E-ISSN : 3025-1311
https://ejournal.cahayailmubangsa.institute/index.php/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 10 No 5 Tahun 2025

ANALISIS SENTIMEN KENDARAAN LISTRIK DI MEDIA SOSIAL X: PERBANDINGAN METODE LSTM DAN *NAÏVE BAYES*

Aziz Rizky Sugiono[1], Nugroho Budhisantosa[2]
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Esa Unggul
Email: arsugiono@student.esaunggul.ac.id

Abstrak

Perkembangan kendaraan listrik sebagai alternatif ramah lingkungan memunculkan berbagai opini publik yang dapat dipantau melalui analisis sentimen di media sosial. Penelitian ini bertujuan untuk membandingkan kinerja metode *Long Short-Term Memory* (LSTM) dan *Naïve Bayes* dalam menganalisis sentimen terhadap kendaraan listrik di platform X (*Twitter*). *Dataset* yang digunakan terdiri dari 42.908 *tweet* yang dikumpulkan menggunakan kata kunci "Mobil Listrik" dan "Motor Listrik" pada periode Oktober 2023 hingga September 2024. Data melalui tahapan *text preprocessing* meliputi *case folding*, *cleansing*, *tokenizing*, normalisasi, penghapusan *stopwords*, dan *stemming*, yang menghasilkan 42.849 *tweet* siap digunakan untuk pelabelan sentimen. Pelabelan data dilakukan menggunakan *Lexicon InSet* dan divalidasi oleh ahli bahasa untuk memastikan akurasi dan konsistensi data. Visualisasi *WordCloud* menunjukkan bahwa kata-kata seperti "listrik", "harga", "baterai", dan "subsidi" sering muncul dalam diskusi positif, menyoroti aspek efisiensi dan dukungan pemerintah yang dianggap penting oleh pengguna. Sebaliknya, sentimen negatif didominasi oleh kata seperti "mahal", "biaya", dan "infrastruktur", yang menunjukkan adanya kekhawatiran terhadap biaya dan ketersediaan fasilitas pendukung. Metode LSTM menunjukkan performa unggul dengan akurasi 86%, jauh lebih tinggi dibandingkan metode Naïve Bayes yang hanya mencapai 45%. Evaluasi menunjukkan bahwa LSTM lebih efektif dalam memahami konteks dan pola teks, sedangkan Naïve Bayes memiliki keterbatasan pada *dataset* yang kompleks. Penelitian ini memberikan kontribusi penting dalam memahami sentimen pengguna sosial X (Twitter) terhadap kendaraan listrik, khususnya di Indonesia. Dengan meningkatnya minat terhadap kendaraan listrik, hasil penelitian ini dapat menjadi acuan strategis untuk produsen otomotif, pembuat kebijakan, dan inovator teknologi. Penelitian ini juga menunjukkan bahwa analisis berbasis *deep learning* dapat digunakan untuk mengatasi tantangan data yang lebih kompleks di masa depan, sehingga memberikan wawasan yang lebih relevan.

Kata Kunci: Analisis Sentimen, *Twitter*, *Dataset*, Kendaraan Listrik, LSTM, *Naïve Bayes*

Abstract

The development of electric vehicles as an environmentally friendly alternative has sparked various public opinions that can be monitored through sentiment analysis on social media. This study aims to compare the performance of Long Short-Term Memory (LSTM) and Naïve Bayes methods in analyzing sentiments

Article History
Received: September 2025
Reviewed: September 2025
Published: September 2025

Plagirism Checker No 234
Prefix DOI : Prefix DOI : 10.8734/Kohesi.v1i2.365
Copyright : Author
Publish by : Kohesi

toward electric vehicles on platform X (Twitter). The dataset consists of 42,908 tweets collected using the keywords "Electric Car" and "Electric Motorcycle" during the period of October 2023 to September 2024. The data underwent text preprocessing stages, including case folding, cleansing, tokenizing, normalization, stopword removal, and stemming, resulting in 42,849 tweets ready for sentiment labeling. The sentiment labeling was conducted using the Lexicon InSet and validated by linguists to ensure the accuracy and consistency of the data. A WordCloud visualization revealed that words such as "electric," "price," "battery," and "subsidy" frequently appeared in positive discussions, highlighting aspects of efficiency and government support deemed important by users. In contrast, negative sentiment was dominated by words like "expensive," "cost," and "infrastructure," indicating concerns about expenses and the availability of supporting facilities. The LSTM model demonstrated superior performance with an accuracy of 86%, significantly outperforming the Naïve Bayes method, which achieved only 45%. The evaluation showed that LSTM is more effective in understanding text context and patterns, whereas Naïve Bayes has limitations with complex datasets. This study provides valuable insights into understanding user sentiments on platform X (Twitter) about electric vehicles, particularly in Indonesia. With the increasing interest in electric vehicles, the results of this research can serve as a strategic reference for automotive manufacturers, policymakers, and technology innovators. Additionally, the study demonstrates that deep learning-based analysis can be used to tackle more complex data challenges in the future, providing more relevant insights.
Keywords *: Sentiment Analysis, Twitter, Dataset, Electric Vehicles, LSTM, Naïve Bayes*

PENDAHULUAN

Kemajuan teknologi dan meningkatnya kesadaran lingkungan mendorong penggunaan kendaraan listrik di berbagai negara, termasuk Indonesia, sebagai alternatif transportasi ramah lingkungan. Kendaraan listrik kini mampu bersaing dengan kendaraan konvensional dalam hal kinerja, seperti kecepatan, jarak tempuh, dan kapasitas muatan. Seiring kemajuan teknologi dan meningkatnya permintaan, kendaraan listrik diprediksi akan semakin populer sebagai opsi transportasi berkelanjutan di masa depan (S. Alfarizi & Fitriani, 2023).

Dalam beberapa tahun terakhir, kendaraan listrik semakin diterima di pasar Indonesia, meskipun industri otomotif mengalami penurunan. Penjualan mobil listrik terus tumbuh signifikan, dengan data Gaikindo mencatat 17.826 unit terjual dari Januari hingga Juli 2024, meningkat 157,30% dibanding periode yang sama tahun sebelumnya. Merek otomotif China mendominasi pasar mobil listrik di Indonesia pada Januari-Juli 2024. Wuling Binguo EV terjual 3.743 unit, disusul Chery Omoda E5 (3.036 unit) dan Wuling Cloud EV (2.097 unit). BYD, merek baru di pasar mobil listrik Indonesia, mencatat penjualan tinggi dengan BYD Seal terjual 1.989 unit hingga Juli 2024, menempati posisi keempat dalam daftar mobil listrik terlaris. (*Penjualan Mobil Listrik Nasional Capai 17.826 Unit Hingga Juli 2024*, n.d.).

Seiring meningkatnya adopsi kendaraan listrik, persepsi pengguna media sosial X (*Twitter*) mengenai penggunaannya tetap beragam. Pemahaman terhadap sentimen dan persepsi pengguna kendaraan listrik adalah krusial, dan analisis sentimen adalah metode yang efektif untuk mengidentifikasi sikap, opini, dan emosi pengguna media sosial X (*Twitter*). Penelitian ini menerapkan metode *Long Short-Term Memory* (LSTM) dan *Naïve Bayes* untuk

menganalisis sentimen terhadap kendaraan listrik di Indonesia. LSTM, sebagai jaringan saraf tiruan yang dirancang untuk mengolah data berurutan, sangat efektif dalam mengenali pola dalam teks, sehingga cocok untuk analisis sentimen. Sementara itu, *Naïve Bayes* merupakan algoritma klasifikasi berbasis probabilitas yang digunakan untuk memprediksi kategori teks. Perbandingan kedua metode ini bertujuan untuk mengklasifikasikan sentimen menjadi positif, negatif, atau netral, serta memberikan wawasan lebih mendalam mengenai persepsi pengguna media sosial X (*Twitter*) terhadap kendaraan listrik di Indonesia.

TINJAUAN PUSTAKA

1. Kendaraan Listrik

Kendaraan listrik adalah alternatif yang ramah lingkungan, beroperasi tanpa bahan bakar fosil dan menggunakan baterai sebagai sumber energi utama untuk menggerakkan motor. Keunggulan kendaraan listrik terletak pada kemampuannya untuk beroperasi tanpa mengeluarkan emisi polutan yang dihasilkan oleh kendaraan bermotor berbahan bakar fosil. Komponen-komponen penting dari kendaraan listrik termasuk motor listrik, baterai, perangkat pengisian, pengontrol kecepatan, dan sistem manajemen energi. Berbagai macam kendaraan listrik yang sudah ada saat ini termasuk motor listrik, mobil listrik, sepeda listrik, bus listrik, dan skuter listrik (Ernawati et al., 2023).

2. X (*Twitter*)

Twitter, sebagai platform media sosial dan *microblogging*, telah meraih popularitas besar sejak didirikan oleh Jack Dorsey pada Maret 2006. Platform ini memungkinkan pengguna mengirim pesan singkat yang disebut "*Tweet*" dengan batasan 280 karakter, yang dirancang untuk mendorong komunikasi ringkas dan cepat. Awalnya dibatasi hanya 140 karakter, *Twitter* memperluas batas tersebut pada 2017 untuk memberikan lebih banyak fleksibilitas dalam berbagi informasi. *Twitter* juga telah berkembang menjadi platform utama dalam penyebaran berita secara *real-time*, terutama dalam situasi darurat atau peristiwa besar dunia. Kemampuan pengguna untuk memberikan pembaruan langsung menjadikannya sumber penting untuk jurnalis, aktivis, dan masyarakat umum.

3. *Twitter* API

API (*Application Programming Interface*) merupakan kumpulan perintah, fungsi, dan protokol yang memainkan peran kunci dalam pengembangan software pada sistem operasi spesifik. API memudahkan pengembang untuk mendekonstruksi aplikasi yang ada untuk pengembangan lebih lanjut atau integrasi dengan software lain, memfasilitasi interaksi antar komponen sistem yang beragam. *Twitter*, sebagai platform besar dengan database yang ekstensif, menyediakan berbagai API yang memungkinkan pengembang mengakses data dan fitur *Twitter* dengan efisien (Agustina et al., 2020). *Twitter* REST API, dibagi menjadi *Twitter* REST yang menyediakan akses ke data dan objek inti *Twitter*, dan *Twitter Search* untuk pencarian objek *Twitter* atau tren yang spesifik. *Twitter Streaming* API, memungkinkan akses data *real-time* dalam volume besar, ideal untuk ekstraksi informasi langsung.

