

**ANALISIS SENTIMEN TERHADAP OPINI MASYARAKAT MENGENAI
PROGRAM KERJA KOTA MEDAN MENGGUNAKAN LSTM (*LONG
SHORT TERM MEMORY*) DENGAN MEDIA SOSIAL *TWITTER***

SKRIPSI

ROSHAN RAM METTA

191401107



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

MEDAN

2023

**ANALISIS SENTIMEN TERHADAP OPINI MASYARAKAT MENGENAI
PROGRAM KERJA KOTA MEDAN MENGGUNAKAN LSTM (*LONG
SHORT TERM MEMORY*) DENGAN MEDIA SOSIAL *TWITTER***

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Komputer

ROSHAN RAM METTA

191401107



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

2023

PERSETUJUAN

3

PERSETUJUAN

Judul : ANALISIS SENTIMEN TERHADAP OPINI
MASYARAKAT MENGENAI PROGRAM KERJA KOTA
MEDAN MENGGUNAKAN LSTM (LONG SHORT TERM
MEMORY) DENGAN MEDIA SOSIAL TWITTER

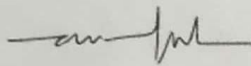
Kategori : SKRIPSI

Nama : ROSHAN RAM METTA

Nomor Induk Mahasiswa : 191401107
: SARJANA (S-1) ILMU KOMPUTER
: ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

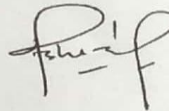
Komisi Pembimbing :

Pembimbing II



Dr. Amalia, S.T., M.T.
NIP. 197812212014042001

Pembimbing I



Amer Sharif S.Si., M.Kom.
NIP. 1271212110690004

Diketahui/Disetujui Oleh
Program Studi Ilmu

KomputerKetua



Dr. Amalia, S.T., M.T.
NIP.

197812212014042001

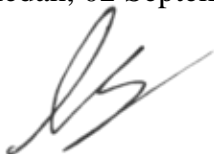
PERNYATAAN

ANALISIS SENTIMEN TERHADAP OPINI MASYARAKAT MENGENAI
PROGRAM KERJA KOTA MEDAN MENGGUNAKAN LSTM (*LONG
SHORT TERM MEMORY*) DENGAN MEDIA SOSIAL *TWITTER*

SKRIPSI

Saya menyatakan bahwasannya skripsi ini merupakan hasil dari kerja keras saya sendiri, tidak termasuk dengan kutipan yang dicantumkan sumbernya.

Medan, 02 September 2023



Roshan Ram Metta

191401107

UCAPAN TERIMA KASIH

Segala puji dan syukur kepada tuhan yang maha esa atas karunianya sehingga penulis diberi kemudahan dan dapat menyelesaikan skripsi ini tepat pada waktunya. Penulis juga mengucapkan terimakasih sebesar-besarnya kepada seluruh pihak yang memberikan banyak bantuan serta dukungan yang sangat berarti dalam proses penyelesaian skripsi ini. Terimakasih penulis ucapkan kepada:

1. Prof. Dr. Muryanto Amin S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
2. Dr. Maya Silvi Lydia B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Dr. Amalia S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara dan selaku dosen pembimbing II yang telah memberikan bantuan serta koreksi pada proses penulisan skripsi ini.
4. Bapak Amer Sharif S.Si., M.Kom. selaku dosen pembimbing I yang selalu membantu mengarahkan, memberikan masukan, dan memberikan dukungan terhadap penulis sampai dengan akhir penulisan skripsi ini.
5. Kedua orangtua penulis, Raj Suhendra dan Malini Wijaya, yang selalu memberikan dukungan, menyemangati dan selalu mendoakan penulis sehingga terselesaikannya skripsi ini.
6. Teman-teman yang selalu memberikan bantuan dan masukan yang positif bagi penulis dalam proses penulisan skripsi ini.

Penulis mengharapkan semoga hasil kerja keras yang dilakukan selama 6 bulan ini memberikan hasil dan dampak positif bagi lingkungan sekitar baik dari sisi ilmu pengetahuan, masyarakat dan sivitas akademik. Di sisi lain penulis juga menyadari pada penelitian ini juga terdapat beberapa hal yang dapat ditingkatkan serta dimaksimalkan, maka dari itu penulis mengharapkan agar hasil dari penelitian ini nantinya dapat dikembangkan dan menjadi dasar pada penelitian-penelitian selanjutnya yang dapat menginspirasi serta berguna bagi banyak orang.

Akhir kata, terima kasih sekali lagi kepada semua yang telah berperan dalam kesuksesan penelitian ini. Semoga penelitian ini dapat memberikan manfaat yang nyata serta berkelanjutan dalam perkembangan ilmu pengetahuan dan praktik di bidang yang diteliti. Dan

juga terimakasih pada Roshan Ram Metta selaku penulis yang selalu berusaha pantang menyerah dalam penulisan skripsi ini.

Medan, 02 September 2023

A handwritten signature in black ink, consisting of stylized, cursive letters that appear to be 'RS'.

Penulis

ABSTRAK

Skripsi ini bertujuan untuk melakukan penelitian terkait sentimen analisis terhadap opini publik mengenai program kerja Kota Medan selama masa pemerintahan Walikota dan Wakil Walikota periode 2020 hingga saat ini. Penelitian ini menggunakan algoritma LSTM (*Long Short Term Memory*) dan metode pembobotan menggunakan BERT (*Bidirectional Encoder Representations From Transformers*). Dataset yang digunakan dalam penelitian ini diperoleh dari Twitter dengan menggunakan tagar seperti #MedanBerkah, #MedanBersih, #MedanKolaborasi, dan kata kunci seperti Kota Medan, Medan Maju, serta Medan sejahtera. Dataset ini melalui proses scraping dan kemudian tahap pre-processing, termasuk *case folding*, *tokenizing*, *stopword removal*, *punctuation removal*, dan *lemmatization*. Data yang telah dibersihkan disebut sebagai data bersih. Selanjutnya, data yang telah dibersihkan dilabeli dengan sentimen, yang dilakukan secara otomatis menggunakan metode *Lexicon Based* dengan pendekatan *word piece*. Total data yang dilabeli terdiri dari 3642 cuitan, dengan 1723 memiliki sentimen positif, 1665 sentimen negatif, dan 256 sentimen netral. Data yang telah dilabeli siap untuk proses pelatihan model. Sebelum pelatihan, peneliti melakukan pencarian *hyperparameter* menggunakan *library* optuna untuk mendapatkan parameter terbaik. Optuna mencoba kombinasi parameter yang telah dipersiapkan dan memilih parameter terbaik berdasarkan hasil perhitungan. Setelah mendapatkan *hyperparameter* terbaik, peneliti melakukan pelatihan model dengan algoritma LSTM dan pembobotan menggunakan algoritma BERT. Hasil pelatihan menunjukkan akurasi terendah sebesar 75% dan akurasi tertinggi sebesar 76%. Hasil ini menunjukkan bahwa model dapat mengklasifikasikan opini masyarakat tentang program kerja Kota Medan dengan baik, meskipun masih ada ruang untuk pengembangan lebih lanjut, terutama dalam proses pelabelan data. Kendala dalam penelitian ini termasuk kualitas pelabelan data menggunakan metode *Lexicon Based*, ketidakmertian model terhadap beberapa kata slang dalam dataset, dan kebutuhan akan dataset yang lebih besar untuk model yang lebih kompleks. Diperlukan juga fitur tambahan dalam proses pre-processing, seperti normalisasi data, untuk meningkatkan kualitas analisis sentimen.

ABSTRACT

This thesis aims to conduct research related to sentiment analysis of public opinion regarding the work program of the City of Medan during the administration of the Mayor and Deputy Mayor for the period 2020 to the present. This research uses the LSTM (Long Short Term Memory) algorithm and a weighting method using BERT (Bidirectional Encoder Representations From Transformers). The dataset used in this research was obtained from Twitter using hashtags such as #MedanBerkah, #MedanBersih, #MedanKolaborasi, and keywords such as Medan City, Medan Maju, and Medan Sejahtera. This dataset goes through a scraping process and then a pre-processing stage, including case folding, tokenizing, stopwords removal, punctuation removal, and lemmatization. Data that has been cleaned is called clean data. Next, the data that has been cleaned will be labeled with sentiment, which is done automatically using the Lexicon Based method with a word piece approach. The total labeled data consists of 3642 tweets, with 1723 having positive sentiment, 1665 negative sentiment, and 256 neutral sentiment. The labeled data is ready for the model training process. Before training, researchers carried out a hyperparameter search using the Optuna library to get the best parameters. Optuna tries a combination of parameters that have been prepared and chooses the best parameters based on the calculation results. After getting the best hyperparameters, researchers trained the model using the LSTM algorithm and weighting using the BERT algorithm. The training results show the lowest accuracy of 75% and the highest accuracy of 76%. These results show that the model can classify public opinion about the Medan City work program well, although there is still room for further development, especially in the data labeling process. Constraints in this research include the quality of data labeling using the Lexicon Based method, the model not understanding some slang words in the dataset, and the need for a larger dataset for a more complex model. Additional features are also needed in the pre-processing process, such as data normalization, to improve the quality of sentiment analysis.

DAFTAR ISI

PERSETUJUAN.....	3
PERNYATAAN	iv
UCAPAN TERIMA KASIH.....	v
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
BAB I.....	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian.....	3
1.4. Manfaat Penelitian.....	3
1.5. Batasan Masalah.....	3
BAB II.....	5
LANDASAN TEORI.....	5
2.1. <i>Twitter</i>	5
2.2. <i>Natural Language Processing (NLP)</i>	5
2.3. Analisis Sentimen.....	6
2.4. <i>Machine Learning</i>	6
2.5. <i>Deep Learning</i>	7
2.6. <i>Neural Network</i>	7
2.7. <i>Lexicon-Based</i>	8
2.8. <i>Text Preprocessing</i>	9
2.8.1. Case Folding	10
2.8.2. Tokenizing	10
2.8.3. Stopwords Removing.....	10
2.8.4. Punctuation Removing.....	10
2.8.5. Lemmatization	10
2.9. <i>BERT Embedding</i>	11
2.10. Algoritma LSTM.....	12
2.11. <i>Hyperparameter</i> Dengan Optuna.....	13

2.12. Evaluasi	14
2.12.1 Confusion Matrix	14
2.12.2 Classification Report	14
2.13. Penelitian Terdahulu	15
BAB III	19
ANALISIS DAN PERANCANGAN	19
3.1. General Arsitektur Sistem	19
3.2. Data <i>Crawling</i>	20
3.3. <i>Labelling</i>	21
3.4. <i>Text Preprocessing</i>	22
3.4.1. Case Folding	23
3.4.2. Tokenizing	24
3.4.3. Stopwords Removing	24
3.4.4. Punctuation Removing	25
3.4.5. Lemmatization	26
3.5. <i>BERT Embedding</i>	27
3.5.1 Preprocessing	27
3.5.2 Pembuatan Input Sequence	27
3.5.3 Embedding	29
3.6. Algoritma LSTM	32
3.6.1 Input Gate	33
3.6.2 Forget Gate	35
3.6.3 Output Gate	36
3.7. Optuna	38
3.8. Evaluasi	39
3.8.1. Confusion Matrix	39
3.8.2 Classification Report	39
3.8.3 Single Sentence Predict	40
BAB IV	41
IMPLEMENTASI DAN PENGUJIAN	41
4.1. Implementasi Sistem	41
4.1.1. Spesifikasi Perangkat Keras	41
4.1.2. Spesifikasi Perangkat Lunak	41
4.2. Implementasi <i>Web Scraping</i>	41
4.3. Labelisasi Dataset	44
4.4. <i>Pre-Processing Dataset</i>	45

4.4.1.	Case Folding	45
4.4.2.	Tokenizing	45
4.4.3.	Stopwords Removing.....	46
4.4.4.	Punctuation Removing.....	46
4.4.5.	Lemmatization	47
4.5.	<i>Split Dataset</i>	47
4.6.	<i>Best Hyperparameter</i> Dengan Optuna	47
4.7.	Implementasi LSTM Dengan <i>BERT Embedding</i>	48
4.8.	<i>Prediction Test</i>	53
BAB V	56
KESIMPULAN DAN SARAN	56
5.1.	Kesimpulan.....	56
5.2.	Saran.....	56
DAFTAR PUSTAKA	58

DAFTAR GAMBAR

Gambar 2. 1. <i>Flowchart</i> Labelisasi Dataset.....	9
Gambar 2. 2. <i>Flowchart</i> Text Preprocessing	11
Gambar 2. 3. Tabel <i>Confusion Matrix</i>	14
Gambar 2. 4. Rumus <i>Classification report</i>	15
Gambar 3. 1. General Arsitektur Sistem	19
Gambar 3. 2. Data <i>Crawling</i>	21
Gambar 3. 3. BERT Tokenize.....	27
Gambar 3. 4. BERT Token Input.....	28
Gambar 3. 5. BERT Padding Input.....	28
Gambar 3. 6. Penjumlahan representasi vektor kata.....	30
Gambar 3. 7. Flowchart BERT Embedding	31
Gambar 3. 8. Input Gate Process	32
Gambar 3. 9. Forget Gate Process	34
Gambar 3. 10. Output Gate Process.....	35
Gambar 3. 11. Flowchart Algoritma LSTM	37
Gambar 3. 12. Arsitektur Algoritma LSTM	37
Gambar 3. 13. Tabel <i>Confusion Matrix</i>	39
Gambar 3. 14. Rumus <i>Classification report</i>	40
Gambar 3. 15. Rancangan <i>UI sentence prediction</i>	55
Gambar 4. 1. Laman Twitter Keyword #MedanBersih	42
Gambar 4. 2. Laman Twitter Keyword Medan Kolaborasi	42
Gambar 4. 3. Laman Twitter Keyword Kota Medan	43
Gambar 4. 4. Dataset Hasil Scrapping Dengan Lokasi.....	43
Gambar 4. 5. Labelisasi Dataset	44
Gambar 4. 6. Case Folding	45
Gambar 4. 7. Tokenizing	45
Gambar 4. 8. Stopwords Removing.....	46
Gambar 4. 9. Punctuation Removing.....	46
Gambar 4. 10. Lemmatization	47
Gambar 4. 11. Splitting Dataset.....	47
Gambar 4. 12. <i>Proses Loop</i> Akurasi Training dan Validation <i>Terbaik</i> Hyperparameter Pertama	49
Gambar 4. 13. Hasil Akurasi Training dan Validation <i>Terbaik</i> Hyperparameter Pertama.....	49
Gambar 4. 14. Classification Report Percobaan Pertama	50
Gambar 4. 15. Confusion Matrix Percobaan Pertama	50
Gambar 4. 16. <i>Proses Loop</i> Akurasi Training dan Validation <i>Terbaik</i> Hyperparameter <i>Kedua</i>	51
Gambar 4. 17. Hasil Akurasi Training dan Validation Hyperparameter <i>Kedua</i>	51
Gambar 4. 18. Classification Report Percobaan <i>Kedua</i>	52
Gambar 4. 19. Confusion Matrix Percobaan <i>Kedua</i>	52
Gambar 4. 20. <i>Prediction Test Pertama</i>	53

Gambar 4. 21. <i>Prediction Test Kedua</i>	54
Gambar 4. 22. <i>Prediction Test Ketiga</i>	54
Gambar 4. 23. <i>Prediction Test Keempat</i>	55

DAFTAR TABEL

Tabel 2. 1. Rincian Singkat Penelitian.....	16
Tabel 3. 1. Dataset Hasil <i>Labelling</i>	22
Tabel 3. 2. Contoh Dataset Hasil <i>Case Folding</i>	23
Tabel 3. 3. Contoh Dataset Hasil <i>Tokenizing</i>	24
Tabel 3. 4. Contoh Dataset Hasil <i>Stopwords Removing</i>	25
Tabel 3. 5. Contoh Dataset Hasil <i>Punctuation Removing</i>	25
Tabel 3. 6. Contoh Dataset Hasil <i>Lemmatization</i>	26

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kota Medan merupakan ibukota provinsi Sumatera Utara, kota ini merupakan kota terbesar ketiga di Indonesia setelah Jakarta dan Surabaya. Kota Medan membutuhkan pemimpin yang dapat memajukan kota Medan. Maka dari itu Bobby Nasution dan Aulia Rachman selaku Walikota dan Wakil Walikota Medan menyusun konsep pemerintahan kolaboratif, konsep tersebut berisi program-program kerja dari kota Medan yang sedang direalisasikan seperti Medan Berkah, Medan Maju, Medan Bersih. Untuk mendukung program-program tersebut agar berjalan secara maksimal pemerintahan kota Medan meminta seluruh masyarakat agar dapat menyalurkan opini masing-masing ke dalam cuitan di *twitter* dengan tagar #MedanMaju #MedanSejahtera #MedanBersih #MedanBekah #MedanBeridentitas.

Pada zaman ini teknologi sudah berkembang pesat, banyak informasi yang bisa didapatkan dengan sangat mudah, cepat dan efisien. *Twitter* merupakan salah satu bentuk perkembangan teknologi yang dapat memberikan kita kemudahan untuk mengakses berita terkini dan topik yang sedang banyak dibicarakan dengan menggunakan fitur yang bernama *trending*. Setiap pengguna *twitter* juga dapat membagikan cuitan mereka kepada pengguna lain atau yang biasa disebut dengan *tweet*, sehingga berita dan informasi dari satu pengguna dapat tersebar secara *instant* kepada orang lain. Dengan menggunakan media sosial ini, pemerintah memanfaatkan *platform twitter* untuk masyarakat agar dapat memberikan opini-opini dan tanggapan positif, netral serta negatif terhadap program kerja yang ada di kota Medan.

Opini-opini dari masyarakat yang didapat pastinya akan sangat berguna menjadi kritik dan saran yang dapat membangun bagi pemerintah dan akan menjadi bahan pertimbangan sebagai acuan dari tingkat kepuasan masyarakat terhadap kinerja dari program kerja kota Medan. Namun opini yang berada di *twitter* masih dalam bentuk yang acak dan belum tertata, maka dari itu demi mendapatkan hasil data yang sudah terklasifikasi harus dilakukannya beberapa tahapan. Tahapan itu ialah proses analisis sentiment, pertama-tama diperlukannya tahapan pengumpulan data dengan menggunakan teknik *crawling* melalui *Application Programming Interface (API) twitter*, tahapan selanjutnya yaitu dilakukan proses *Preprocessing* (*cleansing*, *case folding*, formalisasi, *stemming*, dan

tokenisasi), kemudian dilakukan tahapan klasifikasi dari data opini masyarakat terhadap program kota Medan dengan mengimplementasikan algoritma *Long Short Term Memory* (LSTM).

Beberapa penelitian mengenai sentimen analisis sudah pernah dilakukan sebelumnya dengan menggunakan berbagai metode, seperti penelitian oleh Putra Fissabil Muhammad *et al.* (2021) dengan judul penelitian *Sentiment analysis using Word2vec and long short-term memory (LSTM) for Indonesian hotel reviews*. Penelitian ini menggunakan *review* atau hasil komentar para pelanggan hotel yang telah menggunakan layanan kamar hotel di Indonesia melalui website Traveloka. Penelitian ini menggunakan *word2vec* dengan pendekatan *skip-gram*, menggunakan *Hierarchical Softmax* pada metode evaluasi dan nilai dimensi vektor yang diatur menjadi 300. Sehingga menghasilkan nilai akurasi maksimal sebesar 85,96%.

Milan Tripathi. (2021) dengan judul penelitian *Sentiment analysis of nepali covid19 tweets using NB SVM and LSTM*. Penelitian ini menggunakan kumpulan dataset dari *twitter* mengenai Covid 19 di Nepal. Penelitian ini menggunakan beberapa algoritma seperti *Bernoulli Naïve Bayes*, *Support Vector Machine*, dan *Long Short Term Memory*. Penelitian ini menghasilkan nilai akurasi tertinggi dengan menggunakan model LSTM yaitu 80%, sedangkan metode *Bernoulli Naïve Bayes* menghasilkan 77,5% akurasi dan *Support Vector Machine* dengan nilai akurasi terkecil yaitu 56,9%.

Fei Long *et al.* (2019) dengan judul penelitian *Sentiment analysis of text based on bidirectional LSTM with multi-head attention*. Penelitian ini menggunakan kumpulan dataset *product review* pada media sosial dengan bahasa Chinese melalui *google*. Pada penelitian ini metode yang digunakan adalah *bidirectional LSTM* dan juga menggunakan mekanisme *Multi-Head Attention*, yang menghasilkan nilai akurasi tertinggi yaitu mencapai 92,11%.

Berdasarkan latar belakang, maka dilakukan penelitian dengan topik “Analisis Sentimen Terhadap Opini Masyarakat Mengenai Program Kerja Kota Medan Menggunakan LSTM dengan Media Sosial *Twitter*”, yang dimana algoritma LSTM diharapkan dapat menghasilkan akurasi yang lebih baik daripada algoritma lainnya, serta dengan adanya proses *embedding* menggunakan *Bert* yang meningkatkan efisiensi dalam memprediksi kata dengan dua arah (*bi-directional*) sehingga menghasilkan akurasi hasil penelitian yang lebih baik jika dibandingkan dengan penelitian sebelumnya yang menggunakan pembobotan kata dengan metode konvensional.

1.2. Rumusan Masalah

Pada skripsi ini rumusan masalah yang ditemukan merupakan pemerintahan kota Medan yang dikepalai oleh Walikota dan wakil Walikota sudah berjalan selama 2 tahun, untuk melihat opini masyarakat terhadap kepemimpinan pemerintah kota Medan dan program kerja yang sudah berjalan selama ini maka dibutuhkan suatu metode analisis sentimen yang dapat menunjukkan reaksi atau opini masyarakat terkait program kerja kota medan berdasarkan cuitan pada sosial media dengan algoritma LSTM.

