

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Untuk memperoleh pemahaman yang komprehensif mengenai perkembangan dan pendekatan yang telah digunakan dalam analisis sentimen, khususnya pada teks berbahasa Indonesia, diperlukan kajian terhadap penelitian-penelitian terdahulu yang relevan. Kajian ini bertujuan untuk mengidentifikasi metode yang umum digunakan, capaian kinerja yang dihasilkan, serta celah penelitian yang masih dapat dikembangkan. Oleh karena itu, pada sub bab ini disajikan ringkasan beberapa penelitian terdahulu yang berkaitan dengan analisis sentimen menggunakan pendekatan *machine learning* dan *transformer*, sebagaimana dirangkum dalam Tabel 2.1.

Tabel 2.1 Tabel Jurnal Penelitian Terdahulu

No	Jurnal	Judul	Penulis	Metode	Hasil
1	International Journal of Advanced Computer Science and Applications, Vol. 14, No. 8, 2023	Sentiment Analysis in Indonesian Healthcare Applications using IndoBERT Approach	Helmi Imaduddin, Fiddin Yusfida A'la, dan Yusuf Sulistyo Nugroho [13]	IndoBERT	Penerapan model analisis sentimen berbasis IndoBERT pada ulasan aplikasi layanan kesehatan yang dilabeli secara manual dan mencapai performa dengan akurasi 96%.

No	Jurnal	Judul	Penulis	Metode	Hasil
2	Journal of Software Engineering, Information and Communication Technology (SEICT), Vol. 6, No. 1, 2025	Sentiment Analysis of Tourist Reviews at Waterfront City Pangururan Using Naive Bayes and TF-IDF Algorithm	Ichwanul Muslim Karo Karo, Justaman Arifin Karo Karo, Selvia Djasmayena, Rizki Wahyudi [14]	Naive Bayes + TF-IDF	Analisis sentimen ulasan wisata Waterfront City Pangururan menggunakan Naive Bayes dan TF-IDF mencapai akurasi 90.48%.
3	Journal of Applied Informatics and Computing (JAIC) Vol. 9, No.3, June 2025, pp. 851~862	Sentiment Analysis on BRImo Application Reviews Using IndoBERT	Asyer Aprinando Pratama Simarmata dan Theopilus Bayu Sasongko [15]	IndoBERT	Model IndoBERT menghasilkan 90% akurasi dalam analisis sentimen aplikasi BRImo dengan nilai F1-score 0.89 untuk negatif, 0.91 untuk netral, dan 0.90 untuk positif.
4	Jurnal Rekayasa Sistem dan Teknologi Informasi Vol. 7, No. 2, 2024: 389-396	Sentiment Analysis of Electricity Company Service Quality Using Naïve Bayes	Yuli Astuti, Yova Ruldeviyani, Faris Salbari, dan Aldiansah Prayogi [16]	Naive Bayes	Pelabelan manual sebagai <i>ground truth</i> dan penggunaan Naïve Bayes mampu mengklasifikasikan sentimen pengguna PLNMobile dengan akurasi 73%.

No	Jurnal	Judul	Penulis	Metode	Hasil
5	Jurnal Kecerdasan Buatan dan Teknologi Informasi Vol. 3, No. 2, Mei 2024, hlm. 68-74	Analisis Sentimen terhadap Layanan Nasabah Bank Menggunakan Teknik Klasifikasi Naïve Bayes	Putri Puspa Wulan dan Hasan Basri [17]	Naïve Bayes	Metode Naïve Bayes digunakan untuk mengklasifikasikan sentimen positif dan negatif layanan nasabah bank dengan akurasi model mencapai 76.32%.
6	INTEK Jurnal Penelitian Vizualitation Vol. 7 No. 2. June 2023	Sentimen Analysis Social Media for Disaster using Naïve Bayes and IndoBERT	Sri Mulyani Anugerah, Rifki Wijaya, Moch Arif Bijaksana [18]	Naïve Bayes dan IndoBERT	Penerapan IndoBERT memberikan akurasi tertinggi sebesar 91% dan Naïve Bayes mencapai 74% dalam klasifikasi sentimen tweet bencana yang telah dilabeli secara manual.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

No	Jurnal	Judul	Penulis	Metode	Hasil
7	ILKOM Jurnal Ilmiah Vol. 13, No. 2, August 2021: 101- 109	Sentiment analysis of game product on shopee using the TF-IDF method and naive bayes classifier	Rifki Kosasih dan Anggi Alberto [19]	TF-IDF Naive Bayes	Pembobotan kata menggunakan TF- IDF dan algoritma klasifikasi Naive Bayes mampu menghasilkan akurasi 80.22% dalam mengidentifikasi sentimen dari 1.000 ulasan pengguna.
8	ICIC Express Letters Volume 16, Number 3, March 2022	IndoBERT for Indonesian Fake News Detection	Sani Muhamad Isa, Gary Nico, dan Mikhael Permana [11]	IndoBERT	IndoBERT untuk deteksi berita palsu dengan melakukan <i>fine-tuning</i> serta penyesuaian <i>hyper parameter</i> yang menghasilkan akurasi sebesar 94,66%.
9	MALCOM: Indonesian Journal of Machine Learning and Computer Science Vol. 5 Iss. 3 July 2025	Analisis Sentimen Coretax: Perbandingan Pelabelan Data Manual, Transformer- Based, dan Lexicon-Based pada Performa IndoBERT	Agnia Suci Rizkia, Wufron, dan Fikri Fahru Roji [8]	IndoBERT	Mengevaluasi lima metode pelabelan data terhadap performa model IndoBERT dalam klasifikasi sentimen isu Coretax dengan hasil bahwa IndoBERTweet mencapai F1-Score tertinggi sebesar 0,9802 namun

