# BAB II

# LANDASAN TEORI

## 2.1 Penelitian Terdahulu

Tabel 2.1 Tabel Jurnal *Text Mining* & *Sentiment Analysis*

| **Artikel Jurnal 1** | |
|---|---|
| Penulis | L. Feng, K. Liu, J. Wang, K.-Y. Lin, K. Zhang, and L. Zhang [12] |
| Judul | *Identifying Promising Technologies of Electric Vehicles from the Perspective of Market and Technical Attributes* |
| Tahun | 2022 |
| Metodologi | Penelitian ini menggunakan beberapa metode untuk menganalisis data dan mengidentifikasi teknologi mobil listrik yang menjanjikan. *Text mining* diterapkan untuk mengekstrak kata kunci dari *review online* dan *patent* data. *Sentiment analysis* digunakan untuk mengukur tingkat kepuasan pengguna terhadap mobil listrik berdasarkan komentar di *online reviews*. Untuk menganalisis fitur teknis, analisis *centrality* jaringan sosial dan *DEA-Malmquist model* digunakan untuk menilai pengaruh dan perubahan efisiensi teknologi seiring waktu. Metode *CRITIC* diterapkan untuk memberikan bobot objektif pada indikator teknis. Peta portofolio dibuat untuk memetakan teknologi berdasarkan permintaan pasar dan fitur teknis. |
| Hasil Pembahasan | Hasil analisis menunjukkan bahwa teknologi yang paling menjanjikan dalam mobil listrik meliputi sistem interior, perangkat keselamatan, dan sistem tenaga. Teknologi terkait interior dan keselamatan memiliki permintaan pasar tinggi, sedangkan teknologi sistem tenaga seperti *solar power*, *Battery Management System* (BMS), dan optimasi konsumsi daya menunjukkan potensi besar dalam hal kinerja dan efisiensi mobil. Pemetaan hasil ini menunjukkan teknologi yang berada di kuadran pertama sebagai yang paling menjanjikan, dengan kombinasi permintaan pasar dan fitur teknis yang tinggi. |

Tabel 2.2 Tabel Jurnal *Naive Bayes* & *Community Detection*

| **Artikel Jurnal 2** | |
|---|---|
| Penulis | S. Bhatnagar, N. Choubey [13] |

| Artikel Jurnal 2 | |
|---|---|
| Judul | *Making sense of tweets using sentiment analysis on closely related topics* |
| Tahun | 2021 |
| Metodologi | Penelitian ini menggunakan kombinasi dari analisis sentimen dan deteksi komunitas untuk memahami opini publik terhadap topik-topik terkait melalui *platform* Twitter. Pertama, data teks (*tweet*) yang berhubungan dengan topik-topik tertentu diekstraksi menggunakan fitur *hashtag* yang digunakan oleh pengguna. *Hashtags* ini berfungsi untuk menandai topik yang dibahas, memberikan wawasan langsung tentang area percakapan di Twitter. *Preprocessing* dilakukan pada teks untuk menyiapkan data, termasuk penghapusan URL, *@mention*, dan tokenisasi. Setelah itu, analisis sentimen diterapkan menggunakan model *Naive Bayes* untuk mengklasifikasikan tweet ke dalam kategori sentimen positif, negatif, atau netral. Model *Fluid Community Detection* digunakan untuk mendeteksi komunitas berdasarkan keterkaitan antara *tweet* yang menggunakan *hashtag* yang sama atau saling berhubungan, sehingga memungkinkan identifikasi topik terkait. Kemudian, model ini mengukur sentimen keseluruhan dari topik yang teridentifikasi, mengukur kesetiaan pasar dan sentimen pengguna terhadap berbagai isu. |
| Hasil Pembahasan | Hasil analisis menunjukkan bahwa model yang diusulkan berhasil mengidentifikasi topik terkait dan sentimen publik dengan lebih baik daripada metode tradisional seperti *Latent Dirichlet Allocation* (LDA). |

Tabel 2.3 Tabel Jurnal Sentimen Insentif Pajak EV

| Artikel Jurnal 3 | |
|---|---|
| Penulis | A. S. Wibowo, D. Septiari [14] |
| Judul | *How Does the Public React to the Electric Vehicle Tax Incentive Policy? A Sentiment Analysis* |
| Tahun | 2023 |
| Metodologi | Metodologi dalam penelitian ini menggabungkan analisis sentimen dan analisis emosi menggunakan data Twitter terkait kebijakan insentif pajak mobil listrik (EV) di Indonesia. Data yang digunakan mencakup 99,856 *tweet* yang dipilih dari periode Mei 2022 hingga Mei 2023 dengan kata kunci seperti "mobil listrik" dan "kendaraan listrik." Sentimen *tweet* dianalisis menggunakan model *IndoRoBERTa Base Sentiment Classifier* untuk mengklasifikasikan *tweet* ke |

| Artikel Jurnal 3 | |
|---|---|
| Metodologi | dalam kategori positif, negatif, atau netral dan analisis emosi menggunakan model *IndoRoBERTa Emotion Classifier*. |
| Hasil Pembahasan | Hasil analisis menunjukkan bahwa sentimen netral mendominasi dengan 59,1% dari *tweet*, diikuti oleh sentimen negatif sebesar 22,9% dan sentimen positif sebesar 18%. Sebanyak 56% dari *tweet* mendukung insentif pajak EV, sementara 44% menentangnya, terutama karena kekhawatiran mengenai harga EV yang tinggi, subsidi yang dianggap tidak adil, dan terbatasnya infrastruktur pengisian. Temuan ini menunjukkan bahwa meskipun ada dukungan terhadap kebijakan insentif pajak EV, kekhawatiran terkait harga dan infrastruktur perlu ditangani agar kebijakan ini dapat diterima lebih luas oleh masyarakat. |

