ANALISIS SENTIMEN PADA MEDIA SOSIAL TWITTER TERHADAP PENGGUNAAN KENDARAAN LISTRIK DENGAN MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER DAN INFORMATION GAIN

by 09021282025046 Ivando Sibarani

Submission date: 24-Jul-2024 05:56PM (UTC+0700)
Submission ID: 2421753587
File name: METODE_NA_VE_BAYES_CLASSIFIER_DAN_INFORMATION_GAIN_-_Ivando.docx (140.4K)
Word count: 7017
Character count: 45718

BAB I
PENDAHULUAN

1.1 Latar Belakang

Penggunaan kendaraan listrik semakin mendapatkan perhatian di seluruh dunia sebagai solusi untuk mengurangi emisi karbon dan ketergantungan pada bahan bakar fosil (Alfarizi, & Fitriani. 2023). Namun, tantangan yang dihadapi dalam mengadopsi teknologi kendaraan listrik tidak hanya sebatas pada aspek tersebut, tetapi juga mencakup pada penerimaan dari masyarakat (Ernawati et al. 2023).

Twitter sebagai platform media sosial yang populer di Indonesia, dengan 14.75 juta pengguna aktif pada April 2023, menjadi sumber data yang potensial untuk memahami opini publik[1]. Oleh karena itu, pemahaman mendalam tentang sentimen, dan sikap masyarakat terhadap kendaraan listrik sangatlah penting untuk melihat respon masyarakat, pengembangan kebijakan publik, merancang strategi yang efektif dalam pemasaran dan mendorong adopsi teknologi ini.

Analisis sentimen menjadi hal yang efektif dalam menggali data opini publik. Dengan mengumpulkan dan mengolah opini yang tersebar di *Twitter*, analisis sentimen dapat membantu mengklasifikasikan respons masyarakat ke dalam kategori positif, negatif, atau netral (Alfarizi, & Fitriani, 2023).

Dalam analisis sentimen, salah satu metode klasifikasi yang umum

---

[1] Databoks "Jumlah Pengguna Twitter di Indonesia Capai 14,75 Juta per April 2023"(https://databoks.katadata.co.id/datapublish/2023/05/31/jumlah-pengguna-twitter-di-indonesia-capai-1475-juta-per-april-2023-peringkat-keenam-dunia, diakses pada 15 Maret 2024)

digunakan adalah *Naïve Bayes Classifier*. Algoritma ini didasarkan pada teorema Bayes, yang dikemukakan oleh ilmuwan Inggris Thomas Bayes. Metode ini menggunakan konsep probabilitas dan statistik, dimana *Naïve Bayes Classifier* mencari nilai probabilitas tertinggi untuk mengelompokkan data uji ke dalam kategori yang tepat (Ramadhani, & Suryono. 2024). Namun, ada masalah umum dalam klasifikasi teks yang disebabkan oleh banyaknya dimensi ruang fitur. Pemilihan fitur dapat menjadi solusi untuk masalah ini dengan *Information Gain*. Dengan menggunakan *Information Gain*, fitur yang kurang atau tidak relevan akan dihilangkan, sehingga dimensi data berkurang dan kinerja dapat meningkat (Anggita, & Abdulloh. 2023). Beberapa penelitian telah menunjukkan bahwa penerapan *Information Gain* pada algoritma klasifikasi mampu meningkatkan akurasi yang dihasilkan.

Pada penelitian yang berjudul "Analisis Sentimen Kendaraan Listrik Menggunakan Algoritma *naive bayes* dengan Seleksi Fitur *Information Gain* dan Particle Swarm Optimization " oleh Salman Alfarizi, dan Eka Fitriani pada 2023. Hasil yang diperoleh dengan metode *Naïve Bayes* sebesar 79.43%. Hasil yang didapatkan setelah menggunakan algoritma *Naïve Bayes*, *Information Gain* dan PSO meningkat menjadi 84.54%.

Penelitian selanjutnya dilakakukan oleh Amelia Isnanda, Yuyun Umaidah, dan Jajam Haerul Jaman pada 2021, dengan judul "Implementasi Naïve Bayes *Class*ifier Dan Information Gain Pada Analisis Sentimen Penggunaan E-Wallet Saat Pandemi". Hasil *accuracy* yang didapat dengan metode *naïve bayes* sebesar 84%. Sedangkan hasil klasifikasi menggunakan *Naïve Bayes* dan *Information Gain*

mendapatkan accuracy sebesar 92%.

Penelitian lainnya dengan judul "Perbandingan Metode Klasifikasi Support Vector Machine Dan *Naïve Bayes* Pada Analisis Sentimen Kendaraan Listrik" oleh Ni Wayan Ernawati, I Nyoman Satya Kumara, dan Widyadi Setiawan pada 2023. Pada penelitian ini hasil *accuracy* yang didapatkan dengan metode *Naïve Bayes* sebesar 82% dan hasil dari SVM sebesar 81%. Dengan waktu latih yang dibutuhkan oleh metode SVM yaitu 37.42 detik sedangkan metode *naive bayes* hanya membutuhkan 0.10 detik. Hal ini membuktikan *Naïve Bayes* menjadi metode penelitian yang baik karena memiliki waktu latih yang cepat dan akurasi yang tinggi.

Tujuan dari penelitian ini adalah untuk mengetahui secara sentimen yang diberikan oleh masyarakat terhadap kendaraan listrik dengan menggunakan metode *naive bayes* dan seleksi fitur *information gian*.

1.2 Rumusan Masalah

1. Bagaimana sentimen masyarakat terkait penggunaan kendaraan listrik dapat dianalisis melalui data yang terdapat pada media sosial Twitter dengan menggunakan metode *Naïve Bayes* dan In*formaiton Gain*?
2. Seberapa akurat model *Naïve Bayes* dalam menganalisis sentiment terhadap pengunaan kendaraan Listrik di twitter setelah diterapkan metode *Information Gain?*

1.3 Tujuan Penelitian

1. Tujuan utama penelitian ini adalah untuk menganalisis sentimen masyarakat terkait kendaraan listrik melalui data yang dikumpulkan dari media sosial Twitter, serta mengidentifikasi faktor-faktor yang memengaruhi sentimen tersebut.
2. Mengevaluasi model *Naïve Bayes Classifier* dalam analisis sentimen setelah diterapkan metode *Information Gain*.

1.4 Manfaat penelitian

1. Bagi Peneliti:
   a. Menambah wawasan dan pemahaman mengenai penggunaan metode *Naïve Bayes Classifier* dan *Information Gain* dalam analisis sentimen.
   b. Mengembangkan keterampilan dalam pengolahan data dan analisis teks, khususnya di media sosial.
2. Bagi Industri Kendaraan Listrik:
   a. Mendapatkan wawasan mengenai persepsi publik terhadap kendaraan listrik, yang dapat digunakan untuk strategi pemasaran dan pengembangan produk.
   b. Mengidentifikasi isu atau masalah yang dihadapi pengguna kendaraan listrik melalui sentimen negatif, yang dapat membantu dalam perbaikan layanan dan produk.

1.5 Batasan Masalah

1. Data yang dianalisi adalah data dari media sosial *Twitter* dengan Bahasa Indonesia.
2. Data yang digunakan merupakan data sekunder yagn didapat dari GitHub - ryasakbar060/Dataset-Komentar-Kendaraan-Listrik.
3. Sentimen analisis dibagi menjadi 3 klasifikasi, yaitu positif, negatif, dan netral.

1.6 Sistematika penulisan

Sistematika penulisan skripsi ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah, dan sistematika penulisan

BAB II KAJIAN LITERATUR

Pada bab ini membahasan mengenai dasar-dasar teori yang digunakan dalam penelitian, seperti definisi dari *Text Mining, Text Preprocessing*, analisis sentimen, *Information Gain, naive bayes*, evaluasi, dan metode *waterfall*.

BAB III METODOLOGI PENELITIAN

Pada bab ini membahas mengenai tahapan yang dilakukan pada penelitian. Setiap rencana tahapan penelitian dideskripsikan dengan rinci dengan berfokus pada satu kerangka kerja. Di akhir bab berisi perancangan menejemen proyek pada pelaksanaan penelitian

1.7 Kesimpulan

Dalam penelitian ini, analisis sentimen pada media sosial *Twitter* terhadap penggunaan kendaraan listrik dilakukan menggunakan metode *Naïve Bayes Classifier* dan *Information Gain*. Tujuan utama penelitian adalah untuk mengidentifikasi sentimen pengguna Twitter terhadap kendaraan listrik, mengimplementasikan metode *Naïve Bayes Class*ifier, dan menganalisis efektivitas *Information Gain* dalam meningkatkan akurasi klasifikasi sentimen.