4. *Word2Vec*

Dalam analisis sentimen, fitur yang dapat mengkonversi kata menjadi vektor numerik sangatlah krusial. Teknik yang terkenal dalam hal ini adalah *Word2Vec*, diperkenalkan oleh *Google* pada 2013. *Word2Vec* memanfaatkan dua lapis jaringan saraf untuk memproses data teks, mengolah kumpulan teks (*corpus*) menjadi rangkaian vektor numerik. Terdapat dua pendekatan utama dalam *Word2Vec*: *Continuous Bag of Words* (CBOW) dan *Skip-gram*. CBOW dirancang untuk memprediksi kata target dari konteks kata-kata sekitarnya, sedangkan Skip-gram bekerja dengan cara yang berlawanan, menggunakan kata saat ini untuk memprediksi konteks kata-kata di sekitarnya, dengan tujuan utama memprediksi kata-kata dalam jendela konteks yang berasal dari satu kata (Amin et al., 2020; Santoso et al., 2022).

5. ***Long Short-Term Memory* (LSTM)**

Long Short-Term Memory (LSTM), yang diciptakan oleh Hochreiter dan Schmidhuber, dirancang untuk mengatasi keterbatasan jaringan saraf tiruan tradisional seperti RNN dalam mempertahankan informasi jangka panjang. Dengan kemampuannya yang unik dalam mengatur aliran informasi, LSTM menjadi pilihan yang ideal untuk aplikasi-aplikasi yang memerlukan pemahaman konteks jangka panjang, seperti pemrosesan bahasa alami dan pengenalan suara. LSTM (*Long Short-Term Memory*) dirancang untuk selektif memproses informasi melalui tiga gerbang khusus: gerbang masuk (*input gate*), gerbang keluar (*output gate*), dan gerbang lupa (*forget gate*) (Merdiansah, Wulandari, et al., 2024).

6. ***Naïve Bayes***

Naïve Bayes merupakan metode klasifikasi data yang menggunakan prinsip statistik untuk mengestimasi peluang data dalam kategori tertentu. Metode ini didasarkan pada teori probabilitas yang dikembangkan oleh Thomas Bayes, ilmuwan asal Inggris, yang mengemukakan bahwa kemungkinan kejadian di masa depan dapat diestimasi dari pengalaman sebelumnya. Sebagai algoritma pembelajaran mesin, *Naïve Bayes* bergantung pada perhitungan probabilitas dan statistik dasar, dengan premis independensi antar kelas dalam *dataset* (Afriansyah et al., 2023). *Naïve Bayes* memiliki karakteristik utama, yaitu asumsi kuat (naif) bahwa setiap kondisi atau kejadian bersifat independen satu sama lain.

7. ***Python***

Python merupakan bahasa pemrograman yang interpretatif dan multifungsi, dirancang dengan prinsip yang mengutamakan kemudahan membaca kode. *Python* diciptakan untuk meningkatkan efisiensi waktu *programmer*, mempermudah proses pengembangan, dan menjamin kompatibilitas sistem. *Python* juga sangat fleksibel, dapat digunakan baik untuk pengembangan aplikasi mandiri maupun skrip. Fitur-fitur *Python* mendukung pengembang dalam berbagai aspek, termasuk desain yang mendukung berbagai paradigma pemrograman, kode sumber yang terbuka, kesederhanaan dalam penggunaan, dukungan pustaka kaya, portabilitas lintas platform, kemampuan untuk diperluas dengan modul tambahan, dan skalabilitas untuk proyek-proyek besar (M. R. S. Alfarizi et al., 2023).

8. ***Google Colab***

Google Collaboratory, atau yang lebih dikenal sebagai *Google Colab*, merupakan sebuah platform *Cloud* yang memfasilitasi pengguna dalam menulis, menjalankan, dan berbagi kode *Python* secara langsung melalui peramban web. Platform ini dirancang khusus untuk memenuhi kebutuhan analis data, pengembang, peneliti, serta pendidik yang bergerak di bidang data *science* dan pembelajaran mesin, dengan menyediakan lingkungan komputasi yang tidak hanya fleksibel tetapi juga mudah diakses tanpa mengeluarkan biaya. Selain itu, *Google Colab* mendukung penggunaan *Jupyter Notebook*, sebuah aplikasi web sumber terbuka yang memungkinkan integrasi antara kode, dan visualisasi data, yang dapat dijalankan langsung dari peramban tanpa konfigurasi tambahan (Febrywinata, 2024).

9. **Analisis Sentimen**

Analisis sentimen adalah metode *text mining* yang digunakan untuk mengklasifikasikan ulasan ke dalam kategori tertentu yaitu positif, negatif atau netral. Ketiga kategori ini membantu dalam memahami opini atau emosi seseorang terhadap topik tertentu, seperti produk, layanan, atau isu tertentu. Dalam konteks bisnis, analisis sentimen bisa diterapkan untuk mengevaluasi ulasan produk dan layanan. Analisis ini juga dapat mengidentifikasi emosi seperti kesedihan, kebahagiaan, atau kemarahan. Ini memungkinkan perusahaan untuk memahami persepsi publik terhadap produk, merek, atau individu, dan bagaimana mereka dipersepsikan secara *online* (Farhani & Sutisna, 2024).

10. ***Text mining***

Text mining merupakan proses penggalian informasi dari data yang belum terstruktur. Data ini kemudian diproses dengan menggunakan teknik dan metodologi tertentu untuk menghasilkan informasi yang relevan dan bermanfaat bagi pengguna.

Metode ini sering diterapkan dalam berbagai kasus, termasuk klasifikasi, pengelompokan, ekstraksi informasi, dan pengambilan informasi (Khan et al., 2020). Teknik-teknik umum dalam *text mining* meliputi kategorisasi teks, pengelompokan, ekstraksi konsep atau entitas, pembuatan taksonomi yang detail, analisis sentimen, ringkasan dokumen, dan pemodelan hubungan antar entitas.

11. *Text Preprocessing*

Text Preprocessing adalah proses normalisasi istilah dari kalimat yang bertujuan untuk memastikan kesesuaian antara data latih dan fitur yang diekstraksi dengan kebutuhan analisis. Proses ini esensial untuk menyederhanakan pengolahan data. Dalam pengumpulan data opini dari media sosial seperti *Twitter*, penting untuk diakui bahwa penggunaan bahasa mungkin tidak standar, termasuk kata-kata non-baku, istilah yang tidak ada dalam kamus resmi, atau penggunaan bahasa lokal. Untuk itu, proses *Preprocessing* atau normalisasi menjadi penting untuk mengonversi teks ke dalam bentuk yang lebih alami dan mengeliminasi ekspresi yang tidak tipikal, sehingga dapat mengurangi *noise* pada proses berikutnya.

12. *Lexicon InSet*

Lexicon InSet adalah pendekatan dalam *Machine Learning* yang tidak diawasi, yang kinerjanya sangat tergantung pada kualitas kamus kata yang diaplikasikan. *Lexicon InSet*, atau Indonesian *Sentiment Lexicon*, merupakan salah satu contoh kamus leksikal yang dirancang khusus untuk bahasa Indonesia. Penerapan *Lexicon InSet* ini memudahkan ekstraksi data teks tanpa keharusan untuk menerjemahkannya ke dalam bahasa Inggris, sehingga efisiensi proses analisis dapat ditingkatkan (Artana et al., 2023). Penilaian bobot untuk setiap kata dalam *Lexicon InSet* dilakukan secara manual, berada pada skala -5 hingga 5. Klasifikasi sentimen sebuah kata ditentukan melalui agregasi bobotnya, kata dengan total bobot di atas 0 dianggap memiliki sentimen positif, di bawah 0 dianggap negatif, sedangkan kata dengan total bobot nol dianggap netral (Fathoni et al., 2024).

13. *Natural Language Processing* (NLP)

Natural Language Processing (NLP) merupakan bidang interdisipliner yang berada di persimpangan antara ilmu komputer, kecerdasan buatan, dan linguistik, yang tujuannya adalah untuk memungkinkan komputer memahami, memproses, dan menghasilkan bahasa manusia. Melalui NLP, komputer dapat berkomunikasi dengan manusia menggunakan bahasa alami, baik secara tertulis maupun lisan. Namun, NLP menghadapi tantangan signifikan karena kompleksitas bahasa manusia, termasuk struktur gramatikal yang beragam, makna kata yang ambigu, penggunaan kata yang bervariasi tergantung konteks, dan penggunaan bahasa kiasan. Untuk mengatasi tantangan ini, NLP memanfaatkan metode komputasional yang canggih untuk memproses dan memahami bahasa manusia dengan lebih efisien (Rivaldi et al., 2024).

14. *Deep Learning*

Deep Learning merupakan metode pembelajaran mesin yang maju, yang terinspirasi dari fungsi otak manusia. *Deep Learning* bisa diaplikasikan pada beragam tugas, termasuk pengenalan gambar, pemrosesan bahasa alami, dan robotika. Contoh algoritma *Deep Learning* yang terkenal antara lain *Convolutional Neural Networks* (CNN) untuk pengenalan gambar, *Recurrent Neural Networks* (RNN) untuk pemrosesan bahasa alami, dan *Generative Adversarial Networks* (GAN) untuk penciptaan gambar dan video yang tampak nyata (Sarker, 2021). Dalam *Deep Learning*, setiap lapisan terdiri dari *node* yang berperan sebagai unit pemrosesan data. *Node input* dikombinasikan dengan bobot dan hasilnya dijumlahkan. Perbedaan utama antara *Deep Learning* dan neural network konvensional adalah jumlah *hidden layer* yang lebih banyak dalam *Deep Learning* (Alzubaidi et al., 2021).

15. Evaluasi Klasifikasi

Proses evaluasi dalam klasifikasi seringkali menggunakan *Confusion Matrix*, sebuah tabel yang menyajikan ringkasan dari hasil klasifikasi untuk menilai performa model dalam

mengidentifikasi kategori data dengan tepat. *Confusion Matrix* merupakan teknik yang umum digunakan untuk mengevaluasi performa model klasifikasi melalui perhitungan *Accuracy*, *Recall*, *Precision*, dan *F1-Score*. *Precision* mengukur efektivitas sistem dalam mengidentifikasi entitas yang paling relevan. *Recall* mengukur kemampuan sistem dalam mengambil semua entitas relevan dari kumpulan dokumen. *Accuracy* merupakan perbandingan antara jumlah prediksi yang tepat dengan total jumlah kasus. *F1-Score* merupakan rata-rata harmonik dari *Recall* dan *Precision*, yang berguna untuk menilai model pada *dataset* yang memiliki distribusi kelas yang tidak seimbang (Rolangon et al., 2023).

