

BAB II

LANDASAN TEORI

2.1 Kajian Teori

2.1.1 Analisis Sentimen

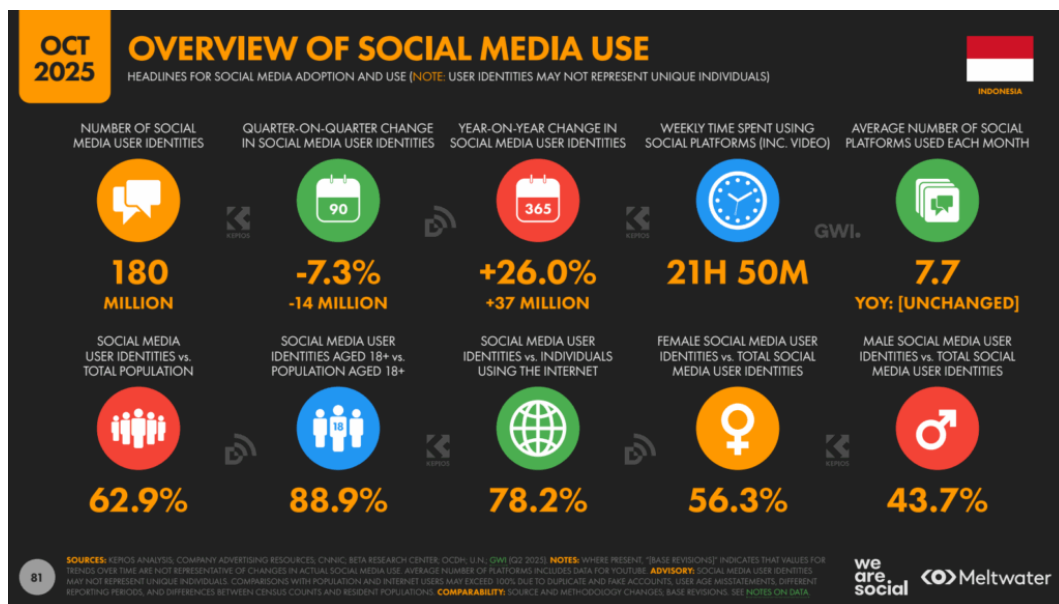
Analisis sentimen merupakan salah satu penerapan *Natural Language Processing* (NLP) yang digunakan untuk mengidentifikasi kecenderungan opini, sentimen, maupun emosi yang terkandung dalam data berbentuk teks. Proses ini bertujuan mengelompokkan suatu dokumen, kalimat, atau pernyataan ke dalam kategori sentimen berdasarkan polaritasnya (Nugraha, 2022). Pada media sosial, analisis sentimen banyak dimanfaatkan untuk menggambarkan kecenderungan tanggapan masyarakat terhadap suatu topik, baik berupa sentimen positif maupun negatif (Wulandari et al., 2021).

Pelaksanaan analisis sentimen umumnya mencakup tahapan *scraping* data, *text preprocessing*, *feature selection*, *classification*, dan *evaluation*. Rangkaian proses tersebut mengubah data teks yang semula tidak terstruktur menjadi informasi yang lebih terorganisasi sehingga dapat dianalisis secara sistematis. Hasilnya dapat dimanfaatkan sebagai bahan evaluasi dalam berbagai bidang, termasuk untuk mengkaji isu publik, pernyataan, atau komentar yang memunculkan beragam tanggapan. Informasi tersebut juga dapat menjadi dasar pertimbangan bagi perusahaan, tokoh publik, maupun pemerintah dalam mendukung proses pengambilan keputusan (Natasuwarna, 2020).

2.1.2 Media Sosial

Media sosial telah menjadi bagian penting dalam kehidupan digital masyarakat Indonesia. Berdasarkan survei *We Are Social*, jumlah penggunaannya pada tahun 2026 mencapai sekitar 180 juta orang atau setara dengan 62,9% dari total populasi. Tingginya angka tersebut menunjukkan bahwa media sosial merupakan platform yang paling banyak diakses untuk berinteraksi dan bertukar informasi dibandingkan jenis situs maupun aplikasi lainnya.

Pesatnya perkembangan teknologi digital turut mendorong peningkatan penggunaan media sosial di berbagai negara, termasuk Indonesia. Sebagai platform berbasis internet, media sosial memungkinkan individu maupun kelompok berkomunikasi tanpa dibatasi ruang dan waktu. Kemudahan akses serta beragam fitur yang tersedia menjadikannya semakin populer, terutama di kalangan generasi Y dan Z (Lestari et al., 2024)



Gambar 2.1 Data Pengguna Media Sosial di Indonesia

Sumber: (Digital 2026: Top digital and social media trends in Indonesia, 2026)

2.1.3 X/Twitter

Media sosial X banyak dimanfaatkan sebagai sumber data dalam penelitian analisis sentimen karena menyediakan beragam opini pengguna dalam bentuk teks yang dapat diakses untuk kebutuhan penelitian. Banyaknya unggahan yang membahas berbagai isu, didukung ketersediaan API serta kemudahan pengguna dalam menyampaikan pendapat, menjadikan platform ini sesuai untuk menganalisis sentimen masyarakat (Achmadi et al., 2024)

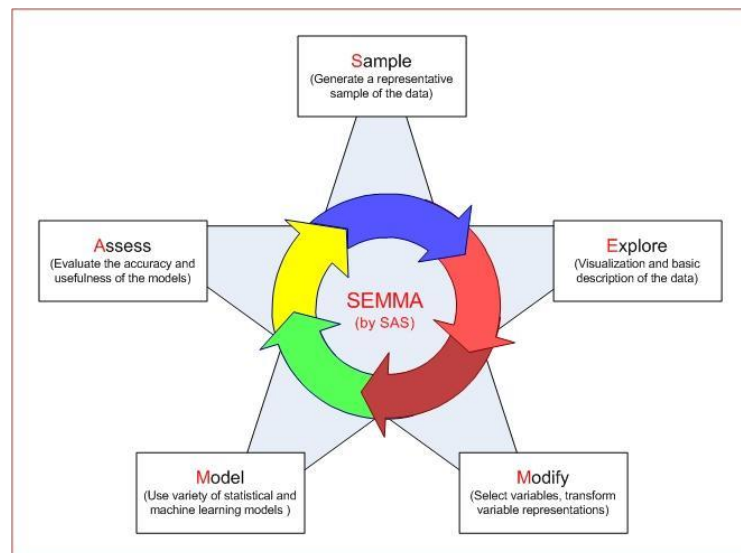
Platform ini sebelumnya dikenal dengan nama Twitter sebelum resmi berganti menjadi X setelah diakuisisi oleh Elon Musk pada Oktober 2022. Melalui unggahan singkat yang disebut *tweet*, pengguna dapat membagikan informasi, berita, maupun opini mengenai berbagai topik.

