

**ANALISIS KESESUAIAN KONTEKS SARAN DAN
KRITIK SEBUAH KALIMAT PADA SIMAK
UNISMUH MENGGUNAKAN ALGORITMA LSTM**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat untuk Mendapatkan
Gelar Sarjana Komputer (S.Kom) Program Studi Informatika



ROSALINDA APRILIA SARI

105841108320

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2024



UNIVERSITAS MUHAMMADIYAH MAKASSAR

FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221

Website: www.unismuh.ac.id, e-mail: unismuh@gmail.com

Website: <http://teknik.unismuh.makassar.ac.id>



PENGESAHAN

Skripsi atas nama Rosalinda Aprilia Sari dengan nomor Induk Mahasiswa 105 84 11083 20, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 104/05/A.5-VI/V/45/2024, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu tanggal 24 Agustus 2024.

Panitia Ujian :

Makassar, - 19 Safar 1446 H
24 Agustus 2024 M

1. Pengawas Umum

- a. Rektor Universitas Muhammadiyah Makassar
Dr. Ir. H. Abd. Rakhim Nasir, ST., MT, IPM
- b. Dekan Fakultas Teknik Universitas Hasanuddin
Prof. Dr. Eng. Muhammad Isran (Rani), ST., MT

2. Penguji

- a. Ketua : Dr. Ir. H. Hafid Nurwan, ST., MT
- b. Sekretaris : Dr. Ir. Ridwan Sukor, M.T.

3. Anggota

- : 1. Muhyiddin A.M. Basri, S.Kom.
- 2. Lukman, S.Kom, M.T.
- 3. Titin Wahyuni, S.Pd., M.T.

Mengetahui

Pembimbing I

Fahrim Irfhamna Rahman S.Kom., M.T

Pembimbing II

Rizki Yustiana Bakti ST., MT

Dekan



Dr. Ir. H. Nurwanaty, ST., MT., IPM.

* NBM: 205 108



UNIVERSITAS MUHAMMADIYAH MAKASSAR

FAKULTAS TEKNIK

GEDUNG MENARA IQRA LT. 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221

Website: www.unismuh.ac.id, e_mail: unismuh@gmail.com

Website: <http://teknik.unismuh.makassar.ac.id>

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : ANALISIS KESESUAIAN KONTEKS SARAN DAN KRITIK SEBUAH KALIMAT PADA SIMAK UNISMUH MENGGUNAKAN ALGORITMA LSTM

Nama : Rosalinda Aprilia Sari

Stambuk : 105 84 11083 20

Makassar, 30 Agustus 2024

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing:

Pembimbing I

Pembimbing II


Fahrir Irhamna Rahman S.Kom., MT.


Rizki Yustiana Bakti, ST., MT.

Mengetahui,

Ketua Program Studi Informatika


Muhyiddin A M Hayat, S.Kom., MT.

NBM : 1504 577

ABSTRAK

ROSALINDA APRILIA SARI. Analisis Kesesuaian Konteks Saran Dan Kritik Sebuah Kalimat Pada Simak Unismuh Menggunakan Algoritma LSTM (dibimbing oleh Fachrim Irhamna Rahman S.Kom., M.T. dan Risky Yusliana Bakti S.T., M.T.).

Dalam upaya meningkatkan pengalaman akademik dan sistem administrasi, Universitas Muhammadiyah Makassar (Unismuh) telah mengedepankan teknologi, termasuk penyediaan platform hak angket bagi mahasiswa untuk menyampaikan pandangan, saran, dan kritik mengenai berbagai aspek kampus. Namun, pengumpulan data ini sering terkendala oleh kualitas tanggapan yang kurang konstruktif dan informasi yang tidak lengkap, sehingga menghambat perbaikan yang efektif. Penelitian ini dilakukan dengan maksud mengembangkan strategi penggunaan algoritma *Long Sort Term Memory* (LSTM) untuk menganalisis konteks saran dan kritik pada platform survei kepuasan simak unismuh pada layanan dan fasilitas yang ditawarkan. Penggunaan algoritma LSTM dipilih karena kemampuan dalam mengekstraksi pola – pola kompleks dalam data teks. Hasil penelitian ini menunjukkan bahwa model LSTM, yang diuji dengan berbagai rasio data pelatihan dan validasi

Kata Kunci: Analisis Konteks, Long Short Term Memory, Natural Language Processing

ABSTRACT

Rosalinda Aprilia Sari. *Analysis of the Contextual Appropriateness of Suggestions and Criticisms in Sentences on the Simak Unismuh Platform Using LSTM Algorithm (supervised by Fachrim Irhamna Rahman S.Kom., M.T. and Risky Yusliana Bakti S.T., M.T.).*

In an effort to enhance the academic experience and administrative systems, Universitas Muhammadiyah Makassar (Unismuh) has prioritized the use of technology, including providing a feedback platform for students to express their views, suggestions, and criticisms regarding various aspects of the campus. However, the data collection process often encounters challenges due to the lack of constructive responses and incomplete information, hindering effective improvements. This study aims to develop a strategy using the Long Short Term Memory (LSTM) algorithm to analyze the context of suggestions and criticisms on the Unismuh satisfaction survey platform regarding the services and facilities offered. The LSTM algorithm was chosen for its ability to extract complex patterns in textual data. The results of this study indicate that the LSTM model, tested with various training and validation data ratio

Keywords: *Context analysis, Long Short Term Memory, Natural Language Processing*

KATA PENGANTAR

Assalamu'alaikum Wr. Wb

Segala puji syukur atas kehadiran Allah SWT, yang telah melimpahkan rahmat, hidayah, serta karunia-Nya. Shalawat serta salam dihaturkan kepada Nabi Muhammad SAW yang telah menjadi suri tauladan bagi seluruh umat manusia. Penulisan proposal skripsi ini berjudul “ANALISIS KESESUAIAN KONTEKS SARAN DAN KRITIK SEBUAH KALIMAT PADA HAK ANGET SIMAK UNISMUH MENGGUNAKAN ALGORITMA LSTM” yang merupakan hasil dedikasi beberapa waktu terakhir dapat diselesaikan dengan sebagaimana mestinya.

Dengan selesainya proposal ini penulis ingin mengucapkan terimah kasih kepada semua pihak yang telah memberikan dukungan dan bantuan selama proses penyusunan proposal ini berlangsung diantaranya:

1. Kedua orang tua, penulis mengucapkan terima kasih telah memberikan dan mengajari banyak hal,
2. Ibu Dr.Ir.Hj Nurnawati, S.T., M.T., I.P.M, selaku Dekan Fakultas Teknik.
3. Bapak Muh. Syafaat S Kuba, S.T., M.T, selaku Wakil Dekan Fakultas Teknik.
4. Bapak Muhyiddin AM Hayat S.Kom., M.T, selaku Ketua Prodi Informatika.
5. Bapak Fahrim Irhamna Rachman S.Kom., M.T, selaku Dosen Pembimbing 1 proposal.
6. Ibu Rizki Yusliana Bakti S.T.,M.T selaku Dosen Pembimbing 2 Proposal.
7. Dosen dan Staf Fakultas Teknik Universitas Muhammadiyah Makassar.
8. Keluarga tercinta yang selalu memberikan doa, dukungan, dan kasih sayang tanpa henti kepada penulis
9. Teman – teman yang telah mendampingi selama proses perkuliahan terkhusus, Lis Indriani, Rizka Adrianingsih, Ayu Andira, David Arian Virgiawan, dan Arya Wibawa Ar, terima kasih telah menemani, mendukung dan mendoakan untuk saya.

10. Temen Evha Diana yang selalu memberikan dukungan moril.
11. Teman-teman mahasiswa Informatika khususnya kelas C 2020 Fakultas Teknik, Universitas Muhammadiyah Makassar, terima kasih atas dukungan dan doanya.

Demikian laporan proposal skripsi ini, dan penulis sadar bahwa laporan ini masih banyak kekurangan di dalamnya oleh karena itu penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca atas laporan ini. Akhir kata penulis ucapkan terima kasih.

Billahi fisabililhaq, fastabiqul khairat.

Wassalamualaikum Wr.Wb.

Makassar, Agustus 2024

Penulis

Rosalinda Aprilia Sari



DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABLE.....	xi
DAFTAR LAMPIRAN.....	xii
DAFTAR ISTILAH	xiii
BAB I PENDAHULUAN.....	1
A. Latar Belakang Masalah.....	1
B. Rumusan Masalah	2
C. Tujuan Penelitian	2
D. Manfaat Penelitian	3
E. Ruang Lingkup.....	3
F. Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA.....	5
A. Landasan Teori.....	5
B. Penelitian Terkait	8
C. Kerangka Pikir	10
BAB III METODE PENELITIAN.....	12
A. Tempat dan Waktu Penelitian	12
B. Alat dan Bahan.....	12
C. Perancangan Sistem	12
D. Teknik Pengujian Sistem	16

E. Teknik Analisis Data.....	16
BAB IV	18
HASIL DAN PEMBAHASAN.....	18
A. Pengambilan Data	18
B. Labeling Data.....	18
C. Preprocessing Data.....	20
D. Pembangunan Model LSTM.....	31
E. Hasil Pengujian	39
BAB V.....	45
KESIMPULAN DAN SARAN.....	45
A. Kesimpulan	45
B. Saran.....	45
DAFTAR PUSTAKA	46
LAMPIRAN.....	50



DAFTAR GAMBAR

Gambar 1 Kerangka pikir.....	11
Gambar 2 Proses penelitian.....	13
Gambar 3 Rancangan pelatihan sistem	15
Gambar 4 Data Survei.....	18
Gambar 5 Jumlah kata dan kata -kata dalam Kosakata.....	25
Gambar 6 Ukuran MatriksPembangunan Model	26
Gambar 7 Hasil Pelatihan Model LSTM	29
Gambar 8 Dimensi Set Testing dan Training.....	31
Gambar 9 Matrix Training Model.....	33
Gambar 10 Training and Validation Loss.....	37
Gambar 11 Training and Validation Accuracy.....	38



DAFTAR TABLE

Table 1 Data Labaling	19
Table 2 Hasil Akurasi 90:10	41
Table 3 Hasil Akurasi 80:20	41
Table 4 Hasil Akurasi 70:30	41
Table 5 Hasil Prediksi	42



DAFTAR LAMPIRAN

Lampiran 1 Pengumpulan Data.....	50
Lampiran 2 Pelabelan Data.....	50
Lampiran 3 Source Code Preprocessing.....	51
Lampiran 4 Source Code LSTM.....	55
Lampiran 5 Source Code Evaluasi Model.....	56
Lampiran 6 Hasil Prediksi.....	58
Lampiran 7 Hasil Turnitin.....	62



DAFTAR ISTILAH

- Deep learning* : Geoffrey Hinton memperkenalkan *deep learning* pada tahun 2006 sebagai bentuk kecerdasan buatan yang meniru cara manusia dalam mendapatkan suatu pengetahuan khusus
- Level Leksikal : Level Leksikal adalah tingkatan dalam analisis bahasa yang berfokus pada kata-kata sebagai unit dasar makna dalam sebuah bahasa. Analisis leksikal mencakup studi mengenai kosakata (leksikon) suatu bahasa, termasuk pengertian kata, bentuk, dan penggunaannya.
- Level Morfologis : Level Morfologis adalah tingkatan dalam analisis bahasa yang berfokus pada struktur dan bentuk kata-kata. Analisis morfologis mencakup studi mengenai morfem, yaitu unit terkecil dari makna atau fungsi dalam sebuah bahasa, serta bagaimana morfem-morfem tersebut digabungkan untuk membentuk kata
- Level Semantic* : *Level Semantic* adalah ingkatan atau lapisan dalam analisis bahasa alami yang berkaitan dengan makna dan interpretasi dari teks atau ucapan
- Level Sintaksis : Level Sintaksis adalah lapisan dalam pemrosesan bahasa alami yang mencakup analisis struktur kalimat berdasarkan aturan-aturan gramatikal. Analisis ini melibatkan identifikasi elemen-elemen kalimat seperti subjek, predikat, objek, dan keterangan, serta hubungan hirarkis dan linier antara elemen-elemen tersebut

- Long short-term memory* : *Long short-term memory* atau disingkat dengan LSTM merupakan jenis arsitektur RNN yang pertama kali diperkenalkan pada tahun 1997 oleh Hochreiter dan Schmidhuber yang dirancang untuk mengatasi kendala *vanishing gradiend* pada RNN Ketika mengelolah data berurutan yang panjang hal ini menyebabkan kesulitan pada RNN untuk menangkap ketergantungan jangka Panjang
- Natural Language Processing* : *Natural Language Processing* atau NLP dimulai segera setelah *computer digital* ditemukan pada tahun 1950-an, merupakan cabang ilmu komputer yang terkait dengan kecerdasan buatan yang mempelajari interaksi antara bahasa manusia dan *computer* dengan menggabungkan prinsip – prinsip linguistic dan kecerdasan buatan
- Supervised learning* : *Supervised learning* merupakan suatu proses yang menggunakan dataset yang telah diberi anotasi, dimana setiap contoh data memiliki label atau hasil yang sudah diketahui sebelumnya
- Tokenisasi : Tokenisasi adalah proses pemecahan teks menjadi kata – kata individu atau token.

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Dalam konteks pembelajaran dan pengembangan institusi pendidikan seperti universitas semakin mengedepankan teknologi guna meningkatkan pengalaman akademik hingga sistem administrative yang dapat ditawarkan kepada mahasiswa. Salah satu upaya yang dilakukan adalah dengan menyediakan *platform* survei kepuasan, tidak terkecuali pada Universitas Muhammadiyah Makassar (Unismuh). Platform ini dapat digunakan untuk menyampaikan pandangan, pengalaman, saran, serta kritik terhadap berbagai aspek kampus termaksud kualitas layanan dan pengalaman yang diberikan. Dalam hal ini, langkah penting yang dapat dilakukan adalah mendapatkan sumber informasi berupa saran dan kritik yang relevan dari para responden.

Namun, efektifitas dari pengumpulan data tersebut sering kali terpengaruh oleh kualitas tanggapan yang diberikan oleh responden dan dalam praktiknya, terdapat kecenderungan dimana responden hanya mengisi jawaban tanpa memberikan umpan balik berupa saran dan kritikan yang konstruktif. Hal ini menyebabkan evaluasi terhadap pengalaman dan kualitas layanan dikampus menjadi tidak efisien, karena informasi yang diberikan tidak memberikan gambaran yang lengkap atau manfaat bagi perbaikan.

Untuk mengatasi permasalahan tersebut, diperlukan pendekatan yang lebih cermat dalam menganalisa konteks saran dan kritik yang disampaikan oleh mahasiswa. Salah satu pendekatan yang dapat digunakan adalah dengan memanfaatkan pengolahan *Natural Language Processing* (NLP) dan *deep learning*, seperti algoritma *Long Short Term Memory* (LSTM).

Disebut pula bahwa LSTM merupakan jaringan saraf tiruan dengan arsitektur yang fleksibel sehingga dapat diterapkan pada berbagai aplikasi yang berbeda (Setiawan et al., 2022). Dengan mengintegrasikan LSTM kedalam survei kepuasan diharapkan dapat mengembangkan model yang dapat mengekstraksi pola-pola kompleks dalam data text data sehingga dapat menentukan apakah suatu jawaban mengandung saran atau kritik yang sesuai dengan konteks evaluasi pengalaman dan pelayanan akademik di Unismuh.

Dengan melakukan penelitian menggunakan algoritma LSTM untuk menganalisis kesesuaian konteks saran dan kritik pada survei kepuasan simak Unismuh, diharapkan dapat memberikan kontribusi dalam meningkatkan efektifitas pada evaluasi pelayanan serta memperbaiki pengalaman mahasiswa di lingkungan akademik. Oleh karena itu, penelitian perlu dilakukan untuk memberikan kontribusi yang signifikan dalam penggunaan teknologi dalam mendukung proses evaluasi dan perbaikan kontinu di Unismuh

B. Rumusan Masalah

Berdasarkan pada latar belakang masalah yang telah diuraikan, dapat dirumuskan masalah sebagai berikut:

1. Bagaimana strategi penggunaan dan keberhasilan algoritma LSTM dalam menganalisis serta mengklasifikasikan konteks saran dan kritik pada survei kepuasan layanan akademik unismuh.

C. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan diatas maka penelitian ini memiliki dua tujuan sebagai berikut:

1. Bertujuan untuk mengembangkan strategi penggunaan algoritma LSTM untuk menganalisis konteks saran dan kritik serta mengevaluasi keberhasilan algoritma LSTM dalam dalam mengklasifikasikan konteks saran dan kritik guna mengevaluasi dan meningkatkan layanan mutu pendidikan.

2. Manfaat Penelitian

Dari hasil penelitian ini, penulis berharap agar manfaat penelitian ini tidak hanya berguna bagi penulis sendiri tetapi juga bagi masyarakat umum, terutama Pelajar.

Adapun manfaat yang diharapkan dari penelitian ini adalah :

1. Bagi Masyarakat

Meningkatkan partisipasi masyarakat khususnya mahasiswa dalam memberikan pendapat, pandangan, saran, serta kritik yang lebih bermakna, sehingga dapat memberikan gambaran yang lebih akurat tentang kebutuhan dan harapan mahasiswa dan dosen terhadap lingkungan akademik.

2. Bagi Mahasiswa

- a. Mendorong pengembangan metode analisis teks yang inovatif, seperti algoritma LSTM, yang dapat diterapkan dalam konteks evaluasi dan perbaikan layanan institusi pendidikan lainnya.
- b. Menjadi dasar dan acuan untuk penelitian mendatang yang berhubungan dengan algoritma LSTM

3. Ruang Lingkup

1. Penelitian ini akan difokuskan pada Universitas Muhammadiyah Makassar, dengan penekanan pada evaluasi layanan, fasilitas, serta kualitas pendidikan yang disediakan oleh pihak kampus untuk mahasiswa dan dosen
2. Penelitian ini akan memperhatikan jenis saran dan kritik yang dikumpulkan melalui platform survei kepuasan Unismuh.
3. Penelitian akan menggunakan algoritma LSTM dalam analisis sentiment

4. Sistematika Penulisan

Adapun pola umum dalam penulisan ini antara lain :

BAB I PENDAHULUAN

Dalam bab ini akan membahas latar belakang penelitian, perumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup masalah, serta sistematika

penulisan untuk memberikan gambaran terhadap konteks yang akan dilakukan dalam penelitian.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan mengulas literatur terkait, serta pendekatan analisis sentiment dan penggunaan algoritma LSTM.