1.3. Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah untuk menganalisa sentimen yang terdapat pada cuitan dan opini publik terhadap kinerja pemerintahan kota Medan yang dikepalai oleh Bobby Nasution di media sosial *twitter* dengan menggunakan algoritma *Long Short Term Memory*.

1.4. Manfaat Penelitian

a. Bagi Masyarakat

1. Dapat menghasilkan bahan ukur dari kinerja pemerintahan kota Medan agar menjadi evaluasi yang akan meningkatkan kinerja pemerintahan kota Medan.
2. Dapat mengetahui respon masyarakat terhadap program kerja pemerintahan kota Medan yang saat ini sedang berjalan.
3. Diharapkan dapat menjadi data yang akan dipertimbangkan oleh pemerintahan untuk mengetahui program kerja yang dilakukan sudah sesuai dengan SOP (*Standart Operating Procedur*) dan tepat sasaran.

b. Bagi Ilmu Pengetahuan

1. Dapat mempelajari, mengimplementasikan dan melakukan analisis terhadap sentiment menggunakan metode *deep learning* algoritma LSTM.
2. Dapat mengetahui tahapan melakukan *crawling* data menggunakan *library tweepy* pada media sosial *twitter*.
3. Dapat memperoleh *accuracy* dari algoritma LSTM dalam melakukan analisis sentimen.

1.5. Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Dataset yang digunakan dalam bahasa Indonesia.
2. Data yang digunakan berupa *tweet* dalam bentuk *text*, tidak menggunakan emoji dan gambar.
3. Data yang diperoleh merupakan hasil *crawling* dari tanggal 1 Maret 2021 hingga 1 Maret 2023, mengenai opini masyarakat terkait program kerja kota Medan.
4. Data yang diperoleh merupakan opini masyarakat terkait pemerintahan kota Medan dimasa jabatan bapak Bobby Nasution dan Aulia Rachman sebagai Walikota dan wakil Walikota Medan.
5. Dataset yang didapat akan dibagi menjadi data *training* dan *testing*, data yang diperoleh berjumlah 3000 data yang berisikan hasil cuitan masyarakat.
6. Data *crawling* pada *twitter* diambil menggunakan *keyword*: Medan, Kota Medan, Medan Berkah dan *hashtag*: #MedanBersih, #MedanMaju, #MedanMetropolitan, #MedanBeridentitas, #MedanKondusif, #MedanSejahtera.
7. Pada penelitian ini akan digunakan bahasa pemrograman *python* versi 3.8.0.

BAB II

LANDASAN TEORI

2.1. *Twitter*

Twitter merupakan jaringan sosial yang pertama kali dibangun pada bulan Maret 2006, *twitter* adalah salah satu sosial media yang memberikan sebuah layanan pada *user* untuk dapat membagikan konten pendek atau pesan singkat hingga 140 karakter per *tweet* yang biasa disebut dengan situs *microblogging* (Wadhwa *et al.*, n). *Twitter* juga merupakan salah satu dari sepuluh situs yang paling sering untuk dikunjungi. Tercatat pada 2022 pengguna global *twitter* mencapai 500 juta dan 302 juta didalamnya merupakan pengguna aktif. *Twitter* dikenal dengan tempat atau *platform* setiap individu untuk menyampaikan aspirasi, keresahan dan setiap opini *public*. Dengan berbasis teks, *twitter* mengizinkan para penggunanya untuk dapat melakukan *tweet* berupa teks hingga 140 karakter. Namun hingga pada tahun 2017 *twitter* telah memperbaharui ketentuannya hingga saat ini, di mana para pengguna sudah dapat mengirimkan cuitan mereka hingga berjumlah 280 karakter.

2.2. *Natural Language Processing (NLP)*

Natural Language Processing (NLP) adalah cabang dari kecerdasan buatan yang memberi kemampuan pada komputer untuk memahami, memproses, dan menganalisis bahasa manusia melalui apa yang diucapkan maupun ditulis oleh manusia. Teknologi ini sudah berkembang pesat dalam beberapa tahun terakhir. Tujuan utama dari NLP adalah memungkinkan komputer untuk memahami dan berinteraksi dengan bahasa manusia secara efektif dan memahami konteks serta makna di balik teks yang diberikan.

Proses dari NLP melibatkan langkah-langkah seperti tokenisasi (pembagian teks menjadi bagian-bagian yang lebih kecil), lemmatisasi (pengembalian kata dasar), pencarian entitas (pengenalan nama orang, tempat, organisasi, dll.), analisis tata bahasa (menganalisis struktur kalimat), dan pemodelan bahasa (menggunakan statistik untuk memahami pola dan kemungkinan urutan kata). Penerapan NLP sangat luas dan mencakup berbagai bidang seperti pencarian *web*, analisis teks, pengenalan suara, dan banyak lagi. Dengan menggunakan teknik dan algoritma NLP, komputer dapat memproses dan memahami bahasa manusia, sehingga dapat menghasilkan aplikasi yang lebih cerdas dan interaksi yang lebih alami antara manusia dan mesin.

2.3. Analisis Sentimen

Sentiment Analysis atau *Opinion Mining* ialah sebuah studi dimana mengkaji tentang pendapat, perilaku, serta emosi setiap individu terhadap suatu entitas yang bisa mewakili suatu individu, acara atau suatu topik eksklusif (Medhat *et al.*, 2014). Selain itu, di analisis sentimen dilakukan proses agar mengetahui, mengekstrak, serta memproses data teks secara otomatis yang pada akhirnya dapat menghasilkan suatu berita yang berguna (Akbari *et al.*, 2012). Lebih tepatnya, bidang ini memiliki tujuan agar mendapatkan pendapat, *sentiment*, serta emosi sesuai dengan pengamatan orang-orang yang dapat diperoleh melalui teks, perkataan, suara, pergerakan, mimik wajah, serta lain sebagainya (Yadollahi *et al.*, 2017).

Internet memiliki banyak sekali jumlah data yang tidak bisa kita bayangkan di dalamnya, contohnya seperti *review*, opini, *response* dan banyak lainnya. Data-data tersebut berbentuk mentah dan masih belum terklasifikasi, dengan analisis sentimen data-data yang terdapat pada internet dapat diproses dan di klasifikasikan. Dikutip dari Cieliebak, pada tahun 2017 “sentimen analisis merupakan metode pengklasifikasian teks menjadi kelas positif, netral dan negatif”.

2.4. Machine Learning

Machine learning merupakan sebuah bentuk dari mesin yang akan dilatih secara berulang melalui sebuah inputan atau yang disebut dengan data *training* yang akan meningkatkan kemampuan dari mesin untuk menentukan atau memprediksi sebuah percobaan. Menurut Tom Mitchell, *machine learning* merupakan sebuah program yang belajar berdasarkan pengalamannya dan melakukan sebuah percobaan untuk menghasilkan sebuah performa.

Machine Learning juga merupakan cabang dari kecerdasan buatan yang memungkinkan komputer untuk belajar dari data dan meningkatkan kinerjanya sendiri melalui pengalaman. Dalam *Machine Learning*, algoritma dan model statistik digunakan untuk menganalisis data, mengidentifikasi pola, dan membuat prediksi atau pengambilan keputusan berdasarkan pola-pola tersebut. Secara umum, *Machine Learning* melibatkan tiga komponen utama yaitu data, model, dan pembelajaran. Data digunakan sebagai *input* untuk melatih model yang kemudian menggunakan algoritma untuk mengenali pola dan

mempelajari hubungan antara fitur-fitur dalam data. Proses pembelajaran ini menghasilkan model yang dapat digunakan untuk melakukan prediksi atau pengambilan keputusan pada data baru yang tidak pernah dilihat sebelumnya.

Machine Learning memiliki berbagai pendekatan dan teknik, termasuk *supervised learning* (pembelajaran terpandu), *unsupervised learning* (pembelajaran tak terpandu), dan *reinforcement learning* (pembelajaran penguatan). Dalam *supervised learning*, model dilatih menggunakan data yang sudah dilabeli dengan jawaban yang diinginkan. Dalam *unsupervised learning*, model mencari pola-pola yang tidak terlabel dalam data. Sedangkan dalam *reinforcement learning*, model belajar melalui interaksi dengan lingkungan dan mendapatkan umpan balik berdasarkan tindakan-tindakan yang diambil. *Machine Learning* memiliki aplikasi yang luas, termasuk dalam pengenalan wajah, pemrosesan bahasa alami, analisis data, prediksi penjualan, dan banyak lagi. Dengan kemampuan belajar dan beradaptasi, *Machine Learning* memberikan komputer kemampuan untuk mengatasi masalah yang kompleks dan melakukan tugas-tugas yang sulit bagi manusia.

2.5. Deep Learning

Deep Learning adalah pendekatan *Machine Learning* yang memungkinkan mesin untuk mempelajari pola-pola yang semakin kompleks dari data dengan menggunakan arsitektur jaringan saraf yang dalam dan kompleks. *Deep learning* memiliki banyak lapisan yang terhubung secara bertingkat, jaringan saraf dalam *deep learning* dapat mengenali pola-pola yang kompleks dan abstrak dalam data termasuk dalam gambar, teks, dan suara. Kelebihan utama *Deep Learning* terletak pada kemampuannya untuk belajar secara *end-to-end* dari data mentah, sehingga tidak memerlukan ekstraksi fitur manual yang rumit. Dengan melibatkan jaringan saraf yang dalam, *Deep Learning* mampu mengatasi masalah yang kompleks dan mencapai kinerja yang luar biasa pada berbagai tugas pemrosesan data.

2.6. Neural Network

Neural Network (jaringan saraf) adalah model matematika yang terdiri dari sejumlah besar neuron yang saling terhubung. Dalam *Neural Network*, neuron-neuron ini bekerja sama untuk memproses informasi dan menghasilkan *output* berdasarkan pola-pola yang terdapat dalam data. Melalui proses pembelajaran, *Neural Network* dapat mengenali dan mempelajari pola-pola kompleks, sehingga mampu melakukan tugas-tugas seperti pengenalan gambar, pemrosesan bahasa alami, dan prediksi. Dalam beberapa tahun

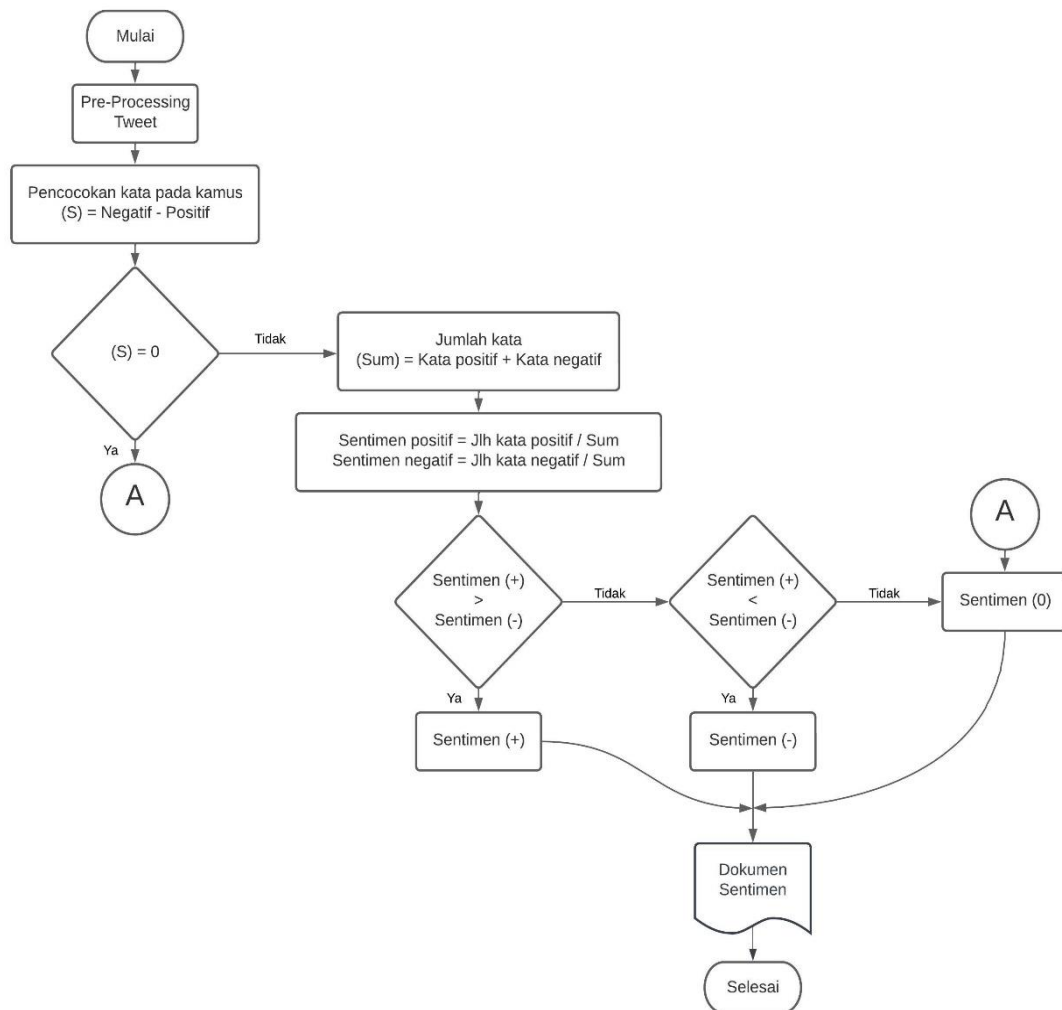
terakhir, kemajuan dalam algoritma dan arsitektur *Neural Network* telah menghasilkan kinerja yang luar biasa dalam berbagai aplikasi seperti pengenalan wajah, deteksi anomali, dan pemrosesan bahasa alami. *Neural Network* memiliki beberapa lapisan yang terhubung secara bertingkat, yaitu lapisan *input*, lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Lapisan *input* menerima data masukan, lapisan tersembunyi melakukan pemrosesan terhadap data tersebut, dan lapisan *output* menghasilkan prediksi atau *output* yang diinginkan.

2.7. *Lexicon-Based*

Pada penelitian ini data yang digunakan berjumlah 3000 data dan dalam melakukan proses penelitian analisis sentimen perlu dilakukan tahapan pembobotan atau pelabelan sentimen pada setiap data yang dimiliki. Proses pelabelan tersebut akan sangat tidak efisien jika dilakukan dengan cara manual satu-persatu, dengan sebab itu pada proses sentimen analisis ini akan dilakukan tahapan pembobotan otomatis menggunakan pendekatan *word level* dengan metode *lexicon based*. Metode *lexicon* ini memiliki basis yang didasari oleh *Dictionary Based Approach*, metode ini memuat kumpulan kata-kata yang terdapat pada data yang ingin diproses lalu menyesuaikan dengan kamus *lexicon* yang nantinya akan memberikan *output* apakah data yang diproses merupakan kalimat yang bersifat opini atau tidak.

Kelebihan dari metode ini, kalimat yang terdapat pada data yang ingin diproses akan dilakukan pengecekan menggunakan tahapan *word level*, yang artinya setiap kata yang terdapat pada kalimat akan dicocokkan dengan kamus yang terdapat pada *lexicon* dan jika kata yang terdapat pada kalimat ditemukan pada kamus *lexicon* maka kalimat tersebut akan dinyatakan sebagai kalimat beropini dan akan ditentukan apakah bernilai positif ataupun negatif. Namun pada metode ini terdapat kekurangan juga yaitu saat kalimat yang diproses tidak terdapat kata yang sesuai pada kamus *lexicon*, maka kalimat tersebut akan dinyatakan tidak beropini, walaupun mungkin saja kalimat tersebut sebenarnya memiliki opini di dalamnya.

Adapun alur dari proses pelabelan dengan menggunakan metode *Lexicon-based* dapat dilihat pada gambar berikut:



Gambar 2. 1. *Flowchart* Labelisasi Dataset

2.8. *Text Preprocessing*

Tahapan yang wajib dilakukan sebelum melakukan proses pada *machine learning* adalah *text preprocessing*. *Text preprocessing* merupakan tahap di mana data mentah akan disiapkan dan dibersihkan sebelum menuju ke tahap-tahap selanjutnya (Dianati *et al.*, 2022). Pada tahap ini, data yang tidak sesuai dan bermasalah akan dihilangkan untuk menghindari terjadinya *missing value*, lalu diubah menjadi bentuk yang lebih sederhana dan mudah untuk dipahami oleh sistem. Sehingga, proses klasifikasi akan menjadi lebih cepat serta data lebih mudah untuk diolah dan cocok untuk *machine learning*.

2.8.1. *Case Folding*

Tahap di mana seluruh kata-kata yang ada dalam data mentah akan diubah dan disamaratakan. Kata-kata tersebut akan menjadi berbentuk sama atau seragam, seperti perubahan seluruh huruf besar dan huruf kapital menjadi huruf kecil (Dianati *et al.*, 2022). Selain itu, simbol dan angka yang tidak memiliki arti penting dan khusus dalam data akan dihapus atau dihilangkan seperti tanda tanya (?), seru (!), koma (,) dan sebagainya.

2.8.2. *Tokenizing*

Tokenisasi merupakan tahap dimana teks dalam data yang diperoleh akan dipecah atau dipisahkan menjadi bagian-bagian yang lebih kecil atau perkata dan diberikan token (Dikiyanti *et al.*, 2021). Sehingga data tersebut menjadi lebih mudah untuk dinilai, serta dilihat frekuensi kemunculannya untuk diartikan oleh mesin.

2.8.3. *Stopwords Removing*

Tahap dimana terjadinya pengurangan atau penghilangan kata-kata umum yang sering muncul dan tidak memiliki makna atau kata-kata yang kurang penting dalam data sesuai dengan *corpus*, seperti kata “di”, “itu”, “ini”, “yang”. Sehingga pada proses selanjutnya dapat terfokus pada kata-kata yang penting saja.

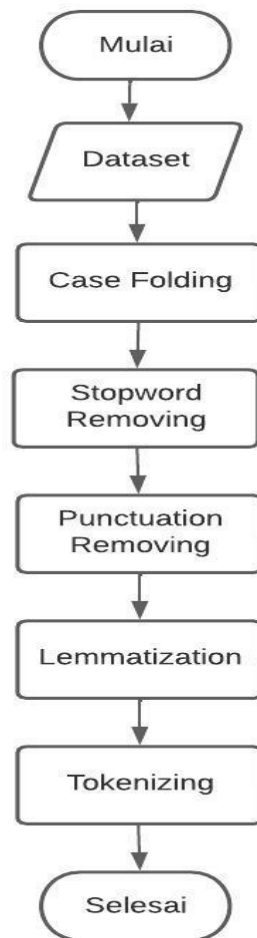
2.8.4. *Punctuation Removing*

Merupakan suatu proses atau tahapan dimana terjadi penghilangan karakter yang tidak memiliki arti dan pengaruh pada data yang nantinya akan diproses, yaitu tanda baca dan *symbol* seperti *hash* (#), *at* (@), *tilde* (~), *exclamation point* (!), and (&) (Dikiyanti *et al.*, 2021).

2.8.5. *Lemmatization*

Lemmatization adalah tahapan perubahan dari data yang diproses agar menemukan bentuk *root word* dengan menghilangkan kata sisipan, akhiran, dan awalan. Hal ini dilakukan untuk dapat meminimalisir kemungkinan mesin untuk melakukan kesalahan pada proses *training* atau pembelajaran dalam pencocokan *keyword*.

Adapun alur dari proses *text pre-processing* dapat dilihat pada gambar berikut:



Gambar 2. 2. *Flowchart Text Preprocessing*

2.9. *BERT Embedding*

BERT (Bidirectional Encoder Representations from Transformers) Embedding merupakan metode pengkodean teks yang menggunakan model BERT untuk menghasilkan representasi vektor yang kaya dan kontekstual untuk setiap kata dalam teks. *BERT Embedding* memanfaatkan kekuatan arsitektur Transformer dalam memahami konteks dan pemodelan bahasa dengan kemampuan untuk memperhitungkan konteks sebelum dan sesudah kata yang sedang diproses (Devlin et al, 2018). Dengan *BERT Embedding*, kata-kata dalam teks diberikan representasi vektor yang menggambarkan makna dan hubungan mereka dalam kalimat secara kontekstual. Representasi vektor ini dapat digunakan dalam berbagai tugas pemrosesan teks seperti sentiment analisis, pemodelan bahasa, dan pemahaman teks.

Berikut adalah langkah-langkah umum dalam pemrosesan sentimen analisis menggunakan *BERT Embedding*:

- a. **Persiapan Data:** pada langkah ini data teks yang akan digunakan untuk analisis sentimen akan dipersiapkan terlebih dahulu dengan cara membersihkan data dan menghapus tanda baca dan karakter khusus yang tidak sesuai.
- b. **Tokenisasi:** data teks yang sudah dipersiapkan harus dipecah menjadi bagian-bagian yang lebih kecil perkata atau token.
- c. **Padding:** karena BERT mengharapkan urutan *input* yang seragam, langkah ini melibatkan menyesuaikan panjang urutan token menjadi panjang yang sama dengan *padding* (misalnya, dengan menambahkan token khusus [PAD]) atau memotong jika terlalu panjang.
- d. **Konversi ke Vektor:** setiap token dalam urutan data diberikan representasi vektor menggunakan model BERT yang telah dilatih sebelumnya. *BERT Embedding* menghasilkan representasi vektor yang memperhitungkan konteks sebelum dan sesudah kata tersebut dalam teks.
- e. **Analisis Sentimen:** Setelah konversi teks menjadi vektor menggunakan *BERT Embedding*, kemudian akan dilakukan proses atau tahapan pengklasifikasian dimana model akan mengklasifikasikan teks menjadi sentimen positif, negatif, atau netral yang akan dilakukan menggunakan algoritma LSTM, untuk pemrosesan LSTM dapat dilihat pada tahapan selanjutnya.

