

**PENERAPAN *DEEP LEARNING* PADA MODEL
KLASIFIKASI EMOSI BERBASIS DATA *SOCIAL MEDIA*
TWITTER (X) MENGGUNAKAN METODE CNN - LSTM**

TUGAS AKHIR

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1) di
Program Studi Teknik Informatika
Fakultas Teknologi Industri, Institut Teknologi Sumatera

Oleh:

SHAKIRA PUTRI ABRAR

NIM 121140053



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2025**

DAFTAR ISI

DAFTAR ISI	ii
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
DAFTAR RUMUS	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Batasan Masalah	6
1.5 Manfaat Penelitian	6
1.6 Sistematika Penulisan	7
1.6.1 Bab I Pendahuluan	7
1.6.2 Bab II Tinjauan Pustaka.....	7
1.6.3 Bab III Metode Penelitian	8
1.6.4 Bab IV Hasil dan Pembahasan.....	8
1.6.5 Bab V Kesimpulan dan Saran	8
BAB IV TINJAUAN PUSTAKA	9
2.1 Tinjauan Pustaka	9

2.2	Dasar Teori.....	18
2.2.1	Emosi	18
2.2.2	Media Sosial.....	18
2.2.2.1	<i>Twitter (X)</i>	18
2.2.3	Prapemrosesan Data	19
2.2.3.1	<i>Tokenization</i>	19
2.2.3.2	<i>Case Folding</i>	19
2.2.3.3	<i>Text Cleaning</i>	20
2.2.3.4	<i>Spell Normalization</i>	20
2.2.3.5	<i>Stopword Removal</i>	20
2.2.3.6	<i>Stemming</i>	21
2.2.4	<i>Deep Learning</i>	21
2.2.3.1	<i>Word2Vec</i>	22
2.2.3.2	<i>Convolutional Neural Network (CNN)</i>	24
2.2.3.3	<i>Long Short-Term Memory (LSTM)</i>	26
2.2.3.4	<i>CNN-LSTM</i>	31
2.2.3.3	<i>Hyperparameter Tuning</i>	32
2.2.5	<i>Confusion Matrix</i>	35
BAB III	METODE PENELITIAN.....	41
3.1	Alur Penelitian	41
3.2	Penjabaran Langkah Penelitian	42
3.2.1	Identifikasi Masalah.....	42

3.2.2	Studi Literatur	43
3.2.3	Pengumpulan Data	43
3.2.4	Prapemrosesan Data	44
3.2.5	<i>Word Embedding</i>	44
3.2.6	<i>Modelling</i> (CNN-LSTM)	45
3.2.7	Evaluasi Model	46
3.3	Alat dan Bahan Tugas Akhir.....	46
3.3.1	Alat.....	46
3.3.2	Bahan	47
3.4	Metode Pengembangan	47
3.5	Penjabaran Langkah Penelitian	48
3.5.1	Data Collecting	48
3.5.2	Validasi <i>Dataset</i>	49
3.5.3	Prapemrosesan Data	49
3.5.4	Pembagian Data	62
3.5.5	Arsitektur Model CNN-LSTM.....	63
3.6	Ilustrasi Perhitungan Metode	65
3.6.1	<i>Word2Vec (Skip-Gram)</i>	65
3.6.2	CNN <i>Layer</i>	65
3.6.3	LSTM <i>Layer</i>	68

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Word2Vec CBOW dan Skip-Gram	22
Gambar 2.2 Arsitektur model CNN untuk kalimat	24
Gambar 2.3 Arsitektur LSTM.....	27
Gambar 2.4 Forget Gate Layer	28
Gambar 2.5 Input Gate Layer	29
Gambar 2.6 Output Gate Layer	30
Gambar 3.1 Alur Penelitian	41
Gambar 3.2 Metode Pengembangan.....	47
Gambar 3.4 Split Data	62
Gambar 3.5 Inisialisasi Nilai Aktual dan Prediksi	76

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu.....	12
Tabel 2.2 Confusion Matrix untuk Klasifikasi Biner.....	36
Tabel 2.3 Confusion Matrix untuk Klasifikasi Multiclass	36
Tabel 3.1 Hyperparameter dalam CNN-LSTM	33
Tabel 3.2 Jumlah data setiap label.....	43
Tabel 3.3 Sampel Dataset Awal	48
Tabel 3.4 Tahapan Tokenizing	54
Tabel 3.5 Tahapan <i>Case Folding</i>	49
Tabel 3.6 Tahapan Text Cleaning.....	51
Tabel 3.7 Tahapan Spell Normalization	55
Tabel 3.8 Tahapan Stopword Removal	57
Tabel 3.9 Tahapan Stemming.....	58
Tabel 3.10 Ilustrasi Inisialisasi Data Training Skip-gram.....	61
Tabel 3.11 Inisialisasi Nilai Bobot dan Bias.....	68
Tabel 3.12 Inisialisasi Nilai Aktual dan Prediksi.....	77

DAFTAR RUMUS

(2.1)	23
(2.2)	24
(2.3)	25
(2.4)	26
(2.5)	28
(2.6)	29
(2.7)	29
(2.8)	30
(2. 9)	31
(2. 10)	37
(2. 11)	38
(2. 12)	38
(2. 13)	38
(2. 14)	39
(2. 15)	39
(2. 16)	40
(2. 17)	40

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Teknologi telah memfasilitasi pertumbuhan media sosial, menumbuhkan kreativitas, dan menyediakan kesempatan bagi *microblogging* sosial untuk mengekspresikan perasaan, emosi, maupun pendapat mereka setiap saat. Media sosial tidak hanya berfungsi sebagai *platform* untuk berbagi konten *multimedia*, membangun jaringan, dan memperoleh pengetahuan, tetapi juga menjadi wadah bagi penggunanya untuk mengekspresikan emosi dan perasaan terkait masalah yang mereka hadapi [1]. Salah satu *platform* yang paling banyak digunakan di Indonesia adalah *Twitter*. *Twitter* merupakan *platform* yang digunakan pengguna untuk mengekspresikan diri. *Platform* media sosial ini memungkinkan kita untuk memperoleh berbagai informasi dari postingan (*tweet*) pengguna *Twitter* secara *realtime* [2]. Teks dalam *tweet* tidak hanya menyampaikan informasi, tetapi juga merefleksikan perilaku dan ekspresi emosional penggunanya, sehingga dapat dijadikan sebagai studi kasus untuk menilai emosi pengguna *Twitter* [3][4].

Penyampaian emosi dalam bahasa dapat diwujudkan melalui penggunaan kata, frasa, kalimat, maupun paragraf. Menurut Kamus Besar Bahasa Indonesia (KBBI), emosi merupakan kondisi yang melibatkan reaksi psikologis dan fisiologis, seperti perasaan senang, sedih, terharu, maupun rasa cinta. Keadaan emosi seseorang memainkan peran penting dalam proses interaksi antarindividu [5]. Model struktur emosi dalam leksikon bahasa Indonesia dibagi menjadi tiga tingkatan, yaitu: tingkat *superordinate*, tingkat *basic*, dan tingkat *subordinate*. Pada tingkat

superordinate, emosi diklasifikasikan ke dalam dua kategori utama, yaitu emosi positif dan emosi negatif. Pada tingkat *basic*, terdapat lima jenis emosi. Emosi positif mencakup dua jenis, yaitu cinta dan senang, sedangkan emosi negatif mencakup tiga jenis, yaitu marah, takut, dan sedih. Pada tingkat *subordinate*, emosi positif dibagi lagi menjadi 13 kelompok, sedangkan emosi negatif terdiri atas 18 kelompok [6].

