SKRIPSI

ANALISIS SENTIMEN VAKSIN KETIGA (*BOOSTER*) PADA *TWITTER* MENGGUNAKAN ALGORITMA *LONG SHORT THERM MEMORY* (LSTM)

OLEH
Reza Hi. Hukum
07351811024

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KHAIRUN
TERNATE

2024

LEMBAR PENGESAHAN

ANALISIS SENTIMEN VAKSIN KETIGA (*BOOSTER*) PADA *TWITTER* MENGGUNAKAN ALGORITMA *LONG SHORT TERM MEMORY* (LSTM)

Oleh
Reza Hi. Hukum
07351811024

Skripsi ini telah disahkan
Tanggal 17 Juli 2024

Menyetujui
Tim Penguji

Ketua Penguji

Ir. ABDUL MUBARAK, S.Kom., M.T., IPM.
NIP. 198212062014041002

Pembimbing I

SAIFUL DO. ABDULLAH, S.T., M.T.
NIDN. 0018029002

Anggota Penguji

ROSIHAN, S.T., M.Cs.
NIP. 197607192010121001

Pembimbing II

Ir. AMAL KHAIRAN, S.T., M.Eng., IPM.
NIP. 197401112003121003

Anggota Penguji

ACHMAD FUAD, S.T., M.T.
NIP. 197606182005011001

Mengetahui/Menyetujui

Koordinator Program Studi
Informatika

ROSIHAN S.T., M.Cs.
NIP. 197607192010121001

Dekan Fakultas Teknik
Universitas Khairun

Ir. ENDAH HARISUN, S.T., M.T., CRP.
NIP. 197511302005011013

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama	: Reza Hi. Hukum
NPM	: 07351811024
Fakultas	: Teknik
Jurusan/Program Studi	: Informatika
Judul Skripsi	: Analisis Sentimen Vaksin Ketiga (*Booster*) Pada *Twitter* Menggunakan Algoritma *Long Short Therm Memory* (LSTM)

Dengan ini menyatakan bahwa penulisan Skripsi yang telah saya buat ini merupakan hasil karya sendiri dan benar keasliannya. Apabila ternyata di kemudian hari penulisan Skripsi ini merupakan hasil plagiat atau penjiplakan terhadap karya orang lain, maka saya bersedia mempertanggung jawabkan sekaligus bersedia menerima sanksi berdasarkan aturan tata tertib di Universitas Khairun.

Demikian pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan.

Penulis

Reza Hi. Hukum

LEMBAR PERSEMBAHAN

Segala pujian saya ucapkan kepada Allah SWT dengan asmanya yang maha pengasih lagi maha penyayang, yang maha memelihara, yang maha mematikan dan menghidupakan karena atas segala pemberian nikmatnya baik nikmat kesehatan dan umur panjang tetapi nikmat yang paling besar atas semua pemberiannya adalah nikmat hidayah berada diatas agama yang mulia ini yang hanya diterima disisinya yaitu islam dan iman. Shalawat dan taslim saya ucapkan kepada nabi yang paling mulia baginda Muhammad Rasulullah Shallallahu'alaihi wasallam beserta para sahabat, keluarga, *tabi'in* dan *tabi tabi'in* semoga kita pengikutnya mendapat syafaatnya. Skripsi ini saya persembahkan sebagai bukti kasih sayang kepada:

1. Surgaku ibunda Fatmah Marsaoly dan ayahanda Murdi Hi. Hukum dimana telah melahirkanku dan merawatku dengan kasih sayang dari kecil hingga sekarang. Yang telah menginfakkan hartanya dan keringatnya. Yang selalu mendoakan anakanya agar bahagia. Saya tidak bisa membalas kebaikannya tapi hanya doa yang bisa saya panjatkan agar Allah ridho sehingga mempertemukan dijannahnya.
2. Adikku Syakirah Hi. Hukum dan Susanti R Tampilang, S.Kom. yang telah memberikan doa dan semangat dalam menyelesaikan skripsi saya.
3. Ketua Prodi Informatika, Dosen Pembimbing, Dosen Penguji, dan seluruh Dosen Prodi Informatika yang telah memberi ilmu, didikan dan pengalaman yang sangat berarti hingga penulisan bisa sampai di tahap ini, saya mengucapkan bayak terima kasih. Semoga kebaikan selalu menyertai bapak/ibu dosen sekalian.

MOTTO

"Apabila Sesuatu Yang Kau Senangi Tidak Terjadi, Maka Senangilah Apa Yang Terjadi"

(Ali Bin Abi Thalib)

"Tetap Sabar ,Sebesar apapun badainya pasti akan tetap berlalu"

(Reza Hi. Hukum)

KATA PENGANTAR

Puji dan syukur penulis persembahkan kehadirat Allah *Subhanahu Wata'ala*, karena berkat rahmat dan karunia-Nya semata sehingga penulis mampu menyelesaikan penyusunan laporan skripsi ini dengan judul "Analissis Sentimen Vaksin Ketiga (*Booster*) Pada *Twitter* Menggunakan Algoritma *Long Short Therm Memory* (LSTM)" Penyusunan laporan Skripsi ini adalah untuk memenuhi salah satu persyaratan kelulusan pada Universitas Khairun Ternate Fakultas Teknik Program Studi Informatika. Penyusunan laporan ini dapat terlaksana dengan baik berkat dukungan dari banyak pihak, untuk itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Muhammad Ridha Ajam M., Hum., selaku Rektor Universitas Khairun Ternate.
2. Bapak Ir., Endah Harisun, S.T., M.T., CRP., selaku Dekan Fakultas Universitas Khairun Ternate.
3. Bapak Rosihan, S.T., M.Cs., selaku Koordinator Program Studi Informatika Dan Penguji III yang telah bersedia meluangkan waktu untuk menguji dan memberikan masukan perbaikan demi menyempurnakan laporan skripsi ini.
4. Bapak Saiful Do Abdullah, S.T, M.T., sebagai dosen Pembimbing I, yang telah meluangkan waktu untuk membimbing, saran, arahan yang sangat berarti serta melayani dan mengarahkan kami dalam proses ujian serta pengumpulan laporan.
5. Bapak Ir. Amal Khairan, S.T., M.Eng., IPM., selaku Dosen Pembimbing II, yang telah bersedia meluangkan waktu untuk membimbing saran, serta arahan yang sangat berarti, hingga terselesaikannya laporan skripsi ini.
6. Bapak Ir. Abdul Mubarak, S.Kom., M.T., IPM., Selaku Penguji I, yang telah meluangkan waktu menguji penulis dan memberikan saran serta arahan yang sangat berarti sehingga terselesaikannya penulisan skripsi ini.
7. Bapak Achmad Fuad, S.T., M.T., selaku Penguji II, yang telah bersedia meluangkan waktu untuk menguji dan memberikan masukan perbaikan demi menyempurnakan laporan skripsi ini.
8. Civitas akademika Fakultas Universitas Khairun yang telah membantu dalam pelaksanaan pembuatan laporan skripsi ini.

9. Orang tua, kakak, adik, serta seluruh keluarga yang selalu memberikan dukungan dan perhatian luar biasa yang penuh hingga penulis dapat menyelesaikan laporan skripsi ini.
10. Teman-teman angkatan 2018 yang telah berpartisipasi baik saran maupun kritik atas laporan skripsi ini dan juga kepada Susanti R Tampilang, S.Kom. yang sangat membantu mau di keadaan susah maupun senang.
11. Semua pihak yang tidak bisa penulis sebutkan satu per satu yang telah membantu penulis baik langsung maupun tidak langsung dalam menyelesaikan laporan skripsi ini.

Akhir kata penulis ucapkan rasa terima kasih kepada semua pihak dan apabila ada yang tidak tersebutkan namanya penulis mohon maaf, dengan besar harapan semoga skripsi ini dapat bermanfaat bagi penulis sendiri dan umumnya bagi pembaca. Dan kepada semua pihak yang telah membantu dalam penulisan ini semoga amal dan kebaikannya dapat balasan yang berlimpah dari *Allah Subhanahu Wata'ala*.

.

Ternate,17 Juli 2024

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

ABSTRAK

ANALISIS SENTIMEN VAKSIN KETIGA (*BOOSTER*) PADA *TWITTER* MENGGUNAKAN ALGORITMA *LONG SHORT THERM MEMORY* (LSTM)

Reza Hi. Hukum[1], Saiful Do Abdullah [2], Amal Khairan[3]
Program Studi Teknik Informatika Universitas Khairun
Jl. Jati Metro, Kota Ternate Selatan
E-mail: rezahukum123@gmail.com[1], Saiful.abdullah@unkhair.ac.id [2], amalkhairan@unkhair.ac.id[3]

Penelitian ini dilakukan untuk mengklasifikasi sentimen masyarakat di *twitter* tentang vaksinasi 3 (*Booster*) berdasarkan kategori positif dan negatif dengan menerapkan algoritma LSTM. Model dari analisis sentimen masyarakat terhadap kebijakan vaksin ke 3 (*Booster*) oleh pemerintah ternyata tidak di respon dengan baik oleh masyarakat, terbukti bahwa dari banyaknya *tweet* negatif yang dihasilkan dibandingkan *tweet* positif tentang kebijakan vaksin ke 3 (*Booster*) dari pemerintah. Model yang dihasilkan dari penelitian ini juga dapat memprediksi dengan baik tentang *tweet* masyarakat sesuai dengan tipenya yaitu sentimen positif terhadap kebijakan vaksin ke 3 (*Booster*) dan sentimen negatif terhadap kebijakan vaksin ke 3 (*Booster*). Dalam penerapan algoritma *long short therm memory* untuk mengetahui berapa banyak klasifikasi sentimen positif dan negatif sentimen vaksin ke 3 (*Booster*) pada *twitter*.Hasil Analisis menggunakan dataset komentar sebanyak 500 sampel data yang terbagi menjadi 2 bagian yaitu 252 sentimen negatif dan 248 sentimen positif, dengan perbandingan 20% data *testing* : 80% data *training*, setelah mendapatkan hasil akurasi pada data *training* didapatkan hasil, nilai *Recall* yang dihasilkan 81%, nilai Presisi sebesar 86% dan nilai *F1-Score* sebesar 84%. Dengan demikian dapat disimpulkan bahwa metode *Long short-term memory* (LSTM) layak digunakan untuk analisis sentimen vaksin ke 3 (*Booster*) pada twitter.

Kata Kunci: analisis sentimen, vaksin *booster,long short therm memory, twitter.*

BAB I

PENDAHULUAN

1.1. Latar Belakang

Coronavirus disease (*COVID-19*) telah menjadi pandemi secara global. Virus ini mulai mewabah di wilayah Wuhan, Cina, pada akhir tahun 2019. Kemudian mulai menyebar ke wilayah Hubei. *Covid-19* kemudian menyebar ke berbagai wilayah di Asia, Amerika, Eropa, Australia, dan Afrika. Sampai dengan awal Maret 2021, kasus terkonfirmasi *Covid-19* telah mencapai 113 juta jiwa di seluruh dunia. Pada akhirnya, *lockdown* menjadi upaya yang diterapkan untuk meminimalisir penyebaran virus tersebut (Rai, 2021).

Tanggal 16 Desember 2020, Presiden Joko Widodo telah mengumumkan untuk memberikan vaksin *Covid-19* secara gratis kepada seluruh warga negara Indonesia (John, 2021). Berbagai reaksi pun muncul dari masyarakat terkait program vaksin ini. Terdapat pihak yang mendukung dan ada pula pihak yang menolak vaksin. Kontroversi efektivitas vaksin *Covid* untuk mencegah penularan penyakit ini ramai diperbincangkan di dunia maya seperti *Twitter*, *Facebook*, dan lain-lain. Pihak yang tidak setuju dan menolak vaksin beralasan karena kandungannya belum teruji dan efikasinya masih rendah.

Kejadian tersebut menimbulkan keresahan di kalangan masyarakat. Respon kekhawatiran tersebut diekspresikan ke dalam media sosial, mayoritas dari masyarakat memberikan respon dan opini terhadap kekhawatiran terhadap vaksinasi melalui media sosial, salah satu media sosial yang digunakan sebagai pilihan untuk menyampaikan respon dan opini tersebut adalah *Twitter*, (Nurhazizah, 2022). Keresahan masyarakat tidak hanya sampai disitu saja karena belum selesai wajib vaksin 1 dan 2 sudah muncul lagi vaksin 3 atau *booster.* Vaksinasi *booster* merupakan vaksinasi ulang atau vaksinasi penguat setelah

selang beberapa waktu,biasanya 1-2 minggu setelah vaksinasi pertama atau tergantung kebutuhan,dengan cara yang sama atau cara vaksin lain yang bertujuan meningkatkan efikasi vaksin (Mulia, 2006).

Twitter merupakan media sosial yang bisa digunakan untuk menyebarkan berita, mendiskusikan ide, dan peristiwa yang sedang terjadi di dunia (John, 2021). Lewat fitur *thread* dan *trending*, *twitter* cocok untuk dijadikan sebagai tempat untuk curhat, bercerita, berdiskusi dan menyuarakan sebuah opini terhadap suatu pembicaraan atau topik. Berdasarkan hal tersebut diperlukan analisis sentimen terhadap opini publik yang yang beredar di *twitter* supaya dapat dikategorikan ke dalam opini yang bersifat positif atau negatif.

Analisis sentimen atau *opinion* mining adalah studi komputasional dari opini-opini orang, sentimen, dan emosi melalui entitas dan atribut yang dimiliki yang diekspresikan dalam bentuk teks (Rahman, 2021).

Algoritma *Long Short-Term Memory* (LSTM) mampu menyimpan informasi untuk jangka waktu yang lama. Hal ini kemudian dapat digunakan untuk memproses, memprediksi, dan mengklasifikasikan informasi berdasarkan data deret waktu. Sehingga cocok dengan objek yang diangkat yaitu mengkalifikasi data komentar masyarakat di *twitter* menggunakan LSTM. Algoritma *Long Short-Term Memory* (LSTM) merupakan salah satu metode dalam *Deep Learning* yang dapat digunakan untuk *Natural Language Processing* (NLP) seperti pengenalan suara, translasi teks, dan juga analisis sentimen. LSTM merupakan pengembangan dari metode *Recurrent Neural Network* (RNN), metode LSTM ini dibuat untuk menyelesaikan permasalahan *vanishing gradient* yang ada pada RNN. Penelitian (Rahman, 2021).

Dari latar belakang di atas penulis ingin melakukan penelelitian dengan mengangkat judul "Analissis Sentimen Vaksin Ketiga (*Booster*) Pada *Twitter* Menggunakan Algoritma *Long Short Therm Memory (LSTM)*".

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, maka yang menjadi rumusan masalah pada penelitian ini adalah bagaimana hasil klasifikasi sentimen masyarakat di *twitter* tentang vaksinasi 3 (*Booster)* berdasarkan positif dan negatif dengan menerapkan algoritma LSTM.

