BAB 2
LANDASAN TEORI

2.1 Pengoplosan BBM

Dikutip dari berita, pada 26 Februari 2025, Badan Perlindungan Konsumen Nasional (BPKN) mengeluarkan siaran pers terkait dugaan pencampuran BBM jenis Pertamax (RON 92) dengan Pertalite (RON 90) oleh PT Pertamina. Ketua BPKN, Mufti Mubarok, menyatakan bahwa apabila dugaan ini terbukti, maka hak konsumen sebagaimana diatur dalam Undang-Undang Perlindungan Konsumen (UUPK) telah dilanggar. Konsumen yang membayar untuk Pertamax dengan harga lebih tinggi justru mendapatkan BBM dengan kualitas lebih rendah, yaitu Pertalite [9].

2.2 X (Twitter)

X merupakan sebuah platform media sosial dan mikroblogging yang memungkinkan penggunanya untuk mengirim serta membaca pesan singkat yang dikenal sebagai ”*tweet*”, dengan panjang maksimal 280 karakter. Melalui platform ini, pengguna dapat menyampaikan informasi, gagasan, maupun pendapat mereka kepada publik, serta berinteraksi dengan pengguna lain melalui fitur seperti *retweet*, *reply*, dan *like* [10].

2.3 Analisis Sentimen

Sentimen analisis merupakan studi komputasional yang berfokus pada sentimen, emosi, opini, komentar, dan segala bentuk ekspresi yang disampaikan dalam teks. Tujuan dari analisis sentimen adalah mengekstrak atribut dan komponen dari sebuah objek yang telah ditanggapi dengan emosi atau pendapat dan untuk mengevaluasi apakah sentimen tersebut bersifat positif atau negatif [11].
Sentimen sering didefinisikan sebagai pandangan atau opini yang diungkapkan. Selain itu, sentimen juga dapat berarti emosi atau perasaan yang diekspresikan melalui kata-kata. Analisis sentimen berpusat pada pengategorian ulasan berdasarkan polaritasnya. Berdasarkan pengkategorian ini, analisis sentimen dibedakan menjadi dua kategori utama. Pertama, adalah klasifikasi antara opini atau fakta, juga disebut klasifikasi subjektivitas. Kedua, adalah pengklasifikasian

sentimen sebagai positif atau negatif, dikenal sebagai analisis sentimen [12].

2.4 Text Mining

Text mining merupakan metode analisis yang digunakan untuk mengidentifikasi serta mengekstraksi informasi bernilai dari kumpulan teks atau dokumen dalam skala besar. Proses ini melibatkan penerapan algoritma dan pendekatan statistik untuk mengolah, mengorganisasi, serta menyajikan data teks ke dalam bentuk yang dapat dipahami dan dianalisis secara efektif oleh manusia maupun sistem komputer[13].

2.5 Text Preprocessing

Tahap *Text Preprocessing* merupakan fase di mana data diolah sehingga siap dianalisis [14]. Dalam melakukan *text preprocessing*, terdapat beberapa tahap yaitu *translate*, *case folding*,*Cleansing*, *Tokenization*, *Stopword removal*, *normalization*, *Stemming* dan *Labelling*. Berikut adalah penjelasan singkat mengenai tahap-tahap tersebut.

2.5.1 Translate

Proses *translation* atau penerjemahan digunakan sebagai salah satu tahap praproses (*preprocessing*) ketika data teks bersumber dari berbagai bahasa. Tujuan utama dari proses ini adalah untuk menyeragamkan bahasa data sehingga analisis lanjutan seperti klasifikasi, klasterisasi, atau ekstraksi fitur dapat dilakukan secara konsisten. Translation menjadi sangat penting dalam sistem lintas-bahasa atau ketika model yang digunakan hanya dilatih pada satu bahasa utama, seperti bahasa Indonesia.
Translation dapat meningkatkan performa ketika data bersifat multibahasa, terutama jika sistem hanya dilatih pada bahasa mayoritas [15]. Namun demikian, proses ini juga memiliki kelemahan seperti hilangnya makna idiomatik, konteks budaya, dan kemungkinan terjadinya kesalahan terjemahan otomatis yang mengganggu kualitas fitur.

