BAB 2
LANDASAN TEORI

Bab ini memaparkan landasan teori yang menjadi fondasi teoretis penelitian analisis sentimen tweet berbahasa Indonesia menggunakan model hybrid IndoBERT-LSTM. Pembahasan dimulai dari konsep analisis sentimen sebagai bidang kajian utama, diikuti dengan Twitter/X sebagai sumber data yang digunakan, pendekatan *deep learning* dalam pemrosesan bahasa alami, arsitektur model yang diterapkan mencakup LSTM (beserta varian *Bidirectional*-nya) dan IndoBERT beserta kombinasi *hybrid*-nya, teknik penanganan ketidakseimbangan kelas dan optimasi hiperparameter otomatis, serta metode evaluasi yang digunakan untuk mengukur kinerja model. Bab ini ditutup dengan identifikasi *research gap* yang memetakan posisi penelitian ini terhadap penelitian-penelitian terdahulu yang paling relevan, khususnya dalam konteks pengembangan model *hybrid* berbasis Transformer untuk analisis sentimen teks berbahasa Indonesia pada domain olahraga nasional.

2.1 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 [10]. 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 [11]. Dalam era digital di mana volume data tekstual dari media sosial meledak secara eksponensial, analisis sentimen menjadi alat yang sangat krusial bagi organisasi, peneliti, dan pembuat kebijakan untuk memahami persepsi publik secara berbasis data.

Secara historis, pendekatan analisis sentimen berevolusi melalui tiga generasi utama. Generasi pertama mengandalkan metode berbasis aturan dan leksikon yang menggunakan kamus kata dengan skor polaritas untuk mengklasifikasikan sentimen. Meskipun sederhana, pendekatan ini kesulitan menangani konteks, ironi, dan sarkasme yang umum ditemukan dalam teks informal media sosial [12]. Generasi kedua menggunakan pendekatan *machine learning* klasik seperti *Naive Bayes* dan SVM yang memanfaatkan fitur-fitur linguistik yang

direkayasa secara manual. Nugraha dan Zakiyah [13] mencontohkan pendekatan ini dengan menerapkan *text mining transformation* pada data Twitter Timnas Indonesia, namun menemukan keterbatasan dalam mengklasifikasikan sentimen minoritas secara akurat. Generasi ketiga dan paling mutakhir adalah pendekatan berbasis *deep learning* yang mampu mempelajari representasi fitur secara otomatis dari data mentah, terbukti memberikan kinerja jauh lebih unggul untuk analisis sentimen teks berbahasa Indonesia [6].

Dalam konteks analisis sentimen pada media sosial Twitter/X, terdapat tantangan unik yang perlu diatasi. *Tweet* berbahasa Indonesia sering kali mengandung bahasa gaul, singkatan tidak baku, ekspresi informal, dan campur kode antara Bahasa Indonesia dengan bahasa daerah maupun bahasa asing [3]. Nanda dkk. [4] mengidentifikasi tantangan ini dalam penelitiannya pada domain sepak bola Indonesia dan menunjukkan bahwa pendekatan berbasis *word embedding* seperti FastText lebih mampu menangkap representasi kata informal dibandingkan TF-IDF konvensional. Tantangan-tantangan inilah yang memperkuat argumen penggunaan model berbasis Transformer seperti IndoBERT yang telah dilatih secara khusus pada korpus Bahasa Indonesia.

Selain tantangan pada bahasa informal, pendekatan pelabelan otomatis berbasis leksikon umum seperti InSet (*Indonesia Sentiment Lexicon*) juga memiliki keterbatasan ketika diterapkan pada domain spesifik seperti olahraga. Leksikon umum disusun berdasarkan korpus berbahasa Indonesia yang luas dan tidak mempertimbangkan konotasi khusus suatu domain, sehingga kata-kata yang netral atau bahkan negatif dalam konteks umum dapat berkonotasi sangat positif dalam konteks olahraga, dan sebaliknya. Hal ini mendorong kebutuhan akan kamus sentimen domain-spesifik yang disusun secara khusus untuk merepresentasikan kosakata dan ekspresi khas suatu komunitas, sebagaimana diterapkan pada penelitian ini (lihat Bab 3).

Pelabelan sentimen merupakan proses pemberian label pada setiap dokumen atau *tweet* berdasarkan polaritas opini yang terkandung di dalamnya. Secara umum, analisis sentimen mengelompokkan data ke dalam tiga kelas utama, yaitu positif, negatif, dan netral. Sentimen positif menggambarkan opini yang mengandung apresiasi, dukungan, kepuasan, maupun penilaian yang menguntungkan terhadap suatu objek. Sebaliknya, sentimen negatif menunjukkan adanya kritik, ketidakpuasan, kekecewaan, atau penilaian yang merugikan terhadap objek yang dibahas. Adapun sentimen netral merepresentasikan informasi, fakta, atau pendapat yang tidak menunjukkan kecenderungan sentimen positif maupun

negatif secara dominan [11].

