

TESIS

**METODE BIDIRECTIONAL LSTM MENGGUNAKAN WORD
EMBEDDING UNTUK ANALISIS SENTIMEN TERHADAP VAKSIN
COVID-19 DI INDONESIA PADA MEDIA SOSIAL TWITTER**



Disusun oleh:

Nama : Kartikasari Kusuma Agustiniingsih
NIM : 20.55.1408
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 PJJ TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2022

TESIS

**METODE BIDIRECTIONAL LSTM MENGGUNAKAN WORD
EMBEDDING UNTUK ANALISIS SENTIMEN TERHADAP VAKSIN
COVID-19 DI INDONESIA PADA MEDIA SOSIAL TWITTER**

**BIDIRECTIONAL LSTM METHOD USING WORD EMBEDDING FOR
SENTIMENT ANALYSIS ON COVID-19 VACCINE IN INDONESIA ON
SOCIAL MEDIA TWITTER**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Kartikasari Kusuma Agustiniingsih
NIM : 20.55.1408
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 PJJ TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2022

HALAMAN PENGESAHAN

**METODE BIDIRECTIONAL LSTM MENGGUNAKAN WORD EMBEDDING
UNTUK ANALISIS SENTIMEN TERHADAP VAKSIN COVID-19 DI INDONESIA
PADA MEDIA SOSIAL TWITTER**

**BIDIRECTIONAL LSTM METHOD USING WORD EMBEDDING FOR
SENTIMENT ANALYSIS ON COVID-19 VACCINE IN INDONESIA ON SOCIAL
MEDIA TWITTER**

Dipersiapkan dan Disusun oleh

Kartikasari Kusuma Agustiniingsih

20.55.1408

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 PJJ Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 02 Juni 2022

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 02 Juni 2022

Rektor

Prof. Dr. M. Suyanto, M.M.

NIK. 190302001

HALAMAN PERSETUJUAN

METODE BIDIRECTIONAL LSTM MENGGUNAKAN WORD EMBEDDING UNTUK ANALISIS SENTIMEN TERHADAP VAKSIN COVID-19 DI INDONESIA PADA MEDIA SOSIAL TWITTER

BIDIRECTIONAL LSTM METHOD USING WORD EMBEDDING FOR SENTIMENT ANALYSIS ON COVID-19 VACCINE IN INDONESIA ON SOCIAL MEDIA TWITTER

Dipersiapkan dan Disusun oleh

Kartikasari Kusuma Agustiniingsih

20.55.1408

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 PJJ Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 02 Juni 2022

Pembimbing Utama

Prof. Dr. Ema Utami, S.Si, M.Kom
NIK. 190302037

Anggota Tim Penguji

Prof. Dr. Kusrini, M.Kom
NIK. 190302106

Pembimbing Pendamping

Hanif Al Fatta, M.Kom
NIK. 190302096

Alva Hendi M., S.T., M.Eng., Ph.D.
NIK. 190302493

Prof. Dr. Ema Utami, S.Si, M.Kom
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 02 Juni 2022
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Kartikasari Kusuma Agustiningih
NIM : 20.55.1408
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Metode Bidirectional LSTM Menggunakan Word Embedding untuk Analisis Sentimen Terhadap Vaksin COVID-19 di Indonesia Pada Media Sosial Twitter

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom
Dosen Pembimbing Pendamping : Hanif Al Fatta, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

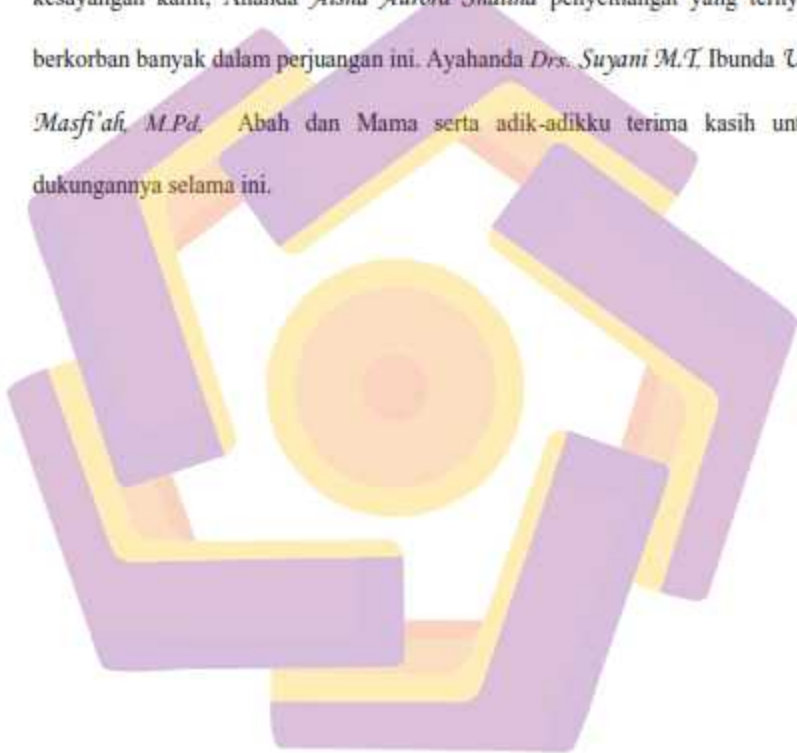
Yogyakarta, 02 Juni 2022
Yang Menyatakan,



Kartikasari Kusuma Agustiningih

HALAMAN PERSEMBAHAN

Kupersembahkan kepada suami tercinta, *Mas Omar Muhammad A.A* yang selalu sabar dan ikhlas memberikan dukungan, bantuan dan do'a. Untuk puteri kesayangan kami, Ananda *Aisha Aurora Shalifa* penyemangat yang ternyata berkorban banyak dalam perjuangan ini. Ayahanda *Drs. Suyani M.T*, Ibunda *Umi Masfi'ah, M.Pd.* Abah dan Mama serta adik-adikku terima kasih untuk dukungannya selama ini.



HALAMAN MOTTO

لَئِنْ شَكَرْتُمْ لَأَزِيدَنَّكُمْ

If you are grateful, I will surely give you more and more

(QS. Ibrahim [14]:7)



KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT yang telah memberikan rahmat-Nya sehingga penulis dapat menyelesaikan laporan penelitian tesis ini dengan baik. Penulisan laporan tesis ini dapat terselesaikan berkat bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan terimakasih kepada pihak-pihak yang terlibat dalam penelitian ini:

1. Prof. Dr. M. Suyanto, MM. selaku Rektor Universitas AMIKOM Yogyakarta.
2. Ibu Prof. Dr. Kusriati, M.Kom. selaku Direktur Program Pascasarjana Universitas AMIKOM Yogyakarta.
3. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom. selaku Wakil Direktur Program Pascasarjana Universitas AMIKOM Yogyakarta sekaligus selaku Pembimbing Utama.
4. Bapak Hanif Al Fatta, M.Kom. selaku dosen Pembimbing Pendamping.
5. Bapak Alva Hendi Muhammad, S.T, M.Eng, Ph.D., selaku penguji Seminar Proposal, Seminar Hasil dan Ujian Tesis.
6. Bapak Dhani Ariatmanto, M.Kom, selaku penguji Seminar Proposal.
7. Bapak Dr. Kumara Ari Yuana, S.T, M.T., selaku penguji Seminar Hasil Penelitian.
8. Serta semua pihak yang tidak bisa penulis sebutkan satu persatu terima kasih atas doa dan sarannya.

Dengan diiringi doa dan ucapan terimakasih, penulis berharap semoga tesis ini dapat bermanfaat. Saran, harapan, kritik yang membangun selalu penulis untuk perbaikan di masa yang akan datang. Terimakasih

Banjarbaru, 2022

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR ISTILAH.....	xv
INTISARI.....	xvi
<i>ABSTRACT</i>	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	6
1.3. Batasan Masalah.....	6
1.4. Tujuan Penelitian.....	8
1.5. Manfaat Penelitian.....	8
BAB II TINJAUAN PUSTAKA.....	9
2.1. Tinjauan Pustaka.....	9

2.2. Keaslian Penelitian.....	15
2.3. Landasan Teori.....	21
2.3.1. Natural Language Processing (NLP).....	21
2.3.2. Analisis Sentimen.....	22
2.3.3. Pre-Processing.....	23
2.3.4. Word Embedding.....	24
2.3.5. Glove.....	25
2.3.6. FastText.....	26
2.3.7. BERT Embedding.....	27
2.3.8. Bidirectional LSTM (Bi-LSTM).....	28
2.3.9. Metrik Pengukuran.....	30
BAB III METODE PENELITIAN.....	33
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	33
3.2. Metode Pengumpulan Data.....	33
3.3. Metode Analisis Data.....	34
3.4. Dataset.....	34
3.5. Alur Penelitian.....	40
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	45
4.1. Hasil Sentimen dan Ragam Sentimen.....	45
4.2. Mekanisme <i>Training</i>	48
4.3. Penanganan <i>Imbalance Data</i>	51
4.4. Model FastText.....	52

4.4.1. Pembuatan Model FastText	52
4.4.2. <i>Training</i> dan <i>Testing</i> FastText	54
4.4.3. Durasi Proses <i>Training</i>	56
4.5. Model GloVe	58
4.5.1. Pembuatan Model GloVe	58
4.5.2. <i>Training</i> dan <i>Testing</i> GloVe	59
4.5.3. Durasi Proses <i>Training</i>	61
4.6. Model BERT	63
4.6.1. Pembuatan Model BERT	63
4.6.2. <i>Training</i> dan <i>Testing</i> BERT	63
4.6.3. Durasi Proses <i>Training</i>	65
4.7. Pembahasan Hasil	67
4.7.1. Analisis Sentimen	67
4.7.2. Analisis Hasil Klasifikasi	70
4.8. Proses Vektorisasi	74
4.9. Proses Klasifikasi	78
BAB V PENUTUP	86
5.1. Kesimpulan	86
5.2. Saran	87
DAFTAR PUSTAKA	89

DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian.....	15
Tabel 3.1. Contoh Dataset	38
Tabel 3. 2. Contoh Hasil Pelabelan	42
Tabel 4.1. Variasi Skenario Training Data.....	50
Tabel 4.2. Hasil Pengujian Model FastText menggunakan Bi-LSTM.....	55
Tabel 4.3. Perbandingan Durasi Training Model FastText.....	57
Tabel 4.4. Hasil Pengujian Model GloVe menggunakan Bi-LSTM.....	60
Tabel 4.5. Perbandingan Durasi Training Model GloVe	62
Tabel 4.6. Hasil uji pada pada model BERT.....	64
Tabel 4.7. Perbandingan Durasi Training Model BERT.....	66
Tabel 4.8. Jumlah Sentimen Perbulan.....	69
Tabel 4.9. Hasil Seluruh Uji yang dilakukan	71
Tabel 4. 10. Contoh tabel perhitungan matriks kata oleh GloVe.....	77
Tabel 4. 11 Contoh kalimat tweet	85
Tabel 4. 12 Contoh Perhitungan Klasifikasi menggunakan Bi-LSTM.....	85

DAFTAR GAMBAR

Gambar 2.1. Arsitektur Glove (Liu dkk., 2019).....	26
Gambar 2.2. Arsitektur BERT (Roy & Ojha, 2020).....	28
Gambar 2.3. Arsitektur Bi-LSTM (Cui dkk., 2017).....	29
Gambar 2.4 Ilustrasi <i>Confusion Matrix</i> untuk Klasifikasi <i>Multiclass</i> (Tharwat, 2018).....	30
Gambar 2.5. <i>Confusion Matrix</i> untuk setiap kelas.....	31
Gambar 3.1. Contoh data hasil <i>scrapping data</i>	35
Gambar 3.2. Contoh data hasil <i>filtering</i>	35
Gambar 3.3. Alur <i>preprocessing Data</i>	40
Gambar 3.4. Alur penelitian.....	41
Gambar 4.1. Jumlah Tweet Berdasarkan Sentimen.....	46
Gambar 4. 2 Sentimen Vaksin COVID-19 Periode September – Desember 2020	47
Gambar 4. 3 Sentimen Vaksin COVID-19 Periode Januari – Maret 2021.....	47
Gambar 4. 4 Sentimen Vaksin COVID-19 Periode April – Juni 2021.....	48
Gambar 4.5. Data Sebelum dan Setelah <i>Oversampling</i>	52
Gambar 4.6. <i>Pseudocode</i> Pembuatan Model FastText.....	53
Gambar 4.7. Contoh Hasil Vektorisasi Kata “covid” pada FastText.....	54
Gambar 4.8. Perbandingan Kinerja Variasi Model FastText dan Bi-LSTM.....	56
Gambar 4.9. Grafik perbandingan durasi <i>training</i> model GloVe.....	57
Gambar 4.10. Contoh Vektorisasi Kata “covid” menggunakan GloVe.....	59
Gambar 4.11. Perbandingan Kinerja Variasi Model GloVe dan Bi-LSTM.....	61

Gambar 4.12. Grafik perbandingan durasi <i>training</i> model GloVe	62
Gambar 4.13. Grafik Perbandingan Variasi Hasil Uji Model BERT.....	65
Gambar 4.14. Grafik Durasi Proses Training Model BERT.....	67
Gambar 4.15. Jumlah <i>Tweet</i> Perbulan.....	68
Gambar 4.16. Grafik Sentimen Perbulan.....	70
Gambar 4.17. Grafik Perbandingan Akurasi Terbaik dari tiap Word Embedding	72



DAFTAR ISTILAH

Dataset: Kumpulan data yang umumnya digunakan dalam penelitian di bidang data science, machine learning dan deep learning. Ada dataset yang bersifat *public*, yakni bisa diakses secara bebas dan ada pula yang bersifat *private*.

Model: Istilah umum yang digunakan untuk merepresentasikan kecerdasan buatan yang sudah dilatih.

Epoch: Sebuah kecerdasan buatan dapat dilatih dengan data yang sama berkali-kali. Setiap menyelesaikan satu kali latihan (epoch), nilai weight atau bobot pada setiap neuron diperbaharui agar dapat nilai keluarannya dapat mendekati output yang seharusnya berdasarkan label data.

Word Embedding : Proses konversi kata berbentuk alphanumeric ke dalam bentuk vektor (kumpulan angka)



INTISARI

Bencana non alam *Corona Virus Disease 2019* (COVID-19) yang melanda dunia sejak akhir tahun 2019 membawa pengaruh yang besar dalam tatanan kehidupan manusia. Tatanan hidup baru berupa protokol kesehatan perlahan mulai diterapkan hingga saat ini seperti menjaga jarak, menggunakan masker, mengurangi mobilitas serta vaksinasi. Khusus untuk vaksinasi dimana setiap individu wajib melakukannya sebanyak 2 kali, bahkan 3 kali mulai banyak terjadi pro dan kontra. Pro dan kontra ini dapat dilihat secara langsung pada sosial media seperti Twitter.

Penelitian ini memanfaatkan opini dari sosial media Twitter untuk dapat dianalisis sentimennya, baik itu sentimen positif, negatif dan netral. Proses melakukan sentimen analisis menggunakan 26.332 data. Data yang diperoleh akan dilakukan *preprocessing* lalu kemudian di vektorisasi menggunakan 3 jenis *word embedding* yaitu, FastText, GloVe dan BERT dan diklasifikasikan menggunakan metode Bi-LSTM.

Karena data yang digunakan tidak seimbang, maka perlu dilakukan penanganan untuk menyeimbangkan data menggunakan SMOTE. Hal ini juga merupakan salah satu variabel dalam variasi skenario yang akan diuji. Karena keterbatasan komputasi, *word embedding* BERT tidak dapat dibandingkan dengan FastText dan GloVe. Dari hasil pengujian dengan berbagai skenario uji diperoleh akurasi terbaik sebesar 71.11% pada saat data di vektorisasi menggunakan *word embedding* FastText.

Kata kunci: sentimen analisis, vaksin COVID-19, FastText, GloVe, BERT

ABSTRACT

The non-natural disaster Corona Virus Disease 2019 (COVID-19) which has hit the world since the end of 2019 has had a great influence on the order of human life. The new normal life as health protocols is slowly starting to be implemented until now, such as maintaining distance, wearing masks, reducing mobility and vaccinations. Especially for vaccinations where each individual is required to do it 2 times, even 3 times, there are many pros and cons. These pros and cons can be seen directly on social media such as Twitter.

This study utilizes opinions from Twitter social media to be able to analyze sentiments, be it positive, negative and neutral sentiments. The process of doing sentiment analysis used 26.332 data. The data obtained were preprocessed and then vectorized using 3 types of word embedding namely, FastText, GloVe and BERT and classified using the Bi-LSTM method.

Because the data used was not balanced, it was necessary to handle the data balance using SMOTE. This is also one of the variables in the variety of scenarios to be tested. From the test results with various test scenarios, the best accuracy was obtained at 71.11% when the data was vectorized using word embedding FastText.

Keyword: sentiment analysis, COVID-19 vaccine, FastText, GloVe, BERT

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Corona Virus Disease 2019 (COVID-19) telah menyebar ke seluruh dunia sejak 2019. Berdasarkan data dari *Worldmeter* pada bulan Juni tahun 2021 bahwa telah terjadi lebih dari 179 juta kasus COVID-19 dengan angka kematian mencapai lebih dari 3 juta pasien. Oleh karena itu, para peneliti mulai melakukan penelitian untuk mengembangkan vaksin COVID-19. Beberapa vaksin yang sudah diakui oleh *World Health Organization* (WHO) yaitu: Pfizer, AstraZeneca dan Ad26.COV2.S, sedangkan beberapa vaksin lain sedang dalam proses *review* dan *assessment* oleh WHO (WHO, 2021).

Proses vaksinasi sudah diawali oleh negara Amerika Serikat dan Israel sejak bulan Desember tahun 2020 dan diikuti oleh negara-negara lain, termasuk Indonesia (Ritchie dkk., 2021). Proses vaksinasi ini menuai kontroversi di beberapa negara. Hasil penelitian (Sallam, 2021; Bono dkk., 2021; Solis Arce dkk., 2021) mengungkapkan bahwa ada beberapa negara yang penerimaan masyarakat terhadap vaksin rendah. Hal ini perlu ditangani untuk menekan penyebaran COVID-19 secara global.

Vaksinasi bertujuan untuk membentuk *Herd Immunity* di dalam suatu wilayah. *Herd Immunity* merupakan perlindungan tidak langsung dari infeksi yang diberikan kepada individu yang rentan ketika proporsi individu yang kebal dalam suatu populasi cukup besar (Randolph & Barreiro, 2020).

Langkah yang tepat sebelum mengambil sebuah kebijakan untuk melakukan vaksinasi ataupun melakukan edukasi kepada masyarakat tentang vaksin, perlu diketahui terlebih dahulu bagaimana persepsi masyarakat terhadap kebijakan vaksinasi yang akan diterapkan. Hal ini penting untuk mengetahui agar keputusan yang diambil tidak mendapat tanggapan negatif dari masyarakat, sehingga kebijakan yang diambil dapat dijalankan segera dengan efektif. Salah satu teknik yang dapat digunakan untuk mengetahui persepsi masyarakat terhadap vaksinasi adalah melalui analisis sentimen.

Analisis sentimen dapat dilakukan pada persepsi masyarakat di sosial media dengan menganalisis bagaimana masyarakat beropini terhadap berbagai topik (Nurdeni dkk., 2021). Hal ini juga bisa digunakan untuk mempelajari pandangan individu, perilaku, perasaan terhadap orang lain, diri sendiri, masalah yang berkembang serta kegiatan yang selama ini sudah dilakukan. Informasi ini menjadi hal yang menarik bagi para pengambil keputusan sebab mampu memastikan apa yang sedang dipikirkan oleh orang lain (Basari dkk., 2013). Oleh sebab itu dikembangkan suatu teknik klasifikasi untuk mengelompokkan suatu opini yang sedang berkembang, baik yang ada di sosial media maupun media massa yang lain.

Beberapa penelitian terkait sentimen analisis vaksin COVID-19 di Indonesia telah banyak dilakukan dengan berbagai metode dan hasil sentimen yang beragam. Seperti penelitian yang dilakukan oleh (Pristiyono dkk., 2021) menggunakan metode Naïve Bayes dengan hasil sentimen negatif lebih besar dari sentimen positif pada sosial media Twitter. Hasil sentimen ini berbeda dengan penelitian yang dilakukan oleh (Yulita dkk., 2021) dimana persentase sentimen

positif lebih besar dibandingkan sentimen negatif yaitu sebesar 60,3%. Perbedaan ini terjadi dataset yang digunakan keduanya memiliki rentang waktu yang berbeda.

Penelitian lain terkait sentimen analisis vaksin COVID-19 pada sosial media Twitter dilakukan oleh (Sistem dkk., 2021) menggunakan Recurrent Neural Network dan Naïve Bayes dengan hasil sentimen positif lebih besar dari sentimen negatif. Penelitian tersebut cukup menarik karena membandingkan hasil klasifikasi menggunakan *machine learning* dan *deep learning*.

Saat ini, dengan bertambahnya jumlah dan ukuran *data*, *traditional machine learning* dianggap belum cukup dalam melakukan analisis sentimen (Zhang dkk., 2018). Penelitian oleh (Jain & Kaushal, 2018) dan (Chandra & Jana, 2020) mencoba membandingkan metode *machine learning* dan *deep learning* dalam melakukan sentimen analisis, hasilnya dalam banyak kasus metode *deep learning* memberikan hasil yang lebih baik daripada *machine learning*. Penelitian (Roy & Ojha, 2020) membandingkan hasil klasifikasi untuk sentimen analisis pada dataset SemEval-2016 menggunakan *deep learning Bidirectional Encoder Representations from Transformers (BERT)*, *Bidirectional Long Short-Term Memory (Bi-LSTM)* dan *Convolutional Neural Network (CNN)*. Tiga metode tersebut dikombinasikan dengan *word embedding GloVe* sebagai dasar representasi data karena biasanya *tweet* yang diposting tidak menggunakan struktur kata yang benar, menggunakan bahasa gaul dan singkatan-singkatan yang tidak dimengerti oleh mesin. Hasil penelitian mereka menunjukkan bahwa metode Bi-LSTM yang berbasis *attention* dan dikombinasikan dengan *GloVe Embedding* mempunyai performa yang lebih baik dari pada metode CNN dengan konfigurasi arsitektur yang sama.

Penelitian (Abdalla & Özyurt, 2021), yaitu CNN, Bi-LSTM dan kombinasi CNN dengan Bi-LSTM. Setiap model dilatih menggunakan tiga dataset dengan jumlah yang berbeda. *Word embedding* juga digunakan dalam penelitian ini karena digunakan untuk merepresentasikan kata ke dalam *vector* agar dipahami oleh mesin. Metode *word embedding* yang digunakan dalam melakukan sentimen analisis adalah Word2Vec. Hasil eksperimen mereka pada metode Bi-LSTM menghasilkan akurasi mencapai 95.35% sedangkan metode CNN hanya mencapai 91.55% dan metode CNN-Bi-LSTM mencapai 91.14%. Peneliti juga mengungkapkan bahwa besarnya dataset yang digunakan untuk *training* akan dapat meningkatkan akurasi model.

Penelitian lain yang menerapkan *word embedding* adalah penelitian (Pasaribu dkk., 2020) yang melakukan peningkatan akurasi algoritma Bi-LSTM dengan BERT *embedding*. Selain itu, dari hasil penelitian (Pasaribu dkk., 2020) dapat diketahui bahwa semakin banyak jumlah dataset yang digunakan untuk *training* maka semakin tinggi akurasi yang dihasilkan oleh model.

Penelitian (Riza & Charibaldi, 2021) melakukan *multiple*-klasifikasi menggunakan LSTM dan *word embedding* FastText, GloVe dan Word2vec. Hasil akurasi tertinggi tercapai ketika menggunakan *word embedding* FastText. Penelitian ini menyarankan untuk menggunakan Bi-LSTM pada penelitian selanjutnya untuk mendapatkan hasil yang optimal.

Beberapa penelitian di atas telah memberikan gambaran sekaligus motivasi untuk melaksanakan penelitian dalam melakukan sentimen analisis menggunakan *Deep learning*. Jika dibandingkan dengan akurasi yang ditunjukkan oleh algoritma

Machine learning, algoritma *Deep learning* menghasilkan akurasi yang lebih baik (Hidayatullah dkk., 2021).

Pada penelitian (Abdalla & Özyurt, 2021) menyatakan bahwa metode Bi-LSTM lebih efektif dibandingkan metode *neural network* yang lain. Penelitian (Riza & Charibaldi, 2021) juga meyarankan untuk menggunakan metode Bi-LSTM dikombinasikan dengan *word embedding* untuk mendapatkan akurasi yang lebih baik. Selanjutnya, penelitian yang dilakukan oleh (Nurdin dkk., 2020) membandingkan kinerja tiga *word embedding*, yaitu Word2Vec, GloVe dan Fasttext yang diuji coba pada metode CNN. Model diuji menggunakan dataset 20 newsgroup dan Reuters Newswire yang berbahasa Inggris. Hasil penelitian mereka menunjukkan bahwa model yang menggunakan *word embedding* Fasttext mempunyai kinerja yang lebih baik dari pada dua *word embedding* lain. Hasil penelitian (Adipradana dkk., 2021) juga membuktikan hasil yang sama dimana *word embedding* FastText yang digunakan pada 4 macam metode, yaitu LSTM, Bi-LSTM, *Gated Recurrent Unit* (GRU) dan *Bidirectional Gated Recurrent Unit* (Bi-GRU) menghasilkan performa yang lebih baik dari pada *word embedding* GloVe yang digunakan pada metode yang sama.

Penelitian (D'Sa dkk., 2020) membandingkan performa *word embedding* FastText, BERT *embedding* dan BERT *fine-tuning* yang dilatih pada metode CNN dan Bi-LSTM. Berdasarkan hasil penelitian mereka, performa BERT *fine-tuning* melampaui metode CNN dan Bi-LSTM baik saat menggunakan Fasttext *embedding* maupun BERT *embedding*.

