PERBANDINGAN ALGORITMA NAIVE BAYES DAN DECISION TREE PADA ANALISIS SENTIMEN MEDIA SOSIAL TERHADAP KENDARAAN LISTRIK INDONESIA MENGGUNAKAN METODE TF-IDF

Yanuar Anggara Firdaus [1], Muhammad Naufal Annabil [2],
Excel Pratama Putra Adi Wibawa [3], Marcelinus Yosep Teguh Sulistyono [4]
[1,2,3,4] Sistem Informasi, Universitas Dian Nuswantoro
teguh.sulistyono@dsn.dinus.ac.id

ABSTRAK

Perkembangan kendaraan listrik sebagai solusi atas krisis energi dan lingkungan memicu perhatian luas di berbagai negara, termasuk di Indonesia. Namun, opini masyarakat terhadap teknologi ini masih menimbulkan perselisihan, sehingga analisis sentimen menjadi penting untuk memahami opini publik. Tujuan dari penelitian guna membandingkan performa algoritma *Naïve Bayes* dan *Decision Tree* dalam mengklasifikasikan sentimen masyarakat terhadap kendaraan listrik dari media sosial Twitter dengan tools Google Colab. Data dikumpulkan melalui proses crawling, lalu dilakukan preprocessing teks serta pembobotan fitur dengan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Selanjutnya, data diklasifikasikan menggunakan kedua algoritma dan dievaluasi dengan metrik akurasi, presisi, *recall*, dan f1-score. Hasil penelitian menunjukkan bahwa algoritma *Decision Tree* menghasilkan performa yang lebih tinggi dibandingkan *Naïve Bayes*, dengan akurasi mencapai 98,85% dibandingkan 81,54%. Hasil performa *Decision Tree* lebih tinggi dari *Naïve Bayes* karena mampu mengekstraksi antar kata secara lebih baik dan memanfaatkan fitur TF-IDF secara optimal. Temuan tersebut menampilkan *Decision Tree* lebih optimal saat mengklasifikasikan data sentimen terhadap kendaraan listrik di Indonesia.

Keyword : *Analisis Sentimen, Kendaraan Listrik, Media Sosial, Naïve Bayes, Decision Tree, TF-IDF.*

1. PENDAHULUAN

Dalam beberapa tahun terakhir, industri otomotif global telah mengalihkan fokusnya ke pengembangan kendaraan listrik sebagai solusi terhadap permasalahan lingkungan dan keterbatasan sumber daya energi. Kendaraan listrik menawarkan beberapa keunggulan daripada kendaraan berbahan bakar bensin, termasuk ramah terhadap lingkungan, tingkat efisiensi energi yang baik, serta biaya operasional yang lebih murah. Sementara itu, bahan bakar fosil, yang merupakan sumber bahan bakar utama untuk mobil konvensional, menjadi semakin langka dan diperkirakan akan semakin langka di masa depan, sehingga menekankan perlunya transisi ke kendaraan listrik [1].

Namun demikian, meskipun kendaraan listrik menawarkan banyak keunggulan, penerimaan masyarakat terhadap teknologi ini masih menimbulkan perselisihan. Beberapa isu yang sering diperdebatkan antara lain harga kendaraan listrik yang dianggap terlalu mahal, keterbatasan infrastruktur pengisian daya di luar kota besar, serta kekhawatiran terhadap umur dan biaya penggantian baterai. Analisis sentimen terhadap opini publik menjadi penting untuk memahami bagaimana masyarakat merespons kebijakan dan inovasi yang berkaitan dengan kendaraan listrik. Media sosial seperti *Twitter* menjadi penghasil data yang beragam dan real-time guna menganalisis opini masyarakat berkaitan topik ini. Dataset yang dikumpulkan cukup besar, yaitu sebanyak 2.552 tweet serta mencakup opini yang beragam, mulai dari dukungan, kritik, hingga kekhawatiran masyarakat terhadap kendaraan listrik. Opini publik dapat mencerminkan berbagai faktor, termasuk ketersediaan infrastruktur pengisian daya, harga mobil yang tinggi, dan pemahaman tentang manfaat jangka panjang kendaraan listrik [2].

Di Indonesia, adopsi kendaraan listrik juga menjadi bagian dari strategi nasional guna meminimalisir gas karbon serta meminimalisir ketergantungan terhadap bahan bakar bensin. Pemerintah telah menerapkan sejumlah kebijakan pendukung seperti insentif pajak dan subsidi pembelian, subsidi langsung pembelian kendaraan listrik yang diatur dalam Peraturan Menteri Keuangan, serta percepatan pembangunan infrastruktur pengisian daya (*charging station*) yang dikoordinasikan oleh PLN dan pemerintah daerah. Namun, penerapan kebijakan ini tidak menjamin keberhasilan tanpa adanya dukungan dari masyarakat secara luas. Persepsi publik menjadi faktor penting dalam menentukan apakah kebijakan yang telah diterapkan benar-benar efektif dan tepat sasaran [3].

Dalam konteks yang lebih luas, analisis sentimen dari media sosial tidak hanya membantu dalam mengevaluasi opini publik, tetapi juga berfungsi sebagai alat strategis bagi pembuat kebijakan, pelaku industri, dan peneliti untuk mengetahui permasalahan atau hambatan yang dihadapi dalam pengembangan kendaraan listrik. Berbagai pendekatan telah digunakan dalam klasifikasi sentimen, salah satunya dengan memanfaatkan algoritma machine learning,

contohnya *Naïve Bayes* dan *Decision Tree* [4]. *Naïve Bayes* dipilih karena efisien dan cocok untuk teks pendek seperti tweet, sementara *Decision Tree* dipilih karena mampu menangkap pola kompleks dan interaksi antar kata. Keduanya digunakan dalam studi analisis sentimen dan relevan untuk dibandingkan dari sisi akurasi, presisi, recall, dan F1-Score.

Oleh karena itu, riset akan dilakukan guna membandingkan performa algoritma *Naïve Bayes* serta *Decision Tree* terhadap klasifikasi sentimen masyarakat terhadap kendaraan listrik di Indonesia. Penggunaan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) sebagai teknik ekstraksi fitur teks, diharapkan bahwa riset ini akan membantu menentukan cara paling efektif untuk memahami opini masyarakat, serta memfasilitasi proses pengambilan keputusan semakin tepat dalam pengembangan kebijakan kendaraan listrik di Indonesia [5].