Pola emosi negatif yang diekspresikan melalui media sosial sering kali dikaitkan dengan meningkatnya risiko gangguan psikologis seperti stres dan depresi pada pengguna. Depresi dicirikan oleh perasaan sedih, mudah tersinggung, kekosongan emosional, kesulitan berkonsentrasi, serta munculnya pikiran untuk mengakhiri hidup [7]. Klasifikasi emosi bertujuan untuk memahami serta menganalisis respons emosional individu dalam konteks tertentu [8]. Proses validasi *dataset* untuk penelitian ini, dilakukan oleh seorang ahli di bidang psikologi, yaitu Ibu Elsy Julianita, S.Psi., M.Psi., Psikolog yang merupakan dosen dan praktisi psikologi klinis di Universitas Malahayati.

Penerapan teknologi ini mencakup berbagai bidang, seperti interaksi manusia dan komputer, analisis sentimen, hingga pemantauan kesehatan mental. Kondisi ini mendorong pentingnya pengembangan sistem klasifikasi emosi secara otomatis melalui analisis teks, yang menjadi aspek krusial dalam penelitian berbasis data teks dan opini publik, terutama dalam memahami kondisi emosional pengguna di ranah digital. Analisis sentimen membutuhkan dukungan algoritma yang tepat, di mana pendekatan seperti *deep learning* menjadi salah satu metode yang efektif untuk menangani kompleksitas data teks [9].

Deep learning merupakan metode khusus dalam pembelajaran mesin yang memanfaatkan jaringan saraf berlapis-lapis untuk mempelajari

data secara bertahap. Pendekatan ini meniru cara kerja otak manusia, sehingga memungkinkan komputer untuk dilatih dalam memahami konsep abstrak dan menyelesaikan permasalahan yang tidak terdefinisi dengan jelas [10]. *Deep learning*, khususnya dalam bidang *Natural Language Processing* (NLP), menunjukkan performa yang unggul dalam memahami konteks dan makna dalam teks. *Natural Language Processing* (NLP) merupakan cabang dari kecerdasan buatan yang berfokus pada pemahaman interaksi kompleks antara komputer dan bahasa manusia [11]. Tujuan utamanya adalah untuk memungkinkan komputer memahami, menganalisis, dan menghasilkan teks secara alami, layaknya kemampuan manusia dalam berbahasa.

Sebelum data teks dari media sosial dapat diproses oleh model pembelajaran mesin, diperlukan tahap transformasi teks ke dalam bentuk numerik yang dapat dipahami oleh komputer. Salah satu teknik representasi teks yang banyak digunakan adalah *word embedding*, yaitu cara merepresentasikan kata ke dalam bentuk vektor berdimensi tetap. Penelitian ini menggunakan metode *Word2Vec* sebagai teknik *word embedding*. *Word2Vec* mampu memetakan kata-kata ke dalam ruang vektor berdasarkan konteks kemunculannya, sehingga kata-kata yang memiliki makna serupa akan memiliki representasi vektor yang saling berdekatan. Keunggulan *Word2Vec* terletak pada kemampuannya menangkap makna semantik dan hubungan antar kata dalam kalimat yang sangat penting untuk memahami emosi dalam data teks pendek seperti *tweet* [12].

Kebutuhan akan pendekatan teknologi yang tepat dalam mengolah data teks dari media sosial menjadi semakin penting seiring meningkatnya volume dan kompleksitas data. Proses klasifikasi yang dilakukan secara

manual membutuhkan waktu yang lama, sehingga hal ini mendorong untuk mengembangkan metode otomatis yang dapat digunakan untuk klasifikasi emosi menggunakan kombinasi metode yakni, *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM), yang umum digunakan dalam berbagai masalah yang berkaitan dengan *deep learning* [13]. Sejumlah penelitian terdahulu telah dilakukan untuk menilai performa metode-metode tersebut dalam konteks data berbahasa Indonesia, khususnya yang bersumber dari media sosial seperti Twitter.

Kustiwa, Akbar, dan Pinandito (2017) melakukan studi perbandingan antara dua model *deep learning*, yaitu CNN-LSTM dan LSTM, dalam mendekripsi emosi dari teks berbahasa Indonesia yang berasal dari media sosial Twitter. Dalam penelitian ini, metode *K-Fold Cross Validation* digunakan untuk mengukur konsistensi performa model dengan fitur yang diekstraksi menggunakan *FastText*. Hasilnya menunjukkan bahwa CNN-LSTM secara statistik memiliki nilai *loss* yang lebih rendah dibanding LSTM, meskipun tidak terdapat perbedaan signifikan pada metrik akurasi, *precision*, *recall*, dan F1-score [14].

Kamarula dan Rochmawati (2022) membandingkan efektivitas model CNN dan Bi-LSTM untuk analisis sentimen dan emosi masyarakat selama pandemi Covid-19 di Twitter. Dengan memanfaatkan representasi kata menggunakan *Word2Vec*, mereka menemukan bahwa Bi-LSTM memberikan akurasi yang lebih tinggi baik untuk klasifikasi sentimen (69,48%) maupun emosi (84,36%), dibandingkan CNN yang memperoleh akurasi sedikit lebih rendah [15].

Metode CNN lebih unggul dalam mengekstraksi fitur terletak pada keberadaan *convolution layer*, yang memungkinkan proses ekstraksi fitur menjadi lebih optimal dan efisien [16]. LSTM merupakan metode yang

lebih unggul dibandingkan metode konvensional, karena kemampuannya dalam menangani dependensi jangka panjang. Hal ini menjadikannya sangat sesuai untuk diterapkan dalam analisis sentimen [17]. Menggabungkan CNN dan LSTM memungkinkan pemanfaatan kekuatan masing-masing metode. Kombinasi kedua metode tersebut diharapkan mampu meningkatkan kinerja dan performa model dalam mengenali teks *tweet* untuk mengklasifikasikan emosi pada pengguna *Twitter*. Hal ini membuat model lebih *robust* dalam mengklasifikasikan nuansa halus dan akan bekerja secara sinergis untuk meningkatkan akurasi dalam identifikasi teks. Evaluasi performa model dalam penelitian ini dilakukan menggunakan metrik evaluasi yang diperoleh dari hasil *confusion matrix*, yaitu akurasi, presisi, *recall*, dan *f1-score*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan sebelumnya, maka didapatkan rumusan masalah sebagai berikut.

1. Bagaimana cara mengklasifikasikan emosi pada teks *tweet* di *platform Twitter* (X) menggunakan metode gabungan dari *Convolutional Neural Network* dan *Long Short Term Memory* (CNN-LSTM)?
2. Bagaimana komparasi performa metode CNN, LSTM, dan kombinasi CNN-LSTM dalam mengklasifikasikan emosi pada *tweet* di *platform Twitter* (X)?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut.

1. Mengimplementasikan metode *Convolutional Neural Network* dan *Long Short Term Memory* (CNN-LSTM) untuk klasifikasi emosi pada teks *tweet* yang menunjukkan indikasi emosi pada *platform Twitter* (X).
2. Menganalisis performa model terbaik dengan menggunakan metrik evaluasi berdasarkan hasil *confusion matrix* untuk menilai efektivitas klasifikasi teks *tweet* yang menunjukkan indikasi emosi pada *platform Twitter* (X).

1.4 Batasan Masalah

Batasan masalah dari penelitian ini dilakukan agar tidak terlalu luas dan dapat dengan mudah dipahami, maka diberikan batasan masalah berikut.

1. Penelitian ini difokuskan pada pengklasifikasian teks *tweet* bahasa Indonesia yang berkaitan dengan emosi.
2. Dataset yang digunakan berasal dari penelitian sebelumnya [18] yang dapat diakses melalui situs *Kaggle* dan terdiri dari 4.403 baris *tweet* yang memiliki lima label kelas, yaitu marah, takut, gembira, cinta, dan sedih yang disediakan dalam bentuk format csv.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut.

