

**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
2.2.4.1	<i>Word2Vec</i>	22
2.2.4.2	<i>Convolutional Neural Network (CNN)</i>	24
2.2.4.3	<i>Long Short-Term Memory (LSTM)</i>	26
2.2.4.4	<i>CNN-LSTM</i>	31
2.2.4.5	<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 (CNN-LSTM)</i>	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	<i>CNN Layer</i>	65
3.6.3	<i>LSTM 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

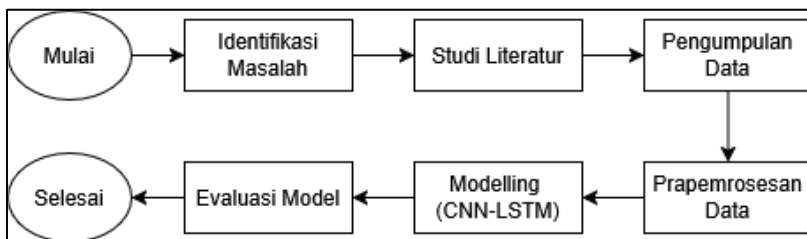
(2.1)	23
(2.2)	24
(2.3)	25
(2.4)	26
(2.5)	28
(2.6)	29
(2.7)	29
(2.8)	30
(2. 9)	31
(2. 10)	37
(2. 11)	38
(2. 12)	38
(2. 13)	38
(2. 14)	39
(2. 15)	39
(2. 16)	40
(2. 17)	40

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Penelitian ini terdiri dari serangkaian langkah yang disusun secara terstruktur untuk mencapai tujuan yang telah ditentukan. Penelitian ini akan mencoba untuk memberikan solusi yang dapat diterapkan dari permasalahan yang ada. Solusi yang akan diberikan adalah dengan membuat sebuah perancangan dan implementasi untuk melakukan klasifikasi emosi berdasarkan data dari media sosial. Metode yang digunakan dalam penelitian ini adalah kombinasi *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM), di mana CNN akan digunakan untuk mengekstrak fitur penting dari teks yang diunggah oleh pengguna, sedangkan LSTM akan membantu dalam memahami pola sekuensial dalam teks untuk meningkatkan akurasi klasifikasi emosi. Data yang digunakan akan diolah melalui beberapa tahap, termasuk preprocessing teks dan gambar, ekstraksi fitur, serta proses pelatihan model menggunakan *dataset* yang relevan. Dimana alur dari proses penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

3.2 Penjabaran Langkah Penelitian

Berikut ini merupakan uraian mendetail mengenai setiap tahap yang terdapat dalam alur penelitian yang dilakukan.

3.2.1 Identifikasi Masalah

Tahap awal dalam penelitian ini dimulai dengan identifikasi permasalahan yang menjadi fokus utama. Penelitian ini bertujuan untuk mengklasifikasikan emosi berdasarkan data dari media sosial *Twitter* menggunakan metode CNN-LSTM. Permasalahan utama yang diangkat adalah sulitnya mengidentifikasi jenis emosi yang terkandung dalam unggahan teks singkat seperti tweet, mengingat sifat bahasa yang digunakan di media sosial cenderung informal, padat, dan sering kali ambigu. Seiring meningkatnya penggunaan *Twitter* sebagai media berekspresi, diperlukan suatu model yang mampu mengenali pola bahasa untuk klasifikasi emosi. Penelitian ini bertujuan untuk membangun model klasifikasi emosi yang dapat memetakan konten teks ke dalam *label* emosi tertentu. Data yang digunakan diperoleh dari *platform* Kaggle, yang berisi kumpulan *tweet* dengan berbagai *label* emosi.

Pengembangan model yang akurat dalam penelitian ini dilakukan dengan menggunakan kombinasi metode CNN dan LSTM. CNN berperan dalam mengekstraksi fitur dari teks, sementara LSTM digunakan untuk menangkap hubungan sekuensial antar kata dalam kalimat. Model ini akan dilatih menggunakan data yang telah diproses, sehingga mampu memberikan prediksi yang lebih optimal dalam mengklasifikasikan emosi berdasarkan pola bahasa yang digunakan oleh pengguna *Twitter*.

3.2.2 Studi Literatur

Pada tahap ini, dilakukan kajian terhadap penelitian sebelumnya yang membahas klasifikasi emosi berbasis media sosial atau penelitian yang menerapkan metode serupa, khususnya dengan model CNN-LSTM. Kajian ini mencakup berbagai sumber seperti paper, jurnal, dan laporan penelitian yang relevan. Hasil dari studi literatur ini berperan sebagai referensi utama yang membantu peneliti dalam memahami konsep, mengembangkan metodologi, serta memperkuat dasar teori dalam penelitian yang sedang dilakukan.

3.2.3 Pengumpulan Data

Setelah menentukan permasalahan dan metode yang akan digunakan dalam pengembangan model, langkah berikutnya adalah mengumpulkan data yang akan digunakan untuk melatih model klasifikasi emosi. Data yang digunakan dalam penelitian ini berasal dari situs *Kaggle* dengan nama dataset *Indonesian Emotion Twitter*¹. *Dataset* ini berisi total 4.401 data dengan lima label emosi, yaitu *love*, *anger*, *sadness*, *joy*, dan *fear*. Data ini akan diolah dan digunakan sebagai dasar dalam proses pelatihan model CNN-LSTM untuk mengklasifikasikan emosi dalam unggahan media sosial.

Tabel 3.2 Jumlah data setiap *label*

<i>Label</i>	Jumlah Data
<i>Anger</i>	1101
<i>Happy</i>	1017

¹ <https://www.kaggle.com/dennisherdi/indonesian-twitter-emotion>

<i>Label</i>	Jumlah Data
<i>Sadness</i>	997
<i>Love</i>	649
<i>Fear</i>	637

3.2.4 Prapemrosesan Data

Data dari dataset *Indonesian Emotion Twitter* dikumpulkan dan dipersiapkan melalui tahap pra-pemrosesan sebelum dianalisis. Pra-pemrosesan data ini mencakup berbagai teknik untuk meningkatkan kualitas data agar lebih sesuai dalam proses analisis dan pelatihan model. Beberapa langkah yang dilakukan meliputi *case folding* untuk mengubah seluruh teks menjadi huruf kecil, *text cleaning* untuk menghilangkan karakter atau simbol yang tidak diperlukan, serta *normalization* guna menyamakan format kata. Selain itu, *stopword removal* diterapkan untuk menghapus kata-kata yang tidak memiliki makna signifikan dalam analisis, dan *stemming* digunakan untuk mengubah kata menjadi bentuk dasarnya. Data yang telah melalui tahap praproses kemudian dibagi menjadi dua bagian, yaitu 80% untuk pelatihan (*training*) dan 20% untuk pengujian (*testing*).

3.2.5 Word Embedding

Data yang telah melewati tahap pra-pemrosesan selanjutnya diproses menggunakan teknik *word embedding* untuk mengubah teks menjadi representasi numerik yang dapat dipahami oleh model pembelajaran mesin. Teknik *word embedding* berfungsi untuk menangkap makna kata serta hubungan semantik di antara kata-kata dalam *dataset*. Metode yang digunakan dalam proses ini adalah *Word2Vec*, yang berperan

dalam mengonversi sekumpulan kata menjadi representasi vektor berdimensi tertentu. *Word2Vec* bekerja dengan cara menganalisis pola kemunculan kata dalam suatu kalimat atau dokumen, sehingga kata-kata yang memiliki makna serupa akan memiliki representasi vektor yang lebih dekat satu sama lain.