Berdasarkan beberapa penelitian di atas, maka pada penelitian ini akan menguji metode Bidirectional-LSTM dengan menggunakan *word embedding* FastText, GloVe dan BERT *embedding*.

1.2. Rumusan Masalah

Latar belakang diatas menghasilkan rumusan masalah sebagai berikut:

- a. Bagaimana ragam sentimen pengguna Twitter terhadap vaksin COVID-19 di Indonesia pada rentang waktu September 2020 – Juni 2021?
- b. Berapa nilai performa (akurasi, presisi, *recall* dan *f1 score*) yang dihasilkan oleh model saat menggunakan masing-masing *word embedding* (Glove, Fast Text dan BERT Embedding) ?
- c. *Word embedding* yang manakah yang menjadikan menghasilkan akurasi tertinggi?

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

- a. *Dataset* yang digunakan adalah dataset berbahasa Indonesia yang diambil dari media sosial Twitter menggunakan kata kunci “vaksin covid” pada periode September 2020 – Juni 2021.
- b. *Dataset* merupakan data berformat teks yang diambil menggunakan metode *scraping* melalui Twitter API menggunakan *tools* SNScrape.

- c. *Dataset* yang dikumpulkan di-*filter* untuk menghasilkan data dari opini masyarakat. Filterisasi yang digunakan adalah menghapus *tweet* dari akun milik lembaga, organisasi dan situs berita.
- d. *Dataset* yang digunakan merupakan *tweet* berbahasa Indonesia dari seluruh masyarakat Indonesia baik yang belum menerima vaksin COVID-19 dan yang telah menerima vaksin COVID-19.
- e. *Tweet* yang diambil sebagai dataset bisa merupakan pernyataan atau pertanyaan yang diposting pada akun Twitter.
- f. Proses pelabelan dilakukan secara manual dan melibatkan pakar ahli dari bidang psikologi.
- g. *Pre-processing* data dilakukan melalui teknik *case folding*, *remove punctuation*, *stopword removal*, *stemming*, *slangword* dan identifikasi frasa.
- h. *Platform* penelitian menggunakan Google Colaboratory dan Google Colaboratory Pro
- i. Sentimen analisis diklasifikasikan menjadi 3, yaitu positif, negatif dan netral.
- j. Sentimen dilakukan terhadap semua jenis vaksin COVID-19 yang digunakan.
- k. Komposisi dataset yang digunakan adalah 80% untuk data *training* dan 20% untuk data *testing*.
- l. Jumlah epoch yang digunakan untuk *training* sebanyak 100 epoch.
- m. Metode yang digunakan untuk melakukan sentimen analisis adalah metode Bi-LSTM dengan beberapa kombinasi *word embedding*, yaitu FastText, GloVe dan BERT.

- n. Membandingkan tingkat performa (akurasi, presisi, *recall* dan *f1 score*) dari algoritma Bi-LSTM + FastText, Bi-LSTM + GloVe dan Bi-LSTM+BERT dalam studi kasus sentimen analisis vaksin COVID-19 di Indonesia pada sosial media Twitter.

1.4. Tujuan Penelitian

- a. Mengetahui ragam sentimen pengguna Twitter terhadap vaksin COVID-19 di Indonesia rentang waktu September 2020 – Juni 2021.
- b. Mengetahui nilai performa yang dihasilkan tiap kelompok sentimen dengan menggunakan Algoritma Bi-LSTM dengan FastText, Bi-LSTM dengan GloVe dan Bi-LSTM dengan BERT berdasarkan evaluasi akurasi, presisi, *recall* dan *f1 score*.
- c. Mengetahui *word embedding* mana yang menghasilkan performa terbaik.

1.5. Manfaat Penelitian

- a. Dapat menjadi pedoman pengembangan penelitian dalam menganalisa sentimen publik mengenai vaksin COVID19 di Indonesia pada sosial media Twitter.
- b. Berkontribusi secara ilmiah terhadap penggunaan *word embedding* dalam pengolahan data Twitter berbahasa Indonesia pada klasifikasi sentimen menggunakan algoritma Bidirectional LSTM.
- c. Menemukan pola dan hubungan antara sentimen pengguna Twitter terhadap isu terkait vaksin COVID-19.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian terdahulu yang pernah dilakukan, relevan dan dijadikan studi literatur adalah sebagai berikut:

Penulis (Pristiyono dkk., 2021) melakukan penelitian terkait sentimen analisis vaksin COVID-19 di Indonesia menggunakan *tweet* yang diambil dari Twitter pada tanggal 15-22 Januari 2021 dengan jumlah *tweet* sebanyak lebih dari 3400 *tweet* yang sebelumnya telah dibersihkan dan di-*preprocessing*. Sentimen analisis diklasifikasikan menjadi tiga, yaitu positif, negatif dan netral menggunakan metode Naïve Bayes. Klasifikasi tersebut menghasilkan persentase sentimen positif sebesar 39% dan sentimen negatif sebesar 56%. Hasil berbeda dikemukakan oleh (Yulita dkk., 2021) dan (Sistem dkk., 2021). Dengan metode yang sama yaitu Naïve Bayes, (Yulita dkk., 2021) memperoleh persentase sentimen positif sebesar 60.3%, sentimen negatif sebesar 5,4% pada 3780 *tweet*. Perbedaan hasil yang cukup signifikan ini dipengaruhi oleh rentang waktu yang digunakan saat melakukan *crawling* data dari Twitter.

Penelitian yang dilakukan oleh (Hidayatullah dkk., 2021) mengklasifikasikan sentimen pemilihan presiden tahun 2019. Dataset dikumpulkan dalam dua periode yang berbeda, yaitu sebelum dan sesudah pemilihan presiden menggunakan *library* yang disediakan oleh Phyton yakni *Tweepy*. Data yang berhasil dikumpulkan sejumlah 115.931 data mentah. Data tersebut kemudian di

pre-processing menggunakan *case folding*, *word normalization* dan *stemming*. Beberapa karakter dihilangkan, seperti karakter ASCII, nama pengguna, tagar, URL, *retweet*, tanda baca, duplikat karakter dalam sebuah kata duplikasi *tweet*.

Data yang telah selesai di *pre-processing* kemudian dilakukan beberapa model percobaan menggunakan *machine learning* dan *deep learning*. Metode *machine learning* yang digunakan dalam penelitian ini adalah Support Vector Machines (SVM), Multinomial Naive Bayes (MNB) dan Logistic Regression (LR) dengan menerapkan *Term Frequency-Inversed Document Frequency* (TF-IDF) dan tanpa TF-IDF. Sedangkan, untuk metode *deep learning* yang digunakan adalah CNN, LSTM, CNN+LSTM, GRU+LSTM and Bi-LSTM. Dalam melakukan percobaan, dataset dibagi menjadi dua bagian, yaitu data latih dan data uji dengan persentase 80% untuk data latih dan 20% data uji. Dari hasil penelitian ini diperoleh akurasi terbaik pada metode SVM dengan TF-IDF sebesar 84.04% untuk *machine learning* dan metode Bi-LSTM dengan akurasi sebesar 84.60%. Penelitian ini yang mendorong penulis untuk melakukan klasifikasi sentimen analisis vaksin COVID-19 menggunakan metode Bi-LSTM.

Penelitian lain terkait sentimen analisis menggunakan *deep learning model* dan memperoleh kinerja terbaik saat menggunakan metode Bi-LSTM adalah penelitian oleh (Roy & Ojha, 2020) yang menggunakan dataset SemEval-2016 untuk diklasifikasikan menggunakan Google BERT, Bi-LSTM, dan CNN. Klasifikasi sentimen dibagi menjadi tiga, yaitu sentimen positif, negatif dan netral. Hampir sama dengan penelitian sebelumnya, data di *pre-processing* terlebih dahulu sebelum diklasifikasikan. Teknik *pre-processing* yang dilakukan antara lain,

mengganti *emoticon* berdasarkan polaritas positif dan negatif, menghapus URL, mengubah ke huruf kecil, menghilangkan tag dan nomor, akronim, dan menghapus karakter yang duplikat. Dari 20.632 *tweet* SemEval-2016 dataset yang diklasifikasikan, hasil akurasi terbaik didapatkan ketika menggunakan model BERT dengan akurasi 64,1%, sedangkan ketika menggunakan Bi-LSTM didapatkan akurasi sebesar 60.02% dan 59.2% pada hasil akurasi menggunakan CNN. Teknik *pre-processing* dalam penelitian tersebut akan diimplementasikan dalam penelitian ini.

Metode Bi-LSTM juga mendapatkan hasil terbaik saat melakukan klasifikasi opini negatif dan positif masyarakat pada perusahaan makanan cepat saji, seperti KFC, Burger King dan McDonald's seperti yang dilakukan oleh (Abdalla & Özyurt, 2021) dimana peneliti membandingkan model *deep learning* yaitu, CNN, Bi-LSTM dan kombinasi CNN + Bi-LSTM. Dataset yang digunakan didapatkan dari Twitter menggunakan Tweepy dan kemudian diberikan label menggunakan *Valence Aware Dictionary and Sentiment Reasoner* (VADER). Dataset yang diperoleh juga dilakukan *pre-processing* berupa konversi ke huruf kecil, penghapusan huruf, menghilangkan URL, tag, mention serta menggunakan *stop words* dan *lemmatization*. Setelah selesai di *pre-processing*, dilakukan penyematan kata menggunakan word2vec. Setiap model dilatih menggunakan tiga dataset yang terdiri dari 50.000 data, 100.000 data dan 200.000 data. Hasil terbaik dari semua model didapatkan ketika dilatih menggunakan 200.000 data. Akurasi, presisi, *recall* dan F1 *recall* paling baik didapatkan ketika data dilatih menggunakan model Bi-LSTM yakni sebesar 95.35%. Penelitian ini mendorong penulis untuk

menambahkan penyematan kata pada sentimen analisis vaksin COVID-19 di Indonesia.

Penyematan kata atau dalam hal ini disebut sebagai *word embedding* banyak dilakukan oleh peneliti dalam melakukan klasifikasi utamanya saat menggunakan metode *deep learning*. Seperti yang dilakukan oleh (Riza & Charibaldi, 2021) untuk mendeteksi emosi pada sosial media Twitter. Peneliti menggunakan *word embedding* FastText, GloVe dan Word2vec pada algoritma LSTM. Dataset di sosial media Twitter dari beberapa *influencer* dan data yang sedang *trending* pada satu minggu terakhir sejumlah 130 data dengan rincian 250 untuk emosi bahagia, 250 data emosi sedih, 200 data emosi ketakutan, 200 data emosi jijik dan 204 data emosi kemarahan. *Pre-processing* data yang digunakan dalam penelitian ini adalah konversi ke huruf kecil, menghapus *punction*, menghapus angka, *tokenizing*, *stop removal* dan *stemming*. *Stemming* data yang digunakan adalah Sastrawi. Dari hasil penelitian menggunakan LSTM dengan tiga *word embedding* yang berbeda, didapatkan hasil akurasi tertinggi pada LSTM dengan FastText sebesar 73%. Penggunaan *word embedding* GloVe dan FastText serta *stemming* data menggunakan *library* sastrawi sangat relevan dengan penelitian yang akan dilakukan.

Penggunaan *word embedding* FastText, GloVe dan Word2vec juga diterapkan oleh (Nurdin dkk., 2020) dalam melakukan perbandingan hasil klasifikasi teks namun menggunakan metode yang berbeda, yaitu CNN. Peneliti melakukan klasifikasi pada dua dataset berbahasa Inggris yang berbeda. Dari hasil eksperimen tersebut menunjukkan hasil *fastText embedding* lebih unggul daripada

dua *embedding* lain. Namun, perbedaan kinerja dari ketiga *embedding* tersebut menunjukkan hasil yang tidak terlalu signifikan.

FastText dan glove *embedding* juga digunakan untuk menganalisis hoax pada berita berbahasa Indonesia seperti yang dilakukan oleh (Adipradana dkk., 2021). Penelitian ini membandingkan dua *embedding* tersebut saat diterapkan menggunakan model-model Recurent Neural Network (RNN) seperti LSTM, Bi-LSTM, GRU dan Bi-GRU. Dari hasil penelitian tersebut fastText menghasilkan kinerja yang lebih baik dibandingkan glove pada hampir semua model RNN. Sedangkan dari keempat model RNN yang digunakan, Bi-LSTM dan Bi-GRU mendapatkan hasil akurasi paling baik yaitu 87,8% dan 93%.

Penyematan kata selain FastText, Glove dan Word2vec adalah BERT *embedding*. BERT *embedding* dan fastText digunakan oleh (D'Sa dkk., 2020) untuk melakukan *binary* dan *multi-class* klasifikasi untuk melakukan deteksi *speech toxic*. Metode yang digunakan untuk melakukan klasifikasi adalah CNN dan Bi-LSTM. Dari hasil penelitian diperoleh hasil terbaik saat melakukan *binary* klasifikasi menggunakan BERT *embedding* baik saat menggunakan metode CNN maupun Bi-LSTM yaitu sebesar 91,9%. Sedangkan saat melakukan *multi-class* klasifikasi, BERT yang dikombinasikan dengan Bi-LSTM mendapatkan hasil terbaik sebesar 72,4%.

Penelitian lain yang juga menggunakan BERT *embedding* adalah penelitian oleh (Pasaribu dkk., 2020) untuk meningkatkan akurasi sentimen ulasan makanan pada situs Amazon. Peneliti menggunakan Bi-LSTM yang dikombinasikan dengan BERT *embedding* pada dataset publik dengan jumlah 85.000 data. Dataset tersebut

di *pre-processing* dengan cara menghapus data yang sama, konversi ke huruf kecil, normalisasi distribusi dan vektorisasi menggunakan BERT *embedding*. Dari hasil penelitian diperoleh akurasi sebesar 93% dengan ukuran *embedding* sejumlah 300. Hasil ini lebih tinggi dibandingkan penelitian sebelumnya menggunakan model yang lain. Penyematan kata menggunakan BERT *embedding* yang dikombinasikan menggunakan Bi-LSTM perlu diuji menggunakan dataset lain.



2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian
Metode Bidirectional Lstm Menggunakan Word Embedding Untuk Analisis Sentimen Terhadap Vaksin Covid-19 Di Indonesia Pada
Media Sosial Twitter

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	<i>Sentiment Analysis of COVID-19 Vaccine in Indonesia using Naïve Bayes Algorithm</i>	Pristiyono, Mulkan Ritonga, Muhammad Ali Al Ihsan, Agus Anjar, Fauziah Hanum Rambe, 2020	Menggunakan algoritma Naïve Bayes untuk melakukan sentimen analisis vaksin COVID-19 di Indonesia dari media sosial Twitter.	Hasil penelitian yang dilakukan terhadap lebih dari 3000 <i>tweet</i> menggunakan algoritma Naïve Bayes didapatkan sentimen negatif sebesar 56%, sentimen positif sebesar 39% dan sentimen netral sebesar 1%.	Penelitian ini hanya melakukan klasifikasi saja, tanpa melakukan pengujian untuk mendapatkan akurasi dari metode yang digunakan.	Penelitian yang akan dilakukan selanjutnya menggunakan metode yang berbeda, yaitu Bi-LSTM dengan <i>word embedding</i> .
2	Peningkatan Akurasi Klasifikasi Sentimen Ulasan Makanan Amazon Dengan Bidirectional LSTM Dan Bert Embedding	David Junggu Manggala Pasaribu, Kusri, Sudarmawan, 2020	Penelitian ini bertujuan untuk meningkatkan akurasi algoritma Bi-LSTM dengan menggunakan BERT Embedding pada dataset public ulasan makan amazon.	Dari hasil pengujian model Bi-LSTM dengan BERT embedding menghasilkan akurasi yang lebih baik sebesar 93% daripada word embedding biasa, seperti Word2Vec dan GloVe dengan deep learning CNN dan LSTM.	Perlu dilakukan variasi pengujian pengukuran vector untuk mengetahui pola hasil akurasi sebanyak lebih dari 500 vektor.	Penelitian yang akan dilakukan menggunakan dataset yang berbeda yang berasal dari sosial media Twitter dengan tweet berbahasa Indonesia dalam melakukan sentiment analisis.

Tabel 2.1. Matriks literatur review dan posisi penelitian (lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3.	<i>Sentiment Analysis on Twitter using Neural Network: Indonesian Presidential Election 2019 Dataset</i>	Ahmad Fathan Hidayatullah, Siwi Cahyaningtyas, Anisa Miladya Hakim, 2020	Penelitian ini bertujuan untuk mengklasifikasikan sentiment dari Twitter pada pemilihan presiden 2019 menggunakan beberapa varian Neural Network, seperti Convolutional Neural Network (CNN), Long short-term memory (LSTM), CNN-LSTM, Gated Recurrent Unit (GRU) -LSTM and Bidirectional LSTM. Selain itu penelitian ini juga melatih dataset yang ada menggunakan machine learning, seperti Support Vector Machine (SVM), Logistic Regression (LR) and Multinomial Naïve Bayes (MNB).	Dari hasil penelitian diperoleh hasil bahwa <i>deep learning</i> memiliki akurasi lebih baik dibandingkan <i>traditional machine learning</i> . Serta dari hasil percobaan, Bi-LSTM memiliki hasil lebih baik dibandingkan metode <i>deep learning</i> yang lain yaitu dengan akurasi 84,0%.	Tahap <i>pre-processing</i> pada dataset yang digunakan sudah ada, tidak digambarkan dengan jelas.	<p>Pada penelitian selanjutnya, akan dijelaskan dan digambarkan dengan rinci alur preprocessing pada dataset yang akan digunakan sehingga dapat menjadi gambaran bagi peneliti yang lain walaupun dataset yang digunakan berbeda.</p> <p>Penelitian selanjutnya akan menggunakan metode Bi-LSTM, sesuai dengan hasil kesimpulan pada penelitian sebelumnya yang menyatakan bahwa metode Bi-LSTM memiliki hasil lebih baik daripada metode yang lain.</p>

Tabel 2.1. Matriks literatur review dan posisi penelitian (lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	<i>Twitter sentiment analysis using deep learning models</i>	Arnab Roy, Muncendra Ojha, 2020	Penelitian ini bertujuan membandingkan klasifikasi sentimen pada social media Twitter menggunakan Google BERT, Bi-LSTM dan Convolutional Neural Network. Menggunakan SemEval-2016 Twitter dataset.	Dari 20.632 tweet yang dari SemEval-2016 dataset diuji menggunakan BERT Model, Bi-LSTM Model dan CNN Model menghasilkan akurasi tertinggi pada model BERT sebesar 61,1%, model Bi-LSTM dengan akurasi 60,02 % dan akurasi terendah pada CNN model dengan 59,2%.	Hasil kinerja dari pengklasifikasian terbuat akan lebih baik jika menggunakan Bahasa Inggris dengan struktur tata Bahasa yang jelas dan benar.	Pada penelitian yang akan dilakukan menggunakan dataset yang berbeda, yaitu dataset mengenai vaksin covid-19 di Indonesia dari social media Twitter. Bahasa yang digunakan dalam dataset penelitian ini adalah Bahasa Indonesia. Penelitian selanjutnya akan melakukan klasifikasi menggunakan BERT embedding dan Bi-LSTM
5	<i>Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text.</i>	M. Alfa Riza, Novrido Charibaldi, 2021	Penelitian ini, mendeteksi emosi dari teks yang berasal dari Twitter menggunakan metode LSTM menggunakan penyematan kata FastText.	Deteksi emosi diklasifikasikan menjadi 6, yaitu bahagia, sedih, jujuk, marah, takut dan terkejut menggunakan model LSTM-Word2vec dan LSTM-Fast Text mendapatkan hasil akurasi, presisi, recall, dan F-1 score terbaik sebesar 73,15%.	Dalam penelitian selanjutnya, diharapkan dapat menerapkan metode deep learning lain seperti, CNN, Bi-LSTM serta lebih banyak data yang digunakan.	Penelitian yang selanjutnya, fitur Word2Vec dan FastText akan dikombinasikan dengan algoritma Bi-LSTM untuk menentukan sentiment analisis vaksin covid 19 pada social media twitter.

Tabel 2.1. Matriks literatur review dan posisi penelitian (lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	Perbandingan Kinerja <i>Word Embedding</i> Word2Vec, Glove dan FastText pada Klasifikasi Teks	Arliyanti Nurdin, Bernadus Anggo Seno, Anugryani Bustamin, Zaenal Abidin. 2020	Tujuan dari penelitian ini adalah membandingkan kinerja <i>Word Embedding</i> Word2Vec, Glove dan FastText untuk klasifikasi teks menggunakan algoritma <i>Convolutional Neural Network</i> pada dataset 20 <i>newsgroup</i> dan <i>Reuters</i> .	Kinerja terbaik dari hasil penelitian diperoleh dengan menggunakan <i>Word Embedding</i> FastText dengan F-1 score sebesar 71,5% pada dataset <i>Reuters</i> . Namun, tidak terdapat perbedaan kinerja yang signifikan dibanding <i>Word Embedding</i> yang lain.	Penggunaan setiap <i>Word Embedding</i> sangat bergantung dengan dataset yang digunakan dan permasalahan yang akan diselesaikan.	Penelitian yang akan dilakukan selanjutnya, tidak hanya membandingkan Glove dan FastText namun juga menambahkan <i>Word Embedding</i> BERT.
7	<i>Hoax Analyzer for Indonesian News using RNNs with FastText and Glove Embedding</i>	Ryan Adiprana, Bagas Pradipta Nagoya, Ryan Suryadi, Derwin Subartono. 2021	Penelitian ini bertujuan untuk membandingkan <i>word embedding</i> FastText dan GloVe terhadap empat <i>Deep Neural Network</i> (DNN) models yaitu LSTM, Bi-LSTM, GRU dan Bi-GRU untuk mengklasifikasi berita palsu berbahasa Indonesia.	Beberapa kesimpulan yang didapatkan dari penelitian tersebut adalah pertama, <i>word embedding</i> FastText memiliki performa yang lebih baik dibandingkan GloVe. Kedua, <i>word embedding</i> FastText yang dikombinasikan dengan Bi-GRU memperoleh kinerja yang paling baik dibandingkan dengan kombinasi yang lain.	Peneliti mengalami masalah terkait dataset dimana sangat sulit untuk menemukan berita yang bersifat satir. Sehingga, akhirnya penelitian menggunakan berita berbahasa Inggris yang diterjemahkan ke dalam Bahasa Indonesia. Hal ini tentunya dapat berpengaruh terhadap hasil penelitian.	Penelitian yang akan dilakukan selanjutnya, tidak hanya membandingkan Glove dan FastText namun juga menambahkan <i>Word Embedding</i> BERT. Penelitian selanjutnya tidak menggunakan banyak model DNN. Namun, hanya satu yaitu Bi-LSTM.

Tabel 2.1. Matriks literatur review dan posisi penelitian (lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
8	<i>BERT ada fastText Embeddings for Automatic Detection of Toxic Speech</i>	Ashwin Geet d'Sa, Irina Illina, Dominique Fohr	Tujuan dari penelitian ini adalah mengklasifikasikan <i>speech toxic</i> menggunakan <i>word embedding fastText</i> , BERT serta BERT fine-tuning dan teknik <i>deep learnig</i> berupa CNN dan Bi-LSTM. Klasifikasi yang dilakukan, terbagi menjadi dua jenis, yaitu <i>binary</i> klasifikasi dan <i>multi-class</i> klasifikasi.	Kinerja BERT <i>embedding</i> dan <i>fastText embedding</i> dengan Bi-LSTM memiliki hasil yang sama saat digunakan untuk melakukan <i>binary</i> klasifikasi. Sedangkan, saat melakukan <i>multi-class</i> klasifikasi, BERT <i>embedding</i> dengan metode Bi-LSTM menghasilkan kinerja terbaik yaitu sebesar 72.4%.	BERT Fine-Tuning lebih baik digunakan saat melakukan <i>binary classification</i> , perlu penelitian lebih lanjut saat melakukan <i>multiclass classification</i> .	Penelitian yang akan dilakukan selanjutnya, tidak mengambil topik mengenai <i>speech detection</i> namun mengenai sentimen analisis vaksin COVID-19 dengan <i>multi-class</i> klasifikasi dan 3 jenis <i>word embedding</i> dan 1 metode <i>deep learning</i> .
9	<i>Sentiments Analysis of Indonesian Tweet About Covid-19 Vaccine Using Support Vector Machine and Fasttext Embedding</i> (Wibowo & Musdholifah, 2021)	Dikih Arif Wibowo, Aina Musdholifah (ISRITI, 2021)	Penelitian ini berfokus pada melakukan sentimen analisis COVID-19 berdasarkan postingan di Twitter menggunakan SVM dan FastText embedding dibandingkan dengan Multinomial Naive Bayes.	Pada saat menggunakan metode SVM dan FastText embedding akurasi yang dihasilkan sebesar 88.1% sedangkan saat menggunakan Multinomial Naive Bayes dan FastTex akurasi yang diperoleh sebesar 73.7%	Perlu dilakukan optimalisasi proses <i>preprocessing</i> ketika menggunakan data kamus <i>slang word</i> yang kecil. Sehingga perlu menggunakan kamus <i>slang word</i> Bahasa Indonesia dengan kosa kata lebih banyak.	Penelitian selanjutnya juga akan melakukan sentimen analisis, namun bukan mengenai COVID-19 melainkan mengenai vaksin COVID-19 dengan menggunakan metode Bi-LSTM serta <i>word embedding</i> FastText, GloVe dan BERT

Tabel 2.1. Matriks literatur review dan posisi penelitian (lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
10	<i>Hate Speech Detection on Indonesian Instagram Comments using FastText Approach</i> (Pratiwi dkk., 2019)	Nur Indah Pratiwi, Indra Budi, dan Ika Alfina, ICACIS 2018	Melakukan klasifikasi ujaran kebencian pada komentar di Instagram menggunakan FastText.	Dari hasil penelitian dalam melakukan klasifikasi ujaran kebencian pada komentar berbahasa Indonesia diperoleh akurasi terbaik sebesar 63.9% saat menggunakan metode FastText.	Dataset yang digunakan dalam penelitian ini jumlahnya masih terlalu sedikit, sehingga disarankan untuk membangun data yang lebih besar lagi untuk penelitian selanjutnya. Selain itu, membangun kumpulan frasa dengan kata-kata <i>offensive</i> juga perlu untuk dilakukan karena kata-kata tersebut sering muncul dalam ujaran kebencian.	Penelitian yang akan dilakukan menggunakan Bi-LSTM untuk klasifikasi dan sentimen analisis vaksin COVID-19 sebagai studi kasusnya.

