

**TESIS**

**PENGEMBANGAN MODEL SENTIMEN ANALISIS MENGGUNAKAN  
KOMBINASI WORD EMBEDDING DAN BI-LSTM PADA KASUS  
CYBER BULLYING**



Disusun oleh:

**Nama : Fldya Farasalsabila**  
**NIM : 22.51.1189**  
**Konsentrasi : Business Intelligence**

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

**2024**

**TESIS**

**PENGEMBANGAN MODEL SENTIMEN ANALISIS MENGGUNAKAN  
KOMBINASI WORD EMBEDDING DAN BI-LSTM PADA KASUS  
CYBER BULLYING**

**SENTIMENT ANALYSIS MODEL DEVELOPMENT USING A  
COMBINATION OF WORD EMBEDDING AND BI-LSTM IN CYBER  
BULLYING CASES**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

**Nama : Fidyfa Farasalsablla**  
**NIM : 22.51.1189**  
**Konsentrasi : Business Intelligence**

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

**2024**

**HALAMAN PENGESAHAN**

**PENGEMBANGAN MODEL SENTIMEN ANALISIS MENGGUNAKAN  
KOMBINASI WORD EMBEDDING DAN BI-LSTM PADA KASUS  
CYBER BULLYING**

**SENTIMENT ANALYSIS MODEL DEVELOPMENT USING A COMBINATION OF  
WORD EMBEDDING AND BI-LSTM IN CYBER BULLYING CASES**

Dipersiapkan dan Disusun oleh

**Fidya Farasalsabila**

**22.51.1189**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Rabu, 03 April 2024.

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

Yogyakarta, 03 April 2023

**Rektor**

**Prof. Dr. M. Suyanto, M.M.**  
**NIK. 190302001**

## HALAMAN PERSETUJUAN

### PENGEMBANGAN MODEL SENTIMEN ANALISIS MENGGUNAKAN KOMBINASI WORD EMBEDDING DAN BI-LSTM PADA KASUS CYBER BULLYING

#### SENTIMENT ANALYSIS MODEL DEVELOPMENT USING A COMBINATION OF WORD EMBEDDING AND BI-LSTM IN CYBER BULLYING CASES

Dipersiapkan dan Disusun oleh

**Fidya Farasalsabila**

**22.51.1189**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Rabu, 03 April 2024.

**Pembimbing Utama**

**Prof. Dr. Ema Utami, S.SI., M.Kom**  
NIK. 190302037

**Anggota Tim Penguji**

**Dr. Kumara Ari Yuana, S.T., M.T**  
NIK. 190302575

**Pembimbing Pendamping**

**M. Hanafi, S.Kom., M.Eng., Ph.D**  
NIK. 190302024

**Tonny Hidayat, S.Kom., M.Kom., Ph.D**  
NIK. 190302182

**Prof. Dr. Ema Utami, S.SI., M.Kom**  
NIK. 190302037

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

Yogyakarta, 03 April 2024  
**Direktur Program Pascasarjana**

**Prof. Dr. Kusri, M.Kom.**  
NIK. 190302106

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Fidyfa Farasalsabila  
NIM : 22.51.1189  
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:  
**Pengembangan Model Sentimen Analisis menggunakan Kombinasi Word Embedding dan BI-LSTM pada Kasus Cyber Bullying**

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom  
Dosen Pembimbing Pendamping : M. Hanafi, S.Kom., M.Eng., Ph.D

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, 03 April 2024  
Yang Menyatakan,



AMIKOM  
METRA TEMPA  
27ALX1680826

Fidyfa Farasalsabila

## HALAMAN PERSEMBAHAN

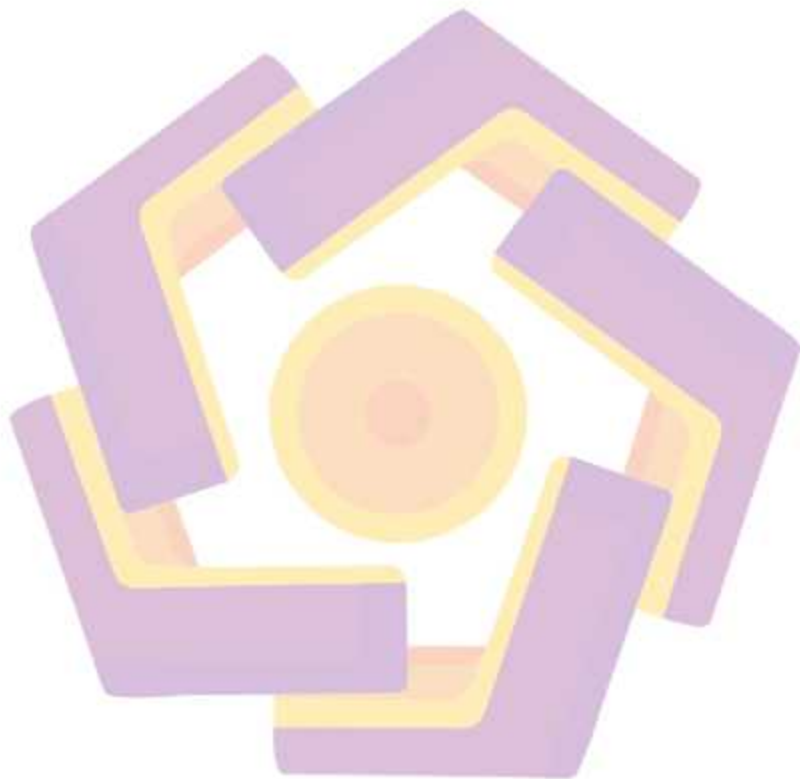
“I dedicated this to my parents. I thanked them for facilitating, supporting, and praying for me to be in this position. I thanked them for sacrificing their own well-being for the sake of my life's worth. I also thanked my family who contributed to supporting and helping my family when I was away from them. Additionally, I thanked my supervisor who was patient and great at guiding me until the end.”



## HALAMAN MOTTO

“Your life is as good your mindset.”

“Fix your prayers, and God will fix your life.”





## 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. Kusriani, 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 M. Hanafi, S.Kom., M.Eng., Ph.D selaku dosen Pembimbing Pendamping.
5. Bapak M. Rudyanto Arief, M.T selaku penguji Seminar Proposal Tesis
6. Bapak Anggit Dwi Hartanto, M.Kom selaku penguji Seminar Proposal Tesis.
7. Bapak Dr. Kumara Ari Yuana, S.T., M.T. selaku penguji Seminar Hasil Proposal Tesis dan Ujian Thesis.
8. Bapak Hanif Al Fatta, M.Kom., Ph.D. selaku penguji Seminar Hasil Proposal Tesis.



9. Bapak Tonny Hidayat, S.Kom., M.Kom., Ph.D selaku penguji Ujian Thesis.
10. Teman-teman Angkatan 28 MTI, serta semua pihak yang tidak bisa penulis sebutkan satu persatu terima kasih atas support dan doa nya.

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

Yogyakarta, 03 April 2024

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.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR ISTILAH.....	xvi
INTISARI.....	xvii
<i>ABSTRACT</i> .....	xviii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	5
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	7
1.5. Manfaat Penelitian.....	8
BAB II TINJAUAN PUSTAKA.....	9
2.1. Tinjauan Pustaka.....	9

2.2. Keaslian Penelitian.....	14
2.3. Landasan Teori.....	21
2.3.1. Analisis Sentimen.....	21
2.3.2. Cyber Bullying.....	22
2.3.3. Preprocessing.....	23
2.3.4. FastText.....	25
2.3.5. Bidirectional Encoder Representations from Transformers (BERT)	25
2.3.6. Bidirectional LSTM (Bi-LSTM).....	27
2.3.7. Confusion Matrix.....	28
<b>BAB III METODE PENELITIAN.....</b>	<b>31</b>
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	31
3.2. Metode Pengumpulan Data.....	32
3.3. Metode Analisis Data.....	33
3.3.1. Datasets.....	33
3.3.2. Proses Data Cleaning.....	34
3.3.3. Proses Tokenization.....	37
3.3.4. Stopword Removal.....	37
3.3.5. Proses Stemming.....	38
3.3.6. Proses Lemmatization.....	39
3.3.7. Exploratory Data Analysis (EDA).....	39

3.4. Alur Penelitian .....	41
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN .....</b>	<b>43</b>
4.1. Hasil Data Cleaning dan Preprocessing .....	43
4.2. Skenario Eksperimen .....	45
4.3. Eksperimen dan Evaluasi Kombinasi FastText dan Bi-LSTM.....	46
4.4. Eksperimen dan Evaluasi Kombinasi BERT dan Bi-LSTM.....	59
4.5. Analisis Hasil Eksperimen.....	76
<b>BAB V PENUTUP.....</b>	<b>82</b>
5.1. Kesimpulan .....	82
5.2. Saran .....	84
<b>DAFTAR PUSTAKA .....</b>	<b>86</b>

## DAFTAR TABEL

Tabel 2. 1. Matriks literatur review Analisis Kombinasi Word Embedding dengan Metode Bidirectional LSTM untuk Klasifikasi Sentimen Cyber Bullying.....	14
Tabel 2. 2. Tabel Confusion Matrix.....	28
Tabel 3. 1. Spesifikasi Hardware dan Software.....	42
Tabel 4. 1. Skenario Eksperimen.....	46
Tabel 4. 2. Hasil Kombinasi FastText+Bi-LSTM.....	53
Tabel 4. 3. Hasil Training FastText+Bi-LSTM.....	56
Tabel 4. 4. Hasil Kombinasi BERT+Bi-LSTM.....	67
Tabel 4. 5. Hasil Training BERT+Bi-LSTM.....	70
Tabel 4. 6. Tabel Perbandingan Word Embedding+Bi-LSTM.....	77

## DAFTAR GAMBAR

Gambar 2.1. Arsitektur BERT (Roy & Ojha, 2020) .....	26
Gambar 2.2. Arsitektur Bi-LSTM (Cui dkk., 2017) .....	28
Gambar 3. 1. Alur Pengumpulan Data .....	32
Gambar 3.2. Proses Case Folding .....	35
Gambar 3.3. Proses penghilangan karakter spesial .....	36
Gambar 3. 4. Proses penghilangan emoji.....	36
Gambar 3. 5. Proses Tokenization .....	37
Gambar 3. 6. Proses stopword removal .....	38
Gambar 3. 7. Proses stemming.....	38
Gambar 3. 8. Proses lemmatization.....	39
Gambar 3. 9. Grafik total jumlah data.....	40
Gambar 3. 10. Grafik persentase data berdasarkan label .....	40
Gambar 3. 11. Alur Penelitian .....	41
Gambar 4. 1. Grafik Total Data .....	43
Gambar 4. 2. Grafik Presentase Data .....	44
Gambar 4. 3. Baris Kode Model FastText .....	47
Gambar 4. 4. Hasil Vektorisasi Kata FastText.....	48
Gambar 4. 5. Baris Kode Proses Hyperparameter Tuning FastText+Bi-LSTM..	49
Gambar 4. 6. Baris Kode Penentuan Parameter FastText+Bi-LSTM.....	50
Gambar 4. 7. Output Hasil Hyperparameter FastText+Bi-LSTM .....	50
Gambar 4. 8. Baris kode kombinasi model FastText+Bi-LSTM.....	51
Gambar 4. 9. Baris kode runtime FastText+Bi-LSTM .....	52



Gambar 4. 10. Hasil training akurasi model FastText-Bi-LSTM .....	54
Gambar 4. 11. Hasil training loss model FastText-Bi-LSTM.....	55
Gambar 4. 12. Confusion Matrik model FastText-Bi-LSTM .....	58
Gambar 4. 13. Grafik Persentase FastText+Bi-LSTM Berdasarkan Prediksi Label .....	59
Gambar 4. 14. Baris kode memuat model BERT.....	60
Gambar 4. 15. Baris kode proses pelatihan dan pengujian model BERT .....	61
Gambar 4. 16. Baris Kode Proses Hyperparameter Tuning BERT-Bi-LSTM .....	62
Gambar 4. 17. Baris Kode Penentuan Parameter Terbaik BERT-Bi-LSTM .....	63
Gambar 4. 18. Output hasil hyperparameter BERT-Bi-LSTM.....	64
Gambar 4. 19. Baris kode kombinasi model BERT+ Bi-LSTM.....	65
Gambar 4. 20. Baris Kode runtime BERT-Bi-LSTM.....	66
Gambar 4. 21. Hasil training akurasi model BERT-Bi-LSTM .....	68
Gambar 4. 22. Hasil training loss model BERT-Bi-LSTM .....	69
Gambar 4. 23. Confusion Matrix model BERT-Bi-LSTM.....	75
Gambar 4. 24. Grafik Persentase BERT+Bi-LSTM Berdasarkan Prediksi Label	76



## DAFTAR ISTILAH

**Datasets:** Kumpulan data terorganisir untuk analisis, pembelajaran mesin, atau tujuan penelitian lainnya.

**Data Latih (Training Data):** Subset dari dataset yang digunakan untuk melatih model pembelajaran mesin.

**Data Uji (Test Data):** Subset lain dari dataset yang digunakan untuk menguji kinerja model yang telah dilatih.

**Word Embedding:** Representasi numerik kata-kata dalam vektor multidimensi untuk digunakan dalam pemrosesan bahasa alami dan pembelajaran mesin.

**Hyperparameter Tuning:** Proses mencari kombinasi terbaik dari hyperparameter untuk suatu model pembelajaran mesin.

**Hyperopt:** Hyperopt adalah sebuah library Python yang digunakan untuk melakukan optimisasi hyperparameter secara otomatis.

**Tree-structured Parzen Estimator (TPE):** TPE adalah salah satu algoritma pencarian hyperparameter yang digunakan dalam optimisasi hyperparameter. TPE sering digunakan dalam library seperti Hyperopt.

**Library:** Library adalah kumpulan kode atau modul yang menyediakan berbagai fungsi atau alat yang dapat digunakan untuk mempermudah pengembangan perangkat lunak.

**Units:** Istilah merujuk pada unit-unit dalam konteks jaringan saraf tiruan (neural networks), yang merupakan blok-blok bangunan dasar dari jaringan saraf, seperti lapisan-lapisan (layers) atau unit-unit dalam lapisan tersebut.

**LR (Learning Rate):** yaitu seberapa besar langkah yang diambil oleh algoritma pembelajaran saat memperbarui bobot-bobot dalam proses pembelajaran.

**Dropout:** Dropout adalah teknik yang umum digunakan dalam jaringan saraf tiruan untuk mencegah overfitting. Ini secara acak mengabaikan sebagian unit selama proses pelatihan untuk mencegah ketergantungan yang terlalu besar pada unit tertentu.

**Epochs:** Epoch adalah satu putaran lengkap dari seluruh dataset melalui model pembelajaran mesin selama proses pelatihan.

## INTISARI

Peningkatan aktivitas online dalam beberapa tahun terakhir telah memberikan dampak signifikan pada kesehatan mental dan kesejahteraan individu, terutama dalam konteks fenomena cyber bullying. Melalui penggunaan platform media sosial dan forum daring, perilaku negatif seperti intimidasi, pelecehan, dan penghinaan semakin merajalela, menghadirkan tantangan serius dalam menciptakan lingkungan online yang aman dan inklusif. Dalam upaya untuk mengatasi masalah ini, penelitian ini mengusulkan pengembangan model analisis sentimen yang menggunakan kombinasi teknik word embedding dan arsitektur jaringan saraf rekuren berbasis LSTM (Bidirectional LSTM) untuk mendeteksi dan mengklasifikasikan pesan-pesan yang mengandung elemen cyber bullying.

Penelitian ini menggunakan data publik yang berasal dari Mendeley Data. Sebanyak 65.343 yang dikumpulkan dari platform youtube dalam rentang waktu 2022-2023. Dataset ini terdiri dari dua label, yakni 0 (non-cyber bullying) dan 1 (cyber bullying). Data yang diperoleh akan dilakukan preprocessing lalu kemudian divektorisasi menggunakan 2 jenis word embedding yaitu, FastText dan BERT dan diklasifikasikan menggunakan metode Bi-LSTM.

Dari percobaan yang dilakukan, diperoleh hasil bahwa kedua model menunjukkan tingkat akurasi yang tinggi, dengan nilai berturut-turut yakni FastText+Bi-LSTM sebesar 92,89% dan BERT+Bi-LSTM sebesar 93,17%. Terdapat perbedaan kecil dalam nilai akurasi namun model BERT+Bi-LSTM menunjukkan sedikit peningkatan performa dibandingkan dengan FastText+Bi-LSTM. Diantara word embedding FastText dan BERT yang digunakan dalam penelitian ini, pada akhirnya word embedding yang diklasifikasi menggunakan Bi-LSTM menghasilkan kinerja yang tinggi dan lebih baik adalah BERT dengan akurasi sebesar 93,17%.

Kata kunci: Analisis Sentimen, cyber bullying, Word Embedding, Bi-LSTM.

## **ABSTRACT**

*The rise in online activity in recent years has had a significant impact on the mental health and well-being of individuals, especially in the context of cyber bullying phenomena. Through the use of social media platforms and online forums, negative behaviors such as intimidation, harassment, and humiliation are increasingly prevalent, posing serious challenges in creating a secure and inclusive online environment. In an effort to address this problem, the study proposes the development of a sentimental analysis model that uses a combination of word embedding techniques and LSTM-based recurrent neural network architecture to detect and classify messages containing elements of cyber bullying.*

*This study uses public data from Mendeley Data. A total of 65,343 were collected from the YouTube platform in the period 2022-2023. This data set consists of two labels, namely 0 (non-cyber bullying) and 1 (cyber bullying). The data obtained will be preprocessed and then divectorized using two types of word embedding namely, FastText and BERT and classified using Bi-LSTM method.*

*From the experiments carried out, the results obtained were that both models showed a high level of accuracy, with respective values, namely FastText+Bi-LSTM of 92.89% and BERT+Bi-LSTM of 93.17%. There is a small difference in the accuracy values but the BERT+Bi-LSTM model shows a slight performance improvement compared to FastText+Bi-LSTM. Between the FastText and BERT word embeddings used in this research, in the end the word embedding that was classified using Bi-LSTM produced high and better performance, namely BERT with an accuracy of 93.17%.*

*Keyword: Sentiment Analysis, cyber bullying, Word Embedding, Bi-LSTM.*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Masyarakat didorong untuk menerima perkembangan dalam berkomunikasi di dunia maya yang merupakan bentuk dari pesatnya perkembangan media sosial saat ini. Pengguna dapat berkomunikasi dan bersosial dengan lebih cepat dan mudah berkat media sosial. Pengguna diberi kemudahan untuk berbagi dan menyebarkan informasi dengan memanfaatkan berbagai platform media sosial. Hal ini pastinya akan berdampak pada cara hidup, cara pandang, dan perilaku masyarakat. cyber bullying atau yang dikenal dengan istilah bullying di dunia maya didefinisikan sebagai tindakan bullying atau pelecehan yang dilakukan melalui pemanfaatan teknologi digital, seperti yang dikemukakan oleh UNICEF. Tujuan dari cyber bullying adalah untuk mengintimidasi korban, membuat mereka marah, sedih atau membuat mereka merasa malu pada diri mereka sendiri. Salah satu perbedaan utama antara bentuk bullying dan cyber bullying adalah cyber bullying akan meninggalkan jejak digital. Jejak digital dapat digunakan sebagai alat bukti untuk membantu menghentikan perbuatan tersebut, dapat berupa rekaman atau capture gambar. Istilah "cyber bullying" mengacu pada praktik intimidasi yang terjadi secara online.

Di sisi lain, masih sulit untuk mengidentifikasi kasus cyber bullying. Oleh karena itu diperlukan klasifikasi cyber bullying agar lebih mudah menentukan



apakah suatu pernyataan mengandung aspek cyber bullying atau tidak. Penggunaan pendekatan machine learning, khususnya analisis sentimen, merupakan salah satu metode yang dapat digunakan untuk mendeteksi cyber bullying. Menurut (Nurdeni dkk., 2021), analisis sentimen terhadap persepsi orang di media sosial dapat dilakukan dengan menganalisis cara individu memiliki gagasan tentang berbagai subjek. Dimungkinkan juga untuk menggunakannya untuk menyelidiki perspektif, perilaku dan sikap individu terhadap orang lain, serta perasaan terhadap diri sendiri atau aktivitas yang telah dilakukan hingga saat ini. Menurut (Basari dkk., 2013) fakta bahwa mengungkapkan pemikiran individu lain membuatnya berharga bagi mereka yang bertanggung jawab untuk mengambil keputusan. Oleh karena itu, dibentuklah metode klasifikasi untuk mengkategorikan suatu opini yang sedang berkembang, baik di media sosial maupun dalam bentuk media massa lainnya.

Beberapa penelitian terkait sentimen analisis cyber bullying di Indonesia telah banyak dilakukan dengan berbagai metode dan hasil sentimen yang beragam. Seperti pada penelitian lain yang menggunakan objek sama dilakukan oleh (Raj dkk. 2022). Dalam penelitiannya, mengusulkan deep learning framework untuk mengevaluasi tweet atau postingan media sosial twitter secara real-time serta mengidentifikasi konten cyber bullying di dalamnya dengan benar. Hasilnya menunjukkan bahwa pendekatan berbasis deep neural network-based lebih efektif daripada teknik konvensional dalam mendeteksi teks cyber bullying dengan akurasi sebesar 95%.

Metode word embedding juga digunakan dalam penelitian (Abdalla & Özyurt, 2021) 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%. 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. 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 tinggi.

Hasan dkk. (2023) telah melakukan investigasi pada sekumpulan penelitian analisis sentimen terhadap cyber bullying dan menyarankan beberapa saran yang salah satunya menambahkan BERT dalam penelitian selanjutnya sehingga model pre-trained dapat dengan mudah memprediksi perilaku pelaku cyber bullying secara online. Selain itu, pada penelitian (Nurdin dkk., 2020) yang juga melakukan penelitian membandingkan kinerja Word Embedding Word2Vec, Glove dan FastText untuk klasifikasi teks menggunakan algoritma Convolutional Neural Network (CNN). Dalam penelitian nya menghasilkan bahwa kinerja word embedding terbaik diperoleh dengan menggunakan Word Embedding FastText dengan F-1 score sebesar 71.5%.

