

BAB 2 LANDASAN TEORI

2.1 Penelitian Terdahulu

Telah dilakukan tinjauan mendalam terhadap berbagai penelitian yang relevan dengan topik analisis sentimen pada data pasar modal dan penerapan metode *machine learning* serta *deep learning*. Tinjauan ini bertujuan untuk memetakan metode yang telah digunakan, kinerja yang dicapai, serta mengidentifikasi celah penelitian secara sistematis. Ringkasan, evaluasi, dan posisi penelitian ini terhadap penelitian terdahulu yang paling relevan disajikan pada Tabel 2.1 hingga 2.4.



Tabel 2.1. Perbandingan Penelitian Terdahulu dan Kontribusi Penelitian (1)

No.	Judul Penelitian	Peneliti (Tahun)	Metodologi	Hasil Utama	Evaluasi	Kontribusi Penelitian
1.	Analisis Sentimen Twitter Terhadap IHSG [17]	Ihsan Puntadewa (2024)	Naïve Bayes, TF-IDF	Mencapai akurasi 90.3% , presisi 90.3% , dan recall 95.9% . Ditemukan juga bahwa sentimen negatif berkorelasi dengan penurunan IHSG.	Belum menggunakan metode <i>deep learning</i> yang lebih kompleks dan tidak fokus pada periode krisis spesifik.	Menerapkan dan membandingkan metode <i>deep learning</i> (LSTM & IndoBERT) pada periode krisis pasar yang spesifik.
2.	Analisis Sentimen Tweet Masyarakat Terhadap Pemerintahan Jokowi Menggunakan Metode LSTM dan CNN [18]	Davina Febryanthi Kurniawan (2024)	LSTM, CNN (TF-IDF, Word Embedding, 7 Optimizers)	LSTM (WE+Adamax): Akurasi Pelatihan 91.43% , Validasi 91.21% , CR Akurasi 91.2% . CNN (WE+Adamax): Akurasi Pelatihan 98.23% , Validasi 91.47% , CR Akurasi 91.4% .	Penelitian ini menguji efektivitas LSTM dan CNN dengan berbagai fitur ekstraksi dan optimizer pada data sentimen pemerintahan Jokowi.	Penelitian ini berfokus pada sentimen publik terhadap anjloknya IHSG pada periode krisis spesifik (Maret-April 2025) dengan membandingkan Bi-LSTM (dari awal) dan IndoBERT (fine-tuning).
3.	Analisis Korelasi Harga Saham dengan Sentimen Twitter [19]	Muhammad Noer Ibnu Sina (2023)	CNN-LSTM	Model CNN-LSTM mencapai akurasi 77.75% dan F1-score 75.42% . Ditemukan korelasi kuat antara sentimen positif dengan harga saham.	Tidak berfokus pada IHSG dan tidak meneliti dalam konteks kejadian pasar spesifik seperti krisis.	Menerapkan analisis pada konteks krisis pasar yang spesifik (anjloknya IHSG) untuk melihat dinamika sentimen pada kondisi anomali.

Tabel 2.2. Perbandingan Penelitian Terdahulu dan Kontribusi Penelitian (2)

No.	Judul Penelitian	Peneliti (Tahun)	Metodologi	Hasil Utama	Evaluasi	Kontribusi Penelitian
4.	Dampak Sentimen Investor dan BI7DRR terhadap Prediksi IHSG [20]	I. F. Ainaya (2023)	Analisis Regresi	Sentimen investor dan suku bunga berpengaruh terhadap IHSG.	Tidak memanfaatkan data media sosial dan belum menggunakan pendekatan <i>machine/deep learning</i> .	Menggunakan data sentimen <i>real-time</i> dari Twitter sebagai proksi sentimen investor dan menerapkan model <i>deep learning</i> .
5.	Analisis Sentimen PSBB dan Pengaruhnya terhadap IHSG [21]	Fitri, dkk. (2022)	Regresi Logistik	Sentimen negatif terhadap kebijakan PSBB berdampak signifikan pada penurunan IHSG.	Belum menggunakan data dari media sosial secara langsung dan belum menerapkan metode LSTM.	Menggunakan data langsung dari Twitter dan secara spesifik mengevaluasi kinerja arsitektur LSTM dan Transformer.
6.	Pemanfaatan Analisis Sentimen Berbasis LSTM dalam Prediksi Harga Saham [22]	M. Fajrul Aslim (2023)	LSTM (Lexicon & TextBlob)	Pendekatan Lexicon memberikan akurasi lebih tinggi.	Tidak difokuskan pada IHSG dan tidak membahas dampak sentimen pada periode krisis pasar.	Menerapkan LSTM pada konteks IHSG selama periode krisis dan membandingkannya dengan arsitektur Transformer.