1. Dapat menerapkan pemodelan CNN-LSTM untuk pengklasifikasian emosi berbasis data *social media Twitter* (X).
2. Dapat mengukur tingkat performa menggunakan metrik evaluasi untuk pengklasifikasian emosi berbasis data *social media Twitter* (X).

1.6 Sistematika Penulisan

Sistematika penulisan pada laporan tugas akhir ini terbagi menjadi beberapa Bab yang disusun secara sistematis untuk memudahkan pemahaman. Adapun rincian sistematika penulisan laporan ini adalah sebagai berikut.

1.6.1 Bab I Pendahuluan

Bab Pendahuluan memberikan gambaran umum dan urgensi mengenai penelitian yang dilakukan yaitu klasifikasi emosi di *platform* media sosial, khususnya *Twitter* (*X*). Rumusan masalah merumuskan pertanyaan yang ingin dijawab. Tujuan utama dari penelitian ini yaitu, untuk mengembangkan model yang dapat secara akurat mengklasifikasikan emosi dalam teks. Manfaat menjelaskan kontribusi penelitian ini bagi pengembangan teknologi kesehatan mental dan pemantauan kondisi psikologis di media sosial. Batasan masalah menetapkan ruang lingkup penelitian, seperti jenis data yang digunakan dan emosi yang diidentifikasi. Sistematika penulisan memberikan gambaran umum tentang struktur laporan, menjelaskan isi dari setiap bab yang akan dibahas.

1.6.2 Bab II Tinjauan Pustaka

Bab Tinjauan Pustaka menyajikan kajian teori dan penelitian terdahulu yang relevan dengan topik klasifikasi emosi. Teori-teori yang relevan menguraikan konsep-konsep dasar tentang emosi, serta teknik-teknik *Natural Language Processing* (NLP) yang digunakan dalam analisis teks. Penelitian terdahulu merangkum studi-studi sebelumnya

yang telah dilakukan dalam bidang klasifikasi emosi dan kesehatan mental di media sosial.

1.6.3 Bab III Metode Penelitian

Bab Metode Penelitian menjelaskan proses pengerjaan penelitian yang mencakup lima bagian yaitu alur penelitian, penjabaran langkah penelitian, alat dan bahan tugas akhir, metode pengembangan, ilustrasi perhitungan metode dan rancangan pengujian.

1.6.4 Bab IV Hasil dan Pembahasan

Bab ini menyajikan hasil penelitian yang telah dilakukan serta mencakup pembahasan dan analisis mendalam mengenai metode penelitian yang diterapkan dalam pengembangan model.

1.6.5 Bab V Kesimpulan dan Saran

Bab ini memuat penjelasan mengenai kesimpulan yang diperoleh dari hasil penelitian yang telah dilakukan. Selain itu, bab ini juga menyajikan saran dari penulis yang dapat menjadi pertimbangan untuk pengembangan penelitian di masa mendatang.

BAB IV

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian ini mengacu serta menggunakan beberapa jurnal penelitian dan berbagai sumber informasi yang didapatkan dari internet untuk mendukung proses riset dan penulisan tugas akhir. Tinjauan pustaka akan mencakup berbagai informasi dan teori yang relevan dengan penelitian serta penulisan tugas akhir ini. Kustiwa, Akbar, dan Pinandito (2017) melakukan studi perbandingan antara dua model *deep learning*, yaitu CNN-LSTM dan LSTM, dalam mengklasifikasikan emosi dari teks berbahasa Indonesia yang berasal dari media sosial Twitter. Dalam penelitian ini, metode *K-Fold Cross Validation* digunakan untuk mengukur konsistensi performa model dengan fitur yang diekstraksi menggunakan *FastText*. Hasilnya menunjukkan bahwa CNN-LSTM secara statistik memiliki nilai *loss* yang lebih rendah dibanding LSTM, meskipun tidak terdapat perbedaan signifikan pada metrik akurasi, *precision*, *recall*, dan F1-score [14].

Kamarula dan Rochmawati (2022) membandingkan efektivitas model CNN dan Bi-LSTM untuk analisis sentimen dan emosi masyarakat selama pandemi Covid-19 di Twitter. Dengan memanfaatkan representasi kata menggunakan *Word2Vec*, mereka menemukan bahwa Bi-LSTM memberikan akurasi yang lebih tinggi baik untuk klasifikasi sentimen (69,48%) maupun emosi (84,36%), dibandingkan CNN yang memperoleh akurasi sedikit lebih rendah [15].

Penelitian oleh Bhat (2024) membahas tantangan dalam klasifikasi emosi pada teks pendek berbahasa Inggris yang disebabkan oleh

keterbatasan konteks dan kompleksitas semantik. Metode yang digunakan meliputi pretrained model seperti DistilBERT, BERT, dan Support Vector Machine (SVM). Hasil penelitian menunjukkan bahwa model DistilBERT menghasilkan performa terbaik dengan akurasi sebesar 77% dan skor macro-F1 sebesar 73% pada dataset SmallEnglishEmotions, serta menunjukkan kinerja yang lebih unggul dibandingkan dengan klasifikasi pada teks yang lebih panjang [19].

Penelitian oleh Saputra, Saragi, dan Ronaldo (2025) menggunakan model *IndoBERT* untuk melakukan klasifikasi emosi pada teks berbahasa Indonesia. Penelitian ini ditujukan untuk mengatasi kendala terbatasnya *dataset* dan model yang dioptimalkan untuk konteks bahasa Indonesia. Hasil evaluasi menunjukkan bahwa model *IndoBERT* mampu mencapai akurasi sebesar 73%, dengan performa tertinggi ditemukan pada kategori emosi jijik yang memperoleh *precision* dan *recall* sebesar 0,98. Meskipun demikian, model masih mengalami kesulitan dalam membedakan emosi marah dan sedih [20].

Andini, dkk (2024) mengembangkan sistem pendekripsi ujaran kebencian dan emosi berbasis CNN yang dilatih menggunakan data dari Twitter. Mereka menerapkan kombinasi metode TF-IDF dan *Word2Vec* untuk representasi fitur kata. Model CNN yang dibangun berhasil mencapai tingkat akurasi sebesar 87% untuk deteksi emosi dan 99% untuk pendekripsi ujaran kebencian, menunjukkan potensi besar dalam aplikasi moderasi konten media sosial secara *real-time* [21].

Sementara itu, Harsemadi, Dharmendra, dan Wijaya (2023) memfokuskan penelitian mereka pada klasifikasi emosi dalam *tweet* berbahasa Indonesia yang memiliki distribusi kelas tidak seimbang. Dengan memanfaatkan teknik *Edited Nearest Neighbours* (ENN) sebagai

metode *sampling* dan menggunakan beberapa algoritma seperti *Random Forest*, *Neural Network*, dan *Support Vector Machine* (SVM), mereka menemukan bahwa *Random Forest* memberikan akurasi terbaik sebesar 62,55%. Di sisi lain, *Neural Network* menunjukkan nilai presisi dan F1-score tertinggi, sedangkan SVM unggul pada presisi namun rendah pada *recall* [22]. Berikut pada Tabel 2.1 merupakan beberapa penelitian terdahulu yang menjadi acuan pada tugas akhir ini.

Terakhir penelitian oleh Islan, dkk. (2025) membahas deteksi pola tidur tidak teratur (ISP) melalui data media sosial dengan model hybrid BiLSTM-CNN-Attention. Menggunakan fitur dari WordNet dan BERT, model ini mencapai akurasi 91% dan F1-score 92%. Kata-kata seperti "anger", "sad", dan "future" dominan pada pengguna dengan ISP [23].

