BAB 2
LANDASAN TEORI

2.1 Penelitian Terkait

Studi sebelumnya telah mengembangkan kerangka kerja berjudul *"A Novel Comprehensive Framework for Detecting and Understanding Health-Related Misinformation"* untuk mendeteksi disinformasi kesehatan yang berfokus pada isu COVID-19. Model yang digunakan merupakan teknik NLP berbasis BERT, Bi-LSTM, dan mekanisme attention untuk mengklasifikasikan informasi ke dalam kategori benar atau palsu dengan tingkat akurasi yang tinggi [10]. Penelitian tersebut menerapkan Optuna dalam optimasi *hyperparameter* dengan mengeksplorasi *search space* yang mencakup *learning rate*, *batch size*, *dropout rate*, dan ukuran *hidden layer* Bi-LSTM. Model ini menunjukkan efektivitas dalam mendeteksi disinformasi dengan meraih akurasi 99,47%. Namun, cakupan penelitian terbatas pada satu topik spesifik dalam domain kesehatan (COVID-19), sehingga belum tentu dapat digeneralisasi ke topik kesehatan lainnya. Rincian metrik performa ditunjukkan pada Tabel 2.1.

Martinez-Rico et al. [11] mengkaji pendekatan deteksi disinformasi kesehatan menggunakan model ensemble BlueBERT-FFNN untuk *fact-checking* yang mengintegrasikan teks, triplet SPO (*subject-predicate-object*), dan terminologi medis UMLS. Penelitian berjudul *"Building a Framework for Fake News Detection in the Health Domain"* ini mengembangkan kerangka kerja enam tahap, mulai dari pengumpulan data otomatis hingga klasifikasi artikel menggunakan BERT Uncased. Evaluasi menunjukkan akurasi 89,4% untuk *check-worthiness*, 70,1% untuk *fact-checking* kalimat, dan 77,0% untuk klasifikasi artikel. Namun, penelitian ini memiliki beberapa keterbatasan signifikan, yaitu hanya mendukung bahasa Inggris, ketergantungan pada terminologi UMLS yang tidak tersedia untuk bahasa lain, adanya *error propagation* antar tahap dengan tingkat kesalahan *fact-checking* sebesar 28,69%, serta ukuran dataset yang relatif kecil yaitu hanya 327 artikel.

Machova et al. [12] mengintegrasikan CNN, LSTM, dan Word2Vec *embedding* dalam penelitian deteksi disinformasi kesehatan terkait COVID-19. Penelitian berjudul *"Deep Learning in the Detection of Disinformation about COVID-19 in Online Space"* mengeksplorasi performa model pada

dataset berbahasa Inggris. Hasil evaluasi menunjukkan bahwa arsitektur LSTM mengungguli CNN dengan meraih *F1-score* terbaik sebesar 87,41%. Temuan ini mengindikasikan efektivitas arsitektur *sequential* seperti LSTM dibandingkan pendekatan *convolutional* seperti CNN dalam menangkap pola temporal pada teks disinformasi kesehatan.

Di Sotto dan Viviani [24] dalam penelitian berjudul *"Health Misinformation Detection in the Social Web: An Overview and a Data Science Approach"* membahas deteksi disinformasi kesehatan secara otomatis pada konten web dan media sosial menggunakan pendekatan *machine learning* dan *deep learning*, mencakup algoritma klasik seperti Random Forest dan Logistic Regression, serta model CNN dan Bi-LSTM berbasis word embedding GloVe. Evaluasi pada tiga dataset publik berbahasa Inggris menunjukkan hasil terbaik AUC 0,973 dan F-Measure 0,953 pada dataset CoAID menggunakan CNN, namun performa menurun pada dataset FakeHealth dengan AUC tertinggi hanya 0,717. Penelitian ini belum menguji pendekatan berbasis transformer dan seluruh dataset yang digunakan terbatas pada Bahasa Inggris, sehingga belum dapat diterapkan langsung untuk konteks Bahasa Indonesia.

Cianciulli et al. [13] dalam penelitian berjudul *"Artificial Intelligence and Digital Technologies Against Health Misinformation: A Scoping Review of Public Health Responses"* melakukan pemetaan sistematis implementasi kecerdasan buatan dan teknologi digital dalam menangkal misinformasi kesehatan periode 2017–2022. *Review* tersebut mencakup 63 studi dengan empat pendekatan utama: pemantauan konten, pengembangan model AI/ML, komunikasi kesehatan, serta keterlibatan digital masyarakat. Hasil analisis menunjukkan bahwa model berbasis deep learning seperti BERT-LSTM mencapai akurasi hingga 93,5%. Namun, mayoritas studi berasal dari wilayah Amerika dan Eropa sehingga kurang merepresentasikan konteks Asia Tenggara. *Review* ini secara eksplisit merekomendasikan pengembangan dataset multibahasa dan studi lintas budaya, yang memperkuat urgensi penelitian deteksi misinformasi kesehatan berbasis bahasa Indonesia.