Secara umum, pelabelan sentimen dapat dilakukan melalui beberapa pendekatan, yaitu anotasi manual oleh pakar (*manual annotation*), pelabelan berbasis kamus sentimen (*lexicon-based labeling*), maupun pelabelan otomatis (*weak supervision*) menggunakan aturan tertentu. Pada pendekatan berbasis kamus, setiap kata atau frasa diberikan skor sentimen yang kemudian diakumulasikan untuk menghasilkan skor keseluruhan suatu dokumen. Apabila skor akhir menunjukkan kecenderungan positif, maka dokumen diberi label positif; apabila menunjukkan kecenderungan negatif, maka dokumen diberi label negatif; sedangkan dokumen yang tidak menunjukkan kecenderungan terhadap kedua polaritas tersebut dikategorikan sebagai netral. Dalam penelitian ini digunakan pendekatan *weak supervision* berbasis kamus sentimen domain sepak bola yang disusun secara khusus untuk konteks Timnas Indonesia. Mekanisme pelabelan tersebut dijelaskan lebih rinci pada Bab 3.

2.2 X sebagai Sumber Data Analisis Sentimen

Twitter, yang kini dikenal sebagai X, merupakan salah satu platform media sosial terbesar di dunia yang menjadi sumber data utama untuk penelitian analisis sentimen. Platform ini menawarkan karakteristik yang sangat berharga bagi analisis sentimen, yaitu bersifat publik, *real-time*, spontan, dan kaya akan ekspresi emosi otentik dari penggunanya [13]. Di Indonesia, Twitter/X menjadi salah satu platform diskusi utama untuk topik-topik yang berkaitan dengan olahraga nasional, termasuk perbincangan seputar performa Timnas Indonesia dalam berbagai kompetisi internasional.

Keunggulan Twitter/X sebagai sumber data analisis sentimen dalam domain olahraga telah dibuktikan oleh berbagai penelitian. Abidin dan Pamungkas [5] berhasil mengumpulkan sebanyak 21.045 komentar dari akun resmi Timnas Indonesia untuk menganalisis sentimen publik selama Piala Asia 2023. Naufal dkk. [2] juga memanfaatkan data Twitter/X dengan kata kunci spesifik terkait Timnas Indonesia di Piala Asia 2023 dan berhasil memetakan distribusi sentimen publik secara komprehensif. Hal ini menunjukkan bahwa data Twitter/X dapat memberikan gambaran yang akurat dan *real-time* mengenai dinamika opini publik terhadap Timnas Indonesia dalam kompetisi internasional.

Sebelum data *tweet* dapat digunakan untuk pemodelan, diperlukan serangkaian tahap *preprocessing* untuk membersihkan dan menormalkan teks dari

noise. Nanda dkk. [4] dalam penelitiannya pada domain sepak bola Indonesia menerapkan tahapan *preprocessing* yang mencakup pembersihan teks, *case folding*, tokenisasi, penghapusan *stopword*, dan normalisasi kata tidak baku. Elhan dkk. [6] juga menegaskan bahwa kualitas *preprocessing* sangat memengaruhi performa akhir model, di mana dataset yang telah melalui *preprocessing* komprehensif menghasilkan akurasi BERT yang lebih tinggi dibandingkan data tanpa perlakuan serupa. Tahapan *preprocessing* teks Twitter/X dalam penelitian ini meliputi:

1. **Pembersihan teks:** Menghapus URL, *mention* (@username), *hashtag* (#), karakter khusus, angka, dan tanda baca berlebihan yang tidak berkontribusi pada makna sentimen.
2. **Case folding:** Mengubah seluruh teks menjadi huruf kecil untuk menyeragamkan representasi kata yang ditulis dengan variasi kapitalisasi berbeda.
3. **Normalisasi kata:** Menstandarisasi kata tidak baku, singkatan, dan bahasa gaul khas *tweet* berbahasa Indonesia ke dalam bentuk yang dapat dikenali oleh model, menggunakan kamus normalisasi yang memetakan kata informal ke bentuk formalnya.
4. **Penghapusan *stopword*:** Menghapus kata-kata umum Bahasa Indonesia yang tidak berkontribusi signifikan terhadap makna sentimen.
5. **Stemming:** Mengubah setiap kata ke bentuk dasarnya untuk menyeragamkan variasi morfologis kata yang berasal dari akar kata yang sama.
6. **Tokenisasi:** Memecah teks menjadi satuan token individual menggunakan tokenizer IndoBERT yang dirancang khusus untuk Bahasa Indonesia, dilakukan setelah teks selesai dibersihkan dan dinormalisasi.

2.3 Deep Learning untuk NLP

Deep Learning merupakan cabang dari *machine learning* yang mengandalkan arsitektur *Artificial Neural Networks* (ANN) berlapis-lapis untuk mempelajari representasi data secara hierarkis langsung dari data mentah [12]. Keunggulan utama *deep learning* dibandingkan metode klasik terletak pada kemampuannya mengekstrak fitur secara otomatis tanpa memerlukan rekayasa fitur

manual, yang sangat menguntungkan untuk data tidak terstruktur seperti teks tweet berbahasa Indonesia yang kaya akan ekspresi informal.