No	Jurnal	Judul	Penulis	Metode	Hasil
					mengalami <i>overfitting.</i>
10	Jurnal Nasional Teknik Elektro dan Teknologi Informasi Volume 13 Number 1, February 2024	Research and Analysis of IndoBERT Hyperparameter Tuning in Fake News Detection	Anugerah Simanjuntak, Rosni Lumbantoruan, Kartika Sianipar, Rut Gultom, Mario Simaremare, Samuel Situmeang, dan Erwin Panggabean [20]	IndoBERT	Bayesian Optimazation sebagai <i>hyperparameter</i> <i>tuning</i> model IndoBERT memiliki pendekatan terbaik dengan <i>precision</i> 88.79%, <i>recall</i> 94.5%, dan <i>F1-</i> <i>score</i> 91.56% untuk deteksi berita palsu dengan

Penelitian terdahulu menunjukkan bahwa teknik pelabelan manual masih menjadi pendekatan yang dominan dalam menghasilkan *ground truth* yang akurat pada analisis sentimen berbahasa Indonesia. Beberapa studi, seperti yang dilakukan oleh Astuti *et al.* pada ulasan aplikasi PLNMobile [14] serta penelitian Coretax oleh Rizkia, Wufron, dan Roji [8], menegaskan bahwa pelabelan manual mampu menghasilkan distribusi kelas yang lebih seimbang serta penilaian konteks sentimen yang lebih tepat dibandingkan metode pelabelan otomatis atau berbasis leksikon. Pendekatan pelabelan manual juga digunakan pada berbagai penelitian lain, termasuk analisis sentimen aplikasi layanan kesehatan berbasis IndoBERT [11] serta studi analisis sentimen bencana alam menggunakan kombinasi Naive Bayes dan IndoBERT [16],

yang sama-sama menekankan pentingnya kualitas anotasi data untuk mencapai performa model yang optimal.

Pada sisi metode, algoritma **Naive Bayes** masih banyak digunakan sebagai pendekatan *machine learning* konvensional dalam klasifikasi sentimen teks. Penelitian Astuti *et al.* (2024) pada layanan PLNMobile [14] mencatat tingkat akurasi sebesar 73% menggunakan pendekatan TF-IDF Naive Bayes, sementara Wulan dan Basri (2024) [15] melaporkan akurasi 76,32% dalam analisis sentimen layanan nasabah bank. Selain itu, Kosasih dan Alberto (2021) menunjukkan bahwa kombinasi TF-IDF dan Naive Bayes mampu menghasilkan akurasi sebesar 80,22% dalam analisis sentimen ulasan produk gim di Shopee [17]. Studi internasional oleh Karo Karo *et al.* (2025) juga membuktikan bahwa metode TF-IDF dan Naive Bayes masih relevan dan efektif sebagai model *baseline* dengan capaian akurasi 90,48% pada analisis sentimen ulasan wisata [12]. Temuan ini memperkuat posisi Naive Bayes sebagai pendekatan pembanding yang valid sebelum penerapan model berbasis *transformer*.

Seiring perkembangan *deep learning*, model transformer seperti IndoBERT semakin banyak digunakan dalam analisis sentimen berbahasa Indonesia. Berbagai penelitian melaporkan performa tinggi IndoBERT, di antaranya akurasi 96% pada ulasan aplikasi layanan kesehatan [11], akurasi 90% pada analisis sentimen aplikasi BRImo [13], serta akurasi 91% pada klasifikasi sentimen bencana alam [16]. Penelitian lain oleh Isa *et al.* [10] dan Simanjuntak *et al.* [18] juga menunjukkan bahwa IndoBERT memiliki kemampuan unggul dalam memahami konteks bahasa Indonesia secara bidirectional dibandingkan pendekatan tradisional berbasis fitur. Selain itu, penelitian Coretax [8] menegaskan bahwa performa IndoBERT sangat dipengaruhi oleh kualitas pelabelan data, sehingga pelabelan manual memberikan kontribusi signifikan terhadap hasil klasifikasi yang optimal.

Dalam konteks perbankan, sebagian besar penelitian analisis sentimen masih berfokus pada layanan digital perbankan atau aplikasi *mobile*, seperti BRImo [13], dibandingkan pengalaman layanan langsung di kantor cabang. Mayoritas studi tersebut menggunakan data dari Google Play Store atau media sosial, bukan data *customer experience* yang dikumpulkan langsung dari nasabah di KCU/KCP. Selain itu, penelitian-penelitian tersebut umumnya hanya menerapkan satu jenis metode tanpa melakukan perbandingan langsung antara model klasik berbasis TF-IDF Naive Bayes dan model *transformer* seperti IndoBERT. Oleh karena itu, penelitian yang mengombinasikan pelabelan manual, pendekatan TF-IDF Naive Bayes sebagai *baseline*, dan IndoBERT sebagai model lanjutan dalam analisis sentimen pengalaman layanan perbankan di kantor cabang masih sangat terbatas sehingga penelitian ini diharapkan dapat mengisi celah penelitian tersebut.