2.5.2 Case Folding

case folding merupakan salah satu bentuk teknik *Text Preprocessing*. Tujuan utama dari proses ini adalah untuk menyeragamkan representasi teks, sehingga kata-kata

yang sama dengan kapitalisasi berbeda (misalnya, ’Bensin’, ’bensin’, ’BENSIN’) diperlakukan sebagai entitas tunggal, yang penting untuk konsistensi analisis [16]. Berikut adalah contoh penerapan *case folding*:

Tabel 2.1. Contoh Case Folding

full_text	**full_text**
Pertamina Ketahuan ’Main Petak Umpet’ Campur Pertamax dengan Pertalite	pertamina ketahuan ’main petak umpet’ campur pertamax dengan pertalite

2.5.3 Cleansing

Cleansing merupakan tahap pembersihan data dari elemen-elemen yang dianggap sebagai noise, seperti tag HTML, tanda baca, sebagian besar angka, serta spasi berlebih. Proses ini bertujuan untuk meningkatkan kualitas data sebelum dianalisis.

Tabel 2.2. Contoh Cleansing

full_text	**Clean_text**
#titipjual vampir bensin & oli mobil 4 pintu	vampir bensin amp oli mobil 4 pintu

2.5.4 Tokenization

Tokenization adalah proses memecah sebuah kalimat menjadi bagian-bagian yang lebih kecil berupa kata-kata. Tahapan ini merupakan langkah awal dalam memecah teks menjadi unit-unit yang lebih kecil (token), yang kemudian dapat dianalisis secara individual [17]. Berikut adalah contoh penerapan *tokenization*:

Tabel 2.3. Contoh Tokenization

full_text	**tokens**
pertamina ketahuan ’main petak umpet’ campur pertamax dengan pertalite	[pertamina, ketahuan, main, petak, umpet, campur, pertamax, dengan, pertalite]

2.5.5 Stopword Removal

Stopword removal merupakan proses dalam pemrosesan teks yang bertujuan untuk menghilangkan kata-kata yang tidak memiliki makna signifikan atau tidak memengaruhi analisis [18], seperti kata imbuhan “di”, “ber”, “ke”, dan sejenisnya. Proses ini dilakukan setelah teks dipisah menjadi kata-kata dasar, sehingga menghasilkan kumpulan kata yang lebih representatif tanpa membentuk kalimat utuh.

Tabel 2.4. Contoh Stopword Removal

full_text	**stopwords**
Data mining adalah proses untuk menemukan pola yang bermanfaat dari sekumpulan data.	Data mining adalah proses menemukan pola bermanfaat dari sekumpulan data.

2.5.6 Normalization

Normalisasi adalah proses mengidentifikasi penulisan kata yang tidak tepat atau berlebihan dan menggantinya dengan kata yang sesuai dengan KBBI [19]. Pada tahap ini, kata-kata yang tidak ditulis sesuai aturan, serta penggunaan singkatan dan sejenisnya, diubah dan disesuaikan dengan pedoman penulisan yang terdapat dalam KBBI. Normalisasi sangat penting untuk memastikan bahwa semua variasi penulisan kata yang memiliki makna, sehingga meningkatkan konsistensi dan akurasi analisis. Berikut adalah contoh penerapan *normalization*:

Tabel 2.5. Contoh Normalization

full_text	**normalization**
Gila sih, pertamax dicampur pertalite? Ngapain juga, harga udh mahal tp kualitas jd turun.	Gila sih, Pertamax dicampur Pertalite? Mengapa juga, harga sudah mahal tapi kualitas jadi turun.