Era modern NLP didominasi oleh arsitektur Transformer yang mengandalkan mekanisme *self-attention* untuk memproses semua kata dalam kalimat secara simultan dan dua arah [14]. Berbeda dengan pendekatan sebelumnya yang memproses teks secara sekuensial, arsitektur Transformer mampu memahami konteks sebuah kata berdasarkan seluruh kata lain dalam kalimat secara paralel, menghasilkan representasi kontekstual yang jauh lebih kaya. Shofwatul [15] memperkuat hal ini dengan membuktikan bahwa model berbasis Transformer seperti BERT secara konsisten mengungguli metode *deep learning* konvensional pada tugas klasifikasi sentimen *multiclass* di Twitter.

Dalam alur kerja proyek NLP untuk klasifikasi sentimen, terdapat beberapa tahapan sistematis yang harus dilalui. Marpaung dan Tarigan [3] menerapkan alur ini secara komprehensif, dimulai dari pengumpulan data Twitter/X, dilanjutkan dengan *preprocessing*, pelabelan otomatis menggunakan InSet Lexicon, tokenisasi, pemodelan, hingga evaluasi menggunakan metrik akurasi dan F1-*score*. Alur kerja yang sistematis ini menjadi fondasi metodologi penelitian dalam domain analisis sentimen teks berbahasa Indonesia.

2.4 Metode Pemodelan

Tahap pemodelan pada penelitian ini menerapkan arsitektur *hybrid* yang menggabungkan IndoBERT sebagai *feature extractor* berbasis Transformer dengan *Bidirectional Long Short-Term Memory* (BiLSTM) sebagai lapisan pemodelan sekuensial. Berbeda dengan LSTM konvensional yang hanya memproses urutan token dari satu arah (*forward*), BiLSTM memproses urutan token dari dua arah, yaitu *forward* dan *backward*, sehingga mampu memanfaatkan informasi konteks sebelum dan sesudah suatu token secara bersamaan. Kemampuan tersebut membuat BiLSTM lebih efektif dalam menangkap hubungan antar kata pada kalimat dibandingkan LSTM satu arah.

Kombinasi IndoBERT dan BiLSTM dipilih berdasarkan penelitian Putra dkk. [8] yang menunjukkan bahwa integrasi representasi kontekstual dari IndoBERT dengan kemampuan BiLSTM dalam memodelkan dependensi sekuensial menghasilkan peningkatan performa klasifikasi teks dibandingkan penggunaan masing-masing arsitektur secara terpisah.

2.4.1 Long Short-Term Memory (LSTM)

Recurrent Neural Network (RNN) merupakan salah satu arsitektur *deep learning* yang dirancang untuk memproses data sekuensial (*sequential data*), seperti teks, suara, maupun deret waktu, dengan memanfaatkan informasi dari urutan sebelumnya untuk memproses urutan berikutnya [12]. Berbeda dengan *feedforward neural network* yang memproses setiap data secara independen, RNN memiliki mekanisme umpan balik (*recurrent connection*) yang memungkinkan informasi dari langkah waktu sebelumnya diteruskan ke langkah waktu berikutnya melalui *hidden state*. Dengan mekanisme tersebut, RNN mampu mempelajari hubungan antar token dalam suatu urutan sehingga konteks dari kata-kata sebelumnya tetap dipertimbangkan ketika memproses kata berikutnya.

Pada setiap langkah waktu (t), RNN menerima dua masukan, yaitu data pada waktu saat ini (x_t) dan *hidden state* dari langkah waktu sebelumnya (h_{t-1}). Kedua informasi tersebut kemudian diproses untuk menghasilkan *hidden state* baru (h_t), yang selanjutnya diteruskan ke langkah waktu berikutnya sebagai memori sementara. Secara matematis, proses pembaruan *hidden state* pada RNN dapat dituliskan sebagai berikut:

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b) \tag{2.1}$$

Di mana h_t merupakan *hidden state* pada waktu ke-t, h_{t-1} adalah *hidden state* pada waktu sebelumnya, x_t merupakan data masukan pada waktu ke-t, W_h dan W_x adalah matriks bobot yang dipelajari selama proses pelatihan, sedangkan b merupakan vektor bias. Fungsi aktivasi tanh digunakan untuk menghasilkan representasi baru yang akan diteruskan ke langkah waktu berikutnya maupun digunakan sebagai dasar dalam menghasilkan keluaran model.

Meskipun mampu memodelkan hubungan antar data yang berurutan, RNN konvensional memiliki keterbatasan dalam mempelajari dependensi jangka panjang akibat permasalahan *vanishing gradient* selama proses *backpropagation*. *Vanishing gradient* merupakan kondisi ketika nilai gradien yang digunakan untuk memperbarui bobot jaringan menjadi semakin kecil seiring semakin panjangnya urutan data yang diproses. Akibatnya, pembaruan bobot pada lapisan-lapisan awal menjadi sangat kecil bahkan mendekati nol, sehingga model kesulitan mempertahankan informasi dari langkah waktu yang jauh sebelumnya. Kondisi tersebut menyebabkan RNN cenderung hanya mampu mengingat informasi jangka pendek, sedangkan informasi penting yang muncul pada awal urutan secara

bertahap hilang selama proses pelatihan.