1.3. Batasan Masalah

Setelah melihat latar belakang masalah yang telah di uraikan di atas maka Batasan masalah dalam penelitian ini yaitu:

1. Data yang digunakan hanya data Komentar *twitte*r yang di ambil dari API *twitter* menggunakan *tools rapidminer* dengan hastag #vaksin 3 dan #vaksin *booster*.
2. Pengolahan data menggunakan *Python.*
3. Menggunakan metode *Long short term memory* (LSTM).

1.4. Tujuan Penelitian

Penelitian bertujuan untuk mengklasifikasi sentimen masyarakat di *twitter* tentang vaksinasi 3 (*Booster)* berdasarkan kategori positif dan negatif dengan menerapkan algoritma LSTM.

1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1. Mengetahui opini publik atau sentimen masyarakat terhadap vaksin ke 3 (*booster*) melalui media sosial *Twitter*.
2. Dapat mengimplementasikan algoritma *long short term memory* (LSTM) dalam bidang

Natural language network (NLP).

1.6. Sistematika Penulisan

Untuk memudahkan pembahasan dalam Skripsi ini, sistematika penulisan dibagi menjdai 5 (lima) bab yang dijelaskan sebagai berikut

BAB I PENDAHULUAN

Terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Memaparkan teori-teori yang didapat dari sumber-sumber relevan untuk digunakan sebagai panduan dalam penelitian serta penyusunan skripsi ini.

BAB III METODE PENELITIAN

Bab ini membahas tentang metode penelitian yang telah dilakukan oleh penulis dengan permasalahan yang diangkat.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan hasil dan pembahasan metode *algoritma long short therm memory* dalam menanalisis komentar sentimen vaksin *booster.*

BAB V PENUTUP

Pada bab ini menjelaskan tentang kesimpulan dan saran.

BAB II

KAJIAN PUSTAKA

2.1. Penelitian Terkait

Penelitian ini bukan dilakukan pertama kali namun sudah di lakukan beberapa penelitian yang terkait dengan analisis sentimen masyarakat di aplikasi *twitter* Pada bagian ini akan di paparkan beberapa penelitian sejenis yang telah dilakukan serta penelitian yang dilakukan oleh penulis.

Tabel 2.1. Penelitian Terkait.

No	Peneliti	Judul penelitian dan tahun	Metode	Hasil
1.	Sri Lestari (2021).	Analisis Sentimen Vaksin *Sinovac* Pada *Twitter* Menggunakan Algoritma Naive Bayes.	*Naive Bayes*	Hasil dari penelitian ini menunjukan bahwa *tweets* dengan sentimen positif sebanyak 86%, sedangkan *tweets* dengan sentimen negatif sebnayak 14%. Dengan nilai akurasi dari perhitungan menggunakan algoritma *Naïve Bayes* adalah 92,96%.
2.	Adi yahyadi Fitri latifha (2022).	Analisis Sentimen *Twitter* Terhadap Kebijakan Ppkm Di Tengah Pandemi *Covid-19* Menggunakan Mode Lstm.	Metode LSTM	Hasil pengujian analisis sentimen *tweet* dengaan menggunakan molde LSTM terhadap penerapan PKKM oleh pemerintah Indonesia didapatkan hasil 36,5 % merespos positif, 15,7%

				merespon netral dan 57.8% merespons negatif, dengan hasil yang diperoleh dapat disimpulkan bahwa LSTM dari algoritma RNN memiliki tingkat keefektifan yang cukup bagus dalam penelitian ini dengan tingkat akurasi sebesar 70% serta memiliki kriteria dengan nilai *loss* yang rendah.
3.	Nabil Ramdhani Rifky Haekal Al-Fadillah (2021)	Analisis Sentimen Pengguna Twitter Terhadap Belajar Daring Selama Pandemi *Covid-19 Dengan Deep Learning.*	Dengan *Deep Learning*,Lstm	Hasil percobaan pada penelitian menunjukan bahwa metode terbaik pada data *tweet* adalah metode *Deep Learning* yaitu dengan akurasi sebesar 100%, ketika dibandingkan dengan metode *Naïve Bayes* yang memiliki akurasi sebesar 99%, dan k-NN yang memiliki akurasi sebesar 82%. Diketahui juga bahwa sebanyak 73% komentar positif, 14% komentar negatif, dan 13% komentar netral.
4.	Rima Tamara Aldisa , Pandu Maulana(2022)	Analisis Sentimen Opini Masyarakat Terhadap Vaksinasi *Booster COVID*	Metode *Naive Bayes, Decision Tree* dan SVM	Hasil pada penelitian ini menunjukkan skor AUC terbesar jatuh kepada model SVM (75.40%), namun

		19 Dengan Perbandingan Metode Naive Bayes, Decision Tree dan SVM.		untuk presisi yang lebih akurat jatuh kepada model Naive Bayes (83.81%). Selain itu, terdapat confusion matrix yang menunjukkan bahwa uji coba model Naive Bayes yang dilakukan berjalan dengan baik.
5.	Frizka Fitriana , Ema Utami , Hanif Al Fatta (2021).	Analisis Sentimen Opini Terhadap Vaksin Covid-19 pada Media Sosial *Twitter* Menggunakan *Support Vector Machine* dan *Naive Bayes.*	*Support Vector Machine* dan *Naive Bayes*	hasil penelitian diperoleh bahwa algoritma SVM memiliki performa lebih baik pada bagian akurasi, presisi dan recall dengan nilai 90,47%, 90,23%, 90,78% dan performa pada algoritma *Naïve Bayes* adalah 88,64%, 87,32%, 88,13%, dengan selisih akurasi 1,83%, presisi 2.91% dan recall 2.65%. Sedangkan untuk waktu algoritma *Naïve Bayes* memiliki tingkat performa lebih baik dengan nilai 8,1 detik dibandingkan SVM dengan kecepatan waktu 11 detik. Hasil sentimen analisis netral diperoleh 8,76%, negatif 42,92% dan positif 48,32% untuk *Naïve Bayes* dan netral

				10,56%, negatif 41,28% dan positif 48,16% untuk SVM.

2.2. Analisis Sentimen

Analisis sentimen adalah suatu proses untuk menemukan suatu makna dari perilaku, opini, pandangan, dan emosi dari suatu teks, perkataan, *tweets*, dan *database* dengan sumber dari *Natural Language Processing* (NLP). Sentimen analisis mengklasifikasi teks menjadi bermakna positif dan negatif (John, 2021).

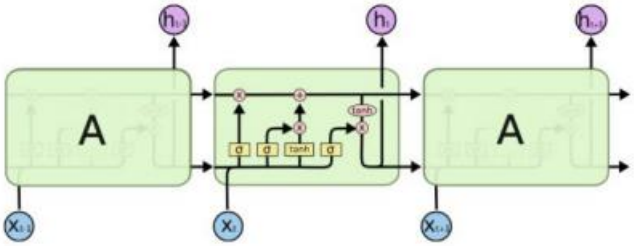
2.3. *Natural language processing* (NLP)

Natural Processing Language (NLP) adalah cabang dari *Artificial Intelligence* yang berhubungan dengan interaksi antara mesin dan manusia menggunakan bahasa natural. Dalam hal ini memanfaatkan salah satu *library python* yaitu *TextBlob* untuk mengerjakan algoritma *text mining*. Salah satu metode dari *text mining* di bidang NLP adalah dengan pendekatan analisis sentimen. Pendekatan analisis sentimen digunakan untuk menganalisa data berbentuk opini publik sebagai pendukung pengambilan keputusan. Tugas analisis sentimen adalah mengelompokkan kumpulan polaritas dari teks yang berada dalam dokumentasi, kalimat atau fitur entitas dengan tingkat aspek yang bersifat positif, netral atau negatif (Buntoro, 2017).

2.4. *Long Short Term Memory* (LSTM*)*

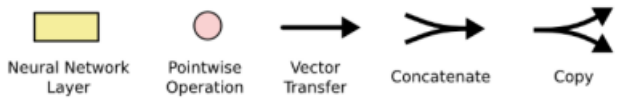
Long Short Term Memory (LSTM) disebutkan untuk pertama kali pada tahun 1997 dijelaskan oleh *Hochreiter* dan *Schmidhuber*. LSTM disebut juga sebagai jaringan saraf dengan arsitektur yang mudah beradaptasi, sehingga bentuknya dapat disesuaikan, tergantung pada aplikasinya. *Long Short Term Memory* merupakan turunan dari metode *Recurrent Neural Network* (RNN). *Recurrent Neural Network* merupakan jaringan saraf

berulang yang didesain khusus untuk *menghandle* data berurutan (*sequence data*). Namun RNN mempunyai masalah *vanishing* dan *exploding gradient* yaitu apabila terjadi perubahan pada jangkauan nilai dari satu lapisan menuju lapisan berikutnya pada sebuah arsitektur. LSTM dibangun dan dirancang untuk mengatasi masalah gradien menghilang dari RNN ketika berhadapan dengan *vanishing* dan *exploding gradient* tersebut. Arsitektur LSTM terdiri dari lapisan input, lapisan *output*, dan lapisan tersembunyi (Aldi, 2018) yang disajikan pada gambar 2.1.



Gambar 2.1. Arsitektur LSTM.

Untuk melihat notasi yang digunakan pada arsitektur LSTM, dapat dilihat pada gambar 2.1. Pada diagram tersebut setiap garis mewakili aliran pengiriman seluruh vektor dari output satu node ke input node lain. Lingkaran merah muda mewakili operasi titik seperti penjumlahan vektor. Kotak kuning adalah lapisan jaringan saraf yang terlatih. Garis yang menyatu berarti penggabungan, dan garis yang terbelah berarti garpu yang menyalin informasi dan mengirimkannya ke sisi lain yang dapat dilihat pada gambar 2.2.



Gambar 2.2. Notasi Dalam LSTM.

Berikut dijelaskan terkait gerbang-gerbang yang terdapat dalam sel memori LSTM:

1. *Forget gate*

Forget gate berfungsi untuk mengolah informasi setiap data inputan dan memilih data mana saja yang perlu disimpan atau dibuang pada *memory cells*. Pada *forget gate*, fungsi aktivasi yang digunakan ialah fungsi aktivasi *sigmoid* (σ). Hasil keluaran yang dihasilkan dari fungsi aktivasi *sigmoid* ialah antara 0 dan 1. Jika hasil keluarannya bernilai 1 maka semua data akan disimpan dan sebaliknya jika keluarannya 0 maka semua data akan dibuang. Rumus dari ft dapat dilihat pada persamaan 2.1.

$$ft=\sigma\ (Uf.Xt+Wfht\text{-}1+bf) \quad \text{.....(2.1)}$$

Dimana:

ft = *Forgate gate*.

σ = Fungsi *sigmoid*.

Uf = Bobot input pada *forgate gate.*

Wf = Nilai bobot untuk *forgate gate*.

$ht-1$ = Nilai keluaran sebelum orde ke t.

bf = Nilai input sebelum orde ke t.

2. Input gate

Memiliki dua *gates* yang akan dilaksanakan, pertama akan diputuskan nilai mana yang akan diperbarui menggunakan fungsi aktivasi sigmoid, kemudian menggunakan fungsi aktivasi *tanh* untuk membuat vektor nilai baru yang akan disimpan pada *memory cell* Rumus dari *input gate* dapat dilihat pada persamaan 2.2 dan 2.3.

$$it = \sigma\ (Ui\times Xt+Wiht-+bi) \quad \text{.....(2.2)}$$

$$\bar{C}t = tanh(Wc\ .\ ht-1,\ xt + bc) \quad \text{.....(2.3)}$$

Dimana:

it = *Input gate*.

σ = Fungsi *sigmoid.*

$\bar{C}$ = *Cell* aktivasi

$tanh$ = Fungsi *tanh*.

xt = Nilai *input* sebelum orde ke t

$ht{-}1$ = Nilai keluaran sebelum orde ke t.

Ui = Bobot *input* pada *input gate*.

Uc = Bobot *input* pada *cell* aktivasi.

Wi = Nilai bobot untuk *input gate.*

Wc = Nilai bobot untuk *cell state*. .

bi = Nilai bias pada *input gate*.

bc = Nilai bias pada *cell state*.

3. Cell gate

Pada *cell gate* nilai yang ada pada *memory cell* sebelummnya akan diganti menggunakan nilai *memory cell* yang baru. Nilai ini didapatkan dengan cara menggabungkan nilai yang ada pada *forget gate* dan *input gate.* Rumus dari ct dapat dilihat pada persamaan 2.4.

$$Ct = (ft * Ct{-}1 + it * \bar{C}) \quad (2.4)$$

Dimana:

Ct = *Cell state*.

fi = *Forgate gate.*

$Ct{-}1$ = *Ouput hidden state* sebelumnya atau *state* pada waktu t-1.

it = Nilai *input gate*.

$\bar{C}$ = Cell aktivasi.

4. Output gate

Memiliki dua *gates* yang akan dilaksanakan, pertama ditentukan nilai pada bagian *memory cell* mana yang akan dikeluarkan menggunakan fungsi aktivasi *sigmoid* kemudian nilai ditempatkan pada *memory cell* dengan menggunakan fungsi aktivasi *tanh*. Tahapan akhir, kedua *gate* ini dikalikan sehingga menghasilkan nilai yang akan dikeluarkan (Aldi, 2018). Rumus dari ot dapat dilihat pada persamaan 2.5 dan 2.6.

$$ot = (\mathrm{U}o\ xt + Wo\ ht{-}1 + bo) \quad (2.5)$$

$$ht = ot * tanh\ (Ct) \quad (2.6)$$

Dimana:

ot = *Output gate*.

Ct = *Cell gate.*

σ = Fungsi *sigmoid*.

Tanh = Fungsi *tanh.*

Uo = Bobot *input* pada *output gate*.

xt = Nilai *input* sebelum orde ke t.

Wo = Nilai bobot untuk *output gate.*

$ht{-}1$ = Nilai keluaran sebelum orde ke t.

bo = Nilai bias pada *output gate*.

2.5. Fungsi Aktivasi

Fungsi aktivasi pada *neural network* merupakan persamaan matematika yang menentukan sebuah *output* dari *neural network.* Fungsi aktivasi tersebut akan diimplementasikan pada setiap *neuron* dalam sebuah jaringan, dan dapat menentukan apakah *neuron* tersebut harus diaktifkan atau tidak berdasarkan relevansi setiap *input neuron* untuk memprediksi sebuah model.

