BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen merupakan sistem yang digunakan untuk melakukan proses analisis otomatis untuk memperoleh informasi meliputi informasi sentimen yang merupakan bagian dari opini online[10]. Proses analisis sentimen berupa analisis atau identifikasi dari opini yang merupakan sikap terhadap suatu topik atau produk tertentu yang terbagi dalam beberapa kategori yaitu positif, negatif, dan netral. [11]

2.2 Machine Learning

Machine learning merupakan sebuah sistem yang mampu untuk belajar sendiri untuk memutuskan sesuatu tanpa harus berulangkali diprogram oleh manusia sehingga komputer dapat berkembang secara mandiri berdasarkan pengalaman data yang dimiliki[12]. Machine learning dibagi menjadi 3 kategori, yaitu supervised learning, unsupervised learning, dan reinforcement learning. Secara singkat, supervised learning memerlukan data berlabel untuk melakukan analisis data dan juga pelatihan. Sebaliknya dengan unsupervised learning tidak memerlukan labeling pada data sehingga lebih cocok untuk masalah yang tidak beraturan. Sedangkan untuk reinforcement learning berbasis trial dan error yang berarti melakukan percobaan pada lingkungannya untuk meningkatkan kemampuan.[13]

2.3 Naive Bayes

Algoritma Naive Bayes merupakan metode klasifikasi sederhana. Algoritma Naive Bayes memanfaatkan teorema probabilitas yaitu mencari peluang terbaik dengan melakukan prediksi probabilitas di masa depan berdasarkan informasi di masa sebelumnya [14]. Berikut adalah rumus perhitungan dengan menggunakan algoritma Naive Bayes:

$$P(C) = \frac{N_c}{N} \tag{2.1}$$

Keterangan:

N_c = Jumlah sampel dengan kelas C.

N = Total jumlah sampel.

Persamaan ini berguna untuk menghitung probabilitas A priori. Persamaan ini menghitung probabilitas kemunculan kelas C secara keseluruhan kelas C secara keseluruhan tanpa mempertimbangkan fitur yang diamati.

$$P(X|C) = P(X1|C) \times P(X2|C) \times ... \times P(Xn|C) \tag{2.2}$$

Keterangan:

$P(Xi|C)$ = Probabilitas kemunculan fitur ke-i dalam kelas C.

Persamaan ini adalah untuk menghitung probabilitas *likelihood* kemunculan fitur X berdasarkan kelas C dengan asumsi bahwa fitur tersebut independen satu sama lain.

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{2.3}$$

Keterangan:

$P(X)$ = Probabilitas kemunculan fitur X secara keseluruhan.

Persamaan ini berguna untuk menghitung probabilitas posterior dari kelas C berdasarkan fitur X. Persamaan ini menghitung probabilitas kelas C berdasarkan fitur yang diamati.

Terdapat beberapa algoritma dalam Naive Bayes, yang pertama adalah Bernoulli Naive Bayes. Bernoulli merupakan model yang hanya memperhatikan kemunculan data dalam dokumen dan mengabaikan frekuensi dari data yang muncul [15]. Kedua adalah Complement Naive Bayes, Complement Naive Bayes merupakan pengembangan dari algoritma Naive bayes yang cenderung memiliki performa lebih baik ketika dataset tidak seimbang [16]. Ketiga adalah Multinomial Naive Bayes, Multinomial Naive Bayes cocok dalam klasifikasi teks atau dokumen tanpa memperhitungkan urutan kata ataupun konteks informasi

melainkan menghitung jumlah kemunculan kata dari dokumen [17]. Terakhir adalah Gaussian Naive Bayes, Gaussian merupakan algoritma yang berbasis nilai kontinu dengan konsep probabilitas yang digunakan untuk menentukan kelas dari dokumen [18].

2.4 Text/Data Mining

Text atau *data mining* merupakan proses menambang data yang berupa teks dimana sumber data didapatkan dari dokumen dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisis keterhubungan antar dokumen [19]. Data mining melakukan eksplorasi pengetahuan dan pola data dengan menggunakan statistik matematika dan juga machine learning. Seiring berjalannya waktu, data akan terus bertambah sehingga muncul aliran data yang besar, maka dari itu algoritma data mining memiliki peran yang efektif untuk menganalisis data besar tersebut.[20]

2.5 Text Preprocessing

Text preprocessing adalah tahap dimana data dipersiapkan untuk menjadi data yang siap dianalisis[21]. Dalam melakukan text preprocessing, terdapat beberapa tahap yaitu remove duplicate, cleansing, case folding, stemming dan tokenizing. Berikut adalah penjelasan singkat mengenai tahap-tahap tersebut.