2. TINJAUAN PUSTAKA

2.1. Kajian Penelitian Terdahulu

Singgalen menggunakan algoritma *Naïve Bayes Classifier* (NBC) dan *Decision Tree* (DT) untuk mengklasifikasikan ulasan wisatawan dari situs web *Tripadvisor*. *Decision Tree* dengan *SMOTE outperformed Naïve Bayes* dalam klasifikasi, dengan akurasi 98,27%, presisi 98,83%, *recall* 97,71%, dan *F-measure* 98,26%. *Naïve Bayes* dengan *SMOTE* memperoleh akurasi 99,81%, namun nilai AUC-nya hanya 0,5, menunjukkan kekurangan dalam pembedaan kelas [6].

Sementara itu, Wijaya dan Utami mengkaji pemanfaatan algoritma *Decision Tree* untuk memahami persepsi masyarakat terhadap kendaraan listrik. Hasilnya menunjukkan bahwa algoritma ini unggul dalam mengolah data yang memiliki pola yang kompleks dan tidak linear [7]. Penelitian serupa dilakukan oleh Pramana et al., yang membandingkan kinerja Naïve Bayes dan Decision Tree menggunakan teknik ekstraksi fitur TF-IDF. Mereka menyimpulkan bahwa hasil klasifikasi sangat dipengaruhi oleh proses pengolahan data awal, khususnya dalam pemilihan metode representasi teks [8].

2.2. *Twitter*

Twitter merupakan platform media sosial yang tren pada mayoritas pengguna internet karena kemudahan penggunaannya dan kepraktisannya. Selain itu, pengguna dapat dengan bebas mengekspresikan pendapat dan pandangan mereka di platform ini [9].

Dalam konteks penelitian ini, *Twitter* dipilih sebagai sumber data karena banyaknya percakapan publik yang berkaitan dengan kendaraan listrik, baik dalam bentuk dukungan, kritik, maupun aspirasi. Kata kunci tertentu dapat digunakan untuk menelusuri kata yang relevan dan kemudian dianalisis menggunakan pendekatan klasifikasi sentimen.

2.3. TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan metode populer dalam pengolahan data teks yang Digunakan untuk menentukan signifikansi suatu istilah dalam sebuah dokumen. Teknik tersebut memprioritaskan istilah-istilah yang sering muncul dalam satu sumber tetapi jarang ditemui dalam dokumen lain, sehingga mampu mengidentifikasi kata-kata yang benar-benar representatif terhadap konteks [10].

Menurut Fitri dan Yuliana, penggunaan TF-IDF terbukti mampu meningkatkan kinerja algoritma klasifikasi, seperti SVM dan *Naïve Bayes*, dibandingkan pendekatan *Bag of Words*. Teknik ini juga membantu mengurangi gangguan dari kata-kata publik yang tidak ada sangkut paut terhadap makna utama [11].

2.4. Naive Bayes

Naïve Bayes adalah algoritma klasifikasi yang banyak digunakan dalam pemrosesan teks karena kesederhanaannya dan efisiensinya dalam waktu komputasi. Meskipun dianggap setiap karakteristik saling tidak bergantung, algoritma ini tetap mampu memberikan hasil klasifikasi yang baik, terutama jika data telah melalui proses pembersihan dan transformasi yang memadai [12].

Kurniawan dan Dewi mencatat bahwa Naïve Bayes sangat cocok untuk digunakan dalam analisis sentimen di media sosial, karena mampu mengelompokkan opini masyarakat secara cepat dengan tingkat akurasi yang konsisten [13].

2.5. Decision Tree

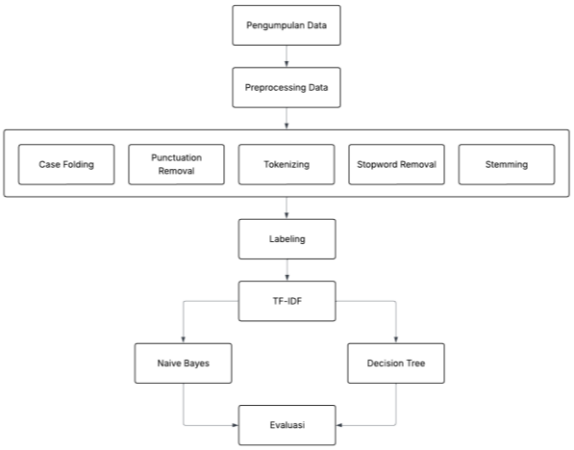
Decision tree adalah pohon keputusan rekursif top-down yang menggunakan algoritma induksi dan menerapkan ukuran pemilihan atribut untuk menentukan atribut mana yang akan diuji. Algoritma *Decision Tree* bertujuan untuk meningkatkan performa dengan menghilangkan cabang pohon yang mewakili informasi yang tidak relevan [14].

Decision Tree dapat mengubah data ke dalam pohon keputusan yang menggambarkan suatu aturan. Aturan yang sudah ada sudah dipahami dengan baik. Teknik Pohon Keputusan mencakup mengubah data menjadi pohon keputusan, kemudian mengubah pohon tersebut menjadi aturan dan menyederhanakan kembali aturan tersebut [15].

3. METODE PENELITIAN

Riset ini menganalisis opini masyarakat terhadap kendaraan listrik di Indonesia dengan mengumpulkan data Twitter. Data dikumpulkan melalui proses crawling menggunakan kata kunci tertentu, kemudian diproses melalui tahapan

preprocessing teks seperti *case folding, punctuation removal, tokenizing, stopword removal*, dan *stemming*. Setelah itu, terdapat proses pengubahan data ke dalam bentuk numerik menggunakan teknik TF-IDF, lalu dikategorikan dengan algoritma *Naïve Bayes* dan *Decision Tree*. Model tersebut diuji dan dievaluasi untuk menentukan performa dalam pengelompokan sentimen.



Gambar 1. Alur metode penelitian

3.1. Pengumpulan Data

Pengumpulan data Twitter dilakukan melalui *web crawling. Web crawling* digunakan untuk mengekstrak data secara otomatis dari halaman Twitter dengan mempertimbangkan kebijakan layanan yang berlaku. Data yang dikumpulkan sebanyak 631 dan 1921 *tweet* dalam rentan waktu mulai dari 1 Januari 2024 hingga 5 Mei 2025 yang meliputi isi *tweet*, waktu posting, informasi pengguna, *hashtag*, *mention*, *retweet*, dan *like*. Proses dimulai dengan penentuan kata kunci dan tagar yang relevan seperti *"*Kendaraan Listrik*"* dan *"*Mobil Listrik*"*, penyaringan *tweet* tidak relevan yang mencakup preprocessing data seperti *case foldiing, puncuation removal, tokenizing, stopword removal,* dan *stemming*, serta penyimpanan data dalam format terstruktur seperti CSV. Untuk menggunakan *Twitter* API, pendaftaran sebagai developer diperlukan guna memperoleh token otentikasi akses [16].