Tabel 2.1. Penelitian Terkait Deteksi Disinformasi Kesehatan

No.	Penulis	Model/Metode	Metrik Performa	State of Arts	Keterbatasan
1	Padalko et al. (2025)	BERT + Bi-LSTM + Attention + Optuna	Accuracy: 99,47% Precision: 0,9947 Recall: 0,9947 F1-Score: 0,9947	Model NLP berbasis BERT dan Bi-LSTM dengan optimasi hyperparameter Optuna mencapai performa tertinggi dalam deteksi disinformasi kesehatan COVID-19	Hanya mencakup satu topik (COVID-19), belum tergeneralisasi ke topik kesehatan lain
2	Martinez-Rico et al. (2024)	BlueBERT-FFNN + SPO Triplet + UMLS	F1-Score: 0,749 (check-worthiness) 0,698 (fact-checking) 0,703 (klasifikasi artikel)	Kerangka kerja enam tahap integrasi teks, triplet SPO, dan terminologi medis UMLS untuk fact-checking artikel kesehatan	Hanya Bahasa Inggris, dataset kecil (327 artikel), error propagation antar tahap
3	Machova et al. (2022)	CNN, LSTM, Word2Vec	Accuracy: 0,8628 F1-Score: 0,8741 (LSTM terbaik)	LSTM mengungguli CNN dalam menangkap pola temporal teks disinformasi kesehatan COVID-19	Dataset hanya Bahasa Inggris
4	Di Sotto & Viviani (2022)	Random Forest, LR, CNN, Bi-LSTM + GloVe	AUC: 0,973 F-Measure: 0,953 (CNN, CoAID)	CNN terbaik pada dataset CoAID, namun performa menurun signifikan pada dataset FakeHealth (AUC 0,717)	Belum menguji transformer, seluruh dataset Bahasa Inggris
5	Cianciulli et al. (2025)	BERT-LSTM (*scoping review*)	Accuracy: 93,5% (BERT-LSTM)	Pemetaan sistematis 63 studi AI untuk misinformasi kesehatan 2017–2025	Mayoritas studi dari Amerika & Eropa, minim representasi Asia Tenggara

2.2 Disinformasi Kesehatan di Indonesia

Suatu disinformasi yang beredar merupakan informasi palsu yang dengan sengaja disebarkan untuk menyesatkan [6]. Disinformasi ini diperparah oleh keberagaman konten sehingga masyarakat, khususnya yang menggunakan media sebagai sumber bacaan, cenderung mudah mempercayai dan mengadopsinya sebagai rujukan utama [7]. Negara Indonesia telah menghadapi permasalahan disinformasi sejak 2014 dengan berbagai topik, dan baru-baru ini semenjak isu COVID-19, topik kesehatan menjadi salah satu yang paling krusial [4]. Di Indonesia terdapat lebih dari 1.800 jenis disinformasi dengan proporsi masyarakat yang pernah terpapar berkisar antara 30–60% [4].

Dari permasalahan ini, dapat dilihat bahwa dampak infodemik di Indonesia terkadang lebih berbahaya daripada penyakit itu sendiri sehingga menjadi ancaman nyata bagi kesehatan masyarakat. Indonesia memiliki tingkat literasi digital yang rendah, berada pada peringkat ke-52 dari 62 negara secara global, meskipun

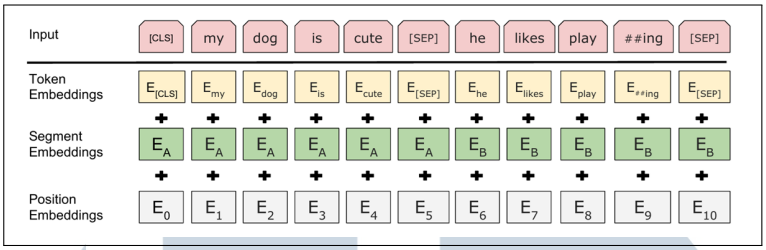
tingkat penetrasi pengguna internet mencapai 71,1% [4]. Tercatat pada tahun 2018 hingga 2023, disinformasi seputar isu kesehatan di Indonesia dilaporkan mencapai 2.256 kasus [6]. Disinformasi kesehatan ini semakin rentan tersebar seiring perkembangan zaman apabila masyarakat tidak mampu membedakan informasi yang benar dan salah.

2.3 Model Transformer

Model Transformer merupakan model *deep learning* berbasis arsitektur *encoder-decoder* yang diperkenalkan oleh Google pada tahun 2017, dirancang untuk mengekstraksi fitur teks secara lebih unggul dibandingkan model sebelumnya seperti LSTM [25]. Keunggulan arsitektur ini mendorong berkembangnya berbagai model pra-latih berbasis Transformer yang kini menjadi standar mutakhir dalam penyelesaian tugas Natural Language Processing (NLP), di mana beberapa model yang paling populer dan banyak digunakan adalah BERT (Bidirectional Encoder Representations from Transformers) dan GPT (Generative Pre-Training) [26]. Seiring perkembangannya, model-model turunan berbasis Transformer terus bertambah, seperti XLNet, RoBERTa, GPT-2, ALBERT, dan berbagai variannya yang masing-masing dirancang untuk mengoptimalkan kinerja pada tugas-tugas NLP tertentu [25].