Pada tahap awal, proses dimulai dengan mengimpor *term*, yaitu sekumpulan kata unik yang terdapat dalam *dataset*. Kemudian, term tersebut dikonversi menjadi vektor menggunakan metode *one hot encoding*. Setelah itu, dilakukan inisialisasi bobot awal secara acak, yakni W_1 sebagai bobot antara *input layer* dan *hidden layer*, serta W_2 sebagai bobot antara *hidden layer* dan *output layer*.

3.2.6 *Modelling (CNN-LSTM)*

Pada tahap ini, dilakukan proses klasifikasi terhadap data yang telah melewati tahap ekstraksi fitur agar model dapat mengenali pola yang berkaitan dengan klasifikasi emosi. Proses klasifikasi ini terdiri dari dua tahapan utama, yaitu tahap *Convolutional Neural Network (CNN)* dan *Long Short-Term Memory (LSTM)*. Tahap CNN dilakukan dua langkah penting, yakni proses konvolusi (*convolution*) dan proses *pooling*.

Proses konvolusi berfungsi untuk mengekstrak fitur penting dari data teks dengan menggunakan filter yang menangkap pola tertentu, sedangkan proses *pooling* bertujuan untuk mengurangi dimensi fitur tanpa kehilangan informasi esensial, sehingga model menjadi lebih efisien. Hasil akhir dari tahap CNN kemudian digunakan sebagai masukan (*input*) untuk tahap LSTM. Data juga dibagi menjadi dua bagian, yaitu data latih (*training data*) yang digunakan untuk melatih model, serta data uji (*testing*

data) yang digunakan untuk mengukur kinerja model dalam mengklasifikasikan data baru.

3.2.7 Evaluasi Model

Setelah model diterapkan, tahap selanjutnya adalah melakukan proses evaluasi guna menilai kinerja model dalam mengklasifikasikan emosi berbasis media sosial. Evaluasi ini dilakukan dengan mengukur beberapa metrik utama, yaitu akurasi, presisi, *recall*, dan *F1-score*. Akurasi digunakan untuk melihat sejauh mana model dapat mengklasifikasikan data dengan benar secara keseluruhan. Presisi mengukur tingkat ketepatan model dalam mengidentifikasi teks yang akan mengklasifikasikan emosi, sedangkan *recall* menilai sejauh mana model mampu menangkap seluruh data yang benar-benar termasuk dalam kategori tersebut. Sementara itu, *F1-score* merupakan kombinasi dari presisi dan *recall*, yang memberikan gambaran lebih seimbang mengenai performa model.

3.3 Alat dan Bahan Tugas Akhir

Adapun alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

3.3.1 Alat

Berikut adalah alat yang digunakan dalam penelitian ini:

1. Pada penelitian ini penulis menggunakan laptop dengan spesifikasi Sistem Operasi Windows 10, 11th Gen Intel(R) Core (TM) i5-1135G7, Memori 8GB DDR 3, *Hardisk* 500GB.

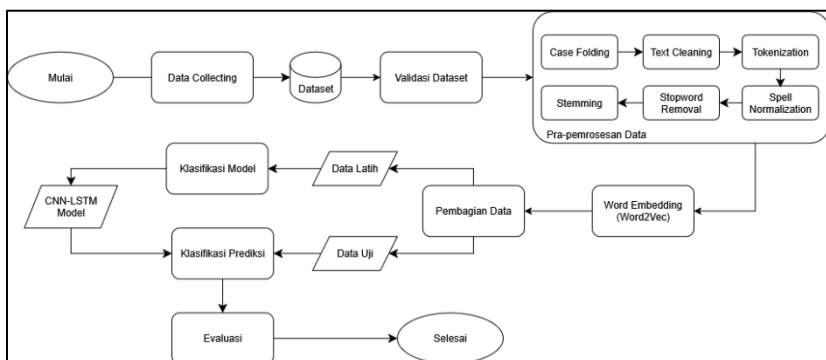
2. *Google Colaboratory*, *runtime type Python* 3, *hardware accelerator CPU*.

3.3.2 Bahan

1. *Dataset Indonesian Twitter Emotion* merupakan data sekunder yang diambil dari penelitian yang dilakukan oleh Ivan Dwi Nugraha dan Yufis Azhar, dengan judul Deteksi Emosi Pengguna Twitter Indonesia Menggunakan LSTM-RNN [18].
2. Jurnal penelitian dari penelitian sebelumnya untuk memberikan landasan teori dan rangkaian konsep serta ide untuk mendukung penelitian.

3.4 Metode Pengembangan

Pada bab ini, akan dibahas mengenai model pengembangan yang digunakan dalam penelitian ini. Seluruh proses yang akan dilakukan dalam klasifikasi emosi menggunakan CNN-LSTM dapat dilihat pada Gambar 3.2.



Gambar 3.2 Metode Pengembangan

3.5 Penjabaran Langkah Penelitian

3.5.1 Data Collecting

Dataset yang digunakan dalam penelitian ini terdiri dari 4.403 baris *tweet* yang dikategorikan ke dalam lima label emosi, yaitu marah, takut, gembira, cinta, dan sedih. Data ini disediakan dalam bentuk format csv. Setiap baris mencakup sebuah *tweet* beserta label emosinya. Berikut ini gambaran secara sederhana mengenai *dataset* yang didapatkan. Adapun contoh dari *dataset* awal yang dilampirkan pada Tabel 3.2.

Tabel 3.3 Sampel *Dataset* Awal

Label	Tweet
Anger	Sesama cewe lho (kayaknya), harusnya bisa lebih rasain lah yang harus sibuk jaga diri, rasain sakitnya haid, dan paniknya pulang malem sendirian. Gimana orang asing? Wajarlah banyak korban yang takut curhat, bukan dibela malah dihujat.
Happy	Kepingin gudeg mbarek Bu Hj. Amad Foto dari Google, sengaja, biar teman-teman jg membayangkannya. Berbagi itu indah.
...	...
Sadness	Sulit menerima kenyataan memang bahwa ada sebagian yang menganggap ketika kita berinteraksi dengan satu orng dari satu golongan tertentu kita langsung mengatakan golongan tertentu sesaat tanpa kita mempelajari secara utuh dan mengklaim diri kita paling benar.

Label	Tweet
Love	Setiap kesempatan yg pernah hadir tuk dapat membuatmu selalu merasa aman dan nyaman, kini jadi suatu kehormatan yg pernah didapat.

3.5.2 Validasi *Dataset*

Validasi Dataset adalah proses pemeriksaan dan penilaian terhadap kualitas serta kelayakan data yang digunakan dalam penelitian. Tahap ini membantu memastikan bahwa *dataset* tidak mengandung kesalahan yang dapat mempengaruhi hasil analisis atau kinerja model.

