BAB II
LANDASAN TEORI

2.1 Penelitian Terkait

Selama penyusunan skripsi ini, penulis mengacu pada berbagai penelitian terdahulu yang relevan dengan topik yang dibahas. Penelitian-penelitian tersebut menjadi dasar dalam memahami perkembangan analisis sentimen serta sebagai pembanding dalam penelitian ini. Selain itu, kajian penelitian terdahulu juga didukung dengan pendekatan Systematic Literature Review (SLR) untuk mengidentifikasi dan menganalisis penelitian secara sistematis.

Berikut ini disajikan rekapitulasi penelitian terdahulu yang relevan dengan penelitian ini dalam satu tabel komprehensif beserta tautan DOI masing-masing penelitian.

Tabel 2.1 Rekapitulasi Penelitian Terdahulu

No	Judul Penelitian	Tahun	Penulis	Algoritma / Metode	Hasil Utama
1	Analisis Sentimen Pengguna Dompet Digital Dana Pada Kolom Komentar Google Play Store Dengan Metode Klasifikasi Support Vector Machine	2023	Saputro et al.	SVM, Confusion Matrix	Akurasi 80%. Sentimen positif 35%, Negatif 65%. Precision negatif 84,06%; positif 74,08%.
2	Analisis Sentimen Pada	2024	Kusuma & Ali	Naïve Bayes, TF-IDF	Akurasi 82%, Precision 81%,

No	Judul Penelitian	Tahun	Penulis	Algoritma / Metode	Hasil Utama
	Pengguna Aplikasi Dana Menggunakan Algoritma Naive Bayes				Recall 82%, F1-Score 81%. Dominan keluhan fitur keamanan.
3	Analisis Sentimen Terhadap Review Aplikasi Brimo Di Google Play Store Menggunakan Algoritma Naive Bayes	2024	Nuraini et al.	Naïve Bayes, TF-IDF, KDD	Akurasi 89,58%, Precision 85,94%, Recall 89,58%, F1-Score 86,90%.
4	Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest	2022	Larasati et al.	Random Forest, Confusion Matrix	Precision 84%, Recall 84%, F1-Score 84%, Akurasi 84% Pada 1.354 data (80:20).
5	Perbandingan Metode Random Forest dan Naive Bayes pada Analisis Sentimen Review Aplikasi BCA	2023	Miftahusalam et al.	Random Forest, Naïve Bayes, TF-IDF	RF unggul: Akurasi 93,93 Precision 93,02 Recall 89,89%, F1-Score 91,43% pada 2.453 ulasan.

No	Judul Penelitian	Tahun	Penulis	Algoritma / Metode	Hasil Utama
	Mobile				
6	Eksplorasi Fitur Seleksi pada SVM dan Random Forest dalam Analisis Sentimen Aplikasi GoPay	2024	Aziz & Yamasari	SVM, Random Forest, Chi-square, Information Gain	SVM unggul 1% di atas RF. Fitur seleksi meningkatkan performa model.
7	Analisis Sentimen Kualitas Layanan Teknologi Pembayaran Elektronik pada Twitter (Studi Kasus OVO dan DANA)	2021	Pratiwi & Yustanti	Naïve Bayes, SVM, KNN	SVM terbaik: 81,33% (OVO) dan 86,23% (DANA). Isu utama: keandalan transaksi.
8	Sentiment Analysis on Google Play Store App Users Reviews Based on Deep Learning Approach	2024	Ramanathan., et al.	Deep Learning (LSTM, BERT), TF-IDF	BERT akurasi tertinggi 94,2%. Deep learning mengungguli metode klasik.
9	Performance Evaluation of Machine Learning	2023	Aslamet al.	Naïve Bayes, SVM, Random Forest,	RF akurasi 91,2%. Dataset imbalance menjadi tantangan utama.

No	Judul Penelitian	Tahun	Penulis	Algoritma / Metode	Hasil Utama
	Models on Large Dataset of Android Applications Reviews			Logistic Regression	
10	Comparative Analysis of Machine Learning Algorithms for Sentiment Classification in Social Media Text	2024	Jahan et al.	Naïve Bayes, SVM, Random Forest, Decision Tree	NB unggul pada teks pendek. SVM dan RF kompetitif pada data seimbang.

2.1.1 Perbandingan Penelitian Terdahulu dengan Penelitian Ini

Terdapat beberapa persamaan dan perbedaan penting antara penelitian ini dan 10 penelitian sebelumnya yang ditunjukkan pada Tabel 2.1. Perbandingan tersebut ditinjau dari berbagai aspek, meliputi objek penelitian, algoritma yang digunakan, jumlah data, kelas sentimen, serta platform pengambilan data.

Sebagian besar penelitian terdahulu menggunakan pendekatan *machine learning* dalam melakukan analisis sentimen pada aplikasi digital dan layanan keuangan berbasis teknologi. Algoritma yang paling sering digunakan meliputi Naive Bayes, Support Vector Machine (SVM), dan Random Forest. Selain itu, sebagian besar penelitian mengukur efektivitas model kategorisasi sentimen menggunakan kriteria penilaian termasuk akurasi, presisi, *recall*, dan *F1-Score*. Aplikasi dompet digital DANA, yang dapat digunakan sebagai tolok ukur yang relevan dalam menilai efektivitas algoritma kategorisasi sentimen, adalah objek penelitian lain yang digunakan oleh beberapa studi.