2.10. Algoritma LSTM

Algoritma LSTM (Long Short-Term Memory) adalah jenis arsitektur jaringan saraf rekurensi (RNN) yang dirancang khusus untuk mengatasi masalah peringatan ketergantungan jangka panjang dalam pemrosesan urutan data. RNN memiliki keterbatasan dalam mempertahankan dan menyimpan informasi jangka panjang dalam urutan data, tetapi LSTM menggunakan sel memori yang kemampuan untuk menyimpan dan mempertahankan informasi dalam jangka waktu yang lebih lama. Pada LSTM terdapat 3 jenis gates antara lain ialah, input gate, output gate dan forget gate. Input gate berfungsi sebagai gerbang yang akan menentukan untuk memperbaharui inputan dari memory state. Forget gate merupakan gerbang yang berfungsi untuk menentukan inputan akan digunakan atau dibuang. Serta output gate berfungsi sebagai gerbang yang menentukan apakah output yang dikeluarkan bernilai sama dengan input dan memory state (Gopalakrishnan et al.,

2020). Dengan adanya gates ini LSTM dapat memiliki kemampuan untuk mempertahankan atau menghapus informasi. Sehingga memungkinkan LSTM untuk mempelajari ketergantungan jangka panjang antara elemen-elemen dalam urutan data, seperti dalam teks, suara, atau rangkaian waktu. Arsitektur LSTM sudah terbukti efektif dan populer dalam berbagai tugas pemrosesan urutan data, karena kemampuannya dalam mempertahankan dan menyimpan informasi dalam jangka waktu panjang.

Dibawah ini merupakan tahapan-tahapan dalam penggunaan algoritma LSTM:

- a. Membangun Arsitektur LSTM: arsitektur LSTM dikonstruksi dengan menentukan jumlah lapisan LSTM, jumlah unit dalam setiap lapisan, dan konfigurasi lainnya. Lapisan LSTM menerima vektor kata sebagai input dan menghasilkan output yang mengandung informasi.
- b. Pencarian Hyperparameter dengan Optuna: sebelum dilakukan pembuatan dan pelatihan model akan dilakukan pencarian best hyperparameter menggunakan optuna, hal ini bertujuan untuk memaksimalkan akurasi dari model itu sendiri dan meminimalisir terjadinya loss value pada dataset pelatihan, validation dan test untuk mencegah terjadinya overfitting pada model
- c. Pelatihan: Arsitektur LSTM dilatih menggunakan data pelatihan yang telah diberi label sentimen. Selama pelatihan, bobot dan parameter dalam LSTM disesuaikan untuk meminimalkan kesalahan prediksi dan meningkatkan akurasi dalam mengklasifikasikan sentimen.
- d. Evaluasi dan Prediksi: Setelah pelatihan, performa LSTM dievaluasi menggunakan data pengujian yang tidak terlihat sebelumnya. LSTM dapat digunakan untuk melakukan prediksi sentimen pada data teks baru dengan memberikan label sentimen positif, negatif, atau netral.

2.11. *Hyperparameter Dengan Optuna*

Optuna adalah sebuah *library Python* yang digunakan untuk melakukan optimasi parameter secara otomatis. Tujuan penggunaan optuna itu sendiri adalah untuk mencari atau memperoleh kombinasi parameter terbaik yang dapat menghasilkan kinerja model yang optimal. Optuna menggunakan algoritma optimasi berbasis probabilitas untuk mencari solusi terbaik dengan mengulangi iterasi dan evaluasi model dengan setiap kombinasi parameter. Optuna adalah kerangka pengoptimalan *hyperparameter* generasi

berikutnya yang secara otomatis menentukan *hyperparameter* dan fitur selama pembelajaran mesin. Optuna menggunakan *sequential model-based optimization* (SMBO) dengan algoritma *tree-structured Parzen estimator* (TPE), yang memungkinkan optimalisasi *hyperparameter* yang efisien dan fleksibel (Akiba et al., 2019).

2.12. Evaluasi

2.12.1 Confusion Matrix

Confusion matrix merupakan sebuah *tools* yang membantu untuk melakukan perhitungan akurasi yang biasa disebut juga dengan *error matrix*. Tabel matrix ini berupa tabel 2 dimensi dengan parameter *true positive*, *true negative*, *false positive* dan *false negative*. Adapun tabel dari *confusion matrix* dapat dilihat pada gambar berikut:

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2.3. Tabel *Confusion Matrix*

2.12.2 Classification Report

Classification report merupakan metrik pada *machine learning* yang digunakan untuk menentukan nilai dari *precision*, *recall*, *F1-score*, dan *support*. Setelah kita mendapatkan nilai dari *confusion matrix* kita dapat menentukan nilai dari *precision*, *recall*, *F1-score*, dan *support* menggunakan rumus seperti dibawah ini:

$$\begin{aligned}
 accuracy &= \frac{TP + TN}{P + N} \\
 precision &= \frac{TP}{TP + FP} \\
 recall &= \frac{TP}{TP + FN} \\
 f1 - score &= 2 \times \frac{precision \times recall}{precision + recall}
 \end{aligned}$$

Gambar 2.4. Rumus *Classification report*

2.13. Penelitian Terdahulu

Berikut ini merupakan beberapa penelitian terdahulu yang berkaitan dengan sentiment analisis dengan menggunakan algoritma LSTM. Pada penelitian mengenai sentimen analisis terhadap *review* hotel di indonesia menggunakan *word2vec* dan *Long Short Term Memory*, penelitian ini dapat menghasilkan akurasi terbaik sampai dengan 85,96% dengan menggunakan parameter-parameter seperti menggunakan pendekatan *skip-gram* pada *word2vec architecture* untuk memprediksi kata daripada menggunakan *CBOW*, menggunakan *Hierarchical Softmax* pada metode evaluasi daripada *Negative Sampling* dan nilai dimensi vektor diatur menjadi 300. (Putra Fissabil Muhammad, Retno Kusumaningrum, Adi Wibowo, 2021).

Pada penelitian mengenai analisis sentimen pada *review* film pada situs IMDB dan Amazon *product* menggunakan algoritma LSTM. Algoritma metode *Deep Learning* seperti metode LSTM ini menampilkan hasil yang lebih baik dengan memperoleh akurasi 85% (Dr. G. S. N. Murthy, S. Rao Allu, B. Andhavarapu, M. Bagadi, M. Belusonti, 2020).

Pada penelitian ini dilakukan sentimen analisis terhadap *review* film menggunakan pendekatan *Attention* menggunakan algoritma LSTM dan dilakukan komparasi akurasi dengan algoritma lainnya seperti *Vanilla Neural Network*, *Convolutional Neural Network*, *LSTM*, *Bidirectional LSTM* dan penelitian ini sendiri yaitu *Attention Based LSTM*. Hasil yang didapat berupa akurasi tertinggi didapat oleh algoritma *Attention Based LSTM* yaitu dengan 87,43% dibawahnya ada, *Bidirectional LSTM* dengan 86,56%, kemudian *LSTM* dengan 85,04%, diikuti oleh *CNN* 82,3% dan yang terakhir *VNN* dengan 74,7%. (Charu Gupta, Geetansh Chwla, Karan Rawlley, Kritarth Bisht, Mahak Sharma, 2021).

Pada penelitian ini sentimen analisis menggunakan dataset dari *twitter* mengenai Covid 19 di Nepal menggunakan algoritma *Bernoulli Naïve Bayes*, *Support Vector Machine*, dan *Long Short Term Memory*. Berdasarkan penelitian ini model LSTM memperoleh nilai akurasi tertinggi yaitu 80%, jika dibandingkan dengan algoritma *Bernoulli Naïve Bayes & Support Vector Machine* yaitu dengan akurasi 77,5% dan 56,9%. (Milan Tripathi, 2021).

Pada penelitian ini sentimen analisis menggunakan algoritma *Bidirectional LSTM* dan mekanisme *Multi Head Attention* serta dilakukan perbandingan hasil akurasi dengan algoritma lainnya seperti *CNN*, *Bidirectional LSTM*, dan *Attention Based LSTM* yang menghasilkan akurasi tertinggi dimiliki dari *Bidirectional LSTM* dengan mekanisme *Multi Head Attention* mencapai 92,11% sedangkan *CNN* memperoleh 90,01%, *BiLSTM* 90,80%, dan *Attention-BiLSTM* 91,16%. (Fei Long, Kai Zhou, Weihua Ou, 2019).

Rincian singkat penelitian yang telah dilakukan ditampilkan dalam tabel berikut:

Tabel 2. 1. Rincian Singkat Penelitian

No	Peneliti	Metode	Judul	Keterangan
1	Putra Fissabil Muhammad, Retno Kusumaningrum, Adi Wibowo, (2021).	<i>Word2vec</i> dan <i>Long Short Term Memory (LSTM)</i> .	<i>Sentiment Analysis Using Word2vec and Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews.</i>	Penelitian ini menggunakan pendekatan <i>skip-gram</i> pada <i>word2vec architecture</i> untuk memprediksi kata, <i>Hierarchical Softmax</i> pada metode evaluasi dan nilai dimensi vektor yang diatur menjadi 300. Sehingga dapat menghasilkan nilai akurasi maksimal sebesar 85,96%.
2	Dr. G. S. N. Murthy, S. Rao Allu, B. Andhavarapu, M. Bagadi, M. Belusonti, (2020).	<i>Long Short Term Memory (LSTM)</i> .	<i>Text Based Sentiment Analysis Using LSTM.</i>	Penelitian ini mengenai analisis sentimen pada <i>review</i> film pada situs IMDB dan Amazon <i>product</i> menggunakan algoritma LSTM. Algoritma LSTM ini menampilkan hasil nilai akurasi sebesar 85%.

3	Charu Gupta, Geetansh Chwla, Karan Rawlley, Kritarth Bisht, Mahak Sharma, (2021).	<i>Long Short Term Memory (LSTM).</i>	<i>Senti_ALSTM: Sentiment Analysis of Movie Reviews Using Attention Based LSTM.</i>	Penelitian ini menggunakan pendekatan <i>Attention</i> dengan algoritma LSTM (<i>Attention Based LSTM</i>). lalu melakukan komparasi akurasi dengan algortima lain seperti, <i>Vanilla Neural Network</i> , <i>Convolutional Neural Network</i> , <i>LSTM</i> , dan <i>Bidirectional LSTM</i> . Nilai akurasi tertinggi didapat oleh algortima <i>Attention Based LSTM</i> yaitu 87,43%. Sedangkan <i>Bidirectional LSTM</i> memperoleh nilai 86,56%, <i>LSTM</i> 85,04%, <i>CNN</i> 82,3% dan <i>VNN</i> 74,7%.
4	Milan Tripathi, (2021).	<i>Bernoulli Naïve Bayes, Support Vector Machine, dan Long Short Term Memory.</i>	<i>Sentiment Analysis of Nepali Covid 19 Tweets Using NB SVM and LSTM.</i>	Penelitian ini menggunakan dataset dari <i>twitter</i> mengenai Covid 19 di Nepal menggunakan algoritma <i>Bernoulli Naïve Bayes</i> , <i>Support Vector Machine</i> , dan <i>Long Short Term Memory</i> . Menghasilkan nilai akurasi tertinggi pada model LSTM yaitu 80%, sedangkan pada algoritma <i>Bernoulli Naïve Bayes</i> dan <i>Support Vector Machine</i> menghasilkan nilai akurasi 77,5% dan 56,9%.
5	Fei Long, Kai Zhou, Weihua Ou, (2019).	<i>Bidirectional LSTM.</i>	<i>Sentiment Analysis of Text Based on Bidirectional LSTM With Multi-Head Attention.</i>	Penelitian ini menggunakan algoritma <i>Bidirectional LSTM</i> dan mekanisme <i>Multi Head Attention</i> , lalu melakukan perbandingan hasil akurasi

				<p>dengan algoritma lain seperti <i>CNN</i>, <i>Bidirectional LSTM</i>, dan <i>Attention Based LSTM</i>. Nilai akurasi tertinggi dihasilkan oleh algoritma <i>Bidirectional LSTM</i> dengan mekanisme <i>Multi Head Attention</i> yaitu 92,11%.</p> <p>Sedangkan <i>CNN</i> memperoleh nilai 90,01%, <i>BiLSTM</i> 90,80%, dan <i>Attention-BiLSTM</i> 91,16%.</p>
--	--	--	--	--

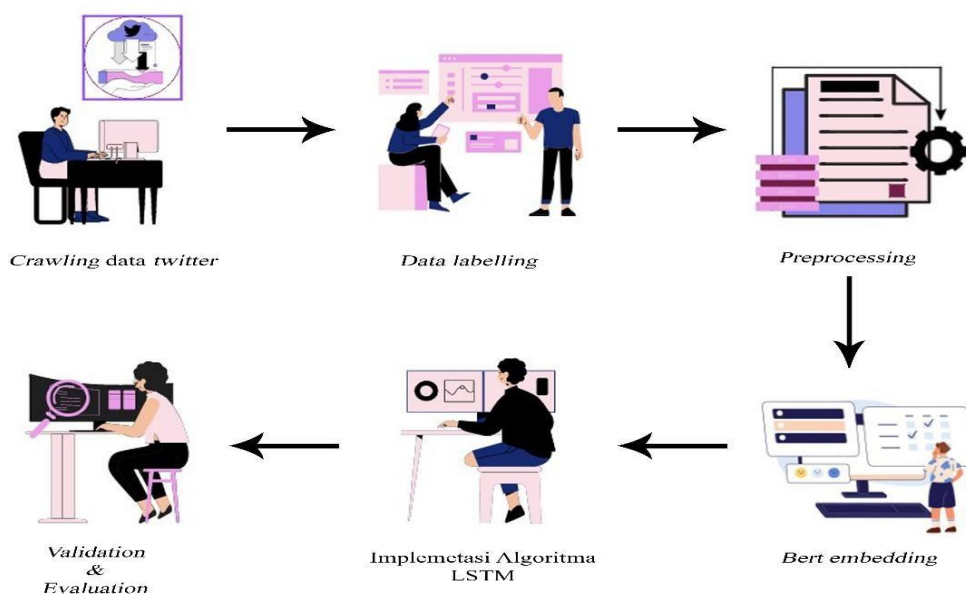
BAB III

ANALISIS DAN PERANCANGAN

Tahapan metodologi penelitian ini akan menjelaskan langkah-langkan yang akan di lewati dalam membangun model sentiment analisis dengan menggunakan metode LSTM. Hal yang pertama kali harus dilakukan sebelum memulai penelitian ini adalah melakukan analisis untuk mengetahui apa permasalahan yang dihadapi dan apa solusi yang akan diterapkan dalam permasalahan yang ingin dipecahkan dengan menggunakan analisis sentimen. Dengan melakukan analisis ini kita dapat memperoleh pemahaman dari kasus yang akan diteliti serta agar memperoleh tujuan akhir, permasalahan yang akan dihadapi, dan ruang lingkup yang akan diteliti. Kemudian akan dilakukan pembentukan rencana sistem yang dibutuhkan atau yang wajib terpenuhi dalam pembentukan model LSTM diantaranya proses pengumpulan data atau *scrapping*, pembobotan data sentiment atau *labelling*, proses penyesuaian data agar data cocok dan dapat digunakan pada model atau *Pre-processing*, lalu tahapan pembobotan kata (*Embedding*) yang sampai pada tahapan akhir yaitu membentuk model LSTM.

3.1. General Arsitektur Sistem

langkah-langkah yang akan dilalui pada penelitian ini “analisis sentiment” terhadap program kerja pemerintahan kota Medan menggunakan metode LSTM (*Long Short Term Memory*) yang dapat dilihat pada, Gambar 3.1 dibawah ini.



Gambar 3. 1. General Arsitektur Sistem

Pada general arsitektur akan ditampilkan tahapan atau proses dari penelitian ini dimulai dari tahapan pengumpulan data (*Crawling data*) menggunakan *library tweepy* pada *twitter*, lalu dilanjutkan tahapan *labelling* data menggunakan pendekatan *lexicon based* agar mempermudah pembobotan nilai sentimen dari data yang sudah dikumpulkan, kemudian data yang sudah di *labelling* akan masuk dalam tahapan *preprocessing* untuk mempersiapkan data agar mudah diproses oleh *machine learning* dan mengurangi *noise* atau *missing value*, lalu akan dilakukan tahapan *Bert embedding* untuk melakukan pembobotan perkata terhadap data yang terkumpul dan yang ingin diproses dan mengubah data menjadi bentuk bilangan riil dalam array, kemudian masuk pada tahapan terpenting yaitu implementasi algoritma *Long Short Term Memory* dan memasuki tahapan akhir yaitu melakukan evaluasi akurasi menggunakan *confusion matrix & classification report*.

3.2. Data Crawling

Data *crawling* merupakan proses pengumpulan data secara otomatis yang dilakukan oleh sebuah program komputer untuk mengambil informasi dari berbagai sumber di internet. Data *crawling* itu sendiri merupakan sebuah metode yang efektif untuk mengakses dan mengumpulkan data dari berbagai situs *web*, termasuk informasi teks, gambar, dan video. Dengan menggunakan algoritma khusus, program tersebut dapat menjelajahi halaman-halaman *web* secara sistematis, mengidentifikasi dan mengekstrak informasi yang relevan, serta menyimpannya dalam format yang dapat dianalisis lebih lanjut. Dengan demikian, data *crawling* memungkinkan peneliti untuk mendapatkan akses ke beragam sumber data yang dan informasi yang luas dan mendalam, memperluas pemahaman dan pengetahuan dalam berbagai bidang studi. Berdasarkan penelitian ini proses *crawling* akan dilakukan pada media sosial *twitter* data yang diperoleh berupa *tweetdate* yaitu waktu pembuatan *tweet* *twitterId* yaitu *unique id* masing masing *tweet*, *location* yaitu lokasi saat *tweet* itu dibuat, *text* yaitu isi dari *tweet*, *profile url* yaitu *link* dari *profile* pembuat *tweet* serta *name* yaitu nama dari pemilik *tweet*. Proses pengumpulan data ini dilakukan dengan menggunakan API *twitter* melalui bahasa pemrograman *python* dan untuk mengakses API dari *twitter* ini diperlukan API *key* sebagai kunci untuk dapat mengakses *documentation* dari API *twitter*. API itu sendiri merupakan antarmuka yang dapat memudahkan *developer* untuk mengakses atau menghubungkan dua aplikasi yang berbeda.

tweetDate	twitterId	location	text	profileUrl	name
Wed May 10 10:29:3	1,62E+09		@bobbynasution_ Baguslah Pak Bobby, buat case "lampu pocong" ini jd lesson-learned	https://twitter.c	John William Girs
Thu May 11 17:08:1	1,46E+18	Medan	@bobbynasution_ @KPK_RI harus segera usut tuntas Proyek "Lampu Pocong" di Kota	https://twitter.c	Erna Sitompul
Fri May 12 12:17:48	1,51E+18	Medan, Indone	Medan tersebut terindikasi ada kelalaian perencanaan."	https://twitter.c	Kubo_comeback
Fri May 12 13:28:28	23343960	Medan	Menurut dia, proyek lampu pocong dilakukan oleh Dinas Kebersihan dan Pertamanan y	https://twitter.c	Erna Sitompul
Wed May 10 03:35:1	57261519		Sementara untuk sanksi dalam proyek lampu pocong akan diberikan pada seluruh ASN	https://twitter.c	Hasmar Kyoto
Mon Mar 20 14:56:1	1,51E+18	Radio KISS FM	@gibran_tweet Harusnya Pak wali Medan @bobbynasution_ belajar yg sama dalam hal	https://twitter.c	akinawa kenzo
Tue May 23 15:00:2	7,05E+08	Tebing Tinggi,	@PartaiSocmed Bahas lampu pocong kota medan dong	https://twitter.c	Akhbaring
Sat May 13 15:16:46	8,84E+08		Dimana proses pengerjaan proyek lampu pocong gagal itu, ditangani oleh Dinas Kebers	https://twitter.c	Kiss Hot Informati
Fri May 12 13:26:19	23343960	Jakarta Pusat,	Bobby menyatakan proyek 1.700 lampu jalan yang disebut mirip pocong itu gagal. Lamp	https://twitter.c	Kompas.com
Sat Mar 04 12:50:44	7,72E+17	Medan Kota, Ir	@HelmiFelis_ Saat ini pembangunan dimedan masih sekitaran medan kota, yang kelih	https://twitter.c	puliadong
Thu Mar 02 05:30:2	1,6E+18	Medan	@DokterTifa @aniesbaswedan Rekam jejak masih gak jelas, langsung jadi panitia. Mec	https://twitter.c	My line
Tue May 09 11:57:0	2,34E+08		@Heraloebs Gede bgt anggaran lampu doang 250m. Seluruh kota medan itu diberi lan	https://twitter.c	hendrik widjaya
Fri May 12 13:26:30	23343960	Kota Medan, S	Pihaknya lantas meminta Dinas Sumber Daya Air Bina Marga dan Bina Konstruksi (SDA	https://twitter.c	isrori nh
Fri May 12 13:28:19	23343960		(SDABMBK) Kota Medan untuk melakukan penagihan menyeluruh terhadap proyek	https://twitter.c	Kompas.com
Mon May 22 14:48:4	1,82E+08		Baru tahu ada banyak lampu pocong tersebar di jalan2 protokol tengah kota, tahunya	https://twitter.c	Erna Sitompul
Sat May 27 01:18:52	44555991	Medan, Indone	@evi_sufiani @budimandjtmiko @KemenkeuRI @DitjenPajakRI @BPKPgoid @bpkri I	https://twitter.c	Erna Sitompul
Tue May 23 05:51:1	7,15E+17	Indonesia	@Syarman59 Kalau peoyek mangkrak lampu pocong 25 M kira2 menfalir ke siapa ya?	https://twitter.c	DatokKong
Tue May 23 08:37:4	47969231	Binjai, Sumate	@lthjosh @PartaiSocmed Walikota mana? Walikota lampu pocong?	https://twitter.c	Ali
Fri May 26 13:43:02	1,23E+18	Indonesia	@mhdarieff_ @FrisianFlagID Pocong ngiri sama Ojol yang bisa sarapan di lampu meral	https://twitter.c	B 455I STA
Thu May 25 16:08:3	1,28E+18	Medan Kota, Ir	@ReVe_kumon Iya biar semangat lagi awak ancurin lampu pocong itu	https://twitter.c	Yerdul
Sun May 21 07:51:3	1,2E+18		@tanyarlffes Lagi famgath akhir tahun kemarin, pasti penuh dong dimana2 tapi villa satu	https://twitter.c	ya

Gambar 3. 2. Data Crawling

3.3. Labelling

Labelling sentiment dataset merupakan sebuah proses atau tahapan dimana memberikan label atau *tag* kepada setiap contoh data dalam dataset yang menggambarkan sentimen atau perasaan yang terkandung dalam teks. Tujuan dari *labelling* ini adalah untuk mengidentifikasi apakah suatu teks mengandung sentimen positif, negatif, atau netral. *Labelling* pada *sentiment analysis* sangat penting dalam pengembangan model dan algoritma untuk analisis sentimen (Li et al, 2019). Dengan memiliki dataset yang dilabeli sentimen, model pembelajaran mesin dapat mempelajari pola dan karakteristik sentimen dalam teks sehingga dapat mengklasifikasikan teks baru ke dalam kategori sentimen yang sesuai. *Labelling sentiment* dataset memberikan landasan yang kuat untuk mengembangkan dan meningkatkan kualitas model analisis sentimen. Pada penelitian ini proses *labelling* akan dilakukan menggunakan metode *lexicon based* dan akan diubah menjadi 3 parameter positif, negatif dan netral, model *lexicon* akan menghitung nilai dari setiap *sentence*, model *lexicon* akan mengidentifikasi kata pada kalimat yang cocok dengan *corpus* pada model contohnya seperti jika terdapat kata “baik, bagus, cantik, memuaskan”, kata-kata pada kalimat akan di hitung berapa jumlah kata positif dan kata negatif yang akan menghasilkan nilai *compound* skala -1 sebagai negatif, 0 sebagai netral dan 1 untuk sentiment positif.