2.3.1 Embedding pada Model Transformer

Sebelum token hasil tokenisasi diproses melalui mekanisme *Self-Attention*, setiap model Transformer terlebih dahulu mengubah token tersebut menjadi representasi vektor melalui lapisan *embedding*. Representasi embedding suatu token pada arsitektur BERT merupakan hasil penjumlahan dari tiga jenis embedding, yaitu *token embedding* yang merepresentasikan identitas kata atau *subword*, *segment embedding* yang membedakan kalimat pertama dan kedua dalam satu pasangan input, serta *position embedding* yang menyimpan informasi urutan posisi token dalam sekuens, mengingat mekanisme *Self-Attention* yang akan memproses representasi ini pada dasarnya tidak memiliki notasi urutan bawaan seperti pada model rekuren [27].



Gambar 2.1. Representasi input BERT sebagai penjumlahan *token embedding*, *segment embedding*, dan *position embedding*

Sumber: [15]

Sebagaimana diilustrasikan pada Gambar 2.1, setiap token pada sepasang kalimat input, termasuk token khusus `[CLS]` di awal sekuens, `[SEP]` sebagai pemisah antar kalimat, serta hasil pemecahan *subword* seperti pada kata ”playing” yang terpecah menjadi ”play” dan ”##ing” untuk kata yang jarang muncul dalam kosakata, diberi *token embedding* sesuai identitasnya, *segment embedding* yang menandai asal kalimat (E_A untuk kalimat pertama, E_B untuk kalimat kedua), dan *position embedding* yang menandai urutan posisinya dalam sekuens [15]. Ketiga representasi tersebut kemudian dijumlahkan secara elemen demi elemen untuk membentuk satu vektor input tunggal bagi setiap token, yang selanjutnya diproses pada lapisan *encoder* melalui mekanisme *Self-Attention* dan *Multi-Head Attention* sebagaimana dijelaskan pada bagian berikutnya [15].

Vektor hasil embedding pada lapisan keluaran BERT mengandung informasi semantik yang bersifat kontekstual, di mana representasi suatu kata dapat berbeda bergantung pada konteks kalimat di sekitarnya, berbeda dengan pendekatan representasi statis seperti Word2Vec atau GloVe yang menghasilkan satu vektor tetap untuk setiap kata tanpa mempertimbangkan konteks penggunaannya [27]. Kemampuan menghasilkan representasi kontekstual inilah yang menjadi salah satu faktor utama keunggulan model berbasis Transformer dibandingkan pendekatan representasi kata sebelumnya dalam berbagai tugas pemrosesan bahasa alami [25].

2.3.2 Self-Attention

Self-Attention merupakan mekanisme utama dalam arsitektur Transformer yang memungkinkan model memahami hubungan antar token dalam suatu urutan teks tanpa harus memproses token secara berurutan seperti pada Recurrent Neural Network (RNN). Melalui mekanisme ini, setiap token dapat memperhatikan token lainnya sehingga model mampu menangkap informasi kontekstual dan

ketergantungan antar kata secara lebih efektif [25]. Pada mekanisme Self-Attention, setiap token direpresentasikan ke dalam tiga vektor, yaitu *Query* (Q), *Key* (K), dan *Value* (V).

Ketiga vektor tersebut digunakan untuk menghitung tingkat relevansi antar token melalui operasi *Scaled Dot-Product Attention*. Semakin tinggi tingkat relevansi antar token, maka semakin besar bobot perhatian (*attention weight*) yang diberikan terhadap token tersebut [25]. Perhitungan Self-Attention dapat dirumuskan sebagai berikut:

$$Attention(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

dengan Q merupakan matriks Query, K merupakan matriks Key, V merupakan matriks Value, dan d_k merupakan dimensi vektor Key yang digunakan sebagai faktor normalisasi.

Melalui proses tersebut, model dapat menentukan token-token yang paling relevan terhadap token yang sedang diproses sehingga menghasilkan representasi kontekstual yang lebih baik. Kemampuan ini menjadi salah satu faktor utama yang membuat Transformer unggul dalam berbagai tugas Natural Language Processing (NLP) [25].

2.3.3 Multi-Head Attention

Untuk meningkatkan kemampuan representasi yang dihasilkan oleh Self-Attention, Transformer menerapkan mekanisme *Multi-Head Attention*. Mekanisme ini menjalankan beberapa proses Self-Attention secara paralel sehingga model dapat mempelajari berbagai hubungan antar token secara bersamaan [25]. Setiap attention head memiliki representasi yang berbeda sehingga mampu menangkap informasi linguistik yang beragam. Hasil dari seluruh attention head kemudian digabungkan untuk membentuk representasi akhir yang lebih kaya dibandingkan penggunaan satu attention head saja [25].