BAB III METODE PENELITIAN

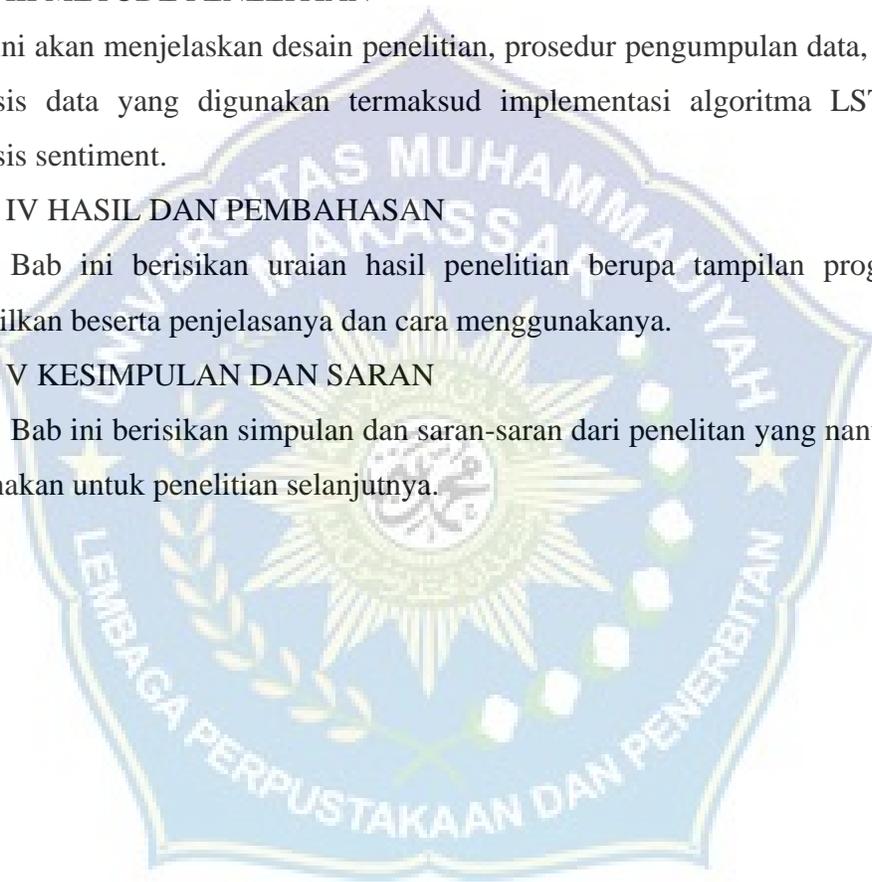
Bab ini akan menjelaskan desain penelitian, prosedur pengumpulan data, dan teknik analisis data yang digunakan termasuk implementasi algoritma LSTM dalam analisis sentiment.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisikan uraian hasil penelitian berupa tampilan program yang dihasilkan beserta penjelasannya dan cara menggunakannya.

BAB V KESIMPULAN DAN SARAN

Bab ini berisikan simpulan dan saran-saran dari penelitian yang nantinya dapat digunakan untuk penelitian selanjutnya.



BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Simak

System manajemen akademik atau yang dikenal dengan SIMAK adalah aplikasi perangkat lunak yang berfungsi untuk menyajikan informasi dan mengelola administrasi terkait kegiatan akademik. Penggunaan perangkat lunak semacam ini diharapkan dapat meningkatkan efisiensi pengolahan akademik dan memudahkan akses terhadap informasi yang dibutuhkan dengan cepat dan mudah (Monalisa et al., 2018).

Penggunaan simak tidak hanya sebagai sumber informasi bagi para mahasiswa tetapi juga memiliki dampak yang signifikan dalam pengolahan dan analisis data yang diperlukan oleh pihak administrasi. Hal ini berperan penting dalam meningkatkan kualitas layanan kepada mahasiswa (Riani et al., 2021).

2. Deep Learning

Deep learning adalah bentuk kecerdasan buatan yang meniru cara manusia dalam mendapatkan suatu pengetahuan khusus (Giarsyani, 2020).

Geoffrey Hinton memperkenalkan *deep learning* pada tahun 2006 sebagai solusi untuk mengatasi keterbatasan metode *machine learning* konvensional. Salah satu keunggulan utama *deep learning* adalah kemampuan dalam melakukan *feature engineering* secara otomatis (Rachman & Santoso, 2021).

Deep learning adalah sebuah struktur jaringan yang terdiri dari berbagai lapisan, dimana setiap lapisan memiliki sejumlah node. Setiap node tersebut berperan sebagai tempat terjadinya operasi perhitungan. *Node input* akan dikombinasikan dengan bobotnya, kemudian hasil penjumlahan ini akan melalui proses yang disebut fungsi aktivasi untuk menentukan tingkat aktivasi node tersebut (Rizki et al., 2020).

3. Supervised Learning

Supervised learning merupakan suatu proses yang menggunakan dataset yang telah diberi anotasi, dimana setiap contoh data memiliki label atau hasil yang sudah diketahui sebelumnya. Pendekatan ini bertujuan untuk melatih model computer agar dapat mengenali dan memahami pola – pola yang terdapat dalam data, serta mampu memberikan prediksi yang akurat terhadap data yang belum pernah dilihat sebelumnya (Mestika et al., 2022).

Dalam algoritma *supervised learning*, sistem akan diberikan data pelatihan yang berisi informasi *input* dan *output* yang di inginkan, sehingga memungkinkan sistem untuk mempelajari berdasarkan data yang telah ada. Sistem akan mencari pola dalam dataset tersebut, kemudian pola tersebut akan dijadikan sebagai acuan untuk kumpulan data yang ada (Abijono et al., 2021).

4. NLP

Studi mengenai NLP atau *Natural Language Processing* dimulai segera setelah computer digital ditemukan pada tahun 1950-an, dan NLP menggabungkan prinsip – prinsip linguistic dan kecerdasan buatan. Namun, kemajuan signifikan dalam beberapa tahun terakhir telah didorong oleh perkembangan dalam pembelajaran mesin (Oracle, n.d.).

NLP adalah cabang ilmu komputer yang terkait dengan kecerdasan buatan yang mempelajari interaksi antara bahasa manusia dan komputer. Dalam ruang lingkup yang lebih luas, NLP mencakup segala bentuk manipulasi komputer terhadap bahasa alami (Fajar Ramadhan et al., 2020).

Dalam tugas NLP juga melibatkan proses penguraian kata atau Bahasa menjadi komponen yang lebih sederhana. Ini melibatkan pemahaman hubungan antar komponen tersebut dan penelusuran cara komponen – komponen tersebut bekerja secara bersamaan untuk membentuk makna yang utuh (Husamuddin et al., 2020).

Terdapat dua aspek utama dalam teori pemahaman tentang sintaksis bahasa alami, pertama menjelaskan tentang bentuk fisik Bahasa, kedua merincikan

makna dari kalimat dalam bahasa tersebut. Disamping dua aspek tersebut, NLP juga memahami enam tingkatan yang berbeda.

Mulai dari level morfologis yang berkaitan dengan komponen pembentuk kata. Level leksikal yang menggabungkan morfem untuk membentuk kata – kata. Level sintaksi yang berfokus pada struktur kalimat. *Level semantic* yang berkonsentrasi pada cara konteks kalimat mempengaruhi makna secara individu. Level wacana yang meneliti bagaimana kalimat saling terhubung dalam teks hingga level pragmatis yang mempertimbangkan makna kata atau kalimat dalam konteks situasional dan pengetahuan dunia (Oktriwina, 2021).

5. LSTM

Long short-term memory atau disingkat dengan LSTM merupakan jenis arsitektur RNN yang pertama kali diperkenalkan pada tahun 1997 oleh Hochreiter dan Schmidhuber (Andiyantama et al., 2021). LSTM dirancang untuk mengatasi kendala *vanishing gradiend* pada RNN. Ketika mengelolah data berurutan yang panjang hal ini menyebabkan kesulitan pada RNN untuk menangkap ketergantungan jangka Panjang (Hanifa et al., 2021).

Di dalam LSTM terdapat tiga gates yang berperan dalam menentukan apakah suatu informasi harus dipertahankan atau dipupakan masing – masing gates tersebut adalah *input gate*, *forget gate*, dan *output gate* (Rahman et al., 2021).

Rumus *input gate* :

$$i_t = \sigma(w_t S_{t-1} + w_t x_i) \dots\dots\dots (1)$$

Dengan

w_t = bobot dari inputan gate

S_{t-1} = state sebelumnya

x_i = input pada waktu t

σ = fungsi aktivasi sigmoid (Wiranda & Sadikin, 2019)

Rumus forget gate

$$f_t = \sigma(w_f S_{t-1} + w_f x_t) \dots \dots \dots (2)$$

Dengan

w_f = bobot dari forget gate

Rumus output gate

$$O_t = \sigma(w_o S_{t-1} + w_o x_t) \dots \dots \dots (3)$$

Dengan

w_o = bobot dari output gate

LSTM juga berhasil digunakan dalam berbagai tugas sekuensial dan pemodelan bahasa. Setiap sel dalam LSTM menyimpan nilai atau keadaan (*cell state*) untuk jangka waktu tertentu, baik itu dalam jangka waktu yang pendek maupun panjang. Lstm memiliki blok memori yang menentukan nilai mana yang paling relevan sebagai keluaran berdasarkan masukan yang diberikan. (Wiranda & Sadikin, 2019).

Dengan keunggulan yang dimiliki, LSTM mampu menampilkan informasi dari memori penyimpanan yang telah berlangsung cukup lama serta dapat mengklasifikasikan dan menghilangkan informasi yang telah kadaluarsa (Algoritma, 2022).

B. Penelitian Terkait

Berikut adalah beberapa referensi jurnal penelitian terdahulu terkait analisis sentiment dan algoritma LSTM.

Penelitian yang dilakukan oleh Andi Aljabar dan Arifin A Abd Karim pada tahun 2022 dengan judul “Analisis Sentimen Menggunakan Algoritma LSTM pada Media Sosial”, yang bertujuan untuk mengevaluasi penggunaan metode LSTM dalam menentukan suatu komentar terhadap acara televisi. Hasilnya menunjukkan bahwa setelah melalui proses *training*, *validation*, dan *testing* menggunakan algoritma LSTM menyatakan nilai *loss* sebesar 12% dengan tingkat akurasi sebesar

97%. Dengan itu LSTM terbukti secara efektif dalam mengklasifikasikan sentiment dari komentar – komentar tersebut (Aljabar & Karim, 2022).

Pada tahun yang sama juga, yakni tahun 2022, Auliya Rahman Isnain, Heni Sulistiani, Bagus Miftaq Hurohman, Andi Nurkholis, dan Styawati melakukan penelitian yang berjudul “Analisis Perbandingan Algoritma LSTM dan Naïve Bayes Untuk analisis Sentimen” (Isnain et al., 2022). Hasil penelitian ini menunjukkan bahwa metode LSTM menghasilkan nilai akurasi, presisi, dan recall sebesar 83,33%, sementara Naïve bayes hanya mencapai 82% . dengan demikian LSTM memiliki nilai kinerja yang lebih unggul sebesar 1,33% dari Naïve Bayes.

Pada tahun 2022, Dioifur Rohman Alghifari, Mohammad Edi, dan Lutfi Firmansyah melakukan penelitian dengan judul “Implementasi Bidirectional LSTM Untuk Analisis Sentimen Terhadap Layana Grab Indonesia” (Alghifari et al., 2022). Hasil pengujian menunjukkan bahwa algoritma BILSTM mampu memberikan akurasi yang lebih baik dengan akurasi mencapai 91% dibandingkan algoritma LSTM biasa hanya mencapai akurasi 76%,

Pada tahun 2022, penelitian berjudul “Pendeteksi nahasa Isyarat Indonesi Secara Real Time Menggunakan *Long Short-Term Memory* (LSTM)” dilakukan oleh penulis Husna Meotia Putri, Fadlisyah, dan Wahyu Fuad menyatakan bahwa model terbaik untuk mendeteksi 10 isyarat mencapai nilai akurasi 92%, dengan menggunakan BILSTM dengan *epoch* 1000, *hiden layer* 46, dan *batch size* 64. Sedangkan model untuk mendeteksi 30 isyarat mencapai akurasi 65% dengan menggunakan 2 *layer* LSTM dengan *epoch* 500, *hiden layer* dan *batch size* 64 pada pengujian *real time* (Putri et al., n.d.).

Penelitian yang dilakukan oleh Aini Suri Talita dan Aristiawan Wiguna pada tahun 2019 dengan judul “Implementasi *Algoritma Long Short-Term Memory* (LSTM) untuk Mendeteksi Ujaran Kebencian (*Hate Speech*) Pada Kasus Pilpres 2019” (Talita & Wiguna, 2019) menunjukkan bahwa LSTM telah menunjukkan hasil

yang cukup baik dalam mendeteksi kalimat ujaran kebencian, dengan parameter recall mencapai 0.7021 pada 190 kalimat data testing dari 950 kalimat dari dataset. Meskipun demikian nilai parameter lainnya cenderung rendah yang mungkin disebabkan oleh penggunaan bahasa Informal, penulisan kata yang sering berubah dan banyaknya ejaan yang tidak benar dan konsisten.

C. Kerangka Pikir





Gambar 1 Kerangka pikir

BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

1. Tempat Penelitian

Penelitian ini dilaksanakan dalam kondisi daring dengan mengumpulkan konteks saran dan kritik pada survei kepuasan simak unismuh

2. Waktu Penelitian

Penelitian ini akan berlangsung mulai Mei hingga Juli 2024

B. Alat dan Bahan

1. Kebutuhan Hardware

a. Laptop Asus X441UA

2. Kebutuhan Software

a. Google Colab

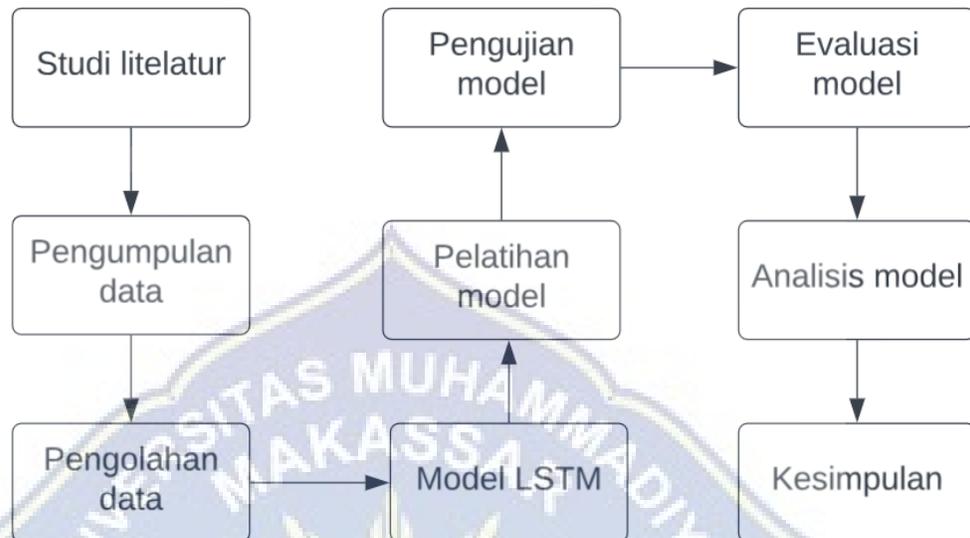
b. Python

c. Excel

C. Perancangan Sistem

Perancang sistem adalah proses rincian dalam pengembangan perangkat lunak dimana konsep dan kebutuhan pengguna diterjemahkan menjadi desain teknik yang spesifik. Tujuan dari perancangan ini adalah untuk merancang struktur dan fungsi system secara keseluruhan. Hal ini melibatkan pembuatan diagram, skema, dan spesifikasi teknis yang lebih detail guna memandu tahapan implementasi dan pengujian selanjutnya.

1. Rancangan Penelitian



Gambar 2 Proses penelitian

Berdasarkan gambar diatas penelitian ini mulai dengan melakukan studi litelatur yang berkaitan dengan analisis konteks dan LSTM serta kajian teori terkait deep learning, supervised learning, dan pengolahan bahasa alami. Selanjutnya adalah pengumpulan data yang dilakukan secara olen melalui platfrom simak unismuh. Setelah data terkumpul maka dilakukan pemrosesan dan pelabelan terhadap data yang diperoleh. Selanjutnya dilakukan tokenisasi, pelatihan, pengujian model menggunakan LSTM. Terakhir evaluasi dan analisis terhadap model serta dilakukan penarikan kesimpulan.