Dari sisi penggunaan algoritma, terlihat bahwa sebagian besar penelitian

terdahulu hanya menggunakan satu atau dua algoritma. Penelitian Saputro et al. [1] menggunakan SVM, Kusuma & Ali [2] serta Nuraini et al. [3] menggunakan Naive Bayes, sedangkan Larasati et al. [4] menggunakan Random Forest. Penelitian Miftahusalam et al. [5] dan Aziz & Yamasari [6] membandingkan dua algoritma, yaitu Naive Bayes dengan Random Forest serta SVM dengan Random Forest. Adapun penelitian yang menggunakan ketiga algoritma secara bersamaan, yaitu Naive Bayes, SVM, dan Random Forest, hanya dilakukan oleh Aslam et al. [9] dan Jahan et al. [10], namun keduanya menggunakan dataset umum yang tidak spesifik pada konteks fintech Indonesia. Penelitian ini melengkapi celah tersebut dengan membandingkan ketiga algoritma sekaligus pada ulasan aplikasi DANA menggunakan dataset sebanyak 15.089 data.

Selain dari aspek algoritma, penelitian ini juga memiliki perbedaan dari sisi objek dan platform penelitian. Penelitian Nuraini et al. [3] menggunakan objek aplikasi BRImo, Aziz & Yamasari [6] menggunakan GoPay, sedangkan Pratiwi & Yustanti [7] menggunakan data Twitter untuk analisis sentimen aplikasi DANA dan OVO. Penelitian ini secara khusus berfokus pada aplikasi DANA dengan sumber data yang berasal dari Google Play Store sebagai platform resmi ulasan pengguna Android, sehingga data yang diperoleh lebih representatif terhadap pengalaman pengguna secara langsung.

Dari aspek jumlah dan kelas data, sebagian besar penelitian terdahulu menggunakan jumlah data yang relatif kecil. Larasati et al. [4] menggunakan 1.354 data, sedangkan Miftahusalam et al. [5] menggunakan 2.453 data. Penelitian ini menggunakan dataset yang jauh lebih besar, yaitu sebanyak 15.089 ulasan pengguna aplikasi DANA periode 1 September 2024 sampai 30 November 2024. Penggunaan dataset yang lebih besar diharapkan mampu menghasilkan model klasifikasi yang lebih andal dan representatif. Selain itu, penelitian ini menggunakan tiga kelas sentimen, yaitu positif, negatif, dan netral, sehingga analisis yang dihasilkan menjadi lebih mendalam dibandingkan penelitian yang hanya menggunakan dua kelas sentimen.

Dari aspek konteks penelitian, Aslam et al. [9] dan Jahan et al. [10] menggunakan dataset umum aplikasi Android atau media sosial yang tidak secara khusus berfokus pada konteks fintech Indonesia. Untuk membuat temuan penelitian

lebih relevan dan mampu menawarkan saran yang lebih bermanfaat bagi pengembang aplikasi dalam meningkatkan kualitas layanan, studi ini secara khusus meneliti sentimen pengguna aplikasi dompet digital DANA, salah satu aplikasi *fintech* terkemuka di Indonesia.

Berdasarkan tinjauan ini, studi yang menggunakan algoritma Naive Bayes, Random Forest, dan SVM dibandingkan dengan penelitian sebelumnya.

2.1.2 Naive Bayes

Algoritma Naive Bayes digunakan pada penelitian Kusuma & Ali [2] yang melakukan analisis sentimen ulasan aplikasi DANA dengan hasil akurasi sebesar 82%, *precision* 81%, *recall* 82%, dan *F1-Score* 81%. Penelitian Nuraini et al. [3] juga menerapkan Naive Bayes pada ulasan aplikasi BRImo menggunakan framework KDD dan menghasilkan akurasi 89,58%, *precision* 85,94%, *recall* 89,58%, dan *F1-Score* 86,90%. Pratiwi & Yustanti [7] membandingkan Naive Bayes dengan SVM dan KNN pada data Twitter aplikasi OVO dan DANA, di mana Naive Bayes menghasilkan akurasi lebih rendah dibandingkan SVM. Miftahusalam et al. [5] membandingkan Naive Bayes dengan Random Forest pada ulasan BCA Mobile dan menyimpulkan bahwa Naive Bayes memiliki performa lebih rendah dibandingkan Random Forest.

Perbedaan utama penelitian ini dengan penelitian-penelitian tersebut terletak pada penggunaan dataset yang lebih besar, yaitu 15.089 ulasan aplikasi DANA, serta penerapan tiga kelas sentimen, yaitu positif, negatif, dan netral. Selain itu, penelitian ini tidak hanya menggunakan Naive Bayes secara tunggal, tetapi membandingkannya secara langsung dengan SVM dan Random Forest pada objek penelitian yang sama.

2.1.3 Random Forest

Penggunaan algoritma Random Forest pada penelitian terdahulu dapat ditemukan pada penelitian Larasati et al. [4] yang menganalisis sentimen ulasan aplikasi DANA menggunakan 1.354 data dan menghasilkan akurasi sebesar 84%, *precision* 84%, *recall* 84%, dan *F1-Score* 84%. Random Forest mengungguli Naive Bayes pada 2.453 ulasan BCA Mobile, dengan akurasi 93,93%, *precision* 93,02%, *recall* 89,89%, dan *F1-Score* 91,43%, menurut penelitian Miftahusalam et al. [5]. Aziz & Yamasari [6] juga menggunakan Random Forest bersama SVM pada ulasan

GoPay dengan metode seleksi fitur *Chi-Square* dan *Information Gain*, di mana SVM unggul sekitar 1% dibandingkan Random Forest.