Dengan demikian, penelitian ini tidak hanya memberikan pemahaman yang lebih baik mengenai sentimen pengguna terhadap kendaraan listrik, tetapi juga menyediakan wawasan yang berguna bagi industri kendaraan listrik dalam merancang strategi pemasaran dan pengembangan produk. Selain itu, penelitian ini juga memberikan kontribusi terhadap pengembangan teknologi analisis sentimen, yang dapat bermanfaat bagi berbagai aplikasi lainnya dalam pemrosesan bahasa alami.

BAB II
KAJIAN LITERATUR

2.1 Text Mining

Text mining mempunyai pengertian yang sama dengan data *mining*, namun tidak untuk beberapa metode dan data yang dikelola nya seperti data teks yang tidak terstruktur, terstruktur sebagian, maupun terstruktur seperti teks email, teks HTML, maupun teks komentar dari berbagai sumber. *Text Mining* merupakan pengetahuan di *database* dalam bentuk tekstual atau *knowledge discovery in textual database* (KDT), merupakan pencarian data-data yang berbentuk teks yang merupakan ketertarikan terhadap pengetahuan yang baru dibuat, diartikan sebagai bagian dari proses pencarian data teks yang sebelumnya tidak diketahui (Firdaus, & Istalma, 2021).

2.2 Text Preprocessing

Text Preprocessing merupkan salah satu bagian dalam proses *Text Mining*, proses mengolah teks yang bertujuan untuk mengubah bentuk dokumen menjadi data yang terstruktur sesuai dengan kepentingannya agar bisa diproses lebih lanjut pada proses text mining. Langkah *preprocessing* dalam klasifikasi bertujuan untuk memaksimalkan akurasi dalam klasifikasi data. (Ridwansyah, 2022). Ada beberapa proses dalam *Text Preprocessing* sebagai berikut (Rahman Isnain dkk. 2021).

2.2.1 *Case Folding*

Proses *Case Folding* melibatkan pemisahan setiap kata dari teks, dengan mengubah semua huruf menjadi huruf kecil. Hasil dari proses *Case Folding* adalah kata yang tidak mengandung karakter selain huruf kecil.

2.2.2 *Cleansing*

Cleansing merupakan proses yang bertujuan untuk menghapus berbagai informasi yang tidak diperlukan dalam analisis sentimen, seperti symbol, angka, tanda baca, link (http, https, pic.twitter), *hashtag*, *username*, dan *retweet*.

2.2.3 *Tokenizing*

Tekonizing adalah proses pembagian suatu teks besar menjadi beberapa bagian, yang dapat berupa kalimat. Kalimat yang dihasilkan kemudian dibagi menjadi kata-kata.

2.2.4 *Normalization*

Normalisasi adalah proses yang bertujuan untuk memperbaiki kata-kata yang mungkin memiliki kesalahan penulisan atau pengejaan, serta kata-kata yang dituliskan dengan singkatan.

2.2.5 *Stemming*

Stemming adalah proses untuk mengubah kata-kata hasil *filtering* menjadi kata dasar dengan cara menghilangkan semua imbuhan, baik itu awalan, akhiran, atau sisipan. Kombinasi awalan dan akhiran pada kata turunan juga harus dihilangkan. *Stemming* digunakan untuk menyederhanakan kata-kata dari bentuk asli yang masih mengandung imbuhan menjadi kata dasar (Ulgasesa dkk. 2021).

2.2.6 *Stopword Removal*

Stopword merupakan proses untuk menghilangkan kata-kata yang dianggap tidak penting dari hasil Tokenisasi. *Stopword* digunakan dengan memeriksa setiap kata dari hasil *Case Folding*, dan jika kata tersebut terdapat dalam daftar *stopword*, maka kata tersebut dihapus (Nofiyani, & Wulandari. 2021).

2.3 Analisis Sentimen

Analisis sentimen merupakan proses yang dilakukan untuk memahami, menganalisis dan mengklasifikasikan informasi yang terkandung dalam suatu teks dengan analisis teks. Analisis sentimen digunakan untuk mengevaluasi dan memahami opini yang terdapat dalamn teks dari berbagai sumber (Ernwati dkk. 2023). Analisis sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seorang, apakah beropini positif, negatif atau netral.

2.4 TF-IDF

TF-IDF adalah salah satu teknik dalam pengolahan teks untuk memberikan bobot pada kata-kata dalam sebuah data. TF-IDF bertujuan untuk mengidentifikasi fitur yang paling penting dalam suatu data. *Term Frequency* (TF) adalah nilai frekuensi kemunculan suatu kata dalam sebuah dokumen. Fitur yang sering muncul dalam suatu data, seperti kata hubung atau kata umum cenderung memiliki nilai TF yang tinggi, tetapi tidak memiliki makna yang penting dalam data tersebut. Oleh karena itu, dibutuhkan teknik lain, yaitu *Inverse Document Frequency* (IDF) yang memberikan bobot pada kata-kata yang jarang muncul di dokumen(Wati dkk. 2023). Nilai TF dan IDF dapat dihitung dengan menggunakan rumus:

$$TF = \frac{jumlah\ kemunculan\ fitur\ dalam\ teks}{jumlah\ kata\ dalam\ teks} \quad \text{(II-1)}$$

$$IDF = log\frac{N}{m} \quad \text{(II-2)}$$

Keterangan

N adalah jumlah dokumen dalam kumpulan dokumen

m adalah jumlah dokumen yang mengandung kata tersebut.

Setelah nilai TF dan IDF diperoleh, nilai TF-IDF dapat dihitung dengan.

$$TF - IDF = TF \; x \; IDF \qquad \text{(II-3)}$$

Fitur yang memiliki nilai TF-IDF tinggi dianggap penting dan memberikan kontribusi yang lebih besar dalam menentukan topik dokumen atau kumpulan dokumen tersebut.

2.5 Seleksi fitur

Seleksi fitur adalah suatu metode yang digunakan untuk mengurangi dimensi atribut dengan memilih sejumlah atribut yang dianggap relevan untuk proses klasifikasi. Terlalu banyak atribut, terutama jika tidak relevan, dapat mempengaruhi efektivitas pengenalan pola selama tahap pemrosesan menggunakan metode yang telah ditentukan (Nur dkk. 2022). Tujuan utama dari seleksi fitur adalah untuk memilih fitur terbaik dari suatu kumpulan fitur.

2.6 *Information Gain*

Information Gain biasanya digunakan untuk menilai dan menentukan atribut mana yang dianggap berpengaruh terhadap kelas. *Information Gain* digunakan untuk menghasilkan atribut yang berkaitan terhadap kelas target, karena setiap atribut memiliki nilai dan dipilih dengan nilai yang terbaik (Nur dkk. 2022).

Information Gain dihitung berdasarkan pengaruh fitur terhadap keseragaman entropy pada data yang dibagi jadi subdata dengan nilai fitur tertentu. Keseragaman entropy dihitung pada data sebelum dipecah dengan persamaan II-1 dan pada data setelah dipecah dengan persamaan II-2 berikut ini.

$$entropy\ (S) = \sum_{i=i}^{k} (P_i) log_2 (P_i) \qquad \text{(II-4)}$$

Keterangan:

P_i : probabilitas kelas i dalam Data.

Entropy (S) : nilai entropy data S

K : jumlah kelas.

$$Entropy\ (S,A) = \sum_{i=1}^{v} (\frac{Sv}{S} * Entropy(Sv)) \quad \text{(II-5)}$$

Keterangan

v : seluruh nilai yang mungkin dari atribut A.

S_v : subset dari S dimana atribut A bernilai v.

Entropy(S.A) : nilai entropy setelah dipisah berdasarkan A.

Nilai dari information Gain dapat dihitung dengan persamaan berikut

$$Gain(S,A) = Entropy(S) - Entropy(S,A) \quad \text{(II-6)}$$

Keterangan:

Gain (S,A) : nilai *information gain*.

2.7 *naive bayes Classifier*

Fungsi lain dari *Data Mining* adalah untuk mengklasifikasikan data, yaitu mengelompokkan data ke dalam satu atau beberapa kelas yang telah ditentukan. Salah satu metode yang digunakan dalam klasifikasi data adalah Naïve Bayes, sebuah metode yang menggunakan perhitungan probabilitas yang dikembangkan oleh ilmuwan Inggris Thomas Bayes. Naïve Bayes bekerja dengan memprediksi probabilitas kejadian di masa depan berdasarkan data dari kejadian-kejadian

sebelumnya (Darwis, Siskawati, & Abidin).

