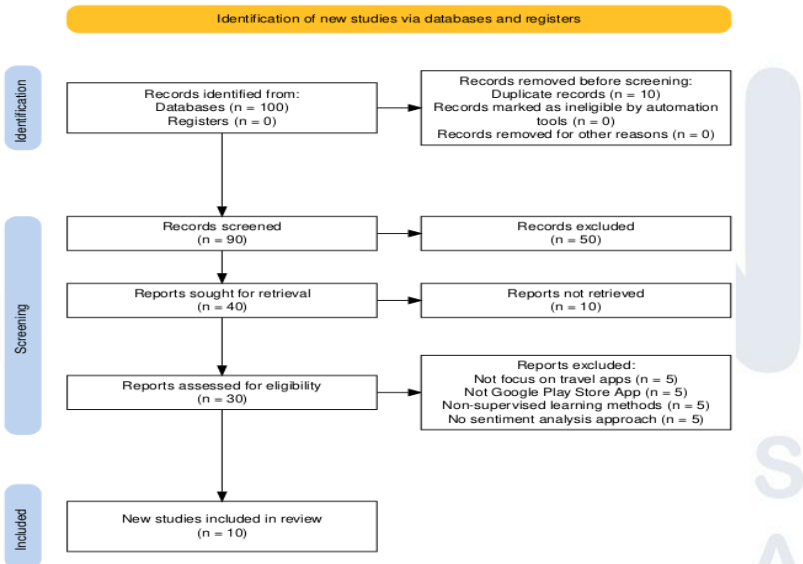
BAB II
LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu diperoleh melalui Systematic Literature Review (SLR) dengan menggunakan metode PRISMA. SLR merupakan salah satu pendekatan yang digunakan untuk mengidentifikasi, menyeleksi, menilai, dan menganalisis penelitian yang telah dilakukan pada bidang tertentu[21]. Sementara itu, metode PRISMA adalah pedoman berbasis bukti yang membantu dalam melakukan tinjauan sistematis dan meta-analisis secara lebih terstruktur[22]. Pada penelitian ini, proses SLR dengan metode PRISMA diawali dengan mengumpulkan artikel jurnal yang relevan dengan topik penelitian, dengan total sebanyak 100 jurnal yang berhasil dikumpulkan. Selanjutnya, dilakukan proses penyaringan untuk memastikan jurnal yang digunakan memiliki relevansi dan kualitas yang sesuai dengan kebutuhan penelitian. Setelah melalui tahap seleksi, diperoleh 10 jurnal yang digunakan sebagai dasar dalam penelitian ini. Berikut merupakan diagram PRISMA yang menggambarkan proses pemilihan penelitian terdahulu dalam penelitian ini.



Gambar 2. 1 Metodologi PRISMA

Gambar 2.1 menunjukkan metodologi PRISMA yang digunakan Dari 100 artikel awal, disaring jadi 90, lalu 40, hingga akhirnya tersisa 30. Setelah seleksi

ketat berdasarkan kriteria: bukan tentang *travel apps* (-5), bukan dari *Google Play Store* (-5), bukan metode *supervised learning* (-5), dan tanpa pendekatan analisis sentimen (-5), akhirnya hanya 10 studi yang relevan dan layak digunakan sebagai acuan penelitian.. Berikut ini adalah 10 artikel jurnal penelitian terdahulu yang didapatkan dengan teknik SLR melalui metode PRISMA.

Tabel 2. 1 Penelitian Terdahulu

Nama Penelitian	Jurnal	Nama Penulis	Metode	Hasil Penelitian
Analisis Sentimen pada Ulasan Pengguna Aplikasi Bibit dan Bareksa dengan Algoritma KNN	Vol 8 No 2 (2021): JATISI (Jurnal Teknik Informatika dan Sistem Informasi)	Aluisius Dwiki, Adhi Putra, Safitri Juanita	**KNN**	**Aplikasi Bibit:** Akurasi yang diperoleh adalah **85,14%.** **Aplikasi Bareksa**: Akurasi yang diperoleh adalah **81,70%**.
Analisis Sentimen Kepuasan Pengguna Aplikasi WhatsApp Menggunakan Algoritma Naïve Bayes dan SVM	Vol 6 No 2 (2021): @is The Best : Accounting Information Systems and Information Technology Business Enterprise	Acep Saepulrohman, Sudin Saepudin, Dudih Gustian	**Naïve Bayes dan SVM**	Akurasi untuk algoritma Naive Bayes adalah **70,40%** untuk algoritma Support Vector Machine (SVM) adalah **77,00%.**
Implementasi Algoritma K-Nearest Neighbor (KNN) untuk Analisis Sentimen Pengguna Aplikasi Tokopedia	Vol. 2 No. 2 (2023): Intellect : Indonesian Journal of Learning and Technological Innovation	M. Rival Ridautal Lillah, Dian Sa'adillah Maylawati,Wildan Budiawan Zulfikar, Wisnu Uriawan,Agung Wahana	**KNN**	Untuk 70:30, akurasinya 89% Untuk 80:20. Akurasinya 90.5%
Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer: Vol 6 No 9 (2022) September 2022	Fanka Angelina Larasati , Dian Eka Ratnawati, Buce Trias Hanggara	**Random Forest**	Akurasi sebesar 84% untuk pembagian data 80:20
Analisis Sentimen Pada Rating Aplikasi				