1. *Sigmoid*

Fungsi aktivasi *sigmoid biner* memiliki nilai output pada range nol sampai satu [0,1] dan didefinisikan, dapat dilihat pada persamaan 2.7

$$Sigmoid = \vdash (1 + 1/ \ 〖1 + e〗 \ ^(-xi) \ \dashv) \quad \text{.....(2.7)}$$

Dimana:

xi = Nilai *input* ke i.

e = Konstanta *euler.*

2. *Tanh*

Fungsi *Hyperbolic Tangent (tanh) Output* dari fungsi ini memiliki *range* antara satu sampai minus satu [-1,1] dan didefinisikan dengan persamaan 2.8:

$$tanh \ (Xi) = \vdash ((e\text{^}2xi - 1)/(e\text{^}2xi + 1) \ \dashv) \quad \text{.....(2.8)}$$

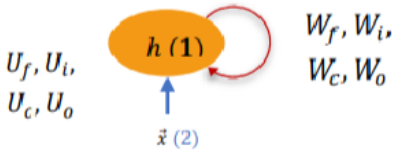
Dimana:

xi = Nilai *input* ke i.

e = Konstanta *euler.*

2.6. Contoh Perhitungan LSTM

Berikut merupakan contoh perhitungan pada jaringan LSTM yang dapat dilihat pada gambar 2.3.



Gambar 2.3. Gambar Jaringan LSTM.

Pada gambar 2.3. menunjukkan pada jaringan LSTM tersebut memiliki 2 dimensi input (xɦ(2)) dengan 1 unit sel LSTM h(1). Dalam contoh perhitungan ini, kita telah memiliki data input yang akan kita proses dengan menggunakan rumus-rumus LSTM yang telah

dijelaskan pada poin sebelumnnya. Data input dapat dilihat pada tabel 2.2. Dimana nilai xt berdimensi 1x2 berupa A1 dan A2 serta nilai yt yang merupakan target.

Tabel 2.2. Input Data.

A1	A2	Target
1	2	0.5
0.5	3	1

Diasumsikan kita sudah memiliki model yang siap untuk digunakan berupa kumpulan matriks bobot U, W, dan b. Setiap matriks U memiliki ukuran 1x2 yang menyatakan jumlah unit LSTM dikali dengan dimensi inputnya. Matriks W dan b masing-masing memiliki ukuran 1x1 serta terdapat nilai$ht-1$ dan $ct-1$. Untuk bobot tiap matriks serta nilai $ht-1$ dan $ct-1$ dapat dilihat pada tabel 2.3.

Tabel 2.3. Bobot Tiap Matriks Dan Nilai $ht-1$ Serta $ct-1$.

Bobot			
U_f		W_f	b_f
0.700	0.450	0.100	0.150
U_i		W_i	b_i
0.950	0.800	0.800	0.650
U_c		W_c	b_c
0.450	0.250	0.150	0.200
U_o		W_o	b_o
0.600	0.400	0.250	0.100
$H\_(t-1)$		$C\_(t-1)$	

Selanjutnya akan melakukan perhitungan untuk menentukan nilai ht dan ct untuk *timestep* t1.

$$"\boldsymbol{t1}" = < 1/2 "\mathbf{0.5}" > \qquad "\boldsymbol{h}" \_"\boldsymbol{t-1}"$$
$$= 0 \qquad "\boldsymbol{C}" \_"\boldsymbol{t-1 = 0}"$$

Perhitungan ht dan *timestep* t1:

Langkah pertama yaitu menghitung nilai *Forget Gate* (ft).

$ft = \sigma\,(UfXt + Wfht-1 + bf)$

$= \sigma\,(1.600 + (0.100 \times 0) + 0.150)$

$= \sigma$ (1.750)

$= 1/\ 〖1 + e〗\ ^(-1.750)\ = 0.852.$

Selanjutnya menghitung nilai *Input Gate* (it) dan nilai Ct.

it = ($UiXt$ + $Wiht$−1 + bi)

= (2.550 + (0.800 × 0) + 0.650)

= (3.200)

$= 1/\ 〖1 + e〗\ ^(-3.200)\ = 0.961.$

Ct= $tanh$ ($UcXt$ + $Wcht$−1 + bc)

= $tanh$ (0.950 + (0.150 × 0) + 0.200)

= $tanh$ (1.150) = ($e^{2}x$1.150 − 1)/($e^{2}x$1.150 + 1)= 0.818.

Untuk tahap selanjutnya kita perlu menghitung nilai Output Gate (ot).

ot = ($UoXt$ + $Woht$−1 + bo)

= (1.400 + (0.250 × 0) + 0.100)

= (1.500)

$= 1/\ 〖1 + e〗\ ^(-1.500)\ = 0.818.$

Setelah mendapatkan nilai, , ot dan Ct , kemudian hitung nilai cell state (ct) dan nilai ht.

ct = ft × ct−1 + it × Ct

= (0.852 × 0) + (0.961 × 0.818)

= 0.786

Menghitung nilai ht.

ht = $ot * tanh$ (ct)

= 0.818 × *tanh* (0.786)

= 0.536.

Perhitungan ht dan: *timestep* t2 Perhitungan dilanjutkan ke *timestep* t2 menggunakan nilai ht−1 dan ct−1 dari *timestep* sebelumnya yaitu *timestep* t1 dan lakukan perhitungan untuk mencari nilai ft, it, dan ot lalu hitung nilai ct dan ht.

$t2$ = < 0.5 3,1> $ht-1$ = 0.536 $ct-1$ = 0.786

Menghitung nilai Forget Gate (ft):

= ($UfXt + Wfht-1 + bf$)

= (1.700 + (0.100 × 0.536) + 0.150)

= (1.904)

=$1/\ 〚1 + e〛\ ^(-1.904)\ = 0.870.$

Menghitung nilai Input Gate (it) dan nilai Ct.

it = ($UiXt + Wiht-1 + bi$)

= (2.875 + (0.800 × 0.536) + 0.650)

= (3.954)

= $1/\ 〚1 + e〛\ ^(-3.954)\ = 0.981$

= $tanh$ ($UcXt + Wcht-1 + bc$)

= $tanh$ (0.975 + (0.150 × 0.536) + 0.200)

= $tanh\ h$ (1.255) =$(e^2x1.255 - 1)/(e^2x1.255 + 1) = 0.850$

Menghitung nilai Output Gate (ot).

ot = ($UoXt + Woht-1 + bi$)

= (1.500 + (0.250 × 0.536) + 0.100)

= (1.734)

= $1/\ 〚1 + e〛\ ^(-1.734)\ = 0.850.$

Setelah mendapatkan nilai ft, ot dan Ct, kemudian hitung nilai cell state (ct) dan nilai ht.

= ft × $ct-1 + it$ × $C\tilde{t}$

= (0.870 × 0.786) + (0.961 × 0.850)

= 1.518.

Menghitung nilai ht.

$ht = ot * tanh\ (ct)$

= 0.850 × $tanh$ (1.518) = 0.772.

2.7. *Confusion matrix*

Confusion Matrix menjadi acuan untuk merepresentasikan prediksi dan kondisi sebenarnya (aktual) dari data yang dihasilkan oleh *machine learning*, *Confusion Matrix* dapat memberikan informasi perbandingan hasil klasifikasi seperti menghitung nilai *accuracy, precision, recall*, dan *F-Measure* (Firmansyah, 2020).

1. *Accuracy* merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Dengan kata lain, *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya). Dapat dilihat pada persamaan 2.10.

   Akurasi = (TP + TN) / (TP + TN + FP + FN)) * 100% ……..(2.10)

2. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Dalam *categorical classification*, precision dapat dibuat sama dengan nilai prediksi positif. Dapat dilihat pada persamaan 2.11.

   Precision = (TP / (TP + FP)) * 100% ……..(2.11)

3. *Recall* adalah data penghapusan yang berhasil diambil dari data yang relevan dengan *query* atau tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Dapat dilihat pada persamaan 2.12.

   Recall = (TP / (TP + FN)) * 100% ……..(2.12)

4. *F-Measure* merupakan perbandingan rata-rata presisi dan *recall* yang dibobotkan. Nilai *recall* dan *precission* pada suatu keadaan dapat memiliki bobot yang berbeda. Ukuran yang menampilkan timbal balik antara *recall* dan *precission* adalah *FMeasure* yang merupakan bobot *harmonic mean* dan *reall* dan *precission*. Dapat dilihat pada persamaan 2.13.

F-Measure = (2 ∗ *Recall* ∗ *Precision*) / (*Recall* + *Precision*)…….. (2.13)

2.8. *Text Preprocessing*

Text preprocessing merupakan Langkah yang sangat penting dalam analisis sentimen, data yang telah dikumpulkan perlu dilakukan *preprocessing* sehingga menghasilkan *clean* data yang digunakan untuk membuat *word vectors* dan klasifikasi sentimen menjadi lebih akurat (Rahman, 2021). Langkah-langkah *preprocessing* sebagai berikut:

1. *Case folding*

Adalah salah satu bentuk *text preprocessing* yang paling sederhana dan efektif meskipun sering diabaikan. Tujuan dari *case folding* untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf „a‟ sampai „z‟ yang diterima.

2. *Tokenizing*

Pada bagian Tokenizing, kalimat hasil dari *Case Folding* di pecah menjadi beberapa bagian kata. Pemecahan kalimat berdasarkan tanda spasi antar kalimat, sehingga dibuatlah *list* yang terdiri dari kumpulan kata yang disebut token.

3. *Stopword*

Adalah tahap mengambil kata penting dari hasil *tokenizing* dengan algoritma *stopword removal* (membuang kata yang kurang penting). *Stopword* merupakan kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag of words*.

4. *Stemming*

Merupakan proses mengembalikan kata-kata menjadi bentuk kata dasar. Proses *stemming* membantu mengurangi kantong kata-kata dengan mengaitkan kata-kata yang mirip dengan kata-kata dasar yang sesuai. Contoh: mengumpulkan menjadi kumpul,

bagaimanakah menjadi bagaimana, perbedaan menjadi berbeda, dan lain-lain.

2.9. *Deep learning*

Deep learning adalah cabang ilmu dari *machine learning* berbasis jaringan saraf tiruan (JST) atau bisa dikatakan perkembangan dari JST. Perbedaan dengan JST sendiri adalah banyaknya hidden layer pada deep learning yang di modelkan sedemikian rupa sehingga mampu memberikan output yang lebih akurat.*Deep learning* mengajari komputer melakukan sesuatu yang natural seperti manusia dan memiliki beberapa algoritma. *Deep learning* menemukan struktur yang rumit dalam kumpulan data yang besar dengan menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana sebuah mesin harus mengubah parameter internalnya yang digunakan untuk menghitung representasi pada setiap lapisan dari representasi pada lapisan sebelumnya (Marifatul, 2018).

2.10. *Rapid Miner*

Rapid miner adalah platform perangkat lunak ilmu data yang dikembangkan oleh perusahaan bernama sama dengan yang menyediakan lingkungan terintegrasi untuk persiapan data, pembelajaran mesin, pembelajaran dalam, penambangan teks, dan analisis prediktif.

Rapid Miner merupakan perangkat lunak yang dibuat oleh Dr. Markus Hofmann dari *Institute of Technologi* Blanchardstown dan Ralf Klinkenberg dari *rapid-i.com* dengan tampilan *Graphical User Interface* (GUI) sehingga memudahkan pengguna dalam menggunakan perangkat lunak ini. Perangkat lunak ini bersifat *open source* dan dibuat dengan menggunakan program Java di bawah lisensi GNU *Public Licence* dan *Rapid Miner* dapat dijalankan di sistem operasi manapun. Dengan menggunakan *Rapid Miner*, tidak dibutuhkan kemampuan koding khusus, karena semua fasilits sudah disediakan. *Rapid*

Miner dikhususkan untuk penggunaan data mining. Model yang disediakan juga cukup banyak dan lengkap, seperti Model *Bayesian, Modelling*, *Tree Induction, Neural Network* dan lain-lain Banyak metode yang disediakan oleh *Rapid Miner* mulai dari kllasifikasi, klustering, asosiassi dan lain-lain (Sudarsono, 2021).

2.11. API *Twitter*

Twitter adalah sebuah media sosial dan layanan *microblogging* yang mengijinkan penggunanya untuk mengirimkan pesan *realtime.* Pesan ini populer dengan sebutan tweet. *Tweet* adalah sebuah pesan pendek dengan panjang karakter yang dibatasi hanya sampai 140 karakter. Dikarenakan keterbatasan karakter yang bisa dituliskan, sebuah *tweet* seringkali mengandung singkatan, bahasa slang maupun kesalahan pengejaan (Agarwal, 2014).

Berikut ini adalah beberapa istilah yang dikenal dalam *Twitter*:

1. *Mention. Mention* adalah menyebut atau memanggil pengguna *Twitter* lain dalam sebuah *tweet*. *Mention* dilakukan dengan menuliskan '@' diikuti dengan nama pengguna lain.
2. *Hashtag. Hashtag* digunakan untuk menandai sebuah topik pembicaraan di *Twitter.* Penulisan *hashtag* dimulai dengan tanda '#' diikuti dengan topik yang sedang dibahas. *Hashtag* biasa digunakan untuk meningkatkan visibilitas *tweet* pengguna.
3. *Emoticon. Emoticon* adalah ekspresi wajah yang direpresentasikan dengan kombinasi antara huruf, tanda baca dan angka. Pengguna biasa menggunakan *emoticon* untuk mengekspresikan *mood* yang sedang mereka rasakan.
4. *Trending topics*. Jika *hashtag* adalah cara untuk menandai sebuah topik pembicaraan di *Twitter,* maka trending *topics* adalah kumpulan dari topik pembicaraan yang sangat

populer di *Twitter.*

2.12. *Covid 19*

COVID-19 merupakan penyakit menular yang disebabkan oleh virus bernama SARS-COV2, atau seringkali disebut Virus *Corona.* Virus *Corona* sendiri merupakan keluarga virus yang sangat besar. Ada yang menginfeksi hewan, seperti kucing dan anjing, namun ada pula jenis Virus *Corona* yang menular ke manusia, seperti yang terjadi pada *COVID-19*. Sampai saat ini terdapat lebih dari 1,2 juta orang terinfeksi dan hampir 65 ribu orang meninggal dunia. Di Indonesia sendiri, ada lebih dari 2 ribu kasus ditemukan dan hampir 200 orang telah meninggal dunia (Ramadhani, 2020).

2.13. Vaksin

Vaksin adalah produk biologi yang berisi antigen berupa mikroorganisme atau bagiannya atau zat yang dihasilkannya yang telah diolah sedemikian rupa sehingga aman, yang apabila diberikan kepada seseorang akan menimbulkan kekebalan spesifik secara aktif terhadap penyakit tertentu (Kemenkes RI, 2021).

2.13.1. Vaksin *Booster*

Vaksinasi *booster* merupakan vaksinasi ulang atau vaksinasi penguat setelah selang beberapa waktu, biasanya 1-2 minggu setelah vaksinasi pertama, atau tergantung kebutuhan, dengan tujuan untuk meningkatkan efikasi (Mulia, 2006).

2.14. Klasifikasi

Klasifikasi merupakan suatu proses menemukan kumpulan pola atau fungsi yang mendeskripsikan serta memisahkan kelas data yang satu dengan yang lainnya untuk menyatakan objek tersebut masuk pada kategori tertentu yang sudah ditentukan. Klasifikasi adalah bentuk analisis data yang mengekstrak model yang menggambarkan kelas data.

