

# Perbandingan *CNN* dan *Bi-Lstm* pada Analisis Sentimen dan Emosi Masyarakat Indonesia Di Media Sosial Twitter Selama Pandemi Covid-19 yang Menggunakan Metode *Word2vec*

Mohamad Romli Firdaus Kamarula<sup>1</sup>, Naim Rochmawati<sup>2</sup>

<sup>1,2</sup>Jurusan Teknik Informatika, Universitas Negeri Surabaya

<sup>1</sup>[mohamad.18050@mhs.unesa.ac.id](mailto:mohamad.18050@mhs.unesa.ac.id)

<sup>2</sup>[naimrochmawati@unesa.ac.id](mailto:naimrochmawati@unesa.ac.id)

**Abstrak**—Pandemi Covid-19 hingga kini masih melanda Indonesia. Sudah banyak kebijakan yang diberlakukan oleh Pemerintah untuk mengatasi penyebaran Covid-19 di Indonesia. Beberapa kebijakan yang telah diberlakukan seperti PPKM, PSBB, penggunaan masker, mematuhi protokol kesehatan hingga pelaksanaan vaksinasi. Namun banyaknya varian Covid-19 membuat pandemi ini belum juga berakhir. Dengan banyaknya kebijakan dan peraturan pemerintah dalam menangani pandemi Covid-19, menimbulkan berbagai macam respon dari masyarakat. Pendapat ini banyak disampaikan melalui media sosial salah satunya *twitter*. Banyaknya respon dari masyarakat ini dapat kita kelompokkan berdasarkan sentimen polaritas dan emosinya. Sehingga kita mampu menilai sudut pandang masyarakat mengenai kebijakan pemerintah untuk penanganan Covid-19. Tujuan dilakukannya penelitian ini untuk mengetahui model yang memiliki kemampuan terbaik untuk melakukan proses analisis sentimen dan emosi menggunakan dataset unggahan media sosial *twitter*. Algoritma *CNN* dan *Bi-LSTM* diusulkan untuk digunakan dalam penelitian analisis sentimen dan emosi menggunakan metode *Word2Vec*. Analisis sentimen dan emosi ini menggunakan data sebanyak 58.984 yang didapatkan melalui proses crawling data unggahan *twitter* yang berjumlah 22.062 data dan pengunduhan dataset dari website *kaggle.com* dengan jumlah data sebanyak 36.922 data. Pada penelitian ini model algoritma *Bi-LSTM* mendapatkan tingkat akurasi sebesar 69,48% untuk analisis sentimen dan 84,36% untuk analisis emosi. Tingkat akurasi ini lebih tinggi dibandingkan dengan algoritma *CNN* yang menghasilkan tingkat akurasi sebesar 68,58% untuk analisis sentimen dan 84,21% untuk analisis emosi. Berdasarkan hasil tersebut dapat disimpulkan bahwa algoritma *Bi-LSTM* lebih baik dibandingkan algoritma *CNN* dalam penelitian analisis sentiment dan emosi menggunakan metode *Word2Vec*.

**Kata Kunci**— Analisis Sentimen dan Emosi, Covid-19 di Indonesia, *Bi-LSTM*, *CNN*, *Word2Vec*.

## I. PENDAHULUAN

Sudah dua tahun lebih pandemi Covid-19 melanda Indonesia. Hingga saat ini, pemerintah masih belum bisa memastikan kapan pandemi ini akan berakhir pemerintah telah melakukan beberapa kebijakan seperti PPKM (Pemberlakuan Pembatasan Kegiatan Masyarakat), PSBB (Pembatasan Sosial Berskala Besar) serta mewajibkan masyarakat untuk selalu patuh menggunakan protokol

kehatan dan melakukan vaksinasi. Namun banyaknya varian Covid-19 membuat pandemi di Indonesia masih belum berakhir. *Coronavirus* (COVID-19) adalah penyakit menular yang disebabkan oleh virus *SARS-CoV-2*. Virus ini dapat menyebar dari mulut atau hidung orang yang terinfeksi dalam partikel cair kecil ketika mereka batuk, bersin, berbicara, bernyanyi atau bernapas [1].

Selama masa pandemi melanda Indonesia, banyak sekali respon yang diberikan masyarakat terkait kebijakan yang diberlakukan oleh pemerintah. Sebagian besar masyarakat menyampaikan pendapat mereka melalui media sosial seperti *facebook*, *twitter*, *instagram* dll. *twitter* adalah sumber informasi yang berharga dan efektif, *twitter* memiliki hampir 350 juta pengguna aktif yang mengunggah sekitar 500 juta *tweet* per hari yang mencerminkan kehidupan sehari-hari pengguna [2]. Oleh karena itu, penelitian ini menggunakan data yang dikumpulkan dari unggahan pengguna di media sosial *twitter* untuk mengetahui respon yang diberikan pengguna selama pandemi covid-19 berlangsung.

Beberapa dekade terakhir telah dilakukan identifikasi keadaan emosional pengguna baik menggunakan suara, mimik wajah, gerakan dan tatapan mata, maupun deteksi teks. Hal ini dilakukan untuk memberikan indikasi interaksi yang jauh lebih baik antar manusia maupun teknologi komputer. Analisis emosi berkaitan erat dengan analisis sentimen. Analisis sentimen merupakan proses komputasi untuk melakukan klasifikasi terhadap data tekstual sesuai dengan sentimen yang dimiliki [3]. Analisis sentimen digunakan untuk mengekstraksi sebuah polaritas opini dari sebuah kata atau kalimat menjadi sebuah polaritas positif, negatif atau netral. Dengan melakukan analisis emosi, maka tidak hanya bentuk polaritas berupa positif, netral atau negatif saja yang mampu dihasilkan, namun juga bentuk emosional yang disampaikan oleh pengguna. Analisis emosi dapat digunakan untuk meningkatkan layanan pelanggan, sektor bisnis, pemasaran, dan juga sebagai ukuran kinerja media sosial.

Beberapa penelitian yang telah dilakukan mengenai analisis sentimen dan emosi, diantaranya adalah : Analisis Sentimen *Movie Review* menggunakan *Word2Vec* dan metode *LSTM Deep Learning* [3]. Penelitian ini dilakukan menggunakan metode *Word2Vec* dan *LSTM deep learning*

yang kemudian akan dilakukan *tuning parameter* untuk mengetahui akurasi yang dihasilkan. Dataset yang digunakan pada penelitian ini merupakan dataset dokumen *review movie* yang berasal dari penelitian Andrew et al (<http://ai.stanford.edu/~amaas/data/sentimen/>). Penelitian ini bertujuan untuk menguji tingkat akurasi dari metode yang digunakan untuk melakukan analisis sentimen terhadap dataset *review movie*. Selain itu dilakukan *tuning parameter* untuk meningkatkan akurasi yang dihasilkan dengan harapan tingkat keakurasian metode yang digunakan setidaknya lebih dari 80%.

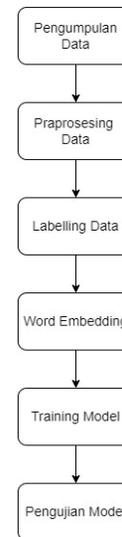
Penelitian dengan metode yang sama juga dilakukan oleh Siti Khomsah dengan judul *Sentimen Analysis On YouTube Comments Using Word2Vec and Random Forest* [4]. Penelitian ini bertujuan untuk mengetahui akurasi klasifikasi sentimen menggunakan Random Forest dengan ekstraksi fitur Word2Vec model Skip-gram. Word2Vec mencari kedekatan makna semantik sebuah kata, sehingga metode ini dianggap lebih baik dari pendekatan leksikal [4]. Penelitian ini menggunakan metode Random Forest dan Word2Vec model Skip-Gram. Dataset yang digunakan untuk penelitian ini adalah komentar-komentar di kanal YouTube yang menayangkan debat calon presiden Indonesia pada 2019.