Untuk mengatasi permasalahan tersebut, dikembangkan arsitektur *Long Short-Term Memory* (LSTM) yang memperkenalkan mekanisme *memory cell* dan tiga gerbang (*forget gate*, *input gate*, dan *output gate*) untuk mengatur aliran informasi secara selektif. Mekanisme tersebut memungkinkan informasi yang masih relevan dipertahankan dalam *cell state*, sedangkan informasi yang tidak diperlukan dapat dibuang. Dengan demikian, LSTM mampu mempertahankan dependensi jangka panjang secara lebih efektif dan mengurangi dampak permasalahan *vanishing gradient* dibandingkan RNN konvensional [12].

LSTM mengatasi keterbatasan RNN melalui mekanisme tiga gerbang (*gate*) yang mengatur aliran informasi secara selektif [16]. Ketiga gerbang tersebut bekerja secara sinergis sebagai berikut:

1. ***Forget Gate*** (f_t): Menentukan informasi dari status sel sebelumnya (C_{t-1}) yang perlu dilupakan atau dipertahankan menggunakan fungsi Sigmoid (σ):

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

2. ***Input Gate*** (i_t): Menentukan informasi baru dari input saat ini (x_t) yang perlu disimpan ke dalam status sel:

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

3. ***Output Gate*** (o_t): Menentukan output tersembunyi (*hidden state*) h_t berdasarkan status sel terkini yang telah diperbarui:

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

Persamaan 2.2, Persamaan 2.3, dan Persamaan 2.4 menunjukkan mekanisme kerja tiga gerbang pada LSTM.

Di mana σ adalah fungsi aktivasi Sigmoid, W adalah matriks bobot masing-masing gerbang, h_{t-1} adalah *hidden state* dari langkah waktu sebelumnya, x_t adalah input pada langkah waktu saat ini, dan b adalah vektor bias. Dalam arsitektur hybrid IndoBERT-LSTM, lapisan LSTM menerima output representasi kontekstual dari IndoBERT dan memproses dependensi sekuensial antar token untuk menghasilkan representasi yang lebih kaya bagi klasifikasi sentimen.

2.4.2 Bidirectional LSTM (BiLSTM)

LSTM konvensional sebagaimana dijabarkan pada Subbab 2.4.1 hanya memproses urutan token searah, dari token pertama hingga token terakhir (*forward pass*), sehingga representasi suatu token hanya dipengaruhi oleh token-token sebelumnya dalam urutan. Keterbatasan ini menjadi kurang ideal untuk tugas pemahaman bahasa, karena makna suatu kata seringkali juga dipengaruhi oleh kata-kata yang muncul setelahnya dalam kalimat [16].

Bidirectional LSTM (BiLSTM) mengatasi keterbatasan ini dengan menjalankan dua lapisan LSTM secara paralel pada arah yang berlawanan: satu lapisan memproses urutan token dari awal ke akhir (*forward*, menghasilkan *hidden state* $\overrightarrow{h_t}$), dan satu lapisan lainnya memproses urutan token yang sama dari akhir ke awal (*backward*, menghasilkan *hidden state* $\overleftarrow{h_t}$). *Hidden state* akhir pada setiap langkah waktu t diperoleh dengan menggabungkan (*concatenate*) kedua arah tersebut:

$$h_t = \left[\overrightarrow{h_t} ; \overleftarrow{h_t}\right] \tag{2.5}$$

Persamaan 2.5 menunjukkan bahwa representasi keluaran BiLSTM diperoleh dengan menggabungkan *hidden state* dari arah *forward* dan *backward* pada setiap langkah waktu.

Dengan demikian, representasi setiap token pada BiLSTM memuat informasi konteks dari kedua arah kalimat secara simultan, sehingga dimensi keluaran BiLSTM menjadi dua kali lipat dimensi *hidden state* satu arah. Ahmadian dkk. [9] membuktikan bahwa integrasi BiLSTM dengan representasi kontekstual IndoBERT—menggabungkan empat lapisan terakhir IndoBERT dengan BiLSTM—mampu mencapai akurasi 93% pada analisis sentimen berbahasa Indonesia, melampaui kinerja IndoBERT yang berdiri sendiri. Karakteristik dua arah inilah yang menjadi alasan utama pemilihan BiLSTM, bukan LSTM satu arah, sebagai lapisan sekuensial tambahan pada arsitektur *hybrid* dalam penelitian ini.

2.4.3 IndoBERT (Arsitektur Berbasis Transformer)

IndoBERT adalah model bahasa berbasis arsitektur BERT (*Bidirectional Encoder Representations from Transformers*) yang dikembangkan dan dilatih secara khusus pada korpus Bahasa Indonesia yang sangat besar [17]. Marpaung dan Tarigan [3] memilih IndoBERT sebagai model utama dalam penelitiannya

karena kemampuan model ini memahami konteks kata secara dua arah sekaligus, yang sangat relevan untuk menganalisis nuansa bahasa informal dalam *tweet* berbahasa Indonesia tentang Timnas. Keunggulan ini membuat IndoBERT mampu membedakan sentimen positif dan negatif dengan lebih akurat dibandingkan model multilingual generik.