(Kuryanti, 2015).

2.15. *Python*

Python adalah salah satu bahasa pemograman tingkat tinggi yang bersifat interpreter, *interactive, objectoriented*, dan dapat beroperasi hampir di semua *platform: Mac, Linux*, dan *Windows. Python* termasuk bahasa pemograman yang mudah dipelajari karena sintaks yang jelas, dapat dikombinasikan dengan penggunaan modul modul siap pakai, dan struktur data tingkat tinggi yang efisien (Prasetya, 2012).

2.16. *Jupyter notebook*

Jupyter Notebook adalah, pengembangan dari *python* atau *Interactive Python*. *Jupyter Notebook* ini suatu editor dalam bentuk *web* aplikasi yang berjalan di *localhost* komputer, adapun beberapa hal yang dapat dilakukan oleh *Jupyter Notebook* seperti menulis kode *python, equations, visualisasi* dan bisa juga sebagai *markdown editor* (Sudarsono, 2021).

BAB III

METODE PENELITIAN

3.1. Objek Dan Waktu Penelitian

Penulis melakukan penelitian mengenai klasifikasi teks dengan objek penelitian adalah artikel berita berbahasa Indonesia. Penelitian ini dilakukan pada semester ganjil tahun ajaran 2021-2022.

3.2. Alat Dan Bahan Penelitian

Dalam melakukan penelitian ini, ada beberapa spesifikasi alat penelitian yang harus dipenuhi. Spesifikasi alat penelitian maksudnya adalah standar minimal dari alat (*tools*) yang digunakan sebagai wadah utama dalam melakukan perbandingan penelitian ini. Alat yang digunkan antara lain sebagai berikut:

3.2.1. Defenisi Spesifikasi *Hardware*

Detail Spesifikasi *Hardware* dapat dilihat pada tabel 3.1.

Tabel 3.1. Detail Spesifikasi *Hardware.*

No	Jenis	Spesifikasi
1	PC	*Acer Aspire* A515-45
2	*Processor*	*AMD Ryzen* 5 5500U *with Radeon Graphics* 2.10 GHz
3	Memori	16.0 GB
4	I/O	*Monitor, Keyboard, Mouse*

3.2.2. Defenisi Spesifikasi *Software*

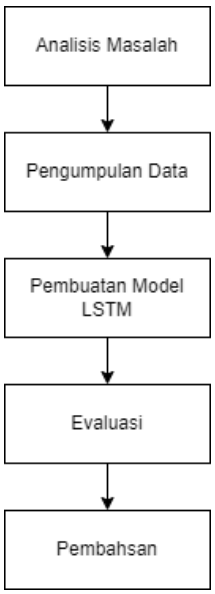
Detail Spesifikasi *Software* dapat dilihat pada tabel 3.2.

Tabel 3.2. Detail Spesifikasi *Software.*

No	Jenis	Spesifikasi	Keterangan
1	Sistem Operasi	Windows 10 64 bit	Sistem operasi yang digunakan saat penelitian
2	Aplikasi Text Editor	Python versi 3.6 Jupyter Notebook, Visual Studio code & Rapidminer	Digunakan untuk menulis dan menjalankan script Python

3.3. Tahapan Penelitian

Metodologi penelitian merupakan langkah-langkah yang dilakukan oleh penulis dalam proses penelitian. Tahapan pada penelitian dapat dilihat pada gambar 3.1.



Gambar 3.1. Tahapan Penelitian

1. Analisis masalah

Tahapan pertama dalam penelitian ini adalah analisis masalah yang tujuannya untuk mengidentifikasi sejumlah masalah yang ada mengenai teks klasifikasi pada sentiment *twitter*.

2. Pengumpulan data

Pada tahapan ini peneliti melakukkan proses pengumpulan data dengan menggunakan tools rapidminer sebagai bagian yang paling utama dalam penelitian. Data yang dikumpulkan merupakan data sekunder yang diambil dari twitter yang terdiri dari 500 data cuitan

3. Pembuuatan model LSTM

Pada tahapan ini dilakukan proses pembuatan model LSTM dan dilakukan proses

pelatihan terhadap model LSTM yang dibuat dengan menggunakan data latih (data training) untuk menghasilkan *output* model hasil pelatihan atau model terlatih yang akan digunakan pada proses evaluasi. Tahapan ini dilakukan secara berulang sampai menghasilkan model yang optimal.

4. Evaluasi

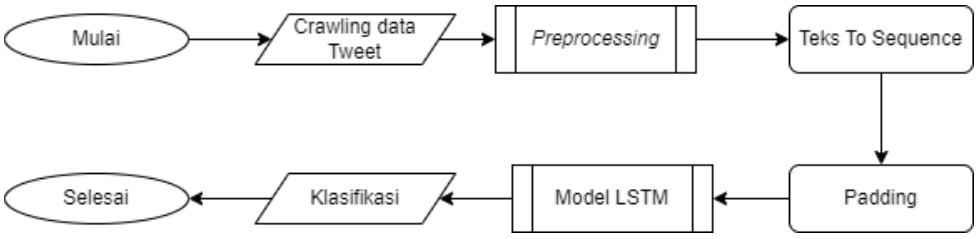
Evaluasi dilakukan sebagai proses pengujian terhadap kinerja atau ketepatan proses klasifikasi *tweet* menggunakan metode LSTM. Tahapan ini dimaksudkan untuk mengevaluasi seberapa baik performa dari System menggunakan *Confusion Matrix* dengan menghitung nilai dari *accuracy, precision, recall*, dan *F-Measure.*

5. Pembahasan

Tahapan ini merupakan tahapan terakhir yang bertujuan untuk mengulas hasil dari evaluasi model LSTM dalam mengklasifikasikan data teks pada *tweet*.

3.4. *Flowchart* Pengolahan Data

Flowchart sistem merupakan tahapan atau langkah-langkah sistem melakukan proses klasifikasi. Lebih jelasnya dapat dilihat pada gambar 3.2.



Gambar 3.2. *Flowchart* Pengolahan Data.

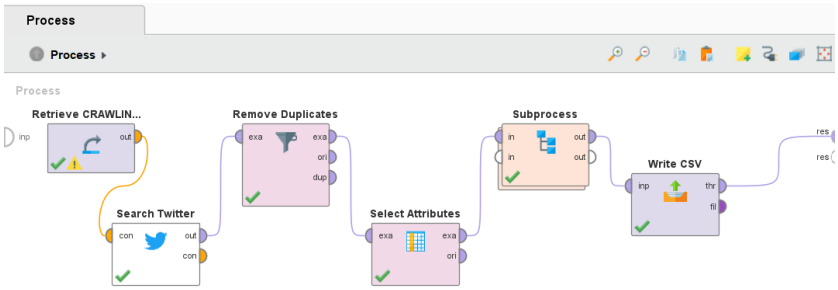
Pada gambar 3.2. Menjelaskan mengenai alur atau tahapan kerja dari sistem yang akan dibangun yang diawali dari input *crawling* data *twitter, preprocess teks, text to sequences, padding,* model LSTM dan yang terakhir *output* berupa hasil klasifikasi oleh sistem.

Data yang akan di ambil adalah data komentar dari masyarakat yang mengandung

pro dan kontra tentang vaksin *booster*, komentar yang dimaksud yaitu seluruh komentar tentang vaksin *booster* dengan menggunakan kata kunci #vaksin*Booster* dan #vakasin3. Sumber datanya berasal dari aplikasi *twitter* dengan menggunakan bantuan *tools rapidminer.*

3.4.1. *Crawling* Data *Twitter*

Peneliti melakukan *crawling* data dengan mengumpulkan data menggunakan *Rapidminer* dengan memasukkan *acces token* yang di dapat melalui pendaftaran di *Twitter Developer*, lebih jelasnya dapat dilihat pada gambar 3.3.



Gambar 3.3 Proses *Crawling* Data *Twitter* Menggunakan *Rapidminer.*

Pengumpulan data menggunakan keyword #vaksin 3 dan #vaksin *booster* Jumlah data 518 *tweet* kemudian akan dibagi menjadi data training dan data testing dengan perbandingan 80%:20%. Berikut merupakan alur *Crawling* data *Twitter:*

1. *Retrieve Crawling*

Peneliti menggunakan Operator ini untuk mengakses informasi yang disimpan di repositori dan memuatnya kedalam proses *crawling* data seperti pada gambar 3.3.

2. *Search twitter*

Operator ini digunakan untuk mengambil data dari media sosial *twitter* data yang bisa diambil dari *twitter* yaitu profil, komentar dan lain lain sebagainya.

3. *Remove* duplikat

Fitur *Remove Duplicates* digunakan untuk menghapus data yang sama dengan membandingkan keseluruhan dataset satu sama lain berdasarkan atribut yang ditentukan. Operator ini menghapus contoh duplikat sehingga hanya satu dari semua contoh duplikat disimpan.

4. *Select attributs*

Operator ini digunakan untuk memilih *attribute* mana saja yang ingin dipilih untuk diproses.

5. Subproses

Operator ini digunakan untuk memproses sebuah proses dalam proses seluruh subproses dieksekusi setelah di eksekusi subproses aliran dikembalikan keproses awal.

6. *Write* CSV

Operator ini digunakan untuk menyimpan dataset dalam bentuk CSV dari data hasil *crawling* yang telah peneliti lakukan. Kemudian data akan disajikan dalam bentuk tabel. Untuk lebih jelasnya dapat dilihat pada table 3.3.

Tabel 3.3. Sampel Data *Tweet.*

NO	Tweet Komentar
1	Vaksinnya gak tamat2 sekalian aja vaksin 3 kali sehari… kalua berlanjut hubungi dokter?
2	1 ampe 3 alhamdulillah langsung olahraga gua abis vaksin
3	Dirasa rasa makin gampang kena flu dll setelah vaksin booster covid 3 yaacchh. Ni ada kali sebulan sekali kena flu
4	Percuma vaksin sampe 3 kali klo masih positif
5	Gue panik anjing kirain baru vaksin 2x ternyata udh 3 wkwowkowkw
6	Meskipun udh vaksin 3 kali jangan lupa untuk selalu menerapkan protocol Kesehatan yaa guys
7	Temenku ada yg vaksin 1 ke vaksin 3 jatuh sakit. Tapi kekebalan
8	Booster gw di akhiri dengan demam 3 hari setelah vaksin dapet semingguan
9	Gw udh vaksin 3×, jujur nih ya bukannya bermaksud apa” tp setelah vaksin badan gua gampang sakit?

500	Kadang mikir, aku yg dah vaksin 3 kali aja masih kenak, jadi apa sebenarnya guna vaksin" ini ya?

3.4.2. *Preprocessing Teks*

Preprocessing merupakan salah satu tahapan menghilangkan permasalahan-permasalahan yang dapat mengganggu hasil daripada proses data. Data yang digunakan dalam proses mining tidak selamanya dalam kondisi yang ideal untuk diproses. Dalam *preprocessing Pyton* terdapat beberapa tahapan perintah yang digunakan dalam mengolah data dengan menggunakan *jupyter notebook*:

1. *Case folding*

Teks dalam komentar biasanya memiliki beragam penulisan, salah satunya adalah menggunakan huruf besar dan kecil. Untuk mengatasi hal ini, teks akan diubah dalam huruf kecil melalui operator dengan perintah pada gamabar 3.4. Sebagai berikut:

```
#case_folding
hapusselainhuruf = re.sub(r'[^A-za-z0-9\/\.]+',' ', komentar)
hapustitikspasi = re.sub(r'\s\.\s', ' ', hapusselainhuruf)
hapustitikawaldanakhir = re.sub(r'(\.)|(\.\s)|(\.$)', ' ', hapustitikspasi)
hapusangka = re.sub(r'(\d)', '', hapustitikawaldanakhir)
sim_komentar = hapusangka.lower()

dataset_y = kelas.lower()
```

Gambar 3.4 Contoh Perintah *Case Folding.*

Contoh penerapan case folding dapat dilihat pada tabel 3.4.

Tabel 3.4. Contoh Penerapan *Case Folding.*

Komentar	penumpang usia 18 tahun keatas wajib sudah vaksin dosis 3 (booster). P emeriksaan -PCR dan antigen sudah tidak diberlakukan ya. Informasi sec ara lengkap bisa dicek
Case folding	penumpang usia 18 tahun keatas wajib sudah vaksin dosis 3 (booster). p emeriksaan -pcr dan antigen sudah tidak diberlakukan ya. informasi secar a lengkap bisa dicek

2. *Stemming*

Dijelaskan tahapan mengubah kata-menjadi kata dasar dengan menghilangkan

imbuhan (*affixes*) yaitu awalan (*prefixes*), sisipan (*infixes)*, akhiran (*suffixes*) dan *conflixes* (kombinasi dari awalan dan akhiran) pada kata turunan yang ada didokumen akan dicocokkan dengan KBBI. Perintah *stemming* dapat dilihat pada gambar 3.5.

```
#stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()

hasil_stem = stemmer.stem(sim_komentar)

stop_sas = stopword.remove(hasil_stem)

dataset_x = stop_sas
```

Gambar 3.5.Contoh Perintah *Stemming.*

Contoh penerapa stemming dapat dilihat pada tabel 3.5.

Tabel 3.5. Contoh Penerapan *Stemming.*

Komentar	Penumpang usia18 tahun keatas wajib sudah vaksin dosis 3 (*booster*). Pemeriksaan -PCR dan antigen sudah tidak diberlakukan ya. Informasi secara lengkap bisa dicek
Stemming	tumpang usia 18 tahun atas wajib sudah vaksin dosis 3 *booster* periksa -pcr dan antigen sudah tidak laku ya informasi cara lengkap bisa cek

3. Stopword

Stopwords adalah kata umum (*common words)* yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh *stopwords* untuk bahasa Inggris diantaranya "*of", "the"*. Sedangkan untuk bahasa Indonesia diantaranya "yang", "di", "ke". Untuk tahapan ini cukup menyulitkan dikarenakan komentar orang saat ini tidak memakai bahasa Indonesia yang baku dan benar sehingga menyulitkan dalam melakukan penghapusan kata-kata. D1 sampai D8 menunjukan dokumen kata-kata yang dihapus ditandai dengan kolom-kolom kata yang dikosongkan berikut. Pada gambar dan tabel 3.6 dijelaskan tahap *stopword* dengan membuang kata-kata yang tidak penting.

```
#stopword
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
```

Gambar 3.6 Contoh Perintah *Stopword.*

Contoh penerapan stopword dapat dilihat pada tabel 3.6.

