

**KLASIFIKASI SARAN DAN KRITIK PADA SIMAK UNISMUH
DENGAN MENGGUNAKAN ALGORITMA
*RECCURENCT NEURAL NETWORK (RNN)***

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mendapatkan Gelar Sarjana

Komputer (S.kom) Program Studi Informatika



AHMAD FAISAL 105841112420

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2024

**KLASIFIKASI SARAN DAN KRITIK PADA SIMAK UNISMUH
DENGAN MENGGUNAKAN ALGORITMA
*RECCURENCT NEURAL NETWORK (RNN)***

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

Disusun Dan Diajukan Oleh:

AHMAD FAISAL

105841112420

**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>

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

PENGESAHAN

Skripsi atas nama Ahmad Faisal dengan nomor induk Mahasiswa 105 84 11124 20, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0009/SK-Y/55202/091004/2024, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Senin tanggal 26 Agustus 2024.

Panitia Ujian : Makassar, 21 Safar 1446 H
26 Agustus 2024 M

1. Pengawas Umum

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, S.T., M.T., IPU

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Isran Ramli, S.T., MT

2. Penguji

a. Ketua : Dr. Ir. Hj. Harsah Nirwana, S.T., MT

b. Sekretaris : Lukman, S.Kom., M.T.

3. Anggota : 1. Muhyiddin A M Hayat, S.Kom., M.T.

2. Rizki Yusliana Bakti, S.T., M.T.

3. Lukman Anas, S.Kom., M.T.

Mengetahui :

Pembimbing I

Titin Wahyuni, S.Pd., M.T

Pembimbing II

Fachrim Irhamna Rahman, S.Kom., M.T

Dekan



Dr. Ir. Hj. Nuzmahaty, S.T., MT., IPM.

NBM : 795 108



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 Informatika (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **KLASIFIKASI SARAN DAN KRITIK PADA SIMAK UNISMUH MAKASSAR DENGAN MENGGUNAKAN ALGORITMA RECCURENT NEURAL NETWORK (RNN)**

Nama : AHMAD FAISAL

Stambuk : 105 84 11124 20

Makassar, 26 Agustus 2024

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Pembimbing II

Titin Wahyuni, S.Pd., M.T

Fachrim Irhamna Rahman, S.Kom., M.T

Mengetahui,
Ketua Program Studi Informatika



Muhyiddin A. W. Sayat, S.Kom., MT.

NEWM - 1504 577

MOTTO DAN PERSEMBAHAN

Motto

“ Selama masih melihat senyum ibu, dunia akan baik baik saja ”

“Ijazah smp ibuku lebih tinggi, dari ijazah sarjanaku”

Persembahan

Kupersembahkan skripsi ini kepada orang-orang yang sangat kusayangi. Karya ini merupakan bentuk rasa syukur kepada Allah SWT. Yang telah memberi nikmat iman, kesehatan, dan kesempatan untuk menyelesaikan skripsi ini. Serta terimakasih kepada Panutanku Ayahanda Amiruddin dan pintu surgaku ibunda Hj Nurysamsi, terimakasih atas segala pengorbanan dan tulus kasih yang di berikan. Beliau memang tidak sempat merasakan bangku perkuliahan, namun mereka mampu senan tiasa memberikan yang terbaik, tak kenal lelah mendo'akan serta memberikan perhatian dan dukungan hingga penulis mampu menyelesaikan studinya sampai meraih gelar sarjana. Semoga ayah dan ibu sehat, panjang umur dan bahagia selalu.

ABSTRAK

AHMAD FAISAL, Klasifikasi Saran Dan Kritik Pada Simak Unismuh Dengan Menggunakan Algoritma *Reccurenct Neural Network* (RNN) (Dibimbing oleh Titin Wahyuni S.Pd.,M.T dan Fahrirm Irhamna S.Kom., M.T)

SIMAK Unismuh Makassar merupakan platform penting yang digunakan oleh mahasiswa untuk menyampaikan saran dan kritik terkait berbagai aspek akademik. Dalam penelitian ini, peneliti mengimplementasikan algoritma *Recurrent Neural Network* (RNN) untuk mengklasifikasikan saran dan kritik yang diterima melalui SIMAK Unismuh. Tujuan dari penelitian ini adalah untuk mengetahui implementasi Algoritma RNN dalam mengklasifikasi saran dan kritik di laman SIMAK Unismuh dan bagaimana keberhasilan Algoritma RNN dalam mengklasifikasi saran dan kritik di laman SIMAK Unismuh. RNN dipilih karena kemampuannya dalam mengolah data teks yang berurutan, seperti masukan dalam bentuk kalimat, yang memungkinkan model untuk menangkap konteks dari masukan tersebut secara lebih efektif. Dataset yang digunakan dalam penelitian ini terdiri dari sejumlah data saran dan kritik yang telah dikategorikan secara manual. Model RNN yang dibangun kemudian dilatih dan diuji menggunakan data tersebut untuk menilai akurasi dan performanya. Hasil penelitian menunjukkan bahwa model mencapai akurasi tertinggi sebesar 91% dan akurasi terendah sebesar 90%. Meskipun terdapat variasi dalam performa model, hasil ini menunjukkan bahwa RNN memiliki potensi yang baik dalam mengklasifikasikan teks saran dan kritik. Model RNN dapat membantu institusi dalam memahami dan merespon masukan dari pengguna dengan lebih efektif, meskipun masih memerlukan optimasi lebih lanjut untuk meningkatkan konsistensi dan akurasi hasil. Kesimpulan dari penelitian ini menunjukkan bahwa model RNN mampu mengklasifikasikan saran dan kritik dengan tingkat akurasi yang memadai. Penerapan model ini diharapkan dapat membantu pihak administrasi Unismuh dalam mengelola masukan dari mahasiswa secara lebih efisien, serta memberikan respons yang lebih tepat dan cepat terhadap kebutuhan akademik.

Kata Kunci: Klasifikasi Teks, *Recurrent Neural Network* (RNN), SIMAK Unismuh, Saran dan Kritik, Sistem Informasi Akademik

ABSTRACT

AHMAD FAISAL, *Classification of Suggestions and Criticisms on Simak Unismuh Using the Recurrence Neural Network (RNN) Algorithm (Supervised by Titin Wahyuni S.Pd., M.T and Fahrini Irhamna S.Kom., M.T)*

SIMAK Unismuh Makassar is an important platform used by students to submit suggestions and criticisms related to various academic aspects. In this study, researchers implemented the Recurrent Neural Network (RNN) algorithm to classify suggestions and criticisms received through SIMAK Unismuh. The purpose of this study was to determine the implementation of the RNN Algorithm in classifying suggestions and criticisms on the SIMAK Unismuh page and how successful the RNN Algorithm was in classifying suggestions and criticisms on the SIMAK Unismuh page. RNN was chosen because of its ability to process sequential text data, such as input in the form of sentences, which allows the model to capture the context of the input more effectively. The dataset used in this study consists of a number of suggestion and criticism data that have been categorized manually. The RNN model that was built was then trained and tested using the data to assess its accuracy and performance. The results showed that the model achieved the highest accuracy of 91% and the lowest accuracy of 90%. Although there were variations in model performance, these results indicate that RNN has good potential in classifying suggestion and criticism texts. The RNN model can help institutions understand and respond to user input more effectively, although it still requires further optimization to improve the consistency and accuracy of the results. The conclusion of this study shows that the RNN model is able to classify suggestions and criticisms with an adequate level of accuracy. The application of this model is expected to help the Unismuh administration in managing student input more efficiently, as well as providing more appropriate and faster responses to academic needs.

Keywords: *Text Classification, Recurrent Neural Network (RNN), SIMAK Unismuh, Suggestions and Criticisms, Academic Information System*

KATA PENGANTAR

Segala Puji bagi Allah SWT atas limpahan rahmat, kesehatan dan kekuatannya sehingga proposal skripsi dengan judul “KLASIFIKASI SARAN DAN KRITIK PADA SIMAK UNISMUH DENGAN MENGGUNAKAN RNN” ini dapat kami selesaikan sebagaimana salah satu syarat untuk penyusunan skripsi Program Studi Informatika. Shalawat dan junjungan Nabi Besar Muhammad SAW sebagai uswatun hasanah dan rahmatan lil alamin.

Penulis menyadari bahwa di dalam penulisan proposal bukanlah tujuan akhir dari suatu pembelajaran, penulis juga menyadari masih jauh dari kata sempurna baik dari segi isi, bahasa maupun dari segi penulisannya, hal ini disebabkan keterbatasan penulis dari segi pengetahuan. Untuk itu penulis mohon maaf atas segala kekurangan tersebut. Dan tidak menutup kemungkinan untuk segala saran dan kritik serta masukan yang bersifat membangun bagi diri penulis.

Tak lupa kami mengucapkan banyak terima kasih kepada semua pihak yang telah membantu dan memberikan motivasi dalam penyusunan proposal ini, terutama kepada :

1. Kedua orang tua saya yang tercinta, yaitu panutanku ayah amiruddin dan surgaku ibu nursyamsi. Penulis mengucapkan terima kasih yang sebesar-besarnya atas segala doa, kasih sayang dan dukungan baik secara moral maupun materi.
2. Keluarga besar H. Djafar tjcongkeng, yang sudah mensupport penulis dalam hal do'a, materi, dan dukungan.
3. Saudara saudara saya, Ainun jariah A, S.E, Ashabul khafi, Anta rindra maulana, yang selalu menjadi semangat saya dalam mengerjakan skripsi ini.
4. Bapak Muhyiddin M Hayat, S.Kom., M.T sebagai Ketua Prodi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

5. Ibu Titin Wahyuni, S.Pd., M.T sebagai pembimbing I dan Bapak Fachrim Irhamna Rahman S.Kom., M.T sebagai pembimbing II yang dengan telah Ikhlas memberikan bimbingan dan arahan selama penyusunan tugas akhir ini.
6. Segenap Bapak – bapak dan Ibu Dosen Prodi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar yang telah memberikan bakat dan ilmu pengetahuan serta mendidik penulis selama proses belajar mengajar di Universitas Muhammadiyah Makassar.
7. Teman-teman Kelas D Informatika 2020, yang tidak bisa saya sebut satu satu terimakasih sudah kebersamai penulis dari awal sampai dititik ini.
8. Rekan-rekan mahasiswa dari program studi teknik informatika atas dukungan dan kerjasamanya selama menempuh pendidikan serta penyelesaian penyusunan skripsi ini.
9. Sahabat sahabat saya, ikbal, andi afnansyap rifman, indrawan irwan, wanda nurul wahida, khaeratul mar'ah, Izzatul abidin, yang sudah menemani dan membantuk banyak hal dalam mengerjakan skripsi ini.
10. Dan terakhir untuk diri saya sendiri Ahmad Faisal, terimakasih sudah mampu berproses dan berjuang sampai titik ini, terimakasih sudah mau berkorban baik dalam hal tenaga ataupun waktu, betul kata pepatah “ tidak ada proses yang menghianati hasil”.

Billahi fisabilhaq, fastabiqul khairat.

Wassalamualaikum Wr. Wb.

Makassar, 26 Agustus 2024

Ahmad Faisal

DAFTAR ISI

HALAMAN JUDUL.....	
HALAMAN PENGESAHAN.....	ii
HALAM PERSETUJUAN	iii
HALAMAN PERSEMBAHAN.....	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah	3
C. Tujuan Penelitian.....	3
D. Ruang Lingkup.....	3
E. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
A. Landasan Teori	5
B. Penelitian Terkait.....	10
C. Kerangka Pikir	12
BAB III METODE PENELITIAN.....	13
A. Tempat dan Waktu Penelitian.....	13
B. Alat dan Bahan	13
C. Perancangan Penelitian	13
D. Teknik Pengujian Sistem.....	18
E. Teknik Analisis Data	19
BAB IV HASIL DAN PEMBAHASAN.....	20

A. Pengambilan Data	20
B. Pelabelan Data.....	22
C. Processing Data.....	23
D. Model <i>Reccurent Neural Network</i>	24
E. Pengujian Metode RNN	36
F. Hasil Pengujian Metode	44
BAB V PENUTUP	58
A. Kesimpulan	58
B. Saran.....	58
TINJAUAN PUSTAKA.....	59



DAFTAR GAMBAR

Gambar 1. Kerangka pikir.....	14
Gambar 2. Rancangan Penelitian.....	16
Gambar 3. Flowchart RNN.....	18
Gambar 4. Epoch 1-15 0.1	47
Gambar 5. Metrik 0.1	47
Gambar 6. Grafik 0.1	48
Gambar 7. Epoch 1-15 0.2.....	51
Gambar 8. Metrik 0.2.....	52
Gambar 9. Grafik 0.2	53
Gambar 10. Epoch 1-15 0.3.....	55
Gambar 11. Metrik 0.3.....	56
Gambar 12. Grafik 0.3	56

DAFTAR TABEL

Tabel 1. Dataset Ulasan.....	22
Tabel 2. Pelabelan Data.....	24
Tabel 3. Tahap Cleaning.....	25
Tabel 4. Tahap Tokenizing	25
Tabel 5. Hasil Prediksi 0.1	49
Tabel 6. Hasil Prediksi 0.2	53
Tabel 7. Hasil Prediksi 0.3	57



DAFTAR LAMIRAN

Lampiran 1. Data Set	62
Lampiran 2. Data Klasifikasi	65
Lampiran 3. Sourcode.....	76



BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi informasi yang semakin pesat di era globalisasi saat ini tidak bisa dihindari lagi pengaruhnya terhadap dunia pendidikan. Tuntutan global menuntut dunia pendidikan untuk selalu senantiasa menyesuaikan perkembangan teknologi terhadap usaha dalam peningkatan mutu pendidikan, terutama penyesuaian penggunaan Teknologi Informasi dan Komunikasi bagi dunia pendidikan khususnya dalam proses pembelajaran. Dampak globalisasi pada dunia pendidikan saat ini, khususnya di perguruan tinggi adalah pemanfaatan teknologi informasi pada semua aspek manajerial, bahkan sampai pada proses pembelajaran. Dalam konteks pembelajaran dan pengembangan institusi pendidikan seperti universitas semakin mengedepankan teknologi guna meningkatkan pengalaman akademik hingga sistem *administrative* yang dapat ditawarkan kepada mahasiswa (Yusny & Yasa, 2019).

Salah satu upaya yang dilakukan adalah dengan menyediakan platform yang disebut dengan SIMAK UNISMUH (Sistem Informasi Manajemen Akademik) Universitas Muhammadiyah Makassar. Platform ini menyediakan berbagai layanan untuk mahasiswa dan calon mahasiswa. Layanan yang tersedia mencakup pendaftaran mahasiswa baru, informasi jadwal kuliah, jadwal ujian, informasi akademik, dan hak angket. Platform ini dapat digunakan untuk menyampaikan pandangan, pengalaman, saran, serta kritik terhadap berbagai aspek kampus termasuk 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 mahasiswa.

Namun, efektifitas dari pengumpulan data tersebut sering kali terpengaruh oleh kualitas tanggapan yang diberikan oleh mahasiswa. Dalam praktiknya,

terdapat kecenderungan dimana mahasiswa 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 mengklasifikasikan konteks saran dan kritik yang disampaikan oleh mahasiswa, salah satu pendekatan yang dapat digunakan adalah dengan mengimplementasikan Algoritma RNN pada konteks saran dan kritik di SIMAK Unismuh Makassar.

Algoritma klasifikasi yang digunakan adalah algoritma RNN adalah algoritma mampu memodelkan dependensi sekuensial pada data masukan dan memiliki sebuah memori data yang berisikan informasi yang dihasilkan sebelumnya, kemampuan ini disebabkan oleh adanya koneksi simpul (*recurrent*) pada model dan memproses data masukan secara sekuensial. Pada tiap langkah waktu, proses dari RNN tidak hanya berupa data masukan asal, tapi juga nilai dari lapis tersembunyi yang terdapat pada langkah waktu sebelumnya sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut (Rozi et al., 2020)

Berdasarkan penelitian sebelumnya dengan judul “Klasifikasi Kalimat Ilmiah Menggunakan “RNN” Penelitian ini telah berhasil membuat model komputasi klasifikasi kalimat menggunakan RNN, dengan fitur yang telah diekstraksi menggunakan fungsi *Word2Vec* untuk menghasilkan satu set vektor (Firmansyah et al., 2020). Berdasarkan latar belakang tersebut, maka peneliti menggunakan metode RNN yang ditujukan untuk mengklasifikasikan atau mengolah kesesuaian konteks saran dan kritik pada hak angket simak Unismuh, sehingga akan diketahui klasifikasi dari kritik dan saran tersebut (positif, negatif, atau netral). Dengan menggunakan metode ini, diharapkan mendapat tingkat akurasi yang cukup tinggi

dalam mengklasifikasikan kesesuaian konteks saran dan kritik pada hak angket simak Unismuh.

