

SENTIMEN ANALISIS DENGAN ANOTASI OTOMATIS
MENGUNAKAN NAMED ENTITY RECOGNITION
DAN LEXICON BASED DICTIONARY

SKRIPSI

Fenni Kristiani Sarumaha

191402035



PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024

SENTIMEN ANALISIS DENGAN ANOTASI OTOMATIS
MENGUNAKAN NAMED ENTITY RECOGNITION
DAN LEXICON BASED DICTIONARY

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana
Teknologi Informasi

Fenni Kristiani Sarumaha

191402035



PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

MEDAN

2024

PERSETUJUAN

Judul : SENTIMEN ANALISIS DENGAN ANOTASI
OTOMATIS MENGGUNAKAN NAMED ENTITY
RECOGNITION DAN LEXICON BASED
DICTIONARY

Kategori : SKRIPSI

Nama : FENNI KRISTIANI SARUMAHA

Nomor Induk Mahasiswa : 191402035

Program Studi : S1 TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Medan, 10 Januari 2024

Komisi Pembimbing :

Medan,

Pembimbing 2



Ivan Jaya, M.Kom

NIP. 198407072015041001

Pembimbing 1




Dr. Erna Budhiarti Nababan M.IT

NIP. 196210262017042001

Diketahui/disetujui oleh

Program Studi S1 Teknologi Informasi

Ketua



Dedy Arisandi, ST., M.Kom.

NIP. 197908312009121002

PERNYATAAN**SENTIMEN ANALISIS DENGAN ANOTASI OTOMATIS
MENGUNAKAN NAMED ENTITY RECOGNITION
DAN LEXICON BASED DICTIONARY****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 10 Januari 2024

A handwritten signature in black ink, consisting of stylized, overlapping loops and a long horizontal stroke at the end.

Fenni Kristiani Sarumaha
191402035

UCAPAN TERIMA KASIH

Pertama-tama, penulis memanjatkan syukur kepada Tuhan Yang Maha Esa karena hanya oleh karena kemurahan dan karunia-Nya, memampukan penulis dalam menuntaskan penulisan skripsi ini sebagai persyaratan dalam meraih gelar Sarjana Komputer, di Program Studi S1 Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Penulis mempersembahkan skripsi ini kepada kedua orangtua tercinta, bapak Noi Tolo Sarumaha dan ibu Suci Hati Fau, yang tidak pernah putus dalam menyertai saya dengan doa, cinta dan dukungan yang hebat. Semoga Tuhan Yang Maha Esa selalu menyertai kedua orangtua saya, mengaruniakan mereeka kebahagiaan, kesehatan, rezeki dan umur yang panjang. Penulis juga berterima kasih kepada keempat adik penulis, yang bernama Handa, Yulianti, serta si kembar, Efraim, Efendi yang selalu memberi dukungan semangat, perhatian dan dorongan dalam proses penulisan skripsi ini.

Penulis menyadari bahwa perjalanan penelitian ini terwujud karena adanya bantuan dan dukungan dari beberapa pihak. Untuk itu dengan tulus dan rendah hati serta rasa hormat, penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Muryanto Amin, S.Sos., M.Si, sebagai Rektor Universitas Sumatera Utara.
2. Bapak Dr. Maya Silvi Lydia, B.Sc., M.Sc., yang menjabat sebagai Dekan Fasilkom-TI USU.
3. Bapak Dedy Arisandi, ST., M.Kom., selaku Ketua Program Studi S1 Teknologi Informasi di Universitas Sumatera Utara dan juga sebagai dosen penguji I dari penelitian yang saya garap.
4. Bapak Ivan Jaya, S.Si., M.Kom., sebagai Sekretaris Program Studi S1 Teknologi Informasi di Universitas Sumatera Utara dan juga sebagai Dosen Pembimbing II saya. Dimana, beliau meluangkan waktunya, memberikan pengarahan, kritik dan saran selama proses penulisan skripsi ini.

5. Ibu Dr. Erna Budhiarti Nababan M.IT, selaku Dosen Pembimbing I, yang mana beliau disela kesibukannya, rela meluangkan waktu, memberikan masukan dan kritik yang membangun bagi saya dalam hal penulisan dan penyelesaian skripsi ini.
6. Ibu Ade Sarah Huzaifah, S.Kom., M.Kom, selaku Dosen Penguji II dari penelitian yang saya lakukan.
7. Semua dosen dan staf, pegawai diProgram Studi Teknologi Informasi dan Fakultas Ilmu Komputer dan Teknologi Informasi, USU. Dimana, telah bersedia membantu kelancaran proses administrasi yang saya jalani selama masa perkuliahan.
8. Michael, Jason, Adib, Daniel, Meily, Vania, Arsyah, dan seluruh teman saya angkatan 2019 yang turut membantu dan mendukung saya selama penyelesaian skripsi ini.
9. Serta berbagai pihak yang ikut campur tangan membantu saya, baik secara langsung maupun tidak langsung, yang tidak dapat penulis sebutkan satu persatu.

Semoga Tuhan Yang Maha Esa melimpahkan berkat-Nya kepada semua pihak yang telah campur tangan dalam memberikan bantuan, buah pemikiran perhatian, dan dukungan semangat kepada penulis dalam menyelesaikan skripsi ini. Penulis sadar akan adanya kekurangan dalam skripsi ini dan dengan rendah hati, penulis mengharapkan masukan atau saran serta kritik yang baik dan membangun dari semua pihak untuk meningkatkan kualitas skripsi ini. Akhir kata, terima kasih atas segala dukungan yang diberikan.

Medan, 10 Januari 2024



Penulis

**SENTIMEN ANALISIS DENGAN ANOTASI OTOMATIS
MENGUNAKAN NAMED ENTITY RECOGNITION
DAN LEXICON BASED DICTIONARY**

ABSTRAK

Penyampaian opini di media sosial Twitter sering dilakukan dengan bahasa tidak formal. Hal ini menyulitkan analisis respon publik dan penarikan informasi pada twit. Sehingga diperlukan pendekatan untuk menganalisis sentimen dan menganotasi entitas pada data twit. Penelitian ini dilakukan dengan algoritma Bidirectional LSTM (BiLSTM) dan kamus InSet leksikon bahasa Indonesia. Performa model NER (BiLSTM) pada pengujian dihitung menggunakan evaluasi yang menghasilkan skor precision, recall, dan f1-score sebesar 98,85%. Hasil evaluasi manual terhadap sentimen analisis berbasis leksikon memperoleh akurasi 79,6%. Penemuan menunjukkan bahwa sentimen analisis dengan anotasi otomatis memudahkan penarikan informasi terhadap topik atau entitas yang dibicarakan, dan respon publik dapat dipahami dari nada sentimen twit pengguna Twitter.

Kata kunci : *Twitter*, *NER*, *BiLSTM*, Analisis Sentimen, Leksikon.

**SENTIMENT ANALYSIS WITH AUTOMATIC ANNOTATION
USING NAMED ENTITY RECOGNITION AND
LEXICON BASED DICTIONARY**

ABSTRACT

The delivery of opinions on the Twitter social media platform is often done using informal language, which complicates the analysis of public responses and the extraction of information from tweets. Therefore, an approach is needed to analyze sentiment and annotate entities in tweet data. This research was conducted using the Bidirectional LSTM (BiLSTM) algorithm and the InSet lexicon dictionary of the Indonesian language. The performance of the NER model (BiLSTM) in testing was evaluated, resulting in precision, recall, and f1-score scores of 98.85%. Manual evaluation of the lexicon-based sentiment analysis yielded an accuracy of 79.6%. The findings demonstrate that sentiment analysis with automatic annotation facilitates the extraction of information on the topics or entities being discussed, and public responses can be understood from the sentiment tone of Twitter users tweets.

Keywords: Twitter, NER, BiLSTM, Sentiment Analysis, Lexicon

DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN.....	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xii
DAFTAR CONTOH	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Penelitian	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	5
2.1 Twitter	5
2.3 Named Entity Recognition	6
2.4 Panduan Anotasi NER.....	7
2.5 Recurrent Neural Network (RNN)	10
2.6 Long Short Term Memory(LSTM)	11
2.7 Bidirectional LSTM	12
2.8 Word Embedding	13
2.9 Lexicon Based Method.....	14
2.10 Integrasi NER dan Lexicon Based Dictionary	16
2.11 Penelitian Terdahulu	16
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	19
3.1 Data yang Digunakan	19
3.2 Arsitektur Umum.....	22

3.2.1	<i>Input</i>	24
3.2.2	<i>Preprocessing</i>	25
3.2.3	<i>Pembagian Dataset</i>	42
3.2.4	<i>Word Embedding</i>	43
3.2.5	<i>Model Building</i>	45
3.2.6	<i>Klasifikasi Sentimen dengan Metode InSet Lexicon</i>	52
3.2.7	<i>Output</i>	54
3.3	Perancangan Aplikasi Sistem	54
3.3.1	<i>Tampilan Antar Muka Halaman Utama</i>	55
3.3.2	<i>Tampilan Antar Muka Halaman Preprocessing</i>	55
3.3.3	<i>Tampilan Antar Muka Halaman Training</i>	56
3.3.4	<i>Tampilan Antar Muka Halaman Testing</i>	57
3.3.5	<i>Tampilan Antar Muka Halaman User Input</i>	58
3.4	Metode Evaluasi	59
3.4.1	<i>Precision, Recall dan F1-score</i>	59
3.4.2	<i>Evaluasi Manual Dengan Akurasi</i>	60
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM		63
4.1	Implementasi Sistem	63
4.1.1	<i>Spesifikasi Perangkat Keras dan Perangkat Lunak</i>	63
4.1.2	<i>Implementasi Rancangan Antarmuka</i>	64
4.2	Implementasi Model	70
4.2.1	<i>Pelatihan Model</i>	70
4.3	Pengujian Model	72
4.4	Evaluasi	82
4.4.1	<i>Metrik Precision, Recall dan F1-Score</i>	82
4.4.2	<i>Evaluasi Manual Dengan Akurasi</i>	83
BAB V KESIMPULAN DAN SARAN		87
5.1	Kesimpulan	87
5.2	Saran	88
DAFTAR PUSTAKA		89

DAFTAR TABEL

Tabel 2.1 Twitter NER BIO Tagging	10
Tabel 2.2 Kamus InSet Lexicon Positif	14
Tabel 2.2 Kamus InSet Lexicon Positif(Lanjutan).....	15
Tabel 2.3 Kamus InSet Lexicon Negative	15
Tabel 2.4 Penelitian Terdahulu	17
Tabel 2.4 Penelitian Terdahulu(Lanjutan).....	18
Tabel 3.1 Dataset NER.....	19
Tabel 3.2 Token Kata dengan Tag PER	20
Tabel 3.3 Token Kata dengan Tag PROD.....	20
Tabel 3.4 Token Kata dengan Tag LOC	21
Tabel 3.5 Token Kata dengan Tag EV	21
Tabel 3.6 Token Kata dengan Tag WA.....	21
Tabel 3.6 Token Kata dengan Tag WA(Lanjutan).....	22
Tabel 3.7 Token Kata dengan Tag ORG.....	22
Tabel 3.8 Proses Data Selection	29
Tabel 3.9 Proses Case Folding	30
Tabel 3.10 Kata ke Indeks	32
Tabel 3.11 Tag ke Indeks	32
Tabel 3.12 Proses Create Data Dictionary	33
Tabel 3.13 Proses Emoticon Removal	36
Tabel 3.14 Proses Punctuation Removal.....	38
Tabel 3.15 Proses Tokenization	38
Tabel 3.16 Daftar Kata Baku dan Tidak Baku	39
Tabel 3.17 Keluaran Proses Normalisasi	40
Tabel 3.18 Proses Stopwords Removal.....	41
Tabel 3.19 Proses Stemming	42
Tabel 3.20 Perbandingan Data Training, Validasi dan Uji	43
Tabel 3.21 Input Pada Embedding Layer.....	46
Tabel 3.22 Output Embedding Layer.....	46

Tabel 3.23 Penerapan Model Building.....	48
Tabel 3. 24 Implementasi Metode Lexicon Based Dictionary	54
Tabel 3.25 Sentimen Analisis dan Anotasi Otomatis.....	54
Tabel 3. 26 Profil Anotator(Pakar).....	61
Tabel 4.1 Kalimat NER Data Uji	72
Tabel 4.1 Kalimat NER Data Uji(Lanjutan)	73
Tabel 4.2 Kata ke Indeks Data Uji	74
Tabel 4.3 Konversi Tag ke Indeks Data Uji.....	74
Tabel 4.3 Konversi Tag ke Indeks Data Uji(Lanjutan)	75
Tabel 4.4 Data Uji (Stemming)	75
Tabel 4.5 Pengujian Model Pada Data Uji Pertama.....	76
Tabel 4.6 Pengujian Model Pada Data Uji Kedua.....	77
Tabel 4.7 Pengujian Model Pada Data Uji Ketiga	77
Tabel 4.7 Pengujian Model Pada Data Uji Ketiga(Lanjutan)	78
Tabel 4.8 Pengujian Model Pada Data Uji Keempat.....	78
Tabel 4.8 Pengujian Model Pada Data Uji Keempat(Lanjutan).....	79
Tabel 4.9 Pengujian Model Pada Data Uji Kelima	79
Tabel 4.9 Pengujian Model Pada Data Uji Kelima(Lanjutan)	80
Tabel 4.10 Prediksi NER.....	80
Tabel 4.10 Prediksi NER(Lanjutan)	81
Tabel 4. 11 Hasil Sentimen Analisis Kamus Leksikon Bahasa Indonesia.....	81
Tabel 4. 11 Hasil Sentimen Analisis Kamus Leksikon Bahasa Indonesia(Lanjutan) .	82
Tabel 4.12 Precision, Recall dan F1-Score(Data Uji)	83
Tabel 4. 13 Perbandingan Sentimen Analisis Otomatis dan Manual	84

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Umum RNN (Amidi, 2018)	10
Gambar 2.2 Arsitektur BiLSTM (Permana & Purnamasari, 2019).....	13
Gambar 2. 3 Konsep COBW dan Skip-gram (Mikolov, Corrado, Chen, & Dean, 2013)	14
Gambar 3.1 Diagram Alir Sentimen Analisis dengan Anotasi Otomatis Text Twitter	23
Gambar 3.2 Proses Preprocessing NER	25
Gambar 3.3 Proses Preprocessing Sentimen Analisis	27
Gambar 3. 4 Pseudocode proses Data Selection	28
Gambar 3.5 Pseudocode proses Case Folding.....	30
Gambar 3.6 Pseudocode Proses Create Unique Dataset	31
Gambar 3. 7 Pseudocode Proses Create Data Dictionary.....	33
Gambar 3. 8 Pseudocode Proses Create Data Dictionary.....	34
Gambar 3. 9 Pseudocode Proses Emoticon Removal.....	36
Gambar 3. 10 Pseudocode Proses Punctuation Removal	37
Gambar 3. 11 Pseudocode Proses Tokenization.....	38
Gambar 3. 12 Daftar Kata Baku dan Tidak Baku	39
Gambar 3. 13 Pseudocode Proses Stopwords Removal	40
Gambar 3.14 Pseudocode Proses Stemming	41
Gambar 3.15 Perbandingan Data Latih, Data Validasi dan Data Uji.....	43
Gambar 3.16 Rincian Model BiLSTM.....	45
Gambar 3. 17 Pseudocode Fungsi Optimasi Adam.....	49
Gambar 3.18 Grafik Loss Dengan Early Stopping.....	51
Gambar 3.19 Grafik Loss Tanpa Early Stopping	52
Gambar 3.20 Alur Klasifikasi Sentimen InSet Lexicon	53
Gambar 3.21 Tampilan Rancangan Halaman Utama	55
Gambar 3.22 Tampilan Rancangan Halaman Preprocessing	56
Gambar 3.23 Tampilan Rancangan Halaman Training	57
Gambar 3.24 Tampilan Rancangan Halaman Testing.....	58

Gambar 3.25 Tampilan Rancangan Halaman User Input.....	58
Gambar 4.1 Tampilan Halaman Utama.....	64
Gambar 4.2 Tampilan Halaman Preprocessing	65
Gambar 4.3 Tampilan Halaman Proses Preprocessing	65
Gambar 4.4 Tampilan Halaman Hasil Preprocessing	66
Gambar 4.5 Tampilan Halaman Hasil Preprocessing.....	66
Gambar 4.6 Tampilan Halaman Training.....	67
Gambar 4.7 Tampilan Data Hasil Training	67
Gambar 4.8 Grafik Hasil Proses Training (Akurasi dan Loss)	68
Gambar 4.9 Tampilan Halaman Testing	68
Gambar 4.10 Tampilan Halaman Hasil Testing	69
Gambar 4.11 Tampilan Slide Confusion Matrix	69
Gambar 4.12 Tampilan Halaman User Input	70
Gambar 4.13 Tampilan Halaman User Input(Hasil)	70
Gambar 4.14 Grafik Akurasi	71
Gambar 4.15 Grafik Loss	72
Gambar 4.16 Perbandingan Sentimen Analisis oleh Sistem dan Pakar	84
Gambar 4.17 Confusion Matrix Sentimen Analisis	85

DAFTAR CONTOH

Contoh 1 Twit dengan Tag PROD	9
Contoh 2. Data yang menjadi data input ke dalam sistem	24
Contoh 3. Kalimat twit	24
Contoh 4. Tag NER.....	24
Contoh 5 Kalimat NER	35
Contoh 6 Daftar kata stopwords.....	40
Contoh 7 Embedding Matriks Kata ‘Hidup’	44
Contoh 8 Pasangan Label NER Numerik dan String	49

BAB I

PENDAHULUAN

1.1 Latar Belakang

Peralihan media sosial dari alat komunikasi menjadi media penyampaian opini publik terhadap suatu topik permasalahan mengalami peningkatan pesat. Terutama media sosial, Twitter yang pada tahun 2022, jumlah penggunaannya di Indonesia mencapai 19,5 juta orang pengguna (Annur, 2022). Sehingga Twitter dapat dimanfaatkan untuk menarik banyak data teks yang tujuannya untuk melakukan analisis data teks. Hasil analisis teks dapat berupa ekstraksi informasi untuk mengetahui topik yang sedang dibicarakan dan analisis sentimen terhadap opini yang disampaikan. Akan tetapi pengguna Twitter dalam menulis *tweet* sering menggunakan bahasa tidak formal atau disebut juga bahasa gaul. Sehingga dalam melakukan ekstraksi informasi berupa entitas pada data twit menjadi hal yang sulit untuk dilakukan. Oleh sebab itu, diharapkan adanya suatu solusi pendekatan untuk melakukan analisis data teks pada data Twitter.

Untuk memperoleh informasi dan juga analisis sentimen dari opini yang diberikan oleh pengguna Twitter ada beberapa hal yang harus dilakukan. Pertama, untuk menganalisis data teks diperlukan proses ekstraksi entitas dengan menggunakan suatu pendekatan yaitu Named entity Recognition(NER). NER merupakan suatu teknik dalam *Natural Language Processing*(NLP) untuk mengidentifikasi dan juga melakukan klasifikasi penamaan entitas dalam suatu kumpulan teks sesuai dengan kategorinya, misalnya nama orang, organisasi, nama tempat, waktu dan lain-lain (Li S. , 2018). Lalu, selanjutnya dalam memahami sentimen pada opini pengguna Twitter, maka perlukan dilakukan proses analisis sentimen analisis pada Twitter.

Ada beberapa penelitian terdahulu yang relevan dengan pendekatan NER pada data teks. Pada tahun 2018, penelitian yang mengangkat topik pendekatan NER pada data *tweet* berbahasa Indonesia dengan pemanfaatan fitur POS(Part Of Speech) *tagger* dan fitur khusus *tweet*, lalu diproses menggunakan algoritma Multinomial Naive Bayes Classifier (Rifani, Bijaksana, & Asror, 2019). model NER yang dibangun memiliki tingkat akurasi cukup baik untuk mengidentifikasi entitas pada data *tweet*. Penelitian yang dilakukan di China terhadap anotasi teks secara otomatis pada dataset teroris menggunakan algoritma Named Entity Recognition(NER) memperoleh hasil bahwa, NER mampu menganotasi secara otomatis kumpulan data dalam klasifikasi teks (Xin et.al, 2019). Namun penelitian tersebut dilakukan pada dataset teks berita. Dimana, dataset teks berita merupakan data teks yang *well-edited* sehingga mudah dilakukan proses ekstraksi entitas. Selanjutnya ada penelitian pada tahun 2018 dengan topik yang sama, menggunakan algoritma BiLSTM-CRF mampu menghasilkan suatu dataset NER Twitter yang baru (Ushio, Neves, Silva, & Barbieri, 2020).

Selanjutnya penelitian yang mengangkat topik sentimen analisis dilakukan pada tahun 2021 (Prasetya, Winarso, & Syahril, 2021). Penelitian ini mengangkat permasalahan mengenai sentimen analisis terhadap tingkat level kepercayaan masyarakat pada isu Covid-19. Objek penelitian ini adalah pengguna Twitter dan dilakukan dengan menggunakan kamus leksikon Indonesia sebagai metode untuk menganalisis sentimen. Selain itu, ada juga penelitian yang mengangkat permasalahan sentimen analisis pada kampanye pemilihan umum menggunakan algoritma SVM yang dikombinasikan dengan kamus leksikon (Apsari, 2018). Hasilnya, algoritma SVM lebih unggul untuk menganalisis sentimen pada data Twitter dibandingkan dengan Lexicon Based Dictionary. Selain itu ada juga penelitian pada tahun 2017 menggunakan Lexicon Based Dictionary yang juga untuk menganalisa sentimen pada data tweet opini masyarakat terhadap produk makanan (Nurfalah, Adiwijaya, & Suryani, 2017).

Berdasarkan penelitian terkait yang sudah dilakukan sebelumnya, penulis memutuskan untuk melakukan kolaborasi antara Named Entity Recognition dan Lexicon Based untuk diajukan sebagai judul dalam penelitian ini: **Sentimen Analisis Dengan Anotasi Otomatis Menggunakan Named Entity Recognition Dan Lexicon Based Dictionary.**

1.2 Rumusan Masalah

Dalam mengekstraksi informasi pada data twit, diperlukan ekstraksi entitas bernama untuk memperoleh entitas penting dalam teks. Selain informasi tersebut, diperlukan yaitu sentimen analisis data tweet untuk memahami nada emosional positif, negatif, maupun netral terhadap topik yang dibicarakan. Akan tetapi twit yang dituliskan oleh pengguna dengan bahasa informal menyebabkan ekstraksi informasi menjadi lebih sulit. Sehingga diperlukan suatu pendekatan untuk memperoleh entitas bernama pada dan melakukan proses analisis sentimen terhadap data twit.

1.3 Tujuan Penelitian

Dari rumusan masalah yang ada, penelitian ini bertujuan untuk melakukan klasifikasi sentimen analisis dengan menganotasi otomatis opini masyarakat di media sosial Twitter. Metode yang digunakan adalah kombinasi pendekatan Named Entity Recognition(BiLSTM) dan Lexicon Based Dictionary.

1.4 Batasan Penelitian

Berdasarkan penjelasan yang telah dipaparkan pada rumusan masalah, agar penelitian ini berjalan efisien dan tepat sasaran, maka, diberikan pembatasan dalam penelitian ini, yaitu:

- 1) Penelitian ini hanya berfokus pada analisis sentimen dan anotasi entitas pada media sosial Twitter.
- 2) Jenis entitas yang akan diproses adalah lokasi(LOC), produk(PROD), *event*(EV), *Work of Art*(WA), *person*(PER) dan *organization*(ORG)
- 3) Implementasi BiLSTM dalam anotasi *Named Entity Recognition* dan Kamus *Lexicon* bahasa Indonesia.

1.5 Manfaat Penelitian

Dalam penelitian ini, diharapkan dapat bermanfaat untuk menghasilkan klasifikasi sentimen analisis dengan anotasi teks Twitter secara otomatis menggunakan *Named Entity Recognition* dan *Lexicon Based Dictionary*.

1.6 Sistematika Penulisan

Adapun sistematika penulisan didalam penelitian ini terdiri atas lima bagian utama, antara lain:

Bab 1: Pendahuluan

Pada bagian ini menjelaskan mengenai latar belakang diadakannya penelitian, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan.

Bab 2: Landasan Teori

Bagian ini menjelaskan hal-hal yang berkenaan dengan teori dasar dari permasalahan dan tujuan yang ingin dicapai pada penelitian ini. Selain itu, pada bagian ini juga dijelaskan secara ringkas tentang penelitian sebelumnya yang relevan dengan penelitian ini.

Bab 3: Analisis dan Perancangan Sistem

Pada bagian ini berisi uraian secara rinci dalam melakukan analisis kasus dalam penelitian dan pembuatan rancangan dari kombinasi algoritma BiLSTM untuk *Named Entity Recognition* dan kamus InSet *Lexicon* bahasa Indonesia dalam melakukan sentimen analisis dengan anotasi otomatis pada cuitan masyarakat dimedia sosial Twitter.

Bab 4: Implementasi dan Pengujian Sistem

Bagian berisi tentang penjelasan dari penerapan atau implementasi dari serangkaian rancangan yang telah disusun pada Bab 3. Selain itu, akan dibahas juga tentang keluaran dari pengujian sistem yang dibangun.

Bab 5: Kesimpulan dan Saran

Bagian terakhir dari penelitian ini menjelaskan tentang hal-hal terkait jawaban permasalahan dan tercapai atau tidaknya tujuan dari penelitian ini. Pada bagian ini berisi kesimpulan yang dapat ditarik dari penelitian yang telah dilakukan serta memberikan saran agar kuliatis penelitian bisa berkembang.

BAB II

LANDASAN TEORI

2.1 Twitter

Merupakan salah satu media sosial yang berfungsi sebagai *platform* bagi masyarakat dalam memberikan pernyataan ataupun opininya tentang suatu topik. Menurut data yang diperoleh dari website Datakita pada bulan Januari tahun 2022, jumlah pengguna Twitter mencapai 19,5 juta (Annur, 2022). Hal ini menjadikan banyak kegiatan penelitian yang bertujuan untuk menganalisis respon atau opini publik yang menjadikan Twitter sebagai sumber penelitiannya.

2.2 Sentimen Analisis

Sentimen analisis merupakan bagian dari penelitian dari suatu bidang ilmu yang dikenal dengan nama, Pemrosesan Bahasa Alami (Medhat, Hassan, & Korashy, 2014). PBA merupakan suatu kategori kecerdasan buatan yang dapat digunakan untuk proses pemahaman, penafsiran maupun proses manipulasi terhadap bahasa manusia dengan memanfaatkan bahasa alami sebagai alat interaksi komputer dan manusia. Menurut Munarnawan & Sinaga (2017, p.110) sebagaimana dikutip dalam (Liu, 2012) analisis sentimen adalah analisis opini orang-orang yang terkomputerisasi terhadap entitas berupa isu permasalahan. Didalamnya dilakukan deteksi emosi dari opini yang dituliskan, lalu dikelompokkan kedalam kategori nilai positif, negatif dan netral.