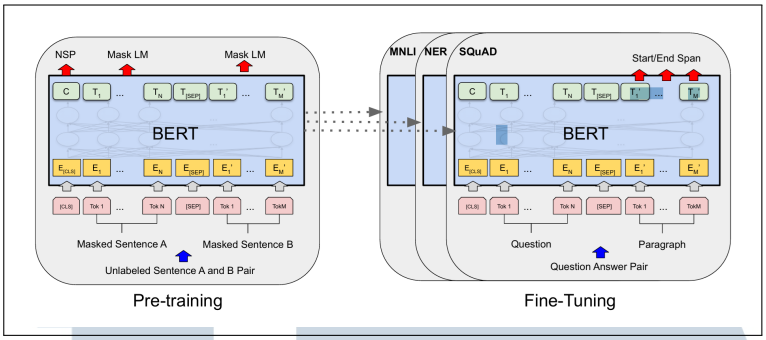
Selain meningkatkan kualitas representasi, Multi-Head Attention juga mendukung proses komputasi secara paralel sehingga membuat arsitektur Transformer lebih efisien dibandingkan pendekatan berbasis pemrosesan sekuensial [25].

2.3.4 BERT (*Bidirectional Encoder Representations from Transformers*)

BERT (*Bidirectional Encoder Representations from Transformers*) merupakan model *pre-trained language model* yang merepresentasikan bahasa berbasis arsitektur transformer dengan mekanisme pelatihan dua arah (*bidirectional*) [28]. Model ini pertama kali diperkenalkan pada tahun 2018 dan telah membawa perubahan signifikan dalam pengembangan model bahasa pra-latih di bidang pemrosesan bahasa alami (*Natural Language Processing*/NLP) [29]. Keunggulan BERT terletak pada kemampuannya menghasilkan *contextual embeddings* yang lebih representatif dibandingkan model berbasis representasi statis seperti Word2Vec dan GloVe [28].

BERT hadir dalam berbagai konfigurasi ukuran untuk mengakomodasi kebutuhan komputasi yang beragam. $\text{BERT}_{\text{Base}}$ terdiri dari 12 lapisan *encoder* dengan 110 juta parameter, sedangkan $\text{BERT}_{\text{Large}}$ hadir dengan 24 lapisan encoder dan 340 juta parameter [28]. Perbedaan konfigurasi ini memengaruhi kapasitas model dalam menangkap pola bahasa yang kompleks, namun juga berdampak pada kebutuhan sumber daya komputasi yang lebih besar [29].

BERT dioperasikan melalui dua tahapan utama, yaitu *pre-training* dan *fine-tuning*. Pada tahap *pre-training*, model dilatih secara tidak supervisi menggunakan data teks tanpa label melalui dua tugas utama: *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP). Setelah proses *pre-training* selesai, model kemudian memasuki tahap *fine-tuning* dengan menginisialisasi parameter dari hasil *pre-training* dan menyesuaikannya lebih lanjut menggunakan data berlabel spesifik domain melalui pelatihan supervisi [15]. Pendekatan dua tahap ini memungkinkan BERT memanfaatkan pengetahuan linguistik umum dari korpus besar, kemudian mengadaptasinya secara efisien untuk tugas klasifikasi atau prediksi tertentu.



Gambar 2.2. Arsitektur model BERT yang terdiri dari dua tahap: *pre-training* dan *fine-tuning*

Sumber: [15]

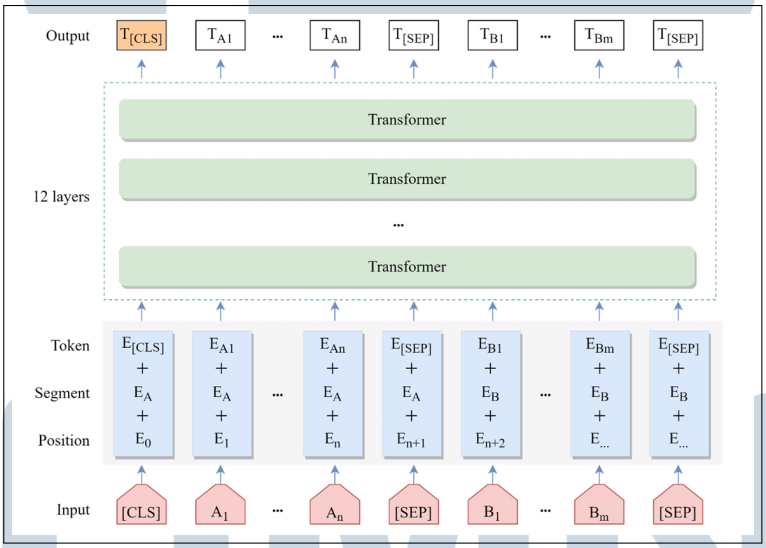
Arsitektur model BERT diilustrasikan dalam dua tahap yang berkesinambungan sebagaimana ditunjukkan pada Gambar 2.2. Pada tahap *pre-training*, model menerima pasangan kalimat tanpa label (*unlabeled sentence pair*) sebagai input, di mana setiap token direpresentasikan melalui *embedding*. Tahap ini menghasilkan dua jenis representasi: vektor `[CLS]` untuk tugas *Next Sentence Prediction* (NSP) dan representasi token (T_1 hingga T_n) untuk tugas *Masked Language Modeling* (MLM) [15]. Selanjutnya, pada tahap *fine-tuning*, seluruh parameter BERT yang telah diinisialisasi dari hasil *pre-training* disesuaikan kembali menggunakan data berlabel guna menyelesaikan tugas spesifik seperti klasifikasi teks, *Named Entity Recognition* (NER), maupun *question answering* [15].