3.5.3 Prapemrosesan Data

Prapemrosesan Data terdiri dari beberapa langkah yang dirancang untuk mempersiapkan dan meningkatkan kualitas data teks agar siap untuk dianalisis lebih lanjut. Langkah-langkah ini umumnya meliputi konversi huruf besar kecil (*case folding*), penghapusan karakter khusus, penyesuaian ejaan *slang* (*spell normalization*), pengembalian kata ke bentuk dasar (*stemming*), serta eliminasi kata-kata yang tidak memiliki makna penting (*stopword*).

1. *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].

Tabel 3.4 Tahapan *Case Folding*

Label	Tweet	Hasil
Anger	Sesama cewe lho (kayaknya), harusnya bisa lebih rasain lah yang harus sibuk jaga diri, rasain sakitnya haid, dan paniknya pulang malem sendirian. Gimana orang asing? Wajarlah banyak korban yang takut curhat, bukan dibela malah dihujat.	sesama cewe lho (kayaknya), harusnya bisa lebih rasain lah yang harus sibuk jaga diri, rasain sakitnya haid, dan paniknya pulang malem sendirian. gimana orang asing? wajarlah banyak korban yang takut curhat, bukan dibela malah dihujat.
Sadness	Sulit menerima kenyataan memang bahwa ada sebagian yang menganggap ketika kita berinteraksi dengan satu orng dari satu golongan tertentu kita langsung mengatakan golongan tertentu sesaat tanpa kita mempelajari secara utuh dan mengklaim diri kita paling benar.	sulit menerima kenyataan memang bahwa ada sebagian yang menganggap ketika kita berinteraksi dengan satu orng dari satu golongan tertentu kita langsung mengatakan golongan tertentu sesaat tanpa kita mempelajari secara utuh dan mengklaim diri kita paling benar.

2. *Text Cleaning*

Text Cleaning adalah proses untuk membersihkan dan memvalidasi kata-kata yang tidak diinginkan. Tahap ini, kata-kata atau karakter yang dianggap tidak relevan atau mengganggu, seperti simbol, tanda baca, emoji, dan *hashtag* akan dihapus dari teks [30].

Tabel 3.5 Tahapan *Text Cleaning*

<i>Label</i>	<i>Tweet</i>	Hasil	Kata/Frasa yang dihapus/diganti	
			Kata/Frasa asli	Perubahan
<i>Anger</i>	Sesama cewe lho (kayaknya), harusnya bisa lebih rasain lah yang harus sibuk jaga diri, rasain sakitnya haid, dan paniknya pulang malem sendirian. Gimana orang asing? Wajarlah banyak korban yang takut curhat, bukan dibela malah dihujat.	Sesama perempuan seharusnya bisa lebih memahami pentingnya menjaga diri, merasakan rasa sakit saat menstruasi, dan rasa takut ketika pulang malam sendirian. Bagaimana dengan orang asing? Wajar jika banyak korban merasa takut untuk bercerita karena khawatir tidak didukung.	Cewe Lho (kayanya) Rasain Haid Paniknya Pulang malem Curhat Dihujat	Perempuan (Dihapus) Merasakan Menstruasi Rasa takut Pulang malam Bercerita Tidak didukung

<i>Label</i>	<i>Tweet</i>	Hasil	Kata/Frasa yang dihapus/diganti	
			Kata/Frasa asli	Perubahan
<i>Sadness</i>	Sulit menerima kenyataan memang bahwa ada sebagian yang menganggap ketika kita berinteraksi dengan satu orng dari satu golongan tertentu kita langsung mengatakan golongan tertentu sesaat tanpa kita mempelajari secara utuh dan mengklaim diri kita paling benar.	Sulit menerima kenyataan bahwa sebagian orang menganggap ketika kita berinteraksi dengan seseorang dari suatu kelompok, kita langsung menilai kelompok tersebut tanpa memahami secara menyeluruh dan merasa diri paling benar.	Org Golongan Mengatakan golongan Sesaat Mengklaim diri kita	Seseorang Kelompok Menilai kelompok (Dihapus) Merasa diri

3. Tokenization

Tokenizing tahap pemecahan teks menjadi bagian-bagian paling kecil seperti kata, frasa, atau kalimat. Tahapan ini memiliki peran krusial dalam

membantu sistem komputer mengenali struktur serta makna dari teks yang dianalisis [28].

Tabel 3.6 Tahapan *Tokenizing*

<i>Label</i>	<i>Tweet</i>	<i>Output</i>
<i>Anger</i>	Sesama cewe lho (kayaknya), harusnya bisa lebih rasain lah yang harus sibuk jaga diri, rasain sakitnya haid, dan paniknya pulang malem sendirian. Gimana orang asing? Wajarlah banyak korban yang takut curhat, bukan dibela malah dihujat.	‘Sesama’, ‘cewe’, ‘lho’, ‘kayaknya’, ‘harusnya’, ‘bisa’, ‘lebih’, ‘rasain’, ‘lah’, ‘yang’, ‘harus’, ‘sibuk’, ‘jaga’, ‘diri’, ‘rasain’, ‘sakitnya’, ‘haid’, ‘dan’, ‘paniknya’, ‘pulang’, ‘malem’, ‘sendirian’, ‘gimana’, ‘orang’, ‘asing’, ‘wajarlah’, ‘banyak’, ‘korban’, ‘yang’, ‘takut’, ‘curhat’, ‘bukan’, ‘dibela’, ‘malah’, ‘dihujat’
<i>Sadness</i>	Sulit menerima kenyataan memang bahwa ada sebagian yang menganggap ketika kita berinteraksi dengan satu orng dari satu golongan tertentu kita langsung mengatakan golongan tertentu sesaat tanpa kita mempelajari	‘Sulit’, ‘terima’, ‘nyata’, ‘memang’, ‘bahwa’, ‘ada’, ‘bagi’, ‘yang’, ‘anggap’, ‘ketika’, ‘kita’, ‘interaksi’, ‘dengan’, ‘satu’, ‘orang’, ‘dari’, ‘satu’, ‘golong’, ‘tentu’, ‘kita’, ‘langsung’, ‘kata’, ‘golong’, ‘tentu’, ‘sesaat’, ‘tanpa’, ‘kita’, ‘pelajari’, ‘secara’, ‘utuh’, ‘dan’,

<i>Label</i>	<i>Tweet</i>	<i>Output</i>
	secara utuh dan mengklaim diri kita paling benar.	‘klaim’, ‘diri’, ‘kita’, ‘paling’, ‘benar’

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].

Tabel 3.7 Tahapan *Spell Normalization*

<i>Label</i>	<i>Tweet</i>	Hasil
<i>Anger</i>	Sesama cewe lho (kayaknya), harusnya bisa lebih rasain lah yang harus sibuk jaga diri, rasain sakitnya haid, dan paniknya pulang malem sendirian. Gimana orang asing? Wajarlah banyak korban	Sesama perempuan seharusnya bisa lebih merasakan bagaimana sibuknya menjaga diri, merasakan sakit saat menstruasi, dan kepanikan ketika pulang malam sendirian. Bagaimana dengan orang asing? Wajar jika banyak

<i>Label</i>	<i>Tweet</i>	<i>Hasil</i>
	yang takut curhat, bukan dibela malah dihujat.	korban merasa takut untuk bercerita, bukan dibela tetapi justru disalahkan.
<i>Sadness</i>	Sulit menerima kenyataan memang bahwa ada sebagian yang menganggap ketika kita berinteraksi dengan satu orng dari satu golongan tertentu kita langsung mengatakan golongan tertentu sesaat tanpa kita mempelajari secara utuh dan mengklaim diri kita paling benar.	Sulit menerima kenyataan bahwa ada sebagian orang yang menganggap ketika kita berinteraksi dengan seseorang dari suatu kelompok tertentu, kita langsung menilai kelompok tersebut tanpa mempelajarinya secara menyeluruh dan menganggap diri sendiri paling benar.

