

## ANALISIS *SELF-ATTENTION* PADA *BIDIRECTIONAL LSTM* DENGAN *FASTTEXT* DALAM MENDETEKSI EMOSI BERDASARKAN TEXT

Letare Saragih<sup>1</sup>, Maria Nababan<sup>2</sup>, Yohana Simatupang<sup>3</sup>, Junita Amalia<sup>4</sup>

<sup>1,2,3,4</sup>Institut Teknologi Del

<sup>1,2,3,4</sup>Program Studi Sistem Informasi Fakultas Informatika dan Teknik Elektro Institut Teknologi Del  
Jl. Sisingamangaraja, Sitoluama Laguboti, Toba Samosir Sumatera Utara, Indonesia 22381

Telp: +62 632 331234 Fax: +62 632 331116

E-Mail: [iss18064@students.del.ac.id](mailto:iss18064@students.del.ac.id)<sup>1</sup>, [iss18019@students.del.ac.id](mailto:iss18019@students.del.ac.id)<sup>2</sup>, [iss18018@students.de.ac.id](mailto:iss18018@students.de.ac.id)<sup>3</sup>,  
[junita.amalia@del.ac.id](mailto:junita.amalia@del.ac.id)<sup>4</sup>

### Abstrak

Cuitan Twitter yang sudah dilabeli berdasarkan jenis emosinya merupakan salah satu bentuk pengekspresian emosi dalam bentuk teks. Teks dapat dijadikan sebagai objek dalam melakukan emotion detection. Tujuan penelitian ini adalah untuk mengetahui pengaruh self-attention pada pemodelan Bi-LSTM dengan FastText dalam mendeteksi emosi pada cuitan Twitter. Pengaruh dilihat dengan membandingkan hasil evaluasi recall, precision, F1-score dan akurasi dari pemodelan Bi-LSTM, Bi-LSTM + Self-Attention dan Self-Attention. FastText digunakan untuk mengubah setiap kata menjadi vector matrix. Bi-LSTM digunakan untuk proses klasifikasi. Dan self-attention untuk membantu model untuk memilih kata yang paling dapat merepresentasikan makna dari kalimat terutama pada kalimat review yang panjang. Hasil yang diperoleh menunjukkan bahwa dari ketiga model, Bi-LSTM memiliki hasil evaluasi yang lebih baik dibandingkan dengan kedua model lainnya. Berdasarkan hasil tersebut, dapat disimpulkan bahwa penambahan self-attention pada model Bi-LSTM tidak memberikan pengaruh pada hasil evaluasi model untuk klasifikasi emosi.

**Kata kunci:** Deteksi Emosi, Word Embedding, FastText, Bi-LSTM, Self-Attention

### Abstract

Twitter tweets that have been labeled based on the type of emotion are a form of emotional expression in the form of text. Text can be used as an object in emotion detection. The purpose of this study was to determine the effect of self-attention on Bi-LSTM modeling with FastText in detecting emotions in Twitter tweets. The effect is seen by comparing the results of the evaluation of recall, precision, F1-score and accuracy of the Bi-LSTM, Bi-LSTM + Self-Attention and Self-Attention modeling. FastText is used to convert each word into a vector matrix. For the classification process, Bi-LSTM is used. And self-attention to help the model choose the word that best represents the meaning of the sentence, especially in a long review sentence. The results obtained indicate that of the three models, Bi-LSTM has a better evaluation result than the other two models. Based on these results, it can be concluded that the addition of self-attention to the Bi-LSTM model has no effect on the evaluation results of the model for emotional classification.

**Keywords:** Emotion Detection, Word Embedding, FastText, Bi-LSTM, Self-Attention.

## 1. PENDAHULUAN

Emosi dapat dipahami sebagai ungkapan seseorang untuk menyampaikan tanggapan maupun pendapat yang dirasakan. Emosi dapat disampaikan secara verbal seperti perkataan atau secara nonverbal seperti tindakan maupun tulisan [1]. Berdasarkan teori komputasi linguistik dan pendapat W. Gerrod Parrott pada bukunya yang berjudul “*Emotions in Social Psychology*”, secara garis besar deteksi emosi manusia dalam teks secara formal mengklasifikasi emosi manusia melalui hirarki emosi dalam enam kelas di tingkat dasar, yaitu: *Love, Joy, Anger, Sadness, Fear* dan *Surprise* [2]. Seiring perkembangan zaman dimana masyarakat sudah bergantung pada teknologi dalam kehidupan sehari-hari, bentuk penyampaian pendapat semakin berkembang. Begitu pula dengan sarana penyampaian pendapat yang semakin lama semakin bebas untuk digunakan yaitu melalui media sosial, salah satunya adalah Twitter [3]. Cuitan pengguna yang disampaikan pada Twitter merupakan salah satu bentuk pengekspresian emosi, pikiran dan perasaan pengguna namun dalam bentuk teks. Emosi yang ditunjukkan pengguna melalui cuitan dalam bentuk teks tersebut dapat dimanfaatkan untuk melihat pola emosi masyarakat dalam kehidupan sosial, sehingga dapat dimanfaatkan menjadi bahan pertimbangan dalam pengambilan keputusan (*decision making*). Dalam penelitian yang dilakukan oleh penulis, dataset yang digunakan bersumber dari dataset hasil gabungan Emotion DW dataset [4] dan RoBERTa dataset [5].