Seiring dengan itu, literatur (Setiawan dkk., 2022) menyoroti adanya peluang untuk mengembangkan, membandingkan, menggabungkan, dan mengintegrasikan beberapa metode dalam upaya meningkatkan efektivitas pendeteksian, termasuk penggunaan teknik word embedding yang menjadi fokus penelitian mendatang. Salah satu pendekatan yang menonjol dalam literatur adalah penggunaan word embedding, yang telah terbukti efektif dalam merepresentasikan teks dalam teks secara numerik sehingga dapat dimengerti oleh model. Word embedding memungkinkan model untuk memahami makna dan hubungan antara kata-kata dalam teks, sehingga meningkatkan kemampuan model dalam melakukan klasifikasi sentimen cyberbullying dengan lebih akurat (Setiawan dkk., 2023). Berdasarkan temuan dari penelitian sebelumnya, dapat disimpulkan bahwa penggunaan word embedding memiliki pengaruh signifikan dalam meningkatkan akurasi klasifikasi (David & Renjith, 2021; Dharma dkk., 2022). Studi-studi ini menarik karena mengevaluasi pembelajaran mesin dan hasil kategorisasi deep learning secara berdampingan. Berdasarkan penelitian beberapa peneliti sebelumnya yang sudah dijelaskan di atas, maka peneliti akan mengembangkan algoritma dengan menambahkan hyperparameter dari Pustaka Hyperopt dan mengkombinasikan Bi-LSTM menggunakan 2 word embedding, yakni FastText dan BERT. Melalui penelitian ini, peneliti berharap dapat memberikan kontribusi baru terhadap pemahaman terkait deteksi dan klasifikasi cyberbullying, serta menawarkan solusi yang lebih efektif dalam menangani masalah yang semakin meningkat di lingkungan online.



## 1.2. Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut.

- a. Bagaimana Hasil Akurasi dan hyperparameter terbaik yang digunakan dalam mengkombinasikan FastText dengan Bi-LSTM pada klasifikasi teks cyber bullying?
- b. Bagaimana Hasil Akurasi dan hyperparameter terbaik yang digunakan mengkombinasikan BERT Embedding dengan Bi-LSTM pada klasifikasi teks cyber bullying?
- c. Word Embedding manakah diantara FastText dan BERT yang dikombinasikan dengan Bi-LSTM menghasilkan performa tertinggi dalam menangani kasus cyber bullying?

## 1.3. Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut.

- a. Dataset yang digunakan merupakan dataset publik yang diakses melalui web Mendeley Data dengan pencarian kata kunci "cyber bullying".
- b. Dataset yang digunakan adalah "Bahasa cyber bullying Datasets" yang merupakan data publik berbahasa indonesia yang dipublish pada 29 Agustus 2023.
- c. Dataset yang digunakan dari sumber data hanya dataset yang berasal dari komentar pada platform youtube.

- d. Menggunakan sebanyak 65.343 data cyber bullying yang didapat dari Mendeley Data.
- e. Dataset telah memiliki 2 label, yakni label 0 (non-cyber bullying) dan label 1 (cyber bullying).
- f. Proses data cleaning dan preprocessing menggunakan proses drop duplicate data, case folding, data cleansing, tokenization, stemming, stopword removal, dan lemmatization.
- g. Komposisi dataset yang digunakan pada eksperimen yakni menggunakan 70% data latih dan 30% data uji.
- h. Platform penelitian menggunakan Google Colaboratory Pro dengan run type A100.
- i. Library yang diantaranya adalah Sastrawi 1.0.1, Nlp-Id 0.1.5, Tensorflow 2.9.0, Transformers 4.35.2, Huggingface 0.20.3, Pandas 1.5.3, Numpy 1.25.2, NLTK 3.8.1, Regex 2023.12.25, Gensim 4.3.2, Sklearn 1.2.2 dan Hyperopt 0.2.7.
- j. Menggunakan 2 word embedding sebagai pembanding, yakni FastText dan BERT.
- k. Menggunakan 2 skenario eksperimen, yakni pada data imbalanced dan balanced
- l. Menggunakan pustaka NumPy yakni fungsi `np.random.randint` untuk penanganan imbalance dan balance pada data
- m. Menggunakan algoritma Bi-LSTM untuk melakukan klasifikasi.

- n. Menggunakan Hyperopt untuk optimasi hyperparameter secara otomatis untuk menemukan best parameters model.
- o. Jumlah epoch yang digunakan pada masing-masing model tergantung hasil hyperparameter yang ditentukan otomatis oleh metode optimasi hyperopt.
- p. Menggunakan Confusion Matrix sebagai evaluasi model.
- q. Output sentimen analisis diklasifikasikan menjadi 2, yaitu No (Non-cyber bullying) dan Yes (cyber bullying).
- r. Fokus pada melakukan perbandingan tingkat performa dari segi akurasi, presisi, recall dan f1 score pada model FastText+Bi-LSTM dan BERT+Bi-LSTM pada studi kasus cyber bullying.

#### **1.4. Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah sebagai berikut.

- a. Mengetahui hasil performa dari mengkombinasikan BERT Embedding dengan Bi-LSTM pada klasifikasi teks cyber bullying.
- b. Mengetahui hasil performa dari mengkombinasikan Fast-Text dengan Bi-LSTM pada klasifikasi teks cyber bullying.
- c. Mengetahui Word Embedding manakah yang memiliki performa tinggi jika dikombinasikan dengan Bi-LSTM dalam menangani kasus klasifikasi teks cyber bullying.
- d. Dapat mengetahui kekurangan yang didapat dari hasil percobaan penelitian untuk perbaikan di penelitian mendatang.

### 1.5. Manfaat Penelitian

Manfaat Penelitian dalam penelitian ini ditentukan sebagai berikut.

- a. Mengetahui hasil kombinasi word embedding manakah antara FastText+Bi-LSTM dan BERT+Bi-LSTM yang memiliki performa baik.
- b. Berkontribusi secara ilmiah dalam penggunaan kombinasi word embedding pada pengolahan data berbahasa Indonesia dengan model Bi-LSTM.
- c. Hasil analisis sentimen ini dapat digunakan sebagai pedoman untuk penelitian mendatang yang menganalisa sentimen publik terhadap cyber bullying di Indonesia pada platform YouTube.
- d. Dapat dijadikan referensi untuk penelitian lain dan wawasan untuk pengembangan penelitian selanjutnya.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Tinjauan Pustaka**

Dalam konteks penelitian mengenai deteksi cyberbullying, literatur yang relevan telah mengulas beragam pendekatan dan teknik yang digunakan untuk meningkatkan akurasi klasifikasi. Sebagai bahan perbandingan, penelitian ini akan mencantumkan beberapa penelitian terdahulu. Studi yang dilakukan sebelumnya (Setiawan dkk., 2023) mengusulkan model klasifikasi untuk mendeteksi cyberbullying dalam bahasa Indonesia dengan mengintegrasikan ekstraksi fitur TF-IDF menggunakan pendekatan dinamis n-Gram, menerapkan algoritma SVM dan k-NN, dengan temuan menunjukkan superioritas SVM dalam menghasilkan F-1 score sebesar 0.889, dan menyarankan studi lanjutan untuk pemilihan model ekstraksi fitur yang sesuai dari variasi TF-IDF, serta mempergunakan dataset yang sama untuk mengoptimalkan akurasi dengan metode yang berbeda sesuai rekomendasi penelitian terdahulu. Pada penelitian lain (Setiawan dkk., 2022) yang dilakukan penulis yang sama juga membahas peluang untuk pengembangan lebih lanjut terbuka, mengingat pola karakteristik cyberbullying yang dinamis dari waktu ke waktu, dan masih ada potensi untuk membandingkan serta mengintegrasikan beberapa metode, termasuk penggunaan word embedding, guna meningkatkan efektivitas dalam proses pendeteksian cyberbullying.

Studi lain (Garrido-Merchan dkk., 2023) membandingkan model BERT dengan metodologi Machine Learning Natural Language Processing (NLP)



tradisional melalui empat eksperimen yang dirancang, dengan tujuan menambahkan bukti empiris untuk mendukung penggunaan BERT sebagai pendekatan default dalam tugas-tugas NLP. Hasil eksperimen menunjukkan keunggulan BERT dalam semua percobaan yang dilakukan serta kemampuan adaptasinya terhadap fitur-fitur masalah NLP, seperti bahasa teks, yang menguatkan argumen untuk mengadopsi BERT sebagai teknik standar dalam domain NLP. Hasil eksperimen ini juga didukung dengan penelitian lain yaitu (Suhartono dkk., 2022) yang menunjukkan bahwa BERT secara signifikan lebih efektif dibandingkan dengan RoBERTa selama proses pelatihan dan validasi, dengan mencapai akurasi sebesar 91%. Penelitiannya mengintegrasikan penyematan kata Glove dan Word2Vec dengan arsitektur berbasis CNN untuk melakukan analisis sentimen terhadap ulasan produk obat, dengan tambahan metode berbasis deep learning seperti BERT dan RoBERTa.

Penelitian lain (Mazari & Kheddar, 2023) berfokus pada menganalisis cyber bullying dalam dataset dialek Aljazair. data multi-label beranotasi yang terdiri dari 14.150 komentar yang diambil dari Facebook, YouTube, dan Twitter, dan diberi label sebagai "hate speech", "offensive language" dan "cyber bullying". Tes eksperimental menggunakan model Deep Learning (DL) seperti Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bidirectional-LSTM (Bi-LSTM) dan Bidirectional-GRU (Bi-GRU) dan menunjukkan keberhasilan model Bi-GRU, yang mencapai hasil tertinggi untuk klasifikasi DL, dengan Akurasi 73,6% dan Skor F1 75,8%.

Selanjutnya pada penelitian (Riza & Charibaldi, 2021), telah melakukan penelitian yang mendeteksi emosi cyber bullying dari teks twitter menggunakan pendekatan LSTM dengan penyisipan kata FastText. Dalam penelitian ini deteksi emosi diklasifikasikan menjadi 6 kategori, yaitu "happiness", "anger", "fear", "disgust", "sadness" dan "surprise". Penelitian menggunakan model LSTM - Word2vec dan LSTM-Fast Text dengan 50 units dan 50 dropouts mendapatkan hasil akurasi, presisi, recall, dan f-1 score terbaik sebesar 73,15%. Penelitian lainnya juga dilakukan oleh (Murshed dkk., 2022) yang menggabungkan Recurrent Neural Networks (RNN) tipe Elman dengan Dolphin Echolocation Algorithm (DEA) dalam mendeteksi cyber bullying. Digunakan sebanyak 10.000 data yang diambil dari jaringan media sosial Twitter. Hasil percobaan menunjukkan bahwa DEA-RNN ditemukan lebih unggul di semua skenario dengan rata-rata akurasi 90,45%, presisi 89,52%, recall 88,98%, skor F1 89,25%, dan kota spesifik 90,94%.

Selanjutnya dalam penelitian (V dan Adolf 2021) melakukan percobaan dari kombinasi dari Convolutional Neural Network (CNN) dan Long Short-term Memory (LSTM) yang menghasilkan prediksi berbasis gambar mencapai akurasi sebesar 86% dan prediksi berbasis teks mencapai akurasi sebesar 85%. Penelitian ini mengidentifikasi kejadian cyber bullying dalam data multimodal untuk dapat mendeteksi gambar dan teks intimidasi secara bersamaan menggunakan data yang di dapat dari GitHub dan Kaggle. Penelitian lainnya juga diusulkan oleh (AlHarbi dkk., 2020) yang menerapkan teknik Ridge Regression (RR) dan Logistic Regression (LR) pada dataset tweet berbahasa arab dengan akurasi tertinggi



dicapai sebesar 91,17%. Hasil penelitian ini dapat memberikan wawasan tentang efektivitas dan kinerja algoritma Machine Learning dalam mengenali dan mengklasifikasikan konten cyber bullying.

Masih dengan objek yang sama yaitu cyber bullying, penelitian ini (Nurdin dkk., 2020) membandingkan kinerja Word Embedding Word2Vec, Glove dan FastText untuk klasifikasi teks menggunakan algoritma Convolutional Neural Network (CNN) pada dataset 20 newsgroup dan Reuters. Hasil akhir penelitian ini memperoleh nilai F-1 score sebesar 71.5% dengan menggunakan Word Embedding FastText. Namun, tidak terdapat perbedaan kinerja yang signifikan dibanding Word Embedding FastText dengan yang lain. Sementara penelitian lain yang dilakukan (Al-Hashedi dkk., 2023) menggunakan model yang dilatih berdasarkan fitur kontekstual, emosi dan sentimen pada kumpulan data cyber bullying menggunakan EDM dan berbasis leksikon. Hasil penelitian ini memaparkan bahwa penggunaan fitur emosi yang ditambahkan ke BERT menghasilkan skor ingatan 0,87 pada dataset Toxic, meningkatkan kinerja pendeteksian cyber bullying sebesar 0,5 dibandingkan dengan menggunakan BERT saja. Sedangkan penggunaan fitur sentimen mendapatkan skor 0,88 recall, meningkatkan model sebesar 0,6 recall dibandingkan dengan menggunakan BERT saja yang secara umum lebih besar dari baseline.

Penelitian lain (Siti Khomsah dkk., 2022) membandingkan keakuratan model analisis sentimen menggunakan Word2Vec dan FastText dengan parameter yang sama, yang kemudian digabungkan dengan Random Forest, Extra Tree, dan AdaBoost. Hasil penelitian menunjukkan bahwa FastText mencapai tingkat

akurasi yang lebih tinggi daripada Word2Vec saat menggunakan Extra Tree dan Random Forest sebagai pengklasifikasi. Leverage akurasi FastText mencapai 8%(baseline: Decision Tree 85%), dengan mencapai akurasi sebesar 93% dengan 100 estimator. Hasil eksperimen ini juga dibuktikan pada penelitian lain (Dharma dkk., 2022) yang menunjukkan bahwa FastText unggul dibandingkan GloVe dan Word2Vec pada dataset 20 newsgroup, dengan akurasi mencapai 97,2% untuk FastText, 95,8% untuk GloVe, dan 92,5% untuk Word2Vec. Selain itu juga didukung dengan penelitian (David & Renjith, 2021) yang membandingkan GloVe dan fastText dalam klasifikasi teks menggunakan dua arsitektur jaringan saraf, yaitu RNN dan CNN. Meskipun dalam klasifikasi gambar, metode pembelajaran mendalam seperti RNN dan CNN telah terbukti populer, dalam konteks klasifikasi teks, deep neural network dengan FastText menunjukkan peningkatan akurasi yang signifikan dibandingkan dengan GloVe. Analisis kinerja CNN menggunakan FastText mencapai akurasi sebesar 90.93%, sementara RNN dengan FastText mencapai akurasi 83.33%.

Dengan adanya Tinjauan Pustaka yang telah ditinjau, peneliti berencana untuk melakukan pengembangan lebih lanjut konsep ini dengan merujuk pada temuan dan saran dari Tinjauan Pustaka. Penelitian ini akan merancang sebuah penelitian yang bertujuan untuk mengimplementasikan metode kombinasi word embedding dengan Bi-LSTM dalam klasifikasi sentimen cyberbullying untuk menjawab beberapa pertanyaan yang masih terbuka dalam literatur. (AlHarbi dkk. 2020; Al-Hashedi dkk. 2023; Mazari dan Kheddar 2023; Murshed dkk. 2022; Nurdin dkk. 2020; Riza dan Charibaldi 2021; V dan Adolf 2021).

## 2.2. Keaslian Penelitian

Tabel 2. 1. Matriks literatur review Analisis Kombinasi Word Embedding dengan Metode Bidirectional LSTM untuk Klasifikasi Sentimen Cyber Bullying

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Sama atau Kelemahan	Perbandingan
1	The Use of Dynamic n-Gram to Enhance TF-IDF Features Extraction for Bahasa Indonesia Cyberbullying Classification	Yudi Setiawan, Nur Ulfa Maulidevi & Kridanto Surendro. 12th International Conference on Software and Computer Applications (ICSCA 2023)	Mengusulkan model klasifikasi mendeteksi cyberbullying bahasa Indonesia dengan mengimplementasikan ekstraksi fitur TF-IDF dengan pendekatan dinamis n-Gram, menggunakan algoritma SVM dan k-NN.	Temuan penelitian menunjukkan bahwa metode SVM mengungguli metode lain ketika model yang dilatih diuji menggunakan dataset cyberbullying dengan nilai F-1 score 0,889.	Studi ekstraksi fitur ini mempunyai peluang untuk dilakukan di masa depan, seperti melakukan studi mendalam mengenai pemilihan model ekstraksi yang sesuai dari variasi ekstraksi fitur TF-IDF.	Penelitian yang dilakukan akan menggunakan datasets cyberbullying yang sama digunakan dalam penelitian ini, namun akan menggunakan metode berbeda untuk mengoptimalkan akurasi sesuai saran penelitian terdahulu melalui studi literatur.
2	Deteksi Cyberbullying dengan Mesin Pembelajaran Klasifikasi (Supervised Learning): Peluang dan Tantangan	Jurnal Teknologi Informasi dan Ilmu Komputer (2022)	Artikel ini menguraikan penerapan metode pembelajaran klasifikasi (Supervised Learning) dengan modifikasi langkah untuk meningkatkan akurasi klasifikasi.	Proses deteksi cyberbullying di media sosial masih sangat terbuka dan masih dapat dikembangkan kedepannya. Pada penentuan pola karakteristik dari cyberbullying masih terdapat perubahan dari waktu ke waktu.	Masih terdapat peluang untuk membandingkan, menggabungkan, dan mengintegrasikan beberapa metode guna mendapatkan efektivitas proses pendeteksian cyberbullying dari berbagai sudut domain permasalahan.	Penelitian yang dilakukan akan menggunakan cyberbullying sebagai objek penelitian dengan metode menggabungkan dan membandingkan beberapa word embedding guna mendapatkan efektivitas pendeteksian cyberbullying.

Tabel 2.1. Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Comparing BERT Against Traditional Machine Learning Models in Text Classification	Eduardo C. Garrido Merchan, Roberto Gozalo-Brizuela & Santiago Gonzalez-Carvajal. Journal of Computational and Cognitive Engineering (2023).	Membandingkan model BERT dengan metodologi ML NLP tradisional dengan merancang empat eksperimen. Tujuan dari penelitian ini adalah menambuhkan bukti empiris untuk mendukung penggunaan BERT sebagai default pada tugas NLP.	Eksperimen menunjukkan keunggulan BERT pada kecempakan eksperimen yang dilakukan dan kemandirian fitur masalah NLP seperti bahasa teks menambahkan bukti empiris untuk menggunakan BERT sebagai teknik default dalam masalah NLP.	Meskipun tampaknya ini merupakan standar yang baik untuk tugas-tugas NLP, hasilnya dapat ditingkatkan. Untuk melakukannya, peneliti ingin meneliti model BERT hyperparameter yang disetel otomatis untuk setiap tugas NLP baru dengan pengoptimalan Bayesian.	Penelitian yang akan dilakukan akan mengikuti saran dari penelitian terdahulu untuk menggunakan hyperparameter yang disetel otomatis.
4	Deep neural networks and weighted word embeddings for sentiment analysis of drug product reviews	Derwin Suhartono, Kartika Purwandari, Hendrik Jeremy, Samuel Philip, Panji Arisaputra & Ivan Halim Parmonangan. Elsevier (2022).	Menggabungkan penyematan kata Glove dan Word2Vec dengan arsitektur berbasis CNN untuk menganalisis sentimen dalam ulasan produk obat. Metode berbasis deep learning (BERT dan RoBERTa) juga dilakukan dalam penelitian ini.	BERT lebih efektif dibandingkan model RoBERTa selama proses pelatihan dan validasi dengan akurasi 91%. Di antara konfigurasi yang dibandingkan, kinerja prediksi tertinggi juga diperoleh oleh model CNN menggunakan penyematan kata Glove dengan akurasi pengujian sebesar 84,87%.	Dibandingkan dengan metode tradisional, penerapan BERT terbukti jauh lebih mudah. Selain itu, perlu dicatat bahwa transfer learning sangat penting kedepannya.	Penelitian ini akan menggunakan salah satu metode yang disarankan yaitu BERT yang akan dikembangkan dengan kombinasi Bi-LSTM. Selain itu, hasilnya akan dibandingkan dengan kombinasi FastText+Bi-LSTM.



Tabel 2.1. Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	Deep Learning-based Analysis of Algerian Dialect Dataset Targeted Hate Speech, Offensive Language and cyber bullying	Ahmed Cherif Mazari & Hamza Kheddar. International Journal of Computing and Digital Systems, 2023.	Mendeteksi teks toxic Aljazair, dimana kumpulan data multi-label beranotasi yang terdiri dari 14.150 komentar yang diambil dari Facebook, YouTube, dan Twitter, dan diberi label sebagai "hate speech", "offensive language" dan "cyber bullying".	Beberapa percobaan telah dilakukan dengan menggunakan model Deep Learning (CNN, LSTM, GRU, Bi-LSTM dan Bi-GRU). Tes eksperimental menunjukkan keberhasilan model Bi-GRU, yang mencapai hasil tertinggi dengan Akurasi 73,6% dan Skor F1 75,8%.	Penelitian kedepannya dapat menggunakan model BERT, GPT-2 atau GPT-3 dan melakukan lebih banyak evaluasi untuk menentukan penyematan dan algoritme mana yang paling cocok.	Penelitian yang akan dilakukan salah satunya akan menggunakan BERT sesuai saran dan akan dikombinasikan dengan metode Bi-LSTM.
6	Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text	M. Alfa Riza & Novrido Charibaldi. International Journal of Artificial Intelligence & Robotics (IAIR), 2021.	Mendeteksi emosi (happiness, anger, fear, disgust, sadness, dan surprise) dari teks Twitter menggunakan model LSTM+Word2vec dan LSTM+Fast Text.	Penelitian menggunakan model LSTM - Word2vec dan LSTM-Fast Text dengan 50 units dan 50 dropouts 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 akan dilakukan akan menggunakan fitur deep learning untuk tugas klasifikasi dan akan menggunakan lebih banyak data sejumlah 65k+.