2.1.4 *Natural Language Processing*

Kemampuan komputer dalam memahami dan mengolah bahasa manusia diwujudkan melalui *Natural Language Processing* (NLP). Bidang ini merupakan salah satu cabang *Artificial Intelligence* (AI) yang berfokus pada pemrosesan bahasa alami yang digunakan manusia dalam komunikasi sehari-hari (Rohman et al., 2019). Melalui NLP, komputer dapat mengenali makna dari teks atau bahasa yang diterima sehingga mampu menghasilkan respons yang sesuai dengan konteks.

2.1.5 SEMMA

Dalam proses *data mining*, metodologi SEMMA banyak digunakan sebagai kerangka kerja untuk mengelola tahapan analisis data secara sistematis. Metodologi yang dikembangkan oleh SAS Institute ini mendukung proses eksplorasi hingga penemuan pola dari data sehingga mampu menghasilkan *Model* yang akurat dan mudah diinterpretasikan (Cerrito & Institute, 2006). Fokus utama SEMMA terletak pada pengolahan dan eksplorasi data secara menyeluruh sebelum *Model* dibangun. Tahapan yang terdapat dalam metodologi SEMMA terdiri atas lima proses sebagaimana ditunjukkan pada Gambar 2.2 berikut.



Gambar 2. 2 Tahapan Model SEMMA

Sumber: (SAS Institute, 2014)

Metodologi data mining ini dibagi menjadi lima tahapan terstruktur (SAS Institute, 2014).

a. *Sample*

Mengekstraksi atau memilih subset data yang representatif dari *dataset* berskala besar yang cukup untuk membentuk informasi signifikan, namun tidak terlalu besar agar proses lebih cepat

b. *Explore*

Menganalisis data secara visual maupun statistik untuk menemukan tren yang tidak terduga, anomali, serta memahami hubungan antar elemen data.

c. *Modify*

Menyiapkan data melalui proses *cleaning*, seleksi, dan transformasi agar sesuai dengan kebutuhan pada tahap pembentukan *Model*.

d. *Model*

Menerapkan algoritma dan teknik statistik secara otomatis untuk mencari kombinasi data terbaik dalam memprediksi hasil akhir yang diinginkan

e. *Assess* (Evaluasi)

Mengevaluasi kegunaan, keandalan, dan tingkat akurasi dari *Model* yang telah dibuat sebelumnya

2.1.6 *Hyperparameter tuning (Grid Search)*

Konfigurasi *hyperparameter* berperan penting dalam menentukan performa *Model Machine learning* karena nilainya ditetapkan sebelum proses pelatihan dimulai. Berbeda dengan parameter *Model* yang diperbarui secara otomatis selama proses pembelajaran, *hyperparameter* harus ditentukan secara manual oleh peneliti. Pemilihan konfigurasi yang kurang tepat dapat menyebabkan *Model* mengalami *underfitting* maupun *overfitting*, sehingga diperlukan proses *hyperparameter tuning* untuk memperoleh konfigurasi yang memberikan kemampuan generalisasi dan performa terbaik pada data uji (Yang & Shami, 2020).

2.1.7 Hyperparameter tuning pada Multinomial Naïve Bayes

Pada algoritma *Multinomial Naïve Bayes*, proses *hyperparameter tuning* difokuskan pada optimasi parameter *alpha* (α) sebagai nilai *smoothing*. Optimasi ini diperlukan untuk mengatasi *zero probability problem*, yaitu kondisi ketika suatu kata pada data uji tidak pernah muncul selama proses pelatihan sehingga probabilitas posterior suatu kelas menjadi nol (Taunk et al., 2019). Dengan menerapkan *smoothing*, *Model* tetap dapat memberikan nilai probabilitas pada kata-kata yang belum ditemukan di data latih sehingga hasil klasifikasi menjadi lebih stabil (Khaira et al., 2023).

Probabilitas pada *Multinomial Naïve Bayes* dengan penambahan parameter *alpha* (α) dinyatakan pada persamaan 2.1 berikut (Khaira et al., 2023).

$$P(w_i|c) = \frac{N_{ci} + \alpha}{N_c + \alpha \cdot |V|} \dots\dots\dots (2.1)$$

Keterangan:

$P(w_i|c)$: Probabilitas kemunculan kata w_i pada kelas c .

N_{ci} : Jumlah kemunculan kata w_i pada kelas c .

N_c : Jumlah total seluruh kata pada kelas c .

α : Parameter *smoothing* (*alpha*).

$|V|$: Jumlah kosakata unik (*vocabulary*) di dalam seluruh *dataset*.

Jika nilai $\alpha = 1.0$, maka metode tersebut dinamakan *Laplace Smoothing*, sedangkan jika nilai $\alpha < 1.0$, metode tersebut dinamakan *Lidstone Smoothing*. Di dalam penelitian ini, pencarian nilai α terbaik dievaluasi menggunakan *5-fold cross-validation* pada *range* nilai tertentu untuk meningkatkan nilai presisi dan *Recall Model* biner (Yang & Shami, 2020).

2.1.8 Hyperparameter tuning pada Support Vector Machine (SVM)

Penentuan nilai parameter C merupakan salah satu faktor yang memengaruhi performa *Support Vector Machine* (SVM), terutama pada penggunaan *kernel* linear. Parameter regularisasi ini mengendalikan keseimbangan antara pembentukan *margin* yang optimal dan tingkat kesalahan klasifikasi pada

data latih. Nilai C yang kurang sesuai dapat menurunkan kemampuan *Model* dalam melakukan generalisasi terhadap data baru, sehingga optimasi *hyperparameter* diperlukan untuk memperoleh konfigurasi yang paling tepat (Wirasasmita & Anisa, 2023). Secara matematis, parameter C diintegrasikan ke dalam fungsi minimisasi *Support Vector Machine* (SVM) sebagai penalti dari variabel *slack* (ξ_i) melalui persamaan berikut (Wirasasmita & Anisa, 2023):

$$\min_{w,b,\xi} \frac{1}{2} ||w||^2 + C \sum_{i=1}^n \xi_i \dots\dots\dots (2.2)$$

Dengan syarat batas:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \dots\dots\dots (2.3)$$

Keterangan:

w : Vektor bobot (*weight vector*) yang tegak lurus terhadap *hyperplane*

b : *Bias*

ξ_i : Variabel *slack* yang mengukur tingkat kesalahan posisi titik data x_i .