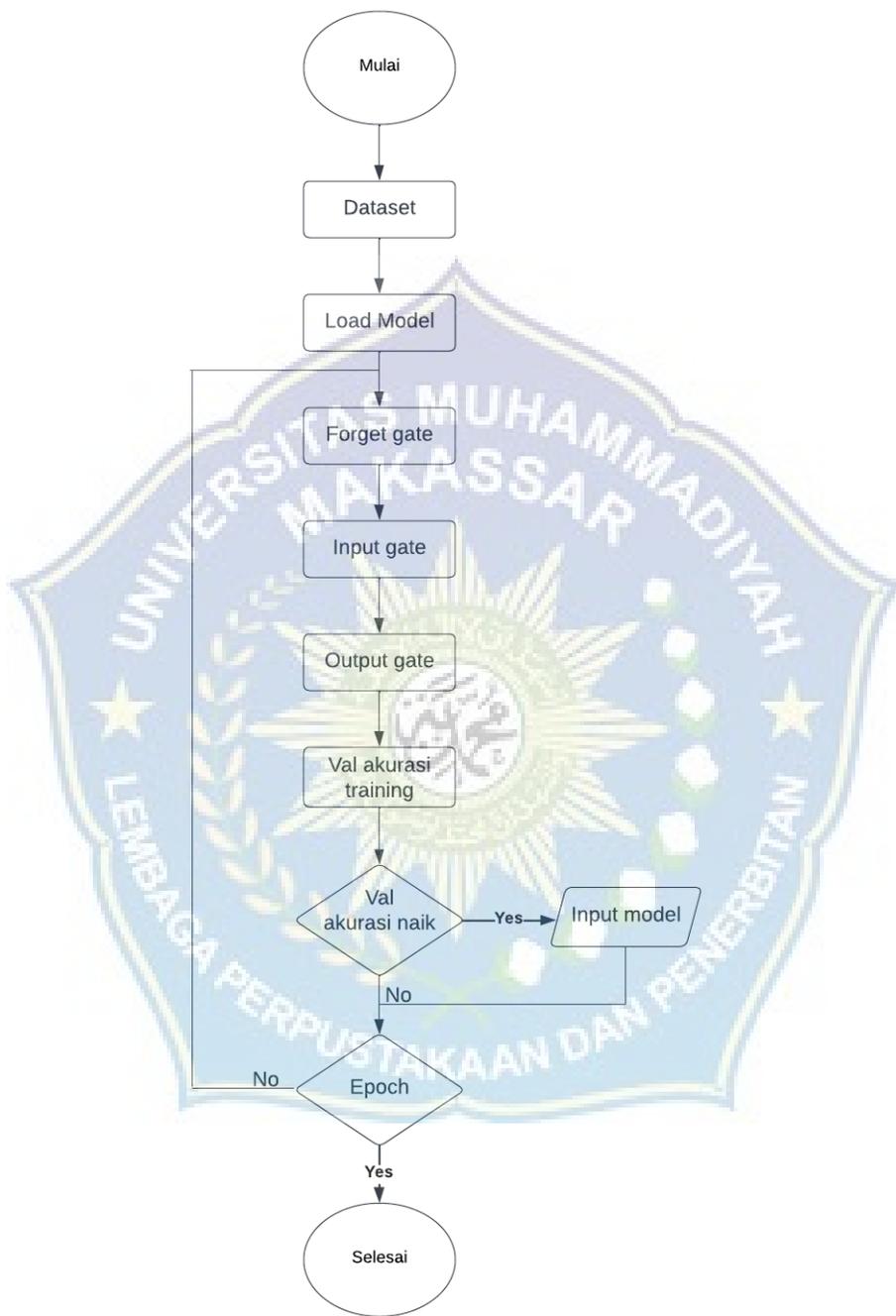
2. Pembangunan Sistem

Dalam penelitian ini, proses pembangunan sistem dimulai dengan memuat dataset yang akan digunakan diikuti dengan melakukan load model. Selanjutnya sistem akan memasuki tahap gate yang harus dilalui dalam model

LSTM, dimulai dengan forget gate untuk menentukan informasi mana yang akan disimpan dan dibuang.

Kemudian input gate akan menerima informasi berupa hidden state dari cell sebelumnya, diikuti oleh output gate yang akan menentukan hidden state mana yang akan dikirim ke cell. Setelah itu dilakukan evaluasi terhadap akurasi pelatihan, jika akurasi pelatihan tinggi model akan dimasukkan kembali namun jika untuk akurasi pelatihan rendah maka jumlah epoch dapat ditambahkan untuk meningkatkan kinerja model. Sesuai yang digambarkan pada dibawah ini,





Gambar 3 Rancangan pelatihan sistem

D. Teknik Pengujian Sistem

Langkah awal yang dilakukan dalam pengujian ini adalah mempersiapkan data yang telah dibersihkan dan dinormalkan untuk menghapus noise dan memastikan konsistensi format. Selanjutnya dataset diproses dengan melakukan tokenisasi teks dan menerapkan word embedding untuk menghasilkan representasi vector kata yang dibutuhkan algoritma LSTM.

Setelah data dipersiapkan, model algoritma LSTM dilatih menggunakan data pelatihan yang telah diproses. Dalam proses ini, parameter model dan hyperparameter disesuaikan untuk meningkatkan kinerja model. Setelah pelatihan selesai, model divalidasi dengan validasi silang untuk mengukur kinerja secara objektif.

Hasil pengujian kemudian dievaluasi dengan mengukur akurasi, presisi, *recall*, dan *F1-score* dari model. Pengujian ini membantu dalam memahami seberapa baik model dapat mengklasifikasikan sentiment dari data uji. Hasil pengujian dianalisis untuk menarik kesimpulan tentang keefektifan algoritma LSTM dalam menganalisis kesesuaian konteks saran dan kritik.

E. Teknik Analisis Data

Analisis data merupakan upaya untuk memastikan dan mengolah data hasil wawancara, observasi, dan sumber lainnya, sehingga peneliti dapat memperoleh pemahaman yang mendalam mengenai kasus yang sedang diteliti dan menyajikannya untuk temuan yang akan datang (Ahmad & Muslimah, 2021). Berikut ini adalah teknik analisis data yaitu :

1. Reduksi data

Sebelum menerapkan model, langkah awal yang harus dilakukan adalah mereduksi data guna memastikan kualitas dan relevansi data yang digunakan. Reduksi data dapat melibatkan penghapusan duplikat, penghapusan data yang tidak relevan, normalisasi data, dan tokenisasi kata.

2. Penyajian data

Setelah melakukan reduksi data, Langkah selanjutnya adalah melakukan penyajian data. Data teks perlu diubah menjadi representasi numerik yang dapat dimengerti oleh model. Salah satu pendekatan umum yang digunakan adalah teknik Word2Vec atau Glove untuk mengonversi kata – kata menjadi vector numerik. Data juga perlu dibagi menjadi data latih dan data uji dengan proporsi yang sesuai.

3. Penarikan kesimpulan

Berdasarkan hasil pengujian, kesimpulan dapat ditarik mengenai kinerja model dalam analisis system kontek saran dan kritik. Kesimpulan ini mencakup evaluasi terhadap akurasi dan konsistensi model dalam mengkalsifikasikan kontek saran dan kritik.



BAB IV

HASIL DAN PEMBAHASAN

A. Pengambilan Data

Penelitian ini memanfaatkan data yang diperoleh dari sistem manajemen Unismuh yang mencakup survei kepuasan mahasiswa dan dosen dari berbagai aspek, seperti fasilitas, layanan, dan kualitas pendidikan. Atribut yang terdapat dalam data ini mencakup id, kategori, konteks saran dan kritik, serta tanggal inser.

id	kategori	saran	tgl_inser
2	MHS	Kedepannya kami bisa diberikan dosen yang lebih produktif dan mampu beoerja sama deng	#####
3	MHS	Saya berharap pelayanan akademik mampu memberikan pelayanan yang baik kepada mahi	#####
5	MHS	Memperbaiki yang baik menjadi lebih baik lagi.	#####
8	MHS	Sebaiknya pihak kampus memantau proses penyaluran kouta internet , pihak kampus juga	#####
9	MHS	Perlunya efisiensi waktu yang baik dalam pembelajaran agar matakuliah yabg lain tidak te	#####
10	MHS	Apakah pelayanan akademik sudah memberikan pelayanan yang terbaik kepada mahasisv	#####
12	MHS	Saran saya agar proses pembelajaran secara online harus juga di selingi dengan pembelajar	#####
14	MHS	Tetap pertahankan pembelajaran yang baik dan tingkat terus-menerus kedepannya	#####
23	MHS	Semoga sarana dan prasarana kampus dapat digunakan secepatnya dan semoga kuliah onli	#####
29	MHS	Semogah pembeljaranya semakin merdeka dan bisa offlinr terutama pada saat praktikum	#####
50	MHS	1) diharapkan agar Dosen mengajar pada waktu yang tepat. 2) diharapkan agar dosen bisa n	#####
51	MHS	Dengan Melihat Keadaan Dalam Pembelajaran Daring (Online), Pembelajaran Online (Dari	#####
53	MHS	semoga dengan adanya peningkatan pelayanan baik itu dalam proses perkuliahan, pembe	#####
54	MHS	Peningkatan fasilitas kampus	#####
56	MHS	-	#####
57	MHS	Saran saya agar lebih ditingkatkan lagi proses pembelajaran daring yang lebig baik lagi	#####
59	MHS	Karena proses perkuliahan masih daring, diharapkan agar sering mengingatkan kepada mah	#####
60	MHS	Semoga para dosen bisa menjelaskan materi dengan kreatif agar mahasiswa mudah mema	#####
62	MHS	Semoga lebih baik kedepannya	#####
64	MHS	Saran sya semoga bisah offline	#####

Gambar 4 Data Survei

Data ini akan digunakan sebagai acuan dalam melakukan analisis kesesuaian kontek saran dan kritik menggunakan algoritma LSTM.

B. Labeling Data

Pada proses pelabelan ini, awalnya peneliti berencana mengambil tiga label yaitu saran, kritik, dan kategori gabungan bukan saran dan kritik. Namun, selama proses pelabelan, ditemukan bahwa jumlah entri data yang termasuk kategori bukan saran dan kritik jauh lebih banyak dibandingkan dengan entri yang

mengandung saran dan kritik. Ketidak seimbangan ini dapat menyebabkan bias dalam analisis dan mengurangi kualitas hasil yang diinginkan.

Untuk mengurangi ketimpangan data dan memastikan analisis lebih representative, peneliti memutuskan hanya fokus pada entri data yang mengandung saran dan kritik. Oleh karena itu, dari total 13.370 entri data awal, diperlukan proses seleksi untuk mengumpulkan 10.000 entri data yang relevan.

Table 1 Data Labeling

saran	label
Kedepannya kami bisa diberikan dosen yang lebih produktif dan mampu bekerja sama dengan mahasiswa menyelesaikan permasalahan dalam pertemuan covid ini. Tapi untuk dosen saat ini sudah luar biasa kok	kritik
Saya berharap pelayanan akademik mampu memberikan pelayanan yang baik kepada mahasiswa dan seluruh masyarakat Unismuh. Terutama untuk para bapak/ibu dosen sekiranya mampu memberikan metode yang baik bagi mahasiswa sehingga mahasiswa lebih aktif dan produktif dalam proses pembelajaran.	kritik
Memperbaiki yang baik menjadi lebih baik lagi.	saran
Sebaiknya pihak kampus memantau proses penyaluran kuota internet, pihak kampus juga harus membuat rancangan kesepakatan perkuliahan online yang disetujui dosen dan mahasiswa serta meningkatkan kualitas layanan akademik seperti portal online untuk kebutuhan kuliah online	kritik
Perlunya efisiensi waktu yang baik dalam pembelajaran agar matakuliah yang lain tidak terganggu	kritik
Apakah pelayanan akademik sudah memberikan pelayanan yang terbaik kepada mahasiswa ?	kritik
Saran saya agar proses pembelajaran secara online harus juga diselingi dengan pembelajaran secara offline agar para mahasiswa lebih mengerti lagi materi perkuliahan	saran
Tetap pertahankan pembelajaran yang baik dan tingkat terus-menerus kedepannya	saran
Semoga sarana dan prasarana kampus dapat digunakan secepatnya dan semoga kuliah online tidak lagi dilaksanakan	saran
Semoga pembelajarannya semakin merdeka dan bisa offline terutama pada saat praktikum	saran
1) diharapkan agar Dosen mengajar pada waktu yang tepat. 2) diharapkan agar dosen bisa menegur dan membimbing kita secara baik-baik apabila kita melakukan kesalahan " Terima kasih	saran

Proses pelabelan melibatkan peninjauan manual yang cermat untuk memastikan bahwa setiap entri dapat ditempatkan dalam kategori yang tepat berdasarkan dengan konteks saran dan kritik yang diberikan responden. Setelah melalui tahap ini, data yang awalnya berjumlah 10.000 entri menyusut menjadi 7.754 entri setelah melakukan penghapusan pada data entri yang tidak mengandung saran dan kritik.

C. Preprocessing Data

Preprocessing data merupakan langkah penting dalam pemrosesan teks untuk memastikan bahwa data yang akan digunakan dalam pelatihan model LSTM bersih, konsisten, dan siap digunakan. Proses ini melibatkan beberapa tahapan yang bertujuan untuk meningkatkan kualitas data serta mengoptimalkan kinerja model. Berikut adalah langkah – langkah preprocessing yang dilakukan dalam penelitian ini:

1. Pembersihan data

Pembersihan data bertujuan untuk menghilangkan elemen – elemen yang tidak relevan dan mengganggu analisis teks.

```
import string
def remove_punctuation(text):
    return text.translate(str.maketrans('', '',
string.punctuation))

df['ULASAN'] = df['ULASAN'].apply(remove_punctuation)
```

Program ini berhasil menghapus semua tanda baca dari teks dalam kolom ulasan pada data *frame* dengan menggunakan fungsi ‘remove_punctuation’. Setiap entri teks dalam kolom tersebut diproses untuk menghapus tanda baca seperti titik, koma, tanda seru, tanda Tanya, dan lainnya.

2. Tokenisasi

Tokenisasi adalah proses pemecahan teks menjadi kata – kata individu atau token.

```
import nltk
nltk.download('punkt')

def tokenize_text(text):
    tokens = []
    for sent in nltk.sent_tokenize(text):
        for word in nltk.word_tokenize(sent):
            if len(word) <= 0:
                continue
            tokens.append(word.lower())
    return tokens
```

Program ini menggunakan *library* NLTK untuk melakukan proses tokenisasi teks, yang terdiri dari dua tahap, pertama pemecahan teks menjadi kalimat – kalimat menggunakan ‘nltk.sent_tokenize’, dan kedua, memecahkan setiap kalimat menjadi kata – kata dengan ‘nltk.word_tokenize’ setelahnya setiap kata diubah menjadi huruf kecil dan dimasukkan ke dalam daftar ‘tokens’.

```
max_features = 500000
max_sequence_length = 50

tokenizer = Tokenizer(num_words=max_features, split=' ',
filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~', lower=True)
tokenizer.fit_on_texts(df['ULASAN'].values)
```

menetapkan parameter ‘max_feature’ ditetapkan sebanyak 500.000, yang menunjukkan jumlah maksimum kata unik yang akan digunakan oleh *tokenizer*. Sementara itu ‘max_sequence_length’ ditetapkan sebesar 50, yang

menentukan panjang maksimum dari setiap urutan teks yang akan diperoleh oleh model.

Selanjutnya, menginisialisasi sebuah objek *tokenizer* dengan beberapa pengaturan. *Tokenizer* ini akan menggunakan 500.000 kata yang paling sering muncul dalam korpus, dengan spasi sebagai pemisah antar kata. Selain itu karakter – karakter tertentu seperti tanda baca akan dihapus dari teks selama proses tokenisasi, dan semua teks akan diubah menjadi huruf kecil.

Dengan objek *tokenizer* yang telah diinisialisasi, peneliti kemudian melatih *tokenizer* ini menggunakan teks ulasan yang ada dalam dataset. Metode ‘fit_on_text’ digunakan untuk membuat indeks untuk setiap kata berdasarkan frekuensi kemunculannya dalam korpus

```
X = tokenizer.texts_to_sequences(df['ULASAN'].values)
X = pad_sequences(X, maxlen=max_sequence_length)

print('Shape dari data tensor:', X.shape)
```

Setelah *tokenizer* dilatih, dilanjutkan dengan mengubah teks ulasan menjadi urutan angka menggunakan metode ‘texts_to_sequences’. Metode ini mengonversi setiap kata dalam teks menjadi angka yang mewakili indeks kata tersebut dalam kosakata yang dihasilkan oleh *tokenizer*. Untuk memastikan bahwa semua urutan teks memiliki panjang yang sama maka peneliti menggunakan metode ‘pad_sequences’. Metode ini menyesuaikan panjang setiap urutan teks menjadi 50 kata, sesuai dengan parameter ‘max_Sequence_length’ yang telah ditetapkan. Jika urutan teks lebih pendek dari 50 kata, maka akan ditambahkan dengan *padding* (nilai nol) diawal atau akhir urutan. Sebaliknya, jika lebih panjang, maka urutan akan dipotong.

```

array([TaggedDocument(words=['dapat', 'memahami', 'keadaan', 'mahasiswa'], tags=['saran']),
      TaggedDocument(words=['diharapkan', 'agar', 'pelayanan', 'lebih', 'maju', 'lagi'], tags=['saran']),
      TaggedDocument(words=['ketiadaan', 'kantin', 'membuat', 'mahasiswa', 'tidak', 'memiliki', 'tempat', 'istirahat', 'yang', 'layak'], tags=['kritik']),
      ...
      TaggedDocument(words=['pengadaan', 'ac', 'di', 'setiap', 'ruang', 'kelas'], tags=['saran']),
      TaggedDocument(words=['jangan', 'berhenti', 'untuk', 'terus', 'belajar', 'dan', 'berusaha', 'mengasah', 'kemampuan', 'diri'], tags=['saran']),
      TaggedDocument(words=['susah', 'nyari', 'tempat', 'belajar', 'yang', 'bersih', 'di', 'kampus'], tags=['kritik']),
      dtype=object)

```

Gambar 5 Tenggged document

Berdasarkan gambar 5 diatas menunjukkan sebuah array yang berisi objek – objek ‘TeggedDocument’ yang terbagi menjadi dua bagian yaitu daftar kata dan tag. Seperti dokumen pertama mencatat sejumlah kata yang menggambarkan bagaimana seseorang dapat memahami keadaan para mahasiswa yang diberi tag saran.

3. Membangun model Doc2Vec

Model Doc2Vec merupakan teknik yang digunakan untuk menghasilkan representasi vektor yang terlatih untuk dokumen teks yang tidak hanya terdiri dari kata –kata individu namun juga dapat menangkap konteks dari keseluruhan dokumen.

```

vector_size = 20
d2v_model = Doc2Vec(dm=1, dm_mean=1,
vector_size=vector_size, window=8, min_count=1, workers=1,
alpha=0.065, min_alpha=0.065)
d2v_model.build_vocab(train_tagged)

```

Peneliti memulai dengan menetapkan berbagai parameter dan menganalisis model Doc2Vec. Model ini akan digunakan untuk menghasilkan vektor representative dari document ulasan yang dimiliki. Parameter yang diatur termasuk ukuran vektor sebesar 20, penggunaan algoritma distributor memori ($dm = 1$), dan laju pembelajaran awal ($alpha$) sebesar 0.065 disamping itu peneliti juga mengatur jumlah *thread* yang digunakan selama pelatihan menjadi satu.