Arsitektur Transformer bekerja dengan memproses seluruh token dalam suatu kalimat secara paralel menggunakan mekanisme *self-attention*, sehingga setiap token dapat mempelajari hubungan dengan seluruh token lainnya secara bersamaan. Proses ini diawali dengan mengubah setiap token menjadi representasi vektor melalui *word embedding*, kemudian ditambahkan informasi posisi menggunakan *positional encoding* agar model tetap dapat mengenali urutan setiap token dalam kalimat. Representasi tersebut selanjutnya diproses oleh lapisan *Multi-Head Self-Attention* untuk menangkap hubungan kontekstual antar token, kemudian diteruskan ke lapisan *Feed Forward Neural Network* (FFN) sehingga dihasilkan representasi yang lebih kaya. Berbeda dengan RNN maupun LSTM yang memproses token secara berurutan, Transformer mampu memproses seluruh token secara simultan sehingga lebih efisien dalam pelatihan serta lebih baik dalam menangkap hubungan antar token yang berjauhan dalam suatu kalimat [14].

Word embedding merupakan teknik representasi kata ke dalam bentuk vektor numerik berdimensi tetap sehingga dapat diproses oleh model *deep learning*. Representasi ini dirancang agar kata-kata yang memiliki makna atau konteks yang serupa memiliki posisi yang berdekatan di dalam ruang vektor. Berbeda dengan representasi tradisional seperti *one-hot encoding* yang hanya menunjukkan keberadaan suatu kata tanpa menangkap hubungan semantik antar kata, *word embedding* mampu merepresentasikan kemiripan makna berdasarkan pola kemunculan kata dalam suatu korpus. Pada arsitektur Transformer, setiap token terlebih dahulu diubah menjadi vektor melalui proses *word embedding* sebelum ditambahkan informasi posisi (*positional encoding*) dan diproses lebih lanjut oleh mekanisme *Multi-Head Self-Attention*. Dengan demikian, setiap token memiliki representasi numerik yang mengandung informasi semantik sekaligus menjadi masukan utama bagi seluruh lapisan Transformer.

Inti dari arsitektur IndoBERT adalah mekanisme *Multi-Head Self-Attention* yang memungkinkan model menimbang pentingnya setiap token dalam kalimat saat merepresentasikan satu token tertentu berdasarkan seluruh token lainnya secara simultan [17]. Pada mekanisme ini, setiap token diubah menjadi tiga buah vektor representasi, yaitu *Query* (Q), *Key* (K), dan *Value* (V). Skor atensi dihitung

berdasarkan tingkat kemiripan antara vektor *Query* suatu token dengan vektor *Key* token lainnya, kemudian dinormalisasi menggunakan fungsi *softmax*. Proses tersebut dirumuskan secara matematis sebagai berikut:

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

Persamaan 2.6 menunjukkan mekanisme perhitungan *self-attention* yang menjadi inti arsitektur Transformer pada model IndoBERT.

di mana Q merupakan matriks *Query*, K merupakan matriks *Key*, V merupakan matriks *Value*, K^T adalah transpos matriks *Key*, dan d_k merupakan dimensi vektor *Key* yang digunakan sebagai faktor penskalaan untuk menjaga stabilitas gradien selama proses pelatihan.

Kekuatan utama IndoBERT terletak pada proses pra-pelatihan (*pre-training*) dua tahap. Tahap pertama, *Masked Language Model* (MLM), melatih model untuk memprediksi token yang disembunyikan secara acak berdasarkan konteks token-token di sekitarnya dari kedua arah, sehingga menghasilkan pemahaman kontekstual yang mendalam. Tahap kedua, *Next Sentence Prediction* (NSP), melatih model untuk menentukan apakah dua kalimat berurutan secara logis [17]. Setelah pra-pelatihan, IndoBERT diadaptasi untuk tugas analisis sentimen melalui proses *fine-tuning* pada dataset berlabel yang lebih kecil, sehingga mampu menghasilkan representasi kontekstual yang efektif untuk berbagai tugas *Natural Language Processing*, termasuk klasifikasi sentimen.

2.4.4 Arsitektur Hybrid IndoBERT-BiLSTM

Arsitektur hybrid IndoBERT-BiLSTM menggabungkan kekuatan dua komponen yang saling melengkapi: IndoBERT sebagai *contextual feature extractor* dan BiLSTM sebagai pemodelan dependensi sekuensial dua arah. Putra dkk. [8] merancang arsitektur serupa dengan cara mengintegrasikan output representasi kontekstual dari lapisan IndoBERT sebagai input langsung ke lapisan LSTM, sehingga lapisan sekuensial dapat memproses urutan representasi kontekstual yang sudah kaya makna tersebut. Ahmadian dkk. [9] membuktikan bahwa integrasi semacam ini—menggabungkan empat lapisan terakhir IndoBERT dengan BiLSTM—mampu mencapai akurasi 93% pada analisis sentimen berbahasa Indonesia, melampaui kinerja IndoBERT yang berdiri sendiri.