2.3. Landasan Teori

2.3.1. Natural Language Processing (NLP)

NLP adalah aplikasi pembelajaran mesin dan teknik komputasi lainnya untuk memahami dan merepresentasikan teks lisan dan tulisan (Bhattacharjee, 2018). Lebih lanjut (Bhattacharjee, 2018) menguraikan beberapa hal yang dapat dipecahkan menggunakan NLP, antara lain:

- a. Pemodelan topik: Secara umum, teks berhubungan dengan topik dan eperti akan membantu pembaca untuk mendapatkan inti dari dokumen dan deskripsi tingkat tinggi tentang apa yang terjadi.
- b. Klasifikasi kalimat: mengklasifikasikan teks ke dalam label yang berbeda.
- c. *Machine Translator*, merupakan model terjemahan pembelajaran mesin yang mampu melatih dari beberapa sumber dengan tingkat daya prediksi yang tinggi.
- d. Sistem tanya jawab (QA): Fokus di sini adalah membangun sistem yang secara otomatis menjawab pertanyaan berdasarkan pertanyaan yang diajukan orang dalam bahasa alami.
- e. Analisis sentimen: tentang memahami kebutuhan dan maksud yang dibagikan pengguna saat membicarakan sesuatu.
- f. Deteksi nama entitas: untuk mengekstrak dan mengklasifikasikan entitas atau informasi spesifik sesuai dengan beberapa kategori yang telah ditentukan sebelumnya, seperti orang, organisasi, geografi, dan sebagainya.

Dalam bidang analisis sentimen, NLP dapat menggunakan berbagai sumber data yang meliputi sosial media, situs berita, komentar pengguna pada sebuah website ataupun akun sosial media produk. Menurut (Farzindar & Inkpen, 2020),

analisis sentimen yang menggunakan data yang bersumber dari sosial media mempunyai tantangan tersendiri karena pengguna di sosial media terkadang menggunakan susunan kalimat yang tidak baku, penulisan kata yang tidak sesuai kamus, maupun gaya bahasa yang tidak formal. Umumnya model NLP dilatih menggunakan bahasa yang formal dan sesuai kaidah. Terdapat dua pendekatan yang dapat dilakukan agar teks dari sosial media dapat diklasifikasi dengan benar oleh model, yang pertama adalah dengan melatih ulang model dengan teks dari sosial media dan yang kedua dengan melakukan normalisasi teks sehingga mendekati bahasa yang digunakan untuk melatih model sebelumnya (Farzindar & Inkpen, 2020).

2.3.2. Analisis Sentimen

Analisis sentimen merupakan salah satu bidang ilmu yang digunakan untuk mengetahui persepsi atau pendapat masyarakat terhadap sesuatu, baik produk, kebijakan, sosok seseorang dan lain-lain, dengan cepat menggunakan teknik-teknik NLP. Menurut (Cambria dkk., 2017) analisis sentimen bertujuan untuk menentukan sikap seorang penulis atau pembicara sehubungan dengan beberapa topik atau polaritas kontekstual keseluruhan dari sebuah dokumen. Selanjutnya (Satapathy dkk., 2017) menjelaskan bahwa pekerjaan dasar dalam sebuah analisis sentimen yaitu mengklasifikasikan *polarity* dari sebuah dokumen, kalimat atau teks, apakah pendapat yang diekspresikan pada teks tersebut positif, negatif atau netral.

Penulis (Pozzi dkk., 2017) menguraikan beberapa tugas yang dilakukan pada analisis sentimen, diantaranya: klasifikasi subjektif, klasifikasi polaritas,

ringkasan opini, deteksi sarkasme, pemegang opini dan ekstraksi waktu, resolusi konferensi dan disambiguasi arti kata, pembuatan leksikon sentimen, pencarian dan pengambilan opini, dan deteksi spam opini. Analisis sentimen dasar umumnya menggunakan dua teknik pertama, yaitu s klasifikasi subjektif dan klasifikasi polaritas. Subjektifitas digunakan untuk mengetahui apakah sebuah kalimat mengekspresikan sentimen atau tidak (Poria dkk., 2018). Jika nilai subjektifitas pada sebuah kalimat tinggi, maka kalimat tersebut bisa dianggap sebagai pendapat pribadi penulis sedangkan jika nilai subjektifitas rendah maka dapat dianggap kalimat tersebut adalah fakta. Dengan kata lain, kalimat tersebut netral. Kalimat dengan subjektifitas tinggi yang mengekspresikan sentimen, kemudian diklasifikasikan menjadi negatif atau positif berdasarkan nilai polaritas. Nilai polaritas berkisar dari -1 sampai dengan 1. Nilai polaritas yang mendekati nilai -1 maka sentimen dianggap negatif dan sebaliknya. Nilai polaritas yang mendekati angka 0 dianggap netral.

2.3.3. Pre-Processing

Pre-processing merupakan salah satu tahapan untuk mempersiapkan data sebelum digunakan ke dalam model. Penulis (Hidayatullah dkk., 2021) menguraikan bahwa terdapat beberapa teknik *pre-processing* yang dapat digunakan pada teks yang bersumber dari sosial media, diantaranya menghapus *username*, *hashtag*, URL, *retweet*, *punctuation*, huruf yang terulang dalam sebuah kata, tanda spasi yang berlebihan, *stopwords* dan *tweet* yang duplikat. Selanjutnya penulis

(Hidayatullah dkk., 2021) juga menggunakan teknik pre-processing lain seperti *case folding*, *word normalization* dan *stemming*.

Case folding dapat digunakan untuk menyeragamkan *case* huruf pada sebuah kalimat (Salam dkk., 2018) sehingga lebih mudah dipahami oleh model. Selanjutnya data berbahasa Indonesia biasanya dilakukan proses *stopword removal* dan *stemming*. *Stopword removal* berfungsi untuk menghapus *common word* yang tidak bermakna dan sering muncul sedangkan *stemming* adalah teknik mengubah kalimat berimbuhan menjadi kata dasar sehingga mempunyai makna yang sama dengan kata imbuhan lain dengan kata dasar yang sama (Riza & Charibaldi, 2021). Setiap bahasa mempunyai struktur imbuhan yang berbeda-beda. Khusus untuk Bahasa Indonesia, terdapat sebuah library yang umum digunakan dalam melakukan *stopword removal* dan *stemming*, yaitu *library* Sastrawi (Robbani, 2016) yang menggunakan algoritma Nazief dan Andriani (Nazief, B. A. A. & Adriani, 2005).

2.3.4. Word Embedding

Word embedding merupakan teknik NLP yang mengubah sebuah kata dasar menjadi vector bernilai *real* (Poria dkk., 2018). Istilah lain yang mengacu ke hal yang sama adalah *Word Vector*. Penulis (Riza & Charibaldi, 2021) menjelaskan *word embedding* sebagai teknik untuk memetakan kata berdasarkan kamus dan kemudian mengubahnya menjadi angka bernilai real. Beberapa teknik telah diusulkan oleh peneliti dalam melakukan *word embedding*, diantaranya *Word2vec* (Mikolov dkk., 2013), *GloVe* (Pennington dkk., 2014), *Fast Text* (Bojanowski dkk., 2017) dan *BERT Embedding* (Devlin dkk., 2019).

2.3.5. GloVe

GloVe (Global Vectors for Word Representation) adalah sebuah algoritma yang diperkenalkan oleh (Pennington dkk., 2014). GloVe merupakan peningkatan dari *matrix factorization-based representations of words* dan model Skip-gram dimana metode *matrix factorization-based representations of words* tidak terlalu baik dalam merepresentasikan kata sehubungan dengan sifat analognya (Beysolow II, 2018). GloVe berbasis *factorization technique* (Poria dkk., 2018). GloVe disebut sebagai modifikasi dari Word2vec (Mengistie & Kumar, 2021).

Skip-Gram dan CBoW melatih dengan cara yang berbeda dari metode faktorisasi berbasis matriks. Misal, teknik LDA digunakan untuk membuat *topic modelling*, teks sebelumnya harus diproses dulu dengan cara meng-*encode* setiap kata dengan informasi statistik yang mewakili kata dalam konteks keseluruhan teks. Dengan demikian, *one-hot encode vector* tidak dapat memahami jenis kompleksitas yang sama saat menggunakan metode Skip-Gram dan CBoW. GloVe sangat efisien dalam menangkap detail semantic sebuah kata di dalam representasi vektornya, namun sangat tidak efisien dalam melakukan analisis sentimen (Poria dkk., 2018). Beberapa solusi telah diusulkan oleh para peneliti dengan cara memodifikasi model C&W (Collobert dkk., 2011).

Metode GloVe dinyatakan oleh (Pennington dkk., 2014) ke dalam fungsi di bawah ini:

$$J = \sum_{i,j=1}^V f(X_{ij}) (w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2 \quad (1)$$

Dimana:

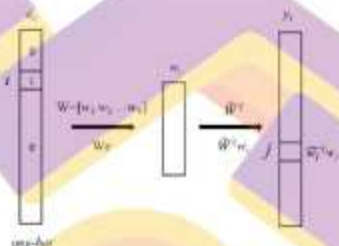
V = ukuran kosa kata

b = bias

W = Weight

X = kata yang diproses pada matrix $i \times j$

Arsitektur GloVe digambarkan pada gambar 1 seperti berikut. (Liu dkk., 2019)



Gambar 2.1. Arsitektur Glove (Liu dkk., 2019)

2.3.6. FastText

FastText adalah *library* milik Facebook yang digunakan untuk membantu menghasilkan representasi kata yang efisien dan memberi dukungan untuk klasifikasi teks (Bojanowski dkk., 2017). FastText merupakan model perbaikan (*update*) dari Skip-gram yang sudah dilatih sebelumnya (*pre-trained*) (Mengistie & Kumar, 2021). Hal yang sama juga dinyatakan oleh (D'Sa dkk., 2020). FastText umumnya digunakan untuk menyelesaikan permasalahan *sentence classification* dan *word representation* agar lebih efisien dan lebih cepat dibandingkan metode Word2vec dan Glove (D'Sa dkk., 2020).

FastText menggunakan pendekatan yang berdasar pada Skip-gram dimana setiap kata direpresentasikan sebagai sebuah bag of character n-gram (Joulin dkk.,

2016). FastText melihat korpus teks yang disediakan dan membentuk model ruang vektor berdimensi tinggi, di mana ia mencoba merangkum makna sebanyak mungkin. Tujuan dari pembuatan ruang vektor adalah agar vektor-vektor dari kata-kata yang serupa harus saling berdekatan (Bojanowski dkk., 2017). Dalam FastText, vektor kata ini kemudian disimpan dalam dua file, mirip dengan apa yang Anda lihat dalam klasifikasi teks: file `.bin` dan file `.vec`. (Bhattacharjee, 2018).

Selanjutnya, penulis (Bojanowski dkk., 2017) menyatakan bahwa penggunaan Softmax pada Skip-gram model karena hanya mampu melakukan prediksi pada satu konteks. Penulis kemudian mengusulkan metode FastText yang direpresentasikan pada fungsi di bawah.

$$\sum_{t=1}^T \left[\sum_{c \in C_t} \ell(s(w_t, w_c)) + \sum_{n \in N_{t,c}} \ell(-s(w_t, n)) \right] \quad (2)$$

Dimana:

s = *scoring function*

w = *weight*

$\ell = \log(1 + e^{-x})$

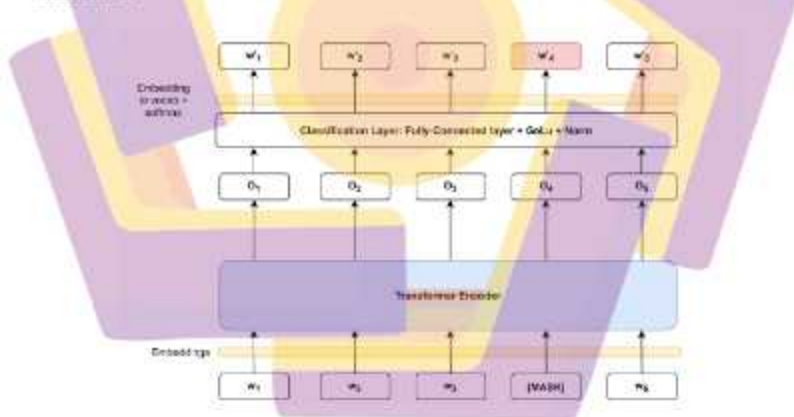
n = jumlah kosa kata (*vocabulary*)

2.3.7. BERT Embedding

BERT (*Bidirectional Encoder Representations from Transformer*) merupakan model yang dipublikasikan oleh Google pada tahun 2018 (Devlin dkk., 2018). Google mempublikasi beberapa *pre-trained* model BERT di situs Github sehingga dapat diakses oleh peneliti di seluruh dunia. BERT sendiri dapat melakukan *word embedding* dengan mencari arti kontekstual sebuah kata dari

sebuah kalimat (Rothman, 2021). BERT berdasar pada metode transformer dan mengimplementasikan attention mechanism (D'Sa dkk., 2020). *Attention* adalah teknik untuk menemukan hubungan antar kata dalam sebuah kalimat (Vaswani dkk., 2017).

Hal ini memungkinkan bagi BERT untuk memahami konteks sebuah kata dengan memperhatikan kata yang ada sebelum dan sesudah kata tersebut. BERT mempunyai kelebihan jika dibandingkan dengan FastText dimana BERT menggunakan teknik *word-piece tokenization* (D'Sa dkk., 2020) sehingga BERT mampu memecah paraprse yang tidak ada di dalam kamus atau corpus yang digunakan menjadi 2 atau lebih kata. Arsitektur BERT digambarkan seperti pada Gambar 2.



Gambar 2.2. Arsitektur BERT (Roy & Ojha, 2020)

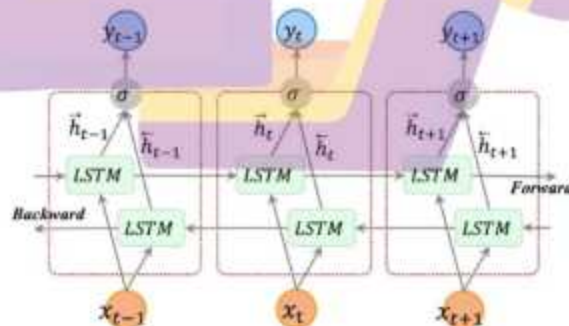
2.3.8. Bidirectional LSTM (Bi-LSTM)

Bi-LSTM merupakan pengembangan dari LSTM yang digabungkan dengan Bidirectional layer (Graves & Schmidhuber, 2005). Pada dasarnya Bi-LSTM

terbentuk dari dua LSTM, yaitu *backward* LSTM dan *forward* LSTM (Z. Chen dkk., 2020). Gabungan dari keduanya akan menangkap informasi dari 2 arah yang berbeda. Pola seperti ini cukup baik untuk mengenali struktur dalam sebuah kalimat karena dapat diproses secara sekuensial dan dapat dipahami setiap kata secara berurutan.

Lapisan *backward* memahami dan memproses kata terakhir hingga kata pertama, sedangkan lapisan *forward* memahami dan memproses dari kata pertama hingga kata terakhir. Dengan adanya 2 arah yang saling berlawanan ini maka model dapat memahami dan memproses kata sebelumnya dan kata sesudahnya, sehingga proses pembelajaran oleh model akan semakin baik (Novalita dkk., 2019).

Dapat dilihat pada Gambar 3 (Cui dkk., 2017) yang merupakan arsitektur Bi-LSTM, setiap *hidden unit* keluaran dari unit h_t pada lapisan *backward* dan *forward* digabungkan untuk membentuk nilai fitur kata tersebut dengan ukuran lebih panjang daripada menggunakan LSTM biasa sehingga akan mengklasifikasikan dengan lebih akurat.



Gambar 2.3. Arsitektur Bi-LSTM (Cui dkk., 2017)

2.3.9. Metrik Pengukuran

Dalam penelitian yang menggunakan *machine learning* dan *deep learning*, umumnya menggunakan 4 macam metrik untuk mengukur kinerja model, yaitu *accuracy*, *precision*, *recall* dan *f1 score* (Grandini dkk., 2020). *Accuracy* adalah perbandingan antara jumlah prediksi yang benar pada semua label dengan jumlah semua data dengan asumsi label pada semua kelas seimbang. *Precision* adalah rasio antara label yang diprediksi dengan benar dan sampel yang diprediksi menjadi data positif.

Metrik ini mewakili tingkat presisi yang ditebak model terhadap label pada dataset. *Recall* didefinisikan sebagai rasio antara sampel positif benar dan sampel positif total dalam dataset. *Recall* mencerminkan kemampuan model untuk mengenali label negatif. *F1 score* adalah *harmonic mean* atau antara *Precision* dan *Recall*. Matriks *F1 score* sering digunakan pada kasus dimana dataset tidak seimbang (Tharwat, 2018). Gambar 2.4 merupakan ilustrasi *confusion matrix* untuk klasifikasi *multiclass*.

		True Class		
		A	B	C
Predicted Class	A	TP_A	E_{BA}	E_{CA}
	B	E_{AB}	TP_B	E_{CB}
	C	E_{AC}	E_{BC}	TP_C

Gambar 2.4 Ilustrasi *Confusion Matrix* untuk Klasifikasi *Multiclass* (Tharwat, 2018)

Berdasarkan Gambar 2.4, *confusion matrix* untuk setiap kelas dapat didefinisikan seperti pada gambar 2.5.

	Label				Label				Label						
	Pos	Neg	Neu		Pos	Neg	Neu		Pos	Neg	Neu				
Prediksi	Pos	TP	FP	FP	Prediksi	Pos	TN	FN	TN	Prediksi	Pos	TN	TN	FN	
	Neg	FN	TN	TN		Neg	FP	TP	FP		Neg	TN	TN	FN	FN
	Neu	FN	TN	TN		Neu	TN	FN	TN		Neu	FP	FP	TP	TP

Confusion Matrix kelas Positive
Confusion Matrix kelas Negative
Confusion Matrix kelas Neutral

Gambar 2.5. Confusion Matrix untuk setiap kelas

Dengan demikian, maka dapat diketahui confusion matrix untuk setiap kelas dari gambar 2.4 dan 2.5 yang direpresentasikan pada formula 3 hingga 8.

$$TP_{Pos} = TP_A, TP_{Neg} = TP_B, \text{ dan } TP_{Neu} = TP_C \quad (3)$$

$$FP_{Pos} = E_{BA} + E_{CA}, FP_{Neg} = E_{AB} + A_{CB}, \text{ dan } FP_{Neu} = E_{AC} + E_{BC} \quad (4)$$

$$FN_{Pos} = E_{AB} + E_{AC}, FN_{Neg} = E_{BA} + E_{BC}, \text{ dan } FN_{Neu} = E_{CA} + E_{CB} \quad (5)$$

$$TN_A = TP_B + E_{CB} + E_{BC} + TP_C \quad (6)$$

$$TN_B = TP_A + E_{CA} + E_{AC} + TP_C \quad (7)$$

$$TN_C = TP_A + E_{BA} + E_{AB} + TP_B \quad (8)$$

Selanjutnya, untuk menghitung performa masing-masing kelas, dapat digunakan formula 9 hingga 12, dimana formula tersebut merupakan formula yang umum digunakan pada binary classification.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (9)$$

$$Precision = \frac{TP}{FP+TP} \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

$$F1 = 2 \frac{Precision \cdot Recall}{Precision + Recall} \quad (12)$$

Pada penelitian ini, performa model secara keseluruhan diketahui dengan cara membuat rerata dari metrik-metrik pada setiap kelas tanpa memperdulikan jumlah data pada setiap kelas. Hal ini digambarkan pada formula 13 sampai 16.

$$\text{Accuracy Model} = (\text{Accuracy}_{\text{Pos}} + \text{Accuracy}_{\text{Neg}} + \text{Accuracy}_{\text{Neu}}) / 3 \quad (13)$$

$$\text{Precision Model} = (\text{Precision}_{\text{Pos}} + \text{Precision}_{\text{Neg}} + \text{Precision}_{\text{Neu}}) / 3 \quad (14)$$

$$\text{Recall Model} = (\text{Recall}_{\text{Pos}} + \text{Recall}_{\text{Neg}} + \text{Recall}_{\text{Neu}}) / 3 \quad (15)$$

$$\text{F1 Score Model} = (\text{Accuracy}_{\text{Pos}} + \text{Accuracy}_{\text{Neg}} + \text{Accuracy}_{\text{Neu}}) / 3 \quad (16)$$

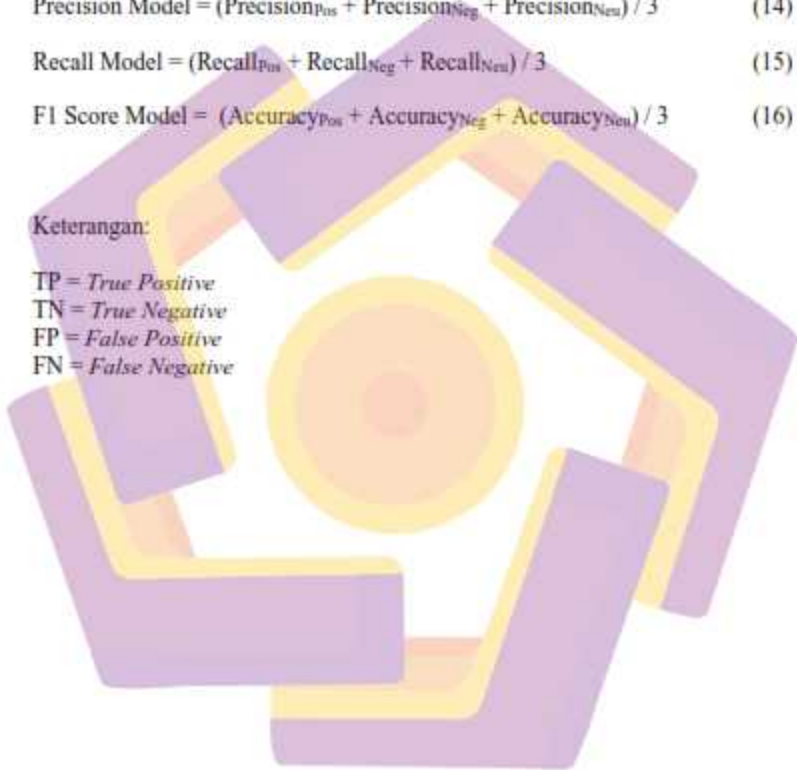
Keterangan:

TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

FN = *False Negative*



BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian ini adalah penelitian eksperimental. Dimana penelitian ini melakukan pengujian tingkat akurasi yang tertinggi menggunakan metode Bi-LSTM+FastText, Bi-LSTM+GloVe dan Bi-LSTM+BERT dengan jumlah data set yang sama. Pengujian ini dilakukan untuk mengetahui metode yang lebih akurat dan tepat dalam melakukan sentimen analisis pada data twitter melalui kata vaksin COVID-19.

Penelitian ini bersifat deskriptif, karena menggambarkan suatu objek yang akan diteliti dan menjabarkan hasil pengujian-pengujian yang dilakukan pada data set yang ada untuk dapat diketahui metode mana yang memiliki akurasi, presisi, recall, F1 score terbaik.

Penelitian ini menggunakan pendekatan kuantitatif yang nantinya hasil dari penelitian ini berupa angka, grafik hasil eksperimen metode Bi-LSTM+FastText, Bi-LSTM+GloVe dan Bi-LSTM+BERT.

3.2. Metode Pengumpulan Data

Data yang digunakan pada penelitian ini merupakan jenis data primer yang dikumpulkan langsung menggunakan teknik *scrapping* dari sosial media Twitter. *Scrapping* data dilakukan menggunakan Twitter API dengan tool bernama

SNScrape dengan kata kunci “vaksin covid”. *Scrapping* data dilakukan pada tweet yang diposting dari bulan September 2020 hingga Juni 2021.

3.3. Metode Analisis Data

Sebelum melakukan analisis data, tweet yang didapatkan dilakukan *pre-processing* data terlebih dahulu, seperti mengkonversi data menjadi huruf kecil (*case folding lower case*) menghilangkan karakter, menghapus *url*, menghapus *punctuation*, menghapus tag HTML, menghapus *stopword*, *stemming*, *slang word* dan identifikasi frasa.

Data yang telah dilakukan *pre-processing* kemudian akan diberikan label dan dianalisis menggunakan metode Bi-LSTM+Glove, Bi-LSTM+FastText dan Bi-LSTM+ BERT untuk didapatkan tingkat akurasi, presisi, recall dan F1 scorenya. Hasil pengukuran tersebut akan dijadikan acuan atau pedoman dalam menentukan hasil atau analisis sentimen pada penelitian ini.

3.4. Dataset

Dataset yang digunakan dalam penelitian ini didapatkan dari sosial media Twitter menggunakan metode *scrapping* menggunakan *tool* SNScrape (JustAnotherArchivist, 2021). Data diambil dari bulan September 2020 sampai dengan Juni 2021 menggunakan kata kunci pencarian “vaksin covid”. Pengambilan data diambil dari bulan September 2020 karena pada bulan sebelumnya belum ditemukan *tweet* yang mengandung kata kunci “vaksin covid”.

Dari hasil *scrapping* didapatkan data sebanyak 262.306 *tweet*. Gambar 3.1. merupakan hasil dari *scrapping* data. Data sejumlah tersebut perlu dilakukan filtering data dimana data yang tidak memenuhi kriteria harus dihapus. Kriteria data yang harus dihapus adalah *tweet* yang berasal dari akun lembaga, organisasi dan akun berita. Penghapusan *tweet* dari akun-akun tersebut perlu dilakukan karena tidak menggambarkan opini dari masyarakat. Selanjutnya, *tweet* yang sama serta kolom yang tidak digunakan juga dihapus. Terakhir, *tweet* yang tidak menggunakan Indonesia juga harus dihapus agar Bahasa yang digunakan menjadi seragam, yaitu Bahasa Indonesia. Dari hasil filtering data tersebut, tersisa 64.332 baris data dan diekspor ke dalam bentuk .csv.

