BAB 2
LANDASAN TEORI

2.1 Depresi dan Media Sosial

2.1.1 Definisi Depresi

Depresi merupakan gangguan kesehatan mental yang ditandai oleh suasana hati yang terus-menerus rendah, hilangnya minat atau kesenangan terhadap aktivitas sehari-hari, serta berbagai gejala kognitif, fisik, dan perilaku yang berlangsung setidaknya selama dua minggu [1]. *World Health Organization* mendefinisikan depresi sebagai kondisi yang berbeda dari perubahan suasana hati yang umum dialami sehari-hari dan respons emosional jangka pendek terhadap tantangan kehidupan [1].

Menurut *Diagnostic and Statistical Manual of Mental Disorders* edisi kelima (DSM-5), diagnosis depresi mayor ditegakkan apabila terdapat setidaknya lima dari sembilan gejala utama dalam periode yang sama selama minimal dua minggu, yaitu suasana hati depresif, kehilangan minat atau kesenangan (*anhedonia*), perubahan berat badan atau nafsu makan, gangguan tidur, agitasi atau retardasi psikomotor, kelelahan, perasaan tidak berharga, kesulitan berkonsentrasi, dan pikiran berulang tentang kematian atau bunuh diri [5]. Kriteria DSM-5 mensyaratkan setidaknya satu dari dua gejala inti hadir, yaitu suasana hati depresif atau *anhedonia*, yang menjadikan kriteria ini lebih ketat dibandingkan sekadar menghitung jumlah gejala [5].

2.1.2 Ekspresi Depresi pada Media Sosial Indonesia

Pertumbuhan penggunaan media sosial terutama di kalangan remaja dan orang dewasa muda membawa perubahan baru dalam cara seseorang mengekspresikan emosi dan tekanan mental [19]. Banyak orang cenderung mengungkapkan perasaannya melalui platform digital seperti X (Twitter), baik secara eksplisit maupun implisit. Dengan demikian, media sosial dapat menjadi sumber informasi penting untuk mendeteksi tanda-tanda awal gangguan mental.

Adanya stigma terkait kesehatan mental di Indonesia seringkali membuat orang merasa terdorong untuk mencari tempat anonim di platform media sosial guna mengekspresikan keluhan atau meluapkan perasaan yang tidak dapat

disampaikan secara langsung di kehidupan nyata [19]. Perilaku linguistik yang berhubungan dengan depresi sering muncul melalui bahasa yang mencerminkan kehilangan rasa nikmat, putus asa, rasa bersalah, dan keinginan untuk menyakiti diri [19].

Penelitian terdahulu mencatat bahwa penggunaan istilah tertentu di kalangan mahasiswa di Indonesia, seperti “capek”, “lelah”, atau “malas hidup”, berkaitan erat dengan tingkat depresi yang tinggi [19, 21].

2.1.3 Pemetaan Emosi ke Indikasi Depresi

Ketiga dataset yang digunakan pada penelitian ini dianotasi pada tataran emosi diskrit, sedangkan tugas yang dikerjakan adalah klasifikasi biner antara kondisi terindikasi depresi dan kondisi normal. Kerangka yang menghubungkan keduanya adalah model tripartit afek yang diajukan Clark dan Watson [6]. Model ini membagi gejala depresi dan ansietas ke dalam tiga komponen, yaitu afek negatif, afek positif, dan *physiological hyperarousal*. Afek negatif yang tinggi bersifat non-spesifik karena dimiliki bersama oleh depresi dan ansietas, *physiological hyperarousal* relatif khas bagi ansietas, sedangkan afek positif yang rendah bersifat spesifik bagi depresi [31]. Depresi karenanya dicirikan oleh kombinasi dua kondisi sekaligus, yaitu afek negatif yang tinggi disertai afek positif yang rendah.