Pada penelitian ini, algoritma *Convolution Neural Network* (CNN) dan *Bidirectional Long Short-Term Memory* (Bi-LSTM) diusulkan untuk digunakan dalam penelitian analisis sentimen dan emosi menggunakan metode *Word2Vec*. Algoritma LSTM memiliki keunggulan dimana algoritma ini mampu menyimpan sebuah nilai atau keadaan (*cell state*), baik untuk periode waktu yang panjang atau singkat [5]. Sedangkan Algoritma CNN menggunakan banyak konvolusi kernel dengan ukuran berbeda untuk mengekstrak informasi kunci teks, sehingga menangkap relevansi lokal dengan lebih baik [6]. Metode *Word2Vec* menerapkan cara kerja jaringan saraf tiruan (JST) untuk merepresentasikan fitur kata yang mempunyai tingkat kemiripan semantik yang tinggi. Ada dua arsitektur yang digunakan yaitu *continuous bag-of-words* (CBOW) dan *Skip-gram* [4]. Hasil akurasi yang diperoleh dengan metode *Skip-Gram* maupun CBOW untuk di beberapa variasi ukuran dimensi *word vector* cenderung stabil [3]. Dataset yang digunakan pada penelitian ini diambil dari unggahan respon pengguna sosial media twitter selama pandemi Covid-19 berlangsung. Pada penelitian Gayathri [6], keseluruhan metode yang disebutkan pada Tabel 3 kemudian dilakukan analisis dampak dari jumlah *layer* dan pengukuran waktu *training*. Hasil dari penelitian tersebut menunjukkan bahwa metode Bi-LSTM menghasilkan akurasi tertinggi yaitu 83.5%. Kemudian pada penelitian Hongjie [7], Grafik di atas (Gambar. 5) menggambarkan model dapat mengekstrak informasi lokal dari kata-kata berturut-turut dalam kalimat. Selain itu, model dapat menangkap informasi semantik dari konteks kalimat, yang memberikan permainan penuh untuk masing-masing keunggulan CNN dan BiLSTM. Hasil penelitian tersebut menunjukkan bahwa metode CNN dan Bi-LSTM menghasilkan tingkat akurasi sebesar 92.73%. Oleh karena itu, penulis melakukan penelitian dengan judul “*Perbandingan CNN dan Bi-LSTM Pada Analisis Sentimen dan Emosi Masyarakat Indonesia di Media Sosial Twitter Selama*

*Pandemi Covid-19 Yang Menggunakan Metode Word2Vec*”. Dengan dilakukannya penelitian ini, penulis berharap penelitian ini dapat bermanfaat bagi pihak yang membutuhkan informasi tentang hasil dari analisis sentimen ini.

## II. METODE PENELITIAN

Program analisis sentiment dan emosi ini dibangun berbasis aplikasi *console* menggunakan bahasa python. Pada penelitian ini digunakan aplikasi web *google colabs* untuk menjalankan program tersebut. Pada bab ini secara khusus menerangkan alur penelitian yang dilakukan menggunakan algoritma CNN dan Bi-LSTM menggunakan metode Word2Vec. Alur sistem dari penelitian ini ditunjukkan pada gambar 1.



Gbr 1. Alur Sistem Penelitian

### A. Dataset

Dataset yang digunakan dalam makalah ini adalah unggahan pengguna sosial media twitter di Indonesia tentang Covid-19. Data unggahan tersebut didapatkan melalui proses *crawling* data unggahan twitter dan dataset yang didapat dari *website kaggle.com*. Data yang didapat melalui *crawling* data menggunakan *library twint* dengan menggunakan kata kunci covid19 atau ppcm atau psbb atau vaksin menggunakan bahasa Indonesia. Data yang terkumpul melalui proses *crawling* sejumlah 22.062 data unggahan. Kemudian data selanjutnya diambil dari dataset yang disediakan oleh *website kaggle*. Dataset tersebut berisi *tweet* bahasa Indonesia dengan kata kunci “corona dan Pemerintah” atau covid dan Pemerintah” dari bulan Mei hingga Juli 2020. Kedua data tersebut digabung menjadi satu dataset untuk digunakan pada penelitian ini.

### B. Praprosesing

Dataset yang telah dikumpulkan, selanjutnya dilakukan tahap praprosesing data. Tahapan praprosesing data pada penelitian ini terdiri dari *cleaning* teks, *case folding*, *tokenizing*, *filtering*, *to sentence* dan *drop\_duplicate*.

### 1) Cleaning Teks

Tahapan *cleaning* teks dilakukan untuk menghilangkan beberapa kata atau karakter yang tidak dibutuhkan seperti mention (@), hastag (#), link dan angka. Selain itu baris baru pada setiap data akan dijadikan spasi, menghapus karakter spasi pada kiri dan kanan teks dan menghilangkan semua tanda baca.

### 2) Case Folding

Tahap selanjutnya adalah *case folding*. Pada tahapan ini data *inputan* yang semula berbentuk *sentence case* atau memiliki huruf kapital diubah menjadi bentuk *lowercase*. Hal ini dilakukan untuk memudahkan dalam proses pencarian atau klasifikasi.

### 3) Tokenizing

Setelah semua kata diubah menjadi *lowercase*, selanjutnya dilakukan proses *tokenizing*. Proses ini dilakukan untuk memisahkan deretan kalimat menjadi token atau potongan kata tunggal dengan menggunakan *word\_tokenize* yang dimiliki oleh *library* nltk. Kata, angka, simbol, tanda baca, dan entitas penting lainnya dapat dianggap sebagai token. Dalam NLP, token didefinisikan sebagai "kata", meskipun *tokenize* juga dapat dilakukan dalam paragraf dan kalimat.

### 4) Filtering

Selanjutnya dilakukan proses *filtering* untuk mengambil kata-kata penting dan menghilangkan kata-kata yang tidak penting dari hasil *tokenizing* sebelumnya seperti kata "di", "yang", "dan", "dari" dan lain-lain. Makna dibalik penggunaan stopwords adalah untuk menghilangkan kata-kata yang memiliki informasi rendah dari sebuah teks, sehingga kita dapat fokus pada kata-kata penting saja.

### 5) To Sentence

Setelah data *difilter* dari kata-kata yang tidak penting, selanjutnya potongan kata tersebut kemudian disatukan kembali menjadi sebuah kalimat.

### 6) Drop Duplicates

Terakhir, dataset yang telah melalui tahap praprosesing kemudian disaring kembali untuk menghapus beberapa data yang sama atau data *tweets* yang sama. Sehingga tidak ada data yang *redundan* pada dataset yang akan digunakan untuk penelitian ini.

### C. Analisis Data

Setelah dataset melalui tahap praprosesing, selanjutnya akan dilakukan proses analisis data. Pada proses ini dataset akan dilakukan *labelling* data untuk menentukan polaritas sentimen serta emosi dari dataset tersebut. Untuk mendapatkan polaritas sentimen dari data tersebut, dilakukanlah proses *labelling* menggunakan "*Indonesia Sentiment Lexicon*". Leksikon sentimen ini dibangun pada tahun 2017 dan memanfaatkan aliran data Twitter sekitar November 2016. Data dikumpulkan selama tiga hari dan disaring dengan Bahasa Indonesia (Bahasa Indonesia) dan dua macam emoticon yang mengekspresikan positif dan negatif polaritas. Kami mengelompokkan tweet menjadi positif dan negatif diatur dengan mengikuti karya Go

et al. di mana mereka memanfaatkan emoticon untuk membubuhi keterangan tweet dengan label sentimen setiap kata pada data tweet akan menentukan nilai polaritas yang kemudian nilai tersebut akan dihitung untuk mengetahui sentimen dari data tersebut [8].