Tabel 2.1. Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
7	DEA-RNN: A Hybrid Deep Learning Approach for cyber bullying Detection in Twitter Social Media Platform	Belal Abdullah Hezam Murshed, Mufeed Ahmed Naji Saif, Suresha Mallappa & Hasib Daowd esmail Al-ariki. IEEE Access, 2022.	Mendeteksi cyber bullying di jaringan media sosial Twitter dengan data sebanyak 10.000 tweet menggunakan Model RNN tipe Elman & Dolphin Echolocation Algorithm (DEA) yang dioptimalkan dan membandingkan kinerjanya dengan Bi-LSTM, RNN, SVM, Multinomial Naive Bayes (MNB), Random Forests.	Hasil percobaan menunjukkan bahwa DEA-RNN ditemukan lebih unggul di semua skenario. Mengungguli pendekatan yang ada dalam mendeteksi cyber bullying di platform Twitter. DEA-RNN lebih efisien pada skenario 3 yang mencapai rata-rata akurasi 90,45%.	Tidak dapat melakukan analisis sehubungan dengan perilaku pengguna. Model di penelitian selanjutnya disarankan menggunakan model yang diusulkan dapat bekerja untuk mendeteksi cyber bullying dari konten seperti gambar, video, dan audio serta real-time.	Penelitian yang akan dilakukan akan menggunakan sumber data dengan platform berbeda, yakni youtube. Selain itu juga menggunakan kombinasi pendekatan deep learning namun dengan kombinasi word embedding berbeda, yakni FastText+Bi-LSTM dan BERT+Bi-LSTM.
8	Multimodal cyber bullying Detection using Hybrid Deep Learning Algorithms	Dr. Vijayakumar V, Dr. Hari Prasad D & Adolf P. 2021	Mengidentifikasi kejadian cyber bullying dalam data multimodal untuk dapat mendeteksi gambar dan teks intimidasi secara bersamaan menggunakan data yang di dapat dari GitHub dan Kaggle.	Model jaringan saraf hibrid dibangun dengan kombinasi dari Convolutional Neural Network (CNN) dan Long Short-term Memory (LSTM) yang menghasilkan prediksi berbasis gambar mencapai akurasi sebesar 86% dan prediksi berbasis teks mencapai akurasi sebesar 85%.	Kedepannya, video dan audio juga dapat diintegrasikan untuk mendeteksi cyber bullying. Opsi multi bahasa, lintas bahasa, dan campuran bahasa juga dapat diintegrasikan..	Penelitian yang akan dilakukan menggunakan data dari platform berbeda dari penelitian sebelumnya, yakni youtube. Selain itu juga menggunakan pendekatan deep learning berbeda, yakni kombinasi word embedding dengan metode Bi-LSTM.

Tabel 2.1. Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
9	Using Machine Learning Algorithms for Automatic cyber bullying Detection in Arabic Social Media	Bedoor Y. AlHarbi, Mashacl S. AlHarbi, Nouf J. AlZahrani, Meshaiel M. Alsheail & Dina M. Ibrahim. Journal of Information Technology Management, 2020.	Mendeteksi cyber bullying otomatis dalam pembelajaran mesin bahasa Arab menggunakan teknik Ridge Regression (RR) dan Logistic Regression (LR) untuk mendeteksi cyber bullying dalam bahasa Arab menggunakan dataset tweet.	91,17%, akurasi yang lebih baik tercapai dalam menerapkan teknik Ridge Regression (RR) dan Logistic Regression (LR).	Menerapkan Ridge Regression (RR) dan Logistic Regression (LR) dan membandingkan akurasi dengan teknik yang ada yang digunakan untuk deteksi otomatis cyber bullying dalam bahasa Arab untuk mengurangi cyber bullying.	Penelitian yang akan dilakukan akan menggunakan dataset berbahasa indonesia dengan kemajuan metode menggunakan pendekatan dan kombinasi deeplearning, yakni FastText+Bi-LSTM dan BERT+Bi-LSTM.
10	Perbandingan Kinerja <i>Word Embedding</i> Word2Vec, Glove dan FastText pada Klasifikasi Teks.	Arliyanti Nurdin, Bernadus Anggo Seno, Anugryani Bustamin & Zaenal Abidin. Jurnal Tekno Kompak, 2020.	Membandingkan kinerja <i>Word Embedding</i> Word2Vec, Glove dan FastText untuk klasifikasi teks menggunakan algoritma CNN pada dataset 20 <i>newsgroup</i> dan <i>Routers</i> .	Kinerja terbaik diperoleh dengan menggunakan <i>Word Embedding</i> FastText dengan F-1 score sebesar 71.5% pada dataset <i>Routers</i> . Namun, tidak terdapat perbedaan kinerja yang signifikan dibanding <i>Word Embedding</i> lain.	Pengunaan setiap <i>Word Embedding</i> sangat bergantung dengan dataset yang digunakan dan permasalahan yang akan diselesaikan.	Penelitian yang akan dilakukan selanjutnya akan menggunakan dataset yang lebih banyak (65k+) dan akan membandingkan <i>Word Embedding</i> BERT dan FastText yang akan dikombinasikan dengan metode Bi-LSTM untuk meningkatkan performa.



Tabel 2.1. Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
11	Cyber bullying Detection Based on Emotion	Mohammed Al-hashedi, Lay-Ki Soon, Hui-Ngo Goh, Amy Hui Lan Lim, & Eu-Gen Siew. IEEE Acces, 2023.	Mendeteksi cyber bullying menggunakan model yang dilatih berdasarkan fitur kontekstual, emosi dan sentimen pada kumpulan data cyber bullying menggunakan EDM berbasis leksikon. Dataset cyber bullying dari Wikipedia dan Twitter.	Hasilnya menunjukkan bahwa fitur emosi dan sentimen meningkatkan ketepatan deteksi cyber bullying dan mengungguli penggunaan fitur kontekstual BERT. Penggunaan fitur emosi yang ditambahkan ke BERT meningkatkan kinerja pendeteksian cyber bullying sebesar 0,5 dibandingkan dengan menggunakan BERT saja.	Penelitian selanjutnya dapat difokuskan pada peningkatan kumpulan data emosi dalam hal ukuran dan anotasi.	Penelitian yang akan dilakukan akan menggunakan kumpulan data yang lebih banyak 65k+. Selain itu juga akan mengkombinasikan salah satunya adalah model BERT dengan metode Bi-LSTM untuk meningkatkan performa.
12	The Accuracy Comparison Between Word2Vec and FastText On Sentiment Analysis of Hotel Reviews	Siti Khomsah, Rima Dias Ramadhani & Sena Wijayanto. JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi) 2022.	Membandingkan keakuratan model analisis sentimen menggunakan Word2Vec dan FastText menggunakan parameter yang sama dan digabungkan dengan Random Forest, Extra Tree, dan AdaBoost.	FastText mencapai akurasi yang lebih tinggi daripada Word2Vec saat menggunakan Extra Tree dan Random Forest sebagai pengklasifikasi. Akurasi leverage FastText 8% (baseline: Decision Tree 85%), dibuktikan dengan akurasi 93%, dengan 100 estimator.	Rekomendasi untuk penelitian lebih lanjut adalah menggabungkan pembelajaran mesin ensemble dengan vektorizer lain, seperti Glove atau Wang2Vec, dan juga mencoba beberapa estimator, parameter berbeda, dan berbagai kumpulan data yang sangat besar.	Penelitian yang akan dilakukan sesuai rekomendasi dari penelitian terdahulu dengan menggabungkan pembelajaran mesin lain, yakni BERT+Bi-LSTM dan FastText+Bi-LSTM.

Tabel 2.1. Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
13	The Accuracy Comparison Among Word2vec, Glove, And Fasttext Towards Convolution Neural Network (CNN) Text Classification	Eddy Muntina Dharma, Ford Lumban Gaol, Harco Leslie Hendric Spits Warnars, Benfano Soewito. Journal of Theoretical and Applied Information Technology (2022).	Penelitian ini bertujuan untuk membandingkan akurasi metode embedding 3 kata yaitu Word2Vec, GloVe dan FastText pada klasifikasi teks menggunakan algoritma Convolutional Neural Network.	FastText mengungguli Glove dan Word2vec untuk dataset 20 newsgroup, akurasiya adalah 97.2% untuk FastText, 95,8% untuk Glove, dan 92.5% untuk Word2Vec. Word2vec dan GloVe tidak dapat merepresentasikan vektor kata yang tidak ada dalam korpus (di luar kosa kata). Berbeda dengan FastText yang bisa diandalkan untuk masalah keluar kosakata ini.	Keakuratan word embedding ini bergantung pada kumpulan data yang digunakan dan domain masalah yang ditangani, oleh karena itu kumpulan data dan masalah domain lain yang harus diselesaikan dapat ditambahkan untuk studi di masa mendatang.	Penelitian akan dilakukan dengan mengambil saran untuk menambah kumpulan data. Sebanyak 65k+ data yang akan digunakan untuk meneliti performa kinerja pengembangan kombinasi FastText+Bi-LSTM dan BERT+Bi-LSTM.
14	Comparison of word embeddings in text classification based on RNN and CNN	Merlin Susan David & Shini Renjith. IOP Conference Series: Materials Science and Engineering (2021)	Menyajikan perbandingan GloVe dan fastText dalam klasifikasi teks menggunakan RNN dan CNN. Di bidang klasifikasi gambar, metode pembelajaran mendalam seperti RNN dan CNN terbukti populer.	Deep neural network dengan fastText menunjukkan peningkatan akurasi yang luar biasa dibandingkan GloVe. Analisis kinerja CNN menggunakan FastText 90.93%, sementara RNN dengan FastText 83.33%.	Hasil mungkin berbeda untuk kumpulan data lain berdasarkan atribut seperti ukuran kumpulan data, dan sebagainya.	Penelitian yang akan dilakukan akan menggunakan FastText namun akan dikombinasikan dengan Bi-LSTM sebagai bentuk pengembangan.

## 2.3. Landasan Teori

### 2.3.1. Analisis Sentimen

Analisis sentimen atau biasa juga disebut opinion mining adalah bidang studi yang khusus menganalisis terkait sentimen, opini, sikap dan emosi orang terhadap suatu entitas atau atribut yang diungkapkan melalui teks tertulis. Entitas yang dimaksud dapat berupa produk, jasa, organisasi, individu, peristiwa, isu dan topik (Liu, 2015). Menurut (Cambria dkk., 2017), tujuan dari analisis sentimen adalah untuk menentukan sikap seorang penulis atau pembicara terhadap beberapa tema atau polaritas kontekstual keseluruhan dari sebuah dokumen.

Dalam Kamus Collegiate Merriam-Webster, sentimen didefinisikan sebagai sikap, pemikiran, atau penilaian yang didorong oleh perasaan, sedangkan opini didefinisikan sebagai pandangan, penilaian, atau penilaian yang terbentuk dalam pikiran tentang suatu hal tertentu. Perbedaannya cukup halus, dan masing-masing berisi beberapa elemen yang lain. Definisi tersebut menunjukkan bahwa pendapat lebih merupakan pandangan konkret seseorang tentang sesuatu, sedangkan sentimen lebih merupakan perasaan (Pozzi dkk., 2017).

Ada dua metode pertama yang dikenal sebagai klasifikasi subjektif dan klasifikasi polaritas yang biasanya digunakan dalam melakukan analisis sentimen. Menurut (Cambria dkk., 2017), penggunaan subjektivitas diperlukan untuk menentukan apakah suatu pernyataan mencerminkan suatu sentimen atau tidak. Jika suatu kalimat memiliki nilai subjektivitas yang tinggi, maka dapat diartikan sebagai pandangan pribadi pengarang, sebaliknya, jika kalimat tersebut memiliki nilai subjektivitas yang rendah, maka kalimat tersebut dapat diartikan sebagai

pernyataan fakta. Dengan kata lain, tidak ada bias yang jelas dalam kalimat tersebut. Nilai polaritas digunakan untuk memutuskan apakah sebuah kalimat dengan tingkat subjektivitas tinggi yang mengekspresikan sentimen harus dikategorikan negatif atau positif. Nilai polaritas mungkin antara -1 dan 1. Secara umum diterima bahwa sentimen tidak menguntungkan ketika nilai polaritas mendekati -1, dan sebaliknya. Nilai polaritas yang mendekati 0 dianggap netral.

### **2.3.2. Cyber Bullying**

Menurut (Watts dkk., 2017), definisi cyber bullying adalah pelecehan yang dilakukan kepada korban dengan maksud untuk menimbulkan kerugian melalui metode elektronik apa pun, termasuk media social. Pelecehan jenis ini dapat mengakibatkan pencemaran nama baik, pengungkapan detail pribadi kepada publik, dan tekanan emosional yang disengaja. Itu bisa merujuk pada tindakan mengirim atau memposting teks atau foto yang menyakitkan melalui media sosial atau bentuk perangkat komunikasi digital lainnya dengan maksud untuk membahayakan korban (Washington, 2015). Karena meningkatnya keterikatan remaja dengan media sosial dan perkembangan teknologi digital, cyber bullying menjadi semakin umum. Meningkatnya penggunaan gadget berteknologi canggih telah menyebabkan peningkatan prevalensi aplikasi media sosial di kalangan anak muda (Kaur & Saini, 2023).

Fenomena ini memiliki dampak positif dan negatif. Di satu sisi, ini memungkinkan berbagi posting, gambar, dan pesan secara pribadi dan rahasia di antara teman-teman, tetapi di sisi lain, ini berkontribusi pada peningkatan jumlah



cyber bullying yang terjadi sebagai akibat dari pembuatan akun penipuan (akun fake) di aplikasi (Ansary, 2020). Ini adalah pola perilaku yang dilakukan oleh individu dengan menggunakan media sosial, game, dan platform perpesanan dengan tujuan utama untuk menurunkan harga diri korban.

### **2.3.3. Preprocessing**

Menurut (Ignatow & Mihalcea, 2018) dalam analisis teks hampir selalu membutuhkan processing. Processing dapat terdiri dari langkah-langkah dasar seperti Tokenization, Stop Word Removal, Stemming dan Lemmatization.

#### **2.3.3.1. Tokenization**

Tokenization adalah proses mengidentifikasi kata-kata dalam urutan input karakter, terutama dengan memisahkan tanda baca tetapi juga dengan mengidentifikasi kontraksi, singkatan, dan sebagainya. Terkadang proses tokenisasi juga mencakup langkah-langkah normalisasi teks lainnya, seperti lowercase atau truecasing atau penghapusan tag HTML, jika teks diperoleh dari halaman web. Tokenization sebagian besar tidak bergantung pada bahasa, misalnya singkatan dan kontraksi seringkali bergantung pada bahasa, dan dengan demikian, seseorang perlu menyusun daftar kata-kata tersebut untuk memastikan bahwa tokenisasi ditangani dengan benar. Hal yang sama berlaku untuk apostrof dan tanda hubung. Perlu juga diperhatikan bahwa proses tokenization mengasumsikan bahwa spasi dan tanda baca digunakan sebagai batas kata yang eksplisit.

### 2.3.3.2. Stop Word Removal

Stop words juga disebut sebagai function words atau closed-class words, yang terdiri dari kata-kata berfrekuensi tinggi termasuk kata ganti, penentu, preposisi dan lain-lain. Proses ini merupakan proses dimana kata-kata yang tidak penting dalam dokumen teks akan dibuang.

### 2.3.3.3. Stemming

Banyak kata terkait dalam bahasa alami, namun memiliki bentuk yang berbeda sehingga membuat pengenalannya menjadi tidak penting. Sederhananya, stemming menerapkan seperangkat aturan ke kata input untuk menghapus sufiks dan prefiks dan mendapatkan induknya (akar kata), yang akan dibagikan dengan kata-kata terkait lainnya. Misalnya, computer, computational, dan computation semuanya akan direduksi menjadi akar yang sama yaitu compute. Stemming seringkali menghasilkan akar kata yang bukan kata-kata valid atau yang tidak relevan jika "konsumen" dari akar kata ini adalah sebuah sistem dan bukan manusia.

### 2.3.3.4. Lemmatization

Alternatif untuk stemming adalah lemmatization, yang mengurangi bentuk infleksi dari sebuah kata ke bentuk akarnya. Misalnya, lemmatization akan mengubah anak-anak menjadi anak, dan saya, sedang, atau akan menjadi. Berbeda dengan stemming, output yang diperoleh dari lemmatization adalah bentuk kata yang valid, yaitu bentuk dasar dari sebuah kata seperti yang terdapat dalam kamus. Dengan demikian, lemmatization memiliki manfaat agar keluarannya dapat dibaca oleh manusia, namun ini membutuhkan

proses komputasi yang lebih intensif, karena memerlukan daftar bentuk tata bahasa untuk menangani infleksi reguler serta daftar ekstensif kata tidak beraturan.

#### 2.3.4. FastText

Menurut (Bojanowski dkk., 2016), Facebook memiliki perpustakaan yang dikenal sebagai FastText, yang digunakan untuk membantu menghasilkan representasi kata yang efektif dan memberikan dukungan untuk klasifikasi teks. Menurut penelitian (Geet D'sa dkk., 2020), FastText biasanya digunakan untuk mengatasi masalah dengan kategorisasi kalimat dan pendekatan representasi kata untuk membuat perangkat lunak lebih efektif dan lebih cepat daripada Word2vec dan Glove. Menurut (Bojanowski dkk., 2016) dalam penelitiannya, metode FastText direpresentasikan sebagaimana fungsi di bawah.

$$\sum_{t=1}^T \left[ \sum_{c \in C_c} \ell(s(w_t, w_c)) + \sum_{n \in \mathcal{N}_{t,c}} \ell(-s(w_t, n)) \right] \quad (1)$$

Keterangan:

s = scoring function

w = weight

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

n = jumlah kosa kata (vocabulary)

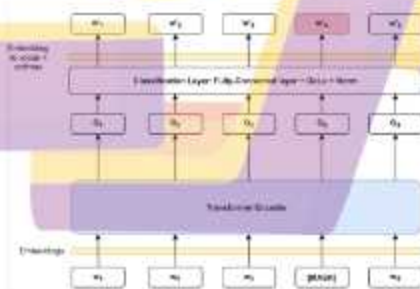
#### 2.3.5. Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (Devlin dkk., 2018) atau disingkat BERT adalah model representasi bahasa terlatih yang dikembangkan oleh para peneliti di Google AI Language pada tahun 2018.



BERT dikembangkan berdasarkan teknik-teknik deep learning dan berbagai metode seperti semisupervised learning, ELMo, ULMFiT, OpenAI Transformers, dan Transformers. Sesuai dengan namanya, BERT menggunakan Transformer yang merupakan sebuah mekanisme yang mempelajari hubungan kontekstual antara kata-kata dalam teks (Vaswani dkk., t.t.).

Arsitektur BERT diilustrasikan pada Gambar 2.1 (Roy & Ojha, 2020). Komponen yang bertanggung jawab untuk decoding adalah setumpuk decoder yang dihitung secara identik. Encoder semuanya memiliki struktur yang mirip satu sama lain. Masing-masing dari mereka dibagi menjadi dua sub-lapisan: input pembuat encode pertama-tama melewati lapisan self-attention, yang merupakan lapisan yang memungkinkan pembuat encode untuk melihat istilah lain dalam ekspresi input saat sedang menyandikan kata tertentu. Hasil dari lapisan self-attention digunakan untuk diumpungkan ke dalam jaringan feedforward sebagai input jaringan.



Gambar 2.1. Arsitektur BERT (Roy & Ojha, 2020)

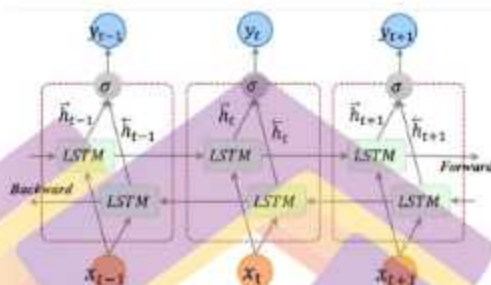
### 2.3.6. Bidirectional LSTM (Bi-LSTM)

LSTM (Long Short-Term Memory) adalah bentuk inovatif dari arsitektur jaringan berulang yang, ketika dipasangkan dengan algoritma pembelajaran berbasis gradien yang sesuai dan menghasilkan hasil yang sangat baik. LSTM dikembangkan secara khusus untuk mengatasi masalah yang terkait dengan aliran balik kesalahan. Bahkan di hadapan urutan input yang bising dan tidak dapat dimampatkan, ia mampu belajar untuk menjembatani rentang waktu lebih dari 1000 langkah tanpa mengalami pengurangan karena kemampuan jeda waktunya yang kecil (Hochreiter & Schmidhuber, 1997). Kesulitan mengelola teks dalam long-term memory dan short-term memory dipecah menjadi dua submasalah oleh LSTM, yang pertama adalah menghapus informasi dari konteks yang tidak lagi diperlukan, dan yang kedua adalah menambahkan informasi yang mungkin diperlukan untuk pengambilan keputusan selanjutnya (Jurafsky & Martin, t.t.).

Menurut (Graves & Schmidhuber, t.t.), Bi-LSTM merupakan pengembangan yang dibuat dengan menggabungkan LSTM dengan lapisan Bidirectional. Menurut penelitian (Z. Chen dkk., 2020), Bi-LSTM dibuat dengan menggabungkan dua LSTM, yaitu LSTM mundur dan LSTM maju. Kombinasi keduanya akan mengumpulkan data dari dua arah yang berbeda secara bersamaan. Karena dapat diolah secara berurutan dan setiap kata dapat dipahami secara berurutan, pola ini cukup untuk mengenali struktur dalam sebuah frase.

Gambar 2.2 (Cui dkk, 2017) menggambarkan arsitektur Bi-LSTM, di mana output unit tersembunyi dari unit ht di lapisan mundur dan maju digabungkan

untuk menghasilkan nilai fitur kata, yang lebih besar dari ukuran fitur yang digunakan oleh LSTM standar dan karenanya lebih akurat mengklasifikasikan kata.



Gambar 2.2. Arsitektur Bi-LSTM (Cui dkk., 2017)

### 2.3.7. Confusion Matrix

Confusion Matrix adalah matriks yang menunjukkan seberapa salah atau benar suatu model diprediksi. Confusion Matrix dapat digunakan untuk menguji kinerja suatu model klasifikasi. Dalam bentuknya yang paling sederhana, confusion matrix memiliki tabel 2x2 untuk mengklasifikasikan model dengan pengamatan sebagai A atau B (Cielen dkk., t.t.). Tabel 2.2 merupakan rumus Confusion Matrix (Markoulidakis dkk. 2021).