Berbeda dengan studi sebelumnya, studi ini menggunakan kumpulan data yang jauh lebih besar dan lebih mutakhir 15.089 ulasan aplikasi DANA dari 1 September 2024 sampai 30 November 2024. Selain itu, untuk memberikan penilaian yang lebih menyeluruh, studi ini secara bersamaan membandingkan Random Forest dengan dua algoritma: Naive Bayes dan SVM.

2.1.4 Support Vector Machine (SVM)

Algoritma SVM digunakan oleh Saputro et al. [1] menggunakan dua kelas sentimen positif dan negatif untuk menganalisis sentimen pengguna aplikasi DANA di Google Play Store. Penelitian tersebut menghasilkan akurasi sebesar 80%, dengan *precision* negatif 84,06% dan positif 74,08%, serta *recall* negatif 87,02% dan positif 69,21%. Pratiwi & Yustanti [7] membandingkan SVM dengan Naive Bayes dan KNN pada data Twitter dan menyimpulkan bahwa SVM merupakan algoritma terbaik dengan akurasi 86,23% pada data DANA. Aziz & Yamasari [6] juga menerapkan SVM bersama Random Forest pada ulasan GoPay dan menunjukkan bahwa SVM memiliki performa lebih unggul dibandingkan Random Forest.

Penelitian ini berbeda dari penelitian sebelumnya karena menggunakan tiga kelas sentimen, membandingkan SVM dengan Naive Bayes dan Random Forest dalam konteks aplikasi DANA, dan menggunakan ulasan Google Play Store sebagai platform resmi untuk pengguna Android.

Berdasarkan seluruh perbandingan tersebut, dapat disimpulkan terdapat beberapa *research gap* yang masih perlu dikaji lebih lanjut. Pertama, belum banyak penelitian yang membandingkan Naive Bayes, SVM, dan Random Forest secara bersamaan pada ulasan aplikasi DANA. Kedua, sebagian besar penelitian sebelumnya menggunakan dataset dengan jumlah yang relatif kecil, berkisar antara 1.354 hingga 2.453 data. Ketiga, mayoritas penelitian hanya menggunakan dua kelas sentimen tanpa menyertakan kelas netral, sehingga hasil analisis menjadi kurang mendalam.

Dengan membandingkan tiga algoritma *machine learning* Naive Bayes, SVM, dan *Random Forest* menggunakan dataset berisi 15.089 ulasan aplikasi DANA dari

1 September 2024 sampai 30 November 2024 dengan tiga kelas sentimen positif, negatif, dan netral penelitian ini berupaya untuk menutup kesenjangan tersebut. Hasilnya, diharapkan studi ini akan memberikan kontribusi yang lebih menyeluruh terhadap kemajuan analisis sentimen berbasis *machine learning* untuk aplikasi keuangan digital di Indonesia.

2.2 Tinjauan Teori

2.2.1 Aplikasi DANA

Google Play Store dan App Store menawarkan aplikasi dompet digital Indonesia DANA untuk diunduh. Pengguna aplikasi DANA dapat melakukan pembayaran digital baik online maupun offline tanpa menggunakan uang tunai atau kartu kredit. DANA menyediakan sejumlah fungsi, seperti membeli pulsa telepon, membayar tagihan listrik, dan menambah saldo. Tujuan utamanya adalah memberikan kemudahan kepada konsumen dalam melakukan transaksi yang cepat, mudah, dan aman [1].

2.2.2 Analisis Sentimen

Analisis sentimen memiliki peran krusial dalam memahami perkembangan opini secara cepat di media sosial dan berbagai situs lainnya. Dalam beberapa tahun terakhir, informasi tersebut telah tersebar luas di platform media sosial dan internet. Oleh karena itu, diperlukan analisis sentimen untuk mengolah data tersebut dan mendukung pengambilan keputusan yang tepat [11].

Tujuan analisis sentimen, yang menggabungkan penambangan teks dan penambangan data, adalah untuk mengklasifikasikan berbagai ulasan, opini, atau sentimen yang diungkapkan oleh para ahli dan pelanggan tentang barang, jasa, perusahaan, atau aktivitas tertentu [9]. Prosedur ini, yang juga disebut klasifikasi teks, digunakan untuk mengidentifikasi apakah suatu teks memiliki makna netral, positif, atau bahkan negatif. Melalui proses interpretasi, ekstraksi, dan pengolahan data, analisis sentimen bertujuan untuk mendistribusikan sentimen publik secara otomatis [10].

Dalam penelitian ini menggunakan analisis sentimen dengan basis *machine learning* sehingga membutuhkan penggunaan algoritma untuk mengkategorikan kata-kata ke dalam kategori positif dan negatif. Contohnya termasuk algoritma

seperti SVM, dan *Random Forest*.

2.2.3 Text Mining

Text mining diartikan sebagai proses yang bertujuan guna menyaring informasi yang terkandung didalam kumpulan teks besar dengan cara mengidentifikasikan pola dan hubungan yang signifikan dalam data tekstual secara otomatis. Proses ini melibatkan berbagai disiplin ilmu, seperti data science, pengolahan bahasa alami, pembelajaran mesin, dan pengambilan informasi. Tujuan utama *text mining* adalah untuk melakukan klasterisasi, klasifikasi, pengambilan informasi, dan pencarian informasi dalam teks [12]. Penelitian ini akan mempergunakan salah satu teknik dalam *text mining*, yaitu analisis sentimen. Dalam analisis sentimen, beberapa tahap *text mining* yang akan diterapkan meliputi *text processing* dan *feature selection*.