C : Parameter regularisasi (*Hyperparameter C*)

y_i : Label aktual kelas numerik sentimen positif atau negatif (1 atau 0)

Nilai C yang kecil (berkisar antara 0,01 hingga 0,10) memberikan penalti yang longgar terhadap kesalahan, menghasilkan *margin* keputusan yang lebih lebar (*soft margin*), tetapi berisiko meningkatkan *underfitting*. Sebaliknya, nilai C yang besar (berkisar antara 1,00 hingga 20,00) memberikan penalti yang ketat terhadap kesalahan latih (*hard margin*), memaksa *Model* untuk mengklasifikasikan seluruh data latih secara benar, namun memiliki risiko tinggi menyebabkan *overfitting*. Oleh karena itu, *GridSearchCV* diterapkan untuk mendapatkan nilai konfigurasi parameter C yang paling stabil dan adaptif pada matriks TF-IDF unigram dan bigram data teks media sosial X (Wirasasmita & Anisa, 2023)

2.1.9 *Naïve Bayes*

Dalam analisis sentimen, *Naïve Bayes* merupakan salah satu algoritma klasifikasi yang banyak diterapkan untuk mengelompokkan data teks berdasarkan probabilitas kemunculan setiap kelas. Algoritma ini mengasumsikan bahwa setiap fitur atau kata pada dokumen saling independen sehingga proses perhitungan probabilitas dapat dilakukan secara sederhana namun tetap memberikan hasil yang baik, khususnya pada *dataset* berukuran besar (Khaira et al., 2023). Efisiensi komputasi yang dimiliki menjadikan *Naïve Bayes* banyak dimanfaatkan pada berbagai tugas klasifikasi teks.

Penerapan *Naïve Bayes* dalam analisis sentimen umumnya diawali dengan representasi dokumen menggunakan metode pembobotan *Term frequency–Inverse document frequency* (TF-IDF). Hasil pembobotan tersebut selanjutnya digunakan sebagai masukan bagi *Model* untuk mempelajari pola dari data latih, kemudian mengklasifikasikan data uji atau data baru yang belum memiliki label sentimen, seperti opini masyarakat maupun ulasan pengguna.

Secara matematis, algoritma *Naïve Bayes* dibangun berdasarkan Teorema Bayes yang diperkenalkan oleh Thomas Bayes pada tahun 1763 dan selanjutnya disempurnakan oleh Pierre-Simon Laplace. Teorema ini digunakan untuk menghitung probabilitas suatu hipotesis berdasarkan bukti yang telah diketahui sebelumnya. Persamaan Teorema Bayes dinyatakan sebagai berikut (Khaira et al., 2023).

$$P(A | B) = \frac{P(B|A) \times P(A)}{P(B)} \dots\dots\dots (2.4)$$

dengan syarat:

$$P(B) \neq 0 \dots\dots\dots (2.5)$$

Keterangan:

$P(A | B)$ = probabilitas posterior, yaitu probabilitas kejadian *A* setelah diketahui bahwa kejadian *B* telah terjadi.

$P(B | A)$ = probabilitas likelihood, yaitu probabilitas munculnya bukti B jika diketahui kejadian A benar.

$P(A)$ = probabilitas prior, yaitu probabilitas awal terjadinya kejadian A sebelum mempertimbangkan bukti.

$P(B)$ = probabilitas evidence, yaitu probabilitas terjadinya bukti B .

Pada penerapan *Naïve Bayes* untuk klasifikasi teks, probabilitas digunakan untuk menentukan kemungkinan suatu dokumen termasuk ke dalam kelas sentimen tertentu. Dalam hal ini, kelas sentimen seperti positif atau negatif direpresentasikan sebagai hipotesis yang akan diprediksi, sedangkan informasi pendukung berupa kumpulan kata dalam sebuah *tweet* menjadi dasar perhitungan probabilitas. Berdasarkan prinsip Teorema Bayes, setiap kelas akan dihitung nilai probabilitasnya terhadap dokumen yang dianalisis, kemudian kelas dengan nilai probabilitas posterior paling tinggi dipilih sebagai hasil akhir proses klasifikasi.

2.1.10 *Multinomial Naïve Bayes*

Multinomial Naïve Bayes merupakan salah satu varian dari algoritma *Naïve Bayes* yang dirancang untuk menangani data berbentuk diskrit, terutama data teks yang direpresentasikan berdasarkan frekuensi kemunculan kata dalam dokumen. Karakteristik tersebut menjadikan algoritma ini banyak digunakan pada tugas klasifikasi dokumen dan analisis sentimen, khususnya ketika data telah melalui proses ekstraksi fitur menggunakan metode seperti *Term frequency–Inverse document frequency* (TF-IDF) atau *bag-of-words*.

Dalam proses klasifikasi, *Multinomial Naïve Bayes* menerapkan asumsi *conditional independence*, yaitu setiap kata pada dokumen dianggap tidak saling bergantung dalam proses perhitungan probabilitas. Meskipun asumsi tersebut memiliki keterbatasan dalam merepresentasikan hubungan antar kata pada bahasa alami, algoritma ini tetap mampu memberikan performa yang baik dengan kompleksitas komputasi yang rendah. Probabilitas suatu dokumen (d) termasuk ke dalam kelas (c) dihitung menggunakan persamaan berikut (Khaira et al., 2023).

$$P(c | d) \propto P(c) \prod_{i=1}^n P(w_i | c) \dots\dots\dots (2.6)$$

dengan:

$P(c)$ = probabilitas awal (*prior probability*) dari suatu kelas.

$P(w_i | c)$ = probabilitas kemunculan kata w_i pada kelas c .

n = jumlah kata pada dokumen.

Berdasarkan hasil perhitungan probabilitas tersebut, dokumen akan diklasifikasikan ke dalam kelas yang memiliki nilai probabilitas posterior paling tinggi sebagai hasil prediksi sentimen.