16. *WordCloud*

WordCloud merupakan visualisasi data teks yang menampilkan frekuensi kata-kata dalam bentuk gambar intuitif. *WordCloud*, yang juga dikenal sebagai *tag Cloud* atau *text Cloud*, efektif untuk menyoroti elemen-elemen penting dalam sejumlah data teks. Selain itu, *WordCloud* berguna untuk membandingkan dua teks berbeda guna mengidentifikasi kata-kata umum antara keduanya. Pembuatan visualisasi ini memerlukan *library* tambahan seperti *numpy*, *pandas*, *matplotlib*, dan image. Teks yang telah diproses diubah menjadi *file* berekstensi **.txt* untuk visualisasi selanjutnya. Setelah mengimpor *library* yang diperlukan, fungsi selanjutnya adalah menetapkan warna latar belakang dan jumlah maksimum kata yang akan ditampilkan dalam visualisasi (Fahrudin et al., 2022).

METODE PENELITIAN

Metode penelitian ini menggunakan pendekatan kuantitatif dengan teknik analisis sentimen terhadap data media sosial X (Twitter). Subjek penelitian adalah pengguna Twitter yang terlibat dalam percakapan mengenai kendaraan listrik, baik melalui unggahan maupun komentar yang bernuansa dukungan, penolakan, maupun netral. Populasi penelitian meliputi seluruh Tweet terkait kendaraan listrik di Indonesia selama periode Oktober 2023 hingga September 2024, sedangkan sampel diambil secara purposive dari data tersebut untuk merepresentasikan persepsi publik secara lebih terukur. Pengumpulan data dilakukan melalui teknik web scraping dengan memanfaatkan Pytrends. Kata kunci yang digunakan antara lain “mobil listrik” dan “motor listrik” dengan rentang waktu penelitian yang sama, yakni Oktober 2023 hingga September 2024. Data yang diperoleh diekspor dalam format CSV/JSON kemudian melalui tahap pra-pemrosesan (pre-processing) yang mencakup case folding, tokenisasi, filtering, dan stemming agar siap diolah lebih lanjut. Setelah itu, dilakukan word embedding dengan metode Word2Vec untuk mengubah kata-kata menjadi representasi numerik sehingga dapat digunakan dalam analisis berbasis algoritma machine learning.

Tahap analisis data melibatkan dua metode utama, yaitu Naïve Bayes dan Long Short-Term Memory (LSTM). Naïve Bayes digunakan untuk klasifikasi sentimen berdasarkan probabilitas frekuensi kata, sedangkan LSTM dimanfaatkan karena kemampuannya dalam memahami konteks bahasa secara mendalam. Kedua model diuji dan dibandingkan efektivitasnya menggunakan metrik evaluasi seperti Accuracy, Recall, Precision, F1-Score, serta Confusion Matrix. Visualisasi tambahan berupa WordCloud juga digunakan untuk memperlihatkan kata-kata yang paling sering muncul dalam diskusi di Twitter. Hasil analisis diinterpretasikan untuk menggambarkan pola persepsi publik terhadap kendaraan listrik di Indonesia, baik dalam bentuk dukungan, kritik, maupun netralitas. Dengan metode ini, penelitian diharapkan mampu memberikan gambaran yang komprehensif mengenai bagaimana kampanye, sosialisasi, dan opini tentang kendaraan listrik terbentuk dan berkembang di media sosial Twitter sepanjang periode penelitian.

HASIL DAN PEMBAHASAN

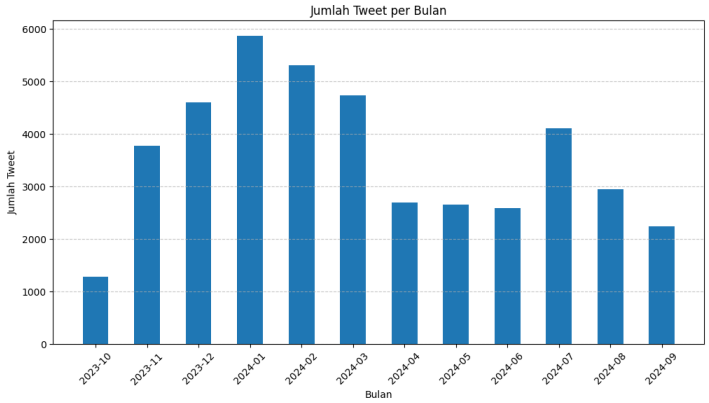
A. Pengumpulan Data

Data dikumpulkan menggunakan kata kunci "mobil listrik" dan "motor listrik" untuk mengidentifikasi *Tweet* relevan, dengan periode pencarian yang ditentukan mulai dari 1

Oktober 2023 sampai 31 Oktober 2023. Proses pengumpulan data dilakukan berulang sesuai dengan periode yaitu hingga akhir September 2024. Hasil dari pengumpulan data tersebut mencapai 42.908 data yang bisa disimpan dalam bentuk CSV atau Excel agar bisa dianalisis ke tahapan selanjutnya. Berikut hasil dari pengumpulan data dari seluruh periode yang telah di tentukan.

	A	B	C	D	E	F	G	H	I	J	K	L
1	full_text											
2	keseruan BRImo USS Yard Sale makin menjadi-jadi hadiahnya motor listrik ALVA Cervo mending ikutan lah. lelang Nike Jordai											
3	Makin rame aja BRImo USS Yard Sale day 2 makin rame makin banyak pilihan sepatu kecenya mana ada hadiah motor listrik											
4	Day 2 Brimo USS Yard Saleee harus siapin Brimo sihh buat hunting sepatu murah hihi. Rameee bgtttt suasana disana apalagi											
5	BRImo USS Yard Sale Day 2 emang boleh serame ini??? Siapin BRImo kalian deh buat hunting sepatu-sepatu ori dengan harg											
6	Rameee bangeett nihh hunting sepatu ori dengan harga murah antusiasme yang tingginya nihh Dapatin motor listrik ALVA Cer											
7	BRIMO USS Yard Sale Day 2 bener2 making menyala! Jgn sampe kelewatan sih sepatu ori dg harga murah. Siapin BRImo kalia											
8	BRImo USS Yard Sale Day 2 makin ramai aja emang harus Siapin BRImo siap hunting sepatu ori dengan harga murah. Eh ada											
9	TIBA TIBA PENGEN BELI MOTOR LISTRIK											
10	BRImo USS Yard Sale makin rame aja nih! Disini bisa nyari sneakers dengan harga murah loh! Terus bisa berkesempatan dap											

Gambar 1. Hasil Scrapping Data Dari Twitter



Gambar 2. Grafik Jumlah Tweet per Bulan Terkait Kendaraan Listrik

Grafik jumlah *Tweet* per bulan menggambarkan bahwa puncak diskusi terjadi pada Januari 2024 dengan total 5.872 *Tweet*. Sebaliknya, jumlah *Tweet* terendah tercatat pada Oktober 2023 dengan hanya 1.280 *Tweet*. Tren ini mengindikasikan adanya periode tertentu dengan peningkatan perhatian pengguna media sosial X (*Twitter*), yang kemungkinan berkaitan dengan kampanye atau isu tertentu terkait kendaraan listrik.

B. *Preprocessing Data*

Data yang terkumpul perlu diproses melalui beberapa tahapan *Preprocessing Data* yaitu *case folding*, *cleansing*, *tokenizing*, *normalize*, *stopword*, dan *stemming*. Proses ini dapat dilakukan dengan menggunakan fitur yang tersedia di *Google Colab*.

1. *Case Folding*

Case folding proses mengubah semua huruf dalam teks menjadi huruf kecil (*lowercase*). Hal ini dilakukan untuk menghilangkan perbedaan antara huruf besar dan huruf kecil yang dapat menyebabkan kata yang sama dianggap berbeda oleh sistem.

```
# === CASE FOLDING ===
def casefolding(text):
    """Mengubah teks menjadi huruf kecil."""
    return text.lower()

df['casefolded_text'] = df['full_text'].apply(casefolding)
```

Gambar 3. Case Folding

Berikut adalah hasil dari tahapan *case folding*.

Tabel 1. Hasil Case Folding

Sebelum	Sesudah
@hi_fiq @mcitraningrum tapi bukannya sebagai pengusaha batubara juga diuntungkan dengan penjualan mobil listrik?	@hi_fiq @mcitraningrum tapi bukannya sebagai pengusaha batubara juga diuntungkan dengan penjualan mobil listrik? karna di indonesia sendiri masih menggunakan pltu.

karna di indonesia sendiri masih menggunakan PLTU.	
TKDN Motor Listrik 40% Mundur ke 2026 Tunggu Industri Tumbuh https://t.co/bGh7InntwU	tkdn motor listrik 40% mundur ke 2026 tunggu industri tumbuh https://t.co/bgh7inntwu
Tambah Lagi Motor Listrik Lokal Kenalin Savart EV Asal Sidoarjo https://t.co/9JAhOumxZZ	tambah lagi motor listrik lokal kenalin savart ev asal sidoarjo https://t.co/9jahoumxzz

2. *Cleansing*

Cleansing adalah proses membersihkan teks dari karakter-karakter yang tidak diinginkan seperti tanda baca, angka, URL, dan karakter khusus. Tujuan dari *cleansing* adalah untuk menghilangkan elemen-elemen yang tidak relevan sehingga teks menjadi lebih bersih dan mudah dianalisis.

```python
# === CLEANSING ===
def remove_html(text):
    """Menghapus tag HTML."""
    return re.sub(r'<.*?>', '', text)

def remove_url(text):
    """Menghapus URL."""
    return re.sub(r'http\S+|www\S+', '', text)

def replace_word_elongation(text):
    """Menghapus pengulangan huruf berlebih."""
    return re.sub(r'(.)\1{2,}', r'\1', text)

def remove_special_characters(text):
    """Menghapus karakter khusus."""
    return re.sub(r'[^a-zA-Z0-9\s]', '', text)

def cleansing_pipeline(text):
    """Pipeline pembersihan teks."""
    text = remove_html(text)  # Menghapus HTML
    text = remove_url(text)  # Menghapus URL
    text = replace_word_elongation(text)  # Menghapus huruf berulang
    text = remove_special_characters(text)  # Menghapus karakter khusus
    return text

df['cleansed_text'] = df['casefolded_text'].apply(cleansing_pipeline)
```

Gambar 4. Cleansing

Berikut adalah hasil dari tahapan *cleansing*.