Tabel 2.1 Penelitian Terdahulu

No.	Peneliti	Judul Penelitian	Permasalahan	Metode Klasifikasi	Ekstraksi Fitur	Hasil Penelitian
1.	Alwan Fauzul Azhim Kustiwa, Muhammad Aminul Akbar, Aryo Pinandito (2017)	Studi Perbandingan pada Metode CNN-LSTM dan LSTM dalam Mendeteksi Emosi pada Data Teks Berbahasa Indonesia pada Media Sosial Twitter [14].	Perbandingan performa antara CNN-LSTM dan LSTM dalam mendeteksi emosi pada teks bahasa Indonesia di Twitter	-K-Fold Cross Validation - CNN-LSTM - LSTM	-FastText	-CNN-LSTM memiliki loss lebih rendah secara signifikan dibanding LSTM, namun tidak ada perbedaan signifikan pada metrik akurasi, <i>precision</i> , <i>recall</i> , dan F1-score.
2.	Mohamad Romli	Perbandingan CNN dan Bi-LSTM pada	Menentukan model terbaik untuk	- CNN - Bi-LSTM	-Word2Vec	Bi-LSTM menghasilkan akurasi

No.	Peneliti	Judul Penelitian	Permasalahan	Metode Klasifikasi	Ekstraksi Fitur	Hasil Penelitian
	Firdaus Kamarula, Naim Rochmawati (2022)	Analisis Sentimen dan Emosi Masyarakat Indonesia di Media Sosial Twitter Selama Pandemik Covid-19 yang Menggunakan Metode Word2Vec [15].	analisis sentimen dan emosi masyarakat selama pandemi Covid-19 di Twitter			lebih tinggi yaitu 69,48% (sentimen) dan 84,36% (emosi), lebih unggul dibanding CNN dengan hasil 68,58% (sentimen) dan 84,21% (emosi).
3.	Siddhanth Bhat (2024)	<i>Emotion Classification in</i>	Kesulitan dalam mengklasifikasikan	- <i>Pretrained DistilBERT</i>		Model DistilBERT memberikan akurasi

No.	Peneliti	Judul Penelitian	Permasalahan	Metode Klasifikasi	Ekstraksi Fitur	Hasil Penelitian
		<i>Short English Texts using Deep Learning Techniques [19].</i>	emosi pada teks pendek berbahasa Inggris karena keterbatasan konteks dan kompleksitas semantik.	-BERT -Support Vector Machine (SVM)		tertinggi (77%) dan <i>macro-F1</i> (73%) pada <i>dataset SmallEnglishEmotions</i> ; performa lebih baik daripada teks panjang.
4.	Ade Chandra Saputra, Agus Sehatan Saragi,	Prediksi Emosi dalam Teks Bahasa Indonesia Menggunakan Model IndoBERT	Keterbatasan <i>dataset</i> dan model yang dioptimalkan untuk Bahasa Indonesia dalam	- <i>IndoBert</i>		Akurasi sebesar 73%, dengan performa terbaik pada emosi jijik (<i>Precision</i> dan <i>Recall</i> 0.98). Emosi seperti

No.	Peneliti	Judul Penelitian	Permasalahan	Metode Klasifikasi	Ekstraksi Fitur	Hasil Penelitian
	Deddy Ronaldo (2025)	[20].	tugas prediksi emosi.			marah dan sedih masih sulit dibedakan.
5.	Nanda Mujahidah Andini, Yulian Findawati, Ika Ratna Indra Astutik, Ade Eviyanti	Implementasi <i>Convolutional Neural Network</i> (CNN) Untuk Mendeteksi Ujaran Kebencian Dan Emosi di Twitter [21].	Penyebaran ujaran kebencian dan kesulitan mendeteksi emosi dan <i>hate speech</i> secara akurat di media sosial (Twitter)	- CNN	- TF-IDF - Word2Vec	Akurasi model mencapai 87% untuk deteksi emosi dan 99% untuk deteksi ujaran kebencian, efektif dalam sistem moderasi konten.

No.	Peneliti	Judul Penelitian	Permasalahan	Metode Klasifikasi	Ekstraksi Fitur	Hasil Penelitian
	(2024)					
6.	I Gede Harsemadi, I Komang Dharmendra, I Made Pasek Pradnyana Wijaya (2023) [15]	Klasifikasi Emosi pada Berbahasa Indonesia Menggunakan Teknik <i>Sampling</i> ENN [22].	Ketidakseimbangan data pada kelas emosi dalam <i>tweet</i> Indonesia menyulitkan klasifikasi emosi yang akurat.	- <i>Edited Nearest Neighbours</i> (ENN) - <i>Random Forest</i> - <i>Neural Network</i> - <i>Support Vector Machine</i>	-TF-IDF	Model <i>Random Forest</i> memiliki akurasi tertinggi (62.55%), sedangkan <i>Neural Network</i> menunjukkan presisi dan F1-score terbaik (67.23% dan 62.57%). SVM memiliki presisi tinggi tapi <i>recall</i> rendah.

No.	Peneliti	Judul Penelitian	Permasalahan	Metode Klasifikasi	Ekstraksi Fitur	Hasil Penelitian
				(SVM)		
7.	Mohammed Jawwadul Islam, dkk. (2025)	<i>Irregular Sleep Pattern Identification and Analysis from Social Media Dataset Using Hybrid Deep Learning Based Attention Mechanism</i> [23].	Minimnya penelitian yang memanfaatkan data media sosial untuk mendeteksi pola tidur tidak teratur (<i>Irregular Sleeping Pattern/ISP</i>), serta keterbatasan alat pendekripsi berbasis data publik.	-Kombinasi BiLSTM+CN N+Attention	-WordNet - Kontekstua l BERT	Model <i>hybrid CNN-BiLSTM-Attention</i> mencapai akurasi 91%, F1-score 92%, dan MCC 0.82. Kata seperti "anger", "sad", dan "future" dominan pada pengguna dengan ISP.

2.2 Dasar Teori

Penelitian ini didasarkan pada beberapa landasan teori yang berperan sebagai acuan dalam proses analisis dan pengembangan. Berikut adalah beberapa dasar teori yang digunakan sebagai acuan dalam pelaksanaan penelitian ini.

2.2.1 Emosi

Emosi merupakan kondisi mental yang kompleks dan dipengaruhi oleh berbagai hal, seperti kejadian di lingkungan sekitar, perubahan fisiologis dalam tubuh, maupun interaksi sosial. Emosi ini dapat memicu berbagai respons, mulai dari perubahan suasana hati, perasaan, cara berpikir, hingga perilaku seseorang [22]. Emosi umumnya muncul sebagai reaksi terhadap suatu kejadian, baik yang berasal dari dalam diri maupun dari lingkungan luar, yang dipersepsikan memiliki nilai positif atau negatif bagi individu [24].

2.2.2 Media Sosial

Media sosial telah berkembang menjadi salah satu sarana komunikasi yang paling populer di kalangan masyarakat. Media sosial adalah *platform digital* yang memungkinkan individu untuk berkomunikasi, membagikan konten, dan menjalin hubungan dengan orang lain melalui jaringan internet [25].

2.2.2.1 Twitter (X)

Twitter, yang kini dikenal sebagai X, adalah *platform* media sosial yang memungkinkan pengguna untuk berbagi pesan singkat (*tweet*) dengan batasan karakter tertentu. *Platform* ini telah menjadi salah satu

sarana komunikasi dan ekspresi diri yang populer, di mana pengguna dapat membagikan pikiran, perasaan, pengalaman, dan opini mereka secara *real-time* [26]. Pengguna dapat membagikan pesan singkat dalam bentuk *tweet*, yang akan ditampilkan di halaman profil mereka dan dapat dilihat oleh para pengikutnya [22].