2.5.1 Remove Duplicate

Remove Duplicate berfungsi untuk membuang data yang bernilai sama. Tahap ini dilakukan untuk menghindari pengulangan data atau spam yang dapat menyebabkan hasil akhir menjadi tidak akurat [22].

2.5.2 Cleansing

Cleansing berfungsi untuk menghapus atau memodifikasi data yang salah, tidak relevan, tidak akurat, maupun yang tidak terformat. Proses yang dilakukan adalah mendeteksi kesalahan data, memperbaiki atau menghapus data sesuai dengan kebutuhan. Contoh dari cleansing adalah menghapus karakter yang bukan alfabet [23]. Berikut adalah contoh dari cleansing :

2.5.3 Case Folding

Case folding berfungsi untuk mengubah huruf dalam kalimat menjadi huruf kecil, dan hanya menerima huruf a-z. Karakter selain huruf dihilangkan dari tweets tersebut [24].

2.5.4 Tokenizing

Tokenizing adalah sebuah proses pemisahan sebuah kata. Berikut adalah contoh penerapan dari tokenizing:

Tabel 2.1. Contoh *Tokenizing*

Sebelum Tokenisasi	Setelah Tokenisasi
resesi membuat warga menjadi resah	”resesi”, ”membuat”, ”warga”, ”menjadi”, ”resah”

2.5.5 Remove Stop Words

Proses ini berguna untuk membuang kata yang tidak memiliki makna seperti misalnya adalah kata hubung. Dalam sebuah data terdapat banyak kata yang tidak diperlukan, oleh karena itu remove stop words diperlukan agar tidak menimbulkan kekacauan dalam proses analisa data[25].

2.5.6 Stemming

Proses stemming adalah proses pemetaan dan penguraian dari berbagai bentuk kata menjadi sebuah kata dasar. Proses stemming secara luas digunakan dalam kegiatan pencarian informasi untuk meningkatkan kualitas data yang didapatkan [26].

2.6 Count Vectorizer

Count Vectorizer adalah proses pengolahan dokumen atau teks menjadi sebuah vektor. Count Vectorizer berguna untuk menghitung frekuensi kemunculan kata dalam dokumen lalu dipresentasikan dalam bentuk vektor [27].

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

TF-IDF merupakan sebuah metode untuk memberikan bobot hubungan suatu kata(*term*) terhadap suatu dokumen berdasarkan 2 konsep, yaitu frekuensi kemunculan data(TF) di dalam sebuah dokumen dan inverse frekuensi dokumen yang mengandung kata tersebut(IDF)[28]. Berikut adalah rumus perhitungan TF-IDF:

$$TF - \text{IDF}_{t,d} = \text{TF}_{t,d} \times iDF_t \tag{2.4}$$

$$TF_{t,d} = \frac{n_{i,j}}{\sum_k n_{i,j}} \tag{2.5}$$

$$IDF_t = log\frac{N}{DF_t} \tag{2.6}$$

Keterangan:

t = kata-kata yang dihitung
d = bobot kalimat
$TF - IDF_{t,d}$ = kalimat bobot (d) terhadap kata (t)
$TF_{t,d}$ = *term frequency*
IDF_t = *inverse document frequency*
N = jumlah kalimat
DF_t = jumlah kata yang terulang
$n_{i,j}$ = total term yang muncul pada satu dokumen
$\sum_k n_{i,j}$ = total seluruh kata dalam satu dokumen

2.8 Confusion Matrix

Confusion Matrix merupakan sebuah tabel yang kerap digunakan dalam mengukur kinerja dari model klasifikasi machine learning. Tabel ini berguna untuk menggambarkan rincian mengenai klasifikasi yang benar ataupun salah [29]. Contoh hasil dari confusion matrix dapat dilihat pada Tabel 2.2

Tabel 2.2. Confusion Matrix

	Data Aktual	
Data Prediksi	**Positive**	**Negative**
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Keterangan:
True Positive = data positif yang terklasifikasi dengan benar.
True Negative = data negatif yang terklasifikasi dengan benar.
False Positive = data negatif yang terklasifikasi menjadi positif.
False Negative = data positif yang terklasifikasi menjadi negatif.
Rumus yang digunakan untuk menghitung performa dalam *confusion matrix* adalah sebagai berikut:

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

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

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

$$f1-score = \frac{2\ x\ Precision\ x\ Recall}{Recall+Precision} \tag{2.10}$$

2.9 Near Miss

Near miss adalah proses dimana data mayoritas dibuang secara acak. Dengan dibuangnya data tersebut, maka jumlah data mayoritas menjadi setara dengan jumlah data minoritas. Proses ini biasa dilakukan ketika mendapatkan data yang tidak balance. Namun terdapat kelemahan dari proses *near miss* yaitu karena

penghapusan data dilakukan secara random maka terdapat kemungkinan data yang dibuang adalah data yang penting untuk model klasifikasi [30]. Near miss terbagi menjadi 3 versi, antara lain adalah [31] :