Tabel 2.4 Tabel Jurnal Sentimen Analisis CNN

| Artikel Jurnal 4 | |
|---|---|
| Penulis | Ginni Yema Sitio, Sri Agustina Rumapea, dan Posma Lumbanraja [8] |
| Judul | Analisis Sentimen Pemindahan Ibu Kota Negara di Media Sosial Twitter Menggunakan Metode *Convolutional Neural Network* (CNN) |
| Tahun | 2023 |
| Metodologi | Penelitian ini menggunakan algoritma *Convolutional Neural Network* (CNN) untuk melakukan analisis sentimen terhadap opini masyarakat di Twitter mengenai pemindahan ibu kota negara. Data diklasifikasikan ke dalam tiga kategori sentimen: positif, netral, dan negatif. |
| Hasil Pembahasan | Hasil analisis menunjukkan bahwa model *Convolutional Neural Network* (CNN) yang diterapkan berhasil mengklasifikasikan 948 data *tweet* mengenai pemindahan ibu kota negara menjadi tiga kategori sentimen, yaitu positif, netral, dan negatif. Dari jumlah tersebut, 424 *tweet* tergolong dalam sentimen positif, 329 termasuk sentimen netral, dan 195 lainnya tergolong dalam sentimen negatif. Model CNN yang digunakan dilatih dengan parameter 100 *epoch* dan *batch size* 4, dan menghasilkan akurasi sebesar 56,06%. Selain itu, model ini juga memperoleh nilai presisi sebesar 67%, *recall* sebesar 62%, dan *F1-score* sebesar 64%. Hasil ini menunjukkan bahwa CNN memiliki performa yang cukup baik dalam menganalisis sentimen dari data media sosial, khususnya Twitter. |

Tabel 2.5 Tabel Jurnal VADER & LDA *Topic Modeling*

| **Artikel Jurnal 5** | |
|---|---|
| Penulis | H. P. Suresha, K. K. Tiwari [15] |
| Judul | *Topic Modeling and Sentiment Analysis of Electric Vehicles of* Twitter Data |
| Tahun | 2021 |
| Metodologi | Metode yang digunakan dalam penelitian ini melibatkan pengumpulan data Twitter terkait mobil listrik menggunakan Twitter API, diikuti dengan *pre-processing* menggunakan teknik *Natural Language Processing* (NLP) untuk membersihkan data, seperti menghapus URL, *hashtag*, dan *stopwords*. Selanjutnya, dilakukan *topic modeling* dengan *Latent Dirichlet Allocation* (LDA) untuk mengidentifikasi topik-topik utama dalam *tweet*, serta analisis sentimen menggunakan *VADER* untuk mengkategorikan *tweet* menjadi positif, negatif, atau netral. |
| Hasil Pembahasan | Hasil analisis menunjukkan bahwa sentimen netral mendominasi (59,1%), diikuti oleh positif (47,1%) dan negatif (10,5%). Tema utama yang ditemukan mencakup harga EV, masalah lingkungan, dan infrastruktur pengisian daya. Tesla menjadi *hashtag* yang paling sering digunakan, dan meskipun ada 56% dukungan terhadap kebijakan mobil listrik, isu terkait harga tinggi dan kurangnya infrastruktur masih menjadi kendala besar yang mempengaruhi adopsi lebih lanjut. |

Tabel 2.6 Tabel Jurnal Perbandingan SVM dan *Naïve Bayes*

| **Artikel Jurnal 6** | |
|---|---|
| Penulis | R. A. Ekatama, M. Rahardi, A. Aminuddin and F. F. Abdulloh [16] |
| Judul | *Sentiment Analysis of Electric Vehicles in Indonesia Using Support Vector Machine and Naïve Bayes* |
| Tahun | 2023 |
| Metodologi | *Support Vector Machine* dan *Naïve Bayes* |

| Artikel Jurnal 6 | |
|---|---|
| Hasil Pembahasan | Studi ini menyimpulkan bahwa analisis sentimen terhadap mobil listrik di Indonesia menunjukkan hasil yang signifikan. Dengan menggunakan dua algoritma pembelajaran mesin, yaitu *Support Vector Machine* (SVM) dan *Naïve Bayes*, penelitian ini menganalisis 3.178 data Twitter yang mencakup 1.818 sentimen netral, 870 sentimen positif, dan 490 sentimen negatif. Hasil penelitian menunjukkan bahwa SVM mencapai metrik kinerja yang sangat tinggi, dengan akurasi sebesar 91,02%, presisi 91,00%, *recall* 91,01%, dan *F1-Score* 91,00%. Sebaliknya, *Naïve Bayes* menghasilkan skor yang lebih rendah, dengan akurasi 83,68%, presisi 83,91%, *recall* 83,61%, dan *F1-Score* 83,51%. Temuan ini menegaskan bahwa SVM lebih efektif dalam mengklasifikasikan sentimen terkait mobil listrik dibandingkan dengan *Naïve Bayes*. |

Tabel 2.7 Tabel Jurnal Analisis Sentimen dengan *Naïve Bayes*

| Artikel Jurnal 7 | |
|---|---|
| Penulis | A. Erfina, R. A. Lestari [17] |
| Judul | *Sentiment Analysis of Electric Vehicles using the Naïve Bayes Algorithm* |
| Tahun | 2023 |
| Metodologi | Metode penelitian ini menggunakan teknik pengumpulan data melalui *scraping* data dari komentar di Youtube terkait dengan mobil listrik. Setelah data terkumpul, dilakukan tahap *preprocessing* yang meliputi pembersihan data dari simbol yang tidak diperlukan, *case folding* (penyeragaman huruf), tokenisasi (pemecahan kalimat menjadi kata-kata), penghapusan *stopword*, dan *stemming* (mengubah kata menjadi bentuk dasar). Selanjutnya, dilakukan analisis sentimen dengan menggunakan algoritma *Naïve Bayes* yang diuji dengan metode *K-Fold Cross Validation*. Pada tahap ini, data dibagi menjadi beberapa bagian untuk melatih dan menguji model. |
| Hasil Pembahasan | Studi ini menyimpulkan bahwa analisis sentimen terhadap mobil listrik di Indonesia menunjukkan hasil yang signifikan. Dengan menggunakan algoritma *Naïve Bayes* dan metode *K-Fold Cross Validation*, penelitian ini memperoleh nilai akurasi sebesar 82%. Hasil analisis menunjukkan bahwa sentimen negatif mendominasi dengan persentase 82%, sementara sentimen positif hanya mencapai 18%. |