2.2 Teori yang berkaitan

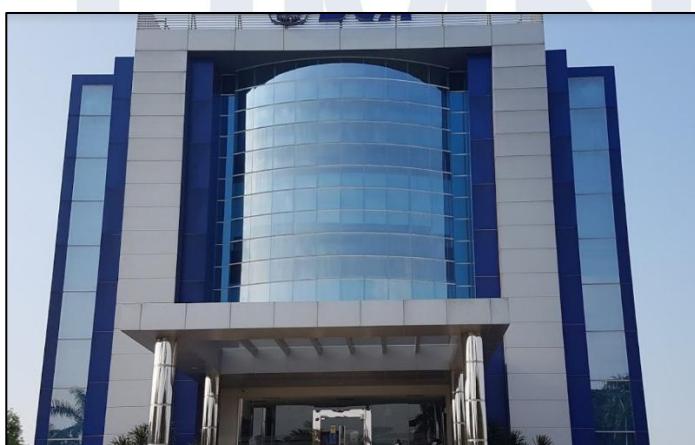
2.2.1 Analisis Sentimen

Analisis sentimen atau *sentiment analysis* merupakan sebuah bagian dari pengolahan bahasa alami (Natural Language Processing-NLP) yang memiliki tujuan untuk mengidentifikasi dan mengklasifikasikan opini atau perasaan seseorang terhadap sesuatu dalam bentuk teks [21]. Hal ini dapat berupa ulasan, komentar, maupun tanggapan seseorang. Melalui pendekatan ini, sistem mampu mengekstraksi makna emosional dari teks yang bersumber dari berbagai media, seperti ulasan pelanggan, komentar di media sosial, maupun hasil survei. Secara umum, analisis sentimen dikategorikan menjadi tiga polaritas utama, yaitu positif, negatif, dan netral [22]. Dalam praktiknya, analisis sentimen menggunakan data teks harus melalui beberapa tahapan, mulai dari *preprocessing* (pembersihan data), ekstraksi fitur, hingga klasifikasi berdasarkan kategori emosional tersebut [23].

Seiring berkembangnya penelitian di bidang analisis sentimen, pendekatan pelabelan data juga mengalami peningkatan dari metode sederhana menuju proses yang lebih terstruktur dan akurat. Pada tahap awal, pelabelan sentimen sering dilakukan secara manual oleh *annotator* untuk menentukan apakah suatu teks bersentimen positif, negatif, atau netral. Metode manual *labeling* memberikan kualitas pelabelan yang baik karena mengandalkan pemahaman manusia terhadap konteks, ironi, dan nuansa bahasa yang tidak dapat ditangkap oleh aturan sederhana. Namun, proses ini membutuhkan waktu, tenaga, serta konsistensi antar-*annotator*. Sebaliknya, pendekatan berbasis *machine learning* mampu menyesuaikan diri dengan variasi bahasa dan konteks karena model belajar langsung dari data yang tersedia [24]. Dengan dukungan jumlah data yang besar serta peningkatan kemampuan komputasi, metode ini memberikan hasil analisis yang lebih akurat dan dapat diandalkan untuk memahami opini publik.

2.2.2 KCU dan KCP Bank

Kantor Cabang Utama (KCU) dan Kantor Cabang Pembantu (KCP) merupakan bagian penting dari perusahaan perbankan yang berfungsi untuk memperluas jangkauan layanan kepada para nasabah.



Gambar 2.1 Kantor Cabang Bank

Sumber: [25]

KCU berperan sebagai pusat koordinasi kegiatan operasional di wilayah tertentu dan biasanya berlokasi di ibu kota provinsi atau kota besar [26]. KCU sendiri memiliki wewenang strategis dalam pengambilan keputusan penting, pengawasan kinerja cabang, serta pelaporan ke kantor pusat [27]. Sementara itu, KCP berada di bawah koordinasi KCU dan berfungsi untuk melayani masyarakat di wilayah yang lebih kecil atau jauh dari pusat kota. Keberadaan KCP memiliki peran sebagai pendukung kegiatan perbankan yang tidak berkewenangan penuh dalam pengambilan keputusan, seperti manajemen risiko ataupun peminjaman kredit dalam jumlah besar [27].

2.2.3 Customer Experience

Customer Experience merupakan pengalaman subjektif yang dirasakan seseorang terhadap suatu produk atau merek yang terbentuk melalui berbagai interaksi antara konsumen dengan perusahaan [28]. Pengalaman ini mencakup reaksi kognitif, emosional, dan perilaku individu terhadap interaksi mereka dengan suatu perusahaan atau layanan yang dilalui melalui berbagai titik kontak (*touchpoints*) sepanjang perjalanan pelanggan (*customer journey*). Sebagai contoh, penelitian literatur pada konteks *mobile banking* menemukan bahwa Customer Experience adalah konsep multidimensional yang dipengaruhi oleh berbagai aspek, seperti antarmuka digital, kecepatan layanan, personalisasi, dan kemudahan akses sehingga pada akhirnya berkontribusi terhadap loyalitas dan kepuasan pelanggan [29]. Dengan demikian, bank yang mampu mengelola pengalaman nasabah secara efektif, mulai dari kualitas layanan di kantor cabang, interaksi digital, hingga respons terhadap umpan balik akan memiliki keunggulan kompetitif dalam mempertahankan nasabah dan mendorong hubungan secara jangka panjang.

