BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Metode Naïve Bayes, Decision Tree, dan Random Forest dapat menganalisis sentiment Sosial Media Twitter tentang maraknya Anti LGBT yang ada di Indonesia (Veny, dkk, 2019). Metode yang digunakan dalam mengklasifikasi analisis sentimen yaitu Naive Bayes Classifier. Naive Bayes Classifier dikombinasikan dengan fitur untuk dapat mendeteksi negasi dan pembobotan menggunakan term frequency serta TF-IDF. Klasifikasi tweet pada penelitian ini diperoleh berdasarkan kombinasi antara kelas sentimen dan kelas kategori. Klasifikasi sentimen terdiri dari positif dan negatif sedangkan klasifikasi kategori terdiri dari kapabilitas, integritas, dan akseptabilitas. Hasil penelitian tersebut memperlihatkan bahwa akurasi dengan term frequency memberikan hasil akurasi yang lebih baik daripada akurasi dengan fitur TF-IDF, Metode Naïve Bayes menghasilkan akurasi performansi yang lebih baik daripada metode Decision Tree maupun Random Forest dengan nilai akurasi 83,43%. Namun demikian, secara keseluruhan penggunaan metode Naive Bayes, Decision Tree, dan Random Forest sama-sama memiliki performansi yang cukup baik untuk melakukan klasifikasi tweet.

Selanjutnya pada metode klasifikasi support vector machine dalam analisis sentimen cyberbullying pada komentar Instagram (Wanda Athira Luqyana, dkk, 2018).Metode ini dilakukan dengan mengetahui setiap sentimen pada komentar digunakan fitur Term Frequency-Inverse Document Frequency (TF-IDF) dan metode klasifikasi Support Vector Machine (SVM), dokumen yang berisi 400 data yang diambil secara luring (offline) dengan total fitur 1799. Dokumen komentar dibagi menjadi 70% data latih dan 30% data uji. Berdasarkan pengujian yang dilakukan didapatkan parameter terbaik pada metode SVM yaitu dengan nilai degree kernel polynomial sebesar 2, nilai learning rate sebesar 0,0001, dan jumlah iterasi maksimum yang digunakan adalah 200 kali. Hasil penelitian tersebut menunjukan bahwa yang didapatkan hasil akurasi tertinggi sebesar 90% pada komposisi data latih 50% dan komposisi data uji 50%. Selanjutnya pada metode K-Nearest Neighbor (K-NN) dalam menganalisis sentimen komentar twitter terhadap

new normal masa covid-19 di Indonesia (Mhd. Furqan, dkk, 2022). Metode yang dilakukan dengan menggunakan algoritma K-Nearest Neighbor (K-NN) dengan data sebanyak 1.000 tweet dengan keyword #NewNormal. Hasil penelitian tersebut menunjukan dengan klasifikasi K-NN dengan nilai k = 1 menghasilkan pengujian use training set dengan accuracy sebesar 100%, 92,60% untuk 10-fold cross-validation dan 94,50% untuk 80% percentage split.

Pada metode hybrid dalam mengintegrasikan analisis sentimen dan pasar utama indikator prediksi tren pencatatan IPO (Ashish Garg dkk., 2023). Metode ini dilakukan dengan mengembangkan model yang menggabungkan analisis sentimen publik dan pendekatan pembelajaran mesin untuk mengoptimalkan ROI untuk tren IPO. Studi ini memberikan pendekatan baru yang menggunakan berbagai fitur berbeda yang terkait dengan IPO seperti opini publik, harga pasar abu-abu (GMP), harga penerbitan, ukuran lot, dll. Hasil pengujian menunjukkan bahwa Decision Tree mengungguli algoritma lainnya, dengan tingkat akurasi 82,3%. Selanjutnya pada Implementasi Algoritma K-Nearest Neighbor (K-NN) Untuk Analisis Sentimen Masyarakat Terhadap Pembelajaran Daring (Auliya Rahman Isnain, dkk, 2021). Metode ini yang dilakukan dengan menggunakan K-Nearest Neighbor (K-NN) dengan menggunakan data tweet sebanyak 1825 data berbahasa Indonesia. Hasil penelitian tersebut menunjukan bahwa yang didapatkan akurasi sebesar 84,65%, presisi 87% dan recall 86% dengan menggunakan K = 10.