2.1 *Text Processing*

Pada tahap ini, teks akan diproses agar dapat digunakan dalam analisis sentimen. Proses yang dilakukan meliputi langkah-langkah berikut:

3.1 *To lower case* adalah metode yang merubah keseluruhan huruf menjadi huruf kecil.

3.2 *Tokenizing* yakni proses yang mengubah kalimat utuh menjadi kata-kata individual.

3.3 *Remove number* adalah proses yang menghapus angka yang terdapat dalam teks.

3.4 *Remove u*rl adalah langkah yang menghilangkan URL yang ada dalam teks.

3.5 *Remove punctuation* adalah proses untuk menghapus tanda baca dan pemisah lainnya.

2.2 *Feature Selection*

Proses yang dilakukan meliputi langkah-langkah berikut:

1) *Stopword* merujuk pada kata-kata yang tak berkontribusi secara signifikan pada makna dari suatu dokumen atau teks.
2) *Stemming* yaitu proses untuk merubah kata menjadi model dasar atau akar kata.

Pada penelitian ini mempergunakan Google Play Store untuk mengambil data.

2.2.4 Confusion Matrix

Confusion Matrix yaitu alat yang berguna dalam pengukuran kinerja dalam masalah klasifikasi yang menghasilkan output berupa dua kelas atau lebih. Fungsi utama dari *confusion matrix* adalah untuk menganalisis suatu *classifier*, sehingga dapat diketahui apakah *classifier* tersebut berkinerja baik atau buruk [4]. *Confusion matrix* ini terdiri dari tabel yang digunakan untuk klasifikasi biner, seperti yang ditampilkan didalam tabel berikut ini.

Tabel 2.2 *Confusion Matrix*

Kelas	**Prediksi Positif**	**Prediksi Negatif**	**Prediksi Netral**
Actual Positif	*True Positive (TP)*	*False Negative (FN)*	*False Netral (FNt)*
Actual Negatif	*False Positve 1 (FP1)*	*False Negative 1 (FN1)*	*False Netral 1 (FNt1)*
Actual Netral	*False Positve 2 (FP2)*	*False Negative 2 (FN2)*	*False Netral 2 (FNt2)*

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

TF-IDF yaitu metode yang mengukur frekuensi kemunculan kata pada dokumen atau dataset, serta mengevaluasi seberapa relevan kata tersebut [5]. Metode ini memiliki fungsi untuk menentukan bobot setiap kata, sehingga dapat menilai pentingnya kata tersebut dalam dokumen atau dataset. TF-IDF terdiri dari dua komponen utama, yakni TF (*Term Frequency*), yang menggambarkan proporsi kemunculan kata didalam teks atau dokumen, serta IDF (*Inverse Document Frequency*), yang mengukur apakah kata tersebut jarang muncul di semua dokumen. Teknik pembobotan token ini bermanfaat untuk memantau seberapa sering kata tertentu muncul dalam teks [8]. Sebagai contoh, TF-IDF bisa mengidentifikasikan kata-kata yang seringkali muncul pada sebuah ulasan, seperti "baik", "bagus", "buruk", dan "jelek". Secara umum, penerapan teknik ekstraksi fitur seperti TF-IDF dapat meningkatkan akurasi kinerja algoritma dalam analisis sentimen. Rumus TF-IDF yakni sebagai berikut:

$$TF - IDF = TF \times IDF$$

$$TF = \frac{Frekuensi\ term\ pada\ satu\ dokumen}{Total\ kata\ pada\ satu\ dokumen}$$

$$IDF = \log \frac{Total\ dokumen + 1}{Frekuensi\ dokumen\ mengandung\ term}$$

Dalam penelitian ini, metode TF-IDF dipilih karena mampu memberikan bobot yang lebih besar terhadap kata-kata yang penting dalam dokumen serta mengurangi pengaruh kata-kata yang terlalu sering muncul pada seluruh dokumen. Dibandingkan dengan metode representasi teks sederhana seperti CountVectorizer yang hanya menghitung frekuensi kemunculan kata, TF-IDF mampu menghasilkan representasi teks yang lebih informatif karena mempertimbangkan tingkat kepentingan suatu kata dalam keseluruhan dokumen. Oleh karena itu, TF-IDF banyak digunakan dalam penelitian analisis sentimen dan terbukti mampu meningkatkan performa algoritma klasifikasi teks

Meskipun TF-IDF banyak digunakan dalam analisis sentimen, terdapat beberapa metode representasi teks lain yang juga sering diterapkan dalam penelitian machine learning. Salah satu metode yang paling sederhana adalah Bag of Words (BoW) yang merepresentasikan dokumen berdasarkan frekuensi kemunculan setiap kata tanpa mempertimbangkan tingkat kepentingan kata tersebut. Kelemahan metode ini adalah seluruh kata diperlakukan memiliki bobot yang sama sehingga kata-kata yang sangat sering muncul tetapi kurang informatif tetap memberikan pengaruh yang besar terhadap proses klasifikasi.[1]

Metode lain yang banyak digunakan adalah Word2Vec, yaitu teknik word embedding yang mampu merepresentasikan setiap kata ke dalam bentuk vektor berdimensi rendah dengan mempertimbangkan hubungan semantik antar kata. Dengan demikian, Word2Vec dapat menangkap kemiripan makna antar kata yang tidak dapat dilakukan oleh TF-IDF maupun Bag of Words.[1] Namun, metode ini umumnya memerlukan data pelatihan yang cukup besar agar mampu menghasilkan representasi kata yang optimal. Selain itu, proses pelatihannya relatif lebih kompleks dibandingkan metode pembobotan berbasis frekuensi.

