BAB II
LANDASAN TEORI

2.1 Pendahuluan

Disini akan menjelaskan secara terperinci tentang landasan teori dari konsep-konsep yang digunakan dalam penelitian ini, serta memberikan penelitian-penelitian lain yang relevan.

2.2 Landasan Teori

2.2.1 *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).

2.2.2 *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 et al., 2021).

2.2.2.1 *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.

2.2.2.2 *Cleaning*

Cleaning 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*.

2.2.2.3 *Tokenizing*

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

2.2.2.4 Normalisasi

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

2.2.2.5 *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 et al., 2021).

2.2.2.6 *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 dan Wulandari. 2021).

2.2.3 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 et al., 2023).

Analisis senitmen adalah merupakan sebuah proses pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistic dan *text mining* yang memiliki tujuan menganalisa pendapat, sentimen, dan sikap dari seseorang (Novina dan Rasal, 2023). Analisis sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seorang, apakah beropini positif, negatif atau netral.

2.2.4 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 et al., 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.

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

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

2.2.5 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 et al., 2022). Tujuan utama dari seleksi fitur adalah untuk memilih fitur terbaik dari suatu kumpulan fitur.

2.2.6 *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 et al., 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:

Entropy (S) : nilai entropy data S

P_i : probabilitas kelas i dalam Data.

K : jumlah kelas.

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

Keterangan

S : seluruh nilai yang mungkin dari atribut A.

S_v : subset dari S dimana atribut A bernilai v.

v : jumlah nilai unik beratribut A.

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*.

2.2.7 *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 et al).

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

$(a_1, a_2, a_3, \dots, a_n)$ 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 $(a_1, a_2, a_3, \dots, a_n)$. Untuk persamaan V_{MAP} dapat ditulis dengan:

$$V_{MAP} = \underset{vj \epsilon v}{argmax} P(v_j | a_1, a_2, a_3, \dots, a_n) \quad \text{(II-8)}$$

Dengan menggunakan teorema Bayes, maka dapat ditulis menjdi

$$V_{MAP} = \underset{vj \epsilon v}{argmax} \frac{P(v_j | a_1, a_2, a_3, \dots, a_n) P(v_j)}{P(a_1, a_2, a_3, \dots, a_n)} \quad \text{(II-9)}$$

$P(a_1, a_2, a_3, \dots, a_n)$ memiliki nilai yang konstan pada semua v_j sehingga dapat ditulis dengan

$$V_{MAP} = \underset{vj \epsilon v}{argmax} P\left(v_j \middle| a_1, a_2, a_3, \dots, a_n\right) P(v_j) \quad \text{(II-10)}$$

Keterangan :

V_{MAP}	: Semua Kategori yang diujikan
$P(v_j \mid a1, a2, a3, \dots, a_n)$	: probabilitas kelas v_j yang diberi fitur $a_1, a_2, a_3, \dots, a_n$
$P(v_j)$	: Probabilitas dari v_j
$P(a_1, a_2, a_3, \dots, a_n)$	: 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 :

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

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

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

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

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_i : $n_i$adalah jumlah kemunculan kata a_i pada kategori v_j

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

2.2.8 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 akurasi digunakan untuk mengetahui seberapa akurat sistem dapat melakukan klasifikasi. Akurasi 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.

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

$$\text{F1} - Score\ negatif$$

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

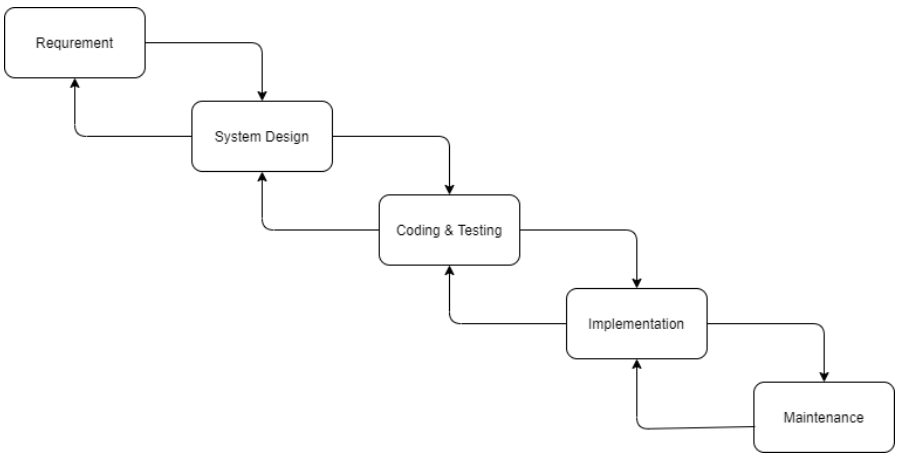
(II-26)

$$\text{F1} - Score\ netral\ = 2\ x\ \frac{Precision\ netral\ \ x\ recall\ netral}{Precision\ netral\ \ + recall\ netral}$$