2.5.7 Stemming

Stemming adalah salah satu tahap dalam proses *preprocessing* teks yang berfungsi untuk mengembalikan kata-kata berimbuhan ke bentuk dasarnya. Proses ini bertujuan untuk mengurangi variasi kata ke bentuk dasarnya, yang membantu dalam

mengelompokkan kata-kata dengan makna serupa dan mengurangi kompleksitas data [20]. Berikut adalah contoh penerapan *stemming*:

Tabel 2.6. Contoh Stemming

Text	**Hasil Stemming**
Permainan	Main
Berbagai	Bagai
Keamanan	Aman

2.5.8 Labelling

Labelling adalah proses pemberian label sentimen (positif atau negatif) pada setiap unit teks (dalam hal ini, tweet) berdasarkan polaritas emosional yang terkandung di dalamnya. Dalam penelitian analisis sentimen, proses ini krusial untuk menyediakan data berlabel yang akan digunakan untuk melatih dan mengevaluasi model klasifikasi [21].

Dalam penelitian ini, pendekatan yang digunakan adalah berbasis *lexicon* (kamus sentimen), di mana setiap kata telah diasosiasikan dengan skor sentimen numerik tertentu. Skor ini umumnya berkisar dari nilai negatif (untuk kata-kata negatif) hingga nilai positif (untuk kata-kata positif). Untuk menentukan polaritas keseluruhan suatu tweet, skor sentimen dari semua kata yang terdapat dalam tweet tersebut dijumlahkan. Jika total skor akhir bernilai di atas 0, tweet dikategorikan sebagai sentimen positif. Sebaliknya, jika skor akhir bernilai di bawah 0, tweet dikategorikan sebagai sentimen negatif. Tweet dengan total skor sama dengan 0 dianggap netral dan tidak digunakan dalam analisis lebih lanjut untuk fokus pada sentimen positif dan negatif [21]. berikut ini contoh dari kata positif dan negatif beserta bobotnya untuk labeling:

Tabel 2.7. Contoh Skor Positif dan Negatif dalam Lexicon Bahasa Indonesia

No	Kata Positif	Bobot	Kata Negatif	Bobot
1	hai	3	putus tali gantung	-2
2	merekam	2	gelebah	-2
3	ekstensif	3	gobar hati	-2
4	paripurna	1	tersentuh (perasaan)	-1
5	detail	2	isak	-5
6	pernik	3	larat hati	-3
7	belas	2	nelangsa	-3
8	welas	4	remuk redam	-5
9	kabung	1	tidak segan	-2
10	rahayu	4	gemar	-1
11	maaf	2	tak segan	-1
12	hello	2	sesal	-4
13	promo	3	pengen	-2
14	terimakasih	5	penghayatan	-2

2.6 Data Splitting

Data *splitting* merupakan tahapan penting dalam pemodelan *machine learning* yang bertujuan untuk membagi dataset ke dalam dua atau lebih bagian, sehingga memungkinkan proses pelatihan, pengujian, dan evaluasi model dilakukan secara terpisah [22]. Dalam penelitian ini, pembagian data dilakukan menggunakan metode *train-test split*, yaitu dengan memisahkan dataset menjadi dua subset utama: data latih (*training* data) dan data uji (*training* data). Data latih digunakan untuk membangun dan melatih model, sedangkan data uji dimanfaatkan untuk mengevaluasi performa model terhadap data yang tidak pernah dilihat sebelumnya. Adapun proporsi pembagian data dalam penelitian ini adalah 80% untuk data latih dan 20% untuk data uji.