Dalam pengerjaan sentimen analisis, beberapa pendekatan yang dapat diterapkan adalah menggunakan metode berbasis leksikon, pembelajaran mesin, serta deep learning. Semuanya memiliki keunggulan dan kelemahan. Untuk mengatasi hal tersebut, sering dilakukan kolaborasi antara beberapa metode (Thomas, Yuliana, & Noviyanti. P, 2021).

Sentimen analisis dengan menggunakan lexicon-based dilakukan dengan cara memanfaatkan daftar kata-kata yang dibentuk menjadi kamus kata yang sudah diberi label sentimen untuk kemudian diterapkan pada klasifikasi sentimen teks (Thomas, Yuliana, & Noviyanti. P, 2021). Teknik ini terbilang sederhana namun kurang akurat

untuk menghasilkan hasil sentimen yang kompleks (Sentiment Analysis, n.d.). Metode sentimen analisis dengan menggunakan pendekatan pembelajaran mesin bekerja dengan cara mengimplementasikan algoritma pada data latih yang besar mampu menghasilkan hasil sentimen yang lebih kompleks akan tetapi, membutuhkan waktu yang lama (Sentiment Analysis, n.d.). Metode sentimen analisis menggunakan *deep learning* memanfaatkan jaringan saraf tiruan dalam menghasilkan klasifikasi sentimen dengan akurasi yang tinggi pada kumpulan data teks, namun sama halnya dengan metode pembelajaran mesin, *deep learning* membutuhkan data latih yang besar (Sentiment Analysis, n.d.). Terakhir, sentimen analisis dengan menggunakan gabungan antara ketiga teknik tersebut, misalnya sentimen analisis dengan *lexicon-based* dan pembelajaran mesin. Pemilihan metode untuk melakukan sentimen analisis tergantung pada sumber data maupun tujuan dari penelitian yang ingin diraih peneliti.

2.3 Named Entity Recognition

Menurut Novi et.al (2021), Named Entity Recognition(NER) merupakan pendekatan yang digunakan untuk mengenali suatu objek dalam suatu teks untuk keperluan pemrosesan bahasa alami. Pada awalnya, terminologi “*named entity*” mengarah pada entitas nama orang, organisasi, dan lokasi geografis yang ada pada sebuah teks (Putra & Hidayatullah, 2021). Hal-hal yang dapat dilakukan dengan menggunakan NER adalah mendukung sejumlah Pemrosesan Bahasa Alami (NLP), seperti menjawab pertanyaan, pencarian semantik, dan terjemahan mesin.

Ada beberapa teknik pendekatan yang dapat diimplementasikan untuk membangun Named Entity Recognition, yaitu mencakup teknik *rule-based*, *statistical*, dan *deep learning* (Li, Han, & Li, 2020). NER dengan menggunakan teknik berbasis aturan mengidentifikasi entitas teks dengan cara membuat serangkaian aturan (Sari, Hassan, & Zamin, 2010). Metode *statistical* pada NER dilakukan dengan memanfaatkan model statistik untuk klasifikasi entitas bernama pada data teks (Li, Han, & Li, 2020). Terakhir, NER dapat dilakukan dengan metode *deep learning* dengan menggunakan cara kerja *deep learning* yaitu memanfaatkan jaringan syaraf tiruan untuk pemrosesan data yang lebih baik dan menghasilkan keluaran yang lebih akurat (Li, Han, & Li, 2020). Dalam melakukan ekstraksi entitas (NER) ketiga metode ini dapat dikolaborasikan untuk meningkatkan akurasi terhadap keluaran yang diinginkan (Tilbe, 2022).

2.4 Panduan Anotasi NER

Menurut (Farihin, 2022) anotasi NER terhadap data Twitter dilakukan setelah membandingkan beberapa literatur, diantaranya adalah sebagai berikut.

- 1) Panduan anotasi NER formal dalam bahasa Inggris (Weischedel, et al., 2017)
- 2) Panduan anotasi NER data Twitter dalam bahasa Inggris (Chinchor & Robinson, 1998; Ritter et al., 2011)
- 3) Serta, panduan anotasi NER formal dalam bahasa Indonesia (Taufik et al., 2016).

Dari ketiga literatur tersebut dihasilkan enam kelas kategori *named entities* yaitu sebagai berikut.

1) PER (Person/Orang)

- a. Tag ini diberikan terhadap token kata yang termasuk nama depan, belakang, ataupun tengah seseorang. Misalnya, "Rudi Santoso" sebagai contoh nama yang dapat mencakup unsur depan, belakang, atau tengah pada identitas seseorang.
- b. Tag ini diberikan terhadap token kata yang termasuk nama dari karakter fiksi manusia. Sebagai contoh, karakter fiksi manusia, "Arya Stark" dari serial Game of Thrones.
- c. Tag ini diberikan terhadap token kata yang termasuk nama samaran, nama yang termasuk inisial ataupun nama lain yang merujuk pada identitas seseorang. Sebagai contoh, "DJ" (singkatan dari Dewi Juwita).

2) ORG (Organization/Organisasi)

- a. Tag ini diberikan terhadap token kata yang termasuk nama dari sebuah instansi, perusahaan, agensi, ataupun institusi swasta. Sebagai contoh, "Tokopedia" dan "Tesla" yang merupakan perusahaan yang bergerak dalam kegiatan di sektor bisnis.
- b. Tag ini diberikan terhadap token kata yang termasuk badan pemerintahan dengan fungsi dan tugas tertentu. Seperti, Badan Pemeriksa Keuangan(BPK), Badan Usaha Milik Negara(BUMN), dan Badan Intelijen Negara(BIN).

- c. Tag ini diberikan terhadap token kata yang termasuk nama dari organisasi yang berdiri berdasarkan pandangan tertentu seperti politik, keagamaan atau kemasyarakatan. Misalnya, Partai Keadilan Sejahtera(PKS), “Muhammadiyah”, dan “Pertamina”.
- d. Tag ini diberikan terhadap token kata yang termasuk nama dari sebuah grup musik atau *band* atau *orchestra*. Misalnya, grup musik "Coldplay".
- e. Tag ini diberikan terhadap token kata yang termasuk nama dari komunitas atau sebuah klub dengan tujuan tertentu. Sebagai contoh, "Bikers Brotherhood". Terdapat juga sebuah klub suporter sepak bola, “Bonek”, dll.

3) LOC(Location/Lokasi)

- a. Tag ini diberikan terhadap token kata yang termasuk nama daerah. Sebagai contoh, "Jawa Barat" sebagai provinsi di Indonesia, "New York City" sebagai kota di Amerika Serikat, dan "Cilandak" sebagai kecamatan di Jakarta.
- b. Tag ini diberikan terhadap token kata yang termasuk lokasi pemandangan alam. Sebagai contoh, "Danau Toba" yang merupakan lokasi pemandangan alam dari sebuah danau di Sumatra Utara.
- c. Tag ini diberikan terhadap token kata yang termasuk kepada fasilitas publik buatan. Contohnya adalah kata “Monas” yaitu kependekan dari bangunan fasilitas publik Monumen Nasional di Jakarta.
- d. Tag ini diberikan terhadap token kata yang termasuk kepada fasilitas publik yang bersifat komersial, baik dibidang pendidikan, budaya, kesehatan maupun transportasi. Sebagai contoh, token kata “SMA 1 Jakarta” sebagai sekolah menengah atas di Jakarta.

4) PROD (Product/Produk)

- a. Tag ini diberikan terhadap token kata yang termasuk nama dari sebuah produk. Sebagai contoh dari kategori ini adalah "iPhone 13 Pro" sebagai nama dari tipe produk yang dijual oleh perusahaan Apple.
- b. Tag ini diberikan terhadap token kata dalam twit yang memiliki keterangan kata “PROD” yang mengikuti dari suatu nama produk. Sebagai contoh kutipan twit, "Samsung Galaxy S21 PROD" dapat menjadi contoh, di mana kata "**PROD**" berfungsi sebagai penanda yang mengindikasikan keterangan atau identifikasi spesifikasi produk tertentu.

Contoh 1 Twit dengan Tag PROD

Contoh twit: “Jual laptop [asusx441Nram4gb]_{PROD} 2.5 fullset 08978000654”

Adapun kata "laptop" tidak diikutsertakan dalam label sebagai tag “PROD”, disebabkan oleh sifatnya sebagai kata benda umum atau *common noun*. Dimana kata tersebut tidak memberikan deskripsi dari produk tersebut. Hal yang sama berlaku terhadap elemen "2.5" yang juga diabaikan dalam anotasi karena merupakan nilai harga dari produk, yang bukan merupakan informasi deskriptif terkait produk. Sementara itu, frasa "RAM 4GB" diikutsertakan dalam label karena merinci spesifikasi produk. Frasa ini dianggap relevan karena mengindikasikan informasi label.

5) WA (Work of Art/Karya Seni)

Tag ini diberikan terhadap token kata yang termasuk nama dari sebuah karya mencakup karya seni seperti, nama dari judul buku, lagu, film, acara TV, nama karya, dan lain sebagainya.

6) EV (Event/Acara)

Tag ini diberikan terhadap token kata yang termasuk nama dari suatu peristiwa atau kejadian, baik yang terencana maupun tidak. Sebagai contoh kejadian yang tidak terencana adalah "Gempa Bumi Lombok 2018". Merupakan kejadian tidak terencana karena termasuk sebagai bencana alam. Contoh dari kejadian yang terencana seperti, "Pemilu Presiden 2024" adalah kejadian yang terencana dan dijadwalkan secara politik untuk pemilihan kepala negara.

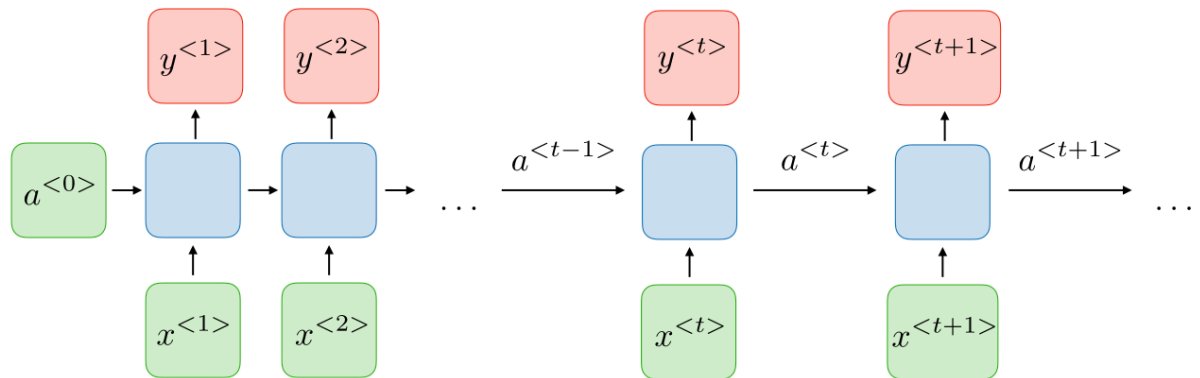
Selain itu pelabelan NER menggunakan format *Beginning(B)*, *Inside(I)*, *Other(O)* atau dikenal dengan format BIO. Dimana “B” adalah awal dari entitas, “I” adalah entitas perantara atau yang masih memiliki keterkaitan terhadap entitas sebelumnya dan “O” merupakan label yang diberikan terhadap token kata yang tidak termasuk entitas. Tujuan anotasi dengan format BIO ini adalah untuk mengatasi permasalahan ambiguitas dalam penentuan batas tanda baca atau penggunaan frasa (Wibisono & Khodra, 2018). Anotasi entitas NER pada data twit menggunakan format BIO dapat dilihat pada tabel 2.1.

Tabel 2.1 Twitter NER BIO Tagging

Tweet	Tags
Kementrian	B-ORG
keuangan	I-ORG
menyatakan	O
inflasi	O
di	O
Indonesia	B-LOC
rendah	O

2.5 Recurrent Neural Network (RNN)

Dalam mengolah data berutan(*sequence*) dapat digunakan dengan teknik RNN. Cara kerja RNN adalah dengan menerima input dari *vector sequence* ($x_0, x_1, x_2, \dots, x_n$) menjadi bentuk *sequence* yang lain ($h_0, h_1, h_2, \dots, h_n$) (Rusliani, 2017). Gambaran mengenai perancangan RNN dapat dilihat pada Gambar 2.1.

**Gambar 2.1** Arsitektur Umum RNN (Amidi & Amidi, 2018)

Penjelasan mengenai Gambar 2.1 adalah setiap *timestep*(t) terdapat input($x^{<t>}$) dan aktivasi fungsi ($a^{<t>}$) serta $y^{<t>}$ sebagai output. Walau demikian, RNN memiliki kelemahan yaitu tidak mampu untuk menyelesaikan permasalahan *long-term dependency* yang menimbulkan masalah *vanishing gradient*. Hal ini karena, RNN cenderung sulit dalam mempertahankan informasi yang relevan dari waktu yang dulu ke waktu yang lebih baru (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016). Untuk mengatasi permasalahan *vanishing gradient* maka dikembangkan sebuah metode yang kini dikenal dengan *Long Short Term Memory*(LSTM).

2.6 Long Short Term Memory(LSTM)

Cara kerja dari LSTM adalah dengan menggunakan *memory block* yang terdiri dari 4 komponen. Adapun penjelasan disetiap bagian blok memori dari cara kerja LSTM adalah sebagai berikut

1) *Input Gate*

Input gate digunakan untuk melakukan perubahan dalam memori sel (C_t) dengan W_i dan U_i yang adalah matriks bobot untuk dilakukan operasi perkalian dengan x_t dan h_{t-1} . Rumus dari *input gate* dapat dilihat pada persamaan 2.1.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad 2.1$$

2) *Forget Gate*

Bagian ini digunakan untuk menghapus dan melupakan hal-hal yang diperoleh dari memori sel (C_t). Fungsi sigmoid (σ) digunakan untuk menghitung informasi yang diperoleh sebelumnya dari *hidden state* (h_{t-1}) dan *input* (x_t). Rumus dari *forget gate* dapat dilihat pada persamaan 2.2.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad 2.2$$

3) *Memory Cell*

Bagian ini, memori sel dikalikan dengan hasil dari perhitungan di *forget gate* ($f_t \odot c_{t-1}$). Lalu selanjutnya nilai dari hasil perkalian tersebut, dikalikan lagi dengan nilai memori sel saat ini. Tujuannya untuk memperoleh nilai memori sel yang baru sebagai nilai memori sel saat ini. Nilai dari memori sel memiliki rentang antara 1 dan -1. Rumus dari memori sel dapat dilihat pada persamaan 2.3.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad 2.3$$

4) *Output Gate*

Rumus dari perhitungan *output gate* digambarkan pada persamaan 2.4

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad 2.4$$

Dimana, *Output gate* memiliki kesamaan dengan bagian *input gate*, yaitu sama-sama digunakan untuk menghitung dan menentukan nilai dari *hidden state* yang baru (h_t). Adapun perhitungan untuk menentukan *hidden state* adalah sebagai berikut pada persamaan 2.5.

$$h_t = o_t \odot \tanh(c_t) \quad 2.5$$

Dengan:

$i_t = \text{input gate}$

$U_i = \text{matriks bias yang lain}$

$f_t = \text{forget gate}$

$c_t = \text{cell gate}$

$o_t = \text{output gate}$

$h_t = \text{hidden state}$

$b = \text{nilai vektor bias}$

$\odot = \text{elemen wise production}$

2.7 Bidirectional LSTM

BiLSTM adalah sebuah algoritma yang memungkinkan model untuk mempelajari informasi dari dua arah kiri dan kanan untuk memprediksi data saat ini (Rachman V. , Septiviana, Augustianti, & Mahendra, 2017). BiLSTM menggunakan dua *hidden layer* secara terpisah, yaitu *forward layer* untuk merepresentasikan konteks sebelumnya dan *backward layer* sebagai representasi konteks selanjutnya (Permana & Purnamasari, 2019). Keluaran dari kombinasi dua arah *hidden layer* \vec{h}_t dan \overleftarrow{h}_t dijelaskan pada persamaan 2.9.

$$y_t = W_{\vec{h}_y} \vec{h}_t + W_{\overleftarrow{h}_y} \overleftarrow{h}_t \quad 2.8$$

Dengan:

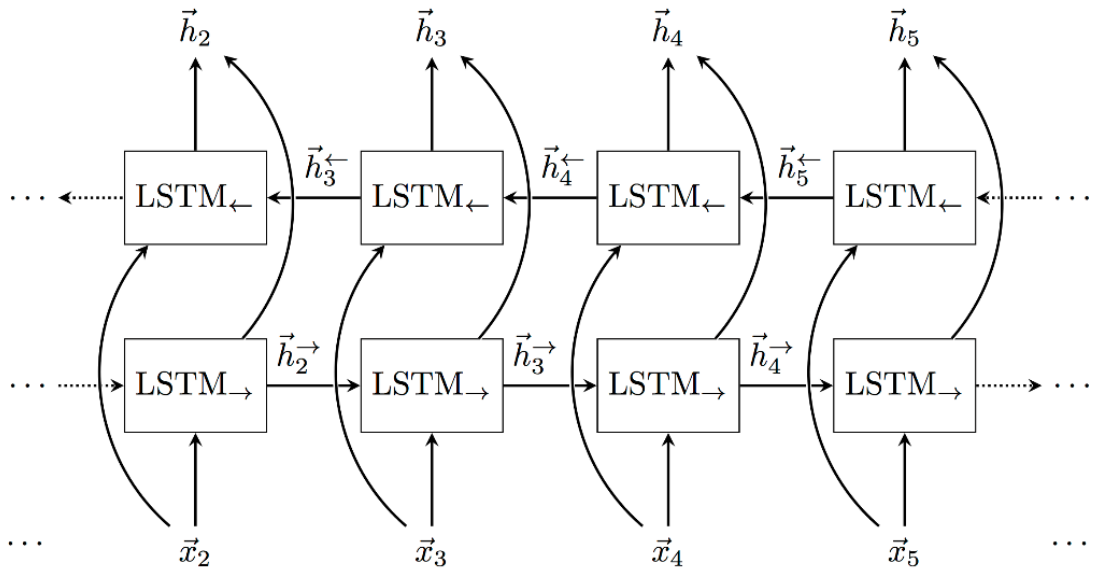
$y_t = \text{Output layer}$

$W = \text{Nilai bobot}$

$\vec{h}_t = \text{hidden layer kanan}$

$\overleftarrow{h}_t = \text{hidden layer kiri}$

Berikut merupakan gambaran arsitektur umum dari BiLSTM pada gambar 2.1.

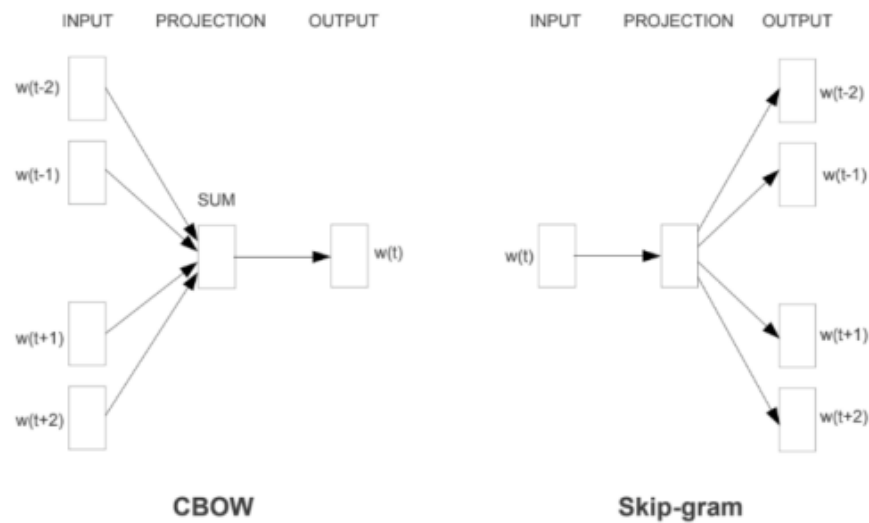


Gambar 2.2 Arsitektur BiLSTM (Permana & Purnamasari, 2019)

2.8 Word Embedding

Teknik yang menggunakan vektor angka berdimensi tinggi sebagai representasi dari kata-kata dikenal dengan istilah *word embedding* (Prasetyo & Kusumaningrum, 2018). Teknik ini memungkinkan komputer lebih efektif dalam melakukan analisis pada data teks yang berukuran besar. Hal ini karena data teks yang sudah diurutkan dan diberi nomor urut berupa integer, tidak bisa diproses begitu saja di komputer tanpa dikonversi sebelumnya menjadi vektor angka yang berdimensi tinggi. Dengan melakukan word embedding, hubungan antar kata dalam korpus dapat dipelajari berdasarkan konteksnya.

Salah satu metode yang populer untuk melakukan *word embedding* adalah *FastText*. FastText digunakan untuk *word embedding* karena keunggulannya untuk menemukan kata-kata yang jarang ditemukan atau berada diluar dari kosa kata yang ada (*out of vocabulary*) (Gunawan, Young, & Rusli, 2021). Terdapat dua arsitektur FastText, yaitu *Continuous Bag of Words (CBOW)* dan *Skip-Gram*. Gambaran arsitektur COBW dan Skip Gram dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Konsep COBOW dan Skip-gram (Mikolov, Corrado, Chen, & Dean, 2013)

Cara kerja dari arsitektur COBOW adalah memahami kata target dari informasi(kata-kata) yang diberikan. Output dari COBOW adalah prediksi kata target yang dipelajari berdasarkan konteks. Sedangkan cara kerja dari arsitektur Skip-gram adalah memahami informasi yang diberikan atau hubungan antar kata target dengan kata-kata disekitarnya(informasi). Output dari arsitektur Skip-Gram adalah prediksi kata target yang dipelajari berdasarkan konteks.

2.9 Lexicon Based Method

Metode yang dikenal juga sebagai pendekatan berbasis leksikal atau *Lexical based approach* (Rifiana & Karel, 2019). Merupakan pemanfaatan kamus yang berisi daftar kata atau frasa yang disebut juga leksikon yang mana setiap kata maupun frasa tersebut dikaitkan dengan sentimen tertentu. Seperti pada kamus InSet Lexicon dari penelitian terdahulu (Koto & Rahmaningtyas, 2017), kata dan frasa pada kamus leksikon akan dilakukan perhitungan *polarity score* pada setiap kalimatnya. Pasangan kata dan skor polaritas leksikon positif dapat dilihat pada tabel 2.1.

Tabel 2.2 Kamus InSet Lexicon Positif

Kata	Bobot
hai	3
merekam	2
Ekstensif	3
paripurna	1
Detail	2

Tabel 2.3 Kamus InSet Lexicon Positif(Lanjutan)

Kata	Bobot
Pernik	3
Belas	2
Welas	4
kabung	1
rahayu	4

Selanjutnya, pasangan kata dan skor polaritas leksikon positif dapat dilihat pada tabel 2.2.

Tabel 2.4 Kamus InSet Lexicon Negative

Kata	Bobot
putus tali gantung	-2
gelebah	-2
gobar hati	-2
tersentuh (perasaan)	-1
isak	-5
larat hati	-3
nelangsa	-3
remuk redam	-5
tidak segan	-2
gemar	-1

Adapun tahapan dalam perhitungan skor polaritas pada kalimat pertama-tama adalah dengan menjumlahkan keseluruhan bobot dari kata yang telah dideteksi sistem lalu, data kemudian dikelompokkan ke masing-masing jenis sentimen melalui algoritma yang sudah ditetapkan. Berikut ini merupakan rumus perhitungan metode *lexicon based* pada persamaan 2.12.

$$\begin{array}{ll}
 \text{if sentiment score} > 0 & \text{then Sentimen Positif} \\
 \text{if sentiment score} = 0 & \text{then Sentimen Netral} \\
 \text{if sentiment score} < 0 & \text{then Negatif.}
 \end{array}
 \quad 2.12$$

2.10 Integrasi NER dan Lexicon Based Dictionary

Metode penggabungan antara NER dan Lexicon-Based Dictionary pada sentimen analisis adalah melakukan integrasi kedua teknik tersebut yang bertujuan untuk menemukan entitas dan kategori sentimen yang terdapat pada teks. Identifikasi entitas bernama dalam teks, seperti orang, lokasi, dan produk dilakukan dengan teknik *Named Entity Recognition*, lalu metode *Lexicon-Based Dictionary* bekerja dengan cara menggunakan daftar kata-kata yang telah dinilai untuk dipakai dalam melakukan klasifikasi sentimen pada data teks (Wikarsa, Angdresey, & Kapantow, 2022).

Penggunaan kombinasi kedua teknik ini akan membutuhkan waktu dan sumber daya yang lebih banyak dalam melakukan *training* pada model dan juga daftar kata-kata yang dinilai harus dikembangkan lebih baik lagi (Wikarsa, Angdresey, & Kapantow, 2022). Beberapa penelitian telah dilakukan terkait penggunaan teknik integrasi ini. Untuk meningkatkan akurasi, selain dengan teknik ini, dilakukan kolaborasi dengan teknik lain misalnya, implementasi metode Naïve Bayes dan Lexicon-Based pada analisis sentimen pada media sosial Twitter.

2.11 Penelitian Terdahulu

Penelitian terdahulu mengenai pembuatan dataset teks bahasa Indonesia untuk keperluan pemrosesan bahasa alami belum banyak jumlahnya. Namun, ada beberapa penelitian terdahulu yang terkait dengan algoritma yang digunakan dan topik permasalahan yang relevan dengan penelitian saat ini.

Penelitian terdahulu mengenai permasalahan anotasi otomatis dataset teks teroris yang berbahasa China yang dilakukan oleh Xin et.al, (2019). Penelitian ini menggunakan pendekatan NER dengan perancangan arsitektur model anotasi teks menggunakan algoritma BiLSTM-CRF, lalu kemudian melakukan klasifikasi teks dengan membandingkan dua algoritma *machine learning* CNN dan RNN. Hasilnya, penelitian ini mampu melakukan anotasi otomatis dan mampu mengklasifikasikan dataset teks teroris dan teks normal dengan *threshold* = 1. Penelitian terkait lainnya dilakukan oleh Widiyanti (2022) membahas tentang pemodelan NER terhadap domain zakat menggunakan algoritma *Conditional Random Fields*. Penelitian ini menghasilkan 12 *named entity class* pada domain zakat dengan nilai *precision*, *recall* dan *f1-score* diatas 80%.