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 kepemilikan, dan kata ganti orang, akan dihilangkan [31].

Tabel 3.8 Tahapan *Stopword Removal*

<i>Label</i>	<i>Tweet</i>	Hasil	Kata yang dihapus
<i>Anger</i>	Sesama cewe lho (kayaknya), harusnya bisa lebih rasain lah yang harus sibuk jaga diri, rasain sakitnya haid, dan paniknya pulang malem sendirian. Gimana orang asing? Wajarlah banyak korban yang takut curhat, bukan dibela malah dihujat.	Sesama cewe (kayaknya), harusnya bisa lebih rasain sibuk jaga diri, rasain sakit haid, panik pulang malem sendirian. Gimana orang asing? Wajarlah banyak korban takut curhat, bukan dibela malah dihujat.	‘lho’, ‘kayanya’, ‘harusnya’, ‘bisa’, ‘lebih’, ‘lah’. ‘yang’, ‘dan’, ‘gimana’, ‘wajarlah’, ‘malah’.
<i>Sadness</i>	Sulit menerima kenyataan memang bahwa ada sebagian yang menganggap ketika kita berinteraksi dengan satu orng dari satu golongan tertentu kita langsung mengatakan golongan tertentu sesaat tanpa kita	Sulit menerima kenyataan, sebagian menganggap berinteraksi satu orng golongan tertentu langsung mengatakan golongan, tanpa mempelajari utuh, mengklaim paling benar.	‘memang’, ‘bahwa’, ‘ada’, ‘yang’, ‘ketika’, ‘dengan’, ‘dari’, ‘kita’, ‘se saat’, ‘secara’,

<i>Label</i>	<i>Tweet</i>	<i>Hasil</i>	Kata yang dihapus
	mempelajari secara utuh dan mengklaim diri kita paling benar.		‘dan’, ‘diri kita’.

1. *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].

Tabel 3.9 Tahapan *Stemming*

<i>Label</i>	<i>Tweet</i>	<i>Output</i>	Kata/Frasa yang diubah	
			Kata asli	Kata dasar
<i>Anger</i>	Sesama cewe lho (kayaknya), harusnya bisa lebih rasain lah yang harus sibuk jaga diri, rasain	Sesama cewe, rasa sibuk jaga diri, rasa sakit haid, panik pulang malam sendiri.	Harusnya Rasain Sakitnya Paniknya Sendirian Dwajarlal	Harus Rasa Sakit Panik Sendiri Wajar

<i>Label</i>	<i>Tweet</i>	<i>Output</i>	Kata/Frasa yang diubah	
			Kata asli	Kata dasar
	sakitnya haid, dan paniknya pulang malem sendirian. Gimana orang asing? Wajarlah banyak korban yang takut curhat, bukan dibela malah dihujat.	Gimana orang asing? Wajar banyak korban takut curhat, bukan bela malah hujat.	Dibela Dihujat	Bela Hujat
<i>Sadness</i>	Sulit menerima kenyataan memang bahwa ada sebagian yang menganggap ketika kita berinteraksi dengan satu orang dari satu golongan tertentu kita	Sulit terima nyata, sebagian anggap interaksi satu orang golongan tentu langsung kata golongan tentu tanpa pelajar utuh, klaim diri paling benar.	Menerima Kenyataan Menganggap Berinteraksi Mengatakan Mempelajari Mengklaim Golongan Tertentu Sesaat	Terima Nyata Anggap Interaksi Kata Pelajar Klaim Golong Tentu Saat

<i>Label</i>	<i>Tweet</i>	<i>Output</i>	Kata/Frasa yang diubah	
			Kata asli	Kata dasar
	langsung mengatakan golongan tertentu sesaat tanpa kita mempelajari secara utuh dan mengklaim diri kita paling benar.			

2. *Word2Vec (Skip-Gram)*

Dalam kamus kata (*vocabulary*), terdapat sekumpulan term seperti ‘hati’, ‘mudah’, ‘tersakiti’, ‘dinilai’, ‘tanpa’, dan ‘mengerti’. Ilustrasi perhitungan pembobotan kata menggunakan metode *Word2Vec* terhadap *term-term* tersebut dilakukan dengan ukuran jendela (*window size*) $c = 2$ dan jumlah dimensi *neuron* sebanyak 5.

1. Representasi kata menjadi *one hot encoding vector*

$$hati (X_0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad mudah (X_1) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned}
 \text{tersakiti } (X_2) &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \text{dinilai } (X_3) &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \\
 \text{tanpa } (X_4) &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \text{mengerti } (X_5) &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
 \end{aligned}$$

Tabel 3.10 Ilustrasi Inisialisasi Data *Training Skip-gram*

Teks Masukan					
hati	mudah	tersakiti	dinilai	tanpa	mengerti
hati	mudah	tersakiti	dinilai	tanpa	mengerti
hati	mudah	tersakiti	dinilai	tanpa	mengerti
hati	mudah	tersakiti	dinilai	tanpa	mengerti
hati	mudah	tersakiti	dinilai	tanpa	mengerti
hati	mudah	tersakiti	dinilai	tanpa	mengerti

Sampel <i>Training</i>
(hati, mudah), (hati, tersakiti)
(mudah, hati), (mudah, tersakiti), (mudah, dinilai)
(tersakiti, hati), (tersakiti, mudah), (tersakiti, dinilai), (tersakiti, tanpa)
(dinilai, mudah), (dinilai, tersakiti), (dinilai, tanpa), (dinilai, mengerti)
(tanpa, tersakiti), (tanpa, dinilai), (tanpa, mengerti)
(mengerti, dinilai), (mengerti, tanpa)

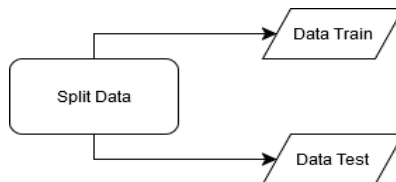
Penggunaan $windows(c) = 2$ menandakan bahwa jarak antar kata $input(w(t))$ dengan kata sekitarnya yaitu 2 yang berarti 2 kata sebelum dan sesudah kata $input$.

