BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen adalah studi komputasional terhadap opini, sentimen, dan emosi yang terungkap dalam sebuah teks [12]. Tugas utama dalam analisis sentimen adalah mengategorikan polaritas teks dalam dokumen, kalimat, atau pendapat yang menunjukkan apakah teks tersebut memiliki aspek positif, negatif, atau netral.

2.2 Alfagift

Alfagift adalah salah satu produk berupa *platform* belanja *online* yang didirikan dan dikembang pada tahun 2019 oleh PT. Global Loyalty Indonesia yang merupakan anak perusahaan dari Alfamart. Alfagift merupakan sebuah aplikasi yang memungkinkan pelanggan berbelanja secara daring dengan menyediakan manfaat keanggotaan yang terintegrasi baik dalam pengalaman berbelanja *online* maupun *offline* [4]. Alfagift memberikan kemudahan bagi masyarakat untuk bisa membeli produk dari Alfamart hanya dari rumah saja.

2.3 Preprocessing

Preprocessing merupakan fase krusial dalam pengembangan model *Natural Language Processing* (NLP). Pada tahap ini, teks disusun sedemikian rupa sehingga lebih terstruktur dan siap untuk tahapan berikutnya[13]. Beberapa prosedur yang dilakukan dalam *preprocessing* mencakup *cleaning* data, *case folding*, *tokenization*, penghapusan *stopword*, *normalization*, dan *stemming*.

2.3.1 Cleaning Data

Dalam tahap pembersihan data, teks dibersihkan dari angka, simbol, dan karakter yang tidak berhubungan. Penghapusan karakter *non*-alfabet diperlukan untuk menghindari pengaruh yang tidak diinginkan pada penelitian ini. Selain itu, karakter tunggal dan spasi berlebih juga dihilangkan dari teks untuk meningkatkan keteraturan struktur kalimat. Tujuan utama dari proses ini adalah membuat data menjadi lebih valid dan akurat [14].

2.3.2 Case Folding

Dalam tahap *case folding*, semua huruf dalam suatu teks akan diubah menjadi huruf kecil. *Case folding* menjadi salah satu hal yang penting, karena dapat mempermudah proses analisis teks lebih lanjut, karena menghilangkan perbedaan huruf besar kecil dalam kata yang sama [14].

2.3.3 Tokenization

Dalam tahapan *tokenization*, kalimat akan dipecah menjadi bagian-bagian kecil berupa kata atau frasa yang memiliki makna [15]. Proses *tokenization* ini akan membantu pengolahan teks menjadi lebih mudah untuk dianalisa dan dimengerti oleh mesin.

2.3.4 Normalization

Normalisasi adalah sebuah tahapan untuk mengatur struktur basis data dengan tujuan untuk meminimalisir adanya kata yang ambigu [16]. Normalisasi dalam penelitian ini berfungsi untuk mengubah kata *slang* atau singkatan, seperti ”yg” menjadi ”yang”, ”dgn” menjadi ”dengan”, dan lain sebagainya.

2.3.5 Penghapusan Stopword

Penghapusan *stopword* adalah proses menghilangkan kata-kata yang dianggap tidak memiliki nilai signifikan untuk analisis teks [15]. Kata-kata ini biasanya adalah kata-kata umum seperti ”dan”, ”atau”, ”tapi”, dan sejenisnya, yang ditemukan dalam banyak bahasa. Penghapusan *stopword* umumnya digunakan dalam tugas-tugas seperti pencarian informasi, analisis sentimen, dan lainnya untuk meningkatkan efisiensi pemrosesan dan fokus pada kata-kata yang lebih informatif dan relevan.

2.3.6 Stemming

Stemming merupakan teknik mengkonversi kata-kata yang memiliki awalan, akhiran, atau imbuhan lainnya ke dalam bentuk dasarnya [17]. Dalam konteks analisis sentimen, penerapan *stemming* dapat membantu meningkatkan keakuratan

hasil analisis dengan menyederhanakan kata-kata ke bentuk dasar mereka, sehingga memudahkan pengidentifikasian dan pemrosesan emosi atau opini dalam teks.

2.4 TF-IDF

TF-IDF atau *Term Frequency-Inverse Document Frequency* adalah sebuah cara *feature extaction* yang menggabungkan konsep *Term Frequency* dan *Inverse Document Frequency*. Perhitungan TF-IDF dihasilkan dengan mengalikan nilai *Term Frequency* dengan nilai *Inverse Document Frequency*, rumus perhitungan TF-IDF ditunjukkan pada persamaan 2.1. Dengan demikian, TF-IDF memberikan bobot yang lebih tinggi pada kata-kata yang sering muncul dalam satu dokumen tetapi jarang muncul di dokumen lainnya, sehingga dapat membantu dalam menemukan kata-kata kunci yang relevan dalam analisis teks [18].

$$w_{ij} = tf_{ij} \times idf \tag{2.1}$$

Keterangan:
w_{ij} = Hasil bobot kata i pada kelas j.
tf_{ij} = Hasil perhitungan *Term Frequency*.
idf = Hasil perhitungan *Inverse Document Frequency*.