Kedua komponen afek tersebut merupakan dimensi yang menjadi dasar pengelompokan label pada penelitian ini. Afek negatif (*negative affect*, NA) adalah dimensi distres subjektif yang menaungi keadaan emosi tidak menyenangkan seperti kemarahan, ketakutan, dan kegugupan. Nilai NA yang tinggi menandakan tekanan psikologis. Afek positif (*positive affect*, PA) adalah dimensi keterlibatan yang menyenangkan dengan lingkungan, mencakup kegembiraan dan kasih sayang. Nilai PA yang rendah menandakan kelesuan dan hilangnya minat atau anhedonia. Keduanya bersifat relatif independen, bukan dua ujung dari satu garis yang sama, sehingga PA yang rendah tidak identik dengan NA yang tinggi [32].

Penempatan setiap label emosi pada kedua komponen tersebut tidak ditentukan secara intuitif, melainkan mengikuti hasil analisis faktor terhadap kata sifat pelaporan mandiri suasana hati. Watson dan Tellegen [32] meninjau dan menganalisis ulang sejumlah penelitian struktur afek, dan menemukan bahwa deskriptor suasana hati terdistribusi secara konsisten ke dalam delapan oktan pada interval 45° berdasarkan muatan faktornya terhadap dimensi NA dan PA. Oktan *high negative affect* memuat deskriptor *distressed*, *fearful*, *hostile*, *jittery*, *nervous*,

dan *scornful*.Pada oktan *unpleasantness* memuat *sad*, *blue*, *lonely*, dan *unhappy*. Pada oktan *pleasantness* memuat *happy*, *content*, *pleased*, *kindly*, dan *warmhearted*. Sedangkan pada oktan *high positive affect* memuat *elated*, *enthusiastic*, *excited*, dan *active*. Pembagian yang sama dioperasionalkan dalam instrumen PANAS [33], yang mengukur NA melalui butir *hostile*, *irritable*, *scared*, *afraid*, *nervous*, dan *jittery*, serta mengukur PA melalui butir *excited*, *enthusiastic*, *active*, dan *interested*.

Berdasarkan struktur tersebut, keenam label emosi pada dataset dapat ditempatkan sebagai berikut. Label *anger* berpadanan dengan deskriptor *hostile*, *scornful*, dan *irritable* sehingga menempati oktan NA tinggi. Label *fear* berpadanan dengan *fearful*, *scared*, *nervous*, dan *jittery* yang seluruhnya berada pada oktan yang sama. Label *sadness* berpadanan dengan *sad*, *blue*, dan *unhappy* yang menempati oktan *unpleasantness*, yaitu posisi di antara NA tinggi dan PA rendah, sehingga label ini memuat kedua komponen penanda depresi sekaligus. Sebaliknya, label *joy* dan *happy* berpadanan dengan *happy*, *pleased*, *enthusiastic*, dan *excited*, sedangkan label *love* berpadanan dengan *warmhearted* dan *kindly*. Pada keduanya menempati wilayah PA tinggi disertai NA rendah. Ketiga label kelompok pertama karenanya membentuk profil NA tinggi disertai PA rendah yang menjadi penanda depresi menurut model tripartit sehingga dipetakan ke kelas terindikasi depresi, sedangkan ketiga label kelompok kedua dipetakan ke kelas normal. Pengelompokan *anger* diperkuat temuan empiris bahwa iritabilitas atau kemarahan terbuka hadir pada 54,5% dari 536 pasien episode depresi mayor unipolar [34], meskipun gejala ini bukan kriteria diagnostik DSM-5 bagi orang dewasa [5]. Rincian pemetaan disajikan pada Tabel 4.1.

2.2 Platform X

Seiring pesatnya perkembangan teknologi digital, media sosial telah menjadi ruang di mana individu secara sukarela mengekspresikan kondisi emosional dan psikologis pengguna secara terbuka dan *real-time* [10]. Tinjauan sistematis terhadap literatur deteksi tanda-tanda depresi dari media sosial menunjukkan bahwa Platform X (sebelumnya Twitter) merupakan sumber data yang paling dominan digunakan, dengan karakteristik teks pendek dan ekspresi emosional spontan yang menjadikannya sumber jejak linguistik yang terstruktur dan dapat dianalisis secara komputasional [10]. Meta-analisis terhadap studi prediksi depresi berbasis teks media sosial menggunakan *machine learning* mengonfirmasi bahwa fitur linguistik dari teks kiriman pengguna memberikan kontribusi paling signifikan terhadap

akurasi prediksi, dengan ketidakseimbangan kelas sebagai tantangan utama yang perlu ditangani dalam perancangan sistem deteksi [9]. Dalam konteks Indonesia, Platform X dengan 27,1 juta pengguna aktif pada tahun 2024 menjadi sumber data yang bernilai tinggi untuk keperluan deteksi gejala depresi berbasis teks [4], mengingat kecenderungan pengguna untuk mengekspresikan tekanan psikologis secara spontan dalam bentuk teks pendek di platform tersebut [25].