Dalam melakukan deteksi emosi pada teks, setiap kata pada cuitan Twitter perlu diubah menjadi vektor kata. Berdasarkan beberapa penelitian terdahulu seperti pada penelitian yang berjudul ‘Deteksi Emosi Media Sosial Menggunakan *Term Frequency- Inverse Document Frequency*’ [6] dan penelitian yang berjudul ‘*Emotion Detection in TWitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text*’ [7] mengenai perbandingan berbagai metode *word embedding* yang digunakan untuk menghasilkan representasi vektor kata dari teks, berdasarkan hasil yang didapatkan, *FastText* menunjukkan akurasi tertinggi dalam melakukan vektorisasi kata. Hal ini dikarenakan *FastText* menggunakan teknik *skip-gram* yang menghasilkan representasi n-banyak *output* kata dari satu kata yang ada, sehingga mampu mengatasi permasalahan *Out of Vocabulary* (OOV). OOV adalah keadaan dimana tidak ditemukannya suatu kata atau istilah selama proses *training* sehingga dapat mengurangi akurasi model. Vektor yang dihasilkan kemudian akan dijadikan sebagai inputan dalam pemodelan yang akan dibangun menggunakan Bi-LSTM dan *Self-Attention*.

Pemilihan Bi-LSTM sebagai model didasarkan pada beberapa hasil penelitian terdahulu seperti pada penelitian yang berjudul ‘Deteksi Depresi Dan Kecemasan Pengguna Twitter Menggunakan Bidirectional LSTM’ [8] dan penelitian yang berjudul ‘*A Deep Neural Network Model for the Detection and Classification of Emotions from Textual Content*’ [9] yang menunjukkan bahwa Bi-LSTM menghasilkan performa nilai akurasi yang lebih baik jika dibandingkan dengan metode lain dalam melakukan klasifikasi emosi. Adapun fokus dari penelitian ini adalah untuk melihat bagaimana pengaruh penambahan *layer Self-Attention* pada model dalam mendeteksi emosi. Penggunaan *Self-Attention* akan membantu model untuk memilih kata yang paling dapat merepresentasikan makna dari kalimat terutama pada kalimat yang panjang [10]. Analisis pengaruh penambahan *layer* akan dilakukan dengan membandingkan hasil evaluasi dari pemodelan menggunakan Bi-LSTM, pemodelan menggunakan Bi-LSTM + *Self-Attention* dan pemodelan menggunakan *Self-Attention*.

## 2. METODE PENELITIAN

Berikut ini dijelaskan mengenai dataset dan skenario yang digunakan pada pelaksanaan penelitian ini.

### 2.1 Data

#### a. Dataset Emotion DW

Dataset Emotion DW merupakan data yang berasal dari hasil klasifikasi emosi oleh data.world yang terdiri dari label emosi sebanyak 13 yaitu: *worry, neutral, sadness, happiness, love, surprise, fun, relief, hate, empty, enthusiasm, boredom* dan *anger* dengan konten yang disimpan pada atribut *tweet* sudah tersedia pada dataset. Data yang akan digunakan adalah kumpulan *tweet* dengan bahasa Inggris dengan jumlah 40.000 data. Pada dataset tersebut, tidak ditemukan *null value* namun

terdapat *duplicate value* pada table *content*. Atribut yang terdapat pada dataset terdiri dari 4 atribut yang ditampilkan pada Tabel 1.

**Tabel 1.** Daftar Atribut Dataset Emotion DW

Nama Atribut	Tipe Atribut	Deskripsi	Keterangan
<i>Tweet_id</i>	Numerik	Berisi <i>id tweet</i>	Tidak dibutuhkan
<i>Sentiment</i>	Kategorikal	Berisi label <i>emotion</i> dari cuitan	Dibutuhkan
<i>Author</i>	Kategorikal	Berisi <i>username</i> pengguna yang mengunggah cuitan	Tidak dibutuhkan
<i>Content</i>	Kategorikal	Berisi cuitan dalam bentuk teks yang memuat opini maupun konten yang diunggah pengguna di Twitter	Dibutuhkan

#### b. Dataset RoBERTa

Dataset RoBERTa merupakan dataset penelitian klasifikasi emosi dengan fine tuning RoBERTa yang terdiri dari 416.809 data *tweet* dengan enam label emosi. Karakteristik dataset ini yaitu tidak terdapat *null value* namun pada atribut text ditemukan adanya *duplicate value*, serta belum melalui proses penghapusan *stopword*, *stemming*, *tokenizer*, dan *lemmatization*. Adapun label emosi dari dataset ini terdiri dari *sadness*, *joy*, *fear*, *anger*, *love*, dan *surprise*. Atribut yang terdapat pada data yang diperoleh yakni terdiri dari 2 atribut yang ditampilkan pada Tabel 2.

**Tabel 2.** Daftar Atribut Dataset RoBERTa

Nama Atribut	Tipe Atribut	Deskripsi	Keterangan
<i>Text</i>	Objek	Berisi text dari pengguna <i>twitter</i> .	Dibutuhkan
<i>Emotions</i>	Kategorikal	Berisi label <i>emotion</i> dari cuitan	Dibutuhkan

## 2.2 Data Preprocessing

Tahapan *data preprocessing* pada kedua dataset yang digunakan bertujuan untuk memperoleh hasil data yang bersih untuk dijadikan sebagai inputan dalam penelitian deteksi emosi. Tahapan *data preprocessing* terdiri dari proses *data integration*, *data selection*, *data cleaning*, dan *data reduction*. *Data integration* dilakukan untuk mengintegrasikan kedua dataset menjadi satu dataset yang terdiri dari atribut *content* dan *sentiment*. Namun pada kedua dataset ini, data disimpan dalam atribut yang berbeda, sehingga perlu dilakukan *renaming* terhadap atribut untuk menyamakan nama atribut pada kedua dataset. *Data selection* dilakukan untuk pemilihan fitur yang sesuai dan dibutuhkan untuk penelitian yang dilakukan. Pada penelitian ini fitur atau atribut yang relevan adalah *content* dan *sentiment*. *Data cleaning* dilakukan untuk menghapus duplikat dan menghapus data yang bersifat null. Dan *data reduction* dilakukan untuk proses *feature subsetting* dimana akan dipilih beberapa label yang menjadi cakupan pada penelitian ini.