Penelitian terkait selanjutnya mengenai penerapan *Lexicon Based Dictionary* dan *Support Vector Machine* dilakukan oleh Seno dan Wibowo(2019). Penelitian ini berhasil melakukan pelabelan dataset Twitter secara otomatis dengan akurasi 92,5%. Penerapan metode yang sama dilakukan oleh Prasetya, Winarso dan Syahril (2021) pada permasalahan analisis sentimen media sosial Twitter mengenai isu Covid-19. Penelitian yang dilakukan dengan data dalam rentang waktu Maret-Oktober 2020 ini melakukan penghitungan bobot setiap kata. Hasil penelitian ini adalah perolehan opini masyarakat mengenai kepercayaan terhadap isu Covid-19 dengan sentimen kategori positif sebesar 58.08%, sentimen kategori negatif sebesar 37.61%, dan opini sentimen kategori netral sebesar 4.31%. Untuk mengetahui uraian singkat dari kajian riset sebelumnya dapat dilihat pada tabel 2.3

Tabel 2.5 Penelitian Terdahulu

No	Peneliti	Tahun	Metode	Keterangan
1	Xin et.al	2019	Named Entity Recognition(Bi-LSTM) dan algoritma <i>Machine Learning(CNN & RNN)</i>	<ul style="list-style-type: none"> • Dibuat dalam bentuk dua data teks, teks teroris dan teks normal • Berhasil melakukan anotasi dan klasifikasi otomatis pada data teks teroris dan normal
2	Widiyanti	2022	NER(CRF)	<ul style="list-style-type: none"> • Menghasilkan 12 named entity class • Akurasi di atas 80% pada nilai f1-score, recall dan precision
3	Hadi Permana, Purnamasari	2019	NER(BiLSTM-CRF)	<ul style="list-style-type: none"> • Pemodelan NER pada media berita online detik.com, compas.com, cnnindonesia.com • Menghasilkan akurasi 87.77%
4	Seno, Wibowo	2019	<i>Lexicon Based Dictionary</i> dan <i>Support Vector Machine</i> dilakukan	<ul style="list-style-type: none"> • Pelabelan otomatis pada dataset Twitter • Akurasi sebesar 92.5%
5	Prasetya, Winarso, & Syahril	2021	Lexicon Based Dictionary	<ul style="list-style-type: none"> • Berhasil dalam mengklasifikasikan teks Twitter secara otomatis ke sentimen kategori positif, negatif dan netral

Tabel 2.6 Penelitian Terdahulu(Lanjutan)

No	Peneliti	Tahun	Metode	Keterangan
6	Aribowo & Khomsah	2021	<i>Lexicon-based dengan kamus emosi EmoLex.</i>	<ul style="list-style-type: none"> • Penerapan pendekatan berbasis kamus EmoLex • Berhasil mengidentifikasi jenis emosi dominan dan kata-kata kunci terkait masing-masing emosi dalam opini masyarakat.
7	Kurniawati & Winarko	2016	Gabungan analisis lokasi dan (<i>gazetteer</i>) dan rule based <i>lexicon sentiment analysis</i>	<ul style="list-style-type: none"> • Berhasil mencapai nilai rata-rata recall 71,3%, precision 99,2%, f-measure 82,6%, dan akurasi 93,8%. Selain itu mampu memberikan informasi tren masyarakat dan opini pariwisata dengan analisis sentimen, mencapai rata-rata recall 73,1%, precision 100%, f-measure 84,1%, dan akurasi 93,2%.
8	Ramdhani, et.al	2022	Lexicon Based Dictionary & Multi Layer Perceptron	<ul style="list-style-type: none"> • Analisis sentimen berdasarkan leksikon, 63,9% tweet menunjukkan sentimen negatif, 29% menunjukkan sentimen positif, dan 7,1% menunjukkan sentimen netral. • Menggunakan <i>multilayer perceptron</i> berhasil memprediksi sentimen mahasiswa terkait kuliah online dengan akurasi sebesar 71%.

Dari tabel 2.3 hasil dari kajian sebelumnya, maka pada penelitian ini, penulis mengusung kombinasi algoritma Named Entity Recognition(BiLSTM) dan Indonesian Lexicon Based Dictionary, untuk dapat melakukan anotasi dan klasifikasi otomatis pada dataset teks Twitter berbahasa Indonesia. Diharapkan dengan menggunakan kombinasi dari kedua algoritma pada penugasan Pemrosesan Bahasa Alami atau dikenal dengan istilah Natural Language Processing ini dapat menghasilkan sebuah dataset teks Indonesia yang dapat digunakan untuk keperluan penelitian dimasa depan.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Data yang Digunakan

Riset yang dilakukan oleh Farihin(2022) yaitu membangun sebuah data twit NER dengan anotasi entitas bernama menghasilkan dataset NER. Dataset ini membuat ribuan token kata sudah dianotasi secara manual oleh anotator. Tag NER sebagai entitas bernama berjumlah 13 tag unik dengan format *Beginning(B)*, *Inside(I)*, *Other(O)*, rinciannya terdapat pada tabel 3.1

Tabel 3.1 Dataset NER

NO	Tag	Token Kata
1	B-PER	5564
2	I-PER	3058
3	B-PROD	3554
4	I-PROD	1177
5	B-LOC	2225
6	I-LOC	957
7	B-EV	809
8	I-EV	978
9	B-WA	164
10	I-WA	271
11	B-ORG	2085
12	I-ORG	1124
13	O	148104

Dari tabel 3.1 dapat dilihat proporsi atau penyebaran token kata setiap tag NER. Tag Other(O) dengan jumlah sebanyak 148104 token kata. Tag NER yang paling sedikit memiliki token kata adalah tag B-WA(*Beginning Work of Art*) dengan jumlah 164 token kata.

Secara rinci token kata yang memiliki tag **PER**, yaitu sebuah tag yang diberikan untuk token kata yang mengindikasikan nama atau identitas seseorang(*person*). Sepuluh contoh token kata dengan tag ini ditampilkan pada tabel 3.2.

Tabel 3.2 Token Kata dengan Tag PER

Nomor Index	Token Kata	Tag Ner
31	Nadiem	B-PER
37	Megawati	B-PER
89	@daya_hadid	B-PER
129	Anies	B-PER
134	Ahok	B-PER
156	Anies	B-PER
180	AHY	B-PER
188	Nadiem	B-PER
189	Makarim	I-PER
198	Megawati	B-PER

Selain itu terdapat rincian token kata yang memiliki tag **PROD**, yaitu sebuah tag yang diberikan untuk token kata yang mengindikasikan nama produk. Sepuluh contoh token kata dengan tag ini ditampilkan pada tabel 3.3.

Tabel 3.3 Token Kata dengan Tag PROD

Nomor Index	Token Kata	Tag Ner
7	ig	B-PROD
9	youtube	B-PROD
27	ml	B-PROD
84	travella	B-PROD
238	Pempek	B-PROD
239	Zada	I-PROD
262	IG	B-PROD
264	pempek_zada	B-PROD
265	Wa	B-PROD
486	tvOne	B-PROD

Rincian token kata selanjutnya memiliki tag **LOC** kependekan dari *location*. Sebuah tag yang mengindikasikan nama dari suatu tempat, wilayah atau lokasi. Sepuluh contoh token kata dengan tag ini ditampilkan pada tabel 3.4.

Tabel 3.4 Token Kata dengan Tag LOC

Nomor Index	Token Kata	Tag Ner
29245	Falah	I-LOC
32313	jatim	B-LOC
10232	Pulau	B-LOC
48350	Riau	B-LOC
32874	jogja	B-LOC
152163	Cina	B-LOC
107660	jogja	B-LOC
46123	Bali	I-LOC
129131	Dangung-Dandung	I-LOC
3674	tangerang	B-LOC

Terdapat juga token kata yang mengindikasikan suatu peristiwa atau kejadian dengan tag **EV**. Tag ini merujuk pada kata “Event”, dipasangkan dengan token kata yang relevan dengan nama suatu peristiwa. Rincian sepuluh token kata dengan tag ini dapat dilihat dengan seksama pada tabel 3.5.

Tabel 3.5 Token Kata dengan Tag EV

Nomor Index	Token Kata	Tag Ner
85460	pemilu	B-EV
23176	#ShopeeRamadanTVSHOW	B-EV
26225	#JumatUntung	B-EV
160070	#ShopeeRamadanSale	B-EV
76320	Shopee	B-EV
73732	#ShopeeRamadanSale	B-EV
2106	"Ngobrol	B-EV
91468	1965.	I-EV
83124	Belanja	I-EV
85513	Giat	B-EV

Terdapat juga tag **WA** yang merupakan kependekan dari *Work of Art* . Tag ini diberikan terhadap token kata yang mengindikasikan nama dari sebuah karya. Secara rinci sepuluh token kata selanjutnya memiliki tag **WA** ditampilkan pada tabel 3.6.

Tabel 3.6 Token Kata dengan Tag WA

Nomor Index	Token Kata	Tag Ner
14566	CARNIVAL	I-WA
18642	Islamic	I-WA
56807	BTS	B-WA
46171	Angsa	I-WA
80932	the	B-WA

Tabel 3.7 Token Kata dengan Tag WA(Lanjutan)

Nomor Index	Token Kata	Tag Ner
145228	itsay	B-WA
101627	tentang	I-WA
77388	box	I-WA
99084	masjid-agung- purwokerto-hasil- rancangan-ridwan...	B-WA
147194	MAPS	B-WA

Terakhir, token kata yang dipasangkan dengan tag **ORG** yang merupakan kependekan dari *Organization*. Tag ini diberikan terhadap token kata yang mengindikasikan nama dari sebuah organisasi, perkumpulan atau klub. Secara rinci sepuluh token kata selanjutnya memiliki tag **ORG** ditampilkan pada tabel 3.7.

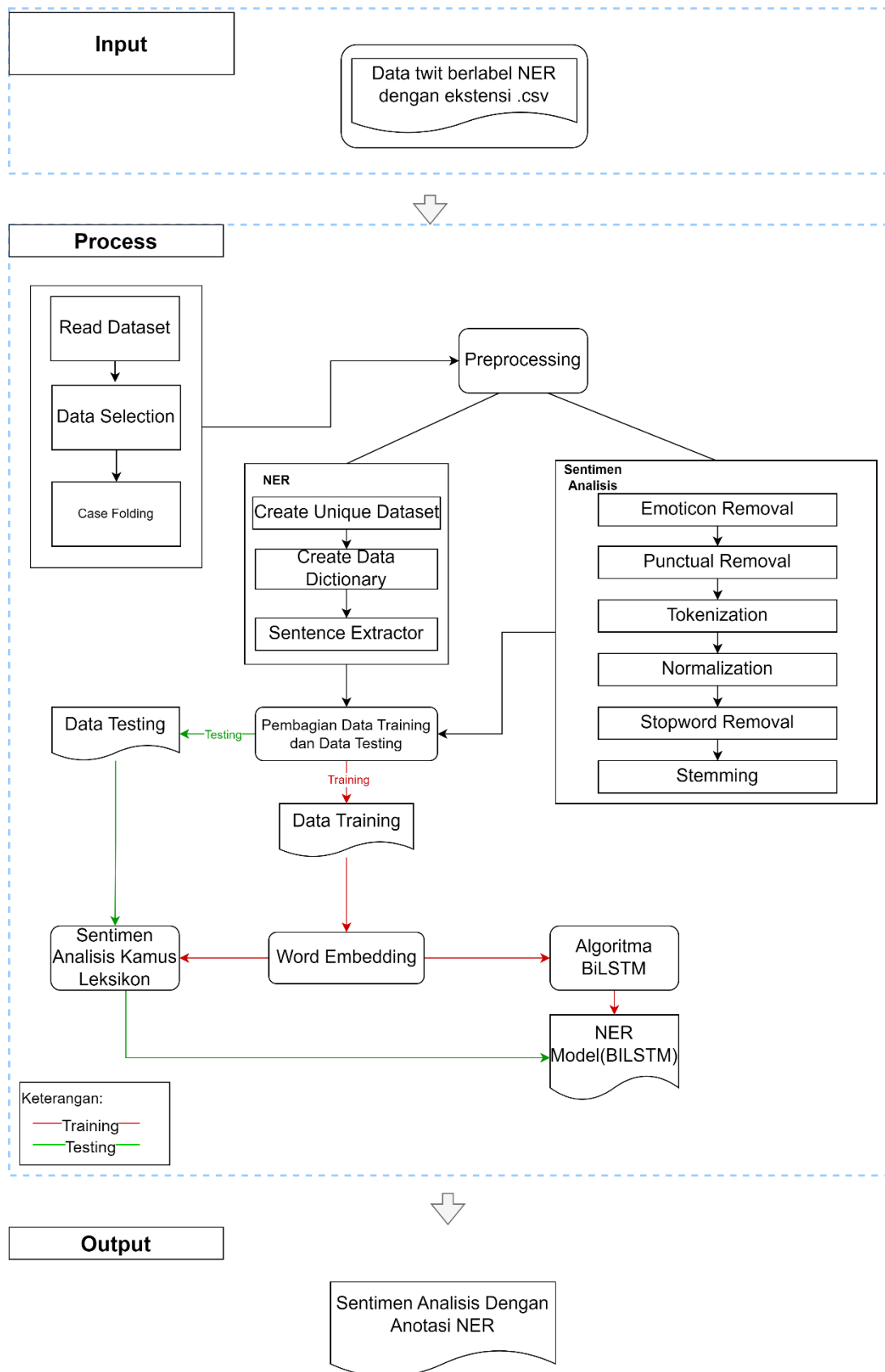
Tabel 3.8 Token Kata dengan Tag ORG

Nomor Index	Token Kata	Tag Ner
165731	INDOMY	B-ORG
5624	BUMDesa	B-ORG
3283	Demokrat	I-ORG
39400	Baung	I-ORG
103691	GEMIRA	B-ORG
81227	UPT	B-ORG
68128	Supreme	B-ORG
59923	Subsatgas	B-ORG
152353	BPKH	B-ORG
124954	Paragon	I-ORG

Sedangkan untuk melakukan pemahaman dari opini masyarakat dilakukan dengan menganalisis sentimen terkait opini yang disampaikan oleh pengguna Twitter, data yang digunakan adalah kamus leksikon berbahasa Indonesia. Dataset berbentuk kamus leksikon berbahasa Indonesia ini secara rinci dijelaskan pada 2.8.

3.2 Arsitektur Umum

Dalam melakukan anotasi otomatis entitas bernama dan klasifikasi sentimen masyarakat terhadap data Twitter dilakukan dengan melalui beberapa tahapan. Pertama, data melalui tahap *pre-processing* dengan tujuan untuk meningkatkan kualitas data agar dapat diproses selanjutnya dengan algoritma BiLSTM. Adapun langkah-langkah pada pembuatan anotasi otomatis dan sentimen analisis terhadap data twit pada gambar 3.1



Gambar 3.1 Diagram Alir Sentimen Analisis dengan Anotasi Otomatis Text Twitter

3.2.2 Preprocessing

Tahapan selanjutnya adalah melakukan proses pengolahan pada data input. Tahapan ini terbagi dalam beberapa bagian, salah satunya adalah tahap *preprocessing* terhadap data(input). Proses prapemrosesan untuk NER Gambar 3.2.

```
function process_data_ner(filename):
    df_tweet ← read_raw_data(filename)
    df_tweet ← case_folding(df_tweet)
    df_tweet, num_words, num_tags, word2idx, tag2idx,
    idx2tag, words, tags ← create_unique_dataset(df_tweet)
    output_path ← "./data/kamus_data.json"
    save_data_dictionary(word2idx, tag2idx, idx2tag,
    tags, output_path)
    getter ← SentenceGetter(df_tweet)
    sentences ← getter.sentences
    temp ← create_dataframe(sentences)
    temp.index_name ← 'tweet_id'
    df_tweet ← merge_dataframes(df_tweet, temp)
    df_tweet ← factorize_column(df_tweet, 'tweet_id')
    df_tweet['tweet_id'] ← 'tweet_id:' +
    df_tweet['tweet_id'].astype(str)
    return df_tweet, num_words, num_tags, word2idx,
    tag2idx, idx2tag, words, tags
```

Gambar 3.2 Proses Preprocessing NER

Penjelasan mengenai Gambar 3.2 adalah sebagai berikut.

- Membuat sebuah fungsi dengan nama `read_raw_data` dengan parameter `filename` sebagai nama file data mentah yang akan dibaca oleh sistem.
- File yang sudah dibaca sistem akan diubah menjadi huruf kecil menggunakan fungsi ``case_folding`` dengan parameter ``df_tweet`` yaitu nama baru dari file yang sudah dibaca sebagai *dataframe* sistem.
- Mengembalikan variabel dataframe, `num_words`, `num_tags`, `word2idx`, `tag2idx`, `idx2tag`, `words`, `tags` dari proses yang sudah dilakukan sebelumnya.
- Inisiasi fungsi ``create_unique_dataset`` dengan parameter ``df_tweet`` untuk menyimpan data yang ada dari kolom `word2idx`, `tag2idx`, `idx2tag`, `words`, `tags` yang dipasangkan dengan variabel yang sama agar menjadi sebuah kamus data dengan form JSON.

- Inisialisasi objek `getter = SentenceGetter(df_tweet)` dengan parameter `df_tweet` untuk membuat kalimat ekstraktor dari setiap token kata dan label NER dari dataframe.
- Pasangan token kata dan label NER sebagai kalimat dari objek `getter` yang sudah diinisiasi sebelumnya lalu diambil dan disimpan ke dalam variabel `sentences`
- Inisiasi variabel `temp` sebagai dataframe sementara yang menyimpan hasil kalimat ekstraktor ke dalam kolom `ner_sentence` dan mengatur nomor indeks.
- Menggabungkan variabel `temp` (*dataframe* sementara) dengan `df_tweet` (*dataframe* sistem) dengan menambahkan kolom `ner_sentence` ke *dataframe* sistem.
- Lalu mengembalikan variabel `df_tweet`, `num_words`, `num_tags`, `word2idx`, `tag2idx`, `idx2tag`, `words`, `tags` untuk digunakan pada tahap selanjutnya, yaitu pra-pemrosesan sentimen analisis.

Sama halnya dengan melakukan prapemrosesan terhadap data yang tujuannya adalah mempersiapkan data NER, maka proses prapemrosesan dilakukan juga untuk sentimen analisis. Proses prapemrosesan data untuk sentimen analisis dapat dilihat secara lengkap pada gambar 3.3.

```

function preprocessing_SA(df_tweet):
    df_tweet['tanpa_emoji'] ←
df_tweet['lower'].apply(remove_emoji)
    df_tweet['tanpa_punct'] ←
df_tweet['tanpa_emoji'].apply(punct_removal)
    df_tweet['word_tokens'] ←
df_tweet['tanpa_punct'].apply(tokenization)

    kamus_baku ← read_excel('data/kamus_kata_alay.xlsx')
    dict_kamus_baku ←
create_dictionary_from_kamus_baku(kamus_baku)
    df_tweet['normal'] ←
apply_normalization(df_tweet['word_tokens'],
dict_kamus_baku)

    df_tweet['tanpa_stopword'] ←
df_tweet['normal'].apply(stopwords_rem)
    stem_text(df_tweet, 'tanpa_stopword')

return df_tweet

```

Gambar 3.3 Proses Preprocessing Sentimen Analisis

Penjelasan mengenai *pseudocode* dari Gambar 3.3 adalah sebagai berikut.

- Inisiasi fungsi `remove_emoji` untuk menghapus emoji. Hasilnya disimpan pada kolom baru yaitu `tanpa_emoji`.
- Inisiasi fungsi `punct_removal` untuk menghapus tanda baca, URL, *mention* akun, *hashtag* serta karakter khusus HTML akibat penarikan data dari API Twitter oleh pengolah data sebelumnya. Hasilnya disimpan dalam kolom baru, kolom `tanpa_punct`.
- Melakukan pemecahan kata pada setiap baris data di kolom `word_tokens` menjadi token-token dengan menggunakan method `word_tokens()`. Hasilnya disimpan ke kolom baru, kolom `word_tokens`.
- Mendeklarasi `kamus_baku ← baca_kamus_bahasa_alay()` yaitu untuk mengambil pasangan kata slang(bahasa tidak baku) dan kata baku.
- Mengubah kamus yang berisi kata tidak baku menjadi sebuah *dictionary*.
- Melakukan normalisasi kalimat dengan fungsi `apply_normalization(df_tweet['word_tokens'],`

`dict_kamus_baku)` terhadap data dengan parameter kolom ``word_tokens`` dan kamus kata non formal.

- Inisiasi fungsi ``stopwords_rem`` untuk menghapus *stopwords*. Proses ini dilakukan dengan cara menentukan sekumpulan stopwords yang tidak boleh dihapus karena berpengaruh pada makna setiap kalimat dalam baris data. Lalu, setiap kata(*stopwords*) akan dihapus dalam kolom ``normal`` dengan pengecualian kata(*stopwords*) yang tidak boleh dihapus yang sudah diinisiasi sebelumnya. Hasil pengolahan pada tahap ini akan disimpan ke kolom ``tanpa_stopword``.
- Melakukan iterasi menggunakan perulangan ``for`` pada setiap baris data yang ada di kolom ``tanpa_stopword``, lalu memotong setiap kata yang berimbuhan hingga hanya tersisa kata dasarnya saja. Hasil proses ini akan disimpan ke kolom ``stemmed``.
- Mengembalikan *dataframe* sistem untuk pengolahan selanjutnya.

3.2.2.1 Data Selection

Pada tahap ini, data yang diambil hanyalah data pada kolom token yang berisi sekumpulan kata, dan data dalam kolom ner yang berisi label NER dari setiap token kata berdasarkan kolom tweet_id-nya. Gambaran lengkap dari data selection dapat dilihat pada gambar 3.4.

```
function read_raw_data(filename):
  data ← read_csv(filename, encoding='latin1')
  data.fillna(method='ffill')
  selected_columns ← ["tweet_id", "token", "ner"]
  data ← data[selected_columns]
  data.rename(columns={"tweet_id": "tweet_id", "token":
"token", "ner": "ner"})
  data['tweet_id'] ← factorize(data['tweet_id'])
return data
```

Gambar 3. 4 Pseudocode proses Data Selection

Penjelasan mengenai Gambar 3.4 secara rinci adalah sebagai berikut.

- Inisiasi variabel data yang digunakan untuk menyimpan hasil dari pembacaan data. Data yang dibaca menggunakan format Comma-Separated Values(CSV).

- ``method='ffil'`` digunakan untuk mencegah adanya data yang tidak memiliki nilai dengan mengisi data tersebut menggunakan isi dari baris data yang ada di atasnya. Metode ini dikenal dengan nama *forward fill*.
- Selanjutnya mengurutkan nama kolom data, agar data dimulai dari kolom 'tweet_id', 'token', 'ner'.
- Mengubah isi kolom 'tweet_id' agar menjadi angka indeks yang simple dari 0 hingga ke baris data twit terakhir menggunakan ``factorize(data['tweet_id'])``
- Mengembalikan *dataframe*

Hasil dari penerapan fungsi data selection terhadap data original dari dataset NER pada Contoh 1, digambarkan pada tabel 3.8.

Tabel 3.9 Proses Data Selection

tweet_id	token	ner
1382601382042103808	Hidup	O
1382601382042103808	sesedih	O
1382601382042103808	dan	O
1382601382042103808	secaper	O
1382601382042103808	apa	O
1382601382042103808	yak	O
1382601382042103808	nyamperin	O
1382601382042103808	ig	B-PROD
1382601382042103808	atau	O

3.2.2.2 Case Folding

Merupakan proses pengubahan seluruh karakter alphabet menjadi huruf kecil. Tujuannya adalah untuk menyederhanakan kalimat agar dapat memperoleh nilai dan makna yang sama. Sehingga dapat dibaca dan dikenali dengan mudah oleh sistem. Pseudocode dari proses *case folding* dapat dilihat pada gambar 3.5.

```

fungsi case_folding(df):
    df['lower'] ← df['token'].lower()

    for setiap baris in df:
        df['lower'] ← strip_whitespace(df['lower'])

    df ← filter_empty_rows(df)
    df ← reset_index(df)
    return df

```

Gambar 3.5 Pseudocode proses Case Folding

Penjelasan dari Gambar 3.5 adalah sebagai berikut.

- Melakukan iterasi menggunakan perulangan `for` untuk setiap kata dalam kolom `token` untuk diubah menjadi huruf kecil.
- Menyimpan hasilnya kedalam kolom `lower` dan mengembalikan dataframe.
- Melakukan iterasi dengan perulangan `for` pada kolom `lower` untuk menghilangkan spasi tambahan, karakter *new line* disetiap baris data.
- Mengatur ulang indeks pada dataframe agar memudahkan penemuan nomor indeks data bila terjadi kesalahan dalam pengolahan.

Perubahan data sebelum dan sesudah proses *case folding* dapat dilihat pada tabel 3.9.

Tabel 3.10 Proses Case Folding

Sebelum	Sesudah
Hidup sesedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain , perasaan dari dulu gua ngefans bola orang pada santai eh di ml ! malah gini\u00ef\u00bf\u00bd\u00ef\u00bf\u00bd	hidup sesedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain , perasaan dari dulu gua ngefans bola orang pada santai eh di ml ! malah gini\u00ef\u00bf\u00bd\u00ef\u00bf\u00bd

3.2.2.3 Create Unique Dataset

Pada tahapan ini, kata yang sudah diubah menjadi huruf kecil dikumpulkan menjadi daftar kata unik lalu dijumlahkan. Selanjutnya, diberi nomor urut dari 0 hingga ke kata terakhir. Demikian pula dengan tag NER akan dijumlahkan dan diurutkan dari indeks ke-0 hingga indeks ke-12, karena jumlah tag NER adalah 13 tag. Proses membuat dataset unik dapat dilihat pada gambar 3.6.

```

function create_unique_dataset(df)
  words ← daftar nilai unik dalam df['lower']
  words.append("endpad")
  tags ← daftar nilai unik dalam df['ner']
  num_words ← panjang dari words
  num_tags ← panjang dari tags
  word2idx ← kamus kosong
  for i, w in enumerate(words)
    word2idx[w] ← i + 1
  tag2idx ← kamus kosong
  for i, t in enumerate(tags)
    tag2idx[t] ← i
  idx2tag ← kamus kosong
  for k, v in tag2idx.items()
    idx2tag[v] ← k
  df['word2idx'] ← df['lower'].map(word2idx)
  df['idx2tag'] ← df['ner'].map(idx2tag)
  return df, num_words, num_tags, word2idx, tag2idx,
  idx2tag, words, tags
end

```

Gambar 3.6 Pseudocode Proses Create Unique Dataset

Penjelasan dari Gambar 3.6 adalah sebagai berikut.

- Inisiasi variable `words` untuk menyimpan kata-kata unik dari kolom `lower`.
- Menambahkan satu elemen kata ke variable `words`, yaitu `endpad`.
- Inisiasi variable `tags` untuk menyimpan label NER unik dari kolom `ner`.
- Inisiasi variable `num_words`, `num_tags`, untuk menyimpan panjang dari nilai data yang dari variable `words` dan `tags`.
- Inisiasi variabel `word2idx` sebagai kamus kosong. Melakukan perulangan `for` pada elemen `w` untuk setiap kata yang ada dalam variabel `words` untuk dipasangkan dengan nomor indeks yang dimulai dari 1, dengan cara `i+1`.
- Hasil pasangan kata dan nomor index disimpan di kamus kosong `word2idx`.
- Inisiasi variabel `tag2idx` sebagai kamus kosong. Melakukan perulangan `for` pada elemen `t` untuk setiap label yang ada dalam variabel `tags` untuk dipasangkan dengan nomor indeks yang dimulai dari 1, dengan cara `i+1`.
- Hasil pasangan label dan nomor index disimpan di kamus kosong `tag2idx`.


```

        'tag2idx': tag2idx,
        'idx2tag': idx2tag,
        'tags': tags
    }
    with membuka_file(output_path, 'w') as f:
        tulis_json(data, f)

```

Gambar 3.7 Pseudocode Proses Create Data Dictionary

Penjelasan mengenai Gambar 3.7 adalah sebagai berikut.