Tabel 2.3. Perbandingan Penelitian Terdahulu dan Kontribusi Penelitian (3)

No.	Judul Penelitian	Peneliti (Tahun)	Metodologi	Hasil Utama	Evaluasi	Kontribusi Penelitian
7.	Prediksi Saham dengan Bidirectional LSTM dan Analisis Sentimen [23]	Nurchayyo, dkk. (2023)	Bi-LSTM	Model Bi-LSTM yang digabungkan dengan data sentimen terbukti lebih unggul, mencapai nilai error (MAPE) 23.85% .	Terfokus pada saham individu (misal: BBCA), bukan indeks IHSG secara keseluruhan.	Mengambil pendekatan makro dengan menganalisis sentimen terhadap indeks IHSG, bukan saham per saham.
8.	Analyze News Effect on Trend Stock Price in Indonesia [24]	M. A. Satriaman (2023)	Bi-LSTM, Word2Vec	Model Bi-LSTM berhasil memprediksi tren harga saham dengan akurasi mencapai 80.03% .	Sumber data bukan dari opini publik di Twitter, melainkan sentimen dari berita (opini media) dan tidak spesifik pada momen krisis.	Menggunakan opini publik murni dari Twitter sebagai sumber data utama selama periode krisis yang spesifik.

Tabel 2.4. Perbandingan Penelitian Terdahulu dan Kontribusi Penelitian (4)

No.	Judul Penelitian	Peneliti (Tahun)	Metodologi	Hasil Utama	Evaluasi	Kontribusi Penelitian
9.	Pemanfaatan Analisis Sentimen dalam Prediksi Harga Saham: Studi pada investor retail Indonesia [25]	Christophorus Bintang Saputra (2023)	Bi-LSTM vs LSTM	Model Bi-LSTM terbukti lebih unggul dari LSTM, menghasilkan nilai error (MAPE) yang lebih rendah. Menambahkan data sentimen secara signifikan meningkatkan akurasi prediksi.	Sumber data dari YouTube dan cakupan penelitian hanya pada dua saham emiten (BBCA dan BBRI), bukan IHSG.	Menggunakan sumber data Twitter yang lebih dinamis untuk sentimen keuangan dan fokus pada indeks IHSG secara keseluruhan.
10.	Prediksi Harga Saham Telekomunikasi Indonesia Menggunakan Metode Bi-LSTM [26]	Siregar, dkk. (2022)	Bi-LSTM	Mencapai nilai error (RMSE) yang sangat rendah, yaitu 0.0051 .	Fokus pada prediksi harga satu saham (TLKM) berdasarkan data historis, belum mengintegrasikan data sentimen.	Mengintegrasikan data sentimen sebagai fitur tambahan dan membandingkan kinerja model dasar (LSTM) dengan arsitektur modern (IndoBERT).

Berdasarkan tinjauan pustaka yang telah dilakukan, berbagai penelitian sebelumnya telah mengeksplorasi analisis sentimen dan hubungannya dengan pasar modal. Metode yang umum digunakan meliputi model klasik seperti Naïve Bayes dan SVM, serta arsitektur *deep learning* seperti LSTM dan CNN, atau kombinasinya seperti CNN-LSTM. Akurasi yang dicapai dalam tugas klasifikasi sentimen bervariasi, dengan beberapa studi menunjukkan kinerja yang tinggi, seperti 90.3% oleh Ihsan Puntadewa [17] dan bahkan 91.2% (LSTM) serta 91.4% (CNN) dalam penelitian Davina Febryanthi Kurniawan terhadap sentimen pemerintahan Jokowi [18]. Namun, meskipun terdapat kemajuan, beberapa celah penelitian utama dapat diidentifikasi.

Pertama, banyak penelitian terdahulu yang berfokus pada prediksi harga saham atau korelasi sentimen dengan harga, namun tidak secara spesifik menganalisis dinamika sentimen publik terhadap indeks pasar secara keseluruhan seperti IHSG, terutama dalam konteks periode anomali atau krisis pasar. Beberapa studi yang melibatkan IHSG cenderung menggunakan analisis regresi tanpa memanfaatkan data media sosial langsung atau metode *deep learning* [20, 21]. Kedua, sumber data yang digunakan bervariasi; meskipun Twitter populer, beberapa penelitian menggunakan sentimen dari berita [24] atau platform lain seperti YouTube [25], yang mungkin tidak mencerminkan opini publik secara *real-time* dan spontan seperti Twitter. Ketiga, meskipun *deep learning* semakin banyak diadopsi, belum banyak penelitian yang secara langsung membandingkan kinerja model yang dilatih dari awal (*from scratch*) pada data bervolume besar dengan model *pre-trained* yang di-*fine-tune* pada data berlabel manual yang lebih kecil, khususnya dalam domain sentimen pasar modal Indonesia. Penelitian Davina Febryanthi Kurniawan [18] telah menunjukkan kinerja yang baik untuk sentimen pemerintahan Jokowi menggunakan LSTM dan CNN, namun fokusnya berbeda dan tidak secara langsung membahas sentimen terhadap IHSG selama periode krisis. Oleh karena itu, penelitian ini bertujuan untuk mengisi celah tersebut dengan menganalisis sentimen publik terhadap anjloknya IHSG pada periode Maret-April 2025 secara spesifik, menggunakan dan membandingkan arsitektur Bi-LSTM yang dilatih dari awal pada data *pseudo-labeled* bervolume besar, serta arsitektur Transformer (IndoBERT) yang di-*fine-tune* pada data berlabel manual.