2.2.4 Text Mining

Text mining merupakan proses untuk mengeksplorasi informasi berharga dan pola tersembunyi dari data berbentuk teks yang tidak terstruktur [30]. Data teks tersebut umumnya bersumber dari artikel, ulasan pelanggan, media sosial, aplikasi, dan lain-lain sehingga berbeda dengan data numerik yang terorganisir dalam tabel. *Text mining* menggabungkan teknik Natural Language Processing (NLP), linguistik, dan *machine learning* untuk memahami isi konteks di dalam teks secara komprehensif [31]. Proses *text mining* mencakup beberapa tahapan, mulai dari *text preprocessing*, *feature extraction*, dan teknik klasifikasi. Tujuan akhirnya ialah untuk memperoleh *insight* atau pengetahuan yang belum ditemukan sebelumnya dalam mendukung keputusan bisnis [32].

2.2.5 Text Preprocessing

Text preprocessing adalah tahap awal yang sangat penting dalam proses *text mining* karena kualitas hasil analisis akan bergantung pada seberapa baik teks mentah dipersiapkan sebelum masuk ke tahap analisis lebih lanjut [33]. Data teks yang dikumpulkan dari sumber tertentu biasanya mengandung banyak *noise*, misalnya tanda baca, emotikon, singkatan, serta kata-kata yang kurang relevan. Oleh karena itu, tujuan utama dari *text preprocessing* adalah untuk menstrukturkan teks mentah menjadi bentuk yang siap untuk dianalisis secara komputasional. Berikut langkah-langkah dalam *text preprocessing*:

1) Tokenization

Tokenization merupakan proses pemecahan teks menjadi token atau unit-unit kecil yang bermakna [34]. Token dapat berbentuk kata, frasa, ataupun kalimat tergantung pendekatan yang digunakan.

2) Case Folding

Case Folding adalah suatu tahap untuk mengubah seluruh huruf atau teks yang ada menjadi bentuk huruf kecil [35]. Hal ini dilakukan agar tidak terjadi perbedaan antara kata yang sama. .

3) Stopword Removal

Stopword removal ialah sebuah teknik untuk menghapus kata-kata umum yang tidak memiliki makna signifikan, biasanya seperti kata penghubung [40]. Sebab kata tersebut tidak akan memberikan informasi penting terhadap konteks analisis.

4) Remove Punctuation

Remove punctuation adalah salah satu tahap dalam proses *text preprocessing* yang bertujuan untuk menghapus tanda baca seperti titik (.), koma (,), tanda tanya (?), tanda seru (!), dan simbol lainnya dari teks [40].

5) Stemming dan Lemmatization

Stemming dan *lemmatization* sasama-sama bertujuan untuk mengembalikan kata-kata ke dalam bentuk dasarnya [40]. *Stemming* akan memotong akhiran kata (imbuhan kata) sedangkan *lemmatization* mempertimbangkan konteks linguistik dan tata bahasa untuk menghasilkan kata dasar yang lebih akurat.

2.2.6 TF-IDF

Term Frequency–Inverse Document Frequency (TF-IDF) adalah metode pengukuran bobot yang digunakan dalam pengolahan teks untuk menentukan seberapa penting suatu kata dalam sebuah dokumen dibandingkan keseluruhan kumpulan dokumen atau *corpus* [36]. Nilai TF-IDF akan meningkat apabila sebuah kata sering muncul pada dokumen tertentu (tinggi TF), tetapi jarang sekali muncul di dokumen secara keseluruhan (tinggi IDF) [37]. Dengan begitu, kata-kata yang

umum atau sering muncul di banyak dokumen memperoleh bobot yang lebih kecil. Berikut rumus perhitungan dari TF-IDF.

1.) Term Frequency (TF)

$$tf = 0.5 + 0.5 \frac{tf}{\max (tf)}$$

Rumus 2.1 Rumus Perhitungan TF [38]

Diketahui:

tf = Banyaknya data yang muncul di dalam suatu dokumen

$\max (tf)$ = Panjang kata pada suatu dokumen

2.) Inverse Document Frequency (IDF)

$$idf = \ln \frac{N}{df} + 1$$

Rumus 2.2 Rumus Perhitungan IDF [38]

Diketahui:

ln = Logaritma natural

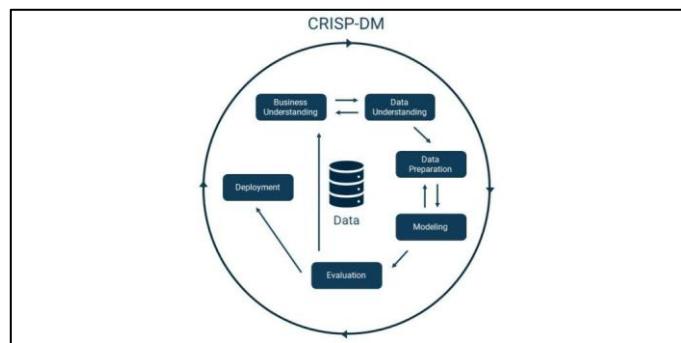
N = Jumlah seluruh dokumen

df = Jumlah data di dalam dokumen

2.3 Framework/Algoritma yang digunakan

2.3.1 CRISP-DM

CRISP-DM atau Cross Industry Standard Process for Data Mining merupakan suatu kerangka kerja yang menjadi standar industri dalam sebuah pelaksanaan proyek *data mining* [39]. Metodologi ini dikembangkan untuk memberikan tahapan sistematis dalam mengelola proyek analisis data yang dimulai dari tahap awal hingga implementasi hasil.