2.2.3 Prapemrosesan Data

Praproses data ini mencakup berbagai teknik untuk meningkatkan kualitas data agar lebih sesuai dalam proses analisis dan pelatihan model.

2.2.3.1 Tokenization

Tokenization tahap pemecahan teks menjadi bagian-bagian paling kecil seperti kata, frasa, atau kalimat. *Tokenization* merujuk pada langkah awal dalam pemrosesan bahasa alami yang bertujuan memisahkan deretan karakter menjadi satuan kata yang teridentifikasi [27]. Tahapan ini memiliki peran krusial dalam membantu sistem komputer mengenali struktur serta makna dari teks yang dianalisis [28].

2.2.3.2 Case Folding

Case folding adalah proses yang bertujuan untuk menyamakan format huruf dalam sebuah dokumen teks dengan cara mengubah semua karakter yang ditulis dalam huruf besar (*uppercase*) menjadi huruf kecil (*lowercase*) [29]. Langkah ini penting dalam pengolahan teks karena membantu mengurangi variasi yang tidak perlu dalam data, sehingga memudahkan analisis dan pencocokan kata. Penerapan *case folding* memungkinkan tercapainya konsistensi dalam representasi teks dapat

dicapai, yang pada gilirannya meningkatkan akurasi dalam pemrosesan lebih lanjut.

2.2.3.3 *Text Cleaning*

Text Cleaning adalah proses yang dilakukan untuk membersihkan dan memvalidasi kata-kata yang tidak diinginkan, dengan tujuan mengurangi gangguan yang dapat menghambat proses klasifikasi. Dalam tahap ini, kata-kata atau karakter yang dianggap tidak relevan atau mengganggu, seperti simbol, tanda baca, emoji, dan *hastag* akan dihapus dari teks [30]. Dengan melakukan *cleaning*, kualitas data yang digunakan dalam analisis dapat ditingkatkan, sehingga hasil klasifikasi menjadi lebih akurat dan efektif.

2.2.3.4 *Spell Normalization*

Spell Normalization dilakukan penyesuaian dan perbaikan tata bahasa pada kata-kata. Kata-kata yang tidak baku, termasuk singkatan, bahasa gaul, dan *slang* akan diubah menjadi bentuk yang sesuai dengan kaidah penulisan Bahasa Indonesia yang baik dan benar, sesuai dengan pedoman yang terdapat dalam Kamus Besar Bahasa Indonesia (KBBI) [31]. Proses ini bertujuan untuk memastikan bahwa teks yang dihasilkan memenuhi standar bahasa yang diakui, sehingga lebih mudah dipahami dan dianalisis.

2.2.3.5 *Stopword Removal*

Stopword removal bertujuan untuk menghapus kata-kata umum yang sering muncul dalam jumlah besar dan dianggap tidak memiliki makna signifikan [32]. Kata-kata yang sering muncul namun tidak penting

dan relevan, seperti kata hubung (konjungsi), kata kepunyaan, dan kata ganti orang, akan dihilangkan [31]. Dengan melakukan penghapusan ini, diharapkan analisis yang dilakukan dapat lebih fokus pada kata-kata yang memiliki makna dan relevansi, sehingga meningkatkan kualitas hasil yang diperoleh.

2.2.3.6 *Stemming*

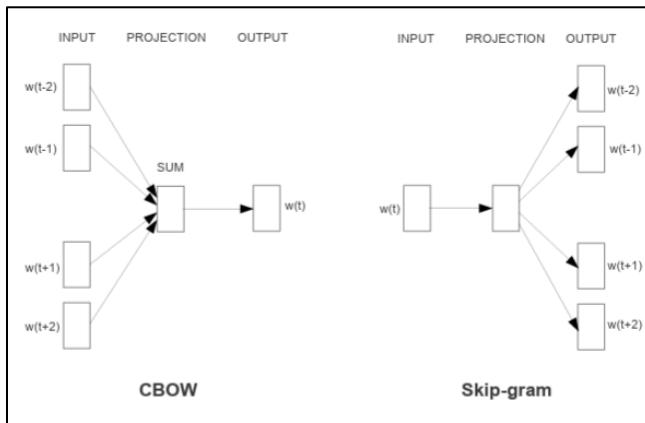
Stemming melibatkan pengubahan kata-kata menjadi bentuk kata dasarnya dengan cara menghapus imbuhan yang terdapat pada kata tersebut. Transformasi ini dilakukan dengan menghilangkan awalan dan akhiran, sehingga kata-kata yang memiliki makna serupa dapat dianggap sebagai satu kesatuan [31]. Tujuan dari langkah ini adalah untuk menyederhanakan analisis, sehingga kata-kata yang semakna dapat dikelompokkan bersama, meningkatkan konsistensi dan akurasi dalam pemrosesan data [11].

2.2.4 *Deep Learning*

Deep Learning merupakan salah satu cabang dari machine learning yang berfokus pada proses pembelajaran dari data yang sudah ada dengan memanfaatkan algoritma yang tersedia [33]. *Deep Learning* adalah suatu pendekatan pembelajaran yang melibatkan berbagai lapisan representasi, yang dihasilkan melalui penggabungan modul-modul sederhana namun *non-linier*. Setiap modul ini berfungsi untuk mengubah representasi pada satu tingkat menjadi representasi yang lebih kompleks pada tingkat yang lebih tinggi [34].

2.2.3.1 Word2Vec

Word2Vec memanfaatkan prinsip kerja jaringan saraf tiruan untuk membentuk representasi fitur kata yang didasarkan pada kemiripan makna atau semantik antar kata. *Word2Vec* berfungsi untuk memetakan setiap kata dalam teks ke dalam ruang vektor, di mana kata-kata dengan makna serupa atau yang sering muncul dalam konteks yang sama akan memiliki representasi vektor yang saling berdekatan. Arsitektur *Word2Vec* terbagi menjadi dua model utama, yaitu *Continuous Bag of Words* (CBOW) dan *Skip-gram*.



Gambar 2.1 Arsitektur *Word2Vec* CBOW dan *Skip-Gram*

CBOW dan *Skip-Gram* merupakan dua algoritma yang termasuk dalam metode *Word2Vec*. Perbedaan utama antara keduanya dapat dilihat pada Gambar 2.1. Pada algoritma CBOW, sejumlah kata konteks di sekitar kata target digunakan sebagai *input* untuk memprediksi kata target tersebut. CBOW cenderung memiliki waktu pelatihan yang lebih singkat dan mampu memberikan akurasi yang sedikit lebih tinggi pada kata-kata yang sering muncul dalam korpus [35]

Sebaliknya, pada metode *Skip-Gram*, satu kata konteks digunakan sebagai *input* untuk memprediksi kata target atau kata-kata yang berada di sekitarnya sebagai *output* [36]. Karakteristiknya yang mempelajari pola hubungan antara satu kata dan banyak konteks memungkinkan *Skip-Gram* untuk membangun hubungan yang lebih mendalam antar kata [37].

Word2vec memungkinkan pembuatan vektor yang merepresentasikan kata-kata dalam bentuk numerik. Gambar 2.1 menyajikan arsitektur *Word2Vec* secara lebih terperinci. Model ini terdiri atas tiga lapisan utama, yaitu *input layer*, *hidden layer* (sering disebut sebagai *projection layer*), dan *output layer*. *Input layer* berisi representasi *One Hot Encoding* dari kata input, yang dilambangkan dengan variabel v . Kata *input* tersebut disimbolkan dengan $w(t)$. *Hidden layer* berfungsi sebagai tempat berlangsungnya proses pembobotan atau pemetaan terhadap kata. Sementara itu, *output layer* menghasilkan representasi vector dari kata yang memiliki hubungan makna atau semantik dengan kata *input*, berdasarkan hasil pembobotan yang dilakukan pada *hidden layer*. Beberapa rumus yang digunakan dalam pembobotan kata dengan algoritma *Skip-gram* sebagai berikut.

