BAB II
KAJIAN LITERATUR

2.1 Kecerdasan Buatan (Artificial Intelligence/AI)

Kecerdasan buatan (*Artificial Intelligence/AI*) adalah cabang ilmu komputer yang berfokus pada pengembangan sistem yang mampu meniru kemampuan kognitif manusia, seperti penalaran, pemahaman bahasa, pengambilan keputusan, dan pembelajaran dari pengalaman [21]. Secara umum, AI dirancang untuk memungkinkan mesin melakukan tugas-tugas yang secara tradisional memerlukan kecerdasan manusia [22]. Salah satu kemampuan utama AI adalah otomatisasi pengolahan data dan informasi dalam skala besar. Sistem berbasis AI dapat menganalisis, menginterpretasikan, dan mengambil kesimpulan dari data yang tidak terstruktur, termasuk teks, gambar, dan suara, dengan kecepatan dan konsistensi yang jauh melampaui kemampuan manusia. Kemampuan ini menjadikan AI sangat relevan dalam berbagai bidang, mulai dari kesehatan, keuangan, hingga analisis teks di media sosial [20, 21, 22].

Dalam perkembangannya, AI tidak bekerja sebagai satu metode tunggal, melainkan mencakup berbagai pendekatan dan teknik. Salah satu pendekatan yang paling dominan dan banyak digunakan saat ini adalah *Machine Learning*, yang memungkinkan sistem untuk belajar secara otomatis dari data tanpa harus diprogram secara eksplisit untuk setiap tugas [23, 24].

2.1.1 Machine Learning

Machine Learning (ML) merupakan bagian dari kecerdasan buatan yang memungkinkan sistem komputer untuk belajar dan meningkatkan kinerjanya secara otomatis melalui pengalaman, tanpa diprogram secara eksplisit [27]. Alih-alih mengandalkan aturan yang ditulis secara manual, model *Machine Learning* dilatih menggunakan data historis untuk mengenali pola dan membuat prediksi terhadap data baru [28]. Salah satu paradigma utama dalam *Machine Learning* adalah *supervised learning*, yaitu pendekatan di mana model dilatih menggunakan data berlabel, artinya setiap data masukan telah disertai dengan keluaran yang diharapkan. Model kemudian belajar memetakan hubungan antara masukan dan keluaran tersebut, sehingga mampu melakukan prediksi pada data yang belum pernah dilihat sebelumnya [26, 27]. *Supervised learning* sangat efektif untuk tugas klasifikasi, yaitu pengelompokan data ke dalam kategori-kategori tertentu [31].

Dalam konteks penelitian ini, analisis sentimen diperlakukan sebagai tugas klasifikasi teks, di mana model dilatih untuk mengklasifikasikan teks ulasan ke dalam kategori sentimen, seperti positif, negatif, atau netral [32]. Dua algoritma yang digunakan dalam penelitian ini, yaitu SVM dan IndoBERT , keduanya beroperasi dalam kerangka *supervised learning* ini. Untuk membangun model yang lebih kompleks dan mampu memahami representasi bahasa secara mendalam, *Machine Learning* terus berkembang ke arah pendekatan yang lebih canggih, yang dikenal sebagai *Deep Learning* [33].

2.1.2 Deep Learning

Deep Learning adalah subbidang dari *Machine Learning* yang menggunakan arsitektur jaringan saraf tiruan (*neural network*) berlapis-lapis untuk mempelajari representasi data secara hierarkis [31, 32]. Setiap lapisan jaringan secara otomatis mengekstraksi fitur dari data masukan, mulai dari fitur sederhana pada lapisan awal hingga fitur yang semakin abstrak dan kompleks pada lapisan yang lebih dalam, tanpa memerlukan rekayasa fitur secara manual [36]. Kemampuan *Deep Learning* dalam merepresentasikan fitur secara otomatis menjadikannya sangat unggul dalam memproses data tidak terstruktur, khususnya teks. Pada tugas pemrosesan bahasa alami (*Natural Language Processing*/NLP), *Deep Learning* mampu menangkap konteks, ambiguitas, dan hubungan semantik dalam teks yang kompleks dengan jauh lebih baik dibandingkan metode konvensional [34, 35].

Perkembangan *Deep Learning* melahirkan *Transformer*, yang menjadi fondasi dari model-model bahasa modern. Arsitektur ini memperkenalkan mekanisme *Self-Attention* yang memungkinkan model untuk mempertimbangkan seluruh konteks kalimat secara bersamaan [39]. BERT (*Bidirectional Encoder Representations from Transformers*) merupakan salah satu model yang dibangun di atas *Transformer* ini, dan menjadi dasar dari IndoBERT yang digunakan dalam penelitian ini [40].