2.1.11 Support Vector Machine

Pada bidang *Machine learning*, *Support Vector Machine* (SVM) merupakan algoritma *supervised learning* yang bekerja dengan memanfaatkan data berlabel sebagai dasar proses pembelajaran. Algoritma ini dikembangkan oleh Boser, Guyon, dan Vapnik pada tahun 1992 sebagai metode yang dapat diterapkan untuk permasalahan klasifikasi maupun regresi. Kemampuan SVM dalam membangun batas keputusan yang optimal menjadikannya salah satu metode yang banyak digunakan pada berbagai tugas prediksi (Hermawan et al., 2023).

Dalam analisis sentimen, SVM dimanfaatkan untuk memisahkan data teks ke dalam kelas tertentu, seperti sentimen positif dan negatif. Sebelum proses klasifikasi dilakukan, data teks terlebih dahulu direpresentasikan dalam bentuk vektor fitur. Selanjutnya, *Model* mempelajari pola dari data latih untuk menentukan kelas pada data yang belum diketahui.

Prinsip utama SVM adalah mencari *hyperplane* terbaik yang mampu memisahkan dua kelas data dengan jarak pemisah (*margin*) maksimum. *Hyperplane* tersebut ditentukan berdasarkan posisi data terdekat dari masing-masing kelas yang disebut sebagai *Support Vector*. Secara matematis, persamaan *hyperplane* dapat dituliskan pada persamaan 2.7 berikut (Hermawan et al., 2023):

$$w \cdot x + b = 0 \dots\dots\dots (2.7)$$

Keterangan:

w : bobot (*weight*)

x : data input

b : bias

Konsep utama yang digunakan dalam SVM meliputi:

- a. *Hyperplane*, yaitu bidang pemisah yang digunakan untuk membedakan data antar kelas.
- b. *Margin*, yaitu jarak antara *hyperplane* dengan titik data terdekat pada setiap kelas.
- c. *Support Vector*, yaitu titik data yang memiliki posisi paling dekat dengan *hyperplane* dan menjadi penentu lokasi batas pemisah.

Salah satu keunggulan SVM adalah kemampuannya dalam menangani data berdimensi tinggi, termasuk data berbentuk teks. Dengan memaksimalkan nilai *margin*, algoritma ini mampu menghasilkan batas klasifikasi yang lebih optimal.

2.1.12 Google Colaboratory

Dalam proses pengolahan data dan pengembangan *Model Machine learning*, *Google Colaboratory (Colab)* banyak digunakan sebagai lingkungan pemrograman berbasis *cloud* yang memungkinkan pengguna menjalankan kode *Python* melalui peramban tanpa perlu melakukan konfigurasi sistem secara kompleks. Platform yang dikembangkan oleh Google Research ini mendukung berbagai kebutuhan, seperti analisis data, pengembangan *Machine learning*, dan *deep learning*.

Kemudahan akses menjadi salah satu keunggulan utama Colab karena menyediakan fasilitas komputasi seperti *General Processing Unit (GPU)* secara gratis serta mendukung penyimpanan dan berbagi dokumen secara daring. Melalui *Colab Notebook*, pengguna dapat mengintegrasikan kode program dengan elemen pendukung lain seperti teks, gambar, HTML, dan LaTeX dalam satu dokumen yang sama. Fitur tersebut memungkinkan proses pengembangan maupun kolaborasi kode dilakukan secara lebih fleksibel oleh beberapa pengguna (Rohim & Zuliarso, 2022).

2.1.13 TF-IDF

Sebelum proses klasifikasi dilakukan, data teks perlu diubah ke dalam bentuk numerik agar dapat diproses oleh algoritma *Machine learning*. Salah satu metode yang umum digunakan untuk merepresentasikan karakteristik kata dalam dokumen adalah *Term frequency–Inverse document frequency* (TF-IDF). Metode ini memberikan bobot pada setiap kata berdasarkan tingkat kemunculannya dalam suatu dokumen serta tingkat penyebarannya pada seluruh kumpulan dokumen (Manguri et al., 2020).

Pembobotan TF-IDF mempertimbangkan dua aspek utama, yaitu frekuensi kemunculan kata dalam dokumen (*Term frequency*) dan tingkat keunikan kata dalam keseluruhan *dataset* (*Inverse document frequency*) (Muzakki et al., 2019).

Kata yang sering muncul pada suatu dokumen akan memiliki nilai TF yang lebih tinggi, sedangkan kata yang jarang ditemukan pada dokumen lain akan memperoleh nilai IDF yang lebih besar karena dianggap memiliki informasi yang lebih spesifik.

1. *Term frequency* (TF)

Komponen *Term frequency* digunakan untuk mengetahui seberapa sering suatu kata muncul pada dokumen tertentu. Nilai TF dihitung menggunakan persamaan 2.8 berikut (Muzakki et al., 2019):

$$TF(t, d) = \frac{f(t, d)}{\sum f(t, d)} \dots \dots \dots (2.8)$$

Keterangan:

$f(t, d)$: jumlah kemunculan kata t pada dokumen d

$\sum f(t, d)$: jumlah keseluruhan kata yang terdapat dalam dokumen.

Semakin tinggi frekuensi kemunculan suatu kata dalam dokumen, semakin besar nilai TF yang diperoleh.

2. *Inverse document frequency* (IDF)

Komponen *Inverse document frequency* digunakan untuk mengukur tingkat kepentingan suatu kata berdasarkan jumlah dokumen yang memuat kata tersebut

dalam keseluruhan *dataset*. Persamaan IDF dapat dituliskan sebagai berikut (Muzakki et al., 2019).

$$IDF(t) = \log \frac{N}{df(t)} \dots \dots \dots (2.9)$$

Keterangan:

N : jumlah seluruh dokumen

$df(t)$: jumlah dokumen yang mengandung kata t

Nilai IDF akan semakin besar apabila suatu kata hanya muncul pada sedikit dokumen, karena kata tersebut dianggap memiliki kemampuan yang lebih baik dalam membedakan antar dokumen.

3. TF-IDF

Bobot akhir suatu kata diperoleh melalui hasil perkalian antara nilai TF dan IDF. Persamaan TF-IDF dinyatakan sebagai berikut (Muzakki et al., 2019).

$$TF-IDF(t, d) = TF(t, d) \cdot IDF(t) \dots \dots \dots (2.10)$$

Hasil pembobotan TF-IDF berupa representasi numerik dari data teks yang selanjutnya dapat digunakan sebagai masukan (*input*) pada algoritma klasifikasi.

