BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang berhungan dengan analisis sentimen dilakukan oleh Meifitrah R, Darmawan I, & Nurul P.O (2020) menjelaskan dalam melakukan analisis sentimen dibutuhkan pemilihan metode dan algoritma yang tepat. Banyak metode yang sering digunakan dalam analisis sentimen diantaranya Naive Bayes Classifier, Support Vector Machine, Decision Tree, namun dijelaskan pula oleh peneliti bahwa sebelumnya telah dilakukan penelitian oleh peneliti lain terhadap tingkat akurasi setiap metode yang menghasilkan data yaitu Naive Bayes 63%, KNN 60%, dan SVM 61%. Keunggulan metode Naive Bayes sederhana dan intuitif dapat menghasilkan kinerja yang baik dengan sedikit data pengujian.

Penelitian lainnya juga dilakukan oleh Widiastuti Dwi, Rasal I & Wulandari A.P.D (2022) yang membandingkan proses klasifikasi, yaitu model SVM, Random Forest, dan Naïve Bayes terhadap polaritas kategorisasi sentimen review produk di Tokopedia. Data yang digunakan dalam penelitiannya adalah review produk online yang dikumpulkan dari aplikasi Tokopedia. Klasifikasi dilakukan pada kategorisasi level kalimat dan kategorisasi level rating bintang. Dalam penelitian yang dilakukan, kategorisasi polaritas sentimen menjadi dasar permasalahan analisis sentimen dalam review produk.

Melina Salsabila S, Alim M.A & Fadhilah N (2022) melakukan penelitian untuk memberikan solusi dalam menentukan kepuasan pelanggan dan umpan balik pada aplikasi berdasarkan umpan balik pelanggan terhadap Tokopedia, di mana umpan balik tersebut akan digunakan sebagai panduan untuk meningkatkan kualitas layanan, kualitas produk, dan operasi bisnis internal. Studi ini memiliki keuntungan memberi keterangan kepada manajemen Tokopedia sehingga mereka selalu konsisten dalam memberikan layanan kepada pelanggan agar pelanggan percaya dan pada akhirnya dapat memberikan keuntungan yang sejalan dengan target yang telah dicapai.

Penelitian yang dilakukan oleh Mutmainah & Yunita (2020) dalam penerapan metode TOPSIS pada pemilihan kendaraan ekspres memiliki beberapa faktor yang mempengaruhi pemilihan kendaraan ekspres, seperti harga, waktu pengiriman, dan waktu pengiriman. Menemukan metode terbaik diperlukan untuk pengambilan keputusan dan dapat membantu dengan masalah yang ada. Ini dapat dilakukan dengan menggunakan metode TOPSIS bersamaan dengan sistem pendukung keputusan. Metode TOPSIS memiliki konsep yang mudah dipahami dan kemampuan untuk mengurangi produksi kerja relatif dari alternatif keputusan dalam format matematis yang sederhana.

Penelitian yang dilakukan oleh Wizsa, Yuspita & Rahayu (2022) membendingkan metode SAW dan metode TOPSIS sebagai sistem pengambilan keputusan penerima beasiswa KIP IAIN. Selama ini proses seleksi masih dilakukan dengan menghitung setiap kriteria pemilihan diberi persentase berdasarkan kepentingan. Namun kekurangan dari prosedur tersebut tidak dapat mencakup sistem pengambilan keputusan. Maka di perlukan sistem pendukung keputusan yang cepat dan akurat untuk menentukan penerima beasiswa yang tepat.

Penelitian oleh Salwa (2021) menyimpulkan dengan data sebanyak 1.134 ulasan, menunjukkan tingkat akurasi tertinggi yang dimiliki oleh skenario data latih 70% dan data uji 30% sebesar 93,45%. Hasil prediksi untuk ulasan positif dan negatif masing-masing adalah 145 dan 653 ulasan. Informasi yang didapatkan dari proses klasifikasi dalam kelas sentimen positif yang berupa pujian terhadap pelayanan dan kelas sentimen negatif yang berisi rasa kekecewaan serta keluhan terdahap aplikasi yang telah digunakan.