Tabel 2.8 Tabel Jurnal Sentimen EV *FastText* & *IndoBERT*

| **Artikel Jurnal 8** | |
|---|---|
| Penulis | D. R. Wijaya, G. M. A. Sasmitha, W. O. Vihikan [18] |
| Judul | *Sentiment Analysis of Indonesian Citizens on Electric Vehicle Using FastText and BERT Method* |
| Tahun | 2024 |
| Metodologi | Metode *FastText* dan *IndoBERT* |
| Hasil Pembahasan | Studi ini menyimpulkan bahwa analisis sentimen terhadap mobil listrik di Indonesia menunjukkan hasil yang positif. Dengan menggunakan metode *FastText* dan *IndoBERT*, penelitian ini menganalisis 119.310 data *tweet* dari Januari 2020 hingga Juni 2023. Model *IndoBERT* mencapai akurasi tertinggi sebesar 82,5%, dengan hasil menunjukkan bahwa 58% dari persepsi masyarakat terhadap mobil listrik adalah positif. |

Tabel 2.9 Tabel Jurnal Akurasi *Logistic Regression* dengan PCA

| **Artikel Jurnal 9** | |
|---|---|
| Penulis | F. Jingga, R. Kosala, S. H. Supangkat and B. Ranti [19] |
| Judul | Analisis Sentimen Kendaraan Listrik Pada Media Sosial Twitter Menggunakan Algoritma *Logistic Regression* dan *Principal Component Analysis* |
| Tahun | 2023 |
| Metodologi | Metode yang digunakan dalam penelitian ini adalah analisis sentimen terhadap *tweet* yang berkaitan dengan kendaraan listrik di Twitter. Data yang digunakan terdiri dari 1.874 *tweet* yang dibagi menjadi data pelatihan dan data pengujian dengan rasio 80:20. Untuk mengklasifikasikan opini, penelitian ini menggunakan metode *Logistic Regression* (LR) yang dioptimalkan dengan *Principal Component Analysis* (PCA) untuk meningkatkan akurasi. PCA digunakan untuk mereduksi dimensi data, sehingga mempermudah proses klasifikasi. |

| Artikel Jurnal 9 | |
|---|---|
| Hasil Pembahasan | Hasil penelitian menunjukkan bahwa mayoritas opini masyarakat terkait kendaraan listrik bersifat positif, dengan persentase 86,9%, sementara sisanya 13,1% bersifat negatif. Akurasi yang diperoleh dari model *Logistic Regression* tanpa optimasi adalah 87,9%, dan setelah dilakukan optimasi dengan PCA, akurasi meningkat menjadi 90%. Temuan ini mengindikasikan bahwa secara umum, masyarakat memiliki pandangan positif terhadap kendaraan listrik di media sosial Twitter, meskipun masih ada sebagian kecil yang menyuarakan pendapat negatif. |

Tabel 2.10 Tabel Jurnal Evaluasi LSTM dalam Analisis Sentimen EV

| Artikel Jurnal 10 | |
|---|---|
| Penulis | F. Jingga, R. Kosala, S. H. Supangkat and B. Ranti [7] |
| Judul | Analisis Sentimen Mobil Listrik di Indonesia Menggunakan *Long-Short Term Memory* (LSTM) |
| Tahun | 2023 |
| Metodologi | *Long Short Term Memory* |
| Hasil Pembahasan | Studi ini menyimpulkan bahwa analisis sentimen terhadap mobil ramah lingkungan menggunakan algoritma *Long-Short Term Memory* (LSTM) menunjukkan hasil yang memadai. Dengan mengumpulkan data dari komentar YouTube dalam bentuk teks bahasa Indonesia dan membagi *dataset* dengan rasio 67:33 untuk data pelatihan dan pengujian, model ini mencapai tingkat akurasi sebesar 63%. Hasil analisis menunjukkan bahwa sentimen negatif mendominasi, terutama terkait dengan harga mobil listrik, yang menjadi perhatian utama masyarakat. Metrik evaluasi lainnya mencakup *macro average precision* 62%, *macro average recall* 60%, *macro average F1-score* 60%, *weighted average precision* 62%, *weighted average recall* 63%, *weighted average F1-score* 62%, dan *ROC AUC* 81%. |

Berdasarkan Tabel 2.1 hingga Tabel 2.10, dapat disimpulkan bahwa penelitian mengenai analisis sentimen terhadap mobil listrik telah dilakukan dengan pendekatan yang beragam, baik dari sisi data, metodologi, maupun fokus analisisnya. Hasil dari studi-studi ini menunjukkan variasi sentimen yang cukup signifikan tergantung pada konteks data dan model yang digunakan. Beberapa studi menemukan dominasi sentimen negatif, seperti pada analisis komentar YouTube

[17] dan studi berbasis LSTM [7], yang mengungkapkan kekhawatiran masyarakat terkait harga mobil listrik dan keterbatasan infrastruktur. Namun, ada pula penelitian yang menunjukkan mayoritas sentimen positif terhadap mobil listrik, terutama ketika data diambil dari Twitter dan dianalisis menggunakan metode yang lebih kompleks seperti *IndoBERT* [18] dan *Logistic Regression* dengan PCA [19]. Dari sisi teknis, telah terbukti bahwa SVM lebih unggul dibandingkan *Naïve Bayes* dalam hal akurasi dan metrik evaluasi lainnya [16], sementara *IndoBERT* juga menunjukkan performa yang kuat dengan akurasi tinggi [18]. Beberapa studi turut menambahkan analisis lanjutan seperti analisis emosi [14], pemodelan topik dengan LDA [15], dan deteksi komunitas [13], yang memberikan pemahaman yang lebih komprehensif terhadap opini publik.