- Inisiasi fungsi `create_data_dictionary` dengan parameter yang disebutkan.
- Deklarasi beberapa variabel kamus(*dictionary*) dari proses yang dilakukan pada gambar 3.6.
- Menyimpan kamus data ke dalam direktori lokal dengan format JSON.

Hasil keluaran dari proses ini ditampilkan pada tabel 3.12.

Tabel 3.13 Proses Create Data Dictionary

kamus_data.json
{"word2idx": {"#inikarakita": 1, "semangka": 2, "feri": 3, "@beritasatu": 4, "guk": 5, "avanza": 6, "https://t.co/w5a4cxhfrq": 7, "create": 8, "@mnyoongi93": 9, "course": 10, "mudah2an": 11, "tmp": 12, "\u00e2\u0097\u0087\u00e2\u0097\u0086": 13, "https://t.co/bqfoikvbmp": 14, "@itsmeyaw": 15, "faefahnya": 16,...} "tag2idx": {"B-LOC": 0, "B-PER": 1, "I-PROD": 2, "B-PROD": 3, "I-PER": 4, "B-WA": 5, "I-LOC": 6, "I-WA": 7, "B-ORG": 8, "I-EV": 9, "B-EV": 10, "I-ORG": 11, "O": 12}, "idx2tag": {"B-LOC": 0, "B-PER": 1, "I-PROD": 2, "B-PROD": 3, "I-PER": 4, "B-WA": 5, "I-LOC": 6, "I-WA": 7, "B-ORG": 8, "I-EV": 9, "B-EV": 10, "I-ORG": 11, "O": 12}, "tags": ["B-LOC", "B-PER", "I-PROD", "B-PROD", "I-PER", "B-WA", "I-LOC", "I-WA", "B-ORG", "I-EV", "B-EV", "I-ORG", "O"]}

Dari tabel 3.12, semua variabel penting termasuk, kata dan tag yang dikonversi menjadi nomor indeks disimpan dalam suatu file dengan format JSON di direktori lokal komputer.

3.2.2.5 Sentence Extractor

Pada proses ini dilakukan pengambilan kalimat yang telah diproses pada tahap *case folding* untuk dipasangkan dengan label NER-nya menjadi tupel (kata, label NER), lalu semua tupel ini dijadikan sebuah kalimat. Proses ekstraksi kalimat dapat dilihat pada gambar 3.8.


```

class SentenceGetter(object)
    inisialisasi(self, data)
    self.data ← data
    self.sentences ← panggil self.get_sentences()

    function get_sentences(self)
        sentences ← kamus kosong
        for setiap tweet_id, group in
data.groupby('tweet_id')
            words ← group['lower'].values.tolist()
            ners ← group['ner'].values.tolist()
            sentence ← daftar kosong
            for setiap w, n in zip(words, ners)
                tambahkan (w, n) ke sentence
            sentences[tweet_id] ← sentence
    return sentences

```

Gambar 3. 8 Pseudocode Proses Create Data Dictionary

Penjelasan dari proses pada Gambar 3.8 adalah sebagai berikut.

- Saat membuat objek `SentenceGetter`, diperlukan pemanggilan konstruktor dengan cara `inisialisasi(self, data)`. Ketika objek `data` diinisiasi, maka atribut `self.data` akan menyimpan `data`. Selanjutnya, dalam menghasilkan kalimat-kalimat yang sesuai, dilakukan pemanggilan terhadap metode `get_sentences()`.
- Inisiasi `get_sentences(self)` dilakukan untuk menghasilkan kalimat-kalimat yang sesuai berdasarkan kolom `tweet_id`. Ada beberapa langkah yang ada dalam tahapan ini, antara lain sebagai berikut.
 - Inisiasi kamus `sentences`. Kalimat-kalimat yang ada berdasarkan `tweet_id` akan dikelompokkan dan disimpan dalam kamus ini.
 - Dilakukan pemecahan *dataframe* `data` berdasarkan `tweet_id` dengan menggunakan bantuan dari metode `groupby('tweet_id')`. Dalam melakukan ini terbagi lagi menjadi beberapa langkah sebagai berikut.
 - Inisiasi `words` yang digunakan untuk melakukan penarikan nilai dari kolom 'lower' dalam kelompok tersebut dan selanjutnya dilakukan konversi menjadi daftar (*list*)
 - Inisiasi `ners` untuk mengambil nilai dari kolom 'ner' dalam kelompok tersebut. Lalu melakukan konverksi menjadi daftar (*list*).

- Inisiasi ``sentence`` sebagai daftar kosong(*empty list*). Tujuannya adalah untuk menyimpan pasangan kata dan label (w, n).
- Melakukan iterasi menggunakan ``for`` dan inisiasi ``fungsi zip()``. Tujuannya untuk melakukan perulangan pada setiap kata (w) dari ``words`` dipasangkan dengan label (n) dari ``ners``. Hasil pasangan kata dan label ini dimasukkan ke dalam daftar ``sentence``.
- Hasil pasangan kata dan label dengan kolom ``tweet_id`` sebagai kunci disimpan dalam kamus ``sentences``.
- Mengembalikan variabel ``sentences`` untuk digunakan pada proses selanjutnya.

Hasil dari proses ekstraksi kalimat NER terhadap teks original (contoh 2) dan tag (contoh 3) dapat dilihat pada contoh 5.

Contoh 5 Kalimat NER

```
[["hidup","O"],["sesedih","O"],["dan","O"],["secaper","O"],["apa","O"],["yak","O"],
["nyamperin","O"],["ig","B-PROD"],["atau","O"],["youtube","B-
PROD"],["tim","O"],["lain","O"],["cuma","O"],["buat","O"],["ngatain","O"],["","O
"],["perasaan","O"],["dari","O"],["dulu","O"],["gua","O"],["ngefans","O"],["bola","
O"],["orang","O"],["pada","O"],["santai","O"],["eh","O"],["di","O"],["ml","B-
PROD"],["!","O"],["malah","O"],["gini\u00ef\u00bf\u00bd\u00ef\u00bf\u00bd","O
"]]
```

3.2.2.6 Emoticon removal

Tahapan *emoticon removal* diperlukan untuk meningkatkan kelancaran proses sentimen analisis. Kebanyakan pengguna sosial media khususnya pengguna Twitter dalam menulis *tweet* terkadang menggunakan *emoticon* yang kurang tepat. Sehingga dapat menyebabkan inkonsistensi data. Proses menghapus emoji digambarkan pada Gambar 3.9.

```

fungsi remove_emoji(string):
    emoticons_happy<- set dari emotikon bahagia
    emoticons_sad<- set dari emotikon sedih

    emoji_pattern<- pola regex untuk mengenali emoji

    gabungkan emoticons_happy dan emoticons_sad menjadi
    satu set emoticons

    for setiap emot in string, lakukan:
        if emot tidak ada dalam set emoticons:
            tambahkan emot ke dalam string

    hapus spasi di awal dan akhir string

    ganti semua emoji dalam string dengan string kosong

    return string sebagai hasilnya

```

Gambar 3.9 Pseudocode Proses Emoticon Removal

Adapun penjelasan dari Gambar 3.9 adalah sebagai berikut.

- Dilakukan pengelompokkan terhadap sekumpulan emoji atau emotikon yang akan dihapus, meliputi emoji yang menunjukkan ekspresi senang, sedih, dan pola emoji lain dari kombinasi ekspresi regular.
- Menggabungkan sekumpulan emoji yang akan dihapus.
- Melakukan iterasi menggunakan `for` untuk melakukan perulangan terhadap string untuk mengecek apakah dalam string terdapat emotikon dalam daftar kelompok emotikon yang akan dihapus, jika ada maka dihapus.
- `strip()` digunakan untuk menghapus spasi yang mungkin ada diawal dan diakhir.
- Menghapus dan mengganti semua emotikon dengan *string* kosong.
- Mengembalikan *string* yang telah dihapus dari emoji.

Keluaran dari proses penghapusan emotikon dapat ditampilkan pada tabel 3.13.

Tabel 3.14 Proses Emoticon Removal

Sebelum	Sesudah
Hidup sesedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain , perasaan dari dulu gua ngefans bola orang pada santai eh di ml ! malah gini\u00ef\u00bf\u00bd\u00ef\u00bf\u00bd	Hidup sesedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain , perasaan dari dulu gua ngefans bola orang pada santai eh di ml ! malah gini\u00ef\u00bf\u00bd\u00ef\u00bf\u00bd

3.2.2.7 Punctuation Removal

Merupakan tahap selanjutnya setelah penghapusan emoji. Pada tahap ini, yang dilakukan menghilangkan tanda baca, karakter *double space*, *new line*, mentions, *retweet* (RT), URL, dan lain-lain. Tujuannya adalah untuk mempersingkat teks agar dapat dilakukan pemrosesan sentimen analisis. Pseudocode proses penghapusan tanda baca dapat dilihat pada gambar 3.10.

```
fungsi punctual_removal(df)
  df ← menghapus @mentions dari df
  df ← menghapus tagar '#' dari df
  df ← menghapus RT dari df
  df ← menghapus hyperlink dari df
  df ← mengganti karakter \n menjadi spasi pada df
  df ← menghapus tanda baca dari df
  df ← menambahkan spasi setelah penghapusan tanda
  baca pada df
  df ← menghapus spasi ganda dan karakter enter pada
  df
  df ← menghapus tag HTML dari df
  return df
end
```

Gambar 3. 10 Pseudocode Proses Punctuation Removal

Penjelasan mengenai Gambar 3.10 adalah sebagai berikut.

- Menghapus semua *mentions* yang dimulai dengan tanda '@' dari tweet balasan yang menyebut *username*.
- Menghapus semua hashtag('#') dari semua data yang ada pada *dataframe*.
- Menghapus teks 'RT' dari kutipan tweet yang ditujukan untuk *retweet*.
- Menghapus URL atau *hyperlink* yang ada pada data.
- Menghapus tanda baca yang ada pada teks data.
- Menghapus karakter *newline* dan menggantikan menjadi spasi.
- Menghapus karakter *double* spasi dan *newline* yang tercipta akibat proses sebelumnya
- Menghapus tag HTML dari *dataframe* dari data yang ada pada teks.
- Mengembalikan *dataframe*.

Keluaran dari proses penghapusan tanda baca diperlihatkan di tabel 3.14.

Tabel 3.15 Proses Punctuation Removal

Sebelum	Sesudah
Hidup sesedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain , perasaan dari dulu gua ngefans bola orang pada santai eh di ml ! malah gini	hidup sesedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain perasaan dari dulu gua ngefans bola orang pada santai eh di ml malah gini

3.2.2.8 Tokenization

Proses tokenisasi adalah kegiatan pengurutan kata dengan memecah-mecah kata dalam teks menjadi bagian-bagian kecil yang sering disebut token. Token yang dimaksud dapat berupa kata, frasa, dan elemen penting dalam data teks. Proses ini dilakukan untuk memudahkan proses eksplorasi data. Proses tokenisasi dapat dilihat pada gambar 3.11.

```
function tokenization(text) :
    word_tokens ← word_tokenize(text)
    return word_tokens
```

Gambar 3. 11 Pseudocode Proses Tokenization

Penjelasan dari gambar 3.11 adalah sebagai berikut.

- Teks ditokenisasi dengan menggunakan method `word_tokenize(text)`.
- Hasilnya disimpan ke variable word_tokens.
- Mengembalikan token-token kata.

Keluaran dari proses tokenisasi dapat dilihat pada tabel 3.15.

Tabel 3.16 Proses Tokenization

Sebelum	Sesudah
hidup sesedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain perasaan dari dulu gua ngefans bola orang pada santai eh di ml malah gini	"hidup","sesedih","dan","secaper","apa","yak","nyamperin","ig","atau","youtube","tim","lain","cuma","buat","ngatain","perasaan","dari","dulu","gua","ngefans","bola","orang","pada","santai","eh","di","ml","malah","gini"

3.2.2.9 Normalization

Tahapan ini dilakukan untuk melakukan konversi kata non formal menjadi kata formal. Kata-kata baku(formal) yang dimaksud adalah daftar kata yang merupakan ‘*slang words*’ atau bahasa gaul, misalnya ‘yuuuk’, ‘ngakak’, ‘bkn’, dan lain sebagainya. Contoh dari kata-kata tidak baku diperlihatkan pada tabel 3.16.

Tabel 3.17 Daftar Kata Baku dan Tidak Baku

slang	formal
woww	wow
aminn	amin
met	selamat
netaas	menetas
eeeehhhh	eh
kata2nyaaa	kata-katanya
hallo	halo
kaka	kakak
...	

Pseudocode proses normalisasi data teks dapat dilihat pada gambar 3.12

```

function normalization(text, dict_kamus_baku):
    inisialisasi res dengan string kosong

    for setiap kata in text yang dipisahkan oleh spasi,
    lakukan:
        if kata tersebut ada dalam dict_kamus_baku:
            normalized_token ← dict_kamus_baku.get(token, token)
            result_tokens.append(normalized_token)
        else:
            res ← join_tokens_into_text(result_tokens)
    return res sebagai hasil normalisasi

```

Gambar 3. 12 Daftar Kata Baku dan Tidak Baku

Penjelasan mengenai Gambar 3.12 adalah sebagai berikut.

- Inisiasi variabel `res` sebagai daftar kosong(*empty list*) untuk menyimpan kumpulan kata hasil dari proses normalisasi.
- Melakukan perulangan menggunakan `for` untuk mengecek setiap kata yang dipecah berdasarkan spasi
- Selanjutnya dilakukan pengecekan `if` terhadap kata dalam kalimat tersebut yang artinya, jika kata berada dalam kamus kata non formal tabel 3.11, maka akan diubah dengan pasangan kata baku yang sesuai.
- Jika terjadi sebaliknya, `else`, maka kata ditambahkan ke variabel `res` sebagai kata baku.

- Setelah perulangan selesai dijalankan, variabel `res` digunakan untuk menyimpan hasil dari langkah sebelumnya.
- Mengembalikan *dataframe* yang sudah dinormalisasi.

Keluaran dari proses normalisasi ditunjukkan pada table 3.17.

Tabel 3.18 Keluaran Proses Normalisasi

Sebelum	Sesudah
"hidup","sesedih","dan","secaper","apa","yak","nyamperin","ig","atau","youtube","tim","lain","cuma","buat","ngatain","perasaan","dari","dulu","gua","ngefans","bola","orang","pada","santai","eh","di","ml","malah","gini"	'hidup', 'sesedih', 'dan', 'secaper', 'apa', 'yak', 'nyamperin', 'ig', 'atau', 'youtube', 'tim', 'lain', 'cuma', 'buat', 'ngatain', 'perasaan', 'dari', 'dulu', 'gua', 'ngefans', 'bola', 'orang', 'pada', 'santai', 'eh', 'di', 'ml', 'malah', 'gini'

3.2.2.10 Stopwords removal

Tahapan penghapusan *stowords* adalah proses untuk menghilangkan kata yang memiliki fungsi namun tidak memiliki arti sehingga dapat diabaikan. Misalnya kata-kata yang berada di Contoh 5.

Contoh 6 Daftar kata stopwords

['sih', 'ya', 'hahaha', 'terus', 'tt', 'jadi', 'lah', 'gue', 'dulu', 'kok', 'an', 'nya', 'e', 'kan', 'lo', 'per', 'ba', 'lu', 'gp', 'si', 'bor', 'ah', 'tak']

Proses *stopword removal* ditampilkan pada gambar 3.13.

```
function stopwords_rem(text):
    factory ← buat objek StopWordRemoverFactory()
    set_stopword ← factory.get_stop_words()
    kata ← [daftar kata stop]
    set_stopword.extend(kata)
    text ← pecah(teks) # Mengubah string menjadi daftar kata-kata
    text ← [kata for kata in text jika kata tidak ada dalam set_stopword]
    text ← gabungkan kembali teks
    return text
```

Gambar 3. 13 Pseudocode Proses Stopwords Removal

Penjelasan mengenai Gambar 3.13 adalah sebagai berikut.

- Membuat inisiasi objek *factory* dari `StopWordRemover`
- Mengambil sekumpulan kata dari *factory* yang merupakan kata *stopword*.
- Menetapkan daftar kata(contoh 5) apa saja yang akan dihapus.

- Melakukan penggabungan kata yang akan dihapus ke kata stopwords yang sebelumnya sudah diambil. Hal ini dilakukan dengan cara ``set_stopword.extend(kata)``.
- Melakukan pemecahan kata pada data teks.
- Membuang setiap kata *stopwords*.
- Melakukan penggabungan kembali daftar kata-kata menjadi *string*.
- Mengembalikan teks.

Keluaran dari proses penghapusan kata stop ditampilkan pada tabel 3.18.

Tabel 3.19 Proses Stopwords Removal

Sebelum	Sesudah
'hidup', 'sesedih', 'dan', 'secaper', 'apa', 'yak', 'nyamperin', 'ig', 'atau', 'youtube', 'tim', 'lain', 'cuma', 'buat', 'ngatain', 'perasaan', 'dari', 'dulu', 'gua', 'ngefans', 'bola', 'orang', 'pada', 'santai', 'eh', 'di', 'ml', 'malah', 'gini'	'hidup', 'sesedih', 'dan', 'secaper', 'apa', 'yak', 'nyamperin', 'ig', 'atau', 'youtube', 'tim', 'lain', 'cuma', 'buat', 'ngatain', 'perasaan', 'dari', 'dulu', 'gua', 'ngefans', 'bola', 'orang', 'pada', 'santai', 'eh', 'di', 'ml', 'malah', 'gini'

3.2.2.11 Stemming

Merupakan proses pengubahan seluruh kata pada dataset yang memiliki imbuhan baik diawal maupun diakhir kata. Proses ini menggunakan *package library* Sastrawi. *Package library* khusus yang meng-handle pemrosesan teks bahasa Indonesia. Proses *stemming* secara lengkap dapat dilihat pada Gambar 3.14

```
function stem_text(dataframe, column_name):
    factory<- objek StemmerFactory()
    stemmer<- objek stemmer yang dibuat oleh factory
    stem[]<-inisialisasi list stem
    for setiap nilai i in kolom dataframe[column_name],
    lakukan:
        i = konversi i ke dalam bentuk string (str(i))
        stemming_result = stemmer.stem(i)
        tambahkan stemming_result ke dalam list stem
    dataframe['stemmed'] = stem
```

Gambar 3.14 Pseudocode Proses Stemming

Penjelasan mengenai Gambar 3.14

- ``factory<- objek StemmerFactory()``, menginisiasi objek *factory* sebagai objek dari kelas *StemmerFactory()*.
- ``stemmer<- objek stemmer yang dibuat oleh factory``, inisiasi objek *stemmer* untuk melakukan *stemming*.

- Melakukan iterasi menggunakan perulangan ``for`` untuk setiap kata dalam dataframe dengan kolom data yang diinginkan.
- Dilakukan konversi kata menjadi string.
- String di-*stem* dengan objek *stemmer*. Hasilnya disimpan ke variabel ``stemming_result``.
- Hasil proses pemotongan kata berimbuhan disimpan ke variabel daftar kosong ``stem``.

Keluaran dari proses *stemming* ditunjukkan pada tabel 3.19.

Tabel 3.20 Proses Stemming

Sebelum	Sesudah
'hidup', 'sesedih', 'dan', 'secaper', 'apa', 'yak', 'nyamperin', 'ig', 'atau', 'youtube', 'tim', 'lain', 'cuma', 'buat', 'ngatain', 'perasaan', 'dari', 'dulu', 'gua', 'ngefans', 'bola', 'orang', 'pada', 'santai', 'eh', 'di', 'ml', 'malah', 'gini'	hidup sedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain asa dari dulu gua ngefans bola orang pada santai eh di ml malah gini

3.2.3 Pembagian Dataset

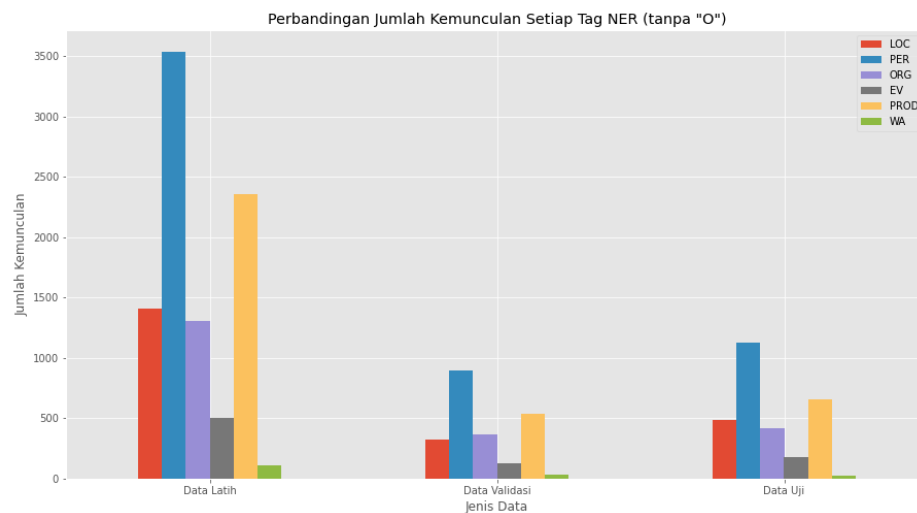
Pembagian dataset merupakan bagian penting dalam pembelajaran mesin yang tujuannya adalah untuk memisahkan data yang akan dimasukkan pada proses pelatihan, pengujian serta untuk mengoptimalkan model. Dataset yang dipakai untuk melatih model NER dengan menggunakan algoritma BiLSTM, sebelumnya akan dibagi dengan perbandingan persentase sebesar 80:20. Dimana, sebanyak 80% dari jumlah dataset NER akan dipakai pada proses pelatihan, dan sisa 20% dari jumlah dataset akan dipakai pada proses pengujian model.

Selanjutnya pembagian data pelatihan dilakukan saat proses *training* dimana, data dilakukan pemecahan lagi dengan perbandingan jumlah sebesar 80:20. Dimana, sejumlah 80% data akan digunakan untuk proses *training* sedangkan sisa 20% dari data latih akan menjadi data validasi. Untuk melihat perbandingan jumlah data yaitu persebaran token kata dan tag NER yang dipakai untuk proses pelatihan, validasi dan pengujian akan digambarkan pada tabel 3.20.

Tabel 3.21 Perbandingan Data Training, Validasi dan Uji

Tag	Data Latih	Data Validasi	Data Uji
O	95542	23642	28920
PER	3539	901	1124
LOC	1414	326	485
ORG	1304	365	416
PROD	2357	542	655
EV	500	129	180
WA	107	35	22

Visualisasi pembagian data diperlihatkan pada gambar 3.15.

**Gambar 3.15** Perbandingan Data Latih, Data Validasi dan Data Uji

3.2.4 Word Embedding

Setelah melakukan tahap *sentence extractor*, selanjutnya pada data pelatihan dilakukan penambahan *padding* kata. Dengan tujuan supaya seluruh kalimat memiliki panjang yang seragam. Setelah data memiliki keseragaman, dilakukan teknik untuk mengkonversi setiap indeks kata menjadi vektor angka. Untuk langkah tersebut, diperlukan suatu teknik menggunakan vektor angka berdimensi tinggi sebagai representasi dari kata-kata dikenal dengan istilah *word embedding*.

Untuk melakukan pemetaan kata menjadi vektor angka, digunakan FastText, yang merupakan salah satu jenis pustaka. Pustaka *FastText* digunakan untuk *word embedding* karena keunggulannya untuk menemukan kata-kata yang jarang ditemukan (*out of*

vocabulary). Selain itu *FastText* menghasilkan kinerja yang baik pada sentimen analisis dan teks klasifikasi. Pada tahapan ini data yang sudah dilakukan tokenisasi harus diubah menjadi bentuk vektor *array* untuk mendapatkan nilai bobot pada jaringan saraf tiruan.