2.1.3 Natural Language Processing (NLP)

Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan yang mempelajari bagaimana komputer dapat memahami, mengolah, serta menghasilkan bahasa manusia secara otomatis. Bidang ini menggabungkan konsep dari linguistik, ilmu komputer, dan pembelajaran mesin untuk memungkinkan sistem komputer menafsirkan teks atau percakapan

manusia dalam bentuk yang dapat diproses secara komputasional [41]. Tujuan utama NLP adalah memungkinkan mesin untuk memahami makna dari bahasa manusia sehingga dapat digunakan dalam berbagai aplikasi seperti sistem percakapan, penerjemahan bahasa, pencarian informasi, serta analisis sentiment [24]. Dalam konteks perkembangan kecerdasan buatan saat ini, NLP menjadi teknologi utama yang memungkinkan sistem seperti ChatGPT berinteraksi dengan pengguna menggunakan bahasa alami [42].

Dalam penelitian berbasis teks, NLP banyak digunakan untuk menganalisis opini dan informasi yang disampaikan pengguna melalui media sosial [41]. Data teks pada media sosial umumnya bersifat tidak terstruktur, mengandung variasi bahasa informal, singkatan, serta penggunaan kata yang kontekstual, sehingga memerlukan pendekatan pemrosesan bahasa yang mampu memahami makna teks secara lebih baik. Salah satu penerapan NLP yang banyak digunakan adalah analisis sentimen, yaitu proses mengidentifikasi dan mengklasifikasikan opini pengguna ke dalam kategori sentimen tertentu berdasarkan teks yang ditulis [43].

2.2 Analisis Sentimen

Analisis sentimen merupakan bidang riset yang mengkategorikan opini dan emosi dari teks menjadi polaritas positif, negatif, atau netral, yang telah banyak diterapkan dalam berbagai sektor seperti keuangan, politik, dan studi tentang opini public [44]. kemampuan tersebut membuktikan bahwa analisis sentimen memiliki peran krusial dalam mendukung pengambilan keputusan, mulai dari keputusan bisnis hingga kebijakan politik [45]. Proses ini melibatkan pengelompokan teks untuk menentukan label sentimen yang dikemukakan, baik itu positif, negatif, atau netral, terhadap suatu topik tertentu [46]. Pendekatan ini juga terbagi menjadi berbagai metode komputasi yang memanfaatkan teknik *Natural Language Processing* untuk secara otomatis mengekstraksi dan memahami informasi sentimen tersirat dalam data teks tidak terstruktur [47].

Dalam penerapannya, terdapat dua skema klasifikasi sentimen yang umum digunakan, yaitu klasifikasi tiga label dan dua label. Klasifikasi tiga label (positif, negatif, dan netral) bertujuan untuk memetakan opini publik secara menyeluruh, di mana label positif merepresentasikan apresiasi atau dukungan, label negatif menunjukkan kritik atau kekhawatiran, dan label netral menampung informasi objektif atau pertanyaan tanpa tendensi emosional [48]. Di sisi lain, klasifikasi dua label (positif dan negatif) sering digunakan untuk

mempertajam analisis pada polaritas opini yang kontras dengan mengeliminasi data netral, sehingga mempermudah evaluasi performa model dalam membedakan sentimen ekstrem secara tegas [49]. Pada penelitian ini, penentuan label tersebut dilakukan melalui pendekatan semi-otomatis menggunakan model IndoBERT, yang kemudian diikuti dengan tahap verifikasi dan koreksi manual oleh peneliti guna menjamin validitas serta akurasi *ground truth* yang dihasilkan.

Pada tahap awal perkembangannya, analisis sentimen banyak dilakukan menggunakan pendekatan berbasis leksikon atau kata-kata yang telah ditentukan sebelumnya, pendekatan ini tidak menggunakan algoritma pembelajaran mesin, melainkan memanfaatkan kamus kata yang telah diberi nilai sentimen tertentu [50]. Metode ini mengandalkan daftar kata positif dan negatif yang telah dikurasi untuk menilai polaritas sentimen dalam sebuah teks [51]. terutama ketika menerapkan kamus sentimen untuk memberikan bobot pada kata-kata yang mengandung sentimen negatif atau positif dalam teks [52].