Selain fokus pada sentimen pengguna, studi [12] mengambil pendekatan yang berbeda dengan memadukan *text mining*, *sentiment analysis*, dan penilaian teknis terhadap teknologi mobil listrik. Hasilnya menyoroti bahwa teknologi seperti *Battery Management System* (BMS) dan sistem tenaga surya dianggap paling menjanjikan karena performa teknis dan minat pasar yang tinggi. Karakteristik data pada sebagian besar jurnal menunjukkan bahwa data yang digunakan relatif bersih, relevan, dan berasal dari *platform* populer seperti Twitter, YouTube, forum otomotif, Kaggle, hingga data paten. Tahapan *preprocessing* yang dilakukan umumnya mencakup proses dasar seperti *text cleaning*, *case folding*, tokenisasi, dan penghapusan *stopword*. Namun, secara umum, tahapan ini masih bersifat standar dan tidak terlalu mendalam. Secara keseluruhan, rangkaian studi ini mengindikasikan bahwa analisis sentimen merupakan alat penting untuk memahami persepsi publik terhadap mobil listrik, yang dapat menjadi dasar dalam penyusunan kebijakan, pengembangan produk, maupun strategi pemasaran. Pemilihan metode yang tepat, kualitas data, serta konteks lokal menjadi faktor penting yang memengaruhi akurasi dan relevansi hasil analisis.

## 2.2 Teori tentang Topik Skripsi

### 2.2.1 Analisis Sentimen

Analisis sentimen adalah teknik yang digunakan untuk menganalisis dan memahami opini, perasaan, atau sentimen yang terkandung dalam teks yang

dihasilkan oleh pengguna di berbagai *platform* digital, seperti media sosial, forum, blog, atau ulasan produk [20]. Tujuan dari analisis sentimen adalah untuk mengekstrak opini yang terkandung dalam teks tersebut, yang kemudian dapat digunakan untuk membuat keputusan yang lebih baik dalam berbagai bidang, seperti bisnis, pemasaran, kebijakan pemerintah, atau analisis sosial. Proses ini berfungsi untuk mengolah data teks yang tidak terstruktur menjadi informasi yang lebih terorganisir dan mudah dipahami. Dalam era digital yang penuh dengan informasi, analisis sentimen membantu mengidentifikasi persepsi publik terhadap topik tertentu dengan cepat dan efisien.

Sebagai bagian dari *Natural Language Processing* (NLP), analisis sentimen memanfaatkan teknik pengolahan bahasa alami untuk mengenali pola-pola dalam teks yang menunjukkan sentimen positif, negatif, atau netral. Proses ini melibatkan analisis terhadap beberapa komponen penting, yaitu subjek (topik yang dibicarakan), polaritas (apakah sentimen tersebut positif, negatif, atau netral), dan pemegang opini (siapa yang mengungkapkan opini tersebut) [21]. Dengan demikian, analisis sentimen tidak hanya membantu dalam mengidentifikasi apakah teks tersebut mengandung opini, tetapi juga menggali lebih dalam terkait konteks dan sentimen yang terkandung di dalamnya.

### 2.2.2 Mobil Listrik

Mobil listrik adalah kendaraan yang digerakkan oleh motor listrik yang mendapatkan sumber energi dari baterai yang dapat diisi ulang. Berbeda dengan kendaraan konvensional yang menggunakan mesin pembakaran internal dan bahan bakar fosil, mobil listrik lebih efisien dalam penggunaan energi dan menghasilkan emisi yang lebih rendah. Mobil listrik ditemukan pada tahun 1832 oleh Robert Anderson dari Skotlandia [22]. Hingga pada akhir abad ke-20 dan awal abad ke-21, mobil listrik kembali menarik perhatian, terutama karena meningkatnya kesadaran akan isu lingkungan dan pentingnya transportasi yang lebih ramah lingkungan. Kemajuan teknologi baterai, seperti pengembangan baterai lithium-ion, serta dukungan kebijakan pemerintah dan investasi dari perusahaan otomotif, telah mendorong kebangkitan mobil listrik di pasar global [23]. Saat ini, mobil listrik

semakin populer dan menjadi bagian dari upaya global untuk mengurangi emisi karbon dan ketergantungan pada bahan bakar fosil.

### 2.2.3 Youtube

YouTube adalah *platform* berbagi video yang memungkinkan penggunanya untuk mengunggah, menonton, dan berbagi video dari seluruh dunia. Didirikan pada Februari 2005 oleh Steve Chen, Chad Hurley, dan Jawed Karim, yang sebelumnya bekerja di PayPal, YouTube kini menjadi situs web terpopuler kedua di dunia setelah Google. Sebagai *platform* yang mudah diakses, YouTube telah menjadi sumber hiburan, informasi, dan edukasi yang penting, dengan jutaan video tersedia untuk ditonton oleh berbagai kalangan. Bagi bisnis, YouTube berfungsi sebagai alat pemasaran yang efektif untuk memperkenalkan produk atau layanan melalui video promosi, tutorial, atau testimoni pelanggan. *Platform* ini juga memungkinkan interaksi sosial antara pembuat konten dan audiens melalui komentar, *live chat*, dan fitur komunitas yang mempererat hubungan. Pengguna YouTube bisa melakukan berbagai aktivitas, seperti menonton video secara bebas, mengunggah konten pribadi atau profesional, dan berinteraksi dengan video melalui *like*, *comment*, atau *share* [6]. Pengguna juga dapat berlangganan *channel* favorit mereka untuk menerima pembaruan video terbaru. YouTube menyediakan opsi berlangganan premium, yang memungkinkan pengguna menikmati video tanpa gangguan iklan, mengunduh konten untuk ditonton *offline*, serta menikmati musik secara latar belakang. Selain itu, YouTube terus berkembang dengan fitur-fitur baru, seperti YouTube Shorts yang menawarkan video pendek dan YouTube *Music* untuk *streaming* musik [6]. YouTube digunakan oleh berbagai kalangan, mulai dari pelajar, profesional, hingga pebisnis. Di Indonesia, YouTube adalah salah satu media sosial yang paling banyak digunakan, dengan lebih dari 139 juta pengguna aktif pada tahun 2023. *Platform* ini memainkan peran besar dalam kehidupan digital sehari-hari, baik untuk tujuan hiburan, pendidikan, maupun pengembangan bisnis. Dengan keanekaragaman konten dan fitur yang tersedia, YouTube tetap menjadi salah satu *platform* terdepan dalam dunia digital.