Tabel 3.6 Contoh Penerapan *Stopword.*

Komentar	penumpang usia 18 tahun keatas wajib sudah vaksin dosis 3 (*booster*). Pemeriksaan -PCR dan antigen sudah tidak diberlakukan ya. Informasi secara lengkap bisa dicek
Stopwords	tumpang usia 18 tahun atas wajib vaksin dosis 3 *booster* periksa -pcr antigen tidak laku informasi cara lengkap cek

4. Tokenizing

Dijelaskan tahapan *tokenizing* atau disebut juga parsing. Pada tahapan ini sebuah dokumen kalimat dipecah-pecah menjadi kata-kata kemudian menganalisis terhadap kumpulan kata dengan memisahkan kata tersebut dan menentukan struktur sintaksis data tiap kata tersebut. 1 sampai 1000 menunjukan dokumen yang sudah di *tokenizing* dan setiap kalimat dalam dokumen yang dipecah-pecah menjadi kata-kata sehingga yang terpisah bahasa kata yang satu dengan kata yang lain dapat dilihat pada gambar dan tabel 3.7.

```
#tokenizing
def hitung_token(wm, feat_names):
    doc_names = ['data{:d}'.format(idx) for idx, _ in enumerate(wm)]
    df = pd.DataFrame(data=wm.toarray(), index=doc_names, columns=feat_names)

    return(df)

token = CountVectorizer(lowercase=True)
wm = token.fit_transform(dt_x)
hasil_token = token.get_feature_names_out()

hitung_token(wm, hasil_token)
```

Gambar 3.7 Contoh Perintah *Tokenizing.*

Contoh penerapan tokenizing dapat dilihat pada tabel 3.7.

Tabel 3.7. Contoh Penerapan *Tokenizing.*

Komentar	penumpang usia 18 tahun keatas wajib sudah vaksin dosis 3 (*booster).* Pemeriksaan -PCR dan antigen sudah tidak diberlakukan ya. Informasi secara lengkap bisa dicek

Token	['18' 'antigen' 'atas' '*booster*' 'cara' 'cek' 'dosis' 'informasi' 'laku' 'lengkap' 'pcr' 'periksa' 'tahun' 'tidak' 'tumpang' 'usia' 'vaksin' 'wajib']

3.4.3. *Teks To Sequence*

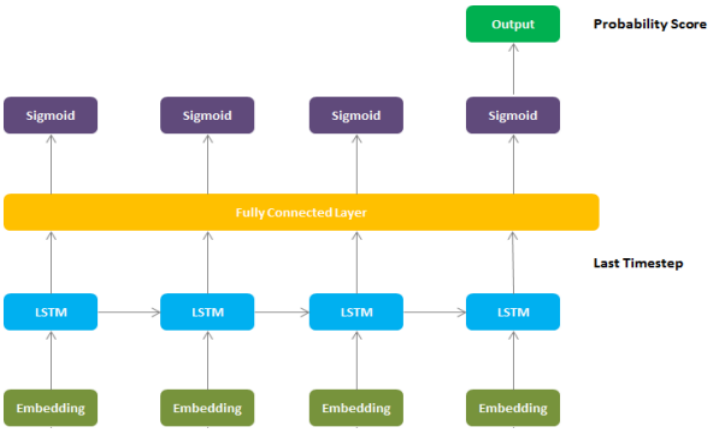
Pada tahapan ini akan memanfaatkan fungsi *text to sequens* di Keras yang berfungsi untuk melakukan vektorisasi korpus teks dengan mengubah setiap teks menjadi *sequens* bilangan.

3.4.4. *Padding*

Selanjutnya akan dilakukan proses menyamakan panjang data yang biasa disebut dengan proses *padding*. Hal ini dilakukan karena tiap kalimat yang ada memiliki panjang yang berbeda. Sehingga diperlukan untuk menyeragamkan panjang kata agar dapat diproses pada model *neural network*. Konsep dari *Padding* sendiri adalah memotong kalimat yang terlalu panjang dan akan memberikan nilai 0 pada data yang terlalu pendek.

3.4.5. Model *Long Short Term Memory* (LSTM)

Tahapan terakhir barulah data akan diproses dengan menggunakan LSTM. Tahapan pertama yaitu data masuk ke dalam *forget gate* yang fungsinya mengontrol dan melakukan seleksi terhadap informasi pada memori, setelah itu data masuk menuju *input gate* yang fungsinya mengontrol dan menentukan informasi baru apa yang akan ditambahkan kedalam sel LSTM tersebut. Selanjutnya memori pada sel LSTM tersebut diupdate. Kemudian masuk menuju output gate yang fungsinya menghasilkan atau mengeluarkan data yang sudah dilakukan perhitungan pada sel LSTM tersebut. Arsitektur LSTM pada penelitian ini terdiri dari *Embedding Layer, LSTM Layer, Fully Connected Layer* dan *Sigmoid Layer.* Lebih jelasnya dapat dilihat pada gambar 3.8.



Gambar 3.8. Model LSTM

Pada gambar 3.8. Tahapan pertama yaitu *Embedding Layer* yang berfungsi untuk mengkonstruksi Vektorisasi kata dengan besar dimensi 200 yang akan digunakan sebagai *input layer*, selanjutnya ada LSTM *layer* (*hidden Layer*) dengan jumlah unit 256 *neuron* didalam LSTM *layer* juga *dropout Layer* sebesar 50% untuk menghindari overfitting dan mempercepat proses pembelajaran, untuk *Fully Connected Layer* bertujuan untuk memetakan keluaran lapisan LSTM ke ukuran yang dinginkan, nilai *input* untuk *layer* ini sama dengan jumlah unit di LSTM *layer* dan nilai *output*nya sama dengan 1 karena sebagai *binary* klasifikasi anatara positif dan negatif, dan ada *sigmoid activation layer* Ini diperlukan hanya untuk mengubah semua nilai keluaran dari lapisan yang terhubung sepenuhnya menjadi nilai antara 0 dan 1 atau positif dan negatif.

3.5. Confusion Matrix

Dalam penelitian ini pengujian *confusion matrix* digunakan untuk mengukur performa model yang dibangun. Parameter yang digunakan adalah *true positif* (TP), *true negative* (TN), negative palsu (FN) dan positif palsu (FP). Parameter ini digunakan untuk proses perhitungan *accuracy, precision, recall* dan *f-measure.* Lebih jelasnya dapat dilihat

pada tabel 3.8.

Tebel 3.8. Pengujian *Confusion Matrix.*

Accuracy	$(\mathrm{TP}+\mathrm{TN})/(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN})$	Perbandingan antara kelas yang diprediksi dengan benar oleh sistem terhadap total data yang ada
Precision	$TP/(TP+FP)$	Perbandingan antara kelas yang di prediksi dengan benar oleh sistem terhadap total data yang terklasifikasikan suatu kelas.
Recall	$TP/(TP+FN)$	Perbandinga antara kelas yang di prediksi dengan benar oleh sistem terhadap total data yang ada pada suatu kelas.
F-Maesure	$(2 \times reccal \times precision)/(recal + precision)$	Perbandingan rata-rata presisi dan recall yang dibobotkan.

3.6. Library

Model *Deep learning* yang akan dibangun pada penelitian kali ini akan dibangun menggunakan beberapa *Library. Library* utama yaitu *TenserFlow* dan Keras*. TensorFlow* adalah salah satu *Library* dikhususkan untuk melakukan perhitungan matriks multidimensi secara optimal yang dikhususkan untuk membangun model jaringan saraf tiruan.

Keras adalah *Aplication Programming Interface* (API) jaringan saraf tiruan tingkat tinggi yang ditulis dengan *python* dan mampu berjalan diatas *TenserFlow*. Keras merupakan *framework* yang terdapat pada *python* yang diperuntukan untuk membuat aplikasi atau *prototype Deep Neural Network*.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil dari pengambilan data dan menghasilkan data yang sudah di*preprocessing* serta menunjukan visualisasi pada setiap kelas dari hasil labeling. Serta akan dipaparkan grafik hasil *training* data dan menunjukkan hasil uji coba parameter untuk mendapatkan model yang optimal sampai mendapatkan evaluasi dengan menunjukan hasil dari *confusion matrix.*

4.1. Implementasi

Data komentar aplikasi *twitter* yang didapatkan melalui *rapid miner* dari API *twitter* selanjutnya di simpan dengan format CSV. Setelah data didapatkan dalam bentu CSV kemudian data seluruhnya diberi label sentiment. Dikarenakan data tersebut merupakan data teks yang tidak terstruktur maka perlu dilakukan beberapa tahapan *preprocessing,* tahapan yang dilakukan diantaranya adalah *case folding, removestopword dan tokenization*. Hasil dari tahap *preprocessing* dan pelabelan seperti pada Tabel 4.1

Tabel 4.1. Sampel Data *Tweet.*

Tweet	Pelabelan Manual
sia sia aku vaksin kali mending aku minum dettol aja sekali	negatif
jasa joki vaksin booster tembus peduli lindung	negatif
vaksin booster sama vaksin beda apa sama sih	Positif
hai ka kalau butuh jasa vaksin boleh dm aja proses cepat aman percaya tembus pedulilindungi sinkron sistem pcare permanent	Positif
mau vaksin booster blm dpt tiket karna vaksin nya kurang bulan aku input paksa rb aja	Negatif
jaga diri meski vaksin kali virus masih pantau di mana mana kalau masker tetap pasang pas nongkrong syukur syukur kalau kena gejala kalau gejala sangat tidak enak soal semua jadi tunda stay safe	Positif

kemarin adeku vaksin terus nyampe stasiun ditawarin buat vaksin lokasi gituuu cuma gtau di semua stasiun engga	Positif
kalau kereta jarak jauh vaksin sekarang swab sama pcr udah ga laku	Positif
jasa joki tembak vaksin	Negatif
umur berapa kalo udah atas wajib vaksin kalo di bawah wajib vaksin	Positif
jadi hari pas bgt minggu negatif sempet positif covid terus tadi siang vaksin padahal kabar minimal vaksin bulan nyata negatif jujur skrg gue parno wkwk	Negatif
aku udah vaksin kali	Positif
alhamdulillah langsung olahraga gua abis vaksin	Positif
kita patuh prokes laku pakai masker benar baik maupun luar ruang segera vaksin dosis tiga vaksin booster jaga perilaku hidup bersih sehat tangkal covid dengan prokes	Positif
iya udah tapi emang baru buat nakes lansia aja buat apa buat tingkat kebal tubuh wkwkwkwk aku dah vaksin kali kemarin sempet kena kok	negatif
ak ga tkut ak dah vaksin	Positif
rasa rasa makin gampang kena flu telah vaksin booster covid yaacchhh ni kali bulan sekali kena flu,negatif	negatif
gak usah bukti di tiket kereta waktu cetak udah lihat udah vaksin apa kalau misal booster pakai surat swab kalau gak salah	Positif
sekarang naik kereta wajib vaksin kai	Positif
suntik vaksin dr kemenkes ndak manjur dong	negatif

Setelah data didapatkan dari tahap *Rapid miner*. Data kemudian dianalisis dengan memberikan label setiap ulasannya. Label pada penelitian kali ini menggunakan positif dan negatif. Label pada penelitian ini digunakan label 1 untuk kalimat yang bernilai sentiment positif misalnya kepuasan dalam melakukan vaksin *booster* sedangkan 0 untuk yang bernilai sentimen negatif misalnya tanggapan kekecewaan dan keluhan dari masyarakat. Tahap ini bertujuan memberikan pembelajaran pada model yang akan dibentuk di tahap pelatihan data. Pada tahap pelabelan ini dilakukan secara langsung oleh Dra. Bekti Nirmala GDP.,

M.Pd., yang bertujuan untuk menyamakan presepsi kalimat ulasan termasuk kedalam label yang mana. untuk akhirnya memutuskan kalimat tersebut masuk ke label positif atau negative yang dapat dilihat pada gambar 4.1.

Out[234]:

	komentar	kelas
0	zulaikha tumpang usia tahun atas wajib vaksin ...	positif
1	bangediii vaksin gak tamat sekali aja vaksin k...	negatif
2	brownmactha hai ka kalau butuh jasa vaksin bol...	positif
3	tuzaklar hai ka kalau butuh jasa vaksin boleh ...	positif
4	woopuy hai ka kalau butuh jasa vaksin boleh dm...	positif
...	...	...
496	hari sedikit dosis vaksin ikut orang tenaga me...	positif
497	jasa joki vaksin booster tembus peduli lindung	negatif
498	prof segera booster atau vaksin dosis pakai va...	positif
499	sudah vaksin ingat covid akhir	positif
500	ak g vaksin kmrn krn gk cukup umur skrg ap dha...	positif

501 rows × 2 columns

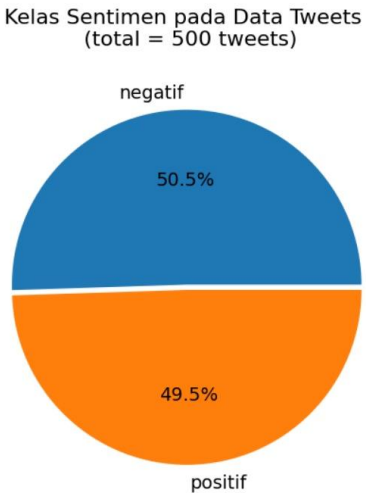
Gambar 4.1. Sampel Data *Tweet* Pada Program

Perintah untuk menampilkan grafik dapat dilihat pada code pada gambar 4.2.

```
fig, ax = plt.subplots(figsize = (6, 6))
sizes = [count for count in df['kelas'].value_counts()]
labels = list(df['kelas'].value_counts().index)
explode = (0.03, 0)
ax.pie(x = sizes, labels = labels, autopct = '%1.1f%%', explode = explode, textprops={'fontsize': 14})
ax.set_title('Kelas Sentimen pada Data Tweets \n (total = 500 tweets)', fontsize = 16, pad = 20)
plt.show()
```

Gambar 4.2. Perintah Grafik Pie

Data tweet ini kemudian dapat digunakan dalam berbagi cara dalam program, *dataset* diatas dapat digambarkan grafik perbandingan antara kelas positif dengan kelas negatif pada gambar 4.3.



Gambar 4.3. Diagram Perbandingan Kelas Sentimen

Kelas positif ditandai dengan warna *orange* dan kelas negatif ditandai dengan warna biru, yang mana diagram lingkaran atau diagram *pie* dimana setiap potong *pie* akan menampilkan ukuran tertentu seperti pada gambar 4.3 diatas menunjukan hasil perbandingan kelas negatif sebesar 252 dengan nilai 50,5% dengan positif 248 dengan nilai 49,5% yang artinya sentimen negatif lebih besar dari sentimen positif. Kemudian agar data komentar dapat diolah dengan mode LSTM maka data *tweet* di atas harus diubah menjadi angka yaitu di tandai dengan angka 1 dan 0 dimana 1 adalah positif dan 0 adalah negatif untuk lebih jelasnya dapat dilihat pada tabel 4.2.