Gambar 2.2 Metode CRISP-DM

Sumber: [40]

Metodologi ini dikembangkan oleh konsorsium perusahaan pada akhir tahun 1990-an sebagai alternatif dari model Knowledge Discovery in Databases (KDD) [41]. Metodologi ini tidak hanya menjadi pengganti, tetapi juga merupakan hasil pengembangan dan penyempurnaan dari model KDD beserta berbagai ekstensi yang telah ada sebelumnya. CRISP-DM menjadi salah satu kerangka kerja paling populer karena memiliki sifat yang fleksibel dan dapat diterapkan pada berbagai domain, seperti keuangan, kesehatan, telekomunikasi, dan pemasaran. Tujuan utama dari CRISP-DM adalah menyediakan panduan umum dalam memahami data, membangun model, serta menghasilkan pengetahuan yang dapat digunakan untuk mendukung pengambilan keputusan berbasis data. CRISP-DM sendiri terdiri dari enam tahapan utama, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment* yang dijelaskan sebagai berikut:

1) Business Understanding

Tahap ini menjadi fase awal dalam metodologi CRISP-DM yang berfokus pada pemahaman konteks bisnis serta tujuan utama dari proyek atau proses *data mining* [42]. Pada tahap ini, peneliti perlu memahami kebutuhan organisasi, permasalahan yang ingin diselesaikan, serta nilai bisnis yang diharapkan dari hasil analisis data. Pemahaman yang baik

terhadap tujuan bisnis sangat penting agar proses *data mining* menghasilkan *insight*.

2) Data Understanding

Tahap kedua dalam metode CRISP-DM ialah *data understanding*, yakni fase kedua dalam metodologi CRISP-DM yang berfokus pada proses pengumpulan dan pemahaman awal terhadap data yang akan digunakan dalam proyek *data mining* [42]. Pada tahap ini, peneliti melakukan pengumpulan data awal, eksplorasi karakteristik data, identifikasi kualitas data, serta deteksi potensi masalah seperti data hilang, duplikasi, ataupun anomali.

3) Data Preparation

Kemudian, *data preparation* berfokus pada proses pengolahan data mentah agar siap digunakan dalam pemodelan [42]. Pada tahap ini, dilakukan berbagai aktivitas, mulai dari seleksi data, pembersihan data (*data cleaning*), transformasi data, serta integrasi dari berbagai sumber data untuk memastikan kualitas data yang optimal. Tujuan utamanya adalah menghasilkan sebuah data yang terstruktur dari data mentah sehingga siap dikonsumsi model *data mining*.

4) Modeling

Tahap *modeling* berfokus pada pembangunan dan penerapan algoritma model analisis menggunakan data yang telah dipersiapkan sebelumnya [42]. Tahap ini memanfaatkan teknik pemodelan statistik, *machine learning*, atau *data mining* yang diterapkan untuk menemukan pola, hubungan, atau prediksi yang sesuai dengan tujuan bisnis. Kesesuaian pemilihan algoritma akan menjadi penentu dalam memastikan hasil yang optimal.

5) Evaluation

Tahap *evaluation* merupakan fase kelima dalam metodologi CRISP-DM yang bertujuan untuk menilai kualitas serta validitas model yang telah dibangun pada tahap Modeling [42]. Model akan dilakukan evaluasi melalui beberapa metriks penilaian, seperti akurasi, presisi, *recall*, *error rate*, F1-Score, dan lain-lain. Evaluation akan melihat dari sejauh mana model tersebut mampu menjawab tujuan bisnis yang telah ditetapkan di awal proyek.

6) Deployment

Tahap terakhir dalam metode CRISP-DM adalah *deployment* di mana model yang telah dievaluasi dan disetujui diterapkan ke lingkungan operasional yang sebenarnya [42]. Tujuan dari tahap ini untuk memastikan bahwa hasil dari proyek *data mining* dapat memberikan manfaat nyata bagi pengambilan keputusan bisnis. Implementasi model dapat dilakukan dalam berbagai bentuk, seperti aplikasi, *website*, *dashboard*, dan lain sebagainya.

2.3.2 Naive Bayes

Naive Bayes adalah algoritma klasifikasi *machine learning* berbasis probabilistik yang didasarkan pada Teorema Bayes dengan asumsi bahwa fitur-fitur yang ada dalam data bersifat independen satu sama lain yang dikenal dengan prinsip *conditional independence* [43]. Meskipun asumsi ini sering kali tidak sepenuhnya tepat dalam praktik, Naive Bayes tetap menunjukkan kinerja yang baik dalam berbagai aplikasi, terutama dalam analisis teks dan klasifikasi data dengan dimensi tinggi, seperti dalam analisis sentimen. Algoritma ini menghitung probabilitas posterior dari setiap kelas berdasarkan probabilitas kondisi dari setiap fitur dalam data, kemudian memilih kelas dengan probabilitas tertinggi [44]. Salah satu keunggulan utama dari Naive Bayes adalah

kemampuannya untuk menangani data yang tidak seimbang dengan efisien serta kesederhanaannya yang memungkinkan implementasi baik pada *dataset* besar. Berikut rumus perhitungan dari Teorema Bayes.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Rumus 2.3 Rumus Perhitungan Teorema Bayes [45]

Diketahui:

h = Hipotesis data dengan suatu *class*

D = Data yang belum memiliki *class*

$P(h)$ = Probabilitas hipotesis (*prior probability*)

