

BAB II

KAJIAN LITERATUR

2.1 Work From Home

Work From Home (WFH) merupakan suatu sistem kerja yang memungkinkan karyawan untuk melaksanakan tugas dan tanggung jawabnya dari rumah atau lokasi di luar kantor dengan memanfaatkan teknologi informasi dan komunikasi. Konsep ini menjadi bagian dari transformasi digital dalam dunia kerja yang memberikan fleksibilitas dalam hal waktu dan tempat kerja. Penerapan WFH memungkinkan pekerja untuk tetap terhubung dengan organisasi melalui berbagai platform digital seperti aplikasi komunikasi, sistem manajemen proyek, serta layanan berbasis cloud yang mendukung kolaborasi secara daring [10].

Di Indonesia, penerapan *Work From Home* (WFH) mulai dilakukan secara luas sejak tahun 2020 sebagai respons terhadap pandemi COVID-19. Kebijakan ini diterapkan sebagai upaya untuk mengurangi mobilitas masyarakat dan menekan penyebaran virus, sehingga aktivitas kerja yang sebelumnya dilakukan secara tatap muka dialihkan menjadi berbasis daring. Perubahan ini memberikan dampak yang signifikan terhadap pola kerja di berbagai sektor, baik pemerintahan maupun swasta, serta mendorong percepatan adopsi teknologi digital dalam aktivitas pekerjaan sehari-hari [11].

Secara umum, penerapan WFH memberikan berbagai manfaat, seperti meningkatkan fleksibilitas kerja, mengurangi waktu perjalanan (*commuting time*), serta memberikan kesempatan bagi pekerja untuk mencapai keseimbangan antara kehidupan pribadi dan pekerjaan (*work-life balance*). Selain itu, bagi organisasi, WFH dapat membantu menekan biaya operasional seperti penggunaan listrik, transportasi, dan fasilitas kantor. Namun demikian, sistem ini juga memiliki sejumlah tantangan, seperti keterbatasan komunikasi antar tim, berkurangnya interaksi sosial, serta potensi penurunan produktivitas akibat kurangnya pengawasan langsung [12].

Seiring berjalannya waktu, konsep WFH tidak lagi hanya digunakan dalam kondisi darurat seperti pandemi, tetapi mulai dipertimbangkan sebagai strategi kerja jangka panjang. Pada tahun 2026, kebijakan WFH kembali diwacanakan sebagai bagian dari upaya efisiensi energi nasional. Hal ini menunjukkan bahwa WFH telah berkembang menjadi bagian dari kebijakan strategis yang memiliki dampak luas terhadap berbagai aspek kehidupan, termasuk ekonomi, sosial, dan

lingkungan. Oleh karena itu, penting untuk memahami bagaimana persepsi masyarakat terhadap kebijakan ini, terutama melalui analisis sentimen terhadap komentar di media sosial [13].

2.2 Analisis Sentimen

Analisis sentimen merupakan representasi dari opini, pandangan, atau emosi yang terkandung dalam suatu data teks yang biasanya didefinisikan ke dalam beberapa kategori, seperti positif, negatif, dan netral [14]. Analisis sentimen berfungsi sebagai media dalam menyampaikan serta mengidentifikasi suatu pesan, makna, maupun tanggapan masyarakat terhadap sebuah topik, kebijakan, produk, atau peristiwa tertentu [15]. Untuk menciptakan klasifikasi dari suatu opini, analisis sentimen mengombinasikan berbagai tahapan pemrosesan teks seperti *cleaning*, *tokenizing*, *stemming*, dan *labeling* yang kemudian diproses menggunakan metode klasifikasi [16].

Akibat dari kemajuan teknologi, konsep analisis sentimen telah melangkah ke era baru dengan munculnya metode berbasis *machine learning* dan *deep learning* sehingga mampu melakukan klasifikasi data teks secara lebih detail dan akurat. Oleh karena itu, model kecerdasan buatan seperti BERT dan IndoBERT dapat menghasilkan klasifikasi sentimen yang lebih sulit dibedakan tingkat akurasinya dibandingkan metode tradisional karena mampu memahami konteks bahasa secara lebih mendalam [17].

Data teks asli merupakan data yang dirancang untuk menampilkan opini, persepsi, serta tanggapan pengguna terhadap suatu topik tertentu, seperti komentar pada media sosial, ulasan aplikasi, maupun forum diskusi. Dalam proses pengambilan data teks asli, media sosial berperan sebagai sumber utama data yang merekam opini masyarakat secara langsung dan real time. Proses data yang dihasilkan tampak sesuai dengan pandangan asli pengguna, sehingga mampu menangkap sentimen secara natural dan realistis. Data yang dihasilkan mempertimbangkan struktur bahasa, konteks kalimat, dan penggunaan kata yang relevan agar sesuai dengan maksud penulis aslinya, sehingga memberikan gambaran persepsi publik secara nyata.

Berbeda dengan model BiLSTM yang berfokus pada pemrosesan urutan kata secara sekuensial meskipun dari dua arah (*forward* dan *backward*), model berbasis transformer seperti IndoBERT mampu memahami konteks bahasa secara menyeluruh melalui mekanisme *bidirectional context* yang lebih kompleks, sehingga menghasilkan klasifikasi sentimen yang lebih akurat. Penelitian oleh Purwanto (2026) menunjukkan bahwa IndoBERT memperoleh akurasi

sebesar 87,50%, lebih tinggi dibandingkan BiLSTM yang hanya mencapai 77,93% dalam klasifikasi sentimen teks berbahasa Indonesia [17].

Dalam proses analisis sentimen, terdapat beberapa tahapan penting yang harus dilakukan agar data teks dapat diproses dengan baik oleh model. Tahapan-tahapan tersebut dapat diuraikan sebagai berikut:

1. Cleaning

Tahapan *cleaning* atau pembersihan data merupakan langkah awal yang krusial dalam pipeline pemrosesan teks sebelum data dimasukkan ke dalam model IndoBERT. Pada tahap ini, teks mentah yang berasal dari komentar media sosial mengalami serangkaian proses pembersihan untuk menghilangkan elemen-elemen yang tidak relevan dan berpotensi mengganggu proses pembelajaran model. Proses pembersihan tersebut meliputi penghapusan *URL* dan tautan eksternal, menghilangkan *mention* pengguna (simbol @), *hashtag* (simbol #), serta karakter-karakter khusus seperti emoji dan simbol yang tidak bermakna secara linguistik. Selain itu, dilakukan pula normalisasi teks yang mencakup konversi seluruh karakter menjadi huruf kecil (*case folding*), perbaikan kata-kata tidak baku atau singkatan yang umum digunakan dalam percakapan informal di media sosial, serta penghapusan karakter berulang yang tidak memiliki nilai semantik. Tahapan ini menjadi sangat penting dalam konteks data media sosial berbahasa Indonesia karena karakteristik teks yang sangat heterogen, penuh variasi bahasa tidak formal, campuran bahasa daerah, dan gaya penulisan yang tidak konsisten. Teks yang telah melalui proses *cleaning* kemudian diteruskan ke tahap tokenisasi menggunakan *tokenizer* IndoBERT, yang akan memecah teks menjadi unit-unit token sesuai dengan kosakata yang dipelajari selama proses *pre-training* model, sehingga representasi teks yang dihasilkan lebih bersih, terstruktur, dan siap untuk diproses lebih lanjut dalam tahap *fine-tuning* IndoBERT.