Seiring berkembangnya penelitian di bidang pengolahan bahasa alami, analisis sentimen kemudian banyak dilakukan menggunakan metode pembelajaran mesin (*Machine Learning*) [53]. Pendekatan ini bekerja dengan melatih model klasifikasi menggunakan dataset teks yang telah diberi label sentimen sebelumnya [53]. Data teks biasanya diubah terlebih dahulu menjadi representasi numerik menggunakan teknik representasi fitur seperti *TF-IDF* sebelum diproses oleh algoritma klasifikasi. Beberapa algoritma yang sering digunakan dalam analisis sentimen antara lain Naïve Bayes dan *Support Vector Machine* [54]. Perkembangannya, analisis sentimen juga memanfaatkan pendekatan *Deep Learning* yang mampu memahami hubungan antarkata secara lebih kompleks [55].

2.3 Preprocessing dan Feature Extraction

Data yang diperolah dari media sosial umunya masih berbentuk teks tidak terstruktur yang mengandung berbagai elemen tambahan seperti simbol, singkatan, serta variasi penulisan kata [32]. Oleh karena itu, sebelum dilakukann proses analisis sentimen, diperlukan tahapan untuk pengolahan teks untuk mengubah data mentah menjadi bentuk yang lebih terstruktur sehingga dapat diproses oleh model pembelajaran mesin maupun model yang berbasis *Deep Learning* [56]. Pada penelitian ini, proses *Preprocessing* dilakukan terhadap data cuitan Twitter/X berbahasa Indonesia yang memiliki ciri khas teks pendek, informal, serta sering

mengandung singkatan atau variasi penulisan kata, tahapan *Preprocessing* yang akan digunakan meliputi beberapa proses sebagai berikut:

1. Data *Cleaning*

   Data *Cleaning* merupakan proses pembersihan teks dari berbagai elemen yang tidak di perlukan kontribusinya terhadap analisis sentimen [57]. Pada penelitian ini, proses pembersihan data meliputi penghapusan beberapa elemen berikut:

   a) *Username* atau *Mention* pengguna (contoh *@username*)
   b) *URL* atau *hyperlink*
   c) Hastag (#)
   d) Angka dan emoji serta karakter khusus yang tidak relevan terhadap analisis sentimen.

2. *Case Folding*

   Case Folding merupakan proses pengubahan seluruh huruf dalam teks menjadi huruf kecil [58].

3. *Tokenization*

   Tokenization merupakan proses memecah teks menjadi unit-unit kata yang lebih kecil yang disebut *token*. Proses ini bertujuan untuk mempermudah sistem dalam pengolahan teks secara komputasional [59].

4. Normalisasi Kata

   Teks pada media sosial sering menggunakan kalimat tidak baku, singkatan maupun bahasa gaul yang umumnya digunakan dalam percakapan sehari-hari [60]. Kondisi ini dapat mempengaruhi proses analisis teks model akan menganggap kata tersebut sebagai bentuk kata yang berbeda.

5. *Stop Removal*

   Stopword Removal merupakan proses penghapusan kata-kata umum yang sering muncul dalam kalimat, akan tetapi tidak memberikan informasi yang signifikat terhadap analisis sentimen [61]. Contoh kalimat yang termasuk dalam kategori stopword antara lain “dan”, “di”. Serta “dari”, terutama pada pendekatan berbasis fitur *TF-IDF*.

6. *Stemming*

   Stemming merupakan proses mengubah kata berimbuhan menjadi bentuk kata dasar untuk mengurangi variasi kata yang memiliki makna serupa. Sebagai contoh, kata “mengerjakan”, “dikerjakan”, dan “pengerjaan” akan diubah menjadi kata dasar “kerja”.

Dalam penelitian teks berbahasa Indonesia, proses *Stemming* umumnya dilakukan menggunakan algoritma *Stemming* bahasa Indonesia seperti Sastrawi [62].

7. *TF-IDF*

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan teknik ekstraksi fitur yang berfungsi memetakan data teks ke dalam representasi numerik agar dapat diproses oleh algoritma mesin pembelajar. Teknik ini menentukan bobot atau nilai kepentingan suatu kata berdasarkan kemunculannya di sebuah teks relatif terhadap seluruh dataset dalam korpus [63]. Metode *TF-IDF* menggunakan dua indikator utama dalam perhitungannya, yaitu *Term Frequency (TF)* dan *Inverse Document Frequency* (IDF) [56]. *Term Frequency (TF)* mengukur frekuensi kemunculan sebuah kata atau term dalam satu dokumen tertentu. Karena panjang dokumen dalam sebuah korpus dapat bervariasi, sebuah term memiliki probabilitas kemunculan yang lebih tinggi pada dokumen yang lebih panjang dibandingkan dokumen yang lebih pendek. Sementara itu, Inverse Document Frequency (IDF) berfungsi memberikan bobot lebih tinggi pada kata-kata yang jarang muncul di seluruh dokumen agar kata-kata tersebut memiliki nilai kepentingan yang lebih menonjol dibandingkan kata umum [64].