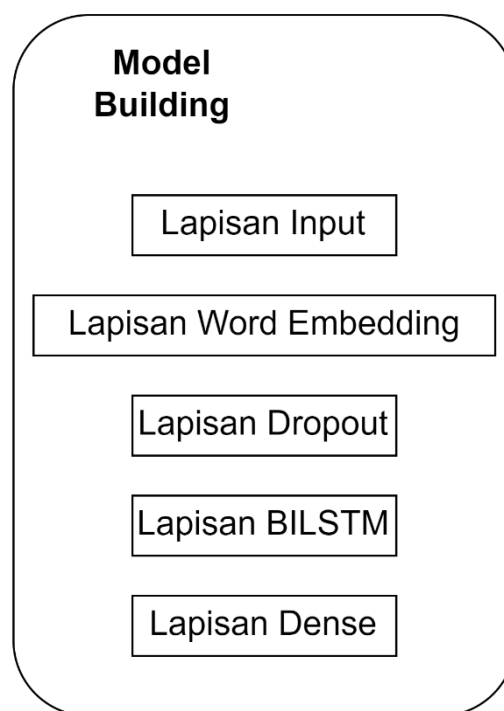
Contoh 7 Embedding Matriks Kata ‘Hidup’

[-0.0248	-0.1097	0.0032	0.085	-0.0111	-0.0804
-0.0441	0.0557	0.0128	-0.086	0.0018	0.0307
0.0031	0.0046	0.036	-0.0318	-0.0016	-0.0041
0.0914	-0.0219	-0.0073	0.0284	-0.0104	0.0564
0.0092	0.0114	-0.0193	-0.0387	0.0606	-0.0518
0.046	0.043	0.089	-0.0376	0.032	0.0212
0.0275	0.027	-0.0129	0.0173	-0.0252	0.0454
-0.0198	0.0786	0.0491	0.023	-0.0149	-0.0477
0.0431	-0.038	-0.0208	-0.0305	0.0254	-0.0759
0.0031	0.055	0.0736	0.0235	-0.0263	0.0003
0.0138	0.1018	-0.0441	0.0082	-0.0145	-0.0457
0.0144	0.0593	0.0318	-0.0474	0.0032	0.0393
0.0413	0.0005	-0.0275	-0.0347	-0.0062	-0.0362
-0.0109	0.0208	-0.0152	-0.0257	-0.0662	0.0281
-0.0649	0.0598	-0.16580001	-0.0945	0.0197	0.0085
0.0173	0.0357	0.0021	-0.0346	-0.1788	-0.1085
-0.0358	0.0344	-0.0342	0.0452	-0.0546	0.0214
-0.0495	0.0028	0.0861	-0.0284	-0.0817	-0.0055
-0.0246	0.0532	-0.0496	-0.0204	0.0122	0.0312
-0.0122	-0.0372	0.0181	0.009	0.0031	0.0105
-0.0574	-0.0455	0.0048	-0.04	-0.0325	-0.0222
0.0364	-0.0111	-0.0156	0.044	0.0507	-0.0153
0.0647	0.0304	0.0501	0.0249	0.0083	0.078
0.0476	-0.0347	-0.0021	-0.1683	0.0251	-0.0118
-0.0183	0.0197	-0.0424	-0.0606	-0.0456	-0.1045
0.0144	-0.0849	-0.16949999	-0.0071	0.17039999	-0.0499
-0.0012	-0.0081	-0.0639	-0.0117	0.0851	0.0279
-0.0494	-0.0078	0.0035	-0.0233	0.0103	0.0223
0.0257	-0.0102	0.0155	0.0058	0.03	0.0028
-0.0279	-0.014	-0.0099	-0.0225	0.0488	0.0107
0.0404	0.0145	-0.0348	-0.0078	0.0063	0.0084
0.106	0.0565	-0.0401	-0.0499	-0.0557	0.0114
-0.003	0.0127	-0.0611	0.0029	0.071	-0.0096
0.0267	0.0913	-0.0136	0.0478	0.0199	0.0058
-0.0758	0.0265	-0.0097	0.0267	0.0018	0.0102
-0.0258	-0.0197	0.0233	-0.017	-0.0195	0.0095
0.0185	0.0277	0.0233	-0.0078	-0.0206	-0.0115
-0.0068	-0.0579	-0.0549	0.011	-0.1006	-0.0064
0.017	-0.0442	-0.0118	0.0254	0.0124	-0.035
-0.0459	-0.0178	-0.0684	-0.0645	0.0019	-0.0529
0.0251	-0.0553	-0.0136	-0.0901	-0.0663	-0.0126
0.0468	0.0377	-0.0575	0.0402	0.0035	-0.0074

0.0736	0.0327	0.0312	0.0524	-0.0326	-0.0166
-0.0035	-0.0681	-0.034	-0.0488	-0.0928	0.0139
0.0582	-0.0105	-0.0177	0.0767	-0.0093	0.0133
0.0215	-0.0542	0.0112	-0.0649	-0.1358	0.0052
-0.1181	-0.0142	-0.0503	-0.0011	-0.0716	-0.0154
0.01	0.0533	0.0122	-0.0344	0.0486	0.0766
0.0099	0.0014	-0.048	-0.0305	-0.0684	-0.
-0.0259	0.0092	0.0914	-0.0429	-0.0015	0.0015]

3.2.5 Model Building

Pada proses ini, setelah data dilakukan tahapan preprocessing hingga *word embedding*, maka selanjutnya dilakukan pemodelan NER pada tweet. Pada pembuatan model NER ini, algoritma yang digunakan adalah BiLSTM. Pembuatan model BiLSTM ini terdiri dari beberapa layer yang dapat dilihat pada gambar 3.16.



Gambar 3.16 Rincian Model BiLSTM

1. Word embedding layer

Pada lapisan ini data menerima input dari proses tokenisasi, dan kata demi kata yang direpresentasikan dengan vektor *array* yang sudah memiliki bobot pada tahapan 3.2.4. Tabel 3.21 menunjukkan contoh data yang akan dijadikan sebagai masukan pada lapisan ini.

Keluaran dari lapisan ini akan digunakan untuk nilai input pada lapisan selanjutnya yaitu, lapisan *Bidirectional LSTM*. Setelah sebelumnya diproses melalui lapisan *dropout*.

2. Dropout Layer

Pada lapisan selanjutnya adalah lapisan *dropout*. Layer ini menggunakan jenis *Spatial Dropout 1D*. Kegunaannya adalah untuk menghapus seluruh vektor dalam satu dimensi pada lapisan *embedding* untuk kata tertentu di setiap *time step* tertentu. Perbedaannya dengan lapisan *dropout* biasa adalah lapisan *dropout* biasa, menghapus elemen-elemen penting secara acak dalam vektor *embedding* yang mengakibatkan terganggunya struktur urutan data. Sedangkan lapisan *spatial dropout* 1 mampu mencegah terjadinya ketergantungan (dependence) yang berlebihan terhadap suatu fitur tertentu dan memperkenalkan variasi keacakan elemen yang bermanfaat selama proses pelatihan(training).

Pada penelitian ini, menggunakan tingkat dropout 0.1. Artinya, dalam output embedding layer yang panjangnya 300 dimensi, pada setiap langkah waktu (*time step*) akan mematikan 10% nilai dimensi vektor. Proses ini diulang pada setiap iterasi dan nilai dimatikan secara acak. Tujuannya adalah untuk mencegah terjadinya *overfitting*. *Overfitting* adalah suatu kondisi dimana model terlalu cocok dengan data training sehingga gagal dalam proses testing dengan data yang baru. Konsep penggunaan lapisan *spatial dropout* 1D adalah dengan menghilangkan(meng-dropout) beberapa informasi(elemen) tertentu pada waktu(time step) tertentu untuk membuat variasi dalam representasi data disetiap perulangan(*epoch*) dalam proses training.

3. Bidirectional LSTM Layer

Setelah proses dropout dari tahapan sebelumnya, selanjutnya adalah lapisan yang menggunakan metode algoritma *Bidirectional LSTM*. Lapisan *Bidirectional LSTM* bekerja secara dua arah, yaitu dengan cara memproses urutan input asli dan urutan input yang dibalik secara simultan. Hasil(output) dari lapisan *drop out*, akan menjadi input bagi lapisan selanjutnya(BiLSTM) dengan struktur neuron sebanyak 100 unit. Lapisan ini juga menerapkan dropout rekuren sebesar 0.1 yang artinya, dalam proses *training* disetiap langkah waktu, 10% unit neuron dimatikan. Hasil dari lapisan ini mencakup informasi masa lalu

dan masa depan yang memungkinkan model untuk dapat memahami secara konteks hubungan dalam data teks.

4. Dense Layer

Pada lapisan ini, lapisan keluaran(output) diinisiasikan. Penggunaan lapisan ini adalah menghubungkan semua neuron aktivasi dari lapisan sebelumnya menuju ke lapisan selanjutnya seperti konteks jaringan syaraf tiruan. Lapisan ini menggunakan aktivasi *softmax*, karena aktivasi jenis ini mampu melakukan klasifikasi multi label dengan cara kerjanya yang mendistribusikan untuk setiap label. Jumlah *nodes* yang dibuat disesuaikan dengan jumlah kelas tag NER.

Keluaran dari lapisan yang *fully connected layer* ini adalah matriks P berukuran $n \times k$. Dimana, k adalah jumlah tag yang akan menjadi input pada lapisan selanjutnya. Sedangkan n merujuk pada *timestep* atau panjang dari sekuens output. Output yang dihasilkan dari lapisan ini adalah output yang terletak dengan rentang nilai $0 - 1$. Jika output yang diberikan mendekati angka 1, maka bisa dipastikan pasangan tag NER pada indeks kata tersebut.

Penerapan model building yang dimulai dari input yang diberikan kemudian diteruskan ke lapisan lainnya hingga ke lapisan output di lapisan dense yang *fully connected layer* pada lapisan dense dapat dilihat pada tabel 3.23.

Tabel 3.24 Penerapan Model Building

Input Model Building(Word to Index)	Output Model Building	Tag NER(Index to Tag)
17639	0.9983	10
2985	0.9868	10
24123	0.9985	10
681	0.9971	10
20719	0.9999	10
24808	0.9927	10
28323	0.8038	10
13467	0.9378	4
21180	0.9981	10
2645	0.9962	4

Dimana, nomor index yang mewakili tag NER dalam kamus data dapat dilihat pada Contoh 8 .

Contoh 8 Pasangan Label NER Numerik dan String

"B-LOC": 0, "B-PER": 1, "I-PROD": 2, "B-PROD": 3, "I-PER": 4, "B-WA": 5, "I-LOC": 6, "I-WA": 7, "B-ORG": 8, "I-EV": 9, "B-EV": 10, "I-ORG": 11, "O": 12

Pada pemodelan NER yang menggunakan algoritma BiLSTM, fungsi optimasi yang digunakan adalah *Adam optimizer*. Fungsi optimasi ini memiliki kemampuan dalam penentuan nilai parameter yang terbaik bagi model untuk meminimalisir nilai *loss*. Keunggulan dari fungsi Adam tersebut menjadikannya sebagai fungsi optimasi yang saat ini umum digunakan dalam hal pembangunan model pembelajaran mesin. Keunggulan dari Adam optimizer merupakan kombinasi dari beberapa fungsi optimasi seperti RMSProp dan AdaGrad, yang mana fungsi-fungsi tersebut digunakan untuk memproses dataset yang besar dan memiliki parameter yang berdimensi tinggi (*high dimentional*). Tahapan dalam perhitungan menggunakan fungsi optimasi Adam digambarkan di gambar 3.16

```

Inisialisasi:  $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \eta = 10^{-8}$  (Default)
Inisialisasi:  $m_0 \leftarrow 0, v_0 \leftarrow 0, k_0 \leftarrow 0$  :

while  $\Theta_i$  #tidak konvergen do:
     $k \leftarrow k+1$ 
     $g_k \leftarrow \nabla_{\theta} f_k(\Theta_{k-1})$  # (Mengambil gradien dari step ke- k)
     $m_k \leftarrow \beta_1 \cdot m_{k-1} + (1 - \beta_1) \cdot g_k$  # (Perbaharuan informasi perkiraan awal)
     $m_k \leftarrow \beta_2 \cdot m_{k-1} + (1 - \beta_2) \cdot g_k^2$  # (Perbaharuan informasi perkiraan kedua)

    #Koreksi Bias
     $\widehat{m}_k \leftarrow \frac{m_k}{1 - \beta_1^k} \cdot g_k$  # (Kalkulasi penyesuaian bias perkiraan ke-1)
     $\widehat{v}_k \leftarrow \frac{v_k}{1 - \beta_2^k} \cdot g_k^2$  # (Kalkulasi penyesuaian bias perkiraan ke-2)

    #Pembaharuan Parameter Nilai
     $\Theta_k \leftarrow \Theta_{k-1} - \alpha \cdot \frac{\widehat{m}_k}{(\sqrt{\widehat{v}_k} + \eta)}$ 

Return  $\Theta$  #parameter yang dihasilkan
  
```

Gambar 3. 17 Pseudocode Fungsi Optimasi Adam

Dengan:

m_0 = Inisialisasi momen vektor awal

v_0 = Inisialisasi momen vektor selanjutnya

k_0 = Inisialisasi step(waktu)

β = Inisialisasi bias

α = inisialisasi rentang pembelajaran (learning rate)

η = inisialisasi epsilon

Penjelasan mengenai gambar 3.17 adalah fungsi adam *optimizer* melakukan perubahan(*update*) secara berulang terhadap parameter dari jaringan syaraf berdasarkan data training. Perhitungan terhadap bias momen vektor pertama dan selanjutnya yang dilakukan pada fungsi Adam menghasilkan kalkulasi kecepatan pembelajaran adaptif independen selama proses training.

Model NER yang tujuannya adalah untuk menghasilkan klasifikasi banyak kelas label NER terhadap suatu token kata, maka fungsi untuk menghitung *loss* yang digunakan adalah *categorical cross entropy*. Rumus perhitungan nilai *loss* dari *categorical cross entropy* dapat dilihat pada persamaan 3.1

$$H(\chi) = \begin{cases} -\int_x p(x) \log p(x), & \text{if } X \text{ kontinu} \\ -\sum_x p(x) \log p(x), & \text{if } X \text{ diskrit} \end{cases} \quad 3.1$$

Dengan:

χ = Variabel acak kontinu

$p(x)$ = distribusi probabilitas aktual dari variabel acak

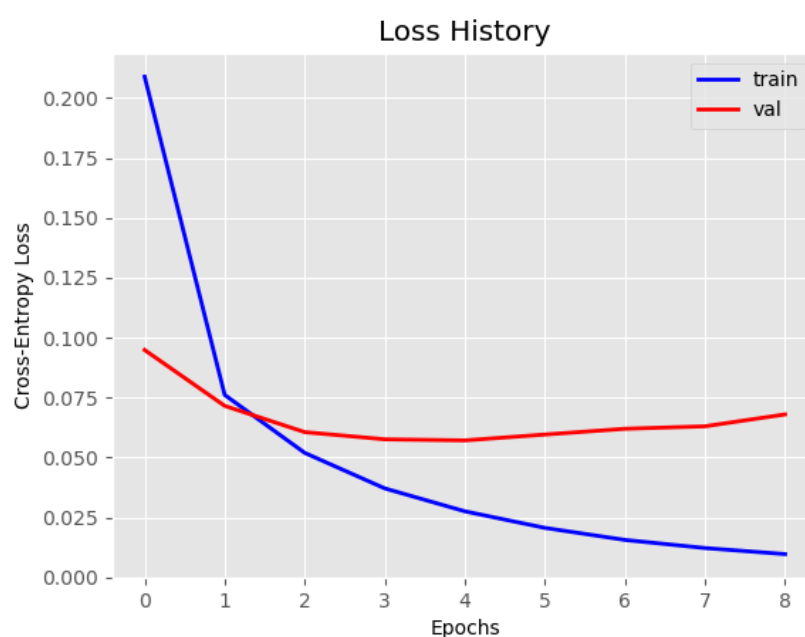
Proses training dilakukan dalam beberapa kali perulangan agar model dapat mempelajari data dengan baik. Sebelum proses training, dilakukan penentuan nilai pada *batch size*, *epoch* yang dalam hal ini bergantung pada kemampuan perangkat dan data yang digunakan. Pada penelitian ini *epoch* yang digunakan adalah 15 dengan ukuran *batch* sebesar 32 dan jumlah data yang dilatih adalah 4768 data. Untuk menghitung jumlah langkah iterasi yang akan dilakukan dalam 1 *epoch*, dapat dilihat pada persamaan 3.2

$$\text{Jumlah iterasi} = \frac{\text{ukuran batch}}{\text{jumlah data pelatihan}} \quad 3.2$$

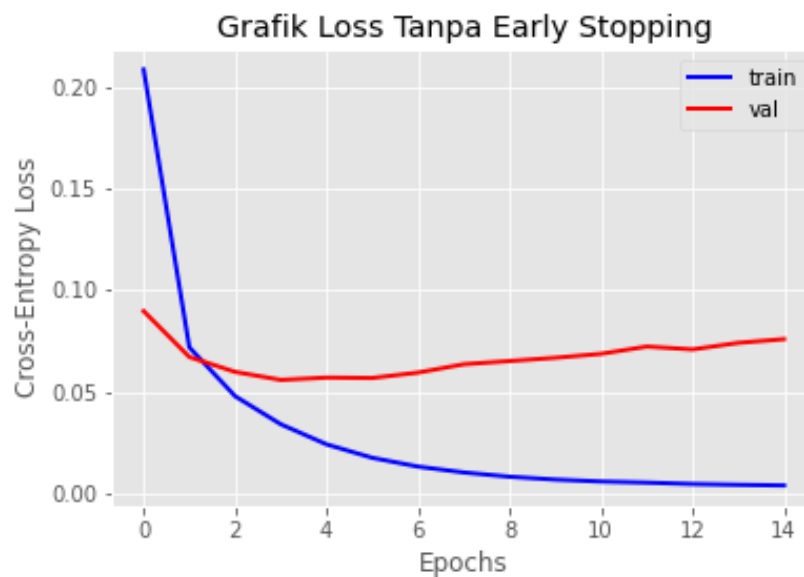
Dengan menggunakan rumus perhitungan dari persamaan 3.2, jumlah iterasi dalam 1 epoch dapat diperoleh sebanyak 149 iterasi.

Demi menghindari terjadinya *overfitting*, yaitu suatu kondisi dimana model pembelajaran mesin terlalu rumit dan terlalu sesuai dengan data latih. Hal ini

mengakibatkan model mengalami penurunan performa pada data yang baru, sehingga tidak mampu memperoleh hasil yang baik pada data testing. Untuk itu, dilakukan penambahan teknik regularisasi, *early stopping*. Teknik ini berfungsi sebagai *callbacks*, yang akan memberhentikan proses pembelajaran. Teknik ini dipanggil ketika nilai kerugian mencapai nilai minimum pada data validasi dan tidak adanya peningkatan akurasi pada *epoch* berikutnya sebagaimana yang diharapkan. Gambar 3.18 dan 3.19. menunjukkan bagaimana jika pemodelan dibangun dengan dan tanpa penggunaan teknik *callbacks*.



Gambar 3.18 Grafik Loss Dengan Early Stopping

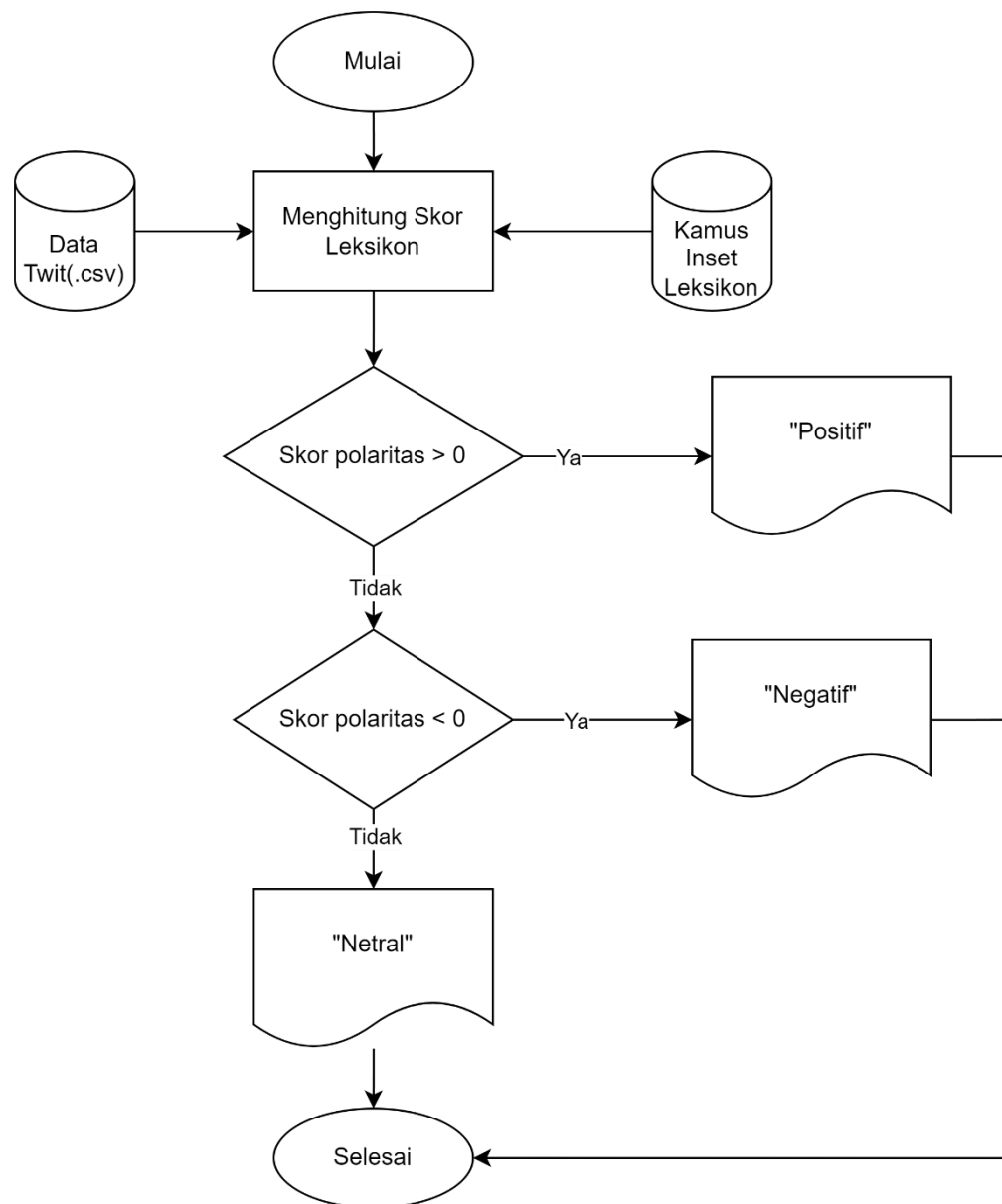


Gambar 3.19 Grafik Loss Tanpa Early Stopping

Dengan melihat gambar 3.19 nilai *loss* pada data latih semakin mengecil, hal ini berbanding terbalik dengan nilai *loss* pada data validasi yang justru semakin meningkat hingga di akhir epoch. Keadaan inilah yang disebut overfitting pada model. Sedangkan pada gambar 3.17, dimana pelatihan berhenti pada epoch ke-8 menunjukkan bahwa *early stopping* mencegah adanya kenaikan nilai *loss* pada epoch berikutnya.

3.2.6 Klasifikasi Sentimen dengan Metode InSet Lexicon

Setelah melakukan anotasi pada data dan diperoleh informasi terstruktur dari tweet, maka selanjutnya adalah melakukan klasifikasi sentimen terhadap data tersebut. Klasifikasi data dilakukan menggunakan pendekatan Lexicon Based Dictionary. Metode ini menerapkan penggunaan dari kamus leksikon berbahasa Indonesia pada tabel 2.1. Alur proses klasifikasi sentimen dengan menggunakan metode *inset lexicon* diuraikan pada gambar 3.20.



Gambar 3.20 Alur Klasifikasi Sentimen InSet Lexicon

Penjelasan dari gambar 3.30 adalah data twit yang sudah dilakukan pra pengolahan selanjutnya dilakukan pemeriksaan terhadap setiap token kata pada data dengan kamus kata positif dan negatif pada tabel 2.2 dan 2.3. Bila ditemukan kata dalam kamus, maka skor polaritas pada kumpulan kata akan dihitung. Implementasi dari metode *lexicon based dictionary* ditampilkan pada tabel 3.24

Tabel 3. 25 Implementasi Metode Lexicon Based Dictionary

Kata InSet Lexicon	Skor	Total Skor	Label
Hidup / sedih / secaper / apa / nyamperin / ig / youtube / tim / Cuma / buat / katai / asa / gua / fan / bola / orang / santai / eh / ml / malah / begini	-4/-5/0/-3/0/0/0/- 4/-3/1/0/-2/- 3/0/0/0/2/0/0/0/-1	-22	Negatif

Selanjutnya jumlah skor kata pada tabel 3.24 akan dimasukkan ke perhitungan skor *dictionary based / lexicon based*. Formula perhitungan skor sentimen yang dipakai ditunjukkan pada tabel 2.4.1. Jika, total skor mencapai angka lebih kecil dari nol, maka data diberi label negatif. Jika, total skor mencapai angka lebih dari 0, maka data dikategorikan positif. Jika skor total data berjumlah sama dengan 0, maka, data dikategorikan netral. Masukkan yang dijadikan sebagai input adalah kata yang sudah melalui proses stemming pada 3.2.2.11. Sehingga nilai sentimen yang dihasilkan pada input adalah negatif.

3.2.7 Output

Adapun output yang dihasilkan pada penelitian ini adalah data twit sudah dianotasi dan berhasil diklasifikasikan jenis sentimennya.

Contoh data: “Hidup sesedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain, perasaan dari dulu gua ngefans bola orang pada santai eh di ml! malah gini”

Tabel 3.26 Sentimen Analisis dan Anotasi Otomatis

Teks Original	Hidup sesedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain , perasaan dari dulu gua ngefans bola orang pada santai eh di ml ! malah gini
Label Sentimen	negative
Kalimat NER	hidup sesedih dan secaper apa yak nyamperin [ig] atau [youtube] [youtube] tim lain cuma buat ngatain , perasaan dari dulu gua ngefans bola orang pada santai eh di ml ! malah gini
Entitas	PROD: [ig] [youtube] tim

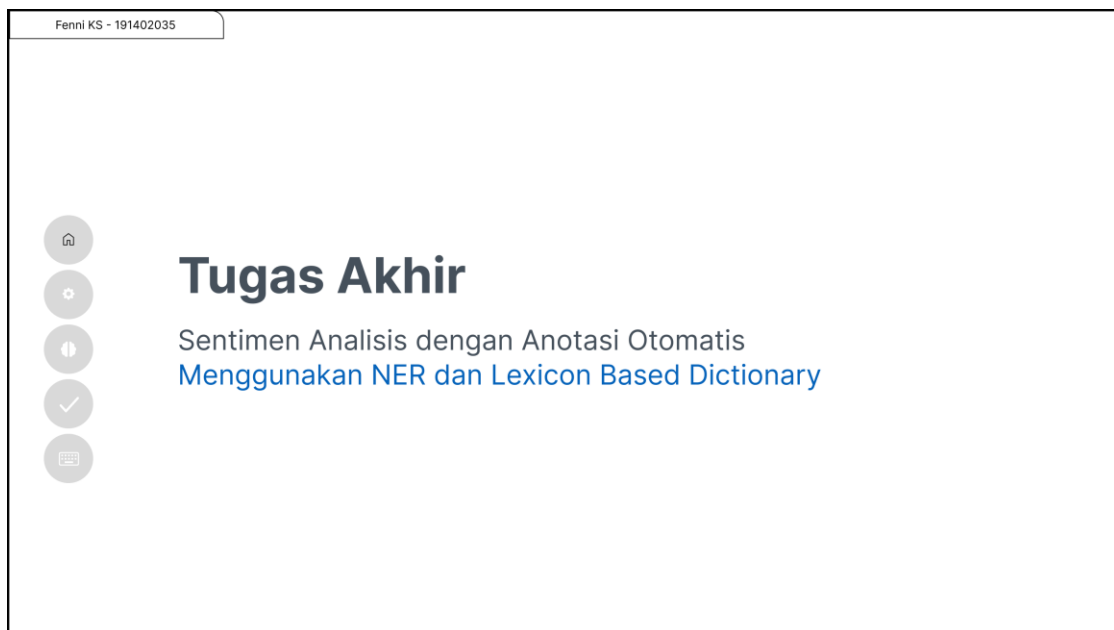
3.3 Perancangan Aplikasi Sistem

Tahap ini menjelaskan perancangan antarmuka analisis sentimen otomatis dan sistem anotasi menggunakan pendekatan NER dan kamus bahasa Indonesia. Tujuan dari

perancangan antarmuka sistem ini adalah untuk memberikan gambaran umum dan memudahkan pengguna dalam menavigasi sistem.