Tabel 2. 2. Tabel Confusion Matrix

Actual Class	Predicted Class	
	Positive	Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

Ada beberapa teknik evaluasi yang digunakan untuk mengevaluasi seberapa baik kinerja klasifikasi pada data uji yang tidak terlihat. Ukuran yang paling umum digunakan dalam klasifikasi teks adalah precision, recall, F-measure, dan accuracy. Seringkali, tujuannya adalah untuk memaksimalkan semua ukuran, yang berkisar dari 0 hingga 1. Oleh karena itu, nilai yang lebih tinggi sesuai dengan kinerja klasifikasi yang lebih baik.

Precision dan recall adalah dua statistik dalam kategorisasi teks yang sering digabungkan untuk menilai efektivitas pencarian informasi. Lebih khusus lagi, recall menghitung jumlah dokumen relevan yang berhasil diambil, sedangkan akurasi menghitung jumlah dokumen relevan yang benar-benar diambil. Rumus yang terdapat pada persamaan (2) dan (3) dapat digunakan untuk menghitung kedua metrik tersebut.

$$\text{Precision} = TP / (TP + FP) \quad (2)$$

$$\text{Recall} = TP / (TP + FN) \quad (3)$$

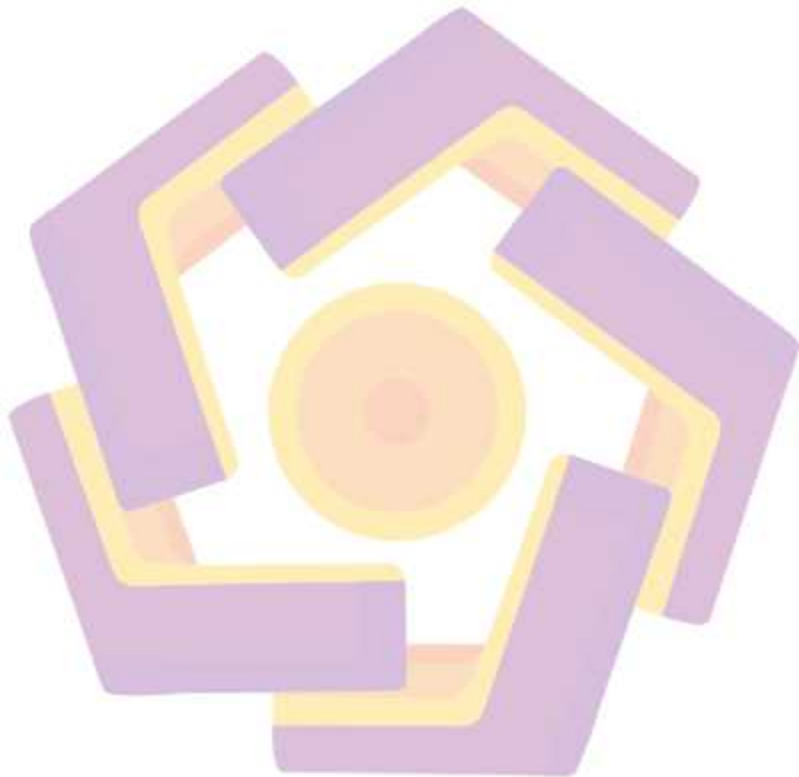
Precision dan recall tidak sering diperhitungkan sendirian. Kedua ukuran ini sering digunakan bersama dalam F-measure, yang menyediakan metrik berbobot tunggal untuk mengevaluasi kinerja secara keseluruhan. F-measure dapat dihitung menggunakan rumus seperti pada persamaan (4).

$$\text{F-measure} = 2 \times (\text{precision} \times \text{recall}) / (\text{precision} + \text{recall}) \quad (4)$$

Metrik lain yang digunakan untuk menilai kinerja kategorisasi adalah akurasi. Jumlah sampel yang diidentifikasi dengan benar digunakan untuk menghitung akurasi. Akurasi digunakan ketika klasifikasi secara konsisten memprediksi satu

kelas, yang dapat ditentukan dengan menggunakan rumus seperti persamaan (5) (Williams t.t.).

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FN+FP+TN)} \times 100 \quad (5)$$





## BAB III

### METODE PENELITIAN

#### 3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian yang akan dilakukan adalah penelitian eksperimen, dimana penelitian yang dilakukan menerapkan serangkaian tindakan guna membuktikan sebuah konsep mengkombinasikan metode word embedding dengan Bi-LSTM. Word embedding yang akan dikombinasikan dengan Bi-LSTM adalah FastText dan BERT. Studi kasus yang diambil merupakan studi kasus cyber bullying yang datasetnya didapat melalui data publik dari Mendeley Data dengan keyword cyber bullying.

Sifat dari penelitian ini adalah deskriptif yang dimana penelitian ini akan menjelaskan dampak penggunaan metode word embedding terhadap performa Bi-LSTM untuk analisis sentimen cyber bullying. Apakah peningkatan performa signifikan atau tidak, dan bagaimana dampak tersebut dapat diinterpretasikan.

Pendekatan pada penelitian ini adalah pendekatan kuantitatif yang dimana penelitian ini akan menghitung akurasi dan performa pada algoritma kombinasi FastText+Bi-LSTM dan BERT+Bi-LSTM dalam klasifikasi sentimen cyber bullying. Data yang diambil merupakan dataset berkaitan cyber bullying pada komentar platform youtube yang didapat dari Mendeley Data. Setelah didapatkan, dilakukan pembagian pembersihan data dengan preprocessing dan dilakukan pembagian dataset seperti data latih dan data uji. Kemudian data tersebut

digunakan sebagai dasar untuk eksperimen model menentukan klasifikasi sentimen.

### 3.2. Metode Pengumpulan Data

Data yang digunakan dalam penelitian ini yaitu terkait tentang komentar cyber bullying. Data ini dapat diakses secara publik di Mendeley Data dan berformat .csv. Gambar 3.1 merupakan alur pengumpulan data yang dilakukan untuk penelitian ini.



Gambar 3. 1. Alur Pengumpulan Data

Langkah pertama dalam pengumpulan data melibatkan pencarian berdasarkan kata kunci "cyber bullying" di basis data Mendeley. Setelah melakukan pencarian, identifikasi dataset yang paling relevan dengan fokus penelitian yang akan dilakukan. Selanjutnya unduh dataset dari Mendeley Data dan melakukan eksplorasi awal terhadap data dengan memeriksa struktur data, tipe variabel, label pada data dan bagaimana informasi mengenai komentar-komentar cyber bullying diwakili dalam dataset.

### 3.3. Metode Analisis Data

Dataset yang digunakan akan dibagi menjadi 2 jenis data untuk keperluan penelitian ini, yakni data latih dan data uji. Data Latih digunakan untuk melatih algoritma dalam mencari model yang sesuai. Sedangkan Data Uji merupakan data baru yang belum memiliki kelas sehingga diperlukan proses klasifikasi untuk menentukan kelas yang cocok. Terdapat 2 label yang ditetapkan yakni 0 dan 1. Data yang didapat dan dikumpulkan akan di olah menggunakan text preprocessing agar dapat diproses pada tahap selanjutnya hingga mendapatkan hasil akurasi terhadap model yang digunakan. Adapun tahapan yang akan di gunakan untuk pengolahan data adalah sebagai berikut.

#### 3.3.1. Datasets

Datasets yang digunakan dalam penelitian ini bersumber dari Mendeley Data. Mendeley Data adalah platform yang disediakan oleh Mendeley untuk membantu peneliti dalam mengelola, menyimpan, dan berbagi data penelitian mereka. Mendeley Data juga menyediakan alat untuk mengorganisir, menganalisis, dan berbagi data dengan rekan penelitian, kolaborator, dan masyarakat ilmiah secara umum. Platform ini membantu mendukung transparansi dan reproduktibilitas dalam penelitian dengan menyediakan akses terbuka terhadap data penelitian.

Penelitian ini menggunakan data "Bahasa cyber bullying" yang di publish di Mendeley Data pada tanggal 29 Agustus 2023 yang diterbitkan dengan nomor

DOI 10.17632/x6pg5wps8j.1. Dataset yang digunakan dalam penelitian merupakan data dari komentar pada platform youtube yang dikumpulkan oleh (Yudi Setiawan, 2023) dalam bahasa indonesia dan dikumpulkan selama rentang waktu 2022-2023. Dataset ini memiliki 65.343 baris data dan telah memiliki label, yakni label 0 (non-cyber bullying) dan label 1 (cyber bullying). Total keseluruhan data yang akan digunakan dalam penelitian ini berjumlah 65.343 data yang selanjutnya akan dilakukan proses preprocessing terlebih dahulu sebelum digunakan untuk melatih model. Jumlah data dari hasil preprocessing nantinya yang akan digunakan untuk eksperimen implementasi model FastText+Bi-LSTM dan BERT+Bi-LSTM.

### 3.3.2. Proses Data Cleaning

Pada tahap ini, data akan diperiksa untuk mengidentifikasi dan menangani berbagai masalah seperti nilai yang hilang, outlier, duplikat, dan format yang tidak sesuai. Proses ini bertujuan untuk memastikan bahwa data yang digunakan dalam analisis atau model memiliki kualitas yang baik, sehingga hasilnya dapat diandalkan dan representatif. Data cleaning adalah langkah penting dalam menyiapkan data untuk analisis lebih lanjut, sehingga keputusan yang diambil berdasarkan data tersebut dapat lebih akurat dan relevan. Sebelum melakukan data cleaning, proses case folding diterapkan pada data. Case folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil atau huruf kapital. Proses case folding dilakukan menggunakan baris kode pada gambar 3.2.



Gambar 3.2 merupakan baris kode untuk proses case folding, yang merupakan proses mengubah semua huruf kapital menjadi huruf kecil. Case folding dilakukan dengan menggunakan metode 'str.lower()' pada kolom teks ('Text') dari 'dataframe df\_tweets'. Hasil dari proses case folding kemudian disimpan kembali dalam dataframe 'df\_preprocessing' dengan nama kolom 'casefolding'.

```
# Do preprocessing casefolding = change capital letters to lowercase
df_tweets['Text'] = df_tweets['Text'].str.lower()
df_preprocessing['casefolding'] = df_tweets['Text']

print('Case Folding Result : \n')
print(df_tweets['Text'].head(10))
```

Gambar 3.2. Proses Case Folding

Selanjutnya akan dilakukan proses data cleaning menggunakan baris kode sesuai pada gambar 3.3. Tahapan pembersihan menggunakan baris kode pada gambar 3.3 dimulai dengan menambahkan spasi setelah titik atau koma untuk memisahkan kata-kata yang berdekatan dengan tanda baca tersebut. Kemudian, karakter non-ascii dihapus dari string, dan dua atau lebih titik diubah menjadi satu spasi. Baris baru dihapus, dan selanjutnya, hashtag dihapus dari teks, karakter huruf tunggal dihapus, angka dihapus, serta URL dihapus dari teks untuk membersihkan teks dari elemen-elemen tersebut. Setelah itu, spasi dihapus dari awal dan akhir teks, dan spasi yang berlebihan diubah menjadi satu spasi. Tanda baca dihapus dari teks menggunakan fungsi 'str.translate()'. Terakhir, URL yang tidak lengkap dihapus dari teks. Setelah proses pembersihan selesai, teks yang telah dibersihkan disimpan kembali dalam kolom 'Text'.





### 3.3.3. Proses Tokenization

Dalam proses ini, data akan dijadikan menjadi token atau dipisahkan per-kata. Tokenisasi juga mencakup langkah-langkah normalisasi teks lainnya, seperti lowercase, simbol simbol dan karakter, spasi, dan penghapusan tag HTML/link jika teks diperoleh dari halaman web. Proses tokenization dilakukan menggunakan baris kode pada gambar 3.5. Dalam fungsi pada gambar 3.5, tokenisasi dilakukan menggunakan library Natural Language Toolkit (NLTK), yang membagi teks menjadi token-token kata.

```
# Proses Tokenizing
def proses_tokenizing(data_ulasan):
    return word_tokenize(data_ulasan)

df_tweets['Text'] = df_tweets['Text'].astype(str).apply(proses_tokenizing)
df_preprocessing['tokenizing'] = df_tweets['Text']

print('Tokenizing Result : \n')
print(df_tweets['Text'].head(5))
```

Gambar 3. 5. Proses Tokenization

### 3.3.4. Stopword Removal

Dalam proses ini, kata kata yang dianggap tidak berguna (delimiter) akan dibuang karena kedepannya akan mempengaruhi kinerja komputer dalam mengenal kata. Stopwords adalah kata-kata umum yang sering muncul dalam bahasa tetapi cenderung tidak memberikan informasi yang signifikan terkait dengan sentimen atau konten esensial dalam konteks analisis cyber bullying. Gambar 3.6 merupakan baris kode yang digunakan untuk memproses stopwords removal. Kata-kata lain yang dianggap delimiter dapat ditambahkan manual pada baris kode jika dibutuhkan.



### 3.3.6. Proses Lemmatization

Pada proses ini data akan diolah komputer menjadi kata dasar atau kata akar. Stemming seringkali menghasilkan akar kata yang bukan kata-kata valid atau yang tidak relevan jika “konsumen” dari akar kata ini adalah sebuah sistem dan bukan manusia. Alternatif untuk stemming adalah lemmatization, yang mengurangi bentuk infleksi dari sebuah kata ke bentuk akarnya. Misalnya, lemmatization akan mengubah anak-anak menjadi anak, dan saya, sedang, atau akan menjadi. Berbeda dengan stemming, output yang diperoleh dari lemmatization adalah bentuk kata yang valid, yaitu bentuk dasar dari sebuah kata seperti yang terdapat dalam kamus. Dengan demikian, lemmatization memiliki manfaat agar keluarannya dapat dibaca oleh manusia. Baris kode yang digunakan dalam proses lemmatization dapat dilihat pada gambar 3.8.

```
# Proses Lemmatization
lemmatizer = Lemmatizer()
def proses_lemmatization(data_dikawat):
    return lemmatizer.lemmatize(data_dikawat)

df_tweets['text'] = df_tweets['text'].astype(str).apply(proses_lemmatization)
df_preprocessing['lemmatization'] = df_tweets['text']

print("Lemmatization Result : \n")
print(df_tweets['text'].head(5))
```

Gambar 3. 8. Proses lemmatization

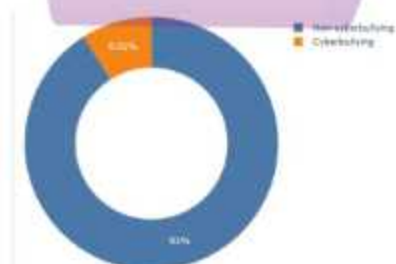
### 3.3.7. Exploratory Data Analysis (EDA)

Proses Exploratory Data Analysis (EDA) adalah langkah analisis data yang bertujuan untuk memahami karakteristik dan pola yang mungkin terdapat dalam dataset. Tujuan utama EDA adalah menggali wawasan (insight) dan pemahaman yang lebih dalam tentang data cyber bullying sebelum dilakukan

analisis statistik formal atau pembuatan model. Selanjutnya, gambar 3.9 dan 3.10 di bawah merupakan grafik yang memvisualisasikan jumlah data berdasarkan label. Setiap batang dalam grafik mewakili jumlah sampel data untuk setiap label. Dapat dilihat pada gambar 3.9 bahwa kedua kategori memiliki jumlah data yang tidak seimbang, yaitu masing-masing 59.451 untuk data berlabel 0 (Non-cyber bullying) dan 5.892 untuk data berlabel 1 (cyber bullying). Sementara pada gambar 3.10 menunjukkan visualisasi persentase data berdasarkan jumlah label nya. Diketahui sebanyak 91% data mentah memiliki label 0 (non-cyber bullying), sementara sisanya 9% memiliki label 1 (cyber bullying).



Gambar 3. 9. Grafik total jumlah data

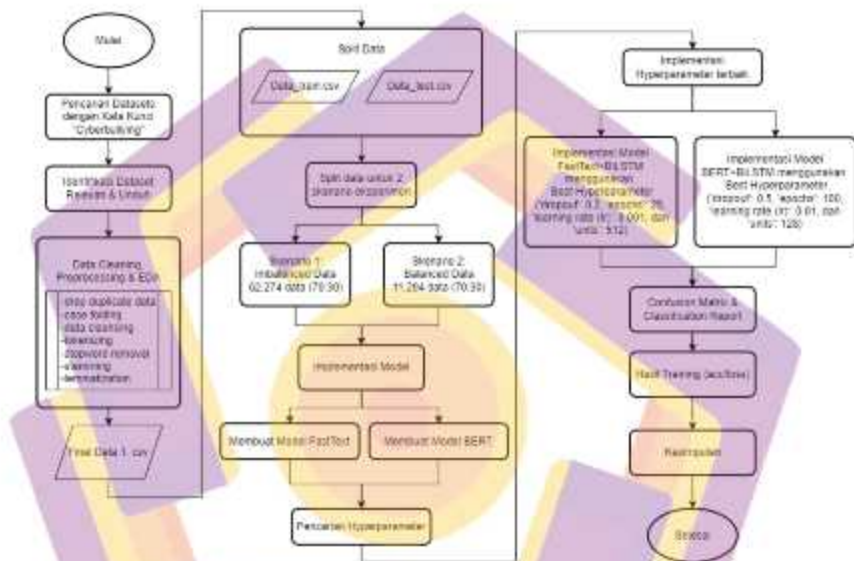


Gambar 3. 10. Grafik persentase data berdasarkan label



### 3.4. Alur Penelitian

Alur Penelitian dalam penelitian ini mencakup mulai dari bagaimana teknik pengumpulan data hingga aspek perancangan implementasi dan pengujian dalam penelitian ini dapat dilihat pada gambar 3.11.



Gambar 3.11. Alur Penelitian

Gambar 3.11 merupakan langkah demi langkah untuk melakukan penelitian ini. Proses dimulai dengan mencari dataset yang relevan menggunakan kata kunci "cyber bullying" dan mengunduh dataset yang relevan setelah mengidentifikasinya. Langkah berikutnya adalah pembersihan dan preprocessing data, yang melibatkan penghapusan data duplikat, mengubah semua teks menjadi huruf kecil (case folding), membersihkan data, tokenisasi teks menjadi kata-kata individu, menghapus stopwords, dan melakukan stemming atau lemmatization

untuk mengurangi kata-kata menjadi bentuk akarnya. Data yang sudah diproses kemudian disimpan sebagai file CSV. Pada langkah selanjutnya, data dibagi menjadi set pelatihan dan pengujian. Dua model kemudian dibangun yakni FastText dan BERT, yang keduanya diimplementasikan dalam kombinasi dengan metode Bidirectional Long Short-Term Memory (Bi-LSTM). Hyperparameter tuning digunakan untuk setiap model mengoptimalkan kinerja. Model yang terlatih kemudian digunakan untuk memprediksi label data uji, dan confusion matrix dihasilkan untuk mengevaluasi kinerja model. Terakhir akan ditarik kesimpulan terkait dari performa dan kinerja model yang diusulkan.

Adapun hardware dan software yang dibutuhkan untuk mengelola data dapat dilihat pada tabel berikut 3.1.

Tabel 3. 1. Spesifikasi Hardware dan Software

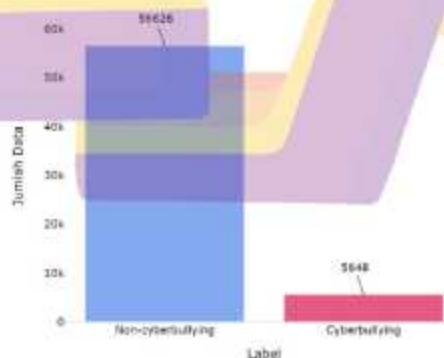
No	Nama Software/Hardware	Versi/Kapasitas
1	Akun Google Colab & Google Drive	l***@students.amikom.ac.id
2	Google Colab Pro 1.0.0	Runtime Type A100 GPU
3	Python	Python Version 3.11.0
4	Library/Package	<ul style="list-style-type: none"> <li>• sastrawi 1.0.1</li> <li>• nlp-id 0.1.5</li> <li>• tensorflow 2.9.0</li> <li>• transformers 4.35.2</li> <li>• huggingface 0.20.3</li> <li>• pandas 1.5.3</li> <li>• numpy 1.25.2</li> <li>• NLTK 3.8.1</li> <li>• regex 2023.12.25</li> <li>• gensim 4.3.2</li> <li>• sklearn 1.2.2</li> <li>• hyperopt 0.2.7</li> </ul>
5	Laptop Asus	<ul style="list-style-type: none"> <li>• Model: Vivobook_ASUS Laptop X415JAB_A416JA</li> <li>• Processor: Intel Core i3-1005G1</li> <li>• Memory: 4GB</li> <li>• Harddisk: 500GB</li> </ul>

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

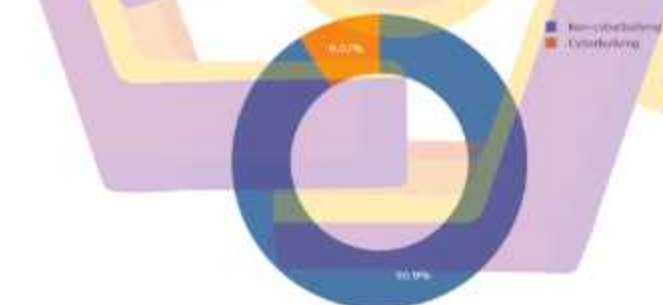
#### 4.1. Hasil Data Cleaning dan Preprocessing

Adapun data yang digunakan dalam penelitian ini adalah data komentar youtube dengan total 65.343 data. Sebanyak 65.343 data dari total keseluruhan data dilakukan proses preprocessing. Bahasa pemrograman Python beserta sejumlah library digunakan untuk menjalankan proses preprocessing melalui baris kode menggunakan platform Google Colab Pro. Setelah dilakukan berbagai tahapan preprocessing yang tersisa menjadi sebanyak 62.274 data yang disimpan dalam bentuk File Comma Separated Values (CSV) dengan nama 'Final Data 1.csv'. Gambar 4.1 memvisualisasikan hasil akhir pengolahan data preprocessing, yakni warna merah untuk 5.648 data berlabel 1 (cyber bullying) dan warna biru untuk 56.626 data berlabel 0 (non-cyber bullying).