2.2 Analisis Sentimen

Analisis sentimen, yang juga dikenal sebagai *opinion mining*, adalah cabang dari *Natural Language Processing* (NLP) yang berfokus pada identifikasi, ekstraksi, kuantifikasi, dan analisis afek, sentimen, serta subjektivitas yang diekspresikan dalam data teks [27]. Tujuan utamanya adalah untuk secara otomatis menentukan polaritas emosional yang terkandung dalam suatu teks apakah positif, negatif, atau netral—terhadap suatu entitas, peristiwa, atau topik tertentu [28]. Dalam era digital di mana volume data tekstual dari media sosial, ulasan produk, dan berita daring meledak, analisis sentimen menjadi alat yang sangat krusial bagi organisasi untuk memahami persepsi publik dan membuat keputusan berbasis data [29].

Secara historis, pendekatan analisis sentimen berevolusi dari metode berbasis aturan dan leksikon (kamus kata) menjadi pendekatan berbasis *machine learning* dan kini didominasi oleh *deep learning*. Pendekatan awal mengandalkan kamus kata-kata yang telah diberi skor polaritas untuk mengakumulasi skor sentimen sebuah kalimat [30]. Meskipun sederhana, metode ini seringkali kesulitan menangani konteks, ironi, dan sarkasme. Oleh karena itu, penelitian modern lebih banyak berfokus pada model *deep learning* yang dapat mempelajari representasi fitur dan konteks secara otomatis dari data, yang terbukti memberikan kinerja yang jauh lebih unggul [14].

Penerapan analisis sentimen di bidang keuangan telah menjadi area riset yang sangat aktif. Teori Keuangan Perilaku (*Behavioral Finance*) menyatakan bahwa keputusan investor tidak selalu rasional dan seringkali dipengaruhi oleh bias kognitif dan emosi kolektif, atau yang dikenal sebagai sentimen investor [30]. Sentimen yang diekspresikan di platform seperti Twitter dapat dianggap sebagai proksi dari sentimen investor ritel secara *real-time*. Berbagai studi terkini telah mengonfirmasi bahwa metrik sentimen yang diekstraksi dari media sosial memiliki kekuatan prediktif terhadap volatilitas dan arah pergerakan pasar saham, termasuk di Indonesia [13, 14].

2.3 *Natural Language Processing* (NLP)

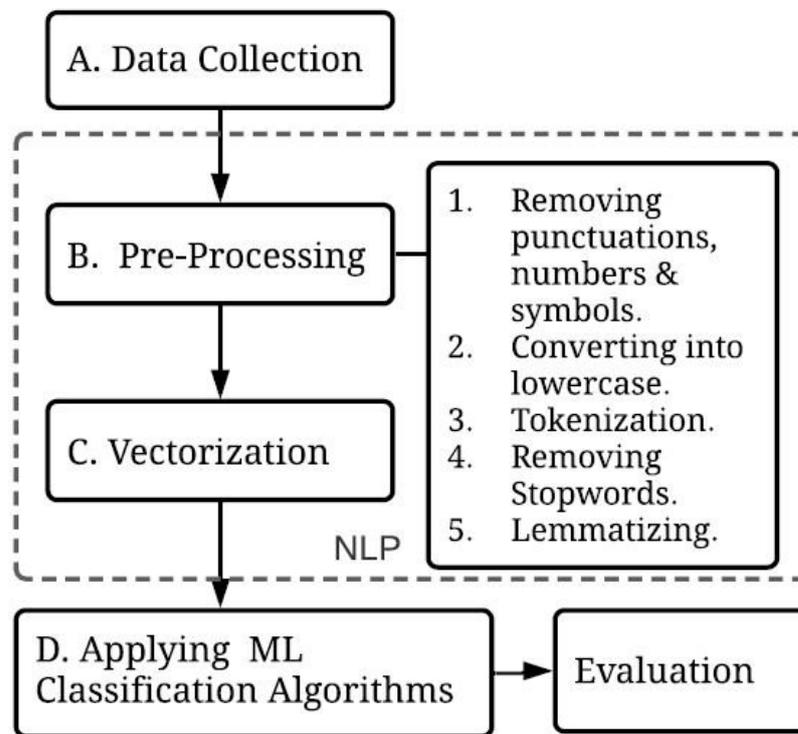
Natural Language Processing Pemrosesan Bahasa Alami (PBA) atau *Natural Language Processing* (NLP) merupakan bidang interdisipliner yang berada pada irisan antara ilmu komputer, kecerdasan buatan, dan linguistik, yang bertujuan untuk mengembangkan sistem yang mampu memproses, memahami, menganalisis,

serta menghasilkan bahasa manusia secara otomatis [31, 32]. Tujuan utamanya adalah untuk menjembatani kesenjangan komunikasi antara manusia dan mesin, mengubah data teks dan ucapan yang tidak terstruktur menjadi format terstruktur yang dapat diinterpretasikan oleh algoritma komputer [33].