### 2.2.4 CRISP-DM

*Cross-Industry Standard Process for Data Mining* (CRISP-DM) adalah kerangka kerja standar yang digunakan untuk melakukan data *mining* [24]. Kerangka kerja ini terdiri dari enam fase, yaitu:

1. *Business Understanding*

adalah tahap awal untuk memahami secara mendalam tujuan bisnis dan kebutuhan pemangku kepentingan [24]. Kegiatan yang dilakukan diantaranya, mengidentifikasi dan menyusun strategi untuk mencapai tujuan tersebut dengan mempertimbangkan keterbatasan yang ada. Hasilnya, langkah-langkah yang diambil akan langsung berkaitan dengan masalah bisnis yang perlu diselesaikan.

2. *Data Understanding*

adalah proses yang mencakup pengumpulan, eksplorasi, dan pemahaman mendalam terhadap data yang akan digunakan dalam proyek [24]. Tujuan utama dari fase ini adalah untuk memperoleh pemahaman yang kuat tentang karakteristik, struktur, dan konten data. Pemahaman yang baik terhadap data merupakan fondasi kritis untuk kesuksesan proses data mining selanjutnya, memastikan bahwa data yang digunakan adalah representatif, berkualitas tinggi, dan sesuai dengan tujuan bisnis yang ditetapkan.

3. *Data Preparation*

mencakup langkah-langkah untuk membersihkan dan menyusun data mentah agar siap digunakan dalam proses pemodelan. Ini melibatkan pemilihan data yang relevan, transformasi, serta pembersihan untuk memastikan data dalam format yang tepat, memungkinkan model yang akan dibangun untuk mendapatkan hasil yang optimal [24].

4. *Modelling*

merupakan fase di mana dilakukan pembangunan model prediktif atau deskriptif berdasarkan data yang telah dikumpulkan dan dipersiapkan. Tujuan utama dari fase ini adalah menghasilkan suatu representasi matematis atau statistik

dari hubungan antara variabel dapat digunakan untuk membuat prediksi atau menggambarkan pola yang ada [24].

5. *Evaluation*

merupakan fase mengevaluasi performa model yang telah dibangun pada tahap *modeling*. Evaluasi ini dilakukan untuk memastikan model yang dikembangkan dapat memenuhi tujuan bisnis dan kriteria keberhasilan yang telah ditetapkan [24].

6. *Deployment*

adalah fase terakhir dalam siklus proses *data mining*. Pada tahap ini, model atau hasil analisis yang telah dikembangkan dan dievaluasi pada tahap sebelumnya diterapkan ke lingkungan bisnis atau organisasi secara praktis [24]. *Deployment* adalah fase kunci yang mengarah pada penerapan praktis dari hasil data mining ke dalam operasi bisnis. Keberhasilan implementasi bergantung pada pemahaman yang baik tentang konteks bisnis, koordinasi yang baik serta perencanaan dan pemantauan yang efektif.

### 2.2.5 API Data *Extraction*

API Data *Extraction* adalah sebuah metode yang memungkinkan aplikasi untuk mengambil data dari suatu sumber menggunakan antarmuka pemrograman aplikasi (API) [25]. Dengan menggunakan YouTube Data API v3, pengembang dapat mengakses dan mengekstrak data komentar untuk berbagai tujuan, seperti analisis, pengumpulan informasi, atau interaksi pengguna. Cara kerja API *Data Extraction* dimulai dengan memperoleh API *Key* dari Google *Cloud Console* sebagai bagian dari proses otentikasi. Setelah API *Key* didapat, pengembang dapat memanfaatkan metode *commentThreads.list* untuk mengambil komentar dari video dengan menyertakan parameter *videoId* yang menunjukkan video target [26]. Data yang diperoleh dalam format JSON mencakup teks komentar, nama pengirim, jumlah suka, serta informasi waktu publikasi. Jika ada balasan terhadap komentar, metode *comments.list* dapat digunakan dengan parameter *parentId* untuk mengambil data balasan tersebut. API ini juga menyediakan mekanisme *nextPageToken* untuk mengakses komentar lebih banyak jika data yang diambil melebihi batas yang dapat diproses dalam satu permintaan [26]. Secara keseluruhan, API Data *Extraction*

untuk YouTube memberikan cara yang efisien dan terstruktur bagi pengembang untuk mengambil data komentar, yang dapat dimanfaatkan untuk riset, analisis sentimen, ataupun pengembangan aplikasi yang lebih interaktif.

### 2.2.6 *Text Preprocessing*

*Text preprocessing* merupakan langkah untuk membersihkan dan menyesuaikan data dengan format yang lebih sesuai, menghapus duplikasi, dan mengurangi gangguan atau ketidaksesuaian yang ada [27].

1. *Text Cleaning*

   merupakan langkah untuk menghilangkan simbol-simbol, karakter non-alfabet, serta elemen-elemen lain yang tidak relevan dari teks. Dengan membersihkan data dari *noise* ini, kualitas data yang digunakan dalam model analisis sentimen akan meningkat [28].

2. *Case Folding*

   adalah teknik untuk menyamakan seluruh huruf dalam teks menjadi huruf kecil (*lowercase*). Tujuan dari *case folding* adalah untuk menghindari perbedaan huruf kapital yang dapat menyebabkan variasi yang tidak perlu dalam proses pemodelan [27].

3. *Tokenization*

   adalah proses untuk memecah teks menjadi unit-unit yang lebih kecil (token) berdasarkan spasi atau tanda baca sebagai pemisah. Dalam tahap ini, teks yang panjang akan dibagi menjadi kata-kata atau frasa yang lebih mudah untuk dianalisis [27].

4. *Stemming*

   adalah proses yang digunakan untuk mengurangi kata-kata turunan atau berimbuhan menjadi bentuk dasarnya. Tujuannya adalah untuk menyederhanakan kata-kata dalam teks agar model dapat lebih fokus pada makna dasar dari kata tersebut, tanpa memperhatikan variasi bentuk kata yang ada [27].