Gambar 4. 1. Grafik Total Data

Persentase kelas divisualisasikan melalui diagram bulat yang ditunjukkan pada Gambar 4.2. Diagram ini menggambarkan distribusi label cyber bullying dan non-cyber bullying di platform YouTube. Non-cyber bullying sentimen direpresentasikan menggunakan warna biru, cyber bullying sentimen direpresentasikan menggunakan warna kuning. Dengan 90,9% dari konten yang disematkan dengan label non-cyber bullying dan 9,07% dengan label cyber bullying, memberikan wawasan penting tentang prevalensi dan jenis konten yang ada di platform tersebut. Konten non-cyber bullying, yang mencakup mayoritas dengan persentase 90,9%, mungkin meliputi berbagai topik konten pada platform youtube. Dengan jumlah yang signifikan, konten-konten ini memberikan nilai tambah bagi pengguna dengan memberikan informasi, hiburan, atau wawasan yang relevan tanpa menimbulkan risiko cyber bullying atau perasaan tidak nyaman.



Gambar 4. 2. Grafik Presentase Data

Di sisi lain, label cyber bullying menyumbang 9,07% dari keseluruhan data yang mencakup komentar yang memiliki niat atau dampak negatif terhadap

individu atau kelompok tertentu, yang berupa intimidasi, pelecehan verbal, atau penyebaran informasi palsu atau merugikan tentang individu atau kelompok. Dengan adanya persentase yang cukup signifikan, keberadaan konten-konten ini menyoroti pentingnya tindakan pencegahan dan penanganan cyber bullying di platform-platform seperti YouTube.

#### **4.2. Skenario Eksperimen**

Pada bagian hasil data cleaning dan preprocessing disebutkan bahwa dataset final yang akan diproses terdiri dari 62.274 data dengan komposisi 5.648 data dengan label 1 (cyberbullying) dan 56.626 berlabel 0 (non-cyberbullying). Hal ini mendeskripsikan bahwa adanya ketidakseimbangan pada data cyberbullying yang digunakan. Ketidakseimbangan data menurut (Tanha dkk., 2020) akan menyebabkan pengaruh pada hasil klasifikasi serta akan menghasilkan hasil yang baik pada kelas mayor dan mengabaikan kelas minor. Penanganan ketidakseimbangan data dapat dilakukan menggunakan teknik Random Undersampling (Madabushi dkk., 2020), di mana sebagian sampel dari kelas mayoritas dihapus secara acak sehingga jumlahnya sebanding dengan kelas minoritas. Selain itu (Kulkarni dkk., 2021) mengungkapkan bahwa penggunaan Random Undersampling salah satu metode paling sederhana dan umumnya digunakan sebagai metode dasar. Penangan imbalanced data dalam penelitian ini diterapkan menggunakan fungsi '*np.random.randint*' yang tersedia dalam pustaka NumPy.



Pada penelitian ini, peneliti akan menggunakan dua skenario pembagian data atau pengujian. Skenario pertama yang dilakukan adalah menggunakan data keseluruhan dan merupakan data tak seimbang (imbalanced) sebanyak 62.274 data, dengan komposisi 5.648 data dengan label 1 (cyberbullying) dan 56.626 berlabel 0 (non-cyberbullying). Sedangkan pada skenario kedua akan dilakukan menggunakan data seimbang (balanced) sebanyak 11.294 data, dengan komposisi 5.647 data berlabel 0 (non-cyberbullying) dan 5.674 data berlabel 1 (cyberbullying). Proporsi pembagian data latih dan data uji dalam dua skenario yang dilakukan dapat dilihat pada tabel 4.1 berikut.

Tabel 4. 1. Skenario Eksperimen

Opsi	Dataset	Jumlah Datasets	Persentase data latih	Persentase data uji
Skenario 1	Imbalanced	62.274	70%	30%
Skenario 2	Balanced	11.294	70%	30%

#### 4.3. Eksperimen dan Evaluasi Kombinasi FastText dan Bi-LSTM

Model FastText yang digunakan dalam penelitian ini merupakan jenis FastText self-train. FastText Self-Train mengacu pada proses pelatihan model FastText menggunakan data yang dimiliki secara langsung. Dalam konteks penelitian ini, FastText Self-Train berarti bahwa model FastText dilatih menggunakan dataset cyber bullying dari platform twitter yang didapat secara langsung, tanpa menggunakan model yang sudah dilatih sebelumnya. Proses pelatihan FastText Self-Train melibatkan pembuatan vektor embedding kata-kata

berdasarkan distribusi teks dalam dataset yang diberikan agar model dapat memahami hubungan antara kata-kata dalam konteks spesifik dataset tersebut.

Proses pembuatan model FastText menggunakan library NLTK (Natural Language Toolkit) untuk tokenisasi kata dan library Gensim untuk membangun model FastText. Pada baris kode dalam gambar 4.3, NLTK digunakan untuk melakukan tokenisasi kata menggunakan fungsi 'word\_tokenize()', sementara Gensim digunakan untuk membangun model FastText menggunakan fungsi 'FastText()'. FastText menggunakan library Gensim untuk membentuk model FastText dengan parameter-parameter yang telah ditentukan, di mana model tersebut diberi input berupa token-token kata dari teks yang telah diolah sebelumnya, dengan vektor embedding kata berukuran 100 untuk merepresentasikan makna kata, window size sepanjang 5 kata untuk memperhitungkan hubungan antarkata, serta memasukkan setiap kata yang muncul minimal sekali dalam proses pembentukan model, dengan menggunakan 4 workers untuk mempercepat pelatihan, dan menerapkan metode skip-gram untuk memprediksi kata-kata di sekitar kata target.

```
# Mengubah data dalam text menjadi bentuk list
sentences = [word_tokenize(text) for text in all_data['text'].tolist()]

# Membangun model FastText
fasttext_model = FastText(sentences=sentences, vector_size=100, window=5, min_count=1, workers=4, sg=1)
```

Gambar 4. 3. Baris Kode Model FastText

Gambar 4.4 menunjukkan contoh kata yang telah mengalami vektorisasi, secara khusus menunjukkan representasi sebagian hasil vektorization untuk istilah "asyik". Vektorisasi pasti bersifat unik sehingga setiap hasil vektorisasi sebuah kata tidak akan pernah sama dengan hasil vektorisasi kata yang lainnya.

```

Row 60 - Text: anyik
Vektor for row 60: [-0.40570197 -0.11424825 0.4860738 0.3095311 0.30714587 0.00052573
0.87873006 0.22723943 0.22069205 -0.137643 -0.18234721 0.2393887
-0.35251438 0.83311093 -0.77024905 0.21459537 0.7673176 0.87800674
0.25986185 -0.8979755 -0.10255885 0.13817306 -0.41984524 0.28818895
-0.11881454 0.5358972 -0.20386404 0.8368305 -0.40582524 0.81188685
0.8511427 -0.11443844 0.38895025 0.8752232 0.22877882 -0.8526468
0.7802382 0.80702932 -0.01761955 -0.12614254 -0.10481339 0.36939204
-0.9790893 0.4822496 0.86483912 -0.39019397 -1.4172379 -0.18129531
-0.5312403 -0.45047846 -0.84177418 -0.1296588 -0.3480371 0.14057833
-0.27811524 -0.16189171 0.15434278 0.8169527 0.39162818 0.21376833
1.1282584 0.43657818 -0.61870701 0.19211715 -0.38232347 0.46929988
0.19980806 -0.121424 0.82577612 0.19365456 0.99928593 0.34065115
-0.61661106 -0.5135853 0.58718723 -0.81898884 -0.38001874 0.89225609
0.42231953 -0.21792810 0.342948 -0.14928632 0.87906049 0.4654026
-0.738361 0.86845822 0.4892578 0.8996108 -0.13717014 0.8758241
-0.24556591 1.1916498 0.44481277 0.854044 -0.39640517 0.89172580
-0.21751874 0.44189617 -0.26179018 0.1793886 ]

```

Gambar 4. 4. Hasil Vektorisasi Kata FastText

Pada tahap selanjutnya eksperimen kombinasi FastText dengan Bi-LSTM dimulai dengan melatih model data FastText yang telah dibuat terlebih dahulu menggunakan hyperparameter. Hyperparameter tuning digunakan pada tahap ini untuk mencari kombinasi nilai hyperparameter optimal untuk model guna mencapai performa terbaik. Proses pencarian hyperparameter terbaik menggunakan algoritma optimasi yakni Hyperopt. Fungsi baris kode 'tuningFastTextBI-LSTM' pada gambar 4.5 bertujuan untuk melakukan penyetelan hyperparameter pada model yang menggabungkan FastText dengan lapisan Bi-LSTM. Pertama, model inisial dibangun menggunakan lapisan embedding yang menerima vektor embedding dari model FastText, dengan input\_length yang disesuaikan dengan dimensi dari data teks. Kemudian, ditambahkan lapisan Bi-LSTM dengan jumlah unit yang ditentukan oleh parameter yang diberikan. Selanjutnya, diterapkan lapisan dropout untuk mengurangi overfitting, dan ditambahkan lapisan output dengan fungsi aktivasi softmax untuk klasifikasi. Model kemudian dicompiled dengan fungsi loss

categorical\_crossentropy dan optimizer RMSprop dengan learning rate yang ditentukan oleh parameter. Selanjutnya, model dilatih dengan data pelatihan dan dievaluasi dengan data pengujian selama jumlah epoch yang telah ditentukan. Hasil akurasi terbaik dari data validasi kemudian dicatat untuk dijadikan metrik evaluasi.

```
def tuning_keras_lstm(param, *args, **kwargs):
    # Inicial model
    model = tf.keras.models.Sequential()
    # Embedding layer
    model.add(tf.keras.layers.Embedding(vocab_size, feature_model_embed_size, weights='embedding_matrix',
                                       input_length=vocab.FeatureEmbed(1), trainable=False))

    # Hidden layer
    model.add(tf.keras.layers.LSTM(units=param['units'], activation='tanh', recurrent_dropout=param['dropout']))
    # Output layer
    model.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer=tf.keras.optimizers.Adam(learning_rate=param['lr']), metrics=['accuracy'])
    # Summary model
    model.summary()

    # Training model
    history = model.fit(x_train, y_train, epochs=param['epochs'], batch_size=32,
                       validation_data=(x_test, y_test), callbacks=[tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)])

    acc = history.history['val_accuracy'][-1]

    return (loss, acc, param['units'], param['lr'], param['epochs'])
```

Gambar 4. 5. Baris Kode Proses Hyperparameter Tuning FastText+Bi-LSTM

Selanjutnya, gambar 4.6 merupakan baris kode yang mendefinisikan ruang pencarian hyperparameter terbaik otomatis untuk model, berbeda dengan penelitian sebelumnya (Nurdin dkk., 2020) yang tidak menggunakan hyperparameter melainkan menggunakan parameter yang diinputkan. Variabel-variabel yang akan dioptimasi meliputi units, lr (learning rate), dropout, dan epochs. Pertama, jumlah unit dalam lapisan LSTM yang akan dipilih dari opsi 128, 256, atau 512. Selanjutnya, lr adalah nilai learning rate yang akan dipilih dari opsi 0.0001, 0.001, atau 0.01, yang mempengaruhi seberapa cepat atau lambat model belajar selama pelatihan. Dropout yang akan dipilih dari opsi 0.2, 0.3, atau 0.5, yang digunakan untuk mengurangi overfitting dengan secara acak mengabaikan sebagian unit



selama pelatihan. Terakhir, jumlah epoch yang akan dipilih dari opsi 25, 50, atau 100, yang merupakan jumlah iterasi pelatihan yang akan dilakukan. Objek 'search\_space' mendefinisikan ruang pencarian hyperparameter menggunakan objek dari modul hyperopt yang mencakup hyperparameter seperti 'units', 'lr', 'dropout', dan 'epochs'.

```
# Definiskan ruang pencarian hyperparameter
search_space = (
    "X_train_fasttext": ("X_train_fasttext"),
    "units": hp.choice("units", [128, 256, 512]),
    "lr": hp.choice("lr", [0.0001, 0.001, 0.01]),
    "dropout": hp.choice("dropout", [0.2, 0.3, 0.5]),
    "epochs": hp.choice("epochs", [25, 50, 100])
)

# Lakukan hyperparameter tuning
trials = Trials()
best = fmin(tuningFastTextBiLSTM, search_space, algo=tpe.suggest, max_evals=1, trials=trials)

# Cetak hyperparameter terbaik
print("Best hyperparameters:")
best_params_fasttext_BiLSTM = space_eval(search_space, best)
print(best_params_fasttext_BiLSTM)
```

Gambar 4. 6. Baris Kode Penentuan Parameter FastText+Bi-LSTM

Hasil dari proses pencarian hyperparameter terbaik ditampilkan dalam gambar 4.7 yang menunjukkan bahwa pencarian hyperparameter telah selesai 100%, dengan satu percobaan (trial). Proses tersebut memakan waktu 61.53 detik per-percobaan dan memiliki nilai loss sebesar -0.924369752407074.. Hasil terbaik ditunjukkan dalam output yakni dengan parameter nilai dropout 0.2, epochs 25, lr 0.001, dan units 512.

```
200/241 [-----] 100% | ETA: 0s | loss: 0.0001 | accuracy: 0.9244
250/241 [-----] 100% | ETA: 0s | loss: 0.0007 | accuracy: 0.9175
302/241 [-----] 100% | ETA: 0s | loss: 0.0000 | accuracy: 0.9274
380/241 [-----] 100% | ETA: 0s | loss: 0.0001 | accuracy: 0.9271
418/241 [-----] 100% | ETA: 0s | loss: 0.0008 | accuracy: 0.9219
491/241 [-----] 100% | ETA: 0s | loss: 0.0011 | accuracy: 0.9288
491/241 [-----] 100% | ETA: 0s | loss: 0.0012 | accuracy: 0.9244
512/241 [-----] 100% | ETA: 0s | loss: 0.0007 | accuracy: 0.9272
615/241 [-----] 100% | ETA: 0s | loss: 0.0010 | accuracy: 0.9279
647/241 [-----] 100% | ETA: 0s | loss: 0.0017 | accuracy: 0.9278
-----
2000 [██████████] 1/1 [100%] (100000, 61.53) (1/1), best loss: -0.924369752407074
Best hyperparameters:
{'X_train_fasttext': 'X_train_fasttext', 'dropout': 0.2, 'epochs': 25, 'lr': 0.001, 'units': 512}
```

Gambar 4. 7. Output Hasil Hyperparameter FastText+Bi-LSTM



Selanjutnya parameter terbaik diterapkan pada kombinasi FastText dengan Bi-LSTM. Gambar 4.8 menunjukkan baris kode yang digunakan untuk membangun arsitektur model FastText-Bi-LSTM. Dimulai dengan inisialisasi model sebagai 'Sequential()', diikuti dengan penambahan lapisan Embedding yang menggunakan representasi vektor kata FastText sebagai bobotnya, dengan parameter 'input\_length' yang sesuai dengan panjang data latih. Kemudian, lapisan Bi-LSTM ditambahkan dengan jumlah unit yang disesuaikan dengan hyperparameter terbaik yang telah ditentukan sebelumnya. Lapisan dropout diterapkan untuk mengurangi overfitting dengan nilai dropout yang sesuai dengan hyperparameter terbaik.

```
# Membangun arsitektur FastText-BiLSTM Embedding
# Inisialisasi model
model = tf.keras.models.Sequential()
# Embedding Layer
model.add(tf.keras.layers.Embedding(vocab_size, fasttext_model.vector_size, weights=[embedding_matrix],
                                   input_length=X_train_fasttext.shape[1], trainable=False))
# Bidirectional LSTM Layer(s)
model.add(tf.keras.layers.Bidirectional([tf.keras.layers.LSTM(units=best_params_fasttext_BiLSTM['units'])]))
model.add(tf.keras.layers.Dropout(best_params_fasttext_BiLSTM['dropout']))
# Output Layer
model.add(tf.keras.layers.Dense(units=2, activation='softmax'))
# Summary model
model.summary()
```

Gambar 4. 8. Baris kode kombinasi model FastText+Bi-LSTM

Langkah selanjutnya, pada gambar 4.9 memaparkan baris kode yang digunakan untuk mengevaluasi waktu yang dibutuhkan untuk menjalankan kode di seluruh sel. Kemudian optimisasi model dilakukan dengan mengatur optimizer sesuai dengan nilai learning rate terbaik yang telah ditentukan sebelumnya. Proses pelatihan model dilakukan menggunakan data latih X\_train dan y\_train, dengan jumlah epoch yang telah ditentukan sebelumnya berdasarkan hyperparameter

terbaik, dan ukuran batch sebesar 128. Selama proses pelatihan, data validasi ( $X_{test}$  dan  $y_{test}$ ) digunakan untuk mengevaluasi performa model pada setiap epoch. Jika tidak terjadi peningkatan dalam loss validasi selama 5 epoch berturut-turut, pelatihan akan dihentikan lebih awal.

```

title
# Model definition
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001, amsgrad=False, clipnorm=1.0)
model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

# Fitting model with validation tracking capabilities
history = model.fit(X_train, y_train, epochs=100, validation_data=(X_test, y_test), callbacks=[tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5)])

```

Gambar 4. 9. Baris kode runtime FastText+Bi-LSTM

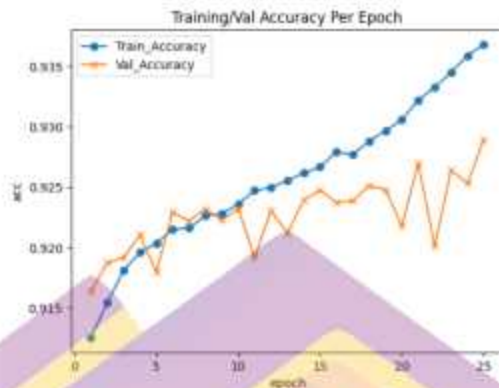
Hasil evaluasi dari kombinasi FastText+Bi-LSTM dalam dua skenario yang berbeda menunjukkan performa yang beragam. Pada skenario pertama, model mencapai akurasi sebesar 92.89%, dengan nilai presisi sebesar 91.99%, recall sebesar 92.89%, dan F1-Score sebesar 91.51%. Namun, pada skenario kedua, meskipun akurasi masih cukup tinggi, yaitu sebesar 81.82%, nilai presisi, recall, dan F1-Score menunjukkan adanya penurunan performa. Hal ini mungkin disebabkan oleh adanya ketidakseimbangan dalam distribusi kelas dalam data. Dengan demikian, hasil evaluasi menunjukkan bahwa performa model FastText+Bi-LSTM dapat dipengaruhi oleh karakteristik dari data yang digunakan. F1-score keseluruhan, mencerminkan rata-rata tertimbang dari nilai F1-score tiap kelas, menunjukkan keseimbangan antara presisi dan recall, dengan nilai sebesar 0.9151, menggambarkan kualitas keseluruhan performa model FastText+Bi-LSTM dalam memprediksi kelas-kelas yang ada lebih tinggi dengan menggunakan parameter berbeda dibandingkan dengan penelitian sebelumnya (Riza & Charibaldi, 2021). Hasil penelitian ini juga menjadi lebih baik dari

penelitian sebelumnya karena mengikuti saran dari (Al-Hashedi dkk., 2023; Nurdin dkk., 2020) yaitu untuk menambah jumlah dataset. Hasil evaluasi model FastText+Bi-LSTM klasifikasi ditunjukkan pada tabel 4.2

Tabel 4. 2. Hasil Kombinasi FastText+Bi-LSTM

Hasil Kombinasi FastText+Bi-LSTM				
Skenario	Accuracy	Precision	Recall	F1-Score
1	0.9289	0.9199	0.9289	0.9151
2	0.8182	0.8249	0.8182	0.8180

Hasil dari kombinasi model FastText dan Bi-LSTM pada skenario yang memiliki performa yang paling tinggi (skenario 1) juga dipaparkan dalam gambar 4.10 dimana grafik tersebut menunjukkan akurasi pelatihan dan validasi dari sebuah model analisis sentimen untuk cyber bullying menggunakan kombinasi FastText dan Bi-LSTM sebanyak 25 epoch. Hal ini tentunya berbeda dengan penelitian sebelumnya (Riza & Charibaldi, 2021) yang menghasilkan akurasi tinggi dengan menggunakan lebih banyak epoch sebesar 50 epoch. Sumbu x mewakili nomor epoch, yang merupakan iterasi seluruh dataset pelatihan. Model dilatih dan divalidasi sebanyak 25 kali, dan akurasi direkam untuk setiap epoch. Sumbu y mewakili akurasi model, yang merupakan persentase prediksi yang benar.



Gambar 4. 10. Hasil training akurasi model FastText-Bi-LSTM

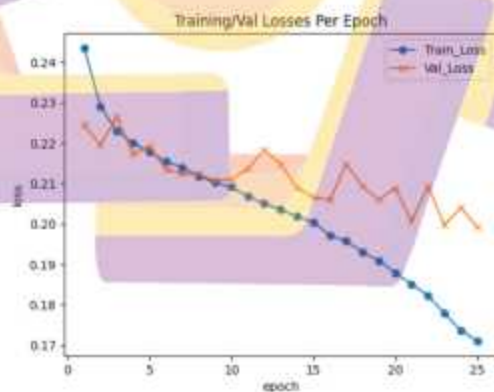
Garis biru mewakili akurasi pelatihan (Train Accuracy), yaitu akurasi model pada dataset pelatihan. Garis oranye mewakili akurasi validasi (Val Accuracy), yaitu akurasi model pada dataset validasi yang terpisah yang tidak dilihat oleh model selama pelatihan. Pada epoch 10, akurasi pelatihan adalah 0.935 (93.5%), dan akurasi validasi adalah 0.930 (93%). Pada epoch 15, akurasi pelatihan adalah 0.930 (93%), dan akurasi validasi adalah 0.925 (92.5%). Pada epoch 20, akurasi pelatihan adalah 0.925 (92.5%), dan akurasi validasi adalah 0.920 (92%). Pada epoch 25, akurasi pelatihan adalah 0.915 (91.5%), dan akurasi validasi adalah 0.915 (91.5%). Akurasi tertinggi dicapai dalam pelatihan ini lebih tinggi menggunakan hyperparameter dibandingkan dengan penelitian sebelumnya (Al-Hashedi dkk. 2023; Nurdin dkk. 2020; Riza dan Charibaldi 2021).