Nama Penelitian	Jurnal	Nama Penulis	Metode	Hasil Penelitian
Shopee Menggunakan Metode Decision Tree Berbasis SMOTE	JURNAL TEKNOLOGI INFORMASI: Vol. 18 No. 2 (2021)	Christian Cahyaningtyas, Yessica Nataliani, Indrastanti Ratna WIdiasari	**Decision Tree**	Akurasi yang didapat menggunakan SMOTE sebesar 99,91%. Tanpa SMOTE Akurasi yang didapat sebesar 99,89%
Analisis Sentimen Review Pada Aplikasi Media Sosial Tiktok Menggunakan Algoritma KNN dan SVM Berbasis PSO	Jurnal INformatika Kaputama (JIK), Vol. 7. No. 2, Juli 2023	Dian Ardiansyah, Atang Saepudin, Riska Aryanti, Eka Fitriani, Royadi	**SVM dan KNN**	Mendapat hasil ahkir Akurasi SVM 86.40 % dan hasil akhir Akurasi KNN 83.40 %
Analisis Sentimen Pengguna Online Travel Agent Pada Perusahaan Pegipegi.com Menggunakan Random Forest	JURNAL GAUSSIAN E-ISSN:2339-2541: Vol 12, No 4 (2023)	Ayu Lestari, Rukun Santoso, Suparti	**Random Forest**	Mengklasifikasikan opini pengguna aplikasi **Pegipegi**, dengan akurasi sebesar **92.27%** dan nilai **AUC-ROC sebesar 82.35%**
Analisis Sentimen Aplikasi Tiket Online di Play Store Menggunakan Metode SVM	SISMATIK (Seminar Nasional Sistem Informasi dan Manajemen Informatika) Universitas Nusa Putra, 7 Agustus 2021	FathurahmanBei, Sudin Saepudin	**SVM**	Akurasi model tertinggi diperoleh pada aplikasi **Pegipegi (78.21%)**, diikuti oleh **Agoda (77.00%)**, **Traveloka (75.03%)**, dan **Mister Aladin (64.00%)**. Sementara itu, **Tiket.com** memiliki akurasi terendah sebesar **58.68%**.
Sentimen Analisis Terhadap Aplikasi pada Google PLaystore Menggunakan Algoritma Naïve Bayes	Jurnal Komtika (Komputasi dan Informatika) : Vol, 5 No. 1 Mei 2021	Arif Rahman, Ema Utami, Sudarmawan	**Naïve Bayes**	Akurasi Naïve Bayes mencapai 96,53 %, 95,54% dan 95,54% dari 3 aplikasi pada Google Play Store

Nama Penelitian	Jurnal	Nama Penulis	Metode	Hasil Penelitian
dan Algoritma Genetika				
Analisis Sentimen Access by Bus Kota se-Indonesia Menggunakan Metode K-Nearest Neighbors	JURNAL ILMIAH SAINS TEKNOLOGI DAN INFORMASI (JITI):Volume 3 Nomor 1 Januari 2025	M. Andrik Muqorrobin P, Zaehol Fatah	**KNN**	dengan pembagian 90% data latih, 10% data uji dan menggunakan nilai k = 5 dengan nilai akurasi, presisi, dan recall secara berurutan sebesar 90,23%

Beberapa informasi pada Tabel 2.1 telah dilakukan terkait analisis sentiment pada aplikasi berbasi layanan digital, termasuk aplikasi keuangan, e-commerce, transportasi, hingga *online travel agency* (OTA). Mayoritas penelitian terdahulu menggunakan algoritma machine learning seperti KNN, Naïve Bayes, SVM, Decision Tree dan Random Forest, dengan hasil akurasi yang bervariasi.

Penelitian terdahulu belum banyak membahas optimasi model melalui hyperparameter tuning, padahal teknik ini berpotensi meningkatkan performa model dalam menganalisis sentiment pengguna. Selain itu, hanya sedikit penelitian yang mempertimbangkan penanganan data imbalance, dengan satu studi yang menerapkan SMOTE pada metode Decision Tree [20].