$P(D)$ = Probabilitas dari D

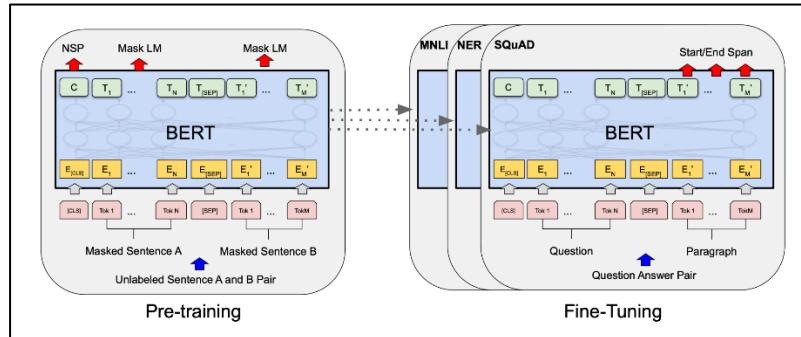
$P(h|D)$ = Probabilitas h berdasarkan kondisi D (*posterior probability*)

$P(D|h)$ = Probabilitas D berdasarkan kondisi pada hipotesis h (*posterior probability*)

2.3.3 BERT

BERT atau Bidirectional Encoder Representations from Transformers merupakan sebuah model representasi bahasa yang dikembangkan oleh Google AI Language pada tahun 2018. BERT berbasis arsitektur *transformer* dan dilatih secara *pre-training* pada korpus teks besar tanpa pengawasan [46]. Model ini dirancang untuk memproses konteks secara dua arah (*bidirectional*), artinya setiap kata dalam suatu kalimat dipahami dengan mempertimbangkan konteks dari kiri dan kanan. Hal ini menjadi sebuah perkembangan signifikan jika dibandingkan dengan model *transformer* sebelumnya yang cenderung memahami hanya ke satu arah. BERT sendiri dilatih menggunakan dua tugas utama, yaitu Masked Language Modeling (MLM) dan Next Sentence Prediction (NSP). Melalui dua proses ini, model belajar memahami hubungan semantik antarkata dan antar kalimat tanpa memerlukan data berlabel. Konfigurasi arsitektur BERT terdiri dari 12 *encoder layers*, 12 *self-attention heads*, 768 *dimensional hidden size*, dan

110 juta parameter [47]. Secara umum, BERT memiliki dua tahap utama sebagai berikut:



Gambar 2.3 Arsitektur BERT

Sumber: [48]

1) Pre-trained BERT

Pre-trained BERT adalah model BERT yang telah dilatih sebelumnya menggunakan korpus teks berskala besar dengan tujuan untuk memahami konteks dan struktur bahasa secara umum [49]. BERT pada tahap ini tidak dilatih untuk tugas tertentu, seperti klasifikasi ataupun sentimen analisis sehingga berfokus pada pengenalan hubungan antar kata dan kalimat. Proses dilakukan dengan dua mekanisme, yaitu MLM dan NSP.

2) Fine-Tuned BERT

Fine-tuned BERT merupakan tahap lanjutan dan penyesuaian dari *pre-trained* BERT untuk melakukan tugas-tugas spesifik (*task-specific*) dalam NLP, seperti analisis sentimen, klasifikasi teks, *question answering*, dan lain-lain [50]. Model BERT yang sudah dilatih sebelumnya tidak dilatih ulang dari awal, tetapi ditambahkan lapisan (*layer*) tambahan di bagian *output*, kemudian dilatih ulang menggunakan *dataset* baru sesuai dengan permasalahan yang ingin diselesaikan.

2.3.4 IndoBERT

IndoBERT merupakan model bahasa berbasis *transformer* yang dikembangkan secara khusus untuk memahami teks berbahasa Indonesia [51]. Secara arsitektur, IndoBERT memiliki struktur yang sama dengan BERT, tetapi yang membedakan keduanya terletak pada tahap *pre-training* saat *dataset* yang digunakan untuk melatih model. *Dataset* yang digunakan dalam pelatihan IndoBERT dikenal dengan nama Indo4B yang berisi sekitar 4 miliar kata dan 250 juta kalimat berbahasa Indonesia [52]. Adapun jenis-jenis IndoBERT, seperti IndoBERT-Base, IndoBERT-Lite, IndoBERTweet, IndoBERT-Base-P1, dan IndoBERT-Base-P2. Dengan jumlah data yang sangat besar dan beragam, IndoBERT mampu menangkap struktur linguistik, semantik, serta nuansa khas Bahasa Indonesia, seperti penggunaan imbuhan, kata serapan, maupun gaya bahasa informal.

2.3.5 Manual Labeling

Manual labeling atau pelabelan manual adalah proses pemberian label atau kategori pada data oleh manusia yang bertujuan untuk menghasilkan *ground truth* yang akurat dan dapat dipercaya [8]. Hal ini digunakan sebagai referensi untuk melatih model *machine learning*. Proses ini sering diterapkan dalam analisis sentimen, di mana *annotator* akan memberikan label seperti positif, negatif, ataupun netral pada teks yang ada, berdasarkan pemahaman kontekstual. Keakuratan *manual labeling* sangat bergantung pada konsistensi dan ketelitian *annotator* dalam menentukan konteks kalimat dan makna yang terkandung. Meskipun lebih memakan waktu dan biaya dibandingkan dengan metode pelabelan otomatis, *manual labeling* memberikan keuntungan berupa data yang lebih akurat dalam membangun model yang lebih baik [53].