Tabel 2. Hasil Cleansing

Sebelum	Sesudah
@hi_fiq @mcitraningrum tapi bukannya sebagai pengusaha batubara juga diuntungkan dengan penjualan mobil listrik? karna di indonesia sendiri masih menggunakan pltu.	hifiq mcitraningrum tapi bukannya sebagai pengusaha batubara juga diuntungkan dengan penjualan mobil listrik karna di indonesia sendiri masih menggunakan pltu
tkdn motor listrik 40% mundur ke 2026 tunggu industri tumbuh https://t.co/bgh7inntwu	tkdn motor listrik 40 mundur ke 2026 tunggu industri tumbuh
tambah lagi motor listrik lokal kenalin savart ev asal sidoarjo https://t.co/9jahoumxzz	tambah lagi motor listrik lokal kenalin savart ev asal sidoarjo

3. *Tokenizing*

Tokenization adalah proses memecah teks menjadi unit-unit kecil yang disebut token, yang biasanya berupa kata-kata. *Tokenization* memungkinkan setiap kata dalam teks dianalisis secara individual.

```python
# === TOKENIZATION ===
def word_tokenize_wrapper(text):
    """Melakukan tokenisasi teks."""
    return word_tokenize(text)

df['tokenized_text'] = df['cleansed_text'].apply(word_tokenize_wrapper)
```

Gambar 5. Tokenizing

Berikut adalah hasil dari tahapan *tokenizing*.

Tabel 3. Hasil Tokenizing

Sebelum	Sesudah
hifiq mcitraningrum tapi bukannya sebagai pengusaha batubara juga diuntungkan dengan penjualan mobil listrik karna di indonesia sendiri masih menggunakan pltu	['hifiq', 'mcitraningrum', 'usaha', 'batubara', 'untung', 'jual', 'mobil', 'listrik', 'indonesia', 'pltu']
tkdn motor listrik 40 mundur ke 2026 tunggu industri tumbuh	['tkdn', 'motor', 'listrik', '40', 'mundur', '2026', 'industri', 'tumbuh']
tambah lagi motor listrik lokal kenalin savart ev asal sidoarjo	['motor', 'listrik', 'lokal', 'kenal', 'savart', 'ev', 'sidoarjo']

4. *Normalize*

Normalize adalah proses mengubah teks ke dalam bentuk standar untuk memastikan konsistensi dan keseragaman data. Proses ini penting untuk meningkatkan kualitas analisis teks, seperti pencocokan kata atau pemrosesan lebih lanjut.

```
# === NORMALIZATION ===
normalized_word1 = pd.read_excel('/content/kata_data.xlsx')  # Kamus pertama
normalized_word2 = pd.read_excel('/content/kamus_kata_alay.xlsx')  # Kamus kedua

normalized_word_dict = dict(zip(normalized_word1.iloc[:, 0], normalized_word1.iloc[:, 1]))
normalized_word_dict.update(dict(zip(normalized_word2['slang'], normalized_word2['formal'])))

def normalize_tokens(tokens):
    """Melakukan normalisasi kata."""
    return [normalized_word_dict.get(token, token) for token in tokens]

df['normalized_text'] = df['tokenized_text'].apply(normalize_tokens)
```

Gambar 6. Normalize

Berikut adalah hasil dari tahapan *normalize*.

Tabel 4. Hasil Normalize

Sebelum	Sesudah
['hifiq', 'mcitraningrum', 'usaha', 'batubara', 'untung', 'jual', 'mobil', 'listrik', 'indonesia', 'pltu']	['hifiq', 'mcitraningrum', 'tapi', 'bukannya', 'sebagai', 'pengusaha', 'batubara', 'juga', 'diuntungkan', 'dengan', 'penjualan', 'mobil', 'listrik', 'karena', 'di', 'indonesia', 'sendiri', 'masih', 'menggunakan', 'pltu']
['tkdn', 'motor', 'listrik', '40', 'mundur', '2026', 'industri', 'tumbuh']	['tkdn', 'motor', 'listrik', '40', 'mundur', 'ke', '2026', 'tunggu', 'industri', 'tumbuh']
['motor', 'listrik', 'lokal', 'kenal', 'savart', 'ev', 'sidoarjo']	['tambah', 'lagi', 'motor', 'listrik', 'lokal', 'memperkenalkan', 'savart', 'ev', 'asal', 'sidoarjo']

5. *Stopwords*

Stopword adalah proses menghapus kata-kata umum yang tidak memiliki makna penting dalam analisis, seperti "dan", "atau", "tetapi". *Stopword* biasanya tidak membawa informasi dan dapat mengganggu hasil analisis jika tidak dihilangkan.

```
# === STOPWORD REMOVAL ===
list_stopwords = stopwords.words('indonesian')
additional_stopwords = ['yg','dg','rt','dgn','ny','klo','kalo','amp','biar','bikin','bilang','jga','hahaha','knp','hai',
                        'sobat','nya','loh','eh','eh','token','garasi','disana','kelewatan','meking',
                        'gak','ga','krn','nya','nih','sih','si','tau','tdk','tuh','utk','ya','jd','kn','yaaa','weleh',
                        'jgn','sdh','aja','nyg','hehe','pen','u','ha','nan','loh','&amp','yah','sdgkan','sdg','jir',
                        'hi','hai','hello','helo','hey','emg','sm','pls','please','thank','thanks','mlu','kannn','ken',
                        'brb','btw','b/c','wtf','kak','wrwrwrwr','cod','cmiiw','fyi','gg','ggwp','idk','ikr','lol','ootd',
                        'lmao','oot','pap','otw','wkwk','nahh','yaaaaa','tfl','vc','ygy','ra','lho','loh','gw','elu','lu',...]
list_stopwords.extend(additional_stopwords)
list_stopwords = set(list_stopwords)

def remove_stopwords(tokens):
    """Menghapus stopwords."""
    return [token for token in tokens if token not in list_stopwords]

df['stopword_removed'] = df['normalized_text'].apply(remove_stopwords)
```

Gambar 7. Stopwords

Berikut adalah hasil dari tahapan *stopwords*.

Tabel 4. Hasil Stopwords

Sebelum	Sesudah
['hifiq', 'mcitraningrum', 'tapi', 'bukannya', 'sebagai', 'pengusaha', 'batubara', 'juga', 'diuntungkan', 'dengan', 'penjualan', 'mobil', 'listrik', 'karena', 'di', 'indonesia', 'sendiri', 'masih', 'menggunakan', 'pltu']	['hifiq', 'mcitraningrum', 'pengusaha', 'batubara', 'diuntungkan', 'penjualan', 'mobil', 'listrik', 'indonesia', 'pltu']

['tkdn', 'motor', 'listrik', '40', 'mundur', 'ke', '2026', 'tunggu', 'industri', 'tumbuh']	['tkdn', 'motor', 'listrik', '40', 'mundur', '2026', 'industri', 'tumbuh']
['tambah', 'lagi', 'motor', 'listrik', 'lokal', 'memperkenalkan', 'savart', 'ev', 'asal', 'sidoarjo']	['motor', 'listrik', 'lokal', 'memperkenalkan', 'savart', 'ev', 'sidoarjo']

6. *Stemming*

Stemming adalah proses mengubah kata-kata dalam sebuah teks menjadi bentuk dasarnya (*root word*). Proses ini bertujuan untuk menyederhanakan kata-kata yang memiliki bentuk turunan, seperti "berlari", "pelari", atau "berlarian", menjadi bentuk dasar yaitu "lari".

```
# === STEMMING ===
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_indonesian(tokens):
    """Melakukan stemming untuk Bahasa Indonesia."""
    return [stemmer.stem(token) for token in tokens]

df['stemmed_text'] = df['stopword_removed'].apply(stem_indonesian)
```

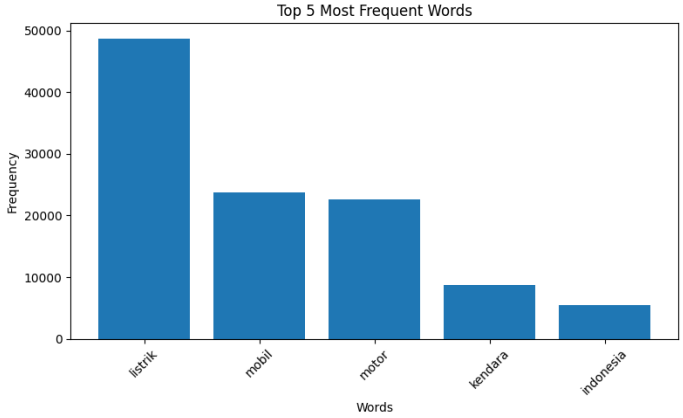
Gambar 8. Stemming

Berikut adalah hasil dari tahapan *stemming*.

Tabel 5. Hasil Stemming

Sebelum	Sesudah
['hifiq', 'mcitraningrum', 'pengusaha', 'batubara', 'diuntungkan', 'penjualan', 'mobil', 'listrik', 'indonesia', 'pltu']	['hifiq', 'mcitraningrum', 'usaha', 'batubara', 'untung', 'jual', 'mobil', 'listrik', 'indonesia', 'pltu']
['tkdn', 'motor', 'listrik', '40', 'mundur', '2026', 'industri', 'tumbuh']	['tkdn', 'motor', 'listrik', '40', 'mundur', '2026', 'industri', 'tumbuh']
['motor', 'listrik', 'lokal', 'memperkenalkan', 'savart', 'ev', 'sidoarjo']	['motor', 'listrik', 'lokal', 'kenal', 'savart', 'ev', 'sidoarjo']

Setelah melalui seluruh tahapan yang ada dalam *Pre-Processing Data*, data tersebut berjumlah menjadi 42.849 data dan akan diolah ke dalam tahapan selanjutnya.



Gambar 9. Kata yang Sering Muncul Setelah Pre-Processing

Gambar 9 merupakan hasil analisis frekuensi kata setelah proses *Pre-Processing*, terdapat lima kata yang paling sering muncul. Hasil ini menunjukkan dominasi tema terkait kendaraan listrik dalam data yang dianalisis.