2.1.14 Python

Dalam bidang pengolahan data dan pengembangan *Model Machine learning*, *Python* banyak dimanfaatkan karena menyediakan lingkungan pemrograman yang mendukung proses analisis secara fleksibel. Bahasa pemrograman ini memiliki sifat interpretatif serta mendukung berbagai paradigma, seperti pemrograman prosedural, berorientasi objek, dan fungsional. Struktur sintaks *Python* yang sederhana membuat proses penulisan maupun pengembangan program menjadi lebih mudah dilakukan.

Selain kemudahan dalam penggunaannya, *Python* memiliki ekosistem pustaka yang luas untuk mendukung berbagai kebutuhan komputasi. Ketersediaan berbagai *library* menjadikan *Python* banyak diterapkan pada pengembangan

perangkat lunak, analisis data, komputasi ilmiah, kecerdasan buatan, hingga pengembangan *Model Machine learning* (Virtanen et al., 2020).



Gambar 2. 3 Logo *Python*

Sumber: (Www.Python.org, 2008)

2.1.15 *Confusion matrix*

Evaluasi terhadap *Model* klasifikasi diperlukan untuk mengetahui sejauh mana *Model* mampu mengenali pola pada data yang diberikan. Salah satu metode evaluasi yang digunakan adalah *Confusion matrix*, yaitu representasi berbentuk matriks yang menunjukkan perbandingan antara hasil prediksi *Model* dengan label aktual pada data uji. Melalui matriks ini, jumlah prediksi yang benar maupun kesalahan klasifikasi pada setiap kelas dapat diketahui (Normawati & Prayogi, 2021).

Selain memberikan gambaran mengenai hasil klasifikasi, *Confusion matrix* juga menjadi dasar dalam perhitungan beberapa metrik evaluasi, seperti *Accuracy*, *Precision*, *Recall*, dan *F1-score* yang digunakan untuk mengukur performa *Model* (Julianto et al., 2022). Bentuk *Confusion matrix* untuk klasifikasi biner ditampilkan pada Tabel 2.1 berikut.

Tabel 2.1 *Confusion matrix*

Nilai Prediksi	Nilai Aktual	
	Positif	Negatif
Positif	TP	FP
Negatif	FN	TN

Sumber: (Julianto et al., 2022)

Tabel 2.1 merupakan tabel *Confusion matrix* dengan keterangan sebagai berikut:

- 1) TP (*True Positive*) = menunjukkan jumlah data dengan label aktual positif yang berhasil diklasifikasikan *Model* sebagai positif.
- 2) TN (*True Negative*) = menunjukkan jumlah data dengan label aktual negatif yang berhasil dikenali *Model* sebagai negatif.
- 3) FP (*False Positive*) = menunjukkan jumlah data negatif yang salah diklasifikasikan sebagai positif.
- 4) FN (*False Negative*) = menunjukkan jumlah data positif yang salah diklasifikasikan sebagai negatif.

Nilai dari setiap komponen *Confusion matrix* kemudian digunakan untuk menghitung metrik evaluasi *Model* sebagai berikut (Normawati & Prayogi, 2021).

1. *Accuracy*

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots (2.11)$$

2. *Precision*

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (2.12)$$

3. *Recall*

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (2.13)$$

4. *F1-score*

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \dots \dots \dots (2.14)$$

Keempat metrik tersebut digunakan sebagai dasar evaluasi untuk membandingkan kemampuan algoritma klasifikasi dalam mengenali sentimen pada data penelitian sehingga dapat ditentukan *Model* dengan performa terbaik.

2.1.16 *Streamlit*

Dalam penelitian berbasis *data science* dan *Machine learning*, hasil pemodelan tidak hanya perlu dievaluasi, tetapi juga dapat disajikan melalui media yang mudah digunakan. Salah satu *framework* yang mendukung proses tersebut adalah *Streamlit*, yaitu *framework* berbasis *Python* yang digunakan untuk membangun aplikasi web interaktif secara praktis tanpa memerlukan pengembangan antarmuka yang kompleks (Azhar et al., 2024).

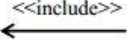
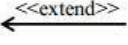
Streamlit menyediakan berbagai komponen yang dapat digunakan untuk menampilkan hasil pengolahan data, seperti tabel, grafik, visualisasi, serta elemen interaktif lainnya. Fasilitas tersebut memungkinkan hasil analisis maupun implementasi *Model* dapat disajikan dalam bentuk yang lebih terstruktur dan mudah dipahami oleh pengguna. Karena kemudahan dalam proses pengembangan dan integrasinya dengan ekosistem *Python*, *Streamlit* banyak digunakan untuk mendistribusikan hasil analisis data dan *Model Machine learning* ke dalam bentuk aplikasi berbasis web.

2.1.17 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan standar pemodelan berbasis grafis yang digunakan untuk melakukan visualisasi, spesifikasi, perancangan, dan dokumentasi terhadap suatu sistem perangkat lunak maupun arsitektur sistem (Hendini, 2016). Melalui pemodelan UML, hubungan antar elemen dalam suatu sistem dapat direpresentasikan secara lebih terstruktur sehingga memudahkan proses pemahaman terhadap rancangan yang dikembangkan.

2.1.18 Use Case Diagram

Use Case Diagram merupakan diagram yang digunakan untuk menggambarkan hubungan interaksi antara aktor, baik pengguna maupun sistem eksternal, dengan sistem yang dikembangkan. Diagram ini menampilkan fungsi utama sistem berdasarkan sudut pandang perilaku aktor (*behavioral*) terhadap sistem (Hendini, 2016). *Use Case Diagram* digunakan sebagai media untuk menjelaskan rancangan sistem kepada pengguna serta membantu mengidentifikasi dan merancang seluruh fitur yang akan tersedia pada sistem yang dikembangkan (Hasanah & Untari, 2020).

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Gambar 2. 4 Komponen Usecase Diagram

Sumber: (Guntoro, 2020)

2.1.19 Activity Diagram

Activity Diagram merupakan diagram yang digunakan untuk menggambarkan alur aktivitas atau proses kerja dalam sistem, baik yang berlangsung secara berurutan maupun bersamaan. Diagram ini dapat menunjukkan proses pengambilan keputusan (*decision*) serta tahapan yang dilakukan secara berulang (*iteratif*) dalam suatu proses komputasi (Hendini, 2016).