B. Rumusan Masalah

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

1. Bagaimana implementasi Algoritma RNN dan dalam mengklasifikasi saran dan kritik pada SIMAK Unismuh Makassar?
2. Bagaimana Tingkat keberhasilan Algoritma RNN dalam mengklasifikasi saran dan kritik pada SIMAK Unismuh Makassar?

C. Tujuan Penelitian

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

1. Untuk mengetahui implementasi Algoritma RNN dalam mengklasifikasi saran dan kritik di laman SIMAK Unismuh
2. Untuk mengetahui keberhasilan Algoritma RNN dalam mengklasifikasi saran dan kritik di laman SIMAK Unismuh

D. Ruang Lingkup

Adapun ruang lingkup pada penelitian ini yaitu :

1. Penelitian ini akan difokuskan pada Universitas Muhammadiyah Makassar, dengan penekanan pada evaluasi layanan dan fasilitas yang disediakan oleh pihak kampus untuk mahasiswa
2. Penelitian ini akan memperhatikan jenis saran dan kritik yang dikumpulkan melalui platform saran pada SIMAK Unismuh Makassar
3. Penelitian akan menggunakan algoritma RNN dalam klasifikasi kritik dan saran yang ada di Simak Unismuh Makassar

E. 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, penggunaan algoritma (RNN) dalam mengklasifikasikan konteks kritik dan saran yang ada di Simak Unismuh Makassar.

BAB III METODE PENELITIAN

Bab ini akan menjelaskan desain penelitian, prosedur pengumpulan data, dan teknik analisis data yang digunakan termasuk implementasi algoritma dan *fast text* dalam mengklasifikasikan konteks kritik dan saran yang ada di Simak Unismuh Makassar.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisikan hasil desain sistem serta pembahasan terhadap desain tersebut

BAB V KESIMPULAN DAN SARAN

Bab ini atau bab terakhir akan membuat kesimpulan isi dari keseluruhan uraian bab – bab sebelumnya dan saran – saran dari hasil yang telah diperoleh serta yang di harapkan dapat bermanfaat dalam pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. SIMAK

Sistem manajemen akademik atau yang dikenal dengan SIMAK adalah aplikasi perangkat lunak yang berfungsi untuk menyajikan informasi dan mengelolah 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 (Ardiansyah et al., 2020). Sistem Informasi Manajemen Akademik adalah segala macam hasil interaksi antara elemen di lingkungan akademik untuk menghasilkan informasi yang kemudian dijadikan landasan pengambilan sebuah keputusan melaksanakan tindakan, baik oleh pelaku proses itu sendiri maupun dari pihak luar kampus (Agustiandra & Sabandi, 2019).

Tujuan Sistem Informasi Manajemen akademik adalah untuk menciptakan layanan akademik yang tertib, efisien dan efektif, sehingga terbentuk kesatuan pemahaman dan tindakan serta disiplin di kalangan mahasiswa, staf pengajar, dan staf administrasi. Merujuk PP Nomor 60 Tahun 1999 pasal 9 ayat 1 administrasi akademik pendidikan tinggi diselenggarakan dengan menerapkan sistem kredit semester, ayat 2 pelaksanaan ketentuan sebagaimana dimaksud pada ayat 1 diatur oleh Menteri (Map et al., 2020). 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).

Berdasarkan pendapat para ahli maka dapat disimpulkan bahwa Sistem Informasi Manajemen Akademik (SIMAK) adalah sebuah platform digital yang dirancang untuk mengelola berbagai aspek administrasi dan operasional akademik di institusi pendidikan tinggi. Tujuan utama SIMAK adalah untuk meningkatkan

efisiensi, transparansi, dan kualitas layanan akademik yang diberikan kepada mahasiswa, dosen, dan staf administrasi. Dengan berbagai fitur dan keuntungannya, Sistem Informasi Manajemen Akademik (SIMAK) telah menjadi elemen kunci dalam modernisasi institusi pendidikan tinggi, menjadikannya lebih adaptif dan siap menghadapi tantangan di era digital.

2. Kritik dan Saran

Kritik berasal dari kata kritein yang artinya hakekat, esensi atau substansi. Yaitu inti daripada sebuah persoalan. Sesungguhnya semua kritik pasti melihat sisi negatif. Tidak ada kritik melihat sisi positif. Tujuan semua kritik sama, supaya sesuatu yang negatif bisa menjadi positif. Masalahnya adalah, banyak orang tidak memahami hakekat daripada kritik itu sendiri

(Ardiansyah et al., 2020)

Saran adalah sebuah solusi yang ditujukan untuk menyelesaikan permasalahan yang dihadapi. Saran harus bersifat membangun, mendidik, dan secara objektif dan sesuai dengan topik yang dibahas (Adanson & Fitriana, 2022). Dalam menyampaikan kritik dan saran harus tetap menggunakan aturan agar yang menerima kritik dan saran tidak tersinggung, adapun caranya menyampaikan kritik dan saran sebagai berikut:

- a. Menyimak topik yang sedang di bahas dengan seksama dan penuh dengan konsentrasi dari awal hingga akhir.
- b. Mencatat pokok-pokok bahasan agar benar-benar paham dalam menentukan bagian yang akan dikritik atau diberi saran.
- c. Menganalisis pokok-pokok isi bahasan dengan menentukan bagianbagian yang hendak di kritik atau di beri saran.
- d. Menentukan kekurangan-kekurangan atau kelemahan-kelemahan isi bahasan tersebut, serta menentukan ideide baru atau perbaikanperbaikan yang tepat untuk menyempurnakan dalam pembahasan.
- e. Mengemukakan atau menyampaikan kritik atau saran berdasarkan bahasan yang dibahas.

Berdasarkan pendapat para ahli maka dapat disimpulkan bahwa kritik dan saran adalah dua elemen penting yang sering digunakan dalam berbagai konteks, seperti pendidikan, pekerjaan, layanan publik, dan interaksi sosial, untuk mendorong perbaikan dan perkembangan. Meskipun keduanya berfokus pada umpan balik, ada perbedaan penting antara kritik dan saran yang mempengaruhi cara mereka diterima dan diaplikasikan.

3. *Reccurent Neuaral Network (RNN)*

RNN merupakan salah satu jenis algoritma pembelajaran mesin yang berbasis pengolahan informasi dari sebuah data dan merupakan algoritma deep learning. RNN menyimpan pola-pola informasi masa lalu dengan melakukan perulangan di dalam arsitekturnya agar dapat membuat informasi tersebut tetap tersimpan. RNN merupakan pengembangan dari Jaringan Saraf Tiruan (JST) dan arsitekturnya mirip dengan Multilayer Perceptron (MLP) (Nurashila et al., 2023).

Pada dasarnya RNN memiliki parameter yang sama dengan neural network biasa, yang membedakan adalah konsep dari RNN itu sendiri. RNN menjadikan hidden layer sebelumnya sebagai input di proses selanjutnya. *Hidden layer* sebelumnya menyimpan informasi dari proses ekstraksi fitur di awal sehingga saat *hidden layer* dijadikan input untuk proses selanjutnya itu menyimpan memory (ingatan) dari input sebelumnya (Bai et al., 2021).

RNN adalah jenis jaringan saraf yang dirancang untuk mengolah data yang berurutan atau memiliki urutan waktu. Berbeda dengan jaringan saraf konvensional yang hanya melihat input dalam satu waktu, RNN dapat mempertimbangkan konteks dari data sebelumnya untuk mempengaruhi prediksi atau output berikutnya. Ini memungkinkan RNN untuk menangani data yang berbasis urutan seperti teks atau sinyal waktu, di mana informasi sebelumnya dapat memberikan wawasan penting untuk memahami data saat ini. Dalam RNN, informasi dari langkah sebelumnya disimpan dalam keadaan tersembunyi yang diupdate secara berulang selama proses

pelatihan, sehingga model bisa "mengingat" informasi dari langkah-langkah sebelumnya (Sherstinsky 2020).

RNN adalah jaringan saraf yang mengkhususkan diri dalam menangani data yang memiliki urutan atau sekuensi waktu. Berbeda dengan jaringan saraf biasa yang hanya mempertimbangkan input saat ini, RNN dapat menggunakan informasi dari langkah-langkah sebelumnya dalam urutan untuk mempengaruhi prediksi atau keputusan yang diambil pada langkah berikutnya. Ini membuat RNN sangat berguna untuk tugas-tugas seperti pemrosesan bahasa alami, di mana konteks dari kata-kata sebelumnya mempengaruhi makna kata berikutnya. Model ini bekerja dengan mengingat informasi dari langkah-langkah sebelumnya dan menggunakannya untuk memproses data yang sedang dianalisis (Ma, Zhang, dan Liu 2023)

Berdasarkan pendapat para ahli maka dapat disimpulkan bahwa, RNN adalah jenis jaringan saraf tiruan yang khusus dirancang untuk memproses data berurutan. Algoritma RNN memiliki kemampuan untuk mengingat informasi dari input sebelumnya dalam serangkaian data yang berurutan, membuatnya sangat cocok untuk tugas-tugas seperti pemrosesan bahasa alami, prediksi waktu seri, dan analisis urutan data lainnya.

4. *Supervised Learning*

Supervised learning melibatkan penggunaan data yang telah diannotasi, di mana setiap sampel data dikaitkan dengan label atau hasil yang diketahui sebelumnya. Tujuan dari *supervised learning* adalah untuk melatih model komputer yang dapat mempelajari pola-pola dalam data dan melakukan prediksi akurat terhadap data yang belum diketahui. Dengan menggunakan algoritma-algoritma *supervised learning*, model dapat digunakan untuk mengklasifikasikan data ke dalam kategori yang tepat atau untuk melakukan prediksi numerik (Mestika et al., 2022).

Dalam algoritma supervised learning, sistem akan diberikan data pelatihan yang berisi informasi input dan output yang diinginkan, 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).

5. NLP

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)

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 (Rumaisa et al., 2021)

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 (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.

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).

B. Penelitian Terkait

Berikut adalah beberapa referensi jurnal penelitian terdahulu tentang implementasi Algoritma RNN dalam mengklasifikasi saran dan kritik:

Penelitian yang dilakukan oleh Widi Afandi pada tahun 2022 dengan judul “Klasifikasi judul berita clickbait menggunakan RNN-LTSM ” bertujuan untuk mengklasifikasikan Hasil dari akurasi klasifikasi tertinggi menunjukkan bahwa model memiliki akurasi yang seimbang pada data training dan data testing di epoch 27 dengan sebesar 79% pada data training dan 77% pada data testing. Pada *loss* mengalami penurunan baik dari data training maupun testing. Hal ini dipengaruhi data *Preprocessing*, arsitektur model, dan keseimbangan data non clickbait dan clickbait. (Afandi et al., 2022)

Selanjutnya pada tahun 2020 Imam Fahrur Rozi melakukan penelitian yang berjudul “Klasifikasi Teks Laporan Masyarakat Pada Situs Lapori! Menggunakan *Recurrent Neural Network* yang bertujuan untuk mengklasifikasikan setiap laporan pengaduan dari masyarakat untuk persiapan proses verifikasi setiap dokumen laporan masyarakat, yang nantinya diharapkan dapat berdampak pada proses percepatan penanganan dan tindak lanjut dari setiap Lembaga atau instansi yang terkait. Pada penelitian ini, *Long Short-Term Memory Recurrent Neural Network* digunakan untuk melakukan proses klasifikasi setiap dokumen laporan masyarakat. Model pembelajaran dievaluasi menggunakan *k-fold cross-validation* sebanyak 10 bagian data. Hasil evaluasi menunjukkan rata-rata persentase *f-measure* sebesar 85,69% untuk dataset seimbang dan 79,44% untuk dataset tidak seimbang, sedangkan nilai evaluasi tertinggi dari semua evaluasi menghasilkan *f-measure* sebesar 88,82%. Akurasi pemodelan yang cukup tinggi menunjukkan bahwa metode yang diusulkan dapat digunakan untuk mengklasifikasikan dokumen laporan masyarakat (Rozi et al., 2020)

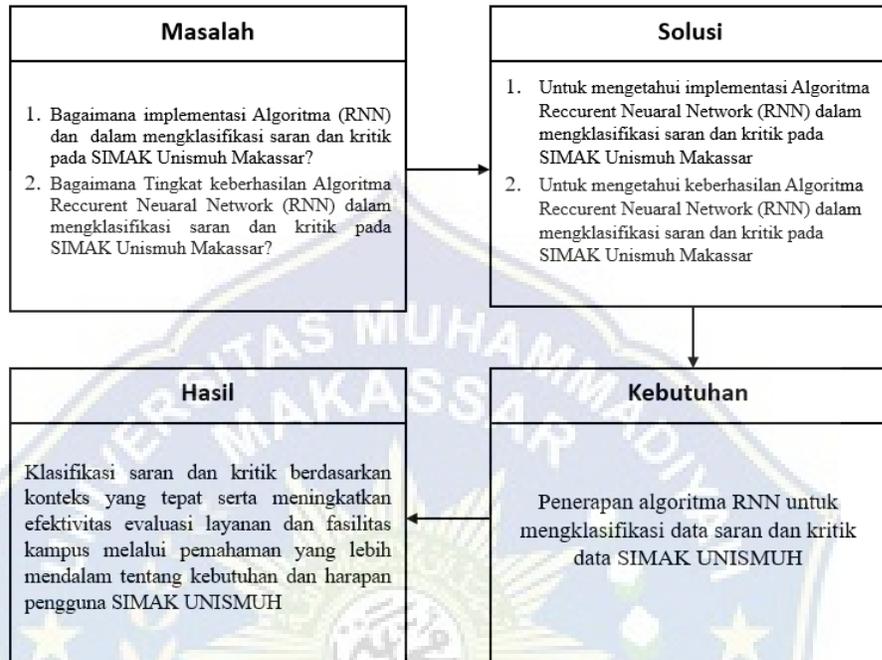
Pada tahun 2020 Muhammad Rizal Firmansyah melakukan penelitian yang berjudul “Klasifikasi kalimat ilmiah menggunakan *Recurrent Neural Network* bertujuan untuk membuat model komputasi klasifikasi kalimat menggunakan RNN,

dengan fitur yang telah diekstraksi menggunakan fungsi *Word2Vec* untuk menghasilkan satu set vektor. Hasil tingkat pembelajaran terbaik diperoleh dengan pengoptimalan SGD dengan nilai akurasi 77,48% dan Loss 0,71%. SGD tidak menggunakan banyak memori Gradient Descent sehingga konvergen lebih cepat. Selain itu SGD bekerja dengan memilih data sampel acak dari satu atau beberapa bagian dari data pelatihan dalam satu iterasi dengan cara yang iteratif (Firmansyah et al., 2020)

Pada tahun 2020, Izza Lutfhi Rais melakukan penelitian berjudul “Klasifikasi Data Kuesioner dengan Metode *Recurrent Neural Network* “ bertujuan untuk mempercepat hasil yang didapat dengan akurasi yang besar dalam waktu yang singkat. Arsitektur RNN yang digunakan untuk pengolahan data ini adalah LSTM (*Long Short Term Memory*). Hasil uji yang didapatkan pada penelitian ini cukup baik melihat akurasi tertinggi pada class positive untuk ratio (50:50) sebesar 56,73%, class neutral untuk ratio (70:30) sebesar 82,49%, dan class negative untuk ratio (80:20) sebesar 79,84%. (Tarkus et al., 2020)

Pada tahun 2020, Exel defrisco Tarkus melakukan penelitian berjudul “Implementasi Metode *Recurrent Neural Network* pada Pengklasifikasian Kualitas Telur Puyuh” dalam penelitian ini, maka dapat menarik kesimpulan bahwa ekstraksi fitur dengan menggunakan *Mel-Frequency Cepstral Coefficients* dan ekstraksi fitur data statistic dari 252 data suara yang menghasilkan 8.316 data ekstraksi sudah mampu menghasilkan model yang baik dalam klasifikasi telur puyuh menggunakan metode RNN. Hasil dari proses ekstraksi fitur berupa nilai statistika, *Fast Fourier Transform* (FFT) dan *Mel-Frequency Cepstral Coefficients* (MFCC) yang kemudian akan di training dengan metode RNN dan menghasilkan file model.h5. Untuk hasil prediksi precision diatas 75%, recall diatas 81% dan akurasi diatas 87% (Tarkus et al., 2020)

C. Kerangka Pikir



Gambar 1. Kerangka pikir

BAB III

METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan dalam kondisi daring dengan mengumpulkan konteks saran dan kritik pada hak angket simak unismuh. Penelitian ini akan berlangsung mulai Juni hingga Agustus 2024 .

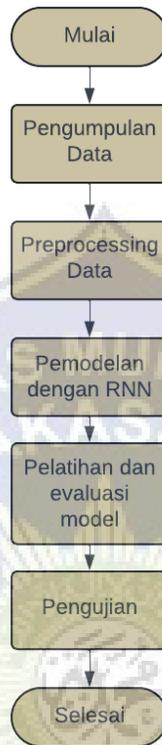
B. Alat dan Bahan

1. Kebutuhan Hardware
 - a. Laptop Asus Vivo Book
2. Kebutuhan Software
 - a. Google Colab
 - b. Python
 - c. Excel

C. Perancangan Sistem

Perancangan 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. Rancangan Penelitian

Flowchart di atas menggambarkan alur proses penelitian atau pengembangan model machine learning berbasis RNN untuk tugas tertentu. Berikut adalah penjelasan dari setiap langkah dalam flowchart:

1. Mulai:

- Tahap awal dari proses penelitian atau pengembangan model.