Untuk mengisi gap tersebut, penelitian ini tidak hanya melakukan perbandingan algoritma dalam analisis sentiment, tetapi juga mengimplementasikan hyperparameter tuning untuk memperoleh parameter terbaik untuk masing masing algoritma. Selain itu, penelitian ini menerapkan SMOTE untuk menangani ketidakseimbangan kelas dalam data, yang dapat mempengaruhi performa model dalam mengklasifikasi sentiment secara lebih akurat. Dengan demikian, penelitian ini diharapkan dapat memberikan hasil yang lebih optimal dibandingkan dengan pendekatan yang digunakan dalam penelitian sebelumnya.

2.2Tinjauan Teori

2.2.1 Analisis Sentimen

Analisis sentimen mengacu kepada bidang yang luas dari pengolahan bahasa alami, komputasi *linguistic* dan *text mining* dengan

tujuan menganalisis sebuah pendapat sentiment, evaluasi, sikap, emosi sesorang, kualitas sebuah produk, layanan, ataupun kegiatan tertentu[23]. Analisis sentimen merupakan Teknik mengekstrak data teks untuk mendapatkan informasi tentang sentiment yang bernilai positif, netral dan negatif[24]. Analisis sentimen juga bisa menjadi metode penggunaan analisis teks untuk mengumpulkan data dari berbagai sumber di internet dan berbagai platform media sosial. Sentiment analysis juga merupakan bidang dari *Natural Language Processing* (NLP) yang membangun system untuk mengenali dan ekstrak opini dalam bentuk teks. Tujuan dari proses ini adalah untuk mendapatkan pandangan atau pendapat dari pengguna yang terdapat di platform tersebut [25].

2.2.2 Teorema Bayes

Teorema Bayes merupakan sebuah rumus matematika yang digunakan untuk menghitung probabilitas suatu kejadian berdasarkan informasi atau bukti yang ada [26]. Teorema bayes juga merupakan konsep fundamental dalam statistika yang memungkinkan pembaruan hipotesis seiring adanya informasi terbaru[27]. Konsep ini banyak diterapkan berbagai bidang seperti AI, pemrosesan bahasa alami serta analisis data[28].

Rumus dari Teorema Bayes adalah[29] :

$$P(A|B) = P(B|A) \cdot P(A) / P(B) \quad (2.1)$$

Keterangan :

P(A|B) : probabilitas kejadian A terjadi, jika diketahui kejadian B telah terjadi

P(B|A) : probabilitas kejadian B terjadi, jika diketahui kejadian A telah terjadi

P(A): probabilitas kejadian A terjadi secara umum

P(B): probabilitas kejadian B terjadi secara umum

2.2.3 Text Mining

Text mining merupakan proses suatu penambangan data dalam bentuk teks[30]. *Text mining* atau praproses teks digunakan untuk *case folding, tekonizing, stopwords*, dan *stemming*[31]. *Text mining* melibatkan pengubahan data teks yang tadinya tidak terstruktur menjadi terstuktur oleh proses yang dilakukan komputer[32]. *Text mining* menganalisis data dalam ukuran yang besar sehingga data teks tidak terstruktur dengan bantuan perangkat lunak yang dapat mengidentifikasi konsep, pola, topik, kata kunci dan atribut lain pada data[33]. *Text mining* juga memiliki tujuaun untuk mendapatkan informasi atau menggali informasi yang berguna dari berbagai dokumen. *Text mining* juga mendukung *Knowledge Discovery* pada beberapa dokumen yang besar[34].

Berikut merupakan penjelasan mengenai *case folding, tokenizing, stopwords* dan *stemming:*

a. Case folding

Case folding merupakan proses pengubahan huruf / kata-kata menjadi huruf kecil. Tujuannya adalah untuk menghilangkan perbedaan kapitalisasi sehingga kata yang sama tidak akan dianggap berbeda oleh sistem[35].

b. Tokenizing

Tokenizing merupakan proses memecah teks menjadi unit-unit kecil yang disebut tokens.Tokenizing bisa berupa kata, frasa, bahkan karakter tergantung pada metode yang digunakan. Tokenizing bertujuan untuk membantu menganalisa teks karena setiap katanya nanti akan di proses secara independent oleh sistem[36].

c. Stopwords

Stopwords merupakan proses menghapus kata-kata umum yang tidak memiliki makna yang signifikan dalam proses analisis teks. Stopwords biasanya terdiri dari kata kata seperti “dan”, “di”, “ke”, “yang”, dll.