Setelah polaritas sentimen didapatkan, selanjutnya menentukan nilai emosi dari data tersebut. Proses penentuan nilai emosi dari data tersebut menggunakan "*Pengklasifikasi emosi Indo RoBERTa*". *Indo RoBERTa Emotion Classifier* adalah *emotion classifier* yang didasarkan pada model *indoroberta*. *Indo RoBERTa* dilatih menggunakan dataset *IndoNLU EMOt*. Berdasarkan *benchmark* *IndoNLU*, model ini mampu melakukan pengklasifikasian dengan akurasi 71,81%. Selanjutnya dataset dibagi menjadi data *training* dan data *testing* dengan proporsi data *training* 67% dan data *testing* 33% [9].

### D. Word Embedding

Setelah tahap *labelling* data untuk menentukan polaritas sentimen dan emosi selesai, selanjutnya dataset akan melalui proses *word embedding*. Pada penelitian ini proses *word embedding* menggunakan metode *Word2Vec* dengan ukuran dimensi 300. Metode ini digunakan untuk menemukan hubungan tersembunyi pada setiap kata. Setiap kata mewakili distribusi bobot pada elemennya. Metode ini memiliki 2 model arsitektur, yaitu *Continous Bag-of-Words* (CBOW) dan *Skip-Gram*. Model CBOW bertujuan untuk memprediksi target kata dari konteks kata, sedangkan model *Skip-Gram* bertujuan untuk memprediksi probabilitas kata yang dapat menjadi konteks kata dari target kata yang ada [10].

Pada tahap ini *input* berupa teks atau *string* akan diubah menjadi angka atau *vektor*. Dengan cara ini *input* dapat diproses oleh arsitektur pembelajaran yang mendalam. *Input* harus memiliki bentuk dan ukuran yang sama, kecuali harus berupa angka atau *vektor*. Saat memproses *inputan* yang berbentuk data urutan sangat umum data tersebut memiliki panjang yang berbeda. Oleh karena itu, kita membutuhkan proses *padding* menggunakan *pad\_sequences* Keras. *Padding* merupakan bentuk khusus *masking* dimana proses *masking* tersebut dapat dilakukan di awal atau di akhir urutan. Kita juga dapat menentukan jumlah maksimum kata per kalimat. Sebelum melakukan proses pengisian, kalimat *input* perlu di-tokenisasi menggunakan *tool tokenizer* Keras.

### E. Training Model

Pada tahapan ini dataset yang sudah melalui proses *word embedding* selanjutnya akan dilatih pada model algoritma CNN dan BiLSTM. Pelatihan ini memiliki ukuran dimensi sebesar 300. Dataset yang sebelumnya sudah dibagi menjadi data *training* dan data test, selanjutnya dilakukan proses tokenisasi untuk menyematkan nilai *index* pada setiap kata. Selanjutnya data tersebut diproses melalui *pad\_sequence* dari *library* keras untuk menyamakan ukuran panjang *padding* pada setiap *array* kata. Selanjutnya data tersebut dijadikan nilai *inputan* untuk diproses pada *embedding layer* dengan ukuran dimensi 300 dan maksimal *inputan* kata sebesar 500.

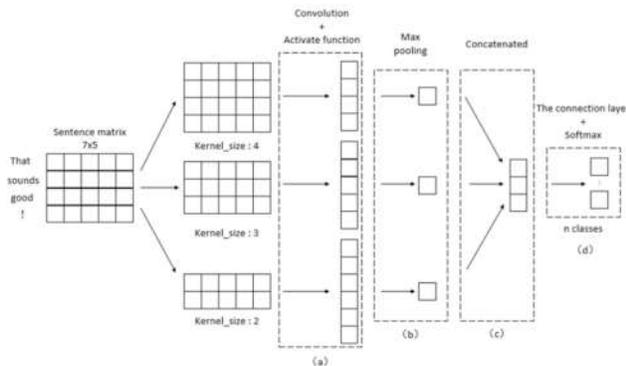
1) Training Data Dengan Model CNN

Setelah *inputan* melalui proses *embedding layer*, *inputan* dimasukkan pada konvolusi kernel satu dimensi. Proses ini menggunakan Conv1D dari *library* keras dengan nilai *filter output* sebesar 256 dan nilai *kernel size* sebesar 3. Setelah itu data *inputan* masuk pada proses pemilihan nilai maksimum dari matriks yang telah ditentukan untuk data temporal 1D menggunakan GlobalMaxPooling1D dari *library* keras. Setelah itu data *inputan* diproses pada *dense layer* untuk mengklasifikasikan data berdasarkan *output* dari lapisan konvolusional.

CNN pertama kali diterapkan pada *computer vision*. Kim [7] mengusulkan TeksCNN, metode pertama untuk menerapkan algoritma CNN untuk tugas klasifikasi teks. TeksCNN menggunakan beberapa kernel konvolusi dengan ukuran berbeda untuk mengekstrak informasi kunci tekstual untuk menangkap korelasi lokal dengan lebih baik.

Seperti terlihat pada gambar 2, model CNN ini masih terdiri dari kombinasi lapisan *convolutional*, *pooling*, dan *fully connected*. Untuk teks input, pertama, ubah menjadi matriks kalimat  $S \in R^{n \times d}$ , n adalah jumlah kata dalam teks, dan d menunjukkan dimensi vektor penyisipan kata dari setiap kata, yang dapat dianalogikan dengan titik piksel asli pada Gambar. Kemudian matriks kalimat digulung oleh beberapa kernel konvolusi  $W \in R^{h \times d}$  dan output diaktifkan oleh fungsi ReLU, dengan h menunjukkan ukuran kernel. *Vektor* pemetaan fitur yang diperoleh dengan konvolusi dimasukkan ke dalam lapisan penyatuan maksimum untuk mendapatkan fitur tingkat yang lebih tinggi, sementara teks dengan panjang yang berbeda dikonversi menjadi panjang tetap.

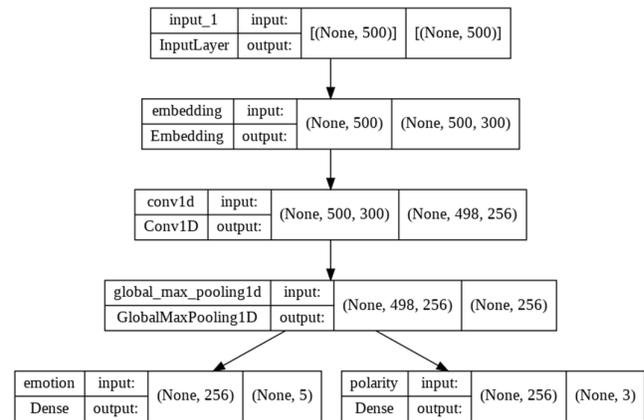
Pada gambar 2 (a)-(d) menunjukkan bahwa matriks kalimat masukan dilewatkan melalui beberapa *filter* dengan ukuran yang berbeda untuk mendapatkan fitur yang berbeda, dan operasi max-pooling mengikuti fitur yang sesuai untuk mendapatkan fitur yang paling penting untuk setiap dimensi. Akhirnya, fitur-fitur ini digabungkan dan diteruskan ke lapisan *Softmax* dari lapisan yang terhubung penuh untuk menampilkan distribusi probabilitas pada label.



Gbr 2. Model CNN untuk Klasifikasi Teks

Sumber : <https://doi.org/10.1016/j.procs.2022.01.092>

Bentuk alur arsitektur model algoritma CNN pada penelitian analisis sentimen dan emosi ini dapat dilihat pada gambar 3 dibawah ini.

