BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen adalah pendekatan komputasional yang digunakan untuk mengekstraksi serta mengkategorikan opini, emosi, dan ekspresi sikap yang terdapat dalam sebuah teks [5]. Tugas utama dalam analisis sentimen adalah mengkategorikan polaritas teks pada level dokumen, kalimat, atau aspek tertentu untuk menentukan apakah teks tersebut memiliki sentimen positif, negatif, atau netral [6]. Dalam penelitian ini, analisis sentimen digunakan untuk memahami opini publik terhadap pelatih baru Tim Nasional Indonesia berdasarkan komentar pengguna di Instagram, dengan menggunakan metode penelitian kuantitatif. Penelitian kuantitatif adalah metode yang menggunakan data dalam bentuk angka yang bertujuan menjawab pertanyaan penelitian. Metode ini menekankan pengukuran secara objektif, berstandar pengumpulan data, dan menggunakan analisis statistik untuk pengujian hipotesis atau menjelaskan suatu fenomena. Metode ini sering digunakan untuk mempelajari hubungan antarvariabel, pengukuran frekuensi, atau dengan mengidentifikasi pola dalam populasi tertentu [7].

2.2 Instagram

Instagram merupakan *platform* media sosial berbagi foto, video, dan komentar yang populer di seluruh dunia, termasuk di Indonesia [8]. Indonesia menempati posisi tertinggi di Asia dengan jumlah pengguna aktif sebanyak 63 juta [9]. Melalui fitur komentar, pengguna dapat menyampaikan opini, kritik, dukungan, atau emosi terhadap sebuah postingan. Komentar pada postingan hasil pertandingan sepak bola seringkali merepresentasikan sentimen publik terhadap tim, pemain, atau pelatih, sehingga menjadi sumber data yang relevan untuk analisis sentimen dalam penelitian ini.

2.3 Preprocessing

Preprocessing teks adalah serangkaian tahapan awal yang dilakukan untuk mempersiapkan data teks mentah sebelum diproses lebih lanjut dalam

analisis data atau *machine learning*. Tahap ini bertujuan untuk membersihkan, menyederhanakan, dan menormalkan teks sehingga dapat meningkatkan kinerja dan akurasi model klasifikasi [10].

Data teks yang diambil langsung dari sumber seperti media sosial biasanya tidak terstruktur dan mengandung banyak noise seperti karakter khusus, singkatan, atau kata-kata tidak relevan. Oleh karena itu, *preprocessing* menjadi tahap krusial agar analisis dapat dilakukan dengan lebih efektif. Berikut ini adalah tahapan-tahapan utama dalam *preprocessing* teks yang digunakan dalam penelitian ini:

2.3.1 Cleaning Data

Cleaning adalah proses membersihkan teks dari elemen-elemen yang tidak diperlukan, seperti angka, simbol, tanda baca, URL, *hashtag*, *mention (@username)*, serta karakter *non*-alfabet lainnya. Tujuan *cleaning* adalah untuk menghilangkan noise sehingga teks menjadi lebih rapi dan mudah dianalisis [11].

2.3.2 Case Folding

Case folding adalah proses mengubah seluruh huruf dalam teks menjadi huruf kecil *(lowercase)*. Langkah ini dilakukan untuk menghindari perbedaan analisis antara kata yang sebenarnya sama, tetapi berbeda dalam penggunaan huruf kapital [11].

2.3.3 Tokenization

Tokenization adalah proses memecah teks menjadi potongan-potongan kata atau token [12]. Setiap token biasanya mewakili satu kata, yang akan menjadi unit analisis dalam pengolahan data.

2.3.4 Normalization

Normalization adalah proses mengubah kata-kata tidak baku, singkatan, atau kata-kata slang ke dalam bentuk standar atau baku [13]. Hal ini penting terutama untuk data komentar media sosial yang sering menggunakan bahasa informal.