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

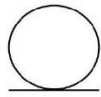
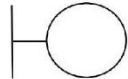
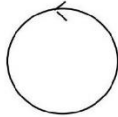
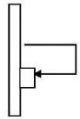
Gambar 2. 5 Komponen Activity Diagram

Sumber: (Esa Unggul, n.d.)

2.1.20 Sequence Diagram

Sequence Diagram merupakan diagram perilaku yang menggambarkan proses interaksi antarobjek dalam sistem melalui pertukaran pesan berdasarkan urutan waktu tertentu (Hendini, 2016). Diagram ini menunjukkan *lifeline* setiap

objek serta aliran pesan atau data yang terjadi antarobjek dalam menjalankan suatu fungsi tertentu (Mubarak et al., 2019).

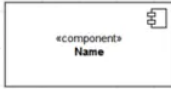
Gambar	Nama	Keterangan
	Entity Class	Gambaran sistem sebagai landasan dalam menyusun basis data
	Boundary Class	Menangani komunikasi antar lingkungan sistem
	Control Class	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
	Recursive	Pesan untuk dirinya
	Activation	Mewakili proses durasi aktivasi sebuah operasi
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek

Gambar 2. 6 Komponen Sequence Diagram

Sumber: (Isnanto, 2023)

2.1.21 Component Diagram

Component Diagram merupakan diagram yang digunakan untuk menggambarkan susunan fisik, struktur, serta hubungan ketergantungan antarkomponen atau modul perangkat lunak dalam sebuah arsitektur sistem (Mubarak et al., 2019). Diagram ini memberikan gambaran mengenai keterkaitan antara modul program, *library*, maupun antarmuka pemrograman aplikasi (*API*) yang saling terhubung dalam membangun suatu sistem secara keseluruhan (Hendini, 2016).

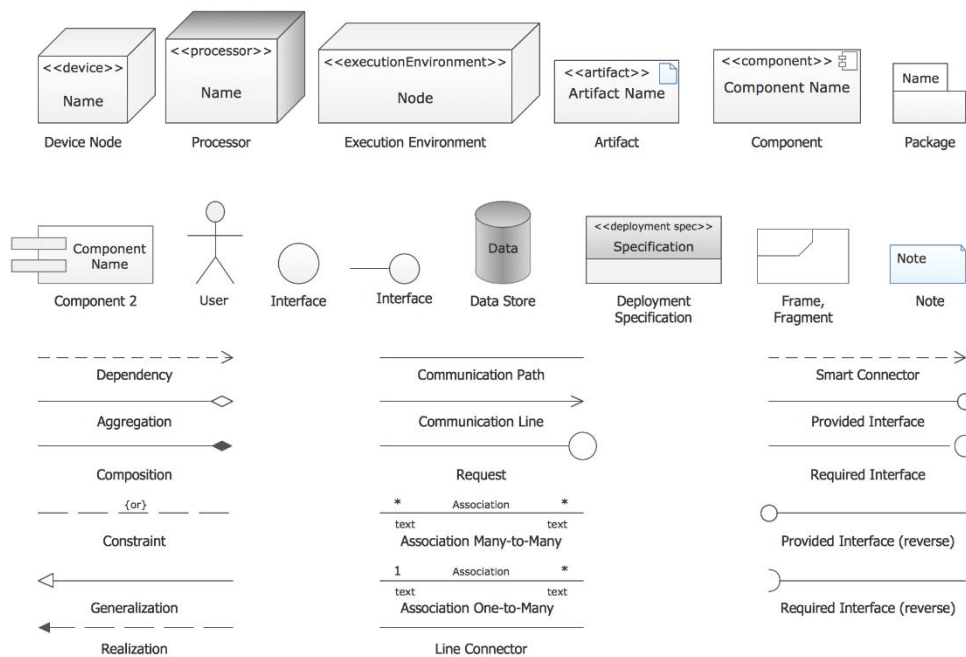
Element	Symbol/Notation	Explanation
Component		Symbol for modules in a system (interaction and communication occur via interfaces).
Package		A package combines multiple elements in a system (e.g. classes, components, interfaces) into a group.
Artifact		Artifacts are physical units of information (e.g. source code, .exe files, scripts, documents) that are created or required during a system's development process or at runtime.
Provided interface		Symbol for one or more clearly defined interfaces that provide functions, services or data to the outside (the open semi-circle is also called a socket).
Required interface		Symbol for a required interface that receives functions, services or data from the outside (the circle-with-stick notation is also referred to as a lollipop).
Port		This symbol indicates a separate point of interaction between a component and its environment.
Relationship		Lines act as connectors and show relationships between components.
Dependency		This special connector indicates a dependency between two parts of a system (not always explicitly shown).

Gambar 2. 7 Component Diagram Symbols

Sumber: (Ionos, 2020)

2.1.22 Deployment Diagram

Deployment Diagram merupakan diagram yang digunakan untuk meModelkan struktur arsitektur fisik suatu sistem dengan menunjukkan penempatan komponen perangkat lunak pada perangkat keras (*node*) yang menjalankannya (Mubarak et al., 2019). Diagram ini menggambarkan konfigurasi infrastruktur fisik serta hubungan komunikasi antarperangkat dalam suatu sistem (Hendini, 2016). Pada penelitian ini, *Deployment Diagram* dimanfaatkan untuk menjelaskan pembagian arsitektur lingkungan penelitian yang terdiri atas lingkungan lokal (*Local Machine*) dan lingkungan berbasis awan (*Cloud service*).



Gambar 2. 8 Deployment Diagram Symbols

Sumber: (ConceptDraw, 2026)

2.1.23 Perbandingan Metodologi Data Mining

Pemilihan metodologi menjadi salah satu aspek penting dalam penelitian berbasis *data mining* karena setiap pendekatan memiliki tahapan dan karakteristik yang berbeda dalam mengelola data hingga menghasilkan informasi yang dapat digunakan. Beberapa metodologi yang sering diterapkan dalam proses pengolahan data antara lain *Sample, Explore, Modify, Model, Assess* (SEMMA), *Knowledge Discovery in Databases* (KDD), dan *Cross-Industry Standard Process for Data Mining* (CRISP-DM) (Wirasasmita & Anisa, 2023).

Ketiga metodologi tersebut memiliki pendekatan yang berbeda dalam mengatur proses analisis data, mulai dari tahap persiapan data, pembentukan *Model*, hingga evaluasi hasil. Perbedaan karakteristik setiap metodologi menjadi pertimbangan dalam menentukan pendekatan yang paling sesuai dengan kebutuhan penelitian (Normawati & Prayogi, 2021).