3.2. Preprocessing Data

Preprocessing data adalah tahap dimulainya mempersiapkan data teks untuk analisis. Tahap tersebut digunakan dalam pembersihan data yang tidak perlu agar algoritma dapat memprosesnya dengan lebih efisien. Berikut ini langkah-langkah preprocessing data:

a. *Case Folding*
Teks diubah seluruhnya menjadi huruf kecil agar kata seperti "EV" dan "ev" dikenali sebagai entitas yang sama.

$$CaseFolding(w) = LowerCase(w) \quad (1)$$

LowerCase(w) mengubah semua karakter dalam *w* ke dalam bentuk huruf kecil.

b. *Punctuation Removal*
Tanda baca seperti titik, koma, tanda seru, dan tanda tanya dihapus karena tidak memiliki kontribusi terhadap makna analisis.

$$PunctuationRemoval(w) = w - \{p \mid p \in Punctuation\} \quad (2)$$

Dengan w mewakili sebuah kata dalam teks, dan *Punctuation* merupakan kumpulan dari seluruh tanda baca.

c. *Tokenizing*
Tokenisasi adalah proses membagi teks menjadi unit kata individu (token). Misalnya, kalimat "Mobil listrik ramah lingkungan" akan diubah menjadi ["mobil", "listrik", 'ramah', "lingkungan"].

$$Tokenize(S) = \{w1, w2, \ldots, wn\} \quad (3)$$

Dengan S yaitu frasa atau dokumen, dan $w1$, $w2$, ..., wn yaitu token yang telah dibuat.

d. *Stopword Removal*
Kata-kata umum yang sering muncul namun tidak memberikan makna penting, seperti "dan," "itu," dan "di," juga dihilangkan melalui proses penghapusan *stopword*.

$$StopwordRemoval(T) = T - \{w \mid w \in Stopwords\} \quad (4)$$

T merupakan kumpulan kata, dan *Stopwords* merupakan kumpulan kata-kata umum yang akan dihilangkan.

e. *Stemming*
Mengganti kata-kata yang berimbuhan dalam bentuk dasarnya, seperti "menggunakan" menjadi "gunakan," membantu menyatukan variasi kata dalam analisis.

$$Stemming\ (w) = Stem\ (w) \quad (5)$$

Dimana *w* sebagai kata majemuk, dan Stem(w) menghasilkan bentuk asalnya w [17].

3.3. Labeling

Proses pelabelan dilakukan untuk mengklasifikasikan *tweet* ke dalam dua kategori sentimen, yaitu positif dan negatif, berdasarkan analisis terhadap isi komentar yang mendukung atau menolak isu kendaraan listrik. Seluruh proses ini dijalankan menggunakan Google Colab dengan bahasa pemrograman Python. Untuk mengatasi ketidakseimbangan data antar kelas, diterapkan metode *Synthetic Minority Oversampling Technique (SMOTE)* sebagai pendekatan *oversampling* yang efektif. Sebelum diterapkan metode *SMOTE*, distribusi kelas menunjukkan sentimen negatif yang sangat signifikan terhadap ulasan *tweet* kendaraan listrik dibandingkan sentimen positif. Setelah diterapkan metode *SMOTE*, data dapat diatasi dengan menambah sampel pada tiap sentimen yang bertujuan menyeimbangkan sentimen positif dan negatif. *SMOTE* mengatasi ketidakseimbangan data dengan menghasilkan data sintetis dari kelas minoritas,

yang seringkali memiliki data yang tidak seimbang [18].

3.4. Metode TF-IDF

TF-IDF merupakan salah satu metode pembobotan kata yang paling umum digunakan dalam analisis teks karena mampu menghasilkan nilai akurasi dan *recall* yang tinggi. Dibandingkan *Word2Vec* yang memerlukan pelatihan *embedding*, TF-IDF tidak membutuhkan proses pelatihan tambahan dan lebih hemat sumber daya komputasi, menjadikannya lebih efisien untuk eksperimen berskala sedang seperti penelitian ini. Metode ini memberikan bobot besar pada kata-kata yang sering muncul dalam satu dokumen namun jarang ditemukan di dokumen lain, sehingga secara efektif merepresentasikan kata-kata yang memiliki makna penting. TF-IDF terdiri dari dua komponen utama, yaitu *term frequency* (TF) dan *inverse document frequency* (IDF).

Term Frequency (TF) mengukur kinerja suatu istilah nampak pada dokumen. Semakin tinggi frekuensi kemunculannya, semakin besar bobot atau relevansi istilah tersebut.

$$TF(tk, dj) = f(tk, dj) \quad (6)$$

Inverse Document Frequency (IDF) mengukur keunikan istilah dalam dokumen. Semakin banyak dokumen yang memuat istilah tersebut, semakin kecil bobot IDF-nya.

$$IDF(tk) = \frac{loglog\ D}{df(t)} \quad (7)$$

TF-IDF dapat dirumuskan sebagai berikut [19]:

$$TF\ IDF(tk, dj) = TF(tk, dj) * IDF(tk) \quad (8)$$

a. *Naive Bayes*
   Naïve Bayes adalah algoritma klasifikasi berbasis probabilistik yang menerapkan Teorema Bayes dengan asumsi bahwa setiap fitur bersifat independen satu sama lain. Data yang digunakan dalam algoritma ini dibagi menjadi dua bagian, yaitu data pelatihan (training) dan data pengujian (testing). Dalam analisis teks Twitter, metode ini mengklasifikasikan data berdasarkan probabilitas kemunculan istilah dalam kategori tertentu.

$$P\left(\frac{A}{B}\right) = \frac{P\left(\frac{B}{A}\right) X\ P(A)}{P(B)} \quad (9)$$

   Keterangan :
   $P\frac{A}{B}$ = probabilitas suatu data A diberikan data B.
   $P\frac{B}{A}$ = probabilitas data BBB muncul dalam data A.
   P(A) = probabilitas awal dari data A.
   P(B) = probabilitas total dari data B [20].

b. *Decision Tree*
   Sementara itu, *Decision Tree* merupakan metode klasifikasi berbasis struktur pohon yang bekerja dengan membagi data secara bertingkat berdasarkan atribut tertentu hingga mencapai keputusan akhir berupa kelas target. Sama seperti pada algoritma *Naïve Bayes*, data yang digunakan juga dibagi menjadi data pelatihan dan data pengujian. Pemilihan fitur terbaik dilakukan menggunakan Gini Index, yang mengukur kemurnian data semakin kecil nilainya, semakin baik pemisahannya. Gini Index memiliki perhitungan komputasi yang lebih sederhana dan lebih cepat dibandingkan dengan Information Gain, karena tidak menggunakan logaritma. Sehingga lebih efisien untuk dataset yang cukup besar, seperti data tweet yang digunakan dalam penelitian ini.

$$Gini = 1 - \sum_{i=1}^{n} p_i^2 \quad (10)$$

   Keterangan :
   Pi = probabilitas suatu kategori di dalam node.
   n = jumlah kategori dalam dataset [21].