Keunggulan arsitektur hybrid ini dibandingkan penggunaan model tunggal

terletak pada kemampuannya memanfaatkan dua jenis pemahaman sekaligus: pemahaman kontekstual dua arah dari IndoBERT yang mampu menangkap makna kata berdasarkan seluruh konteks kalimat, dan pemahaman dependensi sekuensial dua arah dari BiLSTM yang mampu menangkap pola urutan token secara jangka panjang dari kedua arah kalimat [8]. Kombinasi keduanya sangat relevan untuk analisis sentimen tweet berbahasa Indonesia tentang Timnas Indonesia, di mana ekspresi sentimen seringkali memerlukan pemahaman konteks yang mendalam sekaligus pemahaman urutan kata yang spesifik.

Setelah representasi token diproses oleh lapisan BiLSTM, diperlukan satu langkah tambahan untuk meringkas representasi seluruh token tersebut menjadi satu vektor representasi kalimat, yaitu melalui mekanisme *pooling*. Salah satu pendekatan *pooling* yang umum digunakan adalah *mean pooling*, yaitu menghitung rata-rata seluruh *hidden state* token sebagai representasi kalimat. Pada data yang telah melalui proses *padding* (penambahan token kosong agar seluruh kalimat dalam satu *batch* memiliki panjang yang sama), *mean pooling* perlu memperhitungkan *attention mask*—penanda token mana yang merupakan data asli (bernilai 1) dan mana yang merupakan token *padding* (bernilai 0)—agar token *padding* tidak ikut memengaruhi nilai rata-rata yang dihasilkan. Pendekatan inilah yang disebut *masked mean pooling*, dan digunakan pada arsitektur model dalam penelitian ini sebagaimana akan dijabarkan lebih lanjut pada Subbab 3.8.

2.5 Penanganan Ketidakseimbangan Kelas

Ketidakseimbangan kelas (*class imbalance*) terjadi ketika jumlah data pada satu atau beberapa kelas jauh lebih banyak dibandingkan kelas lainnya dalam suatu tugas klasifikasi. Pada tugas analisis sentimen, kondisi ini lazim terjadi karena ekspresi netral atau informatif pada media sosial umumnya jauh lebih banyak dibandingkan ekspresi sentimen yang kuat (positif maupun negatif) [18]. Apabila tidak ditangani, model *machine learning* akan cenderung bias terhadap kelas mayoritas, menghasilkan akurasi keseluruhan yang tampak tinggi namun dengan *recall* kelas minoritas yang sangat rendah, karena model secara implisit ”belajar” bahwa memprediksi kelas mayoritas secara konsisten sudah cukup untuk meminimalkan *loss* rata-rata.

Salah satu teknik untuk menangani ketidakseimbangan kelas tanpa mengubah distribusi data asli adalah *class weighting* pada fungsi *loss*. Berbeda dengan teknik *resampling* seperti SMOTE atau *oversampling* yang menambah

maupun mengurangi jumlah sampel data secara fisik, *class weighting* bekerja dengan memberikan bobot lebih besar pada kontribusi *loss* dari kelas minoritas, sehingga kesalahan klasifikasi pada kelas tersebut ”dihukum” lebih berat dibandingkan kesalahan pada kelas mayoritas, tanpa perlu menduplikasi maupun mensintesis data baru. Bobot kelas dapat dihitung secara seimbang (*balanced*) menggunakan rumus:

$$w_c = \frac{n}{k \times n_c} \tag{2.7}$$

Persamaan 2.7 menunjukkan bahwa bobot setiap kelas dihitung berdasarkan jumlah total data, jumlah kelas, dan jumlah data pada masing-masing kelas.

Di mana w_c adalah bobot untuk kelas c, n adalah jumlah total data, k adalah jumlah kelas, dan n_c adalah jumlah data pada kelas c. Rumus ini menghasilkan bobot yang berbanding terbalik dengan frekuensi kemunculan suatu kelas: semakin sedikit jumlah data suatu kelas, semakin besar bobotnya. Bobot yang diperoleh kemudian dikalikan dengan nilai *loss* setiap data pada fungsi *loss* seperti *Cross-Entropy Loss*, sehingga total *loss* dari kelas minoritas memberikan kontribusi yang setara dengan kelas mayoritas terhadap pembaruan bobot model selama pelatihan.

Pendekatan *class weighting* ini dipilih dalam penelitian ini karena tetap mempertahankan distribusi data asli, berbeda dengan teknik *oversampling* berbasis SMOTE yang menambahkan sampel sintetis dan dapat menyebabkan distribusi data tidak lagi mencerminkan kondisi nyata di lapangan, sebagaimana diterapkan pada penelitian Guntara dkk. [19].