2.2 Dasar Teori

2.2.1 Sentimen Analisis

Sentimen Analisis atau opinion mining merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Sentimen Analysis dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang, apakah cenderung berpandangan atau beropini negatif atau positif (Liu, 2012). Sentimen Analysis dapat dibedakan berdasarkan sumber datanya, beberapa level yang sering digunakan dalam penelitian Sentimen

Analisis adalah Sentimen Analysis pada level dokumen dan Sentimen Analisis pada level kalimat. Berdasarkan level sumber datanya Sentimen Analisis terbagi menjadi 2 kelompok besar yaitu :

1. Coarse-grained Sentiment Analysis

   Coarse-grained Sentiment Analysis adalah teknik analisis sentimen yang membagi sentimen ke dalam beberapa kategori atau level yang lebih umum. Biasanya, ada tiga kategori umum yang digunakan dalam coarse-grained sentiment analysis: positif, negatif, dan netral. Dalam melakukan analisis terkait opini publik terhadap suatu product biasanya Coarse grained biasa di gunakan suatu perusahaan untuk mengetahui positif atau negatifnya respon publik terhadap product atau layanan perusahaan.

2. Fined-grained Sentiment Analysis

   Berbeda dengan coarse-grained, Fined-grained Sentiment Analysis adalah teknik analisis sentimen yang lebih detail seperti marah, senang, sedih, kecewa dan lainnya.analisis menggunakan Fined-grained Sentiment relatif digunakan saat melakukan analisis di sosial media survei atau data teks yang mampu direpresentasikan sebagai sebuah respon konsumen terhadap suatu produk yang dikeluarkan oleh sebuah perusahaan.

2.2.2 Web Scraping

Web Scraping merupakan keseluruhan proses mengekstrasi data seperti teks, gambar, video, dan metadata untuk menemukan informasi berguna dan halaman situs web baik secara Sebagian maupun keseluruhan. Umumnya, halaman website dibuat dengan Bahasa markup seperti HTML atau XHTML (Johnson dan Gupta, 2012). Proses ekstrasi data dari internet dapat dilakukan dengan berbagai Teknik dan metode. *Web Scraping* mengubah data yang tidak terstruktur, umumnya HTML, menjadi suatu format data terstruktur yang dapat disimpan dan divalidasi ke dalam media penyimpanan seperti basis data atau format file lain untuk keperluan analisis lebih lanjut. Inti dari proses web scraping adalah mengumpulkan data, meyimpan data, dan memvalidasi data (Dewi dkk, 2019). Validasi data dilakukan karena

kualitas dan keakuratan data perlu diperhatikan agar data dapat diinterpretasikan menjadi informasi yang bermanfaat.

Terdapat beberapa langkah dalam proses web scraping, yaitu (Josi dkk., 2014):

1. *Create Scraping Template*

   Pembuat program mempelajari dokumen HTML dari website yang akan diambil informasinya dari tag HTML yang mengapit informasi yang akan diambil. Contoh source program :

```
import requests
from bs4 import BeautifulSoup


# Step 1: Mendapatkan halaman web
url = 'https://example.com'
response = requests.get(url)
html_content = response.text


# Step 2: Membuat objek BeautifulSoup
soup = BeautifulSoup(html_content, 'html.parser')


# Step 3: Mempelajari struktur HTML untuk menemukan informasi yang diinginkan
# Misalnya, kita ingin mengambil semua judul artikel yang ada dalam tag <h2> dengan class 'title'
titles = soup.find_all('h2', class_='title')


# Menampilkan judul yang ditemukan
for title in titles:
    print(title.text)
```

2. *Explore Site Navigation*

   Pembuat program mempelajari teknis navigasi pada website yang akan diambil informasinya untuk ditirukan pada aplikasi web scraper. Contoh source program :

```
# Menemukan semua elemen dengan tag <h2> dan class 'article-title'
articles = soup.find_all('h2', class_='article-title')

# Menampilkan judul artikel untuk memverifikasi
for article in articles:
    print(article.text.strip())
```

3. *Automate*

   Navigation and Extraction Berdasarkan informasi yang didapatkan dari langkah 1 dan 2 diatas, aplikasi web scraper dibuat untuk mengotomatisasi pengambilan informasi dari website yang ditentukan. Contoh source program :

```
def scrape_articles(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    articles = soup.find_all('h2', class_='article-title')
    data = []
    for article in articles:
        data.append(article.text.strip())
    return data
# Contoh URL
url = 'https://example-news-site.com'
article_titles = scrape_articles(url)
for title in article_titles:
    print(title)
```

4. *Extracted Data and Package History*

   Informasi yang didapat dari langkah 3 disimpan dalam tabel atau tabel-tabel database. Contoh source program :

```
import csv
from datetime import datetime
# Nama file CSV untuk data
csv_filename = 'articles.csv'
def save_to_csv(data, filename):
    with open(filename, mode='w', newline='', encoding='utf-8')
    as file:
        writer = csv.writer(file)
        writer.writerow(['Title'])  # Menulis header kolom
        for title in data:
            writer.writerow([title])
# Menyimpan data
save_to_csv(article_titles, csv_filename)
print(f'Data telah disimpan di {csv_filename}')
```