1. Rumus perhitungan *hidden layer*

$$oh = w_1^T \cdot x \quad (2.1)$$

Keterangan:

h = *hidden layer*

w_1^T = transpose w_1

x = *input vector*

2. Rumus menghitung *output vector*

$$o = w_2^T \cdot h \quad (2.2)$$

Keterangan:

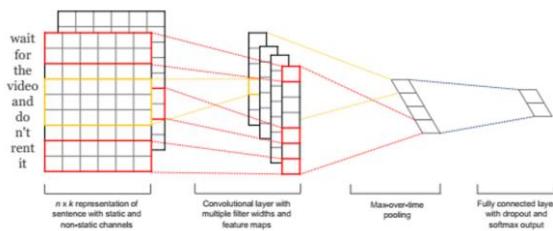
o = output vector

w_2^T = transpose w_2

h = hidden layer

2.2.3.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan algoritma *deep learning* yang dirancang untuk mengolah data berupa gambar, memungkinkan identifikasi objek pada gambar tersebut. Algoritma ini digunakan untuk memahami karakteristik visual dalam gambar, sehingga mampu membedakan setiap gambar berdasarkan fitur yang dikenali [38]. CNN terbukti efektif dalam mengklasifikasikan teks dan kalimat [1]. Pada penelitian ini, metode CNN digunakan untuk melakukan klasifikasi data yang mendukung proses prediksi.



Gambar 2.2 Arsitektur model CNN untuk kalimat

1. *Convolutional Layer*

CNN merupakan jenis *deep learning* yang memanfaatkan *convolutional layer* sebagai penyusun *neural network* yang dibangun. *Convolutional layer* merupakan lapisan pertama dari tahap dalam arsitektur CNN [38]. *Convolutional layer* menggunakan pendekatan *sliding window* dan *weight sharing* untuk menyederhanakan proses perhitungan sehingga mempercepat proses pelatihan.

$$Z[i] = \sum_{j=0}^{k-1} X[i+j] \times W[j] + b \quad (2.3)$$

Keterangan:

$X[i]$ = Elemen ke-I dari urutan *input* pada posisi ke-i

$W[j]$ = Kernel (*filter*) dengan panjang k

b = ias yang ditambahkan setelah operasi konvolusi

$Z[i]$ = Hasil konvolusi atau *feature map* pada posisi i

Aktivasi pada layer ini digunakan fungsi non-linear seperti *Rectified Linear Unit* (ReLU), yang berfungsi untuk menambahkan unsur non-linear dan memungkinkan jaringan mempelajari pola yang lebih kompleks. Jika nilai input $x \leq 0$ maka $x = 0$. Sebaliknya, jika nilai input $x \geq 0$ maka $x = x$.

2. *Pooling Layer*

Pooling layer merupakan komponen penting kedua dalam CNN setelah *convolution layer*. Fungsi dari *pooling layer* adalah melakukan *down-sampling* secara *non-linear*. Tujuan dari *pooling layer* adalah untuk

memilih informasi yang relevan dari feature map. Proses seleksi ini dilakukan dengan menghilangkan nilai-nilai yang kurang signifikan, sehingga dapat mengurangi beban komputasi. Tugas akhir ini menerapkan metode *pooling*, dengan fokus pada *max pooling* yang umum digunakan dalam tugas pemrosesan bahasa alami (NLP). *Max Pooling* yaitu memilih nilai maksimum pada *feature map*.

3. Fully-Connected Layer (Dense)

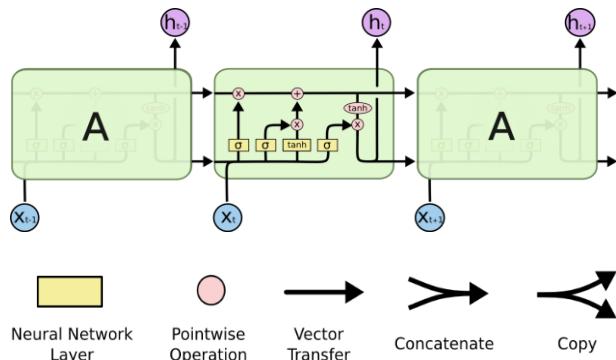
Fully connected layer adalah bagian dari jaringan yang bekerja setelah proses konvolusi. Tugasnya adalah mengambil informasi dari proses sebelumnya, lalu menentukan fitur-fitur yang paling cocok atau paling berkaitan dengan setiap kategori atau kelas yang ingin dikenali. *Fully connected layer* biasanya dipakai di bagian akhir dari jaringan. Fungsinya adalah untuk mengolah semua fitur yang sudah diambil sebelumnya, lalu menghasilkan prediksi. Tujuan utamanya yaitu menggabungkan semua informasi dari node yang ada menjadi satu kesimpulan atau output akhir.

$$y = W_h + b \quad (2.4)$$

2.2.3.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) merupakan pengembangan dalam arsitektur *Recurrent Neural Network* (RNN). LSTM dirancang untuk mampu mengatasi permasalahan *vanishing gradient* [39]. Permasalahan ini membuat RNN tidak mampu mempelajari dan memahami *long term dependencies*. LSTM memiliki struktur yang terdiri dari *memory cell* dan *cell gates*. *Cell gates* ini mencakup tiga jenis gerbang utama, yaitu *forget gate*, *input gate*, dan *output gate* [40]. LSTM memiliki *hidden layer* yang berfungsi untuk menyimpan informasi yang dianggap

penting, sementara informasi yang tidak relevan akan dihapus di setiap sel. Berikut adalah gambaran mengenai struktur gerbang dalam LSTM.

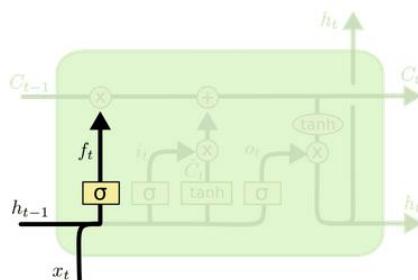


Gambar 2.3 Arsitektur LSTM

Berikut ini terdapat persamaan dari setiap gate pada model LSTM, yaitu:

1. *Forget Gate*

Pada tahap *Forget Gate*, proses penyaringan dilakukan untuk menghapus informasi atau data yang tidak diperlukan dengan bantuan fungsi aktivasi *sigmoid*. Lapisan ini menghasilkan nilai dalam rentang 0 hingga 1. Jika nilai yang diperoleh mendekati 1, informasi tersebut akan dipertahankan. Sebaliknya, jika nilai mendekati 0, informasi tersebut akan dihapus.



Gambar 2.4 Forget Gate Layer

Proses *forget gate* diilustrasikan dengan memori sel menerima nilai hasil keluaran $ht-1$ dan tambahan informasi Xt dari keadaan saat ini. Berikut ini persamaan dari *forget gate* yang didefinisikan di bawah ini:

$$f_t = \sigma (W_f [h_{t-1}, x_t] + b_f) \quad (2.5)$$

Keterangan:

f_t = *forget gate*

σ = fungsi *sigmoid*

Wf = matriks bobot

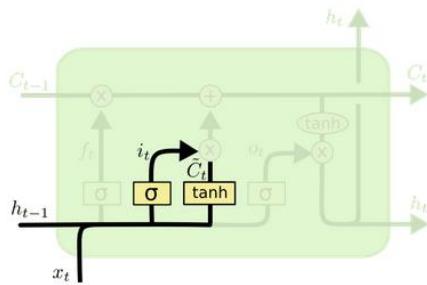
Xt = *input data* (vektor input x dalam *timestep* t)

$ht-1$ = vektor *hidden state* dalam *timestep* sebelumnya t -1

bf = vektor bias

2. Input Gate

Pada tahap *Input Gate*, proses pembaruan informasi dilakukan dengan mengolah data melalui perhitungan khusus pada *input gate*. Selain itu, tahap ini juga menghasilkan kandidat nilai baru dengan menggunakan fungsi aktivasi tanh. Kedua perhitungan tersebut kemudian digabungkan untuk memperbarui status sel sebelum masuk ke tahap berikutnya. Berikut adalah persamaan yang digunakan dalam tahap *Input Gate*.