Selanjutnya, FastText merupakan pengembangan dari Word2Vec yang tidak hanya mempelajari kata secara utuh, tetapi juga memanfaatkan karakter atau subword penyusun kata. Pendekatan ini membuat FastText

lebih baik dalam menangani kata yang jarang muncul (out-of-vocabulary atau OOV), termasuk kesalahan penulisan maupun variasi bentuk kata. Akan tetapi, seperti Word2Vec, metode ini membutuhkan proses pelatihan embedding yang lebih kompleks serta sumber daya komputasi yang lebih besar dibandingkan TF-IDF.[2]

Dalam beberapa tahun terakhir, metode berbasis Transformer, seperti BERT (Bidirectional Encoder Representations from Transformers), juga banyak digunakan pada penelitian analisis sentimen karena mampu memahami konteks kalimat secara dua arah sehingga menghasilkan representasi teks yang lebih baik dibandingkan metode representasi teks konvensional. Meskipun demikian, penggunaan BERT memerlukan kebutuhan memori, waktu pelatihan, dan sumber daya komputasi yang jauh lebih besar. Selain itu, proses implementasinya relatif lebih kompleks sehingga lebih sesuai digunakan pada penelitian yang berfokus pada pendekatan deep learning.[3]

Berdasarkan karakteristik tersebut, penelitian ini memilih menggunakan TF-IDF sebagai metode representasi teks karena mampu menghasilkan pembobotan kata yang lebih informatif dibandingkan Bag of Words maupun CountVectorizer, sementara kebutuhan komputasinya jauh lebih ringan dibandingkan metode word embedding seperti Word2Vec, FastText, maupun BERT. Selain itu, penelitian ini bertujuan membandingkan performa algoritma machine learning klasik, yaitu Naïve Bayes, Support Vector Machine (SVM), dan Random Forest. Ketiga algoritma tersebut telah terbukti bekerja dengan baik pada fitur berbasis TF-IDF sehingga penggunaan metode ini dinilai paling sesuai dengan karakteristik dataset ulasan pengguna aplikasi DANA yang berjumlah 15.089 data serta tujuan penelitian untuk memperoleh perbandingan performa algoritma secara objektif dan efisien.

2.3 Framework dan Algoritma

2.3.1 CRISP-DM

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) yaitu model yang dipergunakan dalam proses data mining, yang dirancang untuk membantu menyelesaikan berbagai masalah proyek data mining di berbagai industri. Model ini menyediakan metodologi dan langkah-langkah yang terstruktur

untuk melakukan data mining, memberikan pendekatan yang komprehensif bagi siapa pun yang terlibat dalam proyek data mining [13].

Gambar 2.1 CRISP-DM Diagram

Model ini menggambarkan siklus hidup data mining, atau data *mining lifecycle*, seperti yang terlampir di gambar 2.1 CRISP-DM terdiri atas enam tahapan, di antaranya: *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*.

Tahap pertama dalam CRISP-DM adalah *Business Understanding*, di mana tujuan, objektif, dan target bisnis yang ingin dicapai harus ditentukan. Untuk menentukan sumber daya yang dibutuhkan, evaluasi kondisi bisnis awal juga dilakukan pada tahap ini. Selanjutnya, di tahap *Data Understanding*, dilakukanlah pengumpulan dan eksplorasi data untuk memastikan kualitas data yang dapat digunakan dalam keperluan bisnis, termasuk melakukan *Exploratory Data Analysis* (EDA). Setelah itu, pada tahap *Data Preparation*, data yang telah dieksplorasi akan diproses lebih lanjut melalui langkah-langkah preprocessing, seperti pembersihan dan penyesuaian format agar sesuai dengan analisis yang akan dilaksanakan. Pada tahap *Data Modeling*, data yang telah dipersiapkan akan diterapkan pada model yang dipilih, dan beberapa model dapat dibuat untuk dilakukan perbandingan. Kemudian, pada tahap *Evaluation*, performa model dan data dievaluasi dan dibandingkan untuk memastikan tidak ada tahap yang terlewat dan mendapatkan model yang paling sesuai dengan kebutuhan bisnis. Tahap terakhir, *Deployment*, merupakan implementasi model yang telah dibangun untuk digunakan dalam operasi bisnis secara nyata.

2.3.2 *Naïve Bayes*

Naïve Bayes adalah algoritma klasifikasi berbasis probabilitas yang bergantung pada independensi fitur. Kelebihannya adalah sederhana dan cepat, namun kelemahannya tidak mempertimbangkan hubungan antar kata [6]. Klasifikasi *Naïve Bayes* ialah metode statistika yang berguna dalam melakukan prediksi probabilitas keanggotaan sebuah kelas.

Algoritma klasifikasi *Naïve Bayes* befungsi dalam mengelompokan data kedalam kategori khusus. Kinerja pengklasifikasian dinilai berdasarkan tingkat akurasi prediksi yang dihasilkan. Proses pada algoritma *Naïve Bayes* terdiri dari beberapa langkah, yaitu: 1) Memperhitungkan total jumlah kelas/label. 2) Memperhitungkan jumlah kasus yang mempunyai label yang sama. 3) Mengkalikan nilai dari seluruh variabel kelas. 4) Memperbandingkan hasil yang diperoleh dari seluruh variabel tersebut [3]. Berdasarkan pada teorema Bayes, klasifikasi ini berkemampuan sama dengan *decision tree* dan *neural network*, serta terbukti efektif dalam menghasilkan keakuratan yang tinggi dan waktu pemrosesan yang cepat ketika diterapkan pada database dengan dataset yang besar [14].