2.2.3 Instagram

Instagram merupakan salah satu media sosial yang digemari pada zaman ini. Aplikasi smartphone ini ditujukan untuk berbagi foto ini, telah mencapai angka statistik yang tinggi disetiap harinya. Berdasarkan data pada tahun 2017, telah didapatkan informasi mengeneai pengguna aktif Instagram di Indonesia yang mencapai 45 juta orang. Pengguna aktif Instagram yang telah diakumulasikan pada tahun 2017, telah mencapai 700 juta pengguna aktif disetiap bulannya (Adi & Hidayat, 2017). Namun perkembangan pengguna instagram semakin meningkat seiring dengan fitur-fitur tambahan yang diberikan. Pada bulan September 2017 telah terhitung peningkatan pengguna sebesar 100juta, sehingga jika diakumulasikan terdapat 800 juta pengguna disetiap bulannya (Yusuf, 2017).

Instagram memberikan fasilitas para penggunanya untuk mengambil foto, memberikan filter secara digital pada foto, dan kemudian untuk dibagikan ke berbagai media sosial termasuk di akun Instagram tersebut. Seiring dengan perkembangan teknologi, Instagram turut berkembang. Tidak hanya dapat membagikan foto, kini Instagram telah menambahkan fasilitas pengguna untuk mengambil video. Fungsionalitas utama Instagram yang dapat anda rasakan ialah memiliki akun pribadi Instagram, dapat mengikut dan tidak mengikuti (follow

unfollow) akun Instagram lainnya, menyukai foto orang lain, menyimpan foto orang lain, dan memberi komentar.

2.2.4 Cyberbullying

Seiring dengan perkembangan zaman, teknologi pun ikut berkembang. Berkembangnya teknologi memberikan pengaruh terhadap kehidupan sosial. Seperti pada tindakan bullying. Mulanya tindakan bullying menyerang secara fisik maupun psikologi secara langsung, namun kini tindakan tersebut dapat dilakukan pada dunia maya yang dikenal dengan cyberbullying. Cyberbullying merupakan suatu tindakan tidak menyenangkan yang dilakukan secara sengaja dan terus menerus melalui teks elektronik (Stauffer, et al., 2012). Berdasarkan sumber lain mengatakan bahwa cyberbullying merupakan tindakan bullying yang dilakukan pada dunia cyber. Dimana dalam tindakannya dapat dibagi menjadi beberapa kriteria dan dilakukan secara berulang-ulang. Terdapat beberapa aspek yang memenuhi pada cyberbullying seperti, flaming, harrassment, cyberstalking, dan lainnya (Pratiwi, 2017). Cyberbullying memiliki jenis dengan masing-masing definisinya (Willard, 2007), seperti yang diketahui sebagai berikut:

1. Flaming, perkelahian dengan media pesan elektronik dengan bahasa yang menunjukkan amarah dan bersifat kasar.
2. Harassment, mengirim pesan jahat, kasar, dan menghina secara berulang kali.
3. Denigration, mengunggah gossip atau rumor mengenai seseorang dengan tujuan menjatuhkan reputasi orang tersebut.
4. Impersonation, berpura-pura menjadi orang lain dengan tujuan mengunggah mengenai orang (yang ditiru) sehingga membahayakan orang tersebut.
5. Outing, menyebarkan rahasia atau informasi mengenai seseorang, membagikan gambar secara online.
6. Trickery, mengungkapkan informasi seseorang atau mengenai hal yang memalukan dan dibagikan secara online.
7. Exclusion, tidak menganggap seseorang pada grup online dengan secara sengaja.
8. Cyberstalking, perilaku berulang yang dapat bersifat melecehkan, menghina hingga mengancam sehingga mengakibatkan ketakutan.

Cyberbullying merupakan bentuk serangan yang bersifat dengki untuk memberikan kepuasan atau kesenangan pelaku dengan cara menyiksa orang lain. Perempuan lebih banyak terlibat dalam cyberbullying, baik sebagai pelaku maupun sebagai korban. Dimana 50% korban cyberbullying umumnya tidak mengetahui identitas pelaku bully meski hanya gender. Cyberbullying merupakan tindakan yang mengakibatkan serangan psikologi, emosional, dan trauma sosial (Kowalski & Limber, 2007).