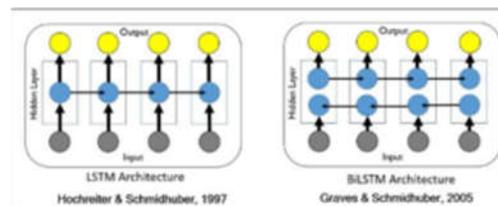


Gbr 3. Alur Arsitektur Model Algoritma CNN untuk Analisis Sentimen dan Emosi

2) Training Data Menggunakan Model Bi-LSTM

Pada proses *training* algoritma Bi-LSTM setelah melalui *embedding layer*, setiap *inputan* kata dimasukkan ke dalam unit Bi-LSTM satu per satu sesuai dengan urutan index dari *sequence*. Setelah itu data *inputan* diproses pada *dense layer* untuk mengklasifikasikan data berdasarkan *output* dari lapisan konvolusional.

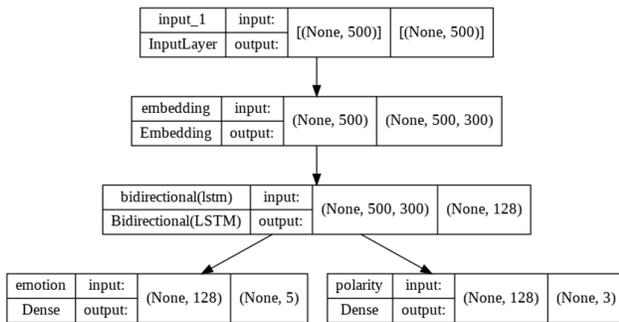
Arsitektur BiLSTM terdiri dari LSTM maju (*forward*) dan LSTM mundur (*backward*). BiLSTM dapat menyesuaikan data bolak-balik dan kemudian menggabungkan prediksi. Maju (*forward*) dan mundur (*backward*) di BiLSTM dapat meningkatkan jumlah informasi yang tersedia untuk jaringan dan konteks yang tersedia untuk algoritma, seperti mengetahui kata mana yang segera mengikuti dan mendahului kata dalam sebuah kalimat. Untuk memahami sebuah kata dalam bidang *Natural Language Processing* (NLP), terkadang Anda tidak hanya membutuhkan kata sebelumnya, tetapi juga kata yang akan datang. Perbedaan arsitektu pada model LSTM dan Bi-LSTM dapat dilihat pada gambar 4 [11].



Gbr 4. Arsitektur Algoritma LSTM dan Bi-LSTM

Sumber : <https://doi.org/10.52985/insyt.v3i1.148>

Bentuk alur arsitektur model algoritma Bi-LSTM untuk analisis sentimen dan emosi pada penelitian ini dapat dilihat pada gambar 5 dibawah ini.



Gbr 5. Alur Arsitektur Model Algoritma Bi-LSTM untuk Analisis Sentimen dan Emosi

### F. Testing Model

Setelah dilakukan tahap *training* data menggunakan model algoritma CNN dan Bi-LSTM, selanjutnya dilakukanlah tahap *testing* data. Dataset yang sebelumnya telah dibagi menjadi data *training* dan data *testing* dimana data testing memiliki jumlah data 33% dari dataset utuh akan dimasukkan pada proses *testing* model. Proses testing ini menggunakan jumlah *batch\_size* sebesar 256. Dalam artian proses testing ini akan menggunakan 256 sampel data pertama untuk dimasukkan ke dalam algoritma CNN dan Bi-LSTM. Selain itu proses ini menggunakan jumlah *epoch* sebanyak 15 untuk melakukan proses *testing*.

## III. HASIL DAN PEMBAHASAN

Pada penelitian analisis sentimen dan emosi masyarakat Indonesia di media sosial twitter selama pandemi Covid-19 ini menggunakan dua model algoritma yaitu CNN dan Bi-LSTM. Tujuannya untuk mengetahui hasil terbaik dari perbandingan metode CNN dan Bi-LSTM dalam kasus analisis sentimen dan emosi serta memberikan label polaritas sentimen dan emosi pada unggahan masyarakat Indonesia mengenai pandemi Covid-19 yang berlangsung di Indonesia.

Pada bab ini akan membahas hasil dari setiap tahapan yang dilakukan dalam penelitian analisis sentiment dan emosi masyarakat Indonesia di media sosial twitter selama masa pandemi Covid-19.

### A. Pengumpulan Dataset

Seperti yang telah dibahas pada subbab metodologi penelitian bahwa dataset yang digunakan pada penelitian ini berasal dari dua data. Dataset pertama diambil dari hasil *crawling* data dari unggahan twitter menggunakan library *twint*. Beberapa parameter yang digunakan yaitu :

- 1) Kata kunci : “psbb atau ppkm atau covid atau covid19 atau vaksin atau vaksin booster”
- 2) Bahasa : bahasa Indonesia (lang : id)

- 3) Rentan waktu : mulai 01-11-2019 hingga 15-06-2022 (31 bulan 15 hari)
- 4) *Limit* data : 35000

Data yang didapatkan pada proses *crawling* tersebut sejumlah 22.062 data. Dataset kedua diambil dari website [www.kaggle.com](http://www.kaggle.com) dimana dataset tersebut berisi data tweet dari masyarakat Indonesia di media social twitter selama masa pandemi Covid-19 dimana rentan tanggal data yang diambil mulai dari tanggal 01 Mei 2020 hingga tanggal 22 Juli 2020. Jumlah data yang dihasilkan dari rentan tanggal tersebut adalah sejumlah 36.922 data.

Selanjutnya kedua dataset tersebut digabung menjadi satu dan dilakukan tahap praprosesing dan *labelling* data. Setelah dataset melalui tahap praprosesing dan *labelling* data, selanjutnya dataset dibagi menjadi dua bagian dimana 67% dari data tersebut akan digunakan untuk data *training*, dan 33% dari data tersebut digunakan untuk data *testing*. Jadi jumlah keseluruhan data yang digunakan pada penelitian ini adalah sejumlah 58.978

### B. Hasil Praprosesing

Tahapan praprosesing data dilakukan untuk mendapatkan data yang bersih dan sesuai dengan kebutuhan input data untuk proses pengujian data sehingga mendapatkan hasil yang maksimal. Beberapa tahapan praprosesing yang dilakukan adalah *cleaning* teks, *case folding*, *tokenizing*, *filtering* dan *to sentence*. Setelah melalui tahap praprosesing dataset kemudian disaring kembali untuk menghapus data yang sama agar tidak ada data *tweets* yang *redundan*. Berikut ini dapat dilihat pada gambar 6 yang merupakan sampel data yang telah melalui tahap praprosesing.

```

51000 klinik eh lupa sebelah medifeet dominos tu yee...
51001 pemerintah dukung penuh umkm pandemi covid buk...
51002 pesakit elakkan sentuhan fizikal orang kena ca...
51003 kaya bisnis pcr vaksin ya om menggiurkan 😊
51004 vaksin logika sekolah dasar
51005 vaksin chicken pox x wajib rasenye kecil mcm x...
51006 covid data sebaran
51007 naiknya elektabilitas anies ya bang memicu tre...
51008 karanganyar sertu sujamil anggota koramil tsm ...
51009 knapa covid daun lagi👁
Name: data_text, dtype: object
  
```

Gbr 6. Sampel Data Hasil Praprosesing

### C. Hasil Labelling

Dataset yang telah melalui tahap praprosesing, selanjutnya dilakukan *labelling* data untuk menentukan nilai polaritas sentimen dan emosi dari setiap data tweet tersebut. Proses *labelling* dilakukan secara terpisah antara *labelling* polaritas sentiment dan *labelling* emosi. *Labelling* polaritas sentiment dilakukan menggunakan “Indonesia Sentiment Lexicon”. Proses ini akan menghasilkan nilai polaritas *positive*, *negative* atau *neutral* untuk menunjukkan nilai polaritas sentimen dari data tersebut. Berikut ini dapat dilihat pada Gambar 7 yang menunjukkan hasil proses *labelling* polaritas sentimen menggunakan “Indonesia Sentiment Lexicon”. Fajri Koto

mengatakan pada penelitiannya[8], *InSet Lexicon* terdiri dari 3.609 kata positif dan 6.609 kata negatif dalam bahasa Indonesia. Setiap kata diberi label secara manual berdasarkan polaritasnya dan kemudian ditingkatkan dengan menambahkan himpunan *stemming* dan sinonim. *Leksikon* sentimen dibangun untuk analisis sentimen bahasa Indonesia yang lebih cocok untuk *mikroblog*. Pendekatan ini menunjukkan bahwa hasilnya mengungguli semua metode dasar yang ada di mana akurasi tertinggi adalah 65,78%.