2.2 Dasar Teori

2.2.1. Video Streaming

Sebuah teknologi yang disebut streaming memungkinkan untuk mendengarkan file audio atau video dengan tenang, atau bahkan dengan perekam sebelumnya, dari server web. Dengan kata lain, file video atau audio yang

disimpan di server dapat diluncurkan secara diam-diam di komputer pengguna kapan saja setelah permintaan pengguna dibuat. Ini berarti bahwa proses unduhan aplikasi dapat dihentikan kapan saja tanpa perlu menyelesaikan proses penyimpanan. Streaming video adalah fitur yang sering digunakan saat menonton video online melalui browser; tidak perlu menonton video tersebut untuk melihatnya. Contoh aplikasi streaming video adalah Vidio.

Salah satu platform yang dapat digunakan untuk streaming online adalah Vidio. Layanan aplikasi video tersedia di situs web www.Vidio.com. Selain itu, aplikasi Vidio dapat di unduh melalui perangkat seluler untuk pengguna Android atau iOS. Vidio dulunya dimiliki oleh Kreatif Media Karya, tetapi saat ini dimiliki oleh PT Surya Citra Media Tbk; keduanya adalah anak perusahaan Emtek. Aplikasi Vidio dibuat oleh Adi Sariaatmadja pada 15 Oktober 2014. Layanan ini mencakup televisi siaran gratis, streaming langsung, drama dan film, serta televisi. (id.wikipedia.org).

2.2.2. Machine Learning

Machine Learning merupakan aplikasi atau komponen dari kecerdasan buatan yang menciptakan sistem dengan kemampuan pembelajaran otomatis dan meningkatkan kemampuannya berdasarkan pengalaman tanpa pemrograman eksplisit. *Machine Learning* berfokus pada pengembangan sistem yang dapat belajar sendiri untuk mengubah sesuatu tanpa perlu diprogram oleh manusia. *Machine Learning* berfungsi ketika data disediakan sebagai input untuk analisis sejumlah besar data guna mengidentifikasi pola tertentu. Data adalah bahan masukan yang akan digunakan dalam pelatihan atau pendidikan agar mesin dapat menghasilkan analisis yang mendalam. Dua jenis data digunakan dalam *machine learning*: data pengujian dan data pelatihan. Pengujian data bertujuan untuk memahami kinerja algoritma yang digunakan dalam pembelajaran mesin yang telah dilatih, yaitu ketika data baru disediakan yang belum pernah digunakan dalam pelatihan data (Fikriya, Soetrisno, & Irawan, 2017).

Salah satu aspek kecerdasan buatan adalah *machine learning*, yang bertujuan untuk memahami atau menganalisis struktur dari sekumpulan data yang diberikan dan mengubahnya menjadi sebuah model. Ini berbeda dari

pemrograman tradisional, seperti yang dijelaskan dalam program kode *machine learning*. Kemampuan belajar ini adalah hasil dari terus-menerus mengakses fakta dan informasi. Implementasi *machine learning* dalam memecahkan masalah memerlukan beberapa asumsi, dan jika asumsi-asumsi tersebut tidak terpenuhi, hasil atau solusinya akan kurang akurat. Pertama, proses *machine learning* memerlukan sejumlah besar memori atau penyimpanan untuk memproses data, yang sering kali juga besar. Kedua adalah proses pelabelan, khususnya untuk pembelajaran terawasi. Proses pelabelan selalu memerlukan banyak waktu. ketiga, hasil atau solusi yang muncul tidak dapat diandalkan karena adanya bias dalam data (Kusuma, 2020).