Setelah parameter diatur, peneliti membangun kosakata dari data pelatihan yang sudah diberi tag menggunakan metode 'build_vocab'. Proses ini membuat model memahami kata – kata apa saja yang ada dalam korpus dan mempersiapkan model untuk pelatihan lebih lanjut.

```
for epoch in range(30):
    d2v_model.train(utils.shuffle(train_tagged),
                    total_examples=len(train_tagged), epochs=1)
    d2v_model.alpha -= 0.002
    d2v_model.min_alpha = d2v_model.alpha
```

Kemudian melatih model selama 30 *epoch*. Dimana setiap *epoch*, peneliti mengacak urutan dokumen pelatihan untuk memastikan model tidak *overfit* pada urutan tertentu. Model dilatih dengan semua dokumen dalam data pelatihan untuk setiap *epoch*. Setiap *epoch* laju, laju pembelajaran dikurangi sebesar 0.002, dan laju pembelajaran minimum disesuaikan dengan nilai *alpha* saat ini. Hal ini dilakukan untuk memastikan bahwa laju pembelajaran berkurang secara bertahap untuk membantu stabilitas pelatihan.

```
num_words = len(d2v_model.wv.key_to_index)
print("Jumlah kata dalam kosakata:", num_words)

words_in_vocab = list(d2v_model.wv.key_to_index.keys())
print("Kata-kata dalam kosakata:", words_in_vocab)
```

setelah proses pelatihan selesai, peneliti mendapatkan jumlah kata dalam kosakata model. Jumlah ini mencerminkan berapa banyak kata unik yang telah dipelajari oleh model dari data pelatihan. Peneliti juga mengumpulkan daftar kata – kata dalam kosakata dan menampilkannya. Informasi ini memberikan wawasan tentang dimensi dan konten kosakata yang digunakan oleh model untuk merepresentasikan dokumen – dokumen ulasan

```
Jumlah kata dalam kosakata: 5325
Kata-kata dalam kosakata: ['yang', 'lebih', 'mahasiswa', 'dan', 'di',
```

Gambar 6 Jumlah kata dan kata -kata dalam Kosakata

4. Matrix Embedding

Matrix embedding adalah representasi numerik dari teks yang digunakan dalam NLP. Matriks ini memetakan kata – kata atau dokumen ke dalam vektor – vektor numerik dalam ruangan berdimensi tinggi.

```
embedding_matrix = np.zeros((len(d2v_model.dv.vectors),
d2v_model.vector_size))

for i in range(len(d2v_model.dv.vectors)):
    embedding_matrix[i] = d2v_model.dv.vectors[i]

print("Ukuran matriks embedding:", embedding_matrix.shape)
print("Contoh vektor untuk dokumen pertama:",
embedding_matrix[0])
```

program ini dimulai dengan inisialisasi matriks *embedding* kosong. Matriks ini dirancang untuk menampung vektor – vektor dokumen yang dihasilkan oleh model Doc2Vec yang telah dilatih sebelumnya. Ukuran matriks ini ditentukan oleh jumlah dokumen atau ukuran vektor yang digunakan dalam model. Misalnya, jika ada 2 dokumen dan ukuran vektornya 20 maka, Matriks embedding akan memiliki ukuran (2, 20).

Berikutnya, matriks *embedding* ini di isi dengan vektor – vektor dokumen dari model Doc2Vec. Untuk setiap dokumen dalam model, program mengambil vektor dokumen tersebut dan menempatkannya pada baris yang sesuai dalam matriks *embedding*. Proses ini dilakukan dalam *loop* yang berjalan dari indeks 0 hingga jumlah dokumen dalam model Doc2Vec.

Setelah matriks *embedding* terisis penuh, program menampilkan informasi mengenai ukuran matriks dan memberikan contoh vektor untuk dokumen pertama.

```
Ukuran matriks embedding: (2, 20)
Contoh vektor untuk dokumen pertama: [-2.9239912 -1.45175827 -2.05696511 -0.09968083 1.3265425 -1.23221016
-0.74578297 0.17178792 -1.39420021 0.68175834 0.94615746 2.40266585
-3.29900527 -0.33511746 0.0836632 -0.21221231 1.19361734 0.4310289
-2.30758405 -1.61501944]
```

Gambar 7 Ukuran Matriks Pembangunan Model

Gambar 7 diatas menunjukkan proses dari hasil embedding dokumen menjadi representasi numerik. Matriks embedding berukuran 2 x 20, yang berarti dua dokumen masing – masing dipresentasikan oleh vektor dengan Panjang 20 elemen. Contoh vektor untuk dokumen pertama terdiri dari deretan angka yang mencerminkan fitur-fitur dokumen tersebut. Proses embedding ini digunakan dalam pemrosesan bahasa alami (NLP) untuk memungkinkan algoritma memahami dan menganalisis teks berdasarkan representasi numerik, memfasilitasi tugas-tugas seperti klasifikasi teks dan pencarian informasi berdasarkan kesamaan konten.

```
from keras.models import Sequential
from keras.layers import LSTM, Dense, Embedding
```

Pustaka – pustaka yang diperlukan untuk diimport antaranya, ‘sequential’, ‘LSTM’, ‘Dense’, dan ‘Embedding’ dari library. Pustaka – pustaka ini digunakan untuk membangun model *neural network*.

```
MAX_SEQUENCE_LENGTH = 50
num_unique_words = len(tokenizer.word_index) + 1
embedding_matrix = np.random.rand(num_unique_words, 20)
```

Selanjutnya, beberapa parameter didefinisikan ‘MAX_SEQUENCE_LENGTH’ ditetapkan dengan nilai 50, yang berarti panjang maksimum dari setiap urutan teks yang akan digunakan sebagai input ke dalam model. ‘num_unique_words’ dihitng dari *tokenizer* untuk

mendapatkan jumlah kata unik dalam dataset dan 'embedding_matrix' diinisialisasi secara acak dengan ukuran yang sesuai dengan jumlah kata unik dan dimensi *embedding* sebesar 20.

```
model = Sequential()
```

Model kemudian diinisialisasi menggunakan objek 'sequential'. Hal ini memungkinkan untuk membangun model secara berurutan dengan menambahkan layer – layer satu per satu.

```
model.add(Embedding(num_unique_words, 20,  
input_length=MAX_SEQUENCE_LENGTH,  
weights=[embedding_matrix], trainable=True))
```

Langkah berikutnya menambahkan *layer embedding* ke dalam model. Layer *embedding* ini menggunakan jumlah kata unik ('num_unique_words'), dimana dimensi *embedding* sebesar 20, dan panjang input sebesar 'MAX_SEQUENCE_LENGTH'. Bobot *embedding* layer di inisialisasikan dengan 'embedding_matrix' yang telah dibuat sebelumnya. Dengan 'trainable=True', bobot *embedding* ini akan diperbarui selama proses pelatihan model.

```
model.add(LSTM(50, return_sequences=False))
```

Setelah itu, layer LSTM ditambahkan ke dalam model dengan 50 unit. LSTM adalah jenis *recurrent neural network* yang mampu menangani data urutan dan mengingat informasi untuk jangka waktu yang lama. 'return_sequences=False' menunjukkan bahwa hanya output dari langkah waktu terakhir yang akan digunakan, bukan seluruh urutan.

```
model.add(Dense(2, activation="softmax"))
```

Kemudian, *layer fully connected* (Dense) dengan 2 unit *output* ditambahkan ke dalam model, menggunakan fungsi aktivasi 'softmax'. Fungsi

aktivasi ini sering digunakan untuk tugas klasifikasi dua kelas, di mana hasilnya adalah probabilitas dari setiap kelas.

```
model.summary()
model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=['acc'])
```

Model yang telah dibangun kemudian ditampilkan menggunakan ‘model.summary()’ yang memberikan ringkasan tentang struktur model, jumlah parameter, dan informasi lainnya.

Kemudian model dikompilasikan dengan menggunakan *optimizer* ‘adam’ untuk optimisasi yang efisien untuk pembelajaran mesin, *loss function* ‘binary_crossentropy’ yang umum digunakan untuk klasifikasi biner, dan matriks akurasi (‘acc’) yang akan digunakan untuk mengevaluasi kinerja model selama pelatihan.

```
def split_input(sequence):
    return sequence[:-1], sequence[1:]
sequence_example = np.array([1, 2, 3, 4, 5])
x, y = split_input(sequence_example)
print("Input:", x)
print("Output:", y)
```

Terakhir fungsi ‘split_input’ digunakan untuk mengambil urutan dan membagi semua elemen kecuali yang terakhir sebagai *inputan* (‘x’), dan semua elemen kecuali yang pertama sebagai *output* (‘y’). Contoh penggunaannya menunjukkan bagaimana fungsi ini bekerja dengan *array* contoh [1, 2, 3, 4, 5].

Model: "sequential_2"

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 50, 20)	107260
lstm_2 (LSTM)	(None, 50)	14200
dense_2 (Dense)	(None, 2)	102

Total params: 121562 (474.85 KB)
Trainable params: 121562 (474.85 KB)
Non-trainable params: 0 (0.00 Byte)

Input: [1 2 3 4]
Output: [2 3 4 5]

Gambar 8 Hasil Pelatihan Model LSTM

Berdasarkan gambar 7 diatas, model dengan nama 'sequential_2' ini terdiri dari tiga lapisan utama dimana setiap lapisan memiliki fungsi dan karakter khusus masing – masing diantaranya.

- Lapisan Embedding
Lapisan pertama ini bertanggung jawab untuk mengubah indeks kata dari input menjadi vektor berdimensi tetap. Setiap kata dalam urutan *input* dipetakan ke dalam vektor dengan panjang 20. Jika *input* memiliki panjang urutan 50, maka output dari lapisan ini akan berupa tensor dengan bentuk (batch_size, 50, 20). Jumlah total parameter yang dilatih di lapisan ini adalah 107,260.
- LSTM Layer
Lapisan kedua adlah LSTM yang dirancang untuk menangkap hubungan temporal dalam kata sequensial.LSTM menerima inputan dari lapisa *embedding* dan menghasilkan representasi dengan panjang urutan yang sama, yaitu 50, namun dengan dimensi yang berbeda, yakni 50 unit. Lapisan ini memiliki 14.200 parameter yang dilatih.

- Dense Layer

Lapisan terakhir ini adalah lapisan *full connected* yang memproses output dari lapisan LSTM untuk menghasilkan klasifikasi akhir. Lapisan ini memetakan 50 unit dari LSTM ke dalam dua kelas *output*. Jumlah parameter yang dilatih dilapisan ini adalah 102

Dengan total 121,562 parameter yang dilatih, model ini siap untuk belajar dari data dan memberikan prediksi yang akurat.

5. Pembagian Data Training Dan Data Testing

Dalam proses ini, peneliti melakukan beberapa tahapan untuk mempersiapkan data ulasan yang akan digunakan dalam pelatihan model dan pembelajaran mesin. Langkah – langkah tersebut meliputi pengubahan label menjadi format *hot-one encoding* dan pembagian dataset. Berikut adalah penjelasan lebih rinci terkait proses yang dilakukan.

```
Y = pd.get_dummies(df['LABEL']).values
```

Langkah awal yang dilakukan adalah mengubah label kategori yang terdapat dalam kolom 'LABEL' pada data ulasan menjadi format *One-Hot encoding*. Format ini merupakan teknik representasi yang mengubah label kategori menjadi vektor biner. Pada vektor ini, setiap kategori diwakili oleh sebuah vektor dengan panjang yang sama dengan jumlah kategori, dimana hanya satu elemen yang bernilai 1 (yang menunjukkan kategori tersebut) dan sisanya bernilai 0.

Missal, jika kita memiliki dua kategori yaitu saran dan kritik maka:

- Saran akan diwakili dengan vector [1, 0]
- Kritik akan diwakili dengan vector [0, 1]

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.01, random_state=42)
```

Setelah label diubah menjadi format yang sesuai, langkah berikutnya memisahkan data menjadi set pelatihan dan pengujian. Ini dilakukan menggunakan fungsi 'training_test_split' dari pustaka *Sklearn*. Dalam pembagian ini, kita menentukan bahwa 10% dari data akan digunakan sebagai data uji dan sisahnya akan digunakan sebagai data latih. Pembagian ini dilakukan secara acak namun dengan menetapkan 'random_state=42' untuk memastikan bahwa pembagian data tetap konsisten setiap kali kode dijalankan.

```
print("Shape of X_train:", X_train.shape)
print("Shape of Y_train:", Y_train.shape)
print("Shape of X_test:", X_test.shape)
print("Shape of Y_test:", Y_test.shape)
```

Setelah data dibagi, program menampilkan bentuk dimensi dari set pelatihan dan pengujian untuk memastikan bahwa pembagian telah dilakukan dengan benar. Bentuk data ini menampilkan jumlah sampel dan fitur dalam set pelatihan dan pengujian. Misalnya, hasil yang ditampilkan sebagai berikut

```
Shape of X_train: (6710, 50)
Shape of Y_train: (6710, 2)
Shape of X_test: (746, 50)
Shape of Y_test: (746, 2)
```

Gambar 9 Dimensi Set Testing dan Training

D. Pembangunan Model LSTM

Model LSTM yang digunakan dalam penelitian ini terdiri dari beberapa lapisan, masing – masing dengan peran yang spesifik untuk memproses data teks dan menghasilkan prediksi yang akurat.

1. Pelatihan Model LSTM

Pada bagian ini, program melatih model LSTM yang telah dibangun menggunakan data pelatihan dan validasi, serta menampilkan hasil matrik evaluasi selama proses pelatihan.

```
history = model.fit(X_train, Y_train, epochs=20,  
batch_size=batch_size, verbose=2, validation_data=(X_test,  
Y_test))
```

Model dilatih dengan memanggil fungsi 'model.fit()'. Fungsi ini menerima beberapa parameter penting seperti data pelatihan ('X_train' dan 'Y_train'), jumlah *epoch* ('epochs=20'), ukuran *batch* ('batch_size'), tingkat *verbosity* ('verbose=2'), dan data validasi ('validation_data=(X_test, Y_test)'). Pelatihan dilakukan selama 20 *epoch*, di mana pada setiap *epoch* model akan memproses seluruh data pelatihan dalam beberapa *batch*, kemudian dievaluasi dengan data validasi untuk memantau kinerja model secara berkala.

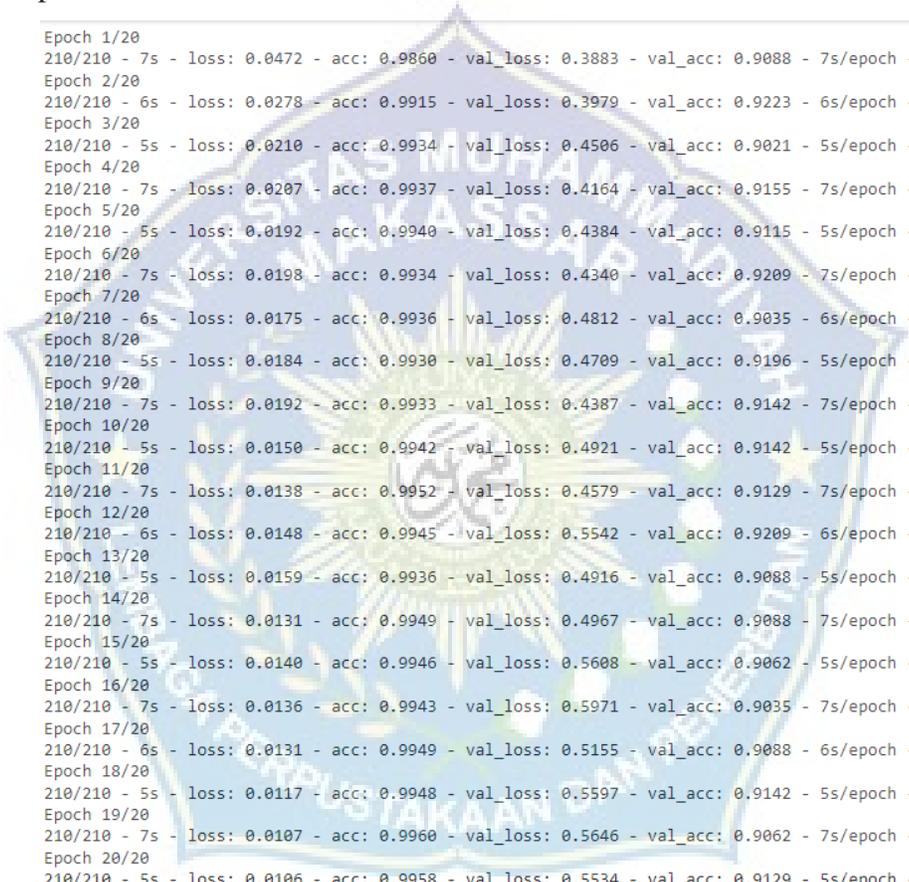
```
print(history.history.keys())
```

Setelah pelatihan selesai, hasilnya disimpan dengan objek 'history', yang merupakan dictionary berisi matrik yang dicatat selama pelatihan. Untuk melihat matrik yang ada kita dapat mencetak kunci dari dictionary.

```
val_loss = history.history['val_loss']  
val_acc = history.history['val_acc']  
print("Validation Loss:", val_loss)  
print("Validation Accuracy:", val_acc)
```

Biasanya, metrik yang dicatat mencakup 'loss' (kerugian pelatihan), 'val_loss' (kerugian validasi), 'acc' (akurasi pelatihan), dan 'val_acc' (akurasi validasi). Selanjutnya, nilai dari kerugian validasi dan akurasi validasi diekstraksi dari objek *history* dan ditampilkan. Hal ini memberikan wawasan tentang kinerja model pada data yang tidak terlihat selama pelatihan, membantu dalam mengevaluasi kemampuan generalisasi model.

Nilai ‘val_loss’ menunjukkan seberapa baik model memprediksi data validasi dalam hal kerugian. Sementara ‘val_acc’ menunjukkan presentase prediksi yang benar pada data validasi. Matrik – matrik ini penting untuk memastikan bahwa model tidak hanya belajar mengingat data pelatihan namun juga mampu menggeneralisasi ke data baru yang tidak terlihat selama proses pelatihan.