No	Account	Text
1	sarkawati	ipn best ng gawe di tim lab ipn
2	JemberBapak	Jember Penangan ini berparah labang ipn
3	ALWARY	Salah satu... Suka...
4	rahmatramani	Orang yang...
5	yulius_mak	Bekas...

Gambar 3.1. Contoh data hasil *scrapping* data

No	Account	Text
1	Dipendit17700	...
2	Liberalisme	...
3	Maman	...
4	Mamak	...

Gambar 3.2. Contoh data hasil *filtering*

Tahap selanjutnya dari pengolahan dataset ini adalah melakukan pelabelan. Proses pemberian label dilakukan secara manual melibatkan pakar dari bidang psikologi. Pemberian label diklasifikasikan menjadi 3 jenis, yaitu netral, positif dan negatif. Data yang telah selesai diberi label akan mulai memasuki tahap *pre-processing*. Ada beberapa tahapan *preprocessing* yang dilakukan, dimulai dari *case folding*, *remove punctuation*, *url* dan *mention*, *stopword removal*, *stemming*, *slang normalization* serta identifikasi frasa. Secara singkat, alur *preprocessing* data dapat dilihat pada gambar 3.2.

- a. *Case folding* merupakan tahapan untuk menyeragamkan karakter, pada penelitian ini semua karakter diubah menjadi huruf kecil (*lower case*).
- b. *Remove punctuation* digunakan untuk menghapus karakter-karakter khusus seperti "!", "#", "\$", "%", "&", "(", "*", "+", "-", ".", ":", ";", "<", "=", ">", "?", "@", "[", "\", "]", "^", "_", "`", "{", "|", "}" dan "~". *Remove url* dan *mention* dilakukan untuk menghapus *link* dan *mention* yang ada di dalam data.
- c. *Stopword removal* berfungsi untuk menghapus *common word* yang tidak bermakna dan sering muncul, seperti "adalah, ialah" kata penghubung seperti "dan, atau" dan kata ganti orang seperti "aku, kamu dan dia". Penggunaan *stopword removal* menurut (Hidayatullah, 2016) dapat meningkatkan performa klasifikasi meski tidak terlalu signifikan. Kamus *stopword* yang digunakan dalam penelitian ini menggunakan *library* Sastrawi (Nazief, B. A. A. & Adriani, 2005).

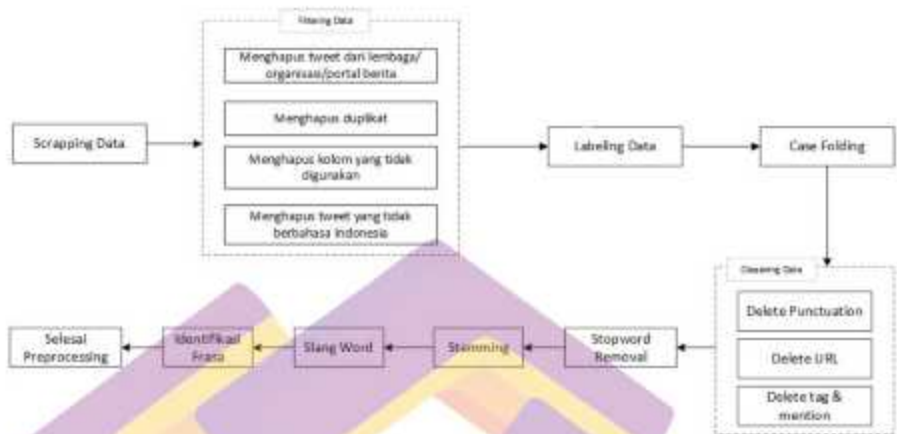
- d. *Stemming* merupakan salah satu tahapan dimana sebuah kata diubah menjadi bentuk dasar dengan menghilangkan imbuhan, baik awalan, sisipan, akhiran serta awalan dan akhiran. Proses *stemming* dilakukan dengan menggunakan algoritma Nazief-Andriani (Nazief, B. A. A. & Adriani, 2005).
- e. *Slang normalization* merupakan tahapan mengubah bentuk kata *slang* (kata tidak baku/formal) menjadi kata baku sesuai aturan Kamus Besar Bahasa Indonesia (KBBI). Proses *slang normalization* ini dilakukan menggunakan kumpulan kata dari kamus Alay milik (Aliyah Salsabila dkk., 2019). Contoh penerapan *slang normalization* adalah sebagai berikut, kata “gak” diubah menjadi “tidak”, kata “hadehh” diubah menjadi “aduh” dan kata “pke” menjadi “pakai”.
- f. Identifikasi frasa dilakukan untuk mengidentifikasi sebuah frasa (gabungan dua kata atau lebih yang memiliki satu makna) untuk menjadi satu kesatuan. Proses ini dilakukan dengan tujuan ketika dilakukan proses tokenisasi, frasa yang ada tidak diperlakukan sebagai kata terpisah. Contoh, frasa “rumah sakit” tidak bisa diperlakukan tokenisasi seperti kata “rumah” dan “sakit”. Penerapan identifikasi frasa menurut (Sinaga, 2017) dan (Hirzani dkk., 2015) dapat mempengaruhi performa yang dihasilkan.

Tabel 3.1. Contoh Dataset

username	tweet	label	casefolding	punctuation	stopword	stemming	Slang word	frasa
hossbellkim94	@iyazheyenk aku sepertinya B+ ka, pernah covid bulan Januari, tapi sampai saat ini aku belum vaksin sama sekali.	netral	@iyazheyenk aku sepertinya b+ ka, pernah covid bulan januari, tapi sampai saat ini aku belum vaksin sama sekali.	aku sepertinya b ka pernah covid bulan januari tapi sampai saat ini aku belum vaksin sama sekali	aku sepertinya b ka pernah covid bulan januari sampai ini aku vaksin sama sekali	aku seperti b ka pernah covid bulan januari sampai ini aku vaksin sama sekali	aku seperti b kakak pernah covid bulan januari sampai ini aku vaksin sama sekali	aku seperti b kakak pernah covid bulan januari sampai ini aku vaksin sama sekali
DwiHary12524735	@jokowi Apabila sudah di vaksin apa kita sudah tidak bisa tertular virus covid-19	netral	@jokowi apabila sudah di vaksin apa kita sudah tidak bisa tertular virus covid-19	apabila sudah di vaksin apa kita sudah tidak bisa tertular virus covid	apabila di vaksin apa sudah bisa tertular virus covid	apabila di vaksin apa sudah bisa tular virus covid	apabila di vaksin apa sudah bisa tular virus covid	apabila di vaksin apa sudah bisa tular virus covid
UNajib4	@fadlizon Bang komentarnya tentang vaksin covid gimana .	netral	@fadlizon bang komentarnya tentang vaksin covid gimana .	bang komentarnya tentang vaksin covid gimana	bang komentarnya vaksin covid gimana	bang komentar vaksin covid gimana	bang komentar vaksin covid gimana	bang komentar vaksin covid gimana
_wyp	Kado terbaik hari ini adalah. . . Bapakku sudah dpt vaksin covid dadakan td pagi. Alhamdulillah* ☺	positif	kado terbaik hari ini adalah. . . bapakku sudah dpt vaksin covid dadakan td pagi. alhamdulillah* ☺	kado terbaik hari ini adalah bapakku sudah dpt vaksin covid dadakan td pagi alhamdulillah* ☺	kado terbaik hari adalah bapakku dpt vaksin covid dadakan td pagi alhamdulillah* ☺	kado terbaik hari adalah bapak dpt vaksin covid daifa td pagi alhamdulillah	kado terbaik hari adalah bapak dapat vaksin covid dada tadi pagi alhamdulillah	kado terbaik hari adalah bapak dapat vaksin covid dada tadi pagi alhamdulillah
amira_susud	@SilhamanRi Alhamdulillah yaa setelah vaksin beredar kita tak prnh dengar berita lagi dokter dan tenaga medis meninggal krn covid δΥ™• δΥ™‡	positif	@silhamanri alhamdulillah yaa setelah vaksin beredar kita tak prnh dengar berita lagi dokter dan tenaga medis meninggal krn covid δΥ™• δΥ™‡	alhamdulillah yaa setelah vaksin beredar kita tak prnh dengar berita lagi dokter dan tenaga medis meninggal krn covid δΥ™• δΥ™‡	alhamdulillah yaa vaksin beredar tak prnh dengar berita lagi dokter tenaga medis meninggal krn covid δΥ™• δΥ™‡	alhamdulillah yaa vaksin edar tak prnh dengar berita dokter tenaga medis tinggal krn covid	alhamdulillah ya vaksin edar tak pernah dengar berita dokter tenaga medis tinggal karena covid	alhamdulillah ya vaksin edar tak pernah dengar berita dokter tenaga medis tinggal karena covid
AbyadHasyim	Akhirnya Lansia bisa segera di vaksin covid 19 nih gaes. Alhamdulillah @VaksinUntukLansia https://t.co/YgezbtmI0	positif	akhirnya lansia bisa segera di vaksin covid 19 nih gaes. alhamdulillah vaksinuntuklansia https://t.co/ygezbtmi0	akhirnya lansia bisa segera di vaksin covid 19 nih gaes alhamdulillah vaksinuntuklansia	akhirnya lansia segera vaksin covid 19 nih gaes alhamdulillah vaksinuntuklansia	akhir lansia segera vaksin covid 19 nih gaes alhamdulillah vaksinuntuklansia	akhir lansia segera vaksin covid nih gaes alhamdulillah vaksinuntuklansia	akhir lansia segera vaksin covid nih gaes alhamdulillah vaksinuntuklansia

Tabel 3. 1. Contoh Dataset (lanjutan)

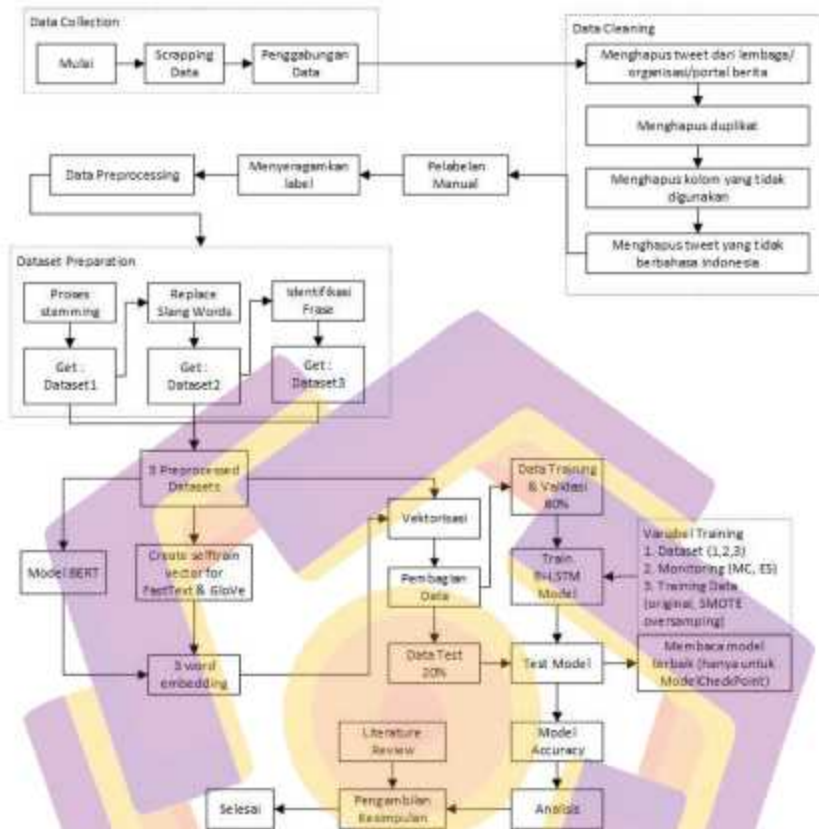
radhyaksa_	disuntik vaksin covid-19 bukannya sembuh malah jadi titan	negatif	disuntik vaksin covid-19 bukannya sembuh malah jadi titan	disuntik vaksin covid19 bukannya sembuh malah jadi titan	disuntik vaksin covid19 bukannya sembuh malah jadi titan	suntik vaksin covid19 bukan sembuh malah jadi titan	suntik vaksin covid bukan sembuh malah jadi titan	suntik vaksin covid bukan sembuh malah jadi titan
LAODE06368283	@msaid_didu saya rasa vaksin yg berasal dari cina itu sengaja di buat untuk menghancurkan rakyat indonesia,melalui pemularan covid 19,makanya para pemerintah tidak ada yg mau jadi relawan untuk uji coba vaksin tersebut krna mereka da tau itu akan memamatkan.	negatif	@msaid_didu saya rasa vaksin yg berasal dari cina itu sengaja di buat untuk menghancurkan rakyat indonesia,melalui pemularan covid 19,makanya para pemerintah tidak ada yg mau jadi relawan untuk uji coba vaksin tersebut krna mereka da tau itu akan memamatkan.	saya rasa vaksin yg berasal dari cina itu sengaja di buat untuk menghancurkan rakyat indonesia melalui pemularan covid makanya para pemerintah tidak ada yg mau jadi relawan untuk uji coba vaksin tersebut krna mereka da tau itu akan memamatkan	rasa vaksin yg berasal cina sengaja buat hancur rakyat indonesia melalui pemularan covid makanya pemerintah ada yg mau jadi rawan uji coba vaksin tersebut krna da tau akan memamatkan	rasa vaksin yg asal cina sengaja buat hancur rakyat indonesia melalui tular covid 19 makanya perintah ada yg mau jadi rawan uji coba vaksin tersebut krna da tau akan mati	rasa vaksin yang asal cina sengaja buat hancur rakyat indonesia melalui covid makanya perintah ada yang mau jadi rawan uji coba vaksin tersebut karena da tahu akan mati	rasa vaksin yang asal cina sengaja buat hancur rakyat indonesia melalui tular covid makanya perintah ada yang mau jadi rawan uji coba vaksin tersebut karena da tahu akan mati
kjhansj	@bonz_ade sampaikan ini sobat: besok kalo u atau keluarga ukenu jangan ke rumah sakit, jangan test, jangan minta tolong siapapun termasuk bantuan pemerintah, jangan ikutan vaksin, tidur aja dirumah, tar juga sembuh sendiri, tergantung sembuh sehat apa qoid	negatif	@bonz_ade sampaikan ini sobat: besok kalo u atau keluarga ukenu covid jangan ke rumah sakit, jangan test, jangan minta tolong siapapun termasuk bantuan pemerintah, jangan ikutan vaksin, tidur aja dirumah, tar juga sembuh sendiri, tergantung sembuh sehat apa qoid	sampaikan ini sobat besok kalo u atau keluarga ukenu covid jangan ke rumah sakit jangan test jangan minta tolong siapapun termasuk bantuan pemerintah jangan ikutan vaksin tidur aja dirumah tar juga sembuh sendiri tergantung sembuh sehat apa qoid	sampaikan sobat besok kalo u keluarga ukenu covid jangan rumah sakit jangan test jangan minta tolong siapapun termasuk bantuan pemerintah jangan ikutan vaksin tidur aja dirumah tar sembuh sendiri tergantung sembuh sehat apa qoid	sampai sobat besok kalo u keluarga ukenu covid jangan rumah sakit jangan test jangan minta siapa masuk bantu perintah jangan ikut vaksin tidur aja rumah tar sembuh sendiri tergantung sembuh sehat apa qoid	sampai sobat besok kalau kamu keluarga ukenu covid jangan rumah sakit jangan test jangan minta siapa masuk bantu perintah jangan ikut vaksin tidur aja rumah nanti sembuh sendiri tergantung sembuh sehat apa qoid	sampai sobat besok kalau kamu keluarga ukenu covid jangan rumah sakit jangan test jangan minta siapa masuk bantu perintah jangan ikut vaksin tidur aja rumah nanti sembuh sendiri tergantung sembuh sehat apa qoid



Gambar 3.3. Alur *preprocessing* Data

3.5. Alur Penelitian

Alur penelitian diilustrasikan pada gambar 3.3. dimulai dari pengumpulan data, *filtering data*, *preprocessing data*, pembuatan model, pelatihan dan pengujian model hingga mendapatkan akurasi.



Gambar 3.4. Alur penelitian

Alur penelitian dimulai dengan melakukan pengumpulan data dari sosial media Twitter menggunakan *tools* SNScrape (JustAnotherArchivist, 2021) menggunakan kata kunci "Vaksin Covid". Dari hasil pengumpulan data tersebut didapatkan data mentah sebanyak 262.306 baris data yang selanjutnya akan dilakukan *cleaning* dan *filtering* sehingga jumlah data yang tersisa berjumlah 64.332 data.

Sisa data yang tersisa tersebut dilakukan proses pelabelan oleh pakar dari bidang psikologi (Psikologi Industri). Contoh hasil pelabelan yang dilakukan oleh ahli ditunjukkan pada Tabel 3.2. Selanjutnya, dataset dilakukan *preprocessing*, dibagi menjadi 3 jenis dataset yaitu, pertama dataset yang di-*preprocessing* sampai tahapan *stemming*, karena berdasarkan penelitian yang dilakukan oleh (Khairunnisa dkk., 2021) penggunaan *stemming* dapat meningkatkan performa klasifikasi, proses *stemming* digunakan menggunakan *library* Sastrawi khusus untuk Bahasa Indonesia (Nazief, B. A. A. & Adriani, 2005)

Tabel 3. 2. Contoh Hasil Pelabelan

tweet	label
@nyazbeyenk aku sepeertinya B+ ka, pernah covid bulan Januari, tapi sampai saat ini aku belum vaksin sama sekali.	netral
@jokowi Apabila sudah di vaksin apa kita sudah tidak bisa tertular virus covid-19	netral
@fadlizon Bang komentarnya tentang vaksin covid gimana..	netral
Kado terhaiq hari ini adalah . . . Bapakku sudah dpt vaksin covid dadakan id pagi. Alhamdulillah! ☺	positif
@SilumanRi Alhamdulillah yaa setelah vaksin beredar kita tak pernah dengar berita lagi dokter dan tenaga medis meninggal krn covid. 🙏🙏🙏🙏	positif
Akhirnya Lansia bisa segera di vaksin covid 19 nih gaes. Alhamdulillah #VaksinUntukLansia https://t.co/YgezbTmdz0	positif
disuntik vaksin covid-19 bukannya sembuh malah jadi titan	negatif
@msasil dihi saya rasa vaksin yg berasal dari china itu sengaja di buat untuk menghancurkan rakyat indonesia, melalui penularan covid 19, makanya para pemerintah tidak aila yg mau jadi relawan untuk uji coba vaksin tersebut kma mereka da tau itu akan mematikan.	negatif
@bonz_ade sampaikan ini sobat: besok kalo u atau keluarga terkena covid jangan ke rumah sakit, jangan test, jangan minta tolong siapapun termasuk bantuan pemerintah, jangan ikutin vaksin, tidur aja dirumah, tar juga sembuh sendiri, tergantung sembuh sehat apa qoid	negatif

Kedua, dataset yang di-*preprocessing* sampai tahapan *slangword normalization* penggunaan *slang word* ini dilakukan atas saran yang diberikan oleh (Yulianto dkk., 2019) dimana perlu ditambahkan *preprocessing* berupa *slang word* untuk menangani kata-kata *slang* dari sosial media, untuk menangani *slang word* kumpulan kata-kata yang digunakan bersumber dari (Aliyah Salsabila dkk., 2019) dan ketiga, dataset yang di-*preprocessing* sampai tahap identifikasi frasa yang mana

(Putranto dkk., 2016) penggunaan identifikasi frasa ini mampu meningkatkan akurasi klasifikasi teks berbahasa Indonesia.

Dataset yang tersedia, selanjutnya dibuat model latih untuk vektorisasi menggunakan tiap *word embedding* baik FastText, GloVe dan BERT. Proses selanjutnya adalah pemisahan data latih dan data uji dengan proporsi 80:20. Pembagian dengan proporsi ini dapat merepresentasikan data, evaluasi model yang cukup cepat, dapat menggambarkan distribusi yang sama pada data latih dan data uji, dan mendapatkan akurasi terbaik (Pratama & Tjahyanto, 2022).

Proses *training* data dilakukan menggunakan dua jenis data, yaitu data original dan data yang telah diseimbangkan. Hal ini dilakukan karena dataset yang digunakan tidak seimbangkan jumlah tiap kelasnya. Beberapa penelitian sebelumnya melatih model menggunakan jumlah iterasi yang berbeda-beda. Sebagai contoh, penelitian yang dilukan oleh (Lestandy dkk., 2021) melatih model dengan jumlah iterasi 10, 20, 30, 40 dan 50. Penelitian oleh (Roy & Ojha, 2020) dan (Riza & Charibaldi, 2021) melatih model dalam 50 iterasi. Sayangnya, beberapa penelitian yang sudah disebutkan tersebut tidak memonitor proses *training* model sehingga tidak diketahui apakah jumlah iterasi yang dilakukan sudah cukup atau sebenarnya model masih dapat dilatih lebih lanjut. Oleh karena itu, pada penelitian ini model dicoba untuk dilatih hingga mencapai 100 iterasi dengan *monitoring* proses.

Dalam hal ini, ada dua teknik *monitoring* yang diuji, yaitu Early Stopping dan ModelCheckPoint. Dengan melakukan monitoring pada proses *training*, maka model akhir yang dihasilkan *training* akan terhindar dari *overfit* pada data *training*.

Di sisi lain, hasil penelitian (Listyarini & Anggoro, 2021) mengungkapkan bahwa model yang dilatih hingga 100 iterasi mempunyai performa yang lebih tinggi dibandingkan dengan model yang dilatih dalam 50 dan 75 iterasi.

Performa model LSTM yang dilatih dalam penelitian yang dilakukan oleh (Nurrohmat & SN, 2019) dalam 100 iterasi menghasilkan performa yang lebih baik jika dibandingkan dengan metode lain yang diuji dalam penelitian tersebut. Hal yang sama juga terjadi pada hasil penelitian (Fatur Rahman dkk., 2020) pada saat dilakukan proses pengujian epoch. Model yang dilatih dalam 100 iterasi menghasilkan performa yang lebih tinggi dibandingkan dengan model yang dilatih dalam 50, 150 dan 200 iterasi. Terakhir, setelah *training* dan *testing* data selesai dilakukan maka akan didapatkan hasil akurasi yang selanjutnya akan dianalisis dan ditarik kesimpulan.



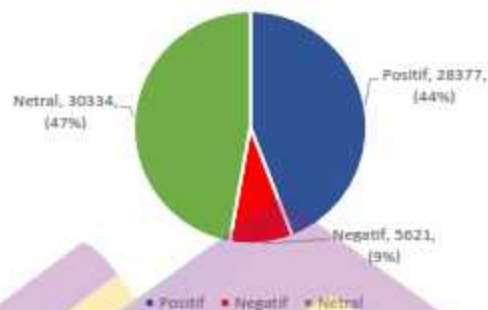
BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

d. Hasil Sentimen dan Ragam Sentimen

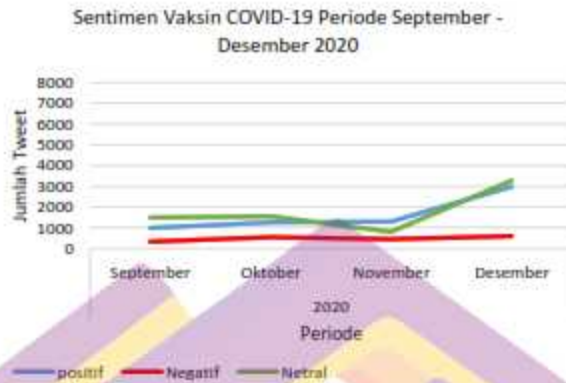
Berdasarkan hasil pengumpulan data yang telah dilakukan sejak bulan September 2020 hingga Juni 2021 pada sosial media Twitter menggunakan kata kunci “vaksin covid” ditemukan data mentah sebanyak 262.306 data. Setelah dilakukan berbagai tahapan *cleaning* dan *filtering* data yang tersisa menjadi 64.332 data yang kemudian diberikan label positif, negatif dan netral. Pemberian label positif, negatif dan netral dilakukan secara manual bekerjasama dengan pakar ahli dari bidang psikolog. Hasilnya ditemukan *tweet* netral sebanyak 30.334 data atau sekitar 47%, data positif sebanyak 28.377 setara dengan 44% dan data negatif sebesar 5621 data atau sebanyak 9%.

Persentase sentimen tersebut divisualisasikan melalui diagram yang ditunjukkan pada gambar 4.1. Positif sentimen direpresentasikan menggunakan warna biru, netral sentimen pada warna hijau dan negatif sentimen direpresentasikan menggunakan warna merah. Dari gambar 4.1. tersebut dapat disimpulkan bahwa penolakan vaksin di Indonesia oleh masyarakat pengguna Twitter masih tergolong rendah. Namun, diskusi publik tentang vaksin COVID-19 cukup tinggi dimana data netral mencapai hampir separuh dari data yang ada. Selain itu, jumlah orang yang memiliki sentimen positif tentang vaksin lebih dari lima kali lipat jumlah orang dengan sentimen negatif



Gambar 4.1. Jumlah Tweet Berdasarkan Sentimen

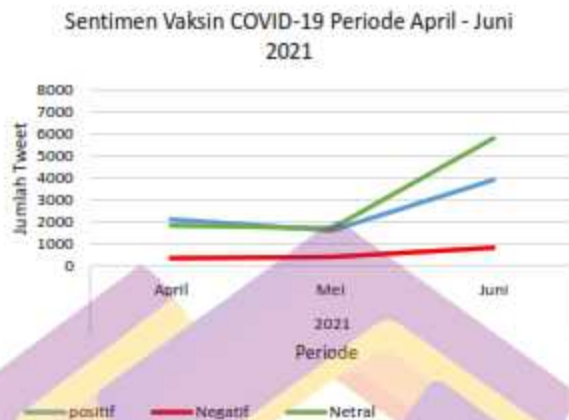
Selama rentang periode September 2020 – Juni 2021 diperoleh hasil sentimen yang cukup beragam. Gambar 4.2 menunjukkan ragam sentimen masyarakat dari sosial media Twitter selama akhir tahun 2020. Terlihat dari gambar 4.2 belum terlalu banyak *cuitan* mengenai vaksin COVID-19. Hal cukup berbeda ditunjukkan pada Gambar 4.3 dimana mulai terjadi peningkatan *cuitan* mengenai vaksin COVID-19 pada rentang waktu Januari 2021 - Maret 2021. Pada rentang waktu April 2021 – Juni 2021 secara umum terjadi penurunan *cuitan* mengenai vaksin COVID-19 namun mulai terjadi peningkatan lagi pada bulan Juni 2021.