Tabel 4.2. Angka *Integer*

Angka *Integer*
1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0,
0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0,
0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1,
0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0,
1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0,
1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,
1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0,
0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1,
1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0,

```
1,  0,  0,  0,  1,  1,  0,  0,  1, 0,  0,  0,  1,   0,  1,  1,  0,  1,  0,   1,  1, 0,
1,  1,  1,  0,  0,  0,  0,  0,  1,  1,  0,  1,  0,  0,  0,  0,  1,  1,  0,  0,  0, 1,
0,  1, 0,   1,  0,  1, 1,  0,  0,  1,  1,  0,  0,  1,  1,  0,  0,  1, 1,   1,  1, 1,
0,  0, 0,   1,  0,  0,  0,  1,  1,  0,  0,  1,  1,  0,  1,  0,  0,  0,  1,   1,  1,  1,
1,  1, 1,   0,  1,  0,  1,  0,  1,  0,  1,  1,  0,  1,  1,  1,  1,  0,  1,   1,  1, 1,
1,  0,  0,  1,  0,  1,  1,  1,  1,  0,  1,  0,  0,  1,  0,  0,  0, 1,  1,   0, 0, 1,
1,  1, 1,   1,  1,  0,  1,  0, 0,   0,  0,  1,  1,  0,  0,  0,  0, 0,  1,   0, 0, 0,
1,  1,  0,  0,  0,  0,  1,  0,  1,  0,  1,  1,  0,  1,  0,  1,  0,  0,  1,  1, 1, 1,
1,  1,  1,  1,  1,  0,  1, 1,  0,  1,  1,  0, 0,  0,  0,  0,  1, 1,  0,  1, 1, 0,
1,  0,  1,  0,  0,  1,  1, 1,  0,  1,  0,  1,  0,  1,  0,  1,  0, 0,  0, 0, 0, 1,
1,  0,  1,  0,  1,  0,  1,  0,  0,  1,  1,  0,  1,  0,  1,  0,  0,  0,  1,  1,  0,  0,
0,  0,  1,  0,  0,  0,  0,  1,  1,  1,  1,  1,  1,  0,  1,  1,  1
```

4.2. Pembagian Data *Trening* Dan *Testing*

Pembagian data ini dilakukan dengan perbandingan menggunakan rasio 80% data *training* : 20 data *testing* dengan akurasi=0,83 Data yang didapat menyatakan bahwa persentase sebesar 0.8 data *training* dan 0.2 data *testing* memiliki akurasi tinggi. Maka diinisialisasikan dengan *test_size* = 0.2, dengan keseluruhan data 500 terbagi menjadi 400 data *training* dan 100 data *testing*, pada pembagian data ini peneliti menggunakan *library* pada python berupa sklearn. Untuk melakukan validasi dan memeriksa data saat menjalankan program berkali-kali, Mengatur *random_state* agar nilai akan menjamin urutan nomor yang acak sama hasilnya setiap dijalankan pada program, pada penelitian ini menginisialisasi *random_state* = 42.

4.3. Implementasi Model Long Short Therm Memory

Setelah *datasets* telah selesai melewati tahapan prapemrosesan data dan *splitting* data, maka tahapan berikutnya sebelum di proses dalam model LSTM *datasets* harus dilakukan beberapa tahapan diantara yaitu *indexing, sequences of integer,* dan *padding*. Karena model *deep learning* tidak bisa mengolah data yang berupa *string*/teks maka data *string*/teks harus dilakukan proses *Indexing* untuk mengkonversi setiap token menjadi bilangan numerik integer. Berikut dapat dilihat pada tabel 4.3.

4.3.1. Prosess *Indexing*

Setelah data dilakukan beberapa tahapan *preprocessing* dan menghasilkan ulasan dalam berbentuk list kata-kata yang sudah di tokenisasi. Selanjutnya adalah tahap memberikan indeks pada setiap kata di *dataset*. Pada tahap ini memerlukan parameter *num_words* yang berguna untuk mengatur ukuran *vocabulary* yang ingin digunakan. Pada penelitian kali ini parameter *num_word* menggunakan jumlah 2300 kata bedasarkan pembulatan dari jumlah kata yang umum pada pada *dataset* sebanyak 2290. Maka hasil dari memberikan indeks pada *dataset* terlihat pada Tabel 3.4.

Tabel 4.3 Proses *Indexing*

Hasil *Indexing*
'UNK': 1,'vaksin': 2,'dosis': 3,'booster': 4,'mau': 5,'jasa': 6,'aja': 7, 'yg': 8,'tembus': 9,'aku': 10,'covid': 11,'langsung': 12,'peduli': 13,'tembak': 14,'udah': 15,'yaa': 16,'pedulilindungi': 17,'kalau': 18,'bulan':19,'dm': 20,'proses': 21,'k': 22,'sertifikat': 23,'open': 24,'vaksinasi': 25,'butuh': 26,'udh': 27,'lindung': 28,'aman': 29,'kali': 3,'ka':31,'sistem': 32,'daftar': 33,'wajib': 34,'hai': 35,'boleh': 36,'paksa':37,'joki': 38,'cepat': 39,'percaya': 40,'pcare': 41,'pfizer': 42,'hari':43,'kalo': 44,'permanent': 45,'sinkron': 46,'hanya': 47,'wa': 48'bisa':49,'tiket':50,'apk':51,'ga':52,'sinopharm':53,'kereta':54'ayo':55,'atas':56,'baru':57,' umur':58,'laku':59,'jenis':60,'naik':61,'suntik': 62,'lansia':63'gak':64'kena':65,'yang': 66,'apa': 67,'di':68'masyarakat': 69,'laksana': 70,'jumat': 71,'tahun': 72,'belum': 73, 't': 74,'masuk': 75,'anak': 76,'info': 77,'cus': 78,'johnson': 79,'th': 80,'https': 81,'co': 82,'kak': 83,'nya': 84,'akhir': 85,'pas': 86,'jadi': 87,'sama': 88,'masalah': 89,'umum': 90,'sehat': 91,'kai': 92,'sudah': 93, 'syarat': 94,'beli': 95,'orang': 96,'prefix': 97,'akan': 98,'buat': 99,'do': 100,'min': 101,'zonauang': 102,'dulu': 103,'padahal': 104,'collegemenfess': 105,'gue': 106,'jalan': 107, 'mulai': 108,'muncul': 109, 'tidak': 110, 'sekarang': 111,'blm': 112,'dan': 113,'lwaozu': 114,'pakai': 115,'jamin': 116,'cek': 117,'soal': 118,'zonaba': 119,'bayar': 120, 'nder': 121,'tinggal': 122,'benerin': 123,'nembak': 124,'bsa': 125, 'tp': 126, 'usia': 127,'tau': 128,'punya': 129,'harus': 130,'dah': 131, 'juga': 132, 'gotong': 133,'aplikasi': 134,'sih': 135,'gw': 136,'pcr': 137,'satu': 138,'ya': 139, 'zonajajan': 140, 'swab': 141,'jogja': 142,'jarak': 143,'masker': 144,'jaga': 145,'selasa': 146, 'harga': 147, 'ak': 148, 'sampe': 149,'antigen': 150, 'lalu': 151, 'dos': 152,'tgl': 153,'sakit': 154, 'testi': 155,'waktu': 156, 'sini': 157, 'lama': 158, 'untuk': 159, 'cuma': 160, 'data': 161, 's': 162, 'maju': 163, 'kan': 164,'royong': 165, 'x': 166,'cepet': 167, 'terus': 168,'beri': 169,'banyak': 170,'tahap': 171,'desember': 172,'mana': 173,'lebih': 174,'ready': 175,'klo': 176,'virus': 177,'ada': 178, 'kemenkesri': 179,'gimana': 180,'lengkap': 181,'halo': 182,'semua': 183,'pinned': 184,'yaw': 185,'yuk': 186,'jg': 187,'jauh': 188,'baik': 189,'berapa': 190,'ingat': 191,'lo': 192,'sekali': 193,'sebar': 194,'juta': 195,'only': 196,'buka': 197,'poli': 198,'atau': 199,'asli': 200,'jangan': 201,'lupa': 202,'cegah': 203, 'ke': 204,'backtoens': 205,'kayak': 206,'kalian': 207,'datang':

208,'badan': 209,'nih': 210,'d': 211,'sy': 212,'uda': 213,'salah': 214,'gampang': 215,'gapapa': 216,'utk': 217,'sdh': 218,'perlu': 219,'bgt': 220,'tiap': 221,'kita': 222,'patuh': 223,'tiga': 224,'sumandogaek': 225,'ikut': 910,'sbb': 911,'hahahahaha': 912,'ma': 913,'dosen': 914,'ngeremehin': 915,'ribet': 916,'ina': 917,'inu': 918, 'gawe': 919, 'pala': 920,'puyeng': 921,'motor': 922,'mata': 923,'rebah': 924,'disakitin': 925,'widino': 926,'sering': 927,'kreta': 928,'dibolehin': 929,'tolak': 930,'rabies': 931,'zuhri': 932,'musniumar': 933,'kartu': 934,'kertas': 935,'biru': 936,'pak': 937,'yappp': 938,'bekas': 939,'ngerasa': 940,'rasa': 941,'linu': 942,'cvd': 943,'keras': 944,'menghimbau': 945, 'ttp': 946, 'disiplin': 947,'rawat': 948,'bhayangkara': 949,'lamongan': 950,'nggak': 951,'vaksiningat': 952,'jembermfs': 953,'ngasi': 954,'jatuh': 955,'api': 956,'gila': 957,'thn': 958,'skeptis': 959,'ketika': 960,'tiba': 961,'dekat': 962,'guru': 963,'olah': 964,'raga': 965,'satpam': 966,'satpol': 967,'fisik': 968,'kekar': 969,'curiga': 970,'walau': 971,'malas': 972, 'seertif': 973,'shittshows': 974,'doktertifa': 975,'agustushanhan': 976,'samping': 977,undur': 978, 'pratamanio': 979,'ampe': 980,'olahraga': 981,'abis': 982,'speckofdusk': 983,'nggk': 984,'swaban': 985,'cancel': 986,'pemkot': 987,'surabaya': 988,'gencar': 989,'fasyankes': 990,'selenggara': 991,'rek': 992,'sikk': 993,'cm': 994,'jabat': 995, 'lgsg': 996,'tita': 997,'standar': 998, 'mn': 999, 'islam': 1000,

Setelah semua kata atau token telah dikonversi menjadi numerik *integer* kemudian setiap *row* data pada *datasets* direpresentasikan kedalam *sequences of integer* yaitu merepresentasikan teks dalam larik yang terdiri dari kumpulan token dari setiap dokumen.

4.3.2. Proses *Sequence Of Integer*

Proses ini akan memanfaatkan fungsi pada keras *text_to_sequences*. Pada tahapan ini berfungsi untuk mengubah setiap dokumen menjadi *sequens*. Setiap *sequences* merepresentasikan teks dalam larik yang terdiri dari kumpulan token dari sebuah dokumen yang dihasilkan dari tahapan *indexing* pada langkah sebelumnya. Lebih jelasnya dapat dilihat pada tabel 4.4.

Tabel 4.4. *Sequences Of Integer*

Sentence	*Sequence*
- Kemenkes Belum Tiket Vaksin Booster Padahal Saya Vaksin Dan Booster Gimana Solusinya	[179,73,50,2,4,104,212,2,113, 4,180,684]
- Udah Gaboleh Pcr Kak Vaksin Dosis	[105,15,685,137,686,2,3]

Selanjutnya setiap *sequens* pada tabel 4.4 akan diterapkan proses *padding* untuk menyeragamkan panjang atau dimensi dari setiap *sequens*. Pada penelitian ini setiap *sequences* akan diberikan panjang maksimal 210 dimensi.

4.3.3. Proses *Padding*

Selanjutnya dilakukan proses penyeragaman panjang *sequeces* yang biasa disebut dengan proses *padding*. Yang berguna untuk mengatur panjang urutan vektor. Karena setiap ulasan memiliki jumlah kata yang berbeda-beda maka untuk mengisi panjang vektor agar semua sama panjang digunakan *pad_sequences* yang ada pada keras *preprocessing sequence* untuk mengisi urutan kata dengan angka 0 secara otomatis. Sehingga jika terdapat kalimat yang pendek maka makin banyak pula angka 0 pada *feature vektor* tersebut. Pada penelitian ini jumlah *input_length* adalah 30.

Konsep dari *padding* sendiri adalah memotong *sequneces* yang terlalu panjang dan memberikan nilai 0 pada tiap *sequences* yang terlalu pendek baik secara sufiks atau prefiks agar sesuai dengan panjang maksimal *sequences* yang telah ditentukan Sehingga didapatkan vektor *pad_sequence* terlihat pada Tabel 4.5.

Tabel 4.5 *Padding*

Sequence	*Padding*
[179,73,50,2,4,104,212,2,113, 4,180,684] [105,15,685,137,686,2,3]	[179, 73, 50, 2, 4, 104, 212, 2, 113, 4,180, 684, 0, 0, 0, 0, 0, 0, 0, 0] [105,15,685,137,686,2,3 ,0 ,0,0]

4.4. Arsitektur jaringan

Arsitektur jaringan dibentuk untuk menghasilkan akurasi yang optimal. Setelah menerima *outputan* dari lapisan *embedding* selanjutnya adalah melakukan pelatihan model LSTM. Pada lapisan LSTM menggunakan jumlah neuron. Selanjutnya setelah semua dibangun model dikonfigurasi terlebih dahulu dengan menggunakan optimasi Adam dan