2. Pengumpulan Data:

- Mengumpulkan data yang akan digunakan untuk melatih dan menguji model. Data ini bisa berasal dari berbagai sumber, seperti dataset publik, data perusahaan, atau hasil survei.

3. Preprocessing Data:

- Melakukan preprocessing atau pra-pemrosesan data yang telah dikumpulkan. Langkah ini termasuk pembersihan data, tokenisasi, penghapusan *stop words*, dan konversi teks menjadi representasi yang dapat digunakan oleh model.

4. Pemodelan dengan RNN:

- Membangun model RNN menggunakan data yang telah dipreproses. Pada tahap ini, struktur model RNN dirancang dan diimplementasikan.

5. Pelatihan dan Evaluasi Model:

- Melatih model RNN menggunakan data pelatihan. Setelah model dilatih, kinerjanya dievaluasi menggunakan metrik evaluasi seperti *akurasi*, *presisi*, *recall*, dan *F1 score* untuk menilai performanya.

6. Pengujian:

- Menguji model yang telah dilatih dan dievaluasi menggunakan data uji untuk memastikan model bekerja dengan baik pada data yang belum pernah dilihat sebelumnya.

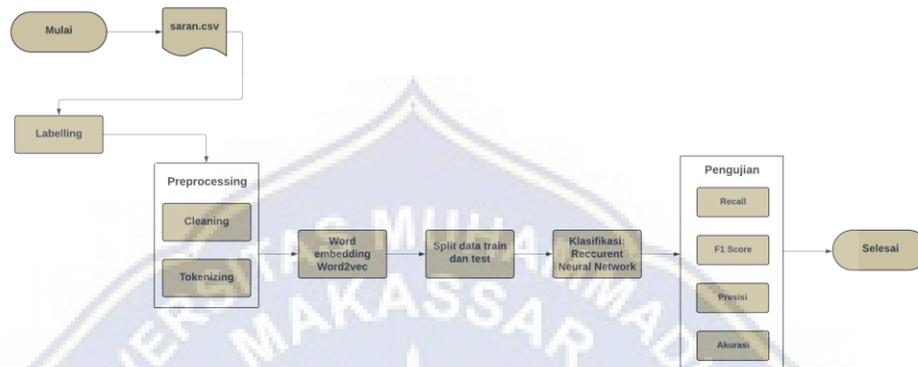
7. Selesai:

- Tahap akhir dari proses, menandakan bahwa penelitian atau pengembangan model telah selesai.

Flowchart ini memberikan gambaran umum tentang langkah-langkah utama dalam proses pengembangan model *machine learning* berbasis RNN, mulai dari pengumpulan data hingga pengujian dan penyelesaian.

Flowchart RNN

Flowchart di bawah menggambarkan alur proses pengolahan data untuk klasifikasi teks menggunakan Algoritma (RNN).



Gambar 3.Flowchart RNN

Berikut adalah penjelasan rinci dari setiap langkah dalam flowchart:

1. Mulai : Proses dimulai dari titik ini.

2. Data Input (saran.csv):

Data masukan yang berupa file CSV dengan nama `saran.csv`, yang berisi saran dan kritik dari SIMAK UNISMUH.

3. *Labeling* :

Pada tahap ini, setiap saran atau kritik dalam dataset diberikan label. Label ini biasanya berupa kategori atau kelas yang ingin diidentifikasi oleh model.

4. *Preprocessing*:

- *Cleaning* : Membersihkan data dari elemen yang tidak diperlukan seperti tanda baca, angka, atau karakter khusus yang tidak relevan.

- *Tokenizing* : Memecah teks menjadi unit-unit yang lebih kecil, seperti kata atau token.

5. *Word Embedding (Word2vec)* :

Proses ini mengubah teks menjadi representasi numerik menggunakan teknik *Word2vec*. *Word2vec* adalah metode pembelajaran distribusi kata yang mengubah kata-kata menjadi vektor berdimensi tetap, yang memungkinkan pemrosesan lebih lanjut oleh model pembelajaran mesin.

6. *Split Data (Train dan Test)* :

Data yang telah diproses kemudian dibagi menjadi dua set, yaitu set pelatihan (train) dan set pengujian (test). Set pelatihan digunakan untuk melatih model, sementara set pengujian digunakan untuk mengevaluasi performa model.

7. *Klasifikasi (Recurrent Neural Network)*:

Pada tahap ini, model RNN digunakan untuk mengklasifikasikan data saran dan kritik. RNN dipilih karena kemampuannya dalam menangani data sekuensial dan konteks temporal, yang penting dalam pemahaman teks.

8. *Pengujian*:

Setelah model dilatih, dilakukan pengujian untuk mengukur performa model menggunakan metrik evaluasi seperti:

- *Recall* : Mengukur seberapa baik model dapat menemukan semua sampel relevan.

- *F1 Score* : Menggabungkan *presisi* dan *recall* untuk memberikan gambaran umum tentang kinerja model.

- *Presisi* : Mengukur akurasi dari prediksi positif yang benar.

- *Akurasi* : Mengukur proporsi total prediksi yang benar, baik positif maupun negatif.

9. Selesai : Proses berakhir di sini, dengan hasil evaluasi performa model yang digunakan untuk interpretasi dan analisis lebih lanjut.

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 RNN.

Setelah data dipersiapkan, model algoritma RNN 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 sentimen dari data uji. Hasil pengujian diklasifikasi untuk menarik kesimpulan tentang keefektifan algoritma RNN dalam mengklasifikasikan 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 mengurangi data, langkah berikutnya adalah mengekspresikan data. Data teks harus diubah menjadi representasi numerik yang dapat dipahami oleh model. Salah satu metode umum yang digunakan adalah teknik untuk mengubah kata-kata menjadi vektor numerik. Selain itu, data juga perlu dibagi menjadi data latih dan data uji dengan proporsi yang sesuai.

3. Penarikan kesimpulan

Dari hasil pengujian, dapat disimpulkan mengenai kinerja model dalam menganalisis konteks saran dan kritik sistem. Kesimpulan ini mencakup penilaian terhadap akurasi dan konsistensi model dalam mengklasifikasi konteks saran dan kritik.

BAB IV

HASIL DAN PEMBAHASAN

A. Pengambilan Data

Tahap pengumpulan data yang digunakan dalam penelitian ini diperoleh dari Universitas SIMAK Muhammadiyah Makassar. SIMAK merupakan sumber data resmi yang mencatat informasi akademik mahasiswa, antara lain riwayat akademik, kualifikasi mata kuliah, dan status akademik lainnya.

Selain itu, SIMAK juga menyediakan hak angket bagi mahasiswa untuk memberikan saran dan kritik terhadap sistem akademik yang ada. Hak angket ini merupakan bagian penting dalam upaya peningkatan kualitas pendidikan, karena memungkinkan mahasiswa untuk berkontribusi dalam evaluasi dan perbaikan sistem secara konstruktif. *Feedback* yang diberikan oleh mahasiswa akan dianalisis dan digunakan untuk melakukan perbaikan yang diperlukan, sehingga menciptakan lingkungan akademik yang lebih baik dan responsif terhadap kebutuhan mahasiswa.

1. Dataset Ulasan

Setelah mengumpulkan data Tanggapan dari SIMAK UNISMUH, hasilnya kemudian disimpan dalam format Excel. Berikut adalah hasil pengumpulan data ulasan yang telah disimpan dalam Excel:

Tabel 1. Dataset Ulasan

ULASAN	LABEL
Kedepanya 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	Kritik

bagi mahasiswa sehingga mahasiswa lebih aktif dan produktif dalam proses pembelajaran.	
Memperbaiki yang baik menjadi lebih baik lagi.	Saran
Sebaiknya pihak kampus memantau proses penyaluran kouta internet, pihak kampus juga harus membuat rancangan kesepakatan perkuliahan online yang disetujui dosen dan mahasiswa serta meningkatkan kualitas layanan akedemik seperti portal online untuk kebutuhan kuliah online.	Kritik
Perlunya efesiensi 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 di selingi 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
Dengan Melihat Keadaan Dalam Pembelajaran Daring (Online), Pembelajaran Online (Daring) Sangat Tidak Baik Bagi Kami Untuk Fokus belajar, Karena Banyak-Nya Kendela2 Yang Kami Alamin Seperti : Jelek Jaringan, Sakit Mata, Dan Pembelian Kuota Tiap Bulan. Saya Berharap Kedepannya Sudah Bisa Melakukan Pembelajaran Tatap Muka.	Kritik
Semoga dengan adanya peningkatan pelayanan baik itu dalam proses perkuliahan, pembelajaran ataupun dari pelayanan akademik akan menjadi jauh lebih baik untuk kedepannya.	Saran
Peningkatan fasilitas kampus.	Kritik

Saran saya agar lebih ditingkatkan lagi proses pembelajaran daring yang lebih baik lagi.	Saran
Karena proses perkuliahan masih daring, diharapkan agar sering mengingatkan kepada mahasiswa masalah koneksi ketika ada kuis/UTS/UAS agar mahasiswa bisa mempersiapkan kemungkinan buruk yang terjadi terutama masalah koneksi internet.	Saran
Semoga para dosen bisa menjelaskan materi dengan kreatif agar mahasiswa mudah memahami.	Saran
Semoga lebih baik kedepannya.	Saran
Saran saya semoga bisa offline.	Saran

B. Pelabelan Data

Pelabelan sentimen dilakukan secara manual untuk mengidentifikasi pola dan karakteristik dalam teks yang menunjukkan sentimen Saran dan kritik. Ulasan yang diperoleh SIMAK UNISMUH disimpan dalam atribut ulasan, sedangkan nilai klasifikasi saran dan kritik disimpan dalam atribut label. Berikut tabel pelabelan data yang dapat dilihat di bawah ini:

Tabel 2. Tahap Pelabelan Data

ULASAN	LABEL
Beberapa dosen terutama pejabat kampus seringkali sangat sibuk sehingga perkuliahan tidak berjalan dengan optimal sehingga mahasiswa dirugikan	kritik
Semoga ruangan perkuliahan bisa memiliki pendingin ruangan (AC)	Saran
proses perkuliahan sangat baik dan disiplin	Tidak termasuk saran dan kritik

C. Preprocessing

Preprocessing data adalah tahap di mana data mentah disiapkan dan dibersihkan sebelum dianalisis. Proses ini melibatkan beberapa langkah untuk memastikan data siap digunakan. Berikut tahapan-tahapan dalam *preprocessing* data:

1. Cleaning

Tahap *cleaning* data melibatkan proses menghapus tanda baca seperti koma, titik, tanda tanya, tanda seru, bintang, dan pagar dari teks atau data. Langkah ini penting karena karakter-karakter tersebut biasanya tidak memberikan kontribusi signifikan dalam analisis data atau pemrosesan teks. Misalnya, dalam analisis teks, tanda baca sering dianggap sebagai *noise* yang tidak relevan dan dihapus untuk memfokuskan perhatian pada informasi utama dalam teks tersebut.

Tabel 3. Tahap Cleaning

Sebelum	Sesudah
Sangat komunikatif dengan mahasiswa	Sangat komunikatif dengan mahasiswa
Sudah baik, terimakasih atas ilmu yang telah diberikan	Sudah baik terimakasih atas ilmu yang telah diberikan
Walaupun yang masuk diklas asdos tetapi cara mengajarnya sangat baik, terimakasih atas ilmu yang telah diberikan	Walaupun yang masuk diklas asdos tetapi cara mengajarnya sangat baik terimakasih atas ilmu yang telah diberikan

2. Tokenizing

Tokenizing adalah langkah dalam pemrosesan teks yang melibatkan pemecahan teks menjadi unit-unit lebih kecil yang disebut token. Unit-unit ini dapat berupa kata-kata, frasa, atau entitas lainnya.

Tabel 4. Tahap *Tokenizing*

Sebelum	Sesudah
maksimalkan pertemuan dalam perkuliahan	['maksimalkan', 'pertemuan', 'dalam', 'perkuliahan']
sangat komunikastif dengan mahasiswa	['sangat', 'komunikastif', 'dengan', 'mahasiswa']
kedepannya bisa lebih ditingkatkan lagi	['kedepannya', 'bisa', 'lebih', 'ditingkatkan', 'lagi']

D. Model *Reccurent Neural Network* (RNN)

Recurrent Neural Networks (RNN) adalah jenis jaringan saraf yang sangat efektif untuk memproses data urutan atau data sekuensial, seperti teks, audio, video, dan data waktu lainnya. Mereka digunakan dalam berbagai aplikasi, termasuk pengenalan suara, pemrosesan bahasa alami, dan analisis deret waktu. RNN memiliki *neuron* yang berulang, yang artinya keluaran dari satu langkah waktu diberikan kembali ke *neuron* sebagai masukan untuk langkah waktu berikutnya. Ini memungkinkan jaringan untuk mempertahankan "memori" tentang informasi sebelumnya dalam urutan.

```
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

```

Kode di atas mengimpor berbagai pustaka penting untuk pemrosesan teks dan pembelajaran mesin. Pustaka `'pandas'` dan `'numpy'` digunakan untuk manipulasi data dan operasi numerik, sedangkan `'tqdm'` menampilkan progress bar untuk operasi `'pandas'`. `Tokenizer` dari Keras mengubah teks menjadi token, sementara `Doc2Vec` dari Gensim menghasilkan representasi vektor dari dokumen. Utilitas dari `Scikit-learn` membantu membagi dataset menjadi set pelatihan dan pengujian, dan fungsi `padding` dari Keras memastikan semua urutan teks memiliki panjang yang sama. Model regresi logistik dari `Scikit-learn` digunakan untuk klasifikasi, dan pustaka `'re'` membantu manipulasi teks menggunakan ekspresi *reguler*. Terakhir, `'seaborn'` dan `'matplotlib'` digunakan untuk visualisasi data statistik dan pembuatan plot.

```

df =
pd.read_excel('//content/drive/MyDrive/Skripsi/2020
/Ahmad Faisal
RNN/saran.xlsx', sheet_name="Sheet1")
# Replace 'path_to_your_excel_file.xlsx' with your
actual file path
df = df[['ULASAN', 'LABEL']]
# Selecting relevant columns
df = df[pd.notnull(df['ULASAN'])]
# Dropping rows with null 'ULASAN' values
df.rename(columns={'ULASAN': 'ULASAN'},
inplace=True)
# Rename 'ULASAN' to 'ULASAN' for consistency

```

Kode di atas membaca data dari file Excel menggunakan `'pandas'`, khususnya dari *sheet* "Sheet1", dan menyimpan data tersebut dalam variabel `'df'`. Kemudian, hanya kolom 'ULASAN' dan 'LABEL' yang dipilih untuk analisis lebih lanjut. Baris dengan nilai 'ULASAN' yang kosong dihapus untuk

memastikan hanya ulasan yang valid yang dipertahankan. Terakhir, kolom 'ULASAN' diberi nama ulang menjadi 'ULASAN' meskipun sebenarnya tidak mengubah apa pun, mungkin untuk memastikan konsistensi atau mengatasi potensi masalah nama kolom sebelumnya.

```
df.head()
```

Kode `df.head()` menampilkan lima baris pertama dari DataFrame `df`. Ini berguna untuk mendapatkan gambaran awal tentang data yang dimuat dan memastikan bahwa data telah dibaca dan diproses dengan benar.

```
df.shape
```

Kode `df.shape` digunakan untuk mengetahui dimensi dari DataFrame `df`. Ini akan menghasilkan sebuah tuple yang berisi dua angka: jumlah baris dan jumlah kolom dalam DataFrame.

```
# 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)
```

Kode di atas pertama-tama mengatur ulang indeks baris pada DataFrame `df` agar berurutan dari 0 hingga jumlah baris dikurangi satu. Setelah itu, kode menghitung total jumlah kata dalam kolom 'ULASAN' dengan menerapkan fungsi lambda pada setiap elemen di kolom tersebut. Fungsi lambda ini membagi setiap ulasan menjadi daftar kata berdasarkan spasi dan kemudian menghitung jumlah kata dalam setiap daftar. Hasilnya kemudian dijumlahkan untuk mendapatkan total kata dari semua ulasan. Terakhir, total jumlah kata dalam semua ulasan dicetak ke layar.

```

# 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()