2.3.6 Hyperparameter Tuning

Hyperparameter tuning adalah sebuah proses penyesuaian nilai parameter yang tidak dipelajari secara langsung oleh model, tetapi memiliki pengaruh besar terhadap performa dan hasil akhir yang optimal dari proses model pembelajaran mesin [54]. *Hyperparameter* ini ditetapkan sebelum proses pelatihan (training) dimulai, berbeda dengan parameter model bobot dan bias yang dilakukan selama proses *training*. Pemilihan nilai *hyperparameter* yang kurang tepat dapat berdampak negatif terhadap kinerja model yang digunakan. Jika nilai yang dipilih terlalu ekstrem atau tidak seimbang, model dapat mengalami *overfitting* sedangkan jika nilai *hyperparameter* terlalu sederhana, maka model bisa mengalami *underfitting* [55]. Contoh *hyperparameter* ialah *learning rate*, *batch size*, jumlah *epoch*, jumlah *layer*, *dropout rate*, jumlah *neuron* pada *hidden layer*, dan lain sebagainya.

2.3.7 GridSearchCV

Grid Search Cross Validation merujuk pada teknik yang menggabungkan metode Grid Search dan Cross-Validation, di mana proses ini melibatkan pengujian berbagai kombinasi model dan *hyperparameter* satu per satu, serta melakukan validasi untuk setiap kombinasi yang diuji [56]. Teknik ini sangat berguna untuk menemukan kombinasi parameter yang menghasilkan performa terbaik untuk model. Selama pencarian, GridSearchCV akan menguji setiap kemungkinan kombinasi parameter dan menghitung skor model menggunakan teknik *cross-validation* sehingga membantu menghindari *overfitting* dan memberikan gambaran lebih baik tentang generalisasi model. GridSearchCV mengembalikan parameter terbaik, skor terbaik, serta estimator yang telah disesuaikan (*refitted estimator*) dengan parameter optimal yang ditemukan.

2.3.8 Bayesian Optimization

Salah satu metode *hyperparameter tuning* ialah Bayesian Optimization. Metode ini bekerja dengan pendekatan probabilistik untuk mencari kombinasi nilai *hyperparameter* terbaik secara lebih efisien dibandingkan dengan metode pencarian acak (*random search*) ataupun pencarian *grid* (*grid search*) [57]. Bayesian Optimization membangun model *surrogate* yang biasanya menggunakan Gaussian Process (GP) dan *acquisition function* untuk memperkirakan hubungan antara nilai *hyperparameter* dan performa model [58]. Model *surrogate* akan disesuaikan berdasarkan seluruh hasil observasi terhadap fungsi target yang telah dilakukan sebelumnya. Kemudian, *acquisition function* akan mencari kombinasi parameter yang paling berpotensi meningkatkan hasil pencarian dan memfokuskan proses pencarian pada area tersebut untuk menemukan *set hyperparameter* terbaik.

2.3.9 Confusion Matrix

		Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)	
	False Negatives (FNs)	True Negatives (TNs)	
Predicted Negative (0)			

Gambar 2.4 Confusion Matrix

Sumber: [59]

Confusion matrix adalah salah satu metode evaluasi kinerja model klasifikasi yang paling umum digunakan dalam teknik *machine learning*. Matriks ini berbentuk tabel dua dimensi yang menggambarkan perbandingan antara hasil prediksi model dengan label sebenarnya sehingga mampu memberikan gambaran mengenai seberapa baik model

dalam mengklasifikasikan setiap kelas [60]. Terdapat empat komponen dalam *confusion matrix* ialah sebagai berikut:

- 1) True Positive (TP): Jumlah data positif yang berhasil diprediksi dengan benar sebagai positif.
- 2) True Negative (TN): Jumlah data negatif yang berhasil diprediksi dengan benar sebagai negatif.
- 3) False Positive (FP): Jumlah data negatif yang salah diprediksi sebagai positif.
- 4) False Negative (FN): Jumlah data positif yang salah diprediksi sebagai negatif.

Berdasarkan keempat komponen tersebut, maka berbagai metriks evaluasi dapat dihitung untuk menilai performa suatu model. Metriks ini mencakup nilai akurasi, presisi (*precision*), *recall*, dan F1-Score [61].

1.) Accuracy

Akurasi mengukur seberapa banyak model terprediksi benar dibandingkan dengan seluruh jumlah data [61]. Nilai akurasi tinggi menunjukkan model memiliki performa baik secara keseluruhan, namun metrik ini kurang tepat apabila data memiliki distribusi kelas yang tidak seimbang. Berikut merupakan rumus akurasi:

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

Rumus 2.4 Perhitungan Accuracy [61]

2.) Precision

Precision menunjukkan proporsi data yang benar-benar positif dari seluruh data yang diprediksi positif [60]. Metrik ini penting ketika kesalahan positif palsu (*false positive*) harus diminimalkan. Berikut rumus perhitungan *precision*:

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

Rumus 2.5 Perhitungan Precision [60]

3.) Recall

Recall mengukur kemampuan model dalam menemukan seluruh data positif yang sebenarnya ada [60]. Metrik ini penting ketika kesalahan negatif palsu (*false negative*) harus diminimalkan. Berikut rumus perhitungan *recall*:

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

Rumus 2.6 Perhitungan Recall [60]

4.) F1-Score

F1-Score merupakan rata-rata harmonik antara *precision* dan *recall* dengan memberikan keseimbangan antara keduanya [60]. Nilai F1 tinggi menunjukkan bahwa model memiliki performa baik dalam mengenali data positif tanpa terlalu banyak kesalahan. Berikut rumus perhitungan F1-Score:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Rumus 2.7 Perhitungan F1-Score [60]

2.3.10 Model Waterfall

Model Waterfall merupakan salah satu model pengembangan sistem informasi yang bersifat sekuensial dan terstruktur, di mana setiap tahapan pengembangan dilakukan secara berurutan dan sistematis, dimulai dari tahap analisis kebutuhan, perancangan sistem,

implementasi, pengujian, hingga tahap *deployment* dan pemeliharaan [62]. Setiap fase dalam model Waterfall harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase berikutnya, sehingga memungkinkan proses pengembangan yang terdokumentasi dengan baik dan mudah dikontrol. Pendekatan ini umumnya digunakan pada sistem yang memiliki kebutuhan yang jelas dan relatif stabil sejak awal pengembangan, sehingga sesuai untuk pengembangan sistem informasi berbasis penelitian yang berfokus pada kejelasan alur kerja dan evaluasi hasil sistem secara terstruktur.