2.2.1 X sebagai Sumber Data Mental Health

Penelitian terdahulu menjelaskan bahwa X (Twitter) berfungsi sebagai layanan *microblogging* yang memungkinkan pengguna menyebarkan informasi, mempromosikan bisnis, dan mengekspresikan pandangan dalam batasan karakter yang dikenal sebagai *tweet* [18]. Penelitian terdahulu juga menambahkan bahwa X secara khusus berfungsi sebagai media yang mendukung percakapan singkat dan sering kali penuh emosi, sehingga menjadi sumber informasi yang berharga untuk analisis kesehatan mental [19].

Namun, cara komunikasi di media sosial memiliki ciri khas tersendiri seperti penggunaan bahasa yang santai, istilah gaul, singkatan, emotikon, hingga gaya bahasa yang tidak literal seperti sarkasme atau ironi [19]. Hal ini menjadi tantangan yang membuat metode tradisional dalam analisis teks kurang efektif saat diterapkan pada data media sosial yang berbahasa Indonesia.

2.3 Natural Language Processing

Natural Language Processing (NLP) bekerja melalui serangkaian proses yang memungkinkan komputer untuk memahami, menganalisis, dan menghasilkan bahasa alami secara efektif [35]. Proses ini merupakan gabungan dari pendekatan linguistik, statistik, serta teknik pembelajaran mesin yang saling melengkapi untuk mengekstrak informasi bermakna dari teks dalam bahasa manusia [35].

Dalam konteks penelitian ini, NLP diterapkan pada teks berbahasa Indonesia dari Platform X untuk mengidentifikasi pola-pola linguistik yang berkaitan dengan kondisi depresi [35]. Tugas klasifikasi teks merupakan salah satu aplikasi utama NLP, di mana model dilatih untuk memetakan urutan teks masukan ke dalam kategori-kategori yang telah ditentukan [35]. Perkembangan metode NLP dari pendekatan berbasis aturan menuju representasi terdistribusi (*distributed representations*) dan arsitektur *deep learning* berbasis *Transformer* telah secara

signifikan meningkatkan kemampuan model dalam menangkap makna kontekstual dari teks [35].

2.4 Text Pre-Processing

Text Pre-processing merupakan tahap awal dalam *Natural Language Processing* (NLP) yang mengubah teks mentah menjadi bentuk yang lebih standar dan seragam agar sesuai dengan kebutuhan model *machine learning* [35]. Secara umum, tahap ini mencakup *case-folding*, penghapusan karakter dan simbol yang tidak bermakna, normalisasi kata tidak baku, serta pemisahan teks menjadi token [35]. Teks media sosial seperti X menuntut penanganan khusus karena memuat elemen non-linguistik seperti *retweet* (RT), *mention*, URL, dan emoji yang tidak relevan terhadap tugas klasifikasi [22].

2.5 Tokenisasi

Tokenisasi adalah proses memecah teks menjadi unit-unit kecil bermakna yang disebut token, seperti kata, sub-kata, atau karakter [35]. Token inilah yang menjadi input model NLP sehingga teks dapat direpresentasikan secara numerik [35]. Tokenisasi pada level kata memiliki keterbatasan dalam menangani kata diluar kosakata (*out-of-vocabulary*), sehingga model berbasis *Transformer* umumnya menggunakan tokenisasi sub-kata yang memecah kata langka menjadi unit yang lebih kecil [35]. Tokenisasi pada penelitian ini menggunakan tokenizer bawaan IndoBERTweet yaitu WordPiece

2.5.1 WordPiece

WordPiece merupakan algoritma tokenisasi sub-kata yang digunakan dalam keluarga model BERT, termasuk IndoBERTweet [17]. Algoritma ini bekerja berdasarkan kosakata tetap (*fixed vocabulary*) yang dibentuk pada tahap pra-pelatihan model. Dengan demikian, tokenisasi WordPiece tidak menciptakan token baru, melainkan memetakan teks input ke dalam kombinasi subword yang telah didefinisikan sebelumnya dalam kosakata.