2.4.1 Term Frequency

Term Frequency digunakan untuk mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen, rumus perhitungan TF ditunjukkan pada persamaan 2.2.

$$tf(t,d) = \frac{n_i j}{\sum_k n_i, j} \tag{2.2}$$

Keterangan:
$tf(t,d)$ = Frekuensi *term*.
$n_i j$ = Jumlah *term* yang muncul dalam satu dokumen.
$\sum_k n_i, j$ = Jumlah seluruh kata dalam satu dokumen.

2.4.2 Inverse Document Frequency

Inverse Document Frequency digunakan untuk mengukur seberapa jarang kata tersebut muncul di seluruh dokumen dalam koleksi data, rumus perhitungan IDF ditunjukkan pada persamaan 2.3.

$$idf = \log \frac{N}{df_j} \tag{2.3}$$

Keterangan:

idf = *Inverse Document Frequency*.

N: Jumlah kelas.

df_j: Jumlah kelas j yang berisi kata i.

2.5 SMOTE

Synthetic Minority Over-sampling Technique (SMOTE) digunakan untuk menangani ketidakseimbangan kelas dalam *dataset*. Teknik ini bekerja dengan menghasilkan sampel baru dari kelas minoritas untuk menyeimbangkan *dataset*. SMOTE menciptakan sampel baru dengan membentuk kombinasi konveks dari sampel-sampel minoritas yang berdekatan [19]. Dengan menggunakan metode ini, *dataset* dapat dibuat seimbang melalui pembuatan sampel sintetik, daripada hanya menduplikasi sampel yang ada.

2.6 Naïve Bayes

Algoritma *Naïve Bayes* digunakan untuk melakukan klasifikasi pada hasil text mining dalam analisis sentimen [20]. Metode ini termasuk dalam kategori metode *machine learning* yang memanfaatkan kalkulasi probabilitas dan statistik. Algoritma *Naïve Bayes* beroperasi dengan mekanisme prediksi kemungkinan kejadian di masa depan berdasarkan pengalaman masa lalu [21]. Algoritma ini utamanya berbasis pada Teorema Bayes, yang mengoperasikan dengan asumsi bahwa setiap atribut adalah independen satu sama lain, bila diberikan nilai pada variabel kelas [22]. Rumus perhitungan dasar dari algoritma *Naïve Bayes* dapat dilihat pada persamaan

Keterangan:

H: Data yang merupakan kelas spesifik.

X: Data dengan kelas yang belum diketahui.

P(H—X): Probabilitas hipotesis berdasarkan kondisi.

P(H): Probabilitas hipotesis.

P(X—H): Probabilitas berdasar kondisi pada hipotesis.

P(X): Probabilitas X.

Algoritma *Naïve Bayes* memiliki beberapa algoritma turunan, seperti *Multinomial Naive Bayes*, *Gasussian Naïve Bayes*, *Bernoulli Naive Bayes*, dan lain sebagainya. *Multinomial Naïve Bayes* adalah algoritma turunan *Naïve Bayes* yang sangat cocok untuk data diskrit, seperti kata-kata dalam dokumen teks [23]. *Multinomial Naive Bayes* menghitung probabilitas dari suatu kelas berdasarkan frekuensi kata-kata dalam dokumen. Kemudian, *Gaussian Naïve Bayes* digunakan ketika fitur mengikuti distribusi normal (*Gaussian*) [24]. Algoritma ini mengasumsikan bahwa data kontinu yang diinput memiliki distribusi normal. Oleh karena itu, GNB sering digunakan dalam tugas klasifikasi di mana fitur-fitur adalah variabel kontinu, sedangkan *Bernoulli Naïve Bayes* digunakan untuk data biner. Algoritma ini mengasumsikan bahwa fitur memiliki nilai biner (0 atau 1) yang menunjukkan apakah kata tertentu ada atau tidak dalam dokumen [25]. Berdasarkan ketiga jenis turunan dari algoritma *Naïve Bayes* tersebut, algoritma yang paling cocok untuk analisa sentimen pada konteks penelitian ini adalah *Multinonmial Naïve Bayes*.

2.6.1 Multinomial Naïve Bayes

Dalam penelitian ini menggunakan *Multinomial Naïve Bayes*, dimana algoritma ini merupakan varian dari model *Naïve Bayes* yang telah terbukti efektif dalam masalah klasifikasi teks. Model ini mengasumsikan bahwa setiap fitur pada setiap kelas adalah independen dan mengabaikan semua ketergantungan antar atribut [23]. Perhitungan probabilitas dijelaskan pada persamaan 2.4 [26].

$$P(c \mid d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k \mid c) \tag{2.4}$$

Keterangan:

$P(c \mid d)$: Probabilitas kelas c diberikan dokumen d.