2. Tokenizing

Setelah data melalui proses *cleaning*, tahapan selanjutnya adalah tokenisasi, yaitu proses mengubah teks menjadi representasi numerik yang dapat dipahami oleh model IndoBERT. Berbeda dengan tokenisasi konvensional yang memecah teks berdasarkan spasi atau tanda baca, IndoBERT menggunakan metode *WordPiece tokenization* yang memecah kata menjadi sub-kata (*subword*) berdasarkan kosakata yang telah dipelajari selama proses *pre-training*. Pendekatan ini memiliki keunggulan dalam menangani kata-kata yang jarang muncul (*out-of-vocabulary*) atau kata-kata tidak baku yang sering dijumpai pada teks media sosial berbahasa Indonesia, karena kata

tersebut akan dipecah menjadi unit-unit sub-kata yang lebih kecil namun tetap bermakna [18]. Dalam implementasinya, setiap teks yang akan diproses ditambahkan token khusus berupa [CLS] di awal kalimat sebagai representasi keseluruhan kalimat untuk keperluan klasifikasi, serta token [SEP] di akhir kalimat sebagai penanda batas kalimat. Selain itu, setiap teks juga dibatasi panjang maksimumnya (*max length*) sesuai dengan kapasitas model, dan teks yang lebih pendek dari panjang maksimum akan ditambahkan token *padding* [PAD] agar seluruh input memiliki dimensi yang seragam. Hasil akhir dari proses tokenisasi ini berupa tiga komponen utama, yaitu *input IDs* sebagai representasi numerik dari setiap token, *attention mask* sebagai penanda token yang bermakna dan token *padding*, serta *token type IDs* yang digunakan untuk membedakan segmen kalimat dalam kasus input berpasangan.

3. Stemming

Stemming merupakan salah satu tahapan dalam proses *preprocessing* teks yang bertujuan untuk mengubah kata berimbuhan menjadi bentuk dasarnya (*root word*) guna mengurangi variasi morfologi kata yang tidak diperlukan dalam proses analisis. Dalam konteks bahasa Indonesia, proses *stemming* menjadi cukup kompleks karena bahasa Indonesia memiliki sistem afiksasi yang kaya, mencakup awalan (*prefiks*) seperti "me-", "di-", "ke-", "se-", sisipan (*infiks*) seperti "-el-", "-er-", akhiran (*sufiks*) seperti "-kan", "-an", "-i", serta kombinasi awalan dan akhiran (*konfiks*) seperti "pe-an", "ke-an", dan "me-kan". Untuk menangani kompleksitas morfologi bahasa Indonesia tersebut, algoritma *stemming* yang paling umum digunakan adalah algoritma Nazief-Adriani dan PySastrawi, di mana PySastrawi merupakan implementasi berbasis Python yang banyak digunakan dalam penelitian *Natural Language Processing* (NLP) berbahasa Indonesia karena kemampuannya menangani berbagai bentuk imbuhan secara lebih akurat. Sebagai contoh, kata "bekerja", "pekerjaan", "dikerjakan", dan "mengerjakan" setelah melalui proses *stemming* akan diubah menjadi bentuk dasar yang sama, yaitu "kerja". Meskipun demikian, perlu dicatat bahwa dalam konteks penggunaan IndoBERT, tahapan *stemming* tidak selalu wajib diterapkan. Hal ini dikarenakan IndoBERT telah dilatih menggunakan korpus bahasa Indonesia yang besar dan beragam, sehingga model ini secara inheren sudah mampu memahami berbagai bentuk variasi morfologi kata beserta konteks penggunaannya tanpa harus terlebih dahulu mengubah kata ke bentuk dasarnya. Bahkan dalam sejumlah penelitian, penerapan *stemming* sebelum proses tokenisasi IndoBERT justru dapat mengurangi informasi kontekstual yang terkandung dalam teks asli, sehingga berpotensi menurunkan performa model dalam memahami makna kalimat secara

keseluruhan. Oleh karena itu, keputusan untuk menerapkan atau melewati tahapan *stemming* dalam pipeline IndoBERT perlu disesuaikan dengan karakteristik data yang digunakan serta hasil eksperimen empiris yang dilakukan selama proses pengembangan model.

4. Labeling

Tahapan pelabelan merupakan proses pemberian kelas sentimen pada setiap data komentar yang telah melalui proses *cleaning* dan tokenisasi. Dalam penelitian analisis sentimen, setiap komentar diklasifikasikan ke dalam tiga kategori label, yaitu positif, negatif, dan netral, berdasarkan kandungan opini yang terdapat dalam teks tersebut. Proses pelabelan dapat dilakukan melalui dua pendekatan utama, yakni pelabelan manual (*manual labeling*) dan pelabelan otomatis (*automatic labeling*). Pada pelabelan manual, setiap data diberi label oleh anotator manusia yang membaca dan memahami isi komentar secara langsung, sehingga hasil labelnya cenderung lebih akurat namun membutuhkan waktu dan sumber daya yang tidak sedikit, terutama ketika volume data sangat besar. Sementara itu, pelabelan otomatis memanfaatkan pendekatan berbasis leksikon seperti InSet Lexicon yang merupakan kamus sentimen khusus bahasa Indonesia, di mana setiap kata dalam teks dicocokkan dengan nilai polaritas positif atau negatif yang tersimpan dalam leksikon tersebut, kemudian dijumlahkan untuk menentukan label keseluruhan kalimat. Masing-masing pendekatan memiliki kelebihan dan kekurangannya tersendiri, sehingga sejumlah penelitian memilih pendekatan gabungan (*hybrid labeling*) yang mengombinasikan kedua metode tersebut guna menghasilkan label yang lebih representatif dan konsisten terhadap karakteristik data yang digunakan.