Secara keseluruhan, grafik menunjukkan bahwa model memiliki akurasi tinggi baik pada dataset pelatihan maupun validasi, menunjukkan bahwa model mampu generalisasi dengan baik pada data baru yang tidak terlihat sebelumnya.



Namun, terdapat sedikit penurunan akurasi setelah epoch 15, yang menandakan bahwa model mulai overfitting pada data pelatihan.

Pada gambar 4.11 menampilkan Grafik tersebut menampilkan grafik yang menunjukkan kerugian (loss) pelatihan dan validasi dari sebuah model menggunakan kombinasi fastText dan Bi-LSTM. Sumbu x mewakili nomor epoch, yang merupakan iterasi seluruh dataset pelatihan. Model dilatih dan divalidasi sebanyak 25 kali, dan kerugian direkam untuk setiap epoch. Sumbu y mewakili kerugian (loss) dari model, yang merupakan ukuran seberapa baik model cocok dengan data. Kerugian yang lebih rendah menunjukkan kesesuaian yang lebih baik. Garis biru mewakili kerugian pelatihan (Train\_Loss), yaitu kerugian dari model pada dataset pelatihan. Garis oranye mewakili kerugian validasi (Val\_Loss), yaitu kerugian dari model pada dataset validasi yang terpisah yang tidak dilihat oleh model selama pelatihan.



Gambar 4. 11. Hasil training loss model FastText-Bi-LSTM



Pada tabel 4.3, selama proses pelatihan model FastText+Bi-LSTM terjadi perkembangan yang konsisten dan signifikan selama 25 epoch. Dalam setiap epoch, waktu yang diperlukan untuk melatih model adalah sekitar 4 detik, menandakan efisiensi komputasi yang baik. Dari hasil pelatihan, terlihat bahwa nilai loss pada data pelatihan secara bertahap menurun dari 0.2436 pada epoch pertama menjadi 0.1710 pada epoch ke-25. Selain itu, akurasi pelatihan juga mengalami peningkatan yang signifikan dari 91.26% menjadi 93.68%. Hal yang menarik adalah bahwa penurunan loss dan peningkatan akurasi tidak hanya terjadi pada data pelatihan, tetapi juga pada data validasi. Pada data validasi, loss menurun dari 0.2245 menjadi 0.1992, sementara akurasi naik dari 91.64% menjadi 92.89%. Pada penelitian yang dilakukan sekarang, terdapat peningkatan kinerja yang signifikan dengan menggunakan arsitektur yang berbeda dengan penelitian terdahulu (Riza & Charibaldi, 2021) yang menyoroti bahwa hasil terbaik dari evaluasi model Bi-LSTM menggunakan arsitektur FastText dengan 50 unit dan 50 dropouts dan menghasilkan akurasi sebesar 73,15%. Dengan demikian, penelitian ini menunjukkan peningkatan yang signifikan dalam akurasi dan kinerja model klasifikasi teks cyber bullying dengan parameter dropout 0.2, epochs 25, lr 0.001, dan units 512. Tabel 4.3 menampilkan secara keseluruhan nilai epoch selama training.

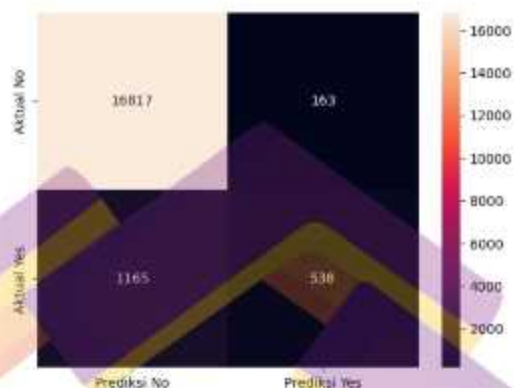
Tabel 4. 3. Hasil Training FastText+Bi-LSTM

Hasil Training FastText+Bi-LSTM					
Epoch	Time	Loss (Train)	Acc (Train)	Loss (Validation)	Acc (Validation)
1	7s	0.2436	91.26%	0.2245	91.64%

Tabel 4.3, Tabel Lanjutan

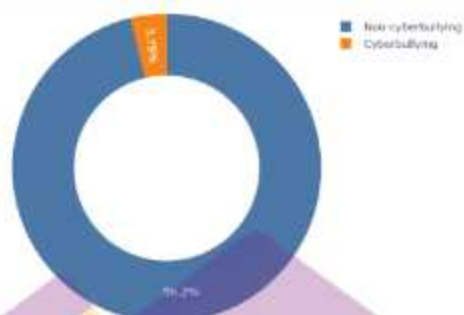
Epoch	Time	Loss (Train)	Acc (Train)	Loss (Validation)	Acc (Validation)
2	4s	0.2291	91.55%	0.2195	91.87%
3	4s	0.2230	91.81%	0.2264	91.92%
4	4s	0.2200	91.96%	0.2173	92.11%
5	4s	0.2178	92.04%	0.2191	91.80%
6	4s	0.2155	92.15%	0.2135	92.29%
7	4s	0.2139	92.16%	0.2124	92.22%
8	4s	0.2119	92.26%	0.2123	92.32%
9	4s	0.2102	92.28%	0.2105	92.23%
10	4s	0.2091	92.36%	0.2111	92.32%
11	4s	0.2068	92.47%	0.2137	91.92%
12	4s	0.2050	92.50%	0.2183	92.31%
13	4s	0.2035	92.56%	0.2148	92.12%
14	4s	0.2018	92.62%	0.2090	92.39%
15	4s	0.2004	92.67%	0.2066	92.47%
16	4s	0.1972	92.79%	0.2059	92.37%
17	4s	0.1957	92.77%	0.2148	92.39%
18	4s	0.1930	92.88%	0.2093	92.51%
19	4s	0.1908	92.97%	0.2050	92.48%
20	4s	0.1879	93.06%	0.2088	92.18%
21	4s	0.1850	93.22%	0.2007	92.70%
22	4s	0.1822	93.33%	0.2092	92.01%
23	4s	0.1778	93.45%	0.1998	92.64%
24	4s	0.1736	93.59%	0.2040	92.53%
<b>25</b>	<b>4s</b>	<b>0.1710</b>	<b>93.68%</b>	<b>0.1992</b>	<b>92.89%</b>

Kinerja model menggabungkan fastText dan Bi-LSTM selama 25 epoch ditampilkan pada gambar 4.12.



Gambar 4. 12. Confusion Matrik model FastText-Bi-LSTM

Selain itu grafik lingkaran pada gambar 4.13 menampilkan distribusi persentase cyber bullying dan non-cyber bullying berdasarkan prediksi yang dibuat oleh model menggunakan kombinasi fastText dan Bi-LSTM. Warna biru mewakili proporsi yang diprediksi sebagai non-cyber bullying, yaitu sebesar 96.2%. Warna oranye mewakili proporsi yang diprediksi sebagai cyber bullying, yaitu sebesar 3.75%. Secara keseluruhan, grafik lingkaran menunjukkan bahwa model lebih cenderung memprediksi sebuah instansi sebagai non-cyber bullying (96.2%) daripada sebagai cyber bullying (3.75%).



Gambar 4. 13. Grafik Persentase FastText+Bi-LSTM Berdasarkan Prediksi Label

#### 4.4.Eksperimen dan Evaluasi Kombinasi BERT dan Bi-LSTM

Penelitian ini memanfaatkan arsitektur BERT pre-trained yang telah dilatih sebelumnya sebagai landasan utama dalam analisis dan pemrosesan data. Dengan menggunakan model BERT pre-trained, penelitian ini dapat memanfaatkan pengetahuan yang terdapat dalam korpus yang luas dan beragam untuk memahami teks yang kompleks, sehingga model dapat diadaptasi untuk tugas tertentu seperti klasifikasi, pemodelan bahasa, atau generasi teks.

Proses pembuatan model BERT menggunakan library TensorFlow dan Hugging Face's Transformers. Library TensorFlow digunakan untuk memuat model dan bekerja dengan model BERT dalam lingkungan TensorFlow, sementara Hugging Face's Transformers digunakan untuk memuat tokenizer dan model BERT dari pre-trained weights. Pada gambar 4.14 merupakan baris kode yang digunakan untuk memuat tokenizer dan model IndoBERT dalam pemrosesan bahasa alami berbahasa Indonesia. Langkah pertama adalah menginisialisasi tokenizer yang digunakan untuk mengonversi teks ke dalam

representasi numerik yang dapat dimengerti oleh model. Kemudian memuat model BERT yang telah dilatih sebelumnya guna untuk melakukan berbagai tugas pemrosesan bahasa alami yang diperlukan seperti klasifikasi teks, ekstraksi informasi, atau generasi teks dalam bahasa Indonesia.

```
# load tokenizer dan model IndoBERT
tokenizer = BertTokenizer.from_pretrained('indobenchmark/indobert-base-pl')
bert_model = TFBertModel.from_pretrained('indobenchmark/indobert-base-pl')
```

Gambar 4. 14. Baris kode memuat model BERT

Selanjutnya menyiapkan data dan label untuk proses pelatihan dan pengujian model BERT dalam sebuah tugas klasifikasi menggunakan baris kode pada gambar 4.15. Pertama, panjang maksimum urutan token ditetapkan menjadi 20 untuk membatasi panjang teks input. Kemudian, data pelatihan dan pengujian di-tokenisasi dan di-encode menggunakan tokenizer BERT, dengan memastikan bahwa setiap urutan token sesuai dengan panjang maksimum yang ditetapkan, dan jika perlu, teks yang terlalu panjang dipotong. Data teks yang kosong diisi dengan string kosong, dan setelah tokenisasi, data tersebut diubah menjadi array numpy. Selanjutnya, label pelatihan dan pengujian diubah menjadi bentuk one-hot encoding menggunakan fungsi 'to\_categorical' dari library Keras, dan juga diubah menjadi array numpy. Proses ini mempersiapkan data dan label dalam format yang sesuai untuk proses pelatihan dan evaluasi model klasifikasi menggunakan model BERT. Proses-proses tersebut secara keseluruhan dilakukan menggunakan fungsi pada baris kode pada gambar 4.15.



```

# Tolaklah panjang urutan maksimum untuk tokenizer BERT
max_length = 20

# Tokenisasi dan siapkan data pelatihan
X_train = [tokenizer.encode(text, max_length=max_length, truncation=True, padding='max_length') for text in train_df['text']]
X_train = np.array(X_train)

# Tokenisasi dan siapkan data uji
test_df['text'] = test_df['text'].fillna('')
X_test = [tokenizer.encode(text, max_length=max_length, truncation=True, padding='max_length') for text in test_df['text']]
X_test = np.array(X_test)

# Label pelatihan dan pengujian
y_train = tn_categorical(train_df['label'], num_classes=2)
y_train = np.array(y_train)
y_test = tn_categorical(test_df['label'], num_classes=2)
y_test = np.array(y_test)

```

Gambar 4. 15. Baris kode proses pelatihan dan pengujian model BERT

Pada tahap selanjutnya eksperimen kombinasi BERT dengan Bi-LSTM dimulai dengan terlebih dahulu menggunakan hyperparameter. Hyperparameter tuning digunakan pada tahap ini untuk mencari kombinasi nilai hyperparameter optimal untuk model guna mencapai performa terbaik. Proses pencarian hyperparameter terbaik menggunakan algoritma optimasi yakni Hyperopt. Fungsi 'tuningBERTBi-LSTM' pada gambar 4.16 dibawah adalah untuk membangun sebuah model neural network yang menggabungkan arsitektur BERT dengan layer-layer LSTM (Long Short-Term Memory) yang bidirectional, yang sering digunakan dalam pemrosesan bahasa alami. Pertama, fungsi menerima parameter params, yang merupakan sebuah dictionary berisi berbagai parameter untuk model seperti jumlah unit LSTM, dropout rate, dan learning rate. Model dimulai dengan dua input layer, yaitu 'input\_ids' dan 'attention\_mask', yang mewakili urutan token dan mask yang diperlukan untuk input ke model BERT. Kemudian, layer embedding BERT digunakan untuk mengubah token-token menjadi representasi vektor yang kaya secara semantik. Selanjutnya, dilakukan layer Bi-LSTM dengan jumlah unit sesuai dengan nilai yang diberikan pada parameter units. Dropout layer diterapkan untuk mencegah overfitting dalam model. Layer output memiliki

dua unit dan fungsi aktivasi softmax, sesuai dengan tugas klasifikasi dengan dua kelas. Sebelum kompilasi, layer-layer dalam model BERT dibekukan dengan mengatur trainable menjadi False. Ini dilakukan agar bobot dari model BERT tidak ikut diupdate selama proses fine-tuning. Selanjutnya, ringkasan dari model dicetak untuk memudahkan pemantauan dan evaluasi model.

```
def tuningBERTBiLSTM(params):
    # Initial model
    input_ids = tf.keras.layers.Input(shape=(max_length,), dtype='int32')
    attention_mask = tf.keras.layers.Input(shape=(max_length,), dtype='int32')
    # Bert Embedding Layer
    embedding = bert_model(input_ids, attention_mask=attention_mask)[0]
    # Bidirectional LSTM Layer(s)
    biLstm = tf.keras.layers.Bidirectional([tf.keras.layers.LSTM(units=params['units'])])(embedding)
    dropout = tf.keras.layers.Dropout(params['dropout'])(biLstm)
    # Output Layer
    output = tf.keras.layers.Dense(units=2, activation='softmax')(dropout)
    # Define model
    model = tf.keras.models.Model(inputs=[input_ids, attention_mask], outputs=output)
    # Set trainable to False for layer Bert Embedding
    for layer in model.layers[:3]:
        layer.trainable = False
    # compile model
    model.compile(loss='categorical_crossentropy',
                  optimizer=tf.keras.optimizers.Adam(learning_rate=params['lr']),
                  metrics=['accuracy'])
    # Summary model
    model.summary()
```

Gambar 4. 16. Baris Kode Proses Hyperparameter Tuning BERT-Bi-LSTM

Selanjutnya, menjalankan proses hyperparameter tuning untuk model yang telah didefinisikan sebelumnya, yang menggabungkan arsitektur BERT dengan layer-layer Bi-LSTM menggunakan baris kode sesuai pada gambar . Pertama, dilakukan pelatihan model dengan menggunakan fungsi 'model.fit()' pada data pelatihan X\_train dan label y\_train. Model dilatih selama sejumlah epoch yang ditentukan dalam parameter 'params['epochs']', dengan ukuran batch yang ditentukan pula. Data validasi yang terdiri dari X\_test dan y\_test digunakan untuk evaluasi model selama pelatihan, dan callback EarlyStopping digunakan untuk menghentikan pelatihan jika tidak ada peningkatan yang signifikan dalam metrik

validasi, dalam hal ini, 'val\_loss', selama sejumlah epoch yang ditentukan oleh patience. Setelah proses pelatihan selesai, akurasi pada data validasi digunakan sebagai metrik untuk evaluasi hyperparameter, yang kemudian digunakan untuk menghitung nilai fungsi objektif. Kemudian, dilakukan proses hyperparameter tuning dengan menggunakan fungsi 'fmin()' dari library Hyperopt. Parameter-parameter yang akan dioptimalkan termasuk jumlah unit dalam layer Bi-LSTM, learning rate, dropout rate, dan jumlah epoch. Fungsi 'fmin()' menggunakan algoritma TPE (Tree-structured Parzen Estimator) untuk mencari kombinasi hyperparameter yang memberikan hasil terbaik berdasarkan fungsi objektif yang didefinisikan sebelumnya. Setelah proses hyperparameter tuning selesai, hasil hyperparameter terbaik dicetak untuk dianalisis lebih lanjut.

```
# Definisi ruang pencarian hyperparameter
search_space = {
    'units': hp.choice('units', [100, 200, 500]),
    'lr': hp.choice('lr', [0.0001, 0.001, 0.01]),
    'dropout': hp.choice('dropout', [0.2, 0.3, 0.5]),
    'epochs': hp.choice('epochs', [20, 50, 100])
}

# Lakukan hyperparameter tuning
trials = Trials()
best = fmin(tuningBERTBiLSTM, search_space, algo=tpe.suggest, max_evals=1, trials=trials)

# Cetak hyperparameter terbaik
print("Best hyperparameters:")
best_params_BERT_BiLSTM = space_eval(search_space, best)
print(best_params_BERT_BiLSTM)
```

Gambar 4. 17. Baris Kode Penentuan Parameter Terbaik BERT-Bi-LSTM

Hasil dari proses pencarian hyperparameter terbaik ditampilkan dalam gambar 4.18 yang menunjukkan bahwa setiap baris output mewakili metrik loss dan akurasi pada setiap epoch selama proses pelatihan. Terdapat 341 batch data dalam total, dengan setiap batch diikuti oleh nilai loss dan akurasi. Pada akhirnya, hasil pelatihan adalah loss 0.2518 dan akurasi 0.9173 pada data pelatihan, serta

loss 0.2608 dan akurasi 0.9188 pada data validasi. Setelah proses hyperparameter tuning selesai, hasil terbaik yang ditemukan adalah dropout: 0.5, epochs: 100, learning rate (lr): 0.01, dan jumlah unit dalam layer LSTM (units): 128, yang memberikan akurasi validasi sebesar 0.9188.

```

336/331 | ..... | - ETA: 0s - loss: 0.2225 - accuracy: 0.9172
332/341 | ..... | - ETA: 0s - loss: 0.2523 - accuracy: 0.9152
332/341 | ..... | - ETA: 0s - loss: 0.2523 - accuracy: 0.9152
332/341 | ..... | - ETA: 0s - loss: 0.2342 - accuracy: 0.9172
334/331 | ..... | - ETA: 0s - loss: 0.2523 - accuracy: 0.9152
336/341 | ..... | - ETA: 0s - loss: 0.2308 - accuracy: 0.9198
336/341 | ..... | - ETA: 0s - loss: 0.2308 - accuracy: 0.9198
337/341 | ..... | - ETA: 0s - loss: 0.2308 - accuracy: 0.9198
338/341 | ..... | - ETA: 0s - loss: 0.2213 - accuracy: 0.9172
339/341 | ..... | - ETA: 0s - loss: 0.2343 - accuracy: 0.9172
342/331 | ..... | - ETA: 0s - loss: 0.2308 - accuracy: 0.9198
342/331 | ..... | - ETA: 0s - loss: 0.2308 - accuracy: 0.9198
200 epochs - loss: 0.2308 - accuracy: 0.9198 - val_loss: 0.2608 - val_accuracy: 0.9188

200 epochs - loss: 0.2308 - accuracy: 0.9198
Best hyperparameters
{'dropout': 0.5, 'epochs': 100, 'lr': 0.01, 'units': 128}

```

Gambar 4. 18. Output hasil hyperparameter BERT-Bi-LSTM

Selanjutnya parameter terbaik diterapkan pada kombinasi BERT dengan Bi-LSTM. Baris kode pada gambar 4.19 digunakan untuk membangun arsitektur neural network yang menggabungkan model BERT dengan layer Bi-LSTM. Proses dimulai dengan mendefinisikan dua input layer, yaitu 'input\_ids' dan 'attention\_mask', yang mewakili urutan token dan mask yang diperlukan untuk input ke model BERT. Selanjutnya, model BERT dimuat melalui layer embedding, yang mengubah urutan token menjadi representasi vektor berdimensi tinggi yang kaya secara semantik. Proses tokenisasi dan embedding ini membantu dalam menangkap hubungan kontekstual antara kata-kata dalam teks. Layer Bi-LSTM kemudian diterapkan di atas output dari model BERT, dengan jumlah unit sesuai dengan nilai terbaik yang telah ditemukan melalui proses hyperparameter tuning. Bi-LSTM adalah jenis layer rekuren yang mampu mengingat informasi dari kedua arah urutan, sehingga membantu dalam memahami konteks teks



dengan lebih baik. Layer dropout digunakan untuk mengurangi overfitting dengan secara acak menghapus beberapa unit selama pelatihan. Terakhir, output layer dengan fungsi aktivasi softmax digunakan untuk melakukan prediksi klasifikasi, di mana dua unit merepresentasikan probabilitas kelas. Selanjutnya, layer-layer pertama yang terkait dengan model BERT dibekukan agar bobotnya tidak diupdate selama proses pelatihan. Ini dilakukan dengan mengatur atribut trainable menjadi False. Ringkasan dari arsitektur model dicetak untuk memberikan gambaran tentang struktur dan parameter yang digunakan dalam model yang telah dibangun.

```
# Ringkasan arsitektur BERT-BiLSTM embedding
# Input Layer
input_ids = tf.keras.layers.Input(shape=(max_length,), dtype='int32')
attention_mask = tf.keras.layers.Input(shape=(max_length,), dtype='int32')
# Bert Embedding Layer
embedding = bert_model(input_ids, attention_mask=attention_mask)[0]
# Hidden Layer BiLSTM
bi_lstm = tf.keras.layers.Bidirectional([tf.keras.layers.LSTM(best_params_bert_BiLSTM['units'])](embedding))
# Dropout Layer
dropout = tf.keras.layers.Dropout(best_params_bert_BiLSTM['dropout'])(bi_lstm)
# Output Layer untuk pemodelan
output = tf.keras.layers.Dense(2, activation='softmax')(dropout)

model = tf.keras.models.Model(inputs=[input_ids, attention_mask], outputs=output)

# Setel trainable ke false di layer bert embedding
for layer in model.layers[1:]:
    layer.trainable = False

# Ringkasan arsitektur model
model.summary()
```

Gambar 4. 19. Baris kode kombinasi model BERT+ Bi-LSTM

Langkah selanjutnya, pada gambar 4.20 memaparkan baris kode yang digunakan untuk mengevaluasi waktu yang dibutuhkan untuk menjalankan kode di seluruh sel. Perintah '%time' digunakan untuk mengukur waktu yang diperlukan untuk mengeksekusi baris kode selanjutnya dalam sel. Setelah itu, optimizer untuk model didefinisikan menggunakan algoritma RMSprop dengan



learning rate yang diperoleh dari hasil hyperparameter tuning sebelumnya. Kemudian, model dikompilasi dengan menggunakan optimizer yang telah ditetapkan, loss function categorical crossentropy, dan metrik akurasi. Proses pelatihan model kemudian dilakukan dengan memanggil fungsi 'fit()' pada data pelatihan dan validasi. Data pelatihan dan validasi disertakan dalam bentuk tuple yang berisi input dan mask untuk model BERT serta label kategorikal. Proses pelatihan dilakukan selama sejumlah epoch yang telah ditentukan sebelumnya melalui proses hyperparameter tuning, dengan ukuran batch yang ditentukan pula. Output dari proses pelatihan, seperti nilai loss dan metrik akurasi, dapat disimpan untuk analisis lebih lanjut.

```

Xtime
# Setel optimizer
optimizer = tf.keras.optimizers.Adam(learning_rate=best_params_bert_BiLSTM['lr'])

model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

# Fitting model atau melakukan training (pelatihan)
history = model.fit([X_train, np.ones((len(X_train), max_length))], y_train,
                    validation_data=([X_test, np.ones((len(X_test), max_length))], y_test),
                    epochs=best_params_bert_BiLSTM['epochs'], batch_size=128)

```

Gambar 4. 20. Baris Kode runtime BERT-Bi-LSTM

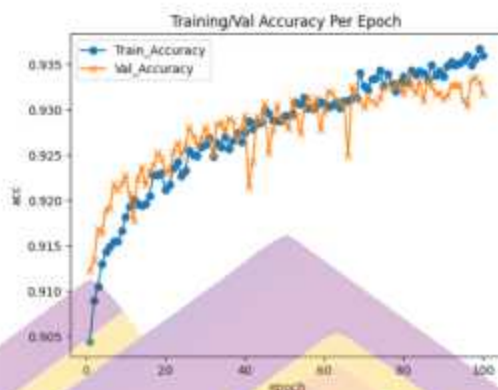
Hasil evaluasi dari kombinasi BERT+Bi-LSTM dalam dua skenario yang berbeda menunjukkan kinerja yang konsisten. Pada skenario pertama, model mencapai tingkat akurasi yang sangat tinggi, yaitu 93.17%, dengan nilai presisi, recall, dan F1-Score masing-masing adalah 92.88%, 93.17%, dan 91.54%. Ini mengindikasikan kemampuan model dalam mengklasifikasikan data dengan tingkat keakuratan yang sangat baik. Pada skenario kedua, meskipun terjadi sedikit penurunan dalam beberapa metrik, akurasi tetap tinggi, yaitu 89.76%, dan

nilai presisi, recall, dan F1-Score tetap seimbang pada 89.76%. Hasil evaluasi ini menunjukkan bahwa model BERT+Bi-LSTM memiliki kinerja yang sangat baik dalam kedua skenario, yang menunjukkan kestabilan dan keandalannya dalam menangani situasi dalam klasifikasi data. Hasil penelitian ini juga menjadi lebih baik dari penelitian sebelumnya karena mengikuti saran dari (Al-Hashedi dkk., 2023; Nurdin dkk., 2020) yaitu untuk menambah jumlah dataset. Hasil evaluasi model FastText+Bi-LSTM klasifikasi ditunjukkan pada tabel 4.3.