Gambar 2.5 Input Gate Layer

$$i_t = \sigma (W_i [h_{t-1}, x_t] + b_i) \quad (2.6)$$

Keterangan:

i_t = input gate

σ = fungsi sigmoid

W_i = matriks bobot

x_t = input data (vektor input x dalam timestep t)

h_{t-1} = vektor hidden state dalam timestep sebelumnya t -1

b_i = vektor bias

New Candidate:

$$\tilde{C}_t = \tanh (W_c [h_{t-1}, x_t] + b_c) \quad (2.7)$$

Keterangan:

\tilde{C}_t = kandidat

\tanh = fungsi \tanh

W_c = matriks bobot

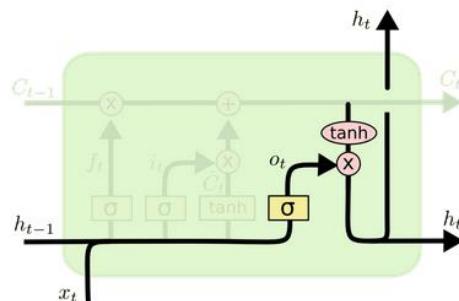
X_t = *input data* (vektor input x dalam *timestep* t)

h_{t-1} = vektor *hidden state* dalam *timestep* sebelumnya t - 1

b_c = vektor bias

3. Output Gate

Pada tahap *Output Gate*, proses pengaturan jumlah keadaan sel yang akan dibuang dilakukan dengan menggunakan fungsi aktivasi *sigmoid*. Setelah itu, keadaan sel saat ini dikalikan dengan fungsi aktivasi \tanh , lalu hasilnya dikalikan kembali dengan *output* dari *Output Gate*. Perhitungan ini menghasilkan nilai *output final*. Berikut adalah persamaan yang digunakan dalam perhitungan *Output Gate*.



Gambar 2.6 Output Gate Layer

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.8)$$

Keterangan:

ot = output gate

σ = fungsi sigmoid

Wo = matriks bobot

X_t = input data (vektor input x dalam timestep t)

$ht-1$ = vektor hidden state dalam timestep sebelumnya t - 1

bo = vektor bias

Berikut ini persamaan dari nilai output akhir sel.

$$h_t = O_t \times \tanh(C_t) \quad (2.9)$$

Keterangan:

h_t = hidden state / hasil akhir

O_t = output gate

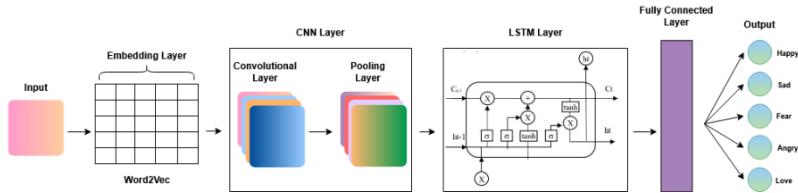
\tanh = fungsi tanh

C_t = cell state

2.2.3.4 CNN-LSTM

CNN-LSTM merupakan pendekatan hibrida yang mengintegrasikan lapisan konvolusional dari CNN dengan jaringan LSTM dalam rangka melakukan prediksi. Kombinasi kedua metode ini bertujuan untuk mengoptimalkan keunggulan CNN dalam mengenali ketergantungan jangka pendek serta memanfaatkan kapasitas memori dari sel LSTM dalam menangkap informasi temporal. Lapisan konvolusional memiliki kemampuan dalam menangkap fitur-fitur lokal yang relevan.

Output dari kedua metode tersebut kemudian dikombinasikan guna memperoleh hasil klasifikasi yang lebih optimal. Hasil integrasi ini menunjukkan peningkatan akurasi dibandingkan ketika masing-masing metode digunakan secara terpisah [16].



Gambar 2.7 Arsitektur Model CNN-LSTM

Model dimulai dengan lapisan *embedding* untuk mengubah input menjadi representasi numerik. Pada lapisan *Convolutional*, berperan untuk mengekstraksi *local feature* dari urutan kata. Lapisan ReLU digunakan sebagai fungsi aktivasi untuk menambahkan non-linearitas dan diikuti oleh lapisan *MaxPooling* untuk mereduksi dimensi serta mempertahankan fitur yang paling penting. Hasil dari *pooling* kemudian digunakan sebagai representasi sekuens dan dimasukkan ke dalam lapisan LSTM, yang bertugas menangkap dependensi jangka panjang dari data teks.

Setelah proses pembelajaran melalui LSTM, hasil keluaran diteruskan ke lapisan *Dense (fully connected)* dan lapisan *output* untuk menghasilkan prediksi label emosi. Model kemudian dievaluasi menggunakan metrik evaluasi.

2.2.3.3 Hyperparameter Tuning

Pemilihan *hyperparameter* yang tepat merupakan langkah krusial dalam membangun model *deep learning* yang efektif. *Hyperparameter* adalah parameter yang ditentukan sebelum proses pelatihan dimulai dan memiliki pengaruh signifikan terhadap performa model. Nilai

hyperparameter yang digunakan masih mengacu pada pengaturan *default* yang diambil dari jurnal terdahulu [14] [16]. Penelitian ini tetap melakukan proses *hyperparameter tuning* untuk mencari kombinasi nilai terbaik guna meningkatkan performa model. Pemilihan kombinasi dilakukan secara bertahap dengan mempertimbangkan hasil evaluasi dari masing-masing percobaan. Pada Tabel 3.1 adalah *Hyperparameter* yang akan digunakan.

Tabel 3.1 *Hyperparameter* dalam CNN-LSTM

<i>Filter Size</i>	<i>Kernel Size</i>	<i>Activation</i>	<i>Units Of LSTM</i>	<i>Learning Rate</i>	<i>Loss</i>	<i>Batch Size</i>	<i>Epoch</i>
[64, 128]	4	ReLU, <i>Softmax</i>	[64, 128]	0.0001	Categorical	32	[50, 100]
					Cross-Entropy	64	
					Cross-Entropy	128	
					Cross-Entropy	192	

Berikut adalah penjelasan mengenai masing-masing *hyperparameter* yang digunakan:

1. *Filter Size*

Jumlah *filter* yang digunakan dalam lapisan konvolusi, yaitu sebanyak 64 dan 128 filter. Berperan dalam menentukan banyaknya filter yang diekstrak dari *input sequence* pada setiap langkah konvolusi.

2. *Kernel Size*

Kernel size yang digunakan adalah 4, yang mengacu pada panjang jendela konvolusi (*convolution window*) yang bergerak melintasi *input sequence*. Ukuran ini menentukan seberapa banyak elemen *input* yang

dipertimbangkan secara bersamaan untuk menghasilkan fitur pada setiap langkah konvolusi.

3. Activation

ReLU berfungsi untuk menambahkan unsur non-linear dan memungkinkan jaringan mempelajari pola yang lebih kompleks. *Softmax* digunakan pada *output layer* untuk klasifikasi *multiclass*.