5. *Stopword Removal*

adalah langkah untuk menghapus kata-kata yang tidak memiliki kontribusi besar terhadap makna teks Menghapus *stopword* dapat membantu meningkatkan efisiensi dan akurasi model dengan hanya memfokuskan pada kata-kata yang lebih relevan untuk analisis sentimen [28].

### 2.2.7 *Semi Supervised Learning*

*Semi Supervised Learning* (SSL) adalah metode dalam *machine learning* yang memanfaatkan kombinasi data berlabel dan tidak berlabel untuk meningkatkan akurasi model. Teknik ini sangat berguna ketika jumlah data berlabel terbatas, namun ada banyak data tidak berlabel yang bisa dimanfaatkan [29]. Salah satu metode yang sering digunakan dalam SSL adalah *self-training*. Pada pendekatan ini, model pertama kali dilatih menggunakan data yang berlabel, kemudian digunakan untuk memprediksi label pada data yang tidak berlabel. Prediksi yang memiliki tingkat kepercayaan tinggi, biasanya berdasarkan probabilitas yang melebihi ambang tertentu, dianggap sebagai label *pseudo* dan ditambahkan ke dalam *dataset* berlabel. Proses ini terus berulang hingga model mencapai konvergensi atau tidak ada data baru yang ditambahkan [29]. Metode *self-training* ini dapat diterapkan dengan berbagai jenis algoritma, salah satunya adalah *XGBoost*. *XGBoost* adalah model yang berbasis pada teknik gradient boosting, yang terkenal efisien dalam menangani data tabular dengan berbagai jenis fitur [30]. Dalam penelitian yang mengaplikasikan metode ini, *XGBoost* mampu mengolah data tidak berlabel yang sebelumnya tidak dimanfaatkan, sehingga meningkatkan akurasi model dalam mendeteksi masalah tersebut [31]. Teknik ini menunjukkan bagaimana kombinasi antara algoritma yang kuat dan data tidak berlabel dapat menghasilkan model yang lebih baik, bahkan ketika data yang berlabel terbatas. Secara keseluruhan, penerapan self-training dengan *XGBoost* memberikan solusi yang efisien dalam mengatasi keterbatasan data berlabel. Metode ini memungkinkan model untuk terus berkembang dengan memanfaatkan data tidak berlabel yang ada, yang pada akhirnya dapat menghasilkan prediksi yang lebih akurat dan optimal dalam berbagai aplikasi.

### 2.2.8 *Term Frequency – Inverse Document Frequency* (TF-IDF)

*Term Frequency-Inverse Document Frequency* (TF-IDF) adalah sebuah metode ekstraksi fitur dengan memberikan bobot pada kata-kata dalam suatu dokumen [32]. Ada dua komponen utama dalam perhitungan TF-IDF, yaitu:

1. *Term Frequency* (TF)

TF mengukur berapa sering suatu kata muncul dalam dokumen dan dihitung dengan membagi jumlah kemunculan kata dalam dokumen dengan total jumlah kata yang ada dalam dokumen tersebut. Semakin sering kata muncul dalam dokumen, semakin besar nilai TF-nya. Namun, TF ini bersifat lokal, artinya hanya mempertimbangkan frekuensi kata dalam dokumen itu sendiri [33].

$$TF\ (t,d) = \frac{Jumlah\ kemunculan\ kata\ \ t\ \ dalam\ dokumen\ \ d}{Jumlah\ total\ kata\ dalam\ dokumen\ \ d}$$

Rumus 2.1 Rumus *Term Frequency*

2. *Inverse Document Frequency* (IDF)

IDF mengukur berapa langka suatu kata di seluruh koleksi dokumen dan dihitung dengan membagi jumlah total dokumen dengan jumlah dokumen yang mengandung kata tersebut. Kemudian hitung dengan *log* untuk mendapatkan nilai IDF. Kata yang jarang muncul di banyak dokumen memiliki nilai IDF yang lebih tinggi dengan kata lain, kata tersebut dianggap lebih penting dan informatif [33].

$$IDF\ (t,D) = \log\frac{Total\ jumlah\ dokumen}{Jumlah\ dokumen\ yang\ mengandung\ kata\ \ t}$$

Rumus 2.2 Rumus *Inverse Document Frequency*

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

TF-IDF dihitung dengan mengalikan TF dan IDF untuk setiap kata dalam dokumen. Kata yang memiliki frekuensi tinggi berarti memiliki nilai TF-IDF yang tinggi. Nilai TF-IDF yang tinggi menunjukkan bahwa kata tersebut lebih relevan dan penting dalam konteks dokumen tersebut [33].

$$TF\text{-}IDF\ (t, d, D) = TF\ (t, d)\ \ IDF\ (t, D)$$

Rumus 2.3 Rumus TF-IDF

### 2.2.9 *Synthetic Minority Oversampling Technique* (SMOTE)

adalah metode untuk mengatasi masalah ketidakseimbangan kelas dalam *dataset* dengan cara *oversampling* [34]. Berbeda dari teknik *oversampling* konvensional yang hanya menggandakan data kelas minoritas, SMOTE menciptakan contoh data baru yang sintetis dengan cara interpolasi antara data yang ada dari kelas minoritas. Pendekatan ini membantu model untuk memahami kelas minoritas lebih baik tanpa memicu *overfitting*. Proses kerja SMOTE dimulai dengan memilih satu titik data secara acak dari kelas minoritas. Kemudian, algoritma akan mencari ke tetangga terdekat dari titik tersebut. Dari tetangga terdekat yang ditemukan, SMOTE memilih salah satu secara acak dan menghasilkan titik data baru yang terletak di antara titik data asli dan tetangga tersebut. Proses ini diulang untuk setiap titik data kelas minoritas, menghasilkan data sintetis sesuai dengan rasio *oversampling* yang diinginkan [34]. Dengan menambahkan data sintetis, *SMOTE* berfungsi untuk menyeimbangkan distribusi kelas dalam *dataset*, membantu model untuk belajar lebih efektif dan mengurangi bias terhadap kelas mayoritas. Namun, penggunaannya harus disesuaikan dengan konteks dan karakteristik *dataset* untuk mencapai hasil terbaik.