2.2.5 Text Mining

Text mining adalah cara pengekstraksian informasi yang tidak diketahui sebelumnya dari suatu sumber data yang berbeda ke dalam bentuk baru dimana bentuk baru dan sumber data tersebut mempunyai keterkaitan. Text Mining merupakan salah satu bentuk explorasi dan analisis data teks yang bertujuan untuk mendapatkan pengetahuan baru baik itu melalui cara otomatis maupun semi otomatis (Weiss, 2010). Berdasarkan kutipan-kutipan diatas, penulis berpendapat bahwa Text mining merupakan proses penggalian informasi-informasi baru dan berguna dari data lama menjadi data baru dimana kedua data tersebut saling berhubungan.

Tahap text preprocessing merupakan tahapan pertama dari text mining. Tahapan ini meliputi segala proses, aktivitas, serta metode untuk mempersiapkan data sebelum digunakan untuk penemuan pengetahuan sistem *text mining*. Preprocessing data tekstual memiliki beberapa tahap, yaitu *Cleaning*, *case folding*, *stopword*, *Tokenisasi*, *Normalizing* , dan *stemming* (Feldman dan Sanger, 2007).

1. *Cleaning*

   Tahapan *Cleaning* adalah proses membersihkan komentar dari kata yang tidak diperlukan untuk mengurangi noise. Kata yang dihilangkan adalah karakter HTML, katakunci, ikon emosi, hashtag (#), username (@username), url (http://situs.com), dan email (nama@situs.com). Contoh dari kata "Wah, keren banget!!! 😈 Lihat deh postingan @username ini! Jangan lupa cek juga http://situs.com. Pokoknya wajib lihat!! #amazing" menjadi "Wah keren banget Lihat deh postingan ini Jangan lupa cek juga Pokoknya wajib lihat".

2. *Case Folding*

   Tahapan Case-folding adalah proses penyamaan case dalam sebuah dokumen, ini dilakukan untuk mempermudah pencarian. Seperti mengubah semua kata menjadi 10 huruf kecil semua atau lower case. Tujuan Case-folding dilakukan untuk mempermudah pencarian, tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu peran case folding dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar (biasanya huruf kecil). Contoh dari kata "Wah keren banget Lihat deh postingan ini Jangan lupa cek juga Pokoknya wajib lihat" menjadi "wah keren banget lihat deh postingan ini jangan lupa cek juga pokoknya wajib lihat".

3. *Normalizing*

   Tahapan *normalizing* adalah proses mengubah bahasa pada kata yang tidak baku. Tahapan ini bertujuan untuk mengembalikan bentuk penulisan dari masing-masing kata yang sesuai dengan Kamus Besar Bahasa Indonesia (KBBI). Proses ini dilakukan dengan mencocokkan setiap kata pada dokumen data latih maupun data uji dengan kata yang ada pada kamus Bahasa tidak baku. Contoh dari kata ['wah', 'keren', 'banget', 'lihat', 'postingan', 'wajib', 'lihat'] menjadi ['wah', 'keren', 'sangat', 'lihat', 'postingan', 'wajib', 'lihat'].

4. *Stopword*

   Tahapan Stopword didefinisikan sebagai sebuah kata yang sangat sering muncul dalam suatu dokumen teks yang kurang memberikan arti penting terhadap isi dokumen. Stopword mempunyai tugas untuk menghilangkan kata yang tidak berpengaruh terhadap akurasi dalam klasifikasi sentimen. Berikut ini adalah contoh stopwords dalam bahasa Indonesia: yang, juga, dari, dia, kami, kamu, aku, saya, ini, itu, atau, dan, tersebut, pada, dengan, adalah, yaitu, ke. Contoh dari kata "wah keren banget lihat deh postingan ini jangan lupa cek juga pokoknya wajib lihat" menjadi "wah keren banget lihat postingan wajib lihat".

5. *Tokenizing*

   Tahapan Tokenisasi adalah proses untuk memisahkan deretan kata di dalam kalimat, paragraf atau halaman menjadi token atau potongan kata tunggal atau termmed word. Tokenisasi juga membuang beberapa karakter tertentu yang

dianggap sebagai tanda baca. Contoh dari kata "wah keren banget lihat postingan wajib lihat" menjadi ['wah', 'keren', 'banget', 'lihat', 'postingan', 'wajib', 'lihat'].

6. *Stemming*

   Tahapan Stemming adalah proses mengubah kata kedalam stem (kata dasar). Agar siap diolah ke tahapan selanjutnya. Proses stemming juga digunakan untuk meniadakan. Pada penelitian ini menggunakan algoritma sastrawi. Contoh dari kata ['wah', 'keren', 'sangat', 'lihat', 'postingan', 'wajib', 'lihat'] menjadi ['wah', 'keren', 'sangat', 'lihat', 'posting', 'wajib', 'lihat'].