3.5. Evaluasi

Evaluasi dilakukan untuk mengukur kinerja model klasifikasi sentimen yang dilatih menggunakan metode TF-IDF serta algoritma *Naïve Bayes* dan *Decision Tree*. Pemisahan dataset dibagi menjadi dua bagian, yaitu training set dan testing set. Pembagian ini dilakukan secara acak (*random state*) dengan skenario split 80:20. Model dievaluasi menggunakan data uji yang telah dipisahkan dari data latih dalam proses pembagian dataset. Evaluasi dilakukan dengan menggunakan confusion matrix, yang mencakup empat metrik utama:

a. *Akurasi* : Mengukur persentase prediksi yang tepat dibandingkan dengan seluruh jumlah data uji. Semakin tinggi nilainya, semakin baik model dalam memahami dan mengklasifikasikan sentimen secara keseluruhan.
b. *Presisi* : Mengukur ketepatan model dalam mengklasifikasi kategori sentimen. Presisi yang tinggi menunjukkan hasil positif model yang sebagian besar valid.
c. *Recall* : *menilai* sejauh mana model dapat mengidentifikasi seluruh data positif yang sebenarnya ada dalam dataset. Nilai recall yang tinggi menunjukkan model dapat mengumpulkan semua data yang relevan.
d. *F1-Score* : *Menilai* keseimbangan rata-rata harmonik antara presisi dan recall. F1-Score penting ketika data memiliki ketidakseimbangan antar kelas [22].

Setelah proses pengujian, dilakukan perbandingan terhadap hasil evaluasi mencakup akurasi, presisi, *recall*, dan *F1-Score* dari dua algoritma, yaitu *Naïve Bayes* dan *Decision Tree*, dalam mengklasifikasikan sentimen pengguna media sosial mengenai kendaraan listrik di Indonesia. Perbedaan performa antara kedua algoritma ini tergolong signifikan secara statistik mengingat confusion matrix yang menunjukkan

bahwa *Decision Tree* mampu mengklasifikasi jauh lebih baik daripada *Naïve Bayes*. Setelah memproses data dan mengekstrak fitur menggunakan TF-IDF, setiap model dievaluasi pada dataset uji. *Naïve Bayes* bekerja pada probabilitas fitur, sedangkan *Decision Tree* menggunakan aturan untuk mengklasifikasikan data. Hasil evaluasi digunakan untuk mengidentifikasi model dengan performa dengan akurasi terbaik dalam analisis sentimen.

4. HASIL DAN PEMBAHASAN

Penelitian ini secara khusus menganalisis sentimen publik pada media sosial yang berkaitan dengan topik kendaraan listrik. Sentimen dikategorikan menjadi dua kelas, yaitu positif dan negatif. Proses pelabelan kelas positif dan negatif dalam penelitian ini dilakukan secara otomatis. Penentuan label dilakukan menggunakan fungsi *label_sentiment(text)* yang dikembangkan di *Google Colab* dengan bahasa pemrograman *Python*.

Proses klasifikasi dilakukan dengan pendekatan pembelajaran mesin, untuk membandingkan performa kedua algoritma, yaitu *Naive Bayes* dan *Decision Tree*, yang masing-masing diterapkan setelah tahap preprocessing data dan pembobotan fitur menggunakan metode TF-IDF. Evaluasi terhadap performa model dilakukan dengan menggunakan *Confusion Matrix*, yang menghasilkan matrix berupa nilai akurasi, presisi, *recall*, dan *f-measure*. *Confusion matrix* digunakan sebagai dasar penilaian untuk mengukur kemampuan masing-masing algoritma dalam mengklasifikasikan sentimen secara akurat dan konsisten berdasarkan data yang tersedia.

4.1. Pengumpulan Data

Dalam riset ini, data dikumpulkan melalui platform media sosial *Twitter* untuk menganalisis opini publik mengenai kendaraan listrik di Indonesia. Pengambilan data dilakukan menggunakan Twitter teknik crawling dengan kata kunci "Kendaraan Listrik" dan "Mobil Listrik", yang menghasilkan masing-masing sebanyak 631 dan 1921. Proses crawling dilakukan dalam rentan waktu mulai dari 1 Januari 2024 hingga 5 Mei 2025. Kata kunci "Kendaraan Listrik" dan "Mobil Listrik" dipilih berdasarkan dua pertimbangan, yaitu relevansi topik dengan isu penelitian dan tren percakapan publik di media sosial. Selain itu, kata kunci ini sering digunakan oleh masyarakat Indonesia saat membahas kendaraan listrik, baik dari segi teknologi maupun kebijakan.

Secara umum, ukuran dataset ini cukup memadai untuk klasifikasi *machine learning*, terutama jika mempertimbangkan kompleksitas model yang digunakan, yaitu *Naïve Bayes* dan *Decision Tree*. Seluruh data kemudian digabung dan disimpan dalam format CSV. Setiap kolom dataset memuat informasi seperti nama pengguna, isi *tweet*, tanggal unggahan, jumlah like dan retweet. Informasi lebih rinci terkait hasil pengumpulan data dapat diperhatikan pada Tabel 1.

Tabel 1. Dataset hasil crawling

Dataset	Jumlah Data
Kendaraan Listrik	631
Mobil Listrik	1921
2552	**2552**

4.2. *Preprocessing Data*

Secara keseluruhan, proses text preprocessing merupakan tahap awal yang krusial dalam pengolahan teks. Tahapan ini mencakup sejumlah langkah pembersihan dan transformasi data teks mentah menjadi format yang lebih siap untuk dianalisis secara komputasional:

a. Case Folding

Tahap *Case Folding* digunakan dengan mengganti seluruh huruf dalam teks menjadi huruf kecil. Hal ini bertujuan untuk menyamakan kata-kata yang secara makna sama namun berbeda penulisan karena penggunaan huruf kapital, seperti "Mobil" dan "mobil", tidak dianggap berbeda. Akibatnya, jumlah kata unik berkurang dan data menjadi lebih konsisten. Berikut hasil dari *Case Folding* pada Tabel 2.