### 2.3 Text Preprocessing

*Text preprocessing* dapat dimanfaatkan untuk membersihkan dataset dari *noise*, misalnya dilakukan koreksi terhadap kesalahan ejaan, atau mengurangi karakter yang berulang [11]. Tahapan pengolahan text dengan teknik berikut: *case folding*, *remove URL*, *remove username*, *remove emoticon*, *remove number*, *remove punctuation*, *remove single char*, *remove special word*, *tokenizing*, *remove stopword*, dan *lemmatization*. *Case folding* merupakan salah satu heuristik yang digunakan untuk mengubah huruf yang bersifat *uppercase* (huruf besar) menjadi huruf *lowercase* (huruf kecil). *Remove URL* digunakan untuk menghapus URL pada teks. *Remove username* digunakan untuk dapat menghapus *username*. *Remove emoticon* digunakan untuk menghapus setiap emoji yang ditemukan dalam teks. *Remove number* digunakan untuk menghapus setiap nomor pada teks. *Remove punctuation* digunakan untuk menggantikan setiap simbol atau tanda baca menjadi char kosong. *Remove Single Char* digunakan untuk menggantikan *single char* menjadi char kosong. *Remove special word* digunakan untuk menghapus kata hasil *punctuation* yang tersisa dan tidak memiliki makna. *Tokenizing* digunakan untuk mengubah himpunan kata yaitu kalimat menjadi satuan kata atau dapat disebut sebagai token. *Remove stopword* digunakan untuk memilih kata atau token apa saja yang dapat mewakili informasi dari sebuah dokumen atau teks. Dan *lemmatization* digunakan untuk proses normalisasi pada kata (*word*) berdasarkan pada bentuk dasar yang merupakan bentuk *lemma*-nya.

### 2.4 FastText

*FastText* akan digunakan untuk mengubah setiap kata menjadi *vector* kata. Hal ini perlu dilakukan pada penelitian sebagai inputan dalam pemodelan klasifikasi dengan menggunakan Bi-LSTM. Adapun masukan kalimat yang diterima pada pembentukan *vector* kata adalah hasil dari *text preprocessing* yang ada dalam bentuk token. Jumlah representasi *vector* dari semua n-gram atau rata-ratanya akan mewakili kata aslinya [12]. Pada penelitian yang dilakukan, akan digunakan beberapa parameter yang dimiliki oleh gensim *library* guna menghasilkan *vector* yang tinggi. Adapun parameter tersebut ialah *vector size* dan *sg. vector size* yang digunakan secara *default* saat menggunakan *FastText* gensim adalah 100. *vector size* pada ukuran 100 hingga 300 memiliki performance yang baik dan dengan mempertimbangkan optimalisasi penggunaan memori komputer, peneliti menggunakan *vector size* 200. *sg* merupakan alternatif *training* yang digunakan oleh peneliti yaitu dengan nilai 1, dimana nilai 1 merepresentasikan alternatif *training* dari *skip-gram*.

### 2.5 Bi-LSTM

Metode Bi-LSTM akan digunakan untuk memeriksa setiap kata dari dua arah yaitu *forward* maupun *backward* [13]. Serangkaian proses Bi-LSTM akan dilakukan pada kedua *layer* yaitu *forward* yang memeriksa nilai berdasarkan kata pertama hingga kata terakhir dan *backward* yang memeriksa nilai berdasarkan kata terakhir terlebih dahulu ke kata pertama. Hasil dari kedua proses dari *forward* maupun *backward* yang selanjutnya dijumlahkan untuk mendapatkan nilai untuk setiap kata. Parameter yang digunakan, antara lain: *Embedding* untuk menampung hasil entri dari *FastText*, Bi-LSTM untuk menampung pendefinisian untuk model Bi-LSTM, *Dropout* digunakan untuk meminimalkan kemungkinan *overfitting* dan *Dense* digunakan untuk melakukan operasi untuk lapisan input.

### 2.6 Bi-LSTM + Self-Attention

Metode Bi-LSTM + *Self Attention* merupakan model yang dibangun dengan menggabungkan penggunaan metode Bi-LSTM dengan metode *Self-Attention*. Parameter yang digunakan, antara lain: *Embedding* digunakan untuk menampung hasil entri dari *FastText*, Bi-LSTM untuk menampung pendefinisian untuk model Bi-LSTM, *Self-Att* digunakan untuk pendefinisian model *Self-Attention*, *Dropout* digunakan untuk meminimalkan kemungkinan *overfitting*, *Flatten* digunakan untuk