2.2.6 Pembobotan Kata

Pembobotan kata merupakan tahapan yang digunakan untuk memberikan nilai setiap kata setelah tahap pre processing. Untuk melakukan pembobotan kata dilakukan menggunakan metode *Term Frequency-Inverse Document Frequency* (*TF-IDF*).

1. *Term Frequency*

   Term Frequency (*TF*) adalah metode yang digunakan saja menjelaskan jumlah kemunculan kata atau fitur dalam dokumen . Rumus Term Frequency dapat dilihat dalam persamaan (2.1) (Nandini dkk., 2019):

$$tf_{t,d} = \begin{cases} 1 + log_{10}\, tf_{t,d}\,, if tf_{t,d} > 0 \\ 0 \qquad\qquad\qquad otherwise \end{cases} \tag{2.1}$$

   Dengan :

   $tf_{t,d}$ = jumlah kata yang muncul di $term\ t$ pada dokumen d

2. *Document Frequency (DF)*

   Document Frequency (DF) adalah jumlah dokumen yang mengandung term t. Pada umumnya muncul di banyak dokumen.

3. *Inverse Document Frequency (IDF)*

   Inverse Document Frequency (IDF) mendefinisikan term yang paling sedikit muncul dalam dokumen merupakan term yang mengandung bobot paling tinggi. Rumus Inverse Document Frequency (IDF) dapat dilihat dalam persamaan (2.2) (Nandini dkk., 2019):

$$idf_t = log_{10}\left(\frac{N}{df_t}\right) \quad (2.2)$$

   Dengan:

   N = jumlah kata yang terdapat di dokumen teks

   df_t = jumlah dokumen yang terdapat term t

4. *Term Frequency-Inverse Document Frequency (TF-IDF)*

   Term Frequency - Inverse Document Frequency (TF-IDF) adalah sebuah algoritma Pembobotan terdiri dari dua algoritma dengan bobot yang berbeda, yaitu Term Frequency (TF) dan Inverse Document Frequency (IDF). keluaran dengan fitur yang sering muncul pada dokumen menghasilkan TF-IDF yang tinggi Fitur yang jarang muncul di dokumen menghasilkan skor rendah (Rahmattullah, 2022). Rumus *Term Frequency-Inverse Document Frequency (TF-IDF)* dapat dilihat dalam persamaan (2.3):

$$W_{t,d} = W_{tft,d}\ x\ idf_t \quad (2.3)$$

   Dengan:

   $W_{tft,d}$ = Nilai $term\ frequency$

   df_t = jumlah dokumen yang terdapat term

2.2.7 Naïve Bayes

Naïve Bayes merupakan algoritma probabilistik yang digunakan untuk memprediksi suatu keadaan (Rini dkk., 2016). Algoritma ini memanfaatkan teori peluang yang dikemukakan oleh ilmuwan Inggris, Thomas Bayes. Cara kerja

metode ini yaitu dengan memprediksi peluang terjadinya kejadian di masa depan berdasarkan data yang ada sebelumnya. Naïve Bayes adalah konsep probabilitas yang bisa digunakan untuk penentuan kelompok kelas dokumen teks dan dapat mengolah data dalam jumlah besar dengan hasil akurasi yang tinggi (Rossi dkk.,2017). Tingkat performa dari sistem klasifikasi yang dibuat menggunakan Naïve Bayes bergantung pada data yang dimiliki dan data yang dipilih sebagai data latih. Jika data yang dipilih sebagai data latih bisa mewakili semua atau sebagian besar data yang dimiliki, maka sistem klasifikasi yang dibuat mempunyai performa yang bagus. Begitu pun sebaliknya, ketika sistem klasifikasi yang dibuat mempunyai performa yang bagus, maka sistem tersebut bisa digunakan untuk melakukan klasifikasi terhadap data yang lebih banyak. Kelebihan dari metode Naïve Bayes yaitu relatif mudah untuk diimplemetasikan karena tidak menggunakan optimasi numerik, perhitungan matriks dan lainnya, efisien dalam pelatihan dan penggunaannya, bisa menggunakan data binary atau polinom, karena diasumsikan independen maka memungkinkan metode ini diimplementasikan dengan berbagai macam dataset, akurasi yang dihasilkan relatif tinggi. Sedangkan kekurangan dari metode Naïve Bayes yaitu perkiraan kemungkinan kelas yang tidak akurat dan batasan atau threshold harus ditentukan secara manual dan bukan secara analisis.