2. Inisialisasi nilai W awal

Nilai W adalah nilai bobot yang tidak memiliki nilai pasti, sehingga diinisialisasikan sebagai nilai acak dengan ukuran vektor. Kita asumsikan *embedding* tiap kata menghasilkan vektor berdimensi 4. Setiap kata akan memiliki representasi vektor sebagai berikut:

$$X = \begin{bmatrix} 0.5 & 0.3 & 0.1 & 0.6 \\ 0.4 & 0.7 & 0.2 & 0.9 \\ 0.2 & 0.6 & 0.1 & 0.8 \\ 0.3 & 0.4 & 0.5 & 0.7 \\ 0.6 & 0.8 & 0.3 & 0.2 \\ 0.9 & 0.2 & 0.1 & 0.4 \end{bmatrix}$$

3.5.4 Pembagian Data



Gambar 3.3 *Split Data*

Pada Gambar 3.4 menunjukkan pembagian *dataset* dilakukan dengan rasio 70% untuk data latih dan 30% untuk data validasi. Data latih digunakan untuk melatih model, sementara data validasi berfungsi untuk menguji kinerja model yang telah dilatih. Dapat diperoleh evaluasi yang

lebih akurat mengenai seberapa baik model dapat melakukan generalisasi terhadap data yang belum pernah dilihat sebelumnya.

3.5.5 Arsitektur Model CNN-LSTM

CNN-LSTM merupakan gabungan antara algoritma *Convolutional Neural Network* (CNN) dan *Long Short Term Memory* (LSTM). Sesuai dengan namanya, struktur model CNN-LSTM menempatkan algoritma LSTM di lapisan terakhir dalam model jaringan CNN. Sebelum vektor input memasuki jaringan LSTM, harus terlebih dahulu melalui proses algoritma CNN. Berikut penjelasan rinci setiap lapisan:

1. *Input Layer*

Input layer bertugas menampung data *tweet* yang telah melalui tahap *preprocessing*, seperti pembersihan teks (menghilangkan karakter tidak penting, stopword removal), tokenisasi (pemecahan kalimat menjadi kata-kata individual), dan stemming. Data yang masuk ke dalam *layer* ini berupa deretan indeks *integer* yang merepresentasikan kata-kata dalam *vocabulary*.

2. *Embedding Layer*

Embedding layer berfungsi mengubah representasi kata berupa indeks menjadi vektor kontinu yang merepresentasikan makna semantik kata. Sebagai contoh, kata 'sedih' dan 'depresi' akan memiliki vektor yang saling berdekatan dalam ruang vektornya. Penelitian ini menggunakan *Word2Vec* untuk menangkap hubungan semantik antar kata secara lebih akurat dibandingkan pelatihan dari awal.

3. CNN Layer

a. Convolutional Layer

Tahapan pertama dalam proses *feedforward* adalah lapisan konvolusi (*convolution layer*). Pada tahap ini, data input yang telah berbentuk vektor kata akan diproses melalui operasi konvolusi untuk mengekstraksi nilai-nilai fitur dari kumpulan vektor tersebut. Setiap *filter* akan menghasilkan *feature map* yang menyoroti bagian teks yang memiliki pola tertentu. Aktivasi ReLU (*Rectified Linear Unit*) diterapkan untuk memperkenalkan non-linearitas dan memastikan hanya fitur yang relevan yang dipertahankan. Lapisan ini efektif menangkap ciri lokal tanpa terpengaruh posisi absolut kata dalam teks.

b. Pooling Layer

Lapisan kedua pada algoritma CNN-LSTM yaitu *pooling layer*. Pada lapisan ini akan diambil nilai maksimum dari *feature map* yang telah didapatkan dari *convolution layer*. *Max-pooling layer* bertugas menyaring *feature map* yang dihasilkan oleh *convolutional layer* dengan mengambil nilai maksimum dari setiap wilayah *pooling*.

4. LSTM Layer

LSTM layer dirancang untuk memahami hubungan antar kata dalam jarak jauh, seperti kalimat kompleks yang menggambarkan emosi. Lapisan ini menggunakan mekanisme *gate* (*input*, *forget*, dan *output gate*) untuk mengontrol aliran informasi, memungkinkan model untuk "mengingat" atau "melupakan" informasi tertentu sesuai konteks. *Output* LSTM adalah representasi teks yang telah mempertimbangkan seluruh *sequence*.

5. *Output Layer*

Output layer menerima *output* dari LSTM dan memetakannya ke dalam satu neuron dengan aktivasi *sigmoid*. *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.

3.6 Ilustrasi Perhitungan Metode

3.6.1 *Word2Vec (Skip-Gram)*

Inisialisasi awal nilai bobot (W)

$$W = \begin{bmatrix} 0.5 & 0.3 & 0.1 & 0.6 \\ 0.4 & 0.7 & 0.2 & 0.9 \\ 0.2 & 0.6 & 0.1 & 0.8 \\ 0.3 & 0.4 & 0.5 & 0.7 \\ 0.6 & 0.8 & 0.3 & 0.2 \\ 0.9 & 0.2 & 0.1 & 0.4 \end{bmatrix}$$

3.6.2 *CNN Layer*

1. *Convolution Layer*

Konvolusi menggunakan *kernel size* = 2, *stride* = 1

Setiap filter memiliki bobot:

$$F = \begin{bmatrix} 0.2 & 0.1 & -0.1 & 0.5 \\ 0.3 & -0.2 & 0.4 & 0.1 \end{bmatrix}$$

Asumsi:

Bias (b) = 0.2

Window size (h) = 2

a. Window 1: Hati + mudah

$$\begin{aligned} output &= (0.5)(0.2) + (0.3)(0.1) + (0.1)(-0.1) + (0.6)(0.5) + \\ &+ (0.4)(0.3) + (0.7)(-0.2) + (0.2)(0.4) + (0.9)(0.1) \\ output &= 0.57 \end{aligned}$$

b. Window 2: mudah + tersakiti

$$\begin{aligned} output &= (0.4)(0.2) + (0.7)(0.1) + (0.2)(-0.1) + (0.9)(0.5) \\ &+ \\ &+ (0.2)(0.3) + (0.6)(-0.2) + (0.1)(0.4) + (0.8)(0.1) \\ output &= 0.64 \end{aligned}$$

c. Window 3: tersakiti + dinilai

$$\begin{aligned} output &= (0.2)(0.2) + (0.6)(0.1) + (0.1)(-0.1) + (0.8)(0.5) \\ &+ \\ &+ (0.3)(0.3) + (0.4)(-0.2) + (0.5)(0.4) + (0.7)(0.1) \\ output &= 0.77 \end{aligned}$$

d. Window 4: dinilai + tanpa

$$\begin{aligned} output &= (0.3)(0.2) + (0.4)(0.1) + (0.5)(-0.1) + (0.7)(0.5) \\ &+ \\ &+ (0.6)(0.3) + (0.8)(-0.2) + (0.3)(0.4) + (0.2)(0.1) \\ output &= 0.54 \end{aligned}$$

e. Window 5: tanpa + mengerti

$$\begin{aligned} output &= (0.6)(0.2) + (0.8)(0.1) + (0.3)(-0.1) + (0.2)(0.5) \\ &+ \\ &+ (0.9)(0.3) + (0.2)(-0.2) + (0.1)(0.4) + (0.4)(0.1) \\ output &= 0.58 \end{aligned}$$

CNN Output		CNN Output + Bias (0.2)
0.57	→	0.77
0.64		0.84
0.77		0.97
0.54		0.74
0.58		0.78

Hasil dari perhitungan filter di atas digabungkan menjadi satu peta aktivasi, kemudian ditambahkan dengan bias yang diperoleh dari perhitungan.