Teorema Bayes merupakan torema yang mengacu pada konsep probabilitas bersyarat. Teorema Bayes dapat dituliskan dengan:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad \text{(II-7)}$$

Keterangan

P(A|B) : Probabilitas dari kelas A yang diberikan Fitur B

P(B|A) : Probabilitas mengamati fitur B dari kelas A

P(A) : Probabilitas dari kelas A

P(B) : Probabilitas fitur B

Pada *naive bayes*, setiap tweet direpresentasikan dalam atribut (a1, a2, a3, ..., an) dimana a1 merupakan kata pertama, a2 merupakan kata kedua, a3 merupakan kata ketiga dan seterusnya, sedangkan V merupakan himpunan kelas. Saat proses klasifikasi, *Naïve Bayes* akan menghasilkan kategori / kelas yang paling tinggi probabilitasnya (V_{MAP}) dengan memasukkan atribut (a1, a2, a3, ..., an). Untuk persamaan V_{MAP} dapat ditulis dengan:

$$V_{MAP} = \underset{v_j \epsilon v}{argmax} P(v_j|a1, a2, a3, \dots, a_n) \quad \text{(II-8)}$$

Dengan menggunakan teorema Bayes, maka dapat ditulis menjdi

$$V_{MAP} = \underset{vj \epsilon v}{^{argmax}} \frac{P(v_j|a1, a2, a3, \dots, a_n)P(v_j)}{P(a1, a2, a3, \dots, a_n)} \quad \text{(II-9)}$$

P(a1, a2, a3, …, a_a) memiliki nilai yang konstan pada semua v_j sehingga dapat ditulis dengan

$$V_{MAP} = \underset{vj \epsilon v}{^{argmax}} P(v_j|a1, a2, a3, \dots, a_n)P(v_j) \quad \text{(II-10)}$$

Keterangan :

V_{MAP} : Semua Kategori yang diujikan

$\underset{vj \epsilon V}{^{argmax}}$: mencari nilai v_j diantara semua kemungkinan kelas V yang memaksimalkan persamaan

$P(v_j|a1, a2, a3, \dots, a_n)$: probabilitas kelas v_j yang diberi fitur $a1, a2, a3, \dots, a_n$

$P(v_j)$: Probabilitas dari v_j

$P(a1, a2, a3, \dots, a_n)$: probabilitas dari fitur

Naïve bayes menyederhanakan hal ini dengan mengasumsikan bahwa setiap kategori, setiap atribut bebas bersyarat satu dengan yang lain. Maka:

$$P(a_1, a_2, a_3, \dots, a_n|V_j) = \prod_i P(a_i|v_j) \quad \text{(II-11)}$$

Jika kedua persamaan diatas digabungkan maka akan mengahasilkan persamaan :

$$V_{MAP} = \underset{v_j \epsilon V}{^{argmax}} P(v_j) \times \prod_i P(a_i|v_j) \quad \text{(II-12)}$$

$P(v_j)$ dan probabilitas kata a_i untuk setiap kategori yang dihitung pada saat *train*ing dapat dirumuskan dengan

$$P(v_j) = \frac{docs_j}{training} \quad \text{(II-13)}$$

$$P(a_i|v_j) = \frac{n_1 + 1}{n + kata} \quad \text{(II-14)}$$

Keterangan:

$docs_j$: banyaknya dokumen pada kategori *j*

$training$: banyaknya fitur yang digunakan dalam proses *train*ing

$P(a_i|v_j)$: probabilitias a_i pada kategori v_j

n : banyaknya kata yang muncul pada kategori v_j

n_1 : n_i adalah jumlah kemunculan kata a_i pada kategori v_j

$kata$: jumlah kata unik pada data *training*

2.8 Evaluasi

Tahap evaluasi bertujuan untuk menilai kinerja algoritma klasifikasi yang digunakan dalam penelitian (Tanggreani, & Sitokdana, 2022). Kemampuan prediksi diukur berdasarkan nilai dari Confusion Matrix, akurasi, *precision*, *recall*, dan F1-*Score* (Ramadhani & Suryono, 2024).

Tabel II- 1 *Tabel Confusion Matrix*

		Prediction *Class*		
		Positif	Negatif	Netral
Actual *Class*	Positif	A	B	C
	Negatif	D	E	F
	Netral	G	H	I

Matriks tersebut memiliki nilai yang dijadikan acuan dalam perhitungan, yaitu:

a. ‘A’ adalah jumlah contoh yang sebenarnya positif dan diprediksi positif.
b. ‘B’ adalah jumlah contoh yang sebenarnya positif tetapi diprediksi negatif.
c. ‘C’ adalah jumlah contoh yang sebenarnya positif tetapi diprediksi netral.
d. ‘D’ adalah jumlah contoh yang sebenarnya negatif tetapi diprediksi positif.
e. ‘E’ adalah jumlah contoh yang sebenarnya negatif dan diprediksi negatif.
f. ‘F’ adalah jumlah contoh yang sebenarnya negatif tetapi diprediksi netral.
g. ‘G’ adalah jumlah contoh yang sebenarnya netral tetapi diprediksi positif.
h. ‘H’ adalah jumlah contoh yang sebenarnya netral tetapi diprediksi negatif.
i. ‘I’ adalah jumlah contoh yang sebenarnya netral dan diprediksi netral.

Untuk menentukan model terbaik, penelitian ini menganalisis nilai *Classification report*. Dalam *Classification report*, terdapat nilai *accuracy*, *precision*, *recall*, dan F1-*Score* yang digunakan sebagai acuan untuk memilih model terbaik.

Untuk *accuracy* digunakan untuk mengetahui seberapa akurat sistem dapat melakukan klasifikasi. *Accuracy* dapat dihitung dengan persamaan (Husada &

Paramita, 2021):

$$Accuracy = \frac{A+E+I}{A+B+C+D+E+F+G+H+I} \quad \text{(II-15)}$$

Precision berfungsi untuk menghitung berapa banyak prediksi pada kelas yang benar dari semua prediksi pada kelas. Untuk menghitung *precision* dapat menggunakan.

$$Precision\ positif = \frac{A}{A+D+G} \quad \text{(II-16)}$$

$$Precision\ negatif = \frac{E}{B+E+H} \quad \text{(II-17)}$$

$$Precision\ netral = \frac{I}{C+F+I} \quad \text{(II-18)}$$

$$Precision - avg = \frac{Precision\ Positif + precision\ negatif + precision\ netral}{3} \quad \text{(II-19)}$$

Recall berfungsi untuk mengukur berapa banyak kelas yang berhasil diidentifikasi dari semua kelas yang sebenarnya. Untuk menghitung *recall* dapat menggunakan.

$$Recall\ positif = \frac{A}{A+B+C} \quad \text{(II-20)}$$

$$Recall\ negatif = \frac{E}{D+E+F} \quad \text{(II-21)}$$

$$Recall\ netral = \frac{I}{G+H+I} \quad \text{(II-22)}$$

$$Recall - avg = \frac{Recall\ Positif + Recall\ negatif + Recall\ netral}{3} \quad \text{(II-23)}$$

F1-*Score* berguna untuk memberikan satu nilai untuk menyeimbangkan *recall* dan *precision*. untuk menghitung f1-*score* dapat menggunakan.

$$\text{F1} - Score\ positif\ = 2\ x\ \frac{Precision\ positif\ x\ recall\ positif}{Precision\ positif + recall\ positif} \quad \text{(II-24)}$$

$$\text{F1} - Score\ negatif = 2\ x\ \frac{Precision\ negatif\ \ x\ recall\ negatif}{Precision\ negatif\ \ + recall\ negatif} \quad \text{(II-25)}$$

$$\text{F1} - Score\ netral\ = 2\ x\ \frac{Precision\ netral\ \ x\ recall\ netral}{Precision\ netral\ \ + recall\ netral} \quad \text{(II-26)}$$

$$\text{F1} - Score\ - avg = \frac{\text{F1} - Score\ Positif + \text{F1} - Score\ negatif + \text{F1} - Score\ netral}{3} \quad \text{(II-27)}$$

2.9 Metode Waterfall

Metode penelitian yang digunakan dalam penelitian ini merupakan metode *waterfall*. Karena dalam pengembangannya bersifat sistematik dan sekuensial. Selain itu metode *waterfall* dilakukan secara berurutan dan berkelanjutan. Berikut beberapa tahapan-tahapan yang dilakukan dalam penelitian ini: *requirement*, *system design*, *Coding & Testing*, *implementation*, *maintenance* (Mahardika dkk. 2023).

Gambar II- 1 Metode Waterfall

2.10 Kesimpulan

Secara keseluruhan, kajian literatur yang dibahas memberikan pemahaman yang komprehensif mengenai konsep-konsep utama dan metodologi yang digunakan dalam penelitian ini. Teori sentimen, analisis sentimen, dan metode *Naïve Bayes Classifier* yang diperkuat dengan seleksi fitur menggunakan *Information Gain* menjadi kerangka dasar untuk analisis sentimen terhadap kendaraan listrik di Indonesia. Dengan pemahaman ini, penelitian ini diharapkan dapat memberikan kontribusi yang berarti dalam mengidentifikasi opini publik dan mendukung pengembangan kebijakan serta strategi pemasaran yang efektif untuk kendaraan listrik.