	data_text	text_preprocessed	polarity_score	polarity
51000	klirik eh lupa sebelah medifeet dominos tu yee...	[klirik, eh, lupa, sebelah, medifeet, dominos, ...]	-9	negative
51002	pesakit elakkan sentuhan fizikal orang kena ca...	[pesakit, elakkan, sentuhan, fizikal, orang, k...]	-14	negative
51003	kaya bisnis pcr vaksin ya om menggijarka...	[kaya, bisnis, pcr, vaksin, ya, om, menggijarka...]	3	positive
51004	vaksin logika sekolah dasar	[vaksin, logika, sekolah, dasar]	3	positive
51005	vaksin chicken pox x wajib rasenye kecil mom x...	[vaksin, chicken, pox, x, wajib, rasenye, kecil...]	-2	negative
51007	naknya elektabilitas anies ya bang memicu tre...	[naknya, elektabilitas, anies, ya, bang, memi...]	-2	negative
51008	karanganyar serti sujamal anggota koramil tm...	[karanganyar, serti, sujamal, anggota, koramil...]	0	neutral
51009	knapa covid daun lagi?	[knapa, covid, daun, lagi?]	0	neutral

Gbr 7. Sampel Data Hasil *Labelling* Polaritas Sentimen

Dari gambar 7 diatas dapat kita lihat hasil dari proses *labelling* polaritas sentimen menggunakan "*Indonesia Sentiment Lexicon*" menghasilkan 30.078 data dengan nilai *negative*, 19162 data dengan nilai *positive* dan 6675 data dengan nilai *neutral*. Selanjutnya data yang sudah melalui proses *labelling* polaritas sentimen dilakukan proses *labelling* emosi. Proses ini menggunakan "*Pengklasifikasi emosi Indo RoBERTa*" yang didasarkan pada model indo-roberta. Indo RoBERTa dilatih menggunakan dataset IndoNLU EMoT. Label emosi yang dihasilkan dari pengklasifikasian ini adalah *sadness*, *anger*, *love*, *fear* dan *happy*. Model pengklasifikasian ini memiliki tingkat akurasi sebesar 71,81% [9]. Berikut ini dapat dilihat hasil pengklasifikasian atau *labelling* emosi pada Gambar 8 dibawah ini.

	conversation_id	username	date	text_clean	text_preprocessed	polarity_score	polarity	emotion
6	1238425982907637780	monologia_id	2020-05-07	pemerintah lampung bantuan mahasiswa dieranya...	[pemerintah, lampung, bantuan, mahasiswa...]	-9	negative	happy
1	1238320972198940672	its_dai	2020-05-07	pemerintah merapkan herd immunity kemampuan...	[pemerintah, merapkan, herd, immunity...]	-8	negative	sadness
2	1238356644427083776	mesrikbaong	2020-05-07	lakukan makanan terbuka sultur supaya keluarga...	[lakukan, makanan, terbuka, sultur, ke...]	4	positive	happy
3	1238424368993931254	ralyadotnews	2020-05-07	warga positif corona bupati kondisi memantul k...	[warga, positif, corona, bupati, kond...]	-4	negative	happy
4	1238423545698246656	annisathali.	2020-05-07	emosi banger kondisi diuraah liputan covid doc...	[emosi, banger, kondisi, diuraah, lip...]	-3	negative	sadness

Gbr 8. Sampel Data Hasil *Labelling* Emosi

Pada gambar 8 diatas dapat kita lihat hasil dari proses *labelling* emosi menggunakan "*Pengklasifikasi emosi Indo RoBERTa*" menghasilkan 25.205 data dengan nilai *sadness*, 14.628 data dengan nilai *happy*, 11.715 data dengan nilai *anger*, 4.266 data dengan nilai *fear* dan 100 data dengan nilai *love*.

#### D. Hasil Word Embedding

Setelah dataset melalui proses *labelling*, kemudian dataset dilakukan proses *word embedding* dimana proses ini dilakukan untuk menangkap konteks kata dalam dokumen, kesamaan semantik dan sintaksis dan menemukan hubungan tersembunyi antar kata pada setiap kalimat. Pada penelitian ini proses *word embedding* dilakukan menggunakan metode Word2Vec. Dimensi *word vector* yang digunakan pada tahapan ini adalah sebesar 300 dimensi. Data *inputan* yang sebelumnya berbentuk string atau teks diubah menjadi angka atau *vektor*. Setiap *inputan* harus memiliki ukuran dimensi yang sama. Namun setiap kalimat masukan memiliki jumlah kata yang berbeda. Oleh karena itu, dilakukan proses *padding* menggunakan *pad sequences* Keras. Selanjutnya kalimat *input* di-tokenisasi menggunakan *tool tokenizer* Keras.

Teks yang berbentuk token ini yang menjadi data *inputan* untuk dimasukkan kedalam model Word2Vec. *Inputan* yang diproses tersebut akan menghasilkan matriks bobot yang memiliki dimensi 300. Berikut ini dapat kita lihat sampel hasil proses *word embedding* pada gambar 9 dibawah ini.

```
[ ] word2vec_weight_matrix[1]
array[[-0.6944235, -0.10669897, -0.02477432, -0.9271279, -0.32875445,
0.12870838, 0.79347676, -0.01032605, -0.1694137, -1.6185087,
-0.06688147, -0.04884438, 0.35320482, 0.7321195, 0.59436476,
0.1342476, -1.48969442, -0.68025023, 0.9057579, -0.7207759,
0.83774173, -0.68071534, 0.57307735, 0.92650904, 0.04274555,
-0.3190206, -0.42055249, 0.51950645, 0.39693084, -0.93446097,
-0.04639543, -0.96070498, -0.32164747, 0.18865055, 0.08867999,
0.09730536, -0.11045329, 0.02063623, -0.73536807, -0.45227212,
0.20541571, -0.16846463, 0.61707312, -0.13305611, 0.48521239,
0.06406575, 1.23288107, -0.40474812, 0.26512668, -0.45614126,
-0.17703757, 0.4480319, -0.13314983, 0.39010128, 1.30026436,
-0.03131105, 0.09598381, 0.51454788, -0.41471088, 0.24411294,
-1.19205248, -0.96548939, 0.89139432, -0.06717692, 0.05889936,
0.15990621, -1.14364839, -0.35532716, 0.21562979, -0.81216084,
0.24881233, -0.16210233, -0.21258764, -0.755683, -0.77583677,
-0.0257823, -0.0609831, -0.38690981, -0.11708438, 0.34416422,
-0.94948918, -0.74077111, -0.87431532, 0.46216735, -0.21650583,
-0.2653966, 0.02265694, -0.63940984, -1.42147171, 0.29159477,
-1.1183961, -0.15503044, -0.1180265, 0.57888281, -0.36095932,
```

Gbr 9. Sampel Data Hasil *Word Embedding*

#### E. Hasil Training Model

Tahapan ini dilakukan untuk melatih mesin menggunakan dataset yang sebelumnya telah dibagi menjadi data *training* dan data *testing*. Tahapan ini dilakukan menggunakan dua model algoritma yaitu algoritma CNN dan algoritma Bi-LSTM. *Inputan* yang digunakan pada proses *training* ini adalah data *training* yang sebelumnya telah diubah dari data teks atau *string* menjadi bentuk token menggunakan *tokenizer* dari library keras. Selanjutnya data tersebut diproses melalui *pad sequence* dari library keras untuk menyamakan ukuran panjang *padding* pada setiap *array* kata. Data ini yang kemudian menjadi data *input* untuk proses *training* model.