Tabel 2. Hasil proses case folding

Sebelum	Sebelum
Harga sel baterai bisa mencapai 75% dari harga total kendaraan listrik meskipun demikian ada baiknya desain baterainya bisa dilepas-pasang dgn mudah sehingga jika sampai kejadian macem gini setidaknya motornya masih bisa diselamatkan gak ikut kobong.	harga sel baterai bisa mencapai 75% dari harga total kendaraan listrik meskipun demikian ada baiknya desain baterainya bisa dilepas-pasang dgn mudah sehingga jika sampai kejadian macem gini setidaknya motornya masih bisa diselamatkan gak ikut kobong.

b. Punctuation Removal

Tahap *Punctuation Removal* dilakukan dengan menghapus seluruh tanda baca. Hasil dari proses ini menunjukkan bahwa teks menjadi lebih sederhana dan berfokus pada kata-kata yang bermakna. Dengan menghapus tanda baca, data menjadi lebih bersih, sehingga dapat meningkatkan akurasi klasifikasi pada algoritma *Naive Bayes* dan *Decision Tree*. Berikut hasil dari *Punctuation Removal* pada Tabel 3.

Tabel 3. Hasil proses punctuation removal

Sebelum	Sesudah
harga sel baterai bisa mencapai 75% dari harga total kendaraan listrik meskipun demikian ada baiknya desain baterainya bisa dilepas-pasang dgn	harga sel baterai bisa mencapai 75 dari harga total kendaraan listrik meskipun demikian ada baiknya desain baterainya bisa dilepas pasang dgn

Sebelum	Sesudah
mudah sehingga jika sampai kejadian macem gini setidaknya motornya masih bisa diselamatkan gak ikut kobong.	mudah sehingga jika sampai kejadian macem gini setidaknya motornya masih bisa diselamatkan gak ikut kobong

c. Tokenizing

Tahapan *Tokenizing* berperan dalam memecah teks mentah menjadi unit-unit kata atau token. Proses ini penting untuk mengubah kalimat panjang menjadi format yang lebih terstruktur, sehingga mempermudah proses analisis selanjutnya. Hasil tokenisasi ditunjukkan pada Tabel 4.

Tabel 4. Hasil proses tokenizing

Sebelum	Sesudah
harga sel baterai bisa mencapai 75 dari harga total kendaraan listrik meskipun demikian ada baiknya desain baterainya bisa dilepas pasang dgn mudah sehingga jika sampai kejadian macem gini setidaknya motornya masih bisa diselamatkan gak ikut kobong	harga, sel, baterai, bisa, mencapai, 75, dari, harga, total, kendaraan, listrik, meskipun, demikian, ada, baiknya, desain, baterainya, bisa, dilepas, pasang, dgn, mudah, sehingga, jika, sampai, kejadian, macem, gini, setidaknya, motornya, masih, bisa, diselamatkan, gak, ikut, kobong

d. Stopword Removal

Stopword Removal merupakan proses penghapusan kata-kata umum yang tidak memiliki makna khusus dalam analisis sentimen, seperti "dan", "atau", dan "itu". Dalam penelitian ini, kata-kata stopword didefinisikan dan dipilih menggunakan kombinasi kata standar dari pustaka Sastrawi, yang merupakan library untuk pengolahan teks berbahasa Indonesia. Tujuannya adalah untuk mengurangi kata-kata yang tidak relevan agar model dapat lebih fokus pada kata-kata yang memiliki nilai informasi tinggi dalam proses klasifikasi. Hasil dari tahap ini ditampilkan pada Tabel 5.

Tabel 5. Hasil proses stopword removal

Sebelum	Sesudah
harga, sel, baterai, bisa, mencapai, 75, dari, harga, total, kendaraan, listrik, meskipun, demikian, ada, baiknya, desain, baterainya, bisa, dilepas, pasang, dgn, mudah, sehingga, jika, sampai, kejadian, macem, gini, setidaknya, motornya, masih, bisa, diselamatkan, gak, ikut, kobong	harga, sel, baterai, bisa, mencapai, 75, harga, total, kendaraan, listrik, meskipun, demikian, ada, baiknya, desain, baterainya, bisa, dilepas, pasang, mudah, sehingga, jika, sampai, kejadian, macem, gini, setidaknya, motornya, masih, bisa, diselamatkan, ikut, kobong

e. Stemming

Tahap Stemming bertujuan menyederhanakan kata-kata yang memiliki imbuhan menjadi bentuk dasar. Proses ini membantu menyatukan imbauan kata agar dianggap sama dalam analisis, seperti kata "pengujian" menjadi "uji", serta "terbaru" menjadi "baru". Dengan stemming, data teks menjadi lebih konsisten dan mengurangi duplikasi, sehingga metode TF-IDF dapat menghasilkan fitur yang lebih representatif. Hal ini berkontribusi meningkatkan performa klasifikasi pada algoritma Naive Bayes dan Decision Tree. Berikut hasil dari Stemming pada Tabel 6.

Tabel 6. Hasil proses stemming

Sebelum	Sesudah
harga, sel, baterai, bisa, mencapai, 75, harga, total, kendaraan, listrik, meskipun, demikian, ada, baiknya, desain, baterainya, bisa, dilepas, pasang, mudah, sehingga, jika, sampai, kejadian, macem, gini, setidaknya, motornya, masih, bisa, diselamatkan, ikut, kobong	harga, sel, baterai, bisa, mencapai, 75, harga, total, kendaraan, listrik, meskipun, demikian, ada, baik, desain, baterai, bisa, lepas, pasang, mudah, sehingga, jika, sampai, kejadian, macem, gini, tidak, motor, masih, bisa, selamat, ikut, kobong