Akibat dari kemajuan teknologi, konsep analisis sentimen telah melangkah ke era baru dengan munculnya metode berbasis *machine learning* dan *deep learning* sehingga mampu melakukan klasifikasi data teks secara lebih detail dan akurat. Oleh karena itu, model kecerdasan buatan seperti BERT dan IndoBERT dapat menghasilkan klasifikasi sentimen yang lebih sulit dibedakan tingkat akurasinya dibandingkan metode tradisional karena mampu memahami konteks bahasa secara lebih mendalam [17].

2.3 Natural Language Processing

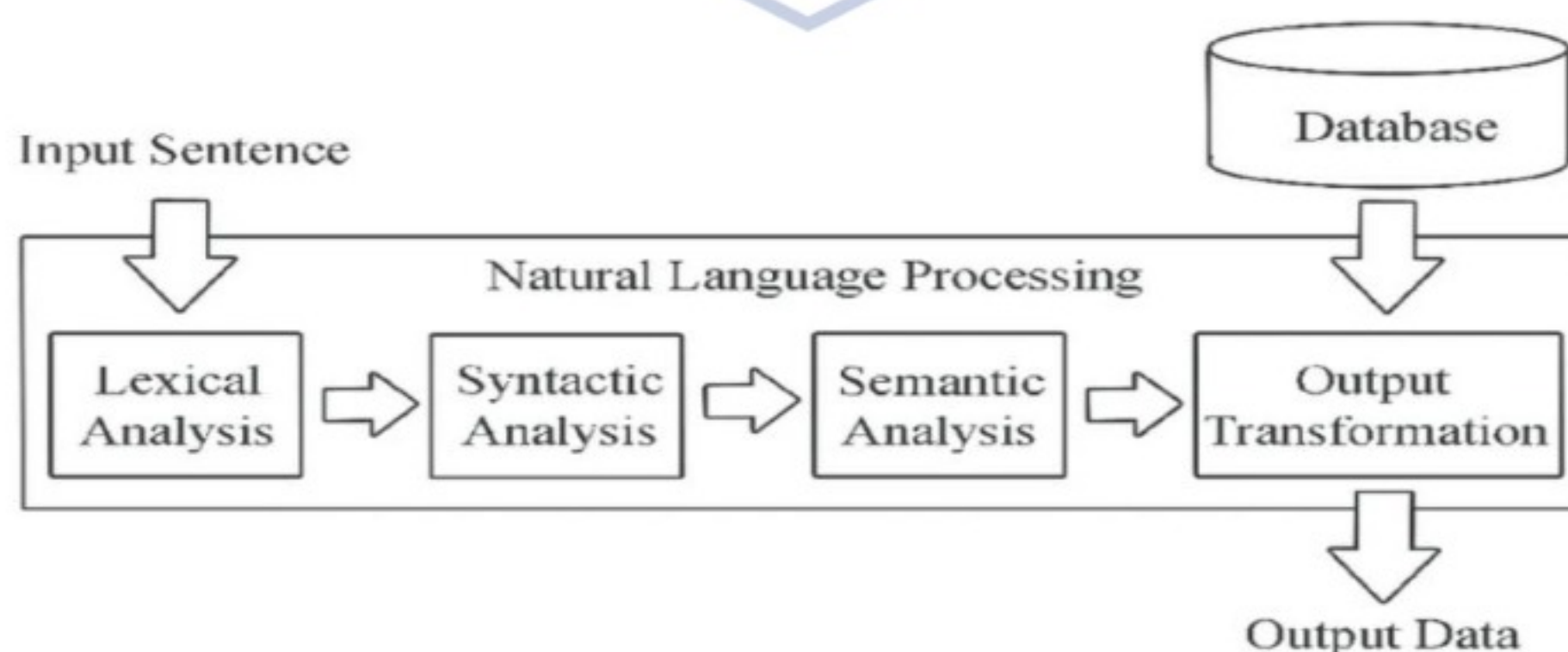
Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan yang berfokus pada kemampuan komputer untuk memahami, memproses, dan menghasilkan bahasa

manusia dalam bentuk teks maupun suara. Dalam penelitian ini, NLP berperan sebagai fondasi utama untuk mengolah komentar pengguna media sosial yang berkaitan dengan kebijakan *Work From Home* (WFH), sehingga opini masyarakat dapat dianalisis secara sistematis menjadi informasi yang bermakna. Menurut Berrajaa (2022), NLP memungkinkan sistem untuk mengekstraksi informasi subjektif seperti opini, emosi, dan sentimen dari teks tidak terstruktur sehingga dapat digunakan dalam proses pengambilan Keputusan [19].

Berbeda dengan pembahasan umum yang hanya menekankan fungsi NLP sebagai alat komunikasi antara manusia dan mesin, pada penelitian ini NLP difokuskan pada pemrosesan teks untuk tugas klasifikasi sentimen. Tahapan yang dilakukan meliputi *text cleaning*, *tokenization*, *stemming*, *vectorization*, hingga proses klasifikasi menggunakan model *deep learning*. Melalui tahapan tersebut, data komentar mentah dapat diubah menjadi representasi numerik yang dapat dipahami oleh model komputasi.

Perkembangan NLP modern sangat dipengaruhi oleh kemajuan metode *deep learning*, khususnya arsitektur Long Short-Term Memory (LSTM) dan Transformer. Model LSTM mampu mempelajari dependensi urutan kata dalam suatu kalimat, sehingga efektif untuk memahami konteks sekuensial teks. Namun, model berbasis Transformer seperti BERT dan IndoBERT memiliki kemampuan yang lebih baik dalam memahami konteks dua arah (*bidirectional context*), sehingga mampu menangkap makna semantik kalimat secara lebih mendalam. Hal ini menjadikan model transformer lebih unggul pada berbagai tugas NLP, termasuk analisis sentimen [20].

Dalam konteks penelitian ini, NLP digunakan sebagai kerangka proses yang menghubungkan data komentar WFH dengan model klasifikasi sentimen. Input penelitian berupa data teks mentah dari media sosial, kemudian diproses melalui tahapan NLP hingga menghasilkan output berupa kategori sentimen positif, negatif, dan netral. Selain itu, penelitian ini juga melakukan perbandingan performa antara model IndoBERT dan LSTM untuk mengetahui metode yang memiliki tingkat akurasi lebih baik dalam memahami opini masyarakat terhadap kebijakan WFH. Berikut diagram tahapan *Natural Language Processing*:



Gambar 2.1 Diagram NLP[21]

2.4 Deep Learning

Deep Learning merupakan salah satu cabang dari *machine learning* yang menggunakan arsitektur jaringan saraf tiruan berlapis (multi-layer neural network) untuk mempelajari pola data secara otomatis. Berbeda dengan metode tradisional yang masih bergantung pada ekstraksi fitur secara manual, *deep learning* mampu mempelajari representasi fitur secara otomatis dari data mentah melalui beberapa lapisan tersembunyi (*hidden layers*). Menurut Nandwani dan Verma (2021), metode *deep learning* banyak digunakan dalam analisis sentimen karena memiliki kemampuan yang lebih baik dalam memahami pola kompleks serta hubungan antar kata dalam teks [14].