2.7 Term Frequency-Inverse Document Frequency (TF-IDF)

Teknik *Term Frequency - Inverse Document Frequency* (TF-IDF) adalah sebuah metode yang digunakan dalam pengolahan teks dan pemodelan bahasa alami. Tujuan dari metode TF-IDF adalah untuk menilai tingkat kepentingan sebuah kata

di dalam dokumen [23]. Rumus yang digunakan TF-IDF untuk kata t dalam dokumen d adalah sebagai berikut:

$$\text{TF-}IDF(t,d) = TF(t,d) \times IDF(t) \tag{2.1}$$

Dalam perhitungannya TF-IDF memperhitungkan dua faktor penting, faktor tersebut antara lain :

2.7.1 Term Frequency (TF)

Term frequency berfungsi untuk mengukur sejauh mana suatu kata sering muncul dalam suatu dokumen, pendekatan umum untuk menghitung nilai TF adalah dengan membagi jumlah kemunculan kata tertentu dengan jumlah keseluruhan kata dalam dokumen tersebut [23]. Rumus yang digunakan untuk menghitung *term frequency* adalah sebagai berikut :

$$\text{TF}_{t,d} = \frac{N_{t,d}}{N_d} \quad [24] \tag{2.2}$$

- $\text{TF}_{t,d}$ = Nilai Term Frequency t di dokumen d.
- $N_{t,d}$: Jumlah kemunculan term t dalam dokumen d.
- N_d : Total jumlah semua term dalam dokumen d.

2.7.2 Inverse Document Frequency (IDF)

Inverse document frequency digunakan untuk mengevaluasi tingkat kepentingan suatu kata dalam keseluruhan kumpulan dokumen (korpus). Kata-kata yang jarang muncul dalam banyak dokumen cenderung memiliki nilai IDF yang lebih tinggi, karena dianggap lebih informatif dan relevan dalam membedakan konten antar dokumen [23]. Rumus yang digunakan untuk menghitung *inverse document frequency* adalah sebagai berikut :

$$\text{I}DF(t) = \log \frac{N}{df(t)} \quad [25] \tag{2.3}$$

- N adalah total jumlah keseluruhan dokumen.
- N_t adalah jumlah dokumen yang memuat term$t$.

2.8 Algoritma Naïve Bayes

Naïve Bayes merupakan metode klasifikasi berbasis probabilistik yang bersifat sederhana, di mana proses prediksi dilakukan dengan menghitung probabilitas berdasarkan frekuensi kemunculan dan kombinasi nilai-nilai fitur yang terdapat dalam dataset [26]. Cara kerja algoritma *Naïve Bayes* dimulai dengan proses pembelajaran (*learning*) terhadap data yang tersedia, di mana model mempelajari pola-pola probabilistik dari fitur-fitur yang ada. Setelah proses pelatihan selesai, model kemudian digunakan untuk melakukan klasifikasi terhadap data baru berdasarkan kategori atau kelas yang telah ditentukan sebelumnya [27]. Kekurangan dari penggunaan algoritma *Naïve Bayes* antara lain adalah Keakuratannya tidak bisa diukur menggunakan satu probabilitas saja dan jika probabilitas kondisionalnya bernilai nol, maka probabilitas prediksi juga akan bernilai nol [28].

A Teorema Bayes

Teorema Bayes adalah fondasi dalam statistik yang berfungsi untuk memperbaharui hipotesis dengan informasi yang baru diperoleh. Ini berguna dalam menghitung peluang *(oppurtunity)* berdasarkan kejadian bersyarat yang tidak bisa diperkirakan [29]. Rumus dari Teorema Bayes adalah sebagai berikut :

$$\mathrm{P}(C|d) = \frac{P(C) \cdot P(d|C)}{P(d)} \tag{2.4}$$

Di mana :

- $P(C|d)$adalah probabilitas suatu dokumen (d) yang termasuk dalam kelas (C).
- P(C) adala probabilitas awal suatu kelas atau *prior probability*.
- $P(d|C)$ adalah probabilitas dokumen (d) muncul dalam kelas (C) *likelihood*.
- P(d) adalah probabilitas dokumen d secara keseluruhan *(evidence)*.