2.3.5 IndoBERT

IndoBERT merupakan varian BERT khusus untuk bahasa Indonesia yang dikembangkan oleh tim IndoNLU. Untuk memajukan penelitian NLP berbahasa Indonesia, Wilie et al. [16] mengumpulkan korpus sekitar empat miliar kata yang diproses menjadi dataset standar Indo4B untuk pembelajaran mandiri (self-supervised learning), kemudian mengembangkan model IndoBERT yang dilatih menggunakan korpus tersebut. IndoBERT dirilis dalam dua konfigurasi: IndoBERT-base dengan 124,5 juta parameter dan IndoBERT-large dengan 335,2 juta parameter. Kedua varian dilatih menggunakan metode Masked Language Modeling (MLM) dalam dua fase dengan panjang sekuens maksimum 128 pada fase pertama dan 512 pada fase kedua [16].

2.3.6 RoBERTa (*Robustly Optimized BERT Pretraining Approach*)

RoBERTa (*Robustly Optimized BERT Pretraining Approach*) merupakan pengembangan dari model BERT (*Bidirectional Encoder Representations from Transformers*) [19]. Model ini termasuk dalam keluarga arsitektur Transformer yang dirancang untuk menghasilkan representasi *word embedding* yang lebih representatif melalui pemrosesan sekuens teks secara efektif. RoBERTa dikembangkan oleh Liu et al. pada tahun 2019 dan dipublikasikan dalam artikel berjudul *"RoBERTa: A Robustly Optimized BERT Pretraining Approach"*. Model ini menyempurnakan pendekatan pelatihan BERT dengan memanfaatkan volume data yang lebih besar, panjang sekuens input yang lebih panjang, serta durasi pelatihan yang diperpanjang [19].



Gambar 2.3. Arsitektur model RoBERTa

Sumber: [27]

BERT merupakan model bahasa *bidirectional* yang memproses teks dengan menumpuk struktur *encoder Transformer* secara parsial, sedangkan model RoBERTa menunjukkan keunggulan lebih lanjut sebagaimana diilustrasikan pada Gambar 2.3 yang menampilkan arsitektur kerja model RoBERTa. Model RoBERTa mewarisi keunggulan model BERT dengan mengadopsi *encoder Transformer bidirectional* sebagai lapisan tengah untuk mengekstraksi informasi kontekstual dari teks [27]. Setiap proses representasi *embedding* dalam teks diindikasikan menjadi berbagai jenis *embedding*. Representasi *embedding* dari teks merupakan kombinasi dari *token embedding*, *segment embedding*, dan *position embedding*

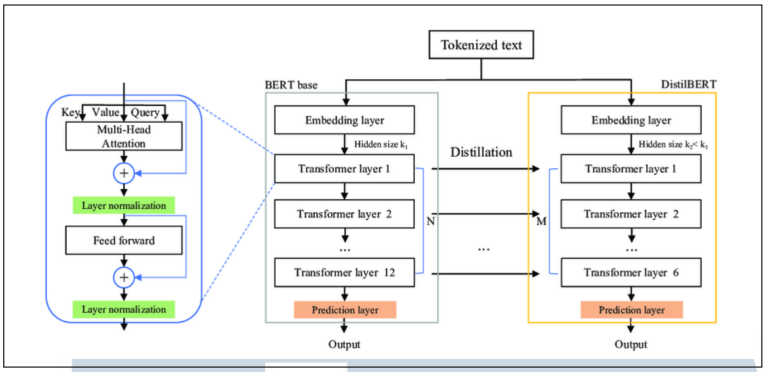
[27]. Vektor kalimat pada lapisan output model RoBERTa mengandung informasi semantik global dan kontekstual, sehingga dapat dimanfaatkan untuk menentukan kesamaan semantik antara teks pendek dengan deskripsi entitas kandidat. Pendekatan ini memungkinkan RoBERTa menangkap nuansa makna yang lebih halus dibandingkan model sebelumnya.