Machine learning menggunakan teknik untuk menganalisis kumpulan data besar (big data) dengan cara yang tepat untuk mendapatkan hasil yang akurat. Bergantung pada metodologi pengajaran, jenis-jenis pembelajaran mesin dapat diklasifikasikan sebagai pembelajaran terawasi, pembelajaran tidak terawasi, pembelajaran semi-terawasi, dan pembelajaran penguatan. Pembelajaran terawasi adalah salah satu jenis teknik pembelajaran mesin yang menggunakan dataset (data pelatihan) yang telah diberi label (diberi label) untuk mengajarkan mesin. Akibatnya, mesin dapat mengidentifikasi input label dengan menggunakan fitur yang tersedia untuk melakukan prediksi atau klasifikasi lebih lanjut, sementara pembelajaran tanpa pengawasan adalah teknik yang menggunakan data yang didasarkan pada input dari data respons yang dilabeli (Iswara, 2019). Algoritma yang termasuk kedalam teknik supervised learning diantaranya Decision Tree, K-Nearest Neighbor (KNN), Naïve Bayes, Regresi, dan Super Vector Machine.

2.2.3. Analisis Sentimen

Salah satu subbidang dari *Text Mining* adalah analisis sentimen, yang kadang-kadang disebut sebagai *opinion mining*. Berdasarkan data, analisis sentimen dapat dibagi menjadi beberapa level. Dua tingkat yang sering digunakan dalam penelitian adalah analisis sentimen pada tingkat dokumen dan analisis sentimen pada tingkat kalimat. Dua kelompok terbesar, Analisis Sentimen Berbutir Kasar dan Analisis Sentimen Berbutir Halus, dipisahkan berdasarkan ringkasan data (Meifitrah R, Darmawan I, & Nurul P.O, 2020).

Analisis sentimen biasanya dilakukan dalam penelitian yang melibatkan analisis; salah satu analisis tersebut adalah tentang opini dan persepsi banyak orang mengenai entitas yang dimaksud sebagai contoh topik atau layanan. Tujuan analisis sentimen adalah untuk mengidentifikasi ide utama atau polaritas konteks dari sebuah dokumen yang diberikan (Savitri, 2021). Tujuan utama dari analisis sentimen adalah untuk merangkum teks yang ada dalam sebuah dokumen atau kalimat dan kemudian menentukan apakah informasi yang disajikan dalam dokumen atau kalimat tersebut bersifat positif atau negatif.

2.2.4. Preprocessing

Data yang telah diambil dari aplikasi vidio melalui scraping data menggunakan google colab masih merupakan data mentah yang masih perlu olah dengan melakukan *preprocessing* untuk mendapatkan data yang siap untuk diproses lebih lanjut. Tahap-tahap dari proses *preprocessing* adalah sebagai berikut :

a. Case Folding

Case folding adalah sebuah proses dimana semua huruf diubah menjadi huruf kecil. Contoh hasil mengubah huruf menjadi huruf kecil dapat dilihat pada tabel 2.1 dibawah ini.

Tabel 2. 1 *Case Folding*

Sebelum	Setelah
Kebanyakan iklan dan sangat terganggu lebih baik uninstal aja	kebanyakan iklan dan sangat terganggu lebih baik uninstal aja

b. Tokenizing

Adalah tahapan untuk memisahkan string input berdasarkan kata-kata yang menyusunnya dengan menggunakan spasi. Berikut contoh hasil proses *tokenizing* dapat dilihat pada tabel 2.2.

Tabel 2. 2 *Tokenizing*

Sebelum	Setelah
kebanyakan iklan dan sangat terganggu lebih	kebanyakan, iklan, dan, sangat, terganggu, lebih,

baik uninstal aja	baik, uninstal, aja

c. Stopword Removal

Adalah proses menghilangkan kata-kata yang tidak perlu. Tabel 2.3 memperlihatkan contoh hasil dari proses *stopword removal*.