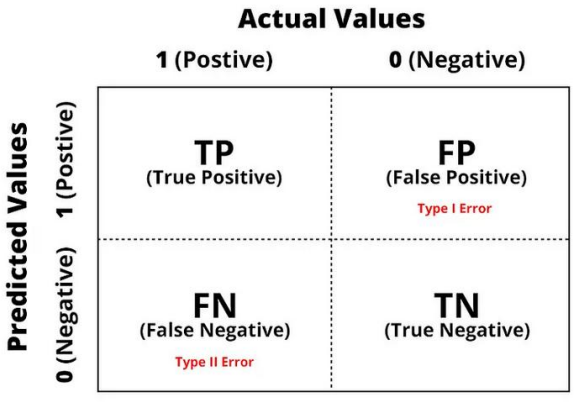
Menghapus kata kata yang tidak penting dapat membantu meningkatkan efisiensi model[37].

d. Stemming

Stemming merupakan proses mengubah kata menjadi bentuk dasarnya dengan menghilangkan imbuhan. Tujuannya adalah untuk menyederhanakan kata sehingga variasi kata dengan makna yang sama dapat dikelompokkan bersama [38]. Proses ini penting dalam analisis teks karena membantu mengurangi kompleksitas data dan meningkatkan konsistensi representasi kata dalam tahap pemrosesan lanjutan seperti klasifikasi atau analisis sentimen.

2.2.4 Confusion Matrix

Confusion matrix merupakan sebuah matriks yang dilakukan oleh system untuk memuat data klasifikasi baik secara actual maupun prediktif[39]. *Confusion matrix* digunakan untuk mengevaluasi performa suatu model klasifikasi dengan membandingkan hasil prediksi model dengan label kelas yang sebenarnya dari data yang digunakan untuk training[40]. Confusion Matrix terdiri dari empat jenis nilai, yaitu true positive (TP), true negative (TN), false positive (FP), dan false negative (FN)[41].



Gambar 2. 2 Confusion Matrix[42].

Berdasarkan Gambar 2.2 masing masing nilai tersebut merepresentasikan jumlah data yang benar benar positif, benar benar negative, salah diprediksi sebagai positif, dan salah diprediksi sebagai negative oleh model. Metrik evaluasi kinerja seperti akurasi, presisi, recall, F1-score dan sejenisnya dapat dihitung berdasarkan nilai nilai tersebut. Confusion matrix menjadi penting dalam evaluasi model klasifikasi dan membantu pengembang model dalam meningkatkan performanya[43].

Berikut merupakan rumus dalam menghitung performa pada *confusion matrix* [44] :

1. *Accuracy*

Accuracy merupakan perbandingan antara data yang telah d iklasifikasikan benar dengan keselurhan data.

$$Accuracy = TP+TN / TP + FP + TN + FN \quad (2.2)$$

2. *Precision*

Precision merupakan perbandingan antara data positif yang diklasifikasikan benar dengan data yang diklasifikasikan positif.

$$Precision = TP / TP + FP \quad (2.3)$$

3. *Recall*

Recall akan menunjukan seberapa berhasil data positif terklasifikasi dengan benar positif.

$$Recall = TP / TP + FN \quad (2.4)$$

4. *F1-score*

Nilai *f1 score* akan direpresentasikan dari hasil antara nilai precision dan juga nilai *recall* antara kategori yang diprediksi dengan kategori sebenarnya.

$$F1 = 2 \times (Recall \times Precision) / (Recall + Precision) \quad (2.5)$$

2.2.5 Web Scraping

Salah satu tahapan penting dalam proses pengumpulan data pada penelitian ini adalah *web scraping*. Teknik ini digunakan untuk mengambil data ulasan pengguna dari situs tertentu, dalam hal ini Google Play Store. Web scraping memungkinkan peneliti memperoleh informasi secara otomatis dan terstruktur tanpa harus mengambil data secara manual. Proses ini dilakukan menggunakan bahasa pemrograman Python dengan bantuan library seperti Beautiful Soup. Penjelasan mengenai konsep dan proses kerja web scraping disajikan pada bagian berikut.

Gambar 2. 3 Web Scraping

Web Scraping merupakan suatu teknik yang digunakan untuk mendapatkan suatu data atau informasi pada website tertentu. Informasi yang didapat berupa, teks, tautan, video, audio ataupun dokumen[45]. Web scraping dapat dilakukan dengan cara ekstraksi informasi menggunakan phyton serta library Beautiful Soup. Beautiful Soup disini nantinya akan digunakan untuk menerjemahkan elemen pada tag html yang bertujuan untuk mengambil isi teks nya[46].

2.2.6 TF - IDF

TF-IDF atau *term frequency- inverse document frequency* yang menjadi salah satu *feature extraction* dapat digunakan pada tahap persiapan data]. *TF-IDF* akan mengevaluasi frekuseni dari kata kata yang dianggap berguna atau memiliki bobot lebih. Hasil akhirnya dapat digunakan untuk mengidentifikasi sentimen[48]. Metode ini digunakan untuk Analisa teks karena dapat menyoroti kata kata yang memiliki relevansi tinggi[49].