Pada tahap pembentukan kosakata, WordPiece diawali dengan unit dasar berupa karakter individual dan simbol khusus. Selanjutnya, pasangan token yang sering muncul secara berurutan dalam korpus pelatihan akan digabungkan secara iteratif berdasarkan kriteria peningkatan probabilitas maksimum terhadap

data [36]. Proses ini menghasilkan kumpulan *subword* dengan ukuran kosakata yang tetap dan terbatas. Pada tahap tokenisasi, WordPiece menerapkan algoritma *Greedy Longest-Match-First* [37]. Proses ini bekerja dengan memecah setiap kata secara deterministik dari kiri ke kanan. Sebagai ilustrasi, dengan asumsi kosakata WordPiece mengandung token `ber`, `##lari`, dan `##an`, maka proses tokenisasi kata *berlarian* dilakukan sebagai berikut:

```
berlarian -> [ber, ##lari, ##an]
```

Apabila pada suatu tahap tidak ditemukan substring yang cocok dalam kosakata, maka karakter yang tidak dikenali akan direpresentasikan sebagai token khusus `[UNK]` (*unknown token*). Pada praktiknya, hal ini jarang terjadi pada teks berbahasa Indonesia karena seluruh karakter Latin sudah tercakup pada kosakata dasar IndoBERTweet, sehingga kata yang sebelumnya tidak dikenal umumnya tetap dapat direpresentasikan melalui pemecahan menjadi karakter atau *subword* yang lebih pendek. Oleh karena itu, kualitas data setelah pembersihan menjadi faktor penting dalam efektivitas tokenisasi WordPiece.

2.6 Transformer & Bidirectional Encoder Representations

Bidirectional Encoder Representations from Transformers (BERT) adalah model bahasa berbasis arsitektur Transformer yang dikembangkan oleh Google AI Language pada thaun 2018 [38]. Model ini menggunakan pendekatan kontekstual dua arah (bidirectional) untuk memahami teks, yang memungkinkan BERT mempertimbangkan konteks dari kedua arah yaitu kiri dan kanan dalam satu kalimat secara simultan [38].

2.6.1 Arsitektur BERT

BERT dibangun sepenuhnya menggunakan arsitektur Transformer, khususnya bagian *encoder* yang terdiri dari lapisan *multi-head self-attention* dan *feed-forward neural network* [38]. Tidak seperti model tradisional seperti RNN atau LSTM yang memproses kata secara sekuensial, BERT memproses semua kata secara paralel menggunakan mekanisme *self-attention* [38].

BERT memiliki dua versi utama [38]:

1. **BERT-Base**: terdiri dari 12 lapisan *encoder*, 768 ukuran vektor tersembunyi (*hidden size*), dan 12 kepala *attention*. Total parameter sekitar 110 juta.

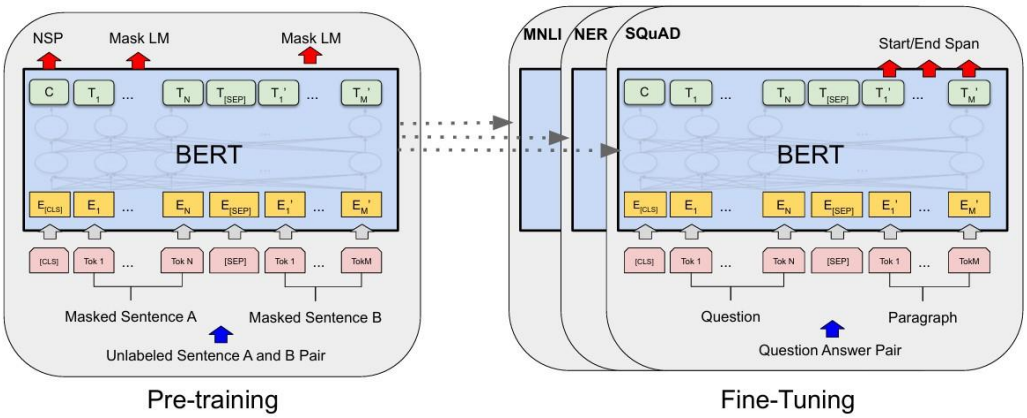
2. **BERT-Large**: terdiri dari 24 lapisan *encoder*, 1024 ukuran vektor tersembunyi, dan 16 kepala *attention*. Total parameter sekitar 340 juta.