Tabel 2. 3 *Stopword Removal*

Sebelum	Setelah
kebanyakan, iklan, dan, sangat, terganggu, lebih, baik, uninstal, aja	kebanyakan, iklan, terganggu, uninstal.

d. Normalization

Adalah menyeragamkan term yang memiliki makna sama namun penulisanya berbeda. Berikut contoh hasil proses *normalization* dapat dilihat pada tabel 2.4.

Tabel 2. 4 *Normalization*

Sebelum	Setelah
kebanyakan, iklan, terganggu, uninstal.	kebanyakan, iklan, terganggu, uninstal.

e. Stemming

Tahap *stemming* dilakukan untuk mengembalikan kata ke bentuk kata dasar. Proses ini dilakukan dengan menghilangkan awalan, akhiran, atau keduanya. Berikut contoh hasil proses *stemming* dapat dilihat pada tabel 2.5.

Tabel 2. 5 *Stemming*

Sebelum	Setelah
Kebanyakan, iklan, terganggu, uninstal.	banyak, iklan, ganggu, uninstal.

2.2.5. TF-IDF

Metode TF-IDF merupakan metode untuk menghitung bobot setiap kata yang paling umum digunakan dalam pencarian informasi. Metode TF-IDF merupakan hasil dari penggabungan dua konsep, yaitu *Term Frequence* (TF) dan *Inverse Document Frequency* (IDF). TF menyatakan banyaknya kemunculan kata

dalam dokumen maupun kalimat. IDF adalah suatu kata yang sering muncul dalam kalimat memiliki bobot kecil (Tanggraeni & Sitokdana, 2022).

$$W_{i,j} = tf_{i,j} \times \log(\frac{N}{df_i})$$

dimana :

N = total dokumen

$tf_{i,j}$ = banyaknya kata-i pada dokumen ke-j

df_i = banyaknya dokumen yang mengandung kata ke-i

TF menunjukkan seberapa sering sebuah kata muncul dalam sebuah dokumen. Semakin sering sebuah kata muncul, semakin penting kata tersebut dianggap dalam dokumen itu. IDF menunjukkan seberapa unik sebuah kata di seluruh kumpulan dokumen. Semakin jarang sebuah kata muncul di seluruh dokumen, semakin penting kata tersebut dianggap.

Contoh Kasus:

Beberapa ulasan pengguna tentang aplikasi Vidio:

- Ulasan 1: "Kualitas video Vidio sangat bagus, tapi sering buffering."
- Ulasan 2: "Antarmuka Vidio mudah digunakan, banyak pilihan filmnya."
- Ulasan 3: "Aplikasi Vidio sering crash, bikin kesel."

Langkah-langkah Perhitungan:

- Tokenisasi:
  - Ulasan 1: ["kualitas", "video", "vidio", "sangat", "bagus", "tapi", "sering", "buffering"]
  - Ulasan 2: ["antarmuka", "vidio", "mudah", "digunakan", "banyak", "pilihan", "filmnya"]
  - Ulasan 3: ["aplikasi", "vidio", "sering", "crash", "bikin", "kesel"]
- Hitung TF (Term Frequency):
  - Ulasan 1: TF("video") = 1/8, TF("buffering") = 1/8, dll.
  - Ulasan 2: TF("mudah") = 1/7, TF("filmnya") = 1/7, dll.
  - Ulasan 3: TF("crash") = 1/6, dll.
- Hitung IDF (Inverse Document Frequency):

- IDF("vidio") = log(3/3) = 0 (karena kata "vidio" muncul di semua dokumen)
- IDF("buffering") = log(3/1) = log(3) (karena kata "buffering" hanya muncul di 1 dokumen)
- ... (hitung untuk kata lainnya)

- Hitung TF-IDF:

  - Ulasan 1: TF-IDF("buffering") = (1/8) * log(3)
  - Ulasan 2: TF-IDF("mudah") = (1/7) * log(3)
  - ... (hitung untuk kata lainnya)

Interpretasi Hasil:

- Kata dengan nilai TF-IDF tinggi menunjukkan kata tersebut penting dan unik dalam dokumen tersebut. Misalnya, "buffering" dan "crash" memiliki nilai TF-IDF tinggi karena sangat spesifik dan terkait dengan masalah pada aplikasi Vidio.
- Kata dengan nilai TF-IDF rendah mungkin terlalu umum atau tidak memberikan informasi yang signifikan untuk analisis sentimen.

