BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen adalah proses komputasional untuk mengidentifikasi, mengekstrak, dan memahami opini, sentimen, emosi, dan sikap yang terkandung dalam teks atau data lainnya terhadap suatu entitas [12]. Analisis sentimen dapat digunakan untuk menganalisis informasi yang dihasilkan pengguna di situs-situs media sosial, situs ulasan film, dan situs e-commerce serta untuk membantu orang membuat keputusan yang lebih baik [13]. Sentimen analisis mengkategorikan teks menjadi positif, negatif, dan netral untuk membantu organisasi meningkatkan kepuasan pelanggan, reputasi merek, dan pendapatan [14].

2.2 Shopee

Shopee merupakan sebuah platform perdagangan elektronik (e-commerce) yang didirikan pada tahun 2015 yang disesuaikan untuk tiap wilayah dan menyediakan pengalaman berbelanja *online* yang mudah, aman, dan cepat bagi pelanggan melalui dukungan pembayaran dan logistik yang kuat. Shopee menyediakan layanan belanja online yang meliputi berbagai kategori produk, mulai dari mode, kecantikan, elektronik, peralatan rumah tangga, hingga makanan dan minuman. Shopee sangat populer di Asia Tenggara dan Taiwan dan telah berkembang pesat menjadi salah satu platform e-commerce terdepan di Asia Tenggara dan Taiwan [15].

2.3 Text Preprocessing

Text preprocessing adalah proses untuk mengubah bentuk dokumen menjadi data yang lebih terstruktur dan lebih siap digunakan untuk tahap berikutnya. Tahap ini berfungsi untuk memaksimalkan akurasi klasifikasi data [16]. Tahapan dalam *text preprocessing* seperti *cleaning*, *case folding*, *tokenizing*, *normalizing*, *stopword removal*, dan *stemming*.

1. *Cleaning*

Cleaning merupakan tahap untuk menghilangkan simbol lain selain karakter alfabet. Tahap ini bertujuan untuk mengurangi karakter yang tidak memiliki makna dalam analisis sentimen [17].

2. *Case Folding*

   Case folding merupakan tahap untuk mengubah semua karakter huruf menjadi huruf kecil [17]. Sehingga tidak ada lagi perbedaan huruf besar dan kecil di dalam dokumen.

3. *Tokenizing*

   Tokenizing merupakan tahap untuk memotong kata berdasarkan penyusunan kata tersebut [17]. Tahap ini akan menghasilkan satuan kata dari sebuah kalimat untuk membantu mesin dalam melakukan pengolahan teks.

4. *Normalizing*

   Normalizing merupakan tahap untuk mengubah kata tidak baku menjadi baku. Contoh dari tahap ini adalah mengubah kata ”pengen” menjadi ”ingin” [18].

5. *Stopword Removal*

   Stopword Removal merupakan tahap untuk menghilangkan kata yang dianggap tidak memiliki arti atau kata yang memiliki nilai informasi yang rendah. Contoh kata ”di”, ”dan”, dan sejenisnya [18].

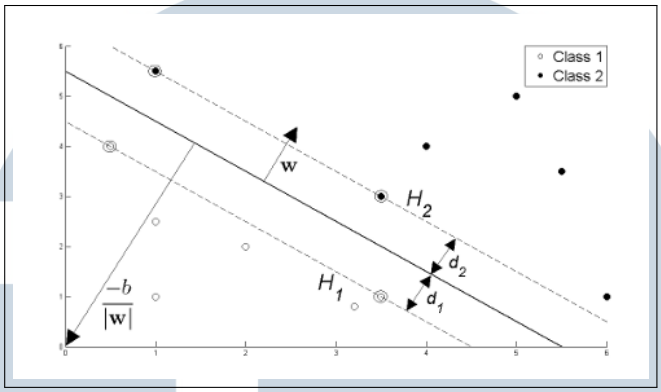
6. *Stemming*

   Stemming merupakan tahap untuk mengubah kata menjadi bentuk dasarnya. Contoh dari tahap *stemming* adalah kata ”semuanya” menjadi ”semua” [18].

2.4 Support Vector Machine

Algoritma *support vector machine* (SVM) adalah sebuah algoritma *supervised learning* yang digunakan untuk pemisahan kelas dan regresi. SVM beroperasi dengan mencari *hyperplane* optimal yang memaksimalkan jarak antara kelas-kelas data. *Hyperplane* ini adalah fungsi yang dapat memisahkan kelas-kelas tersebut [19]. Algoritma SVM memiliki performa yang bagus baik dengan jumlah data yang besar maupun kecil. Algoritma ini awalnya hanya bisa melakukan

klasifikasi biner namun saat ini telah dikembangkan lebih jauh sehingga mampu digunakan untuk mengklasifikasi beberapa kelas sekaligus [20].