```

Kode di atas menghitung jumlah kemunculan setiap nilai dalam kolom 'LABEL' pada DataFrame `df` dan kemudian membuat diagram batang untuk visualisasi hasilnya menggunakan Seaborn dan Matplotlib. Pertama, `df['LABEL'].value_counts()` menghasilkan seri yang berisi jumlah kemunculan setiap label. Lalu, `plt.figure(figsize=(12, 4))` menetapkan ukuran gambar plot. `sns.barplot(x=cnt_pro.index, y=cnt_pro.values, alpha=0.8)` membuat diagram batang dengan label pada sumbu x dan jumlah kemunculan pada sumbu y. Label sumbu y dan x diatur dengan `plt.ylabel` dan `plt.xlabel` masing-masing, serta rotasi label x diatur ke 90 derajat untuk memastikan keterbacaan dengan `plt.xticks(rotation=90)`. Akhirnya, `plt.show()` menampilkan diagram batang tersebut.

```

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)

```

Fungsi `print_message(index)` menerima indeks sebagai argumen dan mengambil baris dari DataFrame `df` berdasarkan indeks tersebut, kemudian menampilkan nilai dari kolom 'ULASAN' dan 'LABEL'. Dalam fungsi ini,

`df.iloc[index][['ULASAN', 'LABEL']].values` mengambil nilai dari kolom 'ULASAN' dan 'LABEL' pada baris dengan indeks yang diberikan, mengembalikan hasil sebagai array. Jika array yang dihasilkan tidak kosong, fungsi akan mencetak teks ulasan dan labelnya. Setelah mendefinisikan fungsi tersebut, contoh penggunaannya ditunjukkan dengan memanggil `print_message(12)`, yang akan menampilkan ulasan dan label dari baris ke-13 (karena indeks dimulai dari 0) dalam DataFrame `df`.

```
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)
```

Kode di atas mendefinisikan fungsi `remove_punctuation(text)` yang menggunakan metode `translate` dari string untuk menghapus semua tanda baca dari teks yang diberikan. Fungsi ini memanfaatkan `str.maketrans('', '', string.punctuation)` untuk membuat tabel translasi yang menghubungkan setiap tanda baca dalam `string.punctuation` dengan string kosong (`''`), sehingga tanda baca tersebut dihilangkan dari teks. Setelah mendefinisikan fungsi `remove_punctuation`, kode menerapkannya pada kolom 'ULASAN' dalam DataFrame `df` dengan menggunakan `df['ULASAN'].apply(remove_punctuation)`. Ini menghasilkan penghapusan semua tanda baca dari setiap baris dalam kolom 'ULASAN', sehingga memungkinkan fokus pada kata-kata dan teks utama tanpa gangguan dari karakter non-alfanumerik. Dengan menghapus tanda baca ini, analisis teks seperti tokenisasi dan pemrosesan lebih lanjut dapat dilakukan lebih efektif karena karakter yang tidak relevan telah dieliminasi.

```

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'].value
s)
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))

```

Kode di atas melakukan beberapa langkah penting dalam pra-pemrosesan teks dan pembagian data untuk analisis lebih lanjut. Pertama, pustaka NLTK digunakan untuk mengunduh sumber daya *'punkt'* yang diperlukan untuk tokenisasi teks. Fungsi *'tokenize_text(text)'* kemudian didefinisikan untuk membagi teks menjadi kalimat dan kata, serta mengubah semua kata menjadi huruf kecil, menghapus kata yang panjangnya nol. Data kemudian dibagi menjadi set pelatihan dan pengujian dengan rasio 99:1 menggunakan *'train_test_split'*. Kedua, *'TaggedDocument'* digunakan untuk membuat dokumen berlabel dari set pelatihan dan pengujian. Selanjutnya, *'Tokenizer'* dari Keras diatur dengan jumlah maksimum kata yang akan digunakan (*'max_features'*) dan panjang maksimum setiap teks (*'max_sequence_length'*). *Tokenizer* ini dilatih pada kolom 'ULASAN' dari DataFrame *'df'*, dan teks diubah menjadi urutan angka. Urutan angka ini kemudian di-padding agar semua urutan memiliki panjang yang sama. Akhirnya, kode mencetak jumlah token unik yang ditemukan oleh *tokenizer*. Langkah-langkah ini adalah bagian penting dari pra-pemrosesan teks untuk analisis dan pelatihan model pembelajaran mesin.

```
# 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)
```

Kode di atas mengonversi teks dari kolom 'ULASAN' dalam DataFrame `df` menjadi urutan angka menggunakan tokenizer yang telah diatur sebelumnya. Proses ini dilakukan dengan memanggil `tokenizer.texts_to_sequences(df['ULASAN'].values)`, yang mengubah setiap ulasan menjadi serangkaian angka berdasarkan indeks kata dalam `tokenizer`. Kemudian, `pad_sequences(X, maxlen=max_sequence_length)` digunakan untuk memastikan bahwa semua urutan angka memiliki panjang yang sama, dengan padding di bagian awal jika urutan lebih pendek dari panjang maksimum yang ditentukan (`max_sequence_length`). Ini penting untuk memastikan konsistensi input saat melatih model pembelajaran mesin, karena banyak model memerlukan input dengan panjang yang sama. Akhirnya, kode mencetak bentuk (`shape`) dari data tensor `X`, yang menunjukkan jumlah sampel dan panjang urutan yang digunakan.

```
#train_tagged.values
train_tagged.values
```

Ketika Anda menggunakan `train_tagged.values`, Anda mengakses nilai-nilai yang terkandung dalam DataFrame `train_tagged`. DataFrame `train_tagged` sendiri telah dibuat dengan menerapkan `TaggedDocument` pada set pelatihan `train`. Ini berarti setiap baris dalam `train_tagged` merupakan objek `TaggedDocument` yang terdiri dari daftar kata-kata yang sudah ditokenisasi (`words`) dan label yang terkait (`tags`). Dengan memanggil `.values` pada `train_tagged`, Anda mengambil representasi nilai dari setiap entri dalam DataFrame, yang dalam konteks ini berisi `array numpy` dari `TaggedDocument` untuk setiap baris. Ini memungkinkan Anda untuk mengakses dan menggunakan data dalam format yang sesuai dengan kebutuhan

analisis atau pelatihan model lebih lanjut, seperti dalam pembuatan model *Doc2Vec* menggunakan Gensim atau metode pembelajaran mesin lainnya yang memerlukan data dalam bentuk yang konsisten dan terstruktur.

```
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)
```

Kode di atas menggunakan library Gensim untuk membangun model *Doc2Vec*. Pertama, model *Doc2Vec* diinisialisasi dengan beberapa parameter yang ditentukan: `dm=1` mengindikasikan penggunaan model PV-DM (Distributed Memory), `dm_mean=1` menunjukkan penggunaan rata-rata vektor kata, `vector_size` menetapkan dimensi vektor fitur yang dihasilkan (dalam contoh ini, 20), `window` mengatur jarak maksimum antara kata target dan kata yang diprediksi dalam konteks (diatur ke 8), `min_count` menetapkan frekuensi minimum kata untuk dimasukkan ke dalam kosakata model (diatur ke 1), `workers` mengatur jumlah thread yang digunakan selama pelatihan (diatur ke 1 untuk demo), dan `alpha` serta `min_alpha` mengatur tingkat pembelajaran model. Selanjutnya, variabel `train_tagged` dibangun dengan menggunakan comprehension list. Ini menerapkan `TaggedDocument` pada set

pelatihan `train`, di mana setiap dokumen dihasilkan dengan mengekstrak kata-kata dari kolom 'ULASAN' menggunakan fungsi `tokenize_text(row['ULASAN'])` dan ditag dengan `row['LABEL']` sebagai labelnya. Proses ini mempersiapkan data dalam format yang sesuai untuk pelatihan model *Doc2Vec*. Langkah terakhir menggunakan `build_vocab(train_tagged)` untuk membangun kosakata dari dokumen-dokumen yang sudah ditag sebelumnya. Kosakata ini akan digunakan saat melatih model untuk menghasilkan vektor representasi kata-kata yang mempertimbangkan konteks dan hubungan semantik antar kata dalam set data pelatihan. Proses ini merupakan langkah awal penting dalam menggunakan model *Doc2Vec* untuk mengekstrak dan memahami struktur semantik dari teks yang direpresentasikan.

```
# 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
```

Kode di atas melakukan pelatihan pada model *Doc2Vec* yang sudah dibangun sebelumnya. Pertama, `d2v_model.build_vocab(train_tagged)` digunakan untuk membangun kosakata dari dokumen-dokumen yang telah ditag sebelumnya dalam `train_tagged`. Setelah kosakata dibangun, model dilatih dengan melakukan iterasi sebanyak 30 epoch. Pada setiap epoch, `d2v_model.train(utils.shuffle(train_tagged), total_examples=len(train_tagged), epochs=1)` digunakan untuk melatih model dengan data pelatihan yang telah diacak (untuk memperbaiki pembelajaran), dengan jumlah total contoh yang disertakan dalam pelatihan (`total_examples=len(train_tagged)`) dan

satu epoch per iterasi. Selama pelatihan, `d2v_model.alpha` dikurangi sebesar 0.002 setiap epoch untuk mengurangi tingkat pembelajaran seiring berjalannya waktu (`d2v_model.alpha -= 0.002`), sementara `d2v_model.min_alpha` diperbarui untuk mempertahankan nilai alpha minimum sesuai dengan nilai alpha saat ini, memastikan bahwa tingkat pembelajaran tidak turun terlalu rendah selama pelatihan berlangsung. Proses ini bertujuan untuk mengoptimalkan vektor representasi kata-kata dalam model *Doc2Vec* agar dapat menangkap dan memahami hubungan semantik yang ada dalam data teks yang diberikan.

```
print(d2v_model)
```

Pemanggilan `print(d2v_model)` pada model *Doc2Vec* akan menghasilkan informasi singkat tentang konfigurasi dan parameter model yang telah dibangun dan dilatih. Ini termasuk detail seperti jenis model yang digunakan (misalnya, PV-DM atau PV-DBOW), ukuran vektor fitur yang telah ditentukan, jumlah kata unik dalam kosakata, serta parameter lain seperti jendela konteks yang digunakan dalam pembangunan vektor. Informasi ini memberikan gambaran tentang bagaimana model *Doc2Vec* telah dikonfigurasi dan siap digunakan untuk keperluan analisis atau tugas pemrosesan bahasa lainnya, seperti pemodelan topik atau pencarian informasi berbasis teks.

```
# 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)
```

Kode di atas bertujuan untuk mengakses informasi tentang kosakata yang telah dibangun oleh model *Doc2Vec* (`d2v_model`). Pertama, `num_words = len(d2v_model.wv.key_to_index)` menghitung jumlah kata

yang ada dalam kosakata dengan mengambil panjang dari dictionary `'key_to_index'` yang dimiliki oleh objek `'wv'` (*Word Vectors*) dari model. Kemudian, `'words_in_vocab = list(d2v_model.wv.key_to_index.keys())'` digunakan untuk mengambil daftar kata-kata yang ada dalam kosakata dengan mengakses kunci-kunci dari dictionary `'key_to_index'`. Hasilnya dicetak untuk menampilkan jumlah total kata dalam kosakata dan daftar kata-kata itu sendiri. Informasi ini berguna untuk memahami cakupan kosakata yang dimiliki oleh model, yang merupakan dasar dari representasi vektor kata-kata yang telah dipelajari selama proses pelatihan.

```
# Inisialisasi 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])
```

Kode di atas melakukan inisialisasi dan pengisian matriks embedding dengan vektor-vektor dokumen yang dihasilkan oleh model *Doc2Vec* (`'d2v_model'`). Pertama, `'embedding_matrix'` diinisialisasi sebagai matriks berukuran `'(len(d2v_model.dv.vectors), d2v_model.vector_size)'`, di mana `'len(d2v_model.dv.vectors)'` mengambil jumlah vektor dokumen yang tersimpan dalam model, dan `'d2v_model.vector_size'` menunjukkan dimensi dari setiap vektor. Selanjutnya, loop `'for'` digunakan untuk mengisi matriks `'embedding_matrix'` dengan vektor-vektor dokumen dari `'d2v_model.dv.vectors'`. Setiap vektor dokumen diambil dari indeks `'i'` dan

disalin ke `'embedding_matrix[i]'`. Hasilnya, penggunaan matriks embedding dicetak untuk menampilkan ukuran matriks yang telah dibuat dan contoh dari vektor yang mewakili dokumen pertama dalam model. Matriks embedding seperti ini berguna dalam banyak tugas pemrosesan bahasa alami, seperti klasifikasi teks atau analisis sentimen, di mana representasi vektor dari teks dapat digunakan sebagai input untuk model pembelajaran mesin.

E. Pengujian Metode RNN

```
from keras.models import Sequential
from keras.layers import SimpleRNN, Dense, Embedding,
Dropout
from keras.callbacks import ModelCheckpoint,
EarlyStopping
import numpy as np
```

from keras.models import Sequential: Mengimpor kelas *Sequential* dari Keras, yang digunakan untuk membangun model berlapis-lapis secara berurutan.

from keras.layers import SimpleRNN, Dense, Embedding, Dropout: Mengimpor berbagai jenis lapisan yang akan digunakan dalam model, termasuk *SimpleRNN*, *Dense*, *Embedding*, dan *Dropout*.

from keras.callbacks import ModelCheckpoint, EarlyStopping: Mengimpor *callback* untuk menyimpan model terbaik dan menghentikan pelatihan lebih awal jika kinerja model berhenti meningkat.

import numpy as np: Mengimpor *NumPy*, sebuah library yang digunakan untuk operasi numerik.

```
# 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
```

MAX_SEQUENCE_LENGTH: Panjang maksimum dari urutan input yang akan digunakan oleh model.

num_unique_words: Jumlah kata unik dalam kamus/*tokenizer* yang digunakan. Ini dihitung sebagai panjang indeks kata dari *tokenizer* ditambah 1.

embedding_matrix: Matriks embedding yang berisi representasi vektor dari kata-kata dalam kamus/*tokenizer*. Dalam hal ini, vektor embedding diinisialisasi dengan nilai acak.

```
# Inisialisasi 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 RNN
model.add(SimpleRNN(50, return_sequences=False))

# Menambahkan lapisan Dropout untuk mencegah overfitting
model.add(Dropout(0.5))

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

# Menampilkan ringkasan model
model.summary()
```

model = Sequential(): Membuat model Sequential baru.

model.add(Embedding(...)): Menambahkan lapisan embedding yang mengubah indeks kata menjadi vektor dengan dimensi 20, menggunakan bobot dari *embedding_matrix*. Lapisan ini juga menerima panjang input maksimum (*input_length*).

model.add(SimpleRNN(...): Menambahkan lapisan SimpleRNN dengan 50 unit. *return_sequences=False* berarti lapisan ini hanya mengembalikan output dari langkah waktu terakhir.

model.add(Dropout(...): Menambahkan lapisan dropout dengan probabilitas dropout sebesar 0.5 untuk mencegah *overfitting*.

model.add(Dense(...): Menambahkan lapisan Dense dengan 2 unit output dan fungsi aktivasi sigmoid. Ini biasanya digunakan untuk masalah klasifikasi biner atau multilabel.

model.summary(): Menampilkan ringkasan arsitektur model.

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

optimizer="adam": Menggunakan optimizer Adam untuk memperbarui bobot.

loss="binary_crossentropy": Menggunakan fungsi *loss binary cross-entropy*, yang cocok untuk masalah klasifikasi biner atau multilabel.

metrics=['acc']: Menggunakan akurasi sebagai metrik evaluasi.

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

Fungsi `split_input` menerima sebuah urutan (*array*) dan membagi urutan tersebut menjadi dua bagian:

- a. Bagian pertama (x) terdiri dari semua elemen kecuali elemen terakhir.
- b. Bagian kedua (y) terdiri dari semua elemen kecuali elemen pertama.

```
# 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)
```

sequence_example adalah urutan [1, 2, 3, 4, 5].

Setelah memanggil `split_input`, kita mendapatkan `x` sebagai [1, 2, 3, 4] dan `y` sebagai [2, 3, 4, 5].

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

'pd.get_dummies(df['LABEL']): Fungsi `pd.get_dummies` digunakan untuk mengubah kolom label ('LABEL') dalam DataFrame `df` menjadi representasi *one-hot encoded*. Ini berarti setiap label kategori diubah menjadi vektor biner dengan satu elemen yang diset ke 1 (menandakan keberadaan kategori) dan yang lainnya ke 0.

.values: Mengonversi DataFrame hasil *one-hot encoding* menjadi *array NumPy*.

```
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)
```

'train_test_split: Fungsi dari `sklearn.model_selection` digunakan untuk membagi dataset menjadi subset pelatihan dan pengujian.

'X: Fitur input untuk model (asumsi ada variabel `X` yang sudah didefinisikan sebelumnya).

'Y: Label hasil *one-hot encoding*.

'test_size=0.01: Mengatur proporsi data untuk subset pengujian menjadi 1% dari total data.

'random_state=42: Menetapkan seed untuk pembangkit bilangan acak agar pembagian data dapat direproduksi.

```
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)
```

X_train.shape: Menampilkan dimensi dari *array* fitur pelatihan.

Y_train.shape: Menampilkan dimensi dari *array* label pelatihan.