2.6.2 Pre-training Tasks

BERT dilatih secara *unsupervised* menggunakan dua tugas utama, yaitu:

1. Masked Language Modeling(MLM): Sebagian token dalam kalimat disembunyikan ([MASK]) dan model dilatih untuk memprediksi token tersebut berdasarkan konteks dua arah. Ini membuat BERT mampu memahami relasi antar kata secara mendalam [38]

2. Next Sentence Prediction (NSP) : Model diberikan dua kalimat dan harus memprediksi apakah kalimat kedua merupakan kelanjutan dari kalimat pertama. NSP membantu model memahami hubungan antar kalimat [38].



Gambar 2.1. Gambaran Pre-training BERT dan Fine-Tuning. [38]

2.6.3 Tokenisasi dan Representasi Input

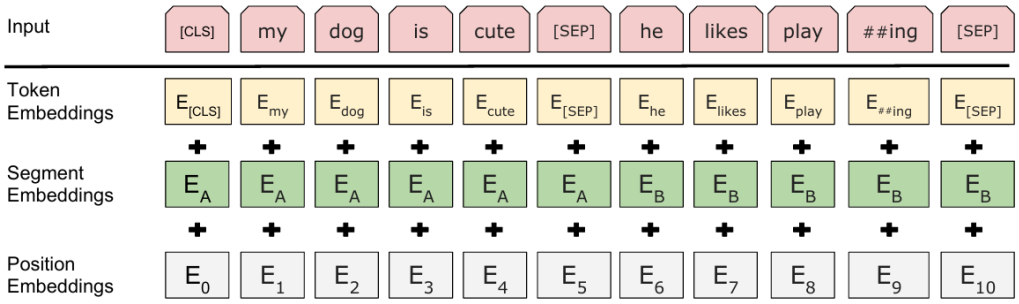
BERT menggunakan tokenisasi WordPiece, yang memecah kata menjadi sub-kata untuk menangani kata yang tidak dikenal (*out-of vocabulary*) [38]. Sebelum dimasukkan ke dalam model, setiap input akan direpresentasikan dalam bentuk kombinasi dari tiga jenis embedding, yaitu:

1. Token Embeddings: Representasi vektor dari token hasil tokenisasi. Misalnya, kata *Fintech* dapat dipecah menjadi menjadi *Fin* dan *##tech*.

2. Segment Embeddings (Token Type IDs): Digunakan untuk membedakan dua kalimat yang dijadikan pasangan dalam tugas NSP, yaitu segmen A dan segmen B

3. Position Embeddings: Memberikan informasi urutan posisi token dalam kalimat [38].

Ketiga jenis embedding tersebut dijumlahkan secara elemen-wise untuk membentuk representasi akhir setiap token. Illustrasi proses ini ditampilkan pada 2.2



Gambar 2.2. Representasi input pada BERT, terdiri dari token embeddings, segment embeddings, dan position embeddings

[38]

2.6.4 Fine-tuning BERT

Setelah proses *pre-training*, model BERT dapat disesuaikan untuk berbagai tugas pemrosesan bahasa alami melalui tahapan *fine-tuning*. Pada tahap ini, arsitektur BERT tidak diubah, melainkan ditambahkan satu layer klasifikasi sederhana diatas token khusus [CLS], kemudian seluruh parameter model dilatih ulang secara end-to-end pada dataset spesifik [38]. Dalam penelitian [38], *fine-tuning* BERT dilakukan untuk berbagai tugas seperti klasifikasi teks, penjawaban pertanyaan, dan pengenalan entitas bernama. Pada tugas klasifikasi teks seperti MNLI, model memprediksi hubungan logis antara dua kalimat. Pada tugas SQuAD, model menjawab pertanyaan berdasarkan konteks paragraf, sedangkan pada tugas NER, model menandai token dalam teks yang merupakan nama entitas seperti orang, lokasi, atau organisasi.