Evolusi NLP telah melalui beberapa fase paradigma. Pendekatan awal sangat bergantung pada aturan linguistik yang dibuat manual (*rule-based systems*), yang terbukti rapuh dan sulit untuk diskalakan. Era berikutnya didominasi oleh pendekatan statistik pada tahun 1990-an hingga 2010-an, yang menggunakan model probabilitas seperti n-gram dan metode klasik seperti *Bag-of-Words* (BoW) dan TF-IDF [34]. Metode ini, meskipun lebih fleksibel, memiliki keterbatasan signifikan dalam menangkap makna semantik dan konteks kalimat. Revolusi sesungguhnya terjadi dengan munculnya representasi kata berbasis jaringan saraf, seperti Word2Vec, yang memungkinkan kata-kata direpresentasikan sebagai vektor padat (*embeddings*) dan mampu menangkap hubungan semantik antar kata [31].

Era modern NLP, terutama sejak tahun 2018, didefinisikan oleh dominasi arsitektur *deep learning* berbasis Transformer, seperti BERT dan GPT [32]. Model-model ini, yang telah melalui pra-pelatihan pada miliaran kalimat, mampu menciptakan representasi kontekstual yang sangat kaya untuk setiap kata dalam sebuah kalimat. Berbeda dengan pendekatan sebelumnya yang memperlakukan kata sebagai entitas independen, model Transformer dapat memahami bahwa makna sebuah kata berubah tergantung pada kata-kata di sekitarnya. Kemajuan ini telah menghasilkan peningkatan kinerja yang drastis di hampir semua tugas NLP, mulai dari terjemahan mesin, rangkuman teks, hingga analisis sentimen yang menjadi fokus penelitian ini [31].

Secara praktis, sebuah proyek NLP untuk tugas klasifikasi teks seperti analisis sentimen umumnya mengikuti alur kerja yang sistematis seperti yang digambarkan pada Gambar 2.1. Alur ini dimulai dengan pengumpulan data, diikuti oleh fase preprocessing, representasi teks (*vectorization*), pemodelan, dan terakhir evaluasi [33].



Gambar 2.1. Ilustrasi Alur Kerja Umum pada Proyek NLP [1]

2.3.1 *Word Cloud*

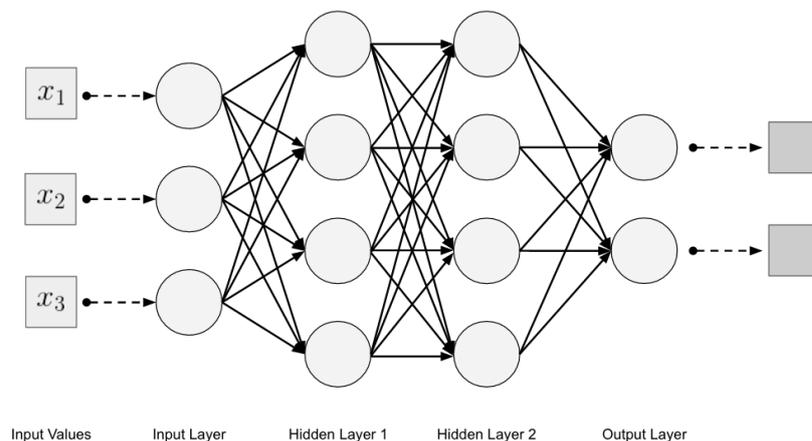
Visualisasi data memegang peranan penting dalam analisis data, terutama pada tahap eksplorasi untuk mendapatkan pemahaman intuitif dari data mentah. Salah satu teknik visualisasi yang populer untuk data tekstual adalah *Word Cloud* atau awan kata. *Word Cloud* adalah representasi visual dari frekuensi kata dalam sebuah korpus teks, di mana kata-kata yang lebih sering muncul akan ditampilkan dengan ukuran font yang lebih besar [35]. Teknik ini sangat efektif untuk mengidentifikasi secara cepat tema atau kata kunci utama yang dominan dalam sekumpulan besar dokumen, sehingga membantu peneliti dalam membentuk hipotesis awal sebelum melakukan analisis kuantitatif yang lebih mendalam.

2.4 *Deep Learning* untuk NLP

Deep Learning merupakan cabang dari *machine learning* yang mengandalkan arsitektur *Artificial Neural Networks* (ANN) berlapis-lapis,

yang dikenal sebagai arsitektur dalam (*deep architecture*), untuk mempelajari representasi data secara hierarkis[36]. Keunggulan utamanya adalah kemampuannya mempelajari representasi fitur secara hierarkis dan otomatis langsung dari data mentah, yang membuatnya sangat unggul untuk data tidak terstruktur seperti teks [37].

Setiap neuron dalam jaringan saraf menerima sekumpulan input, menghitung penjumlahan berbobot (*weighted sum*), dan menerapkannya pada sebuah fungsi aktivasi non-linier untuk menghasilkan output, seperti dijelaskan pada Persamaan 2.1. Dengan menumpuk banyak neuron ini ke dalam beberapa lapisan, terbentuklah sebuah jaringan yang "dalam" seperti diilustrasikan pada Gambar 2.2.



Gambar 2.2. Ilustrasi Arsitektur Deep Neural Network (sumber: [2])

$$y = f \left(\sum_{i=1}^n (w_i \cdot x_i) + b \right) \quad (2.1)$$

di mana:

1. y : Output dari neuron.
2. f : Fungsi aktivasi non-linier (contoh: ReLU, Sigmoid).
3. w_i : Bobot (*weight*) dari input ke- i .
4. x_i : Nilai dari input ke- i .
5. b : Bias, sebuah nilai konstan untuk menyesuaikan output.