Data *inputan* selanjutnya diproses pada *embedding layer* dengan ukuran dimensi 300 dan maksimal kata *inputan* sebanyak 500 kata. Data yang melalui proses *word embedding* sebelumnya akan dijadikan parameter bobot pada *embedding layer* ini. Selanjutnya *inputan* dimasukkan pada konvolusi kernel satu dimensi. Proses ini menggunakan Conv1D dari library keras dengan nilai filter *output* sebesar 256 dan nilai *kernel size* sebesar 3. Setelah itu data *inputan* masuk pada proses pemilihan nilai maksimum dari matriks yang telah ditentukan untuk data temporal 1D menggunakan

GlobalMaxPooling1D dari library keras. Setelah itu data *inputan* diproses pada *dense layer* untuk mengklasifikasikan data berdasarkan *output* dari lapisan konvolusional.

Pada proses *training* algoritma Bi-LSTM setelah melalui *embedding layer*, setiap *inputan* kata dimasukkan ke dalam unit Bi-LSTM satu per satu sesuai dengan urutan *index* dari *sequence*. Setelah itu data *inputan* diproses pada *dense layer* untuk mengklasifikasikan data berdasarkan *output* dari lapisan konvolusional.

#### F. Hasil Testing Data

Tahapan ini merupakan tahapan terakhir dalam penelitian ini dimana dataset yang sebelumnya telah dibagi menjadi data *testing* dan *training* akan diuji pada tahap ini menggunakan model algoritma CNN dan Bi-LSTM. Dataset pada kolom polaritas sentimen dan emosi yang semula berbentuk data teks atau *string* kemudian diubah menjadi data *vector integer* dimana nilai sentimen diubah menjadi *vector integer* dengan *range interval* [0, 1, 2] yang disimpan pada *variable object polarity\_encoding*. Selain itu nilai emosi juga diubah menjadi data *vector integer* dengan *range interval* [0, 1, 2, 3, 4] yang disimpan pada *variable object emotion\_encoding*. Selanjutnya data *inputan* yang berbentuk *vector integer* tersebut diubah menjadi bentuk matriks biner menggunakan *to\_categorical* dari library *tensorflow*.

Data polaritas, emosi dan teks *tweet* yang telah melalui berbagai tahapan kemudian dimasukkan pada model algoritma CNN dan Bi-LSTM yang sebelumnya telah dibuat dan dilatih menggunakan data *training*. Pada proses ini dilakukan *testing* dan validasi data *testing* menggunakan data *training* yang sebelumnya telah di *tokenize*. Proses ini menggunakan *epoch* 15 dengan nilai *batch\_size* sebesar 256.

#### 1) Hasil Testing Model Algoritma CNN

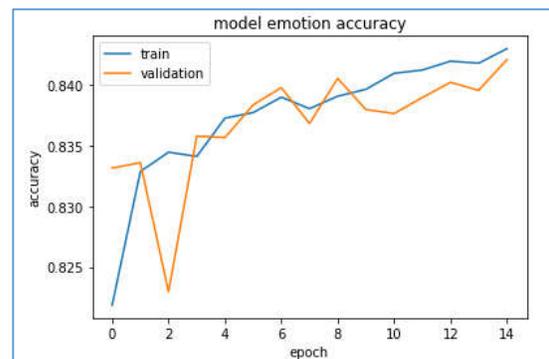
Hasil pengujian dan validasi data *testing* menggunakan model algoritma CNN dengan 15 kali proses putaran dan nilai *batch\_size* sebesar 256 menghasilkan nilai akurasi yang dapat kita lihat pada tabel 3 dibawah ini.

TABEL I  
HASIL PROSES TESTING MODEL CNN

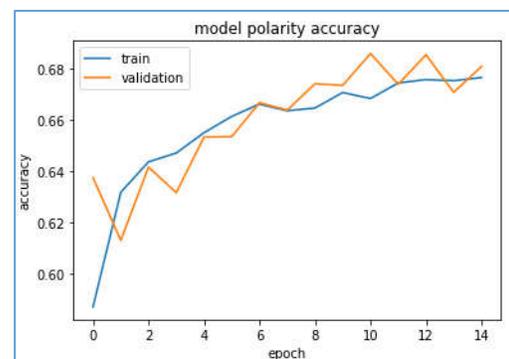
Epoch	Emotion Accuracy	Polarity Accuracy	Val Emotion Accuracy	Val Polarity Accuracy
1	0.8219	0.5870	0.8331	0.6374
2	0.8329	0.6317	0.8336	0.6130
3	0.8344	0.6436	0.8230	0.6415
4	0.8341	0.6470	0.8358	0.6316
5	0.8372	0.6549	0.8357	0.6532
6	0.8377	0.6613	0.8383	0.6534
7	0.8390	0.6661	0.8398	0.6666
8	0.8380	0.6635	0.8368	0.6637
9	0.8391	0.6646	0.8405	0.6740
10	0.8396	0.6706	0.8380	0.6734
11	0.8409	0.6682	0.8376	0.6858
12	0.8412	0.6743	0.8389	0.6738

13	0.8419	0.6756	0.8402	0.6853
14	0.8418	0.6752	0.8395	0.6707
15	0.8430	0.6764	0.8421	0.6807

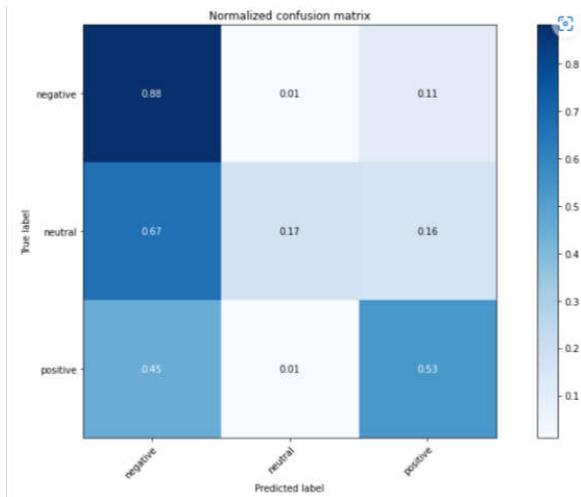
Berdasarkan tabel 3 diatas dapat dijelaskan bahwa hasil dari proses testing model menggunakan algoritma CNN dengan nilai epoch 15 mendapatkan rentan nilai akurasi dari 0,6130 atau 61,30% hingga 0,6858 atau 68,58% untuk analisis sentimen dan rentan nilai akurasi dari 0.8230 atau 82,30% hingga 0,8421 atau 84,21% untuk analisis emosi. Berikut dapat kita lihat pada gambar 10 dan gambar 11 bentuk grafik hasil proses *testing* menggunakan model algoritma CNN.