```

Epoch 1/20
210/210 - 7s - loss: 0.0472 - acc: 0.9860 - val_loss: 0.3883 - val_acc: 0.9088 - 7s/epoch - 33ms/step
Epoch 2/20
210/210 - 6s - loss: 0.0278 - acc: 0.9915 - val_loss: 0.3979 - val_acc: 0.9223 - 6s/epoch - 27ms/step
Epoch 3/20
210/210 - 5s - loss: 0.0210 - acc: 0.9934 - val_loss: 0.4506 - val_acc: 0.9021 - 5s/epoch - 26ms/step
Epoch 4/20
210/210 - 7s - loss: 0.0207 - acc: 0.9937 - val_loss: 0.4164 - val_acc: 0.9155 - 7s/epoch - 34ms/step
Epoch 5/20
210/210 - 5s - loss: 0.0192 - acc: 0.9940 - val_loss: 0.4384 - val_acc: 0.9115 - 5s/epoch - 25ms/step
Epoch 6/20
210/210 - 7s - loss: 0.0198 - acc: 0.9934 - val_loss: 0.4340 - val_acc: 0.9209 - 7s/epoch - 33ms/step
Epoch 7/20
210/210 - 6s - loss: 0.0175 - acc: 0.9936 - val_loss: 0.4812 - val_acc: 0.9035 - 6s/epoch - 27ms/step
Epoch 8/20
210/210 - 5s - loss: 0.0184 - acc: 0.9930 - val_loss: 0.4709 - val_acc: 0.9196 - 5s/epoch - 26ms/step
Epoch 9/20
210/210 - 7s - loss: 0.0192 - acc: 0.9933 - val_loss: 0.4387 - val_acc: 0.9142 - 7s/epoch - 34ms/step
Epoch 10/20
210/210 - 5s - loss: 0.0150 - acc: 0.9942 - val_loss: 0.4921 - val_acc: 0.9142 - 5s/epoch - 25ms/step
Epoch 11/20
210/210 - 7s - loss: 0.0138 - acc: 0.9952 - val_loss: 0.4579 - val_acc: 0.9129 - 7s/epoch - 33ms/step
Epoch 12/20
210/210 - 6s - loss: 0.0148 - acc: 0.9945 - val_loss: 0.5542 - val_acc: 0.9209 - 6s/epoch - 27ms/step
Epoch 13/20
210/210 - 5s - loss: 0.0159 - acc: 0.9936 - val_loss: 0.4916 - val_acc: 0.9088 - 5s/epoch - 26ms/step
Epoch 14/20
210/210 - 7s - loss: 0.0131 - acc: 0.9949 - val_loss: 0.4967 - val_acc: 0.9088 - 7s/epoch - 35ms/step
Epoch 15/20
210/210 - 5s - loss: 0.0140 - acc: 0.9946 - val_loss: 0.5608 - val_acc: 0.9062 - 5s/epoch - 25ms/step
Epoch 16/20
210/210 - 7s - loss: 0.0136 - acc: 0.9943 - val_loss: 0.5971 - val_acc: 0.9035 - 7s/epoch - 33ms/step
Epoch 17/20
210/210 - 6s - loss: 0.0131 - acc: 0.9949 - val_loss: 0.5155 - val_acc: 0.9088 - 6s/epoch - 27ms/step
Epoch 18/20
210/210 - 5s - loss: 0.0117 - acc: 0.9948 - val_loss: 0.5597 - val_acc: 0.9142 - 5s/epoch - 25ms/step
Epoch 19/20
210/210 - 7s - loss: 0.0107 - acc: 0.9960 - val_loss: 0.5646 - val_acc: 0.9062 - 7s/epoch - 34ms/step
Epoch 20/20
210/210 - 5s - loss: 0.0106 - acc: 0.9958 - val_loss: 0.5534 - val_acc: 0.9129 - 5s/epoch - 25ms/step

```

Gambar 10 Matrix Training Model

Gambar 10 tersebut menampilkan perjalanan pelatihan sebuah model pembelajaran mesin selama 20 *epoch*. Setiap epoch mencatat berbagai metrik yang digunakan untuk mengevaluasi kinerja model, baik pada data pelatihan maupun data validasi. Berikut adalah narasi mengenai proses pelatihan tersebut:

Pada awal pelatihan, di *epoch* pertama, model menunjukkan tingkat kesalahan (*loss*) sebesar 0.0472 dengan akurasi 98.60% pada data pelatihan. Namun, ketika diuji pada data validasi, tingkat kesalahan naik menjadi 0.3883, dan akurasi turun menjadi 90.88%. Ini adalah langkah awal yang menunjukkan model sedang mulai belajar pola dari data yang diberikan.

Memasuki *epoch* kedua, model menunjukkan peningkatan signifikan. *Loss* pada data pelatihan menurun menjadi 0.0278 dan akurasi meningkat menjadi 99.15%. Meskipun demikian, *validation loss* sedikit meningkat menjadi 0.3979, tetapi *validation accuracy* naik menjadi 92.23%, menunjukkan bahwa model mulai menggeneralisasi lebih baik.

Seiring berjalannya waktu, dari *epoch* ketiga hingga kelima, model terus memperbaiki diri dengan *loss* yang semakin menurun dan akurasi yang semakin tinggi pada data pelatihan. *Validation loss* dan *validation accuracy* berfluktuasi sedikit tetapi tetap dalam rentang yang menunjukkan performa yang baik.

Pada *epoch* keenam hingga kesepuluh, model mencapai stabilitas yang lebih tinggi. Pada *epoch* kedelapan, *loss* sangat rendah di 0.0184 dengan akurasi 99.42%, sementara *validation loss* mencapai 0.4709 dengan *validation accuracy* 91.96%. Ini menunjukkan model mampu mempertahankan performa yang baik bahkan pada data yang tidak terlihat selama pelatihan.

Selanjutnya, dari *epoch* kesebelas hingga kelima belas, model mempertahankan performa pelatihannya dengan *loss* yang terus menurun dan akurasi yang tetap tinggi. Pada *epoch* kelima belas, *loss* mencapai 0.0140 dan akurasi mencapai 99.64%, sementara *validation loss* dan *validation accuracy* menunjukkan hasil yang konsisten.

Menjelang akhir pelatihan, dari *epoch* keenam belas hingga dua puluh, model menunjukkan performa yang stabil. Pada *epoch* terakhir, model mencapai *loss* sangat rendah di 0.0106 dengan akurasi 99.58% pada data

pelatihan. Meskipun *validation loss* berada di 0.5534, *validation accuracy* tetap tinggi di 91.29%, menunjukkan model telah belajar pola data dengan baik tanpa mengalami *overfitting* yang signifikan.

Secara keseluruhan, proses pelatihan ini menunjukkan bahwa model berhasil mempelajari data pelatihan dengan sangat baik, dengan performa yang memuaskan juga pada data validasi. Waktu pelatihan per *epoch* berkisar antara 5 hingga 7 detik, menunjukkan efisiensi dalam proses pelatihan.

2. Visualisasi Hasil Pelatihan

Setelah melatih model LSTM untuk mengklasifikasikan teks, langkah berikutnya adalah menganalisis kinerja model selama pelatihan dari objek 'history' yang berisi matrik kinerja pada setiap *epoch* pelatihan

```
history_dict = history.history
```

Pertama, mengekstrak nilai –nilai penting dari history, termaksud nilai keraguan dan akurasi baik untuk data latih maupun validasi. Ekstraksi ini dilakukan untuk setiap *epoch*, memungkinkan untuk memvisualisasikan bagaimana kinerja model berkembang seiring waktu.

```
loss_values = history_dict['loss']  
val_loss_values = history_dict['val_loss']  
acc_values = history_dict['acc']  
val_acc_values = history_dict['val_acc']
```

Kemudian membuat *range* untuk jumlah *epoch*, yang akan digunakan sebagai sumbu x pada plot. Ini memberikan konteks waktu untuk bagaimana kinerja matrik berubah dengan setiap iterasi pelatihan.

```
epochs = range(1, len(loss_values) + 1)
```

Dengan data ini peneliti membuat dua plot terpisah dalam satu *figure* untuk visualisasi yang lebih baik. *Plot* pertama menunjukkan nilai keraguan dari data latih dan validasi. *Plot* ini membantu melihat apakah model mampu

mengurangi kesalahan prediksi selama pelatihan dan bagaimana hal ini dapat dibandingkan dengan kinerja pada data yang belum pernah dilihat sebelumnya

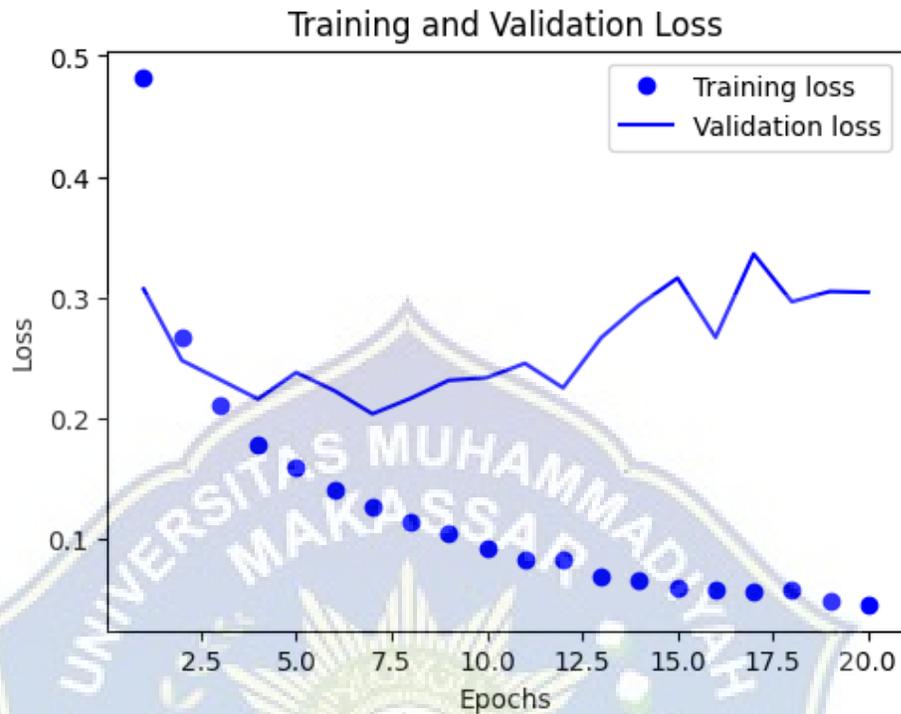
```
plt.figure(figsize=(12, 4))

plt.subplot(1, 2, 1)
plt.plot(epochs, loss_values, 'bo', label='Training loss')
plt.plot(epochs, val_loss_values, 'b', label='Validation
loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
```

Plot kedua menunjukkan nilai akurasi pada data pelatihan dan validasi. Akurasi mengukur seberapa baik model kami mengklasifikasikan ulasan dengan benar. Melalui *plot* ini, kami dapat mengamati apakah model belajar secara efektif dari data pelatihan dan seberapa baik generalisasi model pada data validasi.

```
plt.subplot(1, 2, 2)
plt.plot(epochs, acc_values, 'bo', label='Training
accuracy')
plt.plot(epochs, val_acc_values, 'b', label='Validation
accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
```

Hasil visualisasi yang ditampilkan menggambarkan performa model selama proses pelatihandan validasi dalam hal *loss* dan akurasi seperti gambar dibawa ini.

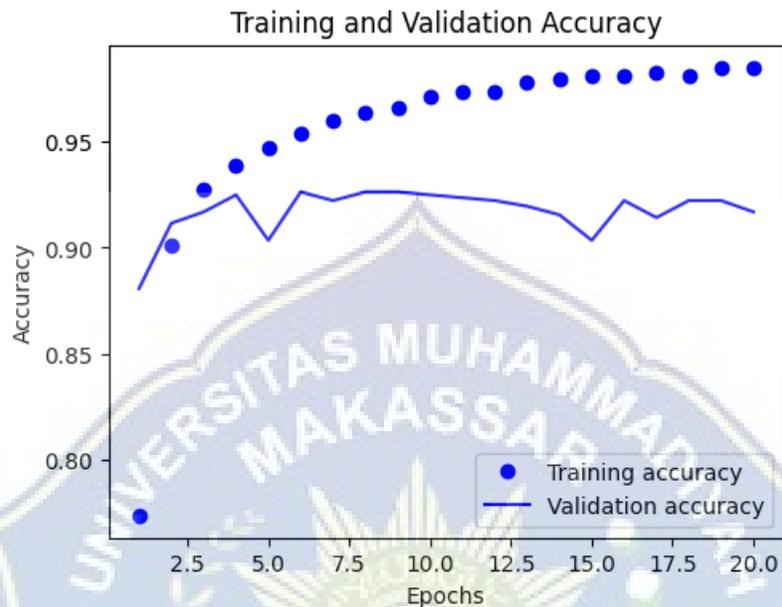


Gambar 11 Training and Validation Loss

Berdasarkan gambar 10 diatas menunjukkan perubahan *loss* selama proses pelatihan dan validasi. Pada sumbu Y, dapat dilihat bahwa nilai *loss* yang mengidentifikasi seberapa baik atau buruk model melakukan prediksi. Pada sumbu X, dapat dilihat jumlah *epoch*, yang merupakan iterasi pelatihan yang telah dilakukan

- Titik biru mewakili nilai *loss* pada data latih untuk setiap *epoch*. Pada awal pelatihan, nilai *loss* sangat tinggi, namun seiring dengan bertambahnya *epoch*, nilai *loss* mengalami penurunan dengan signifikan. Hal ini menunjukkan bahwa mode belajar dari data dan meningkatkan performanya.
- Garis biru mewakili nilai *loss* pada data validasi untuk setiap *epoch*. Garis ini juga menunjukkan penurunan di awal. Tetapi menunjukkan fluktuasi yang lebih dibandingkan training loss setelah sekitar epoch ke-5, setelah sekitar 10 epoch,

validasi loss mulai stabil tetapi sedikit meningkat diakhir, yang mungkin mengidentifikasi adanya overfitting



Gambar 12 Training and Validation Accuracy

Pada gambar 11 menunjukkan perubahan akurasi selama proses pelatihan dan validasi. Pada sumbu Y, dapat dilihat nilai akurasi yang mengidentifikasi presentase prediksi yang benar oleh model. Pada sumbu X, untuk melihat jumlah *epoch*.

- Titik biru mewakili nilai akurasi pada data pelatihan untuk setiap *epoch*. Pada awal pelatihan, nilai akurasi cukup rendah, tetapi meningkat dengan cepat dan stabil pada nilai yang tinggi setelah beberapa *epoch*. Ini menunjukkan bahwa model semakin baik dalam memprediksi dengan benar selama pelatihan berlangsung.
- Garis biru mewakili nilai akurasi pada data validasi untuk setiap *epoch*., Setelah sekitar *epoch* ke-5, *validation accuracy* cenderung stabil di sekitar 95%, menunjukkan performa yang baik pada data yang tidak dilatih. Namun terdapat

fluktuasi dalam validation accuracy setelah epoch ke-5, tetapi secara keseluruhan tetap berada pada nilai yang cukup tinggi.

Secara keseluruhan, hasil visualisasi ini menunjukkan bahwa model belajar dengan baik dari data pelatihan dan mampu melakukan generalisasi yang cukup baik terhadap data validasi. Penurunan *training loss* dan peningkatan *training accuracy* menunjukkan bahwa model semakin baik dalam mempelajari pola dari data pelatihan. Sementara itu, *validation loss* dan validation accuracy yang cukup stabil menunjukkan bahwa model memiliki kemampuan generalisasi yang baik, meskipun ada sedikit indikasi *overfitting* setelah *epoch* ke-10.

Model ini menunjukkan performa yang baik dan kemampuan generalisasi yang cukup kuat. Namun, fluktuasi yang terlihat pada *validation loss* dan *accuracy* menunjukkan bahwa ada ruang untuk lebih banyak *fine-tuning* atau penerapan teknik regularisasi untuk meningkatkan stabilitas dan performa model lebih lanjut.

E. Hasil Pengujian

Pada tahap ini model LSTM digunakan untuk mengklasifikasikan ulasan dari mahasiswa dan dosen terkait fasilitas, layanan, dan kualitas pendidikan di Unismuh terkait kategori saran dan kritik. Untuk memastikan bahwa model ini berfungsi dengan baik, beberapa matriks evaluasi yang digunakan.

1. Hasil Matrik Evaluasi

Model yang telah dilatih digunakan untuk melakukan prediksi pada data uji ('*x_text*'). Hasil prediksi ini berupa probabilitas yang menunjukkan kemungkinan setiap ulasan termaksud dalam kategori tertentu. Probabilitas tersebut kemudian dikonversi menjadi label biner (0 dan 1) dengan menggunakan threshold 0.5. dengan kata lain, jika probabilitas lebih besar dari 0.5, maka ulasan tersebut diklasifikasikan sebagai kategori positif, jika sebaliknya diklasifikasikan sebagai kategori negatif.

```
predictions = model.predict(X_test)
predicted_labels = (predictions > 0.5).astype(int)
```

Setelah memperoleh prediksi, langkah selanjutnya adalah menghitung matriks evaluasi untuk menilai performa model. Matriks – matriks ini dihitung menggunakan fungsi – fungsi *scikit-learn*, yaitu ‘f1_score’, ‘recision_score’, dan ‘recall_score’, yang semuanya menggunakan parameter ‘average=‘weighted’ untuk memberikan rata – rata pertimbangan berdasarkan jumlah setiap kelas.