Dalam beberapa tahun terakhir, *deep learning* telah menjadi pendekatan yang dominan dalam penelitian analisis sentimen, khususnya pada data teks yang berasal dari media sosial. Model *deep learning* bekerja dengan pendekatan *data-driven*, yaitu membangun pola pembelajaran berdasarkan data dalam jumlah besar, bukan hanya berdasarkan aturan atau model yang telah ditentukan sebelumnya. Pendekatan ini memungkinkan model untuk menghasilkan akurasi klasifikasi yang lebih tinggi pada data opini masyarakat, komentar, maupun ulasan pengguna [12].

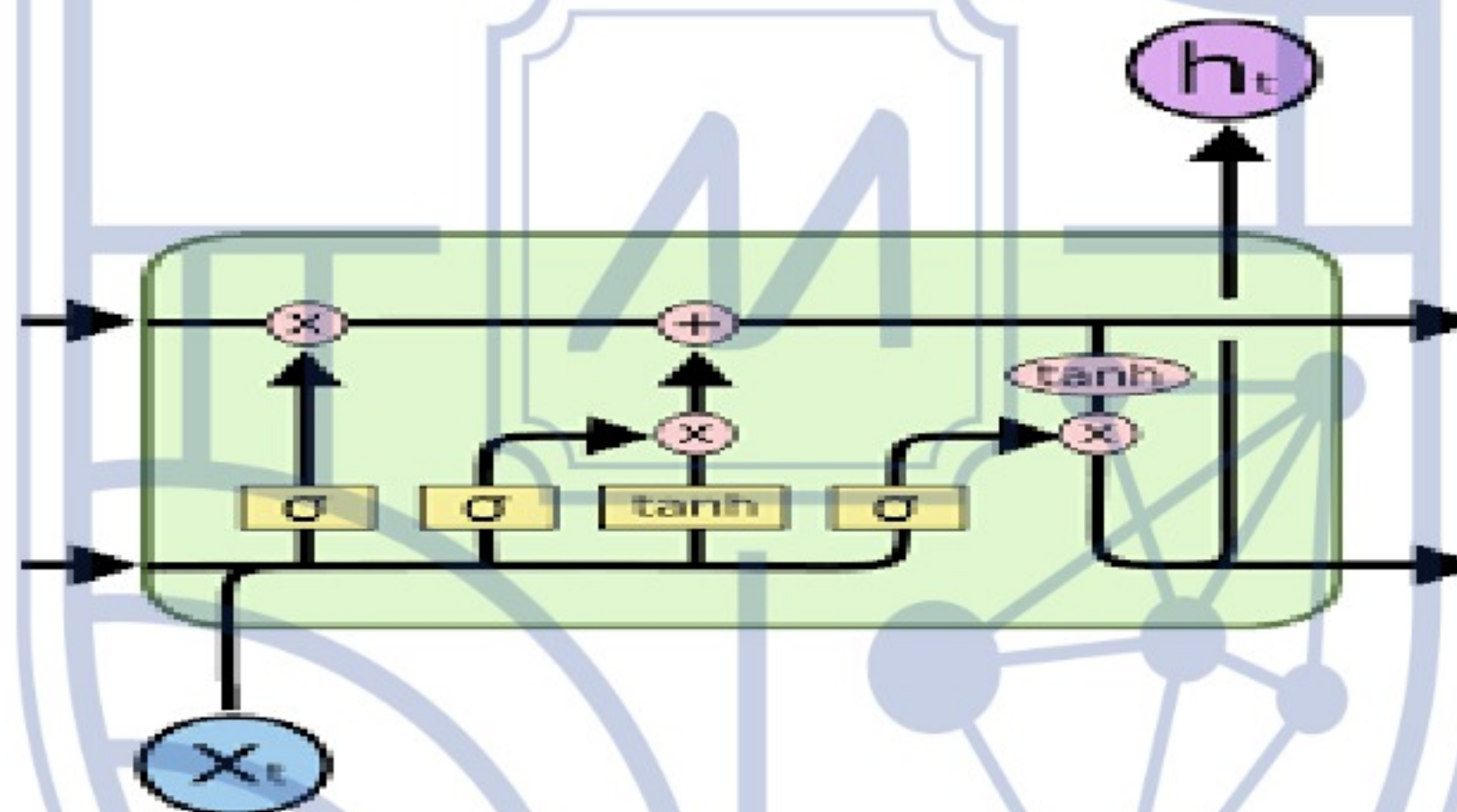
Beberapa arsitektur *deep learning* yang umum digunakan dalam analisis sentimen adalah Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), dan Transformer. Model LSTM merupakan pengembangan dari RNN yang dirancang untuk mengatasi masalah *vanishing gradient* serta lebih efektif dalam memahami hubungan kata dalam urutan teks yang panjang. Sementara itu, model berbasis Transformer seperti BERT dan IndoBERT memiliki kemampuan yang lebih unggul karena mampu memahami konteks kalimat secara dua arah (*bidirectional context*) [17].

Terdapat beberapa varian dari RNN yang dikembangkan untuk meningkatkan kemampuan model dalam memproses teks. Perbedaan utama dari setiap model terletak pada cara menyimpan dan memanfaatkan informasi dari data sebelumnya. Salah satu kelemahan RNN standar adalah keterbatasannya dalam mempertahankan informasi jangka panjang, terutama pada kalimat yang

memiliki urutan kata yang panjang. Untuk mengatasi hal tersebut, dikembangkan model khusus seperti Long Short-Term Memory (LSTM) yang mampu menyimpan data masukan sebelumnya dengan lebih efektif melalui mekanisme *memory cell* dan *gates* [22].

2.4.1 Long Short Term-Memory (LSTM)

Long Short-Term Memory (LSTM) merupakan salah satu pengembangan dari arsitektur *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi keterbatasan RNN dalam menyimpan informasi jangka panjang. Pada analisis sentimen, LSTM sangat efektif digunakan untuk memproses data teks karena mampu memahami hubungan antar kata dalam sebuah kalimat berdasarkan urutan kemunculannya. Hal ini penting karena makna suatu opini sering kali dipengaruhi oleh konteks kata sebelumnya maupun sesudahnya [23].