3.3.1 Tampilan Antar Muka Halaman Utama

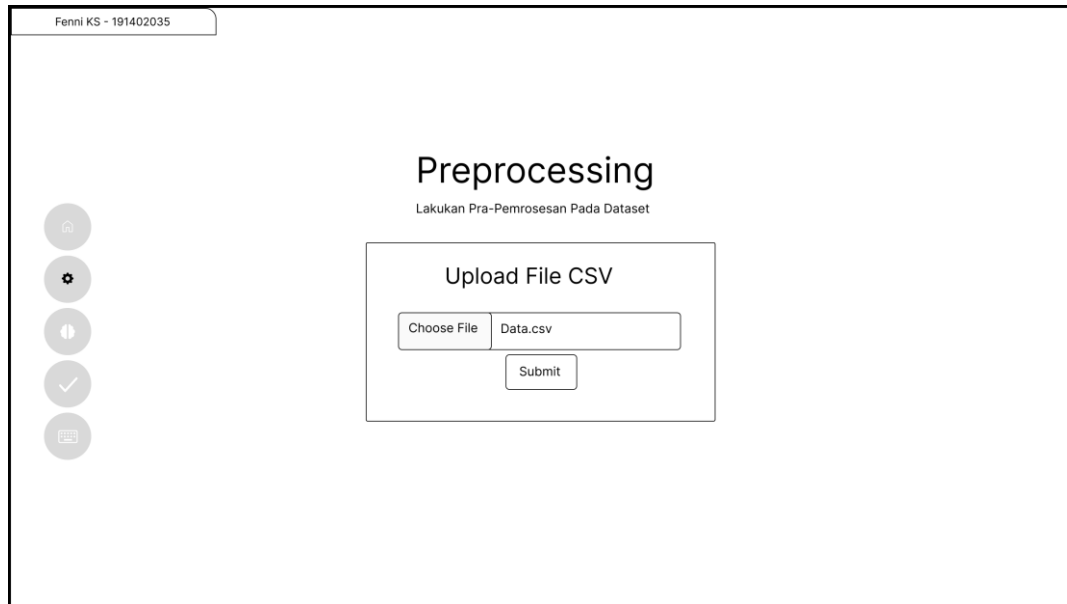
Pertama kali saat sistem dijalankan, pengguna akan ditampilkan halaman utama. Pada halaman ini berisi penjelasan atau informasi singkat mengenai penelitian yang dilakukan. Informasi yang diberikan tersebut meliputi, judul penelitian. Selain itu terdapat fitur sebuah tombol yang berfungsi sebagai penghubung yang menghubungkan ke menu yang lain. Tampilan halaman utama diilustrasikan pada gambar 3.20.



Gambar 3.21 Tampilan Rancangan Halaman Utama

3.3.2 Tampilan Antar Muka Halaman Preprocessing

Tampilan antar muka dari menu proses *preprocessing* adalah antar muka yang ditunjukkan saat meng-klik tombol dengan ikon *gear* dibawah ikon *home*. Pada menu preprocessing akan dilakukan upload file data twit dengan ekstensi *Comma-Separated Values*(.csv) terhadap sistem. Pada menu ini menghasilkan data training dan data testing degan format CSV yang dapat diunduh. Tampilan halaman *preprocessing* dapat diilustrasikan pada gambar 3.22.



Gambar 3.22 Tampilan Rancangan Halaman Preprocessing

3.3.3 Tampilan Antar Muka Halaman Training

Tampilan antar muka menu proses *training* adalah halaman yang ditampilkan saat meng-klik tombol dengan ikon otak(*brain*) dibawah ikon *gear*. Pada menu training akan dilakukan upload file data twit dengan ekstensi *Comma-Separated Values(.csv)* terhadap sistem. Hasil keluaran dari menu ini adalah data training yang sudah melalui proses pelatihan dan diberi label sentimen dari proses sentimen analisis. Hasil training juga dapat diunduh dan memiliki format CSV. Serta grafik *loss* dan *accuracy*. Tampilan halaman training diilustrasikan pada gambar 3.23.

Fenni KS - 191402035

Named Entity Recognition & Sentimen Analisis Training

Model NER BiLSTM & Analisis Sentimen Lexicon Based Dictionary

Masukkan Data Training
Data yang dimasukkan dengan ekstensi.csv

Choose File data_train.csv Submit

Gambar 3.23 Tampilan Rancangan Halaman Training

3.3.4 Tampilan Antar Muka Halaman Testing

Tampilan antar muka dari proses *testing* adalah halaman yang ditampilkan saat mengklik tombol dengan ikon ceklis dibawah ikon *brain*. Pada menu testing akan dilakukan upload file data twit dengan ekstensi *Comma-Separated Values(.csv)* terhadap sistem. Hasil keluaran dari menu ini adalah data testing yang sudah melalui proses testing terhadap model NER dan diberi label sentimen dari proses sentimen analisis. Di menu ini juga terdapat visualisasi *confusion matrix* untuk menghitung prediksi NER yang dihasilkan oleh model. Hasil testing juga dapat diunduh dan memiliki format CSV. Tampilan halaman testing dapat diilustrasikan pada gambar 3.24.

Gambar 3.24 Tampilan Rancangan Halaman Testing

3.3.5 Tampilan Antar Muka Halaman User Input

Tampilan antar muka menu proses masukan pengguna(*user input*) adalah tampilan yang muncul saat meng-klik tombol dengan ikon otak(*brain*) dibawah ikon *gear*. Pada menu *user input*, pengguna akan memasukkan tweet secara acak untuk diinput ke dalam sistem. Hasil keluaran tweet asli yang diinputkan oleh user, kalimat yang terdeteksi label NER oleh sistem, entitas NER dan label sentimen dari proses sentimen analisis. Tampilan halaman *user input* ditampilkan pada gambar 3.25.

Gambar 3.25 Tampilan Rancangan Halaman User Input

3.4 Metode Evaluasi

Untuk mengukur seberapa baik performa sistem dalam mengidentifikasi label NER dan sentimen pada data twit maka, perlu dilakukan evaluasi. Metode yang digunakan untuk mengukur kinerja sistem dalam pemodelan NER adalah metrik *F1-score*. Termasuk kedua elemen yang ada dalamnya yaitu, *precision*, dan *recall*. Sedangkan untuk mengukur performa sentimen analisis digunakan metode evaluasi manual yang dikombinasikan dengan *confusion matrix*.

3.4.1 *Precision, Recall dan F1-score*

Penerapan metode evaluasi dengan menghitung nilai *precision*, *recall* dan *f1-score* sangat penting dilakukan untuk menghitung performa model NER BiLSTM. Hal ini karena persebaran kelas target yang timpang dalam jumlah per tag-nya. Informasi yang disajikan oleh tabel 3.2, memperlihatkan bahwa antara jumlah token kata dengan masing-masing tag NER tidak memiliki persebaran yang seimbang. Tag “EV” dan “WA” memiliki jumlah yang kecil dibandingkan tag NER lainnya.

Jika dalam hal ini hanya digunakan metode evaluasi dengan menghitung akurasi dari performa model saja, maka, hasilnya hanya akan menghitung performa model yang bekerja pada data secara keseluruhan. Sedangkan evaluasi kinerja model NER juga harus dilakukan terhadap tag secara spesifik, untuk mengukur kemampuan model dalam mengidentifikasi entitas NER yang relevan dan penting dalam teks. Itulah sebabnya penting untuk menggunakan metode evaluasi yang lebih komprehensif seperti *precision*, *recall*, dan *f1-score* untuk mengevaluasi kinerja sistem NER secara keseluruhan. Masing-masing penjelasan dari setiap elemen dijelaskan sebagai berikut.

1. *Precision*

Precision digunakan untuk mengukur kemampuan model dalam mengidentifikasi token kata dengan label sebenarnya dibandingkan dengan seluruh token kata yang berhasil diidentifikasi label entitasnya. Caranya adalah dengan menghitung jumlah token kata yang hasil prediksi label NER-nya relevan dengan label NER yang seharusnya. Jika nilai *precision* yang diperoleh tinggi, artinya model NER hanya memiliki sedikit *false positive*. Rumus perhitungan untuk melakukan kalkulasi dari skor *precision* dapat dilihat pada persamaan 3.3.

$$\text{Precision(P)} = \frac{\text{True Positives(TP)}}{\text{True Positives(TP)} + \text{False Positives(FP)}} \quad 3.3$$

2. Recall

Recall bertugas digunakan untuk mengukur kemampuan model dalam mengidentifikasi token kata yang harusnya memiliki label entitas dibandingkan dengan seluruh token kata yang sebenarnya memiliki entitas. Jika nilai *recall* yang diperoleh tinggi, artinya model NER hanya memiliki sedikit *false negative*. Rumus dari perhitungan untuk melakukan kalkulasi dari skor *recall* dapat dilihat pada persamaan 3.4.

$$\text{Recall(R)} = \frac{\text{True Positives(TP)}}{\text{True Positives(TP)} + \text{False Negatives(FN)}} \quad 3.4$$

3. F1-Score

F1-score merupakan metrik evaluasi yang memberikan gambaran keseluruhan mengenai kinerja model NER dalam mengidentifikasi label entitas bernama terhadap data. Caranya adalah dengan menggabungkan hasil perhitungan nilai *precision* dan *recall*. Kalkulasi dari pengukuran skor *f1-score* dapat dilihat pada persamaan 3.5.

$$\text{F1 - score(F1)} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad 3.5$$

3.4.2 Evaluasi Manual Dengan Akurasi

Untuk mengetahui performa kinerja *InSet Lexicon* terhadap sentimen analisis pada data Twitter, dilakukan evaluasi gabungan yaitu evaluasi secara manual dengan perhitungan akurasi. Evaluasi manual ini dilakukan dengan memberikan sampel data *twit* yang sudah berhasil diperoleh label sentimennya oleh sistem lalu, data tersebut diatonasi oleh beberapa anotator. Jika label yang dihasilkan oleh sistem dinilai tidak sesuai oleh pakar maka, pakar menambahkan label baru yang sesuai.

Pakar yang menilai hasil sentimen analisis yang dihasilkan oleh sistem merupakan 5(lima) orang tenaga pengajar mata pelajaran bahasa Indonesia di Sekolah Menengah Atas(SMA). Nama dan lokasi mengajar dari setiap anotator dapat dijelaskan pada tabel 3.26

Tabel 3. 27 Profil Anotator(Pakar)

No.	Nama	Sekolah
1	Bapak Saipul Pujakesuma	SMA NEGERI 13 MEDAN
2	Ibu Irma Suryani Lubis	SMA NEGERI 13 MEDAN
3	Ibu Ayu Retno Antika	SMA NEGERI 13 MEDAN
4	Ibu Nurhesti Rahmadani	SMA NEGERI 13 MEDAN
5	Wira Swasti Buulolo	SMA SWASTA RIZKY ANANDA

Setelah divalidasi oleh pakar, kemudian hasil label sentimen yang diperoleh sistem dan yang divalidasi oleh pakar dihitung menggunakan metrik akurasi. Metrik evaluasi secara umum memiliki empat elemen dasar yaitu, *true positive*, *true negative*, *false positive* dan *false negative*. Setiap elemen dari keempat elemen dasar tersebut menjelaskan tentang performa sistem dalam menghasilkan label sentimen secara otomatis yang dibandingkan dengan label sebenarnya dari hasil validasi oleh pakar. Adapun definisi dari keempat elemen metrik akurasi adalah sebagai berikut.

1) *True Positive*(TP)

Merupakan elemen yang menghitung jumlah label data kelas positif yang dihasilkan oleh sistem dan merupakan label sebenarnya pada data.

2) *True Negative*(TN)

Merupakan elemen yang menghitung jumlah label data kelas negatif yang dihasilkan oleh sistem dan merupakan label sebenarnya pada data.

3) *False Positive*(FP)

Merupakan elemen yang menghitung jumlah label data kelas positif yang dihasilkan oleh sistem akan tetapi sebenarnya merupakan data dengan kelas negatif.

4) *False Negative*(FN)

Merupakan elemen yang menghitung jumlah label data kelas negatif yang dihasilkan oleh sistem akan tetapi sebenarnya merupakan data dengan kelas positif.

Adapun cara untuk melakukan perhitungan nilai akurasi yaitu, dengan menghitung total skor dari *true positive* dan *true negative* kemudian dibagi dengan jumlah data. Untuk menghitung nilai akurasi dapat dilihat pada persamaan 3.6.

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar(TP + TN)}{Jumlah\ Data(TP + TN + FP + FN)} \times 100 \quad 3.6$$

Dari persamaan 3.6, dapat dilakukan pengukuran terhadap kinerja kamus leksikon bahasa Indonesia dalam hal memberikan prediksi label positif, negatif, maupun netral terhadap data twit.

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Eksekusi dari setiap perancangan pada BAB III, diimplementasikan menjadi sistem yang utuh pada bagian ini. Implementasi sistem akan menggunakan alat pendukung seperti, komponen perangkat fisik (hardware) dan non fisik (software). Berikut ini menjelaskan mengenai perangkat-perangkat yang digunakan dalam implementasi sistem.

4.1.1 Spesifikasi Perangkat Keras dan Perangkat Lunak

Pembangunan sistem pada penelitian ini menggunakan perangkat keras dengan spesifikasi sebagai berikut.

1. *Procesor* AMD Ryzen 5
2. Memori (*RAM*): 4 GB
3. *SSD* 512 GB

Perangkat lunak yang dipakai memiliki spesifikasi sebagai berikut.

1. Sistem Operasi: Windows 10 Home 64 Bit.
2. Python versi 3.10
3. Pustaka(*Library Python*) yang digunakan, antara lain:
 - a. Flask
 - b. NumPy
 - c. Pandas
 - d. Seaborn
 - e. Matplotlib
 - f. Openpyxl
 - g. Scikit-learn

- h. Keras
- i. TensorFlow
- j. NLTK
- k. Sastrawi
- l. Emoji
- m. tqdm
- n. Ipython
- o. Plot Model

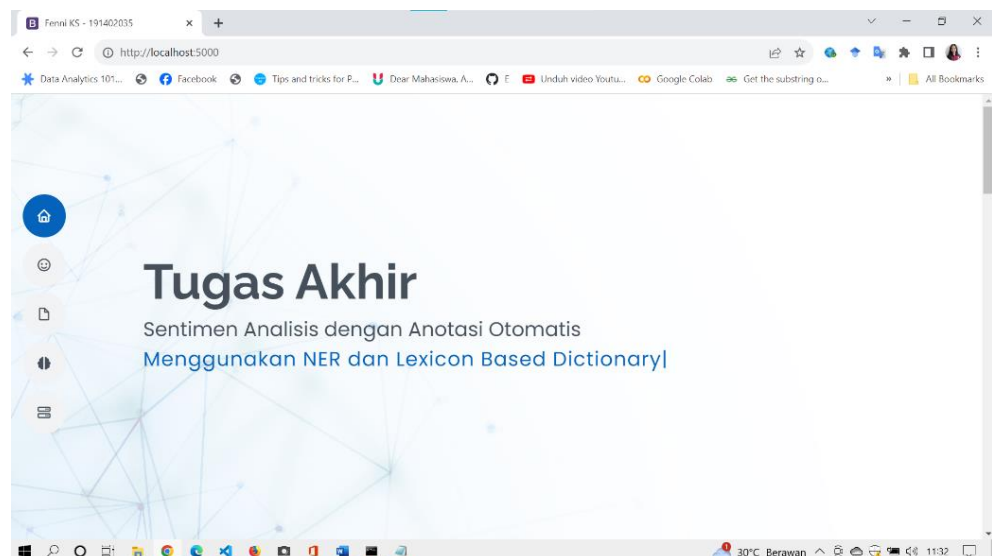
4. *Text Editor*: Visual Studio Code

4.1.2 *Implementasi Rancangan Antarmuka*

Implementasi hasil rancangan halaman tampilan dari setiap proses pada sistem dipenelitian ini adalah sebagai berikut.

1. Tampilan Halaman Utama

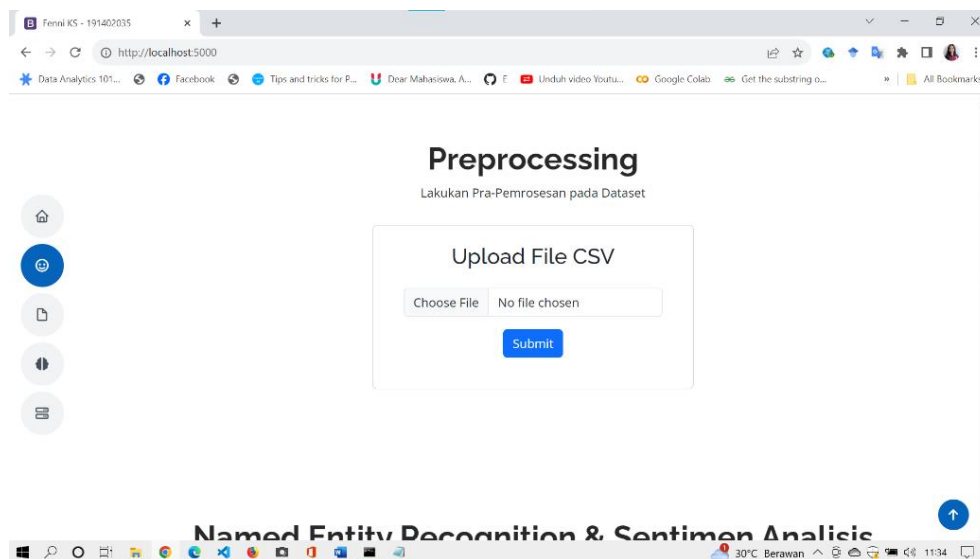
Implementasi dari rancangan halaman utama pada 3.3.1 yang fiturnya adalah memuat informasi yang terkait dengan penulis, karya penelitian dan beberapa fitur yang menghubungkan ke menu-menu lainnya. Tampilan halaman utama(beranda) ditunjukkan pada gambar 4.1.



Gambar 4.1 Tampilan Halaman Utama

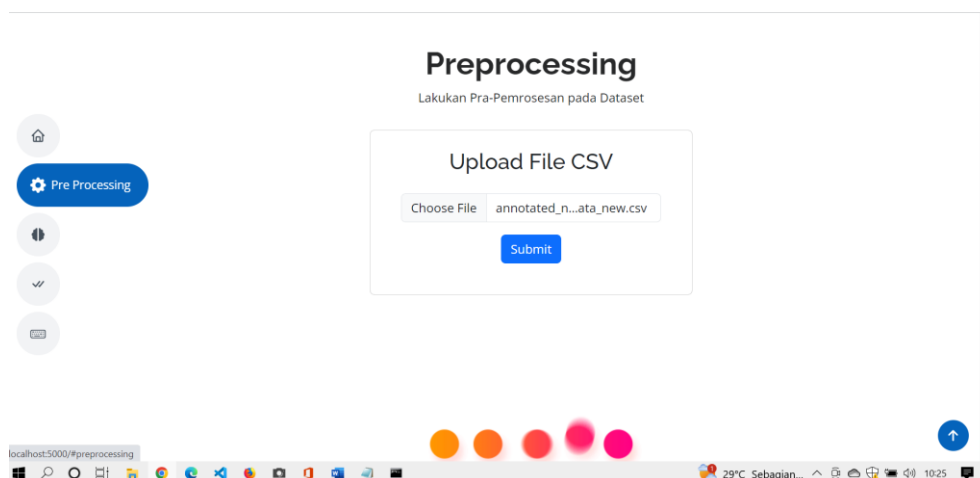
2. Tampilan Halaman Preprocessing

Pada tampilan antar muka *preprocessing* seperti penjelasan pada 3.3.2, pengguna diminta untuk memasukkan data yang akan dilakukan proses prapengolahan. Setelah dimasukkan, pengguna menekan tombol ‘submit’ agar dapat memulai proses prapengolahan. Pada menu ini, pengguna bisa mengunduh data latih dan data testing dengan format CSV. Tampilan implementasi antar muka menu *preprocessing* dapat dilihat pada gambar 4.2.



Gambar 4.2 Tampilan Halaman Preprocessing

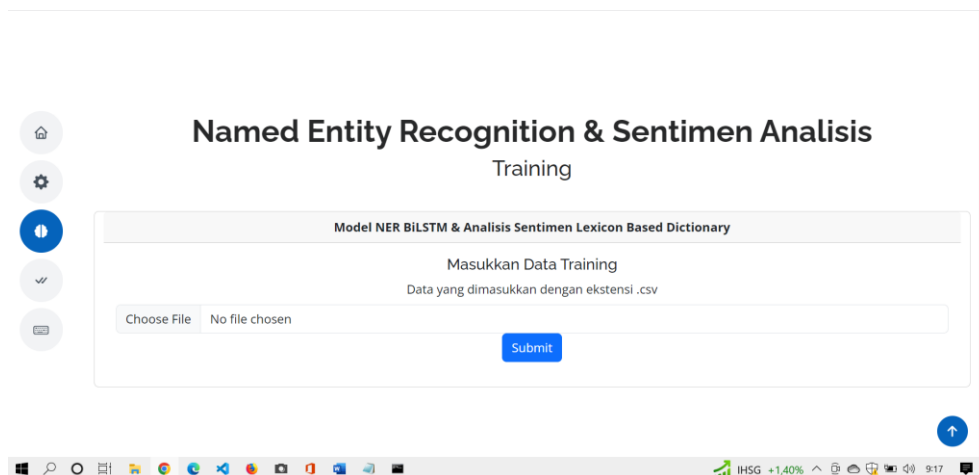
Ketika data sudah dimasukkan dan disubmit maka, proses prapengolahan sedang berlangsung. Pada bagian ini, tampilan akan berubah dengan menampilkan gambar loading seperti ditampilkan di gambar 4.3.



Gambar 4.3 Tampilan Halaman Proses Preprocessing

3. Tampilan Halaman Training.

Dalam melakukan proses training, pengguna akan diarahkan untuk memasukkan data latih dengan ekstensi CSV. Sesuai dengan rancangan yang dijelaskan pada 3.3.3, tampilan halaman training terdapat tombol ‘Submit’ sebagai tombol untuk memulai proses *training* model NER serta sentimen analisis dengan kamus leksikon. Antar muka halaman training ditunjukkan pada Gambar 4.6.



Gambar 4. 6 Tampilan Halaman Training

Saat proses pelatihan data selesai dijalankan, maka akan muncul tabel yang berisi data training yang sudah diberi label sentimen oleh kamus leksikon bahasa Indonesia. Data tersebut dapat diunduh. Tampilan tabel data hasil proses *training* dapat dilihat pada gambar 4.7.

Hasil Sentimen Analisis dengan Lexicon Based Dictionary
Kamus InSet Lexicon

Unduh Hasil Training Search:

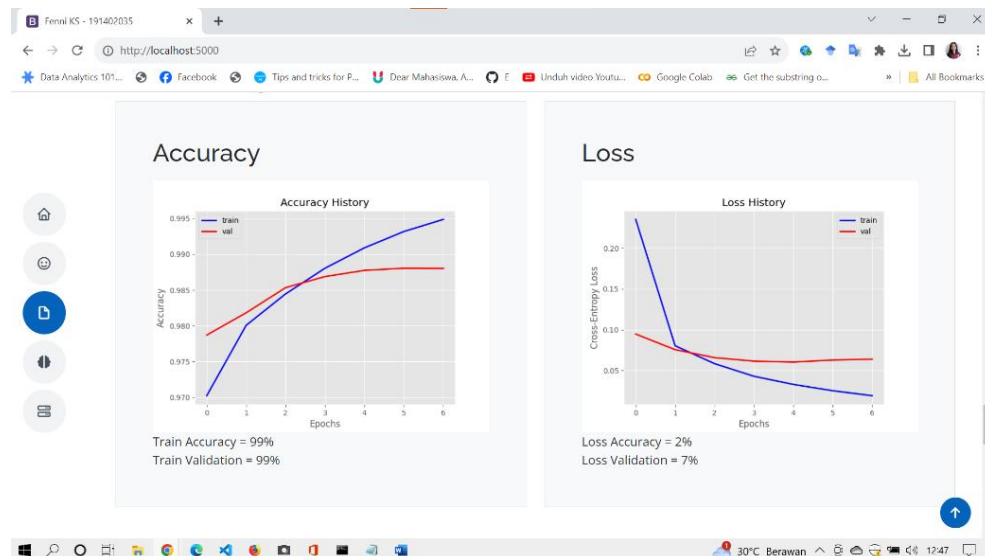
Tweet Id	Text	Bobot Sentiment	Label
tweet_id:1004	PELINDUNG KORUPTOR IKUT2AN JADI KORUPTOR ! JawaPos : Curi Emas 1,9 Kg dari Barang Bukti , Pegawai KPK Langsung Dipecat . https://t.co/h0fEOCVz7m #FPidaniBHRSbukanTerroris #PKIMusuhSemuaAgama @GoogleNews	-16	negative
tweet_id:1006	Daerah Borobudur tempat wisata sek spot fotone apik ngendi yo	-3	negative
tweet_id:1007	RT @DreamOfeuMe : Idk udah pada tau/belum . Berhubung banyak banget kabar meresahkan tentang masker di platform chatting sebelah , yang mau beli masker cek di web info alkes dari kementrian ya . Manteman bisa cek produsen/brand yang sudah dapet izin buat produksi masker . https://t.co/85sz31wuQ2	-23	negative

Showing 1 to 10 of 5,940 entries Previous 1 2 3 4 5 ... 594 Next

Gambar 4.7 Tampilan Data Hasil Training

Selain terdapat data hasil pelatihan, bagian ini juga ditampilkan dua grafik yaitu, grafik loss dan akurasi terhadap data *training* dan validasi selama proses

pelatihan data berlangsung hingga di akhir *epoch*. Tampilan grafik *loss* dan akurasi dari proses pelatihan model terhadap data dapat dilihat pada gambar 4.8.



Gambar 4.8 Grafik Hasil Proses Training (Akurasi dan Loss)

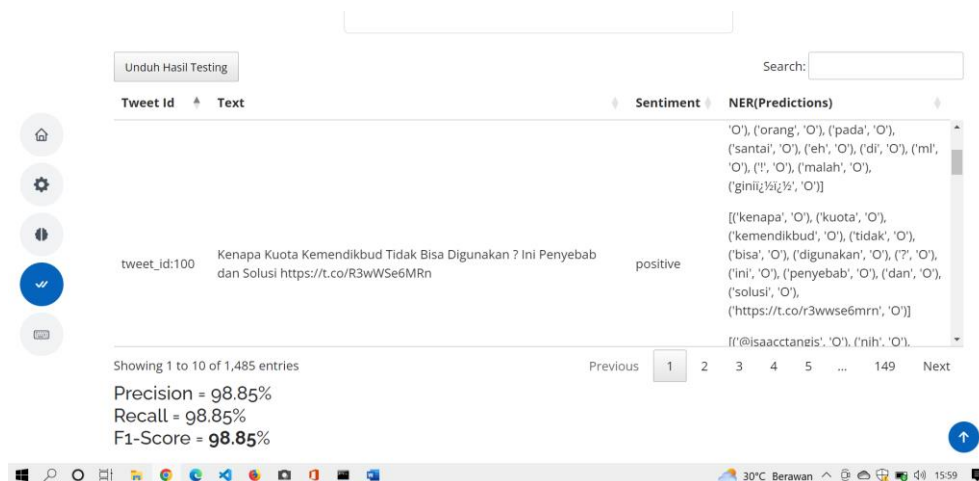
4. Tampilan Halaman Testing

Dalam melakukan proses testing, pengguna akan diarahkan untuk memasukkan data testing dengan ekstensi CSV. Sesuai dengan rancangan yang dijelaskan pada 3.3.4, tampilan halaman testing terdapat tombol ‘Submit’ sebagai tombol untuk memulai proses *testing* model NER serta sentimen analisis dengan kamus leksikon. Antar muka dari halaman proses pengujian(*testing*) ditampilkan digambar 4.9.