Berikut merupakan rumus untuk menghitung *Term Frequency*[50]:

$$Tf = 0{,}5 + 0{,}5 \times (tf) / \max(tf) \quad (2.6)$$

Keterangan:

Tf= banyaknya kata yang dicari pada sebuah data

max(tf)= jumlah kemunculan terbanyak term pada data yang sama

Selain itu, untuk menghitung *IDF* dengan cara [50]:

$$IDF(t, D) = \log(\text{jumlah total dokumen} / (\text{jumlah dokumen yang mengandung kata } t + 1)) \quad (2.7)$$

Keterangan :

T merupakan kata yang dihitung

D merupakan Kumpulan dokumen

2.2.7 E- Commerce

E-commerce dalam bahasa Indonesia dikenal dengan istilah "Perdagangan Elektronik" atau "E-niaga". *E-commerce* adalah suatu proses terjadinya transaksi jual beli yang dalam prakteknya dilakukan secara online melalui media elektronik seperti internet, televisi, dan lain-lain. E-commerce mencakup kegiatan belanja online, namun tidak hanya itu, melainkan juga berbagai aktivitas yang melibatkan transaksi online, seperti

internet banking dan e-wallet, hingga pemesanan tiket, akomodasi, dan lain lain[51].

2.2.9 SMOTE

SMOTE (*Synthetic Minority Over Sampling Technique)* merupakan salah satu metode over-sampling yang digunakan untuk menangani data *imbalance* atau ketidak seimbangan pada data. Ketidakseimbangan ini terjadi ket0069ka satu kelas memiliki jumlah sampel yang jauh lebih sedikit dibandingkan kelas lainnya[52]. SMOTE (*Synthetic Minority Over Sampling Technique)* menghasilkan data *training* sintesis yang baru dengan menginterpolasi linier untuk kelas minorias[53]. Data buatan tersebut nantinya dibangkitkan berdasarkan atribut yang berasal dari KNN. Untuk jumlah KNN, dapat ditentukan sesuai dengan pertimbangan kemudahan untuk menjalankannya [54].

2.2.10 Hyperparameter Tuning

Salah satu langkah penting dalam meningkatkan kinerja model adalah penyesuaian hyperparameter. Hyperparameter adalah parameter yang nilainya harus ditentukan sebelum pelatihan model dimulai dan tidak didapat langsung dari proses pembelajaran[55]. Perilaku algoritma akan dikontrol oleh hyperparameter, seperti kecepatan konvergensi, kompleksitas model, dan kemampuan generalisasi[55]. Tujuan pengaturan hyperparameter adalah untuk menemukan kombinasi terbaik yang akan memaksimalkan kinerja model (akurasi, presisi, atau skor f1) pada data validasi/tes[56].

Berikut merupakan beberapa pendekatan umum dalam hyperparameter tuning:

- Grid Search

Dalam hal ini, Grid Search akan memeriksa semua kombinasi hyperparameter yang ditentukan dalam grid[57].

- Random Search

Random search akan memilih kombinasi hyperparameter secara acak dari distribusi yang ditentukan[57].

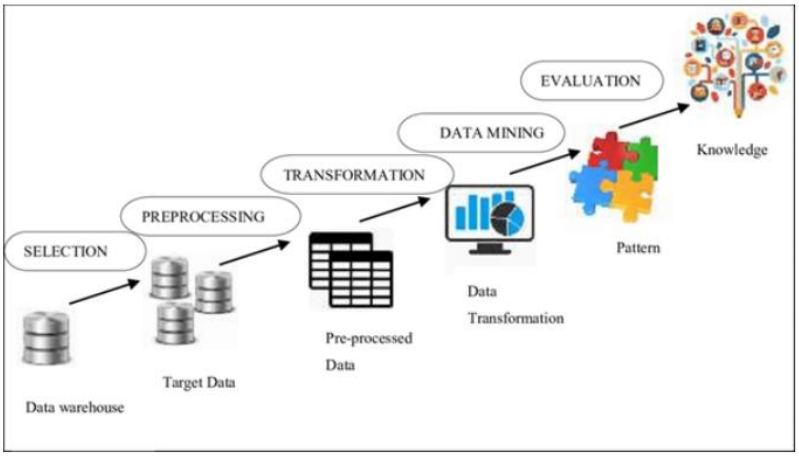
- Bayesian Optimazation

Optimalisasi Bayesian akan menggunakan probabilitas untuk membantu pencarian kombinasi terbaik.[29].