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

2.2.9 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 et al., 2023).



Gambar II- 1 Metode Waterfall

Langkah-langkah dalam metode *waterfall* digambarkan sebagi berikut.

1. *Requirement*

   Melakukan observasi data, menetapkan fitur, dan tujuan dari sistem dibuat. Semua Langkah ditetapkan secara detail dan digunakan sebagai

kualifikasi sistem.

2. *System Design*

   Membuat sebuah desain sistem berdasarkan syarat sistem yang sudah di tetapkan. Pada tahap ini bertujuan untuk memberikan perancangan sistem yang harus dikerjakan.

3. *Coding & Testing*

   Perancangan yang sudah di buat diterjemahkan dalam bahasa pemrograman menjadi serangkaian unit program. Kemudian dilanjutkan pengujian sistem pada setiap unit program.

4. *Implementation*

   Pengintegrasian setiap unit sistem menjadi sebuah sistem utuh. Kemudian dilakukan pengujian program untuk memastikan kesesuaian syarat sistem.

5. *Maintenance*

   Yang terakhir yaitu melakukan penerapan sistem. Dalam tahap ini kesalahan yang belum ditemukan pada tahap – tahap sebelumnya maka akan dilakukan perbaikan secara berkala.

2.3 Penelitian yang relevan

2.3.1 Analisis Sentimen Terhadap Kendaraan Listrik Pada Platform Twitter Menggunakan Metode *Naive bayes* (Alin et al., 2023)

Tujuan dari penelitian ini adalah untuk menganalisis sentimen masyarakat Indonesia terhadap kendaraan listrik melalui platform Twitter. Penelitian ini bertujuan untuk mengklasifikasikan opini masyarakat menjadi sentimen positif,

negatif, dan netral menggunakan metode Naïve Bayes. Dengan pendekatan ini, peneliti berharap dapat memahami kecenderungan pandangan masyarakat mengenai kendaraan listrik, yang diharapkan dapat memberikan wawasan penting bagi perkembangan teknologi kendaraan listrik di Indonesia.

2.3.2 Analisis Sentimen Kendaraan Listrik Menggunakann Algoritma *Naive bayes* dengan Seleksi Fitur *Information gain* dan Particle Swarm Optimization (Alfarizi & Fitriani, 2023)

Penelitian ini bertujuan untuk menganalisis sentimen masyarakat terhadap kendaraan listrik di media sosial Twitter. Dengan menggunakan algoritma *Naïve Bayes* serta metode seleksi fitur *Information gain* dan *Particle Swarm Optimization* (PSO), penelitian ini bertujuan untuk meningkatkan akurasi klasifikasi sentimen positif, negatif, dan netral terkait opini publik tentang kendaraan listrik. Hasil penelitian diharapkan memberikan informasi yang berguna bagi produsen kendaraan listrik dan pemerintah dalam pengembangan produk serta kebijakan yang mendukung penggunaan kendaraan listrik.

2.3.3 Perbandingan Metode Klasifikasi Support Vector Machine Dan Naïve Bayes Pada Analisis Sentimen Kendaraan Listrik (Ernawati et al., 2023)

Tujuan utama dari penelitian ini adalah untuk menganalisis sentimen masyarakat Indonesia terhadap kendaraan listrik menggunakan data dari Twitter. Penelitian ini juga bertujuan membandingkan dua metode klasifikasi, yaitu Support Vector Machine (SVM) dan Naïve Bayes, dari segi akurasi dan efisiensi waktu. Melalui analisis ini, diharapkan dapat diketahui metode yang paling efektif dalam

mengklasifikasikan sentimen positif, negatif, dan netral terhadap kendaraan listrik di Indonesia.

2.4 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.