BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen atau disebut dengan *Opinion Mining*, termasuk dalam *NLP* yang merupakan pembelajaran komputasi tentang komentar, perilaku, berdasarkan emosi orang terhadap entitas [6]. Analisis sentimen dilakukan untuk mengetahui opini publik tentang perilaku yang bersifat positif, negatif, dan netral pengguna yang terdapat di media sosial. Opini dan sentimen memiliki karakteristik subjektif, oleh karena itu penting untuk memeriksa ulasan atau pendapat dari banyak orang karena untuk satu pendapat saja itu hanya mewakili pandangan subjektif yang biasanya tidak cukup untuk diterapkan dalam sentimen[7].

2.2 Labelling Data

Labelling data secara manual yang berfungsi sebagai penentuan setiap *tweet* termasuk ke dalam kelas positif yang berisikan masukan, saran, atau emosi positif seperti senang dan bahagia, kelas negatif yang berisikan kalimat kritik, keluhan dan emosi negatif seperti marah dan kecewa, atau termasuk dalam kelas netral yang berisikan sikap emosi biasa saja atau tidak pada kedua kelas positif dan negatif.

2.3 SMOTE

Metode *Synthetic Minority Over-samplingTechnique* (SMOTE) merupakan metode untuk menangani ketidak seimbangan kelas. Teknik ini menggabungkan sampel baru dari kelas minoritas untuk menyeimbangkan dataset dengan membuat sebuah kelas baru dari kelas minoritas dengan pembentukan *convex* kombinasi dari kelas yang berdekatan. Dengan menerapkan metode ini dapat membuat dataset menjadi seimbang dengan membuat sampel *synthetic* dari pada melalukan duplikat sampel[8].

2.4 Text Preprocessing

Text Processing merupakan sebuah proses otomatis analisis dan memilah data teks yang tidak ter-struktur agar mendapatkan masukan yang baik pada pemo-

delan data dan analisis. Tahap ini sangat berpengaruh terutama dalam melakukan sentimen analisis [4] data teks yang memiliki ukuran besar dan memiliki banyak *noise* pada data mentah, untuk mengoptimalisasi *noise* tersebut dilakukan dengan beberapa tahapan seperti berikut :

1. Case Folding
   Case Folding merupakan pemrosesan yang bertujuan mengubah seluruh teks huruf dalam sebuah data dokumen menjadi huruf kecil dan menghilangkan karakter selain huruf sehingga data teks hanya berupa huruf.

2. Data Cleaning
   Data cleaning merupakan proses pembersihan data teks bertujuan untuk menghilangkan simbol, angka, dan tanda baca pada setiap *tweet*.

3. Tokenization
   Tokenization merupakan pemrosesan memisahkan kalimat menjadi daftar kata tunggal (*token*) yang bertujuan untuk mempermudah proses *stemming* dalam analisis untuk mencari kata dasar. Contoh pemisahan kalimat menjadi token sebagai berikut :

   Kalimat : Gudeg makanan khas Yogyakarta

   Output : | *Gudeg* | | *makanan* | | *khas* | | *Yogyakarta* |

4. Stopword Removal
   Stopword Removal merupakan pemrosesan pada text untuk menghapus kata-kata yang tidak memiliki arti pada sebuah kalimat di *tweet*. Contoh *Stopword* dalam bahasa indonesia yaitu “yang”, “dan”, atau ”di”.

5. Stemming
   Stemming merupakan proses mencari kata dasar pada setiap kata di kalimat dengan cara memisahkan imbuhan awal atau imbuhan akhir. Contoh proses *stemming* yaitu, kata ”mengambil” lalu dilakukan proses *stemming* menjadi ”ambil”, karena ”mengambil” memiliki kata dasar yaitu ”ambil”.

2.5 Multinomial Naïve Bayes

Multinomial Naïve Bayes merupakan model variasi dari *Naïve Bayes* yang teruji baik dalam permasalahan klasifikasi teks. *Multinomial Naïve Bayes* mengasumsikan ketidaktergantungan terhadap fitur tertentu pada setiap kelas dan

mengabaikan seluruh dependensi setiap atribut [9]. *Naïve Bayes Classfier* memiliki kecepatan dan akurasi yang baik, berikut merupakan rumus dari *bayes* di definisikan di bawah ini.

$$P(w_i|x) = \frac{P(x|w_i).P(w_i)}{P(x)} \tag{2.1}$$

1. $P(x|w_i)$ = sebuah peluang kata x muncul di kelas w.
2. $P(w_i)$ = Peluang Kata pada kelas c.
3. $P(x)$ = peluang kemunculan pada kata x.

Proses klasifikasi menggunakan *Multinomial Naïve Bayes* yaitu melakukan perhitungan total kemunculan pada masing-masing kata dalam setiap dokumen, berikut merupakan persamaan dari *Multinomial Naïve Bayes*[4].

$$C_{map} = argmax P(c|d) \prod 1 \leq k \leq n_d P(t_k|C) \tag{2.2}$$

1. *argmax* = Mencari nilai *posterior probability* terbesar pada suatu kelas.
2. $P(t_k|C)$ = *Conditional probability*, yaitu peluang munculnya kata k dalam kelas tertentu.
3. $P(c)$ = *Prior Probability* pada kelas c.