Tabel 4. 4. Hasil Kombinasi BERT+Bi-LSTM

<b>Hasil Kombinasi BERT+Bi-LSTM</b>				
<b>Skenario</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>
1	0.9317	0.9288	0.9317	0.9154
2	0.8976	0.8976	0.8976	0.8976

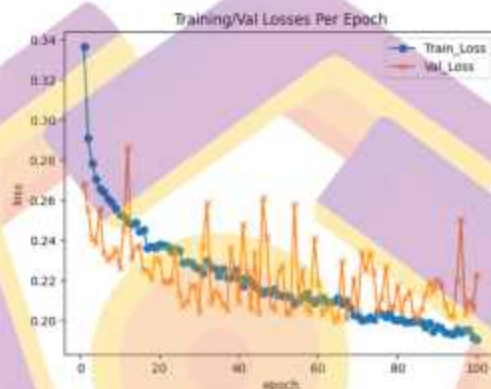
Output dari proses pelatihan dalam skenario yang menghasilkan hasil tertinggi (skenario 1), seperti nilai loss dan metrik akurasi dapat dilihat pada gambar 4.21. Gambar 4.21 menunjukkan akurasi pelatihan dan validasi per-epoch untuk model kombinasi antara BERT dan Bi-LSTM. Pada sumbu-x mewakili nomor epoch, sementara sumbu-y mewakili akurasi. Terdapat dua garis dalam grafik, garis biru untuk akurasi pelatihan (Train\_Accuracy) dan garis oranye untuk akurasi validasi (Val\_Accuracy).



Gambar 4. 21. Hasil training akurasi model BERT-Bi-LSTM

Model ini dilatih selama 100 epoch, dan akurasi pelatihan dimulai pada sekitar 0.935 dan secara perlahan menurun seiring bertambahnya jumlah epoch. Di sisi lain, akurasi validasi dimulai pada sekitar 0.930 dan fluktuatif sepanjang proses pelatihan, dengan tren umum yang naik. Akurasi validasi tertinggi yang dicapai adalah sekitar 0.935, yang terjadi pada epoch 20. Setelah itu, akurasi validasi menurun dan kemudian mulai meningkat lagi, mencapai maksimum lokal sekitar 0.930 pada epoch 60. Akurasi validasi kemudian turun lagi, mencapai minimum lokal sekitar 0.905 pada epoch 80, sebelum kembali meningkat menjadi sekitar 0.910 pada epoch 100. Pada gambar 4.22 menampilkan grafik yang menunjukkan kerugian pelatihan dan validasi per epoch untuk sebuah model yang tampaknya merupakan kombinasi antara BERT dan Bi-LSTM. Pada sumbu-x mewakili nomor epoch, sementara sumbu-y mewakili kerugian. Terdapat dua garis dalam grafik, garis biru untuk kerugian pelatihan (Train\_Loss) dan garis oranye untuk kerugian validasi (Val\_Loss).

Model ini dilatih selama 100 epoch, dan kerugian pelatihan dimulai pada sekitar 0.34 dan secara perlahan menurun seiring bertambahnya jumlah epoch. Ini adalah perilaku yang diharapkan, karena kinerja model menjadi lebih baik dalam prediksinya pada data pelatihan, dan umumnya kerugian pelatihan akan menurun seiring dengan peningkatan kerugian validasi.



Gambar 4. 22. Hasil training loss model BERT-Bi-LSTM

Hasil pelatihan model BERT+Bi-LSTM menunjukkan pola yang menarik dan kompleks selama 100 epoch. Meskipun waktu pelatihan relatif stabil sekitar 26 detik per epoch, pola penurunan loss dan peningkatan akurasi tidaklah linear. Pada awal pelatihan, terlihat penurunan loss yang signifikan dari 0.3366 pada epoch pertama menjadi 0.1906 pada epoch ke-100, serta peningkatan akurasi dari 0.9044 menjadi 0.9360. Namun, di tengah-tengah pelatihan, terdapat fluktuasi yang cukup besar dalam performa model, terutama terlihat dari fluktuasi loss pada data pelatihan dan validasi. Pada beberapa titik, terjadi kenaikan tajam pada loss dan penurunan pada akurasi, namun model mampu pulih dan kembali

meningkatkan performanya. Tabel 4.5 menampilkan secara keseluruhan nilai epoch selama training.

Tabel 4. 5. Hasil Training BERT+Bi-LSTM

Hasil Training BERT+Bi-LSTM					
Epoch	Time	Loss (Train)	Acc (Train)	Loss (Validation)	Acc (Validation)
1	44s	0.3366	0.9044	0.2682	0.9122
2	26s	0.2910	0.9089	0.2553	0.9133
3	26s	0.2784	0.9105	0.2400	0.9168
4	26s	0.2705	0.9130	0.2385	0.9163
5	26s	0.2657	0.9143	0.2559	0.9190
6	26s	0.2639	0.9149	0.2336	0.9192
7	26s	0.2607	0.9155	0.2302	0.9217
8	26s	0.2587	0.9155	0.2319	0.9212
9	26s	0.2558	0.9160	0.2359	0.9218
10	26s	0.2528	0.9182	0.2260	0.9228
11	26s	0.2519	0.9194	0.2458	0.9207
12	26s	0.2493	0.9201	0.2863	0.9177
13	26s	0.2479	0.9196	0.2305	0.9224
14	26s	0.2488	0.9195	0.2363	0.9236
15	26s	0.2444	0.9197	0.2376	0.9217
16	26s	0.2453	0.9205	0.2251	0.9232
17	26s	0.2360	0.9228	0.2249	0.9238
18	26s	0.2371	0.9228	0.2208	0.9253
19	26s	0.2360	0.9230	0.2311	0.9246
20	26s	0.2379	0.9211	0.2303	0.9231
21	26s	0.2374	0.9218	0.2195	0.9233



Tabel 4.5. Lanjutan

Epoch	Time	Loss (Train)	Acc (Train)	Loss (Validation)	Acc (Validation)
22	26s	0.2371	0.9235	0.2194	0.9253
23	26s	0.2346	0.9242	0.2195	0.9264
24	26s	0.2359	0.9227	0.2358	0.9250
25	26s	0.2347	0.9233	0.2131	0.9259
26	26s	0.2287	0.9255	0.2074	0.9281
27	26s	0.2293	0.9250	0.2091	0.9271
28	26s	0.2286	0.9249	0.2176	0.9261
29	26s	0.2265	0.9259	0.2172	0.9267
30	26s	0.2259	0.9262	0.2044	0.9281
31	26s	0.2236	0.9270	0.2372	0.9268
32	26s	0.2299	0.9248	0.2587	0.9250
33	26s	0.2267	0.9264	0.2089	0.9284
34	26s	0.2258	0.9259	0.2147	0.9285
35	26s	0.2225	0.9270	0.2127	0.9275
36	26s	0.2258	0.9257	0.2081	0.9291
37	26s	0.2215	0.9265	0.2054	0.9288
38	26s	0.2216	0.9275	0.2366	0.9265
39	26s	0.2252	0.9264	0.2237	0.9279
40	26s	0.2214	0.9275	0.2091	0.9293
41	26s	0.2171	0.9287	0.2479	0.9214
42	26s	0.2209	0.9279	0.2241	0.9243
43	26s	0.2191	0.9285	0.2058	0.9292
44	26s	0.2171	0.9286	0.2341	0.9287
45	26s	0.2154	0.9299	0.2038	0.9307
46	26s	0.2137	0.9297	0.2608	0.9252

Tabel 4.5. Lanjutan

Epoch	Time	Loss (Train)	Acc (Train)	Loss (Validation)	Acc (Validation)
47	26s	0.2150	0.9288	0.2415	0.9277
48	26s	0.2136	0.9288	0.2069	0.9305
49	26s	0.2159	0.9288	0.2053	0.9290
50	26s	0.2128	0.9293	0.2243	0.9287
51	26s	0.2121	0.9293	0.2276	0.9279
52	26s	0.2124	0.9297	0.2033	0.9311
53	26s	0.2117	0.9308	0.2040	0.9292
54	26s	0.2085	0.9306	0.2581	0.9274
55	26s	0.2103	0.9313	0.2098	0.9300
56	26s	0.2117	0.9302	0.2251	0.9313
57	26s	0.2138	0.9301	0.2061	0.9301
58	26s	0.2105	0.9305	0.2037	0.9313
59	26s	0.2087	0.9309	0.2410	0.9278
60	26s	0.2104	0.9307	0.2188	0.9297
61	26s	0.2115	0.9303	0.2032	0.9320
62	26s	0.2098	0.9305	0.2058	0.9319
63	26s	0.2086	0.9309	0.2086	0.9311
64	26s	0.2095	0.9302	0.1995	0.9308
65	26s	0.2114	0.9309	0.2013	0.9305
66	26s	0.2084	0.9310	0.2296	0.9250
67	26s	0.2095	0.9312	0.2005	0.9326
68	26s	0.2068	0.9314	0.2084	0.9312
69	26s	0.2029	0.9341	0.2208	0.9310
70	26s	0.2022	0.9326	0.2050	0.9303
71	26s	0.2003	0.9322	0.2339	0.9314

Tabel 4.5. Lanjutan

Epoch	Time	Loss (Train)	Acc (Train)	Loss (Validation)	Acc (Validation)
72	26s	0.2005	0.9334	0.2248	0.9310
73	26s	0.2019	0.9335	0.2338	0.9306
74	26s	0.2006	0.9343	0.2242	0.9313
75	26s	0.2045	0.9326	0.2039	0.9324
76	26s	0.2030	0.9340	0.2108	0.9328
77	26s	0.2018	0.9329	0.2270	0.9324
78	26s	0.2037	0.9321	0.2069	0.9327
79	26s	0.2007	0.9330	0.2061	0.9328
80	26s	0.2000	0.9337	0.2175	0.9318
81	26s	0.2005	0.9330	0.2037	0.9338
82	26s	0.1991	0.9343	0.2115	0.9325
83	26s	0.1992	0.9339	0.2145	0.9318
84	26s	0.1997	0.9343	0.2035	0.9331
85	26s	0.1998	0.9340	0.2004	0.9310
86	26s	0.1994	0.9333	0.2053	0.9332
87	26s	0.1968	0.9349	0.2131	0.9326
88	26s	0.1988	0.9338	0.2187	0.9318
89	26s	0.1945	0.9342	0.2142	0.9319
90	26s	0.1972	0.9337	0.2202	0.9312
91	26s	0.1963	0.9348	0.2181	0.9316
92	26s	0.1942	0.9353	0.2079	0.9327
93	26s	0.1935	0.9350	0.2026	0.9327
94	26s	0.1931	0.9351	0.2020	0.9327
95	26s	0.1959	0.9354	0.2203	0.9311
96	26s	0.1942	0.9360	0.2505	0.9304

Tabel 4.5. Lanjutan

Epoch	Time	Loss (Train)	Acc (Train)	Loss (Validation)	Acc (Validation)
97	26s	0.1954	0.9350	0.2030	0.9333
98	26s	0.1953	0.9356	0.2099	0.9335
99	26s	0.1918	0.9368	0.2064	0.9331
100	26s	0.1906	0.9360	0.2228	0.9317

Confusion Matrix yang ditunjukkan pada gambar 4.23 menunjukkan performa model klasifikasi BERT+Bi-LSTM. Jumlah true positive (TP) yang tercatat adalah sebesar 490, yang mengindikasikan jumlah prediksi yang benar untuk kelas "Yes". Selain itu, terdapat 16917 true negative (TN), yang menggambarkan jumlah prediksi yang benar untuk kelas "No". Di sisi lain, terdapat 63 false positive (FP), yang mewakili jumlah prediksi yang salah untuk kelas "Yes" (Type I error), dan 1213 false negative (FN), yang menunjukkan jumlah prediksi yang salah untuk kelas "No" (Type II error). Berdasarkan matriks yang dihasilkan, akurasi model adalah  $(16917+490)/(16917+1213+63+490) = 0,93$ , presisi untuk kelas Yes adalah  $490/(490+63) = 0,88$ , dan recall untuk kelas Yes adalah  $490/(490+1213) = 0,29$ .

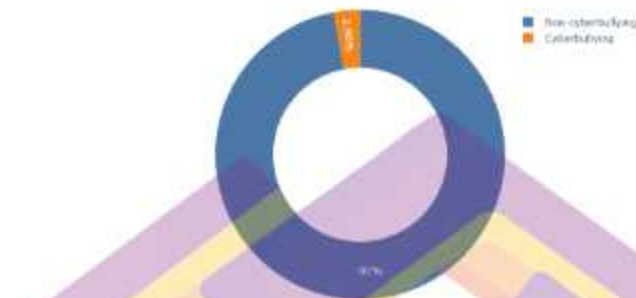


Gambar 4. 23. Confusion Matrix model BERT-Bi-LSTM

Hasil yang diperoleh dari model BERT+Bi-LSTM penelitian ini juga serupa dengan hasil penelitian (Geet D'sa dkk., 2020) dimana penggunaan word embedding BERT memiliki akurasi terbilang terbaik dibandingkan word embedding FastText. Hal ini terjadi karena model BERT yang digunakan pada penelitian merupakan model BERT yang telah dilatih dari dataset yang ada. Grafik yang dihasilkan dalam gambar 4.24 menunjukkan distribusi persentase dari label yang diprediksi model menggunakan kombinasi BERT dan Bi-LSTM. Sumbu y merepresentasikan persentase dari setiap label yang diprediksi, sedangkan sumbu x merepresentasikan kategori label yang diprediksi, yaitu "cyber bullying" dan "Non-cyber bullying". Dari grafik tersebut, model memprediksi "Non-cyber bullying" untuk 97% teks masukan, sementara hanya 2,96% dari teks masukan yang diprediksi sebagai "cyber bullying". Oleh karena itu, kita dapat menyimpulkan bahwa model cenderung sangat bias terhadap



prediksi "Non-cyber bullying", dan mungkin tidak secara akurat mendeteksi kejadian cyber bullying.



Gambar 4. 24. Grafik Persentase BERT+Bi-LSTM Berdasarkan Prediksi Label

#### 4.5. Analisis Hasil Eksperimen

Hasil penelitian menunjukkan bahwa kedua model memiliki kinerja yang nyaris sama dalam hal akurasi, precision, recall, dan F1-score. Secara khusus, dalam skenario pertama yang menggunakan data imbalanced FastText+Bi-LSTM mencapai akurasi 92,89%, presisi 93,17%, recall 91,99%, dan skor F1 91,51%, sementara BERT+Bi-LSTM berhasil mencapai akurasi 92,88%, precision 93,17, recall 92.89%, dan F1-score 91,54%. Pada skenario kedua yang menggunakan data balanced, FastText+Bi-LSTM memiliki hasil akurasi sebesar 81.82%, presisi 82.49%, recall 81.82%, dan F1-Score 81.80% yang dijalankan dalam waktu 28.32 detik. Sementara model BERT+Bi-LSTM memiliki Akurasi sebesar 89.76%, dengan presisi, recall, dan F1-Score masing-masing sebesar 89.76%. Model ini dijalankan dalam waktu 97.44 detik. Dalam dua skenario eksperimen

yang telah dilakukan, model BERT+Bi-LSTM terbukti lebih unggul dan memiliki akurasi yang tinggi dibandingkan model FastText+Bi-LSTM.

Hasil tersebut menunjukkan peningkatan yang signifikan berkat saran dari penelitian terdahulu (Mazari & Kheddar, 2023) dan (Hasan dkk., 2023) untuk menggunakan model BERT telah membawa peningkatan yang berarti dalam kinerja model klasifikasi teks cyberbullying. Agar mudah melihat perbandingan performa model, hasil dua eksperimen dan evaluasi dua model yang diusulkan dalam penelitian ini dapat dilihat tabel 4.4.

Tabel 4. 6. Tabel Perbandingan Word Embedding+Bi-LSTM

Model	Data	Acc(%)	Precision (%)	Recall (%)	F1-Score(%)	Runtime (s)
FastText+Bi-LSTM	Imbalanced	0.9289	0.9199	0.9289	0.9151	61.53
	Balanced	0.8182	0.8249	0.8182	0.8180	28.32
BERT+Bi-LSTM	<b>Imbalanced</b>	<b>0.9317</b>	<b>0.9288</b>	<b>0.9317</b>	<b>0.9154</b>	<b>357.54</b>
	Balanced	0.8976	0.8976	0.8976	0.8976	97.44

Tabel perbandingan menampilkan hasil evaluasi dari FastText+Bi-LSTM dan BERT+Bi-LSTM, pada dua kondisi data yang berbeda pula, yaitu imbalanced dan balanced. Model FastText+Bi-LSTM menunjukkan kinerja yang sangat baik pada kedua kondisi data, dengan akurasi mencapai 92.89% untuk data yang tidak seimbang dan 81.82% untuk data yang seimbang. Sementara itu, model BERT+Bi-LSTM juga menunjukkan hasil yang lebih tinggi dengan akurasi sebesar 93.17% untuk data tidak seimbang dan 89.76% untuk data seimbang. Waktu pelatihan yang berbeda secara signifikan antara kedua model, di mana

BERT+Bi-LSTM membutuhkan waktu yang lebih lama dibandingkan FastText+Bi-LSTM, namun hasil evaluasi menunjukkan bahwa kinerja keduanya sebanding dalam menangani klasifikasi data yang kompleks.

Disisi lain, hasil yang diperoleh dari penelitian ini juga serupa dengan hasil penelitian (Al-Hashedi dkk., 2023) dan (Geet D'sa dkk., 2020) dimana penggunaan word embedding BERT memiliki akurasi terbaik dibandingkan word embedding yang lain. BERT+Bi-LSTM memiliki akurasi yang lebih unggul jika dibandingkan dari segi performa. Namun model FastText+Bi-LSTM dapat menjadi pilihan yang lebih efisien karena runtime yang lebih cepat dan komputasi yang kecil. Meskipun keduanya mengalami proses yang sama dalam pengolahan data dan tahapan eksperimen, perbedaan mendasar terletak pada penggunaan hyperparameter yang diatur secara otomatis. Penggunaan hyperparameter yang berbeda ini mengakibatkan bahwa model-model tersebut menemukan parameter terbaik yang berbeda untuk mencapai hasil terbaik, meskipun menggunakan data dan proses yang serupa. Model FastText+Bi-LSTM dalam penelitian ini menemukan parameter terbaiknya pada 'dropout': 0.2, 'epochs': 25, 'learning rate (lr)': 0.001, dan 'units': 512. Berbeda dengan penelitian sebelumnya (Riza & Charibaldi, 2021) yang menggunakan data lebih sedikit dan parameter manual yakni 'dropout': 0.5 dan 'epochs': 50 dan mendapati hasil tertinggi yaitu 73%. Sementara itu, model BERT+Bi-LSTM menemukan parameter terbaiknya pada 'dropout': 0.5, 'epochs': 100, 'learning rate (lr)': 0.01, dan 'units': 128. Berbeda dengan penelitian sebelumnya (Geet D'sa dkk., 2020) yang tidak menggunakan hyperparameter namun menggunakan parameter sequence length 256, batch size 16,

learning rate  $2 \cdot 10^{-5}$  dan 3 epochs. Penemuan parameter terbaik yang berbeda ini menunjukkan bahwa setiap model memiliki kebutuhan yang berbeda dalam hal tuning hyperparameter untuk mencapai kinerja optimal. Hal ini mungkin disebabkan oleh perbedaan arsitektur model, kompleksitas data, atau karakteristik tugas yang dihadapi. Model FastText+Bi-LSTM mungkin lebih sensitif terhadap dropout rendah dan jumlah unit yang besar, sementara model BERT+Bi-LSTM memerlukan dropout yang lebih tinggi dan jumlah unit yang lebih kecil untuk menghindari overfitting pada data yang lebih kompleks.