C. *Word Embedding*

Word Embedding yang digunakan adalah *Word2Vec*, sebuah teknik pembelajaran tanpa pengawasan (*unsupervised learning*) yang menghasilkan vektor numerik untuk kata-kata. *Word2Vec* memiliki dua arsitektur utama: *Continuous Bag of Words* (CBOW) dan *Skip-gram*. CBOW memperkirakan kata target berdasarkan kata-kata konteks, sedangkan *Skip-gram* memperkirakan kata-kata konteks berdasarkan kata target dan lebih baik dalam menangkap hubungan kata yang jarang muncul.

```python
# Fungsi untuk melakukan tokenisasi teks
def tokenize_text(text):
    return word_tokenize(text)

# Menambahkan kolom tokenized_text dengan teks yang telah ditokenisasi
df['tokenized_text'] = df['processed_text'].apply(tokenize_text)

# Melatih model Word2Vec menggunakan teks yang telah ditokenisasi
model = Word2Vec(
    sentences=df['tokenized_text'],
    vector_size=300,   # Ukuran vektor kata
    window=4,          # Jarak maksimum antara kata saat prediksi
    sg=1,              # Gunakan skip-gram (1) daripada CBOW (0)
    workers=4,         # Jumlah thread untuk melatih model
    min_count=3,       # Abaikan kata dengan frekuensi total lebih rendah dari ini
    sample=1e-5,       # Tingkat down-sampling untuk kata yang sering muncul
    alpha=0.03,        # Learning rate awal
    min_alpha=0.0007,  # Learning rate minimum
)

# Menyimpan vektor kata dalam format teks
model.wv.save_word2vec_format('word2vec_skipgram_embeddings.txt', binary=False)

# Menyimpan model Word2Vec untuk LSTM
model.save('word2vec_model.model')

# Menyimpan embedding Word2Vec untuk Naive Bayes dalam format teks
model.wv.save_word2vec_format('word2vec_embeddings.txt', binary=False)

# Fungsi untuk menghitung rata-rata vektor kata dari token
def get_average_word_vectors(tokens, model, vector_size):
    # Memfilter token yang ada dalam kosakata model
    valid_tokens = [token for token in tokens if token in model.wv]
    if not valid_tokens:
        # Jika tidak ada token valid, kembalikan vektor nol
        return np.zeros(vector_size)

    # Mendapatkan vektor untuk token yang valid
    word_vectors = [model.wv[token] for token in valid_tokens]
    # Menghitung rata-rata vektor
    average_vector = np.mean(word_vectors, axis=0)

    return average_vector

# Menambahkan kolom word_embeddings dengan rata-rata vektor kata
vector_size = model.vector_size
df['word_embeddings'] = df['tokenized_text'].apply(
    lambda tokens: get_average_word_vectors(tokens, model, vector_size)
)

# Menyimpan data ke file baru dengan kolom tambahan 'word_embeddings'
df.to_csv('/content/Hasil_Word_Embedding.csv', index=False)

# Example: Accessing vectors for specific words
word_vector_listrik = model.wv['listrik']
print(f"Vector for 'listrik': {word_vector_listrik}")

word_vector_mobil = model.wv['mobil']
print(f"Vector for 'mobil': {word_vector_mobil}")

word_vector_motor = model.wv['motor']
print(f"Vector for 'motor': {word_vector_motor}")
```

Gambar 10. Word Embedding Word2Vec

Berikut adalah tabel hasil *Word Embedding*.

Tabel 6. Hasil Word Embedding Word2Vec

Sebelum	Sesudah
hifiq mcitraningrum usaha batubara untung jual mobil listrik indonesia pltu	[-0.00559129 0.03888996 0.04330458 ...]
tkdn motor listrik 40 mundur 2026 industri tumbuh	[-4.50963294e-03 3.83835062e-02 4.19708453e-02 ...]
motor listrik lokal kenal savart ev sidoarjo	[-0.00423834 0.04065003 0.04232381 ...]

D. Pelabelan Data

pelabelan data akan menggunakan *Lexicon InSet* untuk mengelompokkan data terkait kendaraan listrik ke dalam tiga kategori yaitu positif, negatif dan netral. Selanjutnya, hasil pelabelan data oleh *Lexicon InSet* akan divalidasi oleh ahli bahasa untuk memastikan keakuratannya.

1. Pelabelan Data *Lexicon InSet*

```python
# === LOAD LEXICON INSET ===
# Load lexicon positif dan negatif
lexicon_positive = pd.read_csv('positive.tsv', sep='\t', header=None)[0].tolist()
lexicon_negative = pd.read_csv('negative.tsv', sep='\t', header=None)[0].tolist()

# === ANALISIS SENTIMEN ===
def classify_sentiment(text):
    tokens = word_tokenize(text)  # Tokenize teks
    positive_count = sum(1 for word in tokens if word in lexicon_positive)  # Hitung kata positif
    negative_count = sum(1 for word in tokens if word in lexicon_negative)  # Hitung kata negatif

    # Tentukan sentimen berdasarkan jumlah kata
    if positive_count > negative_count:
        return "positive"
    elif negative_count > positive_count:
        return "negative"
    else:
        return "neutral"

# Terapkan analisis sentimen ke kolom processed_text dan perbarui kolom sentiment
df['sentiment'] = df['processed_text'].apply(classify_sentiment)

# === PINDAHKAN KOLOM SENTIMENT KE BELAKANG ===
# Buat ulang DataFrame dengan kolom sentiment di posisi paling belakang
columns_order = [col for col in df.columns if col != 'sentiment'] + ['sentiment']
df = df[columns_order]

# Simpan hasil analisis sentimen ke file baru
df.to_csv('Hasil_Pelabelan_Data.csv', index=False)

# Tampilkan beberapa hasil
print(df.head(100000))
print("Analisis sentimen selesai. Hasil disimpan di 'Hasil_Pelabelan_Data.csv'.")
```

Gambar 11. Pelabelan Data Lexicon InSet

Berikut adalah tabel klasterisasi dari data kendaraan listrik ke dalam tiga klaster yaitu positif, negatif dan netral.

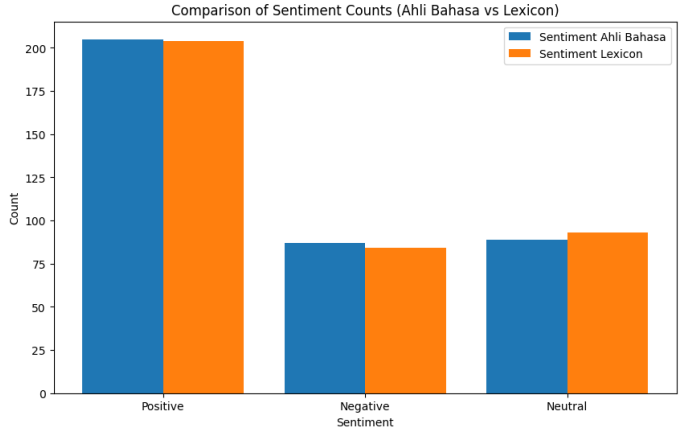
Tabel 7. Hasil Pelabelan Lexicon InSet

Sentimen			Jumlah
Positif	**Negatif**	**Netral**	
22.854	9.536	10.459	42.849
53.3%	22.3%	24.4%	100%

Tabel 8 menyajikan distribusi hasil pelabelan menggunakan *Lexicon InSet*. Dari tabel ini menunjukkan bahwa opini pengguna media sosial X (*Twitter*) terhadap kendaraan listrik cenderung positif berdasarkan analisis otomatis dengan *Lexicon InSet*.

2. Pelabelan Validasi Ahli Bahasa

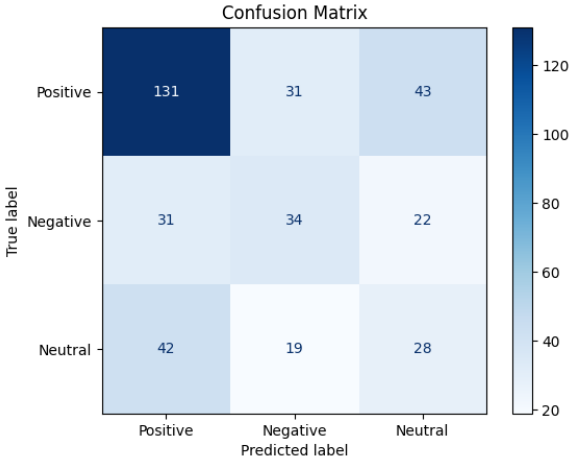
Pada tahap ini, diterapkan teknik *Cochran's Formula*, yang berfungsi untuk menentukan ukuran sampel dalam penelitian survei, terutama saat ukuran populasi sangat besar atau tidak diketahui. Berikut adalah hasil pembagian data sampling sesuai dengan jumlah klaster yang ditentukan.



Gambar 12. Perbandingan Sentimen Lexicon InSet dan Ahli Bahasa

Gambar 12 menampilkan perbandingan jumlah data sentimen yang dilabeli oleh ahli bahasa dan metode *Lexicon InSet*, yang dikelompokkan ke dalam tiga kategori yaitu positif, negatif dan netral. Perbedaan diatas mengindikasikan bahwa metode leksikon cenderung lebih sering mengategorikan data yang ambigu atau kurang jelas sebagai netral, sementara ahli bahasa kemungkinan besar mempertimbangkan lebih banyak aspek kontekstual untuk menentukan apakah data tersebut lebih mendekati positif atau negatif.

Berikut adalah gambar *Confusion Matrix* dari hasil perbandingan keefektifan antara sentimen ahli bahasa dengan *Lexicon InSet*.



Gambar 13. Confusion Matrix Keefektifan Pelabelan Data

Confusion Matrix digunakan untuk menilai kinerja model klasifikasi sentimen dalam membedakan tiga kategori utama yaitu positif, negatif dan netral. Secara keseluruhan, *Confusion Matrix* ini mengungkapkan bahwa meskipun model mampu menghasilkan beberapa prediksi yang akurat, terdapat kesalahan yang cukup signifikan dalam mengklasifikasikan kategori sentimen tertentu. Kesalahan ini dapat memengaruhi tingkat akurasi serta kinerja model secara keseluruhan. Untuk mendapatkan evaluasi yang lebih komprehensif, metrik seperti *Accuracy*, *Precision*, *Recall* dan *F1-Score* dapat digunakan untuk menganalisis kinerja model pada setiap kategori.

Tabel 9. Performa Kefektifan Pelabelan Data

Sentimen	*Precision*	*Recall*	*F1-Score*
Positif	0,64	0,64	0,64
Negatif	0,40	0,39	0,40
Netral	0,30	0,31	0,31
Akurasi	0,51		

Secara keseluruhan, kinerja model terbaik terlihat pada kategori sentimen positif. Namun, kelemahan yang signifikan terlihat dalam mengidentifikasi sentimen negatif dan netral. Ketidakseimbangan ini menunjukkan bahwa model memerlukan peningkatan, baik dari sisi data pelatihan yang lebih bervariasi dan representatif maupun pendekatan pemodelan yang lebih efektif, untuk meningkatkan akurasi dan performa dalam mengklasifikasikan semua kategori sentimen secara menyeluruh.