Binary Cross Entropy untuk mengetahui nilai *loss* dari model yang sudah terbentuk. Parameter lainnya untuk membantu proses training adalah ukuran batch size dan epoch. Bedasarkan uraian parameter yang digunakan berikut adalah arsitektur jaringan pada model yang terbentuk dapat dilihat pada gambar 4.4.

```
Model: "sequential_20"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_20 (Embedding)    (None, 100, 50)           50000

 lstm_26 (LSTM)              (None, 100, 200)          200800

 flatten_6 (Flatten)         (None, 20000)             0

 dense_20 (Dense)            (None, 1)                 20001

=================================================================
Total params: 270801 (1.03 MB)
Trainable params: 270801 (1.03 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Gambar 4.4. Arsitektur Jaringan

4.5. Pengujian Parameter

Penentuaan model yang optimal ditentukan bedasarkan parameter yang menghasilkan nilai yang terbaik. Pada penelitian ini penentuan model yang optimal dengan melakukan pengujian dari beberapa parameter diantaranya adalah pengujian jumlah neuron dan pengujian fungsi aktivasi.

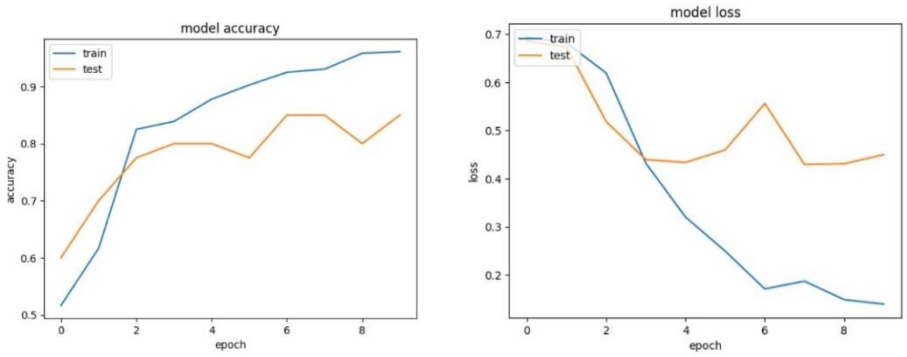
4.6. Pengujian Data

Pengujian yang dilakukan pertama adalah mencoba jumlah neuron pada lapisan yaitu dengan mencoba pada lapisan berjumlah 10 Percobaan ini dilakukan untuk menentukan jumlah neuron berapa yang optimal untuk membangun model dengan menunjukan akurasi yang terbaik seperti yang ditunjukkan pada Tabel 4.6.

Tabel 4.6. Pengujian Data

Jumlah Pengujian	*Loss*	*Accuracy*
1	0.6927	0.4861
2	0.6816	0.5611
3	0.5730	0.7472
4	0.3779	0.8806
5	0.2536	0.9000
6	0.1768	0.9306
7	0.1276	0.9583
8	0.0898	0.9722
9	0.0728	0.9722
10	0.0483	0.9833

Dari hasil pengujian data testing dan training diatas maka dapat digambarkan grafik perbandingan sebagai berikut proses trainning menggunakan jumlah *epoch* sebanyak 10 dan *batz size* sebanyak 64 dengan parameter yang sudah ditentukan akan dilihat berapa akurasi dari data training dan testing dan melihat nilai loss terrendah model akan menyimpan *epoch* yang optimal pada nilai *loss* yang terrendah selama proses *epoch* berlangsung yang dapat dilihat pada gambar 4.5.

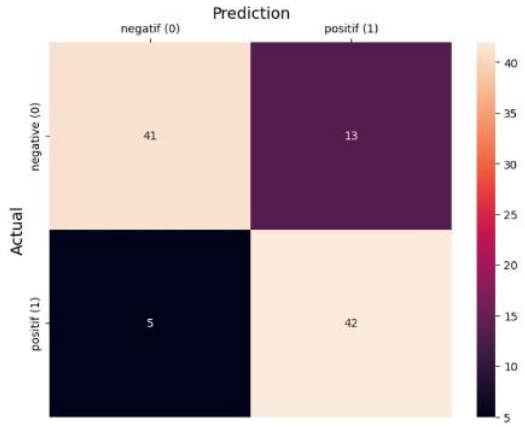


Gambar 4.5. Nilai *Accuracy* dan *loss Train Testing*

4.7. Evaluasi Data Uji

Tahap ini merupakan tahap pengujian model dengan data testing. Data yang akan diuji adalah berjumlah 500 ulasan dengan jumlah label di kelas negatif 252 ulasan dan 248 ulasan pada kelas positif. Selanjutnya setelah model menghasilkan prediksinya di masing-

masing kelas dilakukan perhitungan tingkat kepercayaan model dengan melihat akurasi, presisi dan *recall.* Tujuannya untuk mengetahui seberapa dapat dipercaya model dalam memprediksi kelas. Bedasarkan gambar 4.6 menunjukkan hasil dari pengujian model. Dimana data test pada ulasan yang berlabel negatif benar diprediksi oleh model sebesar 41 dan salah memprediki sebesar 5. Begitupula dengan ulasan yang berlabel positif dapat benar diprediksi oleh model sebesar 42 dan salah memprediksi sebesar 13 komentar.



Gambar 4.6. *Confusion Matrix*

Tabel 4.7. Confusion Matrix

Kelas Sebenarnya	Kelas Prediksi	
	Negatif	Positif
Negatif	TN =*True negative* pada penelitian tersebut nilai TN sebesar 41	FP=*False positive* pada penelitian tersebut FP sebesar 13
Positif	FN=*False Negative* FN penelitian tersebut bernilai 5	TP4= *True Positive* TP penelitian tersebut sebesar 42

Dari tabel 4.7 diatas dapat dilakukan perhitungan Akurasi, Presisi, *Recall* dan *F1-Score.* Berikut merupakan hasil evaluasi dari analisis sentimen masyarat terhadap vaksin *booster* dengan metodel *long short-term memory* untuk lebih jelasnya dapat dilihat pada tabel 4.8.

Tabel 4.8. Tabel Perhitungan Akurasi

Label	*Precision*	*Recall*	*F1-score*
Negatif	0,86	0,81	0,84
Positif	0,80	0,82	0,82
Accuracy	83%		

Evaluasi klasifikasi pada penelitian ini juga dapat dikatakan cukup baik dilihat dari nilai akurasi yang berarti model dapat dipercaya keakuratannya sebesar 83%. Begitupun dengan nilai presisi didapakan sebesar 0,86% angka ini cukup baik untuk memprediksi sentiment negatif yang merupakan sentiment terbanyak pada dataset. Kemudian nilai *recall* sebesar 81% juga memberikan hasil yang baik untuk mengetahui sensitivitas dari model yang didapat.

4.8. Visualisasi Kata Setiap Kelas

Dari hasil model yang telah memprediksi komentar *tweet* pada data testing, selanjutnya adalah visualisasi hasil prediksi untuk melihat opini apa saja yang ada dimasing-masing kelas Hasil Visualisasi dihasilkan oleh *tools* pada *phyton*. Yang berguna untuk melihat keterkaitan kata yang paling banyak muncul pada suatu kelas, dan melihat keberhubungan antar kata didalam suatu kelas. Sehingga hasil dari visualisasi ini dapat berguna untuk tweet komentar.Pada Gambar 4.7 merupakan hasil visualisasi pada kelas Positif. Hasil visualisasi pada kelas positif menunjukkan adanya tweet tentang mengajak masyarat untuk vaksin *booster*. Seperti pada ulasan berikut*"ayo vaksin".* Pada visualisasi dikelas positif ini dapat dijadikan acuan untuk masyarakat agar mau vaksin *booster*

Gambar 4.7. Visualisasi Kelas Negatif Dan Positif

BAB V

PENUTUP

5.1. Kesimpulan

Berdasrakan hasil analisis, implementasi dan pengujian pada analisis sentimen vaksin ke *3 (booster)* pada twitter menggunakan *metode long short therm memory* (LSTM) maka dapat diambil kesimpulan:

1. Berdasarkan Implementasi dan Penjelasan pada bab 4 penelitian ini menggunakan dataset komentar sebanyak 500 yang terbagi menjadi 2 bagian yaitu 252 sentimen negatif dan 248 sentimen positif, dengan perbandingan 20% data *testing* : 80% data *training*, setelah mendapatkan hasil akurasi pada data *training* didapatkan hasil, nilai *Recall* yang dihasilkan 81%, nilai Presisi sebesar 86% dan nilai *F1-Score* sebesar 84%. Dengan demikian dapat disimpulkan bahwa metode *Long short-term memory* (LSTM) layak digunakan untuk analisis sentimen vaksin ke 3 (*booster)* pada *twitter.*
2. Dari hasil analisis sentimen masyarakat terhadap kebijakan vaksin ke 3 (booster) oleh pemerintah ternyata masyatakat lebih merespon dengan tidak baik, terbukti bahwa dari banyaknya *tweet* negatif yang dihasilkan dibandingkan *tweet* yang menyukai tentang kebijakan vaksin ke 3 (booster) dari pemerintah. Model yang dihasilkan dari penelitian ini juga dapat memprediksi dengan baik tentang *tweet* masyarakat sesuai dengan tipenya, yaitu sentimen positif terhadap kebijakan vaksin ke 3 (booster) dan sentimen negatif terhadap kebijakan vaksin ke 3 (booster)
3. Hasil dari penelitian ini yaitu untuk mengimplementasikan *metode long short therm memory* (LSTM) pada komentar sentimen masyarakat terhadap vaksin ke tiga booster di *twitter.*

5.2. Saran

Berdasarkan implementasi penelitian ini masih adanya kekurangan yang harus diperbaiki maka dari itu penulis dapat memberikan saran pada peneliti berikutnya agar mendapatkan hasil yang memuaskan yaitu sebagai berikut:

1. Menggunakan metode yang berbeda untuk melakukan pembobotan kata seperti: *Word2Vec, FastText* dan pembobotan kata lainnya.
2. Pada penelitian ini apabila menggunakan model *Deep learning* maka data yang digunakan dapat dinyatakan bahwa berukuran yang kecil, sedangkan pada model *Deep learning* memerlukan data yang besar, maka peneliti dapat menambahkan data *training* yang digunakan.
3. Data yang digunakan dapat dinyatakan bahwa perbandingan sentimen tidak seimbang negatif jauh lebih banyak daripada sentimen positif, maka dari itu dapat melakukan pertimbangan dalam mengambil dataset.

DAFTAR PUSTAKA

Aldisa, R. T., & Maulana, P. 2022. Analisis Sentimen Opini Masyarakat Terhadap Vaksinasi Booster COVID-19 Dengan Perbandingan Metode *Naive Bayes, Decision Tree* dan *SVM. Technology and Science (BITS)*, *4*(1), 106–109. https://doi.org/10.47065/bits.v4i1.1581.

Aldi, M. W. P., Jondri, & Aditsania, A. 2018. Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin. Jurnal Informatika, 5, No(2), 3548. http://openlibrarypublications.telkomniversity.ac.id.

Firmansyah, M. R., Ilyas, R., & Kasyidi, F. 2020. Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network. Prosiding The 11th Industrial Research Workshop and National Seminar, 11(1), 488–495.

Fitriana, F., Utami, E., & Al Fatta, H. 2021. Analisis Sentimen Opini Terhadap Vaksin Covid - 19 pada Media Sosial Twitter Menggunakan Support Vector Machine dan Naive Bayes. Jurnal Komtika (Komputasi Dan Informatika), 5(1), 19–25. https://doi.org/10.31603/komtika.v5i1.5185.

john dan samuel Balai Pengembangan Sumber Daya Manusia dan Penelitian Kominfo Jakarta, D., Pegangsaan Timur No, J., & Pusat, J. 2021. Kecenderungan Tanggapan Masyarakat Terhadap Vaksin Sinovac Berdasarkan Lexicon Based Sentiment Analysis The Trend of Public Response to Sinovac Vaccine Based on Lexicon Based Sentiment Analysis. Jurnal Ilmu Pengetahuan Dan Teknologi Komunikasi, 23(1), 21–31. http://dx.doi.org/10.33169/iptekkom.23.1.2021.21-31.

Kirana, C., #1, P., Fachri, S., #2, P., Nuraini, R., & Fatonah, S. 2021. Meningkatkan Akurasi Long-Short Term Memory (LSTM) pada Analisis Sentimen Vaksin Covid-19 di Twitter dengan Glove. Jurnal Telematika, 16(2), 85–90. https://journal.ithb.ac.id/telematika/article/view/400.

Kuryanti, J. S. 2015. Rancangan Aplikasi Pengajuan KArtu Kuning Secara Online (studi Kasus : Dinas Tenaga Kerja dan Transmigrasi Kabupaten Musi Rawas). Seminar Nasional Inovasi Dan Tren (SNIT), 33–34. http://seminar.bsi.ac.id/snit/index.php/snit-2015/article/view/109.

Lestari, S., & Saepudin, S. 2021. Analisis Sentimen Vaksin Sinovac Pada Twitter Menggunakan Algoritma Naive Bayes. SISMATIK (Seminar Nasional Sistem Informasi Dan Manajemen Informatika), 163–170.

Marifatul Azizah, L., Fadillah Umayah, S., & Fajar, F. 2018. Deteksi Kecacatan Permukaan Buah Manggis Menggunakan Metode Deep Learning dengan Konvolusi Multilayer. Semesta Teknika, 21(2), 230–236. https://doi.org/10.18196/st.212229.

Mulia, D. S., Rachmansyah, R., & Triyanto, T. 2006. Pengaruh Cara Booster terhadap Efikasi Vaksinasi Oral dengan Debris Sel Aeromonas hydrophila pada Lele Dumbo (Clarias sp.). Jurnal Perikanan Universitas Gadjah Mada, 8(1), 36. https://doi.org/10.22146/jfs.161.

Nurhazizah, E., Ichsan, R. N., & Widiyanesti, S. 2022. Analisis Sentimen Dan Jaringan Sosial Pada Penyebaran Informasi Vaksinasi Di Twitter. Swabumi, 10(1), 24–35. https://doi.org/10.31294/swabumi.v10i1.12474.

Prasetya, D. A., & Nurviyanto, I. 2012. Deteksi wajah metode viola jones pada opencv menggunakan pemrograman python. Simposium Nasional RAPI XI FT UMS, 18–23.

Pratama, S. R., & Mirza, A. H. 2021. Penerapan Data Mining Untuk Memprediksi Tingkat Inflasi Menggunakan Metode Regresi Linier Berganda Pada BPS. Bina Darma Conference on Computer Science, 245–255.

Rahman, M. Z., Sari, Y. A., & Yudistira, N. 2021. Analisis Sentimen Tweet COVID-19 menggunakan Word Embedding dan Metode Long Short-Term Memory (LSTM). Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 5(11), 5120–5127. http://j-ptiik.ub.ac.id.

Rai, A. 2021. Analisis Sentimen Pemberitaan Vaksin Covid-19 dan Kaitannya dengan Perubahan Harga Saham Emiten Farmasi. Jurnal Bisnis Strategi, 30(1), 26–34. https://doi.org/10.14710/jbs.30.1.26-34.

Ramdhani, N., & Al-Fadillah, R. H. 2021. Analisis Sentimen Pengguna Twitter Terhadap Belajar Daring Selama Pandemi Covid-19 Dengan Deep Learning. Jurnal Siliwangi, 7(2), 2021.

Ramadhani, S. P., & Supena, A. 2020. Persepsi Orangtua dan Guru terhadap Pembelajaran Masa Pandemi COVID-19 terhadap Anak Speech Disorder Usia 8 Tahun di Madrasah

Ibtidayah. Jurnal Basicedu, 4(4), 1267–1273. https://doi.org/10.31004/basicedu.v4i4.548.

Sudarsono, B. G., Leo, M. I., Santoso, A., & Hendrawan, F. 2021. Analisis Data Mining Data Netflix Menggunakan Aplikasi Rapid Miner. JBASE - Journal of Business and Audit Information Systems, 4(1), 13–21. https://doi.org/10.30813/jbase.v4i1.2729

Yahyadi, A., & Latifah, F. 2022. Analisis Sentimen Twitter Terhadap Kebijakan PPKM di Tengah Pandemi COVID-19 Menggunakan Mode LSTM. Journal of Information System, Applied, Management, Accounting and Research, 6(2), 464–471.

LAMPIRAN