Analisis Lebih Lanjut:

Kata Kunci Negatif: Kata-kata seperti "buffering", "crash", "lambat", "kesulitan" seringkali terkait dengan sentimen negatif.

Kata Kunci Positif: Kata-kata seperti "mudah", "bagus", "suka", "menyenangkan" seringkali terkait dengan sentimen positif.

Fitur Aplikasi: Kita bisa menganalisis sentimen terhadap fitur-fitur spesifik seperti kualitas video, antarmuka pengguna, pilihan konten, dan lain-lain.

Contoh Analisis:

Jika banyak ulasan mengandung kata "buffering" dengan nilai TF-IDF tinggi, maka dapat disimpulkan bahwa pengguna banyak mengeluhkan masalah buffering pada aplikasi Vidio.

2.2.6. Klasifikasi Naïve Bayes

Metode ini didasarkan pada teorema bayes yang ditemukan oleh Thomas Bayes. *Naïve Bayes Classifier* dapat digunakan untuk menunjukkan bahwa atribut kelas tertentu tidak dapat mempengaruhi atau mengurangi atribut lainnya. Penjelasan ini, yang dikenal sebagai *Naïve Bayes Classifier*, mengasumsikan bahwa sebuah frasa muncul dalam kalimat tertentu yang tidak dapat mempengaruhi kata lain. (Sari & Wibowo, 2019). Dalam analisis sentimen, setiap frasa atau kata yang muncul akan memiliki bobot, dan jika seluruh bobot diperhitungkan, akan mungkin untuk menentukan apakah kalimat yang dimaksud terkait dengan sentimen positif atau negatif.

Teorema Bayes menyempurnakan teorema probabilitas bersyarat yang hanya dibatasi oleh 2 buah kejadian sehingga dapat diperluas untuk n buah kejadian. Umumnya E_1,E_2, … , E_n adalah kejadian-kejadian yang mempunyai ruang sampel s dengan kejadian A sebagai himpunan bagian dari S atau $A \subset S$ maka dengan menggunakan probabilitas bersyarat dapat dikatakan :

$$P(E_i|A) = \frac{P(E_i \cap A)}{P(A)} \qquad (2.1)$$

Diketahui,

$$P(E_i \cap A) = P(A \cap E_i) = P(E_i)P(A|E_i) \qquad (2.2)$$

Dengan hukum probabilitas total :

$$P(A) = \sum_{j=1}^{n} P(A \cap E_j) \qquad (2.3)$$

Jadi, dengan mensubtitusi $P(Ei \cap A)$ dan $P(A)$ diperoleh

$$P(E_i|A) = \frac{P(E_i)P(A|E_i)}{\sum_{j=1}^{n} P(E_j)P(A|E_j)} \qquad dimana \qquad i = 1,2, \dots, n \qquad (2.4)$$

Disini, $P(Ei|A)$ adalah probabilitas bersyarat yang disebut juga probabilitas posterior. Dengan direduksinya penerapan konsep menjadi dua peristiwa, yaitu A dan B, probabilitas bersyarat untuk A dengan B yang diberikan adalah $P(A|B)$.