E. Visualisasi *WordCloud*

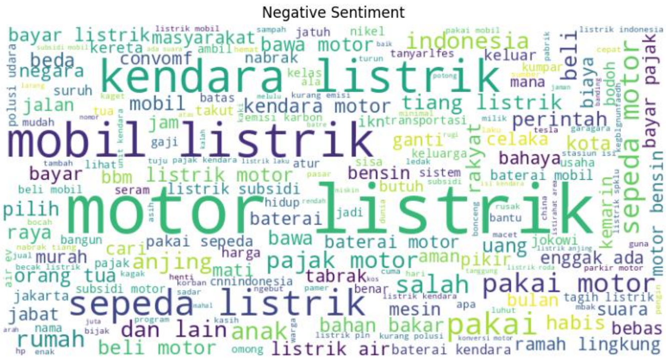
WordCloud adalah visualisasi data teks yang menampilkan kata-kata atau istilah yang paling sering muncul dalam suatu kumpulan teks. Berikut ini adalah visualisasi *WordCloud* untuk sentimen positif, negatif, dan netral.

1. Visualisasi *WordCloud* Positif

Gambar 14. WordCloud Sentimen Positif

WordCloud di atas menunjukkan bahwa kendaraan listrik telah menjadi fokus diskusi pengguna media sosial X (*Twitter*). Selain itu, perhatian pengguna juga terpusat pada aspek harga dan biaya, seperti yang terlihat dari kata-kata "harga", "murah", "biaya" dan "subsidi". Pengguna juga membandingkan kendaraan listrik dengan kendaraan konvensional berbahan bakar fosil, sebagaimana tercermin dari kata-kata seperti "bensin", "konversi" dan "motor". Kesadaran terhadap manfaat lingkungan dan teknologi kendaraan listrik juga semakin meningkat, terlihat dari kata-kata seperti "teknologi", "inovasi", "lingkungan" dan "bersih". Dukungan pemerintah menjadi harapan penting pengguna kendaraan listrik, sebagaimana terlihat dari kata-kata "pemerintah", "program", "infrastruktur" dan "SPKLU". Terakhir, pengguna mulai mengenal berbagai merek kendaraan listrik, yang tercermin dari kemunculan kata-kata seperti "Tesla", "Hyundai" dan "Wuling". Secara keseluruhan, analisis ini mencerminkan peningkatan kesadaran pengguna terhadap kendaraan listrik, baik dari sisi manfaat teknologi, dampak lingkungan, maupun tantangan finansial yang masih perlu diatasi.

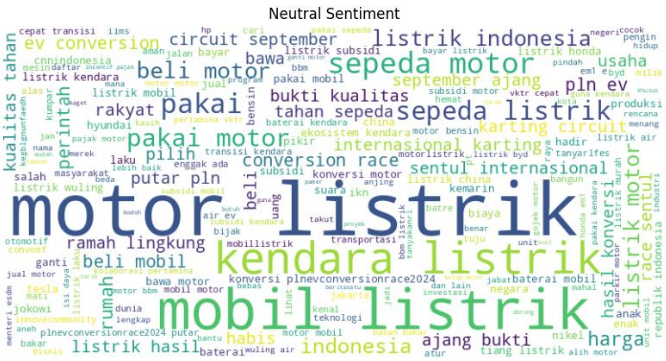
2. **Visualisasi *WordCloud* Negatif**



Gambar 15. WordCloud Sentimen Negatif

WordCloud di atas mencerminkan kekhawatiran pengguna media sosial X (*Twitter*) terhadap aspek-aspek seperti biaya, infrastruktur, dan kinerja. Salah satu isu utama adalah masalah biaya, yang ditunjukkan oleh kata-kata seperti "harga", "mahal", "biaya" dan "pajak." Selain itu, minimnya infrastruktur pengisian daya juga menjadi perhatian, sebagaimana terlihat dari kata-kata "stasiun pengisian", "SPKLU" dan "infrastruktur". Kekhawatiran lainnya terkait dengan kinerja kendaraan listrik, yang terlihat dari kata-kata seperti "jarak tempuh", "baterai" dan "daya". Pengguna juga masih membandingkan kendaraan listrik dengan kendaraan konvensional, sebagaimana terlihat dari kata-kata "bensin", "motor" dan "sepeda". Kekhawatiran terkait keamanan, seperti "bahaya" dan "kebakaran". Selain itu, kata-kata seperti "rumit", "sulit" dan "tidak praktis" mengungkapkan persepsi bahwa kendaraan listrik dianggap terlalu rumit atau tidak sesuai untuk kebutuhan sehari-hari. Kesimpulannya, hambatan utama dalam adopsi kendaraan listrik di Indonesia meliputi harga yang tinggi, infrastruktur yang belum memadai, kurangnya informasi, serta perbandingan negatif dengan kendaraan konvensional. Untuk mengatasi hambatan ini, diperlukan kerja sama antara pemerintah dan industri dalam memberikan edukasi, meningkatkan keterjangkauan kendaraan listrik melalui subsidi atau insentif, serta mempercepat pembangunan infrastruktur pengisian daya, khususnya di daerah terpencil. Kampanye edukasi intensif juga dibutuhkan untuk mengurangi keraguan pengguna terkait keamanan dan kinerja kendaraan listrik.

3. **Visualisasi *WordCloud* Netral**



Gambar 16. WordCloud Sentimen Netral

WordCloud di atas memberikan wawasan menarik terkait topik-topik yang sering dibahas tanpa menunjukkan kecenderungan sentimen yang sangat positif atau negatif. Temuan utama mengindikasikan bahwa pengguna memiliki ketertarikan terhadap teknologi yang mendukung kendaraan listrik, seperti yang tercermin dari kemunculan kata-kata seperti "teknologi", "motor", "baterai" dan "konversi". Selain itu, kata-kata seperti "sepeda", "motor" dan "kendaraan" mencerminkan adanya perbandingan antara kendaraan listrik dan jenis kendaraan lain, baik dari segi performa, harga, maupun

penggunaannya dalam kehidupan sehari-hari. Kata-kata seperti "kualitas", "tahan" dan "kuat" menunjukkan bahwa pengguna masih memerlukan keyakinan lebih besar terhadap daya tahan dan keandalan teknologi kendaraan listrik.

Berdasarkan analisis ini, dapat disimpulkan bahwa pengguna di Indonesia saat ini masih berada dalam tahap awal dalam mengenali dan mengevaluasi kendaraan listrik. Ketertarikan terhadap teknologi serta aspek kepraktisan menunjukkan adanya peluang besar untuk meningkatkan minat pengguna media sosial X (*Twitter*) terhadap kendaraan listrik. Namun, pengguna masih membutuhkan informasi yang lebih mendalam mengenai berbagai aspek, seperti teknologi yang digunakan, performa kendaraan, biaya operasional, serta ketersediaan infrastruktur pendukung. Sentimen netral ini juga menunjukkan bahwa edukasi lebih lanjut sangat diperlukan agar pengguna lebih memahami keunggulan kendaraan listrik dan merasa lebih yakin untuk beralih ke teknologi ramah lingkungan ini.

F. Data Latih dan Data Uji

Data latih berfungsi sebagai kumpulan data yang digunakan untuk melatih model *machine learning*, memungkinkan algoritma mempelajari pola, fitur, dan hubungan antar variabel. Sementara itu, data uji digunakan untuk mengukur kinerja model setelah proses pelatihan selesai, tanpa ikut serta dalam tahap pembelajaran.

Tabel 10. Pembagian Data Latih dan Data Uji

Nama Data	Jumlah	Persentase
Data Latih	34.279	80%
Data Uji	8.570	20%
Total	42.849	100%

Pada tabel di atas, data *Tweet* yang menjadi data latih adalah 34.279 *Tweet* dan data *Tweet* yang menjadi data uji adalah 8.570 *Tweet*.

G. Implementasi Model LSTM

Berikut adalah kode untuk membangun dan melatih model LSTM:

```
# Load pre-trained Word2Vec embeddings
word2vec_model = Word2Vec.load("word2vec_model.model")
embedding_dim = word2vec_model.vector_size  # Dimensi embedding

# Create an embedding matrix
word_index = tokenizer.word_index  # Index dari tokenizer
embedding_matrix = np.zeros((len(word_index) + 1, embedding_dim))  # +1 untuk token pad

# Mengisi embedding_matrix menggunakan Word2Vec
for word, i in word_index.items():
    if word in word2vec_model.wv:  # Kompatibel dengan berbagai versi gensim
        embedding_matrix[i] = word2vec_model.wv[word]

# Build the LSTM model with trainable embeddings
model = Sequential()
model.add(Embedding(
    input_dim=len(word_index) + 1,
    output_dim=embedding_dim,
    embeddings_initializer=Constant(embedding_matrix),  # Inisialisasi embedding
    trainable=True  # Embedding dapat dilatih ulang
))
model.add(SpatialDropout1D(0.2))  # Dropout untuk mencegah overfitting
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))  # LSTM layer
model.add(Dense(64, activation='relu'))  # Dense layer tambahan
model.add(Dropout(0.2))  # Dropout tambahan
model.add(Dense(3, activation='softmax'))  # Output layer dengan 3 kelas (Negatif, Netral, Positif)

# Compile the model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Gambar 17. Kode Pelatihan Model LSTM

Kode ini membangun model LSTM (*Long Short-Term Memory*) dengan menggunakan *pre-trained Word2Vec embeddings* untuk tugas klasifikasi sentimen menjadi tiga kelas yaitu positif, negatif, dan netral.

Untuk menghindari *overfitting* selama pelatihan, *early stopping* digunakan. Model kemudian dilatih menggunakan data latih yang telah diproses sebelumnya:

```
# Early stopping callback
early_stop = EarlyStopping(monitor='val_accuracy', patience=5, restore_best_weights=True, verbose=1)