B Multinomial Naive Bayes (MNB)

Multinomial *Naive Bayes* merupakan salah satu model dari algoritma *Naive Bayes* yang sering digunakan dalam klasifikasi teks. Algoritma Multinomial *Naive Bayes* mengasumsikan bahwa setiap dokumen merupakan kumpulan kata-kata yang memiliki distribusi multinomial dan setiap fitur dianggap independen satu sama lain, sesuai dengan asumsi dasar *Naive Bayes*[30]. MNB sangat efektif digunakan untuk tugas klasifikasi teks, seperti analisis sentimen, klasifikasi email (spam/ham), dan pengkategorian berita. Kelebihannya terletak pada efisiensi komputasi dan keakuratannya pada data dengan distribusi kata yang khas.

C Complement Naive Bayes (CNB)

Complement Naive Bayes merupakan perbaikan dari model MNB yang dirancang khusus untuk menangani masalah ketidakseimbangan kelas dalam klasifikasi teks. CNB menghitung probabilitas berdasarkan dokumen dari kelas pelengkap (bukan dari kelas target), sehingga mengurangi pengaruh dominasi kelas mayoritas. Dalam pengembangannya, CNB telah diadaptasi ke dalam bentuk *Complement-Class Harmonized Naïve Bayes*, yang memperhalus estimasi probabilitas antar kelas dan meningkatkan performa klasifikasi untuk data yang tidak seimbang [31].

2.9 Confusion Matrix

Confusion matrix merupakan tabel yang dimanfaatkan untuk menilai performa model klasifikasi dalam machine learning. Tabel ini menunjukkan perbandingan antara label sebenarnya (*true class*) dan label yang diprediksi (*predicted class*) oleh model [32]. *Confusion matrix* sangat berguna dalam mengidentifikasi kesalahan klasifikasi serta menghitung metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*. Controh hasil dari *Confusion matrix* dapat dilihat pada tabel 2.8

Tabel 2.8. Confusion Matrix

	Prediksi Positif	**Prediksi Negatif**
Aktual Positif	True Positive (TP)	False Negative (FN)
Aktual Negatif	False Positive (FP)	True Negative (TN)

Penjelasan komponen:

- **True Positive (TP)**: Jumlah data positif yang diprediksi dengan benar sebagai positif.
- **False Positive (FP)**: Jumlah data negatif yang salah diprediksi sebagai positif.
- **False Negative (FN)**: Jumlah data positif yang salah diprediksi sebagai negatif.
- **True Negative (TN)**: Jumlah data negatif yang diprediksi dengan benar sebagai negatif.

Perhitungan dalam *confusion matrix* dapat digunakan untuk menentukan nilai *accuracy, precision, recall, dan F1-score*.

2.9.1 Accuracy

Accuracy menggambarkan seberapa akurat model dalam melakukan pengklasifikasian terhadap seluruh data. Metrik ini menghitung proporsi prediksi yang benar terhadap total jumlah prediksi yang dilakukan. Semakin tinggi nilai akurasi, semakin baik kinerja model dalam mengenali seluruh kelas secara keseluruhan.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.5}$$

2.9.2 Precision

Precision menggambarkan akurasi antara data yang diminta (positif) dengan hasil prediksi yang diberikan oleh model. Metrik ini menilai seberapa banyak prediksi positif yang benar-benar relevan. Nilai precision yang tinggi menunjukkan bahwa model menghasilkan sedikit kesalahan dalam mengklasifikasikan data negatif sebagai positif.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.6}$$

2.9.3 Recall

Recall menggambarkan kemampuan model dalam menemukan kembali data yang relevan dari seluruh data aktual yang positif. Metrik ini penting ketika kesalahan dalam melewatkan data positif harus diminimalkan. Semakin tinggi nilai recall, semakin baik model dalam menangkap seluruh kejadian positif yang sebenarnya.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.7}$$

2.9.4 F1-Score

Score menggambarkan perbandingan rata-rata precision dan recall yang dibobotkan. Accuracy tepat kita gunakan sebagai acuan performansi algoritma jika dataset kita memiliki jumlah data False Negatif dan False Positif yang sangat mendekati (symmetric). Namun jika jumlahnya tidak mendekati, maka sebaiknya kita menggunakan *F1-score* sebagai acuan.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.8}$$