Karena probabilitas bersyarat $P(A|B)$ dan $P(B|A)$ tidak sama, Teorema Bayes dapat digunakan untuk membangun hubungan antara probabilitas bersyarat yang berbeda

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{2.5}$$

Keterangan :

A : Hipotesis data B merupakan suatu class spesifik

B : Data dengan class yang belum diketahui

P(A|B) : Probabilitas hipotesis A berdasar kondisi B (posterior probability)

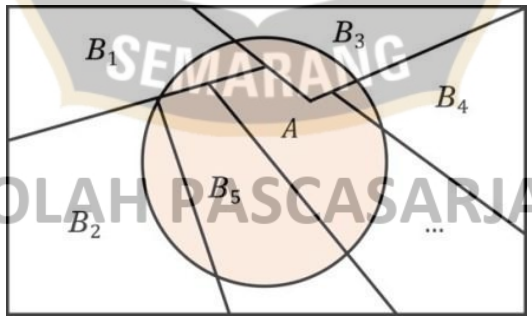
P(A) : Probabilitas hipotesis A (prior probability)

P(B|A) : Probabilitas B berdasarkan kondisi pada hipotesis A

P(B) : Probabilitas B

Dimana,

$P(A)$ adalah probabilitas prior atau marginal dari A, $P(B|A)$ adalah B jika diberi probabilitas bersyarat. $P(A|B)$ adalah probabilitas bersyarat B tertentu, dan $P(B)$ adalah probabilitas prior atau marginal dari B



Gambar 2. 1 Teorema Peluang Total

Sumber : Ahzuri, Zulfah, Ediputra K, 2024

Jika kejadian B_1, B_2,B_3, … , B_k membentuk suatu partisi dari ruang sampel S sehinnga $P(B_i) \neq 0$ untuk $i \in \{1,2,3, \dots , k\}$, maka untuk setiap kejadian A di S berlaku.

$$P(A) = \sum_{i=1}^{k} P(B_i \cap A) = \sum_{i}^{k} P(B_i) \cdot P(A|B_i) \quad (2.6)$$

Probabilitas bersyarat suatu peristiwa A, dengan syarat peristiwa B didefinisikan sebagai :

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (2.7)$$

atau

$$P(B|A) = \frac{P(B \cap A)}{P(A)} \quad (2.8)$$

Dimana

$$P(A \cap B) = P(B \cap A) \quad (2.9)$$

Sehingga dari persamaan (2.8) dan (2.9) didapatkan

$$P(A \cap B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$

$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)} \quad (2.10)$$

Dengan menggunakan teorema peluang total (2.6), aturan bayes kemudian dapat dinyatakan dengan berlandaskan pada definisi peluang bersyarat (2.10), yaitu kasus ketika kita ingin mencari nilai dari $P(B_i|A)$

$$P(B_i|A) = \frac{P(B_i \cap A)}{P(A)} = \frac{P(B_i) \cdot P(A|B_i)}{\sum_{i}^{k} P(B_i) \cdot P(A|B_i)} \quad (2.11)$$

$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)} \quad (2.12)$$

Keterangan :

$P(B|A)$: Peluang Posterior

$P(A)$: Peluang Prior

$P(A|B)$: Peluang Likelihood

$P(B)$: Peluang Evidence

Contoh kasus :

Ulasan	**Sentimen**

Film ini sangat bagus, aktingnya keren	Positif
Ceritanya membosankan, efek visualnya jelek	Negatif
Aktor utama sangat berbakat, layak ditonton	Positif

Kata Kunci:

- **Positif:** bagus, keren, berbakat, layak
- **Negatif:** membosankan, jelek

Langkah-langkah Perhitungan:

Hitung Probabilitas Prior:

P(Sentimen = Positif) = 2/3

P(Sentimen = Negatif) = 1/3

Hitung Probabilitas Bersyarat:

Untuk kelas Positif:

P(bagus | Positif) = 1/2

P(keren | Positif) = 1/2

P(berbakat | Positif) = 1/2

P(layak | Positif) = 1/2

P(membosankan | Positif) = 0

P(jelek | Positif) = 0

Untuk kelas Negatif:

P(bagus | Negatif) = 0

P(keren | Negatif) = 0

P(berbakat | Negatif) = 0

P(layak | Negatif) = 0

P(membosankan | Negatif) = 1

P(jelek | Negatif) = 1

Klasifikasi Ulasan Baru:

Misalkan kita ingin mengklasifikasikan ulasan "Film ini sangat bagus dan aktornya berbakat".