```
f1 = f1_score(true_labels, predicted_labels,
             average='weighted')
precision = precision_score(true_labels, predicted_labels,
                           average='weighted')
recall = recall_score(true_labels, predicted_labels,
                    average='weighted')
```

Hasil dari perhitungan matrik evaluasi kemudian ditampilkan. Selain itu, digunakan fungsi ‘classification_report’ untuk memberikan ringkasan lebih rinci tentang performa model, termaksud *precision*, *recall*, *F1 score*, dan *support* untuk setiap kelas.

```
print(f"F1 Score: {f1}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(classification_report(true_labels, predicted_labels))
```

Untuk memberikan gambaran lebih lanjut tentang hasil prediksi, program juga menampilkan array dari label asli dan label yang diprediksi.

```
print("Array hasil prediksi:")
print(true_labels)
print(predicted_labels)
```

a. Split Data 90:10

Table 2 Hasil Akurasi 90:10

Class	Precision	Recall	F1-Score	Support
0	0.89	0.83	0.86	229
1	0.93	0.96	0.94	517
micro avg	0.92	0.92	0.92	746
macro avg	0.91	0.89	0.90	746
weighted avg	0.92	0.92	0.92	746
samples avg	0.92	0.92	0.92	746

b. Split Data 80:20

Table 3 Hasil Akurasi 80:20

Class	Precision	Recall	F1-Score	Support
0	0.92	0.88	0.90	446
1	0.95	0.97	0.96	1046
micro avg	0.94	0.94	0.94	1492
macro avg	0.93	0.92	0.93	1492
weighted avg	0.94	0.94	0.94	1492
samples avg	0.94	0.94	0.94	1492

c. Split Data 70:30

Table 4 Hasil Akurasi 70:30

Class	Precision	Recall	F1-Score	Support
0	0.91	0.93	0.91	674
1	0.96	0.96	0.96	1563
micro avg	0.95	0.95	0.95	2237
macro avg	0.94	0.94	0.94	2237

weighted avg	0.95	0.95	0.95	2237
samples avg	0.95	0.95	0.95	2237

Berdasarkan hasil evaluasi performa model dengan berbagai rasio perbandingan data pelatihan dan validasi berikut hal yang dapat dirangkum

- Dengan meningkatnya rasio data validasi (dari 10% menjadi 30%), model menunjukkan peningkatan performa, terutama dalam hal precision dan recall untuk kedua kelas.
- Dalam pembagian data 70:30, model mencapai nilai precision, recall, dan F1-Score yang tertinggi, menunjukkan bahwa model memiliki generalisasi yang lebih baik dengan data validasi yang lebih banyak.
- Semua hasil evaluasi menunjukkan bahwa model memiliki performa yang sangat baik dalam melakukan klasifikasi dengan nilai precision, recall, dan F1-Score yang tinggi, menunjukkan bahwa model mampu melakukan prediksi dengan akurat dan sensitif terhadap kedua kelas.

2. Analisis Hasil Prediksi

Untuk lebih memahami performa model, beberapa contoh prediksi dibandingkan dengan label sebenarnya. Sebagian besar prediksi model sesuai dengan label sebenarnya, hal ini dapat menunjukkan bahwa model dapat memahami konteks ulasan dengan baik. Namun, ada beberapa kasus dimana model membuat kesalahan prediksi seperti table dibawah ini.

Table 5 Hasil Prediksi

Ulasan	Label Sebenarnya	Prediksi
Hadirkan dosen dan siap dalam mengajar	kritik	saran
Fasilitas pengolahan limbah kampus tidak memadai	kritik	saran

Sistem simak perlu ditingkatkan
dikarenakan sistem sering mengalami
eror kritik saran

a. Ulasan “Hadirkan dosen dan siap dalam mengajar”

Meskipun model prediksi model berbeda dengan label sebenarnya, dalam konteks ini, prediksi dapat dianggap benar. Ulasan ini memberikan saran agar dosen selalu hadir dan siap dalam mengajar, yang lebih bersifat konstruktif dari pada kritik. Hal ini menunjukkan model mampu menangkap esensi dari ulasan yang lebih positif dan memberikan rekomendasi. Jadi meskipun tidak sesuai dengan label sebenarnya, prediksi sebagai saran tetap relevan dan dapat diterima

b. Ulasan “Fasilitas pengolahan limbah kampus tidak memadai”

Pada ulasan ini, prediksi model tentu saja salah. Ulasan ini jelas merupakan kritik terhadap fasilitas pengolahan limbah yang dianggap tidak memadai. Model tidak berhasil menangkap konteks negative dari ulasan ini, sehingga kesalahan ini menunjukkan bahwa adanya ruang untuk perbaikan dalam menangani ulasan dengan nada negative yang jelas

c. Ulasan “Sistem simak perlu ditingkatkan dikarenakan sistem sering mengalami error”

Ulasan ini sebenarnya dapat dikategorikan sebagai saran maupun kritik. Meskipun label sebenarnya adalah kritik, ulasan ini juga menyampaikan solusi, yaitu perlunya peningkatan sistem. Hal ini menunjukkan bahwa ulasan tersebut memiliki unsur saran yang signifikan, sehingga prediksi model sebagai saran tidak sepenuhnya salah. Ulasan ini mencerminkan ambiguitas yang sering ditemukan dalam teks ulasan, di mana sebuah pernyataan dapat memiliki dua interpretasi yang valid.

Hasil evaluasi secara keseluruhan menunjukkan bahwa model LSTM memiliki kemampuan yang baik dalam mengklasifikasikan ulasan. Dari 75 entri, hanya terdapat

3 entri yang tidak sesuai dengan label sebenarnya, menunjukkan tingkat kesalahan yang rendah. Kesalahan-kesalahan ini sebagian besar disebabkan oleh ambiguitas dalam teks ulasan dan kurangnya data pelatihan yang memadai untuk menangani semua variasi konteks ulasan.

Secara keseluruhan, model berhasil mengidentifikasi sebagian besar ulasan dengan benar, dan sebagian besar prediksi yang salah masih dapat dipertimbangkan sebagai interpretasi yang valid. Misalnya, dalam kasus ulasan pertama dan ketiga, prediksi yang berbeda dari label sebenarnya tetap relevan dan dapat diterima dalam konteks tertentu.



BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Penelitian ini menyatakan bahwa model LSTM menunjukkan performa yang baik dalam mengklasifikasikan konteks saran dan kritik, terutama ketika rasio data pelatihan dan validasi diatur dengan baik dan dapat disimpulkan bahwa, model menunjukkan performa baik dalam berbagai skenario pembagian data. Namun dengan meningkatkan rasio data validasi dari 10% menjadi 30%, terdapat peningkatan performa model, terutama dalam hal precision dan recall untuk kedua kelas dengan tingkat akurasi precision 91%, recall 93%, F1-Score 91% untuk kelas 0 dan precision 96%, recall 96%, F1-Score 96% untuk kelas 1. Hal ini menunjukkan bahwa model mampu menggeneralisasi dengan lebih baik dan memiliki performa optimal Ketika menggunakan lebih banyak data validasi. Disamping itu sebagian besar prediksi model sesuai dengan label sebenarnya, meskipun ada beberapa kesalahan prediksi yang disebabkan oleh ambiguitas dalam konteks saran dan kritik.

B. Saran

Berdasarkan hasil evaluasi dan analisis prediksi, saran perbaikan yang dapat diberikan adalah menggunakan data pelatihan dengan variasi konteks ulasan yang lebih banyak agar model dapat menangkap maksud ulasan dengan lebih tepat. Hal ini dilakukan untuk memastikan bahwa ulasan dengan makna ambigu dapat ditentukan secara objektif dan akurat. Selain itu, disarankan untuk mengeksplorasi penggunaan berbagai teknik yang dapat membantu mengurangi kesalahan prediksi dan menangani ambiguitas dalam teks ulasan.

DAFTAR PUSTAKA

- Abijono, H., Santoso, P., & Anggreini, N. L. (2021). Algoritma Supervised Learning Dan Unsupervised Learning Dalam Pengolahan Data. *Jurnal Teknologi Terapan: G-Tech*, 4(2), 315–318. <https://doi.org/10.33379/gtech.v4i2.635>
- Ahmad, & Muslimah. (2021). Memahami Teknik Pengolahan dan Analisis Data Kualitatif. *Proceedings*, 1(1), 173–186.
- Alghifari, D. R., Edi, M., & Firmansyah, L. (2022). *Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia Bidirectional LSTM Implementation for Sentiment Analysis Against Grab Indonesia Services*. 12, 89–99.
- Algoritma. (2022). *Long Short Term Memory Network (LSTM)*. Algoritma. <https://algorit.ma/blog/lstm-network-adalah-2022/>
- Aljabar, A., & Karim, A. A. A. (2022). *Analisis sentimen menggunakan algoritma lstm pada media sosial*. 1(3).
- Andiyantama, M. Q., Zahira, I., & Irawan, A. (2021). Prediksi Energi Listrik Kincir Angin Berdasarkan Data Kecepatan Angin Menggunakan LSTM. *JITCE (Journal of Information Technology and Computer Engineering)*, 5(01), 1–7. <https://doi.org/10.25077/jitce.5.01.1-7.2021>
- Fajar Ramadhan, D., Noertjahjono, S., & Dedy Irawan, J. (2020). PENERAPAN CHATBOT AUTO REPLY PADA WHATSAPP SEBAGAI PUSAT INFORMASI PRAKTIKUM MENGGUNAKAN ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 4, Issue 1).
- Giarsyani, N. (2020). Komparasi Algoritma Machine Learning dan Deep Learning untuk Named Entity Recognition : Studi Kasus Data Kebencanaan. *Indonesian*

Journal of Applied Informatics, 4(2), 138.

<https://doi.org/10.20961/ijai.v4i2.41317>

Hanifa, A., Fauzan, S. A., Hikal, M., & Ashfiya, M. B. (2021). Perbandingan Metode LSTM dan GRU (RNN) untuk Klasifikasi Berita Palsu Berbahasa Indonesia.

Dinamika Rekayasa, 17(1), 33. <https://doi.org/10.20884/1.dr.2021.17.1.436>

Husamuddin, H., Prasetyo, D. B., & Rustamadji, H. C. (2020). Otomatisasi Layanan Frequently Ask Questions Berbasis Natural Language Processing Pada Telegram Bot.

Telematika, 17(2), 145. <https://doi.org/10.31315/telematika.v1i1.3383>

Isnain, A. R., Sulistiani, H., Hurohman, B. M., & Nurkholis, A. (2022). Analisis Perbandingan Algoritma LSTM dan Naive Bayes untuk Analisis Sentimen. 8(2), 299–303.

Mestika, J. C., Selan, M. O., & Qadafi, M. I. (2022). Menjelajahi Teknik-Teknik Supervised Learning untuk Pemodelan Prediktif Menggunakan Python.

BIIKMA : Buletin Ilmiah Ilmu Komputer Dan Multimedia, 99(99), 216–219.

Monalisa, S., Anggara, P. P., & Kurnia, F. (2018). Analisis Kesuksesan Penerapan Sistem Administrasi Akademik Menggunakan Human Organization Technology Fit Model. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, 4(1), 36–41.

Oktriwina, A. S. (2021). *NLP: Kecerdasan Buatan yang Bantu Komputer Pahami Bahasa Manusia*. Glints. <https://glints.com/id/lowongan/natural-language-processing-adalah/>

Oracle. (n.d.). *What is natural language processing?* Oracle.

<https://www.oracle.com/id/artificial-intelligence/what-is-natural-language-processing/>

Putri, H. M., Fuadi, W., Informatika, T., Teknik, F., Malikussaleh, U., & Holistic, M.

(n.d.). *PENDETEKSIAN BAHASA ISYARAT INDONESIA SECARA REAL-TIME MENGGUNAKAN LONG.*

Rachman, F. P., & Santoso, H. (2021). Perbandingan Model Deep Learning untuk Klasifikasi Sentiment Analysis dengan Teknik Natural Language Processing. *Jurnal Teknologi Dan Manajemen Informatika*, 7(2), 103–112.

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>

Riani, E., Yonathan, J., & Oliver, L. (2021). Audit Sistem Informasi Akademik (SIMAK) Menggunakan Framework COBIT 5 di Universitas Universal, *Journal of Digital Ecosystem for Natural Sustainability (JoDENS)*. *Journal of Digital Ecosystem for Natural Sustainability (JoDENS)*, 1(2), 2798–6179.

Rizki, M., Basuki, S., & Azhar, Y. (2020). Implementasi Deep Learning Menggunakan Arsitektur Long Short Term Memory(LSTM) Untuk Prediksi Curah Hujan Kota Malang. *Jurnal Repositor*, 2(3), 331–338. <https://doi.org/10.22219/repositor.v2i3.470>

Setiawan, A. Y., Darmawiguna, I. G. M., & Pradnyana, G. A. (2022). Sentiment Summarization Evaluasi Pembelajaran Menggunakan Algoritma LSTM (long short term memory). *Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika (KARMAPATI)*, 11(2), 183–191.

Talita, A. S., & Wiguna, A. (2019). *IMPLEMENTASI ALGORITMA LONG SHORT-TERM MEMORY (LSTM) UNTUK MENDETEKSI UJARAN KEBENCIAN (HATE SPEECH) PADA KASUS PILPRES 2019*. 19(1), 37–44.

Wiranda, L., & Sadikin, M. (2019). Penerapan Long Short Term Memory pada Data Time Series untuk Memprediksi Penjualan Produk PT. Metiska Farma. *Jurnal*

Nasional Pendidikan Teknik Informatika (JANAPATI), 8(3), 184–196.



LAMPIRAN

Lampiran 1 Pengumpulan Data

id	kategori	saran	tgl insert
2	MHS	Kedepannya kami bisa diberikan dosen yang lebih produktif dan mampu beoerja sama dengan mahasiswa menyelesaikan permasalahan dalam pertemuan cov	#####
3	MHS	Saya berharap pelayanan akademik mampu memberikan pelayanan yang baik kepada mahasiswa dan seluruh masyarakat Unismuh. Terutama untuk para baj	#####
5	MHS	Memperbaiki yang baik menjadi lebih baik lagi.	#####
8	MHS	Sebaiknya pihak kampus memantau proses penyaluran kouta internet , pihak kampus juga harus membuat rancangan kesepakatan perkuliahan online yang d	#####
9	MHS	Perlunya efisiensi waktu yang baik dalam pembelajaran agar matakuliah yang lain tidak terganggu	#####
10	MHS	Apakah pelayanan akademik sudah memberikan pelayanan yang terbaik kepada mahasiswa ?	#####
12	MHS	Saran saya agar proses pembelajaran secara online harus juga di selingi dengan pembelajaran secara offline agar para mahasiswa lebih mengerti lagi materi p	#####
14	MHS	Tetap pertahankan pembelajaran yang baik dan tingkat terus-menerus kedepannya	#####
23	MHS	Semoga sarana dan prasarana kampus dapat digunakan secepatnya dan semoga kuliah online tidak lagi dilaksanakan	#####
29	MHS	Semogah pembelajarannya semakin merdeka dan bisa offiint terutama pada saat praktikum	#####
11		11 diharap jam dan hari vana sesuai denean iadwal dan tidak mengambil waktu istirahat kami untuk kuliah	

Lampiran 2 Pelabelan Data

ULASAN	LABEL
Berikan contoh nyata dan studi kasus yang memperlihatkan aplikasi dari teori dan konsep ilmu sosial dan budaya dalam kehidupan sehari-hari. Hal ini dapat membantu mahasiswa memahan	saran
menyediakan fasilitas dan sarana pendukung pembelajaran	saran
Sebaiknya mempertahankan konsistennya dalam mengajar	saran
Tolong fasilitas lebih dimaksimalkan lagi	saran
Saran saya semoga lebih ditingkatkan lagi kedisiplinan dalam proses pembelajaran	saran
menyediakan fasilitas dan sarana pendukung pembelajaran	saran
menyediakan fasilitas dan sarana pendukung pembelajaran	saran
Kedisiplinan waktu nya lebih di tingkatkan lagi	saran
sdikit saran agar pada saat mengajar kedepannya Bpk lebih bisa menjelaskan mengenai materi yg akan disampaikan atau dengan memberikan contoh kasus di dunia nyata agar mudah dipah	saran
Sebaiknya mempertahankan konsistennya dalam mengajar	saran
Sebaiknya mempertahankan konsistennya dalam mengajar	saran
Sekiranya dosen juga bisa saling mengerti terhadap kendala yang dialami mahasiswa agar lebih bertoleran.	saran
Memperbaiki kembali komunikasi agar perkuliahan jalan dengan baik	saran
Pelayanan academic perlu di tingkatkan	saran
Proses pembelajaran lebih bagus menggunakan metode yang mudah dimengerti mahasiswa	saran
Proses pembelajaran menggunakan metode yang mudah dimengerti mahasiswa	saran
semoga pendukung perangkat pembelajaran lebih memadai lagi dari sebelumnya	saran
Proses pembelajaran lebih di tingkatkan	saran

saran - Copy .xlsx

File Edit Tampilan Sisipkan Format Data Alat Bantuan

100% Rp % .0 .00 123 Courie... - 11 + B I A

	A	B
1	ULASAN	LABEL
2	Kedepanya kami bisa diberikan dosen yang lebih produktif dan mampu beoerja sama dengan mahasiswa menyelesaikan permasalahan dalam pertemuan covid ini. Tapi untuk dosen saat ini sudah lu	kritik
3	Saya berharap pelayanan akademik mampu memberikan pelayanan yang baik kepada mahasiswa dan seluruh masyarakat Unismuh. Terutama untuk para bapak/ibu dosen sekiranya mampu member	kritik
4	Sebaiknya pihak kampus memantau proses penyaluran kouta internet, pihak kampus juga harus membuat rancangan kesepakatan perkuliahan online yang disetujui dosen dan mahasiswa serta men	kritik
5	Perlu nya efisiensi waktu yang baik dalam pembelajaran agar matakuliah yang lain tidak terganggu	kritik
6	Apakah pelayanan akademik sudah memberikan pelayanan yang terbaik kepada mahasiswa ?	kritik
7	Dengan Melihat Keadaan Dalam Pembelajaran Daring (Online), Pembelajaran Online (Daring) Sangat Tidak Baik Bagi Kami Untuk Fokus belajar, Karena Banyak-Nya Kendala2 Yang Kami Alamin Seperti	kritik
8	Peningkatan fasilitas kampus	kritik
9	Pendidikan agama Islam harus di tingkatkan	kritik
10	Agar kedepannya buku referensi di perpustakaan diperbanyak lagi sesuai dengan buku yang dipakai pada tahun ajaran yang sedang digunakan	kritik
11	Kepada dosen sekiranya dalam metode pembelajarannya harus di tingkatkan lagi keberagaman nya agar mahasiswa nya lebih tertarik apalagi dalam hal kuliah online ini	kritik
12	Jangan mempersulit mahasiswa ketika dalam pelayanan	kritik
13	Konsisten terhadap jadwal yang telah di tentukan	kritik
14	Menurut saya yang harus dilakukan untuk meningkatkan pelayanan akademik di program studi adalah dengan cara membuat ruang belajar dengan nyaman. Seperti dilengkapi penyejuk udara serta	kritik
15	Sebaiknya dalam pembelajaran daring dosen dapat mengerti tidak semua mahasiswa dapat mengakses internet yang baik, dan dosen dapat mengerti jika ada kendala dalam keterlambatan mengirin	kritik
16	Diupayakan dosen masuk di jam mata kuliah dengan tepat waktu dan tidak mengganti jadwal jam kuliah	kritik
17	Pihak akademik lebih fokus pada peningkatan kualitas mahasiswa	kritik
18	Pelayanan perlu di tingkatkan	kritik
19	Fasilitas nya harus bisa lebih baik lagi	kritik

Lampiran 3 Source Code Preprocessing