Tabel 3. 1. Dataset Hasil *Labelling*

<i>Tweet</i>	Sentimen
@PartaiSocmed Sbnarnya jalan di Labura yg rusak itu jalan kab, dan ini hampir di semua kab/kota ada jalannya yg rusak, bahkan di Kota Medan sj masih banyak jalan yg perlu perhatian khusus. bagus sih Pak Jokowi, asal jgn ada niat menjatuhkan Citra gubernur skrg utk menaikkan Calon gubernur	-1
@bobbynasution_ Median jalannya tenggelam Pak □ 😞 Beginilah masalah pengaspalan jalan di Kota Medan, tidak pernah dikeruk/dikupas dulu. Alhasil, jalan makin lama makin tinggi. Kalau drainasenya tidak diperbaiki sekaligus, jalan akan lebih cepat rusak dan diaspal lagi, makin tinggi lg. @KemenPU	-1
Perbaikan Jalan Rusak di Kota Medan Harus Segera Dilakukan Cepat	0
Di Medan-Kabanjahe misalnya banyak jalan rusak yang belum diperbaiki sampai hari ini. Ada yang sudah diperbaiki, tetapi sebatas tambal-sulam	0
Boby nasution pura2 tak tau lampu pocong... Masa tiap hari dilihat dia. Di depan rumahnya sendiri masa ngga nampak... Jgn korbakan yg lain anda sendiri melihat itu lampu.	0
Nyampe d rumah mati lampu. Katanya dari jam 3 tadi. Medan ini bener2 kota gak nyaman ditinggali. Mati lampu...	-1

3.4. *Text Preprocessing*

Text preprocessing adalah tahap pemrosesan awal dimana dataset akan dibersihkan, diubah dan disusun sesuai dengan kebutuhan analisis sebelum menuju ketahap selanjutnya. Proses *text preprocessing* melibatkan serangkaian langkah seperti mengubah teks menjadi huruf kecil (*case folding*), memecah teks menjadi bentuk yang lebih kecil (*tokenizing*), penghilangan kata-kata yang tidak bermakna (*stopwords removing*), penghapusan tanda baca (*punctuation removing*), mereduksi kata-kata ke bentuk dasarnya (*lemmatization*), dan lain sebagainya (Jane Doe, 2021). Tahap *text preprocessing*

membantu dalam penghapusan data yang tidak relevan atau sesuai, penanganan *missing value*, normalisasi data, dan penghapusan *noise* atau *outlier*. Maka dari itu dengan melakukan tahap *text preprocessing* yang baik, maka data dapat dibersihkan dari data yang tidak sesuai atau *error* dan menghasilkan dataset yang tepat serta dapat diandalkan untuk proses analisis (Patil dan Sonawane, 2020).

3.4.1. Case Folding

Case folding yang juga dikenal sebagai *text lowercase* adalah proses mengubah semua huruf dalam teks menjadi bentuk yang seragam, biasanya dengan mengubahnya menjadi huruf kecil atau huruf besar. Proses ini bertujuan untuk mengatasi dan menghindari perbedaan dalam kata-kata yang ditulis dengan huruf besar dan huruf kecil. *Case folding* sangat berguna dalam tugas-tugas seperti klasifikasi teks dan pencarian informasi, dimana perbedaan antara huruf besar dan huruf kecil mungkin tidak relevan untuk analisis. Proses *case folding* membantu mempermudah pemrosesan teks dan konsistensi dalam analisis teks (Ahmad Ali, 2020).

Tabel 3. 2. Contoh Dataset Hasil *Case Folding*

<i>Punctuation_remove</i>	<i>Case Folding</i>
PartaiSocmed Sbnarnya jalan di Labura yg rusak itu jalan kab dan ini hampir di semua kab kota ada jalannya yg rusak bahkan di Kota Medan sj masih banyak jalan yg perlu perhatian khusus bagus sih Pak Jokowi asal jgn ada niat menjatuhkan Citra gubernur skrg utk menaikkan Calon gubernur	@partaisocmed sbnarnya jalan di labura yg rusak itu jalan kab dan ini hampir di semua kab/ kota ada jalannya yg rusak bahkan di kota medan sj masih banyak jalan yg perlu perhatian khusus bagus sih pak jokowi asal jgn ada niat menjatuhkan citra gubernur skrg utk menaikkan calon gubernur
jokowi Kab Langkat juga banyak yg rusak pak Kab Deli Serdang juga apalagi jalan lintas Medan Kabanjahe	jokowi kab langkat juga banyak yg rusak pak kab deli serdang juga apalagi jalan lintas medan kabanjahe
Menteng VII kel Medan tenggara kec Medan denai kota Medan dr pagi jam sampe skrg mati lampu Tolonglah pln panas kali ini ah	menteng vii kel medan tenggara kec medan denai kota medan dr pagi jam sampe skrg mati lampu tolonglah pln panas kali ini ah

3.4.2. *Tokenizing*

Tokenizing adalah proses dimana teks dipecah menjadi bagian-bagian yang lebih kecil yang disebut token. Token ini dapat berupa kata, frasa, atau karakter tertentu yang memiliki arti atau makna tersendiri. Proses ini akan dilakukan oleh peneliti menggunakan library “nltk” yang dapat memenuhi kebutuhan peneliti untuk melakukan tokenisasi pada kalimat. Tahapan ini dilakukan bertujuan untuk mempermudah pemrosesan teks lebih lanjut.

Tabel 3. 3. Contoh Dataset Hasil *Tokenizing*

<i>Case_Fold</i>	<i>Tokenizing</i>
partaisocmed sbarnya jalan di labura yg rusak itu jalan kab dan ini hampir di semua kab kota ada jalannya yg rusak bahkan di kota medan sj masih banyak jalan yg perlu perhatian khusus bagus sih pak jokowi asal jgn ada niat menjatuhkan citra gubernur skrg utk menaikkan calon gubernur	['partaisocmed', 'sbarnya', 'jalan', 'di', 'labura', 'yg', 'rusak', 'itu', 'jalan', 'kab', 'dan', 'ini', 'hampir', 'di', 'semua', 'kab', 'kota', 'ada', 'jalannya', 'yg', 'rusak', 'bahkan', 'di', 'kota', 'medan', 'sj', 'masih', 'banyak', 'jalan', 'yg', 'perlu', 'perhatian', 'khusus', 'bagus', 'sih', 'pak', 'jokowi', 'asal', 'jgn', 'ada', 'niat', 'menjatuhkan', 'citra', 'gubernur', 'skrg', 'utk', 'menaikkan', 'calon', 'gubernur']

3.4.3. *Stopwords Removing*

Stopwords removing adalah proses penghapusan kata-kata yang umum dan tidak bermakna dalam teks. Kata-kata ini disebut dengan *stopwords*, misalnya seperti kata "itu", "dan", "di", "yang", dan sebagainya. Dengan menghapus *stopwords*, teks yang dihasilkan akan lebih fokus pada kata-kata yang memiliki makna dan penting dalam teks, sehingga dapat meningkatkan keakuratan dan keefektifan dalam analisis teks. Dalam proses penghapusan kata henti ini akan digunakan library “sastrawi” dimana library ini menyediakan kumpulan kata henti dalam Bahasa Indonesia yang nantinya akan dilakukan penghapusan berdasarkan menggunakan sistem *word piece*

Tabel 3. 4. Contoh Dataset Hasil *Stopwords Removing*.

<i>Tokenizing</i>	<i>Stopword_Remove</i>
['partaisocmed', 'sbnarnya', 'jalan', 'di', 'labura', 'yg', 'rusak', 'itu', 'jalan', 'kab', 'dan', 'ini', 'hampir', 'di', 'semua', 'kab', 'kota', 'ada', 'jalannya', 'yg', 'rusak', 'bahkan', 'di', 'kota', 'medan', 'sj', 'masih', 'banyak', 'jalan', 'yg', 'perlu', 'perhatian', 'khusus', 'bagus', 'sih', 'pak', 'jokowi', 'asal', 'jgn', 'ada', 'niat', 'menjatuhkan', 'citra', 'gubernur', 'skrg', 'utk', 'menaikkan', 'calon', 'gubernur']	['partaisocmed', 'sbnarnya', 'jalan', 'labura', 'rusak', 'jalan', 'kab', 'kab', 'kota', 'jalannya', 'rusak', 'kota', 'medan', 'sj', 'jalan', 'perhatian', 'khusus', 'bagus', 'jokowi', 'jgn', 'niat', 'menjatuhkan', 'citra', 'gubernur', 'skrg', 'utk', 'menaikkan', 'calon', 'gubernur']

3.4.4. *Punctuation Removing*

Punctuation removing adalah proses menghapus tanda baca yang tidak memiliki arti dan pengaruh dalam teks. Tanda baca yang dihapus seperti tanda titik, dan, koma, tanda tanya, tanda seru, hashtag, dan sejenisnya. Dengan begitu teks yang dihasilkan akan berfokus pada kata-kata yang memiliki arti dan struktur kalimat saja, sehingga analisis teks menjadi lebih mudah dan efektif (Agrawal, 2022) peneliti akan menggunakan library “nltk” yang dianggap memiliki kumpulan data tanda baca yang lengkap untuk melakukan penghapusan tanda baca pada penelitian ini.

Tabel 3. 5. Contoh Dataset Hasil *Punctuation Removing*

<i>Raw_Tweet</i>	<i>Punctuation_Remove</i>
@PartaiSocmed Sbnarnya jalan di Labura yg rusak itu jalan kab, dan ini hampir di semua kab/kota ada jalannya yg rusak, bahkan di Kota Medan sj masih banyak jalan yg perlu perhatian khusus. bagus sih Pak	PartaiSocmed Sbnarnya jalan di Labura yg rusak itu jalan kab dan ini hampir di semua kab kota ada jalannya yg rusak bahkan di Kota Medan sj masih banyak jalan yg perlu perhatian khusus bagus sih Pak Jokowi asal jgn ada niat

Jokowi, asal jgn ada niat menjatuhkan Citra gubernur skrg utk menaikkan Calon gubernur	menjatuhkan Citra gubernur skrg utk menaikkan Calon gubernur
@jokowi Kab. Langkat juga banyak yg rusak pak, Kab. Deli Serdang juga apalagi jalan lintas Medan-Kabangahe	jokowi Kab Langkat juga banyak yg rusak pak Kab Deli Serdang juga apalagi jalan lintas Medan Kabangahe
Menteng VII kel. Medan tenggara kec. Medan denai kota Medan dr pagi jam 6 sampe skrg mati lampu. Tolonglah @pln_123 panas kali ini ahhh 🤔🤔	Menteng VII kel Medan tenggara kec Medan denai kota Medan dr pagi jam sampe skrg mati lampu Tolonglah pln panas kali ini ah

3.4.5. Lemmatization

Lemmatization adalah proses mengubah kata-kata dalam teks menjadi bentuk dasarnya yang disebut "*lemma*". Dengan proses ini, kata-kata yang memiliki akhiran, awalan, atau imbuhan yang berbeda namun memiliki makna yang sama akan disatukan menjadi bentuk dasar yang lebih mudah untuk dipahami dan dianalisis. *Lemmatization* membantu dalam normalisasi data teks dan meningkatkan akurasi tugas analisis teks seperti pencarian informasi, analisis sentimen, dan terjemahan mesin (Alonso, 2022) untuk dilakukannya tahapan *Lemmatization* dibutuhkan sebuah library yang memiliki kumpulan kata dalam bahasa Indonesia agar sistem dapat mencari akar kata dari teks inputan dan yang dapat memenuhi kebutuhan tersebut peneliti menggunakan library "nltk" untuk melakukan tahapan ini.

Tabel 3. 6. Contoh Dataset Hasil *Lemmatization*

<i>Stopword_Remove</i>	<i>Lemmitizing</i>
['partaisocmed', 'sbnarnya', 'jalan', 'labura', 'rusak', 'jalan', 'kab', 'kab', 'kota', 'jalannya', 'rusak', 'kota', 'medan', 'sj', 'jalan', 'perhatian', 'khusus', 'bagus', 'jokowi', 'jgn', 'niat', 'menjatuhkan', 'citra', 'gubernur', 'skrg', 'utk', 'menaikkan', 'calon', 'gubernur']	['partaisocmed', 'sbnarnya', 'jalan', 'labura', 'rusak', 'jalan', 'kab', 'kab', 'kota', 'jalan', 'rusak', 'kota', 'medan', 'sj', 'jalan', 'perhati', 'khusus', 'bagus', 'jokowi', 'jgn', 'niat', 'jatuh', 'citra', 'gubernur', 'skrg', 'utk', 'naik', 'calon', 'gubernur']

3.5. *BERT Embedding*

BERT Embedding atau yang biasa disebut dengan *Bidirectional Encoder Representations from Transformers*, merupakan proses pembobotan kata yang diciptakan oleh Jacob Devlin pada tahun 2018. Algoritma BERT merupakan metode *deep learning* yang telah terbukti memberikan hasil yang baik pada pemrosesan *Natural Language Processing* (Andrea *et al.*, 2021). BERT memiliki beberapa keunggulan seperti dapat mengetahui konteks global dari kata. Pada proses *pre-training*, model dilatih agar memperediksi kata yang hilang pada kalimat serta melakukan pelatihan secara *bi-directional* yang artinya BERT akan mempelajari hubungan dari kata dalam dua arah sehingga BERT dapat memperoleh nilai konteks global dari kata dan memperoleh nilai vektor yang sangat akurat dibandingkan dengan metode *embedding* konvensional. Dalam pengimplementasian BERT Embedding nantinya akan digunakan *library transformers* agar dapat memperoleh representasi teks melalui BERT Embedding. Dan untuk memperoleh representasi dari teks menggunakan *BERT Embedding* ada beberapa tahapan yang harus dilalui seperti:

3.5.1 *Preprocessing*

Pada tahap ini, teks yang akan diolah diproses dengan melakukan normalisasi, misalnya menghapus karakter yang tidak diinginkan, mengubah huruf besar ke huruf kecil, dan sebagainya. Selain itu, teks juga akan di-tokenisasi menjadi bagian-bagian kecil yang disebut token, seperti kata-kata dan tanda baca.

3.5.2 *Pembuatan Input Sequence*

Bert menggunakan *input sequence* yang terdiri dari token-token kata yang diawali dengan token [CLS] diakhiri dengan token [SEP] dan menambahkan token [PAD] pada data yang membutuhkan. *Input sequence* itu sendiri memiliki beberapa tahapan pada pemrosesannya seperti:

a. Tokenisasi

Pada tahapan tokenisasi teks akan diproses dan dipecah menjadi bagian-bagian kecil yang disebut token. Setiap kata atau karakter lainnya dalam teks direpresentasikan sebagai token sebelum memasuki tahapan selanjutnya.

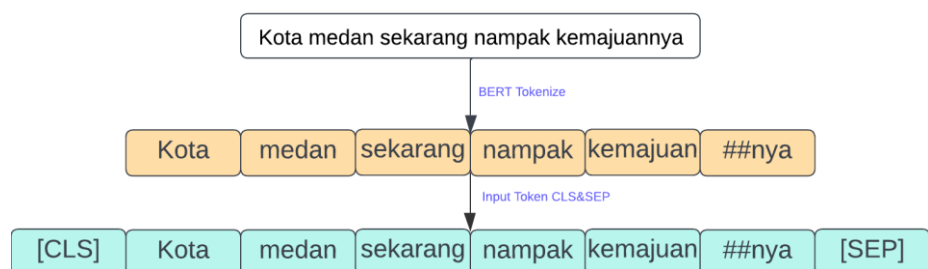


Gambar 3. 3. BERT Tokenize

b. Penginputan Token

Pada tahapan ini akan dilakukan penambahan token pada setiap kalimat yang telah di tokenisasi setidaknya akan ada dua token khusus ditambahkan ke *input sequence* yaitu:

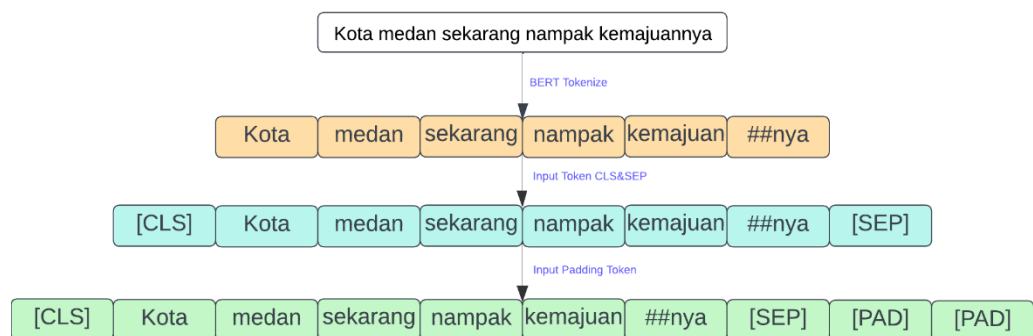
- [CLS]: Token [CLS] (*Classification*), token ini akan ditambahkan pada awalan *input sequence* dan digunakan sebagai tanda khusus untuk menandai awal teks yang akan dianalisis. Pembobotan dari token ini digunakan untuk merepresentasi teks secara keseluruhan dalam klasifikasi teks.
- [SEP]: Token [SEP] (*Separator*), token ini ditambahkan setelah setiap teks atau kalimat yang diproses, token ini sendiri digunakan sebagai tanda khusus untuk menandai akhir teks atau kalimat. Token ini juga akan membantu model dalam memahami struktur dan hubungan antar kalimat dalam teks.



Gambar 3. 4. BERT Token Input

c. Penginputan Token *Padding*

Pada tahapan ini dalam sebuah dataset memiliki ribuan jumlah kalimat, setiap kalimat tersebut memiliki jumlah *length* yang beragam maka dari itu akan dibuat sebuah patokan yang memberikan nilai maksimum *length* dari sebuah kalimat. Jika teks memiliki panjang yang lebih pendek dari panjang maksimum yang diinginkan, token *padding* ditambahkan untuk mencapai panjang yang seragam. Token *padding* biasanya diwakili oleh token khusus [PAD]. *Padding* dilakukan agar semua *input sequence* memiliki panjang yang sama sehingga dapat diolah oleh model secara efisien.



