BAB 2
LANDASAN TEORI

2.1 Amazon

Amazon adalah salah satu perusahaaan *e-commerce* terbesar di dunia yang didirikan oleh Jeff Bezos pada tahun 1994. Awalnya Amazon hanya menjual buku secara *online*. Setelah itu, pada tahun 1998, Amazon berkembang dan mulai menjual *game* untuk PC dan rekaman-rekaman lagu [2]. Dan setelah bertahun-tahun, bisnis Amazon semakin berkembang dan kini Amazon menjual banyak hal mulai dari kebutuhan sehari-hari, barang-barang unik hingga layanan komputer seperti layanan *web*, layanan penyimpanan data dan layanan *cloud computing*. Amazon memiliki kantor pusat di Seattle, Washington dan kini memiliki banyak cabang di negara-negara lain seperti Mexico, Poland dan Singapore [2].

2.2 Machine Learning

Machine learning atau pembelajaran mesin memungkinkan pengguna untuk memberikan ilmu kepada komputer agar komputer dapat menganalisa dan membuat rekomendasi ataupun keputusan berdasarkan data yang diberikan sehingga merupakan pilihan tepat untuk melakukan klasifikasi [8]. Pemrosesan data dalam *machine learning* terbagi 2 yaitu *supervised learning* dan *unsupervised learning*. Penelitian ini menggunakan *supervised learning*. *Supervised learning* adalah teknik yang digunakan untuk mengetahui hubungan antara atribut *input* dan atribut *target* (label) [9]. Algoritma ini bertujuan untuk memperkirakan fungsi pemetaan sehingga ketika ada variabel *input* (X) kita dapat memprediksi variabel *output* (Y). Variabel *output* (Y) juga sering disebut label. Dalam penelitian ini label-labelnya adalah positif dan negatif. Algoritma *supervised learning* dapat digunakan untuk memproses berbagai jenis data, mulai data yang terstruktur hingga yang tidak terstruktur [10].

2.3 Natural Language Processing

Natural Language Processing (NLP) adalah salah satu jenis pemrosesan data pada *machine learning* yang mempelajari komunikasi antara manusia dan komputer [5]. Dengan NLP, komputer akan dapat memproses bahasa manusia dengan baik.

Salah satu metode dalam NLP adalah *sentiment analysis*. *Sentiment analysis* atau analisis sentimen adalah proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Analisis sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek yang dikemukakan oleh seseorang, apakah cenderung berpandangan negatif atau positif [4]. Analisis sentimen biasanya dilakukan terhadap data-data yang mengandung banyak opini seperti berita dan komentar media sosial.

2.4 Preprocessing

Preprocessing adalah tahap yang dilakukan sebelum data dibuat menjadi model. Karena data tidak terstruktur dan memiliki banyak *noise*, diperlukan *text preprocessing* untuk membuat data lebih terstruktur dan menghapus *noise* [11]. Tahap-tahap yang dilalui dalam *text preprocessing* pada penelitian ini adalah sebagai berikut.

Handling Duplicate

Handling duplicate berfungsi sebagai penyaring sebuah data yang sedang diteliti untuk menghindari data memiliki jumlah lebih dari satu dari sekelompok data yang digunakan [7].

Remove Empty Values

Pada tahap ini, data yang kosong atau memiliki *empty value* akan dihapus.

Casefolding

Casefolding berfungsi mengubah semua kararkter dengan huruf besar menjadi huruf kecil [12].

Data Cleaning

Data cleaning berfungsi menghilangkan komponen-komponen yang tidak diinginkan yang ada dalam data yang digunakan pada proses penelitian [7]. Komponen-komponen tersebut misalnya simbol-simbol yang tidak diperlukan.

Tokenization

Tokenization berfungsi memotong dokumen inputan berdasarkan tiap kata yang menyusunnya [12].

Stopword Removal

Stopword removal berfungsi memilih kata-kata penting yang memunyai arti dan tidak sehingga kata-kata yang tidak mempunyai arti akan dibuang [12]. Dalam kasus ini kata-kata yang tidak penting misalnya ”*but*”, ”*as*” dan ”*are*”.

Lemmatization

Lemmatization adalah proses normalisasi kata untuk menemukan bentuk dasar dari kata tersebut berdasarkan lemmanya [12]. Lemma tersebut bisa berbentuk *adjective* (kata sifat) atau *verb* (kata kerja) atau *noun* (kata benda) atau *adverb* (kata keterangan).

Labelling

Labelling adalah melabeli data menjadi bentuk *positive* atau *negative* [13].

2.5 Term Frequency - Inverse Document Frequency

Term Frequency - Inverse Document Frequency (TF-IDF) adalah metode yang digunakan untuk menghitung bobot kata dalam sebuah dokumen. Bobot kata ini digunakan untuk menentukan seberapa penting kata tersebut dalam dokumen tersebut [14]. Dengan demikian, kata-kata yang sering muncul dalam dokumen tertentu tetapi jarang muncul dalam dokumen-dokumen lain dalam sebuah *corpus* akan memiliki bobot yang tinggi, menunjukkan tingkat keunikan dan relevansi kata tersebut terhadap dokumen tersebut [14].