2.3.7 IndoRoBERTa

Indonesian RoBERTa Base merupakan model bahasa berbasis arsitektur RoBERTa yang dilatih khusus menggunakan dataset OSCAR (Open Super-Large Crawled ALManaCH corpus), yaitu korpus multibahasa berskala besar yang mencakup 166 bahasa, termasuk Bahasa Indonesia [20]. Model ini tersedia dalam beberapa varian, salah satunya Indonesian RoBERTa Small, yang telah diuji untuk tugas klasifikasi teks multikelas berbahasa Indonesia bersama model lain seperti IndoBERT dan MBERT [30]. Dalam pengujian pada tugas klasifikasi kepribadian berbahasa Indonesia, Indonesian RoBERTa Base menunjukkan performa yang baik dengan menghasilkan akurasi sebesar 0.7059, *F1-Score* 0.6781, dan ROC AUC 0.7062, sehingga terpilih sebagai salah satu model dalam proses *ensemble learning* [20].

2.3.8 DistilBERT (*Distilled BERT)*

DistilBERT merupakan salah satu varian BERT yang lebih ringkas dan efisien secara komputasi karena memiliki arsitektur yang lebih sederhana, dikembangkan oleh tim HuggingFace [21]. Proses teknis yang diintegrasikan dalam implementasi DistilBERT menggunakan konsep *knowledge distillation* sebagai landasan arsitekturnya. Teknik distilasi melibatkan pelatihan model yang lebih kecil berdasarkan model yang lebih besar yang disebut sebagai *teacher*, sedangkan model yang dilatih disebut sebagai *student* [21]. DistilBERT memiliki sekitar 50 juta parameter lebih sedikit dibandingkan BERT [21].



Gambar 2.4. Arsitektur model DistilBERT

Sumber: [31]

Secara arsitektur, DistilBERT hanya terdiri dari 6 lapisan tersembunyi (*hidden layers*) dibandingkan 12 lapisan pada BERT-base, dengan total 66 juta parameter, di mana seitu lapisan tersebut memuat komponen Multi-Head Attention, Feed Forward, dan Layer Normalization [31]. Arsitektur ini lebih ringkas dibandingkan BERT namun terbukti sekitar 45% lebih cepat dalam proses *fine-tuning* dengan tetap mempertahankan sekitar 96% kemampuan pemahaman bahasa dari BERT [21].

2.3.9 IndoDistilBERT (*Indonesian Distilled BERT*)

IndoDistilBERT merupakan varian ringkas dari model BERT yang dikembangkan khusus untuk bahasa Indonesia melalui teknik *knowledge distillation* [32]. Model ini merupakan adaptasi dari arsitektur DistilBERT yang telah di-*fine-tune* menggunakan korpus berbahasa Indonesia sehingga mampu memahami konteks dan struktur bahasa Indonesia dengan lebih baik [32]. Dalam pengujian klasifikasi teks berbahasa Indonesia, IndoDistilBERT terbukti mampu mencapai akurasi 76,38%, meskipun masih berada di bawah performa IndoBERT yang mencapai akurasi 80% [32].

2.4 *Text Augmentation*

Text augmentation merupakan teknik pengolahan data yang bertujuan mengatasi ketidakseimbangan kelas pada dataset [33]. Teknik ini menghasilkan sampel baru dengan mempertahankan makna semantik asli tetapi mengubah struktur kalimat dan pemilihan kata [33]. Beberapa metode yang umum digunakan meliputi *Easy Data Augmentation* (EDA) dan *back-translation* [33].

2.4.1 *Easy Data Augmentation*

Easy Data Augmentation (EDA) merupakan teknik augmentasi teks berbasis aturan yang menerapkan empat operasi dasar: *synonym replacement* (penggantian sinonim), *random insertion* (penyisipan kata acak), *random swap* (penukaran posisi kata), dan *random deletion* (penghapusan kata acak) [33]. Agar makna semantik asli tetap terjaga, EDA tidak mengganti kata secara sembarangan. Pada operasi penggantian dan penyisipan sinonim, sinonim diambil dari *WordNet* [33]. Untuk mencegah perubahan konteks, modifikasi EDA dapat menerapkan *POS tagging* (*Part-of-Speech tagging*) sehingga kata pengganti memiliki kelas kata yang sama dengan kata asli (misalnya kata sifat diganti dengan kata sifat lain) [33]. Selain itu, kata-kata yang mewakili aspek atau sentimen (kata kunci penentu label) tidak boleh dihapus atau ditukar; jika suatu kata teridentifikasi sebagai kata benda (*noun*) atau kata kerja (*verb*) yang memiliki kemiripan semantik tinggi dengan aspek tertentu (misal makanan, harga), maka kata tersebut dilindungi dari operasi penghapusan dan penukaran [33]. Dengan cara ini, integritas semantik data hasil augmentasi tetap terjaga, sehingga label asli tidak berubah meskipun struktur kalimat dan pilihan kata dimodifikasi [33].