X_test.shape: Menampilkan dimensi dari *array* fitur pengujian.

`Y_test.shape`: Menampilkan dimensi dari *array* label pengujian.

```
# Melatih model dengan data validasi
history = model.fit(X_train, Y_train, epochs=15,
batch_size=16, verbose=2, validation_data=(X_test,
Y_test))
```

`model.fit`: Fungsi ini digunakan untuk melatih model dengan data pelatihan (*X_train*, *Y_train*).

`epochs=20`: Menentukan jumlah iterasi pelatihan seluruh dataset.

`batch_size=16`: Menentukan ukuran *batch* untuk setiap pembaruan bobot.

`verbose=2`: Menampilkan status pelatihan secara rinci pada setiap *epoch*.

`validation_data=(X_test, Y_test)`: Menyediakan data validasi untuk mengevaluasi kinerja model pada data yang tidak terlihat selama pelatihan.

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

`history.history.keys()`: Menampilkan kunci dari dictionary yang berisi nilai-nilai metrik pelatihan dan validasi selama proses pelatihan, seperti *loss*, akurasi, dan metrik lainnya.

```
# 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)
```

`history.history['val_loss']`: Mengambil nilai *loss* pada data validasi untuk setiap *epoch*.

`history.history['val_acc']`: Mengambil nilai akurasi pada data validasi untuk setiap *epoch*.

`print("Validation Loss:", val_loss)`: Menampilkan nilai *loss* pada data validasi.

`print("Validation Accuracy:", val_acc)`: Menampilkan nilai akurasi pada data validasi.

```
# Mendapatkan histori pelatihan
history_dict = history.history
```

Baris ini mengambil sejarah pelatihan model yang disimpan dalam objek *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']
```

Baris-baris ini mengekstrak nilai loss dan accuracy untuk data pelatihan (*loss dan acc*) serta data validasi (*val_loss dan val_acc*).

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

Baris ini membuat rentang angka yang sesuai dengan jumlah *epoch* pelatihan, dimulai dari 1 hingga jumlah *epoch* yang ada.

```
# 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()
```

Bagian ini membuat plot untuk loss pelatihan dan validasi:

- Membuat figure dengan ukuran 12x4
- Membuat subplot untuk loss.
- Memplot nilai loss untuk pelatihan dan validasi.
- Menambahkan judul, label sumbu, dan legenda.

```
# 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()
```

Bagian ini membuat plot untuk *accuracy* pelatihan dan validasi:

- Membuat subplot untuk *accuracy*.
- Memplot nilai *accuracy* untuk pelatihan dan validasi.
- Menambahkan judul, label sumbu, dan legenda.

```
# 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
```

model.predict(X_test) digunakan untuk memprediksi probabilitas dari data uji.

(predictions > 0.5).astype(int) mengubah probabilitas menjadi label biner (0 atau 1) dengan *threshold* 0.5.

true_labels menyimpan label sebenarnya dari data uji.

```
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}")
```

f1_score, *precision_score*, dan *recall_score* digunakan untuk menghitung *F1 score*, *precision*, dan *recall* dengan rata-rata berbobot (*average='weighted'*), yang memperhitungkan proporsi setiap kelas dalam dataset.

Hasil metrik tersebut kemudian dicetak ke layar.

```
print(classification_report(true_labels,predicted_labels))
```

classification_report menampilkan laporan yang berisi *precision*, *recall*, dan *F1 score* untuk setiap kelas, serta *average/macro/micro values*.

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

Menampilkan array *true_labels* dan *predicted_labels* untuk melihat perbandingan antara label sebenarnya dan prediksi yang dihasilkan oleh model.

```
# 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

```

odel.predict(X_test) digunakan untuk memprediksi probabilitas dari data uji.

(predictions > 0.5).astype(int) mengubah probabilitas menjadi label biner (0 atau 1) dengan *threshold* 0.5.

true_labels menyimpan label sebenarnya dari data uji.

```

# Menghitung metrik evaluasi tambahan
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)))

```

true_labels.argmax(axis=1) dan ***predicted_labels.argmax(axis=1)*** digunakan untuk mendapatkan indeks kelas dari label satu-hot encoding.

f1_score, ***precision_score***, dan ***recall_score*** digunakan untuk menghitung ***F1 score***, ***precision***, dan ***recall*** dengan rata-rata berbobot (***average='weighted'***).

classification_report menampilkan laporan yang berisi ***precision***, ***recall***, dan ***F1 score*** untuk setiap kelas, serta ***average/macro/micro values***.

```

# 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)
})

```

Membuat DataFrame ***test_results*** yang berisi ulasan, label sebenarnya, dan prediksi label.

```

# 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)

```

Fungsi *classify_label* digunakan untuk mengubah nilai numerik label menjadi teks ('saran' atau 'kritik').

apply(classify_label) digunakan untuk mengubah kolom Label Sebenarnya dan Prediksi pada DataFrame *test_results*.

```

# Export ke Excel
test_results.to_excel('/content/drive/MyDrive/Skripsi/202
0/Ahmad Faisal RNN/hasil_prediksi10%.xlsx', index=False)
Menyimpan DataFrame test_results ke dalam file Excel di path yang ditentukan.

# Tampilkan hasil
print("\nHasil Prediksi:\n", test_results)
print("Data berhasil diekspor ke
'hasil_prediksi10%.xlsx'.")

```

Menampilkan hasil prediksi dalam bentuk DataFrame dan memberikan pesan bahwa data telah berhasil diekspor ke file Excel.

F. Hasil Pengujian Metode RNN

Dengan langkah-langkah ini, hasil prediksi pada data validasi telah ditampilkan dan disimpan ke dalam file Excel untuk dianalisis lebih lanjut atau dipresentasikan kepada pihak terkait. Dalam kode tersebut, dilakukan iterasi melalui daftar arsitektur neural network. Selama iterasi, program melatih dan menguji model neural network menggunakan setiap arsitektur yang ada dalam daftar tersebut.

Pembagian data dalam machine learning dapat dinyatakan sebagai persentase atau proporsi desimal. Pada pembagian **90:10** atau **0.1, 90%** data digunakan untuk pelatihan dan **10%** untuk pengujian, yang memberikan lebih banyak data pelatihan tetapi lebih

sedikit data untuk pengujian. Pembagian **80:20** atau **0.2**, dengan **80%** untuk pelatihan dan **20%** untuk pengujian, menawarkan keseimbangan yang baik antara keduanya. Pembagian **70:30** atau **0.3**, dengan **70%** untuk pelatihan dan **30%** untuk pengujian, memberikan lebih banyak data untuk evaluasi model, membantu memastikan generalisasi yang lebih baik.

1. Pembagian Data

a. 90 : 10

1) Epoch

```
Epoch 1/15
420/420 - 6s - loss: 0.5924 - acc: 0.7060 - val_loss: 0.4212 - val_acc: 0.8271 - 6s/epoch - 13ms/step
Epoch 2/15
420/420 - 4s - loss: 0.3486 - acc: 0.8714 - val_loss: 0.2550 - val_acc: 0.9142 - 4s/epoch - 10ms/step
Epoch 3/15
420/420 - 4s - loss: 0.2451 - acc: 0.9700 - val_loss: 0.2555 - val_acc: 0.9196 - 4s/epoch - 11ms/step
Epoch 4/15
420/420 - 4s - loss: 0.1971 - acc: 0.9380 - val_loss: 0.2453 - val_acc: 0.9169 - 4s/epoch - 10ms/step
Epoch 5/15
420/420 - 4s - loss: 0.1905 - acc: 0.9387 - val_loss: 0.2530 - val_acc: 0.9196 - 4s/epoch - 11ms/step
Epoch 6/15
420/420 - 4s - loss: 0.1547 - acc: 0.9537 - val_loss: 0.2493 - val_acc: 0.9236 - 4s/epoch - 10ms/step
Epoch 7/15
420/420 - 4s - loss: 0.1322 - acc: 0.9590 - val_loss: 0.2496 - val_acc: 0.9196 - 4s/epoch - 10ms/step
Epoch 8/15
420/420 - 4s - loss: 0.1161 - acc: 0.9620 - val_loss: 0.2599 - val_acc: 0.9129 - 4s/epoch - 10ms/step
Epoch 9/15
420/420 - 4s - loss: 0.1153 - acc: 0.9642 - val_loss: 0.2723 - val_acc: 0.9008 - 4s/epoch - 10ms/step
Epoch 10/15
420/420 - 4s - loss: 0.1125 - acc: 0.9621 - val_loss: 0.2689 - val_acc: 0.9209 - 4s/epoch - 10ms/step
Epoch 11/15
420/420 - 4s - loss: 0.0837 - acc: 0.9753 - val_loss: 0.2871 - val_acc: 0.9196 - 4s/epoch - 10ms/step
Epoch 12/15
420/420 - 4s - loss: 0.0779 - acc: 0.9735 - val_loss: 0.3067 - val_acc: 0.9209 - 4s/epoch - 10ms/step
Epoch 13/15
420/420 - 4s - loss: 0.0745 - acc: 0.9770 - val_loss: 0.3570 - val_acc: 0.9102 - 4s/epoch - 10ms/step
Epoch 14/15
420/420 - 4s - loss: 0.0590 - acc: 0.9830 - val_loss: 0.3184 - val_acc: 0.9182 - 4s/epoch - 10ms/step
Epoch 15/15
420/420 - 4s - loss: 0.0591 - acc: 0.9812 - val_loss: 0.3531 - val_acc: 0.9102 - 4s/epoch - 10ms/step
dict keys(['loss', 'acc', 'val_loss', 'val_acc'])
Validation loss: [0.42120064163208, 0.25503015518188477, 0.2554517388343811, 0.2452619969844818, 0.2525
Validation accuracy: [0.7060000000000001, 0.8714000000000001, 0.9700000000000001, 0.9380000000000001, 0.9387000000000001, 0.9537000000000001, 0.9590000000000001, 0.9620000000000001, 0.9642000000000001, 0.9621000000000001, 0.9753000000000001, 0.9735000000000001, 0.9770000000000001, 0.9830000000000001, 0.9812000000000001]
```

Gambar 4. Epoch 1-15 0.1

Dari gambar di atas dapat dilihat bahwa pada pembagian data **90:10** dengan batch size 16, *Epoch* pertama (1), untuk Training Accuracy mencapai **70%** dan Validasi Accuracy **82%**.

2) Metrik

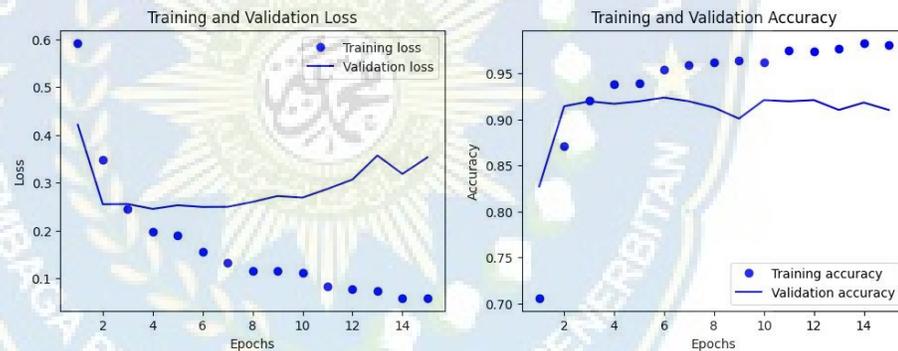
```
[x] 47/47 [=====] - 0s 4ms/step
F1 Score: 0.9042476634055933
Precision: 0.90438188389531
Recall: 0.9041554959785523
```

	precision	recall	f1-score	support
0	0.85	0.83	0.84	446
1	0.93	0.93	0.93	1046
micro avg	0.90	0.90	0.90	1492
macro avg	0.89	0.88	0.89	1492
weighted avg	0.90	0.90	0.90	1492
samples avg	0.90	0.90	0.90	1492

Gambar 5. Metrik 0.1

Gambar tersebut menunjukkan hasil evaluasi model klasifikasi dalam bentuk metrik kinerja. Tabel memperlihatkan nilai *precision*, *recall*, dan *f1-score* untuk dua kelas: 0 dan 1. Kelas 0 memiliki *precision* **0.85**, *recall* **0.83**, dan *f1-score* **0.84** dengan support **446**, sedangkan kelas 1 memiliki *precision* **0.93**, *recall* **0.93**, dan *f1-score* **0.93** dengan support 1046. Selain itu, tabel juga menampilkan nilai rata-rata mikro (*micro avg*), makro (*macro avg*), rata-rata berbobot (*weighted avg*), dan rata-rata sampel (*samples avg*), semuanya dengan nilai *precision*, *recall*, dan *f1-score* masing-masing sebesar **0.90**, **0.90**, dan **0.90** untuk *micro avg* dan *samples avg*, serta nilai sedikit lebih rendah untuk *macro avg* dan *weighted avg*. Hasil ini menunjukkan bahwa model memiliki kinerja yang baik dan konsisten dalam mengklasifikasikan data uji.

3) Grafik



Gambar 6. Grafik 0.1

Gambar di atas menunjukkan grafik yang menggambarkan perubahan nilai *loss* dan akurasi selama proses pelatihan dan validasi model *machine learning*. Grafik di sebelah kiri memperlihatkan nilai *loss* untuk pelatihan dan validasi terhadap jumlah *epoch*, dimana terlihat bahwa nilai *loss* pelatihan terus menurun, namun nilai *loss* validasi cenderung fluktuatif setelah mencapai titik terendah. Grafik di sebelah kanan menunjukkan perubahan akurasi untuk pelatihan dan validasi terhadap jumlah *epoch*. Akurasi pelatihan terus meningkat, sedangkan

akurasi validasi mencapai puncaknya sekitar *epoch* ke-3 dan kemudian mengalami sedikit *fluktuasi* tanpa peningkatan signifikan. Hal ini mungkin mengindikasikan bahwa model mulai mengalami *overfitting* setelah beberapa *epoch*.

4) Hasil prediksi

Tabel 5. Hasil Prediksi 0.1

Ulasan	Label Sebenarnya	Prediksi	
sebaiknya materi yang akan di pelajari terlebih dahulu dijelaskan agar mahasiswa dapat lebih mudah memahaminya	Saran	Saran	1
Mahasiswa sering tidak tahu bagaimana cara mengajukan keluhan tentang penilaian	Kritik	Kritik	1
Agar lebih tepat waktu lagi	Kritik	Kritik	1
Semoga kedepannya proses perkuliahan sudah bisa berjalan dengan baik	Saran	Saran	1
Untuk proses dalam memberikan pengajaran sudah sangat bagus dan juga materi yang di sampaikan sangat bagus hanya saja saran saya yaitu perlu di tingkatkan lagi suasana pada saat proses pembelajaran agar para mahasiswa tidak cepat bosan	Saran	Saran	1
Seharusnya dosen lebih memberi kebijakan dan toleransi dalam pembelajaran daring	Kritik	Saran	0
Menyediakan ruang khusus untuk kegiatan ekstrakurikuler dan organisasi mahasiswa	Saran	Saran	1
kedepannya mahasiswa bisa melakukan sistem pembelajaran di kampus agar dapat menikmati fasilitas yang ada	Saran	Saran	1

Pembelajaran berjalan cukup baik dan selalu tepat waktu akan tetapi perkuliahan kurang bervariasi karena hanya menggunakan GC dan sulit rasanya melakukan diskusi ketika cuma di GC	Kritik	Saran	0
Setiap urusan di TU pasti makan waktu lama	Kritik	Kritik	1
Kursi di kelas sering nggak nyaman bikin pegal	Kritik	Kritik	1
Fasilitas olahraga di kampus sering kali tidak terawat	Kritik	Kritik	1
Saya belum tahu di mana fasilitas Olahraga yang ada di lingkungan Unismuh saya belum tahu tunjangan selain gaji Jaminan Hari Tua Pensiunan sistemnya seperti apa belum ada pencerahan ke dosen-dosen dan Pegawai Sebaiknya hal-hal yang berkaitan dengan SDM disosialisasikan	Kritik	Saran	0
Komputer kampus sering rusak susah buat ngerjain tugas	Kritik	Kritik	1
Ketepatan waktu dalam perkuliahan	Saran	Saran	1
Kampus memberikan Rungan kelas tambahan untuk mahasiswanya	Saran	Saran	1
Ada satu pegawai tata usaha kurang ramah	Kritik	Kritik	1
Saran saya ialah dalam peningkatan pembelajaran akademik untuk selanjutnya agar lebih baik lagi untuk kedepannya perlu ditingkatkan lagi dalam proses belajar mengajar	Saran	Saran	1
Semoga cara mengajarnya bagus dan smoga mempertahankan dalam memberikan materi atau apapun itu	Saran	Saran	1

Tingkatkan fast respon ya kakakkakak	Saran	Saran	1
Fasilitas laboratorium kurang lengkap dan peralatannya banyak yang usang	Kritik	Kritik	1
Jumlah label yang benar			677
Accucay			0.90

Gambar di atas menunjukkan daftar komentar atau umpan balik dari mahasiswa mengenai berbagai aspek kampus mereka, seperti proses pembelajaran, fasilitas, dan pelayanan. Kolom A berisi teks komentar, kolom B dan C berisi label 'kritik' atau 'saran', dan kolom D berisi angka biner (0 atau 1) yang mungkin menunjukkan klasifikasi atau kategori dari komentar tersebut. Di bagian bawah tabel, terdapat jumlah total dari nilai di kolom D (**677**) dan rata-rata atau proporsi tertentu (**0.907507**).

b. **80 : 20**

1) *Epoch*