The screenshot shows the 'Testing' page of the application. It features a header 'Testing' and a sub-header 'Testing Model NER BiLSTM'. Below this, there is a prompt 'Masukkan Data Testing' and a note 'Data yang dimasukkan dengan ekstensi .csv'. A file selection area shows 'Choose File' and 'No file chosen'. A blue 'Submit' button is located at the bottom right of the form area.

Gambar 4.9 Tampilan Halaman Testing

Ketika proses *testing* selesai dilakukan, maka tampilan akan berubah dengan menampilkan data tabel yang berisi data yang sudah diprediksi NER oleh model NER BiLSTM. Serta data juga sudah diberi label oleh kamus leksikon bahasa Indonesia. Keluaran pada halaman testing ditampilkan digambar 4.10.



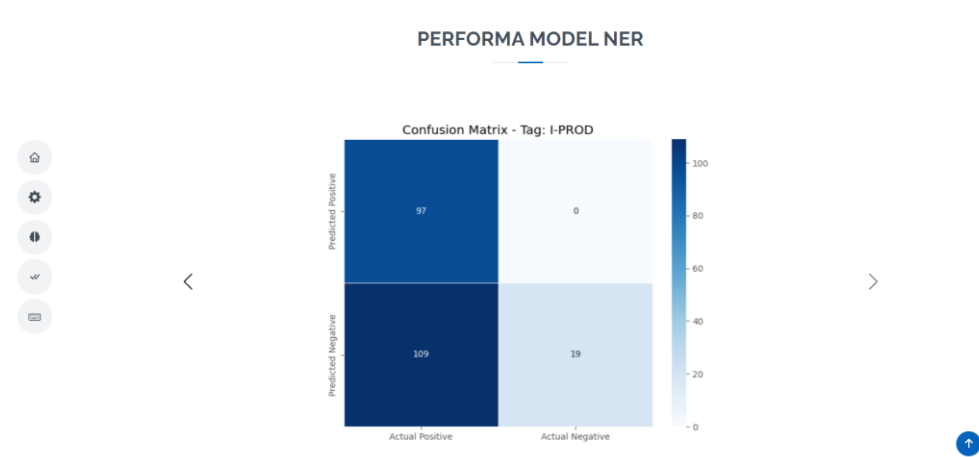
The screenshot shows a web application interface for testing NER results. It includes a search bar, a table with columns for Tweet Id, Text, Sentiment, and NER(Predictions), and a sidebar with navigation icons. The table displays one entry for tweet_id:100 with a positive sentiment and various NER predictions. Performance metrics are shown at the bottom: Precision = 98.85%, Recall = 98.85%, and F1-Score = 98.85%.

Tweet Id	Text	Sentiment	NER(Predictions)
tweet_id:100	Kenapa Kuota Kemendikbud Tidak Bisa Digunakan ? Ini Penyebab dan Solusi https://t.co/R3wWSe6MRn	positive	['O'], ('orang', 'O'), ('pada', 'O'), ('santai', 'O'), ('eh', 'O'), ('di', 'O'), ('ml', 'O'), ('!', 'O'), ('malah', 'O'), ('gini', 'O'), ('kenapa', 'O'), ('kuota', 'O'), ('kemendikbud', 'O'), ('tidak', 'O'), ('bisa', 'O'), ('digunakan', 'O'), ('?', 'O'), ('ini', 'O'), ('penyebab', 'O'), ('dan', 'O'), ('solusi', 'O'), ('https://t.co/r3wwse6mrn', 'O'), ('@isaacctangis', 'O'), ('nih', 'O')]

Precision = 98.85%
Recall = 98.85%
F1-Score = 98.85%

Gambar 4.10 Tampilan Halaman Hasil Testing

Performa model terhadap masing-masing *tag NER* juga dapat dilihat pada gambar slide *confusion matrix* tiap tabel yang ditampilkan pada halaman testing. Gambar confusion matrix setiap tag NER dapat dilihat pada gambar 4.11.



Gambar 4. 11 Tampilan Slide Confusion Matrix

5. Tampilan Halaman User Input

Untuk menganotasi entitas dan label dari data twit secara acak dari pengguna, maka pengguna harus kutipan twit. Lalu terdapat tombol yang digunakan untuk

memulai proses pengolahan data. Sesuai dengan rancangan pada 3.3.5, implementasi antar muka pada menu *user input* dapat dilihat pada gambar 4.12.

Gambar 4.12 Tampilan Halaman User Input

Ketika kutipan twit dimasukkan dan proses sistem berjalan dan selesai, maka tampilan akan berubah dengan menampilkan hasil dari pengolahan sistem. Tampilan akan berisi teks original yang dimasukkan, kalimat NER yang mendeteksi apakah ada entitas dalam kutipan twit, entitas NER serta label sentimen. Tampilan halaman hasil proses *user input* dapat dilihat pada gambar 4.13.

Teks Original	Hidup sesedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain , perasaan dari dulu gua ngefans bola orang pada santai eh di ml ! malah giniilz½z½
Label Sentimen	negative
Kalimat NER	hidup sesedih dan secaper apa yak nyamperin [ig] atau [youtube] [youtube] tim lain cuma buat ngatain , perasaan dari dulu gua ngefans bola orang pada santai eh di ml ! malah giniilz½z½
Entitas	PROD: [ig] [youtube] tim

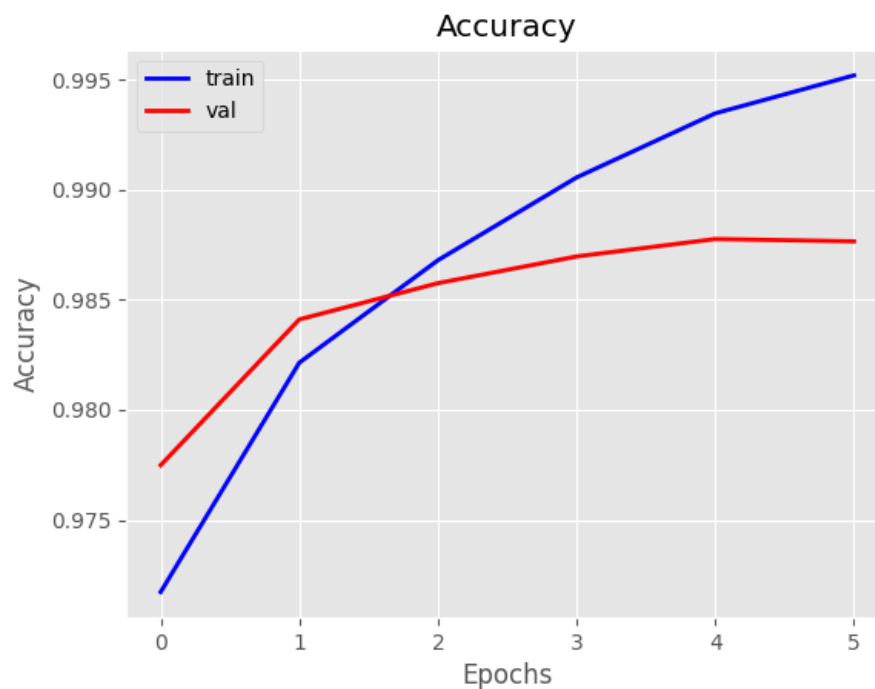
Gambar 4.13 Tampilan Halaman User Input(Hasil)

4.2 Implementasi Model

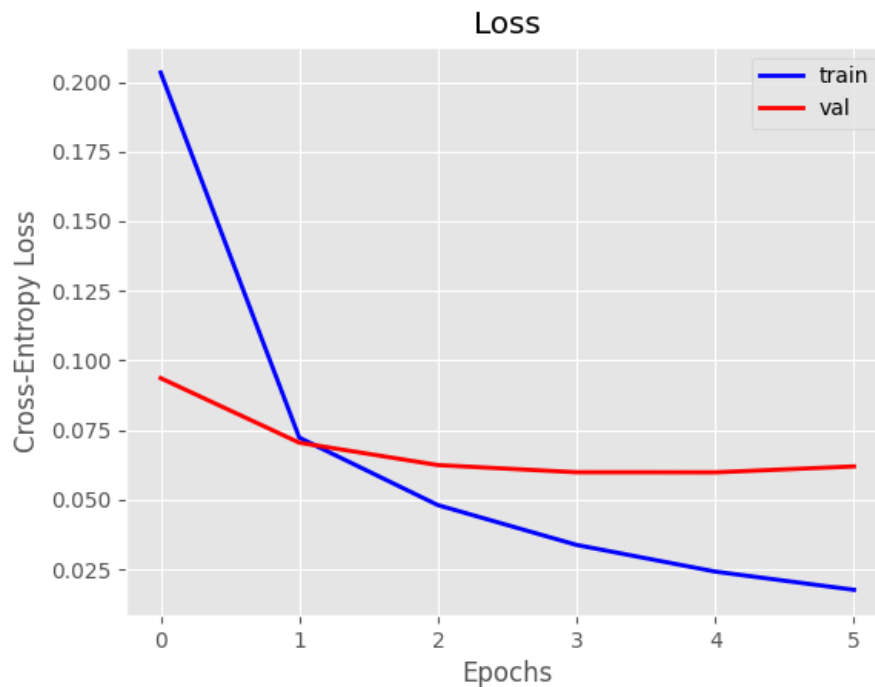
4.2.1 Pelatihan Model

Pelatihan model dilakukan oleh penulis dengan sebelumnya melakukan hyperparameter tuning. Proses ini adalah melakukan beberapa kali proses *training* dengan mengubah

jumlah *epoch*, *batch size* serta jumlah unit neuron. Tujuan dilakukanya *hyperparameter tuning* ini adalah untuk mendapatkan kinerja model yang baik. Model dikatakan memiliki kinerja atau performa yang baik apabila tidak *overfitting*. Maka dari itu untuk menghindari kondisi tersebut, penulis menambahkan lapisan *drop out* dan fungsi *early stopping*. Dalam proses *hyperparameter tuning*, penulis mendapatkan model dengan performa terbaik dengan parameter jumlah neuron 100, ukuran menggunakan ukuran sampel data tiap iterasi (*batch size*) sebesar 32, dan jumlah *epoch* sebesar 15. Proses pelatihan model terhadap data latih dapat dilihat pada gambar grafik akurasi dan loss terhadap data latih dan validasi. Gambar grafik akurasi pada data latih dan validasi dapat digambarkan digambar 4.14 (grafik akurasi).



Gambar 4. 14 Grafik Akurasi



Gambar 4.15 Grafik Loss

4.3 Pengujian Model

Setelah proses pelatihan selesai dilakukan pada model, langkah selanjutnya adalah proses pengujian model terhadap data uji. Data uji yang dipakai dalam proses pengujian model adalah dua puluh persen dari jumlah keseluruhan data pada saat pembagian dataset di 3.2.3. Data yang digunakan untuk diuji terhadap model sudah melalui proses prapemrosesan data NER dan sentimen analisis. Sampel data uji untuk model NER berupa kalimat NER. Dimana setiap token kata dipasangkan dengan labelnya. Pasangan token kata dan label pada kalimat NER dapat dilihat pada tabel 4.1.

Tabel 4.1 Kalimat NER Data Uji

No	Kalimat NER Data Uji
1	[('hidup', 'O'), ('sesedih', 'O'), ('dan', 'O'), ('secaper', 'O'), ('apa', 'O'), ('yak', 'O'), ('nyamperin', 'O'), ('ig', 'B-PROD'), ('atau', 'O'), ('youtube', 'B-PROD'), ('tim', 'O'), ('lain', 'O'), ('cuma', 'O'), ('buat', 'O'), ('ngatain', 'O'), ('.', 'O'), ('perasaan', 'O'), ('dari', 'O'), ('dulu', 'O'), ('gua', 'O'), ('ngefans', 'O'), ('bola', 'O'), ('orang', 'O'), ('pada', 'O'), ('santai', 'O'), ('eh', 'O'), ('di', 'O'), ('ml', 'B-PROD'), ('!', 'O'), ('malah', 'O'), ('gini', 'I-PROD'), ('malah', 'O'), ('gini', 'I-PROD'), ('malah', 'O')]
2	[('kenapa', 'O'), ('kuota', 'B-PROD'), ('kemendikbud', 'I-PROD'), ('tidak', 'O'), ('bisa', 'O'), ('digunakan', 'O'), ('?', 'O'), ('ini', 'O'), ('penyebab', 'O'), ('dan', 'O'), ('solusi', 'O'), ('https://t.co/r3wwse6mrn', 'O')]

Tabel 4.2 Kalimat NER Data Uji(Lanjutan)

No	Kalimat NER Data UJI
3	[('@isaacctangis', 'B-PER'), ('nih', 'O'), ('aku', 'O'), ('kasih', 'O'), ('solusi', 'O'), ('kak', 'O'), ('.', 'O'), ('buka', 'O'), ('app', 'O'), ('@grabid', 'B-PROD'), ('lalu', 'O'), ('#bisalah', 'O'), ('pesannya', 'O'), ('pake', 'O'), ('fitur', 'O'), ('penanan', 'O'), ('terjadwal', 'O'), ('.', 'O'), ('karena', 'O'), ('lagi', 'O'), ('ada', 'O'), ('promo', 'O'), ('grabfood', 'B-PROD'), ('diskon', 'I-PROD'), ('kilat', 'I-PROD'), ('ngabuburit', 'O'), ('yg', 'O'), ('bikin', 'O'), ('untung', 'O'), ('dgn', 'O'), ('potongan', 'O'), ('hingga', 'O'), ('50k', 'O'), ('dari', 'O'), ('13-25', 'O'), ('april', 'O'), ('mulai', 'O'), ('jam', 'O'), ('2-4', 'O'), ('sore.İ½İ½', 'O'), ('note', 'O'), ('.', 'O'), ('pesan', 'O'), ('terjadwal', 'O'), ('min', 'O'), ('2jam', 'O'), ('sebelumnya', 'O'), ('ya', 'O'), ('!', 'O'), ('https://t.co/7w5cvxqjc8', 'O')]
4	[('@rvoseanv', 'B-PER'), ('gapaap', 'O'), ('ce', 'B-PER'), ('.', 'O'), ('lg', 'O'), ('sakit', 'O'), ('aja', 'O')]
5	[('@collegemenfess', 'B-ORG'), ('sering', 'O'), ('banget', 'O'), ('nder', 'O'), ('.', 'O'), ('buat', 'O'), ('yg', 'O'), ('ngekos', 'O'), ('sm', 'O'), ('rumah', 'O'), ('deket', 'O'), ('sih', 'O'), ('gamasalah', 'O'), ('.', 'O'), ('yg', 'O'), ('jauh2', 'O'), ('yg', 'O'), ('kasian', 'O'), ('wkwkwk', 'O'), ('dan', 'O'), ('aku', 'O'), ('termasuk', 'O'), ('yg', 'O'), ('jauh', 'O'), ('.', 'O'), ('biasanya', 'O'), ('aku', 'O'), ('lngsng', 'O'), ('ke', 'O'), ('perpus', 'O'), ('sih', 'O'), ('.', 'O'), ('entah', 'O'), ('perpus', 'O'), ('kampus', 'O'), ('atau', 'O'), ('perpusnas', 'B-LOC'), ('.', 'O'), ('sambil', 'O'), ('ngeringkas', 'O'), ('atau', 'O'), ('nyari2', 'O'), ('referensi', 'O'), ('buat', 'O'), ('tugas', 'O'), ('aja', 'O')]
6	[('kebangkitan', 'O'), ('ekonomi', 'O'), ('jatim', 'B-LOC'), ('ditandai', 'O'), ('kinerja', 'O'), ('ekspor', 'O'), ('yg', 'O'), ('trus', 'O'), ('bergerak', 'O'), ('positif', 'O'), ('sejak', 'O'), ('awal', 'O'), ('taun', 'O'), ('2021', 'O'), ('kata', 'O'), ('gubernur', 'B-PER'), ('jawa', 'I-PER'), ('timur', 'I-PER')]
7	[('@shopeeid', 'B-ORG'), ('İ½İ½', 'O'), ('giveaway', 'O'), ('iphone', 'B-PROD'), ('12', 'I-PROD'), ('mini', 'I-PROD'), ('İ½İ½', 'O'), ('"', 'O'), ('shopee', 'B-PROD'), ('#shopeeramadansale', 'B-EV'), ('#selalubelanjadishopee', 'O'), ('#thrdarishopee', 'O'), ('"', 'O'), ('bismillah', 'O'), ('semoga', 'O'), ('rejeki', 'O'), ('saya', 'O'), ('.', 'O'), ('İ½İ½İ½İ½', 'O'), ('yoh', 'O'), ('iso', 'O'), ('yoh', 'O'), ('!!!!', 'O')]
8	[('propaganda', 'O'), ('hoax', 'O'), ('.', 'O'), ('di', 'O'), ('media', 'O'), ('nasional', 'O'), ('ini', 'O'), ('negara', 'O'), ('apa', 'O'), ('İ½İ½', 'O'), ('sakit', 'O'), ('lu', 'O'), ('pada', 'O')]
9	[('gelar', 'O'), ('khotmil', 'B-EV'), ('quran', 'I-EV'), ('online', 'I-EV'), ('.', 'O'), ('gubernur', 'B-PER'), ('berharap', 'O'), ('keberkahan', 'O'), ('bagi', 'O'), ('masyarakat', 'O'), ('jatim', 'B-LOC'), ('https://t.co/opzxkfsz8r', 'O')]
10	[('@buibaii', 'B-PER'), ('disuruh', 'O'), ('langsung', 'O'), ('bayar', 'O'), ('ukt', 'O')]

Selanjutnya, setiap token kata dan tag NER pada data uji akan diubah menjadi nomor indeks sekuens integer. Konversi token kata ke nomor indeks berurutan tabel 4.2.

Tabel 4.7 Pengujian Model Pada Data Uji Pertama

Token	NER Original	NER Prediksi	Valid
hidup	O	O	1
sesedih	O	O	1
dan	O	O	1
secaper	O	O	1
apa	O	O	1
yak	O	O	1
nyamperin	O	O	1
ig	B-PROD	B-PROD	1
atau	O	O	1
youtube	B-PROD	B-PROD	1
tim	O	O	1
lain	O	O	1
cuma	O	O	1
buat	O	O	1
ngatain	O	O	1
,	O	O	1
perasaan	O	O	1
dari	O	O	1
dulu	O	O	1
gua	O	O	1
ngefans	O	O	1
bola	O	O	1
orang	O	O	1
pada	O	O	1
santai	O	O	1
eh	O	O	1
di	O	O	1
ml	B-PROD	O	0
!	O	O	1
malah	O	O	1
gini	O	O	1

Data pada baris pertama menunjukkan dua token kata terprediksi dengan label B-PROD yaitu, pada token kata “ig” dan “youtube”. Format ‘B’ yang berarti *begin*(awal) terletak sebelum nama entitas. Hal ini mendeskripsikan bahwa kedua token kata tersebut merupakan entitas nama dua produk(PROD) berbeda yang ada pada awal kalimat.

Selanjutnya pengujian model pada data uji baris kedua atau pada kalimat twit kedua dapat dilihat pada tabel 4.6

Tabel 4.8 Pengujian Model Pada Data Uji Kedua

Token	NER Original	NER Prediksi	Valid
kenapa	O	O	1
kuota	B-PROD	O	0
kemendikbud	I-PROD	O	0
tidak	O	O	1
bisa	O	O	1
digunakan	O	O	1
?	O	O	1
ini	O	O	1
penyebab	O	O	1
dan	O	O	1
solusi	O	O	1
https://t.co/r3wwse6mrn	O	O	1

Pada tabel 4.6, dapat dilihat pada kalimat twit kedua, tidak ada label NER yang dapat diprediksi model. Semua token kata diberi label O yang artinya bukan entitas bernama.

Selanjutnya pengujian model pada data uji ketiga, dapat dijelaskan pada tabel 4.7.

Tabel 4.9 Pengujian Model Pada Data Uji Ketiga

Token	NER Original	NER Prediksi	Valid
@isaacctangis	B-PER	O	0
nih	O	O	1
aku	O	O	1
kasih	O	O	1
solusi	O	O	1
kak	O	O	1
,	O	O	1
buka	O	O	1
app	O	O	1
@grabid	B-PROD	B-PROD	1
lalu	O	O	1
#bisalah	O	O	1
pesannya	O	O	1
pake	O	O	1
fitur	O	O	1
penanan	O	O	1
terjadwal	O	O	1
.	O	O	1
karena	O	O	1
lagi	O	O	1
ada	O	O	1
promo	O	O	1
grabfood	B-PROD	B-PROD	1
diskon	I-PROD	O	0

Tabel 4.10 Pengujian Model Pada Data Uji Ketiga(Lanjutan)

Token	NER Original	NER Prediksi	Valid
kilat	I-PROD	O	0
ngabuburit	O	O	1
yg	O	O	1
bikin	O	O	1
untung	O	O	1
dgn	O	O	1
potongan	O	O	1
hingga	O	O	1
50k	O	O	1
dari	O	O	1
13-25	O	O	1
april	O	O	1
mulai	O	O	1
jam	O	O	1
2-4	O	O	1
sore.½i½i½	O	O	1
note	O	O	1
:	O	O	1
pesan	O	O	1
terjadwal	O	O	1
min	O	O	1
2jam	O	O	1
sebelumnya	O	O	1
ya	O	O	1
!	O	O	1
https://t.co/7w5cvxqjc8	O	O	1

Pada tabel 4.7 dapat dilihat terdapat dua token kata yaitu kata terprediksi dengan label B-PROD yaitu, pada token kata “@*grabid*” dan “*grabfood*”. Format ‘B’ yang berarti *begin*(awal) terletak sebelum nama entitas. Hal ini mendeskripsikan bahwa kedua token kata tersebut merupakan entitas nama dua produk(PROD) berbeda, namun masih relevan. Serta kedua entitas tersebut berada pada awal kalimat. Selanjutnya pengujian model terhadap data keempat dapat dilihat pada tabel 4.8.

Tabel 4.11 Pengujian Model Pada Data Uji Keempat

Token	NER Original	NER Prediksi	Valid
@rvoseanv	B-PER	O	0
gapaap	O	O	1
ce	B-PER	O	0
,	O	O	1
lg	O	O	1
sakit	O	O	1

Tabel 4.12 Pengujian Model Pada Data Uji Keempat(Lanjutan)

Token	NER Original	NER Prediksi	Valid
@rvoseanv	B-PER	O	0
@rvoseanv	B-PER	O	0

Pada tabel 4.8, dapat dilihat pada kalimat twit keempat, tidak ada label NER yang dapat diprediksi model. Semua token kata diberi label O yang artinya bukan entitas bernama. Selanjutnya pengujian model pada data uji kelima, dapat dilihat pada tabel 4.9.

Tabel 4.13 Pengujian Model Pada Data Uji Kelima

Token	NER Original	NER Prediksi	Valid
@collegemenfess	B-ORG	O	0
sering	O	O	1
banget	O	O	1
nder	O	O	1
.	O	O	1
buat	O	O	1
yg	O	O	1
ngekos	O	O	1
sm	O	O	1
rumah	O	O	1
deket	O	O	1
sih	O	O	1
gamasalah	O	O	1
,	O	O	1
yg	O	O	1
jauh2	O	O	1
yg	O	O	1
kasian	O	O	1
wkwkwk	O	O	1
dan	O	O	1
aku	O	O	1
termasuk	O	O	1
yg	O	O	1
jauh	O	O	1
.	O	O	1
biasanya	O	O	1
aku	O	O	1
lngsng	O	O	1
ke	O	O	1
perpus	O	O	1
sih	O	O	1
,	O	O	1
entah	O	O	1

Tabel 4.14 Pengujian Model Pada Data Uji Kelima(Lanjutan)

Token	NER Original	NER Prediksi	Valid
atau	O	O	1
nyari2	O	O	1
referensi	O	O	1
buat	O	O	1
tugas	O	O	1
aja	O	O	1
perpus	O	O	1
kampus	O	O	1
atau	O	O	1
perpusnas	B-LOC	O	0
.	O	O	1
sambil	O	O	1
ngeringkas	O	O	1

Pada tabel 4.9, dapat dilihat pada kalimat twit kelima, tidak ada label NER yang dapat diprediksi model. Semua token kata diberi label O yang artinya bukan entitas bernama. Secara keseluruhan prediksi NER pada data twit(data uji) dapat dilihat pada tabel 4.10.