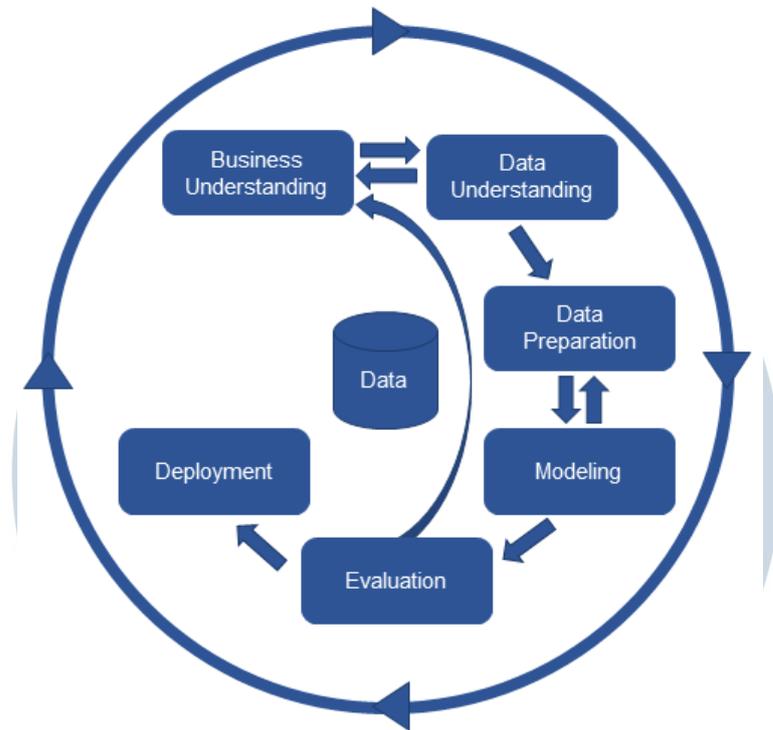
Paradigma terkini dalam NLP didominasi oleh arsitektur Transformer, yang sepenuhnya mengandalkan mekanisme *self-attention* untuk menimbang hubungan antara semua kata dalam teks secara simultan. Hal ini melahirkan model-model canggih seperti BERT dan IndoBERT yang menjadi *state-of-the-art* dalam pemahaman konteks dua arah [38]. Penelitian ini secara langsung akan membandingkan efektivitas dari arsitektur sekuensial (LSTM) dengan arsitektur Transformer.

2.5 Metodologi CRISP-DM

Cross-Industry Standard Process for Data Mining (CRISP-DM) adalah sebuah kerangka kerja dan metodologi standar terbuka yang paling banyak digunakan di dunia untuk memandu proyek-proyek *data science* dan *data mining* [39]. Dikembangkan pada akhir tahun 1990-an oleh sebuah konsorsium industri, CRISP-DM dirancang untuk menjadi independen dari jenis industri maupun teknologi yang digunakan. Tujuannya adalah untuk menyediakan peta jalan yang terstruktur dan sistematis, sehingga proses analisis data menjadi lebih terkelola, efisien, dan dapat direplikasi [40].

Meskipun telah berusia lebih dari dua dekade, relevansi CRISP-DM tetap bertahan kuat di era *data science* modern. Sifatnya yang fleksibel dan siklus hidupnya yang non-linear (iteratif) membuatnya sangat cocok untuk diaplikasikan pada berbagai macam masalah, termasuk proyek-proyek yang melibatkan analisis data tidak terstruktur seperti teks dari media sosial dan pemodelan menggunakan *deep learning* [41]. Kerangka kerja ini membantu peneliti untuk tidak melewatkan langkah-langkah krusial, mulai dari pemahaman masalah hingga evaluasi hasil akhir, sehingga meningkatkan peluang keberhasilan proyek secara keseluruhan [39].

Pendekatan CRISP-DM mencakup enam tahapan inti yang saling berkaitan satu sama lain, sebagaimana digambarkan pada Gambar 2.3. Arah panah dalam diagram menggambarkan bahwa proses ini bersifat iteratif dan berulang, memungkinkan peneliti untuk kembali ke tahapan sebelumnya guna melakukan penyesuaian berdasarkan temuan yang diperoleh di tahap akhir.



Gambar 2.3. Siklus Metodologi CRISP-DM [3]

Berikut adalah penjelasan teoretis untuk setiap fase yang relevan dalam penelitian ini:

2.5.1 *Business Understanding* (Pemahaman Masalah)

Fase pertama dan paling krusial dalam CRISP-DM adalah memahami tujuan proyek dari perspektif penelitian atau bisnis. Tahap ini melibatkan penerjemahan masalah umum menjadi pertanyaan analisis data yang spesifik dan terukur, serta mendefinisikan kriteria keberhasilan proyek [40]. Tanpa pemahaman yang jelas pada fase ini, sebuah proyek berisiko kehilangan arah dan menghasilkan output yang tidak relevan. Dalam konteks skripsi, fase ini diwujudkan dalam perumusan Latar Belakang dan Rumusan Masalah di Bab 1.