Persamaan algoritma Naive Bayes dalam analisis sentimen adalah sebagai berikut:

1. Teorema Bayes :

$$P(S_i|F) = \frac{P(F|S_i) \times P(S_i)}{P(F)} \qquad (2,4)$$

Dengan :

$P(S_i|F)$: Probabilitas posterior dari sentimen S_i (positif atau negatif) berdasarkan fitur F.

$P(F|S_i)$: Probabilitas likelihood dari fitur F dalam kategori sentiment S_i.

$P(S_i)$: Probabilitas prior dari sentimen S_i.

$P(F)$: Probabilitas prior dari fitur F.

2. Probabilitas Prior dari sentimen S_i ($P(S_i)$) :

$$P(S_i) = \frac{n_i}{N} \quad (2,5)$$

Dengan :

$P(S_i)$: Probabilitas prior dari sentimen S_i (positif atau negatif).

n_i : Jumlah sampel dengan sentimen S_i dalam dataset pelatihan.

N : Total jumlah sampel dalam dataset pelatihan.

3. Probabilitas Likehood ($P(F|S_i)$) :

$$P(F|S_i) = \frac{n(F, S_i)}{n(S_i)} \quad (2,6)$$

Dengan :

$P(F|S_i)$: Probabilitas likelihood dari fitur F yang muncul dalam sentimen S_i.

$n(F, S_i)$: Jumlah kemunculan fitur F di dalam sentimen S_i.

$n(S_i)$: Jumlah total semua fitur di dalam sentimen S_i.

4. Probabilitas Prior dari kemunculan F ($P(F)$) :

$$P(F) = \frac{|F|}{|S|} \quad (2,7)$$

Dengan :

$P(F)$: Probabilitas prior dari fitur F.

$|F|$: jumlah kemunculan dari fitur F

$|S|$: Total jumlah kemunculan sampel dalam dataset pelatihan

2.2.8 Decision Tree

Decision tree merupakan algoritma yang sering digunakan dalam banyak penelitian dikarenakan algoritma ini mudah diinterpretasikan berkat strukturnya yang sederhana (Mantas dan Abellan, 2014). Salah satu algoritma untuk membuat

pohon keputusan (decision tree) ialah algoritma Iterative Dichotomiser 3 (ID3). Masukan dari algoritma ID3 ini adalah sebuah basis data dengan beberapa variabel atau atribut (Nafi'iyah dan Fatichah, 2018). Proses klasifikasi dilakukan dari akar (root), lalu bercabang-cabang hingga diperoleh node daun (leaves). Leaves ini menunjukkan hasil akhir dari klasifikasi. Setiap objek yang diklasifikasi pada pohon keputusan harus diuji nilai entropinya. Entropi ini menunjukkan ukuran yang digunakan untuk mengetahui homogenitas dari kumpulan data (Sifaunajah dan Wahyuningtyas, 2022). Entropy digunakan untuk menilai seberapa penting sebuah node(Hikmatuloh dkk., 2021). Perhitungan nilai entropy dapat menggunakan persamaan 2,7. Sesudah memperoleh nilai entropi, pemilihan atribut dilakukan berdasarkan nilai information gain yang terbesar. Information gain merupakan tolok ukur pemilihan atribut untuk dijadikan sebagai akar maupun node pada pohon Keputusan (). Nilai Information gain dihitung dengan persamaan 2,8. Entropi (S) = 0, jika semua contoh dalam S berada di kelas yang sama Entropi (S) = 1, jika jumlah contoh positif sama dengan jumlah contoh negatif dalam S 0 < Entropi (S) < 1, jika jumlah contoh positif dan negatif dalam S berbeda. Kelebihan dari metode Decision Tree adalah sebagai berikut mudah dipahami dan diinterpretasikan secara visual, dapat mengatasi data yang tidak lengkap dan outliers, mampu mengidentifikasi fitur penting dalam data, tidak memerlukan normalisasi data, cocok untuk klasifikasi dan regresi. Sedangkan kekurangan dari metode Decision Tree adalah sebagai berikut cenderung rentan terhadap overfitting, terutama jika pohon terlalu kompleks, tidak efisien dalam mengatasi data dengan banyak fitur, kehilangan informasi yang relevan saat variabel input memiliki banyak kategori, rentan terhadap perubahan data yang kecil, tidak dapat menangani ketergantungan non-linear antara variabel.

$$Entropy(S) = \sum_{i=1}^{k} -p_j \, log_2 \, p_j \qquad (2,8)$$

Dengan :

S : ruang (data) sample yang digunakan untuk training

k : banyaknya partisi pada S

p_j : probabilitas yang didapat dari Sum(Ya) dibagi dengan total sample

$$Gain(A) = Entropy(S) - \Sigma_{i=1}^{k} \frac{|si|}{|s|} Entropy(S_i) \qquad (2,9)$$

Dengan:

S : ruang (data) sample yang digunakan untuk training

A : atribut

$|si|$: jumlah sample untuk nilai V

$|s|$: jumlah seluruh sample data

$Entropy(S_i)$: entropy untuk sample – sample yang memiliki nilai i.