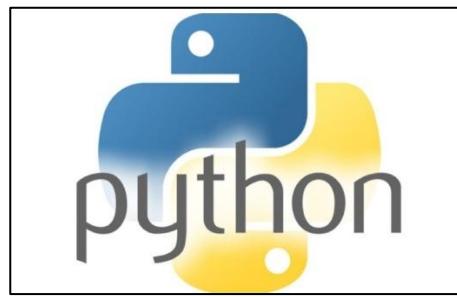
2.3.11 Evaluasi Black Box

Black box testing merupakan salah satu teknik pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas sistem berdasarkan masukan (*input*) dan keluaran (*output*) tanpa memperhatikan struktur internal atau logika kode program [63]. Pendekatan ini bertujuan untuk memastikan bahwa sistem telah berjalan sesuai dengan kebutuhan pengguna dan spesifikasi fungsional yang telah ditetapkan. *Black box testing* efektif digunakan dalam pengujian aplikasi berbasis web karena mampu memvalidasi perilaku sistem dari sudut pandang pengguna akhir (*end-user*), sehingga sangat relevan untuk aplikasi yang dikembangkan untuk kebutuhan bisnis.

2.4 Tools/Software yang Digunakan

2.4.1 Python

Dalam pengembangan sistem dan pemodelan analitik, pemilihan bahasa pemrograman menjadi aspek penting yang menentukan efisiensi proses pengolahan data dan implementasi model. Oleh karena itu, pada subbab ini akan dibahas bahasa pemrograman Python sebagai salah satu teknologi yang digunakan dalam penelitian ini.



Gambar 2.5 Pemrograman Python

Sumber: [64]

Python adalah sebuah bahasa pemrograman tingkat tinggi yang dirilis oleh Guido Van Rossum pada tahun 1991 [65]. Python memiliki sintaksis yang sederhana dan mudah dipahami dibandingkan bahasa pemrograman lain sehingga telah menjadi pilihan utama dalam bidang pengembangan *web*, *machine learning*, kecerdasan buatan (AI) dan lain-lain [66]. Selain itu, Python juga dilengkapi dengan berbagai *library* dan *framework* bawaan maupun eksternal, mulai dari NumPy, Pandas, Matplotlib, dan TensorFlow menjadikan Python sangat unggul dalam pemrosesan data. Python bersifat *open source* sehingga menciptakan komunitas pengguna yang sangat besar dan aktif di seluruh dunia.

2.4.2 Google Colab



Gambar 2.6 Google Colab

Sumber: [67]

Google Colab merupakan *platform* berbasis *cloud* yang disediakan oleh Google untuk menjalankan kode Python secara interaktif tanpa perlu melakukan instalasi perangkat lunak tambahan di komputer lokal [68]. Platform ini sangat populer di kalangan peneliti dan praktisi *data science*

karena menyediakan akses gratis ke sumber daya komputasi seperti CPU, GPU, dan TPU yang memungkinkan pengguna untuk melakukan pelatihan (*training*) model *machine learning* dan *deep learning* yang berukuran besar secara efisien. Sebagai layanan berbasis *cloud*, Google Colab menyimpan dan mengeksekusi notebook di lingkungan virtual Google Drive sehingga pengguna dapat berkolaborasi secara *real-time*.

2.4.3 Microsoft Excel

Microsoft Excel merupakan salah satu perangkat lunak pengolah data yang dikembangkan oleh Microsoft dengan fungsi sebagai sumber data (*data source*) dalam berbagai penelitian dan proyek analisis data. Excel menyediakan kemampuan untuk menyimpan, mengelola, serta memanipulasi data berupa perhitungan, penggunaan rumus, pembuatan grafik, dan lain-lain dalam format tabular yang mudah dipahami [69]. Dalam konteks *data mining* dan *machine learning*, Microsoft Excel sering digunakan sebagai sumber data awal karena formatnya (.xlsx atau .csv) mudah diintegrasikan dengan berbagai bahasa pemrograman, seperti Python.

2.4.4 Streamlit

Streamlit merupakan *framework* berbasis Python yang dirancang khusus untuk membangun aplikasi *web* interaktif dengan fokus utama pada kebutuhan *data science* [70]. Streamlit memungkinkan pengembang dan peneliti untuk mengubah *script* Python menjadi antarmuka *web* hanya dengan beberapa baris kode tanpa memerlukan keahlian dalam pengembangan *front-end* [71]. Streamlit kerap kali digunakan untuk mendemonstrasikan performa model, menampilkan hasil evaluasi, seperti *confusion matrix*, serta memudahkan pengguna melakukan *testing* data untuk *deployment* model. Framework ini juga mendukung berbagai pustaka Python, mulai dari *pandas*, *matplotlib*, *scikit-learn*, dan lain sebagainya.