CNN Output + Bias (0.2)		ReLU
0.77	→	0.77
0.84		0.84
0.97		0.97
0.74		0.74
0.78		0.78

ReLU digunakan untuk menghilangkan nilai-nilai negatif dari *feature map* dengan tujuan agar proses pembelajaran model menjadi lebih efisien. Dengan mengubah semua nilai negatif menjadi nol, ReLU membantu mencegah masalah yang dikenal sebagai "neuron mati" dan memungkinkan model untuk fokus hanya pada informasi penting yang bersifat positif, sehingga kinerja model dalam mengenali pola atau fitur menjadi lebih baik dan stabil.

2. Pooling Layer – Max Pooling

Penerapan *max pooling* bertujuan untuk mengurangi ukuran keluaran dari CNN. Proses ini menggunakan *window* = 2 dan *stride* = 1, yang berarti pergeseran dilakukan satu piksel setiap kali proses berlangsung.

$$Input = [0.77, 0.84, 0.97, 0.74, 0.78]$$

Hasil *Max Pooling* =

- a. $max(0.77, 0.84) = 0.84$
- b. $max(0.84, 0.97) = 0.97$
- c. $max(0.97, 0.74) = 0.97$
- d. $max(0.74, 0.78) = 0.78$

$$Output = [0.84, 0.97, 0.97, 0.78]$$

3.6.3 LSTM Layer

LSTM memproses data secara bertahap berdasarkan urutan waktu (*time step*). Pada studi ini, setiap nilai dianggap mewakili satu *time step*, sehingga *input* yang digunakan adalah [0.84], [0.97], [0.97], dan [0.78]. Untuk menyederhanakan proses perhitungan, digunakan dimensi tersembunyi (*hidden size*) sebesar 1. Nilai parameter yang digunakan pun disederhanakan, yaitu:

Tabel 3.11 Inisialisasi Nilai Bobot dan Bias

Variabel	Nilai
W	0.5
b f	0.1
b i	0.2
b c	0
b o	0.1

Keterangan:

w = bobot

b_f, *b_i*, *b_c*, *b_o* = bias

Sebagai ilustrasi, perhitungan difokuskan pada *cell* pertama ($t = 1$) dengan asumsi bahwa nilai awal h_o dan c_o adalah 0.

1. *Timestep* $t = 1$, $x_1 = 0.84$, $h_o = 0$, $c_o = 0$

a. *Forget Gate*

$$\begin{aligned} f_t &= \sigma (W_f [h_{t-1}, x_t] + b_f) \\ &= \sigma ((0.5 \times 0) + (0.5 \times 0.84)) + 0.1 \\ &= \sigma (0.52) \\ &= 0.627 \end{aligned}$$

b. *Input Gate*

$$\begin{aligned} i_t &= \sigma (W_i [h_{t-1}, x_t] + b_i) \\ &= \sigma ((0.5 \times 0) + (0.5 \times 0.84)) + 0.2 \\ &= \sigma (0.62) \\ &= 0.650 \end{aligned}$$

c. *Candidate Cell*

$$\begin{aligned} \tilde{C}_t &= \tanh (W_c [h_{t-1}, x_t] + b_c) \\ &= \tanh((0.5 \times 0) + (0.5 \times 0.84)) + 0 \\ &= \tanh (0.42) \\ &= 0.397 \end{aligned}$$

d. *Cell State*

$$\begin{aligned} C_t &= f_t \times C_{t-1} + i_t \times \tilde{C}_t \\ &= 0.627 \times 0 + 0.650 \times 0.397 \\ &= 0.258 \end{aligned}$$

e. *Output Gate*

$$\begin{aligned} O_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ &= \sigma((0.5 \times 0) + (0.5 \times 0.84)) + 0.1 \\ &= \sigma(0.52) \\ &= 0.627 \end{aligned}$$

f. *Hidden State*

$$\begin{aligned} h_t &= O_t \times \tanh(C_t) \\ &= 0.627 \times \tanh(0.258) \\ &= 0.627 \times 0.252 \\ &= 0.158 \end{aligned}$$

2. *Timestamp* $t = 2$, $x_1 = 0.97$, $h_1 = 0.158$, $c_1 = 0.258$

a. *Forget Gate*

$$\begin{aligned} f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\ &= \sigma((0.5 \times 0.158) + (0.5 \times 0.97)) + 0.1 \\ &= \sigma(0.664) \\ &= 0.660 \end{aligned}$$

b. *Input Gate*

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\ &= \sigma((0.5 \times 0.158) + (0.5 \times 0.97)) + 0.2 \\ &= \sigma(0.764) \\ &= 0.682 \end{aligned}$$

c. *Candidate Cell*

$$\begin{aligned}
\tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c) \\
&= \tanh((0.5 \times 0.158) + (0.5 \times 0.97)) + 0 \\
&= \tanh(0.564) \\
&= 0.511
\end{aligned}$$

d. *Cell State*

$$\begin{aligned}
C_t &= f_t \times C_{t-1} + i_t \times \tilde{C}_t \\
&= 0.660 \times 0.258 + 0.628 \times 0.511 \\
&= 0.519
\end{aligned}$$

e. *Output Gate*

$$\begin{aligned}
O_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\
&= \sigma((0.5 \times 0.158) + (0.5 \times 0.97)) + 0.1 \\
&= \sigma(0.664) \\
&= 0.660
\end{aligned}$$

f. *Hidden State*

$$\begin{aligned}
h_t &= O_t \times \tanh(C_t) \\
&= 0.660 \times \tanh(0.519) \\
&= 0.660 \times 0.477 \\
&= 0.315
\end{aligned}$$

3. *Timestep t = 3, $x_3 = 0.97, h_2 = 0.315, c_2 = 0.519$*

a. *Forget Gate*

$$\begin{aligned}
f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\
&= \sigma((0.5 \times 0.315) + (0.5 \times 0.97)) + 0.1
\end{aligned}$$

$$= \sigma (0.742)$$

$$= 0.677$$

b. Input Gate

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

$$= \sigma ((0.5 \times 0.315) + (0.5 \times 0.97)) + 0.2$$

$$= \sigma (0.842)$$

$$= 0.699$$

c. Candidate Cell

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

$$= \tanh ((0.5 \times 0.246) + (0.5 \times 0.97)) + 0$$

$$= \tanh (0.842)$$

$$= 0.699$$

d. Cell State

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

$$= 0.677 \times 0.519 + 0.699 \times 0.566$$

$$= 0.747$$

e. Output Gate

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

$$= \sigma ((0.5 \times 0.246) + (0.5 \times 0.97)) + 0.1$$

$$= \sigma (0.742)$$

$$= 0.677$$

f. Hidden State

$$\begin{aligned}
h_t &= O_t \times \tanh(C_t) \\
&= 0.677 \times \tanh(0.747) \\
&= 0.67 \times 0.633 \\
&= 0.428
\end{aligned}$$