Term Frequency

Term Frequency (TF) adalah konsep pembobotan dengan mencari seberapa sering munculnya sebuah kata (*term*) dalam satu dokumen [15]. Rumus TF adalah sebagai berikut [14].

$$TF(t,d) = \frac{n_t}{n} \tag{2.1}$$

Dimana:
TF(t,d) : Frekuensi sebuah *term* (t) dalam dokumen (d)
n_t : Total suatu *term*(t) dalam suatu dokumen
n : Total dari seluruh kata dalam suatu dokumen

Misalnya jika sebuah kata (*term*) muncul 4 kali di sebuah dokumen dengan total 20 kata. Maka TF dihitung dengan cara:

$$TF(t,d) = \frac{4}{20} = 0.2$$

Inverse Document Frequency

Inverse Document Frequency (IDF) adalah banyaknya jumlah dokumen di mana sebuah *term* itu muncul [15]. Kata yang hanya terdapat dalam sedikit dokumen seperti istilah teknikal akan mendapat *score* IDF yang lebih tinggi daripada kata-kata yang sering muncul dalam dokumen-dokumen seperti *is*, *am* dan *are*. Rumus IDF adalah sebagai berikut [14].

$$IDF(t) = \frac{N_d}{N} \tag{2.2}$$

Dimana:
IDF(t) : Frekuensi jumlah dokumen di dalam *corpus* yang mengandung *term* (t)
N_d : Total dokumen di dalam *corpus*
N : Total dokumen di dalam *corpus* yang mengandung *term* (t)

Misalnya jika sebuah kata (*term*) terdapat di 100 dokumen dan ada total 10000 dokumen di sebuah *corpus* (*dataset*), maka nilai IDF dihitung dengan cara:

$$IDF(t) = \frac{100}{10000} = 0.01$$

2.5.1 Term Frequency - Inverse Document Frequency

Dengan mempertimbangkan nilai TF dan IDF, rumus TF-IDF adalah sebagai berikut [14].

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

Dimana:
TF-IDF(t,d) : Bobot sebuah *term* (t) dalam sebuah dokumen (d)
TF(t,d) : Frekuensi jumlah *term* (t) dalam dokumen (d)
IDF(t) : Frekuensi banyak dokumen dalam *corpus* yang mengandung *term* (t)

Berdasarkan perhitungan TF dan IDF sebuah kata (*term*) yang telah dicontohkan di atas, maka nilai TF-IDF dari kata tersebut adalah:

$$TF\text{-}IDF(t,d) = 0.2 \times 0.01 = 0.002$$

2.6 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE (*Synthetic Minority Oversampling Technique*) digunakan untuk mengatasi ketidakseimbangan kelas pada *dataset*. Data-data pada kelas yang memiliki populasi yang rendah akan ditambah jumlahnya [14]. Pada penelitian ini, data dengan kelas 0 (negatif) akan ditambah karena jumlahnya lebih sedikit.

2.7 Naive Bayes

Naive Bayes adalah salah satu metode probabilitas yang digunakan untuk melakukan klasifikasi yang ditemukan oleh ilmuwan yang bernama Thomas Bayes [15]. Algoritma *Naive Bayes* memiliki cara kerja dengan memprediksi kemungkinan di masa depan berdasarkan kejadian di masa lalu [13]. Rumus *Naive Bayes* adalah sebagai berikut [15].

$$P(H \mid X) = \frac{P(X \mid H) \cdot P(H)}{P(X)} \tag{2.4}$$

Dimana:
P(H|X) : Probabilitas kelas (H) berdasarkan prediktor (X)
P(H) : Probabilitas kelas (H)
P(X|H) : Probabilitas prediktor (X) berdasarkan kelas (H)
P(X) : Probabilitas prediktor (X)

Algoritma *Naive Bayes* menghitung probabilitas data yang baru terhadap *classes* yang ada menggunakan *Bayes Theorem* seperti rumus di atas [13].

2.8 Logistic Regression

Logistic Regression adalah bagian dari dari analisis regresi yang digunakan ketika variabel (respon) merupakan variabel dikotomi yaitu yang terdiri dari dua nilai yang mewakili muncul atau tidaknya suatu kejadian yang biasanya diberi angka 0 atau 1. Tidak seperti regresi linier biasa, regresi logistik tidak mengasumsikan hubungan antara variabel independen dan dependen secara linier. Regresi logistik adalah sebuah pendekatan untuk membuat model prediksi seperti halnya regresi linear atau yang biasa disebut *Ordinary Least Squares* (OLS). Perbedaannya adalah pada regresi logistik, peneliti memprediksi variabel terikat yang berskala dikotomi. Skala dikotomi yang dimaksud adalah data nominal dengan dua kategori, misalnya ya dan tidak, baik dan buruk atau tinggi dan rendah [6]. Rumus *Linear Regression* dan *Logistic Regression* adalah sebagai berikut.