Arsitektur transformer pada BERT memungkinkan BERT untuk mengeksplorasi konteks dari kedua arah dalam sebuah kalimat dan menghasilkan representasi kata yang sangat informatif. Namun, arsitektur BERT yang besar dan kompleks memerlukan sumber daya komputasi yang besar untuk melatih dan menerapkan model. Proses pelatihan memakan waktu dan memerlukan GPU yang tinggi karena BERT memiliki lebih banyak parameter yang harus dipelajari selama proses pelatihan. Selain itu, penggunaan BERT juga bergantung pada jumlah data yang besar untuk melatihnya dengan baik, yang dapat menjadi hambatan dalam hal ketersediaan sumber daya dan waktu. Di sisi lain, FastText meskipun kurang rumit secara arsitektural dibandingkan dengan BERT, dapat bekerja dengan cepat karena pendekatannya yang lebih sederhana. FastText menggunakan representasi kata dengan n-gram yang memungkinkannya untuk bekerja secara efisien, terutama pada dataset yang besar. Namun, kelebihan kecepatan FastText datang dengan trade-off atas kurangnya representasi konteks yang dalam. FastText tidak mampu memahami hubungan antara kata-kata dalam



konteks kalimat secara menyeluruh seperti yang dilakukan oleh BERT. Perbedaan kecepatan atau lambatnya komputasi pada model BERT dan FastText disebabkan oleh perbedaan dalam arsitektur dan kompleksitas model mereka, di mana BERT memiliki kompleksitas yang lebih tinggi daripada FastText, yang mengakibatkan waktu komputasi yang lebih lama.

Teknik balancing data di penelitian ini menggunakan pustaka NumPy yakni `'np.random.random'`. Teknik balancing data menggunakan metode ini dimaksudkan untuk menyeimbangkan distribusi kelas dalam dataset (Kulkarni dkk., 2021). Dengan menggunakan pendekatan ini, terdapat risiko bahwa kelas minoritas akan diwakilkan oleh sampel yang tidak memadai, sementara kelas mayoritas mungkin diwakilkan oleh sampel yang berlebihan. Sebagai hasilnya, model yang dilatih dengan dataset yang dihasilkan oleh pendekatan `'np.random.random'` tidak mampu memahami pola atau karakteristik yang sebenarnya dari kelas-kelas minoritas, yang mengakibatkan kinerja kurang akurat dan kurang cerdas dalam memprediksi data baru. Pengurangan data secara acak menggunakan `'np.random.random'` mempengaruhi kinerja model secara keseluruhan. Oleh karena itu, dalam konteks balancing data, penting untuk menggunakan metode yang lebih cerdas dan cermat yang mempertimbangkan distribusi aktual dari kelas-kelas dalam dataset, untuk memastikan bahwa model yang dihasilkan mampu menggeneralisasi dengan baik dan memberikan prediksi yang lebih akurat.

Eksperimen dalam penelitian ini menunjukkan bahwa BERT+Bi-LSTM adalah model yang efektif untuk analisis sentimen dalam konteks deteksi cyber



bullying jika dibandingkan dengan kombinasi model FastText+Bi-LSTM. Penggunaan model BERT membawa pengaruh signifikan dalam segi performa yang tinggi juga didukung dengan hasil penelitian terdahulu (Y. Chen & Ling, 2023; Patil dkk., 2023; Sabbah & Hanani, 2023). Hasil penelitian yang diperoleh menjadi lebih baik dari penelitian sebelumnya karena mengikuti saran dari (Al-Hashedi dkk. 2023; Nurdin dkk. 2020; Riza dan Charibaldi 2021) yaitu untuk menambah jumlah dataset. Tantangan dalam mengumpulkan data dari media sosial membuat sulitnya mendapatkan data berkualitas dari platform sosial, yang pada gilirannya menghambat pengembangan pembelajaran mesin yang memerlukan data yang memadai (Hasan dkk., 2023). Pada akhirnya, hasil penelitian ini lebih baik menggunakan model yang diusulkan dalam penelitian ini dibandingkan penelitian sebelumnya (Setiawan dkk., 2023) dengan eksperimen menggunakan dataset yang sama.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Penelitian ini mengusulkan dan membandingkan dua model analisis sentimen untuk deteksi cyber bullying menggunakan teknik kombinasi word embedding dan Bi-LSTM. Berdasarkan hasil yang telah dicapai dalam penelitian Pengembangan Model Sentimen Analisis menggunakan kombinasi Word Embedding dan Bi-LSTM pada kasus cyber bullying, dapat disimpulkan beberapa hal sebagai berikut:

1. Dari hasil eksperimen menggunakan kombinasi FastText dengan Bi-LSTM pada klasifikasi teks cyber bullying, diperoleh hasil terbaik pada skenario pertama dengan akurasi sebesar 92.89%. Hyperparameter terbaik yang digunakan untuk model FastText+Bi-LSTM adalah dropout sebesar 0.2, 25 epochs, learning rate ( $lr$ ) sebesar 0.001, dan jumlah unit (units) pada layer LSTM sebesar 512. Hasil ini menunjukkan bahwa model ini mampu dengan baik dalam mengklasifikasikan teks cyber bullying dengan menggunakan representasi teks FastText dan arsitektur Bi-LSTM. Penggunaan dropout yang relatif rendah (0.2) dapat membantu dalam mencegah overfitting, sementara jumlah unit yang cukup besar pada layer LSTM (512) memungkinkan model untuk menangkap pola-pola kompleks dalam data teks.
2. Dari hasil eksperimen dengan mengkombinasikan BERT Embedding dengan Bi-LSTM pada klasifikasi teks cyber bullying, diperoleh hasil terbaik pada

skenario pertama dengan akurasi sebesar 93.17%. Hyperparameter terbaik untuk model BERT+Bi-LSTM adalah dropout sebesar 0.5, 100 epochs, learning rate (lr) sebesar 0.01, dan jumlah unit (units) pada layer LSTM sebesar 128. Hasil ini menunjukkan bahwa model ini mampu dengan sangat baik dalam mengklasifikasikan teks cyber bullying dengan menggunakan representasi teks BERT dan arsitektur Bi-LSTM. Penggunaan dropout yang relatif tinggi (0.5) dan jumlah epochs yang cukup besar (100) menunjukkan adanya upaya untuk mengatasi overfitting dan memungkinkan model untuk mempelajari pola-pola yang kompleks dalam data teks secara mendalam. Dengan demikian, kombinasi hyperparameter ini telah terbukti sangat efektif dalam meningkatkan kinerja model pada tugas klasifikasi teks cyber bullying.

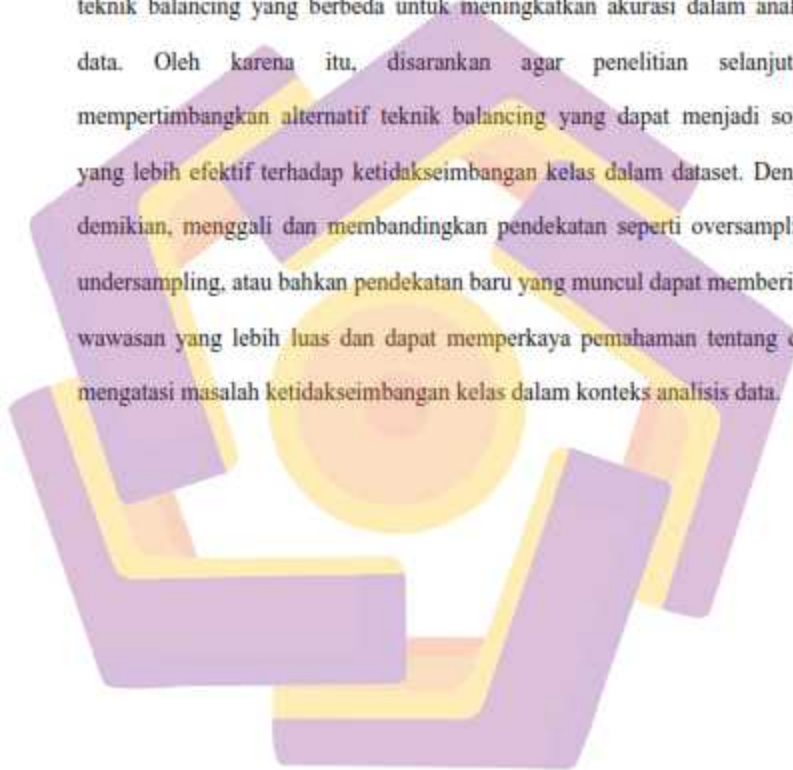
3. Berdasarkan data evaluasi performa model FastText+Bi-LSTM dan BERT+Bi-LSTM, ditemukan bahwa kedua model menunjukkan tingkat akurasi yang tinggi, dengan nilai berturut-turut yakni FastText+Bi-LSTM sebesar 92,89% dan BERT+Bi-LSTM sebesar 93,17%. Terdapat perbedaan kecil dalam nilai akurasi namun model BERT+Bi-LSTM menunjukkan sedikit peningkatan performa dibandingkan dengan FastText+Bi-LSTM. Meskipun model BERT+Bi-LSTM menunjukkan kinerja yang lebih baik, waktu komputasi yang dibutuhkan lebih besar (26 detik) dibandingkan dengan FastText+Bi-LSTM (4 detik). Diantara word embedding FastText dan BERT yang digunakan dalam penelitian ini, pada akhirnya word embedding yang diklasifikasi menggunakan Bi-LSTM menghasilkan kinerja yang tinggi dan lebih baik adalah BERT dengan akurasi sebesar 93,17%.

## 5.2. Saran

Meskipun kedua model yang diusulkan yakni FastText+ Bi-LSTM dan BERT+Bi-LSTM menunjukkan kinerja yang baik dalam dua scenario mengklasifikasikan teks cyber bullying, keduanya memiliki kelemahan yang perlu diperhatikan. Model FastText+Bi-LSTM sangat bergantung pada data pelatihan yang cukup besar dan bervariasi untuk menghasilkan representasi kata yang baik. Jika data pelatihan terbatas atau tidak mewakili dengan baik distribusi kata dalam data uji, maka performa model FastText dapat terpengaruh secara negatif. Model BERT+Bi-LSTM memerlukan sumber daya komputasi yang lebih besar karena kompleksitasnya. Hal ini terlihat dari jumlah epochs yang lebih tinggi (100) yang digunakan dalam pelatihan model ini dibandingkan dengan model FastText+Bi-LSTM. Penggunaan BERT juga memerlukan waktu komputasi yang lebih lama untuk pelatihan dan inferensi dibandingkan dengan metode seperti FastText. Dalam memilih model yang sesuai, perlu dipertimbangkan sumber daya yang tersedia, baik dalam hal data pelatihan yang mencukupi maupun kemampuan komputasi yang ada. Terlebih lagi, evaluasi cermat terhadap kebutuhan spesifik proyek serta ketersediaan sumber daya menjadi kunci dalam memilih model yang paling sesuai untuk kasus tertentu.

Selain itu, kedua model tersebut mungkin tidak sepenuhnya dapat mengatasi masalah kelas tidak seimbang atau adanya noise dalam data, yang dapat mempengaruhi kinerja dalam mengklasifikasikan teks cyber bullying

secara akurat. Oleh karena itu, dalam implementasi praktis, perlu diperhatikan penanganan kelas tidak seimbang yang tepat untuk meningkatkan kinerja model dalam menghadapi situasi yang lebih realistis. Dalam konteks pengembangan penelitian ini, pentingnya untuk melihat lebih lanjut pada teknik balancing yang berbeda untuk meningkatkan akurasi dalam analisis data. Oleh karena itu, disarankan agar penelitian selanjutnya mempertimbangkan alternatif teknik balancing yang dapat menjadi solusi yang lebih efektif terhadap ketidakseimbangan kelas dalam dataset. Dengan demikian, menggali dan membandingkan pendekatan seperti oversampling, undersampling, atau bahkan pendekatan baru yang muncul dapat memberikan wawasan yang lebih luas dan dapat memperkaya pemahaman tentang cara mengatasi masalah ketidakseimbangan kelas dalam konteks analisis data.





## DAFTAR PUSTAKA

- 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>
- AlHarbi, B. Y., AlHarbi, M. S., AlZahrani, N. J., Alsheail, M. M., & Ibrahim, D. M. (2020). Using machine learning algorithms for automatic cyber bullying detection in Arabic social media. *Journal of Information Technology Management*, 12(2), 123–130. <https://doi.org/10.22059/JITM.2020.75796>
- Al-Hashedi, M., Soon, L. K., Goh, H. N., Lim, A. H. L., & Siew, E. G. (2023). Cyberbullying Detection Based on Emotion. *IEEE Access*, 11, 53907–53918. <https://doi.org/10.1109/ACCESS.2023.3280556>
- Ansary, N. S. (2020). Cyberbullying: Concepts, theories, and correlates informing evidence-based best practices for prevention. Dalam *Aggression and Violent Behavior* (Vol. 50). Elsevier Ltd. <https://doi.org/10.1016/j.avb.2019.101343>
- 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, 453–462. <https://doi.org/10.1016/j.proeng.2013.02.059>
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). *Enriching Word Vectors with Subword Information*. <http://arxiv.org/abs/1607.04606>
- Cambria, E., Poria, S., Gelbukh, A., Nacional, I. P., & Thelwall, M. (2017). *AFFECTIVE COMPUTING AND SENTIMENT ANALYSIS Sentiment Analysis Is a Big Suitcase*. [www.computer.org/intelligent](http://www.computer.org/intelligent)

- Chen, Y., & Ling, J. (2023). *Online Twitter Bot Detection: A Comparison Study of Vectorization and Classification Methods on Balanced and Imbalanced Data*.
- Chen, Z., Jiang, C., Masood, M. K., Soh, Y. C., Wu, M., & Li, X. (2020). Deep learning for building occupancy estimation using environmental sensors. Dalam *Studies in Computational Intelligence* (Vol. 865, hlm. 335–357). Springer Verlag. [https://doi.org/10.1007/978-3-030-31760-7\\_11](https://doi.org/10.1007/978-3-030-31760-7_11)
- Cielen, D., Meysman, A. D. B., & Ali, M. (t.t.). *Introducing Data Science*.
- David, M. S., & Renjith, S. (2021). Comparison of word embeddings in text classification based on RNN and CNN. *IOP Conference Series: Materials Science and Engineering*, 1187(1), 012029. <https://doi.org/10.1088/1757-899x/1187/1/012029>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. <http://arxiv.org/abs/1810.04805>
- Dharma, E. M., Lumban Gaol, F., Leslie, H., Warnars, H. S., & Soewito, B. (2022). THE ACCURACY COMPARISON AMONG WORD2VEC, GLOVE, AND FASTTEXT TOWARDS CONVOLUTION NEURAL NETWORK (CNN) TEXT CLASSIFICATION. *Journal of Theoretical and Applied Information Technology*, 31(2). [www.jatit.org](http://www.jatit.org)
- Garrido-Merchan, E. C., Gozalo-Brizuela, R., & Gonzalez-Carvajal, S. (2023). Comparing BERT Against Traditional Machine Learning Models in Text

- Classification. *Journal of Computational and Cognitive Engineering*, 2(4), 352–356. <https://doi.org/10.47852/bonviewJCCE3202838>
- Geet D'sa, A., Illina, I., & Fohr, D. (2020). BERT and fastText Embeddings for Automatic Detection of Toxic Speech. *SIIE 2020-Information Systems and Economic Intelligence. Dalam Organization of Knowledge and Advanced Technologies.*  
<https://www.article19.org/data/files/medialibrary/3548/ARTICLE-19->
- Graves, A., & Schmidhuber, J. (t.t.). *Frame-wise Phoneme Classification with Bidirectional LSTM Networks.*
- Hasan, M. T., Hossain, M. A. E., Mukta, M. S. H., Akter, A., Ahmed, M., & Islam, S. (2023). A Review on Deep-Learning-Based Cyberbullying Detection. *Dalam Future Internet* (Vol. 15, Nomor 5). MDPI. <https://doi.org/10.3390/fi15050179>
- Ignatow, G., & Mihalcea, R. (2018). *An Introduction to Text Mining.*
- Jurafsky, D., & Martin, J. H. (t.t.). *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft Summary of Contents.*
- Kaur, M., & Saini, M. (2023). Indian government initiatives on cyberbullying: A case study on cyberbullying in Indian higher education institutions. *Education and Information Technologies*, 28(1), 581–615. <https://doi.org/10.1007/s10639-022-11168-4>
- Kulkarni, A., Batarseh, F. A., & Chong, D. (2021). *Chapter 5: Foundations of Data Imbalance and Solutions for a Data Democracy.*

- Madabushi, H. T., Kochkina, E., & Castelle, M. (2020). *Cost-Sensitive BERT for Generalisable Sentence Classification with Imbalanced Data*. <http://arxiv.org/abs/2003.11563>
- Markoulidakis, I., Rallis, L., Georgoulas, I., Kopsiaftis, G., Doulamis, A., & Doulamis, N. (2021). Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*, 9(4). <https://doi.org/10.3390/technologies9040081>
- Mazari, A. C., & Kheddar, H. (2023). Deep Learning-based Analysis of Algerian Dialect Dataset Targeted Hate Speech, Offensive Language and Cyberbullying. *International Journal of Computing and Digital Systems*, 13(1), 965–972. <https://doi.org/10.12785/ijcds/130177>
- Murshed, B. A. H., Abawajy, J., Mallappa, S., Saif, M. A. N., & Al-Ariki, H. D. E. (2022). DEA-RNN: A Hybrid Deep Learning Approach for Cyberbullying Detection in Twitter Social Media Platform. *IEEE Access*, 10, 25857–25871. <https://doi.org/10.1109/ACCESS.2022.3153675>
- Nurdeni, D. A., Budi, I., & Santoso, A. B. (2021). Sentiment Analysis on Covid19 Vaccines in Indonesia: From the Perspective of Sinovac and Pfizer. *3rd 2021 East Indonesia Conference on Computer and Information Technology, EIConCIT 2021*, 122–127. <https://doi.org/10.1109/EIConCIT50028.2021.9431852>
- Nurdin, A., Anggo, B., Aji, S., Bustamin, A., & Abidin, Z. (2020). PERBANDINGAN KINERJA WORD EMBEDDING WORD2VEC,



GLOVE, DAN FASTTEXT PADA KLASIFIKASI TEKS, *Jurnal TEKNOKOMPAK*, 14(2), 74.

Patil, A., Han, K., & Jadon, A. (2023). *A Comparative Study of Text Embedding Models for Semantic Text Similarity in Bug Reports*.  
<http://arxiv.org/abs/2308.09193>

Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2017). *Sentiment analysis in social networks*. Todd Green.

Raj, M., Singh, S., Solanki, K., & Selvanambi, R. (2022). An Application to Detect Cyberbullying Using Machine Learning and Deep Learning Techniques. *SN Computer Science*, 3(5). <https://doi.org/10.1007/s42979-022-01308-5>

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>

Roy, A., & Ojha, M. (2020, Desember 10). Twitter sentiment analysis using deep learning models. *2020 IEEE 17th India Council International Conference, INDICON 2020*. <https://doi.org/10.1109/INDICON49873.2020.9342279>

Sabbah, A. F., & Hanani, A. A. (2023). Self-admitted technical debt classification using natural language processing word embeddings. *International Journal of Electrical and Computer Engineering*, 13(2), 2142–2155. <https://doi.org/10.11591/ijece.v13i2.pp2142-2155>



- Setiawan, Y., Maulidevi, N. U., & Surendro, K. (2022). Deteksi Cyberbullying dengan Mesin Pembelajaran Klasifikasi (Supervised Learning): Peluang dan Tantangan. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 9(7), 1577. <https://doi.org/10.25126/jtiik.2022976747>
- Setiawan, Y., Maulidevi, N. U., & Surendro, K. (2023). The Use of Dynamic n-gram to Enhance TF-IDF Features Extraction for Bahasa Indonesia Cyberbullying Classification. *ACM International Conference Proceeding Series*, 200–205. <https://doi.org/10.1145/3587828.3587858>
- Siti Khomsah, Rima Dias Ramadhani, & Sena Wijaya. (2022). The Accuracy Comparison Between Word2Vec and FastText On Sentiment Analysis of Hotel Reviews. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 6(3), 352–358. <https://doi.org/10.29207/resti.v6i3.3711>
- Suhartono, D., Purwandari, K., Jeremy, N. H., Philip, S., Arisaputra, P., & Parmonangan, I. H. (2022). Deep neural networks and weighted word embeddings for sentiment analysis of drug product reviews. *Procedia Computer Science*, 216, 664–671. <https://doi.org/10.1016/j.procs.2022.12.182>
- Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., & Asadpour, M. (2020). Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, 7(1). <https://doi.org/10.1186/s40537-020-00349-y>
- V, V., & Adolf, H. P. D. (2021). Multimodal Cyberbullying Detection using Hybrid Deep Learning Algorithms. *International Journal of Applied*

<https://doi.org/10.37622/ijaer/16.7.2021.568-574>

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (t.t.). *Attention Is All You Need*.

Washington, E. T. (2015). An Overview of Cyberbullying in Higher Education. *Adult Learning*, 26(1), 21–27. <https://doi.org/10.1177/1045159514558412>

Watts, L. K., Wagner, J., Velasquez, B., & Behrens, P. I. (2017). Cyberbullying in higher education: A literature review. Dalam *Computers in Human Behavior* (Vol. 69, hlm. 268–274). Elsevier Ltd. <https://doi.org/10.1016/j.chb.2016.12.038>

Williams, L. A. (t.t.). *Pushing the Envelope of Sentiment Analysis Beyond Words and Polarities*.

Yudi Setiawan. (2023). Bahasa Cyberbullying Dataset (Source Data: Instagram, Twitter, and Youtube). Dalam *Mendeley Data*. Mendeley Data.