Secara matematis, perhitungan nilai IDF dilakukan menggunakan rumus sebagai berikut:

$$Idf_t = \log\left(\frac{N}{df_t}\right) \quad \text{..... (1)}$$

Dimana:

Idf_t = merupakan nilai *Inverse Document Frequency* untuk suatu term.

N = menyatakan jumlah total dokumen dalam korpus.

df_t = menunjukkan jumlah dokumen yang mengandung term tersebut

Setelah nilai TF dan IDF didapatkan, nilai bobot akhir untuk setiap kata dihitung melalui perkalian antara kedua indikator tersebut:

$$w_{t,d} = tf_{t,d} \times idf_t \quad \text{..... (2)}$$

Dimana:

$w_{t,d}$ = Nilai bobot akhir kata t pada dokumen d.

$tf_{t,d}$ = Frekuensi kemunculan kata pada t dokumen d

idf_t = Nilai *Inverse Document Frequency* untuk kata t

Dalam implementasi praktis menggunakan Library Scikit-Learn melalui fungsi *TF-IDF Transformer*, skema pembobotan sering kali menggunakan metode smooth IDF dengan penambahan L2 norm untuk menormalisasi panjang vektor. Rumus yang diterapkan dalam skema ini adalah [65]:

$$\text{TF-IDF} = TF(d,t) \times \left[\ln\left(\frac{1+N}{1+df_t}\right) + 1\right] \quad (3)$$

Dimana:

TF-IDF = Nilai bobot Term Frequency–Inverse Document Frequency untuk kata t pada dokumen d

$\text{TF}(d,t)$ = Frekuensi kemunculan t dalam dokumen d

N = Jumlah total seluruh dokumen dalam dataset

df_t = Jumlah dokumen yang mengandung kata t

ln = Logaritma natural (natural logarithm)

2.4 Support Vector Machine (SVM)

*Support Vector Machine (*SVM*)* merupakan algoritma pembelajaran mesin terbimbing (*Supervised learning*) yang bekerja dengan prinsip mencari *hyperplane* atau bidang pemisah optimal untuk membagi data ke dalam kelas-kelas yang berbeda secara akurat [66]. Dalam penelitian ini, algoritma SVM diterapkan untuk mengklasifikasikan sentimen pengguna Twitter/X mengenai pemanfaatan ChatGPT dalam konteks akademik ke dalam tiga kategori, yaitu positif, negatif, dan netral. Tujuan utama dari mekanisme SVM adalah memaksimalkan margin, yaitu jarak terdekat antara bidang pemisah dengan titik data dari masing-masing kelas. Titik-titik data yang berada tepat di garis margin tersebut disebut sebagai *Support Vector* [67]. Fokus pada *Support Vector* memberikan SVM keunggulan dalam menangani data berdimensi tinggi, seperti vektor kata hasil ekstraksi *TF-IDF*, karena keputusan klasifikasi hanya bergantung pada sejumlah kecil sampel data kunci [68].

Secara teknis, proses klasifikasi pada SVM dilakukan menggunakan fungsi keputusan untuk menentukan posisi data terhadap *hyperplane* sebagai berikut:

$$f(x) = \text{sign}(w \cdot x + b) \quad (4)$$

Dimana:

sign = fungsi penentu kelas berdasarkan posisi data terhadap *hyperplane*.

w = Vektor bobot yang menentukan orientasi bidang pemisah.

x = Vektor fitur numerik dari teks (hasil pemrosesan *TF-IDF*).

b = Bias yang menentukan pergeseran bidang dari titik pusat.

Implementasi SVM dalam penelitian ini merepresentasikan pendekatan pembelajaran mesin klasik yang bersifat deterministik. Proses klasifikasi dimulai dengan transformasi data teks yang telah melalui tahap *Preprocessing* dan pelabelan ke dalam bentuk vektor fitur melalui metode pembobotan *TF-IDF*. Model SVM selanjutnya dilatih menggunakan representasi fitur *TF-IDF* untuk mengklasifikasikan sentimen teks ke dalam kategori positif, negatif, dan netral. Penggunaan SVM sangat relevan dalam penelitian ini karena algoritma tersebut memiliki performa yang stabil pada teks pendek dan memiliki kemampuan generalisasi yang baik pada data berdimensi tinggi pada ruang fitur yang luas [69].