Gambar 2.2
[24]

Diagram LSTM

Arsitektur *Long Short-Term Memory* (LSTM) terdiri dari beberapa komponen utama yang berfungsi untuk mengatur aliran informasi pada data sekuensial. Secara umum, LSTM memiliki tiga gerbang utama, yaitu *forget gate*, *input gate*, dan *output gate*, serta dua komponen penting berupa *cell state* dan *hidden state*. Struktur ini memungkinkan model menyimpan informasi penting dari kata-kata sebelumnya dan menggunakannya dalam proses prediksi sentimen.

Pada tahap pertama, informasi dari *hidden state* sebelumnya (h_{t-1}) dan input saat ini (x_t) diproses melalui *forget gate* untuk menentukan informasi mana yang perlu dipertahankan atau dibuang dari *cell state*. Proses ini dirumuskan sebagai:

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

Keterangan:

(1) f_t

Merupakan nilai output dari *forget gate* pada waktu ke- t . Nilai ini berada pada rentang 0 sampai 1 yang menunjukkan tingkat “kelupaan” terhadap informasi sebelumnya.

(2) σ

Fungsi aktivasi sigmoid yang digunakan untuk mengubah nilai menjadi rentang 0–1.

- Nilai mendekati 0 $\rightarrow$ informasi dibuang
- Nilai mendekati 1 $\rightarrow$ informasi dipertahankan

(3) W_f

Bobot (*weight*) pada *forget gate* yang akan dipelajari selama proses training.

- $[h_{t-1}, x_t]$

Gabungan (*concatenation*) antara:

- h_{t-1} : *hidden state* pada waktu sebelumnya
- x_t : *input* pada waktu sekarang

(4) b_f

Bias pada *forget gate* yang juga dipelajari selama proses training.

Nilai f_t berada pada rentang 0 hingga 1. Jika nilai mendekati 0, maka informasi lama akan dibuang, sedangkan jika mendekati 1, informasi akan tetap dipertahankan pada *cell state*.

Selanjutnya, pada *input gate*, model menentukan informasi baru yang akan ditambahkan ke *cell state*. Tahap ini terdiri dari dua proses, yaitu fungsi sigmoid untuk menentukan bobot informasi baru dan fungsi *tanh* untuk membentuk kandidat nilai memori. Persamaan *input gate* dapat dituliskan sebagai:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \dots\dots\dots (2)$$

Keterangan:

(1) i_t : output *input gate* pada waktu ke- t (nilai antara 0–1)

(2) σ : fungsi aktivasi sigmoid untuk menentukan tingkat kepentingan informasi

(3) W_i : bobot (*weight*) pada *input gate*

(4) $[h_{t-1}, x_t]$: gabungan *hidden state* sebelumnya dan input saat ini

(5) b_i : bias pada *input gate*

Kemudian kandidat memori baru dirumuskan sebagai:

$$\{\tilde{C}\}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \dots\dots\dots(3)$$

Keterangan:

- (1) i_t : kandidat nilai *cell state* baru
- (2) σ : fungsi aktivasi *hyperbolic tangent* (rentang -1 sampai 1)
- (3) W_i : bobot untuk kandidat memori
- (4) $[h_{t-1}, x_t]$: gabungan *hidden state* sebelumnya dan input saat ini
- (5) b_i :bias kandidat memori

Setelah proses *forget gate* dan *input gate* selesai, tahap berikutnya adalah memperbarui nilai *cell state*. Pada tahap ini, informasi lama dari *cell state* sebelumnya ($C_t - 1$)dikombinasikan dengan informasi baru yang dihasilkan dari *input gate*. Proses pembaruan ini bertujuan agar model dapat mempertahankan informasi penting dari kata-kata sebelumnya sekaligus menambahkan konteks baru dari input saat ini. Persamaan pembaruan *cell state* dituliskan sebagai:

$$C_t = f_t C_{t-1} + \tilde{i}_t \{C\}_t \dots\dots\dots(4)$$

Keterangan:

- (1) C_t : *cell state* pada waktu ke-t (memori terbaru)
- (2) C_{t-1} : *cell state* pada waktu sebelumnya
- (3) f_t : nilai dari *forget gate* (mengontrol informasi lama yang dipertahankan)
- (4) $\tilde{i}_t$: nilai dari *input gate* (mengontrol informasi baru yang masuk)
- (5) C_t : kandidat memori baru

Nilai tersebut kemudian dikombinasikan dengan hasil aktivasi *tanh* dari *cell state* untuk menghasilkan *hidden state* akhir, dengan persamaan:

$$h_t = o_t \tanh(C_t) \dots\dots\dots(5)$$

Hidden state inilah yang selanjutnya digunakan sebagai keluaran pada waktu t dan diteruskan ke langkah berikutnya dalam proses analisis teks.