Hitung Probabilitas Posterior:

P(Positif | bagus, berbakat) = P(bagus | Positif) * P(berbakat | Positif) * P(Positif) / P(bagus, berbakat)

Hitung juga P(Negatif | membosankan, jelek) dengan cara yang sama.

Klasifikasi:

Bandingkan kedua probabilitas posterior di atas. Pilih kelas dengan probabilitas tertinggi sebagai prediksi.

Penjelasan Singkat:

Probabilitas Prior: Peluang sebuah ulasan bersifat positif atau negatif secara umum.

Probabilitas Bersyarat: Peluang suatu kata kunci muncul dalam ulasan yang berlabel positif atau negatif.

Probabilitas Posterior: Peluang sebuah ulasan berlabel positif atau negatif diberikan kata-kata kunci tertentu.

2.2.7. Pengujian Klasifikasi

Proses klasifikasi memerlukan proses pengujian yang terkait dengan hasil klasifikasi. Ini harus dilakukan untuk menilai akurasi perhitungan yang sedang berlangsung. Proses klasifikasi menggunakan matriks kebingungan, yang dijelaskan seperti yang ditunjukkan pada tabel berikut. Eko Prasetyo (2012).

Tabel 2. 6 Matriks confusion

Keterangan	Relevan	Tidak Relevan
Terambil	True Positif (TP)	False Positif (FP)
Tidak Terambil	False Negatif (FN)	True Negatif (TN)

Dimana,

TP = terindentifikasi secara benar

FP = teridentifikasi secara salah

FN = tertolak secara benar

TN = tertolak secara salah

Rumus untuk menghitung akurasi seperti pada rumus 2.9 sebagai berikut.

$$Akurasi = \frac{Jumlahdatayangdiprediksibenar}{Jumlahprediksiyangdilakukan} \quad (2.9)$$

$$Accurasi : \frac{TP+TN}{TP+FP+FN+TN} \quad (2.10)$$

$$Recall : \frac{TP}{TP+FN} \quad (2.11)$$

$$Precision : \frac{TP+TN}{TP+FP} \quad (2.12)$$

$$F - Scrore : \frac{2 \times (Precision \times Recall)}{Precision+Recall} \quad (2.12)$$

Ada algoritma klasifikasi yang dirancang untuk membuat model dengan akurasi tinggi. Secara umum, model yang dibangun dapat diprediksi dengan akurat dari semua data yang digunakan sebagai data latih. Namun, ketika model dibandingkan dengan data uji, barulah kinerja model dari algoritma klasifikasi dapat ditentukan.

2.2.8. Metode *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)

TOPSIS diklasifikasikan sebagai metode Multi- Criteria Decision Making, dimana teknik pengambilan keputusannya berdasarkan beberapa pilihan alternatif yang ada. di mana proses pengambilan keputusan didasarkan pada beberapa alternatif yang tersedia. TOPSIS dapat mengidentifikasi alternatif terbaik yang memiliki nilai jarak terpendek dari solusi positif ideal dan nilai jarak terpanjang dari solusi negatif ideal. (Giofani & dkk, 2022). Berikut adalah langkah-langkah dalam metode TOPSIS, yaitu sebagai berikut:

1. Membuat matriks perbandingan berpasangan yang ternormalisasi

$$r_{ij} \frac{Xij}{\sqrt{\sum_{i=1}^{m} X_{ij}^2}} \quad (2.13)$$

Dengan i = 1, 2, … m dan j = 1, 2, … n

Dimana

rij = Elemen matriks ternormalisasi

Xij = Elemen matriks keputusan X

2. Membuat matriks keputusan yang ternormalisasi terbobot

$$yij = wirij \quad (2.14)$$

Dengan i=1, 2, …m dan j=1, 2, …n

Dimana:

yij = Elemen matriks ternormalisasi [i][j]