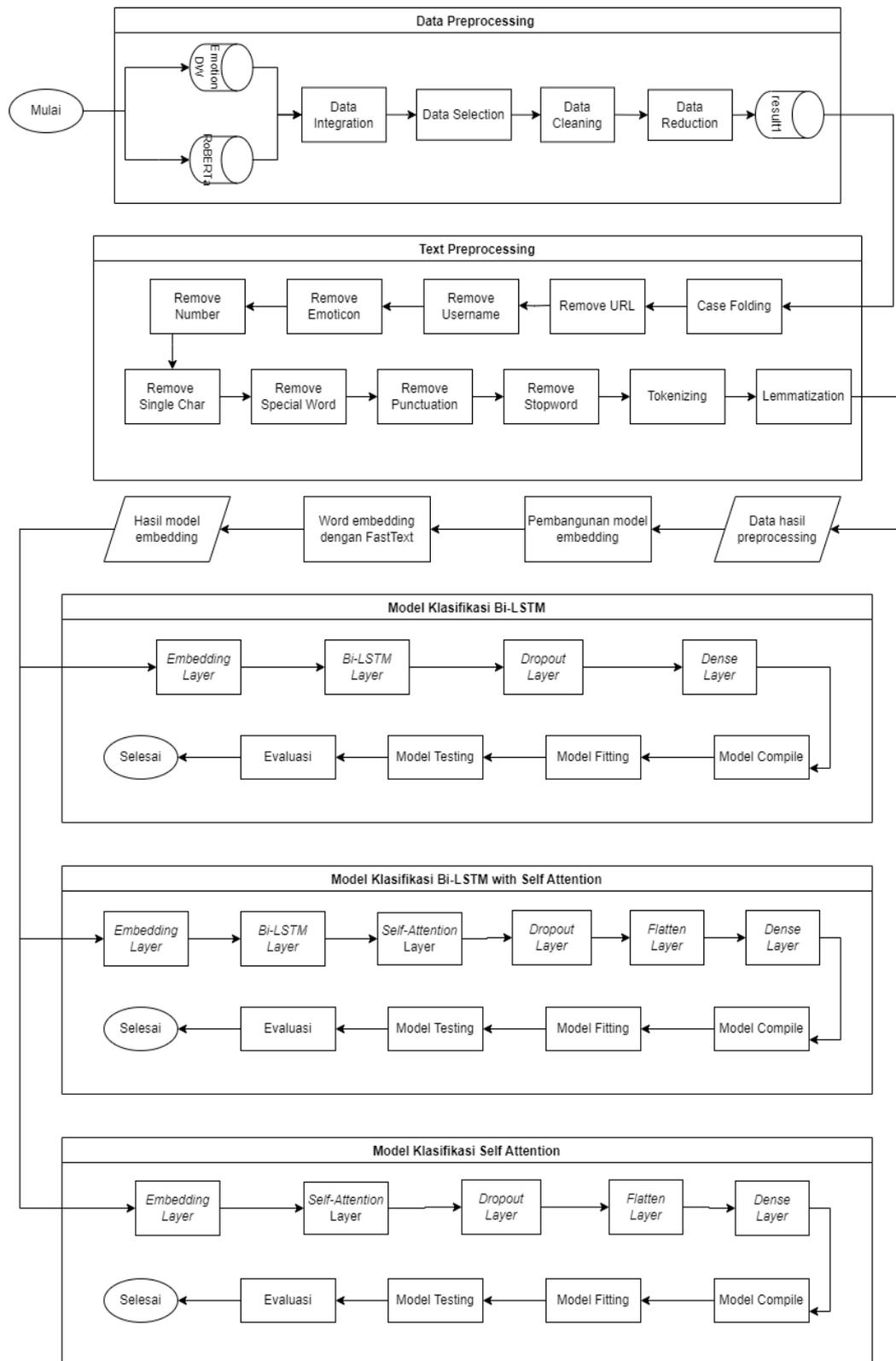
mengurangi input multi-dimensi menjadi satu dimensi, dan *Dense* untuk melakukan operasi untuk lapisan input.

## **2.7 Self-Attention**

Metode *self-attention* memungkinkan model untuk dapat memproses kata inputan sekaligus dan memodelkan hubungan antara semua kata dalam sebuah kalimat. *Self-attention* akan mempelajari hubungan antar satu kata dengan kata lainnya untuk melihat seberapa penting kata tersebut berpengaruh terhadap kata lain [14]. Parameter yang digunakan, antara lain: *Embedding* digunakan untuk menampung hasil entri dari *FastText*, *Self-Att* digunakan untuk pendefinisian model *Self-Attention*, *Dropout* digunakan untuk meminimalkan kemungkinan *overfitting*, *Flatten* digunakan untuk mengurangi input multi-dimensi menjadi satu dimensi, dan *Dense* untuk melakukan operasi untuk lapisan input.

## **2.8 Skenario Penelitian**

Skenario dari keseluruhan dari metode penelitian yang akan dilakukan dapat dilihat pada desain penelitian berikut:



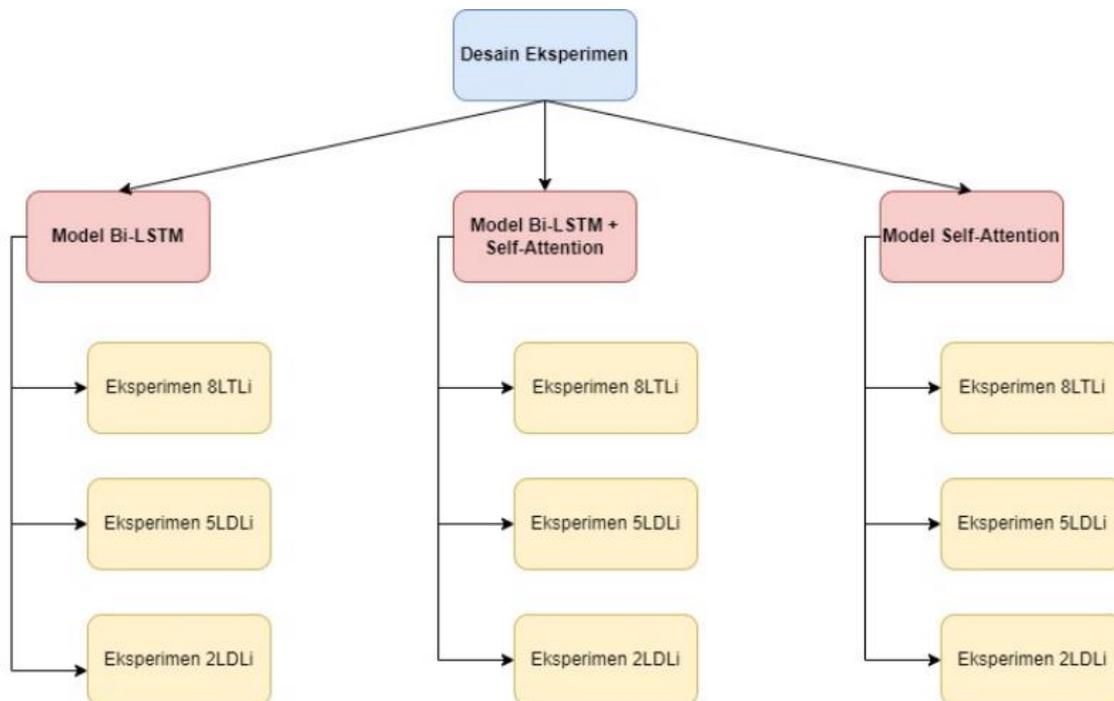
Gambar 1. Metode Penelitian

Terdapat 2 dataset yang digunakan yaitu Emotion DW yang memuat kumpulan data *tweet* berbahasa Inggris yang sudah berlabel dengan jumlah 40.000 *rows* dan RoBERTa yang memiliki 6 label dan 416.809 *rows*. Untuk dataset Emotion DW, terdapat 2 atribut yang digunakan, yaitu: atribut *content* dan atribut *sentiment*. Sementara untuk dataset RoBERTa terdapat 2 atribut yang digunakan,

yaitu: atribut *text* dan atribut *emotions*. Kedua data tersebut disimpan dalam ekstensi .csv dan diproses pada serangkaian tahapan *data preprocessing* dan dilanjutkan ke serangkaian tahapan *text preprocessing*. Hasil *preprocessing* kemudian akan menjadi masukan kepada tahap *word embedding* dengan *FastText* yang dijalankan dengan beberapa *hyperparameter default* seperti *sentences*, *vector size*, *window*, *workers*, *min\_count*, *iterasi*, *skip-gram* dan *hierarchical softmax*. Tahap klasifikasi dilakukan dengan menggunakan metode Bi-LSTM yang mempelajari semantik kata (kata sebelum dan kata sesudah) dan relasi antarkata yang berpengaruh dalam penentuan hasil akhir (deteksi emosi) dari kalimat yang sedang dipertimbangkan dan *Self-Attention* memilih kata yang paling dapat merepresentasikan makna dari kalimat terutama pada kalimat yang panjang.

Terdapat 3 versi klasifikasi yang dilakukan, yaitu: menggunakan Bi-LSTM, menggunakan Bi-LSTM + *Self-Attention* dan menggunakan *Self-Attention*. Setelah data diproses pada serangkaian *layer* pada klasifikasi ketiga model yang dibangun, masing-masing hasil model pada tahap model *compile* akan diperoleh hasil *summary* yang merupakan hasil dari representasi dari masing masing model yang dibuat. Hasil *summary* kemudian dieksekusi menggunakan model *fitting* untuk melihat seberapa baik model dalam menggeneralisasi data latih. Kemudian pada tahap model *testing* dilakukan pengujian model terhadap *data testing*. Tahap akhir adalah evaluasi terhadap sistem apakah sesuai dengan yang diharapkan. Hasil evaluasi diperoleh dengan menghitung nilai *confusion matrix* yang akan memvisualisasikan tingkat kebingungan algoritma pada beberapa kelas yang berbeda satu sama lain dan tidak tergantung pada algoritma klasifikasi konkret, dilihat dari nilai prediksi dan juga pada nilai actual [15] dimana nilainya terdiri dari perhitungan *accuracy*, *precision*, *recall* dan *F1-Score* yaitu nilai Hasil perhitungan akan berada pada rentang 0 sampai 1, dimana jika hasil nilai-nilai tersebut mendekati 1 maka klasifikasi tersebut dikatakan baik dan jika sebaliknya maka klasifikasi yang buruk.

Adapun rangkaian eksperimen yang akan dilakukan dapat dilihat pada gambar berikut:



**Gambar 2.** Desain Eksperimen

Pada 3 versi pemodelan yang dilakukan, masing-masing model akan diimplementasikan dengan 3 eksperimen berbeda seperti bagan desain eksperimen di atas. Berikut dijelaskan masing - masing eksperimen yang dilakukan:

1. Eksperimen 8LTli (8 Label Tanpa Limitasi)

Pada eksperimen 8LTLi (8 Label Tanpa Limitasi), klasifikasi emosi untuk setiap model akan menggunakan dataset yang telah dihasilkan melalui proses *data preprocessing* dan *text preprocessing* dengan total label 8 dan tanpa limitasi panjang kata.

2. Eksperimen 5LDLi (5 Label Dengan Limitasi)

Pada eksperimen 5LDLi (5 Label Dengan Limitasi), klasifikasi emosi untuk setiap model akan menggunakan dataset yang telah dihasilkan melalui proses *data preprocessing* dan *text preprocessing*. Selain itu pada eksperimen ini akan menggunakan 5 label yang dipilih berdasarkan hasil seleksi kata dengan panjang lebih atau sama dengan 15.

3. Eksperimen 2LDLi (2 Label Dengan Limitasi)

Pada eksperimen 2LDLi (2 Label Dengan Limitasi), klasifikasi emosi untuk setiap model akan menggunakan dataset yang telah dihasilkan melalui proses *data preprocessing* dan *text preprocessing*. Namun pada eksperimen ini akan menggunakan 2 label yang dipilih berdasarkan hasil seleksi kata dengan panjang lebih atau sama dengan 15.