### 2.2.10 *Confusion Matrix*

adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dalam *machine learning*. Tabel ini membandingkan hasil prediksi model dengan nilai aktual *(ground truth)*, sehingga dapat diketahui seberapa baik model dalam mengklasifikasikan data [35].

1. *Accuracy*

adalah ukuran yang menunjukkan sejauh mana model klasifikasi dapat memprediksi dengan benar. Ini adalah proporsi prediksi yang benar dari total jumlah prediksi yang dilakukan oleh model [36]. Akurasi dihitung dengan rumus:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Rumus 2.4 Rumus *Accuracy*

- TP (*True Positive*): Jumlah kasus positif yang diprediksi benar-benar positif.
- TN (*True Negative*): Jumlah kasus negatif yang diprediksi benar-benar negatif.
- FP (*False Positive*): Jumlah kasus negatif yang diprediksi positif.
- FN (*False Negative*): Jumlah kasus positif yang diprediksi negatif.

2. *Precision*

adalah proporsi prediksi positif yang memberi info seberapa tepat model dalam memprediksi kelas positif [36].

$$Precision = \frac{TP}{TP+FP}$$

Rumus 2.5 Rumus *Precision*

3. *Recall*

*Recall* adalah proporsi kasus positif yang berhasil diprediksi sebagai positif. Dengan kata lain, ini mengukur kemampuan model dalam menemukan semua kasus positif yang ada [36].

$$Recall = \frac{TP}{TP+FN}$$

Rumus 2.6 Rumus *Recall*

4. *F1-Score*

adalah rata-rata dari *precision* dan *recall* yang memberikan keseimbangan antara *precision* dan *recall* [36].

$$F1 - Score = \frac{Precision \times Recall}{Precision+Recall}$$

Rumus 2.7 Rumus *F1-Score*

### 2.2.11 *User Acceptance Test*

*User Acceptance Test* (UAT) merupakan tahap akhir dari proses pengujian perangkat lunak yang bertujuan untuk memastikan bahwa sistem atau aplikasi telah memenuhi kebutuhan dan harapan pengguna akhir [47]. UAT dilakukan langsung

oleh pengguna bukan oleh pengembang, sehingga pengujian ini lebih menekankan pada aspek fungsionalitas dari perspektif pengguna. Dalam UAT, pengguna mencoba menjalankan fitur-fitur utama sistem sesuai dengan skenario penggunaan yang telah dirancang. Tujuan utamanya adalah mengevaluasi apakah sistem dapat digunakan secara efektif dalam situasi nyata. Apabila ditemukan ketidaksesuaian atau kekurangan selama pengujian, maka akan dijadikan bahan evaluasi untuk penyempurnaan sistem sebelum peluncuran akhir [47].

Metode yang umum digunakan dalam UAT meliputi observasi langsung, pengisian kuesioner, wawancara, dan dokumentasi umpan balik. Kuesioner sering kali disusun berdasarkan skala *likert* untuk menilai berbagai aspek, seperti kemudahan penggunaan (*usability*), antarmuka pengguna (UI), kecepatan, hingga ketepatan hasil [48]. Penilaian ini memberikan gambaran objektif mengenai tingkat kepuasan dan penerimaan pengguna terhadap system. Jumlah partisipan dalam UAT umumnya digunakan 5 responden berdasarkan prinsip 5-*user rule* yang dikemukakan oleh Jakob Nielsen. Prinsip ini menyatakan bahwa 5 pengguna pertama biasanya sudah mampu menemukan mayoritas masalah utama pada sebuah sistem [49]. Namun, dalam praktiknya, beberapa penelitian memilih melibatkan lebih banyak responden, seperti 7 hingga 10 orang, untuk memperoleh masukan yang lebih variatif dan memperkuat validitas hasil uji penerimaan pengguna [50]. UAT memiliki peran penting dalam menjembatani sisi teknis dengan kebutuhan pengguna. Dalam konteks sistem klasifikasi berbasis machine learning, seperti yang digunakan dalam penelitian ini, UAT tidak hanya membantu mengukur fungsionalitas sistem, tetapi juga memastikan hasil klasifikasi benar-benar bermanfaat dan dapat dipahami oleh pengguna akhir.

## 2.3 Teori tentang Framework/Algoritma yang digunakan

### 2.3.1 Streamlit

Streamlit adalah *framework open-source* berbasis Python yang memungkinkan para *data scientist* atau *engineer* untuk dengan cepat dan mudah membangun aplikasi *web* interaktif tanpa memerlukan keahlian dalam pengembangan *frontend*, karena cukup menulis beberapa baris kode Python untuk mengubah skrip analisis data menjadi aplikasi *web* [37]. Aplikasi yang dibangun menggunakan Streamlit akan diperbarui secara otomatis setiap kali kode disimpan sehingga memberikan

pengalaman yang interaktif dan *real-time*. Selain itu, Streamlit kompatibel dengan berbagai *library* Python memungkinkan integrasi data dan visualisasi langsung dalam aplikasi [37]. Untuk memulai, cukup mengunduh Streamlit melalui *pip* dilanjutkan dengan mulai menulis kode aplikasi dengan Python. Aplikasi yang telah dibuat dapat dijalankan di server lokal atau disebarkan menggunakan *platform* seperti Streamlit *Community Cloud* atau Snowflake, yang cocok untuk aplikasi dengan kebutuhan skalabilitas dan keamanan tingkat perusahaan. Dengan demikian, Streamlit adalah alat yang sangat efisien untuk membuat prototipe aplikasi berbasis data dengan cara yang cepat dan efektif [37].

### 2.3.2 *Long Short-Term Memory* (LSTM)

*Long Short-Term Memory* (LSTM) adalah jenis arsitektur *neural network* yang termasuk dalam kategori *Recurrent Neural Network* (RNN). LSTM dirancang untuk mengatasi masalah yang sering dihadapi oleh RNN tradisional, yaitu hilangnya informasi penting dalam urutan data panjang akibat masalah *vanishing gradient* [38]. Pada LSTM, ada beberapa komponen yang dihitung pada setiap langkah waktu:

1. *Forget Gate*

bertanggung jawab untuk menentukan informasi apa yang harus dilupakan dari memori (*cell state*) sebelumnya [38].