Gambar 4. 2 Sentimen Vaksin COVID-19 Periode September – Desember 2020



Gambar 4. 3 Sentimen Vaksin COVID-19 Periode Januari – Maret 2021



Gambar 4. 4 Sentimen Vaksin COVID-19 Periode April – Juni 2021

4.2. Mekanisme *Training*

Secara umum proses eksperimen pada penelitian ini dimulai dengan menyiapkan data. Data yang telah siap untuk diolah kemudian dibuat model atau divektorisasikan menggunakan 3 (tiga) jenis *word embedding* yaitu, FastText, GloVe dan BERT. Setiap model vektorisasi yang dihasilkan oleh tiap-tiap *word embedding* kemudian akan diklasifikasi menggunakan metode Bi-LSTM karena menurut (Pasaribu dkk., 2020) (Abdalla & Özyurt, 2021) dan (Riza & Charibaldi, 2021) metode ini menghasilkan kinerja yang cukup bagus saat melakukan klasifikasi.

Dalam penelitian ini, *framework* yang digunakan adalah Keras, Tensor Flow, Scikit Learn dan NLTK. Google Collaboratory digunakan sebagai *environment* untuk melakukan eksperimen. *File notebook* dibuat untuk setiap skenario sehingga proses *training* dapat dijalankan secara paralel di beberapa

komputer. Parameter yang digunakan untuk melakukan *training* berupa *epoch*, *batch size*, *learning rate* dan *dropout*. Jumlah *epoch* yang digunakan saat menggunakan jenis *monitoring ModelCheckpoint* adalah 100 *epoch*.

Nilai *batch size* yang digunakan adalah 64 seperti penelitian yang dilakukan oleh (An dkk., 2018). Selain itu, menurut (Masters & Luschi, 2018) secara umum nilai *batch size* yang efektif dalam melakukan *training* yang stabil yaitu pada kisaran 4 hingga 64. Nilai *learning rate* yang digunakan sebesar 0,001, besaran nilai ini merupakan nilai standar dari Keras dan dianggap paling ideal untuk menangani berbagai macam kasus secara umum (Moolayil, 2019). Fungsi aktivasi pada penelitian ini menggunakan fungsi aktivasi *default* yang dimiliki oleh Bi-LSTM, yaitu fungsi aktivasi *tanh* sedangkan untuk nilai *dropout* sebesar 0,5 digunakan untuk mencegah terjadinya *overfitting*.

Dataset yang digunakan dalam penelitian ini merupakan dataset yang tidak seimbang (*imbalance dataset*). Dimana dari 64.332 baris data, terdapat 30.334 data dengan sentimen netral, 28.377 data dengan sentimen positif dan 5.621 data dengan sentimen negatif. Dataset yang tidak seimbang dapat ditangani dengan beberapa cara, seperti *resampling data training*. *Resampling* data bisa dilakukan menggunakan metode *undersampling* dan *oversampling*.

Metode *undersampling* dilakukan jika ingin mengurangi jumlah kelas data sesuai dengan kelas data yang paling sedikit, metode ini baik dilakukan jika selisih antar kelas data tidak terlalu jauh. Sedangkan metode *oversampling* adalah menyamakan jumlah data tiap kelas sesuai dengan jumlah data yang paling banyak pada sebuah kelas dan metode ini efektif digunakan jika selisih jumlah data antar kelas terlalu jauh. Sehingga pada penelitian ini metode *oversampling* lebih cocok untuk digunakan. Teknik yang digunakan untuk

melakukan *oversampling* adalah dengan menggunakan Synthetic Minority Over-sampling Technique (SMOTE) hal ini sejalan dengan saran penelitian oleh (Nurdeni dkk., 2021) dimana data yang tidak seimbang sebaiknya ditangani menggunakan teknik SMOTE tersebut.

Tabel 4.1. Variasi Skenario Training Data

Embedding	Training Monitoring	Training Data	Tahapan <i>preprocessing</i> Data		
			Stemming	Slangword	Frasa
FastText	Model Checkpoint	Original	√	-	-
			√	√	√
		SMOTE	√	-	-
			√	√	√
	Early Stopping	Original	√	-	-
			√	√	√
		SMOTE	√	-	-
			√	√	√
GloVe	Model Checkpoint	Original	√	-	-
			√	√	√
		SMOTE	√	-	-
			√	√	√
	Early Stopping	Original	√	-	-
			√	√	√
		SMOTE	√	-	-
			√	√	√
BERT	Early Stopping	Original	√	-	-
			√	√	√
		SMOTE	√	-	-
			√	√	√
	Early Stopping	Original	√	-	-
			√	√	√
		SMOTE	√	-	-
			√	√	√

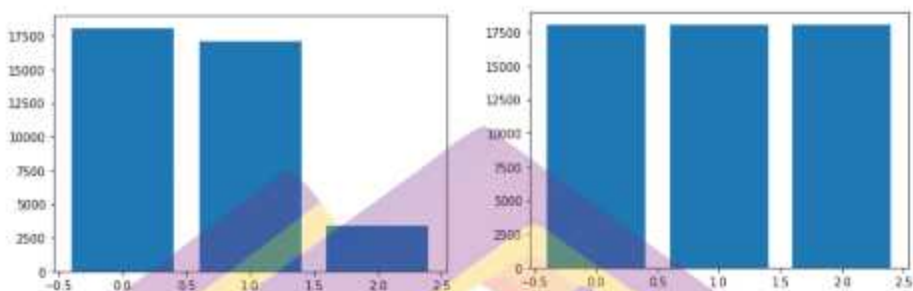
Proses *training* dari penelitian ini dilakukan dengan beberapa skenario dengan beberapa variasi variabel yang berbeda seperti jenis *word embedding* yang digunakan, teknik *monitoring* dari tiap *training*, jenis data *training* yang digunakan serta jenis *preprocessing* data yang digunakan pada dataset. Secara singkat, tabel 4.2. menunjukkan variasi skenario yang digunakan dalam penelitian ini.

4.3. Penanganan *Imbalance Data*

Pada bagian mekanisme *training* sebelumnya sudah disebutkan bahwa dataset final yang akan diproses terdiri dari 64.332 data dengan komposisi 30.334 data dengan label netral, data berlabel positif sebesar 28.377 dan sisanya sebanyak 5621 berlabel negatif. Ketidakseimbangan data menurut (Garcia dkk., 2012) akan menyebabkan penurunan performa hasil klasifikasi serta akan menghasilkan hasil yang baik pada kelas mayor dan mengabaikan kelas minor (L. Chen dkk., 2018). Oleh karena itu, perlu dilakukan penanganan *imbalance* data agar performa klasifikasi menjadi lebih maksimal.

Penanganan *imbalance* data dilakukan menggunakan teknik SMOTE seperti yang disarankan oleh (Nurdeni dkk., 2021). Selain itu (Dablain dkk., 2022) mengungkapkan bahwa penggunaan SMOTE untuk mengatasi *imbalance* data dianggap mudah dan efektif.

Pada gambar 4.5. menunjukkan hasil penanganan *imbalance* data sehingga pada akhirnya data menjadi seimbang dan siap diolah untuk beberapa skenario tertentu. Data yang diseimbangkan merupakan data latih sebesar 80% dari semua data.



Gambar 4.5. Data Sebelum dan Setelah *Oversampling*

4.4. Model FastText

4.4.1. Pembuatan Model FastText

Model FastText yang digunakan dalam penelitian ini merupakan jenis FastText yang dilatih menggunakan dataset yang ada dalam hal ini disebut sebagai *self-train*. Sebenarnya di dalam dokumentasinya, FastText juga menyediakan model *pre-train* berbahasa Indonesia yang telah dilatih menggunakan *corpus* Bahasa Indonesia. Namun, penggunaan model *self-train* dirasa lebih relevan karena dataset yang digunakan diambil dari sosial media sehingga Bahasa Indonesia yang digunakan merupakan Bahasa sehari-hari yang tidak terlalu baku, hal ini cukup berbeda dengan model *pre-train* yang disediakan dengan Bahasa Indonesia baku karena dilatih menggunakan *corpus* Wikipedia.

Dalam proses pembuatan model, *library* yang digunakan adalah Gensim yang telah disediakan oleh Python dan FastText dapat langsung diimpor melalui Gensim. Gambar 4.6. merupakan proses inti pembuatan model FastText, dimana

pada gambar tersebut tertera *embedding size* yang digunakan sebesar 300, *embedding size* digunakan untuk menentukan jumlah vektorisasi dalam hal ini satu kata akan diubah menjadi 300 angka. *Window* yang digunakan sejumlah 5, artinya kata yang akan dicek adalah 5 ke kiri dan 5 ke kanan dari kata yang ada. *Minimum count* ditentukan sebesar 5 dengan maksud jika sebuah kata muncul sebanyak 5 kali, maka kata ini akan dianggap sebagai *vocab* dan dilakukan vektorisasi. Sg adalah skipgram dimana yang digunakan pada penelitian ini sesuai *default* dari FastText dan iterasi yang digunakan sejumlah 100.

```
[ ] embedding_size = 300
    window_size = 5
    min_word = 5

    fast_text_model = FastText(word_tokens,
                              size=embedding_size,
                              window=window_size,
                              min_count=min_word,
                              workers = 4,
                              sg=1,
                              iter=100)
```

Gambar 4.6. *Pseudocode* Pembuatan Model FastText

Contoh dari kata yang telah dilakukan vektorisasi ditunjukkan pada Gambar 4.7. dimana yang ditampilkan hanya potongan dari hasil vektorisasi dari kata "covid". Vektorisasi bersifat unik sehingga tidak mungkin hasil vektorisasi sebuah kata sama dengan hasil vektorisasi kata yang lainnya.

```
w2v["covid"]
array([ 0.10805999, -0.06850228, -0.13077527, -0.24220584,  0.06460719,
        0.06785943,  0.2732286 , -0.12258494,  0.0550986 ,  0.04415696,
       -0.01230397, -0.03490597,  0.11085398, -0.02920009,  0.04270586,
        0.13875069,  0.18144996,  0.01280014,  0.21851833, -0.05364508,
        0.08444549,  0.07842056,  0.09519295,  0.18969256, -0.10723357,
       -0.21983136,  0.04263502,  0.2612632 ,  0.08816692, -0.37300745,
       -0.04878063,  0.18641037,  0.1103454 ,  0.03737861, -0.13315411,
       -0.18489632,  0.2213517 ,  0.2238398 , -0.1650903 ,  0.17796144,
       -0.04503194,  0.04684994, -0.07020257,  0.1416657 , -0.04864604,
       -0.23798104,  0.11920702, -0.11010794,  0.23099364,  0.01691593,
        0.20206687,  0.17697166,  0.29554132, -0.12500507,  0.04565444,
        0.23836517,  0.12948188, -0.15620424,  0.06872442,  0.00161854,
       -0.03445695,  0.24936294,  0.07770012, -0.07656769,  0.14141026,
       -0.08667178,  0.16146204, -0.01971115,  0.120717 , -0.14118402,
        0.153269 , -0.1383846 ,  0.00520265,  0.10474788, -0.15467153,
       -0.16459891, -0.30467320,  0.20371641,  0.03488177,  0.14937036,
        0.04911535,  0.2253473 ,  0.05524678,  0.24567242,  0.08558986,
        0.07368436,  0.21518072, -0.04530065, -0.03290503, -0.1039127 ,
       -0.1219371 ,  0.1527695 , -0.21248153,  0.11337114, -0.3191922 ,
       -0.24645856,  0.00410964, -0.08193823, -0.03173072,  0.01806174,
        0.11562655, -0.13032244,  0.01887573,  0.15178536, -0.02321329,
        0.0357335 ,  0.22297916, -0.14546616,  0.04966676 ,  0.37801933,
        0.23949663, -0.07426744, -0.10225021, -0.19430156, -0.12677464,
       -0.04146657,  0.20711155, -0.11255908, -0.17414166, -0.246851 ,
        0.15095131, -0.3496353 , -0.24417877,  0.0271374 , -0.28142455,
        0.44446033, -0.04036549, -0.12738185,  0.10317118, -0.05003943,
       -0.1275076 , -0.17344557,  0.06823276, -0.20429562,  0.18138662,
       -0.0005722 , -0.11685363, -0.00142398, -0.01096835, -0.04267531,
        0.15676397,  0.0614155 ,  0.23603763,  0.03637194, -0.382244 ,
```

Gambar 4.7. Contoh Hasil Vektorisasi Kata "covid" pada FastText

Seluruh data yang telah selesai dilakukan vektorisasi kini menjadi sebuah model untuk selanjutnya disimpan dan akan digunakan pada proses selanjutnya.

4.4.2. Training dan Testing FastText

Pada tahap ini percobaan dimulai dengan melatih model data FastText yang telah dibuat serta menggunakan metode klasifikasi Bi-LSTM, dataset yang digunakan pada percobaan ini menggunakan dataset asli yang tidak seimbang

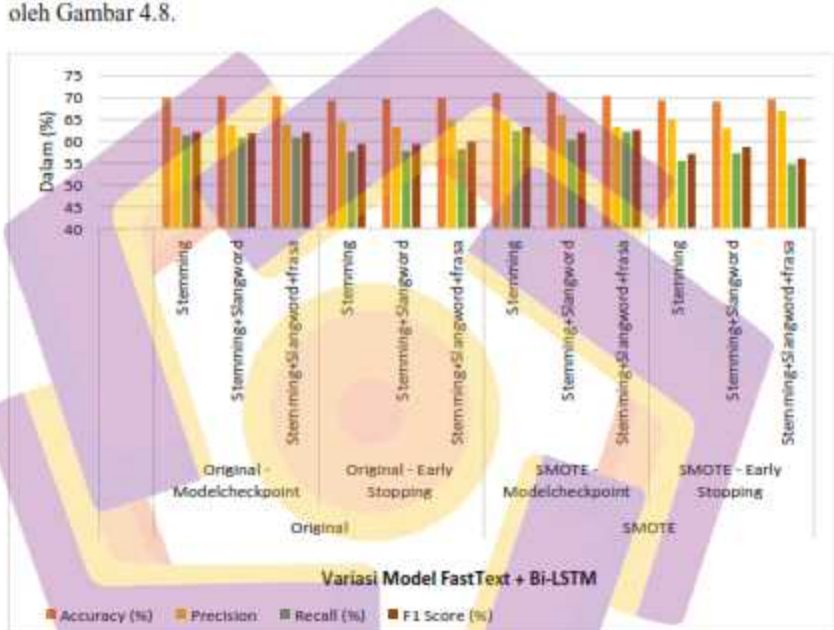
pembagian kelasnya dan dimonitoring menggunakan ModelCheckpoint dan EarlyStopping. Khusus untuk penggunaan dataset yang tidak seimbang pengukuran kinerja yang digunakan dilihat dari nilai *f1 score* yang dihasilkan. Sehingga berdasarkan Tabel 4.3. kinerja terbaik sebesar 62.10% dihasilkan oleh variasi model FastText dengan teknik *monitoring* ModelCheckpoint pada dataset yang dilakukan tahap *preprocessing stemming* saja.

Skenario uji selanjutnya adalah menggunakan data *training* yang telah dilakukan *oversampling* menggunakan SMOTE dengan variasi dataset sama seperti skenario sebelumnya dan dimonitor menggunakan ModelCheckpoint dan EarlyStopping. Berbeda dengan skenario sebelumnya saat menggunakan dataset original, performa terbaik dilihat berdasarkan nilai akurasi yang diperoleh oleh tiap pengujian. Oleh karena itu, dari semua skenario uji pada dataset yang telah dilakukan SMOTE akurasi terbaik dihasilkan oleh variasi model yang dimonitor menggunakan ModelCheckpoint pada dataset yang telah melakukan *preprocessing* sampai tahapan *slangword* dengan nilai akurasi sebesar 71.11% seperti yang ditunjukkan pada Tabel 4.3.

Tabel 4.2. Hasil Pengujian Model FastText menggunakan Bi-LSTM

Embedding	Training Data	Training Monitoring	Tahapan <i>preprocessing</i> Data			Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
			Stem ming	Slang word	Frans				
FastText	Original	Model CheckPoint	√	-	-	69.95	63.18	61.29	62.10
			√	√	-	70.26	63.61	60.82	61.86
			√	√	√	70.20	63.78	60.82	61.95
		Early Stopping	√	-	-	69.36	64.45	57.58	59.34
			√	√	-	69.64	63.24	57.72	59.29
			√	√	√	69.85	64.72	58.16	59.90
	SMOTE	Model CheckPoint	√	-	-	70.94	64.90	62.36	63.19
			√	√	-	71.11	65.90	60.23	61.99
			√	√	√	70.35	63.27	62.05	62.57
		Early Stopping	√	-	-	69.42	65.09	55.48	57.00
			√	√	-	69.01	63.02	57.26	58.73
			√	√	√	69.61	66.93	54.69	56.01

Perbandingan nilai matrik yang dihasilkan oleh semua variasi skenario pada model FastText tampak tidak terlalu signifikan selisihnya. Hal ini diketahui dari grafik hasil pengujian antar variasi menggunakan FastText yang direpresentasikan oleh Gambar 4.8.



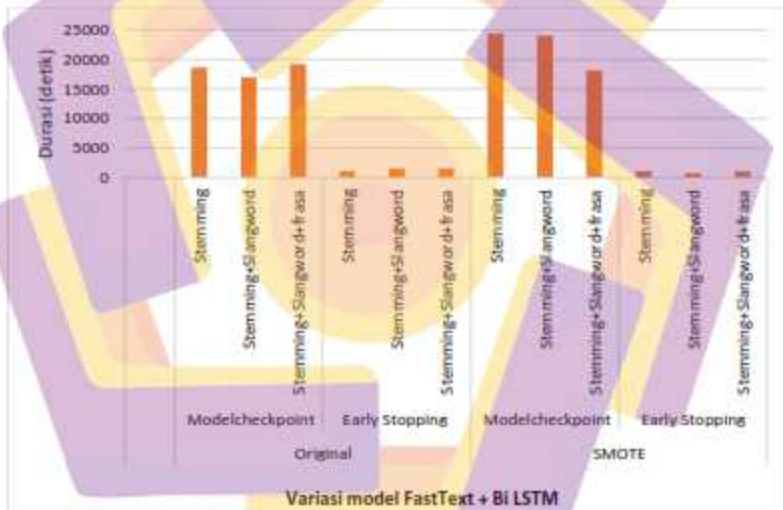
Gambar 4.8. Perbandingan Kinerja Variasi Model FastText dan Bi-LSTM

4.4.3. Durasi Proses *Training*

Proses *training* dilakukan menggunakan 80% data yang tersedia menggunakan *runtime type* standar milik Google Colaboratory yaitu CPU (*Central Processing Unit*). Durasi *training* variasi model FastText ditunjukkan pada Tabel 4.4.

Tabel 4.3. Perbandingan Durasi Training Model FastText

Embedding	Training Data	Training Monitoring	Tahapan <i>preprocessing</i> Data			Durasi (detik)	Durasi (Jam:menit:detik)
			Stem ming	Slang word	Frasa		
FastText	Original	Model CheckPoint	√	-	-	18576.84754	05:09:36
			√	√	-	16951.90948	04:42:31
			√	√	√	19181.71951	05:19:41
		Early Stopping	√	-	-	1051.814243	00:17:31
			√	-	-	1494.641387	00:24:54
			√	√	√	1489.582629	00:24:49
	SMOTE	Model CheckPoint	√	√	√	24330.25343	06:45:30
			√	√	-	23980.5848	06:39:40
			√	√	√	18095.41456	05:01:55
		Early Stopping	√	-	-	1045.785915	00:17:25
			√	-	-	715.512438	00:11:55
			√	√	√	1050.9101	00:17:30

Gambar 4.9. Grafik perbandingan durasi *training* model GloVe

Berdasarkan Gambar 4.9, durasi *training* paling lama dihasilkan ketika model menggunakan variasi data yang telah dilakukan SMOTE, monitoring menggunakan ModelCheckpoint dan dataset dengan *preprocessing* *Stemming* dengan lama waktu *training* 6 jam 45 menit dan 30 detik. Di sisi lain durasi *training* paling

singkat dihasilkan oleh variasi FastText dengan model monitoring *EarlyStopping* pada dataset sampai tahapan *slangword* dengan data yang telah di SMOTE.

Secara umum, model yang dilatih menggunakan teknik *monitoring modelcheckpoint* memiliki durasi *training* rata-rata lebih dari 5 jam karena proses *training* harus diselesaikan sampai epoch ke-100 sedangkan pada saat model dilatih menggunakan *monitoring EarlyStopping* durasi *training* rata-rata berkisar antara 11 menit sampai 24 menit. Hal ini terjadi karena pada saat menggunakan *EarlyStopping*, model akan berhenti berlatih jika metrik yang dipantau berhenti melakukan *improve*.

4.5. Model GloVe

4.5.1. Pembuatan Model GloVe

Sedikit berbeda dengan pembuatan model FastText, pembuatan model GloVe untuk *pre-trained* model kali ini menggunakan bantuan sistem operasi Linux berdasarkan saran dari (Wibisono, 2019) karena *code* yang digunakan merupakan *code* bash dimana cukup sulit untuk dijalankan pada sistem operasi Windows. Namun, pada dasarnya aturan-aturan pembuatan vektorisasi pada GloVe sama dengan FastText. Setelah proses pembuatan model selesai dilakukan, maka model yang berbentuk .txt akan diimpor ke dalam Google Colab. Contoh hasil vektorisasi kata "covid" menggunakan GloVe ditunjukkan pada Gambar 4.10.

```

model['covid']
array([-1.045790e-01,  4.493800e-01, -2.183900e-02,  7.645000e-02,
       -3.920700e-02,  3.392440e-01, -6.333540e-01, -5.899000e-03,
       -7.206200e-02, -3.018390e-01,  4.367400e-02, -2.331900e-02,
       -1.220960e-01, -3.972990e-01,  8.206550e-01,  5.043880e-01,
       1.887430e-01, -1.385440e-01,  6.150500e-01, -1.522040e-01,
       2.825900e-02, -2.849890e-01, -4.373380e-01, -5.431400e-01,
       -4.226600e-01, -6.700420e-01, -2.498760e-01, -4.640400e-02,
       1.262860e-01, -1.713000e-01,  3.365320e-01,  1.084590e-01,
       -1.588070e-01, -8.448690e-01,  4.401970e-01, -3.302500e-02,
       1.892590e-01, -3.066810e-01, -6.565550e-01, -7.123260e-01,
       3.940920e-01,  3.689910e-01, -4.350700e-02,  4.964040e-01,
       -3.583390e-01, -5.200000e-02, -8.201390e-01, -1.348600e-01,
       5.997640e-01, -1.510580e-01,  2.693000e-01, -2.095470e-01,
       9.138100e-02, -1.055300e-02,  2.760240e-01, -4.032410e-01,
       2.525800e-02,  1.014430e-01,  1.442710e-01, -2.143970e-01,
       -7.790000e-03, -2.721450e-01,  4.799400e-02,  1.217900e-03,
       1.766400e-01,  1.640000e-03,  9.586120e-01,  9.299300e-02,
       1.613630e-01, -2.934400e-02, -3.516100e-02, -1.107910e-01,
       8.382780e-01, -2.017900e-02, -3.920900e-01, -3.066850e-01,
       3.259810e-01, -2.407550e-01,  2.191610e-01,  4.136040e-01,
       4.647140e-01, -4.788270e-01, -4.446760e-01,  2.269890e-01,
       3.911750e-01,  9.546000e-03, -7.099610e-01,  5.908890e-01,
       -7.035900e-02,  1.735240e-01, -1.627050e-01,  1.338270e-01,
       5.452220e-01, -5.897300e-02, -2.391540e-01, -8.670360e-01,
       -1.059100e-02, -3.598560e-01, -1.451780e-01, -9.400550e-01,
       -4.744500e-02,  5.936870e-01, -3.454300e-02,  7.268360e-01,
       2.603630e-01, -5.627260e-01,  3.710390e-01, -2.805650e-01,
       9.112200e-02,  7.376950e-01, -3.800000e-03,  5.103380e-01,
       -5.855210e-01,  1.194090e-01,  4.894710e-01,  1.305370e-01,

```

Gambar 4.10. Contoh Vektorisasi Kata "covid" menggunakan GloVe

4.5.2. Training dan Testing GloVe

Hampir sama dengan proses *training* pada tahap sebelumnya, hanya saja kali ini model yang digunakan adalah model *word embedding* GloVe. Variasi *training* yang digunakan juga sama, yaitu menggunakan *training monitoring* ModelCheckPoint dan *earlystopping*, data latih yang masih asli dan data latih yang telah dilakukan *oversampling* serta tahapan *preprocessing stemming*, *stemming+slangword* dan *stemming+slangword+frasa*. Hasil uji yang diperoleh

dari eksperimen ini sedikit berbeda dengan proses *training* menggunakan model FastText.