2.3.5 Stopword Removal

Stopword removal adalah proses menghapus kata-kata umum yang sering muncul tetapi tidak membawa informasi penting dalam analisis, seperti ”yang”, ”di”, ”ke”, ”dan”, ”adalah”. *Stopwords* tidak berkontribusi banyak terhadap pemaknaan teks sehingga dihilangkan untuk meningkatkan efisiensi analisis [14].

2.3.6 Stemming

Stemming adalah proses mengubah kata menjadi bentuk dasarnya dengan menghilangkan imbuhan seperti awalan dan akhiran. Tujuan stemming adalah untuk mengurangi variasi kata sehingga kata-kata yang memiliki makna sama dapat dikelompokkan bersama [15].

2.4 TF-IDF *(Term Frequency – Inverse Document Frequency)*

TF-IDF adalah metode pemberian bobot pada setiap kata dalam dokumen teks yang bertujuan untuk menilai seberapa penting sebuah kata dalam suatu dokumen relatif terhadap keseluruhan korpus (kumpulan dokumen). Metode ini banyak digunakan dalam proses ekstraksi fitur *(feature extraction)* pada analisis teks, terutama sebelum teks dianalisis menggunakan algoritma klasifikasi seperti *Naïve Bayes*.

TF-IDF menggabungkan dua konsep utama, yaitu *Term Frequency* dan *Inverse Document Frequency*. Perkalian kedua nilai ini akan memberikan bobot yang lebih tinggi kepada kata-kata yang sering muncul dalam satu dokumen tetapi jarang muncul di dokumen lain, sehingga dianggap lebih relevan secara kontekstual [16].

2.4.1 Term Frequency

Term Frequency adalah ukuran frekuensi kemunculan suatu kata dalam sebuah dokumen. Dengan konsep dasarnya, semakin sering sebuah kata muncul dalam dokumen, semakin besar pengaruh atau relevansi kata tersebut terhadap dokumen tersebut. Pada Rumus 2.1 menunjukkan cara perhitungan *Term Frequency*.

$$TF(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \tag{2.1}$$

Keterangan:
$TF(t,d)$: *frekuensi term t* dalam dokumen *d*
$f_{t,d}$: jumlah kemunculan kata *t* dalam dokumen *d*
$\sum_k f_{k,d}$: total seluruh kata dalam dokumen *d*

2.4.2 Inverse Document Frequency

Inverse Document Frequency digunakan untuk mengukur seberapa umum atau seberapa jarangnya suatu kata muncul di seluruh dokumen. Kata yang terlalu sering muncul di banyak dokumen (seperti ”dan”, ”yang”) akan memiliki nilai IDF yang rendah, karena dianggap kurang informatif. Pada Rumus 2.2 menunjukkan cara perhitungan *Inverse Document Frequency*.

$$IDF(t) = \log\left(\frac{N}{df(t)}\right) \tag{2.2}$$

Keterangan:
N : jumlah total dokumen dalam korpus
$df(t)$: jumlah dokumen yang mengandung term *t*

2.4.3 TF-IDF

Nilai akhir TF-IDF adalah hasil perkalian antara TF dan IDF, yang menunjukkan pentingnya suatu kata dalam satu dokumen relatif terhadap seluruh korpus. Pada Rumus 2.3 menunjukkan cara perhitungan *TF-IDF*.

$$TFIDF(t,d) = TF(t,d) \times IDF(t) \tag{2.3}$$

Semakin tinggi nilai TF-IDF, semakin penting kata tersebut dalam konteks dokumen tertentu. Sebaliknya, kata-kata yang umum dan sering muncul di seluruh dokumen (seperti “adalah”, “dan”) akan memiliki nilai TF-IDF yang rendah.

2.5 SMOTE

Synthetic Minority Over-sampling Technique (SMOTE) merupakan metode yang digunakan untuk menangani ketidakseimbangan kelas dalam sebuah dataset. Ketidakseimbangan ini terjadi ketika jumlah data pada satu atau lebih kelas jauh lebih sedikit dibandingkan kelas lainnya, yang dapat menyebabkan performa model klasifikasi menjadi bias terhadap kelas mayoritas.