BAB IV
PENGEMBANGAN PERANGKAT LUNAK

4.1 Metode Pengembangan Perangkat Lunak

Pada penelitian ini, metode penelitian perangkat lunak yang digunakan adalah metode waterfall yang memiliki 5 tahapan, yaitu *requirement, system design, coding and testing, implementasi dan maintenance*.

4.1.1 *Requirement*

Pada tahap ini, hal yang perlu dilakukan adalah untuk menganalisis hal apa saja yang diperlukan untuk mengatasi masalah-masalah yang bekaitan dengan kebutuhan sistem. Tahap ini dilakukan dengan melakukan analisis terhadap perangkat keras, perangkat lunak, dan data.

4.1.1.1 Analisis Perangkat Keras

Dalam melakukan pengambangan perangkat lunak, diperlukan sebuah perangkat keras yang digunakan sebagai alat dalam membuat sistem. Perangkat keras yang digunakan dalam mengembangkan perangkat lunak adalah sebagai berikut:

Processor : AMD Ryzen 5 4500U with Radeon Graphics

RAM : 20 GB

SSD : 512 GB

4.1.1.2 Analisis Perangkat Lunak

Dalam pengembangan perangkat lunak ini juga memerlukan perangkat lunak lain. Dalam penelitian ini perangkat lunak yang digunakan adalah sebagai berikut:

Sistem Operasi : Windows 11 64-bit

Bahasa Pemrograman : *Python*

Code Editor : *Visual Studio Code*

Library : *Pandas, sklearn, streamlit, re, nltk, sastrawi, string, os, pickle, io*

4.1.1.3 Analisis Data

Pada penelitian ini, data yang digunakan merupakan data sekunder. Data yang digunakan dalam ini diambil dari https://github.com/ryasakbar060/Dataset-Komentar-Kendaraan-Listrik. Untuk rincian data, dapat dilihat dalam tabel V-1 berikut.

Tabel IV- 1 Rincian Dataset

Klasifikasi	Jumlah
Positif	1903
Negatif	1603
Netral	668
Total	4174

4.1.2 *System Design*

Pada tahap ini hal yang dilakukan yaitu membuat u*secase, scenario usecase, Activity Diagram, sequence Diagram, dan Class Diagram.*

4.1.2.1 *Use case Diagram*

Use case Diagram menggambarkan *user* sebagai aktor terhadap sistem sehingga tercipta sebuah interaksi. *Diagram* usecase untuk penelitian ini digambarkan seperti dibawah ini.

Gambar IV- 1 *Use case Diagram*

4.1.2.2 *Use case* Skenario

Use case skenario merupakan penggambaran tindakan spesifik terhadap actor dan sistem yang dibuat pada Gambar IV-1 Berikut ini skenario penerapan dalam format tabel.

Tabel IV- 2 Skenario *preprocess data*

Identifikasi	
Nomor	1
Nama	Melakukan *Preprocess* Data dan analisis sentimen
Actor	Pengguna
Tujuan	Pengguna memasukkan data yang akan dilakukan *preprocessing*. Setelah itu, jika data sudah selesai diproses akan muncul *radio button* untuk memilih analisis dengan *Naive Bayes* atau *Information Gain dan Naive Bayes*
Deskripsi	pada *use case* ini pengguna akan memasukkan data yang akan digunakan dalam preprocess
Kondisi awal	Belum ada data yang diinput
Skenario utama	
Pengguna	Sistem
	1. Menampilkan *interface* awal untuk menginput file csv dan tombol '*browse file*'

2. Menekan tombol '*browse files'*	
	3. Menampilkan *dialog box* untuk mencari *file*
4. Memilih *file* berekstensi CSV	
5. Menekan tombol '*open'* pada *dialog box*	
	6. Membaca *file*
	7. Menampilkan isi *file* CSV
8. Menekan tombol '*Preprocess Data'*	
	9. Menampilkan data hasil *preprocessing text*
	10. Manmpilkan *radio button* untuk memilih metode analisis sentimen.
11. Memilih metode analisis	
12. Menekan tombol 'Analisis'	
	13. Menampilkan hasil evaluasi berdasarkan metode yang dipilih
Kondisi Akhir	Menampilkan hasil evaluasi berdasarkan metode yang dipilih
Skenario alternatif	

Pengguna	Sistem
	1. Menampilkan interface awal dan dan button '*browse file'*
2. Menekan tombol '*browse files'*	
	3. Menampilkan *dialog box* untuk mencari *file*
4. Memilih *file* yang tidak berekstensi CSV atau tidak menginput salah satu file atau file CSV yang dimasukkan tidak memiliki kolom yang dibutuhkan	
	5. Menampilkan pesan error
Kondisi Akhir	Sistem menampilkan pesan kesalahan input

Tabel IV- 3 *Skenario Analisis sentiment teks*

Identifikasi	
Nomor	2
Nama	Analisis sentiment teks
Actor	Pengguna
Tujuan	Pengguna memasukkan teks yang akan diklasifikasi
Deskripsi	pada use case ini pengguna akan memasukkan teks yang akan digunakan dalam proses perhitungan

Kondisi awal	Belum ada text yang diinput
Skenario utama	
Pengguna	Sistem
	1. Menampilkan interface awal dan kolom untuk input text
2. Memasukkan text	
3. Menekanekan tombol ‘Analisis’	
	4. Melakukan preprocessing text
	5. Melakukan klasifikasi terhadap text berdasarkan model yang telah dilatih
	6. Menampilkan hasil preprocessing text dan klasifikasi text
Kondisi Akhir	interface menampilkan hasil preprocessing text dan klasifikasi text
Skenario alternatif	
Pengguna	Sistem
	1. Menampilkan interface awal dan kolom untuk input text

2. Menekan tombol ‘analisis’ tanpa memasukkan teks	
	3. Menampilkan pesan error ‘silahkan masukkan text’
Kondisi akhir	Interface menampilkan pesan error ‘silahkan masukkan text’

4.1.2.3 Activity *Diagram*

Activity *Diagram* adalah *Diagram* yang menggambarkan alur program yang perankan oleh actor dan dengan sistem. Alur program digambarkan dengan *Diagram* dibawah.

Gambar IV- 2 *Activity diagram Preprocess Data dan Sentiment Analisys*

Gambar IV- 3 *Activity Diagram Analisis sentimen teks*

4.1.2.4 Sequence *Diagram*

Sequence Diagram bertujuan untuk mengetahui bagaimana suatu sistem berinteraksi satu dengan yang lain dengan urutan tertentu. Berikut adalah gambaran dari *sequence Diagram* yang sedang dikembangkan peneliti.

Gambar IV- 4 *Analisis Sentimen Naïve Bayes*

Gambar IV- 5 Analisis Sentimen Naive Bayes dan Infotmation Gain

Gambar IV- 6 *Sequence Diagram sentimen analisis teks*

4.1.2.5 *Class Diagram*

Class diagram merupakan suatu *diagram* dalam UML yang diperlukan untuk menggambarkan struktur dari suatu sistem. *Diagram* ini menggambarkan kelas yang dimiliki sistem serta hubungan diantaranya. Berikut ini adalah *diagram* kelas untuk perangkat lunak yang sedang dikembangkan.

Gambar IV- 7 *Class Diagram*

4.1.3 *Coding and Testing*

Pada tahap ini dibagi menjadi dua tahap yaitu tahap *coding* dan tahap *testing*..

4.1.3.1 *Coding*

Pada tahap *coding*, hasil analisis dan desain sistem yang sudah dibuat dalalm tahap sebelumnya diterjemahkan kedalam bahasa pemrograman

4.1.3.1.1 Preproses Data

Disini, merupakan tahap awal yang dimana data yang akan digunakan, terlebih dahulu dilakukan *preprocess*. Terdapat beberapa proses yang perlu dilakukan, diantaranya sebagai berikut.

4.1.3.1.1.1 Case Folding

Dalam proses ini, semua huruf dalam data diubah menjadi *lower case*. Fungsi ini menerima argumen sebuah teks yang akan diproses dan mengembalikan teks

yang sudah diporoses.

Gambar IV- 8 Fungsi *Case Folding*

4.1.3.1.1.2 Cleansing

Dalam proses ini, semua hal, atribut, atau elemen yang tidak diperlukan atau tidak berpengaruh akah dihapus dan digantikan dengan space (‘ ‘).