2.5.2 *Data Understanding* (Pemahaman Data)

Setelah tujuan proyek ditetapkan, fase ini berfokus pada pengumpulan data awal dan melakukan eksplorasi untuk berkenalan dengan data. Kegiatan utamanya meliputi pengumpulan data, deskripsi data (misalnya, jumlah data, atribut yang ada), eksplorasi data untuk menemukan pola awal, dan verifikasi kualitas data

untuk mengidentifikasi adanya data yang hilang atau tidak konsisten [39]. Fase ini penting untuk membentuk hipotesis awal dan merencanakan tahap persiapan data selanjutnya.

2.5.3 Data Preparation (Persiapan Data)

Fase ini seringkali merupakan fase yang paling intensif, mencakup semua aktivitas untuk membangun dataset final yang akan digunakan untuk pemodelan dari data mentah awal. Tugas-tugas utamanya meliputi pemilihan data (memilih baris dan kolom yang relevan), pembersihan data (menangani *noise* dan data yang hilang), konstruksi fitur baru, integrasi data dari berbagai sumber, dan pemformatan ulang data agar sesuai dengan kebutuhan *tools* pemodelan[41]. Dalam penelitian ini, fase ini mencakup semua tahapan *preprocessing*, pelabelan, ekstraksi fitur, dan pembagian data yang akan dijelaskan secara rinci di Bab 3.

2.5.4 Modeling (Pemodelan)

Pada tahap ini, berbagai metode pemodelan dievaluasi dan diterapkan, disertai dengan penyesuaian parameter untuk memperoleh performa yang optimal. Pemilihan metode yang sesuai sangat dipengaruhi oleh tujuan analisis serta karakteristik data yang digunakan[40]. Tahapan ini umumnya bersifat iteratif, di mana analis melakukan eksplorasi terhadap beberapa pendekatan model dan konfigurasi untuk menentukan alternatif terbaik. Dalam konteks penelitian ini, proses pemodelan melibatkan pengembangan model Bi-LSTM serta penerapan proses fine-tuning terhadap IndoBERT.

2.5.5 Evaluation (Evaluasi)

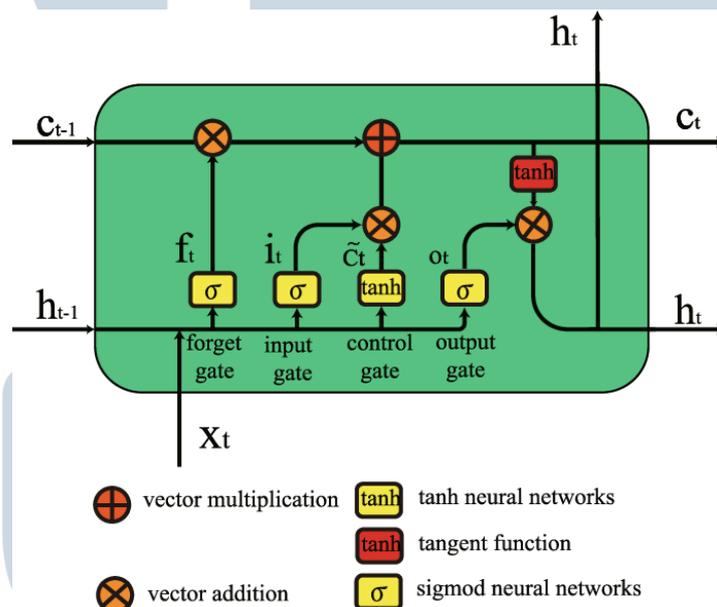
Setelah proses pembangunan model selesai, tahap selanjutnya adalah melakukan evaluasi secara komprehensif guna menilai kualitas dan kesesuaian model terhadap tujuan yang telah ditetapkan pada fase *Business Understanding* ([39]). Evaluasi ini mencakup pengujian performa model menggunakan metrik kuantitatif seperti akurasi dan F1-score terhadap data uji yang belum pernah digunakan sebelumnya. Temuan dari proses evaluasi ini menjadi dasar untuk menentukan apakah model dapat langsung diimplementasikan atau perlu dilakukan revisi dengan kembali ke tahapan sebelumnya.

2.6 Metode Pemodelan

Tahap pemodelan dalam penelitian ini melibatkan implementasi dan perbandingan dua arsitektur *deep learning* yang berbeda secara fundamental. Bagian ini menguraikan landasan teoretis dari setiap model yang digunakan.

2.6.1 Bidirectional Long Short-Term Memory (Bi-LSTM)

Pada ranah Pemrosesan Bahasa Alami (NLP), arsitektur *Long Short-Term Memory* (LSTM) menawarkan pendekatan yang efektif dalam menangani data berurutan, dengan mengatasi kendala *vanishing gradient* yang sering terjadi pada RNN konvensional melalui penggunaan mekanisme gerbang yang kompleks, sebagaimana diperlihatkan pada Gambar 2.4 [36].