$P(c)$: Probabilitas priori munculnya dokumen pada kelas c.

$\prod_{1 \leq k \leq n_d} P(t_k \mid c)$: Produk dari probabilitas kemunculan kata t_k di kelas c.

t_k: kata-kata yang ada dalam dokumen d.
n_d: jumlah kata dalam dokumen d.

Penentuan kelas dilakukan dengan membandingkan hasil probabilitas posterior yang diperoleh, kemudian kelas yang mempunyai probabilitas *posterior* terbesar adalah kelas yang dipilih sebagai hasil prediksi [26]. Rumus probabilitas prior ditunjukkan pada persamaan 2.5.

$$P(c) = \frac{N_c}{N} \tag{2.5}$$

Keterangan:
$P(c)$: Probabilitas priori dari kelas c.
N_c : Jumlah dokumen yang termasuk dalam kelas c.
N : Total jumlah dokumen dalam keseluruhan koleksi.

Rumus peluang kemungkinan ditunjukkan pada persamaan 2.6 [26].

$$P(t_k \mid c) = \frac{T_{tc}}{\sum_{t' \in V} T_{ct'}} \tag{2.6}$$

Keterangan:
$P(t_k \mid c)$: Probabilitas kemunculan kata t_k diberikan kelas c.
T_{tc} : Jumlah kemunculan kata t_k dalam dokumen yang masuk pada kelas c.
$\sum_{t' \in V} T_{ct'}$: Jumlah total kemunculan semua kata dalam kelas c.
V : himpunan dari seluruh kata (*vocabulary*).

2.7 Confusion Matrix

Confusion Matrix adalah salah satu metode yang sering digunakan untuk mengukur tingkat akurasi dalam bidang data *mining*. *Confusion matrix* juga merupakan representasi dari hasil klasifikasi biner pada sebuah dataset [27]. Metode ini merupakan alat yang efektif untuk mengevaluasi model yang telah di latih karena *confusion matrix* dapat diterapkan baik pada masalah klasifikasi biner maupun *multiclass* [28].

Tabel 2.1. *Confusion Matrix*

	Nilai Aktual Positif	Nilai Aktual Negatif
Kelas Prediksi Positif	*True Positive*	*False Positive*
Kelas Prediksi Negatif	*False Negative*	*True Negative*

Sumber: [28]

Keterangan:

- *True Positive* (TP) : Hasil prediksi data positif yang terdeteksi benar.
- *False Positive* (FP) : Hasil prediksi data positif yang terdeteksi salah.
- *False Negative* (FN) : Hasil prediksi data negatif yang terdeteksi salah.
- *True Negative* (TN) : Hasil prediksi data negatif yang terdeteksi benar.

Kemudian, setelah mendapatkan hasil dari pelabelan *confusion matrix*, maka akan dilakukan tahapan proses untuk menghitung akurasi, *precision*, *recall*, dan *F-1 Score*.

1. Akurasi

   Akurasi adalah rasio prediksi yang benar terhadap total prediksi yang dilakukan [27]. Perhitungan dapat dilihat dalam rumus berikut.

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+TN+FN} \tag{2.7}$$

2. *Precision*

   Precision adalah proporsi dari dokumen teks yang relevan dibandingkan dengan total dokumen yang dipilih [27]. Cara perhitungan dapat dilihat dalam rumus berikut.

$$Precision = \frac{TP}{TP+FP} \tag{2.8}$$

3. *Recall*

   Recall adalah proporsi dari teks dokumen yang relevan yang berhasil ditemukan dibandingkan dengan total jumlah teks dokumen relevan yang ada dalam kumpulan data [27]. Dengan kata lain, perhitungan ini untuk menentukan model berhasil dalam menemukan kembali suatu informasi yang terdapat dalam data. Cara perhitungan dapat dilihat dalam rumus berikut.

$$Recall = \frac{TP}{TP+FN} \tag{2.9}$$

4. *F-1 Score*

   F1-Score adalah indikator kesuksesan dalam mengambil parameter tunggal yang mengintegrasikan *recall* dan *precision*. Hasilnya diperoleh dari perkalian precision dan recall, dibagi dengan jumlah *precision* dan *recall*, kemudian hasilnya dikalikan dengan 2 [29]. Perhitungan *f1-score* dapat dilihat dalam rumus berikut.

$$F\ 1\text{- }Score\ = 2 \times \frac{Precision\ \times\ Recall}{Precision\ +\ Recall} \tag{2.10}$$