2.4.2 BiLSTM

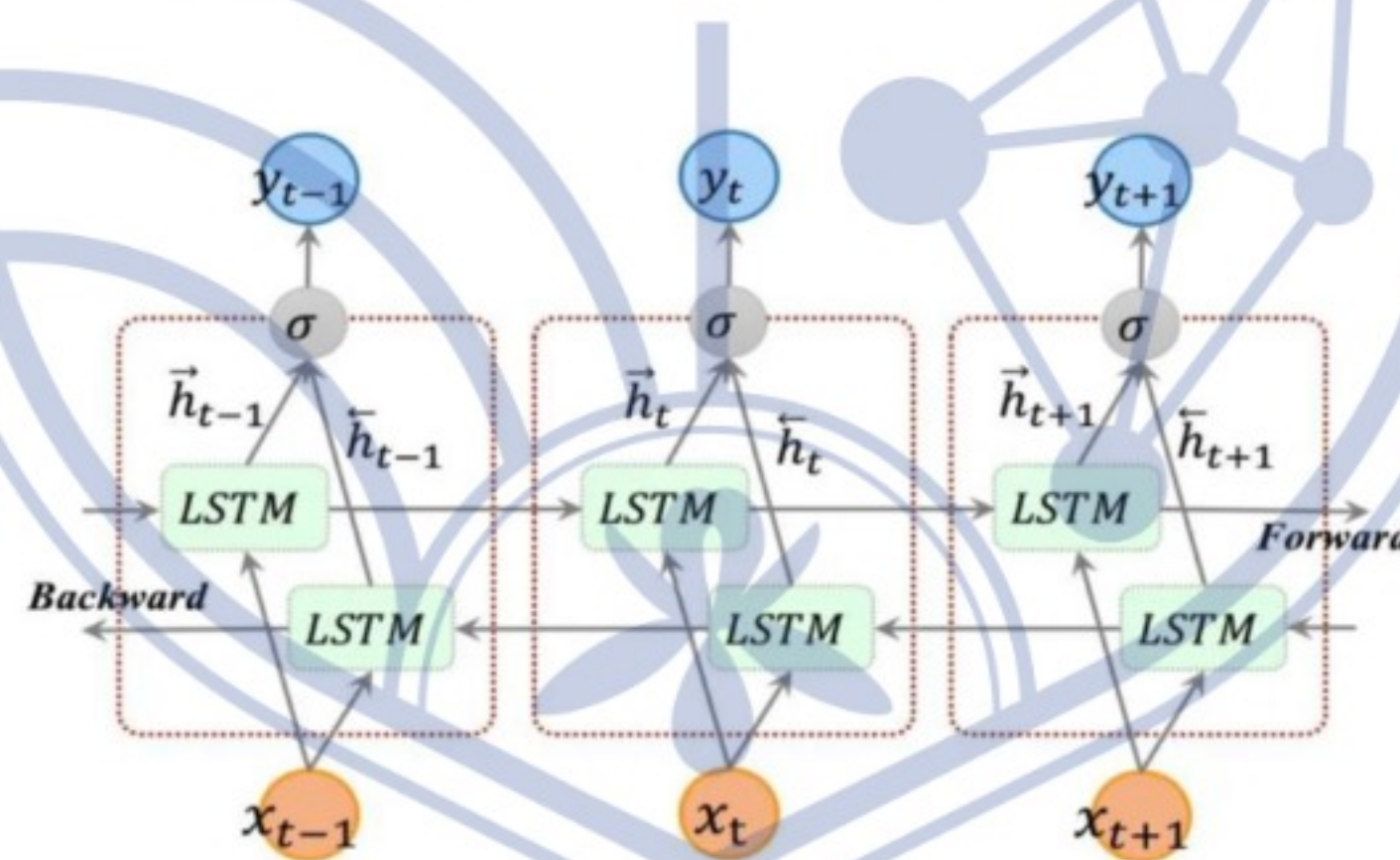
Bidirectional Long Short-Term Memory (BiLSTM) merupakan pengembangan dari arsitektur Long Short-Term Memory (LSTM) yang dirancang untuk meningkatkan kemampuan model dalam memahami konteks data sekuensial dari dua arah, yaitu arah maju (*forward*) dan arah mundur (*backward*). Berbeda dengan LSTM konvensional yang hanya memproses urutan kata dari

awal ke akhir, BiLSTM memproses teks secara simultan dari kiri ke kanan dan dari kanan ke kiri, sehingga mampu menangkap informasi dari kata sebelumnya maupun kata sesudahnya dalam satu kalimat. Kemampuan ini sangat penting pada tugas analisis sentimen karena makna suatu kata sering kali dipengaruhi oleh konteks di sekitarnya [25].

Secara arsitektur, BiLSTM terdiri atas dua lapisan LSTM, yaitu *forward LSTM* dan *backward LSTM*. Lapisan *forward* memproses urutan input dari token pertama hingga token terakhir, sedangkan lapisan *backward* memproses urutan dari token terakhir menuju token pertama. Output dari kedua arah tersebut kemudian digabungkan (*concatenate*) untuk membentuk representasi kontekstual yang lebih kaya. Dengan demikian, BiLSTM mampu memahami hubungan kata dalam konteks dua arah secara lebih baik dibandingkan LSTM biasa [26].

Pada setiap arah, BiLSTM tetap menggunakan tiga gerbang utama, yaitu *forget gate*, *input gate*, dan *output gate*, serta komponen *cell state* dan *hidden state*. *Forget gate* berfungsi menentukan informasi lama yang perlu dipertahankan atau dibuang, *input gate* menentukan informasi baru yang akan ditambahkan ke memori, sedangkan *output gate* menghasilkan keluaran berdasarkan informasi yang tersimpan pada *cell state* [26].

Diagram BiLSTM sebagai berikut:



Gambar 2.3 Diagram BiLSTM[27]

Keterangan:

- x_t : Input pada waktu ke-t (misalnya kata dalam kalimat)
- LSTM (kotak hijau) : Unit pemrosesan yang menangkap dependensi jangka panjang
- Forward* : Aliran informasi dari masa lalu ke masa depan

- d. *Backward* : Aliran informasi dari masa depan ke masa lalu
- e. $\vec{h_t}$ dan $\overleftarrow{h_t}$: *Hidden state* dari masing-masing arah
- f. y_t : Output pada waktu ke- t
- g. σ : Fungsi aktivasi (biasanya sigmoid atau softmax)

2.4.3 Bert (Bidirectional Encoder Representations from Transformers)

Bidirectional Encoder Representations from Transformers (BERT) merupakan model *pre-trained language representation* yang dikembangkan oleh Google berbasis arsitektur *transformer*. BERT dirancang untuk meningkatkan kemampuan pemrosesan bahasa alami (*Natural Language Processing/NLP*) dengan memanfaatkan mekanisme *self-attention* yang memungkinkan model memahami hubungan antar kata dalam suatu kalimat secara lebih mendalam [28].

BERT merupakan model representasi kata kontekstual yang dilatih menggunakan pendekatan *Masked Language Model* (MLM) dan *Next Sentence Prediction* (NSP). Pada metode MLM, sebagian kata dalam kalimat akan disembunyikan (*masked*), kemudian model dilatih untuk memprediksi kata yang hilang tersebut berdasarkan konteks sekitarnya. Sedangkan pada NSP, model dilatih untuk memahami hubungan antar kalimat dengan menentukan apakah suatu kalimat merupakan kelanjutan dari kalimat sebelumnya. Pendekatan ini memungkinkan BERT memahami struktur bahasa secara lebih komprehensif [28].

Secara arsitektur, BERT menggunakan *encoder* dari *transformer* yang terdiri dari beberapa lapisan (*layers*) yang ditumpuk (*stacked*). Setiap lapisan memiliki dua komponen utama, yaitu multi-head self-attention dan feed-forward neural network. Proses dimulai dari tahap tokenisasi, di mana teks dipecah menjadi token berupa kata, sub-kata, atau karakter. Selanjutnya, token tersebut diubah menjadi representasi vektor melalui proses *embedding*. Vektor ini kemudian diproses oleh lapisan *transformer encoder* untuk menghasilkan representasi kontekstual yang semakin kompleks pada setiap lapisannya [28].

Keunggulan utama BERT terletak pada kemampuannya dalam memahami konteks secara dua arah (*bidirectional*), yaitu dengan memperhatikan hubungan kata baik dari sisi kiri maupun kanan dalam suatu kalimat. Hal ini berbeda dengan model sebelumnya seperti LSTM yang umumnya memproses teks secara sekuensial satu arah atau terbatas. Dengan pendekatan ini, BERT mampu menangkap makna kata yang bergantung pada konteks kalimat secara lebih akurat [28].