2.3Framework & Algoritma

2.3.1 Knowledge Discovery in Databases (KDD)

Knowledge Discovery in Database (KDD) merupakan proses persimpangan dari beberapa ilmu statistic, database, AI, komputer parallel, dan visualisasi yang mempengaruhi "*interdisciplinary kwowledge"*[58]. KDD meliputi banyak kegiatan seperti pengumpulan, pemakaian data, historis untuk menemukan keteraturan, pola, serta hubungan dalam set data yang besar[59]. Proses ini bertujuan untuk mengekstrak informasi berharga yang dapat digunakan dalam pengambilan keputusan dan analisis lebih lanjut[60].



Gambar 2. 4 Metode KDD[61]

Gambar 2.4 menunjukkan tahapan-tahapan utama dalam metode KDD yang meliputi[62]:

a) *Selection*

Tahap ini melibatkan pemilihan sumber data yang relevan untuk analisis. Pemilihan data ini harus mempertimbangkan tujuan analisis dan jenis

informasi yang ingin diekstrak. Misalnya, dalam konteks penelitian, pemilihan data bisa mencakup memilih data ulasan pengguna aplikasi Traveloka di Indonesia sebagai sumber informasi.

b) Pre Processing

Pada tahap ini, data yang diperoleh dari sumber dipersiapkan untuk analisis lebih lanjut. *Pre Processing* melibatkan tugas seperti penghapusan data duplikat, penanganan missing values, normalisasi data (misalnya, mengubah skala), dan pembersihan data. Untuk data teks, pemrosesan teks seperti tokenisasi, penghapusan tanda baca, dan stemming juga dapat diterapkan.

c) Transformation

Transformasi data merupakan langkah menyiapkan data untuk diumpankan ke algoritma data mining.

d) *Data Mining*

Tahap data mining adalah inti dari metode KDD. Pada tahap ini, teknik analisis seperti pemodelan statistik, pembelajaran mesin, atau algoritma data mining digunakan untuk menggali pengetahuan dari data. Misalnya, dalam penelitian sentimen pengguna Traveloka, algoritma *supervised learning* digunakan untuk mengklasifikasikan sentimen ulasan.

e) Evaluation

Proses evaluation merupakan tahap akhir yang dilakukan setelah pembentukan model berhasil terbentuk pada proses data mining. Tujuan dari proses evaluation ini untuk mengukur performa dari model yang telah dibangun. Melalui tahap ini, dapat diketahui sejauh mana model mampu melakukan prediksi atau klasifikasi secara akurat berdasarkan data yang diberikan.

2.3.2 Naïve Bayes

Naïve bayes merupakan algoritma metode klasifikasi sederhana untuk mencari peluang yang memanfaatkan teorema probabilititas dengan cara

memprediksi probabilitas dimasa depan berdasarkan informasi di masa sebelumnya[63]. Setiap data baru akan dilakukan probabilitas dengan setiap class yang ada. Hasil akhir dilihat nilai yang paling tinggi, sehingga algoritma ini dirasa cukup baik untuk menentukan probabilitas dalam menentukan hasil[64]. Naïve bayes juga memanfaatkan Teorema Bayes dengan asumsi fitur yang digunakan untuk klasifikasi adalah independent satu sama lain. Naïve bayes sendiri sudah terbukti mempunyai kecepatan dan akurasi yang tinggi. Ketika digunakan untuk database dengan data yang besar. Untuk klasifikasi naïve bayes ini memiliki kemampuan klasifikasi yang serupa dengan neural network dan decision tree [65].

Rumus dari Teorema Bayes adalah[29] :

$$P(A|B) = P(B|A) \cdot P(A) / P(B) \quad (2.8)$$

Keterangan :

P(A|B) : probabilitas kejadian A terjadi, jika diketahui kejadian B telah terjadi

P(B|A) : probabilitas kejadian B terjadi, jika diketahui kejadian A telah terjadi

P(A): probabilitas kejadian A terjadi secara umum

P(B): probabilitas kejadian B terjadi secara umum

Tabel 2. 2 Kelebihan dan Kekurangan Naive Bayes

Kelebihan Naïve Bayes[66]	**Kekurangan Naïve Bayes**[67]
Cepat dan efisien dalam training dan prediksi	Asumsi fitur independent jarang terpenuhi di dunia nyata
Cocok untuk data teks seperti analisis sentiment	Kurang akurat jika data tidak seimbang atau fitur saling terkait
Mudah diimplementasikan dan digunakan	Tidak menangani interaksi antar fitur dengan baik

2.3.3 K-Nearest Neighbors (KNN)

K-nearest neighbor (KNN) merupakan algoritma supervised dimana hasil klasifikasinya berdasarkan yang paling banyak / mayoritas[68]. Algoritma *K-Nearest Neighbor* merupakan salah satu metode klasifikasi data mining, KNN mengklasifikasikan sekumpulan data berdasarkan data pembelajaran diberi label. KNN termasuk ke dalam supervised learning yang digunakan untuk klasifikasi objek baru berdasarkan objek terdekatnya. Hasil query instance yang baru, akan diklasifikasikan berdasarkan yang paling banyak jumlahnya atau mayoritas dari kategori pada KNN, dapat diartikan juga kelas yang paling banyak muncul akan dijadikan sebagai kelas klasifikasi[65].