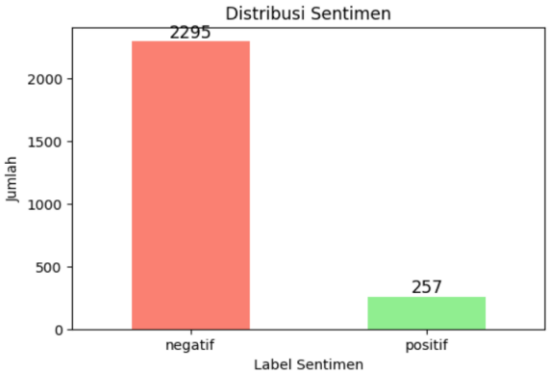
4.3. Labeling

Hasil pelabelan menunjukkan ketidakseimbangan yang signifikan dalam pembagian kelas. Pelabelan menggunakan tools Google Colab bahasa pemrograman Python dengan fungsi *label_sentiment(text)* untuk mengklasifikasi sentimen teks menjadi positif dan negatif berdasarkan kemunculan kata-kata dalam daftar positive_words dan negatve_words seperti pada gambar 1. Kriteria pemilihan kata sentimen positif mencerminkan dukungan, kepuasan, atau opini baik masyarakat seperti *bagus*, *hemat*, *canggih*, *efisien*, *nyaman*, dan *ramah lingkungan*. Sementara untuk kriteria pemilihan kata sentimen negatif mencerminkan ketidakpuasan atau kritik seperti *mahal*, *lambat*, *sulit*, dan *tidak efisien*.

```
import matplotlib.pyplot as plt

def label_sentiment(text):
    text = text.lower()
    pos = sum(word in text for word in positive_words)
    neg = sum(word in text for word in negative_words)
    if pos > neg:
        return 'positif'
    else:
        return 'negatif'

# Terapkan pelabelan ke dataframe
df['label'] = df['final_text'].apply(label_sentiment)

# Tampilkan hasil pelabelan untuk 10 data pertama
print(df[['full_text', 'final_text', 'label']].head(10))

# Tampilkan total hasil positif dan negatif
print("Total sentimen:")
print(df['label'].value_counts())
```

Gambar 1. Program python labeling positif dan negatif

Dari seluruh data yang diperoleh, 2.295 diklasifikasikan sebagai negatif, sedangkan hanya 257 diklasifikasikan sebagai positif, seperti tampak pada Gambar 2.



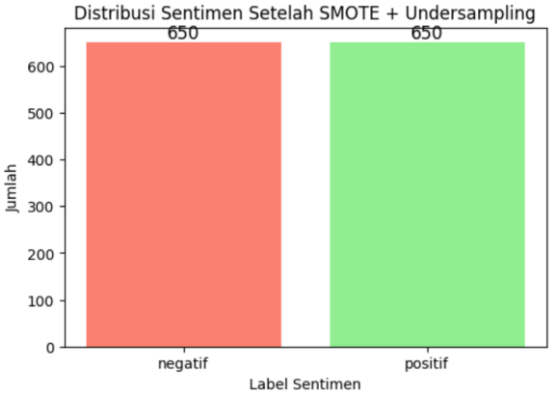
Gambar 2. Jumlah sentimen positif dan negatif

Untuk mengatasi masalah ini, penyeimbangan data diproses menggunakan tools Google Colab bahasa pemrograman Python yang menggabungkan teknik *SMOTE* menggunakan library imblearn.over_sampling dan undersampling. Namun, karena distribusi data awal sangat tidak seimbang (negatif: 2295; positif: 257), penggunaan *SMOTE* saja tidak cukup. Sebagai hasilnya, dilakukan undersampling pada kelas negatif menggunakan fungsi n_sample_per_class sebanyak 650 per label, yaitu dengan mengurangi jumlah data negatif untuk menyeimbangkan hasil oversampling pada kelas positif. Jumlah 650 data dianggap ideal karena memiliki kinerja cukup besar untuk menghindari overfitting dari data SMOTE serta terbukti menghasilkan performa model yang optimal dan efisien. Hasil pemrograman python *SMOTE* terdapat pada gambar 3.

```
from imblearn.over_sampling import SMOTE
from sklearn.utils import resample
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer()
X = tfidf.fit_transform(df['final_text'])
y = df['label']

# SMOTE untuk menyeimbangkan kelas
smote = SMOTE(random_state=42)
# Gunakan X dan y yang baru saja dibuat
X_resampled, y_resampled = smote.fit_resample(X, y)

# Gabungkan hasil SMOTE ke DataFrame sementara
df_resampled = pd.DataFrame(X_resampled.toarray())
df_resampled['label'] = y_resampled

# Undersampling agar total data sekitar 1000 (500 per kelas)
n_sample_per_class = 650
df_balanced = pd.concat([
    resample(df_resampled[df_resampled['label'] == label],
             replace=False,
             n_samples=n_sample_per_class,
             random_state=42)
    for label in df_resampled['label'].unique()])

# Pisahkan kembali fitur dan label
X_final = df_balanced.drop('label', axis=1).values
y_final = df_balanced['label'].values
```

Gambar 3. Program python penggunaan SMOTE

Setelah itu, distribusi data menjadi seimbang, dengan masing-masing kelas positif dan negatif sebanyak 650 data, yang nampak pada Gambar 4.



Gambar 4. Jumlah sentimen positif dan negatif setelah optimasi SMOTE

4.4. Metode TF-IDF

Berdasarkan Tabel 7, diperlihatkan beberapa kolom mengenai hasil pembobotan nilai TF-IDF. Kolom Kata berisi kata-kata yang sering muncul dalam tweet. Kolom Frekuensi menunjukkan jumlah kemunculan masing-masing kata. Kolom TF-IDF memperhitungkan pembobotan kata berdasarkan frekuensi dan tingkat keunikan dalam dokumen. Sedangkan kolom TF-IDF Rata-rata menunjukkan nilai rata-rata bobot TF-IDF per kata.

Tabel 7. Hasil nilai TF-IDF

Kata	Frekuensi	TF-IDF	TF-IDF Rata-rata
listrik	1265	66.570095	0.051208
mobil	1084	70.366494	0.054128
kendaraan	440	37.115979	0.028551
indonesia	263	31.039015	0.023876
lebih	242	34.202769	0.026310
ada	211	20.138116	0.015491
bisa	210	21.221733	0.016324
baterai	203	20.722484	0.015940
buat	202	21.532496	0.016563
jadi	197	19.294204	0.014842

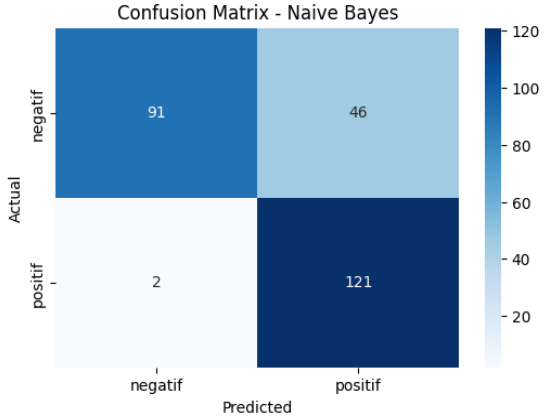
a. *Naive Bayes*

Dalam penelitian ini, proses pembagian data untuk algoritma Naïve Bayes dilakukan dengan rasio 80:20, di mana 80% data digunakan sebagai data pelatihan (training) dan 20% sebagai data pengujian (testing) dari total 1.300 data yang telah diseimbangkan. Sebanyak 1.040 data digunakan untuk pelatihan dan 260 data untuk pengujian, sebagaimana diperlihatkan pada Tabel 8. Pembagian ini bertujuan untuk mengevaluasi kinerja model dalam mengklasifikasikan sentimen secara akurat serta memastikan kemampuan model dalam melakukan generalisasi terhadap data baru yang tidak dilibatkan pada proses pelatihan.