Proses uji menggunakan model GloVe pada data yang belum dilakukan *oversampling* menghasilkan kinerja terbaik pada saat merapkan variasi dataset yang telah dilakukan *stemming*, *slangword* dan frasa. Hasil terbaik ini sama ketika menggunakan ModelCheckpoint maupun EarlyStopping walaupun persentase nilai *f1 score* nya berbeda yaitu sebesar 56.34% pada saat menggunakan ModelCheckpoint dan 54.02% pada saat menggunakan EarlyStopping.

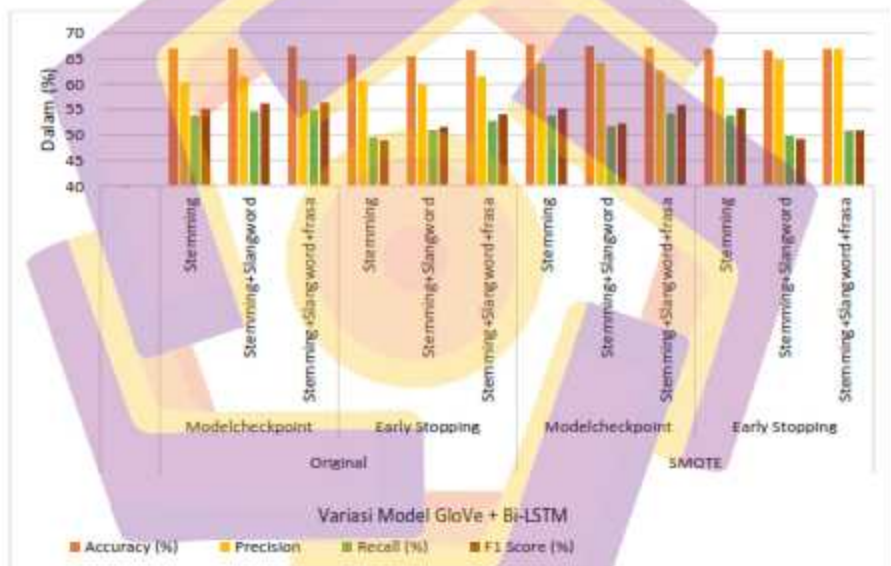
Hal berbeda dihasilkan ketika melakukan uji pada data yang telah dilakukan *oversampling* menggunakan SMOTE. Variasi *training* dengan menggunakan dataset yang telah dilakukan proses *stemming* menghasilkan kinerja terbaik saat dimonitor menggunakan ModelCheckpoint maupun EarlyStopping dengan nilai akurasi 67.74% dan 66.88%. Hasil pengujian menggunakan model GloVe menggunakan metode Bi-LSTM dapat dilihat pada Tabel 4.5.

Tabel 4.4. Hasil Pengujian Model GloVe menggunakan Bi-LSTM

Embedding	Training Data	Training Monitoring	Tahapan <i>preprocessing</i> Data			Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
			Stem mine	Slang word	Frasa				
GloVe	Original	Model CheckPoint	√	-	-	66.92	60.24	53.70	55.07
			√	√	-	67.02	61.47	54.60	56.13
			√	√	√	67.33	60.81	54.88	56.34
		Early Stopping	√	-	-	65.63	60.56	49.48	48.96
			√	√	-	65.55	60.02	50.96	51.49
			√	√	√	66.62	61.51	52.69	54.02
	SMOTE	Model CheckPoint	√	-	-	67.74	64.14	53.78	55.18
			√	√	-	67.51	64.17	51.69	52.21
			√	√	√	67.17	62.54	54.20	55.87
		Early Stopping	√	-	-	66.88	61.36	53.78	55.22
			√	√	-	66.64	64.81	49.81	49.15
			√	√	√	66.88	66.89	50.78	50.91

Berdasarkan Tabel 4.5. dari semua variasi *training* menggunakan model GloVe, model dengan *training monitoring* ModelCheckpoint, data yang telah di

SMOTE dan dataset yang telah *dipreprocessing* menggunakan *stemming* mendapatkan akurasi yang paling baik sebesar 67.74 %. Grafik hasil perbandingan tersebut dapat dilihat pada Gambar 4.11. Selain itu, penggunaan teknik *preprocessing* berupa identifikasi frasa pada hampir semua skenario tidak memiliki pengaruh yang besar karena GloVe memiliki mekanisme menggabungkan *co-occurrence* kata untuk memperoleh hubungan yang semantik antar kata (Pennington dkk., 2014).



Gambar 4.11. Perbandingan Kinerja Variasi Model GloVe dan Bi-LSTM

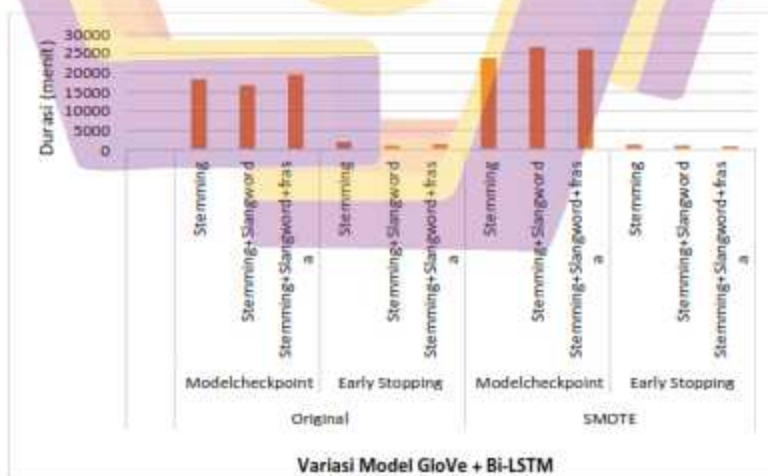
4.5.3. Durasi Proses Training

Secara garis besar, durasi *training* yang dilakukan menggunakan model GloVe tidak jauh berbeda dengan *training* pada saat menggunakan model FastText. Durasi paling lama terjadi pada saat melakukan *training* menggunakan

ModelCheckpoint pada data yang telah dilakukan *oversampling* dengan dataset yang telah dilakukan *stemming* dan *slangword* yaitu selama 7 jam 23 menit 31 detik. Sedangkan proses *training* tercepat selama 15 menit 06 detik dihasilkan ketika melakukan *training* pada *monitoring earllystopping* dengan data *oversampling* dan tahapan *preprocessing stemming, slangword* dan frasa. Tabel 4.6. dan Gambar 4.12. merepresentasikan perbandingan durasi *training* model GloVe

Tabel 4.5. Perbandingan Durasi Training Model GloVe

Embedding	Training Data	Training Monitoring	Tahapan <i>preprocessing</i> Data			Durasi (detik)	Durasi (Jam:menit:detik)
			Stem ming	Slang word	Frasa		
GloVe	Original	Model CheckPoint	√	-	-	18150.30893	05:02:30
			√	√	-	16652.75087	04:37:32
			√	√	√	19351.41871	05:22:31
		Early Stopping	√	-	-	2003.891771	00:33:23
			√	√	-	1086.427014	00:18:06
			√	√	√	1474.16872	00:24:34
	SMOTE	Model CheckPoint	√	-	-	23825.79543	06:37:05
			√	√	-	26611.29552	07:33:31
			√	√	√	26070.5011	07:14:30
		Early Stopping	√	-	-	1238.510224	00:20:39
			√	√	-	1080.365505	00:18:00
			√	√	√	906.9141724	00:15:06

Gambar 4.12. Grafik perbandingan durasi *training* model GloVe

4.6. Model BERT

4.6.1. Pembuatan Model BERT

Pada proses ini model mengadaptasi dari model Indonesian BERT milik (Cahya, 2022) model BERT tersebut telah dilakukan *pre-train* pada corpus Wikipedia Bahasa Indonesia. Penggunaan model ini mengacu pada penelitian yang dilakukan oleh (Nugroho dkk., 2021). Model BERT yang digunakan pada penelitian kali ini tidak menggunakan model BERT yang dilatih sendiri menggunakan dataset yang ada (*fine tuning*) karena dibutuhkan kinerja komputasi yang cukup tinggi untuk melakukannya dan belum bisa dipenuhi dalam penelitian ini.

4.6.2. Training dan Testing BERT

Proses *training* dan *testing* menggunakan model BERT *embedding* cukup berbeda dengan proses *training* dan *testing* pada model-model sebelumnya. Perbedaan utama terletak pada karakteristik model vektorisasi BERT itu sendiri dimana BERT tidak menghasilkan nilai vektor yang sama pada setiap kata, karena BERT memperhatikan kata sebelum dan setelah kata tersebut, sehingga vektor yang dihasilkan adalah vektor dari sebuah kalimat.

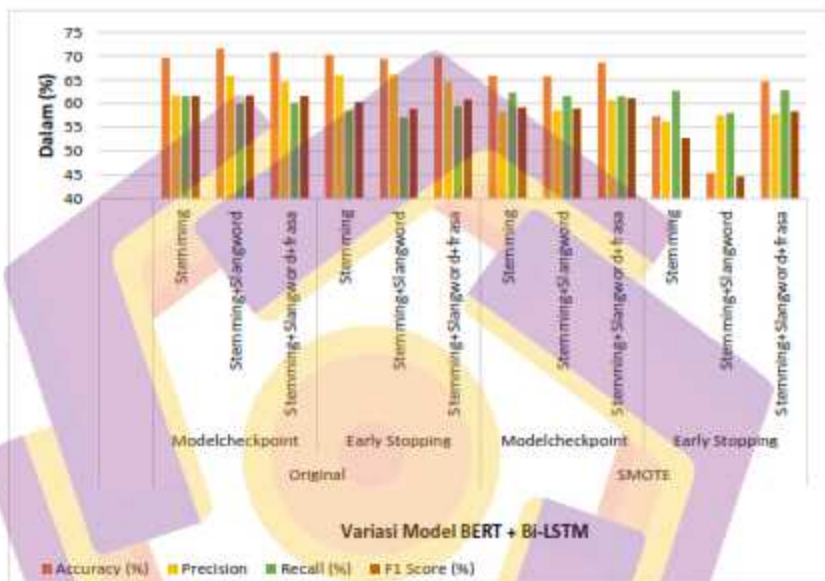
Tabel 4.6. Hasil uji pada pada model BERT

Embedding	Training Data	Training Monitoring	Tahapan <i>preprocessing</i> Data			Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
			Stem ming	Slang word	Frasa				
BERT	Original	Model CheckPoint	√	-	-	69.60	61.68	61.55	61.62
			√	√	-	71.55	65.84	59.97	61.71
			√	√	√	70.82	64.67	60.06	61.60
		Early Stopping	√	-	-	70.18	65.87	58.51	60.32
			√	√	-	69.33	66.09	57.08	58.86
			√	√	√	69.76	64.46	59.31	60.81
	SMOTE	Model CheckPoint	√	-	-	65.84	58.28	62.30	59.20
			√	√	-	65.70	58.51	61.58	58.88
			√	√	√	68.59	60.71	61.51	61.08
		Early Stopping	√	-	-	57.29	36.23	62.58	52.80
			√	√	-	45.36	57.51	57.99	44.57
			√	√	√	64.64	57.93	62.75	58.30

Pada awal melakukan proses *training* dan *testing*, *runtime type* yang digunakan sama dengan *training* dan *testing* pada model sebelumnya, namun ternyata proses ini tidak bisa dijalankan. Solusi pertama yang dilakukan untuk menjalankan proses ini adalah dengan menggunakan *runtime type* GPU (*Graphic Processing Unit*), sayangnya solusi ini pun masih belum membuahkan hasil. Solusi kedua yang dilakukan adalah dengan mengganti *runtime type* menjadi TPU (*Tensor Processing Unit*) hingga pada akhirnya proses *training* dan *testing* dapat berjalan dengan semestinya. Untuk dapat meningkatkan lagi performa komputasi untuk menjalankan proses *training* dan *testing* akhirnya diputuskan untuk melakukan *upgrade* Google Colab menjadi Google Colab Pro. Hasil uji menggunakan model BERT dapat dilihat pada Tabel 4.7.

Berdasarkan hasil yang ditunjukkan pada Tabel 4.7. akurasi paling baik dihasilkan pada saat melakukan *testing* pada data *train* ModelCheckPoint dengan data original yang telah dilakukan *preprocessing* sampai tahap *slangword* dengan nilai *f1 score* sebesar 61.71%. Sedangkan pada saat menggunakan data yang telah

dilakukan *oversampling*, kinerja terbaik dihasilkan oleh dataset yang telah *dipreprocessing* sampai pada tahap identifikasi frasa dengan nilai akurasi sebesar 68.59 %. Perbandingan tiap variasi uji dapat dilihat pada Gambar 4.13.



Gambar 4.13. Grafik Perbandingan Variasi Hasil Uji Model BERT

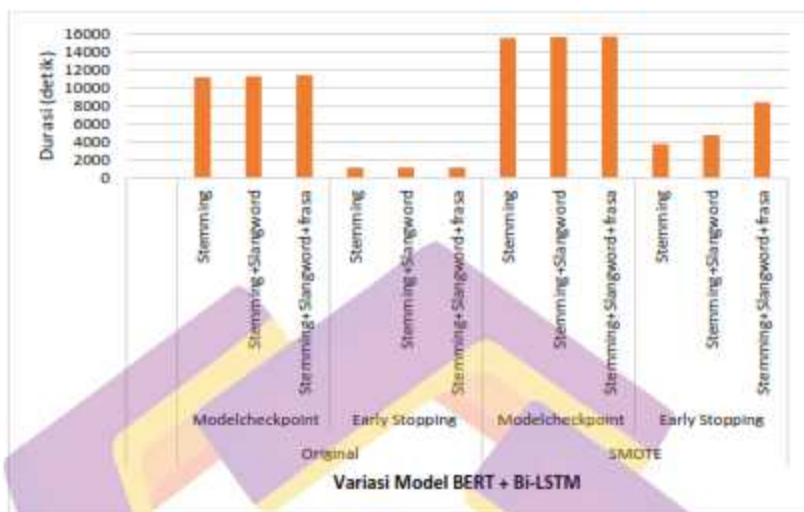
4.6.3. Durasi Proses Training

Pada umumnya durasi proses *training* yang dilakukan pada model BERT lebih cepat dibandingkan dengan durasi proses *training* pada model-model sebelumnya. Namun, tentu saja hal ini tidak dapat dibandingkan secara '*apple-to-apple*' karena jenis komputasi yang digunakan tidak sama. Secara detail, durasi *training* pada model BERT ditunjukkan pada Tabel 4.8.

Tabel 4.7. Perbandingan Durasi Training Model BERT

Embedding	Training Data	Training Monitoring	Tahapan <i>preprocessing</i> Data			Durasi (detik)	Durasi (Jam:menit:detik)
			Stem ming	Slang word	Frasa		
BERT	Original	Model CheckPoint	√	-	-	11162.41178	03:06:02
			√	√	-	11258.04991	03:07:38
			√	√	√	11386.20807	03:09:46
		Early Stopping	√	-	-	1135.622765	00:19:15
			√	√	-	1152.800264	00:19:12
			√	√	√	1168.475012	00:19:28
	SMOTE	Model CheckPoint	√	-	-	15506.51235	04:18:26
			√	√	-	15637.81621	04:20:37
			√	√	√	15682.12338	04:21:22
		Early Stopping	√	-	-	3764.519932	01:02:44
			√	√	-	4804.462025	01:20:04
			√	√	√	8384.899319	02:19:44

Dari Tabel 4.8, dapat disimpulkan bahwa durasi *training* pada ModelCheckPoint memakan waktu kurang lebih 3 hingga 4 jam dengan durasi terlama selama 4 jam 21 menit 22 detik pada saat melakukan *training* data SMOTE pada dataset yang telah dilakukan proses identifikasi frasa. Hasil yang cukup menarik perhatian ditemukan pada saat menggunakan fungsi *monitoring earlystopping* dimana durasi pada *training* data yang telah dilakukan *oversampling* cukup lama sekitar kurang lebih 1 jam. Setelah ditelusuri lebih lanjut ternyata *training* menggunakan model BERT selalu mengalami peningkatan kinerja sehingga proses *training* terus dilanjutkan. Grafik perbandingan durasi waktu *training* ditunjukkan pada Gambar 4.14.



Gambar 4.14. Grafik Durasi Proses Training Model BERT

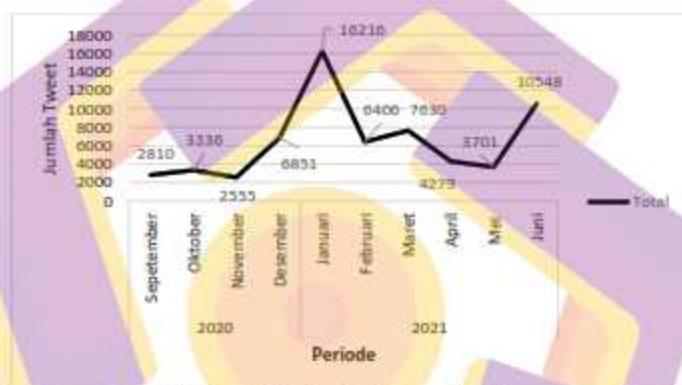
4.7 Pembahasan Hasil

4.7.1. Analisis Sentimen

Seperti yang sudah diuraikan pada bagian hasil bahwa sentiment masyarakat yang bersifat negatif cukup rendah jika dibandingkan dengan sentimen positif dan netral. Untuk lebih memahami data, data kemudian dikelompokkan berdasarkan bulan. Data ini direpresentasikan dalam 4.2. Dari gambar tersebut dapat diketahui bahwa *tweet* tentang “vaksin COVID-19” oleh masyarakat Indonesia di Twitter berada di bawah 3000 *tweet*. Intensitas diskusi ini mulai meningkat perlahan pada Desember 2020 yang mencapai 6851 *tweet* saat terjadi pergantian Menteri Kesehatan Republik Indonesia (Redaksi, 2020a) dan terjadi peningkatan jumlah kasus aktif di Indonesia (Worldometer, 2021). Pada di sisi lain, di bulan yang sama,

Joe Biden memenangkan pemilihan Presiden Amerika Serikat dan menerima yang suntik pertama vaksin COVID-19 (Redaksi, 2020b).

Jumlah *tweet* tentang vaksin mencapai titik tertinggi pada Januari 2021 ketika Presiden Republik Indonesia, Bapak Joko Widodo menjadi orang pertama di Indonesia yang disuntik vaksin COVID-19 (BPMI Setpres, 2021). Jumlah *tweet* mencapai 16.216 *tweet*.



Gambar 4.15. Jumlah *Tweet* Perbulan

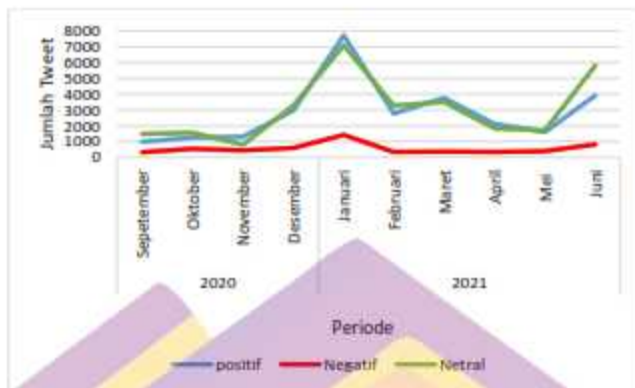
Pada Februari 2021, diskusi publik tentang vaksin COVID-19 di Twitter menurun drastis dibandingkan dengan bulan sebelumnya namun masih cukup tinggi jika dibandingkan dengan tiga bulan pertama saat pembahasan vaksin dimulai di Twitter. Di sisi lain, tiga bulan pertama tahun 2021 akan menjadi implementasi vaksinasi fase I khusus untuk tenaga kesehatan. Diskusi ini mulai meningkat lagi pada bulan Maret hingga mencapai 7630 *tweet* dimana masyarakat umum usia 18-59 tahun mulai terdaftar untuk mendapatkan vaksin dosis pertama. Diskusi melonjak lagi pada bulan Juni 2021 dimana terjadi peningkatan kasus aktif yang signifikan (Worldometer, 2021) dan penyebaran varian Delta (Sofa, 2021).

Tabel 4.8. Jumlah Sentimen Perbulan

Bulan	Positif	Negatif	Netral
September 2020	991	330	1489
October 2020	1251	538	1547
November 2020	1307	443	805
December 2020	2980	597	3274
January 2021	7708	1423	7085
February 2021	2765	356	3285
March 2021	3746	382	3502
April 2021	2108	339	1832
May 2021	1603	395	1703
June 2021	3918	818	5812

Tabel 4.8 dan Gambar 4.16. menjelaskan data sentimen publik bulanan dari September 2020 hingga Juni 2021. Berdasarkan pada Tabel 4.8. dan Gambar 4.16. dapat diamati bahwa jumlah *tweet* dengan sentimen negatif selalu lebih rendah daripada jumlah *tweet* dengan sentimen positif atau netral di setiap bulan. Sedangkan jumlah *tweet* dengan sentimen positif dan netral tidak jauh berbeda setiap bulannya. Jumlah *tweet* dengan sentimen netral selalu lebih banyak dari jumlah *tweet* dengan sentimen positif kecuali pada bulan November 2020, Januari 2021, Maret 2021 dan April 2021.

Dari data tersebut dapat disimpulkan bahwa meskipun ada masyarakat yang merespon negatif terhadap vaksinasi COVID-19, jumlahnya sangat rendah jika dibandingkan dengan orang yang merespon kegiatan vaksinasi dengan positif.



Gambar 4.16. Grafik Sentimen Perbulan

4.7.2. Analisis Hasil Klasifikasi

Seluruh hasil *training* dan *testing* model FastText, GloVe dan BERT dijabarkan secara detail pada Tabel 4.9. Dari tabel tersebut dapat diketahui bahwa performa paling baik pada saat menggunakan data original dihasilkan oleh *word embedding* FastText dengan nilai *f1 score* sebesar 62.10%. Hal yang sama juga ditunjukkan ketika menggunakan *word embedding* FastText ketika dilakukan pengujian menggunakan data yang telah dilakukan *oversampling* SMOTE. Akurasi yang diperoleh ketika menggunakan *word embedding* FastText sebesar 71.11%.

Penggunaan *f1 score* dan akurasi pada saat menentukan performa berdasarkan pada saat dataset tidak seimbang (jumlah sampel dalam satu kelas jauh lebih besar daripada jumlah sampel di kelas lain), akurasi tidak dapat dianggap sebagai ukuran yang andal lagi, karena memberikan estimasi yang terlalu optimis dari kemampuan pengklasifikasi pada kelas mayoritas sehingga pada ke dataset

yang belum seimbang ukuran performa yang digunakan adalah *f1 score* (Gu dkk., 2009).

Tabel 4.9. Hasil Seluruh Uji yang dilakukan

Embedding	Training Data	Training Monitoring	Tahapan preprocessing Data			Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
			Stem ming	Slang word	Frans				
FastText	Original	Model CheckPoint	√	-	-	69.95	63.18	61.29	62.10
			√	√	-	70.26	63.61	60.82	61.86
		√	√	√	70.20	63.78	60.82	61.95	
		√	√	-	69.36	64.45	57.58	59.34	
		√	√	-	69.64	63.24	57.72	59.29	
		√	√	√	69.85	64.72	58.16	59.90	
	SMOTE	Model CheckPoint	√	-	-	70.04	64.90	62.36	63.19
			√	√	-	71.11	65.90	60.23	61.99
		√	√	√	70.35	63.27	62.05	62.57	
		√	-	-	69.42	65.09	55.48	57.00	
		√	√	-	69.01	63.02	57.26	58.73	
		√	√	√	69.61	66.93	54.69	56.01	
GloVe	Original	Model CheckPoint	√	-	-	66.92	60.24	53.70	55.07
			√	√	-	67.02	61.47	54.60	56.13
		√	√	√	67.33	60.81	54.88	56.34	
		√	-	-	65.63	60.56	49.48	48.96	
		√	√	-	65.55	60.02	50.96	51.49	
		√	√	√	66.62	61.51	52.69	54.02	
	SMOTE	Model CheckPoint	√	-	-	67.74	64.14	53.78	55.18
			√	√	-	67.51	64.17	51.69	52.21
		√	√	√	67.17	62.54	54.20	55.87	
		√	-	-	66.08	61.36	53.78	55.22	
		√	√	-	66.64	64.81	49.81	49.15	
		√	√	√	66.88	66.89	50.78	50.91	
Model BERT dengan corpus Wikipedia									
BERT	Original	Model CheckPoint	√	-	-	69.60	61.68	61.55	61.62
			√	√	-	71.55	65.84	59.97	61.71
		√	√	√	70.82	64.67	60.06	61.60	
		√	-	-	70.18	65.87	58.51	60.32	
		√	√	-	69.33	66.09	57.08	58.86	
		√	√	√	69.76	64.46	59.31	60.81	
	SMOTE	Model CheckPoint	√	-	-	65.84	58.28	62.30	59.20
			√	√	-	65.70	58.51	61.58	58.88
		√	√	√	68.59	60.71	61.51	61.08	
		√	-	-	57.29	56.23	62.58	52.80	
		√	√	-	45.36	57.51	57.99	44.57	
		√	√	√	64.64	57.93	62.75	58.30	

Hasil yang diperoleh dari penelitian ini juga serupa dengan hasil penelitian (Nurdin dkk., 2020) dan (Adipradana dkk., 2021) dimana penggunaan *word embedding* FastText memiliki akurasi terbaik dibandingkan *word embedding* yang lain. Disisi lain, hasil penelitian ini berbeda dengan yang dilakukan oleh (Pasaribu

dkk., 2020) dan (D'Sa dkk., 2020) dimana performa terbaik dihasilkan oleh *word embedding* BERT dengan metode klasifikasi Bi-LSTM. Hal ini terjadi karena model BERT yang digunakan pada penelitian ini bukan merupakan model BERT yang dilatih dari dataset yang ada, melainkan berasal dari model yang dilatih menggunakan *corpus* Wikipedia Bahasa Indonesia. Perbandingan hasil kinerja terbaik pada tiap model data dan *word embedding* ditunjukkan pada Gambar 4.17 dan 4.18.