2.5 Transformer

Perkembangan model pemrosessan bahasa alami mengalami peningkatan sejak diperkenalkannya *Transformer* oleh *Ashish Vaswani* dengan rekan-rekannya melalui makalah *Attention is All You Need* pada tahun 2017. Arsitektur ini dikembangkan untuk mengatasi keterbatasan model sebelumnya seperti *Recurrent Neural network (RNN) dan Long Short-Term Memory* (*LSTM*) yang memproses teks secara berurutan sehingga kurang efisien dalam menangkap hubungan antara kata yang akan berjauhan dalam suatu kalimat [70]. *Transformer* memungkinkan pemrosesan teks secara bersamaan sehingga lebih efisien serta mampu menangkap hubungan konteks antarkata dengan lebih baik. Arsitektur ini kemudian menjadi dasar bagi berbagai model bahasan modern lainnya, termasuk BERT dan IndoBERT yang akan digunakan dalam penelitian ini [71].

1. *Self-Attention*

   Komponen pertama dari *Transformer* adalah mekanisme *Self-Attention*, yaitu mekanisme yang memungkinkan model memberikan tingkat perhatian yang berbeda terhadap setiap kata dalam suatu kalimat ketika melakukan pemrosesan teks. Mekanisme ini bekerja dengan menghitung hubungan antara kata menggunakan representasi *Query (Q)*, *key (K)*, dan *Value (V)* [72].

2. *Multi-Head Attention*

Selain *Self-Attention*, *Transformer* juga menggunakan mekanisme *Multi-Head Attention* yang memungkinkan model menjalankan beberapa proses *attention* secara bersamaan. Setiap *attention head* punya cara pandang berbeda dalam melihat hubungan antarkata [72]. Pendekatan ini memungkinkan mekanisme atensi untuk berfungsi secara bersamaan, sehingga menghasilkan kinerja jaringan yang lebih baik [73].

3. *Positional encoding*

   Sifat pemrosesan paralel pada *Transformer* meniadakan pemahaman urutan kata secara otomatis. Oleh karena itu, *positional encoding* digunakan sebagai representasi numerik tambahan untuk menandai posisi tiap kata, yang memungkinkan model menginterpretasikan hubungan sintaksis berdasarkan letaknya dalam kalimat [74]. Fitur ini penting karena meskipun *Multi-Head Attention* dapat menangkap dependensi global, informasi posisi relatif dan absolut antarkata tidak dapat diekstrak tanpa encoding posisi [75].

4. *Encoder* pada BERT/IndoBERT

   Transformer mencakup dua komponen utama, *Encoder* untuk mengekstraksi representasi konteks dan *Decoder* untuk pembangkitan teks [71]. Model IndoBERT secara spesifik hanya mengadopsi modul *Encoder*, sebuah konfigurasi yang memungkinkannya memproses informasi secara *Bidirectional* (dua arah). Melalui mekanisme ini, model mampu mengasimilasi makna suatu kata dengan mempertimbangkan seluruh konteks kalimat, baik yang muncul sebelum maupun sesudah kata tersebut, secara simultan [76].

2.5.1 BERT

BERT merupakan model pemrosesan bahasa alami (*Natural Language Processing*) yang memanfaatkan *Transformer* untuk mempelajari representasi teks secara mendalam. Berbeda dengan model pendahulunya yang bersifat searah, BERT dirancang untuk membaca seluruh urutan kata secara bersamaan (*Bidirectional*). Hal ini memungkinkan model untuk memahami konteks sebuah kata berdasarkan informasi dari sisi kiri dan sisi kanan secara simultan dalam seluruh lapisan arsitektur [77].

1. Arsitektur *Encoder Transformer*

   Arsitektur BERT didasarkan pada mekanisme *Encoder* dari *Transformer*. Dalam pengembangannya, BERT melepaskan mekanisme *Decoder* karena fokus utamanya adalah

menghasilkan representasi bahasa yang kaya, bukan menghasilkan teks secara generative [76]. Penggunaan *Encoder* memungkinkan model untuk memperhatikan keterkaitan antarkata dalam satu kalimat melalui mekanisme *Self-Attention*. Secara matematis, inti dari pemrosesan konteks dalam BERT terletak pada fungsi *Scaled Dot-Product Attention* [77].