2.6 Optimasi Hiperparameter Otomatis

Kinerja model *deep learning* sangat dipengaruhi oleh pemilihan hiperparameter, seperti *learning rate*, ukuran *batch*, dan jumlah lapisan tersembunyi. Pemilihan hiperparameter secara manual melalui *trial-and-error* cenderung tidak efisien dan rentan menghasilkan konfigurasi yang suboptimal, terutama pada arsitektur *hybrid* yang menggabungkan komponen pra-terlatih (IndoBERT) dengan komponen yang diinisialisasi secara acak (BiLSTM), karena kedua komponen tersebut idealnya memerlukan *learning rate* yang berbeda agar bobot pra-pelatihan tidak mengalami *catastrophic forgetting* sementara komponen acak tetap dapat berkonvergensi dengan cepat.

Optuna adalah *framework* optimasi hiperparameter otomatis yang bersifat *define-by-run*, yang memungkinkan ruang pencarian hiperparameter didefinisikan

secara dinamis di dalam kode program itu sendiri. Optuna mendukung berbagai strategi pencarian (*sampler*), salah satunya adalah *Tree-structured Parzen Estimator* (TPE), yaitu algoritma optimasi Bayesian yang membangun model probabilistik dari hasil percobaan (*trial*) sebelumnya untuk memprediksi kombinasi hiperparameter mana yang kemungkinan besar akan menghasilkan kinerja terbaik pada percobaan selanjutnya, sehingga pencarian menjadi lebih terarah dibandingkan pencarian acak (*random search*) maupun pencarian kisi (*grid search*).

Untuk mempercepat proses pencarian, Optuna juga mendukung mekanisme *pruning*, yaitu penghentian dini suatu *trial* yang menunjukkan kinerja jauh lebih buruk dibandingkan *trial* lain pada tahap pelatihan yang sama, sehingga sumber daya komputasi dapat dialokasikan untuk *trial* yang lebih menjanjikan. Salah satu strategi *pruning* yang umum digunakan adalah *Median Pruner*, yang menghentikan suatu *trial* apabila nilai metrik evaluasinya berada di bawah median nilai metrik dari *trial-trial* sebelumnya pada langkah evaluasi yang sama.

2.7 Metode Evaluasi

Untuk mengukur kinerja model hybrid IndoBERT-LSTM secara kuantitatif, digunakan serangkaian metrik evaluasi standar yang berasal dari *Confusion Matrix*. *Confusion Matrix* adalah tabel yang memvisualisasikan performa algoritma klasifikasi dengan membandingkan label kelas aktual dengan label kelas yang diprediksi oleh model [18]. Abidin dan Pamungkas [5] menerapkan metrik-metrik ini secara komprehensif untuk membandingkan kinerja IndoBERT, IndoRoBERTa, dan DistilBERT Multilingual pada analisis sentimen Timnas Indonesia, sehingga menghasilkan perbandingan yang objektif dan terukur antar model. Untuk tugas klasifikasi tiga kelas sentimen dalam penelitian ini, digunakan *Confusion Matrix* berukuran 3×3.

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

1. **Akurasi (*Accuracy*)**

   Mengukur persentase keseluruhan prediksi yang benar terhadap total seluruh data uji. Marpaung dan Tarigan [3] menggunakan akurasi sebagai metrik utama dan melengkapinya dengan macro F1-*score* untuk mendapatkan

gambaran kinerja yang lebih komprehensif pada kondisi ketidakseimbangan kelas.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.8}$$

Persamaan 2.8 menunjukkan perhitungan akurasi sebagai proporsi prediksi yang benar terhadap seluruh data uji.

2. **Presisi (*Precision*)**

   Mengukur proporsi prediksi positif yang benar-benar positif dari seluruh prediksi yang diklasifikasikan sebagai positif oleh model. Presisi yang tinggi menunjukkan rendahnya tingkat *false positive* dalam prediksi model.

$$Precision = \frac{TP}{TP + FP} \tag{2.9}$$

   Persamaan 3.1 menunjukkan perhitungan presisi sebagai proporsi prediksi positif yang benar terhadap seluruh prediksi positif.

3. ***Recall* (*Sensitivity*)**

   Mengukur proporsi data yang benar-benar positif yang berhasil diidentifikasi secara tepat oleh model. Nanda dkk. [4] melaporkan nilai *recall* sebesar 74,92% pada model LSTM terbaiknya, menunjukkan kemampuan model mendeteksi sebagian besar *tweet* dengan sentimen yang relevan.

$$Recall = \frac{TP}{TP + FN} \tag{2.10}$$

   Persamaan 3.2 menunjukkan perhitungan *recall* sebagai proporsi data positif yang berhasil dikenali oleh model.

4. ***F1-Score***

   Merupakan rata-rata harmonik antara nilai Presisi dan *Recall*, sangat relevan untuk kondisi ketidakseimbangan kelas. Putra dkk. [8] menggunakan F1-*score* sebagai metrik utama dalam membandingkan kinerja IndoBERT-LSTM dengan model lainnya, dan berhasil mencapai F1-*score* 98,90% sebagai hasil terbaik.

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.11}$$

Persamaan 2.11 menunjukkan bahwa nilai *F1-Score* diperoleh dari rata-rata harmonik antara *Precision* dan *Recall*.