Gambar 4.17. Grafik Perbandingan Akurasi Terbaik dari Word Embedding FastText dan GloVe



Gambar 4.18 Grafik Perbandingan Akurasi Terbaik dari Word Embedding BERT

Ditinjau lagi secara lanjut, dari seluruh akurasi terbaik dari tiap *word embedding* ditemukan fakta bahwa semua proses *training monitoring* yang digunakan adalah ModelCheckpoint. Fungsi ModelCheckpoint pada dasarnya diterapkan untuk menyimpan model menjadi file dengan ekstensi h5 pada setiap iterasi pelatihan jika matrik *val_accuracy* mengalami peningkatan selama proses iterasi pelatihan model (Oktaviana dkk., 2021). Penerapan ModelCheckpoint dilakukan karena ada kemungkinan akurasi validasi yang dihasilkan oleh model akan menurun pada akhir 100 iterasi (Alsyabani dkk., 2021).

Dari sisi penggunaan 3 (tiga) tahap *preprocessing*, penggunaan *preprocessing* berupa proses identifikasi frasa kurang memiliki pengaruh dalam proses peningkatan kinerja pada tiap model. Setelah ditelusuri lebih lanjut, hal ini terjadi karena *word embedding* dapat mengidentifikasi kata yang sering muncul secara bergandengan/bersamaan dan akan menganggap kata-kata yang muncul secara bersamaan tersebut sebagai sebuah *vocab* (Bhattacharjee, 2018 ; Pennington

dkk.,2014). Secara umum, hasil yang diperoleh dalam penelitian ini bukan yang terbaik secara keseluruhan namun sudah mencapai tujuan penelitian dalam mencari embedding dalam domain bahasa Indonesia

4.8. Proses Vektorisasi

Neural network memproses data berupa angka secara matematis dan menghasilkan output yang juga berupa vektor angka. Di sisi lain, bidang ilmu *Natural Language Processing* mencoba untuk mengelola data input agar dapat dipahami oleh *neural network* dengan cara mengubah data input menjadi angka. Proses perubahan kata-kata menjadi *vector* angka disebut vektorisasi.

Pada penelitian ini, proses vektorisasi dilakukan menggunakan *word embedding*. Teknik *word embedding* akan melatih sebuah model untuk dapat menghasilkan *vector* sesuai dengan dimensi yang ditentukan. Proses vektorisasi tradisional menggunakan *One-Hot Encoding* membuat representasi setiap kata menggunakan angka biner. Metode ini mempunyai kelemahan jika *corpus* berisi jumlah data yang sangat besar dan tidak mampu merepresentasikan hubungan *semantic* dan *statistic* antar data.

Karen Spärck Jones (1972) mengembangkan metode TF-IDF yang mampu mengukur signifikansi atau tingkat pentingnya sebuah kata dalam *corpus* secara statistic berdasarkan frekuensi kemunculan data tersebut dalam korpus. Selain itu, jumlah dokumen dimana sebuah kata muncul juga dikalkulasi untuk menentukan *vector* dari sebuah kata. Teknik ini dapat beroperasi dengan cepat, namun tidak

mampu menyimpan informasi hubungan antar kata. Untuk menutupi kekurangan tersebut, penulis (Mikolov et al., 2013) mengembangkan metode Word2Vec.

Metode Word2Vec akan melatih model neural network secara *unsupervised*. Model yang sudah dilatih ini nanti akan mampu men-*generate vector* dari sebuah kata. Terdapat dua pilihan metode Word2Vec yang dapat digunakan, yaitu Continuous Bag-of-Words dan Skip-gram. Word2Vec akan mencoba memprediksi konteks dari sebuah kata menggunakan *neural network* dengan 1 *hidden layer*. Seperti proses *training neural network* pada umumnya, nilai bobot pada masing-masing input akan diperbaharui secara berkala sesuai dengan korpus yang menjadi input model. Jika dua kata yang berbeda sebagian besar muncul dalam konteks yang sama dan salah satu kata tersebut digunakan sebagai input, maka model akan menampilkan prediksi yang sangat mirip sebagai output.

Berbeda dengan Word2Vec yang hanya bekerja pada *local context* yaitu mempertimbangkan kata-kata lain yang berada sebelum dan sesudah target kata (*neighbouring words*), GloVe yang dikembangkan oleh penulis (Pennington et al., 2014) dibuat untuk melihat konteks kata pada keseluruhan korpus (*global*). Glove didasarkan pada teknik faktorisasi matriks pada matriks kata berkonteks.

Pada proses awal melatih *word embedding* GloVe, model akan membentuk matriks dengan jumlah baris dan kolom yang sama dengan jumlah kata di dalam korpus. Setiap elemen X_{ij} dari matriks tersebut mewakili seberapa sering kata i muncul dalam konteks kata j . Untuk setiap kata akan dicari konteks kata dalam beberapa area yang ditentukan oleh *window_size* sebelum kata dan *window_size* setelah kata tersebut. Hal ini dapat dirumuskan pada Formula 16. Dari matrix yang

sudah dibuat, dapat dihitung kemungkinan atau tingkat kedekatan satu kata dengan kata yang lain. Perhitungan probabilitas ini dapat dilihat pada Formula 17 dan 18 dimana k adalah jumlah kata di dalam korpus.

$$X_{ij} = \#j \text{ muncul pada konteks } i \quad (16)$$

$$P(i|j) = X_{ij} / X_i \quad (17)$$

$$X_i = \sum_k X_{ik} \quad (18)$$

Metode GloVe secara efisien memanfaatkan informasi statistik dengan melatih hanya pada elemen bukan nol dalam matriks kemunculan kata-kata, bukan pada seluruh matriks atau pada jendela konteks individu dalam korpus besar. Model menghasilkan ruang vektor dengan substruktur yang bermakna dan mempunyai kinerja yang sangat baik pada task analogi kata, similarity dan named entity recognition.

Permasalahan pada GloVe adalah besarnya model yang akan dibuat jika model dilatih pada korpus yang besar dan mempunyai jumlah kata yang besar pula. Selain itu, GloVe tidak mampu membuat representasi vector pada kata yang tidak pernah dilihat sebelumnya.

Sebagai contoh, pada sebuah korpus terdapat dua dokumen berikut:

Dokumen 1 : saya sudah divaksin

Dokumen 2 : saya sudah di tes pcr

Dari dua dokumen tersebut, dapat diketahui bahwa korpus sederhana ini berisi 6 kata. Dengan demikian akan dibentuk sebuah matriks berdimensi 6×6 yang kurang lebih seperti pada Tabel 4.10. Dari Tabel 4.10 terlihat bahwa kata "saya" muncul dua kali dalam konteks kata "sudah", maka $X_{\text{saya, sudah}} = 2$ seperti

dirumuskan pada formula 16. Perhitungan yang sama dilakukan pada elemen matriks yang lain.

Tabel 4. 10. Contoh tabel perhitungan matriks kata oleh GloVe

	saya	sudah	divaksin	di	tes	pcr
saya	0	2	0	0	0	0
sudah	2	0	1	1	0	0
divaksin	0	1	0	0	0	0
di	0	1	0	0	1	0
tes	0	0	0	1	0	1
pcr	0	0	0	0	1	0

Metode FastText dikembangkan oleh penulis (Bojanowski et al., 2017) dari metode Word2Vec. Metode ini tidak mempelajari setiap kata menjadi vector secara langsung, namun memecah kata menggunakan teknik n-gram. Sebagai contoh, jika teknik n-gram dengan n=3 digunakan pada kata "vaksin", maka dapat dipecah menjadi <va, vak, aks, ksi, sin, in>. Tanda "<" dan ">" melambangkan awal dan akhir sebuah kata. Setelah sebuah kata direpresentasikan dengan n-gram, maka model skip-gram akan mempelajari *embedding*. Nilai word *embedding* didapatkan dari rerata nilai *vector n-gram embeddings* ditambah nilai vector dari kata itu sendiri. Hal ini direpresentasikan pada formula 19.

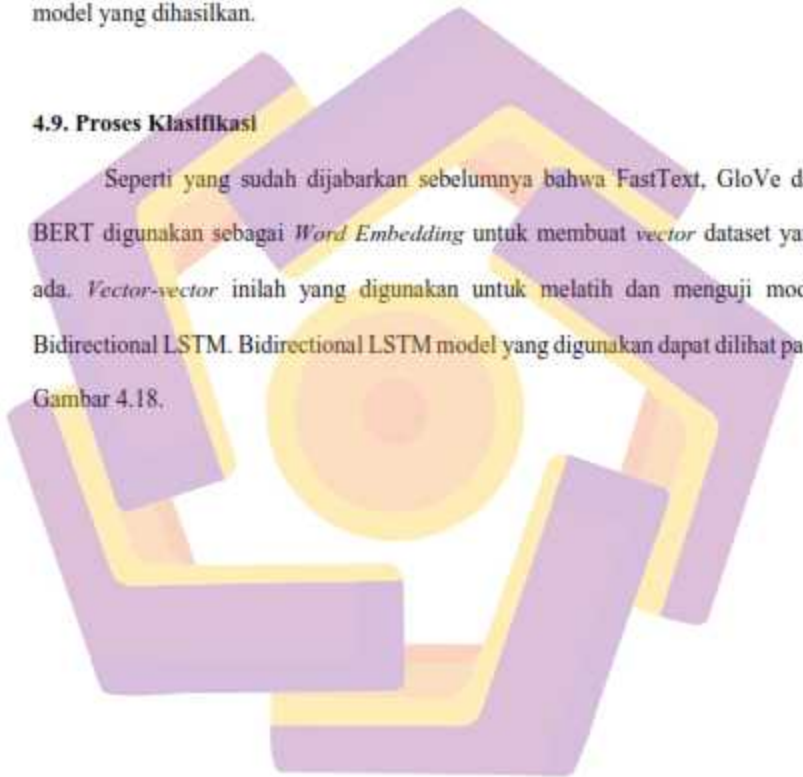
$$h_w = \sum_{g \in w} x_g \quad (19)$$

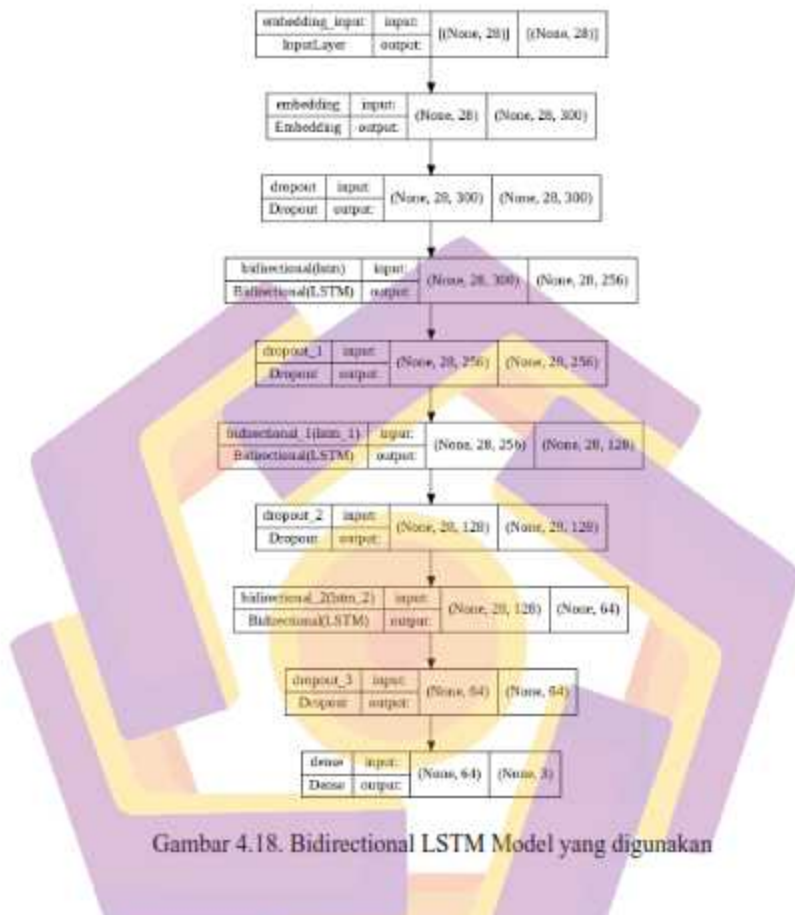
Model FastText ini dianggap sebagai model *bag of words* dimana FastText dapat membuat representasi *vector* bahkan untuk kata yang tidak pernah ditemui sebelumnya pada proses *training*. Nilai *word embedding* untuk kata yang tidak ada dalam *corpus* latih didapatkan dari rerata nilai vector n-gram embeddings saja tanpa ditambahkan nilai vector dari kata itu sendiri.

FastText akan mencari representasi *vector* dan setiap potongan yang dibuat oleh *n-gram*. Kemampuan ini tidak dimiliki oleh GloVe maupun Word2Vec. Di sisi lain, seperti namanya, FastText dapat melakukan training model dalam waktu yang lebih cepat dibandingkan metode embedding lain tanpa mengurangi tingkat akurasi model yang dihasilkan.

4.9. Proses Klasifikasi

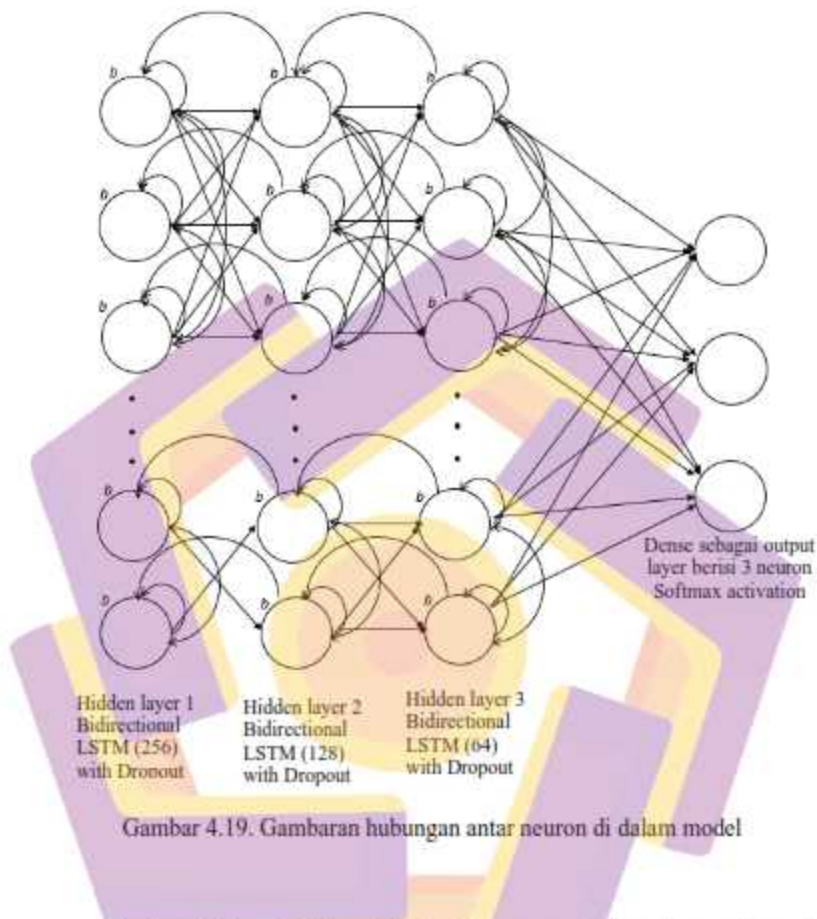
Seperti yang sudah dijabarkan sebelumnya bahwa FastText, GloVe dan BERT digunakan sebagai *Word Embedding* untuk membuat *vector* dataset yang ada. *Vector-vector* inilah yang digunakan untuk melatih dan menguji model Bidirectional LSTM. Bidirectional LSTM model yang digunakan dapat dilihat pada Gambar 4.18.





Gambar 4.18. Bidirectional LSTM Model yang digunakan

Dari Gambar 4.18 dapat terlihat bahwa maksimum jumlah kata perdokumen atau perbaris dataset setelah semua tahapan *preprocessing* adalah 28. Setiap kata akan direpresentasikan dengan vector berdimensi 1×300 dan menjadi output dari *Embedding* layer. Data tersebut kemudian diproses ke beberapa layer Bidirectional LSTM hingga menuju layer Dense yang berfungsi sebagai *output* layer.



Gambar 4.19. Gambaran hubungan antar neuron di dalam model

Gambar 4.19 menjelaskan lebih lanjut mengenai setiap layer pada model.

Luaran dari *embedding* layer berupa *vector* berupa n kata \times 300 akan menjadi input bagi *hidden* layer pertama. Fungsi LSTM dapat gambarkan ke dalam formula (20).

$$y' = a \left(\sum_{j=1}^n W_{ij} X_j + b \right) \quad (20)$$

y' = luaran dari LSTM Cell

a = fungsi aktivasi

n = jumlah data

W = bobot pada setiap input data

b = bias

Secara sederhana, output dari sebuah *neuron* merupakan hasil perkalian antara setiap input dengan bobotnya ditambah nilai bias dan kemudian hasilnya dihitung dengan fungsi aktivasi yang digunakan. Pada penelitian ini, fungsi aktivasi yang digunakan adalah fungsi aktivasi default dari LSTM. Fungsi aktivasi tanh mempunyai luaran dengan range -1 hingga 1. Fungsi aktivasi Tanh dapat didefinisikan dengan formula (21) dimana x adalah nilai yang ingin diubah.

$$F(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (21)$$

Output dari sebuah *neuron* di *hidden layer* pertama akan menjadi masukan untuk setiap *neuron* pada *hidden layer* kedua yang nanti nilainya akan dikali dengan masing-masing bobot lagi. *Output* dari *hidden layer* kedua akan menjadi masukan bagi *hidden layer* ketiga. Yang terakhir, *output* dari *hidden layer* ketiga akan menjadi masukan bagi *dense layer*. *Output* dari *dense layer* akan menjadi hasil prediksi dari model terhadap sebuah dokumen cuitan.

$$E = y - y' \quad (22)$$

Dari perbandingan antara hasil prediksi dengan *output* yang seharusnya akan didapatkan nilai *error*. Hal ini digambarkan pada formula (22). Nilai *error* ini akan digunakan untuk memperbaiki nilai bobot dan bias pada setiap masukan di layer-layer sebelumnya. Pada awalnya, bobot pada setiap masukan akan ditentukan secara acak oleh model. Nilai tersebut kemudian secara perlahan diperbaiki agar *output* yang dihasilkan oleh *neuron* semakin mendekati hasil yang seharusnya. Seperti yang terlihat pada Gambar 4.19, ada aliran data dari setiap *neuron* tidak

hanya kembali ke neuron tersebut, namun juga kembali ke neuron lain pada *hidden* layer sebelumnya. Hal ini dapat terjadi karena pada setiap LSTM layer ditambahkan Bidirectional layer sehingga proses pembelajaran (*update* bobot dan bias) dapat terjadi dalam 2 arah (tidak hanya *feed-forward*). Proses *update* bobot dan bias pada setiap masukan dapat dilihat pada formula (23) dan formula (24). Proses *update* ini akan dilakukan secara terus menerus untuk setiap *batch* data. Pada penelitian ini, nilai *batch* ditentukan sebesar 64 yang artinya proses *update* bobot dan bias akan dilakukan setelah model mencoba memprediksi *output* dari 64 buah dokumen tweet. Proses *update* bobot dan bias akan didasarkan dari nilai *error* hasil prediksi dari 64 dokumen tweet. Proses ini akan terus dilanjutkan hingga model menyelesaikan prediksi untuk keseluruhan dokumen tweet yang artinya menyelesaikan 1 *epoch* training.

$$W' = W + (lr \times E \times X) \quad (23)$$

$$b' = b + (lr \times E) \quad (24)$$

W' = bobot yang baru (hasil update)

W = bobot yang lama

lr = learning rate

e = nilai error

X = nilai masukan yang dihitung sebelum proses update

b' = nilai bias yang baru (hasil update)

b = nilai bias yang lama

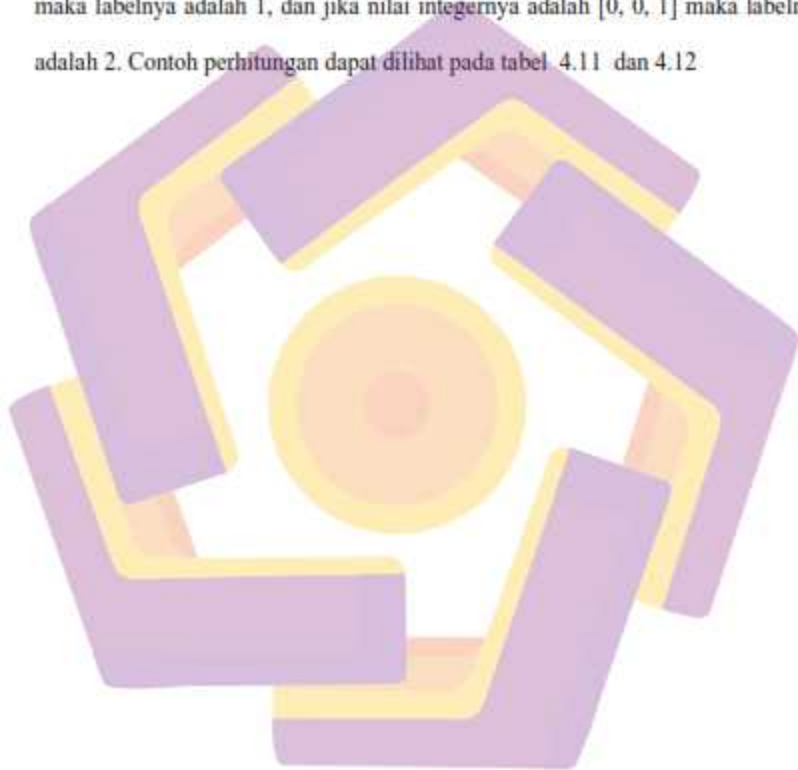
Seperti yang terlihat pada Gambar 4.18 bahwa diantara setiap *hidden layer* dengan *hidden layer* berikutnya terdapat *Dropout layer* yang membantu untuk menghindari model dari *overfit*. Seperti yang sudah dijelaskan sebelumnya bahwa nilai bobot akan di-*update* secara berkala selama proses *training*. Setelah dilatih berkali-kali, terkadang bobot pada sebuah masukan akan menjadi lebih prediktif dari yang lain. Sebaliknya, terkadang nilai bobot yang lain akan menjadi semakin lemah dan akan diabaikan. Kondisi ini disebut sebagai *co-adaptation*. Kondisi ini tidak dapat diperbaiki oleh teknik *regularization* jika model mempunyai banyak *hidden layer*. Untuk mengatasi masalah ini, *dropout layer* mengurangi partisipasi bobot pada input tertentu sesuai dengan nilai parameternya. Pada penelitian ini, nilai parameter yang digunakan adalah 0.5 dimana nilai merupakan nilai standar untuk banyak kasus klasifikasi. Dengan memberikan nilai 0.5 pada *dropout layer*, maka dropout akan mengurangi setengah dari nilai partisipasi bobot dalam perhitungan luaran sebuah neuron.

$$\text{softmax}(x) = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \quad (25)$$

Fungsi aktivasi softmax digunakan sebagai fungsi aktivasi pada output layer. Fungsi aktivasi softmax dapat direpresentasikan pada formula (25). Fungsi aktivasi softmax akan menghitung *luaran* dari masing-masing neuron pada *dense layer* menggunakan formula 22 dan menghasilkan angka diantara 0 dan 1. Pada penelitian ini terdapat 3 neuron pada *dense layer* yang artinya akan menghasilkan sebuah vector 1 x 3. Berikut adalah contoh luaran dari *dense layer* yang sudah dikenai proses perhitungan oleh fungsi aktivasi softmax:

$$[0.8750667, 0.076537415, 0.04839585]$$

Setiap *luaran* yang dihasilkan akan mempunyai salah satu nilai yang bernilai lebih dari 0.5. Dengan mengubahnya menjadi integer, nilai tersebut akan menjadi [1, 0, 0]. Dengan menggunakan fungsi `argmax` dari *library* `numpy`, *luaran* ini dapat diubah menjadi label 0. Selanjutnya, jika nilai integernya adalah [0, 1, 0] maka labelnya adalah 1, dan jika nilai integernya adalah [0, 0, 1] maka labelnya adalah 2. Contoh perhitungan dapat dilihat pada tabel 4.11 dan 4.12.