Gambar IV- 9 Fungsi *Cleansing*

4.1.3.1.1.3 Tokenizing

Dalam proses ini, dilakukan pemotongan terhadap kalimat berdasarkan kata yang menyusunnya menjadi kata tunggal. Fungsi ini menerima argumen sebuah teks dan memecahnya menjadi token.

Gambar IV- 10 Fungsi *Tokenizing*

4.1.3.1.1.4 Normalization

Dalam proses ini, mengkonversikan kata asing, singkatan, dan kata baku menjadi sesuai dengan EYD. Fungsi ini menerima argument token, dan ‘normal_csv’ yang merupakan kamus untuk normalisasi kata.

Gambar IV- 11 Fungsi *Normalization*

4.1.3.1.1.5 Stemming

Dalam proses ini, mengubah token yang diinputkan menjadi kata dasarnya.

Gambar IV- 12 Fungsi *Stemming*

4.1.3.1.1.6 Stopword remove

Dalam proses ini, merupakan proses dimana program menghilangkan kata yang tidak berhubungan atau kata yang tidak relevan dengan topik. Fungsi ini menerima input sebuah token dan ‘stopword_csv’ sebagai *dataset* dari stopwords.

Gambar IV- 13 Fungsi *Stopword Removal*

4.1.3.1.1.7 Preprocess text

Pada proses ini, semua proses yang dibuat sebelumnya disimpan sebelum dilakukannya proses. Fungsi ini menerima 3 argumen, yaitu teks yang akan diproses, ‘normal_csv’ sebagai kamus untuk melakukan normalisasi, dan ‘stopword_csv’ sebagai *dataset* dari *stopword*. Kemudian program ini akan menjalankan proses secara berurutan, dan menggabungkan kembali token-token menjadi sebuah *string* dengan *space* sebagai pemisah.

Gambar IV- 14 Fungsi *Preprocess data*

4.1.3.1.2 Naïve Bayes Classifier

Dalam *file* ini, dilakukan proses klasifikasi dengan metode naïve bayes. Dalam melaskukan klasifikasi terdapat proses-proses yang harus dilakukan, diantaranya sebagai berikut.

4.1.3.1.2.1 Split data

Dalam proses ini, membagi data menjadi dua, yaitu data *train* dan data *test*.

Gambar IV- 15 Fungsi *Split data*

Didalam fungsi *split data*, data dibagi menjadi 2, yaitu *data[‘full_text’]* sebagai ‘X’ atau fitur, dan *data[‘sentimen’]* sebagai ‘y’ atau label. Kemudain data

dibagi menjadi data *test* dan *train* dengan rasio 8:2. Fungsi ini mengembalikan variabel yange berisi set *train* dan *test* untuk fitur dan label.

4.1.3.1.2.2 Train model

Dalam proses ini, dilakukan proses untuk melatih *machine learning* untuk melakukan klasifikasi teks.

Gambar IV- 16 Fungsi *Train Model*

Fungsi ini menerima dua argument, 'X_*train*', dan 'y_*train*'. Berikutnya dilakukan inisialisasi '*CountVectorizer'* yang mana ini adalah alat dari *library* '*sklearn'* yang akan mengubah setiap kata unik dalam data menjadi vektor yang menghitung frekuensi kemunculan kata dalam data. Kemudian '*fit_transform'* yang digunakan pada 'X_*train*' bertujuan untuk mempelajari fitur dari data *train* dan mengubah kata tersebut menjadi vektor. Setelah itu, model *naïve bayes* dapat dilatih pada data yang sudah diubah menjadi vektor fitur dan lebel ('y_*train*'). Terakhir mengembalikan objek berupa 'model' sebagai model dari *naïve bayes* yang sudah dilatih dan '*CountVectorizer'* yang digunakan untuk mengubah teks menjadi vektor fitur.

4.1.3.1.2.3 Test Model

Dalam proses ini, dilakukan pengujian dari model yang sudah dilatih di *train* model.

Gambar IV- 17 Fungsi *Test Model*

'*test*_model' menerima tiga argument, yaitu 'model' sebagai model *naïve bayes* yang sudah dilatih, 'vectorizer' adalah 'CountVectorizer' yang sudah dilatih, dan 'X_*test*' sebagai data uji.

Pertama '*vektorizer.transform(*X_*test)*' digunkan untuk mengubah data uji menjadi vektor fitur. Kemudian 'model' digunakan untuk memprediksi label 'y_*pred'* untuk data uji yang diubah menjadi vektor fitur. Dan terakhir mengembalikan prediksi 'y_*pred'*.

4.1.3.1.2.4 Evaluasi

Proses ini digunakan untuk mengevaluasi kinerja model.

Gambar IV- 18 Fungsi *Evaluasi*

Fungsi ini menerima dua argument, 'y_*test*' sebagai label yang sebenarnya, dan 'y_*pred'* sebagai label yang diprediksi model. Kemudian menghitung nilai '*accuracy', 'report', 'confusion matrix', 'precision', 'recall' dan 'f1-score'*. Dan mengembalikan hasil dari perhitungan.

4.1.3.1.2.5 Sentimen Analisis

Ditahap ini, digunakan untuk menjalankan semua alur kerja mulai dari membaca data, membagi data, melatih model, menguji model, dan evaluasi model.

Gambar IV- 19 Fungsi *Main Program*

Fungsi ini menerima '*file*_csv' yang mana ini merupakan data yang akan dilakukan proses analisis sentiment. Dimulai dngan membaca dan menyimpan data dalam *data frame* "data". kemudian membagi data dengan fungsi "split_data" yang mengembalikan 'X_*test*', 'X_*train*', 'y_*test*' dan 'y_*train*'. Kemudian model dilatih dengan data *train* 'X-*test*' dan 'y_*test*'. Lalu menguji model 'X_*test*" dengan model yang sudah dilatih. Kemudian menggunakan 'evaluate_model' untuk mengevaluasi hasil dan terakhir mengembalikan nilai hasil evaluasi kinerja model.

4.1.3.1.3 Information Gain dan Naïve Bayes

Dalam proses ini, setiap fitur dalam data dihitung nilai Gain. Dalam melakukan perhitungan *information gain* terdapat beberapa proses, diantaranya adalah sebagai berikut.

4.1.3.1.3.1 Hitung Information Gain

Fungsi ini digunakan untuk menghitung skor information gian untuk tiap fitur.

Gambar IV- 20 Fungsi *Hitung Information Gain*

Fungsi ini menerima dua argument, ‘texts’ dan ‘label’. Pertama ‘CountVectorizer’ digunakan untuk mengubah teks menjadi vektor fitur. Kemudian ‘x = fit_transform(texts)’ digunakan untuk mempelajari fitur dari teks dan kemudian mengubah teks menjadi vector. Inisiasi ‘y’ sebagai label.

Berikutnya ‘*mutual_info_Class*if(x,y)’ untuk menghitung skor *information gain* untuk setiap fitur ‘x’ terhadap label ‘y’. hasil dari perhitungan disimpan di ‘ig_score’. ‘get_feature_names_out’ digunakan untuk mengambil nama fitur yang sesuai dengan vector fitur ‘x’.

Hasil perhitungan disimpan kedalam data frame ‘ig_results’ dengan dua kolom ‘feature’ sebagai nama fitur dan ‘Information_Gain’ sebagai skor dari *information gain*. Dan mengembalikan data ‘ig_result’yang berisi fitur dan skor tiap fitur.

4.1.3.1.3.2 *Split data*

Dalam proses ini, data dibagi menjadi 2, yaitu ‘test’ dan ‘train’

Gambar IV- 21 Fungsi *Split Data*

Didalam fungsi *split data*, data dibagi menjadi 2, yaitu *data['full_text']* sebagai 'X' atau fitur, dan *data['sentimen']* sebagai 'y' atau label. Kemudain data dibagi menjadi data *test* dan *train* dengan rasio 8:2. Fungsi ini mengembalikan empat variabel yange berisi set *train* dan *test* untuk fitur dan label.

4.1.3.1.3.3 Seleksi fitur

Fungsi ini digunakan untuk memilih fitur-fitur dari teks berdasarkan *information gain*.

Gambar IV- 22 Fungsi Seleksi Fitur *Information Gain*

Fungsi ini menerima tiga argumen, yaitu 'X_train' sebagai data latih, 'gain_csv' sebagai *file* CSV yang memiliki nama dan skor *information gain* tiap fitur dan 'fitur_gain' merupakan jumlah fitur teratas yang dipilih berdasarkan skor information gian.

Berikutnya dilakukan inisialisasi '*CountVectorizer'* yang mana ini adalah alat dari *library* '*sklearn'* yang akan mengubah setiap kata unik dalam data menjadi vektor yang menghitung frekuensi kemunculan kata dalam data. Kemudian '*fit_transform'* yang digunakan pada 'X_*train*' bertujuan untuk mempelajari fitur dari data *train* dan mengubah kata tersebut menjadi vektor.