```
import re
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
import csv
#pelebelan_data
komentar ="Yang mau vaksin 1,2,3 atau 4 udh bisa yaa atau mau
langsung 4 dosis dan terdaftar di apk peduli 90k/dosis"
kelas = "negatif"

#case_folding
hapusselainhuruf = re.sub(r'[^A-za-z0-9\/\.]+',' ', komentar)
hapustitikspasi = re.sub(r'\s\.\s', ' ', hapusselainhuruf)
hapustitikawaldanakhir = re.sub(r'(\.)|(\.\s)|(\.$)', ' ',
hapustitikspasi)
hapusangka = re.sub(r'(\d)', '', hapustitikawaldanakhir)
sim_komentar = hapusangka.lower()

dataset_y = kelas.lower()

#stopword
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()

#stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()

hasil_stem = stemmer.stem(sim_komentar)

stop_sas = stopword.remove(hasil_stem)

dataset_x = stop_sas

#simpan_hasil_csv
artikel = [(dataset_x, dataset_y)]
f = open('datasentimen-fix.csv', 'a')
w = csv.writer(f)
w.writerows(artikel)

f.close()
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
data = open("datasentimen-fix.csv", 'r')
names = ["komentar", "kelas"]
df = pd.read_csv(data, names=names)
df
df.describe()
df.groupby('kelas').count()
fig, ax = plt.subplots(figsize = (6, 6))
```

```
sizes = [count for count in df['kelas'].value_counts()]
labels = list(df['kelas'].value_counts().index)
explode = (0.03, 0)
ax.pie(x = sizes, labels = labels, autopct = '%1.1f%%',
explode = explode, textprops={'fontsize': 14})
ax.set_title('Kelas Sentimen pada Data Tweets \n (total = 500
tweets)', fontsize = 16, pad = 20)
plt.show()
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split

from keras.layers import Dense, Dropout, LSTM, Embedding,
Flatten
from keras.models import Sequential
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from sklearn.metrics import accuracy_score

from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import seaborn as sns
label = LabelEncoder().fit_transform(df['kelas'])
label
X_train,       X_test,       y_train,       y_test       =
train_test_split(df['komentar'],   label,   test_size=0.20,
random_state=42)
input_length=100
tok = Tokenizer(oov_token='UNK', num_words=1000)
tok.fit_on_texts(X_train)
tok.word_indextok.word_index
X_train_seq = tok.texts_to_sequences(X_train)
X_test_seq = tok.texts_to_sequences(X_test)
X_train
X_train_seq[0]
X_train_pad          =          pad_sequences(X_train_seq,
maxlen=input_length, padding='post')
X_test_pad = pad_sequences(X_test_seq, maxlen=input_length,
padding='post')
X_train_pad[0]
model = Sequential([
    Embedding(1000, 50, input_length=input_length),
    LSTM(200, return_sequences=True),
    Flatten(),
    Dense(1, activation='sigmoid')
])
model.summary()
model.compile(loss='binary_crossentropy',  optimizer='adam',
metrics=['accuracy'])
bprint('Ready Goo...')
hist   =   model.fit(X_train_pad,   y_train,   batch_size=64,
epochs=20, validation_split=0.1, verbose = 2)
```

```
plt.plot(hist.history['accuracy'])
plt.plot(hist.history['val_accuracy'])

plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train','test'], loc='upper left')
plt.show()

plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])

plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train','test'], loc='upper left')
plt.show()
score = model.evaluate(X_test_pad, y_test, verbose = 2)
print("Test Loss:", score[0])
print("Test Accuracy:", score[1])
y_pred = model.predict(X_test_pad)
accuracy = accuracy_score(y_test, y_pred.round())
print('Model Accuracy on Test Data:', accuracy)
confusion_matrix(y_test, y_pred.round())

fig, ax = plt.subplots(figsize = (8,6))
sns.heatmap(confusion_matrix(y_true = y_test, y_pred =
y_pred.round()), fmt = 'g', annot = True)
ax.xaxis.set_label_position('top')
ax.xaxis.set_ticks_position('top')
ax.set_xlabel('Prediction', fontsize = 14)
ax.set_xticklabels(['negatif (0)', 'positif (1)'])
ax.set_ylabel('Actual', fontsize = 14)
ax.set_yticklabels(['negative (0)', 'positif (1)'])
plt.show()
print(classification_report(y_test, y_pred.round()))
from wordcloud import WordCloud
df_negatif = df[df[ 'kelas'] == 'negatif']
df_positif = df[df[ 'kelas'] == 'positif']

negatif_list=df_negatif['komentar'].tolist()
positif_list=df_positif['komentar'].tolist()

filtered_negatif = ("").join(str(negatif_list))
filtered_negatif = filtered_negatif.lower()

filtered_positif = ("").join(str(positif_list))
filtered_positif = filtered_positif.lower()
wordcloud = WordCloud(max_font_size = 160, margin = 0,
background_color = "black", colormap =
"Greens").generate(filtered_positif)
```

```
plt.figure(figsize=[10,10])
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis=("off")
plt.margins(x=0, y=0)
plt.title("Positif WordCloud")
plt.show()
wordcloud = WordCloud(max_font_size = 160, margin = 0,
background_color = "black", colormap =
"Reds").generate(filtered_negatif)
plt.figure(figsize=[10,10])
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis=("off")
plt.margins(x=0, y=0)
plt.title("Negatif WordCloud")
plt.show()
```

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	SENIN, 29 JANUARI 2024
Pukul	:	10:00 - 12:00
Tempat	:	RUANG PRODI

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa	:	REZA HI HUKUM
NPM	:	07351811024
Judul	:	ANALISIS SENTIMEN VAKSIN KETIGA (BOOSTER) PADA TWITTER MENGGUNAKAN ALGORITMA LONG SHORT THERM MEMORY (LSTM)

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

Segera melakukan perbaikan yang diberikan oleh para penguji

Selasa, 09 Juli 2024

Dosen Pembimbing I,

SAIFUL Do. ABDULLAH, S.T., M.T.
NIDN. 0018029002

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	SENIN, 29 JANUARI 2024
Pukul	:	10:00 - 12:00
Tempat	:	RUANG PRODI

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa	:	REZA HI HUKUM
NPM	:	07351811024
Judul	:	ANALISIS SENTIMEN VAKSIN KETIGA (BOOSTER) PADA TWITTER MENGGUNAKAN ALGORITMA LONG SHORT THERM MEMORY (LSTM)

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

Lakukan perbaikan sebagaimana yang diminta para penguji

Dosen Pembimbing II,

Ir. AMAL KHAIRAN, S.T., M.Eng., IPM
NIP. 197401112003121003

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	SENIN, 29 JANUARI 2024
Pukul	:	10:00 - 12:00
Tempat	:	RUANG PRODI

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa	:	REZA HI HUKUM
NPM	:	07351811024
Judul	:	ANALISIS SENTIMEN VAKSIN KETIGA (BOOSTER) PADA TWITTER MENGGUNAKAN ALGORITMA LONG SHORT THERM MEMORY (LSTM)

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

(1) Tambah jumlah dataset yg akan digunakan
(2) Jelaskan defisi analisis hasil pengolahan data di Bab 4
(3) Jelaskan hasil Confusion Matrix dgn lebar
(4) Perbaiki Kesimpulan.

7/5/24.
Acc Revisi proposal.
ABDUL MUBARAK

Dosen Penguji I,

Ir. ABDUL MUBARAK, S.Kom., M.T., IPM
NIP. 198212062014041002

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	SENIN, 29 JANUARI 2024
Pukul	:	10:00 - 12:00
Tempat	:	RUANG PRODI

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa	:	REZA HI HUKUM
NPM	:	07351811024
Judul	:	ANALISIS SENTIMEN VAKSIN KETIGA (BOOSTER) PADA TWITTER MENGGUNAKAN ALGORITMA LONG SHORT THERM MEMORY (LSTM)

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

- Ikuti format penulisan laporan
- jelaskan algoritma dan pseudocode dari parsing kata dari sebuah kalimat
- perbaiki abstrak
- buat program input biodata diri (seperti waktu seminar)

Dosen Penguji II,

ROSIHAN, S.T., M.Cs.
NIP. 197607192010121001

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	SENIN, 29 JANUARI 2024
Pukul	:	10:00 - 12:00
Tempat	:	RUANG PRODI

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa	:	REZA HI HUKUM
NPM	:	07351811024
Judul	:	ANALISIS SENTIMEN VAKSIN KETIGA (BOOSTER) PADA TWITTER MENGGUNAKAN ALGORITMA LONG SHORT THERM MEMORY (LSTM)

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

1. pelebelan data/kriteria jangan manual kecuali ada referensi
2. perlu filter kategori data sentimen negatif/positif/bukan sentimen
3. diagram diganti dan tambahkan analisis terkait diagram yang membentuk opini sentimen
4. tambahkan penjelasan implementasi MC terkait diagram aktual dan prediksi
5. implementasi semua perhitungan dari metode yang ditawarkan belum tuntas sampai dalam kodingan
6. perbaiki format penulisan dan kesimpulan

Dosen Penguji III,

ACHMAD FUAD, S.T., M.T.
NIP. 197606182005011001

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	RABU, 17 JULI 2024
Pukul	:	13:00 - 14:30
Tempat	:	RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa	:	REZA HI HUKUM
NPM	:	07351811024
Judul	:	ANALISIS SENTIMEN VAKSIN KETIGA (BOOSTER) PADA TWITTER MENGGUNAKAN ALGORITMA LONG SHORT THERM MEMORY (LSTM)

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

Segera melakukan perbaikan dari para penguji

Selasa, 13 08 - 2024

OK Acc

Dosen Pembimbing I,

SAIFUL Do. ABDULLAH, S.T., M.T.
NIDN. 0018029002

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	RABU, 17 JULI 2024
Pukul	:	13:00 - 14:30
Tempat	:	RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa	:	REZA HI HUKUM
NPM	:	07351811024
Judul	:	ANALISIS SENTIMEN VAKSIN KETIGA (BOOSTER) PADA TWITTER MENGGUNAKAN ALGORITMA LONG SHORT THERM MEMORY (LSTM)

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

- segera lakukan perbaikan sesuai permintaan para penguji

Dosen Pembimbing II,

Ir. AMAL KHAIRAN, S.T., M.Eng., IPM
NIP. 197401112003121003

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	RABU, 17 JULI 2024
Pukul	:	13:00 - 14:30
Tempat	:	RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa	:	REZA HI HUKUM
NPM	:	07351811024
Judul	:	ANALISIS SENTIMEN VAKSIN KETIGA (BOOSTER) PADA TWITTER MENGGUNAKAN ALGORITMA LONG SHORT THERM MEMORY (LSTM)

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

1. Pelajari kembali Implementasi di Python.
2. Antara Hasil dan Kesimpulan tidak sinkron.
3. Pelajari Dasar Deep Learning.
4. Uji Lagi dgn epoch 20 dan 50.

Acc Prodi
29/7/2024

Dosen Penguji I,

Ir. ABDUL MUBARAK, S.Kom., M.T., IPM
NIP. 19821206201404100 2

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	RABU, 17 JULI 2024
Pukul	:	13:00 - 14:30
Tempat	:	RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa	:	REZA HI HUKUM
NPM	:	07351811024
Judul	:	ANALISIS SENTIMEN VAKSIN KETIGA (BOOSTER) PADA TWITTER MENGGUNAKAN ALGORITMA LONG SHORT THERM MEMORY (LSTM)

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

- Ikuti format penulisan
- Pelajari cara kerja text processing
- Pelajari cara kerja dari LSTM (termasuk cara perhitungan data hasil text processing ke dalam rumus2 yang digunakan dalam LSTM)
- Buat program sederhana menampilkan Biodata dengan PHP, ddan ketentuannya data bersifat dinamis dan proses bisa berulang)

Dosen Penguji II,

ROSIHAN, S.T., M.Cs.
NIP. 197607192010121001

UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal	:	RABU, 17 JULI 2024
Pukul	:	13:00 - 14:30
Tempat	:	RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa	:	REZA HI HUKUM
NPM	:	07351811024
Judul	:	ANALISIS SENTIMEN VAKSIN KETIGA (BOOSTER) PADA TWITTER MENGGUNAKAN ALGORITMA LONG SHORT THERM MEMORY (LSTM)

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

1. revisi format penulisan, kesimpulan, analisis akhir, abstrak dan referensi yang digunakan
2. diagram hasil analisis belum tuntas, perbaiki/lengkapi dan berikan penjelasan yang tepat
3. cek kembali hasil jst dalam lstm ini,, sesuai dengan masukan saat ujian
4. catatan2 saat ujian ditindaklanjuti —

Dosen Penguji III,

ACHMAD FUAD, S.T., M.T.
NIP. 197606182005011001

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KHAIRUN TERNATE

LEMBAR ASISTENSI

Nama : Reza Hi. Hukum

NPM : 07351811024

Pembimbing I : Saiful Do. Abdullah, S.T., M.T.

Judul : Analisis Sentimen Vaksin Ketiga (*Booster*) Pada *Twitter* Menggunakan Algoritima *Long Short Therm Memory* (LSTM)

No	Tanggal	Keterangan	Paraf
1.	5-11-2021	- BAB 4 perbaiki hasil dan pembahasan	
2.	6-11-2021	- Kesimpulan - Tambahkan penjelasan grafik.	
3.	8-12-2021	pelajari algoritma siap untuk dipresentasi Ok Acc	

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KHAIRUN TERNATE

LEMBAR ASISTENSI

Nama : Reza Hi. Hukum

NPM : 07351811024

Pembimbing I : Amal Khairan, S.T., M.Eng.,

Judul : Analisis Sentimen Vaksin Ketiga (*Booster*) Pada *Twitter* Menggunakan Algoritima *Long Short Therm Memory* (LSTM)

No	Tanggal	Keterangan	Paraf
1	20/11/2023	• Abstrak blm ada → lengkapi • Tata Cara penulisan kata non Bahasa Indonesia di perbaiki	
		• Rumusan Masalah perbaiki - pakai yg sy tulis di naskah. • Sistematika penulisan	
		• Matriks jadwal pd bab 3 di hapus	
		• Bab 4, setiap gambar dan tabel pd bab 4, beri penjelasan tambahan yang menjelaskan apa yg ada pd gambar dan tabel.	
		Kalau perlu jelaskan salah satu contoh data dari tabel/gambar (lihat gambar 4-1)	

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KHAIRUN TERNATE

LEMBAR ASISTENSI

Nama : Reza Hi. Hukum

NPM : 07351811024

Pembimbing II : Amal Khairan, S.T., M.Eng.,

Judul : Analisis Sentimen Vaksin Ketiga (*Booster*) Pada *Twitter* Menggunakan Algoritima *Long Short Therm Memory* (LSTM)

No	Tanggal	Keterangan	Paraf
2.	5-12-2023	Perbaiki Abstrak.	
3	10-12-23		