Gbr 10. Grafik Nilai Akurasi Testing Analisis Emosi Model Algoritma CNN

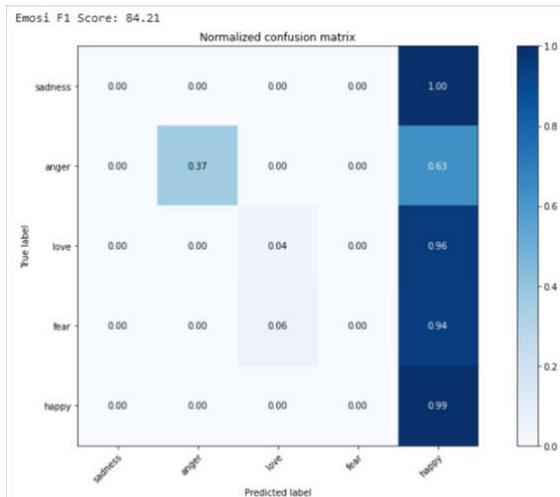


Gbr 11. Grafik Nilai Akurasi Testing Analisis Sentimen Model Algoritma CNN



Gbr. 12 Confusion Matrix Analisis Sentimen Model Algorithma CNN

Pada gambar 12 diatas prediksi *confusion matrix* untuk analisis sentimen model algoritma CNN menyatakan bahwa sebelah kiri merupakan *true label* atau label target dan bagian bawah adalah hasil prediksi sehingga dapat dinyatakan bahwa ada 0,88 atau 88% data tepat memprediksi sentimen *negative*, 0,17 atau 17% data tepat memprediksi sentimen *neutral* dan 0,53 atau 53% data tepat memprediksi sentimen *positive*.



Gbr. 13 Confusion Matrix Analisis Emosi Model Algorithma CNN

Pada gambar 13 diatas prediksi *confusion matrix* untuk analisis emosi pada model algoritma CNN menyatakan bahwa ada 0,00 atau 0% data tepat memprediksi emosi *sadness*, 0,37 atau 37% data tepat memprediksi emosi *anger*, 0,04 atau 4% data tepat memprediksi emosi *love*, 0,99 atau 99% data memprediksi emosi *happy* dan tidak ada data yang tepat atau 0% data yang tepat memprediksi emosi *fear*.

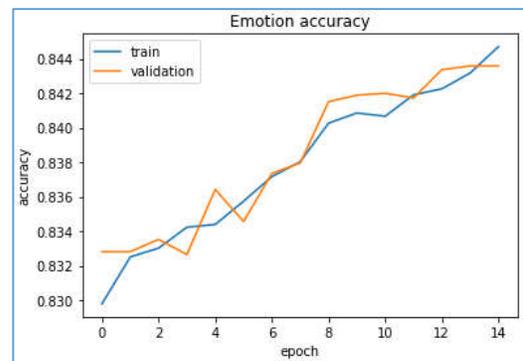
## 2) Hasil Testing Model Algorithma Bi-LSTM

Hasil pengujian dan validasi data *testing* menggunakan model algoritma Bi-LSTM dengan 15 kali proses putaran dan nilai *batch size* sebesar 256 menghasilkan nilai akurasi yang dapat kita lihat pada tabel 4 dibawah ini.

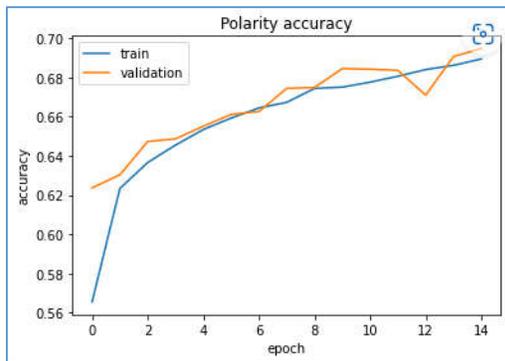
TABEL II  
HASIL PROSES TESTING MODEL BI-LSTM

Epoch	Emotion Accuracy	Polarity Accuracy	Val Emotion Accuracy	Val Polarity Accuracy
1	0.8298	0.5655	0.8328	0.6237
2	0.8325	0.6233	0.8328	0.6303
3	0.8330	0.6366	0.8335	0.6473
4	0.8342	0.6455	0.8326	0.6487
5	0.8344	0.6534	0.8364	0.6551
6	0.8357	0.6593	0.8346	0.6611
7	0.8372	0.6644	0.8374	0.6626
8	0.8380	0.6673	0.8380	0.6744
9	0.8403	0.6744	0.8415	0.6749
10	0.8409	0.6751	0.8419	0.6846
11	0.8407	0.6776	0.8420	0.6842
12	0.8419	0.6806	0.8417	0.6836
13	0.8423	0.6840	0.8434	0.6710
14	0.8432	0.6862	0.8436	0.6907
15	0.8447	0.6894	0.8436	0.6948

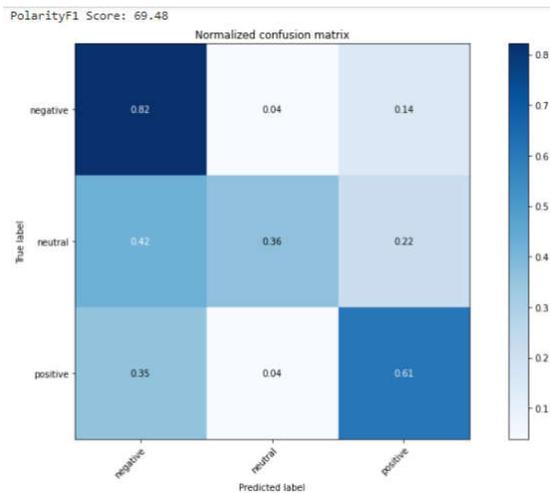
Berdasarkan tabel 4 diatas dapat dijelaskan bahwa hasil dari proses *testing* model menggunakan algoritma Bi-LSTM dengan nilai *epoch* 15 mendapatkan rentan nilai akurasi dari 0,6303 atau 63,03% hingga 0,6948 atau 69,48% untuk analisis sentiment dan rentan nilai akurasi dari 0,8326 atau 83,26% hingga 0,8436 atau 84,36% untuk analisis emosi. Berikut dapat kita lihat pada gambar 12 dan gambar 13 bentuk grafik hasil proses *testing* menggunakan model algoritma Bi-LSTM.



Gbr 14. Grafik Nilai Akurasi Testing Analisis Emosi Model Algorithma Bi-LSTM

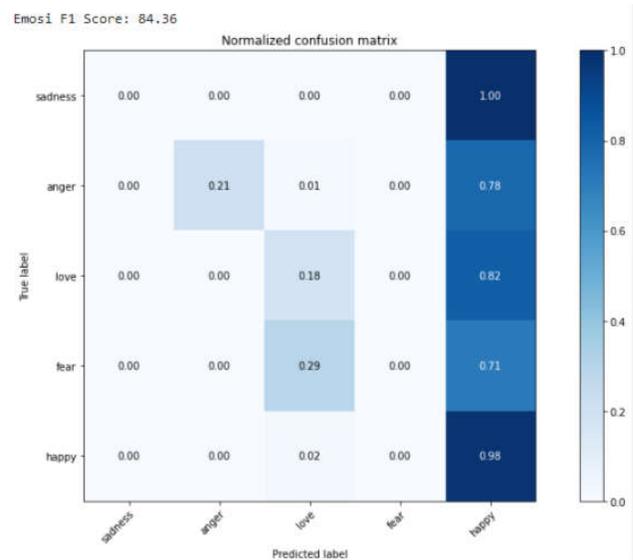


Gbr 15. Grafik Nilai Akurasi Testing Analisis Sentimen Model Algoritma Bi-LSTM



Gbr 16. Confusion Matrix Analisis Sentimen Model Algoritma Bi-LSTM

Pada gambar 16 diatas prediksi *confusion matrix* untuk analisis sentimen pada model algoritma Bi-LSTM menyatakan bahwa sebelah kiri merupakan true label atau label target dan bagian bawah adalah hasil prediksi sehingga dapat dinyatakan bahwa ada 0,82 atau 82% data tepat memprediksi sentimen *negative*, 0,36 atau 36% data tepat memprediksi sentimen *neutral* dan 0,61 atau 61% data tepat memprediksi sentimen *positive*.