Dalam implementasinya, BERT melalui dua tahapan utama, yaitu pre-training dan fine-tuning. Pada tahap *pre-training*, model dilatih menggunakan data dalam jumlah besar tanpa label untuk memahami struktur bahasa. Selanjutnya, pada tahap *fine-tuning*, model disesuaikan dengan tugas spesifik seperti klasifikasi teks atau analisis sentimen menggunakan data berlabel [29].

2.5 IndoBERT

IndoBERT merupakan model bahasa berbasis arsitektur BERT (*Bidirectional Encoder Representations from Transformers*) yang dikembangkan secara khusus untuk memproses dan memahami teks berbahasa Indonesia. Model ini diperkenalkan oleh Koto et al.(2020) sebagai bagian dari proyek IndoLEM (*Indonesian Language Evaluation Montage*), sebuah *benchmark* komprehensif yang mencakup tujuh tugas NLP untuk bahasa Indonesia, mulai dari morfosintaks, semantik, hingga analisis wacana. Berbeda dengan model multilingual seperti mBERT yang dilatih menggunakan data dari ratusan bahasa sekaligus, IndoBERT dilatih secara monolingual menggunakan lebih dari 220 juta kata yang dikumpulkan dari berbagai sumber teks berbahasa Indonesia, meliputi artikel berita, tulisan blog, teks media sosial, dan berbagai konten daring lainnya [30]. Pendekatan pelatihan monolingual ini memungkinkan IndoBERT memiliki pemahaman yang jauh lebih mendalam terhadap struktur, kosakata, dan konteks penggunaan bahasa Indonesia dibandingkan model multilingual yang harus berbagi kapasitas representasinya dengan ratusan bahasa lain secara bersamaan.

Secara arsitektur, IndoBERT dibangun di atas fondasi model BERT yang pertama kali diperkenalkan oleh Devlin et al. (2018) dari Google. BERT sendiri merupakan terobosan besar dalam dunia NLP karena memperkenalkan mekanisme *self-attention* berbasis transformer yang mampu memproses teks secara bidireksional, artinya model ini memahami makna sebuah kata tidak hanya berdasarkan kata-kata yang mendahuluinya, tetapi juga mempertimbangkan kata-kata yang mengikutinya dalam satu kalimat secara bersamaan. Kemampuan pemahaman konteks dua arah inilah yang membedakan BERT dan turunannya, termasuk IndoBERT, dari model-model sebelumnya seperti LSTM dan GRU yang hanya mampu memproses teks secara sekuensial satu arah atau dua arah secara terpisah [30]. IndoBERT tersedia dalam dua varian ukuran, yaitu IndoBERT -base yang terdiri dari 12 lapisan transformer dengan 768 dimensi *hidden state* dan 12 *attention head*, serta IndoBERT -large yang memiliki kapasitas lebih besar dengan 24 lapisan transformer. Dalam praktiknya, IndoBERT -base lebih sering digunakan dalam berbagai penelitian

karena menawarkan keseimbangan yang baik antara performa model dan kebutuhan sumber daya komputasi.

Proses pelatihan IndoBERT menggunakan dua tugas utama yang menjadi ciri khas arsitektur BERT, yaitu *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP). Pada tugas MLM, sebagian token dalam teks input disembunyikan secara acak dan model dilatih untuk memprediksi token yang disembunyikan tersebut berdasarkan konteks kata-kata di sekitarnya, baik dari arah kiri maupun kanan. Mekanisme ini memaksa model untuk benar-benar memahami konteks kalimat secara menyeluruh, bukan sekadar menghafal urutan kata. Sementara itu, tugas NSP melatih model untuk memprediksi apakah dua kalimat yang diberikan merupakan kalimat yang berurutan dalam teks aslinya atau tidak, sehingga model juga mampu memahami hubungan antar kalimat dalam wacana yang lebih panjang. Kombinasi kedua tugas pelatihan ini menghasilkan model yang memiliki representasi bahasa yang kaya dan kontekstual, yang kemudian dapat diadaptasi untuk berbagai tugas NLP hilir (*downstream tasks*) melalui proses *fine-tuning* menggunakan dataset berlabel yang spesifik untuk setiap tugas.

Dalam penerapannya untuk analisis sentimen, IndoBERT bekerja melalui mekanisme *transfer learning* di mana pengetahuan bahasa yang telah diperoleh selama proses *pre-training* dimanfaatkan sebagai titik awal untuk mempelajari tugas klasifikasi sentimen yang lebih spesifik. Token khusus [CLS] yang ditempatkan di awal setiap input teks berfungsi sebagai representasi agregat dari keseluruhan kalimat, dan representasi inilah yang kemudian diteruskan ke lapisan klasifikasi tambahan untuk menghasilkan prediksi label sentimen [31]. Pendekatan ini terbukti jauh lebih efisien dibandingkan melatih model dari nol, karena model hanya perlu melakukan penyesuaian minor pada bobot-bobotnya menggunakan data berlabel yang relatif sedikit untuk mencapai performa yang tinggi [31]. Keunggulan IndoBERT dalam menangani teks informal berbahasa Indonesia juga menjadikannya sangat relevan untuk analisis data media sosial yang penuh dengan variasi linguistik, bahasa tidak baku, singkatan, dan campuran bahasa. Hal ini dibuktikan oleh berbagai penelitian yang secara konsisten menunjukkan bahwa IndoBERT mampu mengungguli metode *deep learning* konvensional seperti BiLSTM dalam berbagai tugas klasifikasi teks berbahasa Indonesia, termasuk analisis sentimen terhadap ulasan produk, komentar media sosial, hingga opini publik terhadap kebijakan pemerintah.

2.6 Confussion Matrix

Dalam mengevaluasi performa model klasifikasi pada tugas analisis sentimen, salah satu alat ukur yang paling banyak digunakan adalah *confusion matrix*. Alat evaluasi ini bekerja dengan cara menyandingkan label aktual dari data uji dengan label hasil prediksi yang dihasilkan oleh model, kemudian menyajikannya dalam bentuk tabel yang memudahkan peneliti untuk melihat secara rinci di mana model berhasil melakukan prediksi dengan benar dan di mana model mengalami kesalahan klasifikasi [32].