Gambar 3. 5. BERT Padding Input

3.5.3 *Embedding*

Pada tahapan ini *embedding* adalah proses di mana teks atau dataset yang akan diproses diubah menjadi representasi vektor yang mencerminkan makna kata dan konteks dari kata tersebut dan untuk memperoleh representasi vektor tersebut membutuhkan beberapa tahapan seperti:

a. *Word Embedding*

Pada level ini setiap token dalam teks akan diberikan nilai representasi vektor yang mencerminkan makna kata tersebut dalam konteks teks. Representasi vektor ini dihasilkan melalui proses *embedding* yang telah dilatih pada model BERT. Tahapan ini melibatkan informasi dalam satu arah (*unidirectional*), yaitu dari kiri ke kanan (Devlin et al, 2018).

b. *Position Embedding*

Pada tahapan ini model BERT tidak hanya mengambil nilai representasi vektor melalui skala kata saja melainkan model BERT juga

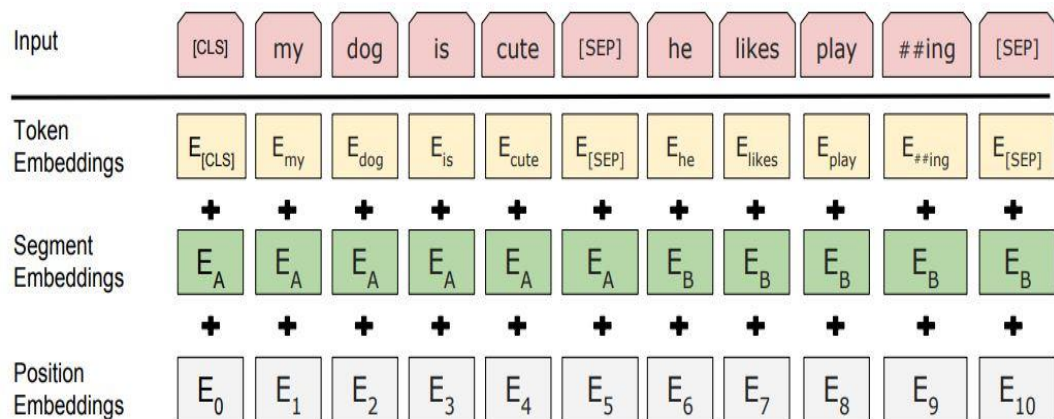
akan mengambil kenyataan bahwa urutan kata dalam teks mempengaruhi makna keseluruhan teks. Untuk memperhitungkan urutan kata, posisi *embedding* ditambahkan pada setiap token (Devlin et al, 2018). Posisi *embedding* adalah representasi vektor yang menunjukkan posisi relatif token dalam teks.

c. *Segment Embedding*

Segment embeddings pada *BERT embedding* digunakan untuk membedakan antara teks-teks yang berbeda dalam *input sequence*. Jika teks terdiri dari beberapa kalimat atau bagian yang berbeda, *segment embeddings* digunakan untuk memberikan tanda *segment embedding* pada setiap token berdasarkan segmen. Hal ini membantu model dalam memahami konteks teks yang lebih luas dan mengenali hubungan antara kalimat-kalimat atau bagian-bagian teks (Devlin et al, 2018).

d. Penjumlahan Representasi

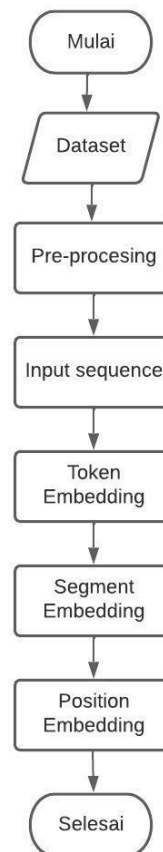
Pada tahapan ini setelah melewati tahapan-tapahan sebelumnya *word embeddings*, *position embeddings*, dan *segment embeddings*, representasi vektor dari ketiga komponen tersebut akan dijumlahkan secara elemen-wise untuk menghasilkan representasi vektor akhir untuk setiap token pada teks. Dengan cara ini, informasi tentang makna kata, posisi relatif, dan *segmen* teks digabungkan dalam representasi vektor (Devlin et al, 2018).



Gambar 3. 6. Penjumlahan representasi vektor kata

e. Normalisasi dan Skalabilitas

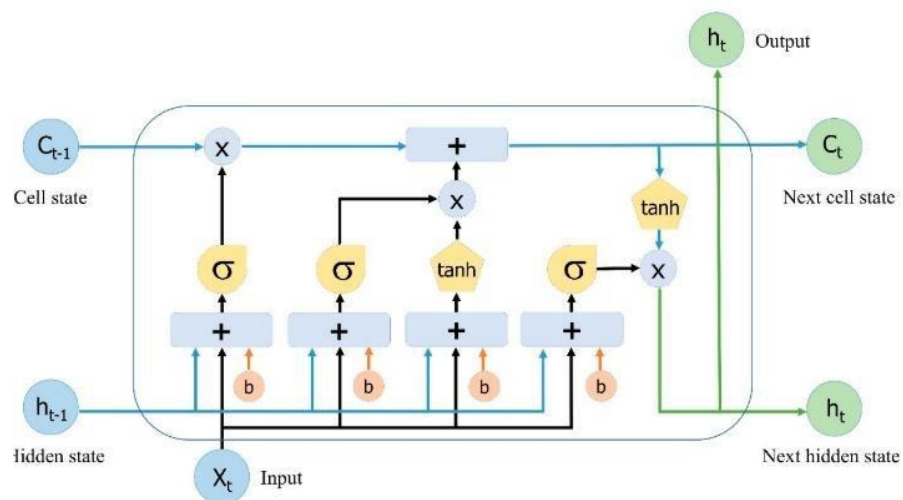
Setelah penjumlahan representasi, tahapan normalisasi akan dilakukan pada setiap token untuk menghasilkan representasi vektor yang stabil dan skalabel. Tahapan normalisasi ini memastikan bahwa representasi vektor memiliki skala yang konsisten dan mempertahankan proporsi relatif antara nilai-nilai (Devlin et al, 2018).



Gambar 3. 7. *Flowchart BERT Embedding*

3.6. Algoritma LSTM

Algoritma LSTM merupakan algoritma *machine learning*, algoritma ini merupakan perbaikan dari *Recurrent Neural Network* yang memperbaiki masalah pada RNN, yaitu LSTM dapat menyimpan informasi dalam memori jangka panjang. Algoritma ini dikembangkan pertama kali oleh Hochreiter dan Schmidhuber (Nurrohmat *et al.*, 2019). Algoritma ini dapat menyimpan data informasi dalam waktu berkepanjangan, sehingga LSTM dapat digunakan untuk mengklasifikasikan, memproses dan memprediksi informasi data lebih baik dari algoritma kebanyakan lainnya. Untuk membangun model ini peneliti membutuhkan sebuah *tools* yang dapat membantu Pembangunan model dalam pemrosesan ini akan digunakan *library torch* hal tersebut disebabkan *torch* merupakan *framework* yang memiliki kemampuan untuk melatih dan mengevaluasi model *machine learning*. LSTM itu sendiri memiliki dua jenis *cell* yang disebut sebagai *hidden state* dan *cell state*. *Cell State* merupakan bagian terpenting yang akan menghubungkan seluruh *output layer* dari LSTM. Pada LSTM terdapat 3 jenis *gates*, dengan adanya *gates* ini LSTM dapat memiliki kemampuan untuk memperbaharui dan menghapus informasi yang terdapat pada *cell state*.

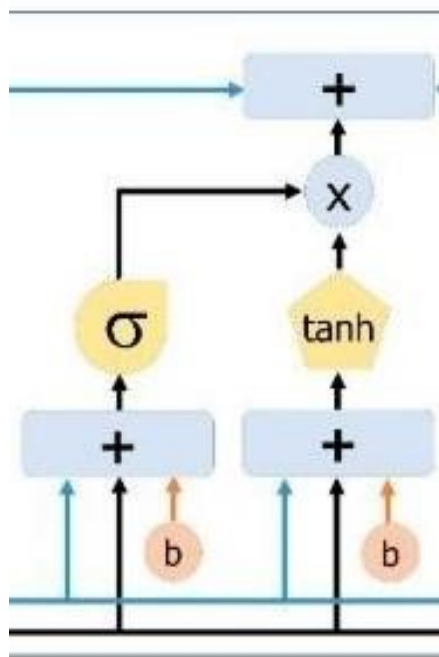


Gambar 3. 12. Arsitektur Algoritma LSTM

Ketiga *gates* ini antara lain yaitu:

3.6.1 *Input Gate*

Pada tahapan ini *input gate* bertanggung jawab untuk mengontrol aliran informasi baru yang akan masuk ke dalam sel memori atau *cell state*. Fungsi utama *input gate* itu sendiri adalah menentukan sampai mana informasi baru akan diperbarui dan disimpan dalam sel memori. Dengan mengontrol aliran informasi, *input gate* memungkinkan LSTM untuk mempelajari dan mengingat informasi penting dalam urutan data.



Gambar 3. 8. *Input Gate Process*

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.1)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

$$N_t = \tanh \cdot (W_n \cdot [h_{t-1}, x_t] + b_n) \quad (2.3)$$

$$f_t = \sigma (w_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.4)$$

$$O_t = \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.5)$$

Untuk mendapatkan hasil *input gate* dapat dilihat dari persamaan (2.1)

a. Nilai Total Bobot *Input*

Pada tahapan ini akan dihitung dengan nilai *input* saat ini yang dilambangkan dengan x_t dan *output* sebelumnya dilambangkan dengan h_{t-1} akan dikalikan dengan matriks bobot yang dilambangkan dengan W_i dan ditambahkan dengan bias.

b. Aktivasi Fungsi Sigmoid

Pada tahapan ini fungsi sigmoid dapat dilihat melalui persamaan (2.2) hasil dari nilai total bobot *input* akan diproses melalui fungsi sigmoid yang akan menghasilkan nilai antara 0 sampai dengan 1 nilai-nilai dalam vector (tanh) ini akan menentukan informasi yang akan diperbarui dan yang akan disimpan pada sel memori.

c. Membuat *Vector* (tanh)

Pada tahapan ini fungsi tanh akan menghasilkan vector yang dihasilkan melalui persamaan (2.3), pada fungsi ini akan menghasilkan nilai dari tingkat kepentingan kata antara -1 sampai dengan 1.

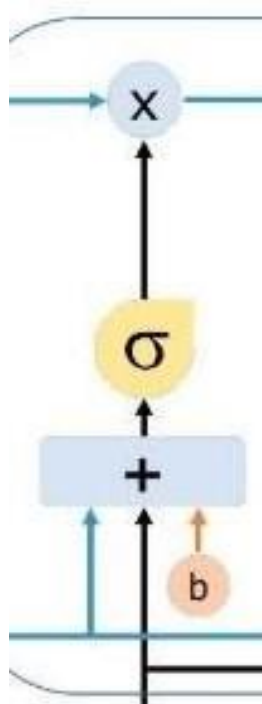
d. Menambahkan Nilai Pada *Cell State*

Pada tahapan ini nilai dari fungsi sigmoid dan fungsi tanh akan dikalikan dan hasil dari proses ini akan ditambahkan ke dalam *cell state*.

Input gate pada LSTM memiliki peran yang sangat penting dalam menjaga aliran informasi yang tepat dan mengontrol penggunaan informasi baru kedalam sel memori. Dengan mengatur jumlah informasi baru yang di *update* dan mengingatnya, *input gate* memungkinkan LSTM untuk mengatasi tantangan seperti dependensi jarak jauh dan penanganan urutan data yang panjang secara efektif.

3.6.2 Forget Gate

Forget gate pada LSTM merupakan komponen penting dalam struktur LSTM yang bertanggung jawab untuk mengendalikan dan mengatur sejauh mana informasi sebelumnya harus dilupakan atau diabaikan dalam sel memori LSTM. *Forget gate* memainkan peran kunci dalam mempertahankan atau menghapus informasi yang tidak relevan atau tidak perlu dari sel memori, sehingga memungkinkan LSTM untuk fokus pada informasi yang lebih penting untuk diteruskan ke dalam *cell state*.



Gambar 3. 9. *Forget Gate Process*

Untuk mendapatkan hasil *forget gate* dapat dilihat dari persamaan (2.4)

a. Nilai Bobot Total *Input*

Tahapan ini akan dihitung dengan nilai *input* saat ini yang dilambangkan dengan x_t dan *output* sebelumnya dilambangkan dengan h_{t-1} akan dikalikan dengan matriks bobot yang dilambangkan dengan W_f dan ditambahkan dengan bias.

b. Aktivasi Fungsi Sigmoid

Pada tahapan ini fungsi sigmoid dapat dilihat melalui persamaan (2.2) hasil dari nilai total bobot *input* akan diproses melalui fungsi sigmoid yang

akan menghasilkan nilai antara 0 sampai dengan 1 nilai-nilai ini akan menentukan informasi yang akan diperbarui dan yang akan disimpan pada sel memori.

c. Melupakan *Cell State* Sebelumnya

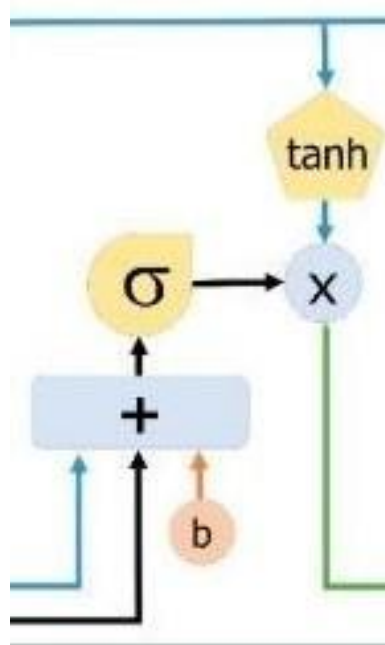
Hasil dari *vector* fungsi sigmoid sebelumnya akan dikalikan dengan *cell memory* sebelumnya hasil dari perkalian tersebut yang akan menentukan sejauh mana elemen dalam sel memori sebelumnya harus dilupakan. Elemen dengan nilai mendekati 1 akan mempertahankan nilai informasi sebelumnya sebaliknya jika elemen bernilai 0.

d. Menambahkan Nilai Pada *Cell State*

Pada tahapan ini nilai yang di dapatkan pada proses-proses sebelumnya akan di *scaling* pada *cell state* baru.

3.6.3 Output Gate

Output gate merupakan bagian yang akan bertanggung jawab untuk mengontrol informasi yang ada dalam sel memori yang harus dikirim ke lapisan *output* atau lapisan berikutnya dalam jaringan. *Output gate* memainkan peran kunci dalam menghasilkan *output* akhir LSTM berdasarkan informasi yang disimpan dalam sel memori.



Gambar 3. 10. *Output Gate Process*

Untuk mendapatkan hasil *Output gate* dapat dilihat dari persamaan (2.5)

a. Nilai Bobot Total *Input*

Tahapan ini akan dihitung dengan nilai *input* saat ini yang dilambangkan dengan x_t dan *output* sebelumnya dilambangkan dengan h_{t-1} akan dikalikan dengan matriks bobot yang dilambangkan dengan W_o dan ditambahkan dengan bias.

b. Aktivasi Fungsi Sigmoid

Pada tahapan ini fungsi sigmoid dapat dilihat melalui persamaan (2.2) hasil dari nilai total bobot *input* akan diproses melalui fungsi sigmoid yang akan menghasilkan nilai antara 0 sampai dengan 1 nilai-nilai ini akan menentukan informasi yang akan diperbarui dan yang akan disimpan pada sel memori.

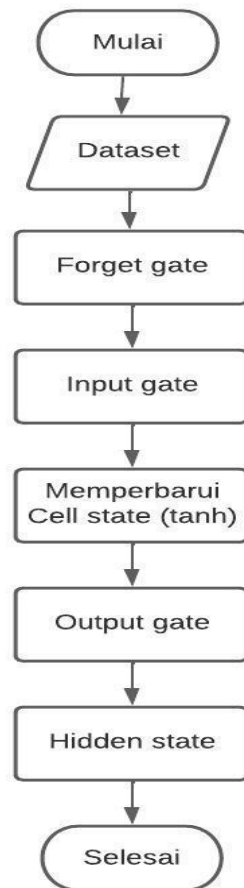
c. Transformasi *Cell State*

Pada tahapan ini nilai *cell state* yang telah melewati beberapa proses dan dilambangkan dengan C_t akan memasuki tahapan fungsi tanh untuk melakukan pembobotan yang dapat dilihat pada persamaan (2.3) untuk menghasilkan nilai *hidden state* namun sebelum memperoleh nilai *hidden state* akan dilakukan *scaling information* dengan hasil dari fungsi *sigmoid output gate*.

d. Kalkulasi *Output*

Pada tahapan ini akan dilakukan penggabungan nilai pada hasil nilai *vector*, tahapan aktivasi fungsi sigmoid akan dikalikan dengan nilai transformasi *cell state* yang telah melalui fungsi tanh. Hasil dari fungsi *output gate* ini akan menghasilkan nilai *hidden state* yang akan diteruskan pada *layer* berikutnya.

Dapat disimpulkan bahwa dalam pemrosesan algoritma LSTM, *input gate* berfungsi sebagai gerbang yang akan menentukan untuk memperbaharui dan menyimpan ke dalam *memory state*. *Forget gate* merupakan gerbang yang berfungsi untuk menentukan inputan akan digunakan atau dibuang. Serta *output gate* berfungsi sebagai gerbang yang menentukan apakah *ouput* yang dikeluarkan akan diteruskan pada lapisan atau jaringan berikutnya atau yang bisa disebut dengan *hidden state* (Gopalakrishnan *et al.*, 2020).



Gambar 3. 11. *Flowchart* Algoritma LSTM

3.7. Optuna

Pada tahapan ini peneliti akan melakukan pencarian *hyperparameter* terbaik yang dapat diperoleh pada proses pelatihan model menggunakan kumpulan parameter yang telah dipersiapkan dalam bentuk *array* yang nantinya *library* optuna ini akan melakukan pemilihan secara otomatis parameter apa yang akan digunakan dalam melakukan pelatihan model parameter yang dipersiapkan antaralain ialah (*hidden_dim*, *num_epoch*, *batch_size*, *learning_rate* dan *dropout_rate*) dengan begitu optuna akan mengambil masing-masing *value* yang telah dipersiapkan pada tiap parameter secara acak dan optuna akan melakukan perhitungan secara *otomatis* agar dapat menyesuaikan parameter terbaik mana yang dapat digunakan sehingga memberikan hasil akurasi terbaik dalam proses pelatihan model ini.

3.8. Evaluasi

Pada tahapan evaluasi ini model akan dimasukan kedalam *eval mode* untuk dapat *merecord* data dan akan dimasukan kedalam *Confusion Matrix & Classification Report* agar memperoleh akurasi serta nilai *f1-score* yang dihasilkan dari model yang telah dibentuk proses ini nantinya akan dilakukan menggunakan *library Sklearn*.

3.8.1. Confusion Matrix

Confusion Matrix adalah metode yang digunakan untuk mengevaluasi kinerja dari sebuah model klasifikasi dengan membandingkan hasil prediksi model dengan nilai sebenarnya. *Confusion Matrix* memiliki empat sel utama yaitu: *True Positive* (TP) yang mewakili jumlah sampel positif yang diklasifikasikan dengan benar, *True Negative* (TN) yang mewakili jumlah sampel negatif yang diklasifikasikan dengan benar, *False Positive* (FP) yang mewakili jumlah sampel negatif yang salah diklasifikasikan sebagai positif, dan *False Negative* (FN) yang mewakili jumlah sampel positif yang salah diklasifikasikan sebagai negatif. Sel-sel ini digunakan untuk menghitung berbagai ukuran evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*. *Confusion Matrix* memungkinkan peneliti untuk membuat keputusan berdasarkan informasi tentang kinerja model dan mengidentifikasi area untuk perbaikan (Johnson, 2022).

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 3. 13. Tabel *Confusion Matrix*

3.8.2 Classification Report

Classification Report adalah sebuah laporan yang menyajikan hasil evaluasi klasifikasi berdasarkan berbagai metrik, seperti presisi (*precision*), *recall*, *f1-score*, dan dukungan (*support*). Presisi mengukur sejauh mana kelas yang

diprediksi sebagai positif benar-benar positif, *recall* mengukur sejauh mana model dapat mengidentifikasi secara akurat semua sampel positif, sedangkan *f1-score* adalah rata-rata harmonis antara presisi dan *recall*, dukungan mengacu pada jumlah sampel dalam dataset yang termasuk dalam setiap kelas target. *Classification Report* membantu memberikan wawasan yang lebih mendalam tentang kinerja model klasifikasi dalam memprediksi kelas-kelas target (John Smith, 2019).

$$accuracy = \frac{TP + TN}{P + N}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$f1 - score = 2 \times \frac{precision \times recall}{precision + recall}$$

Gambar 3. 14. Rumus *Classification report*

3.8.3 *Single Sentence Predict*

Pada tahapan ini peneliti akan membuat sebuah laman depan atau *user interface* menggunakan IDE Pycharm dan *library Seaborn* melalui *UI* ini user akan melakukan prediksi dengan memasukan sebuah inputan berbentuk sebuah kalimat yang dimana kalimat tersebut akan diprediksi oleh model yang telah dibangun sebelumnya pada *google colab* dan disimpan kedalam file .pt (pytorch) untuk menyimpan “*state dict*”.

Gambar 3. 15. Rancangan *UI sentence prediction*

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas mengenai langkah-langkah penerapan algoritma LSTM dengan menggunakan *Embedding* melalui algoritma BERT dalam memprediksi sentimen pada sebuah teks atau kalimat.

4.1. Implementasi Sistem

4.1.1. Spesifikasi Perangkat Keras

Agar bisa melakukan uji analisis sentimen pada komentar-komentar yang digunakan di aplikasi *twitter*, maka dibutuhkan perangkat keras dengan spesifikasi sebagai berikut ini:

- a. *Memory* RAM 8 GB
- b. *Processor* Intel® i5-6200U
- c. Kapasitas SSD 1 *Terabyte*

Penulis juga menggunakan layanan *cloud* gratis dari *Google* yaitu *Google Colab* untuk melakukan pembentukan model serta akurasi dan menggunakan *pycharm* dalam membentuk UI untuk memprediksi sentimen dari model yang telah dibentuk melalui *Google Colab*.