wi = Bobot [i] dari proses AHP

3. Menentukan matrik solusi ideal positif dan solusi ideal negatif

$$A^+ = (y_1^+, y_2^+, \ldots, y_n^+ \quad (2.15)$$

$$A^- = (y_1^-, y_2^-, \ldots, y_n^- \quad (2.16)$$

Dimana:

$$y_j^+ = \{\frac{max_i y_{ij} ; jika\ j\ adalah\ atribut\ keuntungan}{min_i y_{ij} ; jika\ j\ adalah\ atribut\ biaya}$$

$$y_j^- = \{\frac{min_i y_{ij} ; jika\ j\ adalah\ atribut\ keuntungan}{max_i y_{ij} ; jika\ j\ adalah\ atribut\ biaya}$$

4. Menentukan jarak antara nilai setiap alernatif dengan matrik solusi ideal positif dan negatif

$$D_i^+ = \sqrt{\sum_{j=1}^{n} (y_{ij} - y_i^+)^2} \quad (2.17)$$

$$D_i^- = \sqrt{\sum_{j=1}^{n} (y_{ij} - y_i^-)^2} \quad (2.18)$$

Dimana:

D_i^+ = Jarak alternatif ke-i dengan solusi ideal positif

y_i^+ = Elemen solusi ideal positif [i]

y_j^- = Elemen solusi ideal negative [i]

y_{ij} = Elemen matriks ternormalisasi terbobot [i][j]

5. Menentukan nilai preferensi untuk setiap alternatif

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}$$

(2.19)

Dimana:

V_i = Kedekatan tiap alternatif terhadap solusi ideal

D_i^+ = Jarak alternatif ke-i dengan solusi ideal positif

D_i^- = Jarak alternatif ke-i dengan solusi ideal negative

Dengan nilai V_i yang lebih besar menunjukan bahwa nilai alternatif ke-i lebih dipilih.

2.2.9. Python

Python merupakan salah satu bahasa pemrograman yang sering digunakan dalam membuat program. Python memiliki sintaks yang tidak rumit sehingga mudah untuk digunakan (Mahawardana & dkk, 2022). Saat ini, Python adalah bahasa pemrograman yang paling banyak digunakan keempat atau kelima di dunia. Pengembang perangkat lunak dapat memanfaatkan beberapa fitur yang ditawarkan oleh pemrograman Python. Berikut adalah beberapa fitur dari bahasa pemrograman Python (Lutz, 2010):

1. Library Support
2. Multi Paradigm Design
3. Extendable
4. Simplicity
5. Open Source
6. Portability
7. Scalability

Beberapa keuntungan dari pemrograman Python termasuk *readability*, multifungsi, efisiensi, *interoperabilitas*, dan komunitas yang ramah. (Wahyono, 2018). Python dianggap andal karena kode sumbernya yang sederhana, yang membuatnya mudah untuk ditulis, dibaca, dan digunakan dengan cara yang

sederhana. Ini memfasilitasi pengembangan aplikasi untuk pengkodean, pengujian, dan koreksi jika terjadi masalah, bug, atau kesalahan lainnya. Python disebut efisien karena memiliki pustaka yang besar dan kode Python lebih mudah dibaca dibandingkan dengan kode yang ditulis untuk program lain. Python mempunyai banyak modul yang sering dimanfaatkan untuk mengembangkan aplikasi sesuai dengan kebutuhan. Python memiliki interoperabilitas tinggi karena dapat berkomunikasi dengan bahasa pemrograman lainnya. Program yang dikembangkan dengan Python dapat digunakan di hampir setiap sistem operasi, termasuk Windows, Mac OS, Linux, Unix, dan platform seluler seperti Android dan iOS. Berkat sifat open source-nya, Python memiliki komunitas yang sangat kuat. Memiliki komunitas yang baik memudahkan pengguna untuk berkomunikasi dan mengubah Python menjadi bahasa tulisan tangan. (Pramudita, 2020).