2.2.9 K-Nearest Neighbor (K-NN)

Dalam buku Algoritma Data Mining, Kusrini menerangkan bahwa Algoritma K-Nearest Neighbors adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada (Kusrini, 2009). Pada Pandangan yang lain disebutkan bahwa K-Nearest Neighbors adalah sebuah algoritma untuk melakukan klasifikasi pada objek berdasarkan data yang jaraknya paling dekat dengan objek tersebut. Data digambarkan ke dalam ruang berdimensi banyak, dimana masing-masing dimensi mencerminkan fitur dari data. Nilai k yang terbaik untuk algoritma ini bergantung pada data yang secara umum nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur (Romadhoni dkk., 2019). Tujuan utama dari algoritma ini yaitu mengklasifikasikan suatu obyek berdasarkan atribut-atribut dan training sample. Algoritma K-Nearest Neighbor (K-NN) menggunakan klasifikasi kedekatan titik sebagai nilai perkiraan dari query instance yang baru. Kelebihan dari metode K-Nearest Neighbor (K-NN) adalah sebagai berikut sangat sederhana dan mudah untuk diimplementasikan, beberapa parameter untuk acuan yakni jarak matrik dan k, efektif dalam menghitung data berskala besar maupun kecil, kuat untuk melatih data noisy. Sedangkan kekurangan dari metode K-Nearest Neighbor (K-NN) adalah sebagai berikut perlu untuk menentukan nilai k untuk menyatakan tetangga terdekatnya. perhitungan jarak harus dilakukan pada setiap querry instance sehingga membuat

biaya komputasi yang tinggi, rentan terhadap variable noninformative. Berikut persamaan algoritma K-Nearest Neighbor (K-NN) sebagai berikut.

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \qquad (2,10)$$

Dengan :

$d(x_i, x_j)$: Jarak antara dua data x_i dan x_j.

x_{ik} dan x_{jk} : Nilai fitur ke -k dari data x_i dan x_j.

n : Jumlah fitur.

Berikut adalah perbandingan kelebihan dan kekurangan dari Naïve Bayes, Decision Tree, dan K-Nearest Neighbor (K-NN) tersebut yang digunakan dalam analisis sentimen yang dapat dilihat pada table 2.1 :

Tabel 2. 1 Kelebihan dan kekurangan Naïve Bayes, Decision Tree, dan K-NN.

Algoritma	Kelebihan	Kekurangan
Naïve Bayes	• Relatif mudah untuk diimplemetasikan karena tidak menggunakan optimasi numerik, • Perhitungan matriks dan lainnya, efisien dalam pelatihan dan penggunaannya, • Bisa menggunakan data binary atau polinom, • Karena diasumsikan independen maka memungkinkan metode ini	• Perkiraan kemungkinan kelas yang tidak akurat • Batasan atau threshold harus ditentukan secara manual dan bukan secara analisis

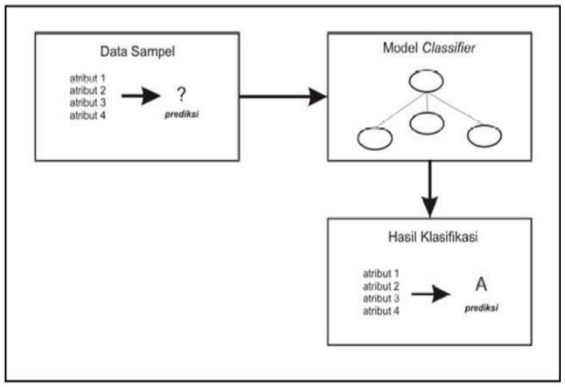
Algoritma	Kelebihan	Kekurangan
	diimplementasikan dengan berbagai macam dataset, • Akurasi yang dihasilkan relatif tinggi.	
Decision Tree	• Mudah dipahami dan diinterpretasikan secara visual. • Dapat mengatasi data yang tidak lengkap dan outliers. • Mampu mengidentifikasi fitur penting dalam data, tidak memerlukan normalisasi data. • Cocok untuk klasifikasi dan regresi.	• Cenderung rentan terhadap overfitting terutama jika pohon terlalu kompleks, • Tidak efisien dalam mengatasi data dengan banyak fitur, • Kehilangan informasi yang relevan saat variabel input memili ki banyak kategori, • Rentan terhadap perubahan data yang kecil, • Tidak dapat menangani ketergantungan non-linear antara variabel.
K-Nearest Neighbor (K-NN)	• Sangat sederhana dan mudah untuk diimplementasikan, • Beberapa parameter untuk acuan yakni jarak matrik dan k • Efektif dalam menghitung data berskala besar maupun	• Perlu untuk menentukan nilai k untuk menyatakan tetangga terdekatnya. • Perhitungan jarak harus dilakukan pada setiap querry instance