4.1.2. Spesifikasi Perangkat Lunak

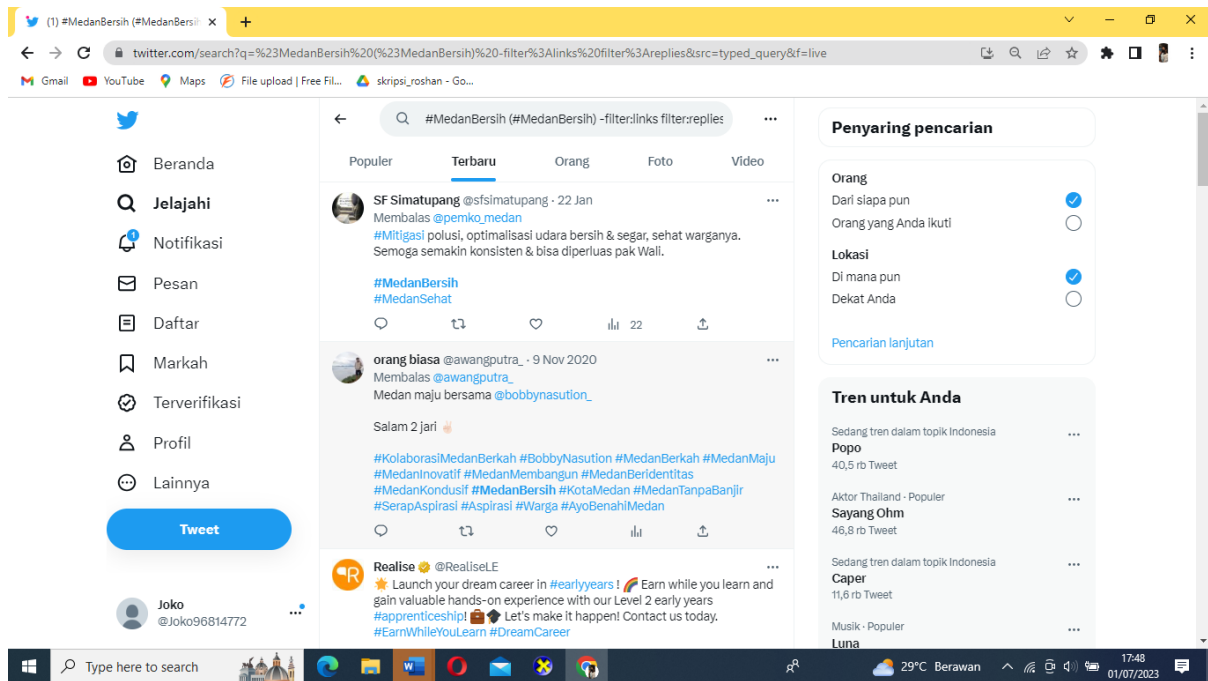
Dalam penelitian ini spesifikasi perangkat lunak yang digunakan adalah sebagai berikut:

- a. Sistem Operasi Windows 10 Pro 64 *Bit operating system*
- b. *Python* 3
- c. *Google Colab*
- d. *Pycharm*
- e. *Library: transformers, sklearn, sastrawi, numpy, seaborn, re, nltk, torch, optuna, pandas.*

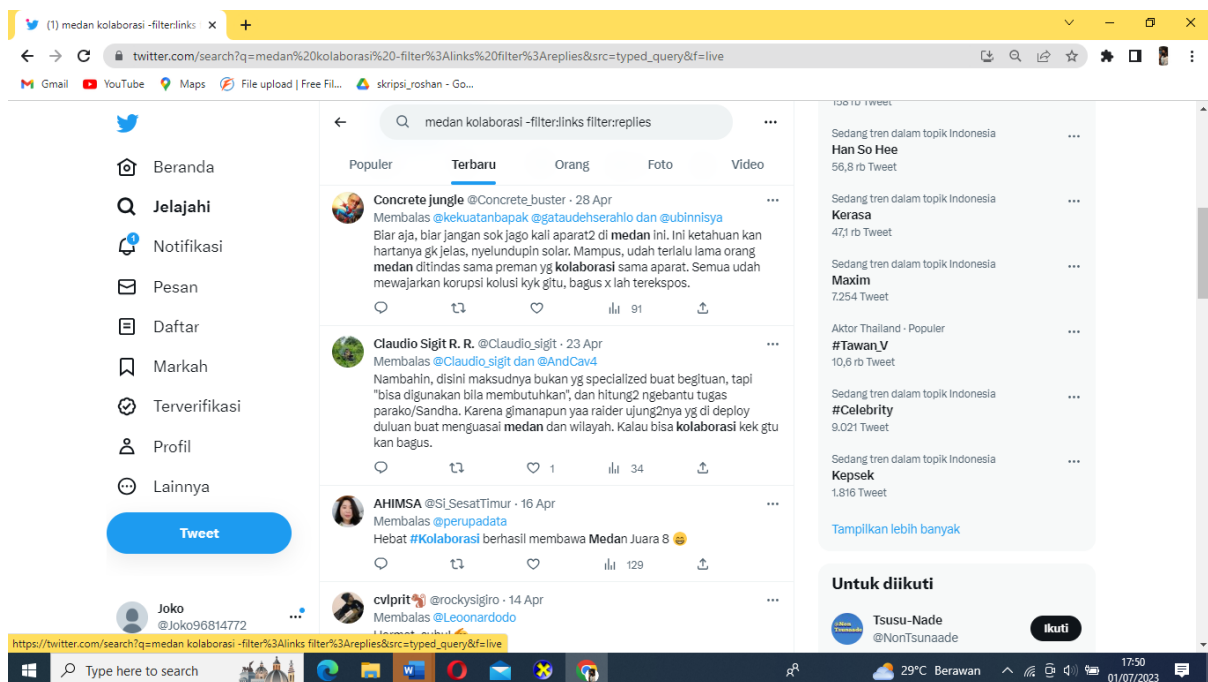
4.2. Implementasi Web Scraping

Pada tahap ini, langkah pertama yang dilakukan penulis adalah membuka halaman aplikasi *twitter* yang kemudian akan menampilkan berbagai *tweet* pada halaman awal

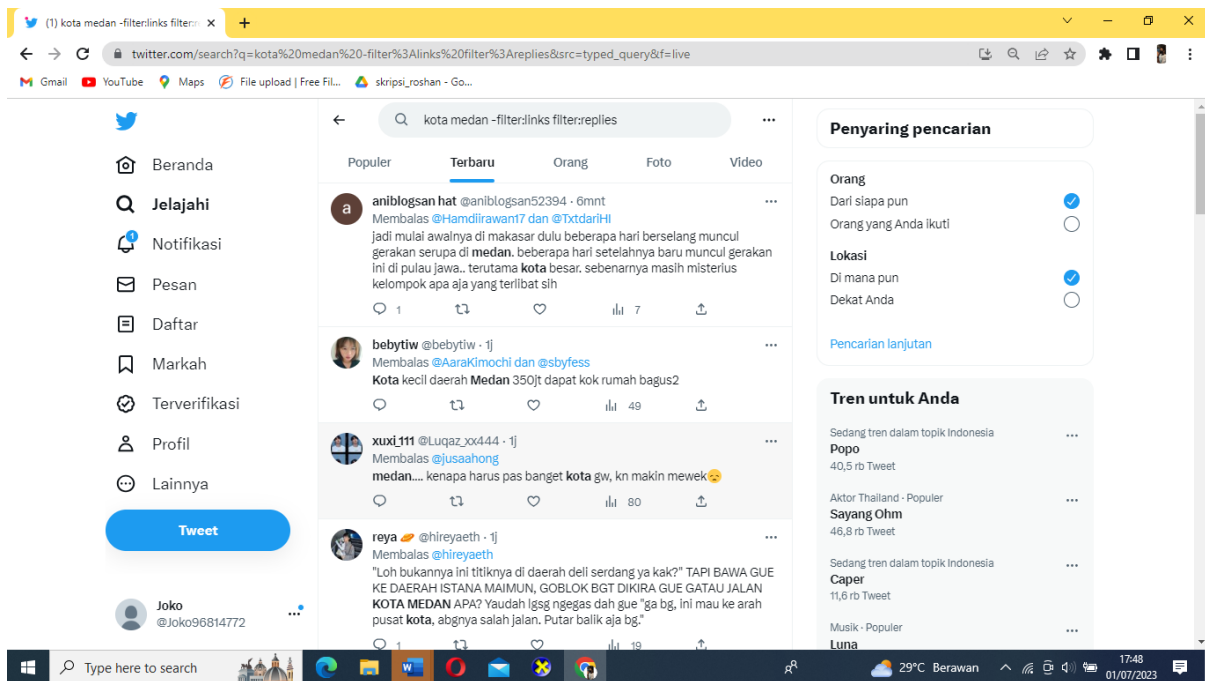
aplikasi tersebut. Lalu penulis melakukan *scraping* data pada *twitter* dengan menggunakan *keyword* tertentu. Setelah itu, penulis melakukan proses data *scraping* dan memperoleh data mentah yang sudah di *scraping*. Data mentah (*tweet*) yang diperoleh selama proses *scraping* berjumlah 5000 data.



Gambar 4. 1. Laman *Twitter* Keyword #MedanBersih



Gambar 4. 2. Laman *Twitter* Keyword Medan Kolaborasi

Gambar 4. 3. Laman *Twitter* Keyword Kota Medan

tweetDate	twitterId	location	text	profileUrName
Wed May 10 10:29:33 +0000 2023	1616266574		@bobbynasution_Bagustah Pak Bobby, buat case "lampu pocong" ini jd lesson-learned agar https://t John William Girsang	
Thu May 11 17:08:15 +0000 2023	1463526618404	Medan	@bobbynasution_ @KPK_Ri harus segera usut tuntas Proyek "Lampu Pocong" di Kota Medan https://t UA NKRI ru	
Fri May 12 12:17:48 +0000 2023	1507695239401	Medan, Indonesia	@CNNIndonesia "Lampu pocong yang telah dipasang di delapan ruas jalan kota Medan terse https://t Kubo_comeback	
Fri May 12 13:28:28 +0000 2023	23343960	ID	Menurut dia, proyek lampu pocong dilakukan oleh Dinas Kebersihan dan Pertamanan yang ki https://t Kompas.com	
Wed May 10 03:35:13 +0000 2023	57261519		Sementara untuk sanksi dalam proyek lampu pocong akan diberikan pada seluruh ASN yang t https://t METRO TV	
Mon Mar 20 14:56:15 +0000 2023	1505486271057	Radio KISS FM Medan	@gibran_tweet Harusnya Pak wali Medan @bobbynasution_ belajar yg sama dalam hal sosn https://t akinawa kenzo	
Tue May 23 15:00:29 +0000 2023	704738840	Tebing Tinggi, Indonesia	@PartaiSocmed Bahas lampu pocong kota medan dong https://t Akhbaring	
Sat May 13 15:16:46 +0000 2023	884075390		Dimana proses pengerjaan proyek lampu pocong gagal itu, ditangani oleh Dinas Kebersihan https://t Kiss Hot Information	
Fri May 12 13:26:19 +0000 2023	23343960	Jakarta Pusat, DKI Jakarta	Bobby menyatakan proyek 1.700 lampu jalan yang disebut mirip pocong itu gagal. Lampu-lan https://t Kompas.com	
Sat Mar 04 12:50:44 +0000 2023	7719301838048	Medan Kota, Indonesia	@HelmiFells_ Saat ini pembangunan dimedan masih sekitaran medan kota, yang kelihatan it https://t puliadong	
Thu Mar 02 05:30:26 +0000 2023	1604372747882	Medan	@DokterTifa @aniesbaswedan Rekam jejak masih gak Jelas, langsung jadi panitia. Medan tu https://t My line	
Tue May 09 11:57:06 +0000 2023	234401938		@Heraloebs Gede bgt anggaran lampu doang 250m. Seluruh kota medan itu diberi lampu p https://t hendrik widjaya	
Fri May 12 13:26:30 +0000 2023	23343960	Kota Medan, Sumatera Ut	Pihaknya lantas meminta Dinas Sumber Daya Air Bina Marga dan Bina Konstruksi (SDABMBK https://t Kompas.com	
Fri May 12 13:28:19 +0000 2023	23343960		@Subur0204 @Miduk17 Lampu pocong kota Medan apa kabar.... 🤔 Baru tahu ada banyak https://t Erna Sitompul	
Mon May 22 14:48:49 +0000 2023	181509540		@evi_sufiani @budimandjatmiko @KemenkeuRI @DitjenPajakRI @BPKPgoid @bpkri Diminta https://t ارسنياوردهنا	
Sat May 27 01:18:52 +0000 2023	44555991	Medan, Indonesia	@Syarman59 Kalau peoyek mangkrak lampu pocong 25 M kira2 menfalir ke siapa ya? https://t DatokKong	
Tue May 23 05:51:13 +0000 2023	7153659044862	Indonesia	@lthjosh @PartaiSocmed Walikota mana? Walikota lampu pocong? https://t Ali	
Tue May 23 08:37:40 +0000 2023	47969231	Binjai, Sumatera Utara	@mhdarieff_ @FrisianFlagID Pocong ngiri sama Ojol yang bisa sarapan di lampu merah https://t B 4551 STA	
Fri May 26 13:43:02 +0000 2023	1233950227620	Indonesia	@ReVe_kumon lya biar semangat lagi awak acurin lampu pocong itu https://t Yerdul	
Thu May 25 16:08:34 +0000 2023	1276873161997	Medan Kota, Indonesia	@tanyarifes Lagi famgath akhir tahun kemarin, pasti penuh dong dimana2 tapi villa satu ini https://t ya	
Sun May 21 07:51:38 +0000 2023	1198457495871		@dennyindrayana Lampu pocong mangkrak, cuma minta pemborong kembalikan dana..BTS n https://t DatokKong	
Sat May 20 12:54:41 +0000 2023	7153659044862		@FaGtn Tower Bts mangkrak, ada tersangka..Lampu pocong mangkrak, kembalikan duit saji https://t DatokKong	
Thu May 25 13:44:08 +0000 2023	7153659044862	Jakarta Capital Region	Tired of endless meeting ? https://t Tom Medema	
Wed May 17 17:08:00 +0000 2023	22298291		@matawardha @Naz_lira Lampu pocong https://t Rayap Radikal	
Tue May 23 07:08:40 +0000 2023	1500712952403			

Gambar 4. 4. Dataset Hasil *Scrapping* Dengan Lokasi

4.3. Labelisasi Dataset

Pada tahap labelisasi ini, dataset yang telah dibersihkan atau telah memasuki tahap *pre-processing* akan di persiapkan untuk dilakukan *labelling* dengan melakukan pencocokan setiap teks dengan pendekatan *word piece* (kata per kata) menyesuaikan dengan *corpus* yang telah tersedia. Kemudian akan dilakukan proses pengidentifikasian, jika nilai *score* dibawah 0 maka teks tersebut bersentimen negatif, jika diatas 0 maka positif dan jika sama dengan 0 akan bersentimen netral.

Semua dataset yang memiliki label sentimen positif, netral, dan negatif akan diganti labelnya menjadi sebuah angka yaitu 1, 0, dan -1. Seperti, jika *tweet* memiliki label sentimen positif maka akan diganti menjadi label angka 1, *tweet* yang memiliki label sentimen netral diganti menjadi label angka 0, dan *tweet* yang memiliki label sentiment negatif diganti menjadi label angka -1. Jumlah data komentar-komentar yang digunakan dalam penelitian ini adalah 3642 data. Jika tahapan ini selesai dilakukan maka dataset sudah siap untuk digunakan dan dapat masuk ke tahap selanjutnya yaitu pembuatan model dan proses pelatihan menggunakan dataset yang telah dilabelisasi ini.

	A	B	C	D
105	lampu masyarakat lampu pocong bentuknya menyerupai pocong warna putih terang	negative		
106	bobbynasution penasaran nih anggaran segede gaban cuman bikin lampu pocong gagal	negative		
107	cnindonesia kocak lampu pocong	negative		
108	detikcom tanggunjiajab pemipinnya ngapain pengerjaan lampu pocong lolos seleksi pengerjaan tanda tanda menyетуjuinya tindakan klo penyimpangan	negative		
109	keberadaan lampu pocong september Pemkot Medan membuka tender proyek lampu jalan paket bernilai Rp miliar	positive		
110	cnindonesia orang Medan gatakut pocong lampu begu proyek gatakut	negative		
111	hasbii lrs fungsinya lampu pocong penerangan ati hiasan	neutral		
112	bobby proyek lampu jalan pocong gagal lampu lampu sejatinya menghiasi mempercantik delapan ruas jalan kota Medan	positive		
113	xix siagianharry dpt proyek lampu pocong bermasalah tuk aparat hukum berani sentuh	negative		
114	livinhumanss moviemendes paa liat tuh keliatan dokter mata trus abis ngebatin oohh keren bentuk tengkorak gini lampu operasinya trus liat mulutnya kebentuk kirain teh kerat	positive		
115	lantas dinas sumber daya air bina marga bina konstruksi sdabmbk kota Medan penagihan proyek lampu pocong mengembalikan anggaran ketiga kontraktor	negative		
116	mochmarkam syahrial nst pemimpin wawasan luas proyek lampu pocong gagal	negative		
117	coklatthazenut dibilang lampu pocong	negative		
118	metro tv tdk gagal proyek namanya lampu pocong setan pocong sukanya ditempat gelap	negative		
119	bobbynasution meng maksud lampu pocong Medan Amelisyf	positive		
120	restupa sadar tindakan gkbs noleh tp matakuk bs digerakin nangkup sosok pocok berdiri dipintu kma pintunya ditutup atau pocong karna samar remang jg lampu kosnya mar	negative		
121	bobbynasution baguslah bobby case lampu pocong jd lesson learned lth mengutamakan implementasi proyek program prioritas mendukung perbaikan infrastruktur utilitas pri	positive		
122	hotellercrypto bobbynasution lampu pocong lokasinya	negative		
123	bobby nasution kontraktor mengembalikan uang Rp miliar dipakai proyek lampu pocong proyek lansekap lampu pocong proyek gagal	negative		
124	cheshlre proyek lampu pocong gajelas gasie anjir	negative		
125	btdariladi desain tahapannya kebaca outputnya menerangi trotoar lampu pocong menerangi parit anethnya tunggu program tuntas mengajak anarki bongkar aset negara	negative		
126	kumpulan kontraktor bayarin utangnya kelak tata kota di dunia Eropa lampu jalan kaya Eropa lampu pocong jeneng ogg lampu pocong lho suwe lampu gendruwo wewe gombor	positive		

Gambar 4. 5. Labelisasi Dataset

4.4. Pre-Processing Dataset

Tahap ini terdiri dari beberapa proses yaitu sebagai berikut:

4.4.1. Case Folding

Pada tahapan ini dilakukan penyamarataan keseluruhan teks menjadi teks *lowercase*.

cleaning_text	case_fold
bobbynasution Baguslah Pak Bobby buat case lam...	bobbynasution baguslah pak bobby buat case lam...
bobbynasution KPK RI harus segera usut tuntas ...	bobbynasution kpk ri harus segera usut tuntas ...
CNNIndonesia Lampu pocong yang telah dipasang ...	cnnindonesia lampu pocong yang telah dipasang ...
Menurut dia proyek lampu pocong dilakukan oleh...	menurut dia proyek lampu pocong dilakukan oleh...
Sementara untuk sanksi dalam proyek lampu poco...	sementara untuk sanksi dalam proyek lampu poco...
...	...
pln pak tolong di cek listrik di jalan syaile...	pln pak tolong di cek listrik di jalan syaile...
Jadi orang konservasi itu berat Medan ke resor...	jadi orang konservasi itu berat medan ke resor...
dedisus Gini aja drpd keja mrt yg terlalu lama...	dedisus gini aja drpd keja mrt yg terlalu lama...
Sungguh saya bahagia lihat saudara saudara yan...	sungguh saya bahagia lihat saudara saudara yan...
jokowi Assamualikum pk alhamdulillah seandai n...	jokowi assamualikum pk alhamdulillah seandai n...

Gambar 4. 6. Case Folding

4.4.2. Tokenizing

Pada tahap ini teks dipecah menjadi bagian-bagian yang lebih kecil yang disebut token. Token ini dapat berupa kata, frasa, atau karakter tertentu yang memiliki arti atau makna tersendiri.

case_fold	tokenizing
bobbynasution baguslah pak bobby buat case lam...	[bobbynasution, baguslah, pak, bobby, buat, ca...
bobbynasution kpk ri harus segera usut tuntas ...	[bobbynasution, kpk, ri, harus, segera, usut, ...
cnnindonesia lampu pocong yang telah dipasang ...	[cnnindonesia, lampu, pocong, yang, telah, dip...
menurut dia proyek lampu pocong dilakukan oleh...	[menurut, dia, proyek, lampu, pocong, dilakuka...
sementara untuk sanksi dalam proyek lampu poco...	[sementara, untuk, sanksi, dalam, proyek, lamp...
...	...
pln pak tolong di cek listrik di jalan syaile...	[pln, pak, tolong, di, cek, listrik, di, jalan...
jadi orang konservasi itu berat medan ke resor...	[jadi, orang, konservasi, itu, berat, medan, k...
dedisus gini aja drpd keja mrt yg terlalu lama...	[dedisus, gini, aja, drpd, keja, mrt, yg, terl...
sungguh saya bahagia lihat saudara saudara yan...	[sungguh, saya, bahagia, lihat, saudara, sauda...
jokowi assamualikum pk alhamdulillah seandai n...	[jokowi, assamualikum, pk, alhamdulillah, sean...