2.1.24 Knowledge Discovery in Databases (KDD)

Knowledge Discovery in Databases (KDD) merupakan pendekatan dalam *data mining* yang digunakan untuk memperoleh pengetahuan atau pola bermakna dari kumpulan data melalui serangkaian proses yang sistematis (Hendini, 2016). Tahapan dalam KDD meliputi pemilihan data (*Selection*), pembersihan data (*Preprocessing*), transformasi data (*Transformation*), proses penemuan pola melalui *Data Mining*, serta tahap interpretasi dan evaluasi hasil (*Interpretation/Evaluation*) (Normawati & Prayogi, 2021).

Dalam konteks penelitian analisis sentimen, KDD memiliki kesesuaian karena sama-sama digunakan untuk mengolah data teks berbahasa Indonesia hingga menghasilkan informasi berupa pola atau klasifikasi. Namun, tahapan KDD memiliki struktur proses yang memisahkan *preprocessing* dan *transformation* secara lebih eksplisit. Pemisahan tersebut kurang sesuai dengan karakteristik penelitian ini yang membutuhkan fleksibilitas dalam pengolahan teks media sosial X, karena proses pembersihan data, penyesuaian fitur, hingga pembentukan representasi TF-IDF dilakukan dalam satu rangkaian pemrosesan menggunakan *Google Colaboratory*.

2.1.25 Cross-Industry Standard Process for Data Mining (CRISP-DM)

Cross-Industry Standard Process for Data Mining (CRISP-DM) merupakan salah satu kerangka kerja yang banyak digunakan dalam pengelolaan proyek *data mining* karena menyediakan alur proses yang sistematis mulai dari pemahaman tujuan hingga penerapan hasil analisis (Wirasasmita & Anisa, 2023). Metodologi ini terdiri atas enam tahapan utama, yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment* (Mubarak et al., 2019).

Salah satu karakteristik utama CRISP-DM adalah penempatan tahap *Business Understanding* sebagai dasar dalam menentukan arah proses *data mining*. Pendekatan tersebut menjadikan CRISP-DM lebih sesuai untuk penelitian atau proyek yang memiliki tujuan bisnis dan kebutuhan organisasi yang jelas (Wirasasmita & Anisa, 2023). Sementara itu, penelitian analisis sentimen Program

Makan Bergizi Gratis (MBG) lebih menitikberatkan pada pengolahan opini masyarakat serta evaluasi perbandingan performa algoritma klasifikasi *Naïve Bayes* dan *Support Vector Machine* (SVM). Oleh karena itu, CRISP-DM dinilai kurang sesuai dibandingkan SEMMA karena fokus penelitian tidak berorientasi pada pengambilan keputusan bisnis.

Selain itu, tahap *Deployment* pada CRISP-DM yang umumnya diarahkan pada implementasi hasil analisis dalam lingkungan organisasi atau industri, pada penelitian ini diterapkan dalam bentuk aplikasi berbasis *Streamlit* sebagai media pendukung untuk menampilkan hasil klasifikasi dan pengujian *Model*.

Berdasarkan uraian tersebut, metodologi SEMMA dipilih karena lebih sesuai dengan karakteristik penelitian yang berfokus pada pengolahan data teks dan evaluasi performa *Model* klasifikasi dibandingkan metodologi lain yang lebih berorientasi pada proses bisnis.

2.2 Penelitian Terdahulu

Kajian mengenai analisis sentimen pada media sosial telah berkembang dengan penerapan berbagai algoritma klasifikasi dan pendekatan pengolahan data. Beberapa penelitian terdahulu yang memiliki keterkaitan dengan penelitian ini digunakan sebagai acuan dalam menentukan metode, objek kajian, serta posisi penelitian yang dilakukan.

Penelitian oleh (Sitanggang et al., 2024) mengkaji opini masyarakat terhadap program makan siang gratis melalui media sosial X dengan menerapkan algoritma *Naïve Bayes* menggunakan metodologi *Knowledge Discovery in Databases* (KDD). Data diperoleh melalui proses *scraping* sebanyak 2.211 *tweet* yang kemudian melalui tahap seleksi dan pelabelan ke dalam dua kategori sentimen, yaitu positif dan negatif. Evaluasi dilakukan menggunakan beberapa skenario pembagian data *training* dan *testing* dengan pendekatan *Confusion matrix*. Hasil terbaik diperoleh pada rasio 60:40 dengan nilai akurasi 72,2%, presisi 63,2%, *Recall* 66,1%, dan *F1-score* 64%.

Penelitian lain oleh (Zainuri, 2026) membahas pemanfaatan analisis sentimen untuk mengevaluasi persepsi masyarakat terhadap Program Makan Bergizi Gratis (MBG). Penelitian tersebut menggunakan algoritma *Naïve Bayes* sebagai metode klasifikasi serta menerapkan teknik ADASYN untuk mengatasi ketidakseimbangan jumlah data antar kelas. Berdasarkan 2.351 *tweet* yang dianalisis, diperoleh distribusi sentimen positif sebesar 76,2% dan sentimen negatif sebesar 23,8%. Hasil evaluasi *Model* menunjukkan akurasi sebesar 86%, sementara penerapan ADASYN mampu meningkatkan kemampuan *Model* dalam mengenali kelas minoritas melalui peningkatan nilai *Recall* dari 42% menjadi 83% dan *F1-score* dari 57% menjadi 67%.