Rumus 2.3 merupakan persamaan untuk menghitung $P(c)$.

$$P(c) = \frac{N_c}{N} \tag{2.3}$$

1. N_c = Jumlah kelas c pada seluruh dokumen.
2. N = Jumlah seluruh dokumen.

Untuk mencari conditional probability dapat menggunakan pada rumus 2.4.

$$P(t|c) = \frac{T_{ct}}{\sum t' \in V T_{ct'}} \tag{2.4}$$

1. T_{ct} = Frekuensi suatu kata pada kelas c dalam dokumen yang berulang.
2. $T_{ct'}$ = Jumlah seluruh kata dalam kelas c.

Seringkali ketika proses klasifikasi terdapat kata yang tidak pernah muncul pada kelas tertentu menyebabkan kelas tersebut memiliki nilai nol karena *conditional probability* bernilai nol, sehingga mengakibatkan kesalahan sistem dalam melakukan klasifikasi terhadap kata-kata dalam sebuah dokumen[4]. Cara untuk menghilangkan peluang dari nilai nol dilakukan dengan cara *add one smoothing (Laplace Smoothing*, Rumus 2.5 ialah rumus *laplace smoothing* yaitu menambahkan angka satu pada setiap perhitungan angka.

$$P(t|c) = \frac{T_{ct} + 1}{(\sum t' \in V T_{ct'}) + B'} \tag{2.5}$$

1. B' = Jumlah keseluruhan kosakata unik pada seluruh kelas dalam dokumen.

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

Term Frequency - Inverse Document Frequency (TF-IDF) merupakan sebuah ekstraksi fitur untuk memberikan bobot nilai pada setiap kata. TF-IDF menghitung banyaknya *term* yang muncul pada suatu dokumen. untuk mencari bobot nilai TF-IDF, dapat digunakan rumus berikut[4].

1. *Term Frequency* (TF)
   Term Frequency merupakan cara pembobotan kata yang paling sederhana yaitu dengan:

   $$tf(t,d) = \frac{n_i j}{\sum_k n_i, j} \tag{2.6}$$

   Keterangan:

   (a) $tf(t,d)$ = Frekuensi *term*

   (b) $n_i j$ = Total suatu *term* yang muncul pada suatu dokumen

   (c) $\sum_k n_i, j$ = Total seluruh kata dalam suatu dokumen

2. *Inverse Document Frequency (IDF)*
   Inverse Document Frequency (IDF) berguna melihat kemunculan setiap kata pada kumpulan kelas, dengan cara seperti berikut:

   $$idf = log \frac{N}{df_j} \tag{2.7}$$

Keterangan:

(a) N = Total kelas

(b) df_j = Total kelas j yang berisi kata i

3. Menghitung TF-IDF *(*Term Frequency Inverse Document Frequency)
   Rumus 2.8 berguna untuk menjumlahkan kedua hasil TF (*term frequency*) dan IDF(*inverse document frequency*):

$$w_{ij} = tf_{ij} \times idf \tag{2.8}$$

Keterangan:

(a) w_{ij} = Bobot kata i pada kelas j

(b) tf_{ij} = Total kemunculan kata i pada kelas j

(c) df_j = Total kelas j yang berisi kata i

2.7 Confusion Matrix

Confusion matrix ialah metode yang digunakan dalam mengukur kinerja pada suatu klasifikasi. Metode ini pada dasarnya banyak mengandung informasi yang membandingkan sebuah hasil klasifikasi yang dilakukan sistem dengan hasil klasifikasi yang seharusnya. *Confusion Matrix* merupakan *tools* penting dalam metode visualisasi pada *machine learning* yang biasanya memuat dua kategori maupun lebih [10]. Tabel di bawah merupakan contoh gambaran hasil *Confusion Matrix* prediksi dua kelas

Tabel 2.1. Confusion Matrix

		Kelas Sebenarnya	
		Positive	Negative
Kelas Prediksi	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

1. *True Positive*(TP), Jumlah data dengan kelas positif yang diklasifikasikan positif.

2. *True Negative*(TN), Jumlah data dengan kelas negatif yang diklasifikasikan negatif.

3. *False Positive*(FP), Jumlah data dengan kelas positif yang diklasifikasikan negatif.

4. *False Negative*(FN), Jumlah data dengan kelas negatif yang diklasifikasikan positive.

Selanjutnya menghitung metrik evaluasi pada tabel 2.1 yang bertujuan untuk mengukur performa pada proses pembelajaran yang telah dilakukan oleh mesin[4], metrik evaluasi yang digunakan meliputi *Accuracy*, *Precision*, *Recall*, dan *F-score*. Berikut rumus metrik evaluasi yaitu :

1. *Accuracy*

   Accuracy nilai prediktif benar (*positive, negatif* dengan jumlah keseluruhan data.

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + FalseNegative + FalsePpositive + TrueNegative} x100 \tag{2.9}$$

2. *Precision*

   Precision ialah nilai prediktif positif yang di jumlahkan berdasarkan total data yang teridentifikasi sebagai kelas positif .

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{2.10}$$

3. *Recall*

   Recall menampilkan nilai berdasarkan total data yang teridentifikasi sebagai kelas positif dibagi keseluruhan sampel yang memiliki label positif.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2.11}$$

4. *F-Score*

   F-Score merupakan perbandingan antara *precision* dan *recall* yang telah dihitung.

$$F1 = 2X \frac{Recall * Precision}{Recall + Precision} \tag{2.12}$$