Pada klasifikasi biner, *confusion matrix* tersusun atas empat komponen utama yang menjadi dasar perhitungan berbagai metrik evaluasi. Pertama, *True Positive* (TP) merujuk pada jumlah data berlabel positif yang berhasil diprediksi secara tepat oleh model sebagai positif. Kedua, *True Negative* (TN) menunjukkan jumlah data berlabel negatif yang juga berhasil diprediksi dengan benar sebagai negatif. Adapun dua komponen lainnya mencerminkan kesalahan prediksi model, yaitu *False Positive* (FP) yang terjadi ketika model keliru mengklasifikasikan data negatif sebagai positif, dan *False Negative* (FN) yang terjadi ketika data yang sebenarnya positif justru diprediksi sebagai negatif oleh model [33].

Keempat komponen tersebut kemudian menjadi landasan dalam menghitung berbagai metrik evaluasi yang lebih informatif. *Accuracy* dihitung sebagai proporsi keseluruhan prediksi yang benar terhadap total data uji, namun metrik ini perlu diinterpretasikan dengan hati-hati ketika distribusi kelas dalam dataset tidak seimbang. Untuk mengatasi keterbatasan tersebut, digunakan *precision* yang mengukur seberapa tepat model dalam memprediksi kelas positif, *recall* yang menilai kemampuan model dalam mendeteksi seluruh data positif yang sesungguhnya ada, serta *F1-score* yang menggabungkan keduanya dalam satu nilai tunggal melalui perhitungan rata-rata harmonik. Kombinasi keempat metrik inilah yang memberikan gambaran paling komprehensif tentang kualitas model klasifikasi, termasuk dalam konteks penelitian ini yang menggunakan IndoBERT untuk menganalisis sentimen masyarakat terhadap kebijakan *Work From Home* (WFH) di media sosial [32].

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2.4 Confusion Matrix[34]

Keterangan:

- True Positive (TP): kondisi ketika model memprediksi data sebagai kelas positif dan hasil prediksi tersebut sesuai dengan label sebenarnya.
- True Negative (TN): kondisi ketika model memprediksi data sebagai kelas negatif dan prediksi tersebut benar sesuai data actual
- False Positive (FP): kondisi ketika model memprediksi data sebagai positif, namun label sebenarnya adalah negatif. Kesalahan ini sering disebut sebagai error tipe I.
- False Negative (FN): kondisi ketika model memprediksi data sebagai negatif, padahal label sebenarnya adalah positif. Kesalahan ini dikenal sebagai error tipe II.

Berdasarkan nilai yang terdapat pada *confusion matrix*, selanjutnya dapat dilakukan perhitungan berbagai metrik evaluasi, seperti *accuracy*, *precision*, *recall*, dan *F1-score*.

1. Accuracy

Accuracy digunakan untuk menunjukkan tingkat ketepatan model dalam melakukan klasifikasi terhadap seluruh data uji. Nilai ini diperoleh dari perbandingan antara jumlah prediksi yang benar dengan total keseluruhan data.

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

Keterangan:

- TP (True Positive): jumlah data positif yang diprediksi benar sebagai positif
- TN (True Negative): jumlah data negatif yang diprediksi benar sebagai negative
- FP (False Positive): jumlah data negatif tetapi diprediksi sebagai positif

- d FN (False Negative): jumlah data positif tetapi diprediksi sebagai negatif

2. Precision

Precision menunjukkan tingkat ketepatan model dalam memprediksi suatu kelas positif. Metrik ini mengukur seberapa banyak data yang diprediksi positif benar-benar termasuk ke dalam kelas positif.

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (7)$$

Keterangan:

- a TP (True Positive): prediksi positif yang benar
- b FP (False Positive): prediksi positif yang salah

3. Recall

Recall digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh data yang benar-benar termasuk ke dalam kelas positif.

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (8)$$

Keterangan:

- a TP (True Positive): data positif yang berhasil terdeteksi
- b FN (False Negative): data positif yang gagal terdeteksi

4. F1- Score

F1-score merupakan metrik yang menggabungkan nilai *precision* dan *recall* dalam satu ukuran evaluasi. Metrik ini sangat penting terutama ketika distribusi kelas dalam dataset tidak seimbang.

$$F1-Score = 2 \times \frac{precision \times recall}{precision+recall} \dots\dots\dots (9)$$

Keterangan:

- a Precision × Recall: kombinasi performa ketepatan dan kelengkapan
- b Precision + Recall: normalisasi agar seimbang
- c 2 ×: faktor pengali untuk menghasilkan rata-rata harmonis

2.7 Cohen's Kappa

Cohen's Kappa (κ) merupakan ukuran statistik yang digunakan untuk mengukur tingkat kesepakatan (agreement) antara dua penilai (annotator) dalam memberikan label terhadap suatu data dengan mempertimbangkan kemungkinan kesepakatan yang terjadi secara kebetulan (chance agreement). Berbeda dengan persentase kesepakatan biasa, Cohen's Kappa memberikan hasil yang lebih objektif karena mengurangi pengaruh kesamaan yang muncul secara acak.

Dalam penelitian analisis sentimen, Cohen's Kappa banyak digunakan untuk mengevaluasi konsistensi proses pelabelan data sebelum dataset digunakan pada proses pelatihan model klasifikasi. Nilai Kappa yang tinggi menunjukkan bahwa label yang diberikan oleh dua penilai memiliki tingkat kesesuaian yang baik sehingga dataset dapat dianggap reliabel sebagai ground truth.

Nilai Cohen's Kappa dihitung menggunakan persamaan berikut.

$$\kappa = \frac{P_o - P_e}{1 - P_e} \dots\dots\dots (10)$$

Keterangan:

- κ = nilai Cohen's Kappa
- P_o = proporsi kesepakatan aktual antar penilai
- P_e = proporsi kesepakatan yang terjadi secara kebetulan

Interpretasi nilai Cohen's Kappa mengacu pada klasifikasi yang dikemukakan oleh Landis dan Koch (1977) sebagaimana ditunjukkan pada Tabel berikut.

Tabel 2.1 Nilai Parameter *Cohen's Kappa*

Nilai Kappa	Interpretasi
< 0	Tidak ada kesepakatan
0,00–0,20	:Sangat rendah
0,21–0,40	Rendah
0,41–0,60	Sedang
0,61–0,80	Tinggi (Substantial Agreement)

Nilai Kappa	Interpretasi
0,81–1,00	Sangat tinggi (Almost Perfect Agreement)

Pada penelitian ini, Cohen's Kappa digunakan untuk mengevaluasi tingkat kesesuaian antara hasil pelabelan awal yang dihasilkan oleh model Artificial Intelligence (AI) dengan hasil validasi manual oleh peneliti. Penggunaan metrik ini bertujuan untuk memastikan bahwa proses pelabelan memiliki tingkat reliabilitas yang baik sehingga dataset layak digunakan dalam proses pelatihan dan evaluasi model IndoBERT maupun BiLSTM.