```

Epoch 1/15
373/373 - 6s - loss: 0.6218 - acc: 0.6950 - val_loss: 0.5278 - val_acc: 0.7218 - 6s/epoch - 15ms/step
Epoch 2/15
373/373 - 4s - loss: 0.3904 - acc: 0.8397 - val_loss: 0.3043 - val_acc: 0.8975 - 4s/epoch - 10ms/step
Epoch 3/15
373/373 - 4s - loss: 0.2627 - acc: 0.9130 - val_loss: 0.2641 - val_acc: 0.9048 - 4s/epoch - 11ms/step
Epoch 4/15
373/373 - 4s - loss: 0.2191 - acc: 0.9274 - val_loss: 0.2472 - val_acc: 0.9202 - 4s/epoch - 10ms/step
Epoch 5/15
373/373 - 4s - loss: 0.1803 - acc: 0.9418 - val_loss: 0.2515 - val_acc: 0.9202 - 4s/epoch - 10ms/step
Epoch 6/15
373/373 - 4s - loss: 0.1542 - acc: 0.9524 - val_loss: 0.2585 - val_acc: 0.9155 - 4s/epoch - 11ms/step
Epoch 7/15
373/373 - 4s - loss: 0.1274 - acc: 0.9584 - val_loss: 0.2790 - val_acc: 0.9102 - 4s/epoch - 11ms/step
Epoch 8/15
373/373 - 4s - loss: 0.1172 - acc: 0.9656 - val_loss: 0.2805 - val_acc: 0.9209 - 4s/epoch - 11ms/step
Epoch 9/15
373/373 - 4s - loss: 0.0973 - acc: 0.9703 - val_loss: 0.3050 - val_acc: 0.9169 - 4s/epoch - 11ms/step
Epoch 10/15
373/373 - 4s - loss: 0.0788 - acc: 0.9750 - val_loss: 0.3285 - val_acc: 0.8968 - 4s/epoch - 11ms/step
Epoch 11/15
373/373 - 4s - loss: 0.0800 - acc: 0.9728 - val_loss: 0.3038 - val_acc: 0.9068 - 4s/epoch - 11ms/step
Epoch 12/15
373/373 - 4s - loss: 0.0580 - acc: 0.9832 - val_loss: 0.3260 - val_acc: 0.9048 - 4s/epoch - 11ms/step
Epoch 13/15
373/373 - 4s - loss: 0.0516 - acc: 0.9844 - val_loss: 0.4559 - val_acc: 0.8861 - 4s/epoch - 11ms/step
Epoch 14/15
373/373 - 4s - loss: 0.0533 - acc: 0.9847 - val_loss: 0.3387 - val_acc: 0.9142 - 4s/epoch - 11ms/step
Epoch 15/15
373/373 - 4s - loss: 0.0461 - acc: 0.9851 - val_loss: 0.3723 - val_acc: 0.9062 - 4s/epoch - 11ms/step
dict_keys(['loss', 'acc', 'val_loss', 'val_acc'])
validation loss: [0.5278254151344299, 0.304330050945282, 0.26413100957870483, 0.24721147119988932, 0.2514709532260895, 0.25850147008895874, 0.27902042865

```

Gambar 7. *Epoch* 1-15 0.2

Dari gambar di atas dapat dilihat bahwa pada pembagian data **80:20** dengan *batch size* 16, *Epoch* pertama (1), untuk Training Accuracy mencapai **69%** dan Validasi Accuracy **72%**.

2) Metrik

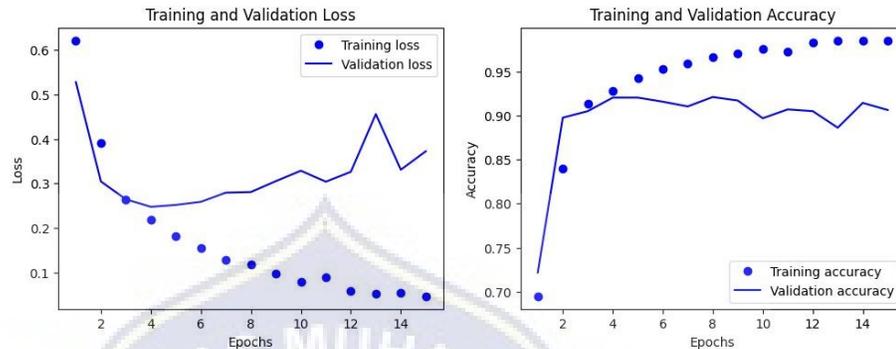
```
47/47 [=====] - 0s 4ms/step
F1 Score: 0.9042476634055933
Precision: 0.90438188389531
Recall: 0.9041554959785523
```

	precision	recall	f1-score	support
0	0.85	0.83	0.84	446
1	0.93	0.93	0.93	1046
micro avg	0.90	0.90	0.90	1492
macro avg	0.89	0.88	0.89	1492
weighted avg	0.90	0.90	0.90	1492
samples avg	0.90	0.90	0.90	1492

Gambar 8. Metrik 0.2

Gambar tersebut menunjukkan hasil evaluasi model klasifikasi dalam bentuk metrik kinerja. Tabel memperlihatkan nilai *precision*, *recall*, dan *f1-score* untuk dua kelas: 0 dan 1. Kelas 0 memiliki *precision* **0.85**, *recall* **0.83**, dan *f1-score* **0.84** dengan support **446**, sedangkan kelas 1 memiliki *precision* **0.93**, *recall* **0.93**, dan *f1-score* **0.93** dengan support **1046**. Selain itu, tabel juga menampilkan nilai rata-rata mikro (*micro avg*), makro (*macro avg*), rata-rata berbobot (*weighted avg*), dan rata-rata sampel (*samples avg*), semuanya dengan nilai *precision*, *recall*, dan *f1-score* masing-masing sebesar **0.90**, **0.90**, dan **0.90** untuk *micro avg* dan *samples avg*, serta nilai sedikit lebih rendah untuk *macro avg* dan *weighted avg*. Hasil ini menunjukkan bahwa model memiliki kinerja yang baik dan konsisten dalam mengklasifikasikan data uji.

3) Grafik



Gambar 9. Grafik 0.2

Gambar tersebut menunjukkan dua grafik yang membandingkan metrik pelatihan dan validasi model pembelajaran mesin selama beberapa *epoch*. Grafik di sebelah kiri menampilkan *Loss* (kerugian) dengan sumbu y dan *Epochs* dengan sumbu x. Terlihat bahwa kerugian pelatihan terus menurun, sementara kerugian validasi menurun hingga sekitar *epoch* 8, kemudian mengalami *fluktuasi*. Grafik di sebelah kanan menampilkan *Accuracy* (akurasi) dengan sumbu y dan *Epochs* dengan sumbu x. Akurasi pelatihan meningkat dan mendekati 1, sementara akurasi validasi meningkat hingga sekitar *epoch* 4, kemudian *fluktuatif* dengan kecenderungan mendatar. Hal ini menunjukkan model mengalami *overfitting*, di mana kinerja pada data pelatihan meningkat terus, tetapi kinerja pada data validasi tidak stabil dan cenderung menurun setelah beberapa *epoch*.

4) Hasil prediksi

Tabel 6. Hasil Prediksi 0.2

Ulasan	Label Sebenarnya	Prediksi	
Sudah baguspembawaan materinya tetap di pertahankan saja dan kalau bisa dalam proses pembelajaran	Saran	Kritik	0

suasananya dosennya bisa membuat suasananya tidak terlalu tegang sehingga mahasiswa juga senang dalam menerima materi			
lebih baik di adakan perkuliahan tatap muka	Saran	Saran	1
Jam kuliah yang singkat sering sekali terasa kurang	Kritik	Kritik	1
Semoga kedepannya lebih baik	Saran	Saran	1
Cara menerangkan lebih diperjelas	Saran	Saran	1
Pemberian materi sudah bagus penjelasannya pun sudah bagus yang perlu ditingkatkan komunikasi antara mahasiswa dan dosen	Saran	Saran	1
Semoga dapat secepatnya melakukan perkuliahan secara offline agar pembelajaran lebih mudah di pahami	Saran	Saran	1
Kenyamanan saat proses pembelajaran harus di tingkatkan karena sangat berpengaruh pada hasil yang di dapat mahasiswa	Saran	Saran	1
Dosen sebaiknya menyediakan waktu untuk sesi tanya jawab setelah kuliah selesai	Saran	Saran	1
akreditasi bisa ditingkatkan lagi	Saran	Saran	1
memberikana solusi atas mahasiswa yang kurang dalam pembelajaran	Saran	Saran	1
Saran sy jgn hanya mengirim materi dan disuruh baca terimakasih	Saran	Saran	1
Beberapa area kampus masih rawan pencurian	Kritik	Kritik	1
Akses ke fasilitas olahraga di kampus susah banget	Kritik	Kritik	1

Saya harap di semester berikutnya bisa kuliah secara offline agar akademik semakin meningkat	Saran	Saran	1
Pembelajaran efektif harus di tingkatkan	Saran	Saran	1
Mahasiswa sering melihat coretan di dinding kampus	Kritik	Kritik	1
Jumlah label yang benar			1350
Accucay			0.90

Gambar di atas menunjukkan daftar komentar atau umpan balik dari mahasiswa mengenai berbagai aspek kampus mereka, seperti proses pembelajaran, fasilitas, dan pelayanan. Kolom A berisi teks komentar, kolom B dan C berisi label 'kritik' atau 'saran', dan kolom D berisi angka biner (0 atau 1) yang mungkin menunjukkan klasifikasi atau kategori dari komentar tersebut. Di bagian bawah tabel, terdapat jumlah total dari nilai di kolom D (**1350**) dan rata-rata atau proporsi tertentu (**0.90483**).

c. 70 : 30

1) Epoch

```

[68] epoch 1/15
327/327 - 5s - loss: 0.5974 - acc: 0.7099 - val_loss: 0.4267 - val_acc: 0.8248 - 5s/epoch - 15ms/step
Epoch 2/15
327/327 - 4s - loss: 0.3541 - acc: 0.8647 - val_loss: 0.2745 - val_acc: 0.9021 - 4s/epoch - 11ms/step
Epoch 3/15
327/327 - 4s - loss: 0.2679 - acc: 0.9088 - val_loss: 0.2633 - val_acc: 0.9057 - 4s/epoch - 11ms/step
Epoch 4/15
327/327 - 4s - loss: 0.2269 - acc: 0.9214 - val_loss: 0.2572 - val_acc: 0.9066 - 4s/epoch - 11ms/step
Epoch 5/15
327/327 - 4s - loss: 0.1812 - acc: 0.9439 - val_loss: 0.2710 - val_acc: 0.9048 - 4s/epoch - 11ms/step
Epoch 6/15
327/327 - 4s - loss: 0.1586 - acc: 0.9492 - val_loss: 0.2448 - val_acc: 0.9177 - 4s/epoch - 11ms/step
Epoch 7/15
327/327 - 4s - loss: 0.1302 - acc: 0.9600 - val_loss: 0.2560 - val_acc: 0.9191 - 4s/epoch - 11ms/step
Epoch 8/15
327/327 - 3s - loss: 0.1084 - acc: 0.9532 - val_loss: 0.2787 - val_acc: 0.9093 - 3s/epoch - 11ms/step
Epoch 9/15
327/327 - 4s - loss: 0.1156 - acc: 0.9642 - val_loss: 0.2536 - val_acc: 0.9173 - 4s/epoch - 11ms/step
Epoch 10/15
327/327 - 4s - loss: 0.0973 - acc: 0.9693 - val_loss: 0.2735 - val_acc: 0.9200 - 4s/epoch - 11ms/step
Epoch 11/15
327/327 - 4s - loss: 0.0869 - acc: 0.9732 - val_loss: 0.2993 - val_acc: 0.9115 - 4s/epoch - 11ms/step
Epoch 12/15
327/327 - 4s - loss: 0.0882 - acc: 0.9720 - val_loss: 0.3061 - val_acc: 0.9061 - 4s/epoch - 11ms/step
Epoch 13/15
327/327 - 4s - loss: 0.0739 - acc: 0.9770 - val_loss: 0.3357 - val_acc: 0.9008 - 4s/epoch - 11ms/step
Epoch 14/15
327/327 - 4s - loss: 0.0650 - acc: 0.9808 - val_loss: 0.2815 - val_acc: 0.9200 - 4s/epoch - 12ms/step
Epoch 15/15
327/327 - 4s - loss: 0.1032 - acc: 0.9680 - val_loss: 0.3385 - val_acc: 0.9115 - 4s/epoch - 11ms/step
dict keys(['loss', 'acc', 'val_loss', 'val_acc'])
validation loss: [0.4266500174999237, 0.2745116055011749, 0.2632636427879335, 0.2572135659713745, 0.27101102471351624, 0.24482442438602448, 0.25595033966064]
✓ 0d total pada 01:13

```

Gambar 10. Epoch 1-15 0.3

Dari gambar di atas dapat dilihat bahwa pada pembagian data **80:20** dengan *batch size* 16, *Epoch* pertama (1), untuk Training *Accuracy* mencapai **70%** dan Validasi *Accuracy* **82%**

2) Metrik