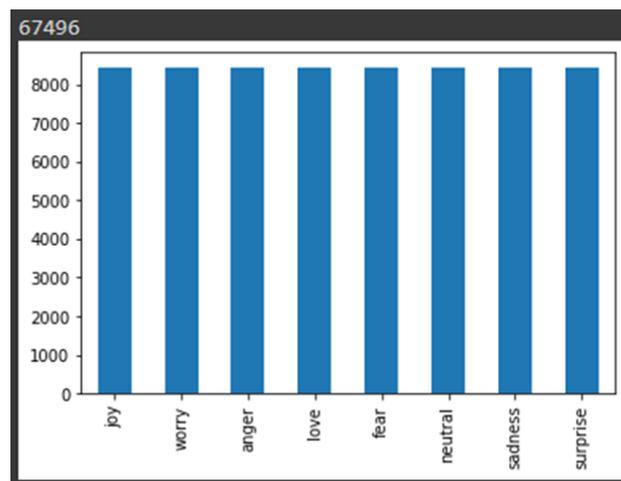
### 3. HASIL DAN PEMBAHASAN

#### 3.1 Hasil Penelitian

Berikut ini akan dijelaskan hasil dari setiap tahapan penelitian yang dilakukan.

##### 3.1.1 Data Preprocessing

Tahapan *data preprocessing* yang dilakukan adalah *data integration*, *data selection*, *data cleaning* dan *data reduction*. Hasil dari proses *data integration* masih berupa data asli yang berasal dari data sebelumnya dan belum melalui seleksi untuk data apa saja yang akan digunakan. Hasil yang diperoleh dari *data selection* belum bersih dan masih berupa data asli dari dataset awal yang hanya mengalami perubahan atribut. Hal ini dilanjutkan pada pengolahan data kembali untuk menghasilkan data yang bersih melalui tahapan *data cleaning*. Tahapan *data cleaning* dilakukan dengan terlebih dahulu memeriksa nilai yang hilang (*null*) dan hasil yang diperoleh adalah tidak terdapat data yang hilang pada kedua atribut yang ada di dataset. Data yang bersih dari nilai *null* selanjutnya melalui tahapan pemeriksaan nilai yang duplikat (*duplicate value*) dan hasil yang diperoleh menunjukkan bahwa terdapat nilai yang berulang pada atribut *content* yang kemudian akan dihapus. Data yang telah dibersihkan selanjutnya melalui proses *reduction*. Pengaruh dari proses ini adalah jumlah data yang berkurang dengan adanya penghapusan data '*sentiment*' yang tidak digunakan sehingga total data yang dimiliki adalah sebanyak 67.496. Hasil ini menunjukkan bahwa data *sentiment* yang dimiliki adalah seimbang (*balance dataset*) seperti yang ditunjukkan pada gambar 3.



Gambar 3. Visualisasi Jumlah Data untuk Setiap Kategori Emosi

##### 3.1.2 Text Preprocessing

Dari serangkaian tahapan *text preprocessing* yang dilakukan, berikut ini akan ditampilkan contoh hasil setiap tahapan *text preprocessing* yang dilakukan secara sekuensial.

Tabel 3. Hasil Text Preprocessing

Text Preprocessing	Hasil
Case Folding	why are money problems so bad these days
Remove URL	@mileycyrus - poor thing
Remove Username	ouch!! that looks so painful
Remove Emoticon	my wife took my last 20, no mcdonalds breakfast for me today
Remove Number	nobody will not talk me
Remove Punctuation	still here unfortunately
Remove Single Char	miss my puppy
Remove Special Word	tried that already
Tokenization	['ouch', 'that', 'looks', 'so', 'painful']
Stopword Removal	['feel', 'like', 'going', 'mad']
Lemmatization	['love', 'lesleys', 'idea', 'rule', 'word', 'rule', 'make', 'feel', 'rebellious']

### 3.1.3 Word Embedding

Data yang telah melalui proses *text preprocessing* selanjutnya melalui proses *word embedding* yang mengubah setiap kata menjadi representasi *vector*. Pada proses *word embedding*, digunakan *library FastText* dengan beberapa parameter yaitu *sentences*, *window size* dengan nilai 5, *min\_count* dengan nilai 5, *sg* bernilai 1 yang menunjukkan *penggunaan skip gram* dan *size* sebesar 200. Adapun hasil dari proses *word embedding* adalah nilai vektor untuk setiap kata yang dihasilkan dari proses sebelumnya yaitu *text preprocessing*.

Tabel 4. Hasil Word Embedding

Term	Vector Result
love	[2.17555955e-01, 7.23503530e-02, 6.15230203e-02, 1.03915088e-01, 4.84224083e-03, 6.24683321e-01, 2.59951174e-01, 2.41386950e-01, 5.30076995e-02, ...]
feel	[-0.18025161, 0.07817043, 0.23360524, -0.12686697, -0.10576086, -0.0130881, -0.10623472, -0.21278976, -0.10912774, ...]
today	[ 5.48940264e-02, -3.17040294e-01, -2.08223253e-01, -1.3832609e-01, -3.35311368e-02, -4.7222423e-02, 3.52269417e-04, -1.94431096e-01, -2.60763317e-01, ...]

Hasil dari proses *word embedding* merupakan nilai dari ukuran 200 vektor yang dibentuk oleh model *FastText*. Dari hasil vektor di atas kemudian diubah menjadi ukuran 2 vektor dan kemudian ukuran 1 vector. Kemudian hasil dari perubahan ukuran vektor dapat menunjukkan *similarity* dari setiap *term*. Vektor yang diperoleh untuk setiap kata kemudian akan dimanfaatkan dalam pembangunan model.