Tabel 4.15 Prediksi NER

No.	NER (Predictions)
1	"[(('hidup', 'O'), ('sesedih', 'O'), ('dan', 'O'), ('secaper', 'O'), ('apa', 'O'), ('yak', 'O'), ('nyamperin', 'O'), ('ig', 'B-PROD'), ('atau', 'O'), ('youtube', 'B-PROD'), ('tim', 'O'), ('lain', 'O'), ('cuma', 'O'), ('buat', 'O'), ('ngatain', 'O'), ('', 'O'), ('perasaan', 'O'), ('dari', 'O'), ('dulu', 'O'), ('gua', 'O'), ('ngefans', 'O'), ('bola', 'O'), ('orang', 'O'), ('pada', 'O'), ('santai', 'O'), ('eh', 'O'), ('di', 'O'), ('ml', 'O'), ('!', 'O'), ('malah', 'O'), ('giniü½zi½', 'O'))]"
2	"[(('kenapa', 'O'), ('kuota', 'O'), ('kemendikbud', 'O'), ('tidak', 'O'), ('bisa', 'O'), ('digunakan', 'O'), ('?', 'O'), ('ini', 'O'), ('penyebab', 'O'), ('dan', 'O'), ('solusi', 'O'), ('https://t.co/r3wwse6mrn', 'O'))]"
3	"[(('@isaacctangis', 'O'), ('nih', 'O'), ('aku', 'O'), ('kasih', 'O'), ('solusi', 'O'), ('kak', 'O'), ('', 'O'), ('buka', 'O'), ('app', 'O'), ('@grabid', 'B-PROD'), ('lalu', 'O'), ('#bisalah', 'O'), ('pesannya', 'O'), ('pake', 'O'), ('fitur', 'O'), ('penanan', 'O'), ('terjadwal', 'O'), ('.', 'O'), ('karena', 'O'), ('lagi', 'O'), ('ada', 'O'), ('promo', 'O'), ('grabfood', 'B-PROD'), ('diskon', 'O'), ('kilat', 'O'), ('ngabuburit', 'O'), ('yg', 'O'), ('bikin', 'O'), ('untung', 'O'), ('dgn', 'O'), ('potongan', 'O'), ('hingga', 'O'), ('50k', 'O'), ('dari', 'O'), ('13-25', 'O'), ('april', 'O'), ('mulai', 'O'), ('jam', 'O'), ('2-4', 'O'), ('sore.ü½zi½', 'O'), ('note', 'O'), (':', 'O'), ('pesan', 'O'), ('terjadwal', 'O'), ('min', 'O'), ('2jam', 'O'), ('sebelumnya', 'O'), ('ya', 'O'), ('!', 'O'), ('https://t.co/7w5cvxqjc8', 'O'))]"
4	"[(('@rvoseanv', 'O'), ('gapaap', 'O'), ('ce', 'O'), ('', 'O'), ('lg', 'O'), ('sakit', 'O'), ('aja', 'O'))]"

Tabel 4.16 Prediksi NER(Lanjutan)

No.	NER (Predictions)
5	"['@collegemenfess', 'O'), ('sering', 'O'), ('banget', 'O'), ('nder', 'O'), ('.', 'O'), ('buat', 'O'), ('yg', 'O'), ('ngekos', 'O'), ('sm', 'O'), ('rumah', 'O'), ('deket', 'O'), ('sih', 'O'), ('gamasalah', 'O'), ('.', 'O'), ('yg', 'O'), ('jauh2', 'O'), ('yg', 'O'), ('kasian', 'O'), ('wkwkwk', 'O'), ('dan', 'O'), ('aku', 'O'), ('termasuk', 'O'), ('yg', 'O'), ('jauh', 'O'), ('.', 'O'), ('biasanya', 'O'), ('aku', 'O'), ('lngsng', 'O'), ('ke', 'O'), ('perpus', 'O'), ('sih', 'O'), ('.', 'O'), ('entah', 'O'), ('perpus', 'O'), ('kampus', 'O'), ('atau', 'O'), ('perpusnas', 'O'), ('.', 'O'), ('sambil', 'O'), ('ngeringkas', 'O'), ('atau', 'O'), ('nyari2', 'O'), ('referensi', 'O'), ('buat', 'O'), ('tugas', 'O'), ('aja', 'O')]"
6	"['kebangkitan', 'B-ORG'), ('ekonomi', 'O'), ('jatim', 'I-ORG'), ('ditandai', 'O'), ('kinerja', 'O'), ('ekspor', 'O'), ('yg', 'O'), ('trus', 'O'), ('bergerak', 'O'), ('positif', 'O'), ('sejak', 'O'), ('awal', 'O'), ('taun', 'O'), ('2021', 'O'), ('kata', 'O'), ('gubernur', 'O'), ('jawa', 'O'), ('timur', 'O')]"
7	"['@shopeeid', 'O'), ('i½i½', 'O'), ('giveaway', 'O'), ('iphone', 'B-PROD'), ('12', 'O'), ('mini', 'O'), ('i½i½', 'O'), ('"', 'B-PROD'), ('shopee', 'O'), ('#shopeeramadansale', 'O'), ('#selalubelanjadishopee', 'O'), ('#thrdarishopee', 'O'), ('"', 'O'), ('bismillah', 'O'), ('semoga', 'O'), ('rejeki', 'O'), ('saya', 'O'), ('.', 'O'), ('i½i½i½i½i½', 'O'), ('yoh', 'O'), ('iso', 'O'), ('yoh', 'O'), ('!!!!', 'O')]"
8	"['propaganda', 'O'), ('hoax', 'O'), ('.', 'O'), ('di', 'O'), ('media', 'O'), ('nasional', 'O'), ('ini', 'O'), ('negara', 'O'), ('apa', 'O'), ('i½i½', 'O'), ('sakit', 'O'), ('lu', 'O'), ('pada', 'O')]"
9	"['gelar', 'O'), ('khotmil', 'O'), ('quran', 'O'), ('online', 'O'), ('.', 'O'), ('gubernur', 'O'), ('berharap', 'O'), ('keberkahan', 'O'), ('bagi', 'O'), ('masyarakat', 'O'), ('jatim', 'O'), ('https://t.co/opzxkfsz8r', 'O')]"
10	"['@buibaii', 'O'), ('disuruh', 'O'), ('langsung', 'O'), ('bayar', 'O'), ('ukt', 'O')]"

Selanjutnya, data uji yang sudah di-stem dan dilakukan proses sentimen analisis dengan kamus leksikon, menghasilkan keluaran berupa data twit dengan skor sentimen dan label sentimen. Hasil sentimen analisis menggunakan kamus leksikon bahasa Indonesia terhadap data dapat ditunjukkan pada tabel 4.11

Tabel 4. 17 Hasil Sentimen Analisis Kamus Leksikon Bahasa Indonesia

No.	Data Uji	Sentiment	Label
1	Hidup sedih dan secaper apa yak nyamperin ig atau youtube tim lain cuma buat ngatain , perasaan dari dulu gua ngefans bola orang pada santai eh di ml ! malah gini i½i½	-25	Negative
2	Kenapa Kuota Kemendikbud Tidak Bisa Digunakan ? Ini Penyebab dan Solusi https://t.co/R3wWSe6MRn	5	Positive
3	@isaacctangis nih aku kasih solusi kak , buka App @GrabID lalu #Bisalah Pesannya pake fitur Penanan Terjadwal . Karena lagi ada promo GrabFood DISKON KILAT ngabuburit yg bikin untung dgn	26	Positive

Tabel 4. 18 Hasil Sentimen Analisis Kamus Leksikon Bahasa Indonesia(Lanjutan)

No.	Data Uji	Sentiment	Label
	potongan hingga 50K dari 13-25 April mulai jam 2-4 sore.🙄🙄🙄 Note : pesan terjadwal min 2jam sebelumnya ya ! https://t.co/7w5CVXQJc8		
4	@rvoseanv gapaap ce , lg sakit aja	-4	Negative
5	@collegemenfess Sering banget nder . Buat yg ngekos sm rumah dekat sih gamasalah , yg jauh2 yg kasian wkwwk dan aku termasuk yg jauh . Biasanya aku lngsng ke perpustakaan , entah perpustakaan atau perpustakaan . Sambil ngeringkas atau nyari2 referensi buat tugas aja	6	Positive
6	kebangkitan ekonomi Jatim ditandai kinerja ekspor yg trus bergerak positif sejak awal taun 2021, kata Gubernur Jawa Timur	2	Positive
7	@ShopeeID 🙄🙄🙄 GIVEAWAY IPHONE 12 MINI 🙄🙄🙄 ' Shopee #ShopeeRamadanSale #SelaluBelanjadiShopee #THRDariShopee ' Bismillah semoga rejeki saya . 🙄🙄🙄🙄🙄🙄🙄 YOH ISO YOH !!!!	6	Positive
8	propaganda hoax , di media nasional Ini negara apa 🙄🙄🙄 Sakit lu pada	-16	Negative
9	Gelar Khotmil Quran Online , Gubernur Berharap Keberkahan Bagi Masyarakat Jatim https://t.co/OPZxKFSZ8r	14	Positive
10	@buiibaii Disuruh langsung bayar ukt	2	Positive

Dari informasi yang dijelaskan oleh tabel 4.11, dapat diketahui bahwa, kamus *InSet Lexicon* berbahasa Indonesia mampu memproduksi label sentimen secara otomatis. Metode ini menghitung skor polaritas pada setiap token kata yang sudah di-stem, sehingga label sentimen bisa diputuskan untuk setiap kalimat tweet.

4.4. Evaluasi

Pada bagian ini yang dilakukan adalah membuat evaluasi terhadap performa model dan sentimen analisis otomatis. Dalam melakukan evaluasi pada model NER dan sentimen analisis otomatis menggunakan kamus leksikon bahasa Indonesia, penulis menggunakan metrik pengukuran sebagai berikut.

4.4.1 Metrik Precision, Recall dan F1-Score

Pengujian model yang dilakukan terhadap data uji dapat menghitung bagaimana performa model pada anotasi entitas bernama terhadap data uji. Hasil pengukuran

performa model dapat dilihat pada persentase nilai precision, recall dan F1-score ditabel 4.12.

Tabel 4.19 Precision, Recall dan F1-Score(Data Uji)

No.	Tag	TP	TN	FP	FN	Precision	Recall	F1-Score
1	O	28471	2535	1992	449	0.934609	0.984474	0.958894
2	B-PROD	424	32580	212	231	0.666667	0.647328	0.656855
3	I-PROD	0	33222	0	225	0.000000	0.000000	0.000000
4	B-PER	718	32199	124	406	0.852732	0.638790	0.730417
5	I-PER	432	32711	94	210	0.821293	0.672897	0.739726
6	B-ORG	149	32921	110	267	0.575290	0.358173	0.441481
7	I-ORG	66	33129	54	198	0.550000	0.250000	0.343750
8	B-EV	16	33259	8	164	0.666667	0.088889	0.156863
9	I-EV	55	33145	53	194	0.509259	0.220884	0.308123
10	B-WA	0	33425	0	22	0.000000	0.000000	0.000000
11	I-WA	0	33401	0	46	0.000000	0.000000	0.000000
12	B-LOC	268	32860	102	217	0.724324	0.552577	0.626901
13	I-LOC	60	33189	39	159	0.606061	0.273973	0.377358

Dari tabel 4.12 dapat dilihat bahwa model NER memiliki performa yang baik dalam menganotasi entitas dalam data twit. Hampir semua tag NER bisa diprediksi benar oleh model, terbukti dengan nilai precision, recall dan f1-score pada beberapa tag NER. Terkecuali pada tag I-PROD, B-WA dan I-WA. Skor setiap elemen metrik menghasilkan nilai yang kurang baik. Hal ini karena adanya ketimpangan pada jumlah data dengan label tersebut pada dataset.

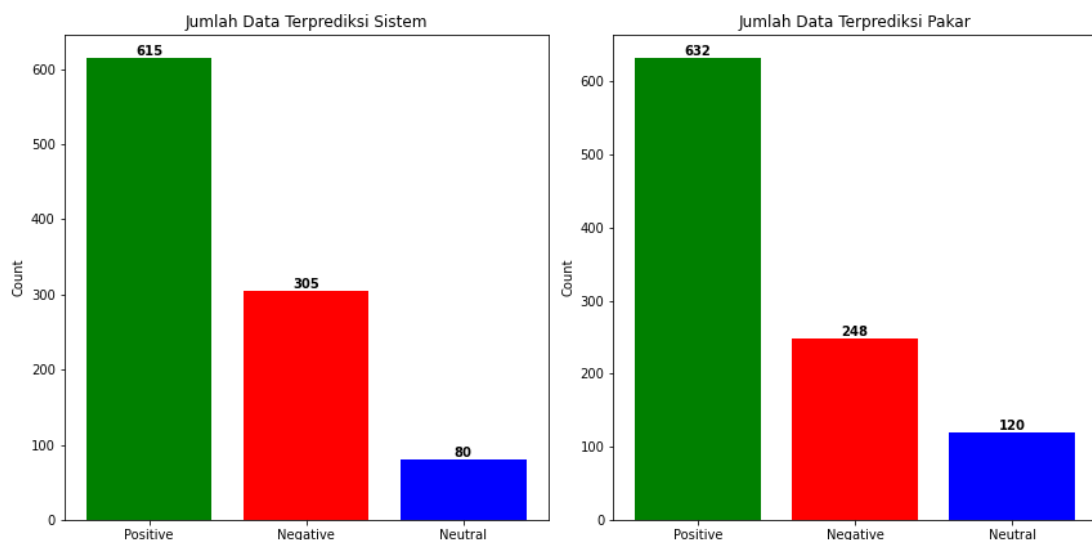
4.4.2 Evaluasi Manual Dengan Akurasi

Dalam melakukan evaluasi terhadap kinerja sentimen analisis secara otomatis menggunakan kamus leksikon bahasa Indonesia terhadap data twit, dilakukan beberapa proses. Penulis mengirim sampel data yang sudah diklasifikasi sentimen oleh sistem kepada anotator. Penulis mengirimkan 1000 baris data twit sebagai sampel data yang akan diberi label secara manual oleh pakar. Pakar dalam hal ini adalah lima orang guru bahasa Indonesia SMA. Para pakar melakukan pemberian label pada data twit sesuai dengan pengetahuan di bidang ilmu bahasa Indonesia, tanpa adanya campur tangan penulis. Perbandingan data yang diberi label sentimen secara otomatis oleh sistem dengan pelabelan oleh anotator secara manual dapat dilihat pada tabel 4.13.

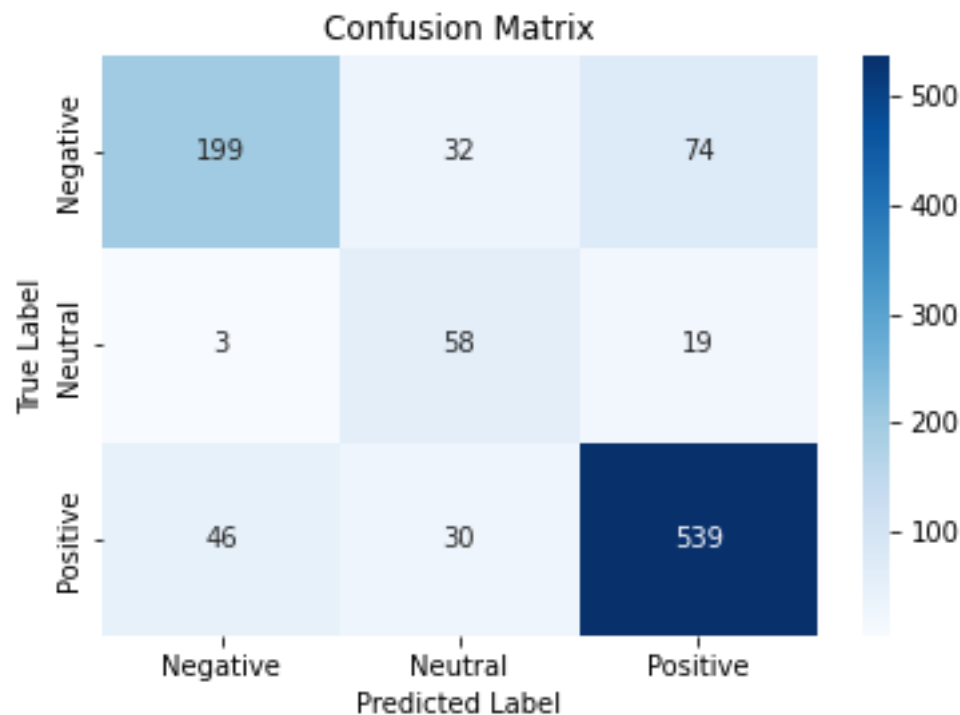
Tabel 4. 20 Perbandingan Sentimen Analisis Otomatis dan Manual

No.	Text	Sistem	Pakar
0	Hidup sesedih dan secaper apa yak nyamperin ig...	-1	-1
1	Kenapa Kuota Kemendikbud Tidak Bisa Digunakan ...	1	1
2	@isaacctangis nih aku kasih solusi kak , buka ...	1	1
3	@rvoseanv gapaap ce , lg sakit aja	-1	1
...
995	Daihatsu Rocky Dijual Resmi Mulai Rp 214 Jutaa...	1	0
996	@Rslnzhr wiranto aja nyamar jadi tukang becak ...	-1	-1
997	@FeatPictures @ParkChanyeolINA YA ALLAH AKU BE...	1	1
998	@susruby @auproms mau shopeepay aja plsss	1	1
999	Penghalauan pada pemudik adalh u / menghindari...	-1	-1

Hasil perbandingan sentimen analisis oleh sistem dan pakar dapat dilihat pada gambar 4.16.

**Gambar 4. 16** Perbandingan Sentimen Analisis oleh Sistem dan Pakar

Lalu selanjutnya adalah menghitung jumlah keseluruhan data yang diprediksi benar dan total data yang diprediksi tidak benar menggunakan *confusion matrix*. Tampilan *confusion matrix* dari perbandingan sentimen analisis otomatis dengan sistem dan pelabelan manual oleh anotator dapat dilihat pada gambar 4.16.



Gambar 4. 17 Confusion Matrix Sentimen Analisis

Pada informasi dari tampilan gambar 4.17 dapat dilihat bahwa, terdapat 199 data dengan label negatif yang diprediksi benar. Terdapat 32 data yang sebenarnya memiliki label negatif namun diprediksi sebagai netral. Serta itu terdapat 74 data yang sebenarnya memiliki label negatif tetapi diprediksi positif. Selain itu terdapat 58 data yang memiliki label netral dan diprediksi benar dikelas netral. Ada 3 data yang harusnya memiliki label netral namun diprediksi negatif. Serta ada 19 data yang harusnya diberi label netral namun diprediksi positif. Pada baris terakhir, terdapat 539 data dengan label positif dan diprediksi dengan benar. Selain itu ada 46 data yang harusnya diberi label positif namun diprediksi sebagai negatif. Terakhir, ada 30 data yang harusnya diprediksi positif namun diberi label netral. Berdasarkan gambar 4.17 dapat diperoleh nilai akurasi dari sentimen analisis menggunakan kamus leksikon bahasa Indonesia, yaitu sebagai berikut.

$$Akurasi = \frac{199+58+539}{1000} = 79,6\%$$

Atas perolehan nilai akurasi yang terhadap 1000 data uji yang divalidasi oleh kelima anotator, dapat disimpulkan bahwa analisis sentimen otomatis dengan menggunakan kamus leksikon bahasa Indonesia dapat dijadikan suatu pilihan dalam melakukan

pelabelan sentimen secara otomatis . Kelemahan dari sentimen analisis menggunakan kamus leksikon ini sehingga akurasi yang dicapai belum optimal karena seiring bertambahnya waktu, terdapat kosa kata baru dan kata tersebut belum ditambahkan pada kamus InSet leksikon yang dipakai pada penelitian ini.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian yang sudah dilakukan oleh penulis dapat ditarik kesimpulan bahwa:

1. Nilai akurasi yang didapatkan dari pelatihan model NER dengan menggunakan BiLSTM anotasi otomatis pada data twit yaitu 98% dan dengan nilai *loss* 2%. Lalu, saat pengujian model NER dengan data uji, jumlah keseluruhan skor precision adalah 98,74%, recall sebesar 98,74% dan f1-score sebesar 98,74%. Hal ini menunjukkan bahwa model NER BiLSTM mampu melakukan anotasi entitas bernama(NER) secara otomatis pada data twit.
2. Perbandingan dari perolehan total nilai prediksi sentimen dan sentimen sebenarnya dalam sentimen analisis menggunakan kamus leksikon bahasa Indonesia, yang tidak begitu besar menjadikan sentimen analisis secara otomatis menggunakan kamus leksikon sebagai suatu pilihan yang bisa dilakukan untuk melakukan pelabelan data. Dengan nilai akurasi 79,64%, data twit dapat diprediksi memiliki sentimen positif, negatif dan netral.
3. Integrasi anotasi otomatis menggunakan BiLSTM dan sentimen analisis berbasis leksikon pada data twit dapat dilakukan dalam membangun sistem untuk mengekstrak entitas bernama serta sentimen pada data twit.
4. Ketimpangan data pada dataset NER terutama pada persebaran setiap tag memiliki pengaruh signifikan pada pembuatan model NER. Hal ini dapat menyebabkan bias pada model NER, sehingga ada tag yang tidak dapat diprediksi dengan benar oleh model.
5. Beberapa kata tidak ditemukan dalam kamus leksikon bahasa Indonesia. Hal ini menyebabkan adanya kata dengan leksikal negatif ataupun positif tidak bisa diberi bobot sentimen. Sehingga ada data yang memiliki label sentimen yang tidak tepat.

6. Proses casefolding pada pemrosesan NER mengurangi performa model NER dalam mengannotasi entitas pada data. Hal ini karena data twit adalah data teks yang memiliki tingkat tidak terstruktur yang tinggi, dimana ada kata yang memang harus ditulis dengan menggunakan huruf kapital dan memiliki suatu label NER, namun karena dilakukan proses casefolding, kata tersebut memiliki label yang berbeda. Sehingga terjadi ambiguitas.

5.2 Saran

Beberapa masukan yang dapat dilakukan untuk mengembangkan penelitian selanjutnya dengan tema dan permasalahan yang relevan, yaitu:

1. Pada penelitian selanjutnya, diharapkan untuk melakukan penambahan data pada dataset NER, terutama pada tag WA, PROD dan LOC pada proses training. Sehingga model bisa mempelajari lebih banyak tag dan dapat memprediksi entitas pada teks dengan baik.
2. Pada penelitian selanjutnya diharapkan untuk menambahkan penerapan konsep *POS tagging*, *chunking* kalimat ataupun teknik-teknik lain yang mampu memproses data NER.
3. Penelitian selanjutnya pada sentimen analisis berbasis leksikon, diharapkan untuk menambahkan penerapan algoritma untuk menganalisis sentimen dengan lebih baik. Sehingga dapat diperoleh perbandingan pelabelan data menggunakan kamus leksikon dengan pelabelan data secara manual.
4. Pada penelitian selanjutnya, kosa kata pada kamus leksikon bahasa Indonesia diharapkan dapat ditambah untuk dapat menganalisis sentimen pada data twit dengan lebih baik lagi.
5. Penelitian selanjutnya diharapkan menambah teknik-teknik lain seperti pemrosesan sarkasme, pengolahan emoji dalam melakukan sentimen analisis berbasis leksikon. Hal ini untuk meminimalisir kesalahan pada pelabelan kategori sentimen.

DAFTAR PUSTAKA

- Amidi , A., & Amidi, S. (2018). VIP Cheatsheet: Recurrent Neural Networks.
- Annur, C. M. (2022, Maret 23). *katadata*. Retrieved from Pengguna Twitter Indonesia Masuk Daftar Terbanyak di Dunia, Urutan Berapa?: <https://databoks.katadata.co.id/datapublish/2022/03/23/pengguna-twitter-indonesia-masuk-daftar-terbanyak-di-dunia-urutan-berapa>
- Azarine, I. S., Bijaksana, M. A., & Asror, I. (2019). Named Entity Recognition on Indonesian Tweets using Hidden Markov Model. *2019 7th International Conference on Information and Communication Technology (ICoICT)*, 1-5.
- Gunawan, Y., Young, J. C., & Rusli, A. (2021). FastText Word Embedding and Random Forest. *Ultimatics : Jurnal Teknik Informatika*, 13.
- Hernikawati, D. (2021). Kecenderungan Tanggapan Masyarakat Terhadap Vaksin Sinovac Berdasarkan Lexicon Based Sentiment Analysis. *Jurnal IPTEK-KOM(Jurnal Ilmu Pengetahuan dan Teknologi Komunikasi)*, 23, 21-31.
- Koto, F., & Rahmaningtyas, G. Y. (2017). InSet Lexicon: Evaluation of a Word List for. *2017 International Conference on Asian Language Processing (IALP)*, 391-394.
- Kurniawati, N. R., & Winarko, E. (2016). Penentuan Destinasi Wisata Favorit Berbasis Aturan Dan Analisis Sentimen Pada Tweet Berbahasa Indonesia (Doctoral dissertation, Universitas Gadjah Mada).
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural Architectures for Named Entity Recognition. *arXiv preprint arXiv:1603.01360*.
- Munarnawan, & Sinaga, A. (2017). Pemanfaatan Analisis Sentimen Untuk Peningkatan Popularitas Tujuan Wisata. *Jurnal Penelitian Pos Dan Informatika*, 7(2), 109-120.

- Novi et.al. (2021). Named-Entity Recognition Pada Teks Berbahasa Indonesia Menggunakan Metode Hidden Markov Model Dan POS-Tagging. *Jurnal Linguistik Komputasional*, 4, 13-20.
- N. Chinchor and P. Robinson. 1998. Appendix E: MUC-7 Named Entity Task Definition (version 3.5). In Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998.
- Nurfalah, A., Adiwijaya, & Suryani, A. A. (2017, Maret). Analisis Sentimen Berbahasa Indonesia dengan pendekatan Lexicon-Based pada Media Sosial. *II*(1).
- Permana, H., & Purnamasari, K. K. (2019). Named Entity Recognition Menggunakan Metode Bidirectional Lstm-Crf Pada Teks Bahasa Indonesia (Doctoral dissertation, Universitas Komputer Indonesia).
- Prasetya, Y. N., Winarso, D., & Syahril. (2021). Penerapan Lexicon Based Untuk Analisis Sentimen Pada Twitter. *JURNAL FASILKOM*, 97-103.
- Prasetyo, S. A., & Kusumaningrum, R. (2018). Klasifikasi Dokumen Berita Bahasa Indonesia Menggunakan Metode Latent Dirichlet Allocation (Lda) Dan Word2Vec.
- Putra, M. F., & Hidayatullah, A. F. (2021). Tinjauan Literatur : Named Entity Recognition pada Ulasan Wisata. *Jurnal UII/Automata*.
- Rachman, V., S. S., Augustianti, F., & Mahendra, R. (2017). Named entity recognition on Indonesian Twitter posts Using Long Short-Term Memory Networks. *2017 International Conference on Advanced Computer Science and Information Systems (ICACISIS)*, 228-232.
- Rifani, R., Bijaksana, M. A., & Asror, I. (2019). Named Entity Recognition for an Indonesian Based Language Tweet using Multinomial Naive Bayes Classifier. *Indonesia Journal on Computing (Indo-JC)*, IV(2), 119-126.
- Rusliani. (2017). Named Entity Recognition Pada Teks Berbahasa Indonesia Untuk Pembangkit Pertanyaan Otomatis. *UNIKOM*.

- Sari, Y., Hassan, M. F., & Zamin, N. (2010, June). Rule-based Pattern Extractor and Named Entity Recognition: A Hybrid Approach. in *2010 International Symposium on Information Technology*, 2, 563-568.
- Taufik, N., Wicaksono, A. F., & Adriani, M. (2016, November). Named entity recognition on Indonesian microblog messages. In *2016 International Conference on Asian Language Processing (IALP)* (pp. 358-361). IEEE.
- Tilbe, A. (2022, July 9). *Top 5 Approaches to Named Entity Recognition (NER) in 2022*. Retrieved from Towards AI: <https://pub.towardsai.net/top-5-approaches-to-named-entity-recognition-ner-in-2022-38afdf022bf1>
- Ushio, A., Neves, L., Silva, V., & Barbieri, F. (2020). Named Entity Recognition in Twitter: A Dataset and Analysis on Short-Term Temporal Shifts. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (pp. 3186-3196).
- Weischedel, R., Pradhan, S., Ramshaw, L., Kaufman, J., Franchini, M., El-Bachouti, M., . . . Houston, A. (2017). OntoNotes Release 5.0 with OntoNotes DB Tool v0.999 beta.
- Wibisono, Y., & Khodra, M. L. (2018). Pengenalan Entitas Bernama Otomatis untuk Bahasa Indonesia dengan Pendekatan Pembelajaran Mesin. *The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018)*, (pp. 1-5).
- Wikarsa, L., Angdresey, A., & Kapantow, J. (2022, April). Implementasi Metode Naïve Bayes dan Lexicon-Based Approach untuk Mengklasifikasi Sentimen Netizen pada Tweet Berbahasa Indonesia. *Jurnal Ilmiah Realtech*, 18(1).
- Xin et.al, Z. (2019). Automatic Annotation of Text Classification Data Set in Specific Field Using Named Entity Recognition. *2019 IEEE 19th International Conference on Communication Technology*, 1403-1407.