Berikut merupakan persamaan dari dasar metode pada theorema Bayes:

$$P(HX) = \frac{P(XH)P(H)}{P(X)}$$

Keterangan:

X : Data dengan *class* yang belum diketahui

H : Hipotesis data X yang termasuk suatu *class* spesifik

P(H|X) : Probabilitas hipotesis H berlandaskan pada kondisi X (posteriori probabilitas)

P(H) : Probabilitas hipotesis H (prior probabilitas)

P(X|H) : Probabilitas X berlandaskan pada kondisi pada hipotesis H P(X) : Probabolitas X

Dalam penelitian ini, algoritma Naïve Bayes diselaraskan dengan *class* yang sesuai terhadap data yang akan diklasifikasikan. Berikut rumus Naïve Bayes yang telah diselaraskan:

$$P(WkCi) = \frac{Nk + 1}{N + vocabulary}$$

Keterangan:

P(Wk): Probabilitas Kemunculan kata Ci: Kategori Kelas

P(Wk|Ci): Probabilitas kemunculan kata (Wk) pada sebuah dokumen dengan kategory kelas (Ci)

nk: Nilai Kemunculan kata pada kategori Ci n: Jumlah seluruh kata pada kategory Ci Vocabulary: Jumlah seluruh kata

Naïve Bayes dipilih karena memiliki proses komputasi yang sederhana, cepat, dan efisien dalam melakukan *klasifikasi* teks. Algoritma ini mampu memberikan hasil yang baik pada data berdimensi tinggi dan sering digunakan sebagai metode dasar dalam analisis sentimen.

Pada penelitian ini, algoritma Naïve Bayes diimplementasikan menggunakan model Multinomial Naïve Bayes (MultinomialNB) dari library scikit-learn. Model ini menggunakan nilai parameter alpha = 1.0 sebagai teknik Laplace smoothing untuk menghindari probabilitas bernilai nol serta fit_prior = True sehingga probabilitas prior dihitung berdasarkan distribusi kelas pada data pelatihan. Pemilihan MultinomialNB dilakukan
karena algoritma ini sesuai untuk klasifikasi teks dengan fitur berbasis TF-IDF.Support Vector Machine (SVM)

Support Vector Machine (SVM) diperkenalkan oleh Vapnik pada tahun 1992 sebagai model klasifikasi yang efektif dalam menangani permasalahan non-linear [10]. SVM yaitu teknik *machine learning* yang terkenal dalam klasifikasi teks, yang menunjukkan kinerja sangat baik di berbagai domain. Teknik ini bekerja dengan mengidentifikasi hyperplane yang memisahkan dua kelas berbeda, sehingga hasil klasifikasi dapat dimaksimalkan. SVM juga berusaha untuk memaksimalkan jarak diantara data yang terdekat dengan *hyperplane* [14]. Proses klasifikasi dilaksanakan dengan pencarian *hyperplane* ataupun garis pembatas (*decision boundary*) yang memisahkan kelas-kela s, di mana SVM menggunakan *support vector* dan margin untuk menentukan nilai hyperplane tersebut [15] Metode ini dapat diandalkan untuk menyelesaikan masalah klasifikasi data dengan pendekatan yang kuat. Permasalahan klasifikasi diselesaikan melalui penyelesaian persamaan Lagrangian, yang termasuk wujud dual dari *Support Vector Machine*, menggunakan *quadratic programming* [16]. SVM pada dasarnya bekerja berdasarkan prinsip

linear classifier, di mana kasus klasifikasi yang dapat dipisahkan secara linier menjadi fokus utamanya. Namun, metode ini sudah diperkembangkan dalam mengatasi permasalahan non-linear melalui penerapan konsep kernel dalam ruangan dengan dimensi tinggi. Pada ruangan tersebut, *hyperplane* dicari untuk mengoptimalkan margin atau jarak antar kelas data. Persamaan yang digunakan dalam metode ini adalah sebagai berikut [14]:

Menurut Santosa, Hyperplane klasifikasi linier dapat dianotasikan:

$$f(X) = w^T . x + b$$

sehingga berdasarkan Vapnik dan Cortes didapatkan persamaan: $[(w^T . xi) + b] \geq 1$ untuk yi = +1

$[(w^T . xi) + b] \leq 1$ untuk yi = -1

Dimana xi = himpunan data *training*, i = 1,2,. n dan yi = label kelas dari xi

Hyperplane paling baik diperoleh dengan menentukan posisi yang berada tepat pada tengah-tengah diantara dua bidang pembatas dari kelas yang berbeda. Proses ini bertujuan untuk mengoptimalkan margin maupun jarak diantara dua kelompok objek yang berasal dari kelas yang berbeda [14].

Support Vector Machine (SVM) dipilih karena memiliki kemampuan yang baik dalam menangani data teks yang memiliki dimensi tinggi. SVM mampu menentukan batas pemisah (hyperplane) yang optimal sehingga sering menghasilkan tingkat akurasi yang tinggi pada kasus klasifikasi teks.

Pada penelitian ini, algoritma Support Vector Machine diimplementasikan menggunakan kernel linear dengan parameter C = 1.0 dan gamma = scale. Kernel linear dipilih karena data hasil ekstraksi TF-IDF memiliki dimensi yang tinggi sehingga lebih efektif dipisahkan menggunakan hyperplane linear. Nilai parameter tersebut merupakan konfigurasi bawaan (default) dari library scikit-learn.