2.4.2 *Back Translation*

Back-translation merupakan metode augmentasi teks yang terbukti meningkatkan performa model melalui penerjemahan bolak-balik kalimat asli ke bahasa perantara lalu kembali ke bahasa asal [33]. Berbeda dengan EDA yang beroperasi pada level kata, *back-translation* bekerja pada level kalimat utuh sehingga menghasilkan variasi linguistik yang lebih alami dan kontekstual [33]. Untuk mempertahankan makna semantik asli, *back-translation* mengandalkan penerjemah saraf (*neural machine translation*) yang mampu menjaga inti informasi meskipun struktur kalimat berubah [33]. Agar hasil terjemahan tidak persis sama dengan kalimat asli (yang dapat menyebabkan duplikasi), *stop word* pada kalimat input dihapus terlebih dahulu. Hal ini memaksa proses penerjemahan menghasilkan susunan kata dan tata bahasa yang berbeda, namun makna global tetap sama [33]. Dalam implementasinya, bahasa perantara seperti bahasa Mandarin terbukti menghasilkan variasi yang lebih beragam dibandingkan bahasa lain, karena perbedaan struktur gramatikal yang signifikan [33]. Dengan demikian, *back-translation* mampu memperkaya data latih dengan kalimat-kalimat baru yang secara

semantik setara dengan aslinya, sehingga model dapat belajar pola yang lebih bervariasi tanpa mengubah kebenaran label [33].

2.5 Hyperparameter Tuning dengan Grid Search

Hyperparameter adalah konfigurasi yang menentukan perilaku model *deep learning* selama pelatihan dan ditetapkan sebelum proses dimulai, karena nilainya tidak dipelajari dari data [34]. Optimasi *hyperparameter* bertujuan menemukan kombinasi nilai yang menghasilkan akurasi klasifikasi tertinggi pada data uji. Salah satu metode yang digunakan adalah *Grid Search*, yaitu mengevaluasi setiap kombinasi dalam ruang pencarian dan memilih konfigurasi dengan performa validasi terbaik [34].

Meskipun *Grid Search* bersifat menyeluruh dan berpotensi menemukan konfigurasi optimal, metode ini memiliki kelemahan utama berupa biaya komputasi yang tinggi karena jumlah kombinasi meningkat secara eksponensial seiring bertambahnya jumlah hyperparameter [34]. Wojciuk et al. [34] menemukan bahwa penerapan optimasi *hyperparameter* seperti grid search dapat meningkatkan akurasi model hingga 6% dibandingkan dengan penggunaan konfigurasi default [34].

2.6 Inverse Frequency Class Weights

Ketidakseimbangan distribusi kelas (*class imbalance*) merupakan tantangan umum dalam tugas klasifikasi dunia nyata, di mana model *machine learning* cenderung bias terhadap kelas mayoritas sehingga kesulitan mengenali kelas minoritas secara konsisten [35]. Salah satu strategi penanganan yang lazim digunakan adalah *inverse frequency class weights*, yaitu memberikan bobot lebih tinggi pada kelas dengan jumlah sampel lebih sedikit dan bobot lebih rendah pada kelas mayoritas, secara proporsional terbalik terhadap frekuensi kemunculannya dalam data latih [35].

Pendekatan ini diintegrasikan ke dalam fungsi loss (misalnya *Weighted Cross-Entropy*), sehingga kesalahan prediksi pada kelas minoritas diberi penalti lebih besar selama pelatihan. Hal ini mendorong model untuk lebih memperhatikan kelas yang kurang terwakili [35]. Bobot untuk setiap kelas dihitung sebagai berikut:

$$w_c = \frac{N}{K \cdot n_c} \tag{2.2}$$

di mana w_c adalah bobot kelas c, N adalah total jumlah sampel, K adalah jumlah kelas, dan n_c adalah jumlah sampel pada kelas c.

Meskipun *weighted cross-entropy* dengan bobot frekuensi terbalik umum digunakan, pendekatan statis ini belum tentu optimal untuk semua kasus ketidakseimbangan kelas, sehingga dikembangkan skema pembobotan dinamis yang lebih adaptif, namun *inverse frequency class weights* tetap menjadi *baseline* standar dalam penanganan ketidakseimbangan pada model deep learning [35].