Gambar 2.1. *Support Vector Machine*

Pada gambar 2.1 terdapat *hyperplane* yang memisahkan kelas pada sebuah data dua dimensi. *Hyperplane* tersebut didapatkan dari persamaan berikut [21]:

$$wx + b = 0 \tag{2.1}$$

Pada gambar 2.1, margin untuk H1 dapat dideskripsikan seperti berikut:

$$w \cdot x_i + b = 1 \tag{2.2}$$

Pada gambar 2.1, margin untuk H2 dapat dideskripsikan seperti berikut:

$$w \cdot x_i + b = -1 \tag{2.3}$$

Keterangan:
w = nilai normal ke *hyperplane*
x = vektor *input*
b = konstanta bias

Ada beberapa parameter dalam menggunakan Model Klasifikasi dengan Support Vector Machine yang berguna untuk meningkatkan kinerja model, yaitu gamma, cost (C), dan kernel. Parameter gamma adalah faktor yang menentukan seberapa jauh pengaruh dari sampel dalam dataset yang dilatih. Nilai rendah pada gamma menunjukkan pengaruh yang jauh, sedangkan nilai tinggi menunjukkan pengaruh yang dekat. Parameter cost (C) digunakan untuk mengoptimalkan metode SVM untuk menghindari kesalahan klasifikasi pada data pelatihan. Kernel Trick,

dalam algoritma Support Vector Machine, digunakan untuk mengubah representasi data yang non-linear dan berdimensi rendah menjadi ruang dimensi yang lebih tinggi. Hal ini memungkinkan SVM untuk menangani pemisahan yang lebih kompleks dan efektif pada data yang sulit dipisahkan secara linear. Terdapat beberapa jenis fungsi kernel yang umum digunakan, seperti: kernel linear, kernel polynominal, dan kernel RBF (Radial Basic Function) [22].

2.5 TF-IDF

Term Frequency-Inverse Document Frequeny (TF-IDF) adalah sebuah metode dalam *feature extraction* yang menggabungkan dua konsep perhitungan, yaitu dari perhitungan *Term Frequency* dan *Inverse Document Frequency*. *Term Frequency-Inverse Document Frequeny* (TF-IDF) adalah metode yang sangat diakui untuk mengevaluasi pentingnya sebuah kata dalam sebuah dokumen. *Term Frequency* merupakan jumlah sebuah kata tertentu (t) dalam sebuah dokumen dibagi dengan total jumlah kata dalam dokumen tersebut. Sedangkan *Inverse Document Frequency* merupakan logaritma dari frekuensi dokumen yang mengandung kata tersebut. Perhitungan TF-IDF dihasilkan dengan mengkalikan nilai TF dan IDF [23].

Rumus perhitungan TF-IDF:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t) \tag{2.4}$$

Dimana rumus *Term Frequency* (TF):

$$\text{TF}(t,d) = \frac{\text{jumlah kemunculan term } t \text{ dalam dokumen } d}{\text{total jumlah term dalam dokumen } d} \tag{2.5}$$

Dan rumus *Inverse Document Frequency* (IDF):

$$\text{IDF}(t) = \log\left(\frac{N}{\text{DF}(t)}\right) \tag{2.6}$$

Keterangan:
t = kata
d = dokumen
N = jumlah dokumen
DF(t) = jumlah dokumen yang memiliki kata t

2.6 *Confusion Matrix*

Confusion matrix adalah cara standar untuk merangkum kinerja suatu metode klasifikasi [24]. *Confusion matrix* juga merupakan tabel ringkasan dari jumlah prediksi yang benar dan salah yang dihasilkan oleh pengklasifikasi (atau model klasifikasi) untuk tugas klasifikasi biner [25].

Actual	*Prediction*		
	Positive	***Negative***	***Neutral***
Positive	*True Positive* (TP)	*False Negative1* (FNg1)	*False Neutral1* (FNt1)
Negative	*False Positive1* (FP1)	*True Negative* (TNg)	*False Neutral2* (FNt2)
Neutral	*False Positive2* (FP2)	*False Negative2* (FNg2)	*True Neutral* (TNt)

Gambar 2.2. Tabel *Confusion Matrix*

Sumber: [26]

1. *True Positive* (TP): Model memprediksinya positif, dan itu benar.
2. *False Positive* (FP): Model memprediksinya positif, dan itu salah.
3. *True Negative* (TNg): Model memprediksinya negatif, dan itu benar.
4. *False Negative* (FNg): Model memprediksinya negatif, dan itu salah.
5. *True Neutral* (TNt): Model memprediksinya netral, dan itu benar.
6. *False Neutral* (FNt): Model memprediksinya netral, dan itu salah.

Setelah mendapatkan hasil tabel *confusion matrix*, maka bisa didapatkan performa dari model tersebut berdasarkan nilai *accuracy*, *precision*, *recall*, dan *F-1 score*.

1. *Accuracy*

   Akurasi adalah ukuran untuk seberapa akurat model dalam klasifikasi prediksi yang benar.

$$\text{Accuracy} = \frac{\text{TP} + \text{TNg} + \text{TNt}}{\text{TP} + \text{FNg1} + \text{FNt1} + \text{FP1TNg} + \text{FNt2} + \text{FP2} + \text{FNg2} + \text{TNt}}$$

2. *Precision*

   Precision mengukur seberapa banyak prediksi positif yang sebenarnya benar dibandingkan dengan total prediksi positif. *Precision* menunjukkan seberapa baik model dalam menghindari false positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP1} + \text{FP2}}$$

3. *Recall*

   Recall mengukur seberapa banyak dari kelas positif yang sebenarnya berhasil diidentifikasi oleh model. *Recall* menunjukkan seberapa baik model dalam menangkap semua kasus positif yang sebenarnya.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FNg1} + \text{FNt1}}$$

4. *F-1 score*

   F-1 score adalah perbandingan dari rata-rata presisi dan recall yang diboboti.

$$F-1Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.7 *Synthetic Minority Oversampling Technique* (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) adalah metode yang digunakan untuk mengatasi ketidakseimbangan data pada kelas minoritas. Teknik ini menambahkan contoh sintetik ke kelas minoritas dengan menggunakan setiap sampel dari kelas tersebut dan membuat contoh baru di sepanjang garis yang menghubungkan salah satu atau semua tetangga terdekat dari kelas minoritas [27].