2. *Framework Pre-Training* dan *Fine-tuning*

   Implementasi BERT terdiri dari dua tahapan utama yang saling berkesinambungan, yaitu tahap *Pre-Training* dan tahap *Fine-tuning* [78].

   a. Tahap *Pre-Training* : Model dilatih menggunakan korpus data teks yang sangat besar tanpa label (*Unlabeled Data*). Pada tahap ini, BERT menjalankan dua tugas utama: *Masked Language Model* (MLM), di mana model memprediksi kata yang disembunyikan dalam kalimat, dan *Next Sentence Prediction* (NSP) untuk memahami hubungan logis antar dua kalimat [79].

   b. Tahap *Fine-tuning*: Model yang telah memiliki bobot parameter terlatih (*Pre-trained*) Diaktivasi ulang untuk tugas yang lebih spesifik (*Downstream Tasks*), seperti analisis sentimen. Pada tahap ini, *hyperparameter* disesuaikan kembali menggunakan dataset yang telah memiliki label, seperti data opini dari platform Twitter/X [80].

3. Relevansi dalam Analisis Sentimen

   Keunggulan utama BERT dalam analisis sentimen terletak pada kemampuannya dalam menangani ambiguitas bahasa dan konteks yang kompleks. Dengan melakukan sedikit modifikasi pada lapisan *output* terakhir, representasi vektor dari token khusus Classification dapat digunakan sebagai input untuk *classification layer*. Hal ini memungkinkan model untuk menentukan polaritas sentimen (positif, negatif, atau netral) dari sebuah teks secara akurat berdasarkan pemahaman konteks menyeluruh, bukan sekadar berdasarkan keberadaan kata kunci tertentu [76].

2.5.2 IndoBERT

IndoBERT merupakan model bahasa berbasis *Transformer* yang dikembangkan khusus untuk memproses bahasa Indonesia dengan performa tinggi pada berbagai tugas *Natural Language Processing (*NLP*)*, termasuk analisis sentimen. Model ini dibangun di atas fondasi BERT (*Bidirectional Encoder Representations from Transformer*s), namun dilatih menggunakan korpus bahasa Indonesia yang masif untuk memahami nuansa linguistik lokal

[81]. Karakteristik utama IndoBERT terletak pada mekanismenya yang bersifat *Bidirectional*, di mana IndoBERT mengadopsi mekanisme *Bidirectional* dari BERT sehingga mampu memahami konteks kata berdasarkan hubungan antarkata dalam kalimat secara menyeluruh [76]. Hal ini memungkinkan IndoBERT menangkap makna kata yang ambigu berdasarkan kata-kata di sekitarnya dalam sebuah kalimat.

a. Arsitektur dan Representasi *Input*

IndoBERT (varian base) memiliki struktur yang terdiri dari 12 *hidden layers*, 768 *hidden units*, dan 12 *attention head* [82]. Dalam memproses data tekstual, IndoBERT melakukan transformasi input melalui tiga jenis *embedding* utama yang dijumlahkan untuk membentuk representasi vektor tunggal:

1. Token *Embedding*: Mengonversi kalimat menjadi unit-unit kecil yang disebut *token*s menggunakan metode *WordPiece Tokenization*. Proses ini diawali dengan penambahan token khusus sebagai representasi klasifikasi di awal kalimat dan sebagai pemisah antar kalimat [83].
2. *Segment Embedding*: Digunakan untuk membedakan antara pasangan kalimat. Dalam analisis sentimen yang umumnya terdiri dari satu kalimat tunggal, seluruh token akan diberi Indeks Segmen 0 [84].
3. *Positional Embedding*: Karena *Transformer* tidak memiliki rekursi atau konvolusi, *Positional Embedding* ditambahkan untuk memberikan informasi mengenai posisi absolut setiap token dalam urutan kalimat [85].

b. Mekanisme Kerja IndoBERT

Secara matematis, kekuatan IndoBERT terletak pada mekanisme *Self-Attention*. Setiap token dalam kalimat berinteraksi dengan token lainnya untuk menghitung bobot relevansi. Representasi *output* dari lapisan terakhir untuk token digunakan sebagai input bagi *classification layer* akhir dalam analisis sentimen [72].

c. Proses *Fine-tuning*

Dalam penelitian analisis sentimen, IndoBERT yang telah dilatih sebelumnya (*Pre-trained* Model) menjalani proses *Fine-tuning*. Pada tahap ini, *hyperparameter* disesuaikan kembali menggunakan dataset spesifik [86]. Model tidak hanya menggunakan pengetahuan bahasa yang bersifat umum, tetapi juga mempelajari pola sentimen (positif, negatif, atau netral) yang terdapat dalam korpus data tersebut [87].