Tabel 8. Pembagian split data Naive Bayes

Persentase Data		Jumlah Data	
Training	**Testing**	**Training**	**Testing**
80%	20%	1040	260
Total		**1300**	

Setelah dilakukan pengujian, model *Naive Bayes* divisualisasikan sebagaimana ditampilkan pada Gambar 5.



Gambar 5. Visualisasi algoritma Naive Bayes

Model klasifikasi *Naive Bayes* dievaluasi menggunakan confusion matrix dengan hasil 91 true negatives, 121 true positives, 46 false positives, dan 2 false negatives dari total 260 data uji. Akurasi model mencapai 81,54%, dengan nilai recall kelas positif sebesar 98,37%, menunjukkan kemampuan tinggi dalam mendeteksi data positif. Namun, nilai precision hanya 72,46%, menandakan masih tingginya prediksi positif yang salah (false positives). Meskipun model cukup akurat dalam menganalisa sentimen positif, peningkatan diperlukan pada aspek precision, terutama untuk mengurangi kesalahan klasifikasi data negatif.

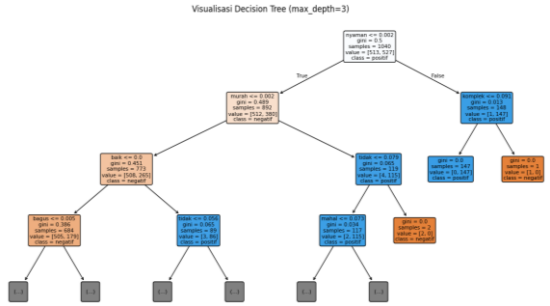
b. *Decision Tree*

Pembagian data pada algoritma *Decision Tree* dilakukan dengan menggunakan skenario rasio 80:20, yaitu sebanyak 1.040 data diuji berdasarkan Training data serta 260 data diuji berdasarkan Testing data, dari total 1.300 data. Pembagian ini bertujuan untuk mengukur dan membandingkan performa klasifikasi algoritma *Decision Tree* terhadap algoritma *Naive Bayes* dalam menganalisis sentimen media sosial mengenai kendaraan listrik di Indonesia. Dalam proses pembangunan model *Decision Tree*, digunakan *parameter max_depth* sebesar 3 guna mengendalikan kompleksitas struktur pohon dan menghindari terjadinya *overfitting*. Penetapan parameter *max_depth* = 3 dianggap cukup untuk pemisahan kelas yang optimal tanpa memerlukan kompleksitas lebih lanjut. Nilai ini terbukti optimal berdasarkan hasil uji awal yang menunjukkan akurasi tinggi (0,99) serta kemampuan model dalam menggeneralisasi data dengan baik. Hasil dari pembagian dan pengujian ini disajikan pada Tabel 9.

Tabel 9. Pembagian split data Decision Tree

Persentase Data		Jumlah Data	
Training	**Testing**	**Training**	**Testing**
80%	20%	1040	260
Total		**1300**	

Setelah dilakukan pengujian, model Decision Tree divisualisasikan seperti pada Gambar 6.



Gambar 6. Visualisasi algoritma Decision Tree

Visualisasi *Decision Tree* dengan parameter *max_depth* sebesar 3, menggambarkan proses klasifikasi sentimen dari 1.040 data, yang terdiri dari 513 data negatif dan 527 data positif. Fitur awal yang digunakan adalah kata "nyaman", yang membagi data menjadi dua cabang utama. Cabang kiri didominasi sentimen negatif, dengan fitur penting seperti "murah", "baik", dan "bagus". Sedangkan cabang kanan didominasi sentimen positif, dengan fitur seperti "tidak", "mahal", dan "komplek". Nilai gini impurity rendah pada beberapa node menunjukkan pemisahan kelas yang cukup baik. *Decision Tree* memperlihatkan bahwa fitur kata tertentu sangat berpengaruh dalam memprediksi sentimen secara akurat.

4.5. Evaluasi

Setelah dilakukan pembagian data dengan skenario split 80:20 pada kedua algoritma, proses pengujian model dilaksanakan. Pengujian ini menghasilkan nilai evaluasi berupa akurasi, presisi, recall, dan F1-score. Model *Naïve Bayes* dan *Decision Tree* masing-masing dilatih dan diuji dengan proporsi data 80% untuk pelatihan dan 20% untuk pengujian.

Berdasarkan hasil pada Tabel 10, algoritma *Naïve Bayes* memperoleh nilai akurasi sebesar 0,82, presisi 0,85, *recall* 0,82, dan *F1-score* sebesar 0,81. Di sisi lain, algoritma *Decision Tree* menunjukkan performa yang lebih tinggi dan konsisten dengan nilai akurasi, presisi, *recall*, dan *F1-score* yang sama, yaitu sebesar 0,99. Kinerja tinggi *Decision Tree* dengan akurasi sebesar 0,99 disebabkan oleh kombinasi representasi fitur yang kuat dari TF-IDF, struktur *Decision Tree* yang optimal dengan max_depth = 3, dan penggunaan *Gini Index* sebagai pemisah yang efisien.

Tabel 10. Tabel perbandingan naive bayes dan decision tree

Algoritma	Akurasi	Presisi	Recall	F1-Score
Naive Bayes	0,82	0,85	0,82	0,81
Decision Tree	0,99	0,99	0,99	0,99

Berdasarkan Tabel 11, diketahui nilai akurasi model *Naive Bayes* sebesar 0,82 atau sekitar 82%, sedangkan akurasi model *Decision Tree* sebesar 0,99 atau 99%. Nilai akurasi ini menunjukkan seberapa besar performa data yang berhasil diklasifikasikan oleh masing-masing model. Dengan perbedaan akurasi yang cukup signifikan, *Decision Tree* mampu mengklasifikasikan data dengan lebih akurat dibandingkan *Naive Bayes* dalam menganalisis sentimen media sosial terhadap kendaraan listrik di Indonesia.