Selanjutnya, 'info_gain' untuk membaca *file* CSV yang memiliki nilai *information gain*. Lalu mengambil daftar nama fitur dari "CountVectorizer". Kemudian mengambil dan menetapkan kolom 'Feature' dari data sebagai indeks. Mengurutkan fitur berdasarkan nilai *information gain* secara descending.

Setelah itu, membuat 'mask' untuk fitur terpilih yang menunjukkan apakah

setiap fitur dalam ‘feature_name’ termasuk kedalam fitur yang dipilih berdasarkan skor *information gain*. Lalu menerapkan ‘mask’ kedalam ‘X_train_vec’ untuk memilih fitur yang termasuk kedalam fitur yang dipilih. Terakhir mengembalikan nilai dari fitur terpilih ‘*X_train_selected’* , *mask* dari fitur yang terpilih ‘*mask’*, dan ‘*vectorizer’*.

4.1.3.1.3.4 *Train* dan *test model*

Dalam proses ini, dilakukan proses untuk melatih *machine learning* untuk melakukan klasifikasi teks, dan dilakukan pengujian dari model yang sudah dilatih di *train* model.

Gambar IV- 23 Fungsi *Train dan Test*

‘train_model’ menerima dua argumen, yaitu ‘X_train_selected’ yang merupakan data dari fitur yang sudah dipilih berdasarkan nilai *information gain*, dan ‘y_train’ sebagai label. Kemudian melatih model berdasarkan dua argumennya. Fungsi ini mengembalikan model yang sudah dilatih

‘test_model’ menerima tiga argumen, ‘model’, ‘mask’, dan ‘X_test’. Pertama, mengubah teks dalam ‘X_text’ menjadi vector fitur. Kemudian menerapkan’mask’ dalam data train sebelumnya untuk memilih fitur-fitur yang relevan dari ‘X_test_vec’. Lalu menggunakan model yang sudah dilatih untuk memprediksi label dalam data uji. Terakhir mengembalikan label prediksi ‘y_pred’.

4.1.3.1.3.5 Evaluasi

Untuk bagian evaluasi masih sama seperti pada proses ‘naïve_bayes.py’. Ditahap ini, digunakan untuk menjalankan semua alur kerja mulai dari membaca

data, membagi data, melatih model, menguji model, evaluasi model, dan menyimpan model sentimen.

Gambar IV- 24 Fungsi *Evaluasi*

Fungsi ini menerima dua argument, 'y_*test*' sebagai label yang sebenarnya, dan 'y_*pred'* sebagai label yang diprediksi model. Kemudian menghitung nilai '*accuracy', 'report', 'confusion matrix', 'precision', 'recall' dan 'f1-score'*. Dan mengembalikan hasil dari perhitungan.

4.1.3.1.3.6 Sentimen Analisis

Gambar IV- 25 Fungsi *analisis sentimen*

Fungsi '*sentiment_analysis'* menerima dua argumen, yaitu 'data' dan 'gain_df' yang berisi nilai *Information Gain* untuk fitur-fitur dalam dataset. Pertama, data dibagi menjadi *training* dan *testing* menggunakan'*split_data*'. Inisialisasi variabel untuk menyimpan performa model terbaik selama proses iterasi. '*fitur_gain'* dihitung sebagai panjang '*gain_df'* yang dibulatkan ke bawah ke ratusan terdekat. Variabel seperti *best_accuracy, best_model, best_vectorizer, best_mask, best_precision, best_recall, best_f1, best_conf_matrix, best_fitur, dan best_epoch* diinisialisasikan untuk menyimpan nilai terbaik dari masing-masing metrik.

Fungsi kemudian memasuki perulangan *while* yang berlanjut selama '*fitur_gain'* sama atau lebih dari 100. Dalam setiap iterasi, fitur-fitur dipilih menggunakan fungsi '*select_features'* berdasarkan '*fitur_gain'* saat ini, model dilatih menggunakan data pelatihan dengan '*train_model'*, prediksi dibuat menggunakan data pengujian dengan '*test_model'*, dan model dievaluasi

menggunakan fungsi '*evaluate_model*' yang mengembalikan metrik performa seperti *accuracy, confusion matrix, precision, recall dan f1-score*. Jika akurasi model saat ini lebih baik dari yang sebelumnya, model dan metrik tersebut disimpan sebagai yang terbaik. Setelah setiap iterasi, jumlah '*fitur_gain*' dikurangi 100. Jika fitur_gain menjadi kurang dari 100, diset menjadi 100 dan perulangan dihentikan.

Setelah perulangan selesai, nilai evaluasi terbaik dicetak. Model terbaik, bersama dengan vectorizer dan mask, disimpan sebagai model untuk klasifikasi menggunakan modul pickle.

4.1.3.1.4 Klasifikasi Teks

Dalam proses ini, dilakukan klasifikasi sentimen terhadap suatu teks yang diinputkan. Dimana program melakukan preprocessing text terlebih dahulu, lalu melakukan klasifikasi terhada teks sesuai dengan model yang disimpan. Berikut adalah tahap yang diperlukan dalam proses ini.

4.1.3.1.5.1 Load Model Sentimen

Muat kembali model yang disimpan pada proses sebelumnya, beserta dengan vectorizer dan mask fitur dari *file* pickle.

Gambar IV- 26 *Load Model Senitiment*

Pertama membuka *file* 'sentimen_model.pkl' dalam mode baca biner, kemudian 'pickle.load(f)' berfungsi untuk memuat model, vectorizer, dan mask yang telah disimpan dalam *file*.

4.1.3.1.5.2 Preprocess Teks

Dalam tahap ini teks yang akan diklasifikasi dilakukan preprocessing text dengan proses sebagai berikut.

Gambar IV- 27 fungsi *case folding, cleansing, tokenizing, normalization, stemming*

Gambar IV- 28 Fungsi *stopword remove, preprocess_text*

4.1.3.1.5.3 Load Resource

Fungsi ini digunakan untuk memuat sumber daya tambahan , yang disini merupakan 'normal_csv' sebagai kamus normalisasi, dan 'stopword_csv' sebagai data dari stopword.

Gambar IV- 29 *load resource*

Didalam '*preprocess_csv'* terdapat beberapa proses yang dilakukan, yaitu membaca *file* CSV normalisasi yang diinput kedalam *'normalization_df*", kemudian *data frame* ini dikonversi menjadi kamus '*normal_csv'*. Berikutnya ada membaca *file stopword*, dimana *file* CSV yang berisi daftar *stopword* dibaca kedalam '*stop_words'*, yang kemudian dikonversi menjadi set '*stopword_csv'.*

4.1.3.1.5.4 Classify

Fungsi ini digunakan untuk melakukan klasifikasipada teks yang sudah di proses. Berikut adalah langkah dalam proses klasifikasi

Gambar IV- 30 fungsi *Classify*

Mengubah 'processed_text' menjadi matriks jumlah token dengan 'vectorizer'. Kemudian menerapkan 'mask' yang telah dimuat sebelumnya data teks yang telah diproses untuk memilih fitur-fitur yang relevan. Terakhir menggunakan

‘model’ yang telah dimuat sebelumnya untuk memprediksi sentiment teks yang sudah diproses. Kemudian mengembalikan hasil dari prediksi sentiment ‘prediction’ dari teks.

4.1.3.1.5.5 Streamlit

Disini menggunakna streamlit untuk mengimplementasikan caching data pada fungsi *‘preprocess_text_cached ‘* dan *’Classify_text_cached’*.

Gambar IV- 31 *Streamlit*

Dekorator ‘@st.cache_data’ digunakan untuk mendekorasi fungsi ‘preprocess_text_cached’. Ini adalah fungsi bawaan dari Streamlit yang digunakan untuk menyimpan hasil dari fungsi di cache. Lalu memanggil fungsi ‘load_resource’ dari module ‘*Class*ify’ untuk memuat kamus normalisasi dan stopwords. Fungsi ‘preprocess_text’ dari modul ‘*Class*ify’ dipanggil dengan parameter ‘text’ yang akan diproses, ‘normalization_dict’, dan ‘stop_words’ yang telah dimuat

Dekorator ‘@st.cache_data’ juga digunakan di sini untuk mendekorasi fungsi ‘*Class*ify_text_cached’. Fungsi ini bertugas untuk mengklasifikasikan sentimen dari teks yang telah diproses tersebut.

4.1.3.2 *Testing*

Pada tahap *test*ing, hasil dari tahap *coding* dilakukan *test*ing dengan metode *blackbox*. Testing ini dilakukan untuk memastikan bahwa sistem berjalan dengan baik dan sesuai dengan sebagaimana seharusnya.