```
[ ] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[ ] import pandas as pd
import numpy as np
from tqdm import tqdm
from keras.preprocessing.text import Tokenizer
tqdm.pandas(desc="progress-bar")
from gensim.models import Doc2Vec
from sklearn import utils
from sklearn.model_selection import train_test_split
from keras.preprocessing.sequence import pad_sequences
import gensim
from sklearn.linear_model import LogisticRegression
from gensim.models.doc2vec import TaggedDocument
import re
import seaborn as sns
import matplotlib.pyplot as plt

[ ] # Load data from Excel file
df = pd.read_excel('/content/drive/MyDrive/Rosalinda - LSTM/saran.xlsx', sheet_name="Sheet1") # Replace 'path_to_your_excel_file.xlsx' with your actual path
df = df[['ULASAN', 'LABEL']] # Selecting relevant columns
df = df[df['ULASAN'].notnull()] # Dropping rows with null 'ULASAN' values
df.rename(columns={'ULASAN': 'ULASAN'}, inplace=True) # Rename 'ULASAN' to 'ULASAN' for consistency

df.head()
```

	ULASAN	LABEL
0	Kedepanya kami bisa diberikan dosen yang lebih...	kritik
1	Saya berharap pelayanan akademik mampu memberi...	kritik
2	Memperbaiki yang baik menjadi lebih baik lagi.	saran
3	Sebaiknya pihak kampus memantau proses penyalu...	kritik
4	Perlu nya efisiensi waktu yang baik dalam pembe...	kritik

```
[ ] df.shape
```

```
(7456, 2)
```

```
[ ] # Mengatur ulang indeks baris
df.index = range(len(df))

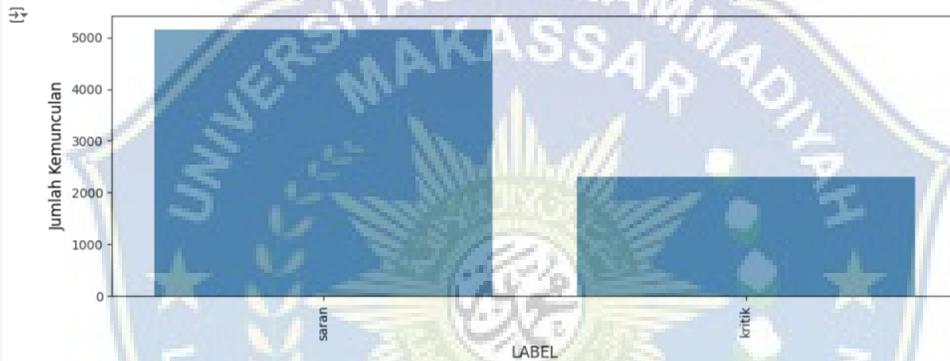
# Menghitung total jumlah kata dalam kolom 'ULASAN'
total_words = df['ULASAN'].apply(lambda x: len(x.split(' '))).sum()

print("Total jumlah kata dalam semua ulasan:", total_words)
```

```
Total jumlah kata dalam semua ulasan: 82685
```

```
[ ] # Menghitung jumlah kemunculan setiap nilai dalam kolom 'LABEL'
cnt_pro = df['LABEL'].value_counts()

# Menggambar diagram batang menggunakan Seaborn
plt.figure(figsize=(12, 4))
sns.barplot(x=cnt_pro.index, y=cnt_pro.values, alpha=0.8)
plt.ylabel('Jumlah Kemunculan', fontsize=12)
plt.xlabel('LABEL', fontsize=12)
plt.xticks(rotation=90)
plt.show()
```



```
[ ] def print_message(index):
    example = df.iloc[index][['ULASAN', 'LABEL']].values
    if len(example) > 0:
        print('ULASAN:', example[0])
        print('LABEL:', example[1])

# Menggunakan fungsi print_message() dengan indeks tertentu
print_message(12)
```

```
ULASAN: semoga dengan adanya peningkatan pelayanan baik itu dalam proses perkuliahan, pembelajaran ataupun dari pelayanan akademik akan menjadi jauh
LABEL: saran
```

```
import string
def remove_punctuation(text):
    return text.translate(str.maketrans('', '', string.punctuation))

# Menghapus tanda baca dari kolom ULASAN
df['ULASAN'] = df['ULASAN'].apply(remove_punctuation)
```

```
import nltk
# Download the 'punkt' resource
nltk.download('punkt')

# Tokenisasi teks menggunakan nltk
def tokenize_text(text):
    tokens = []
    for sent in nltk.sent_tokenize(text):
        for word in nltk.word_tokenize(sent):
            if len(word) <= 0:
                continue
            tokens.append(word.lower())
    return tokens

# Memisahkan data menjadi train dan test
train, test = train_test_split(df, test_size=0.01, random_state=42)
```

```
# TaggedDocument untuk train dan test set
train_tagged = train.apply(
    lambda r: TaggedDocument(words=tokenize_text(r['ULASAN']), tags=[r.LABEL]), axis=1)
test_tagged = test.apply(
    lambda r: TaggedDocument(words=tokenize_text(r['ULASAN']), tags=[r.LABEL]), axis=1)

# Pengaturan tokenizer
max_features = 500000 # Jumlah maksimum kata yang akan digunakan
max_sequence_length = 50 # Panjang maksimum setiap teks

tokenizer = Tokenizer(num_words=max_features, split=' ', filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~', lower=True)
tokenizer.fit_on_texts(df['ULASAN'].values)

# Konversi teks ke dalam urutan angka (sequences)
X_train = tokenizer.texts_to_sequences(train['ULASAN'].values)
X_train = pad_sequences(X_train, maxlen=max_sequence_length)

X_test = tokenizer.texts_to_sequences(test['ULASAN'].values)
X_test = pad_sequences(X_test, maxlen=max_sequence_length)

print('Found %s unique tokens.' % len(tokenizer.word_index))

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
Found 5362 unique tokens.

# Konversi teks ke dalam urutan angka (sequences)
X = tokenizer.texts_to_sequences(df['ULASAN'].values)
X = pad_sequences(X, maxlen=max_sequence_length)

print('Shape dari data tensor:', X.shape)

Shape dari data tensor: (7456, 50)

#train_tagged.values
train_tagged.values

array([TaggedDocument(words=['dapat', 'memahami', 'keadaan', 'mahasiswa'], tags=['saran']),
      TaggedDocument(words=['diharapkan', 'agar', 'pelayanan', 'lebih', 'maju', 'lagi'], tags=['saran']),
      TaggedDocument(words=['ketiadaan', 'kantin', 'membuat', 'mahasiswa', 'tidak', 'memiliki', 'tempat', 'istirahat', 'yang', 'layak'], tags=
['kritik']),
      ...,
      TaggedDocument(words=['pengadaan', 'ac', 'di', 'setiap', 'ruang', 'kelas'], tags=['saran']),
      TaggedDocument(words=['jangan', 'berhenti', 'untuk', 'terus', 'belajar', 'dan', 'berusaha', 'mengasah', 'kemampuan', 'diri'], tags=['saran']),
      TaggedDocument(words=['susah', 'nyari', 'tempat', 'belajar', 'yang', 'bersih', 'di', 'kampus'], tags=['kritik']),
      dtype=object)

[ ] from gensim.models.doc2vec import Doc2Vec, TaggedDocument

# Ubah ukuran vektor (vector_size) sesuai kebutuhan Anda
vector_size = 20

# Inisialisasi model Doc2Vec
d2v_model = Doc2Vec(dm=1, dm_mean=1, vector_size=vector_size, window=8, min_count=1, workers=1, alpha=0.065, min_alpha=0.065)

[ ] # Membangun kosakata dari tagged documents pada data pelatihan
train_tagged = [TaggedDocument(words=tokenize_text(row['ULASAN']), tags=[row['LABEL']]) for index, row in train.iterrows()]
d2v_model.build_vocab(train_tagged)

[ ] # Bangun vocab dari train_tagged
d2v_model.build_vocab(train_tagged)

# Latih model
for epoch in range(30):
    d2v_model.train(utils.shuffle(train_tagged), total_examples=len(train_tagged), epochs=1)
    d2v_model.alpha -= 0.002 # Reduksi alpha setiap epoch
    d2v_model.min_alpha = d2v_model.alpha # Tetapkan min_alpha sesuai alpha saat ini

WARNING:gensim.models.KeyedVectors:sorting after vectors have been allocated is expensive & error-prone

[ ] print(d2v_model)

Doc2Vec<dm/m,d20,n5,w8,s0.001>
```

```
[ ] # Mendapatkan jumlah kata dalam kosakata
num_words = len(d2v_model.wv.key_to_index)
print("Jumlah kata dalam kosakata:", num_words)

# Mengakses kata-kata dalam kosakata
words_in_vocab = list(d2v_model.wv.key_to_index.keys())
print("Kata-kata dalam kosakata:", words_in_vocab)
```

Jumlah kata dalam kosakata: 5325
Kata-kata dalam kosakata: ['yang', 'lebih', 'mahasiswa', 'dan', 'di', 'dosen', 'tidak', 'untuk', 'dalam', 'pembelajaran', 'baik', 'semoga', 'bisa', 'l

```
[ ] # Inialisasi matriks embedding kosong
embedding_matrix = np.zeros((len(d2v_model.dv.vectors), d2v_model.vector_size))

# Mengisi matriks embedding dengan vektor-vektor dokumen dari model Doc2Vec
for i in range(len(d2v_model.dv.vectors)):
    embedding_matrix[i] = d2v_model.dv.vectors[i]

# Contoh penggunaan matriks embedding
print("Ukuran matriks embedding:", embedding_matrix.shape)
print("Contoh vektor untuk dokumen pertama:", embedding_matrix[0])
```

Ukuran matriks embedding: (2, 20)
Contoh vektor untuk dokumen pertama: [-2.41089797 -1.83616626 -1.99523568 1.01581132 -0.03091831 -1.13246322
-0.8185643 -0.04435334 -1.75505257 -0.66514939 2.00743747 0.99225533
-3.88084745 -0.71095061 0.38927159 -0.92159224 0.86452615 0.27829656
-2.26162648 -1.13632429]

```
[ ] from keras.models import Sequential
from keras.layers import LSTM, Dense, Embedding
import numpy as np

# Definisikan panjang maksimum urutan
MAX_SEQUENCE_LENGTH = 50

# Definisikan jumlah kata unik
num_unique_words = len(tokenizer.word_index) + 1

# Pastikan bahwa embedding_matrix memiliki bentuk yang sesuai
embedding_matrix = np.random.rand(num_unique_words, 20)

# Inialisasi model Sequential
model = Sequential()

# Menambahkan lapisan Embedding dengan bobot yang sesuai
model.add(Embedding(num_unique_words, 20, input_length=MAX_SEQUENCE_LENGTH, weights=[embedding_matrix], trainable=True))

# Menambahkan lapisan LSTM
model.add(LSTM(50, return_sequences=False))

# Menambahkan lapisan Dense untuk output
model.add(Dense(2, activation="softmax"))

# Menampilkan ringkasan model
model.summary()

# Kompilasi model
model.compile(optimizer="adam", loss="binary_crossentropy", metrics=['acc'])

# Contoh pemanggilan fungsi split_input
def split_input(sequence):
    return sequence[:-1], sequence[1:]

# Contoh penggunaan split_input
sequence_example = np.array([1, 2, 3, 4, 5])
x, y = split_input(sequence_example)
print("Input:", x)
print("Output:", y)
```

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 50, 20)	107260
lstm (LSTM)	(None, 50)	14200
dense (Dense)	(None, 2)	102

=====
Total params: 121562 (474.85 KB)
Trainable params: 121562 (474.85 KB)
Non-trainable params: 0 (0.00 Byte)
=====
Input: [1 2 3 4]
Output: [2 3 4 5]

```

Y = pd.get_dummies(df['LABEL']).values
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.01, random_state=42)

print("Shape of X_train:", X_train.shape)
print("Shape of Y_train:", Y_train.shape)
print("Shape of X_test:", X_test.shape)
print("Shape of Y_test:", Y_test.shape)

```

Shape of X_train: (7381, 50)
 Shape of Y_train: (7381, 2)
 Shape of X_test: (75, 50)
 Shape of Y_test: (75, 2)

Lampiran 4 Source Code LSTM

```

# Melatih model dengan data validasi
history = model.fit(X_train, Y_train, epochs=20, batch_size=32, verbose=2, validation_data=(X_test, Y_test))

# Mendapatkan histori pelatihan
print(history.history.keys())

# Menampilkan val_loss dan val_accuracy
val_loss = history.history['val_loss']
val_acc = history.history['val_acc']
print("Validation Loss:", val_loss)
print("Validation Accuracy:", val_acc)

```

Epoch 1/20
 231/231 - 7s - loss: 0.4850 - acc: 0.7689 - val_loss: 0.2141 - val_acc: 0.9467 - 7s/epoch - 29ms/step
 Epoch 2/20
 231/231 - 4s - loss: 0.2484 - acc: 0.9092 - val_loss: 0.1464 - val_acc: 0.9600 - 4s/epoch - 19ms/step
 Epoch 3/20
 231/231 - 4s - loss: 0.1955 - acc: 0.9316 - val_loss: 0.1443 - val_acc: 0.9600 - 4s/epoch - 19ms/step
 Epoch 4/20
 231/231 - 4s - loss: 0.1744 - acc: 0.9412 - val_loss: 0.1417 - val_acc: 0.9467 - 4s/epoch - 18ms/step
 Epoch 5/20
 231/231 - 4s - loss: 0.1514 - acc: 0.9492 - val_loss: 0.1426 - val_acc: 0.9333 - 4s/epoch - 16ms/step
 Epoch 6/20
 231/231 - 4s - loss: 0.1399 - acc: 0.9522 - val_loss: 0.1243 - val_acc: 0.9467 - 4s/epoch - 17ms/step

Epoch 7/20
 231/231 - 4s - loss: 0.1263 - acc: 0.9591 - val_loss: 0.1153 - val_acc: 0.9600 - 4s/epoch - 18ms/step
 Epoch 8/20
 231/231 - 4s - loss: 0.1113 - acc: 0.9646 - val_loss: 0.1046 - val_acc: 0.9600 - 4s/epoch - 19ms/step
 Epoch 9/20
 231/231 - 5s - loss: 0.1103 - acc: 0.9631 - val_loss: 0.1351 - val_acc: 0.9467 - 5s/epoch - 20ms/step
 Epoch 10/20
 231/231 - 4s - loss: 0.0949 - acc: 0.9703 - val_loss: 0.1115 - val_acc: 0.9600 - 4s/epoch - 19ms/step
 Epoch 11/20
 231/231 - 4s - loss: 0.0888 - acc: 0.9702 - val_loss: 0.1236 - val_acc: 0.9600 - 4s/epoch - 18ms/step
 Epoch 12/20
 231/231 - 4s - loss: 0.0780 - acc: 0.9740 - val_loss: 0.1456 - val_acc: 0.9467 - 4s/epoch - 19ms/step
 Epoch 13/20
 231/231 - 4s - loss: 0.0712 - acc: 0.9764 - val_loss: 0.1486 - val_acc: 0.9600 - 4s/epoch - 18ms/step
 Epoch 14/20
 231/231 - 4s - loss: 0.0760 - acc: 0.9744 - val_loss: 0.1413 - val_acc: 0.9467 - 4s/epoch - 19ms/step
 Epoch 15/20
 231/231 - 4s - loss: 0.0728 - acc: 0.9748 - val_loss: 0.1058 - val_acc: 0.9733 - 4s/epoch - 17ms/step
 Epoch 16/20
 231/231 - 4s - loss: 0.0603 - acc: 0.9794 - val_loss: 0.1336 - val_acc: 0.9600 - 4s/epoch - 16ms/step
 Epoch 17/20
 231/231 - 4s - loss: 0.0567 - acc: 0.9795 - val_loss: 0.1499 - val_acc: 0.9600 - 4s/epoch - 16ms/step
 Epoch 18/20
 231/231 - 4s - loss: 0.0530 - acc: 0.9823 - val_loss: 0.1708 - val_acc: 0.9600 - 4s/epoch - 17ms/step
 Epoch 19/20
 231/231 - 5s - loss: 0.0514 - acc: 0.9817 - val_loss: 0.1674 - val_acc: 0.9600 - 5s/epoch - 19ms/step

Epoch 20/20
 231/231 - 4s - loss: 0.0528 - acc: 0.9816 - val_loss: 0.2168 - val_acc: 0.9333 - 4s/epoch - 19ms/step
 dict_keys(['loss', 'acc', 'val_loss', 'val_acc'])
 Validation Loss: [0.21413564682006836, 0.14643655717372894, 0.14434942603111267, 0.14171966910362244, 0.14262758195400238, 0.1243191510438919, 0.11533333333333333, 0.10463655717372894, 0.1351111111111111, 0.1115111111111111, 0.1236111111111111, 0.1456111111111111, 0.1486111111111111, 0.1413111111111111, 0.1058111111111111, 0.1336111111111111, 0.1499111111111111, 0.1708111111111111, 0.1674111111111111, 0.2168111111111111]
 Validation Accuracy: [0.9466666579246521, 0.9599999785423279, 0.9599999785423279, 0.9466666579246521, 0.933333333069763, 0.9466666579246521, 0.9599999785423279, 0.960000011912603, 0.960000011912603, 0.960000011912603, 0.960000011912603, 0.960000011912603, 0.960000011912603, 0.960000011912603, 0.960000011912603, 0.960000011912603, 0.960000011912603, 0.960000011912603, 0.960000011912603, 0.960000011912603]