2.7 IndoBERT

IndoBERT adalah model BERT yang dilatih secara khusus pada korpus besar berbahasa Indonesia. Model ini dikembangkan oleh IndoNLU(2020) untuk mengatasi tantangan dalam pemrosesan bahasa alami di Indonesia, seperti struktur gramatikal yang unik dan ragam bentuk kata. IndoBERT menggunakan arsitektur BERT-Base dengan 12 lapisan transformer dan 110 juta parameter, serta dilatih pada lebih dari 200 juta kata dari berbagai sumber termasuk berita, media sosial, dan Wikipedia Indonesia [39].

2.8 IndoBERTweet

IndoBERTweet adalah model *Transformer* berbasis arsitektur BERT-Base yang dikembangkan oleh Koto, Lau, dan Baldwin pada tahun 2021 secara khusus untuk teks tweet berbahasa Indonesia [17]. Model ini diinisialisasi dari bobot IndoBERT, kemudian dilanjutkan dengan *pre-training* pada korpus tweet berbahasa Indonesia berukuran sekitar 26GB melalui tugas *Masked Language Modeling* (MLM) [17]. Kosakata IndoBERTweet juga dilatih ulang menggunakan algoritma *WordPiece* pada korpus tweet tersebut, sehingga 46% di antaranya merupakan *token* baru dibanding kosakata IndoBERT yang berfokus pada teks formal [17].

Pelatihan ulang kosakata ini memungkinkan IndoBERTweet menangani karakteristik unik dari teks tweet berbahasa Indonesia seperti singkatan, kata gaul, tagar (*hashtag*), emoji, dan tata bahasa informal yang umum dijumpai pada *platform* X. Model ini telah dilaporkan mencapai performa yang lebih baik dibandingkan IndoBERT pada berbagai tugas klasifikasi teks Twitter berbahasa Indonesia, termasuk klasifikasi sentimen dan emosi [17].

2.9 Arsitektur BiLSTM dan Gerbang Logika

Long Short-Term Memory (LSTM) adalah varian dari *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi masalah *vanishing gradient* pada data sekuensial panjang. Inti dari LSTM adalah sel memori yang diatur oleh mekanisme gerbang (*gates*) untuk mengontrol aliran informasi [40]. Gerbang pada LSTM terdiri dari:

1. Gerbang lupa (*forget gate*, f_t): menentukan informasi lama yang dibuang.
2. Gerbang masukan (*input gate*, i_t): menentukan informasi baru yang disimpan.

3. Gerbang keluaran (*output gate*, o_t): menentukan keluaran berdasarkan kondisi memori saat ini.

Setiap gerbang menggunakan fungsi aktivasi *sigmoid* (σ) yang menghasilkan nilai antara 0 dan 1 sebagai pengatur aliran informasi. Operasi LSTM pada langkah waktu t dirumuskan melalui persamaan (2.1)–(2.6), dengan x_t sebagai masukan, h_{t-1} sebagai *hidden state* sebelumnya, C_t sebagai sel memori, W dan b sebagai bobot dan bias, serta $\odot$ sebagai perkalian elemen demi elemen (*element-wise*).