Penggunaan IndoBERT terbukti lebih unggul dibandingkan model mesin pembelajaran tradisional karena kemampuannya dalam menangani ketergantungan jarak jauh antarkata dan memahami konteks kalimat yang kompleks

d. Model Pelabelan Sentimen Otomatis

merupakan model yang digunakan untuk memberikan label sentimen pada data teks secara otomatis berdasarkan hasil prediksi model pembelajaran mesin atau *deep learning*. Salah satu model yang banyak digunakan untuk klasifikasi sentimen bahasa Indonesia adalah *mdhugol/indonesia-bert-sentiment-classification*, yaitu model berbasis IndoBERT yang telah melalui proses *fine-tuning*. Model ini mampu mengklasifikasikan teks ke dalam tiga kategori sentimen, yaitu positif, negatif, dan netral, serta menghasilkan *confidence score* terhadap setiap prediksi[88].

e. *Classification layer* dan Fungsi *Softmax*

Setelah proses *fine-tuning*, representasi keluaran dari token *Classification* (CLS) pada lapisan terakhir IndoBERT diteruskan ke *classification layer* untuk menentukan kelas sentimen. Lapisan ini menghasilkan nilai logit untuk setiap kelas sentimen yang kemudian dikonversi menjadi probabilitas menggunakan fungsi *Softmax*[89].

$$P(\text{kelas}_i) = \frac{e^{z_i}}{\sum_{j=1}^{3} e^{z_j}} \quad \text{..... (5)}$$

Dimana:

P(kelas$i$)= merupakan probabilitas untuk kelas ke-i

zi = merupakan nilai logit pada kelas ke-i

$\sum$ ezj = merupakan total eksponensial seluruh nilai logit

2.6 Metrik Evaluasi

Evaluasi model klasifikasi merupakan tahap yang digunakan untuk mengukur kinerja model dalam mengklasifikasikan data ke dalam kategori yang telah ditentukan [90]. Proses evaluasi dilakukan dengan membandingkan hasil prediksi model dengan label sebenarnya pada *testing set*. Hasil klasifikasi kemudian direpresentasikan dalam bentuk *Confusion Matrix*, yang terdiri dari empat komponen utama, yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)* [91]. *Confusion Matrix* digunakan untuk mengetahui jumlah prediksi yang benar maupun yang salah dari masing-masing kelas [92]. Berikut beberapa performance Metrics yang digunakan:

1. *Confusion Matrix*

   Confusion Matrix merupakan tabel evaluasi yang digunakan untuk menggambarkan performa model klasifikasi dengan membandingkan hasil prediksi model terhadap label sebenarnya pada *testing set* [93]. Melalui *Confusion Matrix*, dapat diketahui jumlah prediksi yang benar maupun salah pada setiap kelas sentiment [94]. Pada klasifikasi sentimen, *Confusion Matrix* terdiri dari empat komponen utama, yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. Nilai-nilai tersebut selanjutnya digunakan sebagai dasar dalam perhitungan metrik evaluasi seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score*[93].

2. *Accuracy*

   Akurasi merupakan ukuran yang menunjukkan proporsi jumlah prediksi yang benar dibandingkan dengan seluruh data yang diuji. Nilai akurasi dapat dihitung menggunakan persamaan berikut [59]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

   Dimana:

   TP_i = *True Positive*, data positif terklasifikasi benar pada kelas- i

   TN_i = *True Negative,* data negatif terklasifikasi benar pada kelas ke- i

   FP_i = *False Positive,* data negatif terklasifikasi salah pada kelas ke- i

   FN_i = *False Negative* (data positif terklasifikasi salah pada kelas ke- i

   n = Jumlah total kelas yang diuji.

3. *Precision*

   Presisi menunjukkan tingkat ketepatan model dalam mengklasifikasikan data sebagai kelas tertentu. Presisi dihitung dengan membandingkan jumlah prediksi benar pada suatu kelas terhadap seluruh prediksi pada kelas tersebut [95]:

$$Precision = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (FP_i + TP_i)} \quad (7)$$

   Dimana:

   TP_i = *True Positive* pada kelas ke- i

   FP_i = *False Positive* pada kelas ke- i

FP_i = *False Positive* pada kelas ke- i

n = Jumlah total kelas.