### 3.1.4 Pemodelan Bi-LSTM

Model pertama dilakukan dengan membangun model klasifikasi emosi menggunakan algoritma Bidirectional LSTM dengan penggunaan *layer* yang telah ditentukan yaitu *embedding layer*, *bi-lstm layer*, *dropout layer* dan *dense layer*. Adapun hasil yang diperoleh untuk setiap eksperimen pada pemodelan dengan Bi-LSTM adalah sebagai berikut.

**Tabel 5.** Hasil Pemodelan Bi-LSTM

<b>Eksperimen</b>	<b>Sentiment</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>	<b>Accuracy</b>
Eksperimen 8LTLi	<i>anger</i>	0.94	0.93	0.93	<b>0.79</b>
	<i>fear</i>	0.89	0.93	0.91	
	<i>joy</i>	0.91	0.93	0.92	
	<i>love</i>	0.88	0.83	0.86	
	<i>neutral</i>	0.52	0.54	0.53	
	<i>sadness</i>	0.90	0.90	0.90	
	<i>surprise</i>	0.77	0.82	0.79	
	<i>worry</i>	0.51	0.46	0.48	
Eksperimen 5LDLi	<i>sadness</i>	0.96	0.96	0.97	0,96
	<i>love</i>	0.96	0.96	0.96	
	<i>anger</i>	0.98	0.96	0.97	
	<i>joy</i>	0.86	0.91	0.88	
Eksperimen 2LDLi	<i>fear</i>	0.97	0.98	0.98	1
	<i>Joy</i>	1	1	1	
	<i>Sadness</i>	1	1	1	

### 3.1.5 Pemodelan Bi-LSTM + Self-Attention

Model kedua dilakukan dengan membangun model klasifikasi emosi menggunakan algoritma *Bidirectional LSTM* yang dikombinasikan dengan *self-attention*. Adapun hasil untuk ketiga eksperimen yang diperoleh dari pemodelan kedua ini adalah sebagai berikut.

**Tabel 6.** Hasil Pemodelan Bi-LSTM dengan *Self-Attention*

<b>Eksperimen</b>	<b>Sentiment</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 Score</b>	<b>Accuracy</b>
Eksperimen 8LTLi	<i>anger</i>	0.92	0.92	0.92	<b>0.79</b>
	<i>fear</i>	0.90	0.92	0.91	
	<i>joy</i>	0.92	0.89	0.90	
	<i>love</i>	0.81	0.85	0.83	
	<i>neutral</i>	0.52	0.54	0.53	
	<i>sadness</i>	0.87	0.89	0.88	
	<i>surprise</i>	0.77	0.83	0.80	
	<i>worry</i>	0.53	0.44	0.48	
Eksperimen 5LDLi	<i>sadness</i>	0.97	0.96	0.97	<b>0,96</b>
	<i>love</i>	0.97	0.96	0.96	

Eksperimen	Sentiment	Precision	Recall	F1 Score	Accuracy
Eksperimen 2LDLi	<i>anger</i>	0.97	0.97	0.97	<b>0,99</b>
	<i>joy</i>	0.88	0.89	0.89	
	<i>fear</i>	0.97	0.98	0.98	
	<i>Joy</i>	1	0.98	0.99	
	<i>Sadness</i>	0.98	1	0.99	

### 3.1.6. Pemodelan *Self-Attention*

Model eksperimen ketiga dilakukan dengan membangun model klasifikasi emosi menggunakan algoritma *self-attention*.

Tabel 7. Hasil Pemodelan *Self-Attention*

Eksperimen	Sentiment	Precision	Recall	F1 Score	Accuracy
Eksperimen 8LTLi	<i>anger</i>	0.93	0.87	0.90	<b>0,77</b>
	<i>fear</i>	0.91	0.87	0.89	
	<i>joy</i>	0.91	0.83	0.87	
	<i>love</i>	0.77	0.86	0.81	
	<i>neutral</i>	0.53	0.61	0.57	
	<i>sadness</i>	0.85	0.86	0.85	
	<i>surprise</i>	0.69	0.86	0.76	
	<i>worry</i>	0.57	0.39	0.46	
Eksperimen 5LDLi	<i>sadness</i>	0.9	0.91	0.91	<b>0,91</b>
	<i>love</i>	0.89	0.9	0.89	
	<i>anger</i>	0.91	0.95	0.93	
	<i>joy</i>	0.92	0.7	0.8	
Eksperimen 2LDLi	<i>fear</i>	0.91	0.93	0.92	<b>0,95</b>
	<i>Joy</i>	0.96	0.96	0.96	
	<i>Sadness</i>	0.95	0.94	0.95	

### 3.2 Pembahasan

Berdasarkan hasil implementasi pada di atas, model Bi-LSTM memiliki nilai *accuracy* tertinggi di setiap eksperimen yang telah diimplementasikan. Adapun nilai *accuracy* pada eksperimen 8LTLi sebesar 79%, eksperimen 5LDLi sebesar 99% dan eksperimen 2LDLi sebesar 100%. Kemudian, *Self-Attention* dan Bi-LSTM menghasilkan *accuracy* tertinggi kedua dengan hasil eksperimen secara berurutan sebesar 79%, 96% dan 99%. Sementara model *Self-Attention* menjadi model dengan tingkat *accuracy* terkecil di setiap eksperimen pada penelitian ini dengan nilai secara berurutan 77%, 91% dan 95%. Pada ketiga eksperimen, semakin sedikit jumlah label yang digunakan, maka semakin baik nilai *accuracy* yang dihasilkan. Hal ini menunjukkan bahwa *accuracy* dari setiap model juga dipengaruhi oleh banyaknya label yang digunakan, dimana semakin banyak label sentimen yang digunakan semakin kecil nilai *accuracy* model. Maka dari hasil ketiga eksperimen, model Bi-LSTM menjadi model terbaik berdasarkan hasil evaluasi *accuracy* sedangkan model *Self-Attention* menjadi model dengan nilai *accuracy* yang paling kecil untuk setiap eksperimen yang dilakukan. Sementara ntuk perbedaan nilai *precision*, *recall* dan *F1-Score* yang dihasilkan pada setiap label emosi

disebabkan oleh pengaruh jumlah panjang rata-rata dan maksimal kata pada label, sehingga tingkat keakuratan dari data yang dibutuhkan dengan nilai prediksi yang dihasilkan oleh model dan tingkat keberhasilan model dalam menemukan dan mengenali informasi mengenai karakteristik kata yang menggambarkan emosi selama proses *training* menjadi berbeda pada setiap label emosi.