Gbr 17. Confusion Matrix Analisis Emosi Model Algoritma Bi-LSTM

Pada gambar 17 diatas prediksi *confusion matrix* untuk analisis emosi pada model algoritma Bi-LSTM menyatakan bahwasannya ada 0,00 atau 0% data tepat memprediksi emosi *sadness*, 0,21 atau 21% data tepat memprediksi emosi *anger*, 0,18 atau 18% data tepat memprediksi emosi *love*, 0,98 atau 98% data memprediksi emosi *happy* dan tidak ada data yang tepat atau 0% data yang tepat memprediksi emosi *fear*.

Pada gambar 10, gambar 11, gambar 12 dan gambar 13 diatas dapat dinyatakan bahwa garis warna biru menunjukkan akurasi yang dihasilkan pada proses *training* di setiap *epoch*. Sedangkan garis warna oranye menunjukkan validasi akurasi yang dihasilkan pada proses *testing*. Dari grafik diatas dapat kita lihat bahwa nilai validasi akurasi pada proses *testing* cenderung tidak stabil. Ini bisa menandakan terjadi *overfitting* pada model yang dilatih.

Hasil akurasi yang rendah dari hasil penelitian perbandingan algoritma CNN dan Bi-LSTM pada analisis sentiment dan emosi menggunakan dataset unggahan twitter masyarakat Indoensia selama masa pandemik menggunakan metode Word2Vec disebabkan beberapa faktor antara lain :

1. Tidak dilakukannya proses *stemming* pada tahap praprosesing data. Dimana proses ini bertujuan untuk memperkecil jumlah indeks yang berbeda dari satu data sehingga sebuah kata yang memiliki suffix maupun prefix akan kembali ke bentuk dasarnya. Selain itu juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk yang berbeda karena mendapatkan imbuhan yang berbeda pula. Proses ini tidak dilakukan dikarenakan terbatasnya spesifikasi perangkat yang digunakan sehingga dibutuhkan waktu yang lama untuk melakukan proses *stemming* dengan jumlah dataset 58.978 data.

2. Rendahnya tingkat akurasi proses *labelling* polaritas sentiment menggunakan "Indonesia Sentiment Lexicon" yang memiliki tingkat akurasi sebesar 65.78%.

#### IV. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa model algoritma Bi-LSTM memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan model algoritma CNN dengan dimensi *word vector* 300 pada penelitian analisis sentimen dan emosi menggunakan dataset unggahan masyarakat Indonesia pada media social twitter selama masa pandemik Covid-19 menggunakan metode Word2Vec. Model algoritma Bi-LSTM mampu mencapai nilai akurasi hingga 0,8436 atau 84,36% untuk analisis emosi dan nilai akurasi tertinggi untuk analisis sentimen hingga 0,6948 atau 69,48%. Akurasi ini lebih tinggi dibandingkan model algoritma CNN yang mencapai nilai akurasi analisis emosi hingga 0,8421 atau 84,21% dan nilai akurasi tertinggi untuk analisis sentimen sebesar 0,6858 atau 68,58%. Beberapa hal yang mempengaruhi hasil akurasi pada penelitian ini seperti tidak dilakukannya proses stemming pada tahap pra-prosesing data. Kemudian rendahnya tingkat akurasi proses *labelling* data menggunakan "Indonesia Sentiment Lexicon" yang memiliki tingkat akurasi sebesar 65.78% untuk *labelling* polaritas sentimen dan "Pengklasifikasi emosi Indo RoBERTa" dengan tingkat akurasi sebesar 71,81% untuk *labelling* emosi pada dataset yang digunakan pada penelitian ini.

#### V. SARAN

Penelitian mengenai analisis sentimen dan emosi menggunakan dataset unggahan masyarakat Indonesia selama masa pandemik Covid-19 ini memiliki peluang besar untuk terus dikembangkan. Salah satu pengembangan yang dapat dilakukan adalah dengan mencoba metode dan model algoritma yang lain. Selain itu pada setiap tahapan dapat dikembangkan kembali terutama pada tahapan *labelling* sehingga penelitian tersebut dapat menghasilkan nilai akurasi yang lebih tinggi.

#### UCAPAN TERIMA KASIH

Ucapan syukur serta terima kasih penulis disampaikan kepada :

1. Allah SWT. Yang telah memberikan rahmat serta ridhonya sehingga penulis mampu menyelesaikan penelitian ini.
2. Nabi Muhammad SAW. yang telah memberikan suri tauladan yang baik yang dapat membuat saya tetap tenang dan sabar dalam melaksanakan penelitian ini.

3. Kedua orang tua saya Bapak Khosim dan Ibu Nur Khamidah yang selalu memberikan dukungan baik secara moril dan materiil sehingga saya tetap kuat dan semangat untuk menyelesaikan penelitian ini.
4. Pembimbing skripsi serta akademik saya Ibu Naim Rochmawati, S.Kom., M.T. yang telah membimbing saya dengan sabar dari mulai penelitian ini dilakukan hingga selesai.
5. Seluruh teman-teman serta semua pihak yang telah membantu dan memberikan dukungan kepada saya.

#### REFERENSI

- [1] "Coronavirus." [https://www.who.int/health-topics/coronavirus#tab=tab\\_1](https://www.who.int/health-topics/coronavirus#tab=tab_1) (accessed Apr. 25, 2022).
- [2] N. Chen and P. Wang, "Advanced Combined LSTM-CNN Model for Twitter Sentiment Analysis," *Proc. 2018 5th IEEE Int. Conf. Cloud Comput. Intell. Syst. CCIS 2018*, pp. 684–687, 2019, doi: 10.1109/CCIS.2018.8691381.
- [3] W. Widayat, "Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning," *J. Media Inform. Budidarma*, vol. 5, no. 3, p. 1018, 2021, doi: 10.30865/mib.v5i3.3111.
- [4] S. Khomsah, "Sentiment Analysis On YouTube Comments Using Word2Vec and Random Forest," *Telematika*, vol. 18, no. 1, p. 61, 2021, doi: 10.31315/telematika.v18i1.4493.
- [5] L. Wiranda, M. Sadikin, J. T. Informatika, and F. I. Komputer, "PENERAPAN LONG SHORT TERM MEMORY PADA DATA TIME SERIES UNTUK MEMREDIKSI PENJUALAN PRODUK PT. METISKA FARMA," vol. 8, pp. 184–196, 2019.
- [6] G. Rajendran, B. Chitturi, and P. Poornachandran, "Stance-In-Depth Deep Neural Approach to Stance Classification," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1646–1653, 2018, doi: 10.1016/j.procs.2018.05.132.
- [7] H. Deng, D. Ergu, F. Liu, Y. Cai, and B. Ma, "Text sentiment analysis of fusion model based on attention mechanism," *Procedia Comput. Sci.*, vol. 199, pp. 741–748, 2021, doi: 10.1016/j.procs.2022.01.092.
- [8] F. Koto and G. Y. Rahmanyngtyas, "Inset lexicon: Evaluation of a word list for Indonesian sentiment analysis in microblogs," *Proc. 2017 Int. Conf. Asian Lang. Process. IALP 2017*, vol. 2018-Janua, no. December, pp. 391–394, 2018, doi: 10.1109/IALP.2017.8300625.
- [9] Steven Limcorn, "indonesian-roberta-base-emotion-classifier," <https://huggingface.co/>, 2021. <https://huggingface.co/StevenLimcorn/indonesian-roberta-base-emotion-classifier/tree/main>
- [10] A. F. Niasita, P. P. Adikara, and S. Adinugroho, "Analisis Sentimen Pembangunan Infrastruktur di Indonesia dengan Automated Lexicon Word2Vec dan Naive-Bayes," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 3, pp. 2673–2678, 2019, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [11] E. I. Setiawan and I. Lestari, "Stance Classification Pada Berita Berbahasa Indonesia Berbasis Bidirectional LSTM," *J. Intell. Syst. Comput.*, vol. 3, no. 1, pp. 41–48, 2021, doi: 10.52985/insyst.v3i1.148.