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Rumus 2.8 Rumus *Forget Gate*

- $f_t$ adalah vektor bernilai antara 0 dan 1 yang menandakan seberapa besar informasi dari $C_{t-1}$akan dipertahankan.
- Jika $f_t$ mendekati 0 → informasi dilupakan.
- Jika $f_t$ mendekati 1 → informasi dipertahankan.
- *Input*nya adalah gabungan dari *hidden state* sebelumnya dan *input* saat ini, diproses melalui fungsi *sigmoid*.

2. *Input Gate*

menentukan informasi mana yang akan ditambahkan ke *cell state* pada langkah waktu saat ini [39].

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\hat{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Rumus 2.9 Rumus *Input Gate*

- $i_t$ adalah vektor *sigmoid* yang menentukan nilai mana yang akan diperbarui.
- $\hat{C}_t$ adalah vektor kandidat nilai baru untuk ditambahkan ke *cell state*, menggunakan fungsi aktivasi $tanh$.
- Keduanya mengontrol masuknya informasi baru yang relevan ke dalam *cell state*.

3. *Update Cell State*

merupakan langkah kunci dalam LSTM. *Cell state* yang baru, $\hat{C}_t$ dihitung dengan cara menggabungkan informasi yang telah dilupakan dan informasi yang baru ditambahkan [38].

$$\hat{C}_t = f_t * C_{t-1} + i_r * \hat{C}_t \quad (2.11)$$

Rumus 2.10 Rumus *Update Cell State*

4. *Output Gate*

menentukan informasi apa yang akan dikeluarkan sebagai *output* (*hidden state*) dari sel LSTM saat ini [38].

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = O_t * \tanh(\hat{C}_t)$$

Rumus 2.11 Rumus *Output Gate*

### 2.3.3 *Convolutional Neural Network* (*CNN*)

*Convolutional Neural Network* (*CNN*) adalah jenis arsitektur *neural network* tiruan yang banyak digunakan untuk pengolahan data berbentuk *grid*, seperti gambar atau teks yang telah diubah menjadi representasi spasial. *CNN* sangat efektif dalam mengenali pola spasial dan struktur lokal karena menggunakan operasi konvolusi untuk mengekstraksi fitur dari data *input* [39]. *CNN* terdiri dari beberapa lapisan diantaranya adalah:

1. *Convolutional Layer*

   Lapisan ini bertugas mengekstraksi fitur dari *input* dengan menggunakan *filter* yang setiap *filter* akan mendeteksi pola tertentu dari *input* [39].

$$Z_{i,j} = (X * W)_{i,j} + b = \sum_m \sum_n X_{i+m,j+n} \cdot W_{m,n} + b$$

Rumus 2.12 Rumus *Convolutional Layer*

- $X$ adalah *input*
- $W$ adalah *filter* atau *kernel*
- $b$ adalah bias
- $*$ adalah operasi konvolusi
- $Z_{i,j}$ adalah hasil konvolusi di posisi $(i, j)$
- Jika $f_t$ mendekati 0 → informasi dilupakan.

2. *Activation Layer*

   *Activation Function* (ReLU) berfungsi memberikan non-linearitas agar jaringan bisa mempelajari hubungan kompleks [39].

$$f(x) = \max(0, x)$$

Rumus 2.13 Rumus *Activation Layer*

3. *Pooling Layer*

   *Pooling Layer* (*Max Pooling*) berfungsi mengurangi dimensi (*downsampling*) dan mempertahankan fitur penting [39].

$$Z_{i,j} = \max\{Z_{m,n} | (m,n) \in window(i,j)\}$$

Rumus 2.14 Rumus *Pooling Layer*

4. *Fully Connected Layer*

*Fully Connected Layer* berfungsi menggabungkan semua fitur menjadi prediksi akhir [39].

$$y = f(W_x + b)$$

Rumus 2.15 Rumus *Fully Connected Layer*

5. *Output Layer*

*Output Layer* memberikan hasil jaringan [39].

$$Softmax(Z_i) = \frac{e^{zi}}{\sum_j e^{zj}}$$

Rumus 2.16 Rumus *Output Layer*

## 2.4 Teori tentang tools/software yang digunakan

### 2.4.1 Google Colaboratory

Google Colaboratory merupakan suatu *platform* yang disediakan oleh Google berbasis *cloud* [40]. Google Colab memungkinkan pengguna untuk menulis, menjalankan, dan membagikan kode dengan mudah tanpa perlu mengunduh perangkat lunak secara lokal, serta menyediakan akses gratis ke GPU dan TPU, yang sangat bermanfaat untuk proyek *machine learning* dan analisis data berskala besar. *Platform* ini terintegrasi langsung dengan Google Drive, sehingga memudahkan penyimpanan dan kolaborasi waktu nyata antar pengguna [40]. Dengan kemudahan akses dan fitur yang lengkap, Google Colab menjadi alat penting dalam dunia komputasi ilmiah modern, terutama bagi pengguna yang tidak memiliki perangkat keras kelas atas namun ingin belajar atau bekerja di bidang kecerdasan buatan dan data analitik.

### 2.4.2 Visual Studio Code

Visual Studio Code (VS Code) adalah *platform* gratis yang dikembangkan oleh Microsoft yang dirancang untuk pengembangan perangkat lunak modern [41]. VS Code mendukung berbagai bahasa pemrograman salah satunya Python. Salah satu keunggulan utamanya adalah antarmuka yang ringan namun kaya fitur, termasuk

fitur *live share* memungkinkan kolaborasi kode secara langsung antar pengembang [41]. VS Code dapat dikustomisasi secara luas, mulai dari tema, *shortcut keyboard*, hingga ekstensi yang memperluas fungsionalitasnya sesuai kebutuhan pengguna. VS Code telah menjadi salah satu editor kode yang paling populer di kalangan pengembang di seluruh dunia karena kemudahan penggunaan, performa yang cepat, dan fitur lengkapnya [41].