4.1.4.2.1. Preprocess data dan Analisis sentimen

Tabel IV- 4 Rencana Pengujian *Preprocess Data dan Analisis Sentimen*

Kode	Skenario Test

A1	Menginput *file* CSV dengan menekan tombol '*browse file*' untuk data dan menekan "Preprocess Data"
A2	Menginput *file* yang bukan CSV
A3	Memilih salah satu metode analisis dan menekan tombol "Analyze"

4.1.4.2.2. Klasifikasi teks

Tabel IV- 5 Rencana Pengujian Klasifikasi teks

Kode	Skenario Test
D1	Menginput teks yang akan dianaalisis, dan menekan tombol 'analisis'
D2	Tidak ada teks yang diinput

4.1.4 Implementasi

Tahap implementasi di sini mencakup serangkaian aktivitas yang dilakukan untuk menerapkan dan menguji sistem. Implementasi ini terdiri dari dua langkah, yaitu uji coba implementasi, dan pelatihan pengguna.

4.1.4.1. Implementasi Interface

Untuk interface sistem dapat dilihat pada gambar-gambar berikut.

4.1.4.1.1. *Preprocess Data dan analisis sentimen*

Gambar IV- 32 *Interface* halaman awal

Gambar IV- 33 *Interface* seteleh *input file*

Gambar IV- 34 *Interface* setelah tombol '*Preprocess Data'* ditekan

Gambar IV- 35 *Hasil Sentimen Analisis Naive Bayes*

Gambar IV- 36 *Sentimen Analisis Information Gain dan Naive Bayes*

Gambar IV- 37 hasil *Analisis Sentimen Information Gain dan Naive Bayes*

4.1.4.1.2. Klasifikasi Data

Gambar IV- 38 *Interface* awal klasifikasi teks

Gambar IV- 39 *Interface* setelah input teks dan menekan 'analisis'

4.1.4.2. Implementasi Pengujian

Langkah pertama dalam tahap ini adalah melakukan uji coba terhadap sistem yang telah dikembangkan. Pengujian dilakukan dengan metode *blackbox* dengan teknik *Equivalence Partitioning*. Dalam pengujiannya menggunakan metode ini meliputi, yaitu skenario test, hasil yang diharapkan, hasil pengujian dan Kesimpulan. Uji coba ini bertujuan untuk memastikan bahwa sistem dapat berjalan sesuai dengan yang diharapkan.

4.1.4.2.1. Preprocess data dan analisis sentimen

Tabel IV- 6 Pengujian *Preprocess Data dan analisis sentimen*

Kode	Skenario Test	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
A1	Menginput *file*	Menampilkan	Sistem	Berhasil

	CSV dengan menekan tombol '*browse file*' dan menekan "Preprocess Data"	data awal yang belum dilakukan *preprocessing text*, lalu menekana tombol "*Preprocess Data"* untuk melakukan preproses teks, setelah selesai, muncul 2 radio *button* untuk memilih metode anlaisis	menampilkan data awal yang belum dilakukan *preprocessing text*, lalu menekana tombol "*Preprocess Data"* untuk melakukan preproses teks, setelah selesai, muncul 2 radio *button* untuk memilih metode anlaisis	
A2	Menginput *file* yang bukan CSV	Menampilkan pesan error	Sistem menampilkan pesan error	Berhasil
A3	Memilih salah satu metode analisis dan menekan	Menampilkan hasil evaluasi berupa *accuracy,*	Sistem menampilkan hasil evaluasi berupa	Berhasil

	tombol "Analyze"	*onfusion matrix, precision, recall, dan f1-score* untuk metode yang dipilih	*accuracy, onfusion matrix, precision, recall, dan f1-score* untuk metode yang dipilih	

4.1.4.2.2. Klasifikasi teks

Tabel IV- 7 Pengujian Klasifikasi teks

Kode	Skenario Test	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
D1	Menginput teks, lalu menekan tombol 'analisis'	Melakukan preprocess text dan menampilkan teks hasil preprocessing dan sentimen hasil analisis	Sistem melakukan preprocess text dan menampilkan teks hasil preprocessing dan sentimen hasil analisis	Berhasil
D2	Tidak ada teks yang diinput	Menampilkan pesar error	Sistem menampilkan	Berhasil

			pesar error	

4.1.4.3. Pelatihan pengguna

Tahap ini adalah memberikan pelatihan kepada pengguna. Pelatihan ini bertujuan untuk memastikan bahwa pengguna dapat menggunakan sistem dengan efektif. pelatihan mencakup pengenalan antarmuka sistem, serta cara penggunaan fitur-fitur utama.

4.1.5 *Maintenance*

Tahap ini adalah langkah penting dalam memastikan keberlanjutan sistem analisis sentimen yang telah dikembangkan. Pada tahap ini, beberapa hal yang dilakukan untuk menjaga agar sistem tetap berfungsi dengan baik terhadap perubahan atau masalah yang mungkin timbul selama penggunaan. Hal yang dapat dilakukan adalah sebagai berikut

4.1.5.1. *Monitoring*

Monitoring adalah langkah awal dalam proses maintenance, di mana sistem diawasi secara terus-menerus untuk mendeteksi setiap anomali atau masalah yang mungkin terjadi. Aktivitas monitoring meliputi pemantauan kinerja sistem dan pemantauan penggunaan sistem.

4.1.5.2. Evaluasi sistem

Evaluasi sistem dilakukan secara berkala untuk menilai apakah sistem masih memenuhi kebutuhan pengguna dan standar performa yang diharapkan. Evaluasi ini mencakup pengujian terhadap komponen perangkat keras dan perangkat lunak, serta analisis umpan balik dari pengguna.

4.1.5.3. Pembaharuan sistem

Seiring dengan perkembangan teknologi dan perubahan kebutuhan pengguna, sistem mungkin memerlukan pembaruan atau peningkatan. Proses pembaruan sistem meliputi pembaharuan perangkat lunak, dan pembaharuan data.

4.1.5.4. Perbaikan dan pemulihan

Jika terdeteksi masalah atau kerusakan pada sistem, langkah-langkah perbaikan dan pemulihan harus segera dilakukan untuk meminimalkan dampak negatif pada pengguna. Aktivitas perbaikan dan pemulihan meliputi identifikasi masalah, dan *fixing bug*.

4.1.5.5. Dokumentasi

Semua aktivitas *maintenance* harus didokumentasikan dengan baik untuk tujuan audit dan referensi di masa yang akan datang. Dokumentasi ini mencakup laporan monitoring, catatan pembaruan sistem, dan log perbaikan.

4.2 Kesimpulan

Dalam penelitian ini, metode *waterfall* digunakan untuk merancang dan membangun perangkat lunak. Metode ini terdiri dari 5 tahap, yaitu *requirement, system design, coding and testing,* implementasi, dan *maintenance*. Penelitian ini menunjukkan metode *waterfall* dapat dipakai untuk mengembangkan perangkat lunak untuk melakukan analisis sentimen.

BAB V
HASIL DAN PEMBAHASAN

5.1 Data Hasil Pengujian

Disini akan dijelaskan mengenai data hasil pengujian sistem yang berfungsi untuk mengetahui kinerja program dalam proses klasifikasi. Pengujian dilakukan untuk menguji *accuracy, confusion matrix, precision, recall, dan f1-score* dengan metode naive bayes dan information gain ditambah naive bayes.

Dalam melakukan perngujian ini, akan dilakukan beberapa skenarion dimana skenario ini akan dibaig berdasarkan data latih dan data uji. Dari total data sebanyak 4174 akan dibagi untuk beberapa skenario. Untuk rincian skenario dapat dilihat dalam tabel V-1

Tabel V- 1 Rincian data Pengujian

Pembagian	Data Latih	Data Uji
50 : 50	2087	2087
60 : 40	2504	1670
70 : 30	2922	1252
80 : 20	3339	835
90 : 10	3757	417

Pengujian dialakukan 2 kali, dimana pengujian pertama hanya menggunakan *naive bayes classifier* dan pada penelitian yang kedua menggunakan seleksi fitur *information gain* dan *naive bayes classifier*. Pengujian ini dilakukan dengan menggunakan dataset yang sudah dilakukan *preprocessing text* untuk melihat perbandingan *accuracy, confusion matrix, precision, recall,* dan *f1-score*.

Pengujian pertama, metode yang digunakan dalam analisis sentimen adalah *naive bayes classifier*. Pengujian berikutnya, metode yang digunakan dalam analisis sentimen adalah *information gian* dan *naive bayes classifier*. Dalam pengujian ini

untuk melakukan seleksi fitur, diberikan sebuah *threshold* sebesar 0.0002 untuk membatasi jumlah dari fitur berdasarkan skor dari *information gain*. Hasil dari penelitian ini akan menampilkan hasil dan menamppilkan jumlah fitur yang digunanakan jumlah fitur yang digunakan. Dari pengujian didapatkan data seperti berikut.