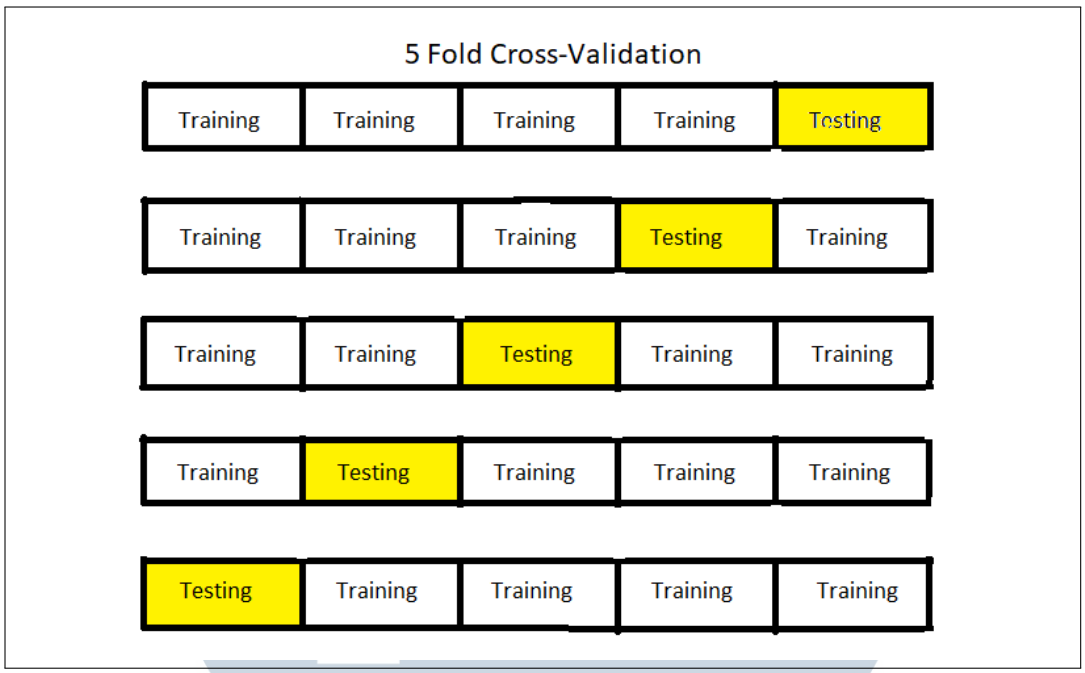
1. Near Miss versi 1, memilih contoh dari kelas mayoritas di mana rata-rata jarak dari contoh terdekat dari kelas minoritas paling kecil.
2. Near Miss versi 2, memilih contoh dari kelas mayoritas di mana jarak rata-rata ke contoh terjauh dari kelas negatif paling kecil
3. Near Miss versi 3, proses ini memperlukan 2 langkah, yang pertama adalah tetangga terdekat akan dipertahankan, kemudian sampel kelas mayoritas yang diambil adalah yang memiliki jarak terdekat terhadap kelas minoritas.

2.10 Synthetic Minority Over-Sampling Technique (SMOTE)

Metode Synthetic Minority Over-Sampling Technique atau biasa yang disebut dengan SMOTE adalah teknik yang digunakan berdasarkan prinsip over sampling yang memiliki arti menambah data dari kelas minor agar jumlah kelasnya sama dengan data dari kelas mayor [32]. Cara kerja dari SMOTE adalah dengan memilih baris mathching secara acak yang kemudian digunakan kombinasi konveks untuk sampel baru [33].

2.11 K-Fold Cross Validation

Cross validation merupakan teknik validasi dengan membagi data menjadi k-grup yang nantinya pengujian juga dilakukan sebanyak jumlah k yang telah ditentukan. Pengujian dilakukan secara bergantian antara data training dan data testing. Data training adalah data yang akan digunakan untuk proses pembelajaran, sedangkan data testing adalah data yang belum digunakan untuk pembelajaran dan berfungsi untuk memeriksa akurasi [34]. Untuk lebih jelasnya dapat dilihat pada gambar berikut :



Gambar 2.1. Contoh Cross Validation

Pada gambar 4.22 adalah contoh dari implementasi cross validation dengan contoh k = 5. Misal dalam gambar tersebut terdapat 100 data, pada percobaan pertama data 1-20 adalah data yang digunakan untuk testing, sisanya adalah data training. Percobaan kedua 1-20 adalah data train lalu 21-40 data test, dan 41-100 adalah data train dan begitu seterusnya. Setelah tiap fold sudah dihitung hasilnya, maka akurasi final akan dihitung melalui rata-rata dari hasil setiap fold.