Gambar 4. 7. Tokenizing

4.4.3. Stopwords Removing

Pada tahapan ini akan dilakukan proses penghapusan kata-kata yang umum dan tidak bermakna dalam teks. Kata-kata ini disebut dengan *stopwords*, misalnya seperti kata "itu", "dan", "di", "yang", dan sebagainya.

tokenizing	stopword_removing
[bobbynasution, baguslah, pak, bobby, buat, ca...]	[bobbynasution, baguslah, bobby, case, lampu, ...]
[bobbynasution, kpk, ri, harus, segera, usut, ...]	[bobbynasution, kpk, ri, usut, tuntas, proyek, ...]
[cnnindonesia, lampu, pocong, yang, telah, dip...]	[cnnindonesia, lampu, pocong, dipasang, delapa...]
[menurut, dia, proyek, lampu, pocong, dilakuka...]	[proyek, lampu, pocong, dinas, kebersihan, per...]
[sementara, untuk, sanksi, dalam, proyek, lamp...]	[sanksi, proyek, lampu, pocong, asn, bertangu...]
...	...
[pln, pak, tolong, di, cek, listrik, di, jalan...]	[pln, cek, listrik, jalan, syailendra, kecamat...]
[jadi, orang, konservasi, itu, berat, medan, k...]	[orang, konservasi, berat, medan, resort, sung...]
[dedisus, gini, aja, drpd, kerja, mrt, yg, terl...]	[dedisus, gini, drpd, kerja, mrt, kasih, contoh...]
[sungguh, saya, bahagia, lihat, saudara, sauda...]	[sungguh, bahagia, lihat, saudara, saudara, be...]
[jokowi, assamualikum, pk, alhamdulillah, sean...]	[jokowi, assamualikum, pk, alhamdulillah, sean...]

Gambar 4. 8. Stopwords Removing

4.4.4. Punctuation Removing

Pada tahapan ini dilakukan penghapusan tanda baca yang tidak memiliki arti dan pengaruh dalam teks, baik tanda baca, *mention*, *hashtag*, *link*, nomor, dan *retweet*.

text	cleaning_text
@bobbynasution_ Baguslah Pak Bobby, buat case ...	bobbynasution Baguslah Pak Bobby buat case lam...
@bobbynasution_ @KPK_RI harus segera usut tunt...	bobbynasution KPK RI harus segera usut tuntas ...
@CNNIndonesia "Lampu pocong yang telah dipasan...	CNNIndonesia Lampu pocong yang telah dipasang ...
Menurut dia, proyek lampu pocong dilakukan ole...	Menurut dia proyek lampu pocong dilakukan oleh...
Sementara untuk sanksi dalam proyek lampu poco...	Sementara untuk sanksi dalam proyek lampu poco...
...	...
@pln_123 pak tolong di cek listrik di jalan sy...	pln pak tolong di cek listrik di jalan syaile...
Jadi orang konservasi itu berat. Medan ke reso...	Jadi orang konservasi itu berat Medan ke resor...
@dedisus34740718 Gini aja drpd kerja mrt yg ter...	dedisus Gini aja drpd kerja mrt yg terlalu lama...
Sungguh saya bahagia lihat saudara-saudara yan...	Sungguh saya bahagia lihat saudara saudara yan...
@jokowi Assamualikum pk^alhamdulillah seandai ...	jokowi Assamualikum pk alhamdulillah seandai n...

Gambar 4. 9. Punctuation Removing

4.4.5. Lemmatization

Pada tahap ini kata-kata dalam teks akan diubah menjadi bentuk dasarnya.

stopword_removing	Lemmatize
[bobbynasution, baguslah, bobby, case, lampu, ...]	[bobbynasution, bagus, bobby, case, lampu, poc...]
[bobbynasution, kpk, ri, usut, tuntas, proyek, ...]	[bobbynasution, kpk, ri, usut, tuntas, proyek, ...]
[cnnindonesia, lampu, pocong, dipasang, delapa...]	[cnnindonesia, lampu, pocong, pasang, delapan, ...]
[proyek, lampu, pocong, dinas, kebersihan, per...]	[proyek, lampu, pocong, dinas, bersih, taman, ...]
[sanksi, proyek, lampu, pocong, asn, bertanggu...]	[sanksi, proyek, lampu, pocong, asn, tanggung, ...]
...	...
[pln, cek, listrik, jalan, syailendra, kecamat...]	[pln, cek, listrik, jalan, syailendra, camat, ...]
[orang, konservasi, berat, medan, resort, sung...]	[orang, konservasi, berat, medan, resort, sung...]
[dedisus, gini, drpd, kerja, mrt, kasih, contoh...]	[dedisus, gin, drpd, kerja, mrt, kasih, contoh, ...]
[sungguh, bahagia, lihat, saudara, saudara, be...]	[sungguh, bahagia, lihat, saudara, saudara, ta...]
[jokowi, assamualikum, pk, alhamdulillah, sean...]	[jokowi, assamualikum, pk, alhamdulillah, anda...]

Gambar 4. 10. Lemmatization

4.5. Split Dataset

Pada tahap ini dataset akan dibagi menjadi tiga bagian yaitu data *training*, data validasi, dan data *testing*. Pada pemrosesan *splitting* data ini menggunakan perbandingan 80:20 data *training* dan *testing*.

```
# Split dataset into training, validation, and testing sets
train_data, test_data, train_labels, test_labels = train_test_split(dataset['text'], dataset['label'], test_size=0.2, random_state=42)
train_data, val_data, train_labels, val_labels = train_test_split(train_data, train_labels, test_size=0.2, random_state=42)
```

Gambar 4. 11. Splitting Dataset

4.6. Best Hyperparameter Dengan Optuna

Pada tahapan ini dilakukan pencarian *hyperparameter* terbaik untuk memperoleh nilai akurasi pelatihan, *validation* dan *test* yang terbaik dan meminimalisir *loss* yang akan terjadi pada proses pelatihan dalam model yang sedang dibentuk, tahapan ini dilakukan dengan melakukan iterasi secara terus menerus dalam waktu yang sangat panjang sehingga dibutuhkan komputasi *power* yang banyak untuk memperoleh akurasi *hyperparameter* terbaiknya. Pada penelitian ini dilakukan pencarian *Best Hyperparameter* sebanyak dua kali dan hasil yang diperoleh dapat dilihat pada diagram dibawah:

Berikut adalah *hyperparameter* yang diperoleh dan hasil akurasi yang didapatkan.

- Percobaan Pertama

<i>Hidden_dim</i>	<i>Num_epoch</i>	<i>Batch_size</i>	<i>Learning_rate</i>	<i>Dropout_rate</i>
128	10	32	0.000848578501478349	0.3987110285640226

- Percobaan Kedua

<i>Hidden_dim</i>	<i>Num_epoch</i>	<i>Batch_size</i>	<i>Learning_rate</i>	<i>Dropout_rate</i>
128	13	64	0.00024412636786250215	0.32844858973723867

4.7. Implementasi LSTM Dengan *BERT Embedding*

Tahapan implementasi model LSTM dengan *BERT Embedding* dalam skripsi ini bertujuan untuk memanfaatkan kekuatan kedua model tersebut dalam pemrosesan bahasa alami. LSTM (*Long Short-Term Memory*) merupakan jenis arsitektur *recurrent neural network* (RNN) yang memiliki kemampuan untuk mempertahankan dan mengingat informasi jangka panjang. Sedangkan BERT (*Bidirectional Encoder Representations from Transformers*) adalah model bahasa yang menggunakan transformer dan telah dilatih pada tugas pemodelan yang besar. Dan untuk membangun model tersebut dibutuhkan sebuah proses yang dapat menampung setiap dataset untuk melakukan iterasi yang disebut dengan data *loader* selain itu dibutuhkan juga beberapa *hyperparameter* yang diperoleh melalui tahapan pencarian *best hyperparameter* menggunakan Optuna.

Selain dengan *hyperparameter* diatas terdapat juga beberapa parameter yang dibutuhkan seperti *Embedding dim & num_classes*, *embedding dim* itu sendiri bertujuan untuk menjadi representasi dari token atau kata dalam teks. BERT sendiri telah menentukan nilai untuk *embedding dim* itu sendiri dan menyarankan untuk menggunakan nilai 768 pada *embedding dim*. Sedangkan, *num_classes* merupakan jumlah kategori label yang terdapat pada dataset. pada penelitian ini menggunakan 3 kategori.

Pada tahap pembuatan model dan pelatihan dataset menggunakan *hyperparameter* dengan percobaan pertama

<i>Hidden_dim</i>	<i>Num_epoch</i>	<i>Batch_size</i>	<i>Learning_rate</i>	<i>Dropout_rate</i>
128	10	32	0.000848578501478349	0.3987110285640226


```

Epoch 1/13 - Train Loss: 0.8354 - Train Accuracy: 61.43% - Val Loss: 0.7960 - Val Accuracy: 64.97%
Epoch 2/13 - Train Loss: 0.7789 - Train Accuracy: 65.70% - Val Loss: 0.8084 - Val Accuracy: 66.01%
Epoch 3/13 - Train Loss: 0.7481 - Train Accuracy: 68.67% - Val Loss: 0.7667 - Val Accuracy: 69.02%
Epoch 4/13 - Train Loss: 0.7153 - Train Accuracy: 71.64% - Val Loss: 0.7786 - Val Accuracy: 69.41%
Epoch 5/13 - Train Loss: 0.6771 - Train Accuracy: 72.65% - Val Loss: 0.7626 - Val Accuracy: 69.02%
Epoch 6/13 - Train Loss: 0.6384 - Train Accuracy: 75.39% - Val Loss: 0.7430 - Val Accuracy: 71.11%
Epoch 7/13 - Train Loss: 0.6069 - Train Accuracy: 76.91% - Val Loss: 0.8084 - Val Accuracy: 70.98%
Epoch 8/13 - Train Loss: 0.6107 - Train Accuracy: 75.00% - Val Loss: 0.7475 - Val Accuracy: 70.98%
Epoch 9/13 - Train Loss: 0.5671 - Train Accuracy: 77.86% - Val Loss: 0.8412 - Val Accuracy: 67.97%
Epoch 10/13 - Train Loss: 0.5436 - Train Accuracy: 78.64% - Val Loss: 0.8087 - Val Accuracy: 69.80%
Epoch 11/13 - Train Loss: 0.4953 - Train Accuracy: 81.89% - Val Loss: 0.8293 - Val Accuracy: 70.59%
Epoch 12/13 - Train Loss: 0.4481 - Train Accuracy: 83.18% - Val Loss: 0.9833 - Val Accuracy: 66.80%
[I 2023-07-02 20:32:58,800] Trial 0 finished with value: 0.7430207009116808 and parameters: {'hidden_
Epoch 13/13 - Train Loss: 0.3866 - Train Accuracy: 85.59% - Val Loss: 0.8823 - Val Accuracy: 70.72%
<ipython-input-4-5e71e952192c>:69: FutureWarning: suggest_loguniform has been deprecated in v3.0.0. T
learning_rate = trial.suggest_loguniform("learning_rate", 1e-5, 1e-3)
<ipython-input-4-5e71e952192c>:70: FutureWarning: suggest_uniform has been deprecated in v3.0.0. This
dropout_rate = trial.suggest_uniform("dropout_rate", 0.3, 0.7)
Some weights of the model checkpoint at indolem/indobert-base-uncased were not used when initializing
- This IS expected if you are initializing BertModel from the checkpoint of a model trained on anothe
- This IS NOT expected if you are initializing BertModel from the checkpoint of a model that you expe
Epoch 1/18 - Train Loss: 0.9428 - Train Accuracy: 51.96% - Val Loss: 0.8928 - Val Accuracy: 55.56%
Epoch 2/18 - Train Loss: 0.8603 - Train Accuracy: 59.53% - Val Loss: 0.8533 - Val Accuracy: 62.48%
Epoch 3/18 - Train Loss: 0.8111 - Train Accuracy: 66.20% - Val Loss: 0.8132 - Val Accuracy: 67.19%
Epoch 4/18 - Train Loss: 0.7756 - Train Accuracy: 66.20% - Val Loss: 0.7946 - Val Accuracy: 67.97%
Epoch 5/18 - Train Loss: 0.7523 - Train Accuracy: 68.83% - Val Loss: 0.7685 - Val Accuracy: 67.58%
Epoch 6/18 - Train Loss: 0.7065 - Train Accuracy: 71.30% - Val Loss: 0.7822 - Val Accuracy: 67.45%
Epoch 7/18 - Train Loss: 0.6748 - Train Accuracy: 73.32% - Val Loss: 0.7700 - Val Accuracy: 68.89%
Epoch 8/18 - Train Loss: 0.6457 - Train Accuracy: 74.89% - Val Loss: 0.7677 - Val Accuracy: 69.41%
Epoch 9/18 - Train Loss: 0.6181 - Train Accuracy: 76.35% - Val Loss: 0.7687 - Val Accuracy: 67.32%
Epoch 10/18 - Train Loss: 0.5897 - Train Accuracy: 78.08% - Val Loss: 0.7607 - Val Accuracy: 68.50%
Epoch 11/18 - Train Loss: 0.5584 - Train Accuracy: 79.60% - Val Loss: 0.7681 - Val Accuracy: 69.28%
Epoch 12/18 - Train Loss: 0.5283 - Train Accuracy: 80.27% - Val Loss: 0.8290 - Val Accuracy: 67.32%
Epoch 13/18 - Train Loss: 0.5019 - Train Accuracy: 81.61% - Val Loss: 0.8055 - Val Accuracy: 68.63%
Epoch 14/18 - Train Loss: 0.4718 - Train Accuracy: 82.79% - Val Loss: 0.7859 - Val Accuracy: 69.93%
Epoch 15/18 - Train Loss: 0.4459 - Train Accuracy: 83.35% - Val Loss: 0.8374 - Val Accuracy: 70.20%
Epoch 16/18 - Train Loss: 0.4020 - Train Accuracy: 85.09% - Val Loss: 0.9048 - Val Accuracy: 67.97%
Epoch 17/18 - Train Loss: 0.3522 - Train Accuracy: 87.78% - Val Loss: 0.9224 - Val Accuracy: 68.89%
[I 2023-07-02 22:50:05,825] Trial 1 finished with value: 0.7606755706171194 and parameters: {'hidden_
Epoch 18/18 - Train Loss: 0.3483 - Train Accuracy: 88.17% - Val Loss: 0.9166 - Val Accuracy: 69.93%

```

Gambar 4. 12. Proses Loop Akurasi *Training* dan *Validation* Terbaik *Hyperparameter* Pertama

Hasil akurasi *training* dan validasi terbaik yang digunakan sebagai berikut:

```

Some weights of the model checkpoint at indolem/indobert-base-uncased were not used when initializing Bert
- This IS expected if you are initializing BertModel from the checkpoint of a model trained on another ta
- This IS NOT expected if you are initializing BertModel from the checkpoint of a model that you expect to
Epoch 1/10 - Train Loss: 0.8415 - Train Accuracy: 60.09% - Val Loss: 0.7767 - Val Accuracy: 64.67%
Epoch 2/10 - Train Loss: 0.7701 - Train Accuracy: 67.30% - Val Loss: 0.7184 - Val Accuracy: 68.95%
Epoch 3/10 - Train Loss: 0.7167 - Train Accuracy: 70.34% - Val Loss: 0.7660 - Val Accuracy: 68.78%
Epoch 4/10 - Train Loss: 0.6969 - Train Accuracy: 71.80% - Val Loss: 0.7086 - Val Accuracy: 69.81%
Epoch 5/10 - Train Loss: 0.6609 - Train Accuracy: 74.33% - Val Loss: 0.6884 - Val Accuracy: 72.56%
Epoch 6/10 - Train Loss: 0.6190 - Train Accuracy: 75.75% - Val Loss: 0.6963 - Val Accuracy: 71.18%
Epoch 7/10 - Train Loss: 0.5763 - Train Accuracy: 77.85% - Val Loss: 0.6868 - Val Accuracy: 73.58%
Epoch 8/10 - Train Loss: 0.5348 - Train Accuracy: 79.14% - Val Loss: 0.6954 - Val Accuracy: 73.76%
Epoch 9/10 - Train Loss: 0.4898 - Train Accuracy: 81.07% - Val Loss: 0.7595 - Val Accuracy: 71.87%
Epoch 10/10 - Train Loss: 0.4420 - Train Accuracy: 83.95% - Val Loss: 0.7042 - Val Accuracy: 72.90%

```

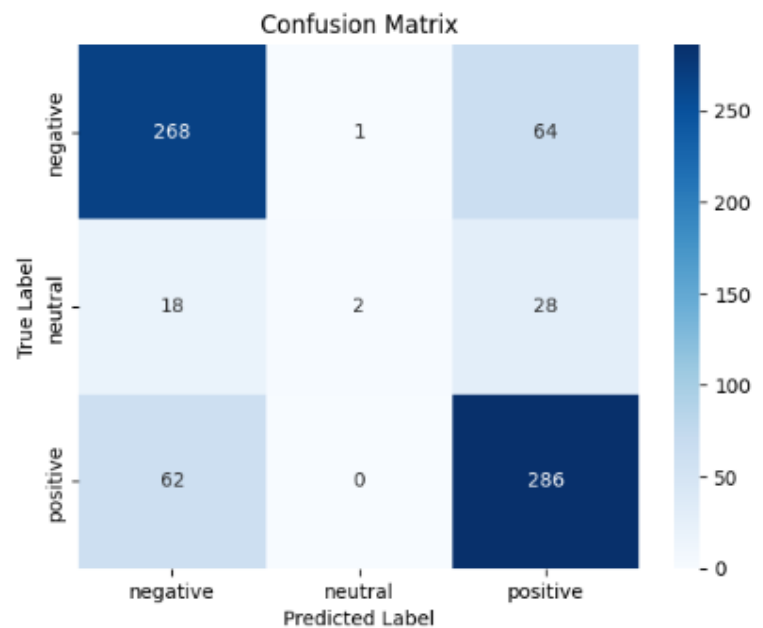
Gambar 4. 5. Hasil Loop Akurasi *Training* dan *Validation* Terbaik *Hyperparameter* Pertama

Pada gambar 4.13 diatas diperoleh akurasi terbaik pada 83.95% *training* dan 72.9% *validation* yang dihasilkan melalui percobaan dengan *hyperparameter* 128 *hidden dim*, 10 *num epoch*, 32 *batch size*, 0.000848578501478349 *learning rate* dan 0.3987110285640226 *dropout rate*. Pada percobaan ini dilakukan juga pencarian akurasi menggunakan *classification report* untuk memperoleh nilai akurasi dan nilai *f1-score* dari percobaan diatas hasil dapat dilihat sebagai berikut:

Classification Report:				
	precision	recall	f1-score	support
negative	0.77	0.80	0.79	333
neutral	0.67	0.04	0.08	48
positive	0.76	0.82	0.79	348
accuracy			0.76	729
macro avg	0.73	0.56	0.55	729
weighted avg	0.76	0.76	0.74	729

Gambar 4. 6. *Classification Report* Percobaan Pertama

Pengujian pada percobaan pertama ini memberikan hasil akurasi *F1-Score* berada di angka 76% dan untuk melihat hasil dari diagram *confusion* matriks pada percobaan ini dapat dilihat sebagai berikut:



Gambar 4. 7. *Confusion Matrix* Percobaan Pertama

Pada tahapan pembuatan model dan pelatihan dataset menggunakan *hyperparameter* dengan percobaan kedua dengan *value* yang dapat dilihat dibawah

<i>Hidden_dim</i>	<i>Num_epoch</i>	<i>Batch_size</i>	<i>Learning_rate</i>	<i>Dropout_rate</i>
128	13	64	0.00024412636786250215	0.32844858973723867

```
Epoch 1/6 - Train Loss: 1.0120 - Train Accuracy: 48.93% - Val Loss: 0.9506 - Val Accuracy: 50.59%
Epoch 2/6 - Train Loss: 0.9181 - Train Accuracy: 53.25% - Val Loss: 0.8914 - Val Accuracy: 56.08%
Epoch 3/6 - Train Loss: 0.8759 - Train Accuracy: 58.24% - Val Loss: 0.8762 - Val Accuracy: 60.52%
Epoch 4/6 - Train Loss: 0.8546 - Train Accuracy: 59.02% - Val Loss: 0.8641 - Val Accuracy: 60.13%
Epoch 5/6 - Train Loss: 0.8345 - Train Accuracy: 61.88% - Val Loss: 0.8421 - Val Accuracy: 62.48%
[I 2023-07-02 23:33:06,075] Trial 2 finished with value: 0.8292885720729828 and parameters: {'hidden_
<ipython-input-4-5e71e952192c>:69: FutureWarning: suggest_loguniform has been deprecated in v3.0.0. T
learning_rate = trial.suggest_loguniform("learning_rate", 1e-5, 1e-3)
<ipython-input-4-5e71e952192c>:70: FutureWarning: suggest_uniform has been deprecated in v3.0.0. This
dropout_rate = trial.suggest_uniform("dropout_rate", 0.3, 0.7)
Some weights of the model checkpoint at indolem/indobert-base-uncased were not used when initializing
- This IS expected if you are initializing BertModel from the checkpoint of a model trained on another
- This IS NOT expected if you are initializing BertModel from the checkpoint of a model that you expect
Epoch 1/11 - Train Loss: 0.8877 - Train Accuracy: 54.88% - Val Loss: 0.8274 - Val Accuracy: 64.05%
Epoch 2/11 - Train Loss: 0.7667 - Train Accuracy: 67.43% - Val Loss: 0.7850 - Val Accuracy: 67.58%
Epoch 3/11 - Train Loss: 0.7157 - Train Accuracy: 69.67% - Val Loss: 0.7779 - Val Accuracy: 66.93%
Epoch 4/11 - Train Loss: 0.6872 - Train Accuracy: 72.65% - Val Loss: 0.7809 - Val Accuracy: 66.80%
Epoch 5/11 - Train Loss: 0.6454 - Train Accuracy: 73.65% - Val Loss: 0.7972 - Val Accuracy: 66.93%
Epoch 6/11 - Train Loss: 0.6229 - Train Accuracy: 73.99% - Val Loss: 0.7606 - Val Accuracy: 70.59%
Epoch 7/11 - Train Loss: 0.5660 - Train Accuracy: 77.86% - Val Loss: 0.7777 - Val Accuracy: 69.15%
Epoch 8/11 - Train Loss: 0.5221 - Train Accuracy: 80.10% - Val Loss: 0.8256 - Val Accuracy: 66.14%
Epoch 9/11 - Train Loss: 0.4653 - Train Accuracy: 81.89% - Val Loss: 0.8117 - Val Accuracy: 70.33%
Epoch 10/11 - Train Loss: 0.4161 - Train Accuracy: 85.54% - Val Loss: 0.8710 - Val Accuracy: 67.45%
[I 2023-07-03 00:43:18,810] Trial 3 finished with value: 0.7606188580393791 and parameters: {'hidden_
Epoch 11/11 - Train Loss: 0.3574 - Train Accuracy: 86.77% - Val Loss: 0.8951 - Val Accuracy: 68.89%
```