4. *Timestep* $t = 4$, $x_4 = 0.78$, $h_3 = 0.428$, $c_3 = 0.747$

a. *Forget Gate*

$$\begin{aligned}
f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\
&= \sigma((0.5 \times 0.428) + (0.5 \times 0.78)) + 0.1 \\
&= \sigma(0.704) \\
&= 0.669
\end{aligned}$$

b. *Input Gate*

$$\begin{aligned}
i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\
&= \sigma((0.5 \times 0.428) + (0.5 \times 0.78)) + 0.2 \\
&= \sigma(0.804) \\
&= 0.690
\end{aligned}$$

c. *Candidate Cell*

$$\begin{aligned}
\tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c) \\
&= \tanh((0.5 \times 0.428) + (0.5 \times 0.78)) + 0 \\
&= \tanh(0.604) \\
&= 0.540
\end{aligned}$$

d. *Cell State*

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

$$\begin{aligned}
&= 0.669 \times 0.747 + 0.690 \times 0.540 \\
&= 0.872
\end{aligned}$$

e. *Output Gate*

$$\begin{aligned}
O_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\
&= \sigma((0.5 \times 0.428) + (0.5 \times 0.78)) + 0.1 \\
&= \sigma(0.704) \\
&= 0.669
\end{aligned}$$

f. *Hidden State*

$$\begin{aligned}
h_t &= O_t \times \tanh(C_t) \\
&= 0.669 \times \tanh(0.872) \\
&= 0.669 \times 0.702 \\
&= 0.470
\end{aligned}$$

3.6.4 *Fully Connected Layer (Dense Layer)*

Misalkan memiliki vektor bobot W dan bias b yang ukurannya sesuai dengan 5 kelas *output*.

Asumsi:

Vektor bobot W dengan dimensi 1×5 dapat dituliskan sebagai:

$$W = [0.8 \quad 0.2 \quad 0.1 \quad 0.4 \quad 0.3]$$

Diketahui: $h_4 = 0.470$

$$y_1(sad) = (0.8 \times 0.470) = 0.376$$

$$y_2(happy) = (0.2 \times 0.470) = 0.094$$

$$y_3(love) = (0.1 \times 0.470) = 0.047$$

$$y_4(fear) = (0.4 \times 0.470) = 0.188$$

$$y_5(angry) = (0.3 \times 0.470) = 0.141$$

Output dari *dense layer* adalah
 $= [0.376 \quad 0.094 \quad 0.047 \quad 0.188 \quad 0.141]$

3.6.5 Softmax

Eksponensial dari y_1 sampai y_5 :

$$e^{y_1} = e^{0.376} \approx 1.465$$

$$e^{y_2} = e^{0.094} \approx 1.098$$

$$e^{y_3} = e^{0.047} \approx 1.048$$

$$e^{y_4} = e^{0.188} \approx 1.207$$

$$e^{y_5} = e^{0.141} \approx 1.151$$

$$\begin{aligned} \text{softmax}(y_1) &= \frac{1.465}{1.465 + 1.098 + 1.048 + 1.207 + 1.151} = \frac{1.465}{5.969} \\ &\approx 0.245 \end{aligned}$$

$$\begin{aligned} \text{softmax}(y_2) &= \frac{1.098}{1.465 + 1.098 + 1.048 + 1.207 + 1.151} = \frac{1.098}{5.969} \\ &\approx 0.183 \end{aligned}$$

$$\begin{aligned} \text{softmax}(y_3) &= \frac{1.048}{1.465 + 1.098 + 1.048 + 1.207 + 1.151} = \frac{1.048}{5.969} \\ &\approx 0.175 \end{aligned}$$

$$\begin{aligned} \text{softmax}(y_4) &= \frac{1.207}{1.465 + 1.098 + 1.048 + 1.207 + 1.151} = \frac{1.207}{5.969} \\ &\approx 0.202 \end{aligned}$$

$$\text{softmax}(y_5) = \frac{1.151}{1.465 + 1.098 + 1.048 + 1.207 + 1.151} = \frac{1.151}{5.969} \\ \approx 0.192$$

Probabilitas untuk masing-masing kelas berdasarkan *output* model:

- a. Kelas 1 (*sad*) memiliki probabilitas sebesar 0.245
- b. Kelas 2 (*happy*) memiliki probabilitas sebesar 0.183
- c. Kelas 3 (*love*) memiliki probabilitas sebesar 0.175
- d. Kelas 4 (*fear*) memiliki probabilitas sebesar 0.202
- e. Kelas 5 (*angry*) memiliki probabilitas sebesar 0.192

Berdasarkan nilai *output* $h_4 = 0.470$ yang dihasilkan oleh model LSTM, diperoleh bahwa model memberikan probabilitas tertinggi pada kelas emosi *sad*, yaitu sekitar 24,5%. Hal ini menunjukkan bahwa model mengidentifikasi *tweet* tersebut mengandung emosi *sad* dibandingkan dengan emosi lainnya.

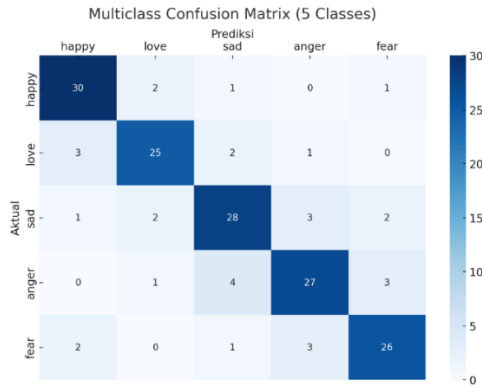
3.7 Rancangan Pengujian

3.7.1 Ilustrasi Pengujian

Evaluasi dilakukan untuk mengukur tingkat akurasi model yang telah dilatih dalam memprediksi data pada tahap pengujian. Proses evaluasi dalam analisis sentimen ini menggunakan beberapa metrik, yaitu: *Accuracy* yang dihitung berdasarkan Rumus (2.11), *Precision* berdasarkan Rumus (2.12), *Recall* berdasarkan Rumus (2.13), serta *F1-score* yang diperoleh melalui Rumus (2.14).

Asumsi:

Gambar 3.4 Inisialisasi Nilai Aktual dan Prediksi



Tabel 3.12 Inisialisasi Nilai Aktual dan Prediksi

Aktual	Prediksi					
	Happy	Love	Sad	Anger	Fear	Total Prediksi
Happy	30	2	1	0	1	34
Love	3	25	2	1	0	31
Sad	1	2	28	3	2	36
Anger	0	1	4	27	3	35
Fear	2	0	1	3	26	32
Total Aktual	36	30	36	34	32	168

Perhitungan tiap kelas:

1. Kelas: *Happy*

Dari total 34 data dengan label asli *happy*, sebanyak 30 data berhasil diklasifikasikan dengan benar (*True Positive*). Terdapat 4 kesalahan dalam

proses klasifikasi, yaitu 2 data diprediksi sebagai *love*, 1 sebagai *sad*, dan 1 sebagai *fear*.

$$TP = 30$$

$$FP = 3 \text{ (Love)} + 1 \text{ (Sad)} + 0 \text{ (Anger)} + 2 \text{ (Fear)} = 6$$

$$FN = 2 \text{ (Love)} + 1 \text{ (Sad)} + 0 \text{ (Anger)} + 1 \text{ (Fear)} = 4$$

$$Precision = \frac{TP}{TP + FP} = \frac{30}{30 + 6} = \frac{30}{36} = 0.833$$

$$Recall = \frac{TP}{TP + FN} = \frac{30}{30 + 4} = \frac{30}{34} = 0.882$$

$$F1-Score = 2 \times \frac{0.833 \times 0.882}{0.833 + 0.882} = 0.857$$

2. Kelas: *Love*

Pada kelas *Love*, nilai *True Positive* (TP) adalah 25, yang berarti sebanyak 25 data yang benar-benar termasuk kelas *Love* berhasil diprediksi dengan benar oleh model. Sementara itu, terdapat *False Positive* (FP) sebanyak 5, yang menunjukkan adanya lima data dari kelas lain yang secara keliru diprediksi sebagai *Love*. Nilai *False Negative* (FN) sebanyak 6, menunjukkan bahwa terdapat enam data yang seharusnya diklasifikasikan sebagai *Love*, namun diprediksi sebagai kelas lain oleh model.