Tabel 4. 11 Contoh kalimat tweet

No.	Tweet										Label	
	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10		
1	apabila	di	vaksin	apa	sudah	bisa	tular	virus	covid			Netral
2	suntik	vaksin	covid	bukan	sembuh	malah	jadi	titan				Negatif
3	bang	komentar	vaksin	covid	gimana							Netral

Tabel 4. 12 Contoh Perhitungan Klasifikasi menggunakan Bi-LSTM

	id	Neuron 1 output layer										Neuron 2 output layer										Neuron 3 output layer										Y' Softmax			Y' Argmax			Y			R
				W _{1,1}		
				
Tweet	0.074700	...	0	0.1	...	0.17	0.6	0.23	0.2	0.5	0.30	...	0.07	0.341	0.8	-0.7	0.2	0.0	...	0.4	0.0	0.4	-0.6	0.2	0.2	0.2	0.2	1	0	0	1	0	0	0	0	0	0	0	0	T			
Tweet	-0.294022	...	0	0.1	...	0.15	0.4	0.31	0.8	0.4	0.34	...	0.07	0.34	0.1	-0.1	0.2	0.0	...	0.4	0.0	0.0	0.47	0.0	0.4	0.2	0.4	1	0	0	0	0	0	1	0	1	0	0	0	F			
Tweet	-0.032367	...	0	0.1	...	0.17	0.6	0.2	0.2	0.0	0.30	...	0.07	0.34	0.2	0.23	0.2	0.0	...	0.4	0.0	0.4	0.0	0.2	0.4	0.2	0.2	1	0	0	1	0	0	1	0	0	0	0	0	T			

lr (learning rate) = 0.001

 $w_{1,1} - w_{10,1}$ = randombh₁ = randombh₂ = bh₁ + (lr x e)bh₃ = bh₂ + (lr x e)

R = Result (T: True ; F:False)

 $oh_1 = (x_1 \times w_{1,1}) + \dots + (x_{10} \times w_{10,1}) + (bh_1)$ $oh_2 = (x_1 \times w_{1,2}) + \dots + (x_{10} \times w_{10,2}) + (bh_2)$ $oh_3 = (x_1 \times w_{1,3}) + \dots + (x_{10} \times w_{10,3}) + (bh_3)$ $w_{1,2} - w_{10,2}$ = W sebelumnya + (lr x error x input) $w_{1,3} - w_{10,3}$ = W sebelumnya + (lr x error x input)

e = Y - Y'

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil yang telah dicapai dalam penelitian mengenai sentimen analisis vaksin COVID-19 menggunakan beberapa jenis *word embedding* dan metode klasifikasi Bi-LSTM, dapat disimpulkan beberapa hal sebagai berikut:

- a. Sentimen masyarakat Indonesia melalui sosial media Twitter pada periode September 2020 hingga Juni 2021 mengenai vaksin COVID-19 mayoritas bersifat netral dengan pesentase sebesar 47% disusul dengan sentimen positif sebesar 44% dan sentimen negatif sebesar 9%. Dalam sudut pandang lain sentimen positif mengenai vaksin COVID-19 lebih besar daripada sentimen negatif. Artinya, meskipun ada masyarakat yang merespon negatif terhadap vaksinasi COVID-19, jumlahnya sangat rendah jika dibandingkan dengan masyarakat yang merespon kegiatan vaksinasi dengan positif.
- b. Nilai performa yang dihasilkan untuk tiap *word embedding* yang digunakan dalam penelitian ini cukup berbeda. Ketika dataset yang digunakan tidak seimbang, nilai performa diambil dari *f1 score* sehingga performa terbaik pada saat menggunakan *word embedding* FastText adalah sebesar 62.10%, dengan nilai *precision* 63.18%, *recall* 61.29% dan akurasi sebesar 69.95% sedangkan saat menggunakan *word embedding* GloVe nilai *f1 score* tertinggi sebesar 56.34%, *precision* 60.81%, *recall* 54.88% dan akurasi 67.33%. Ketika dataset yang digunakan dilakukan *oversampling* parameter yang digunakan untuk

melihat performa diambil dari nilai akurasi setiap hasil uji, pada *word embedding* FastText akurasi tertinggi yang diraih sebesar 71.11% , *precision* 65.90%, *recall* 60.23% dan *f1 score* 61.99% sedangkan saat menggunakan GloVe akurasi tertinggi sebesar 67.74%, dengan nilai *precision* 64.11%, *recall* 53.78% dan *f1 score* 55.18%. Sebagai tambahan, pada saat menggunakan *word embedding* BERT *f1 score* tertinggi sebesar 61.71%, *precision* 65.84%, *recall* 59.97% dan akurasi 71.55%.

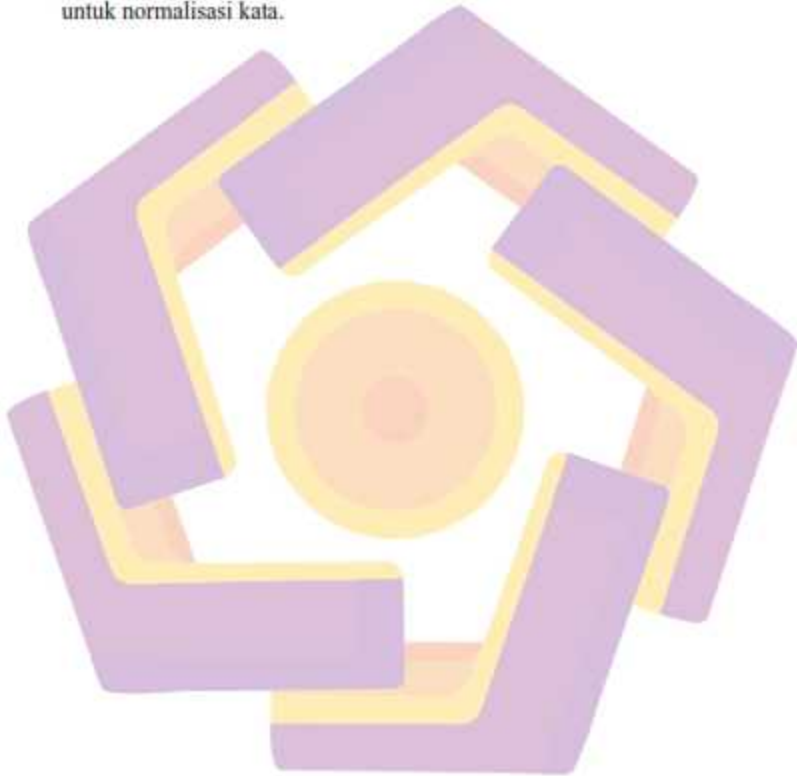
- c. Di antara *word embedding* FastText, GloVe dan BERT yang digunakan dalam penelitian ini, pada akhirnya hanya dua *word embedding* yang dapat dibandingkan secara “*apple to apple*” yaitu FastText dan GloVe. Karena terkendala komputasi, *word embedding* BERT tidak dianalisis bersamaan dengan *word embedding* FastText dan GloVe. Sehingga dari hasil pengujian model yang menggunakan *word embedding* FastText dan diklasifikasi menggunakan Bi-LSTM menghasilkan kinerja yang lebih baik, yaitu sebesar 71.11%.

5.2. Saran

Ada beberapa saran yang direkomendasikan untuk penelitian selanjutnya pada topik penelitian yang sama, antara lain:

- a. Jika peneliti selanjutnya memiliki *resource* komputasi yang besar maka disarankan untuk melakukan *fine-tuning* pada model BERT menggunakan dataset sendiri dengan harapan proses ini dapat meningkatkan akurasi dari hasil pengujian.

- b. Percobaan lebih lanjut dapat mengeksplorasi lebih dalam lagi *hyper-paramater* pada metode klasifikasi Bi-LSTM sebagai upaya optimalisasi model.
- c. Performa kinerja dapat ditingkatkan dengan melakukan perbaikan teknik *pre-processing* seperti melakukan evaluasi terhadap kamus *slang word* dan *frasa* untuk normalisasi kata.



DAFTAR PUSTAKA

PUSTAKA BUKU

- Bhattacharjee, J. (2018). *FastText Quick Start Guide*. Packt Publishing.
- Moolayil, J., 2019, Learn Keras for Deep Neural Networks – A Fast-Track Approach to Modern Deep learning with Python, Apress, Canada.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- Abdalla, G., & Özyurt, F. (2021). Sentiment Analysis of Fast Food Companies with Deep Learning Models. *Computer Journal*, 64(3), 383–390. <https://doi.org/10.1093/comjnl/bxaa131>
- Adipradana, R., Nayoga, B. P., Suryadi, R., & Suhartono, D. (2021). Hoax analyzer for Indonesian news using rnn with fasttext and glove embeddings. *Bulletin of Electrical Engineering and Informatics*, 10(4), 2130–2136. <https://doi.org/10.11591/eei.v10i4.2956>
- Aliyah Salsabila, N., Ardhito Winatmoko, Y., Akbar Septiandri, A., & Jamal, A. (2019). Colloquial Indonesian Lexicon. *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*. <https://doi.org/10.1109/IALP.2018.8629151>
- Alsyabani, O. M. A., Utami, E., & Hartanto, A. D. (2021). Intrusion Detection System Model Based on Gated Recurrent Unit to Detect Anomaly Traffic. *ICOIACT 2021 - 4th International Conference on Information and Communications Technology: The Role of AI in Health and Social Revolution in Turbulence Era*. <https://doi.org/10.1109/ICOIACT53268.2021.9564003>
- An, B., Wu, W., & Han, H. (2018). Deep active learning for text classification. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3271553.3271578>
- Basari, A. S. H., Hussin, B., Ananta, I. G. P., & Zeniarja, J. (2013). Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization. *Procedia Engineering*, 53. <https://doi.org/10.1016/j.proeng.2013.02.059>
- Beysolow II, T. (2018). *Applied natural language processing with python*. Springer.
- Bhattacharjee, J. (2018). *fastText Quick Start Guide* (Juli 2018). Packt Publishing Ltd.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146.

- Bono, S. A., Vilella, E. F. de M., Siau, C. S., Chen, W. S., Pengpid, S., Hasan, M. T., Sessou, P., Ditekemena, J. D., Amodan, B. O., Hosseinipour, M. C., Dolo, H., Fodjo, J. N. S., Low, W. Y., & Colebunders, R. (2021). Factors affecting COVID-19 vaccine acceptance: an international survey among low-and middle-income countries. *Vaccines*, 9(5). <https://doi.org/10.3390/vaccines9050515>
- BPMI Setpres. (2021). *Presiden Jokowi Menerima Vaksin Covid – 19 Perdana*. <https://www.presidentri.go.id/siaran-pers/president-jokowi-menerima-vaksin-covid-19-perdana/>
- Cahya. (2022). *cahya / bert-base-indonesian-522M*. <https://huggingface.co/cahya/bert-base-indonesian-522M>
- Cambria, E., Poria, S., Gelbukh, A., & Thelwall, M. (2017). Sentiment Analysis Is a Big Suitcase. *IEEE Intelligent Systems*, 32(6). <https://doi.org/10.1109/MIS.2017.4531228>
- Chandra, Y., & Jana, A. (2020). Sentiment analysis using machine learning and deep learning. *Proceedings of the 7th International Conference on Computing for Sustainable Global Development, INDIACom 2020*. <https://doi.org/10.23919/INDIACom49435.2020.9083703>
- Chen, L., Fang, B., Shang, Z., & Tang, Y. (2018). Tackling class overlap and imbalance problems in software defect prediction. *Software Quality Journal*, 26(1). <https://doi.org/10.1007/s11219-016-9342-6>
- Chen, Z., Jiang, C., Masood, M. K., Soh, Y. C., Wu, M., & Li, X. (2020). Deep learning for building occupancy estimation using environmental sensors. In *Studies in Computational Intelligence*. https://doi.org/10.1007/978-3-030-31760-7_11
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12.
- Cui, Z., Ke, R., & Wang, Y. (2017). Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. *Mathematics*.
- D'Sa, A. G., Illina, I., & Fohr, D. (2020). BERT and fastText Embeddings for Automatic Detection of Toxic Speech. *Proceedings of 2020 International Multi-Conference on: Organization of Knowledge and Advanced Technologies, OCTA 2020*. <https://doi.org/10.1109/OCTA49274.2020.9151853>
- Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15. <https://doi.org/10.1109/TNNLS.2021.3136503>

- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1*.
- Farzindar, A. A., & Inkpen, D. (2020). *Natural Language Processing for Social Media* (G. Hirst (ed.); Third Edit). Morgan & Claypool. <https://doi.org/10.2200/S00999ED3V01Y202003HLT046>
- Faturohman, F., Irawan, B., & Setianingsih, C. (2020). Analisis Sentimen Pada Bpjs Kesehatan Menggunakan Recurrent Neural Network. *e-Proceeding of Engineering, 7*(2).
- García, V., Sánchez, J. S., & Mollineda, R. A. (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems, 25*(1). <https://doi.org/10.1016/j.knosys.2011.06.013>
- Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*, 1–17. <http://arxiv.org/abs/2008.05756>
- Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks, 18*(5–6), 602–610.
- Gu, Q., Zhu, L., & Cai, Z. (2009). Evaluation measures of the classification performance of imbalanced data sets. *Communications in Computer and Information Science, 51*. https://doi.org/10.1007/978-3-642-04962-0_53
- Hidayatullah, A. F. (2016). Pengaruh Stopword Terhadap Performa Klasifikasi Tweet Berbahasa Indonesia. *JISKA (Jurnal Informatika Sunan Kalijaga), 1*(1). <https://doi.org/10.14421/jiska.2016.11-01>
- Hidayatullah, A. F., Cahyaningtyas, S., & Hakim, A. M. (2021). Sentiment Analysis on Twitter using Neural Network: Indonesian Presidential Election 2019 Dataset. *IOP Conference Series: Materials Science and Engineering, 1077*(1), 012001. <https://doi.org/10.1088/1757-899x/1077/1/012001>
- Hirzani, F. A., Maharani, W., Informatika, F., & Telkom, U. (2015). Analisis Sentimen Review Produk Menggunakan Pendekatan Berbasis Kamus. *e-Proceeding of Engineering, 2*(2), 5891–5898.
- Jain, K., & Kaushal, S. (2018). A Comparative Study of Machine Learning and Deep Learning Techniques for Sentiment Analysis. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2018*.

<https://doi.org/10.1109/ICRITO.2018.8748793>

- Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.
- JustAnotherArchivist. (2021). *snsrape*.
<https://github.com/JustAnotherArchivist/snsrape>
- Khairunnisa, S., Adiwijaya, A., & Faraby, S. Al. (2021). Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19). *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(2). <https://doi.org/10.30865/mib.v5i2.2835>
- Lestandy, M., Abdurrahim, A., & Syafa'ah, L. (2021). Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent Neural Network dan Naive Bayes. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(2).
- Listyarini, S. N., & Anggoro, D. A. (2021). Analisis Sentimen Pilkada di Tengah Pandemi Covid-19 Menggunakan Convolution Neural Network (CNN). *Jurnal Pendidikan dan Teknologi Indonesia*, 1(7). <https://doi.org/10.52436/1.jpti.60>
- Liu, C., Zhang, P., Li, T., & Yan, Y. (2019). Semantic Features Based N-Best Rescoring Methods for Automatic Speech Recognition. *Applied Sciences*, 9(23), 5053.
- Masters, D., & Luschi, C. (2018). Revisiting Small Batch Training for Deep Neural Networks. *CoRR, abs/1804.0*. <http://arxiv.org/abs/1804.07612>
- Mengistie, T. T., & Kumar, D. (2021). Deep Learning Based Sentiment Analysis On COVID-19 Public Reviews. *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, 444–449.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.
- Nazief, B. A. A. & Adriani, M. (2005). Confix- stripping: Approach to Stemming Algorithm for Bahasa Indonesia. *Conferences in Research and Practice in Information Technology Series*, 38(4).
- Novalita, N., Herdiani, A., & Lukmana, I. (2019). *Identifikasi Cyberbullying Pada Media Sosial Twitter Menggunakan Metode Klasifikasi Random Forest*.
- Nugroho, K. S., Sukmadewa, A. Y., Wuswilahaken Dw, H., Bachtiar, F. A., & Yudistira, N. (2021). BERT Fine-Tuning for Sentiment Analysis on Indonesian Mobile Apps Reviews. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3479645.3479679>
- Nurdeni, D. A., Budi, I., & Santoso, A. B. (2021). *Sentiment Analysis on Covid19 Vaccines in Indonesia: From The Perspective of Sinovac and Pfizer*. 122–127.

<https://doi.org/10.1109/eiconcit50028.2021.9431852>

- Nurdin, A., Anggo Seno Aji, B., Bustamin, A., & Abidin, Z. (2020). Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks. *Jurnal Tekno Kompak*, 14(2), 74. <https://doi.org/10.33365/jtk.v14i2.732>
- Nurrohmah, M. A., & SN, A. (2019). Sentiment Analysis of Novel Review Using Long Short-Term Memory Method. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 13(3). <https://doi.org/10.22146/ijccs.41236>
- Oktaviana, U. N., Hendrawan, R., Annas, A. D. K., & Wicaksono, G. W. (2021). Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(6) SE-Artikel Teknologi Informasi. <https://doi.org/10.29207/resti.v5i6.3607>
- Pasaribu, D. J. M., Kusriani, K., & Sudarmawan, S. (2020). Peningkatan Akurasi Klasifikasi Sentimen Ulasan Makanan Amazon dengan Bidirectional LSTM dan Bert Embedding. *Inspiration: Jurnal Teknologi Informasi dan Komunikasi*, 10(1), 9–20. <https://doi.org/10.35585/inspir.v10i1.2568>
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.
- Poria, S., Hussain, A., & Cambria, E. (2018). *Multimodal Sentiment Analysis* (A. Hussain & E. Cambria (ed.)). Springer Nature. <https://doi.org/10.1007/978-3-319-95020-4>
- Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2017). Challenges of Sentiment Analysis in Social Networks: An Overview. In *Sentiment Analysis in Social Networks*. <https://doi.org/10.1016/B978-0-12-804412-4.00001-2>
- Pratama, R. P., & Tjahyanto, A. (2022). The influence of fake accounts on sentiment analysis related to COVID-19 in Indonesia. *Procedia Computer Science*, 197, 143–150. <https://doi.org/https://doi.org/10.1016/j.procs.2021.12.128>
- Pratiwi, N. I., Budi, I., & Alfina, I. (2019). Hate speech detection on Indonesian instagram comments using FastText approach. *2018 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2018*. <https://doi.org/10.1109/ICACSIS.2018.8618182>
- Pristiyono, Ritonga, M., Ihsan, M. A. Al, Anjar, A., & Rambe, F. H. (2021). Sentiment analysis of COVID-19 vaccine in Indonesia using Naïve Bayes Algorithm. *IOP Conference Series: Materials Science and Engineering*, 1088(1). <https://doi.org/10.1088/1757-899x/1088/1/012045>
- Putranto, H. A., Setyawati, O., & Wijono, W. (2016). Pengaruh Phrase Detection dengan POS-Tagger terhadap Akurasi Klasifikasi Sentimen menggunakan SVM. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*,

5(4). <https://doi.org/10.22146/jnteti.v5i4.271>

- Randolph, H. E., & Barreiro, L. B. (2020). Herd Immunity: Understanding COVID-19. In *Immunity* (Vol. 52, Nomor 5). <https://doi.org/10.1016/j.immuni.2020.04.012>
- Redaksi, C. I. (2020a). *Hot Issue: Jokowi Tunjuk Menkes Baru, Terawan Diganti BGS?* <https://www.cnbcindonesia.com/news/20201221121604-4-210539/hot-issue-jokowi-tunjuk-menkes-baru-terawan-diganti-bgs>
- Redaksi, C. I. (2020b). *Joe Biden Disuntik Vaksin Covid-19.* <https://www.cnbcindonesia.com/news/20201222145043-8-210917/joe-biden-disuntik-vaksin-covid-19>
- Ritchie, H., Ortiz-Ospina, E., Beltekian, D., Mathieu, E., Hasell, J., Macdonald, B., Giattino, C., Appel, C., & Roser, M. (2021). *Coronavirus (COVID-19) Vaccinations.* <https://ourworldindata.org/covid-vaccinations>
- Riza, M. A., & Charibaldi, N. (2021). Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text. *International Journal of Artificial Intelligence & Robotics (IJAIR)*, 3(1), 15–26. <https://doi.org/10.25139/ijair.v3i1.3827>
- Robbani, H. A. (2016). *Sastrawi*. MIT.
- Rothman, D. (2021). *Transformers for Natural Language Processing - Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more.* Packt Publishing Ltd.
- Roy, A., & Ojha, M. (2020). Twitter sentiment analysis using deep learning models. *2020 IEEE 17th India Council International Conference, INDICON 2020*, 3–8. <https://doi.org/10.1109/INDICON49873.2020.9342279>
- Salam, A., Zeniarja, J., & Khasanah, R. S. U. (2018). *Analisis Sentimen Data Komentar Sosial Media Facebook Dengan k-Nearest Neighbor (Studi Kasus Pada Akun Jasa Ekspedisi Barang J&T Ekspres Indonesia).*
- Sallam, M. (2021). Covid-19 vaccine hesitancy worldwide: A concise systematic review of vaccine acceptance rates. In *Vaccines* (Vol. 9, Nomor 2). <https://doi.org/10.3390/vaccines9020160>
- Satapathy, R., Cambria, E., & Hussain, A. (2017). *Sentiment Analysis in the Bio-Medical Domain - Techniques, Tools, and Applications* (A. Hussain & E. Cambria (ed.)). Springer Nature. <https://doi.org/10.1007/978-3-319-68468-0>
- Sinaga, A. (2017). Implementasi Sentiment Analysis untuk Menentukan Tingkat Popularitas Tujuan Wisata. *Prosiding Seminar Nasional Teknologi dan Rekayasa Informasi Tahun 2017, November.*
- Sistem, R., Lestandy, M., Abdurrahim, A., & Syafa, L. (2021). *Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent.* 5(10), 802–808.

- Sofa, A. (2021). *Covid-19 Varian Delta dan Hal-hal yang Harus Kamu Perhatikan* | Blog | Portal Jakarta Smart City. Jakarta Smart City. <https://smartcity.jakarta.go.id/blog/758/covid-19-varian-delta-dan-hal-hal-yang-harus-kamu-perhatikan>
- Solis Arce, J. S., Warren, S. S., Meriggi, N. F., Scacco, A., McMurry, N., Voors, M., Syunyaev, G., Malik, A. A., Aboutajdine, S., Adejo, O., Anigo, D., Armand, A., Asad, S., Atyera, M., Augsburg, B., Awasthi, M., Ayesiga, G. E., Bancalari, A., Björkman Nyqvist, M., ... Omer, S. B. (2021). COVID-19 vaccine acceptance and hesitancy in low- and middle-income countries. *Nature Medicine*, 27(8). <https://doi.org/10.1038/s41591-021-01454-y>
- Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*, 17(1). <https://doi.org/10.1016/j.aci.2018.08.003>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 5998–6008.
- WHO. (2021). *Status of COVID-19 Vaccines within WHO EULPQ evaluation process*. https://extranet.who.int/pqweb/sites/default/files/documents/Status_COVID_VAX_16Feb2021.pdf
- Wibisono, Y. (2019). *GloVe untuk Wikipedia Bahasa Indonesia*. <https://yudiwbs.wordpress.com/2018/04/02/glove-untuk-wikipedia-bahasa-indonesia/>
- Wibowo, D. A., & Musdholifah, A. (2021). Sentiments Analysis of Indonesian Tweet about Covid-19 Vaccine Using Support Vector Machine and Fasttext Embedding. *2021 4th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2021*. <https://doi.org/10.1109/ISRITI54043.2021.9702871>
- Worldometer. (2021). <https://www.worldometers.info/coronavirus/country/indonesia/>
- Yulianto, A., Herdiani, A., & Sardi, I. L. (2019). Klasifikasi Keberpihakan Tweet Menggunakan Multinomial Naive Bayes (studi Kasus: Pemilihan Presiden 2019). *eProceedings of Engineering*, 6(2).
- Yulita, W., Nugroho, E. D., & Algifari, M. H. (2021). Analisis Sentimen Terhadap Opini Masyarakat Tentang Vaksin Covid - 19 Menggunakan Algoritma Naive Bayes Classifier. 2(2), 1–9.
- Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4). <https://doi.org/10.1002/widm.1253>

SITUS INTERNET

- BPMI Setpres. (2021). *Presiden Jokowi Menerima Vaksin Covid – 19 Perdana*. <https://www.presidentri.go.id/siaran-pers/president-jokowi-menerima-vaksin-covid-19-perdana/> (diakses 02 Januari 2022)
- Cahya. (2022). *cahya / bert-base-indonesian-522M*. <https://huggingface.co/cahya/bert-base-indonesian-522M> (diakses 06 Maret 2022)
- JustAnotherArchivist. (2021). *snsrape*. <https://github.com/JustAnotherArchivist/snsrape> (diakses 12 Juli 2021)
- Redaksi, C. I. (2020a). *Hot Issue: Jokowi Tunjuk Menkes Baru, Terawan Diganti BGS?* <https://www.cnbcindonesia.com/news/20201221121604-4-210539/hot-issue-jokowi-tunjuk-menkes-baru-terawan-diganti-bgs> (diakses 02 Januari 2022)
- Redaksi, C. I. (2020b). *Joe Biden Disuntik Vaksin Covid-19*. <https://www.cnbcindonesia.com/news/20201222145043-8-210917/joe-biden-disuntik-vaksin-covid-19> (diakses 02 Januari 2022)
- WHO. (2021). *Status of COVID-19 Vaccines within WHO EUL/PQ evaluation process*. https://extranet.who.int/pqweb/sites/default/files/documents/Status_COVID_VAX_16Feb2021.pdf (diakses Juni 2021)
- Wibisono, Y. (2019). *GloVe untuk Wikipedia Bahasa Indonesia*. <https://yudiwbs.wordpress.com/2018/04/02/glove-untuk-wikipedia-bahasa-indonesia/> (diakses 03 Januari 2022)
- Worldometer. (2021). <https://www.worldometers.info/coronavirus/country/indonesia/> (diakses 15 Oktober 2021)

LAMPIRAN



SYNERGY CONSULTANT INDONESIA

Pusat: Komplek Cahaya Ratu Elok Jl. Sanggiringan No. 52 Banjarbaru, Kalimantan Selatan
Cabang: Green Residence Jl. Banjar Indah Raya, Jl. A. Yani KM. 5 Banjarmasin, Kalimantan Selatan
Praktek Bersama: LPT GLOBAL Jl. Banjar Indah Raya Permai 1 Banjarmasin, Kalimantan Selatan
Email: Synergyconsultant.id@gmail.com
Phone: 081351139523 (Shanty Komalasari, M. Psi., Psikolog)

Banjarbaru, 14 April 2022

No : 01/SCI/IV/2021

Ket : Surat Keterangan Validasi Data Penelitian

SURAT KETERANGAN

Yang bertanda tangan di bawah ini:

Nama : Shanty Komalasari, M. Psi., Psikolog

No. SIPP : 04804-14/0065-21-2-3

Dengan ini menyatakan bahwa data penelitian milik saudara:

Nama : Kartikasari Kusuma Agustiningsih

No. Mlis : 20.55.1408

Judul Penelitian : Metode Bidirectional LSTM Menggunakan Word Embedding Untuk Analisis Sentimen Terhadap Vaksin COVID-19 Di Indonesia Pada Media Sosial Twitter

Sudah divalidasi dan dapat dilanjutkan ke tahap penelitian selanjutnya. Demikian surat keterangan ini dibuat untuk digunakan sebagaimana mestinya.

Penanggung Jawab Konsultan Psikologi

Shanty Komalasari, M. Psi., Psikolog
SIPP. 0065-21-2-3