2.3.3 *Random Forest*

Random Forest adalah teknik yang bermanfaat untuk regresi dan klasifikasi. Teknik ini merupakan pendekatan *ensemble* yang membangun dan menggabungkan serangkaian pohon keputusan sebagai *base classifier*. Beberapa faktor penting pada *Random Forest* meliputi *bootstrap sampling* guna menciptakan pohon prediksi, penggunaan prediktor acak oleh setiap pohon keputusan, dan penggabungan hasil

prediksi dengan cara *majority vote* untuk klasifikasi serta rata-rata untuk regresi [17].

Random Forest bisa dibangun mempergunakan teknik *bagging* dengan memilih atributnya secara acak. Proses ini sering menggunakan metode CART (*Classification and Regression Tree*) guna membangun pohon keputusan, yang dibiarkan tumbuh sampai ukurannya maksimal tanpa pemangkasan. Hasilnya adalah sekumpulan pohon yang disebut *forest*. Keunggulan dari metode ini antara lain:

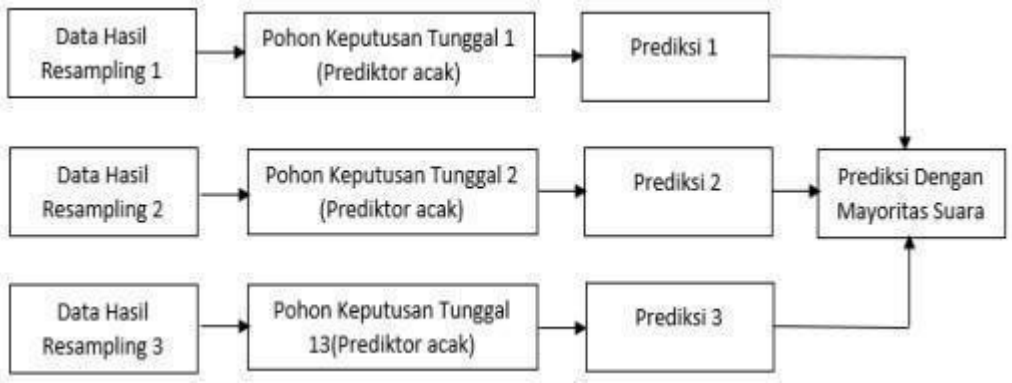
2.1 Memiliki tingkat akurasi yang tinggi.

2.2 Relatif tahan terhadap *outliers* dan *noise*.

2.3 Lebih cepat dibanding dengan metode *bagging* dan *boosting* lainnya.

2.4 Sederhana serta mudah untuk diimplementasikan secara paralel [14].

Proses algoritma sederhana dari *Random Forest* dapat ditinjau pada penjelasan berikutnya.



Gambar 2.2 Algoritma Sederhana *Random Forest*

Gambar 2.2 menggambarkan alur kerja sederhana algoritma Random Forest. Secara umum, proses dimulai dari kumpulan data pelatihan yang kemudian diambil sampelnya secara acak menggunakan teknik bootstrap sampling untuk membangun sejumlah pohon keputusan (decision tree) secara paralel. Setiap pohon keputusan dibangun menggunakan subset fitur yang dipilih secara acak, sehingga menghasilkan prediksi yang beragam. Prediksi dari semua pohon kemudian digabungkan menggunakan mekanisme pemungutan suara mayoritas untuk menghasilkan keputusan klasifikasi akhir. Pendekatan ensemble ini menjadikan Random Forest lebih robust dan akurat dibandingkan pohon keputusan tunggal.

Random Forest dipilih karena merupakan metode ensemble yang mampu

mengurangi risiko overfitting dan memiliki kemampuan yang baik dalam menangani data yang kompleks. Selain itu, Random Forest juga relatif tahan terhadap noise dan mampu menghasilkan performa klasifikasi yang stabil.

Pada penelitian ini, algoritma Random Forest menggunakan 100 decision tree (n_estimators = 100) dengan criterion = gini, max_depth = None, dan random_state = 42. Konfigurasi tersebut digunakan untuk menghasilkan model yang stabil serta mengurangi risiko overfitting. Parameter yang digunakan merupakan parameter bawaan dari library scikit-learn.

2.3.4 Alasan Pemilihan Algoritma

Pemilihan algoritma dalam penelitian analisis sentimen merupakan salah satu faktor penting yang dapat memengaruhi performa klasifikasi. Berbagai algoritma machine learning telah banyak digunakan dalam penelitian analisis sentimen, seperti Decision Tree, K-Nearest Neighbor (KNN), Logistic Regression, Naïve Bayes, Support Vector Machine (SVM), Random Forest, hingga pendekatan deep learning seperti Long Short-Term Memory (LSTM) dan Bidirectional Encoder Representations from Transformers (BERT). Masing-masing algoritma memiliki karakteristik, kelebihan, dan keterbatasan yang berbeda sehingga pemilihannya perlu disesuaikan dengan tujuan penelitian, karakteristik data, serta sumber daya komputasi yang tersedia.

Pada penelitian ini dipilih tiga algoritma, yaitu Naïve Bayes, Support Vector Machine (SVM), dan Random Forest karena ketiganya merupakan algoritma klasifikasi yang paling banyak digunakan dalam penelitian analisis sentimen berbasis machine learning. Selain memiliki karakteristik yang berbeda, ketiga algoritma tersebut juga telah terbukti memberikan performa yang baik pada berbagai penelitian klasifikasi teks sehingga hasil penelitian ini dapat dibandingkan secara lebih objektif dengan penelitian-penelitian sebelumnya.