```

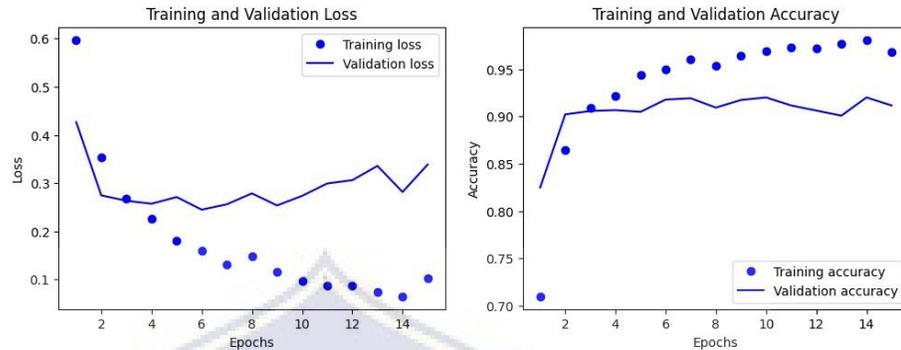
70/70 [
F1 Score: 0.9122293742772783
Precision: 0.912628665868699
Recall: 0.911995629873125
precision  recall  f1 score  support
0         0.84   0.86   0.85     674
1         0.94   0.93   0.94    1563
micro avg 0.91   0.91   0.91    2237
macro avg 0.89   0.90   0.90    2237
weighted avg 0.91   0.91   0.91    2237
samples avg 0.91   0.91   0.91    2237

```

Gambar 11. Metrik 0.3

Gambar tersebut menunjukkan hasil evaluasi model klasifikasi dalam bentuk metrik kinerja. Tabel memperlihatkan nilai *precision*, *recall*, dan *f1-score* untuk dua kelas: 0 dan 1. Kelas 0 memiliki *precision* **0.84**, *recall* **0.86**, dan *f1-score* **0.85** dengan support **674**, sedangkan kelas 1 memiliki *precision* **0.94**, *recall* **0.93**, dan *f1-score* **0.94** dengan support **1563**. Selain itu, tabel juga menampilkan nilai rata-rata mikro (*micro avg*), makro (*macro avg*), rata-rata berbobot (*weighted avg*), dan rata-rata sampel (*samples avg*), semuanya dengan nilai *precision*, *recall*, dan *f1-score* masing-masing sebesar **0.91**, **0.91**, dan **0.91** untuk *micro avg* dan *samples avg*, serta nilai sedikit lebih rendah untuk *macro avg* dan *weighted avg*. Hasil ini menunjukkan bahwa model memiliki kinerja yang baik dan konsisten dalam mengklasifikasikan data uji.

3) Grafik



Gambar 12. Grafik 0.3

Gambar tersebut menunjukkan dua grafik yang membandingkan metrik pelatihan dan validasi untuk model pembelajaran mesin selama beberapa *epoch*. Grafik di sebelah kiri menunjukkan *loss* (kerugian) dengan sumbu y dan *epochs* dengan sumbu x. Terlihat bahwa *loss* pelatihan (ditandai dengan titik-titik biru) terus menurun secara konsisten, sementara *loss* validasi (ditandai dengan garis biru) menurun hingga sekitar *epoch* ke-8, kemudian mengalami *fluktuasi* yang cukup signifikan. Grafik di sebelah kanan menunjukkan akurasi dengan sumbu y dan *epochs* dengan sumbu x. Akurasi pelatihan (ditandai dengan titik-titik biru) meningkat secara stabil hingga mendekati 1, sementara akurasi validasi (ditandai dengan garis biru) meningkat hingga sekitar *epoch* ke-4, kemudian *fluktuatif* dengan kecenderungan mendatar dan sedikit menurun setelah *epoch* ke-10. Pola ini menunjukkan adanya kemungkinan *overfitting*, di mana model menunjukkan kinerja yang sangat baik pada data pelatihan tetapi tidak konsisten pada data validasi.

4) Hasil prediksi

Tabel 7. Hasil prediksi 0.3

Ulasan	Label Sebenarnya	Prediksi	
Banyak kaca jendela yang buram dan tidak pernah dibersihkan	Kritik	Kritik	1

Harus lebih tepat waktu dan sesuai jadwal perkuliahan	Saran	Kritik	0
Semoga lebih baik lagi kedepannya	Saran	Saran	1
semoga kedepannya menjadi lebih baik	Saran	Saran	1
Saran agar bapak dosen bisa lebih tegas terhadap mahasiswatanpa menghilangkan kebiasaan atau metode belajar bapak yang unik	Saran	Saran	1
Semoga lebih baik kedepannya	Saran	Saran	1
Semoga pertemuan selanjutnya bisa offline supaya pembelajarannya lebih mudah untuk dipahami	Saran	Saran	1
Dari segi kedisiplinan dosen dalam proses belajar mengajar sudah sesuai Rencana Pembelajaran Semester Untuk masalah sarana alat proses mengajar yang kurang memadai contohnya LCD Terima kasih	Kritik	Kritik	1
Mohon maaf sebelumnya saran saya dalam hal ini yaitu perlunya pembelajaran tatap muka sehingga mata kuliah dapat dipahami dengan baik Maaf bila ada salah kata dan Terima kasih	Saran	Saran	1
Semoga kedepannya pembelajaran bisa dilakukan secara tatap muka agar materinya akan lebih mudah lagi untuk dipahami dari sebelumnya	Saran	Saran	1
Perlu monev integrasi AIK sbg penciri Muhammadiyah	Saran	Saran	1
Semoga kedepannya perkuliahan bisa terlaksana sesuai ketentuan yang berlaku	Saran	Saran	1
Pelayanan dan sarpras semakin baik dan lebih ditingkatkan lagi	Saran	Saran	1

Sistem pelayanan terintegrasi tetap dijaga dan ditingkatkan terutama dalam pemanfaatan ICT disemua sektor pelayanan	Saran	Saran	1
supaya kuliah of line sebaiknya dilakukan	Saran	Saran	1
Materi kuliah digital hampir tidak tersedia di perpustakaan	Kritik	Kritik	1
Banyak ruang kelas yang tidak dilengkapi dengan alat bantu pengajaran yang memadai	Kritik	Kritik	1
Jumlah label yang benar			2038
Accucay			0.91

Gambar di atas menunjukkan daftar komentar atau umpan balik dari mahasiswa mengenai berbagai aspek kampus mereka, seperti proses pembelajaran, fasilitas, dan pelayanan. Kolom A berisi teks komentar, kolom B dan C berisi label 'kritik' atau 'saran', dan kolom D berisi angka biner (0 atau 1) yang mungkin menunjukkan klasifikasi atau kategori dari komentar tersebut. Di bagian bawah tabel, terdapat jumlah total dari nilai di kolom D (**2038**) dan rata-rata atau proporsi tertentu (**0.911042**).

BAB V

PENUTUP

A. Kesimpulan

1. Penelitian ini berhasil mengimplementasikan Metode *Recurrent Neural Network* (RNN) untuk klasifikasi saran dan kritik pada SIMAK UNISMUH dengan menggunakan 7.457 data. Hasil evaluasi menunjukkan bahwa model mencapai akurasi tertinggi sebesar 91% dan akurasi terendah sebesar 90%. Meskipun terdapat variasi dalam performa model, hasil ini menunjukkan bahwa RNN memiliki potensi yang baik dalam mengklasifikasikan teks saran dan kritik.
2. Model RNN dapat membantu institusi dalam memahami dan merespon masukan dari pengguna dengan lebih efektif.

B. Saran

Untuk meningkatkan performa model, disarankan untuk melakukan pengembangan lebih lanjut dengan mencoba berbagai arsitektur RNN lainnya seperti LSTM atau GRU, serta melakukan optimasi *hyperparameter* dan penambahan data pelatihan.

TINJAUAN 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>
- Adanson, Y. N., & Fitriana, G. F. (2022). Rancang Bangun Aplikasi Midesk Berbasis Website Menggunakan Metode User Centered Design. *Jurnal Sistem Komputer Dan Informatika (JSON)*, 4(1). <https://doi.org/10.30865/json.v4i1.4376>
- Afandi, W., Saputro, S. N., Kusumaningrum, A. M., Ardiansyah, H., Kafabi, M. H., Sudianto, S., Informatika, J. T., Informatika, F., & Purwokerto, K. (2022). Klasifikasi Judul Berita Clickbait menggunakan RNN-. 7(2), 85–89.
- Agustiandra, V., & Sabandi, A. (2019). Persepsi Guru Terhadap Penerapan Sistem Informasi Manajemen Di Sekolah Menengah Kejuruan (Smk) Negeri 3 Padang. *Jurnal Bahana Manajemen Pendidikan*, 8(1), 1. <https://doi.org/10.24036/bmp.v8i1.103704>
- Ardiansyah, D., Saepudin, A., Aryanti, R., & Fitriani, E. (2020). Perancangan Sistem Informasi Akademik Madrasah Aliyah Negeri (Man) 4 Karawang Berbasis Web. *Jurnal Teknologi Dan Open Source*, 3(2), 187–201. <https://doi.org/10.36378/jtos.v3i2.783>
- Bai, Y., Xie, J., Liu, C., Tao, Y., Zeng, B., & Li, C. (2021). Regression modeling for enterprise electricity consumption: A comparison of recurrent neural network and its variants. *International Journal of Electrical Power and Energy Systems*, 126. <https://doi.org/10.1016/j.ijepes.2020.106612>
- Fajar Ramadhan, D., Noertjahjono, S., & Dedy Irawan, J. (2020). PENERAPAN CHATBOT AUTO REPLY PADA WHATSAPP SEBAGAI PUSAT INFORMASI PRAKTIKUM MENGGUNAKAN ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 4(1). <https://doi.org/10.36040/jati.v4i1.2375>

- Firmansyah, M. R., Ilyas, R., & Kasyidi, F. (2020). Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network. *Prosiding The 11th Industrial Research Workshop and National Seminar*, 11(1).
- 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>
- Map, J., Ilyasi, A., Susanto, H., & Astuti, S. J. W. (2020). PELAKSANAAN SISTEM ADMINISTRASI AKADEMIK DI UNIVERSITAS ISLAM JEMBER. *MAP (Jurnal Manajemen Dan Administrasi Publik)*, 3(2). <https://doi.org/10.37504/map.v3i2.245>
- 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).
- Nurashila, S. S., Hamami, F., & Kusumasari, T. F. (2023). Perbandingan Kinerja Algoritma Recurrent Neural Network (Rnn) Dan Long Short-Term Memory (Lstm): Studi Kasus Prediksi Kemacetan Lalu Lintas Jaringan Pt Xyz. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 8(3), 864–877. <https://doi.org/10.29100/jupi.v8i3.3961>
- 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.
- Ma, Zhifeng, Hao Zhang, dan Jie Liu. 2023. “MM-RNN: A Multimodal RNN for Precipitation Nowcasting.” *IEEE Transactions on Geoscience and Remote Sensing* 61. doi: 10.1109/TGRS.2023.3264545.
- Sherstinsky, Alex. 2020. “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network.” *Physica D: Nonlinear Phenomena* 404. doi: 10.1016/j.physd.2019.132306.
- Rozi, I. F., Wijayaningrum, V. N., & Khozin, N. (2020). KLASIFIKASI TEKS LAPORAN MASYARAKAT PADA SITUS LAPOR! MENGGUNAKAN RECURRENT NEURAL NETWORK. *SISTEMASI*, 9(3). <https://doi.org/10.32520/stmsi.v9i3.977>

- Rumaisa, F., Puspitarani, Y., Rosita, A., Zakiah, A., & Violina, S. (2021). Penerapan Natural Language Processing (NLP) di bidang pendidikan. *Jurnal Inovasi Masyarakat*, 1(3), 232–235. <https://doi.org/10.33197/jim.vol1.iss3.2021.799>
- Tarkus, E. D., Sompie, S. R. U. A., Jacobus, A., Elektro, T., Sam, U., Manado, R., & Bahu, J. K. (2020). *Implementasi Metode Recurrent Neural Network pada Pengklasifikasian Kualitas Telur Puyuh*. 15(2), 137–144.
- Yusny, R., & Yasa, G. I. (2019). MENGEMBANGKAN (PEMBELAJARAN) BLENDED LEARNING DENGAN SISTEM LINGKUNGAN PEMBELAJARAN VIRTUAL (VLE) DI PTKIN. *Jurnal Ilmiah Islam Futura*, 19(1). <https://doi.org/10.22373/jiif.v19i1.3707>



LAMPIRAN

Dataset Set

	ULASAN	LABEL
1	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
2	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
3	Memperbaiki yang baik menjadi lebih baik lagi.	Saran
4	Sebaiknya pihak kampus memantau proses penyaluran kouta 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
5	Perlunya 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	Saran saya agar proses pembelajaran secara online harus juga di selingi dengan pembelajaran secara offline agar para mahasiswa lebih mengerti lagi materi perkuliahan.	Saran
8	Tetap pertahankan pembelajaran yang baik dan tingkat terus-menerus kedepannya.	Saran
9	Semoga sarana dan prasarana kampus dapat digunakan secepatnya dan semoga kuliah online tidak lagi dilaksanakan.	Saran
10	Semoga pembelajarannya semakin merdeka dan bisa offline terutama pada saat praktikum.	Saran

11	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
12	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 : Jelek Jaringan, Sakit Mata, Dan Pembelian Kuota Tiap Bulan. Saya Berharap Kedepannya Sudah Bisa Melakukan Pembelajaran Tatap Muka.	Kritik
13	Semoga dengan adanya peningkatan pelayanan baik itu dalam proses perkuliahan, pembelajaran ataupun dari pelayanan akademik akan menjadi jauh lebih baik untuk kedepannya.	Saran
14	Peningkatan fasilitas kampus.	Kritik
15	Saran saya agar lebih ditingkatkan lagi proses pembelajaran daring yang lebih baik lagi.	Saran
16	Karena proses perkuliahan masih daring, diharapkan agar sering mengingatkan kepada mahasiswa masalah koneksi ketika ada kuis/UTS/UAS agar mahasiswa bisa mempersiapkan kemungkinan buruk yang terjadi terutama masalah koneksi internet.	Saran
17	Semoga para dosen bisa menjelaskan materi dengan kreatif agar mahasiswa mudah memahami.	Saran
18	Semoga lebih baik kedepannya.	Saran
19	Saran saya semoga bisa offline.	Saran

	ULASAN	LABEL
7437	Penilaian tugas sering bikin bingung, nggak jelas.	Kritik
7438	Syarat administrasi ribet, bikin pusing.	Kritik
7439	Administrasi akademik nggak ada panduan jelas, bingung.	Kritik

7440	Ruang tunggu di kampus kotor dan berantakan.	Kritik
7441	fasilitas belajar dalam kelas dilengkapi	Saran
7442	lebih baik dalam proses mengajar dan belajar	Saran
7443	Proses perkuliahan berjalan dengan lancar, namun lebih bagus jika sering-sering bertatapan langsung dengan mahasiswa	Saran
7444	Apabila dosen berhalangan untuk mengajar agar sekiranya dosen yang bersangkutan konfirmasi terlebih dahulu kepada ketua tingkat sebelum jadwal perkuliahan	Saran
7445	Saran untuk proses pembelajaran pada program studi Manajemen yaitu dosen dapat membangkitkan motivasi mahasiswa yang merasa tidak bersemangat saat mengikuti perkuliahan terutama dalam pembelajaran daring ini	Saran
7446	lebih mengutamakan waktu mengajar	Saran
7447	Lebih efektif kedepannya	Saran
7448	Semoga fasilitas belajar dalam kelas dilengkapi	Saran
7449	Disarankan agar dosen bisa lebih berbaur dan bersahabat dengan semua mahasiswa.	Saran
7450	Tolong leboh mempertimbangkam hasil dari kuliah harian jangan cuma fokus terhadap fainal	Saran
7451	Sudah cukup baik, tapi jarang ada mahasiswa yang kurang memahami karna faktor kendala dlam jringan	Kritik
7452	Perlu menghadirkan variasi mengajar baru, seperti menyajikan materi perkuliahan yang menarik.	Saran
7453	KULIAH OFFLINE segera dilaksanakan, sehingga proses penerimaan materi mudah dipahami mahasiswa. Krna menurut saya sangat tidak maksimal pengajaran online ini.	Saran
7454	TETAP OBJEKTIF DALAM PEMBERIAN NILAI AGAR KEDEPANNYA TETAP TERCERMIN SIKAP MANDIRI DAN	Saran

	KETELADANAN MAHASISWA MAUPUN DOSEN PENGAJAR, KARNA SESUATU KECURANGAN AKAN MENGHASILKAN BIBIT GAGAL YANG TIDAK BAIK UNTUK DIPASARKAN.	
7455	Mohon pengarahannya terhadap buku apa saja yang harus di baca guna meningkatkan kualitas dan kemampuan dalam berdiskusi perihal mata kuliah ini	Saran
7456	jadi dosen yang tidak mempersulit mahasiswa nya	Saran
7457	Pelaksanaan kuliah yg saat ini masi online saya rasa berpengaruh terhadap proses pemahaman pembelajaran yg kurang mudah untuk dicerna saya harap kedepannya dapat segera dilaksanakan perkuliahan offline supaya pemahaman dalam proses pembelajaran dapat sedikit dipahami dan dimengerti.	Saran

Data klasifikasi

A. Pembagian data 90 : 10

	Ulasan	Label Sebenarnya	Prediksi	
1	Pengurusan berkas sering kali terhambat karena masalah teknis yang tidak segera diatasi	Kritik	Kritik	1
2	Tetap melayani dengan baik	Saran	Saran	1
3	Hadirkan dosen dan siap dalam mengajar	Kritik	Saran	0
4	Menyediakan materi kuliah dalam berbagai format seperti video teks dan audio	Saran	Saran	1
5	tabe dok mungkin lebih meningkatkan lagi proses pembelajaran secara kelompok	Saran	Saran	1
6	Hadirkan quiz di setiap akhir pembelajaran	Saran	Saran	1

7	Staf tata usaha sering terlihat tidak peduli terhadap kepentingan mahasiswa	Kritik	Kritik	1
8	Sebaiknya seluruh ruangan dalam proses belajar mengajar dilengkapi fasilitas berupa tiap ruang kelas memiliki LCD dan AC	Saran	Saran	1
9	Perpustakaan online bisa diaktifkan kembali dengan koleksi buku yang lebih banyak dan bervariasi	Saran	Saran	1
10	Tugas kelompok tidak terorganisir dengan baik	Kritik	Kritik	1
11	Pintu kelas susah dibuka bikin ribet	Kritik	Kritik	1
12	Mahasiswa sering kesulitan mengetahui jadwal kegiatan kampus	Kritik	Kritik	1
13	Saran saya lebih atif lagi dalam menjelaskan pada saat pelajaran mata kuliah berlangsung	Saran	Saran	1
14	Pada kenyataannya perkuliahan dilaksanakan secara daring dikarenakan adanya pandemi covid ini yang terus saja melanda Indonesia sehingga kita perlu meningkatkan perkuliahan secara daring untuk meningkat rasa kepuasan dan rasa nyaman mahasiswa untuk terus melaksanakan perkuliahan secara daring	Saran	Saran	1
15	Fasilitas pengolahan limbah kampus tidak memadai	Kritik	Kritik	1
16	Masih perlu tambahan penjelasan dalam mata kuliah	Kritik	Saran	0
17	Semoga pembelajarannya kedepanya lebih efektif	Saran	Saran	1

18	Perlunya menghargai para dosen dalam aktivitas pengrusan administrasi di kampus sendiri	Saran	Saran	1
19	Perlunya menghargai para dosen dalam aktivitas pengrusan administrasi di kampus sendiri	Kritik	Kritik	1
20	Ada baiknya proses praktikum secara langsung didalami secara inti Mempraktekkan beberapa komoditikomoditi perkebunan	Saran	Saran	1
	Ulasan	Label Sebenarnya	Prediksi	
727	sebaiknya materi yang akan di pelajari terlebih dahulu dijelaskan agar mahasiswa dapat lebih mudah memahaminya	Saran	Saran	1
728	Mahasiswa sering tidak tahu bagaimana cara mengajukan keluhan tentang penilaian	Kritik	Kritik	1
729	Agar lebih tepat waktu lagi	Kritik	Kritik	1
730	Semoga kedepannya proses perkuliahan sudah bisa berjalan dengan baik	Saran	Saran	1
731	Untuk proses dalam memberikan pengajaran sudah sangat bagus dan juga materi yang di sampaikan sangat bagus hanya saja saran saya yaitu perlu di tingkatkan lagi suasana pada saat proses pembelajaran agar para mahasiswa tidak cepat bosan	Saran	Saran	1
732	Seharusnya dosen lebih memberi kebijakan dan toleransi dalam pembelajaran daring	Kritik	Saran	0
733	Menyediakan ruang khusus untuk kegiatan ekstrakurikuler dan organisasi mahasiswa	Saran	Saran	1

734	kedepannya mahasiswa bisa melakukan sistem pembelajaran di kampus agar dapat menikmati fasilitas yang ada	Saran	Saran	1
735	Pembelajaran berjalan cukup baik dan selalu tepat waktu akan tetapi perkuliahan kurang bervariasi karna hanya menggunakan GC dan sulit rasanya melakukan diskusi ketika cuma di GC	Kritik	Saran	0
736	Setiap urusan di TU pasti makan waktu lama	Kritik	Kritik	1
737	Kursi di kelas sering nggak nyaman bikin pegal	Kritik	Kritik	1
738	Fasilitas olahraga di kampus sering kali tidak terawat	Kritik	Kritik	1
739	Saya belum tahu di mana fasilitas Olahraga yang ada di lingkungan Unismuh saya belum tahu tunjangan selain gaji Jaminan Hari Tua Pensiunan sistemnya seperti apa belum ada pencerahan ke dosen-dosen dan Pegawai Sebaiknya hal-hal yang berkaitan dengan SDM disosialisasikan	Kritik	Saran	0
740	Komputer kampus sering rusak susah buat ngerjain tugas	Kritik	Kritik	1
741	Ketepatan waktu dalam perkuliahan	Saran	Saran	1
742	Kampus memberikan Rungan kelas tambahan untuk mahasiswanya	Saran	Saran	1
743	Ada satu pegawai tata usaha kurang ramah	Kritik	Kritik	1
744	Saran saya ialah dalam peningkatan pembelajaran akademik untuk selanjutnya agar lebih baik lagi untuk kedepannya perlu ditingkatkan lagi dalam proses belajar mengajar	Saran	Saran	1

745	Semoga cara mengajarnya bagus dan smoga mempertahankan dalam memberikan materi atau apapun itu	Saran	Saran	1
746	Tingkatkan fast respon ya kakakkakak	Saran	Saran	1
747	Fasilitas laboratorium kurang lengkap dan peralatannya banyak yang usang	Kritik	Kritik	1
Jumlah label yang benar				677
Accucay				0.90

B. Pembagian Data 80:20

	Ulasan	Label Sebenarnya	Prediksi	
21	Semoga semester kedepan menjadi lebih baik dari sebelumnya	Saran	Saran	1
22	Staf tata usaha sering tidak ada di tempat saat dibutuhkan	Kritik	Kritik	1
23	Tetap selalu memotivasi mahasiswa dan memberikan respon apabila ada kendala	Saran	Saran	1
24	Kebutuhan perangkat Lab Komputer perlu di tingkatkan Perlunya laboratorium yang spesifik di bidangnya contoh Lab Data Science dll	Saran	Saran	1
25	Semoga kedepannya lebih baik lagi	Saran	Saran	1
26	Harapan kami semoga ke depan metode dalam proses belajar dapat lebih baik dan menghadirkan pengalaman baru	Saran	Saran	1
27	Kemudahan dalam proses pembelajaran selama daring	Saran	Saran	1
28	Semoga lebih baik lagi kedepannya	Saran	Saran	1

29	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	1
30	Semoga bisa lebih baik lagi	Saran	Saran	1
31	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	1
32	Mahasiswa tidak mendapatkan bantuan dalam menghadapi wawancara magang	Kritik	Kritik	1
33	Saran saya lebih atif lagi dalam menjelaskan pada saat pelajaran mata kuliah berlangsung	Saran	Saran	1
34	Saran saya semoga ibu bisa menerapkan metode pembelajaran yang lebih variatif	Saran	Saran	1
35	Semoga kedepannya sarana dalam proses pembelajaran dapat lebih ditingkatkan lagi	Saran	Saran	1
36	Pihak Kampus Sebaiknya Memantau Proses Penyaluran Kuota Internet	Kritik	Kritik	1
37	semoga kedepannya bisa lebih baik dan sering sering offline	Saran	Kritik	0
38	Saran saya adalah memaklumi jika ada mahasiswa yang terganggu jaringannya	Saran	Saran	1
39	Semoga lebih baik untuk kedepannya	Saran	Saran	1

40	sebaiknya meningkatkan pengaturan classroom mata kuliahnya agar mempermudah mahasiswa mengirim tugas terimakasih	Saran	Saran	1
	Ulasan	Label Sebenarnya	Prediksi	
1477	Sudah bagus pembawaan materinya tetap di pertahankan saja dan kalau bisa dalam proses pembelajaran suasananya dosennya bisa membuat suasananya tidak terlalu tegang sehingga mahasiswa juga senang dalam menerima materi	Saran	Kritik	0
1478	lebih baik di adakan perkuliahan tatap muka	Saran	Saran	1
1479	Jam kuliah yang singkat sering sekali terasa kurang	Kritik	Kritik	1
1480	Semoga kedepannya lebih baik	Saran	Saran	1
1481	Cara menerangkan lebih diperjelas	Saran	Saran	1
1482	Pemberian materi sudah bagus penjelasannya pun sudah bagus yang perlu ditingkatkan komunikasi antara mahasiswa dan dosen	Saran	Saran	1
1483	Semoga dapat secepatnya melakukan perkuliahan secara offline agar pembelajaran lebih mudah di pahami	Saran	Saran	1
1484	Kenyamanan saat proses pembelajaran harus di tingkatkan karena sangat berpengaruh pada hasil yang di dapat mahasiswa	Saran	Saran	1
1485	Dosen sebaiknya menyediakan waktu untuk sesi tanya jawab setelah kuliah selesai	Saran	Saran	1
1486	akreditasi bisa ditingkatkan lagi	Saran	Saran	1

1487	memberikana solusi atas mahasiswa yang kurang dalam pembelajaran	Saran	Saran	1
1488	Saran sy jgn hanya mengirim materi dan disuruh baca terimakasih	Saran	Saran	1
1489	Beberapa area kampus masih rawan pencurian	Kritik	Kritik	1
1490	Akses ke fasilitas olahraga di kampus susah banget	Kritik	Kritik	1
1491	Saya harap di semester berikutnya bisa kuliah secara offline agar akademik semakin meningkat	Saran	Saran	1
1492	Pembelajaran efektif harus di tingkatkan	Saran	Saran	1
1493	Mahasiswa sering melihat coretan di dinding kampus	Kritik	Kritik	1
	Jumlah label yang benar			1350
	Accucay			0.90

Pembagian Data 70:30

	Ulasan	Label Sebenarnya	Prediksi	
41	Semoga fasilitas di kampus dapat di perbaharui	Saran	Saran	1
42	Semoga fasilitas di kampus dapat di perbaharui	Kritik	Kritik	1
43	Mahasiswa sering kali harus menunggu lama karena keterbatasan staf	Kritik	Kritik	1
44	Semoga kedepannya pembelajaran bisa dilakukan secara tatap muka agar materinya akan lebih mudah lagi untuk dipahami dari sebelumnya	Saran	Saran	1
45	Semoga mtkul ipb di offlinekan	Saran	Saran	1

46	Penjelasan materi kurang jelas	Kritik	Kritik	1
47	Agar lebih meningkatkan kebersihan dilingkungan kampus dan bebsa dari asap rokok	Saran	Saran	1
48	TU lambat bikin mahasiswa susah dapat dokumen tepat waktu	Kritik	Kritik	1
49	Harap meningkatkan lagi kualitas belajar daring	Saran	Saran	1
50	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	1
51	Sebaiknya media pembelajaran bervariasi agar tidak monoton Dan mahasiswa lebih semangat lagi Terima kasih	Saran	Saran	1
52	Sistem simak perlu ditingkatkan dikarenakan sistem sering mengalami eror	Kritik	Kritik	1
53	Tempat parkir mobil sering kali tidak cukup luas untuk menampung kendaraan mahasiswa	Kritik	Kritik	1
54	banyak banyak mendekati diri dan memahami mahasiswanya pak	Saran	Kritik	0
55	saran tolong kalau di chat setidaknya di balas atau konfirmasi sudah 2 smester mengajar tetap susah di hubungi	Saran	Saran	1
56	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	Saran	Saran	1

	tetap bisa mengabdikan di amal usaha ini			
57	Saran saya hanya 1 semoga bapak bisa kasih nilai sesuai kemampuan mahasiswa	Saran	Saran	1
58	Keterlambatan dosen menunjukkan kurangnya dedikasi terhadap pengajaran	Kritik	Kritik	1
59	Banding nilai susah banget diajukan	Kritik	Kritik	1
60	Pelayanan akademik sangat memadai dan inovasi terbaru selalu di munculkan	Saran	Kritik	0
	Ulasan	Label Sebenarnya	Prediksi	
2222	Banyak kaca jendela yang buram dan tidak pernah dibersihkan	Kritik	Kritik	1
2223	Harus lebih tepat waktu dan sesuai jadwal perkuliahan	Saran	Kritik	0
2224	Semoga lebih baik lagi kedepannya	Saran	Saran	1
2225	semoga kedepannya menjadi lebih baik	Saran	Saran	1
2226	Saran agar bapak dosen bisa lebih tegas terhadap mahasiswa tanpa menghilangkan kebiasaan atau metode belajar bapak yang unik	Saran	Saran	1
2227	Semoga lebih baik kedepannya	Saran	Saran	1
2228	Semoga pertemuan selanjutnya bisa offline supaya pembelajarannya lebih mudah untuk dipahami	Saran	Saran	1
2229	Dari segi kedisiplinan dosen dalam proses belajar mengajar sudah sesuai Rencana Pembelajaran Semester Untuk masalah sarana alat proses mengajar yang kurang	Kritik	Kritik	1

	memadai contohnya LCD Terima kasih			
2230	Mohon maaf sebelumnya saran saya dalam hal ini yaitu perlunya pembelajaran tatap muka sehingga mata kuliah dapat dipahami dengan baik Maaf bila ada salah kata dan Terima kasih	Saran	Saran	1
2231	Semoga kedepannya pembelajaran bisa dilakukan secara tatap muka agar materinya akan lebih mudah lagi untuk dipahami dari sebelumnya	Saran	Saran	1
2232	Perlu monev integrasi AIK sbg penciri Muhammadiyah	Saran	Saran	1
2233	Semoga kedepannya perkuliahan bisa terlaksana sesuai ketentuan yang berlaku	Saran	Saran	1
2234	Pelayanan dan sarpras semakin baik dan lebih ditingkatkan lagi	Saran	Saran	1
2235	Sistem pelayanan terintegritas tetap dijaga dan ditingkatkan terutama dalam pemanfaatan ICT disemua sektor pelayanan	Saran	Saran	1
2236	supaya kuliah of line sebaiknya dilakukan	Saran	Saran	1
2237	Materi kuliah digital hampir tidak tersedia di perpustakaan	Kritik	Kritik	1
2238	Banyak ruang kelas yang tidak dilengkapi dengan alat bantu pengajaran yang memadai	Kritik	Kritik	1
	Jumlah label yang benar			2038
	Accucay			0.91

Sourcecode