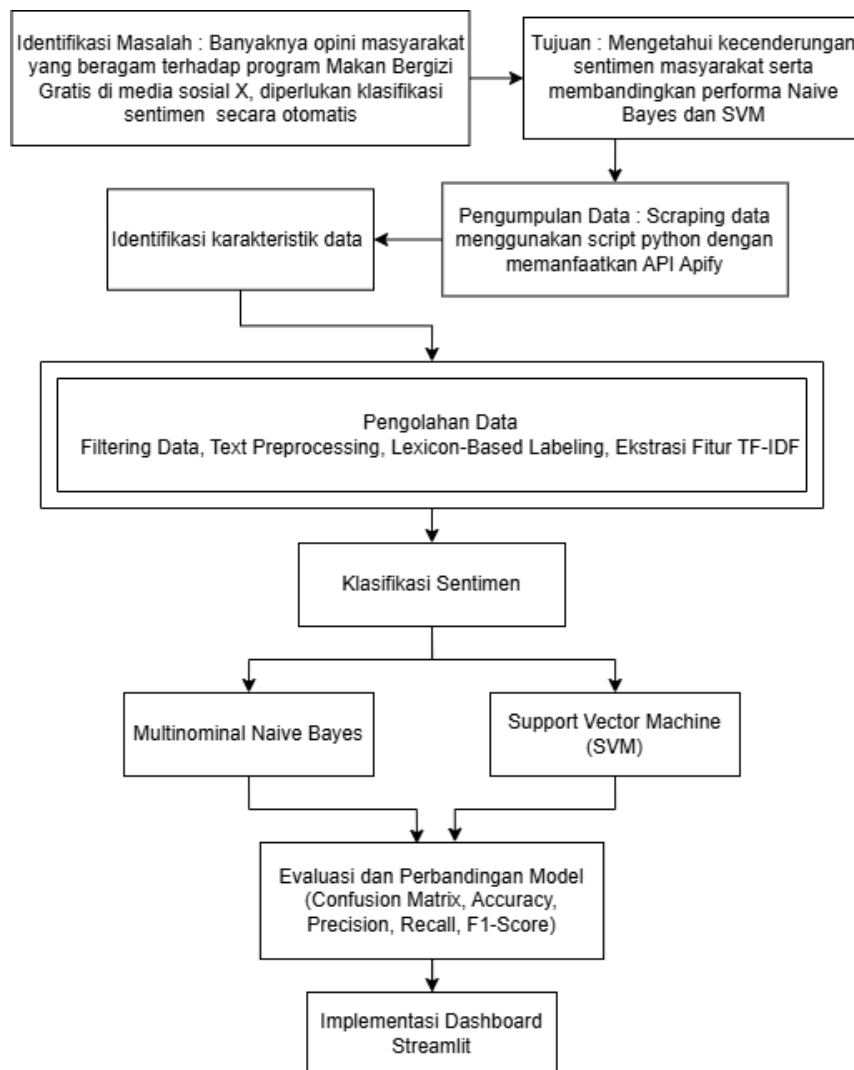
Perbandingan algoritma klasifikasi juga dilakukan oleh (Ilmawan & Mude, 2020) melalui penelitian mengenai klasifikasi ulasan teks berbahasa Indonesia pada Google Play Store. Penelitian tersebut membandingkan algoritma *Support Vector Machine* (SVM) dan *Naïve Bayes* menggunakan metode *K-Fold Cross-validation* dengan $K=3$. Hasil pengujian menunjukkan bahwa SVM menghasilkan performa yang lebih baik dibandingkan *Naïve Bayes*, dengan akurasi masing-masing sebesar 81,46% dan 75,41%.

Selain perbandingan algoritma, penelitian (Rizki et al., 2021) menerapkan SVM untuk analisis sentimen pada konten *cyberbullying* di media sosial X melalui pengembangan sistem berbasis web. Data yang digunakan berupa *tweet* dengan hashtag tertentu seperti #cebong dan #kadrun sebanyak maksimal 100 data. Tahapan penelitian meliputi *text preprocessing* dan pembobotan fitur menggunakan TF-IDF. Hasil evaluasi menunjukkan nilai *Recall* sebesar 64%, *Precision* sebesar 58%, dan *Accuracy* sebesar 70%.

Berdasarkan beberapa penelitian tersebut, dapat diketahui bahwa algoritma *Naïve Bayes* dan SVM memiliki kemampuan yang baik dalam melakukan klasifikasi teks, namun performanya dapat berbeda tergantung karakteristik data, metode pengolahan, serta konfigurasi *Model* yang digunakan. Oleh karena itu, penelitian ini melakukan perbandingan kedua algoritma tersebut pada objek Program Makan Bergizi Gratis (MBG) di media sosial X untuk mengetahui metode yang memberikan performa klasifikasi sentimen terbaik.

2.3 Kerangka Pemikiran

Kerangka pemikiran disusun sebagai dasar dalam menggambarkan hubungan antara permasalahan penelitian, pendekatan yang digunakan, tahapan pengolahan data, metode klasifikasi, hingga hasil yang ingin dicapai. Kerangka pemikiran dalam penelitian ini menunjukkan alur penelitian analisis sentimen masyarakat terhadap Program Makan Bergizi Gratis (MBG) berdasarkan data yang diperoleh dari media sosial X. Alur tersebut mencakup proses pengolahan data, pembentukan dan evaluasi *Model* klasifikasi, serta penerapan *Model* terbaik pada aplikasi pendukung berbasis *Streamlit*. Kerangka pemikiran penelitian ini disajikan pada Gambar 2.9.



Gambar 2. 9 Kerangka Pemikiran

Kerangka pemikiran pada Gambar 2.9 menggambarkan tahapan pemikiran yang menjadi dasar pelaksanaan penelitian, mulai dari identifikasi permasalahan hingga menghasilkan *Model* klasifikasi sentimen. Permasalahan penelitian berangkat dari munculnya berbagai opini masyarakat terhadap Program Makan Bergizi Gratis (MBG) pada media sosial X. Keberagaman opini yang disampaikan dalam bentuk teks tidak terstruktur menyebabkan diperlukan suatu pendekatan komputasi untuk mengidentifikasi kecenderungan sentimen masyarakat secara sistematis.

Pendekatan *text mining* diterapkan untuk mengolah data opini masyarakat yang diperoleh dari media sosial X. Data tersebut dipersiapkan melalui beberapa tahapan pengolahan, yaitu *filtering*, *text preprocessing*, pelabelan sentimen berbasis *lexicon*, serta transformasi data teks menjadi representasi numerik menggunakan metode TF-IDF. Hasil representasi fitur tersebut kemudian digunakan sebagai masukan dalam proses klasifikasi menggunakan algoritma *Multinomial Naïve Bayes* dan *Support Vector Machine* (SVM).

Performa kedua algoritma dianalisis melalui hasil evaluasi menggunakan *Confusion matrix*, *Accuracy*, *Precision*, *Recall*, dan *F1-score*. Perbandingan tersebut digunakan untuk menentukan algoritma dengan kemampuan klasifikasi terbaik. *Model* terpilih kemudian diterapkan pada *Dashboard* berbasis *Streamlit* sebagai media pendukung untuk menampilkan hasil analisis serta melakukan pengujian prediksi terhadap data baru.