Gambar 2.4. Diagram Arsitektur Sel LSTM [4]

Persamaan matematis untuk setiap gerbang pada sel LSTM adalah sebagai berikut [42]:

1. **Forget Gate** (f_t): Menentukan informasi apa dari status sel sebelumnya (C_{t-1}) yang akan dibuang.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.2)$$

2. **Input Gate** (i_t): Menentukan informasi baru apa yang akan disimpan ke dalam status sel.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.3)$$

3. **Output Gate** (o_t): Menentukan apa yang akan menjadi output (h_t) berdasarkan status sel saat ini (C_t).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.4)$$

di mana:

1. f_t, i_t, o_t : Vektor hasil dari forget, input, dan output gate.
2. σ : Fungsi aktivasi Sigmoid, yang menghasilkan nilai antara 0 dan 1.
3. W_f, W_i, W_o : Matriks bobot (*weight matrices*) untuk setiap gerbang.
4. h_{t-1} : Output (*hidden state*) dari sel LSTM pada langkah waktu sebelumnya.
5. x_t : Input pada langkah waktu saat ini.
6. b_f, b_i, b_o : Vektor bias untuk setiap gerbang.

Meskipun demikian, versi standar dari LSTM hanya memproses urutan data dalam satu arah, yaitu ke depan. Untuk memperluas pemahaman terhadap konteks sekuens secara menyeluruh, penelitian ini mengimplementasikan varian *Bidirectional LSTM* (Bi-LSTM). Model Bi-LSTM menggabungkan dua lapisan LSTM yang berjalan secara paralel: satu lapisan mengolah urutan dari awal ke akhir (*forward layer*), sedangkan lapisan lainnya memproses dari akhir ke awal (*backward layer*).

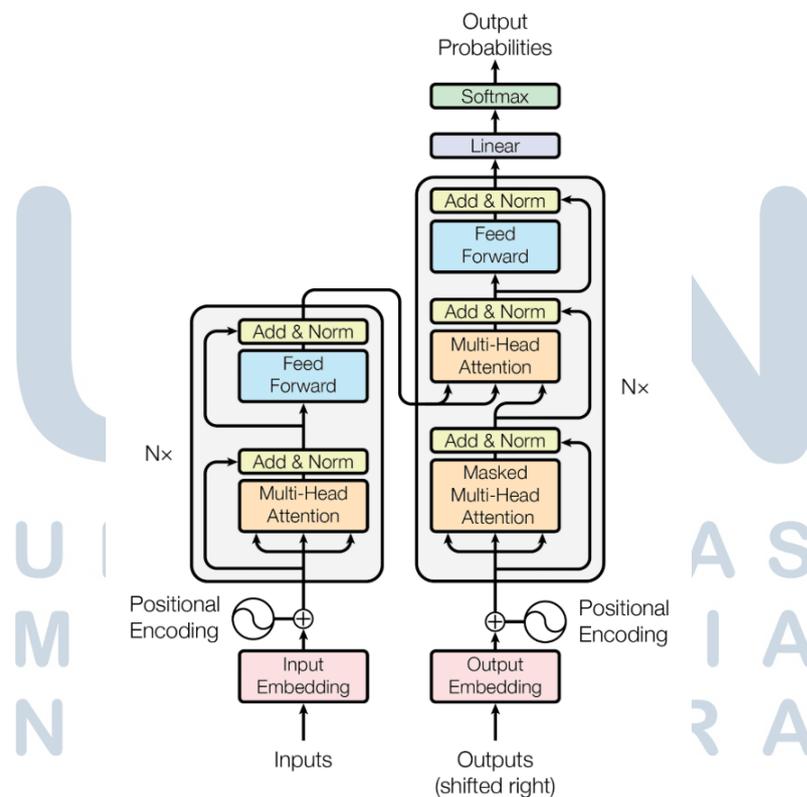
Pada setiap langkah waktu (*time step*) t , output dari kedua lapisan tersebut (\vec{h}_t dari forward layer dan \overleftarrow{h}_t dari backward layer) digabungkan (biasanya dengan konkatenasi) untuk membentuk representasi akhir. Persamaan untuk output Bi-LSTM dapat dirumuskan sebagai:

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (2.5)$$

Pendekatan dua arah ini memungkinkan model untuk memiliki akses ke informasi kontekstual dari masa lalu (kata-kata sebelumnya) dan masa depan (kata-kata sesudahnya) saat membuat prediksi untuk kata saat ini. Kemampuan ini terbukti secara konsisten memberikan peningkatan kinerja dibandingkan LSTM satu arah pada berbagai tugas NLP, termasuk analisis sentimen [43].

2.6.2 IndoBERT (Arsitektur Berbasis Transformer)

Berbeda dengan LSTM yang memproses teks secara sekuensial, IndoBERT dibangun di atas arsitektur **Transformer** yang sepenuhnya mengandalkan mekanisme **self-attention**. Mekanisme ini memungkinkan model untuk menimbang pentingnya setiap kata dalam sebuah kalimat saat merepresentasikan satu kata tertentu, dengan melihat semua kata lain dalam kalimat yang sama secara bersamaan. Arsitektur lengkap dari Transformer, yang terdiri dari blok *Encoder* dan *Decoder*, diilustrasikan pada Gambar 2.5. Model IndoBERT secara spesifik menggunakan tumpukan blok *Encoder* (sisi kiri) untuk pemrosesan teks.