```
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/Skripsi/2020/Ahmad
Faisal RNN/saran.xlsx', sheet_name="Sheet1") # Replace
'path_to_your_excel_file.xlsx' with your actual file path
df = df[['ULASAN', 'LABEL']] # Selecting relevant
columns
df = df[pd.notnull(df['ULASAN'])] # Dropping rows with
null 'ULASAN' values
df.rename(columns={'ULASAN': 'ULASAN'}, inplace=True) #
Rename 'ULASAN' to 'ULASAN' for consistency
df.head()
df.shape
# 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)
```

```

# 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)
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))
# 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)
#train_tagged.values
train_tagged.values
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
print(d2v_model)
# 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)

# Inisialisasi 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])

from keras.models import Sequential
from keras.layers import SimpleRNN, Dense, Embedding,
Dropout
from keras.callbacks import ModelCheckpoint,
EarlyStopping
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)

# Inisialisasi 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 RNN
model.add(SimpleRNN(50, return_sequences=False))

```

```

# Menambahkan lapisan Dropout untuk mencegah overfitting
model.add(Dropout(0.5))

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

# 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)

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)
# Melatih model dengan data validasi
history = model.fit(X_train, Y_train, epochs=20,
                   batch_size=16, 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)
# Save the model
model.save('/content/drive/MyDrive/Skripsi/2020/Ahmad
Faisal RNN/RNN.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']

# 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()
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)
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))

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
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)

# Export ke Excel
test_results.to_excel('/content/drive/MyDrive/Skripsi/202
0/Rosalinda - LSTM/hasil_prediksi.xlsx', index=False)

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

```



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 : Ahmad Faisal

Nim : 105841112420

Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	10 %	10 %
2	Bab 2	25 %	25 %
3	Bab 3	8 %	10 %
4	Bab 4	7 %	10 %
5	Bab 5	0 %	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, 10 Agustus 2024
Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,



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

BAB I Ahmad Faisal -
105841112420

by Tahap Tutup

Submission date: 10-Aug-2024 08:23AM (UTC+0700)

Submission ID: 2429765057

File name: BAB_I_FAISAL_1.docx (18.2K)

Word count: 769

Character count: 5092

BAB I Ahmad Faisal - 105841112420

ORIGINALITY REPORT

10%

SIMILARITY INDEX

10%

INTERNET SOURCES

7%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

Rank	Source	Percentage
1	digilib.unila.ac.id Internet Source	3%
2	repo.undiksha.ac.id Internet Source	2%
3	scholar.unand.ac.id Internet Source	2%
4	ojs.unimal.ac.id Internet Source	2%
5	repository.pip-semarang.ac.id Internet Source	2%



Exclude quotes On
Exclude bibliography On

Exclude matches < 2%

BAB II Ahmad Faisal -
105841112420

by Tahap Tutup

Submission date: 10-Aug-2024 08:23AM (UTC+0700)

Submission ID: 2429765150

File name: BAB_II_FAISAL_1.docx (78.74K)

Word count: 1663

Character count: 10931

BAB II Ahmad Faisal - 105841112420

ORIGINALITY REPORT

25%

SIMILARITY INDEX

25%

INTERNET SOURCES

2%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1	digilibadmin.unismuh.ac.id Internet Source	5%
2	sistemasi.ftik.unisi.ac.id Internet Source	5%
3	blogelissss.blogspot.com Internet Source	3%
4	docplayer.info Internet Source	2%
5	issuu.com Internet Source	2%
6	www.scribd.com Internet Source	2%
7	bagawanabiyasa.wordpress.com Internet Source	1%
8	123dok.com Internet Source	1%
9	ejournal.undip.ac.id Internet Source	1%



BAB III Ahmad Faisal - 105841112420

by Tahap Tutup

Submission date: 09-Aug-2024 01:59PM (UTC+0700)

Submission ID: 2429425042

File name: BAB_III_FAISAL.docx (57.09K)

Word count: 947

Character count: 5978

BAB III Ahmad Faisal - 105841112420

ORIGINALITY REPORT

8%

SIMILARITY INDEX

8%

INTERNET SOURCES

0%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

dqlab.id

Internet Source

3%

2

ejournal.gunadarma.ac.id

Internet Source

2%

3

docplayer.info

Internet Source

2%

4

cyberarea.id

Internet Source

2%

Exclude quotes On

Exclude matches < 2%

Exclude bibliography On



BAB IV Ahmad Faisal -
105841112420

by Tahap Tutup

Submission date: 09-Aug-2024 02:00PM (UTC+0700)

Submission ID: 2429425371

File name: BAB_IV_FAISAL.docx (897.02K)

Word count: 6388

Character count: 44710

AB IV Ahmad Faisal - 105841112420

ORIGINALITY REPORT

7% SIMILARITY INDEX
7% INTERNET SOURCES
0% PUBLICATIONS
% STUDENT PAPERS

PRIMARY SOURCES

- | | | |
|----------|---|-----------|
| 1 | digilibadmin.unismuh.ac.id
Internet Source | 5% |
| 2 | towardsdatascience.com
Internet Source | 2% |



Exclude quotes On Exclude matches < 2%
Exclude bibliography On



BAB V Ahmad Faisal -
105841112420

by Tahap Tutup

Submission date: 09-Aug-2024 02:00PM (UTC+0700)

Submission ID: 2429425588

File name: BAB_V_FAISAL.docx (13.89K)

Word count: 119

Character count: 786

AB V Ahmad Faisal - 105841112420

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES



Exclude quotes On

Exclude bibliography On

Exclude matches 2%