Simple Linear Regression

Rumus *Simple Linear Regression* adalah sebagai berikut [16].

$$\begin{aligned} y &= \alpha + \beta x \\ g(x) &= \alpha + \beta x \end{aligned} \tag{2.5}$$

Multiple Linear Regression

Rumus *Multiple Linear Regression* adalah sebagai berikut [16].

$$\begin{aligned} y &= \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \\ g(X) &= \alpha + \beta X \end{aligned} \tag{2.6}$$

Logistic Regression

Rumus *Logistic Regression* adalah sebagai berikut [14].

$$g(x) = sigmoid(\alpha + \beta X) \tag{2.7}$$

$$sigmoid(x) = \frac{1}{1 + \exp^{-x}} \tag{2.8}$$

Dimana:

y: Variabel *response* atau variabel akibat (dependen)

α : Konstanta

β : Koefisien regresi (kemiringan), besaran *response* yang ditimbulkan oleh prediktor.

x : Variabel prediktor atau variabel faktor penyebab (independen)

X : Setiap variabel prediktor atau faktor penyebab (independen) yang dikali dengan setiap koefisien regresi

2.9 Confusion Matrix

Confusion matrix adalah salah satu cara yang sampai dengan saat ini masih dianggap efektif untuk mengukur dan mengevaluasi kinerja dari sebuah model klasifikasi [17]. Ilustrasi *confusion matrix* untuk mengukur dan mengevaluasi kinerja klasifikasi khususnya pada klasifikasi 2 kelas (*binary classification*) dapat dilihat pada tabel dibawah ini [18].

	Diidentifikasikan Sebagai Tidak Layak	Diidentifikasikan Sebagai Layak
Tidak Layak	a	b
Layak	c	d

Gambar 2.1. *Confusion Matrix*

Dimana:

a = *True Negative* (TN)

b = *False Positive* (FP)

c = *False Negative* (FN)

d = *True Postive* (TP)

Penggunaan *confusion matrix* bertujuan untuk mengetahui seberapa besar nilai prediksi yang dihasilkan oleh sistem untuk kemudian dibandingkan dengan nilai aktual dari datanya. Penilaian yang dilakukan berdasarkan indikator dari *confusion matrix* diambil dari banyaknya nilai pada komponen *True Positive* (TP), *False Positive* (FP), *True Negative* (TN) dan *False Negative* (FN). TP merupakan komponen yang menunjukkan banyaknya jumlah prediksi nilai positif terhadap keseluruhan data yang secara aktual juga bernilai positif. Sedangkan FP, merupakan komponen yang menunjukkan banyaknya jumlah prediksi nilai positif namun secara aktual bernilai negatif. Berikutnya adalah FN, yang berisi banyaknya jumlah data yang diprediksi bernilai negatif, namun secara aktualnya bernilai positif dan TN menunjukkan banyaknya jumlah prediksi sistem yang bernilai negatif, dengan data aktual juga bernilai negatif. Nilai TP, FP, TN dan FN selanjutnya digunakan untuk menghitung *precision*, *accuration*, *recall* dan *f1-score* [17].

Accuration

Accuration atau akurasi merupakan metrik yang digunakan untuk mengetahui proporsi dari jumlah prediksi sistem yang bernilai benar (TP dan TN) dengan jumlah dari keseluruhan hasil prediksi (TP, FP, FN, TN). Berdasarkan tabel pada gambar 1, rumus akurasi adalah sebagai berikut [18].

$$accuracy = \frac{(a+d)}{(\ total\ sample\ )} \times 100\% \tag{2.9}$$

Precision

Precision atau presisi adalah metrik untuk mengukur seberapa besar rasio prediksi nilai positif yang sesuai dengan nilai aktual (TP) dibandingkan dengan keseluruhan hasil prediksi yang bernilai positif (TP dan FP). Berdasarkan tabel pada gambar 1, rumus presisi adalah sebagai berikut [18].

$$precision = \frac{(d)}{(b+d)} \times 100\% \tag{2.10}$$

Recall

Recall adalah metrik yang bertujuan untuk mengetahui proporsi jumlah data yang diprediksi bernilai positif (TP) dibanding dengan seluruh data yang secara aktual bernilai positif (TP dan FN). Berdasarkan tabel pada gambar 1, rumus *recall* adalah sebagai berikut [18].

$$recall = \frac{(d)}{(c+d)} \times 100\% \tag{2.11}$$

F1-Score

F1-Score adalah metrik pengukuran untuk mengetahui perbandingan rata-rata antara nilai *precision* dan nilai *recall*. Rumus *f1-score* adalah sebagai berikut [17].

$$F1 = 2x\frac{Recall\ x\ Precision}{(\ Recall\ +\ Precision\ )} \tag{2.12}$$