```

# Save the model
model.save('/content/drive/MyDrive/Rosalinda - LSTM/my_model.h5') # Saves the model to a file named 'my_model.h5'

```

```

# Mendapatkan histori pelatihan
history_dict = history.history

# Ekstrak nilai untuk setiap metrik
loss_values = history_dict['loss']
val_loss_values = history_dict['val_loss']
acc_values = history_dict['acc']
val_acc_values = history_dict['val_acc']

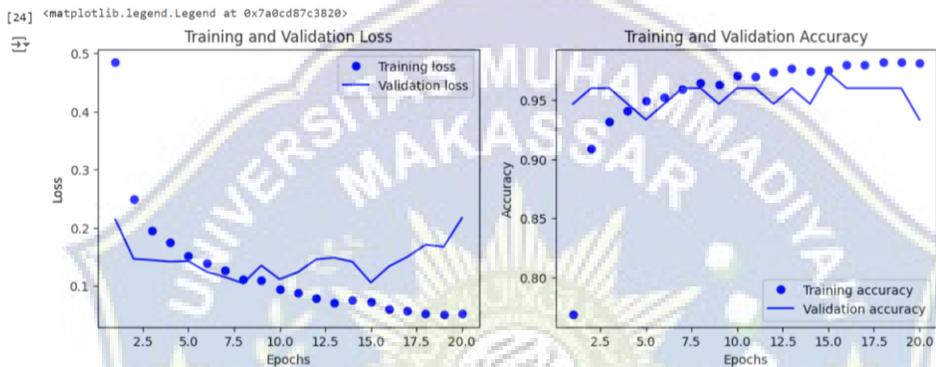
```

```
[24] # Buat range untuk jumlah epoch
epochs = range(1, len(loss_values) + 1)

# Plot Loss
plt.figure(figsize=(12, 4))

plt.subplot(1, 2, 1)
plt.plot(epochs, loss_values, 'bo', label='Training loss')
plt.plot(epochs, val_loss_values, 'b', label='Validation loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

# Plot Accuracy
plt.subplot(1, 2, 2)
plt.plot(epochs, acc_values, 'bo', label='Training accuracy')
plt.plot(epochs, val_acc_values, 'b', label='Validation accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
```



Lampiran 5 Source Code Evaluasi Model

```
[25] from sklearn.metrics import classification_report, f1_score, precision_score, recall_score

# Melakukan prediksi
predictions = model.predict(X_test)
predicted_labels = (predictions > 0.5).astype(int) # Konversi probabilitas menjadi label biner (0 atau 1)
true_labels = Y_test

# Menghitung metrik evaluasi tambahan
f1 = f1_score(true_labels, predicted_labels, average='weighted')
precision = precision_score(true_labels, predicted_labels, average='weighted')
recall = recall_score(true_labels, predicted_labels, average='weighted')
print(f"F1 Score: {f1}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(classification_report(true_labels, predicted_labels))

# Tampilkan hasil prediksi dalam array
print("Array hasil prediksi:")
print(true_labels)
print(predicted_labels)
```

```
3/3 [=====] - 0s 7ms/step
F1 Score: 0.9336835724388766
```

```

F1 Score: 0.9336835724388766
Precision: 0.9344
Recall: 0.9333333333333333
      precision  recall  f1-score  support
0           0.88    0.92    0.90     24
1           0.96    0.94    0.95     51

   micro avg    0.93    0.93    0.93     75
   macro avg    0.92    0.93    0.92     75
  weighted avg    0.93    0.93    0.93     75
   samples avg    0.93    0.93    0.93     75

Array hasil prediksi:
[[ True False]
 [False True]
 [ True False]
 [False True]
 [False True]
 [False True]
 [ True False]
 [False True]]

[26] print("Panjang Tes Ulasan:", len(test['ULASAN']))
     print("Panjang X_test:", len(X_test))
     print("Panjang Y_test:", len(Y_test))
     print("Panjang true_labels:", len(true_labels))
     print("Panjang predicted_labels:", len(predicted_labels))

Panjang Tes Ulasan: 75
Panjang X_test: 75
Panjang Y_test: 75
Panjang true_labels: 75
Panjang predicted_labels: 75

import pandas as pd
from sklearn.metrics import classification_report, f1_score, precision_score, recall_score

# Melakukan prediksi
predictions = model.predict(X_test)
predicted_labels = (predictions > 0.5).astype(int) # Konversi probabilitas menjadi label biner (0 atau 1)
true_labels = Y_test

# Menghitung metrik evaluasi tambahan

[27] f1 = f1_score(true_labels.argmax(axis=1), predicted_labels.argmax(axis=1), average='weighted')
     precision = precision_score(true_labels.argmax(axis=1), predicted_labels.argmax(axis=1), average='weighted')
     recall = recall_score(true_labels.argmax(axis=1), predicted_labels.argmax(axis=1), average='weighted')
     print(f"F1 Score: {f1}")
     print(f"Precision: {precision}")
     print(f"Recall: {recall}")
     print(classification_report(true_labels.argmax(axis=1), predicted_labels.argmax(axis=1)))

# Ambil ulasan, label sebenarnya, dan prediksi label
test_results = pd.DataFrame({
    'Ulasan': test['ULASAN'].values,
    'Label Sebenarnya': true_labels.argmax(axis=1),
    'Prediksi': predicted_labels.argmax(axis=1)
})

# Klasifikasi label 'saran' dan 'kritik' berdasarkan nilai
def classify_label(label):
    return 'saran' if label == 1 else 'kritik'

# Menambahkan kolom klasifikasi label
test_results['Label Sebenarnya'] = test_results['Label Sebenarnya'].apply(classify_label)
test_results['Prediksi'] = test_results['Prediksi'].apply(classify_label)

[27] # Export ke Excel
     test_results.to_excel('/content/drive/MyDrive/Rosalinda - LSTM/hasil_prediksi.xlsx', index=False)

# Tampilkan hasil
print("\nHasil Prediksi:\n", test_results)
print("Data berhasil diekspor ke 'hasil_prediksi.xlsx'.")

3/3 [=====] - 0s 7ms/step
F1 Score: 0.9336835724388766
Precision: 0.9344
Recall: 0.9333333333333333
      precision  recall  f1-score  support
0           0.88    0.92    0.90     24
1           0.96    0.94    0.95     51

   accuracy    0.93    0.93    0.93     75
   macro avg    0.92    0.93    0.92     75
  weighted avg    0.93    0.93    0.93     75

Hasil Prediksi:
      Ulasan  Label Sebenarnya  Prediksi
0  [True False]                0          0
1  [False True]                1          1
2  [ True False]                0          0
3  [False True]                1          1
4  [False True]                1          1
5  [False True]                1          1
6  [ True False]                0          0
7  [False True]                1          1

```

```

[27] Hasil Prediksi:
      Ulasan Label Sebenarnya \
0  Pengurusan berkas sering kali terhambat karena... kritik
1  Tetap melayani dengan baik saran
2  Hadirkan dosen dan siap dalam mengajar kritik
3  Menyediakan materi kuliah dalam berbagai forma... saran
4  tabe dok mungkin lebih meningkatkan lagi prose... saran
..  ...
70 Di berikan arahan mengenai buku yang cocok seb... saran
71 Semoga lebih ditingkatkan lagi kinerjanya saran
72 Saran saya agar kedepanya materi di paparkan d... saran
73 Mahasiswa sering kali harus menunggu lama untu... kritik
74 Banyak mahasiswa yang mengeluh tentang sulitny... kritik

Prediksi
0  kritik
1  saran
2  saran
3  saran
4  saran
..  ...
70 saran
71 saran
72 saran
73 kritik

```

Lampiran 6 Hasil Prediksi

Ulasan	Label Sebenarnya	Prediksi
Pengurusan berkas sering kali terhambat karena masalah teknis yang tidak segera diatasi	kritik	kritik
Tetap melayani dengan baik	saran	saran
Hadirkan dosen dan siap dalam mengajar	kritik	saran
Menyediakan materi kuliah dalam berbagai format seperti video teks dan audio	saran	saran
tabe dok mungkin lebih meningkatkan lagi proses pembelajaran secara kelompok	saran	saran
Hadirkan quiz di setiap akhir pembelajaran	saran	saran
Staf tata usaha sering terlihat tidak peduli terhadap kepentingan mahasiswa	kritik	kritik
Sebaiknya seluruh ruangan dalam proses belajar mengajar dilengkapi fasilitas berupa tiap ruang kelas memiliki LCD dan AC	saran	saran
Perpustakaan online bisa diaktifkan kembali dengan koleksi buku yang lebih banyak dan bervariasi	saran	saran
Tugas kelompok tidak terorganisir dengan baik	kritik	kritik
Pintu kelas susah dibuka bikin ribet	kritik	kritik
Saran saran untuk dosen yang menempuh pelajaran aik agar kiranya bisa membuat hubungan emosional yang lebih baik agar kondusif kelas semakin nyaman plus referensi sekali sekali jangan kopas Mulu di google ajari kami agar berfikir kedepan bukan hanya kopas jawaban demi mendapatkan perilaku baik dan model belajarnya mungkin bisa di gunakan beberapa aplikasi terutama zoom meet dll agar emosional kami dosen dan teman teman yang lain bisa terjalin demi kenyamanan kelas	saran	saran
Mahasiswa sering kesulitan mengetahui jadwal kegiatan kampus	kritik	kritik

Saran saya lebih aktif lagi dalam menjelaskan pada saat pelajaran mata kuliah berlangsung	saran	saran
Pada kenyataannya perkuliahan dilaksanakan secara daring dikarenakan adanya pandemi covid ini yang terus saja melanda Indonesia sehingga kita perlu meningkatkan perkuliahan secara daring untuk meningkatkan rasa kepuasan dan rasa nyaman mahasiswa untuk terus melaksanakan perkuliahan secara daring	saran	saran
Fasilitas pengolahan limbah kampus tidak memadai	kritik	saran
Masih perlu tambahan penjelasan dalam mata kuliah	kritik	kritik
Semoga pembelajarannya kedepannya lebih efektif	saran	saran
Perlunya menghargai para dosen dalam aktivitas pengurusan administrasi di kampus sendiri	saran	saran
Toilet kampus jorok nggak terawat	kritik	kritik
Ada baiknya proses praktikum secara langsung didalami secara inti Mempraktekkan beberapa komoditikomoditi perkebunan	saran	saran
Semoga semester kedepan menjadi lebih baik dari sebelumnya	saran	saran
Staf tata usaha sering tidak ada di tempat saat dibutuhkan	kritik	kritik
Tetap selalu memotivasi mahasiswa dan memberikan respon apabila ada kendala	saran	saran
Kebutuhan perangkat Lab Komputer perlu di tingkatkan Perlunya laboratorium yang spesifik di bidangnya contoh Lab Data Sience dll	saran	saran
Semoga kedepannya lebih baik lagi	saran	saran
Harapan kami semoga ke depan metode dalam proses belajar dapat lebih baik dan menghadirkan pengalaman baru	saran	saran
Kemudahan dalam proses pembelajaran selama daring	saran	saran
Semoga lebih baik lagi kedepannya	saran	saran
Semoga fasilitas mengajar didalam kelas diperadakan dengan lengkap misalnya kipas angin agar kami mahasiswa bisa nyaman dalam mendapatkan ilmu materi baru dalam kelas Terimakasih	saran	saran
Semoga bisa lebih baik lagi	saran	saran
Keterbukaan antara dosen dengan mahasiswa perlu ditingkatkan lagi sehingga dapat tercipta komunikasi yang baik sehingga proses pembelajaran menjadi lebih menyenangkan dengan adanya saran dan solusi serta umpan balik yang tercipta dari suatu komunikasi yang baik	saran	saran
Mahasiswa tidak mendapatkan bantuan dalam menghadapi wawancara magang	kritik	kritik
Saran dari saya sebagai mahasiswa jika berkenan dosen perlu menghadirkan variasi mengajar baru Tidak hanya materi melainkan juga menyajikan materi perkuliahan menarik berupa video	saran	saran

pembelajaran atau yang lebih menarik lainnya supaya bisa lebih mudah di pahami oleh kami sebagai Mahasiswa		
Saran saya semoga ibu bisa menerapkan metode pembelajaran yang lebih variatif	saran	saran
Semoga kedepannya sarana dalam proses pembelajaran dapat lebih ditingkatkan lagi	saran	saran
Pihak Kampus Sebaiknya Memantau Proses Penyaluran Kuota Internet	kritik	kritik
semoga kedepannya bisa lebih baik dan sering sering offline	saran	saran
Saran saya adalah memaklumi jika ada mahasiswa yang terganggu jaringannya	saran	saran
Semoga lebih baik untuk kedepannya	saran	saran
sebaiknya meningkatkan pengaturan classroom mata kuliahnya agar mempermudah mahasiswa mengirim tugas terimakasih	saran	saran
Semoga fasilitas di kampus dapat di perbaharui	saran	saran
Banyak ruang kelas yang tidak memiliki komputer untuk presentasi	kritik	kritik
Mahasiswa sering kali harus menunggu lama karena keterbatasan staf	kritik	kritik
Semoga kedepannya pembelajaran bisa dilakukan secara tatap muka agar materinya akan lebih mudah lagi untuk dipahami dari sebelumnya	saran	saran
Semoga mtkul ipb di offlinekan	saran	saran
Penjelasan materi kurang jelas	kritik	kritik
Agar lebih meningkatkan kebersihan dilingkungan kampus dan bebas dari asap rokok	saran	saran
TU lambat bikin mahasiswa susah dapat dokumen tepat waktu	kritik	kritik
Harap meningkatkan lagi kualitas belajar daring	saran	saran
untuk memudahkan mahasiswa memahami materi dengan baik di harapkan perkuliahan di lakukan secara offline dan diharapkan kepada ibu dan bapak dosen agar masuk dan keluar dari kelas dengan tepat waktu	saran	saran
Sebaiknya media pembelajaran bervariasi agar tidak monoton Dan mahasiswa lebih semangat lagi Terima kasih	saran	saran
Sistem simak perlu ditingkatkan dikarenakan sistem sering mengalami eror	kritik	saran
Tempat parkir mobil sering kali tidak cukup luas untuk menampung kendaraan mahasiswa	kritik	kritik
banyak banyak mendekati diri dan memahami mahasiswanya pak	saran	saran
saran tolong kalau di chat setidaknya di balas atau konfirmasi sudah 2 semester mengajar tetap susah di hubungi	saran	saran

dosen diberikan ruang yang luas untuk lebih mengeksplor dan meningkatkan kompetensi dengan dukungan kuat dari pimpinan Dukungan kami sangat harapkan agar kekuatan kami tetap ada untuk tetap bisa mengabdikan di amal usaha ini	saran	saran
Saran saya hanya 1 semoga bapak bisa kasih nilai sesuai kemampuan mahasiswa	saran	saran
Keterlambatan dosen menunjukkan kurangnya dedikasi terhadap pengajaran	kritik	kritik
Banding nilai susah banget diajukan	kritik	kritik
Pelayanan akademik sangat memadai dan inovasi terbaru selalu di munculkan	saran	saran
Menggabungkan teknologi dan alat bantu visual dalam proses mengajar untuk membuat materi lebih menarik	saran	saran
Area parkir kampus sering tidak aman karena kurangnya pengawasan	kritik	kritik
Lebih meningkatkan proses sarana dan prasarana mengajarnya	saran	saran
Harus adanya jaminan di hari tua pensiunan	saran	saran
Proses pengurusan berkas sering kali berubah-ubah tanpa pemberitahuan	kritik	kritik
diharapkan agar pengembangan ke depan diutamakan kepada kelengkapan laboratorium komputer dan sebisanya ruangan untuk tiap dosen	saran	saran
Meningkatkan fasilitas ruang kelas kampus	saran	saran
Semoga semester ini bisa full offline dan fasilitas di ruang kelas diperbarui misal dengan mejanya mungkin bisa tidak menggunakan kayu lagikarena meja dan kursinya sdh penuh coretan dan itu kadang membuat baju jadi kotor dan penambahan kipasAC karena di beberapa kelas kipasnya sudah tidak berfungsi sangat disayangkan karena berpengaruh ada proses PBM dalam ruangan Terima kasih	saran	saran
baiknya ada toleran terhadap keterlambatan	saran	saran
Di berikan arahan mengenai buku yang cocok sebagai referensi belajar untuk masing-masing mata kuliah	saran	saran
Semoga lebih ditingkatkan lagi kinerjanya	saran	saran
Saran saya agar kedepannya materi di paparkan dengan baik	saran	saran
Mahasiswa sering kali harus menunggu lama untuk pengurusan surat keterangan	kritik	kritik
Banyak mahasiswa yang mengeluh tentang sulitnya mendapatkan informasi yang akurat	kritik	kritik

Lampiran 7 Hasil Turnitin



MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN

Alamat Kantor: Jl.Sultan Alauddin No.259 Makassar 90221 Tlp. (0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Rosalinda Aprilia Sari

Nim : 105841108320

Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	9 %	10 %
2	Bab 2	9 %	25 %
3	Bab 3	6 %	10 %
4	Bab 4	2 %	10 %
5	Bab 5	5 %	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan
Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan
seperlunya.

Makassar, 09 Agustus 2024

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,


Nursilafitri Himmah Lili
NBM. 964.894

Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593, fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id