Metode ini bergantung pada kesamaan antara dokumen pleatihan dan dokumen tes dalam hal albel kategori yang melekat pada dokumen tersebut. Cara kerja algoritma ini adalah mencari sejumlah k pola terdekat dengan pola masukan, dan kemudian menentukan keputusan berdasarkan jumlah pola terbanyak diantara k pola tersebut [69].

KNN menggunakan metrik jarak untuk menentukan kedekatan antar data. Salah satu metrik yang paling umum digunakan adalah *Euclidean Distance,* yang dihitung dengan rumus[70]:

$$d(x, y) = \sqrt{((x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2)} \quad (2.9)$$

Keterangan :

d(x, y) adalah jarak antara dua titik data x dan y.

x_i dan y_i adalah nilai fitur ke-i dari masing-masing data.

n adalah jumlah fitur dalam dataset.

Semakin kecil nilai d(x, y), semakin dekat dua data tersebut.

Berikut merupakan kelebihan dan kekurangan KNN[71] :

Tabel 2. 3 Kelebihan dan Kekurangan KNN

Kelebihan KNN[72]	**Kekurangan KNN**[73]
Algoritma dapat mengatasi data noisy	KNN perlu menentukan nilai parameter K (jumlah dari tetangga terdekat)

Kelebihan KNN[72]	Kekurangan KNN[73]
Algoritma KNN dapat menanggulangi data yang jumlahnya besar	Pembelajaran berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk dapat hasil yang terbaik
Mudah diimplementasikan	Daya komputasi cukup tinggi karena memerlukan perhitungan sebuah jarak dari tiap sample uji pada keseleruhan sample

2.3.4 Decision Tree

Pohon keputusan, atau decision tree merupakan salah satu metode klasifikasi dalam data mining yang populer dan sering digunakan. Metode ini banyak dipilih karena kemudahannya dalam interpretasi oleh pengguna. Decision tree berfungsi untuk membagi sekumpulan data menjadi subset yang lebih kecil dengan menerapkan aturan-aturan keputusan yang terbentuk[74].

Metode ini banyak digunakan karena mudah dipahami dan dipahami oleh orang-orang, bahkan mereka yang tidak terlalu teknis. Karena menyerupai cara manusia membuat keputusan logis berdasarkan serangkaian kondisi, proses klasifikasi menggunakan pohon keputusan sangat mudah dipahami[75].

Tabel 2. 4 Kelebihan dan Kekurangan Decision Tree

Kelebihan Decision Tree[76]	Kekurangan Decision Tree[76]
Tidak memerlukan persiapan data yang signifikan, seperti normalisasi.	Rentan terhadap penyesuaian berlebihan atau overfitting. Terutama jika pohonnya dalam atau kompleks
Bisa menangani data kategorikal dan numerik.	Struktur pohon dapat sangat berbeda jika ada perubahan kecil dalam data
Memungkinkan untuk menginterpretasikan aturan keutusan dalam bentuk logika if-else.	Jika tidak dipruning atau dikombinasikan (seperti dalam random forest), kurang akurat

2.3.5 SVM

Support Vector Machine (SVM) adalah metode pembelajaran terawasi (supervised learning) yang digunakan untuk tugas klasifikasi. Model SVM bekerja dengan memisahkan setiap kelas atau label menggunakan margin seluas mungkin. SVM sendiri dapat diterapkan dalam bentuk Support Vector Classification (SVC) maupun Support Vector Regression (SVR)[77].

Algoritma SVM menentukan hyperplane terbaik dengan mencari jarak maksimum antara kelas-kelas yang dipisahkan. Dalam SVM, hyperplane

berfungsi sebagai batas keputusan yang memisahkan satu kelas dengan kelas lainnya. Pada ruang dua dimensi, batas keputusan ini disebut garis (line), sedangkan pada tiga dimensi disebut bidang (plane). Jika data memiliki lebih dari tiga dimensi, batas keputusan tersebut dikenal sebagai hyperplane[78].