4. *Recall*

Recall menunjukkan kemampuan model dalam menemukan seluruh data yang termasuk dalam suatu kelas. Nilai *Recall* dihitung sebagai berikut [95]:

$$Recall = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n}(TP_i + FN_i)} \quad \text{..... (8)}$$

Dimana:

TP_i = *True Positive*, data positif terklasifikasi benar pada kelas- i

FN_i = *False Negative* (data positif terklasifikasi salah pada kelas ke- i

n = Jumlah total kelas yang dianalisis.

5. *F1-Score*

F1-score merupakan rata-rata harmonis antara presisi dan *Recall* yang digunakan untuk memberikan keseimbangan antara kedua metrik tersebut. Nilai *F1-score* dihitung menggunakan persamaan berikut [92]:

$$F1 = 2 \times \frac{r \times p}{r + p} \quad \text{..... (9)}$$

Dimana:

$F1$ = Nilai *F1-score*

r = Nilai *Recall*

p = Nilai *Precision*

2.7 Penelitian Terdahulu

Sebagai landasan dalam melakukan penelitian ini, diperlukan tinjauan terhadap beberapa studi relevan yang telah dilakukan sebelumnya. Rangkuman mengenai perbandingan metodologi serta hasil dari penelitian-penelitian tersebut disajikan dalam tabel berikut.

Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti, Judul, Tahun	Metode	Dataset Yang Digunakan	Hasil
1	Yuma Akbar dan Tri Sugiharto, "Analisis Sentimen Pengguna Twitter di Indonesia Terhadap ChatGPT Menggunakan Algoritma	Algoritma C4.5 dan Naïve Bayes (menggunakan RapidMiner)	500 data cuitan dari Twitter/X dengan kata kunci "ChatGPT"	Algoritma Naïve Bayes memiliki performa lebih baik dengan tingkat akurasi sebesar 78,00%, sementara algoritma

	C4.5 dan Naïve Bayes", 2023 [96]			C4.5 menghasilkan akurasi sebesar 74,00%.
2	Muhammad Galih Ramaputra dan Hendri Purnomo, "Analisis Sentimen Opini Masyarakat Terhadap Penggunaan ChatGPT di Bidang Pendidikan Berbasis Twitter", 2024 [97]	Decision Tree dan k-Nearest Neighbors (k-NN)	1.000 data cuitan Twitter/X (*scraping* menggunakan Python/Twint) terkait ChatGPT di bidang pendidikan.	Algoritma k-NN memberikan akurasi tertinggi sebesar 82,5% pada perbandingan *training set* dan uji tertentu, mengungguli Decision Tree dalam mengklasifikasikan sentimen positif dan negatif.
3	Kardandi Alfarizi Siregar, Salsabila Nasution, dan Putri Nabawy, "Analisis Sentimen Netizen Indonesia Terhadap Kampanye Penggunaan Kecerdasan Buatan Oleh Pemerintah Menggunakan Algoritma Naive Bayes", 2025 [98]	Algoritma Naive Bayes	Data dari platform Twitter terkait kampanye penggunaan AI oleh pemerintah Indonesia.	Mayoritas sentimen netizen bersifat negatif (kekhawatiran akan regulasi dan etika). Model Naive Bayes berhasil mengklasifikasikan data dengan performa yang cukup baik untuk pemetaan opini publik.
4	Lina Rhomaningtias, Adenda Khairunisa, Shindi Shella May Wara, dan Kartika Maulida Hindrayani, "Analisis Sentimen Ulasan Aplikasi SMILE Indonesia Menggunakan Metode Naive Bayes dan *Support Vector Machine (*SVM*)*", 2025 [99]	Naive Bayes dan *Support Vector Machine (*SVM*)*	383 ulasan pengguna aplikasi SMILE Indonesia yang diambil dari Google Play Store.	Metode SVM menunjukkan performa yang lebih stabil dan unggul dalam mengklasifikasikan ulasan aplikasi dibandingkan Naive Bayes, terutama pada data teks ulasan yang beragam.
5	Riski Nursafitri, Sabarudin, dan Nur Munajat, "Penggunaan Teknologi ChatGPT Terhadap Efisiensi Penyelesaian Tugas Karya Ilmiah di Kalangan Mahasiswa", 2025 [100]	Kuantitatif dengan teknik Survei (Deskriptif)	43 responden mahasiswa Fakultas Ilmu Tarbiyah dan Keguruan UIN Sunan Kalijaga yang pernah menggunakan ChatGPT.	Penggunaan ChatGPT terbukti efektif meningkatkan efisiensi mahasiswa dalam mengumpulkan data dan menyusun kerangka karya ilmiah, meskipun terdapat risiko terkait orisinalitas karya.