SMOTE bekerja dengan cara menghasilkan sampel sintetis dari kelas minoritas, bukan dengan sekadar menduplikasi data yang ada. Teknik ini membuat data baru dengan membentuk kombinasi nilai-nilai dari sampel minoritas yang saling berdekatan dalam ruang fitur [17]. Dengan cara ini, distribusi kelas dalam dataset menjadi lebih seimbang, sehingga model dapat belajar dengan lebih baik dan adil terhadap semua kelas. Penggunaan SMOTE dalam penelitian ini bertujuan untuk menyamakan jumlah data komentar berlabel positif, negatif, dan netral agar model Naïve Bayes yang dilatih tidak cenderung bias terhadap kelas dominan.

2.6 Naïve Bayes

Naïve Bayes adalah algoritma klasifikasi berbasis probabilitas yang digunakan secara luas dalam pemrosesan bahasa alami *(Natural Language Processing)* dan *text mining*, termasuk dalam analisis sentimen [18]. Algoritma ini disebut *"naïve"* (naïf) karena mengasumsikan bahwa setiap fitur dalam data bersifat independen satu sama lain terhadap kelas target, padahal dalam kenyataannya, asumsi tersebut sering kali tidak sepenuhnya benar [19]. Meski demikian, asumsi ini justru membuat perhitungannya sederhana, cepat, dan efektif dalam banyak kasus nyata, khususnya klasifikasi teks. Dalam konteks analisis sentimen, algoritma ini akan menghitung seberapa besar kemungkinan sebuah komentar termasuk ke dalam kategori positif, negatif, atau netral, berdasarkan kata-kata yang muncul di dalamnya. Meskipun metode ini sederhana, *Naïve Bayes* dikenal cukup efektif dalam melakukan klasifikasi teks, terutama pada data dengan dimensi tinggi seperti dokumen atau komentar media sosial.

Algoritma *Naïve Bayes* memiliki beberapa varian turunan, di antaranya adalah *Multinomial Naïve Bayes*, *Gaussian Naïve Bayes*, dan *Bernoulli Naïve Bayes*. Varian *Multinomial Naïve Bayes* sangat sesuai untuk menangani data diskrit seperti kata-kata dalam dokumen teks, karena algoritma ini memperhitungkan probabilitas suatu kelas berdasarkan frekuensi kemunculan kata dalam teks [20].

Sementara itu, *Gaussian Naïve Bayes* digunakan untuk data dengan fitur yang bersifat kontinu dan mengikuti distribusi normal (Gaussian) [21]. Metode ini mengasumsikan bahwa data yang dimasukkan memiliki distribusi normal, sehingga sering diterapkan pada klasifikasi dengan variabel numerik. Adapun *Bernoulli Naïve Bayes* digunakan untuk data biner, dengan asumsi bahwa fitur hanya memiliki dua nilai, yaitu 0 dan 1, yang menunjukkan keberadaan atau ketidakhadiran suatu kata dalam dokumen [22]. Dari ketiga jenis varian tersebut, *Multinomial Naïve Bayes* dipilih sebagai metode yang paling sesuai untuk kebutuhan analisis sentimen pada penelitian ini, karena mampu menangani representasi data berupa frekuensi kata dalam teks.

2.6.1 Multinomial Naïve Bayes

Dalam penelitian ini, digunakan *Multinomial Naïve Bayes* untuk memproses data berbentuk teks atau data diskrit seperti komentar, yang merupakan salah satu varian dari *Naïve Bayes* dan dikenal cukup efektif dalam menyelesaikan permasalahan klasifikasi pada teks. Algoritma ini menghitung probabilitas sebuah dokumen d masuk ke dalam kelas c berdasarkan frekuensi kata-kata (token) dalam dokumen tersebut [20]. Simbol $\propto$ berarti ”sebanding dengan”, karena biasanya nilai probabilitas akhir (setelah mempertimbangkan data) hanya dibandingkan antarkelas untuk menentukan klasifikasi. Pada Rumus 2.4 adalah perhitungan *Multinomial Naïve Bayes* sebagai berikut [23].