4. Units

Jumlah unit (*neuron*) yang digunakan pada LSTM adalah sebanyak 64 dan 128. Unit ini merepresentasikan banyaknya sel memori dalam lapisan LSTM, yang berfungsi untuk menyimpan dan memproses informasi urutan.

5. Learning Rate

Learning rate adalah salah satu *hyperparameter* yang mengatur seberapa besar langkah pembaruan bobot yang dilakukan oleh model pada setiap iterasi selama proses pelatihan.

6. Loss

Loss categorical crossentropy digunakan pada permasalahan klasifikasi dengan lebih dari dua kelas (*multiclass*). Berfungsi untuk mengukur sejauh mana perbedaan antara distribusi probabilitas hasil prediksi model dengan distribusi kelas sebenarnya (*ground truth*).

7. Batch Size

Batch Size merupakan jumlah sampel data yang diproses dalam satu iterasi pelatihan model. Penggunaan *batch size* yang kecil dapat membuat proses pelatihan lebih cepat, namun bisa berdampak pada penurunan akurasi model. Sebaliknya, *batch size* yang lebih besar cenderung meningkatkan akurasi, tetapi memerlukan waktu pelatihan yang lebih lama.

8. Epoch

Epoch merupakan proses ketika model menyelesaikan satu siklus pelatihan penuh terhadap seluruh *dataset*. Semakin banyak jumlah *epoch*, model memiliki peluang lebih besar untuk mempelajari informasi dari data. Namun, jika terlalu banyak, hal ini dapat menyebabkan *overfitting*, yaitu kondisi saat model terlalu menghafal data latih dan gagal mengenali pola pada data baru. Sebaliknya, jumlah *epoch* yang terlalu sedikit bisa menyebabkan *underfitting*, di mana model belum cukup belajar untuk menangkap pola yang ada dalam data pelatihan.

2.2.5 Confusion Matrix

Confusion matrix merupakan salah satu metode evaluasi yang sering digunakan dalam analisis sentimen. Metode ini berupa tabel yang berfungsi untuk menilai kinerja model dengan membandingkan hasil prediksi model terhadap nilai aktual dari dataset. *Confusion matrix* terdiri dari empat komponen utama, yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Berdasarkan *confusion matrix*,

berbagai metrik evaluasi seperti akurasi, presisi, *recall*, dan F1-score dapat dihitung untuk mengukur performa model secara lebih mendetail.

Pada evaluasi model klasifikasi *multiclass*, terdapat beberapa pendekatan yang umum digunakan untuk menghitung metrik seperti *precision*, *recall*, dan F1-score, yaitu *micro average*, *macro average*, dan *weighted average*. Setiap pendekatan memiliki karakteristik dan tujuan yang berbeda. Penelitian ini menggunakan metode evaluasi *macro average* karena memberikan bobot yang sama pada setiap kelas, tanpa memandang jumlah data yang dimiliki oleh masing-masing kelas. *Macro average* sangat sesuai untuk data multi-kelas yang memiliki distribusi tidak seimbang, sehingga mampu memberikan gambaran yang lebih adil terhadap performa model pada setiap kelas.

Tabel 2.2 *Confusion Matrix* untuk Klasifikasi Biner

Aktual	Prediksi	
	Positif	Negatif
Positif	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
Negatif	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

Tabel 2.3 *Confusion Matrix* untuk Klasifikasi *Multiclass*

Aktual	Prediksi				
	C1	C2	C3	C4	C5
C1	TP	TN	FP	TN	TN
C2	TN	TP	FP	TN	TN
C3	FN	FN	TP	FN	FN
C4	TN	TN	FP	TP	TN

Aktual	Prediksi				
	C1	C2	C3	C4	C5
C5	TN	TN	FP	TN	TP

Berdasarkan Tabel 2.2 dan Tabel 2.3, berikut adalah penjelasannya:

1. *True Positive* (TP): Jumlah observasi yang sebenarnya memiliki kategori positif dan berhasil diprediksi dengan benar sebagai positif oleh model.
2. *True Negative* (TN): Jumlah observasi yang sebenarnya negatif dan berhasil diprediksi dengan benar sebagai negatif oleh model.
3. *False Positive* (FP): Jumlah observasi yang sebenarnya negatif tetapi keliru diprediksi sebagai positif oleh model.
4. *False Negative* (FN): Jumlah observasi yang sebenarnya positif tetapi keliru diprediksi sebagai negatif oleh model.

Pada penelitian ini digunakan parameter tersebut untuk mengukur akurasi, presisi, *recall* dan *f1-score* dengan rumus sebagai berikut.

1. *Accuracy*

Accuracy mengukur seberapa baik model dalam mengklasifikasikan semua data dengan benar. Nilai akurasi diperoleh dengan membandingkan jumlah prediksi yang benar terhadap total data yang diuji. Persamaan sebagai berikut.

Klasifikasi Biner:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (2. 10)$$

Klasifikasi *Multiclass*:

$$Accuracy = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Prediksi}} \quad (2.11)$$

$$Accuracy = \frac{\sum_{i=1}^k TP_i}{\sum_{i=1}^k (TP_i + FP_i)}$$

Keterangan:

k = Jumlah kelas dalam klasifikasi

TP_i = Jumlah data dari kelas ke- i yang diprediksi benar sebagai kelas tersebut

FP_i = Jumlah data dari kelas ke- i yang diprediksi salah sebagai kelas tersebut

2. *Precision*

Precision mengukur sejauh mana model mengklasifikasikan data dengan benar dibandingkan dengan keseluruhan data yang diprediksi sebagai positif. Persamaan sebagai berikut.

Klasifikasi Biner dan *Multiclass*:

$$Precision: \frac{(TP)}{(TP + FP)} \quad (2.12)$$

Keterangan:

TP = Jumlah data positif yang diprediksi benar oleh model

FP = Jumlah data negatif yang diprediksi salah sebagai positif oleh model

Persamaan *Macro Average Precision* sebagai berikut.

$$Precision (macro): \frac{1}{N} \sum_{i=1}^N Precision (C_i) \quad (2.13)$$

Keterangan:

$TP (C_i) = \text{True Positive}$ untuk kelas ke-i

$FP (C_i) = \text{False Positive}$ untuk kelas ke-i

3. Recall

Recall mengukur seberapa baik model dalam mengidentifikasi data yang benar dalam suatu kelas tertentu. Perhitungannya dilakukan dengan membandingkan jumlah data yang diklasifikasikan dengan benar terhadap total data dalam kelas tersebut. Persamaan sebagai berikut.

Klasifikasi Biner dan *Multiclass*:

$$\text{Recall}: \frac{(TP)}{(TP + FN)} \quad (2. 14)$$

Keterangan:

TP = Jumlah data positif yang diprediksi benar oleh model

FN = Jumlah data positif yang diprediksi salah sebagai negatif oleh model

Persamaan *Macro Average Recall* sebagai berikut.

$$\text{Recall (macro)}: \frac{1}{N} \sum_{i=1}^N \text{Recall} (C_i) \quad (2. 15)$$

4. F1-Score

F1-Score adalah metrik yang menggabungkan *precision* dan *recall* dalam satu nilai rata-rata berbobot untuk menilai keseimbangan keduanya. Persamaan sebagai berikut.

Klasifikasi Biner dan *Multiclass*:

$$\boxed{\text{F1-Score: } 2 \times \frac{Precision \times Recall}{Precision + Recall}} \quad (2. 16)$$

Persamaan *Macro Average F-1 Score* sebagai berikut.

$$\text{F1-Score: } 2 \times \frac{Precision (macro) \times Recall(macro)}{Precision (macro) + Recall (macro)} \quad (2. 17)$$