Tabel V- 2 Hasil pengujian

Naive Bayes				
	Accuracy	*Precision*	*Recall*	*F1-score*
50 : 50	0.6003	0.4028	0.4686	0.4280
60 : 40	0.5904	0.5693	0.4685	0.4297
70 : 30	0.6009	0.4058	0.4689	0.4286
80 : 20	0.6027	0.4075	0.4668	0.4255
90 : 10	0.5789	0.3861	0.4520	0.4130
Naive Bayes dan Information Gain				
	Accuracy	*Precision*	*Recall*	*F1-score*
50 : 50	0.5936	0.5697	0.5697	0.4220
60 : 40	0.5940	0.5697	0.4622	0.4231
70 : 30	0.5977	0.4065	0.4649	0.4243
80 : 20	0.6059	0.4092	0.4718	0.4313
90 : 10	0.5813	0.3903	0.4528	0.4135

5.2 Analisis Data Hasil Pengujian

Berdasarkan data pada tabel V-2, terlihat untuuk model *Naive Bayes* menunjukkan nilai *accuracy* antara 0.5789 dan 0.6027. Perbandingan data latih

80:20 memberikan nilai *accuracy* tertinggi di angka 0.6027, sementara perbandingan 90:10 memberikan nilai *accuracy* terendah di angka 0.5789. Secara umum, peningkatan perbandingan data latih cenderung meningkatkan *accuracy*, kecuali pada perbandingan 90:10. *Precision* tertinggi tercapai pada rasio 60:40 sebesar 0.5693 dan cenderung turun pada rasio yang lebih tinggi. Nilai *recall* relatif konsisten di sekitar 0.468, dengan rasio 70:30 memberikan nilai *recall* tertinggi 0.4689. *F1-score* tertinggi juga tercapai pada rasio 70:30 sebesar 0.4286 dan cenderung stabil di sekitar 0.428.

Sementara itu, untuk model *Naive Bayes* dengan *Information Gain* menunjukkan nilai *accuracy* antara 0.5813 dan 0.6059, dengan perbandingan data latih 80:20 memberikan nilai *accuracy* tertinggi sebesar 0.6059 dan perbandingan 90:10 memberikan nilai *accuracy* terendah, yaitu 0.5813. - perbandingan data latih, menunjukkan bahwa *Information Gain* dapat meningkatkan kemampuan model *Naive Bayes* dalam melakukan analisis sentimen.

5.3 Kesimpulan

Penelitian analisis sentimen pada media sosial *twitter* terhadap penggunaan kendaraan listrik dengan menggunakan metode *naïve bayes classifier* dan *information gain* dapat berhasil melakukan analisis sentimen. Sebanyak 4174 data tweet bahasa indonesia yang sudah diberikan label klasifikasi dengan 3 kelas, yaitu positif, negatif, dan netral. Hasil perhitungan evalasi menunjukkan bahwa analisis sentimen dengan menggunakan seleksi fitur memberikan hasil kinerja yang lebih baik dibandingkan dengan metode naive bayes tanpa menggunakan seleksi fitur.

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Penelitian ini membahas tentang penggunaan seleksi fitur *information gain* dan *naive bayes* dalam melakukan analisis senitmen. Dengan melakukan analisis berdasarkan hasil dari penelitian, peneliti dapat menyimpulkan hal sebagai berikut:

1. Penelitian ini menunjukkan bahwa sentimen masyarakat terhadap kendaraan listrik dapat dianalisis secara efektif menggunakan data dari media sosial Twitter dengan memanfaatkan metode *Naïve Bayes Classifier* dan *Information Gain*. Data yang digunakan dalam penelitian ini diambil dari Twitter, diklasifikasikan menjadi sentimen positif, negatif, dan netral. *Naïve Bayes Classifier* digunakan untuk mengklasifikasikan sentimen ini berdasarkan probabilitas, dan *Information Gain* digunakan untuk seleksi fitur guna mengurangi dimensi data dan meningkatkan akurasi klasifikasi.
2. Hasil penelitian menunjukkan bahwa penggunaan *Information Gain* dalam seleksi fitur mampu meningkatkan akurasi model *Naïve Bayes* dalam menganalisis sentimen terhadap kendaraan listrik. Model yang menggunakan *Information Gain* mendaptkan akurasi sebesar 63%, lebih tinggi dibandingkan dengan model yang hanya menggunakan *Naïve Bayes* yang mendapatkan nilai akurasi sebesar 61%. Dalam penelitian ini, akurasi model *Naïve Bayes* setelah diterapkan *Information Gain* mencapai tingkat yang lebih tinggi, memperlihatkan peningkatan signifikan dalam kemampuan klasifikasi sentimen.

6.2 Saran

Berdasarkan penelitian yang sudah dilakukan, terdapat beberapa saran yang dapat diberikan untuk penelitian selanjutnya dan pengembangan yang lebih lanjut:

1. Penelitian ini dilakukan dengan menggunakan dataset dari satu sumber, yaitu Twitter. Disarankan agar penelitian selanjutnya mempertimbangkan penggunaan dataset yang lebih beragam dari berbagai platform media sosial seperti *Facebook, Youtube, Instagram,* atau forum-forum diskusi *online*. Hal ini dapat memberikan perspektif yang lebih luas dan mendalam mengenai sentimen masyarakat terhadap kendaraan listrik.
2. Meskipun metode *text preprocessing* yang digunakan dalam penelitian ini sudah cukup, namun untuk penelitian selanjutnya dapat mempertimbangkan penggunaan teknik pra-pemrosesan tambahan seperti *lemmatization*, *Named Entity Recognition* (NER), dan penggunaan model bahasa yang lebih canggih untuk meningkatkan kualitas data yang diolah.
3. Penelitian ini menggunakan metode *Naïve Bayes Classifier*. Disarankan agar penelitian selanjutnya mencoba menggunakan metode machine learning lain seperti *Support Vector Machine* (SVM), *Random Forest,* atau *deep learning* seperti *Long Short-Term Memory* (LSTM) dan *Transformer* untuk melihat perbandingan performa dan akurasi dalam analisis sentimen.
4. Untuk mendapatkan hasil yang lebih relevan dan terkini, disarankan agar penelitian selanjutnya mencoba melakukan analisis sentimen dengan data real-time. Ini akan membantu dalam memahami tren sentimen yang sedang berlangsung di masyarakat.

Dengan mengikuti saran-saran tersebut, diharapkan penelitian di masa mendatang dapat memberikan kontribusi yang lebih besar dan bermanfaat dalam bidang analisis sentimen serta mendukung perkembangan industri kendaraan listrik di Indonesia.

ANALISIS SENTIMEN PADA MEDIA SOSIAL TWITTER TERHADAP PENGGUNAAN KENDARAAN LISTRIK DENGAN MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER DAN INFORMATION GAIN

ORIGINALITY REPORT

9%	9%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	etheses.uin-malang.ac.id Internet Source	3%
2	ojs.unikom.ac.id Internet Source	2%
3	Submitted to Sriwijaya University Student Paper	2%
4	ojs.unud.ac.id Internet Source	1%
5	repository.ub.ac.id Internet Source	1%
6	Mochamad Amzah Yamin, Kusnadi Kusnadi, Luhur Bayuaji. "Optimasi Algoritma Support Vector Machine (SVM) Dengan Menggunakan Feature Selection Gain Ratio Untuk Analisis Sentimen", INOVTEK Polbeng - Seri Informatika, 2024 Publication	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		

SURAT KETERANGAN PENGECEKAN SIMILARITY

Saya yang bertanda tangan di bawah ini

Nama	: Ivando Sibarani
Nim	: 09021282025046
Prodi	: Teknik Informatika

Menyatakan bahwa benar hasil pengecekan similarity Skripsi~~/Tesis/Disertasi/Lap~~. Penelitian yang berjudul "ANALISIS SENTIMEN PADA MEDIA SOSIAL TWITTER TERHADAP PENGGUNAAN KENDARAAN LISTRIK DENGAN MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER DAN INFORMATION GAIN" adalah 9%.

Dicek oleh operator *: ~~1. Dosen Pembimbing~~ [redacted]

2. UPT Perpustakaan

Demikianlah surat keterangan ini saya buat dengan sebenarnya dan dapat saya pertanggung jawabkan.

Inderalaya, 25 Juni 2024

Menyetujui
Dosen pembimbing,

Rizki Kurniati, M.T.
NIP. 199107122019032016

Yang menyatakan,

Ivando Sibarani
NIM. 09021282025046