#### 4. KESIMPULAN

Berdasarkan hasil perolehan *nilai recall*, *precision*, *F1-Score* dan akurasi pada setiap eksperimen disetiap model pada penelitian ini, diketahui bahwa setiap eksperimen menghasilkan nilai evaluasi yang baik dan penambahan *self-attention* pada model Bi-LSTM tidak memberikan pengaruh pada hasil evaluasi model untuk klasifikasi emosi. Hasil penelitian menunjukkan jika dataset yang belum bersih menyebabkan banyak data yang terbuang dan informasi penting yang dihilangkan pada tahap data dan teks preprocessing. Karena itu, jumlah panjang rata-rata dan maksimal kata pada label mengurangi keakuratan dari data yang dibutuhkan oleh model untuk menemukan dan mengenali informasi mengenai karakteristik kata yang menggambarkan emosi selama proses *training* menjadi tidak akurat. Untuk itu peneliti menyarankan pada peneliti selanjutnya untuk menggunakan dataset yang bersih dan menggunakan metode penyeimbangan dataset lain untuk mendapatkan hasil yang lebih maksimal. Atau menggunakan dataset dan metode yang sama namun diterapkan pada objek penelitian lain seperti *machine translation* untuk melihat perbandingan kinerja algoritma *self-attention* pada objek translator dibandingkan dengan objek klasifikasi.

#### Daftar Pustaka

- [1] A. Bagus, "Klasifikasi Emosi pada Teks dengan Menggunakan Metode Deep Learnin," *J. Ilm. Indones.*, vol. 06, 2021.
- [2] W. Parrot, "Emotions in Social Psychology," in *Emotions in Social Psychology*, Psychology Press, 2001.
- [3] H. Juwiantho *et al.*, "Sentiment Analysis Twitter Bahasa Indonesia Berbasis WORD2VEC Menggunakan Deep Convolutional Neural Network," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 1, pp. 181–188, 2020, doi: 10.25126/jtiik.202071758.
- [4] Microsoft's Cortana Intelligence Gallery, "Sentiment Analysis in Text summary," *CrowdFlower*, 2016. <https://data.world/crowdfLOWER/sentiment-analysis-in-text/workspace/project-summary?agentid=crowdfLOWER&datasetid=sentiment-analysis-in-text>
- [5] E. Saravia, "RoBERTa\_Fine\_Tuning\_Emotion\_classification." <https://colab.research.google.com/drive/1nwCE6b9PXIKhv2hvbqf1oZKIGkXMTi1X#scrollTo=t23zHggkEpc->
- [6] M. A. Riza and N. Charibaldi, "Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text," *Int. J. Artif. Intell. Robot.*, vol. 3, no. 1, pp. 15–26, 2021, doi: 10.25139/ijair.v3i1.3827.
- [7] M. Polignano, M. De Gemmis, P. Basile, and G. Semeraro, "A comparison of Word-Embeddings in Emotion Detection from Text using BiLSTM, CNN and Self-Attention," *ACM UMAP 2019 Adjun. - Adjun. Publ. 27th Conf. User Model. Adapt. Pers.*, no. June, pp. 63–68, 2019, doi: 10.1145/3314183.3324983.
- [8] K. S. Nugroho, I. Akbar, A. N. Suksmawati, and I. Istiadi, "Deteksi Depresi dan Kecemasan Pengguna Twitter," *4th Conf. Innov. Appl. Sci. Technol. (CIASTECH 2021)*, no. Ciastech, pp. 287–296, 2021.
- [9] M. Z. Asghar *et al.*, "A Deep Neural Network Model for the Detection and Classification of Emotions from Textual Content," *Complexity*, vol. 2022, 2022, doi: 10.1155/2022/8221121.
- [10] X. Zhao, "Book rating model based on self-attention and LSTM," *ACM Int. Conf. Proceeding Ser.*, pp. 153–156, 2020, doi: 10.1145/3449301.3449327.
- [11] M. Ahsan, M. Mahmud, P. Saha, K. Gupta, Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *technologies*, vol. 9, 2021.
- [12] P. N. D. L. C Wang, "A Comparative Study on Word Embeddings in Deep Learning for Text Classification," *NLPIR*, 2020.
- [13] C. A. S. D. B Kuyumcu, "An automated new approach in fast text classification (fastText): A case study for Turkish text classification without pre-processing". *NLPIR, 2019*

- [14] N. S. N. P. J. U. L. J. A. G. Ł. K. I. P. A Vaswani, " Attention Is All You Need," *31st Conference on Neural Information Processing Systems*, 2017.
- [15] X. Jia X. Jianfeng, Z. Yuanjian, M. Duoqian, "Three-way confusion matrix for classification: A measure driven view," *Information Sciences*, pp. 772-794, 2020



*ZONasi: Jurnal Sistem Informasi*  
is licensed under a [Creative Commons Attribution International \(CC BY-SA 4.0\)](https://creativecommons.org/licenses/by-sa/4.0/)