Gambar 2.5. Diagram Arsitektur Lengkap Model Transformer [5]

Inti dari arsitektur ini adalah komponen *Multi-Head Attention*. Secara konseptual, untuk setiap kata, model menciptakan tiga vektor: *Query* (Q), *Key* (K), dan *Value* (V). Vektor *Query* merepresentasikan kata saat ini, sementara vektor *Key* dan *Value* merepresentasikan semua kata lain (termasuk kata itu sendiri) dalam kalimat. Skor atensi dihitung dengan mengukur kemiripan antara *Query* dari satu kata dengan *Key* dari setiap kata lainnya. Skor ini kemudian digunakan untuk membobot vektor *Value*, yang pada akhirnya menghasilkan representasi kontekstual untuk kata tersebut. Proses ini dirumuskan secara matematis pada Persamaan 2.6.

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.6)$$

di mana:

1. Q : Matriks *Query*, merepresentasikan kata yang sedang diproses.
2. K : Matriks *Key*, merepresentasikan semua kata dalam kalimat untuk dibandingkan dengan *Query*.
3. V : Matriks *Value*, merepresentasikan semua kata dalam kalimat untuk kemudian dibobot berdasarkan skor atensi.
4. K^T : Transpos dari matriks *Key*.
5. d_k : Dimensi dari vektor *Key*, digunakan sebagai faktor penskalaan untuk menjaga stabilitas gradien selama pelatihan [5].

Kekuatan utama model seperti IndoBERT terletak pada strategi pra-pelatihan (*pre-training*) dan *fine-tuning*. Model ini pertama-tama dilatih pada korpus data Bahasa Indonesia yang sangat masif (tahap *pre-training*) untuk mempelajari pola bahasa, tata bahasa, dan pengetahuan umum. Proses pra-pelatihan BERT (dan turunannya seperti IndoBERT) memiliki dua tujuan utama [44]:

1. ***Masked Language Model* (MLM)**: Secara acak, beberapa kata dalam sebuah kalimat disembunyikan (*di-mask*), dan model ditugaskan untuk menebak kata yang disembunyikan tersebut berdasarkan konteks kata-kata di sekitarnya (baik dari kiri maupun kanan). Ini memaksa model untuk mempelajari representasi kontekstual dua arah (*bidirectional*) yang mendalam.

2. **Next Sentence Prediction (NSP):** Model diberi dua kalimat (A dan B) dan harus menentukan apakah kalimat B adalah kalimat yang logis mengikuti kalimat A. Tugas ini membantu model memahami hubungan antar kalimat.

Setelah melalui pra-pelatihan yang intensif, model yang sudah "pintar" ini kemudian dapat diadaptasi untuk tugas spesifik (seperti analisis sentimen IHSG) melalui proses *fine-tuning* pada dataset yang lebih kecil dan berlabel [16]. Pendekatan *transfer learning* ini telah menjadi standar emas dalam NLP modern karena terbukti sangat efisien dan mampu mencapai kinerja canggih bahkan dengan data latih yang relatif terbatas [44].

2.7 Metode Evaluasi

Untuk mengukur dan membandingkan kinerja model klasifikasi sentimen secara kuantitatif, digunakan matriks evaluasi standar yang berasal dari *Confusion Matrix*. *Confusion Matrix* adalah sebuah tabel yang memvisualisasikan performa sebuah algoritma klasifikasi dengan membandingkan kelas aktual dengan kelas yang diprediksi oleh model. Gambar 2.6 mengilustrasikan konsep dasar dari *confusion matrix* untuk kasus klasifikasi biner.

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Gambar 2.6. Ilustrasi Konsep Dasar Confusion Matrix (2x2) [6]

Dari nilai-nilai dalam *Confusion Matrix*—yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) dapat dihitung beberapa metrik evaluasi utama sebagai berikut:

1. **Akurasi (Accuracy):** Metrik ini menghitung persentase prediksi yang tepat terhadap seluruh data yang diuji. Meskipun memberikan gambaran umum

mengenai performa model, penggunaan metrik ini pada data yang tidak seimbang (*imbalanced dataset*) dapat menghasilkan interpretasi yang keliru.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.7)$$

2. **Presisi (*Precision*):** Menunjukkan proporsi prediksi yang benar pada kategori positif dibandingkan dengan seluruh prediksi yang diklasifikasikan sebagai positif. Metrik ini merepresentasikan sejauh mana model mampu menghasilkan prediksi positif yang akurat untuk suatu kelas tertentu.

$$Precision = \frac{TP}{TP + FP} \quad (2.8)$$

3. **Recall (*Sensitivity*):** Mengukur seberapa besar proporsi prediksi positif yang benar di antara seluruh data yang benar-benar termasuk dalam kelas positif. Metrik ini merefleksikan kemampuan model dalam mendeteksi semua instance relevan dari suatu kelas tertentu.

$$Recall = \frac{TP}{TP + FN} \quad (2.9)$$

4. **F1-Score:** Merupakan rata-rata harmonik antara nilai Presisi dan Recall. Metrik ini sangat bermanfaat dalam kondisi ketidakseimbangan kelas, karena mampu menggabungkan kedua metrik tersebut secara seimbang untuk memberikan penilaian kinerja model yang lebih menyeluruh.

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.10)$$

Dalam penelitian ini, metrik-metrik tersebut akan dihitung untuk setiap model guna mendapatkan gambaran kinerja yang komprehensif.