Tahap pertama, *forget gate* menentukan proporsi informasi pada sel memori sebelumnya (C_{t-1}) yang perlu dipertahankan atau dibuang. Gerbang ini memetakan gabungan h_{t-1} dan x_t ke nilai antara 0 (buang sepenuhnya) dan 1 (pertahankan sepenuhnya), sebagaimana persamaan (2.1).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.1}$$

Selanjutnya, LSTM menentukan informasi baru yang akan disimpan melalui dua komponen. *Input gate* (i_t) pada persamaan (2.2) memutuskan nilai mana yang akan diperbarui, sedangkan kandidat sel memori ($\tilde{C}_t$) pada persamaan (2.3) menghasilkan nilai-nilai baru yang berpotensi ditambahkan ke sel memori menggunakan aktivasi *tanh*.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.3}$$

Sel memori kemudian diperbarui dengan menggabungkan dua aliran tersebut, yaitu informasi lama yang dipertahankan ($f_t \odot C_{t-1}$) dan informasi baru yang dipilih ($i_t \odot \tilde{C}_t$), seperti ditunjukkan pada persamaan (2.4). Mekanisme penggabungan inilah yang memungkinkan LSTM mempertahankan konteks jangka panjang tanpa mengalami *vanishing gradient*.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.4}$$

Tahap akhir, *output gate* (o_t) pada persamaan (2.5) menentukan bagian sel memori yang akan dikeluarkan. *Hidden state* (h_t) pada persamaan (2.6) kemudian dihasilkan dengan menyaring sel memori yang telah diaktivasi *tanh* melalui *output gate*. Nilai h_t inilah yang diteruskan ke langkah waktu berikutnya sekaligus menjadi keluaran pada langkah saat ini.

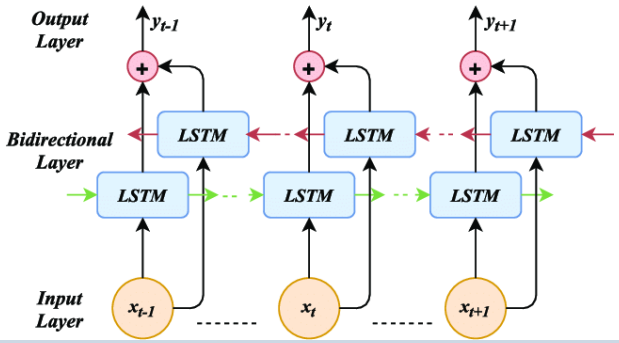
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.5}$$

$$h_t = o_t \odot \tanh(C_t) \tag{2.6}$$

LSTM standar hanya memproses urutan dalam satu arah (kiri ke kanan), sehingga konteks suatu kata hanya ditentukan oleh kata-kata sebelumnya. *Bidirectional LSTM* (BiLSTM) mengatasi keterbatasan ini dengan menggabungkan dua lapisan LSTM yang dilatih secara simultan pada arah waktu positif dan negatif [41]. Lapisan maju(*forward*) memproses urutan dari kiri ke kanan, sedangkan lapisan mundur(*backward*) memproses dari kanan ke kiri, sehingga output pada setiap langkah waktu memuat informasi dari konteks sebelum maupun sesudahnya [41]. *Hidden state* akhir pada setiap posisi diperoleh dengan menggabungkan (*concatenate*) *output* kedua arah, sebagaimana ditunjukkan pada persamaan (2.7).

$$h_t = [\overrightarrow{h}_t ; \overleftarrow{h}_t] \tag{2.7}$$

Gambar 2.3 menunjukkan arsitektur *Bidirectional LSTM* (BiLSTM) yang memproses dari dua arah untuk menangkap depedensi karakter kiri-ke-kanan dan kanan-ke-kiri.



Gambar 2.3. Arsitektur *Bidirectional LSTM* (BiLSTM).

2.10 Evaluation Metrics

Dalam Penelitian ini, untuk mengukur performa model digunakan beberapa metrik evaluasi yang umum pada tugas klasifikasi, yaitu *accuracy, precision, recall, F1-score, dan confusion matrix* [42].