Naïve Bayes dipilih sebagai algoritma pembanding (baseline classifier) karena memiliki mekanisme klasifikasi berbasis probabilitas yang sederhana, efisien, dan mampu memberikan hasil yang baik pada data teks berdimensi tinggi. Selain memiliki waktu komputasi yang relatif cepat, Naïve Bayes juga banyak digunakan sebagai acuan awal dalam penelitian analisis sentimen sehingga dapat memberikan gambaran dasar mengenai performa klasifikasi terhadap dataset yang

digunakan.

Support Vector Machine (SVM) dipilih karena dikenal sebagai salah satu algoritma yang memiliki performa tinggi pada permasalahan klasifikasi teks. SVM mampu membentuk hyperplane optimal yang memisahkan setiap kelas dengan margin maksimum sehingga efektif digunakan pada data hasil ekstraksi fitur TF-IDF yang memiliki dimensi tinggi. Berbagai penelitian terdahulu juga menunjukkan bahwa SVM sering memperoleh nilai akurasi yang lebih tinggi dibandingkan algoritma klasifikasi konvensional lainnya pada kasus analisis sentimen.

Random Forest dipilih karena merupakan metode ensemble learning yang mampu meningkatkan stabilitas dan akurasi klasifikasi melalui kombinasi banyak decision tree. Pendekatan ini relatif tahan terhadap noise maupun overfitting sehingga sesuai digunakan pada dataset ulasan pengguna yang memiliki variasi kata dan karakteristik data yang cukup kompleks. Selain itu, Random Forest mampu menangani data dalam jumlah besar dengan performa yang tetap baik.

Pemilihan ketiga algoritma tersebut juga bertujuan untuk membandingkan tiga pendekatan machine learning yang memiliki mekanisme klasifikasi yang berbeda. Naïve Bayes mewakili pendekatan klasifikasi berbasis probabilitas, Support Vector Machine mewakili pendekatan berbasis pencarian hyperplane optimal, sedangkan Random Forest mewakili pendekatan ensemble learning berbasis decision tree. Dengan menggunakan ketiga algoritma tersebut pada dataset yang sama, penelitian ini dapat memberikan perbandingan performa yang lebih komprehensif berdasarkan nilai accuracy, precision, recall, dan F1-score sehingga dapat diketahui algoritma yang paling sesuai untuk melakukan analisis sentimen pada ulasan pengguna aplikasi DANA di Google Play Store.

2.4 Tools dan Software

2.4.1 *Transformers*

Transformers adalah sebuah pustaka yang dikembangkan menggunakan bahasa pemrograman Python, dirancang untuk mengunduh model pre-trained. Pustaka ini sering digunakan dalam pengolahan bahasa alami (*Natural Language Processing/NLP*), termasuk untuk tugas seperti klasifikasi teks dalam analisis sentimen. Salah satu contoh model yang digunakan untuk klasifikasi teks adalah *Indonesian-roBERTa*, yang telah dilatih khusus untuk mengkategorikan teks dalam

bahasa Indonesia menjadi sentimen positif atau negatif [18].

2.4.2 *CountVectorizer*

CountVectorizer adalah metode ekstraksi fitur yang berfungsi mengonversi kumpulan teks menjadi matriks. Metode ini mengubah teks menjadi vektor yang menunjukkan seberapa sering setiap kata muncul dalam dataset. Vektor dibuat berdasarkan jumlah kata dalam dataset dan akan bertambah seiring ditemukannya kata-kata baru. CountVectorizer juga mengekstraksi kata-kata dengan panjang minimal dua huruf [19].

2.4.3 Microsoft Excel

Microsoft Excel adalah *software* yang berguna dalam pengolahan data, dilengkapi dengan *spreadsheet* yang tersusun atas baris dan kolom. *Software* ini dapat digunakan untuk berbagai tugas, termasuk modifikasi data, penambahan, dan pengurangan. Selain itu, *Microsoft Excel* menyediakan sejumlah fitur yang mempermudah analisis data yang efisien bagi pengguna [20].

2.4.4 Google *Colaboratory*

Google Colaboratory, atau Google Colab, ialah media berbasis *cloud* yang dikembangkan oleh Google dalam rangka mengoperasikan dan mengembangkan kode Python. Platform ini menggunakan *Jupyter Notebook* sebagai lingkungan interaktif untuk menulis dan mengeksekusi kode. Google Colab menyediakan akses ke sumber daya *cloud* seperti CPU, GPU, dan TPU (*Tensor Processing Units*) dari Google, memungkinkan pengembangan model *machine learning* dilakukan dengan lebih efisien. Selain itu, Google Colab dapat terintegrasi dengan Google Drive, layanan penyimpanan berbasis *cloud*, sehingga mendukung proses *backup* data secara *real-time* [21].

2.4.5 Python

Python diartikan sebagai bahasa pemrograman yang pertama kalinya diperkembangkan oleh Guido van Rossum di awal 1990-an. Bahasa ini termasuk dalam kategori pemrograman tingkat tinggi dan sering dimanfaatkan untuk pengembangan aplikasi, web, *machine learning*, serta kecerdasan buatan. Python menyediakan berbagai *library* dan modul seperti *NumPy*, *Pandas*, *PyMongo*, dan *PyTorch*, yang mempermudah pengguna dalam mengolah data secara efisien.

Selain itu, Python bersifat *open-source*, memungkinkan banyak pengembang dalam komunitas untuk berkontribusi dengan menciptakan *library* dan modul baru yang dapat membantu pengguna lain dalam menganalisis data [22].