Berdasarkan nilai-nilai tersebut, diperoleh nilai *precision* sebesar 0,833, yang mengindikasikan bahwa 83,3% dari semua prediksi *Love* adalah benar. Nilai *recall* sebesar 0,806 menunjukkan bahwa model berhasil mengidentifikasi 80,6% dari seluruh data kelas *Love* secara tepat. Nilai *F1-score*, yaitu rata-rata harmonis dari *precision* dan *recall*, berada pada angka 0,819.

3. Kelas: *Sad*

Pada kelas *Sad*, diperoleh *True Positive* (TP) sebanyak 28, menandakan bahwa sebanyak 28 data dari kelas *Sad* berhasil diklasifikasikan dengan benar. Jumlah *False Positive* (FP) sebanyak 8 menunjukkan jumlah data dari kelas lain yang secara salah diklasifikasikan sebagai *Sad*. Sementara itu, *False Negative* (FN) sebanyak 8 menunjukkan jumlah data kelas *Sad* yang tidak berhasil dikenali dan diprediksi sebagai kelas lain.

Nilai *precision* untuk kelas ini adalah 0,778, yang berarti bahwa 77,8% dari prediksi kelas *Sad* adalah benar. Nilai *recall* juga sebesar 0,778, yang menunjukkan bahwa model dapat mengenali 77,8% data yang benar-benar berasal dari kelas *Sad*. Karena *precision* dan *recall* memiliki nilai yang sama, *F1-score* untuk kelas ini adalah 0,778.

4. Kelas: *Anger*

Pada kelas *Anger*, model menghasilkan *True Positive* (TP) sebanyak 27, menunjukkan bahwa 27 data dari kelas ini diprediksi dengan tepat. *False Positive* (FP) sebanyak 7 berarti terdapat tujuh data dari kelas lain yang salah diklasifikasikan sebagai *Anger*. Nilai *False Negative* (FN) sebanyak 8 menunjukkan jumlah data kelas *Anger* yang tidak dikenali oleh model.

Dari hasil tersebut, diperoleh nilai *precision* sebesar 0,794, yang berarti dari seluruh prediksi kelas *Anger* sebanyak 79,4% benar. Nilai *recall* sebesar 0,771 menunjukkan bahwa 77,1% dari seluruh data kelas *Anger* berhasil dikenali secara tepat. Nilai *F1-score* untuk kelas ini adalah 0,782.

5. Kelas: *Fear*

Pada kelas *Fear*, diperoleh *True Positive* (TP) sebanyak 26. Ini berarti model mampu mengenali 26 data kelas *Fear* dengan tepat. Nilai *False Positive* (FP) sebanyak 6 menunjukkan adanya enam data dari kelas lain yang salah diklasifikasikan sebagai *Fear*. Terdapat *False Negative* (FN) sebanyak 6, yaitu data dari kelas *Fear* yang diprediksi menjadi kelas lain.

Dari perhitungan tersebut, nilai *precision* dan *recall* masing-masing adalah 0,8125. Nilai *precision* ini menunjukkan bahwa 81,25% dari semua prediksi kelas *Fear* adalah benar, sedangkan nilai *recall* menunjukkan bahwa 81,25% dari semua data yang benar-benar merupakan kelas *Fear* berhasil dikenali oleh model. Karena kedua nilai tersebut sama, maka F1-score juga sebesar 0,8125.

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F-1 Score</i>
<i>Happy</i>	0,833	0,882	0,857
<i>Love</i>	0,833	0,806	0,819
<i>Sad</i>	0,778	0,778	0,778
<i>Anger</i>	0,794	0,771	0,782
<i>Fear</i>	0,813	0,813	0,813
Rata-Rata	0,811	0,810	0,810

Perhitungan *Accuracy*:

Jumlah nilai TP dari keseluruhan sisi diagonal

= [Happy, Happy] + [Love, Love] + [Sad, Sad] + [Anger, Anger] + [Fear, Fear]

= 30 + 25 + 28 + 27 + 26 = 136

Jumlah keseluruhan total data pengujian = 168

$$Accuracy = \frac{Jumlah\ Prediksi\ Benar}{Total\ Prediksi}$$

$$Accuracy = \frac{30+25+28+27+26}{168} = \frac{136}{168} = 0,7556 = 75,56\%$$

Perhitungan *Macro Average*:

Diketahui:

$$Precision = 0,833 + 0,833 + 0,778 + 0,794 + 0,813$$

$$Recall = 0,882 + 0,806 + 0,778 + 0,771 + 0,813$$

$$F1 - Score = 0,857 + 0,819 + 0,778 + 0,782 + 0,813$$

1. *Macro Precision Average*

$$= \frac{1}{N} \sum_{i=1}^N Precision (C_i) = \frac{0,833+0,833+0,778+0,794+0,813}{5} = 0,811$$

2. *Macro Recall Average*

$$= \frac{1}{N} \sum_{i=1}^N Recall (C_i) = \frac{0,882+0,806+0,778+0,771+0,813}{5} = 0,810$$

3. *Macro F1-Score Average*

$$= \frac{1}{N} \sum_{i=1}^N F1 - Score (C_i) = \frac{0,857+0,819+0,778+0,782+0,813}{5} = 0,810$$

Nilai-nilai tersebut mengindikasikan bahwa model memiliki performa yang stabil dalam mengklasifikasikan data secara akurat, terutama dengan mempertimbangkan proporsi jumlah data pada masing-masing kelas. Metode *macro average* merupakan metode perhitungan metrik evaluasi (*Precision*, *Recall*, dan *F1-Score*) dengan cara mengambil rata-rata dari masing-masing kelas. Dalam pendekatan ini, jumlah data pada setiap kelas

tidak memengaruhi hasil akhir, sehingga baik kelas yang memiliki banyak data maupun yang sedikit tetap diberikan bobot yang setara.

Dengan nilai presisi dan *recall* yang identik sebesar 0,811 serta F1-*score* sebesar 0,810 yang hampir setara, dapat disimpulkan bahwa model mampu mempertahankan keseimbangan antara ketepatan dalam prediksi dan kemampuan untuk mengklasifikasikan seluruh kelas dengan proporsional. Hal ini menunjukkan bahwa model tidak hanya bekerja baik pada kelas yang dominan, tetapi juga memberikan perhatian yang cukup terhadap kelas lainnya sesuai dengan bobot kemunculannya.