2.7 *Evaluation Metrics*

Evaluation metrics merupakan sekumpulan ukuran kuantitatif yang digunakan untuk menilai performa model klasifikasi secara objektif [36]. Terdapat dua pendekatan utama dalam evaluasi sistem, yaitu evaluasi manusia dan evaluasi otomatis. Evaluasi otomatis lebih hemat biaya dan mudah digunakan untuk membandingkan beberapa sistem sekaligus dibandingkan evaluasi manusia yang memerlukan biaya tinggi dan tenaga yang signifikan [36]. Dalam suatu penelitian, implementasi evaluasi model menggunakan empat metrik utama, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*, untuk memberikan gambaran komprehensif terhadap kemampuan deteksi disinformasi [21]. Seluruh metrik yang digunakan memiliki rentang nilai dengan skor terbaik sebesar 1 dan skor terburuk sebesar 0 [21]. Untuk memahami perhitungan keempat metrik evaluasi tersebut, diperlukan pemahaman terlebih dahulu mengenai konsep confusion matrix yang menjadi landasan dasar seluruh perhitungan evaluasi klasifikasi.

2.7.1 *Confusion Matrix*

Confusion matrix merupakan landasan fundamental dalam evaluasi model klasifikasi pada *machine learning*, yang menyajikan hasil prediksi model dalam bentuk tabel dua dimensi dengan baris merepresentasikan kelas aktual dan kolom merepresentasikan kelas prediksi [37]. Pada klasifikasi biner, setiap instance menghasilkan salah satu dari empat kemungkinan hasil berdasarkan perbandingan prediksi dan label aktual. *True Positive* (TP) terjadi ketika model memprediksi positif dan aktualnya positif; *False Positive* (FP) ketika prediksi positif tetapi aktualnya negatif; *False Negative* (FN) ketika prediksi negatif tetapi aktualnya positif; serta *True Negative* (TN) ketika prediksi dan aktual keduanya negatif [37]. Keempat komponen tersebut dapat direpresentasikan dalam bentuk tabel sebagai

berikut.

Tabel 2.2. Confusion matrix klasifikasi biner.

	Prediksi Positif	**Prediksi Negatif**
Aktual Positif	*True Positive* (TP)	*False Negative* (FN)
Aktual Negatif	*False Positive* (FP)	*True Negative* (TN)

sumber: [37]

2.7.2 *Precision*

Precision merupakan metrik yang mengukur ketepatan model dalam memprediksi kelas positif, di mana nilai tinggi menunjukkan bahwa sebagian besar prediksi positif model benar-benar merupakan kasus positif [37].

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

2.7.3 *Recall*

Recall mengukur kemampuan model menemukan seluruh kasus positif yang sebenarnya ada; nilai recall tinggi berarti model mampu mendeteksi sebagian besar konten positif tanpa melewatkan banyak kasus [37].

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

2.7.4 F_1-*Score*

F_1-*Score* merupakan metrik gabungan yang menyeimbangkan precision dan recall dalam satu nilai harmonis [37].

$$F_1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.5}$$

2.7.5 *Accuracy*

Accuracy mengukur proporsi keseluruhan prediksi yang benar, namun dapat menyesatkan pada dataset tidak seimbang karena tidak membedakan jenis kesalahan klasifikasi [37].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

2.8 LIME (*Local Interpretable Model-agnostic Explanations*)

LIME (*Local Interpretable Model-agnostic Explanations*) merupakan teknik *explainability* yang dirancang untuk menginterpretasikan keputusan model *machine learning* yang kompleks secara lokal, dengan cara mengganggu (*perturb*) data input melalui penggantian atau penghapusan kata, kemudian menganalisis dampaknya terhadap output prediksi [38]. Pendekatan ini menghasilkan aproksimasi lokal menggunakan model sederhana seperti regresi linear yang meniru perilaku model kompleks di sekitar suatu instance, sehingga kontribusi tiap kata terhadap prediksi dapat diukur dan diidentifikasi secara kuantitatif [38].

Secara matematis, algoritma LIME diformulasikan sebagai berikut [38]:

$$\xi(x) = \arg\min_{g \in G} [L(f, g, \pi_x) + \Omega(g)] \tag{2.7}$$

di mana $\xi(x)$ adalah penjelasan optimal untuk instance x, f adalah model kompleks asli, g adalah model *surrogate* sederhana yang dipilih dari keluarga model G, L mengukur kedekatan aproksimasi g terhadap f di sekitar instance x menggunakan bobot lokalitas π_x, serta $\Omega(g)$ merupakan fungsi regularisasi yang menjaga kesederhanaan penjelasan agar tetap mudah dipahami [38].

Algoritma LIME mampu menyorot kata-kata yang berkontribusi signifikan terhadap suatu prediksi, baik yang bersifat positif maupun negatif, disertai dengan skor kontribusi masing-masing kata [38]. Selain itu, LIME menghasilkan visualisasi teks berbasis warna untuk membedakan pengaruh tiap kata, di mana warna merah menandakan kontribusi terhadap kelas positif dan warna biru terhadap kelas negatif, sehingga analisis kesalahan klasifikasi dapat dilakukan secara lebih intuitif [38]. Karena bersifat *model-agnostic*, LIME dapat diterapkan pada berbagai

arsitektur model tanpa bergantung pada struktur internalnya, sehingga berperan penting dalam meningkatkan transparansi dan kepercayaan terhadap keputusan yang dihasilkan oleh model [38].