Tabel 11. Nilai akurasi naive bayes dan decision tree

Algoritma	Akurasi
Naive Bayes	0,8154
Decision Tree	0,9885

5. KESIMPULAN DAN SARAN

Dari hasil penelitian, dapat disimpulkan bahwa algoritma *Decision Tree* menunjukkan kinerja yang lebih optimal dibandingkan *Naïve Bayes* dalam klasifikasi sentimen media sosial Twitter terkait kendaraan listrik di Indonesia. Hal ini dibuktikan dengan nilai akurasi *Decision Tree* sebesar 98,85%, lebih tinggi dibanding dengan *Naïve Bayes* sebesar 81,54%. Selain itu, nilai presisi, *recall*, dan *f1-score* pada *Decision Tree* juga menunjukkan hasil yang konsisten, sehingga algoritma ini dinilai efektif dalam menganalisis data sentimen melalui tahap pembobotan fitur menggunakan metode TF-IDF.

Sebagai rekomendasi untuk penelitian berikutnya, disarankan untuk mengeksplorasi algoritma klasifikasi lain seperti *Random Forest* dan *Support Vector Machine* (SVM) guna membandingkan efektivitas metode klasifikasi dalam konteks serupa, maupun pendekatan berbasis Deep Learning. Selain itu, penggunaan data real-time dan perluasan kategori sentimen dapat menjadi alternatif untuk meningkatkan performa model dan memperluas cakupan analisis.

DAFTAR PUSTAKA

[1] 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.

[2] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends® in information retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[3] A. Mahalana and Z. Yang, "Overview of vehicle fuel efficiency and electrification policies in Indonesia," *The International Council on Clean Transportation*, 2021.

[4] T. Prasetyo, A. A. Waskita, and T. Taryo, "Analisis Sentimen Pengguna Mobil Listrik di Media Sosial Twitter Menggunakan Metode Klasifikasi Naïve Bayes, K-Nearest Neighbor (KNN), dan Decision Tree," *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, vol. 8, no. 2, pp. 108–117, 2025.

[5] J. Ye, X. Jing, and J. Li, "Sentiment analysis using modified LDA," in *Signal and Information Processing, Networking and Computers: Proceedings of the 3rd International Conference on Signal and Information Processing, Networking and Computers (ICSINC) 3*, Springer, 2018, pp. 205–212.

[6] Y. A. Singgalen, "Penerapan Metode CRISP-DM dalam Klasifikasi Data Ulasan Pengunjung Destinasi Danau Toba Menggunakan Algoritma Naïve Bayes Classifier (NBC) dan Decision Tree (DT)," *J. Media Inform. Budidarma*, vol. 7, no. 3, pp. 1551–1562, 2023.

[7] G. H. Kusuma, I. Permana, F. N. Salisah, M. Afdal, M. Jazman, and A. Marsal, "Pendekatan Machine Learning: Analisis Sentimen Masyarakat Terhadap Kendaraan Listrik Pada Sosial Media X," *JUSIFO (Jurnal Sistem Informasi)*, vol. 9, no. 2, pp. 65–76, 2023.

[8] M. Guia, R. R. Silva, and J. Bernardino, "Comparison of Naïve Bayes, Support Vector Machine, Decision Trees and Random Forest on Sentiment Analysis.," *KDIR*, vol. 1, pp. 525–531, 2019.

[9] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, and W. Gata, "Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi," *Jurnal Teknoinfo*, vol. 14, no. 2, p. 115, 2020.

[10] P. H. Prastyo, I. Ardiyanto, and R. Hidayat, "Indonesian Sentiment Analysis: An Experimental Study of Four Kernel Functions on SVM Algorithm with TF-IDF," in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, IEEE, 2020, pp. 1–6.

[11] D. E. Cahyani and I. Patasik, "Performance comparison of tf-idf and word2vec models for emotion text classification," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2780–2788, 2021.

[12] A. Agustian and F. Nurapriani, "Analisis Sentimen, Text Mining Penerapan Analisis

Sentimen Dan Naive Bayes Terhadap Opini Penggunaan Kendaraan Listrik Di Twitter," *Jurnal Tika*, vol. 7, no. 3, pp. 243–249, 2022.

[13] A. Karimah, G. Dwilestari, and M. Mulyawan, "Analisis Sentimen Komentar Video Mobil Listrik Di Platform Youtube Dengan Metode Naive Bayes," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 1, pp. 737–767, 2024.

[14] D. Suyanto, "Data Mining untuk klasifikasi dan klasterisasi data," *Bandung: Informatika Bandung*, 2017.

[15] K. G. Prasetyo, "Analisis Perbandingan Algoritma Decision Tree Dengan Support Vector Machine Untuk Mendeteksi Kompetensi Mahasiswa Konsentrasi Informatika Komputer Studi Kasus: Politeknik Lp3I Jakarta, Kampus Depok," *JURNAL LENTERA ICT*, vol. 5, no. 2, pp. 11–26, 2019.

[16] K. SENTIMEN, "SUPPORT VECTOR MACHINE SENTIMENT CLASSIFICATION OF MOVIE REVIEWS USING ALGORITHM SUPPORT VECTOR MACHINE".

[17] S. J. Angelina, A. B. P. Negara, and H. Muhardi, "Analisis Pengaruh Penerapan Stopword Removal Pada Performa Klasifikasi Sentimen Tweet Bahasa Indonesia," *Jurnal Aplikasi dan Riset Informatika*, vol. 1, no. 2, pp. 165–173, 2023.

[18] U. H. Laksono and R. R. Suryono, "SENTIMENT ANALYSIS OF ONLINE DATING APPS USING SUPPORT VECTOR MACHINE AND NAÏVE BAYES ALGORITHMS," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 1, pp. 229–238, 2025.

[19] D. Ruppert, "The elements of statistical learning: data mining, inference, and prediction," 2004, *Taylor & Francis*.

[20] R. Hidayat, M. Fikry, Y. Yusra, F. Yanto, and E. P. Cynthia, "Penerapan Naïve Bayes Classifier dalam Klasifikasi Sentimen Publik di Twitter terhadap Puan Maharani," *JUKI: Jurnal Komputer dan Informatika*, vol. 6, no. 1, pp. 100–108, 2024.

[21] K. Zerrouki, R. M. Hamou, and A. Rahmoun, "Sentiment analysis of tweets using Naïve Bayes, KNN, and decision tree," in *Research Anthology on Implementing Sentiment Analysis Across Multiple Disciplines*, IGI Global, 2022, pp. 538–554.

[22] S. A. Helmayanti, F. Hamami, and R. Y. Fa'rifah, "Penerapan algoritma Tf-Idf dan Naïve Bayes untuk analisis sentimen berbasis aspek ulasan aplikasi Flip pada Google Play Store," *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, vol. 4, no. 3, pp. 1822–1834, 2023