Accuracy mengukur proporsi prediksi yang benar terhadap total jumlah data dan dirumuskan sebagai berikut:

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

Precision merupakan proporsi dari prediksi positif yang benar dibandingkan seluruh prediksi positif yang dibuat oleh model, dengan rumus sebagai berikut:

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

Recall, atau sensitivitas, adalah rasio antara prediksi positif yang benar terhadap seluruh data positif sebenarnya. Rumus *Recall* ditunjukkan pada persamaan berikut:

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

Sementara itu, *F1-score* merupakan rata-rata harmonik dari *Precision* dan *Recall* yang digunakan untuk memberikan keseimbangan antara keduanya. Rumus *F1-score* ditunjukkan pada persamaan berikut:

$$F1\text{-}score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.11)$$

Confusion matrix memetakan hasil prediksi model terhadap label sebenarnya ke dalam empat kategori pada klasifikasi biner. dalam penelitian ini , kelas depresi diperlakukan sebagai kelas positif dan kelas normal sebagai kelas negatif. Penamaan keempat kategori mengikuti dua komponen: kata pertama (*True/False*) menyatakan apakah prediksi model benar atau salah, sedangkan kata kedua(*Positive/Negative*) menyatakan kelas yang diprediksi model. Berdasarkan Tabel 2.1, keempat kategori tersebut didefinisikan sebagai berikut:

Tabel 2.1. Confusion Matrix untuk klasifikasi biner

		Prediksi	
		Depresi	Normal
Aktual	Depresi	TP	FN
	Normal	FP	TN

Keempat nilai tersebut menjadi dasar perhitungan metrik evaluasi seperti *accuracy*, *precision, recall,* dan *F1-score*.

2.11 Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence (XAI) merupakan subdomain dari AI yang bertujuan meningkatkan transparansi dengan menjelaskan proses pengambilan keputusan internal dari model machine learning [14]. XAI muncul sebagai respons terhadap meningkatnya kompleksitas model seperti *deep neural networks* yang berperforma tinggi namun bersifat *black-box*, sehingga menimbulkan kekhawatiran etis dan isu akuntabilitas [43].Penelitian terdahulu mendefinisikan XAI dengan model yang memberikan detail atau alasan untuk membuat cara kerjanya menjadi jelas atau mudah dipahami [27].

2.11.1 Local Interpretable Model-Agnostic Explanations(LIME)

LIME merupakan teknik XAI yang pertama kali diperkenalkan oleh Ribeiro et. al [44] dan bekerja dengan mengaproksimasi model kompleks dengan model *interpretable* sederhana secara lokal di sekitar prediksi yang ingin dijelaskan

[14, 27]. LIME melakukan perturbasi pada fitur-fitur di sekitar prediksi target dan menganalisis perubahan output model untuk memahami kontribusi setiap fitur [14]. Dalam taksonomi [27], LIME termasuk *model-agnostic post-hoc explainability* yang menyediakan *local explanation*.

Secara formal, LIME menghasilkan penjelasan $\xi(x)$ untuk sebuah *instance* x dengan mencari model *interpretable* g dari kelas model *interpretable* G yang meminimalkan *loss* lokal $\boldsymbol{L}(f, g, \pi_x)$ sekaligus menjaga kompleksitas $\Omega(g)$ tetap rendah, sebagaimana ditunjukkan pada persamaan (2.12) [44].

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \; \boldsymbol{L}(f, g, \pi_x) + \Omega(g) \tag{2.12}$$

Pada persamaan (2.12), f adalah model *black-box* yang ingin dijelaskan, sedangkan g adalah model *interpretable* sederhana (misalnya model linear) yang dipilih untuk meniru perilaku f di sekitar x. Fungsi kedekatan π_x (*proximity*) menentukan seberapa dekat setiap sampel hasil perturbasi terhadap x, sehingga sampel yang lebih dekat memperoleh bobot lebih besar dan membentuk wilayah lokal di sekitar x. Suku $\boldsymbol{L}(f, g, \pi_x)$ mengukur seberapa tidak setia (*unfaithful*) model g dalam meniru f pada wilayah lokal tersebut, sehingga semakin kecil nilainya, semakin baik g merepresentasikan f secara lokal. Adapun $\Omega(g)$ merupakan ukuran kompleksitas model g, misalnya jumlah kata atau fitur yang dilibatkan, yang dijaga tetap rendah agar penjelasan tetap sederhana dan mudah dipahami manusia. Dengan demikian, LIME pada dasarnya mencari keseimbangan antara kesetiaan lokal (*local fidelity*) terhadap model asli dan kesederhanaan penjelasan [44].