Gambar 4. 16. Proses Loop Akurasi *Training* dan *Validation* Terbaik *Hyperparameter* Kedua

Menghasilkan akurasi *training* dan validasi sebagai berikut:

```
Some weights of the model checkpoint at indolem/indobert-base-uncased were not used when initializing
- This IS expected if you are initializing BertModel from the checkpoint of a model trained on another
- This IS NOT expected if you are initializing BertModel from the checkpoint of a model that you expect
Epoch 1/13 - Train Loss: 0.9361 - Train Accuracy: 52.35% - Val Loss: 0.8709 - Val Accuracy: 60.39%
Epoch 2/13 - Train Loss: 0.8287 - Train Accuracy: 64.13% - Val Loss: 0.8306 - Val Accuracy: 63.92%
Epoch 3/13 - Train Loss: 0.7804 - Train Accuracy: 67.26% - Val Loss: 0.7936 - Val Accuracy: 67.32%
Epoch 4/13 - Train Loss: 0.7442 - Train Accuracy: 69.51% - Val Loss: 0.7936 - Val Accuracy: 65.62%
Epoch 5/13 - Train Loss: 0.7088 - Train Accuracy: 71.41% - Val Loss: 0.7589 - Val Accuracy: 68.10%
Epoch 6/13 - Train Loss: 0.6766 - Train Accuracy: 73.43% - Val Loss: 0.7645 - Val Accuracy: 69.80%
Epoch 7/13 - Train Loss: 0.6472 - Train Accuracy: 74.83% - Val Loss: 0.7756 - Val Accuracy: 68.76%
Epoch 8/13 - Train Loss: 0.6035 - Train Accuracy: 76.85% - Val Loss: 0.7770 - Val Accuracy: 69.80%
Epoch 9/13 - Train Loss: 0.5826 - Train Accuracy: 76.96% - Val Loss: 0.8087 - Val Accuracy: 67.97%
Epoch 10/13 - Train Loss: 0.5747 - Train Accuracy: 77.47% - Val Loss: 0.7891 - Val Accuracy: 69.80%
Epoch 11/13 - Train Loss: 0.4976 - Train Accuracy: 80.33% - Val Loss: 0.8373 - Val Accuracy: 68.63%
Epoch 12/13 - Train Loss: 0.5055 - Train Accuracy: 79.82% - Val Loss: 0.7920 - Val Accuracy: 69.67%
Epoch 13/13 - Train Loss: 0.4359 - Train Accuracy: 84.08% - Val Loss: 0.8212 - Val Accuracy: 71.37%
```

Gambar 4. 17. Hasil Akurasi *Training* dan *Validation* *Hyperparameter* Kedua

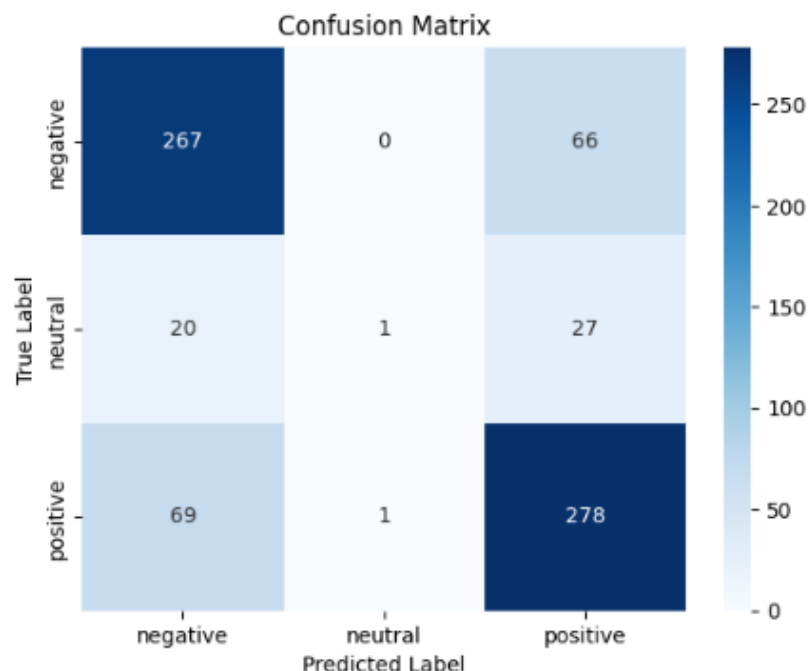
Pada gambar 4.16 diatas diperoleh akurasi terbaik pada 84,08% *training* dan 71,37% *validation* yang dihasilkan melalui percobaan dengan *hyperparameter* 128 *hidden dim*, 13 *num epoch*, 64 *batch size*, 0.00024412636786250215 *learning rate* dan 0.32844858973723867 *dropout rate*.

Pada percobaan ini dilakukan juga pencarian akurasi menggunakan *classification report* untuk memperoleh nilai akurasi dan nilai *f1-score* dari percobaan diatas hasil dapat dilihat sebagai berikut:

Classification Report:				
	precision	recall	f1-score	support
negative	0.75	0.80	0.78	333
neutral	0.50	0.02	0.04	48
positive	0.75	0.80	0.77	348
accuracy			0.75	729
macro avg	0.67	0.54	0.53	729
weighted avg	0.73	0.75	0.73	729

Gambar 4. 18. *Classification Report* Percobaan Kedua

Pengujian pada percobaan kedua ini memberikan hasil akurasi *F1-Score* berada di angka 75% dan untuk melihat hasil dari diagram *confusion* matriks pada percobaan ini dapat dilihat sebagai berikut:



Gambar 4. 19. *Confusion Matrix* Percobaan Kedua

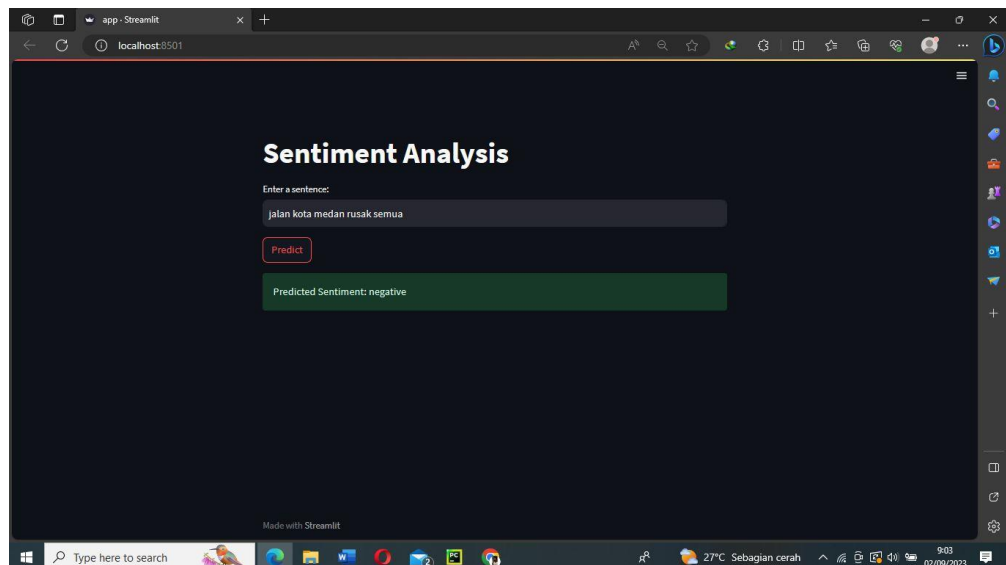
Berdasarkan diagram *confusion matrix* diatas dapat dilihat bahwa sistem dapat memprediksi sentimen dengan cukup merata, namun berdasarkan akurasi dapat diduga

bahwa prediksi yang dilakukan masih memiliki kemungkinan mengalami *false predict*. Hal tersebut diduga terjadi karena ketidak seimbangan dataset pada sentiment positif dan netral yang menyebabkan sistem kurang dapat melakukan prediksi dengan lebih akurat.

4.8. *Prediction Test*

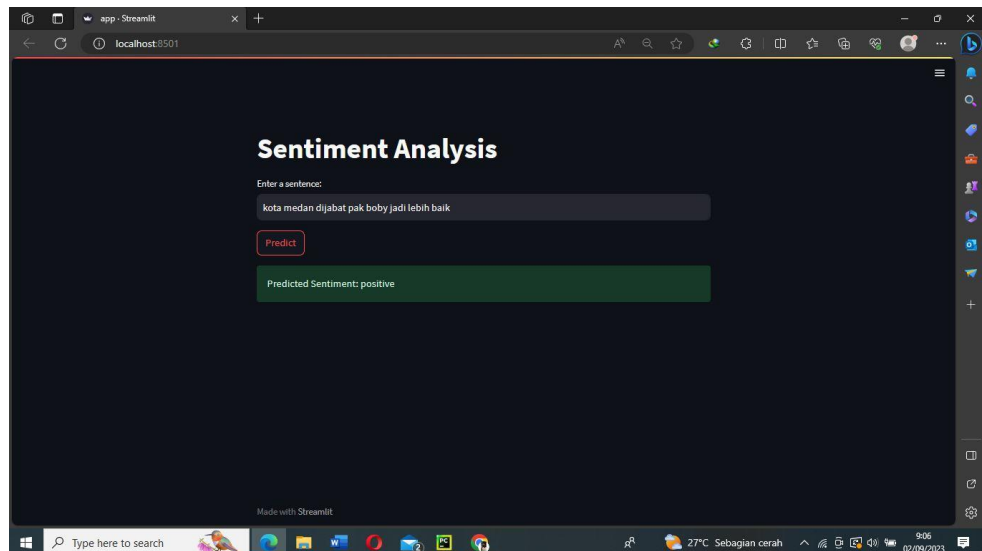
Pada tahapan ini peneliti melakukan *testing* ke beberapa *sentence* yang telah dipersiapkan pada model yang telah dibangun, di proses *prediction* ini peneliti membuat UI agar mempermudah proses *testing* pada model menggunakan *streamlit*

1. Pada proses *prediction* pertama kalimat yang digunakan adalah “Jalan kota medan rusak semua” pada prediksi pertama ini model memprediksi dengan sentimen yang tepat yaitu sentimen *Negative* untuk hasil dapat dilihat pada gambar 4.20.



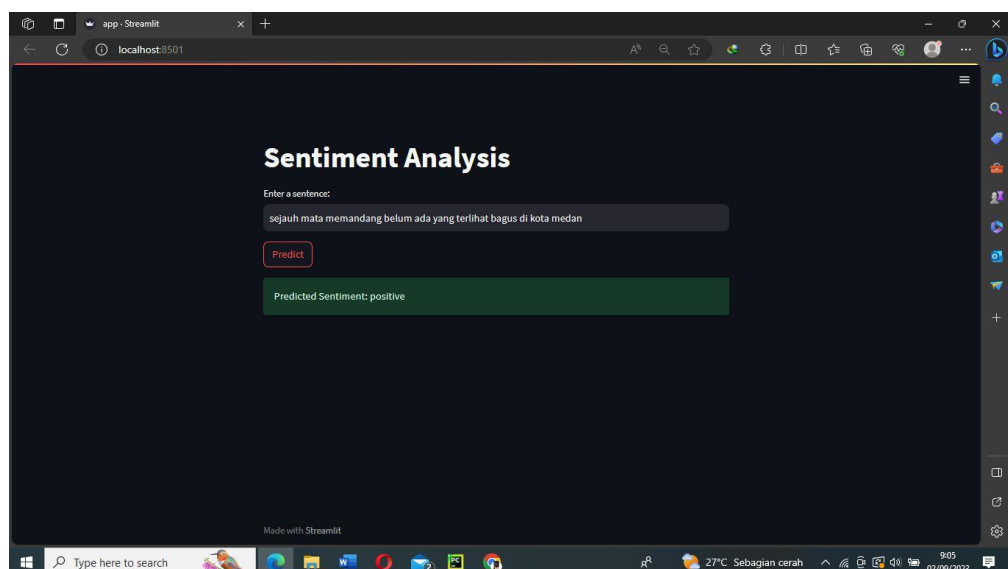
Gambar 4. 20. *Prediction Test Pertama*

2. Peneliti melanjutkan pada proses *prediction* kedua dan kalimat yang digunakan adalah “Kota medan dijabat pak boby jadi lebih baik” pada prediksi kedua ini model memprediksi dengan sentimen yang tepat yaitu sentimen *Positive* untuk hasil dapat dilihat pada gambar 4.21.



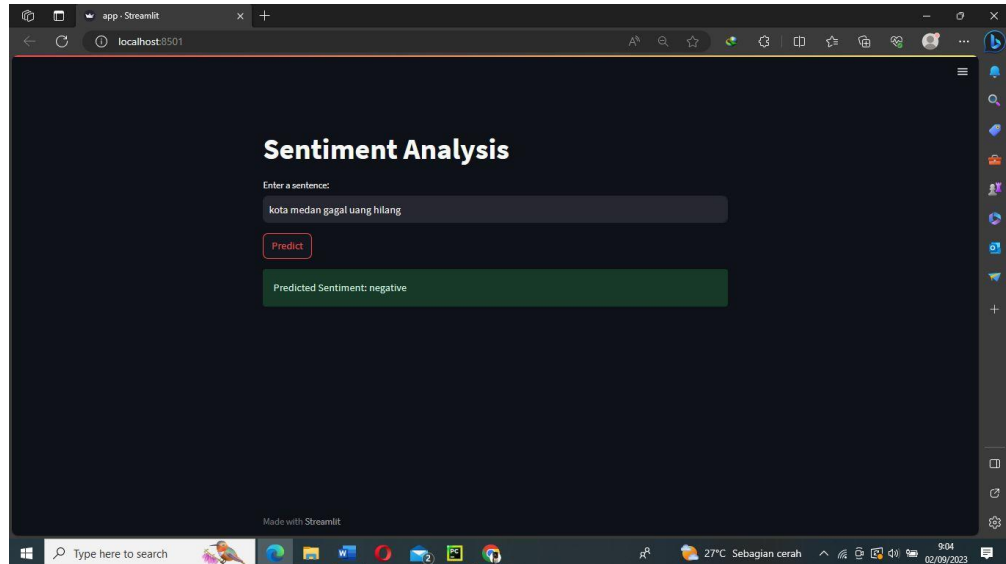
Gambar 4. 21. *Prediction Test Kedua*

3. Peneliti melanjutkan pada proses *prediction* ketiga dan kalimat yang digunakan adalah “Sejauh mata memandang belum ada yang terlihat bagus di kota medan” pada prediksi ketiga ini muncul sebuah perbedaan ternyata saat model diberikan kalimat yang panjang dan memiliki makna tersirat model tidak dapat memprediksi sentimen dengan tepat, hasil menunjukkan sentimen *positive* dimana seharusnya sentimen yang dihasilkan adalah *negative* untuk hasil dapat dilihat pada gambar 4.22.



Gambar 4. 22. *Prediction Test Ketiga*

4. Peneliti melanjutkan pada proses *prediction* keempat dan kalimat yang digunakan adalah “Kota medan gagal uang hilang” pada prediksi keempat ini model memprediksi dengan sentimen yang tepat lagi yaitu sentimen *negative* untuk hasil dapat dilihat pada gambar 4.22.



Gambar 4. 23. *Prediction Test Keempat*

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dan berlandaskan setiap tahapan yang dilalui dari analisis, perancangan, implementasi serta hasil pengujian penulis mengambil kesimpulan:

1. Pada penelitian ini yang bertema analisis sentimen menggunakan algoritma LSTM dan mengkombinasikan dengan algoritma BERT telah dilakukan beberapa kali percobaan dan dua percobaan telah diambil dari yang terbaik untuk didokumentasikan, hasil akurasi yang diperoleh ialah 76 % dan 75% dengan menggunakan *hyperparameter* yang berbeda.
2. Hasil yang diperoleh melalui diagram *confusion matrix* memberikan hasil bahwa model yang dibentuk dapat melakukan prediksi dengan merata melalui baik positif maupun negatif, namun masih terdapat *false predict* pada model yang diakibatkan oleh ketidak seimbangan jumlah dataset negatif dengan dataset positif juga netral, serta proses pelabelan *automatis* yang tidak sempurna mengakibatkan model kesulitan dalam proses pelatihan model.
3. Model mengalami *value loss* yang cukup besar dikarenakan ketidak seimbangan dataset dan kompleksitas model yang berlebih.
4. Dataset sangat berpengaruh dalam menghasilkan model yang baik serta akurasi yang maksimal, semakin baik dataset yang dimiliki semakin baik pula hasil penelitian yang diperoleh.

5.2. Saran

Penulis juga menemukan beberapa saran yang mungkin dapat di kembangkan atau di lakukan pada sistem selanjutnya:

1. Mencoba untuk melakukan metode *labelling* dengan yang lebih optimal agar diperolehnya hasil sentimen yang akurat pada setiap teks yang terdapat di penelitian selanjutnya.

2. Mencoba untuk memperhatikan dataset lebih rinci pada kesalahan *typo* atau penggunaan bahasa berbeda di tengah teks yang dapat menyulitkan pelatihan model.
3. Mencoba untuk melakukan peningkatan pada model untuk memperoleh hasil akurasi model yang lebih baik dengan mencoba beberapa cara seperti meningkatkan jumlah dataset, melakukan *cross-validation* dan melakukan penyeimbangan kelas pada dataset
4. Dapat melakukan pendeteksian kesalahan-kesalahan yang terjadi pada klasifikasi yang terjadi pada algoritma LSTM dan *BERT Embedding*.

DAFTAR PUSTAKA

- Murthy, G., Allu, S. R., Andhavarapu, B., Bagadi, M., & Belusonti, M. (2020). Text based sentiment analysis using LSTM. *International Journal of Engineering Research & Technology*, 9(05).
- Nurrohmat, M. A., & Azhari, S. (2019). Sentiment analysis of novel review using long short term memory method. *Indonesian Journal of Computing and Cybernetics Systems*.
- Kusum, S. P. P. (2022). Sentiment analysis using global vector and long short-term memory. *Indonesian Journal of Electrical Engineering and Computer Science*, 26(1), 414–422. ISSN: 2502-4752.
- Gopalakrishnan, K., & Salem, F. M. (2020). Sentiment analysis using simplified long short-term memory recurrent neural networks. *Cornell University*.
- Andrea, C., Claudia, D., Alex, M., & Domenico, P. (2021). Emotion and Sentiment analysis of tweets using BERT.
- Gupta, C., Chawla, G., Rawlley, K., Bisht, K., & Sharma, M. (2021). *Senti_ALSTM: sentiment analysis of movie reviews using attention-based-LSTM*. 211–219.
- Tripathi, M. (2021). Sentiment analysis of nepali covid19 tweets using nb svm and lstm. *Journal of Artificial Intelligence*, 3(03), 151–168.
- Long, F., Zhou, K., & Ou, W. (2019). Sentiment analysis of text based on bidirectional LSTM with multi-head attention. *IEEE Access*, 7, 141960–141969.
- Muhammad, P. F., Kusumaningrum, R., & Wibowo, A. (2021). Sentiment analysis using Word2vec and long short-term memory (LSTM) for Indonesian hotel reviews. *Procedia Computer Science*, 179, 728–735.
- Ahmed, A., & Yousuf, M. A. (2020). *Sentiment analysis on Bangla text using long short-term memory (LSTM) recurrent neural network*. 181-192.

- Brownlee, J. (2019). *Long short-term memory networks with python: Develop sequence prediction models with deep learning* (v1.5 ed). Tersedia di Google Books.
- Putri, D. D., Nama, G. F., & Sulistiono, W. E. (2022). Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada *Twitter* Menggunakan Metode Naïve Bayes Classifier. *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, 10(1), 34-40.
- Dikiyanti, T. D., Rukmi, A. M., & Irawan, M. I. (2021). Sentiment analysis and topic modeling of BPJS Kesehatan based on twitter crawling data using Indonesian Sentiment Lexicon and Latent Dirichlet Allocation algorithm. *Journal of Physics: Conference Series*.
- Wadhwa, S., & Babber, K. (2021). Performance Comparison of Classifiers on Twitter Sentimental Analysis. *European Journal of Engineering Science and Technology*, 4(3), 15-24. ISSN: 2538-9181.
- Smith, J. (2019). A Comparative Study of Classification Evaluation Metrics for Imbalanced Datasets. *IEEE International Conference on Data Mining*.
- Johnson, E. (2022). Evaluating Classification Models using Confusion Matrix: A Comprehensive Study. *Journal of Machine Learning Research*.
- Agrawal, M., Hegselmann, S., Lang, H., Kim, Y., & Sontag, D. (2022). Large language models are zero-shot clinical information extractors.
- Alonso, J. C., Martin-Lopez, A., Segura, S., Garcia, J. M., & Ruiz-Cortes, A. (2022). ARTE: Automated Generation of Realistic Test Inputs for Web APIs, 49(1), 348–363.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.
- Li, X., Bing, L., Zhang, W., & Lam, W. (2019). Exploiting BERT for end-to-end aspect-based sentiment analysis.