# Train the model
history = model.fit(
    X_train_pad,
    y_train,
    epochs=50,
    batch_size=32,
    validation_data=(X_test_pad, y_test),
    verbose=2,
    callbacks=[early_stop]
)
```

Gambar 18. Kode Pelatihan Model LSTM Menerapkan Early Stopping

Setelah pelatihan selesai, model diuji menggunakan data uji untuk mengevaluasi performanya. Prediksi dilakukan terhadap data uji, dengan hasil yang disimpan dalam variabel *y_pred_classes*. Evaluasi model dilakukan dengan menghitung akurasi serta menggunakan metrik evaluasi lainnya seperti *Precision*, *Recall* dan *F1-Score*.

H. Implementasi Model *Naïve Bayes*

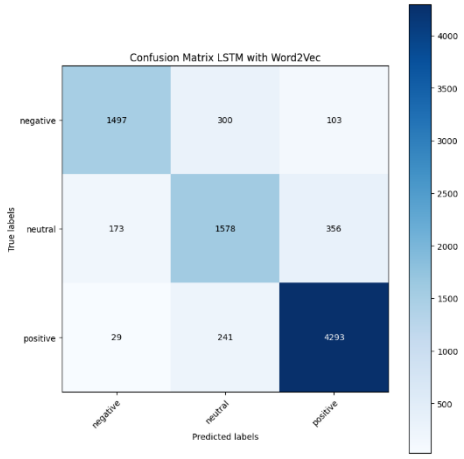
Berikut adalah kode untuk membangun dan melatih model *Naïve Bayes*:

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['full_text'], df['sentiment'], test_size=0.2, random_state=42)

# Load pre-trained Word2Vec embeddings for Naïve Bayes
from gensim.models.keyedvectors import KeyedVectors
word2vec_model = KeyedVectors.load_word2vec_format("word2vec_embeddings.txt", binary=False)

# Fungsi untuk menghitung rata-rata vektor kata dari teks
def get_average_word_vectors(text, model, vector_size):
    tokens = text.split()  # Tokenisasi sederhana dengan split
    valid_tokens = [token for token in tokens if token in model]
    if not valid_tokens:
        return np.zeros(vector_size)
    word_vectors = [model[token] for token in valid_tokens]
    return np.mean(word_vectors, axis=0)

# Dimensi vektor Word2Vec
vector_size = word2vec_model.vector_size

# Mengonversi teks menjadi rata-rata embedding Word2Vec
X_train_vectors = np.array([get_average_word_vectors(text, word2vec_model, vector_size) for text in X_train])
X_test_vectors = np.array([get_average_word_vectors(text, word2vec_model, vector_size) for text in X_test])

# Naive Bayes tidak mendukung nilai negatif, jadi kita gunakan Min-Max Scaling
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train_vectors)
X_test_scaled = scaler.transform(X_test_vectors)

# Create a Multinomial Naive Bayes classifier
nb = MultinomialNB()

# Train the classifier
nb.fit(X_train_scaled, y_train)

# Make predictions on the testing data
y_pred = nb.predict(X_test_scaled)

# Evaluate the model
print("Naive Bayes with Word2Vec Accuracy:", accuracy_score(y_test, y_pred))
print("Naive Bayes with Word2Vec Classification Report:")
print(classification_report(y_test, y_pred))
```

Gambar 19. Kode Pelatihan Model Naïve Bayes

Kode ini mengimplementasikan model *Multinomial Naïve Bayes* untuk analisis sentimen menggunakan *Word2Vec embeddings*. Setelah model dilatih, hasil prediksi diuji menggunakan data uji yang telah diproses sebelumnya. Evaluasi dilakukan dengan menghitung akurasi serta menggunakan metrik *Precision*, *Recall* dan *F1-Score* untuk mengevaluasi kinerja klasifikasi model.

I. Evaluasi Model



Gambar 20. Confusion Matrix Model LSTM

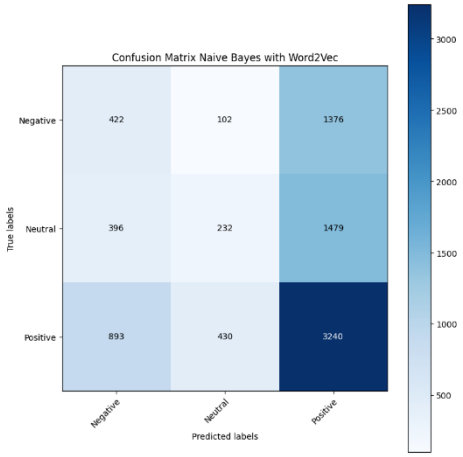
Dari gambar *Confusion Matrix* tersebut kita dapat menghitung akurasi dan metrik lainnya dengan menggunakan nilai-nilai dari *Confusion Matrix* tersebut.

Tabel 11. Hasil Classification Report dari Confusion Matrix Model LSTM

Kelas	Performa		
	Precision	*Recall*	*F1-Score*
Negatif	0.88	0.79	0.83
Positif	0.90	0.94	0.92
Netral	0.74	0.75	0.75
Akurasi	0.86		

Hasil evaluasi ini menunjukkan bahwa model memiliki performa yang baik dalam mengidentifikasi sentimen positif, tetapi masih memerlukan perbaikan dalam mengenali sentimen negatif dan netral, terutama dalam meningkatkan ketepatan klasifikasi pada kategori netral yang masih sering salah dikategorikan sebagai sentimen negatif atau positif.

Selain hasil dari *Confusion Matrix* dari model LSTM, dalam penelitian ini juga terdapat hasil visualisasi *Confusion Matrix* dari penggunaan model *Naïve Bayes*.
Berikut adalah hasil dari *Confusion Matrix* nya.



Gambar 21. Confusion Matrix Model Naïve Bayes

Dari gambar *Confusion Matrix* tersebut kita dapat menghitung akurasi dan metrik lainnya dengan menggunakan nilai-nilai dari *Confusion Matrix* tersebut.

Tabel 12. Hasil Classification Report dari Confusion Matrix Model Naïve Bayes

Kelas	Performa		
	Precision	*Recall*	*F1-Score*
Negatif	0.25	0.22	0.23
Positif	0.53	0.71	0.61
Netral	0.30	0.11	0.16
Akurasi	0.45		

Laporan klasifikasi di atas menunjukkan model lebih sering mengategorikan data netral sebagai sentimen negatif atau positif, menyebabkan tingkat kesalahan klasifikasi yang cukup tinggi pada kategori ini.

J. Interpretasi Hasil

Dalam penelitian ini, data dikumpulkan dari media sosial X (Twitter) selama periode Oktober 2023 hingga September 2024, dengan total 42.908 data *Tweet*. Setelah melalui tahapan *Preprocessing Data*, jumlah menjadi 42.849 data *Tweet*. Pelabelan menggunakan metode *Lexicon InSet*, dengan distribusi sentimen yang terdiri dari 22.854 *Tweet* (53,3%) positif, 9.536 *Tweet* (22,3%) negatif, dan 10.459 *Tweet* (24,4%) netral. Dilakukan juga validasi oleh ahli bahasa menggunakan sampel acak sebanyak 381 *Tweet*. Validasi ini penting karena *Lexicon InSet* sering mengalami kesalahan dalam menangkap makna sentimen yang lebih kompleks yang sering muncul di media sosial. Hasil validasi oleh ahli bahasa terdiri

dari 205 *Tweet* (53,8%) sentimen positif, 87 *Tweet* (22,8%) sentimen negatif, dan 89 *Tweet* (23,4%) sentimen netral.

Hasil visualisasi *WordCloud* menunjukkan bahwa kata-kata seperti "kendaraan listrik", "baterai", dan "harga" sering muncul dalam berbagai kategori sentimen. Hal ini mengindikasikan bahwa biaya kendaraan, daya tahan baterai, serta ketersediaan infrastruktur pengisian daya menjadi isu utama dalam percakapan pengguna media sosial X (*Twitter*) mengenai kendaraan listrik di media sosial. Pola kemunculan kata-kata ini menunjukkan bahwa diskusi pengguna media sosial X (*Twitter*) cenderung berfokus pada aspek ekonomi dan daya tahan teknologi, yang sering menjadi faktor utama dalam keputusan pembelian kendaraan listrik.

Penelitian ini juga menunjukkan bahwa LSTM lebih unggul dibandingkan *Naïve Bayes* dalam analisis sentimen kendaraan listrik di media sosial X (*Twitter*). Dari hasil pengujian, LSTM mencapai akurasi 86%, jauh lebih tinggi dibandingkan *Naïve Bayes* yang hanya memperoleh 45%, menandakan bahwa metode berbasis *Deep Learning* lebih efektif dalam menangani data *Tweet* yang memiliki struktur bahasa bervariasi dibandingkan metode konvensional seperti *Naïve Bayes*. Rendahnya akurasi Naïve Bayes disebabkan oleh ketidaksesuaian penggunaan *Word2Vec*, yang menghasilkan representasi kata dalam bentuk vektor berkelanjutan, sedangkan *Naïve Bayes* lebih cocok untuk fitur berbasis frekuensi diskrit.

Mayoritas sentimen yang diklasifikasikan bersifat positif, menunjukkan penerimaan terhadap teknologi kendaraan listrik, meskipun harga dan infrastruktur masih menjadi tantangan utama. Sentimen positif meningkat setelah adanya subsidi atau promosi, sementara sentimen negatif lebih dominan ketika harga tinggi dan keterbatasan infrastruktur menjadi sorotan. Hal ini menunjukkan bahwa dukungan terhadap kendaraan listrik dipengaruhi oleh insentif ekonomi dan kesiapan ekosistemnya. Selain itu, sentimen netral menunjukkan bahwa banyak pengguna Twitter membahas kendaraan listrik tanpa menyatakan sikap yang jelas, kemungkinan karena masih mencari informasi atau sekadar menyampaikan opini tanpa emosi. Dengan demikian, meskipun tren sentimen positif meningkat, adopsi kendaraan listrik masih bergantung pada faktor ekonomi dan infrastruktur, yang memerlukan perhatian lebih lanjut.

KESIMPULAN

Berdasarkan hasil pengumpulan data selama satu tahun yang berlangsung dari Oktober 2023 hingga September 2024, terkumpul sebanyak 42.908 data *Tweet*. Setelah melewati proses *Preprocessing Data*, jumlah data yang terkumpul sebanyak 42.849 data *Tweet*.

Pelabelan data yang dilakukan menggunakan metode *Lexicon InSet* menunjukkan bahwa 22.854 *Tweet* (53,3%) bersentimen positif, 9.536 *Tweet* (22,3%) negatif, dan 10.459 *Tweet* (24,4%) netral. Namun, karena metode berbasis leksikon memiliki keterbatasan dalam menangkap konteks bahasa alami, dilakukan validasi oleh ahli bahasa menggunakan sampel acak sebanyak 381 *Tweet*, yang terdiri dari 205 *Tweet* (53,8%) bersentimen positif, 87 *Tweet* (22,8%) negatif, dan 89 *Tweet* (23,4%) netral. Hasil ini menunjukkan bahwa model berbasis leksikon masih memiliki keterbatasan dalam menangkap makna bahasa yang lebih kompleks dan memerlukan validasi tambahan untuk meningkatkan keakuratannya.