Dalam penelitian ini, keempat metrik tersebut dihitung secara terpisah untuk setiap kelas sentimen (positif, negatif, netral) maupun secara keseluruhan dalam bentuk *macro average* dan *weighted average*. Nugraha dan Zakiyah [13] menerapkan pendekatan serupa dalam menganalisis distribusi sentimen terhadap Timnas Indonesia, sehingga menghasilkan pemahaman yang lebih granular mengenai kemampuan model dalam mengklasifikasikan masing-masing kelas sentimen secara individual.

2.8 Identifikasi Research Gap

Berdasarkan tinjauan terhadap penelitian-penelitian terdahulu yang telah dipaparkan pada bab sebelumnya, terdapat dua penelitian yang paling relevan dan menjadi acuan utama dalam penelitian ini, yaitu Putra dkk. [8] dan Guntara dkk. [19].

Putra dkk. [8] telah membuktikan bahwa kombinasi *hybrid* IndoBERT-LSTM mampu menghasilkan *F1-score* sebesar 98,90% pada klasifikasi *multiclass* tweet berbahasa Indonesia, mengungguli penggunaan IndoBERT maupun Word2Vec-LSTM secara terpisah. Arsitektur *hybrid* ini memanfaatkan kemampuan IndoBERT dalam menghasilkan representasi kontekstual yang kaya, yang kemudian diproses lebih lanjut oleh lapisan LSTM untuk menangkap pola dependensi sekuensial antar token. Meskipun demikian, penelitian tersebut memiliki keterbatasan yang signifikan — dataset yang digunakan bersifat umum dan tidak domain-spesifik, distribusi kelasnya relatif seimbang sehingga tidak menghadapi tantangan *class imbalance*, serta *task* yang diselesaikan adalah klasifikasi topik bukan analisis sentimen tiga kelas. Putra dkk. [8] sendiri merekomendasikan eksplorasi lebih lanjut pada berbagai domain dan dataset berbahasa Indonesia sebagai arah penelitian selanjutnya.

Di sisi lain, Guntara dkk. [19] mengembangkan model *hybrid* NusaBERT-BiLSTM dan NusaBERT-BiGRU untuk analisis sentimen suporter sepak bola. Pada penelitian tersebut, NusaBERT berfungsi sebagai *pre-trained Transformer* yang menghasilkan representasi kontekstual (*contextual embedding*) dari setiap teks

hasil tokenisasi sebelum diproses lebih lanjut oleh lapisan BiLSTM atau BiGRU untuk melakukan klasifikasi sentimen. Untuk mengatasi ketidakseimbangan kelas, penelitian tersebut menerapkan teknik SMOTE pada data latih, kemudian membandingkan performa kedua arsitektur tersebut. Hasil penelitian menunjukkan bahwa model NusaBERT-BiLSTM memperoleh performa terbaik dengan akurasi sebesar 70,83%. Nilai akurasi tersebut menunjukkan bahwa klasifikasi sentimen pada domain suporter sepak bola masih menjadi tantangan yang cukup kompleks. Namun, penelitian Guntara dkk. berfokus pada perbandingan performa antara NusaBERT-BiLSTM dan NusaBERT-BiGRU, sehingga tidak membahas secara khusus faktor-faktor yang menyebabkan nilai akurasi tersebut. Selain itu, penelitian tersebut hanya melakukan klasifikasi dua kelas sentimen (positif dan negatif), menggunakan dataset sebanyak 1.039 *tweet*, serta belum mengeksplorasi penggunaan IndoBERT maupun klasifikasi tiga kelas sentimen.

Nilai akurasi yang relatif lebih rendah dibandingkan penelitian Putra dkk. [8] dipengaruhi oleh beberapa faktor. Pertama, NusaBERT dikembangkan untuk mendukung berbagai bahasa dan dialek daerah di Indonesia sehingga representasinya tidak secara khusus dioptimalkan untuk teks berbahasa Indonesia standar yang banyak digunakan pada platform X (Twitter). Kedua, penelitian tersebut hanya melakukan klasifikasi dua kelas sentimen sehingga belum mampu merepresentasikan kompleksitas opini publik yang mengandung sentimen netral. Ketiga, penanganan *class imbalance* dilakukan menggunakan SMOTE yang menghasilkan sampel sintetis, sehingga distribusi data yang digunakan pada proses pelatihan tidak sepenuhnya mencerminkan kondisi data asli. Penelitian ini hadir untuk mengisi celah tersebut, sebagaimana dirangkum pada Tabel 2.1.

Tabel 2.1. Identifikasi *research gap*

No	Peneliti	Metode	Domain	Kelas Sentimen	Penanganan *Class Imbalance*	Hasil	Keterbatasan
1	Putra dkk. (2022)	IndoBERT-LSTM	Tweet umum	*Multiclass* topik	Tidak ada	F1 98,90%	Dataset umum, bukan sentimen, tidak ada *class imbalance*
2	Guntara dkk. (2026)	NusaBERT-BiLSTM	Suporter sepak bola	2 kelas	SMOTE	Akurasi 70,83%	NusaBERT, hanya 2 kelas, SMOTE menambah sampel buatan