Selain itu, SVM sangat efektif dalam menangani data berdimensi besar dan juga dalam kasus klasifikasi non-linear. Untuk data yang tidak dapat dipisahkan secara linear, SVM menggunakan pendekatan trick kernel, yaitu teknik yang memproyeksikan data ke ruang dimensi yang lebih besar sehingga dapat dipisahkan secara linear di ruang tersebut[79]. Beberapa fungsi kernel yang biasa digunakan oleh SVM antara lain:

- Jika data linear, kernel linear digunakan.
- Kernel polinomial cocok untuk data yang memiliki hubungan polinomial.
- Radial Basis Function (RBF), juga dikenal sebagai Gaussian Kernel, sangat populer untuk menangani data non-linear.
- Sigmoid Kernel Fungsinya mirip dengan fungsi aktivasi neural network.

Tabel 2. 5 Kelebihan dan Kekurangan SVM

Kelebihan SVM[80]	**Kekurangan SVM**[81]
Akurat untuk data dengan dimensi tinggi	Proses training bisa lama untuk dataset
Efektif memisahkan kelas yang tidak linear dengan kernel	Pemilihan kernel dan parameter butuh tuning yang tepat
Tahan terhadap overfitting pada ruang fitur tinggi	Kurang cocok untuk data yang sangat besar

2.3.6 Random Forest

Random Forest adalah pengembangan dari Decision Tree yang menggunakan beberapa pohon keputusan untuk meningkatkan akurasi model. Setiap pohon dilatih dengan sampel yang dipilih secara acak, sementara pemisahan atribut dilakukan dari subset atribut yang juga dipilih secara acak. Metode ini memiliki keunggulan seperti ketahanan terhadap outlier, kemampuan menangani data yang hilang, serta efisiensi dalam penyimpanan.

Selain itu, Random Forest dapat melakukan seleksi fitur untuk memilih atribut terbaik, sehingga meningkatkan kinerja model dalam klasifikasi[82].

Random Forest adalah contoh metode pembelajaran kelompok di mana banyak model (dalam hal ini pohon keputusan) digabungkan untuk menghasilkan prediksi yang lebih akurat dan stabil[53]. Proses penggabungan ini dilakukan dengan menggunakan teknik bagging yang dikenal sebagai Bootstrap Aggregating, yaitu dengan membuat subset data yang diambil secara acak dengan pengembalian, lalu hasil prediksinya digabungkan (misalnya, dengan memberikan suara untuk klasifikasi atau rata-rata untuk regresi)[83].

Tabel 2. 6 Kelebihan dan Kekurangan Random Forest

Kelebihan random forest[84]	**Kekurangan random forest**[84]
Mengurangi *overfitting*: Random Forest lebih stabil daripada Decision Tree satu karena prediksi dibuat dari agregasi beberapa pohon.	Proses pelatihan dan prediksi lebih lama dibandingkan dengan satu Decision Tree, terutama ketika ada banyak pohon.
tahan terhadap gangguan dan data tidak seimbang.	Tidak seperti pohon keputusan yang dapat divisualisasikan, model yang dibuat seperti "kotak hitam", sehingga sulit bagi pengguna untuk memahaminya secara langsung.

2.4 Tools

2.4.1 Python

Python merupakan salah satu bahasa pemrograman yang banyak digunakan oleh perusahaaanbesar maupun para developer untuk mengembangkan berbagai macam aplikasi berbasis desktop, web dan mobile. Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak sintuitif dan memiliki pustaka yang luas[85].

Python memiliki keragaman kerangka kerjanya, Python mampu memberikan potongan kode dalam jumlah yang besar untuk mengembangkan suatu proyek[86]. Python juga sering digunakan dalam analisis sentimen. Dalam analisis sentimen, Python memberikan fleksibilitas kepada pengguna dalam mengelola teks dengan berbagai metode, baik itu melalui pendekatan statistik yang sederhana atau

pemanfaatan model machine learning yang lebih kompleks. Ini memberikan kesempatan kepada perusahaan dan peneliti untuk mendalami pandangan serta perasaan pengguna dengan lebih mendalam, yang selanjutnya dapat digunakan sebagai dasar untuk pengambilan keputusan strategis, perbaikan produk, dan pengembangan layanan yang lebih unggul.

2.4.2 Google Colab

Google colab merupakan *Web Integrated Development Environment* (IDE) web untuk Python[87]. *Google Colab* merupakan platform yang berbasi cloud dan free untuk menjalankan dan berbagi *notebook Jupyter*[88]. *Google colab* menyediakan hamper semua *library* yang dibutuhkan[89]. *Web scraping* merupakan teknik pengumpulan data otomatis dari halaman web. *Google Colab* menyediakan *library* dan lingkungan yang cocok untuk menjalankan script Python. Oleh karena itu, kita bisa memanfaatkan tools / platform *Google colab* sebagai sarana untuk melakukan *web scraping*.