Algoritma	Kelebihan	Kekurangan
	kecil, kuat untuk melatih data noisy	sehingga membuat biaya komputasi yang tinggi, • Rentan terhadap variable noninformative

2.2.10 Klasifikasi

Klasifikasi adalah proses pengelompokan teramati darisuatu objek data ke dalam suatu kelas tertentu berdasarkan kelas-kelas yang ada.Sedangkan menurut (Liu, 2012), klasifikasi didefinisikan sebagai teknik analisis data yang digunakan untuk menghasilkan model prediksi untuk mendeskripsikan label atau kelas data.

Menurut Han dan Kamber (2006) terdapat dua tahap dalam proses klasifikasi. Tahap pertama merupakan learned model, pada tahap ini dilakukan pembangunan model menggunakan hasil analisa record database dari serangkaian kelas yang ada. Masing-masing record diasumsikan mempunyai predefined class yang didasarkan pada atribut kelas label, oleh karena itu klasifikasi termasuk dalam supervised learning. Tahap ini juga disebut sebagai tahap pembelajaran atau pelatihan. Sebuah algoritma klasifikasi akan membangun sebuah model klasifikasi dengan cara menganalisis data training yang bisa dilihat di gambar 2.2.



Gambar 2.1 Proses Klasifikasi.
(Han dan Kamber, 2006)

Tahap pembelajaran dapat juga dipandang sebagai tahap pembentukan fungsi atau pemetaan y = f(x), dimana y merupakan kelas hasil prediksi dan x adalah record yang ingin dipediksi kelasnya. Kualitas hasil kualifikasi dapat dinilai dan dievaluasi berdasarkan beberapa ukuran (Pratama dkk., 2023). Evaluasi biasanya di tampilkan dalam tabel confusion matrix seperti tabel 2.1 yang dimana confusion matrix yaitu sebuah tabel yang digunakan untuk mengevaluasi performa suatu model klasifikasi atau algoritma prediksi:

Tabel 2. 2 Confusion Matrix (Pratama dkk., 2023)

	Actual Classification		
	Label	*Positive*	*Negative*
Prediction Classification	*Positive*	*True Positive (TP)*	*False Positive (FP)*
	Negative	*False Negative (FN)*	*True Negative (TN)*

1. *Accuracy*

   Accuracy adalah jumlah proporsi prediksi yang benar. Akurasi digunakan sebagai tingkat ketepatan antara nilai aktual dengan nilai prediksi (Nandini dkk.,2019). Rumus Accuracy dapat dilihat pada persamaan (2.10):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \qquad (2.11)$$

   dengan

   $TP = True\ Positif$

   $TN = True\ Negatif$

2. *Precision*

   Precision adalah proporsi jumlah dokumen teks yang relevan terkendali diantara semua dokumen yang dipilih sistem. Precision digunakan sebagai tingkat ketepatan informasi yang diminta dengan jawaban yang diberikan

oleh sistem (Nandini dkk., 2019). Rumus Precision dapat dilihat pada persamaan (2.11):

$$Precision = \frac{TP}{\sum TP + FP} \times 100\% \quad (2.12)$$

3. *Recall*

   Recall adalah proporsi jumlah dokumen teks yang relevan terkendali diantara semua dokumen teks relevan yang ada pada koleksi. Recall digunakan sebagai ukuran keberhasilan sistem dalam menemukan kembali informasi (Nandini dkk., 2019). Rumus Recall dapat dilihat pada persamaan (2.12):

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2.13)$$

4. *F1 Score*

   F1 Score adalah merupakan kata-kata harmonis dari nilai recall dan nilai precision sehingga dapat memberikan penilaian kinerja yang lebih seimbang. F1 score digunakan untuk mengukur kinerja sistem secara menyeluruh dalam pengklasifikasian (Nandini dkk., 2019). Rumus F1 score dapat dilihat dengan persamaan (2.13);

$$F1\ Score = \frac{2 \times Precision \times Re\ call}{Precision + Recall} x100\% \quad (2.14)$$