Hasil evaluasi menunjukkan bahwa LSTM memiliki performa yang jauh lebih baik dibandingkan *Naïve Bayes* dalam mengklasifikasikan sentimen kendaraan listrik. Dari hasil pengujian, LSTM mencapai akurasi 86%, jauh lebih tinggi dibandingkan *Naïve Bayes* yang hanya memperoleh 45%, mengindikasikan bahwa model berbasis *Deep Learning* lebih efektif dalam menangani bahasa alami yang kompleks dan beragam. Kelemahan utama *Naïve Bayes* adalah ketidaksesuaian dengan penggunaan *Word2Vec* sebagai metode *Word Embedding*. *Word2Vec* menghasilkan representasi kata dalam bentuk vektor berkelanjutan di ruang dimensi tinggi, yang bertujuan untuk menangkap hubungan semantik antar kata. Ketidaksesuaian ini membuat *Word2Vec* kurang optimal jika digunakan bersama *Naïve Bayes*, sehingga meningkatkan tingkat kesalahan prediksi dan menurunkan akurasi model.

Visualisasi menggunakan *WordCloud* menunjukkan bahwa diskusi pengguna media sosial X (*Twitter*) tentang kendaraan listrik didominasi oleh kata-kata seperti "kendaraan listrik", "baterai" dan "harga". Hal ini mengindikasikan bahwa biaya kendaraan, daya tahan baterai, dan infrastruktur pengisian daya masih menjadi faktor utama yang diperbincangkan dalam adopsi kendaraan listrik. Selain itu, analisis pola sentimen menunjukkan bahwa sentimen positif meningkat setelah adanya kebijakan subsidi atau promosi, sementara sentimen negatif lebih sering muncul ketika isu harga tinggi dan keterbatasan infrastruktur menjadi perbincangan utama.

Berdasarkan hasil *Confusion Matrix*, LSTM lebih unggul dalam menangkap pola sentimen dibandingkan *Naïve Bayes*, terutama dalam mengklasifikasikan sentimen negatif dan netral dengan lebih akurat. Hal ini menunjukkan bahwa model berbasis *Deep Learning* lebih mampu memahami konteks bahasa secara menyeluruh, sehingga lebih direkomendasikan untuk analisis sentimen kendaraan listrik dibandingkan metode konvensional seperti *Naïve Bayes*.

Secara keseluruhan, penelitian ini memberikan wawasan mendalam mengenai persepsi pengguna media sosial X (*Twitter*) terhadap kendaraan listrik di Indonesia, yang dapat menjadi referensi bagi produsen otomotif, pembuat kebijakan, serta akademisi dalam memahami tren opini pengguna media sosial X (*Twitter*). Selain itu, hasil penelitian ini menunjukkan bahwa pemilihan metode analisis sentimen sangat berpengaruh terhadap akurasi klasifikasi, dengan LSTM terbukti lebih unggul dalam menangani kompleksitas bahasa dibandingkan metode konvensional seperti *Naïve Bayes*. Dengan meningkatnya sentimen positif terhadap kendaraan listrik, diharapkan industri dan pemerintah dapat memanfaatkan tren ini dengan menyediakan insentif yang lebih luas serta mempercepat pengembangan infrastruktur pendukung, guna mempercepat adopsi kendaraan listrik di Indonesia.

DAFTAR PUSTAKA

Afriansyah, M., Saputra, J., Sa'adati, Y., & Ardhana, V. Y. P. (2023). Optimasi Algoritma Naïve Bayes Untuk Klasifikasi Buah Apel Berdasarkan Fitur Warna RGB. *Bulletin of Computer Science Research*, *3*(3), 242–249. https://doi.org/10.47065/bulletincsr.v3i3.251

Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning. *Karimah Tauhid*, *2*(1), 1–6. https://doi.org/10.30997/karimahtauhid.v2i1.7518

Anwar, K. (2022). Analisa sentimen Pengguna Instagram Di Indonesia Pada Review Smartphone Menggunakan Naïve Bayes. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, *2*(4), 148–155. https://doi.org/10.30865/klik.v2i4.315

Ardiyanti, D., Kurniawan, F., Raokter, U., & Wikansari, R. (2023). Analisis penjualan mobil listrik di Indonesia dalam rentang waktu 2020-2023. *ECOMA: Journal of Economics and Management*, *1*(3), 114–122. https://doi.org/10.55681/ecoma.v1i3.26

Aryanti, P. G., & Santoso, I. (2023). Analisis Sentimen Pada Twitter Terhadap Mobil Listrik Menggunakan Algoritma Naive Bayes. *IKRA-ITH Informatika: Jurnal Komputer Dan Informatika*, *7*(2), 133–137. https://journals.upi-yai.ac.id/index.php/ikraith-informatika/article/view/2821

Azrul, A., Purnamasari, A. I., & Ali, I. (2024). Analisis Sentimen Pengguna Twitter Terhadap Perkembangan Artificial Intelligence Dengan Penerapan Algoritma Long Short-Term Memory (Lstm). *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(1), 413–421. https://doi.org/10.36040/jati.v8i1.8416

Farhani, A., & Sutisna. (2024). Analisis Sentimen Terhadap Kendaraan Listrik di Indonesia Menggunakan Metode Klasifikasi Naïve Bayes. *Jurnal Indonesia: Manajemen Informatika Dan Komunikasi*, *5*(3), 2680–2690. https://doi.org/10.35870/jimik.v5i3.983

Firdaus, R., & Mukhtar, H. (2023). Prediksi Indeks Harga Produsen Pertanian Karet di Indonesia Menggunakan Metode LSTM. *Jurnal Fasilkom*, *13*(01), 1–6. https://doi.org/10.37859/jf.v13i01.4851

Fuad, M., Wahlil, M. A., Hazriani, H., & Yuyun, Y. (2023). Implementasi Klasifikasi Naive Bayes Dalam Memprediksi Lama Studi Mahasiswa. *Prosiding SISFOTEK*, *7*(1), 209–312. https://seminar.iaii.or.id/index.php/SISFOTEK/article/view/392

Gifari, O. I., Adha, M., Freddy, F., & Durrand, F. F. S. (2022). Film Review Sentiment Analysis Using TF-IDF and Support Vector Machine. *Journal of Information Technology*, *2*(1), 36–40. https://doi.org/10.46229/jifotech.v2i1.330

Hasanah, S. U., Sibaroni, Y., & Prasetyowati, S. S. (2024). Word2Vec Optimization on Bi-LSTM in Electric Car Sentiment Classification. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *8*(1), 124–132. https://doi.org/10.30865/mib.v8i1.7200

Irawati, D. Y., Bellanov, A., & Damayanti, F. A. (2024). Sentiment Analysis about Electric Motorbikes in Indonesia Using Twitter Data. *Spektrum Industri*, *22*(1), 25–35. https://doi.org/10.12928/si.v22i1.158

Karimah, A., Dwilestari, G., & Mulyawan, M. (2024). Analisis Sentimen Komentar Video Mobil Listrik Di Platform Youtube Dengan Metode Naive Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(1), 767–773. https://doi.org/10.36040/jati.v8i1.8373

Merdiansah, R., Siska, S., & Ridha, A. A. (2024). Analisis sentimen pengguna X Indonesia terkait kendaraan listrik menggunakan IndoBERT. *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)*, *7*(1), 221–228. https://doi.org/10.55338/jikomsi.v7i1.2895

Mukti, A., Hadiyanti, A. D., Nurlaela, A., & Panjaitan, J. (2023). Sistem Analisa Sentiment Bakal Calon Presiden 2024 Menggunakan Metode NLP Berbasis Web. *SOSCIED*, *6*(1), 128–140. https://doi.org/10.32531/jsoscied.v6i1.621

Nanjundeswaraswamy, T. S., & Divakar, S. (2021). DETERMINATION OF SAMPLE SIZE AND SAMPLING METHODS IN APPLIED RESEARCH. *Proceedings on Engineering Sciences*, *3*(1), 25–32. https://doi.org/10.24874/pes03.01.003

Penjualan Mobil Listrik Nasional Capai 17.826 Unit Hingga Juli 2024. (n.d.). Retrieved October 31, 2024, from https://industri.kontan.co.id/news/penjualan-mobil-listrik-nasional-capai-17826-unit-hingga-juli-2024

Penjualan Mobil Listrik Nasional Naik, Segmennya Mencapai Empat Persen – GAIKINDO. (n.d.). Retrieved October 31, 2024, from https://www.gaikindo.or.id/penjualan-mobil-listrik-nasional-naik-segmennya-mencapai-empat-persen/

Pratama, Y., Murdiansyah, D. T., & Lhaksmana, K. M. (2023). Analisis Sentimen Kendaraan Listrik Pada Media Sosial Twitter Menggunakan Algoritma Logistic Regression dan Principal Component Analysis. *Jurnal Media Informatika Budidarma*, *7*(1), 529–535. https://doi.org/10.30865/mib.v7i1.5575

Raup, A., Ridwan, W., Khoeriyah, Y., Supiana, S., & Zaqiah, Q. Y. (2022). Deep Learning dan Penerapannya dalam Pembelajaran. *JIIP-Jurnal Ilmiah Ilmu Pendidikan*, *5*(9), 3258–3267. https://doi.org/10.54371/jiip.v5i9.805

Rolangon, A., Weku, A., & Sandag, G. A. (2023). Perbandingan Algoritma LSTM Untuk Analisis Sentimen Pengguna Twitter Terhadap Layanan Rumah Sakit Saat Pandemi Covid-19. *TeIKa*, *13*(01), 31–40. https://doi.org/10.36342/teika.v13i01.3063

Setiawan, E. (2023, February 7). *Word2Vec: Embedding Teks Berbahasa Indonesia | by Eko Setiawan | Medium*. https://medium.com/@ekomasterwan993/embedding-teks-berbahasa-indonesia-dengan-gensim-5efcc36f2216

Sudjoko, C. (2021). Strategi pemanfaatan kendaraan listrik berkelanjutan sebagai solusi untuk mengurangi emisi karbon. *Jurnal Paradigma: Jurnal Multidisipliner Mahasiswa Pascasarjana Indonesia*, *2*(2), 54–68. https://doi.org/10.22146/jpmmpi.v2i2.70354

Talaei Khoei, T., Ould Slimane, H., & Kaabouch, N. (2023). Deep learning: Systematic review, models, challenges, and research directions. *Neural Computing and Applications*, *35*(31), 23103–23124. https://doi.org/10.1007/s00521-023-08957-4

Wuling Kuasai Penjualan Mobil Listrik di Indonesia – GAIKINDO. (n.d.). Retrieved October 31, 2024, from https://www.gaikindo.or.id/wuling-kuasai-penjualan-mobil-listrik-di-indonesia/