$$P(c|d) \propto P(c) \prod_{k=1}^{n_d} P(t_k|c) \tag{2.4}$$

Keterangan:
$P(c|d)$: probabilitas dokumen d termasuk dalam kelas c
$P(c)$: probabilitas awal kelas c
$P(t_k|c)$: probabilitas *term* t_k muncul dalam kelas c
n_d : jumlah total kata dalam dokumen d

2.6.2 Langkah Umum Klasifikasi Menggunakan *Naïve Bayes*

1. Ekstraksi fitur dari dokumen, misalnya menggunakan TF-IDF.
2. Hitung probabilitas awal setiap kelas ($P(c)$) dari distribusi kelas dalam data

latih.

3. Hitung probabilitas setiap kata ($P(t_k|c)$) muncul dalam masing-masing kelas berdasarkan frekuensi kata.
4. Hitung $P(c|d)$ untuk setiap kelas menggunakan rumus *Multinomial Naïve Bayes*.
5. Dokumen diklasifikasikan ke dalam kelas dengan nilai $P(c|d)$ tertinggi.

2.6.3 Keunggulan Naïve Bayes

1. Proses pelatihan dan prediksi sangat cepat, bahkan untuk dataset berukuran besar.
2. Efektif untuk klasifikasi teks seperti analisis sentimen dan deteksi spam.
3. Sederhana dan tidak memerlukan banyak parameter.

2.6.4 Kelemahan Naïve Bayes

1. Asumsi independensi antar fitur sering kali tidak realistis dalam konteks teks.
2. Tidak mempertimbangkan konteks atau urutan kata dalam kalimat.

2.7 Confusion Matrix

Confusion Matrix merupakan salah satu metode yang digunakan untuk mengevaluasi performa model klasifikasi, baik untuk data biner maupun *multi*-kelas. *Confusion matrix* menyajikan perbandingan antara hasil prediksi model dengan nilai aktual dari data yang diuji [24]. Melalui tabel ini, performa model dapat diketahui secara lebih terperinci karena tidak hanya mengukur jumlah prediksi benar, tetapi juga kesalahan yang terjadi dalam masing-masing kelas [25]. Tabel 2.1 menunjukkan struktur dasar dari *confusion matrix* pada klasifikasi biner.

Tabel 2.1. Tabel *Confusion Matrix*

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

Keterangan:

True Positive (TP): Jumlah data positif yang benar diprediksi 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 benar diprediksi sebagai negatif.

Dari nilai-nilai tersebut, maka dapat dihitung beberapa metrik evaluasi model klasifikasi seperti berikut.

2.7.1 Accuracy

Akurasi adalah persentase total prediksi yang benar dibandingkan dengan keseluruhan jumlah prediksi yang dilakukan oleh model. Rumus *akurasi* adalah sebagai berikut.

Rumus 2.5 *Accuracy*.

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

2.7.2 Precision

Precision mengukur tingkat ketepatan dari prediksi positif model, yaitu seberapa besar data yang diprediksi sebagai positif benar-benar merupakan data positif.

Rumus 2.6 *Precision*.

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

2.7.3 Recall

Recall (atau sensitivitas) mengukur seberapa banyak data positif yang berhasil dikenali dengan benar oleh model dari seluruh data aktual positif.

Rumus 2.7 *Recall*.

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

2.7.4 F1-Score

F1-Score adalah nilai rata-rata harmonis antara *precision* dan *recall*. Metrik ini memberikan gambaran keseimbangan antara kemampuan model dalam menemukan data positif dan menghindari kesalahan prediksi.

Rumus 2.8 *F1-Score*.

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

F1-Score sangat berguna dalam kasus di mana terdapat ketidakseimbangan jumlah data antar kelas, karena menggabungkan presisi dan sensitivitas secara seimbang [26].

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA