

**IMPLEMENTASI *GATED RECURRENT UNIT* (GRU) DALAM PREDIKSI
GANGGUAN *MENTAL HEALTH* PADA MAHASISWA FAKULTAS
ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

SKRIPSI

**Ayu Prasinta
201401109**



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

MEDAN

2024

**IMPLEMENTASI *GATED RECURRENT UNIT* (GRU) DALAM PREDIKSI
GANGGUAN *MENTAL HEALTH* PADA MAHASISWA FAKULTAS
ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Komputer

Ayu Prasinta

201401109



PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

MEDAN

2024

PERSETUJUAN

Judul : IMPLEMENTASI *GATED RECURRENT UNIT*
(GRU) DALAM PREDIKSI GANGGUAN
MENTAL HEALTH PADA MAHASISWA
FAKULTAS ILMU KOMPUTER DAN
TEKNOLOGI INFORMASI UNIVERSITAS
SUMATERA UTARA

Kategori : SKRIPSI

Nama : AYU PRASINTA

Nomor Induk Mahasiswa : 201401109

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI
INFORMASI UNIVERSITAS
SUMATERA UTARA

Diluluskan di
Medan, 11 Juni 2024

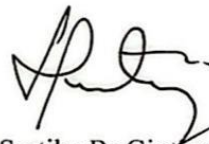
Komisi Pembimbing :

Dosen Pembimbing II



Handrizal S.Si., M.Comp.Sc
NIP. 197706132017061001

Dosen Pembimbing I



Dewi Sartika Br.Ginting, S.Kom., M.Kom
NIP. 199005042019032023

Diketahui / Disetujui Oleh
Ketua Program Studi S-1 Ilmu Komputer



Dr. Amalia S.T., M.T
NIP. 19781221 201404 2 001

PERNYATAAN
IMPLEMENTASI *GATED RECURRENT UNIT* (GRU) DALAM PREDIKSI
GANGGUAN *MENTAL HEALTH* PADA MAHASISWA FAKULTAS
ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

SKRIPSI

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

Medan, 11 Juni 2024



Ayu Prasinta

201401109

UCAPAN TERIMA KASIH

Ungkapan puji dan syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya kepada penulis sehingga penulisan skripsi ini dapat diselesaikan dengan sebaik baiknya, serta sebagai persyaratan untuk memperoleh gelar Sarjana Komputer dari Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Penulis ingin mengekspresikan rasa hormat dan terima kasih yang tulus kepada semua pihak yang telah memberikan bantuan dan dukungan dalam proses penulisan dan penyelesaian skripsi ini. Terima kasih untuk mereka yang telah memberikan doa, bimbingan, kerjasama, dukungan, serta kata-kata yang memberikan ketenangan dalam perjalanan penulisan skripsi ini. Penulis mengapresiasi semua bantuan dan kontribusi yang telah diberikan dari berbagai pihak. Penulis mengucapkan terima kasih kepada:

1. Prof. Dr. Muryanto Amin S.Sos., M.Si. sebagai Rektor Universitas Sumatera Utara
2. Dr. Maya Silvi Lydia B.Sc., M.Sc. sebagai Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Dr. Amalia S.T., M.T. sebagai Ketua Program Studi S- 1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Dewi Sartika Br Ginting S.Kom., M.Kom selaku dosen pembimbing I yang telah banyak memberikan ilmu dan masukan terhadap skripsi yang dikerjakan oleh penulis.
5. Bapak Handrizal S.Si., M.Comp.Sc sebagai dosen pembimbing II yang telah memberikan banyak masukan, kritik dan saran terhadap penulis.
6. Kedua orangtua penulis, Ayahanda tercinta, Alm Habib beliau memang tidak sempat menemani penulis dalam perjalanan selama menempuh pendidikan tapi terima kasih sudah menjadi motivator bagi penulis dan terima kasih telah mengantarkan penulis berada ditempat ini walaupun pada akhirnya penulis harus berjuang sendiri tanpa ditemani lagi Dan kepada Ibunda tercinta, Sulasmi terima kasih karna senantiasa

menyayangi, mendoakan dan mendukung penulis dalam tiap aspek kehidupan yang penulis lalui.

7. Abang kandung penulis Salman Alfarizi, S.M dan adik kandung penulis Rio Sanjaya yang menjadi penyemangat penulis.
8. M. Zuhry Palmindo selaku partner disegala situasi penulis yang senantiasa menemani dan menyemangati penulis.
9. Teman-Teman seperjuangan : Syabrina Ramadhani, Tridinda Permatasari, Rani Syahfitri, Ainun Mardiah, Farrel Dwi Prayogo, Haryanda Fidi, dan Delpiero.
10. Seluruh teman-teman Kom C dan stambuk 2020 Program Studi S-1 Ilmu Komputer yang lagi sama-sama berjuang.
11. Keluarga IMILKOM USU, khususnya departemen KOMINFO periode 2022/2023 yang telah memberikan banyak ilmu dan pengalaman pada penulis.
12. Dan seluruh pihak yang terlibat dalam penyusunan skripsi penulis yang tidak bisa disebutkan satu persatu.

Semoga hasil penelitian ini dapat memberikan kontribusi dan manfaat yang nyata bagi perkembangan ilmu pengetahuan dan teknologi. Penulis menyadari bahwa penelitian ini memiliki keterbatasan dan ruang untuk pengembangan lebih lanjut. Oleh karena itu, Penulis berharap penelitian ini dapat menjadi dasar untuk penelitian lanjutan dan menginspirasi penelitian lainnya.

Akhir kata, Terima kasih kepada Ayu Prasinta yang sudah berjuang dan tidak menyerah dan memilih menyelesaikan skripsi ini sampai akhir walaupun banyak sekali hambatan yang datang disaat proses penyelesaian skripsi ini. semoga skripsi ini bermanfaat bagi pembaca dan masyarakat luas. Penulis berharap agar hasil penelitian ini dapat memberikan sumbangsih positif dalam memahami Kesehatan mental mahasiswa di seluruh Indonesia. Terima kasih.

Medan, 11 Juni 2024



Penulis

ABSTRAK

Kesehatan mental merupakan aspek krusial dari kesejahteraan individu yang mencakup keseimbangan emosional, psikologis, dan sosial. Dalam beberapa tahun terakhir, perhatian terhadap kesehatan mental di kalangan mahasiswa meningkat secara signifikan. Mahasiswa menghadapi berbagai tekanan, seperti tuntutan akademik, masalah keuangan, dan adaptasi sosial, yang dapat berdampak negatif pada kesehatan mental mereka. Penelitian ini bertujuan untuk melakukan prediksi gangguan *mental health* pada mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara menggunakan algoritma GRU (*Gated Recurrent Unit*). Data yang digunakan dalam penelitian ini dikumpulkan melalui *Google Form* sebanyak 645 data. Metode GRU dipilih karena kemampuannya dalam melakukan prediksi. Jenis penyakit yang akan diprediksi dalam penelitian ini adalah *anxiety* dan depresi. Proses penelitian ini melibatkan *training* model GRU dan memasukkannya ke dalam Flask. *Epoch* yang digunakan adalah 50, *learning rate* sebesar 0,001, dan *batch size* sebesar 8. *Dataset* yang terkumpul melalui *Google Form* kemudian melalui tahap *preprocessing*, termasuk konversi kolom ke tipe numerik dan menghapus nilai kosong, sebelum dilakukan proses *training* dan evaluasi. Hasil dari penelitian ini menunjukkan bahwa metode analisis prediksi gangguan mental menggunakan algoritma GRU memberikan akurasi yang signifikan mencapai akurasi sebesar 96%. Hasil ini menunjukkan kemampuan GRU dalam memprediksi gangguan *mental health* akurasi yang tinggi. Penelitian ini memberikan kontribusi dalam pemahaman untuk melakukan prediksi terhadap gangguan *mental health* yang sering terjadi. Namun, penelitian ini memiliki keterbatasan dan masih terdapat ruang untuk pengembangan lebih lanjut, seperti peningkatan jumlah data atau penambahan fitur *preprocessing*.

Kata Kunci: Prediksi, *Mental Health*, *Google Form*, *Gated Recurrent Unit*, *Website*, *Deep Learning*, Flask

*IMPLEMENTATION OF GATED RECURRENT UNIT (GRU) IN PREDICTING
MENTAL HEALTH DISORDERS IN STUDENTS OF THE FACULTY OF
COMPUTER SCIENCE AND INFORMATION TECHNOLOGY,
UNIVERSITY OF NORTH SUMATERA*

ABSTRACT

Mental health is a crucial aspect of individual well-being that includes emotional, psychological and social balance. In recent years, attention to mental health among university students has increased significantly. Students face various pressures, such as academic demands, financial problems, and social adaptation, which can have a negative impact on their mental health. This study aims to predict mental health disorders in students of the Faculty of Computer Science and Information Technology, University of North Sumatra using the GRU (Gated Recurrent Unit) algorithm. The data used in this study were collected through Google Form as much as 645 data. The GRU method was chosen because of its ability to make predictions. The types of diseases that will be predicted in this study are anxiety and depression. This research process involves training the GRU model and inputting it into Flask. The epoch used is 50, the learning rate is 0.001, and the batch size is 8. The dataset collected through Google Form then goes through a preprocessing stage, including column conversion to numeric type and removing empty values, before the training and evaluation process. The results of this study show that the mental disorder prediction analysis method using the GRU algorithm provides significant accuracy, reaching an accuracy of 96%. This result shows the ability of GRU in predicting mental health disorders.

Keywords: *Prediction, Mental Health, Google Form, Gated Recurrent Unit, Website, Deep Learning, Flask*

DAFTAR ISI

PERSETUJUAN	iii
PERNYATAAN.....	iv
UCAPAN TERIMA KASIH.....	v
ABSTRAK	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian.....	5
1.7 Sistematika Penyusunan.....	7
BAB 2 LANDASAN TEORI.....	9
2.1 Prediksi	9
2.2 Kesehatan Mental.....	10
2.3 <i>Machine Learning</i>	11
2.4 <i>Deep Learning</i>	13
2.5 ANN (<i>Artificial Neural Network</i>).....	14
2.6 RNN (<i>Recurrent Neural Network</i>).....	15
2.7 HTML (<i>Hypertext Markup Language</i>)	17
2.8 CSS (<i>Cascading Style Sheet</i>)	18
2.9 GRU (<i>Gated Recurrent Unit</i>).....	18
2.10 <i>Website</i>	23
2.11 <i>Flask</i>	24
2.12 Penelitian Relevan	25

BAB 3 ANALISIS DAN PERANCANGAN	28
3.1 Arsitektur Umum Sistem	28
3.2 <i>Flowchart</i>	29
3.3 Pengumpulan <i>Dataset</i>	30
3.4 Labelisasi <i>Dataset</i>	30
3.5 <i>Pre-processing Dataset</i>	32
3.5.1 Konversi Kolom Ke Tipe Numerik	32
3.5.2 Menghapus Nilai Kosong	33
3.6 Split Dataset.....	34
3.7 Implementasi GRU	35
3.7.1 <i>Hyperparameter Model GRU</i>	35
3.7.2 Algoritma Optimasi.....	36
3.7.3 Mekanisme <i>Update Gate</i>	37
3.7.4 Mekanisme <i>Reset Gate</i>	37
3.7.5 Perhitungan Calon Status	37
3.7.6 Pembaruan Status	37
3.7.8 Tahapan Algoritma GRU	38
3.8 <i>Deploy Model Ke Server</i>	39
3.8.1 Persiapan Model dan Lingkungan Flask	39
3.8.2 Membuat API dengan Flask	39
3.8.3 Menjalankan API Flask	39
3.9 Evaluasi.....	40
BAB 4 IMPLEMENTASI DAN PENGUJIAN	41
4.1 Implementasi Sistem.....	41
4.1.1 Spesifikasi Perangkat Keras	41
4.1.2 Spesifikasi Perangkat Lunak	41
4.2 Pengumpulan <i>Dataset</i>	42
4.3 Labelisasi <i>Dataset</i>	43
4.4 <i>Preprocessing Dataset</i>	43
4.4.1 Konversi Kolom Ke Tipe Numerik	43
4.4.2 Menghapus Nilai Kosong.....	44
4.5 <i>Split / Pembagian Dataset</i>	44
4.6 Implementasi Algoritma GRU	45

4.7	Evaluasi Algoritma GRU	48
4.7.1	Perhitungan Manual Akurasi	48
4.8	Pengujian Sistem oleh Pakar	49
BAB 5 KESIMPULAN DAN SARAN		50
5.1	Kesimpulan	51
5.2	Saran	52
DAFTAR PUSTAKA		52
LAMPIRAN		A-1

DAFTAR TABEL

Tabel 3. 3 Contoh <i>Dataset</i> Hasil Labelisasi	25
--	----

DAFTAR GAMBAR

Gambar 2.1 Hubungan antara Deep Learning, <i>Machine Learning</i> dan Kecerdasan Buatan (Chollet, 2018).....	12
Gambar 2. 2 Arsitektur kerja dari metode RNN.....	16
Gambar 2. 3 Arsitektur Sistem GRU.....	19
Gambar 2. 4 <i>Cell Memory</i> dan Gates GRU	20
Gambar 2. 5 Ilustrasi <i>Cell Memory</i> Pada GRU	20
Gambar 3. 1 Arsitektur Umum Sistem	27
Gambar 3. 2 <i>Flowchart</i> Sistem.....	28
Gambar 4. 1 <i>Google Form</i> Sebagai Sarana Pengumpulan Data.....	37
Gambar 4. 2 <i>Dataset</i> Jawaban Responden	37
Gambar 4. 3 <i>Dataset</i> yang sudah di labelisasi.....	38
Gambar 4. 4 Proses <i>Splitting Dataset</i>	38
Gambar 4. 4 .1 Konversi Kolom Ke Tipe Numerik	39
Gambar 4. 4. 2 Menghapus Nilai Kosong	39
Gambar 4. 5 Akurasi yang Diperoleh dengan 50 <i>Epoch</i>	41
Gambar 4. 6 Proses <i>Training</i> Pada Percobaan Pertama	41
Gambar 4. 7 Tampilan Halaman <i>Input</i> Gejala.....	41
Gambar 4. 8 Tampilan Halaman Hasil	42

DAFTAR LAMPIRAN

Lampiran 1. <i>Listing Program</i>	A-1
Lampiran 2. Daftar Riwayat Hidup.....	B-1

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kesehatan mental yaitu aspek krusial dari kesejahteraan individu yang mencakup keseimbangan emosional, psikologis, dan sosial. Dalam beberapa tahun terakhir, perhatian terhadap kesehatan mental di kalangan mahasiswa meningkat secara signifikan. Mahasiswa menghadapi berbagai tekanan, seperti tuntutan akademik, masalah keuangan, dan adaptasi sosial, yang dapat berdampak negatif pada kesehatan mental mereka. Berbagai penelitian menunjukkan bahwa tingkat kecemasan dan depresi di kalangan mahasiswa cenderung lebih tinggi dibandingkan dengan populasi umum, menjadikan ini isu yang mendesak untuk ditangani. Kesehatan mental pada mahasiswa dapat dipengaruhi oleh beberapa faktor, diantaranya yaitu faktor pendidikan, keluarga, pertemanan, sosial, dan berbagai faktor lainnya. (Fitria Amalia Rochimah, 2020).

Mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi di Universitas Sumatera Utara tidak terkecuali dari fenomena ini. Fakultas ini memiliki kurikulum yang intensif dan berorientasi teknologi, yang menuntut mahasiswa untuk selalu mengikuti perkembangan terbaru di bidang teknologi informasi. Beban akademik yang tinggi ini, ditambah dengan kebutuhan untuk terus berinovasi, dapat menjadi sumber stres yang signifikan bagi mahasiswa. Selain itu, banyak mahasiswa yang merantau dari luar daerah harus menyesuaikan diri dengan lingkungan baru dan jauh dari dukungan keluarga, yang juga dapat meningkatkan risiko gangguan kesehatan mental.

Gangguan kesehatan mental seperti kecemasan dan depresi sering kali tidak terdeteksi atau tidak ditangani dengan baik di kalangan mahasiswa. Banyak mahasiswa yang enggan mencari bantuan profesional karena stigma sosial yang melekat pada gangguan mental. Mereka mungkin merasa malu atau takut dianggap lemah oleh teman sebaya dan masyarakat. Selain itu, kurangnya informasi tentang layanan kesehatan mental yang tersedia di kampus juga menjadi hambatan bagi mahasiswa untuk mendapatkan bantuan yang mereka butuhkan.

Deteksi dini dan intervensi yang tepat adalah kunci untuk mengatasi masalah kesehatan mental ini. Namun, deteksi gangguan mental secara manual sering kali memerlukan waktu dan sumber daya yang tidak sedikit. Di sinilah teknologi dapat berperan penting. Dengan kemajuan dalam bidang *machine learning* dan analisis data, kita dapat mengembangkan sistem yang mampu mendeteksi tanda-tanda awal gangguan mental secara otomatis dan akurat. Teknologi ini dapat digunakan untuk memproses data survei yang dikumpulkan dari mahasiswa dan mengidentifikasi pola-pola yang mengindikasikan risiko gangguan mental.

Gated Recurrent Unit (GRU) merupakan salah satu teknik *machine learning* yang potensial untuk digunakan didalam deteksi gangguan kesehatan mental. GRU adalah jenis jaringan saraf berulang yang dirancang untuk menangani data sekuensial. Dibandingkan dengan *Long Short-Term Memory* (LSTM), GRU memiliki arsitektur yang lebih sederhana namun tetap efektif dalam menangkap dependensi jangka panjang dalam data. Hal ini menjadikan GRU sangat cocok untuk memproses data survei dan riwayat kesehatan mental yang bersifat sekuensial.

Implementasi GRU dalam sistem prediksi kesehatan mental mahasiswa dapat membantu mengidentifikasi mahasiswa yang berisiko mengalami gangguan mental. Dengan menggunakan data survei yang dikumpulkan melalui *Google Form*, model GRU dapat dilatih untuk mengenali tanda-tanda awal kecemasan dan depresi. Sistem ini dapat memberikan rekomendasi intervensi dini sebelum masalah kesehatan mental berkembang menjadi lebih serius. Ini tidak hanya bermanfaat bagi mahasiswa, tetapi juga bagi pihak universitas dalam merancang program dukungan yang lebih efektif.

Pemanfaatan teknologi dalam mendeteksi gangguan kesehatan mental juga dapat mengurangi stigma yang sering dikaitkan dengan masalah ini. Dengan adanya sistem otomatis yang dapat diakses secara anonim, mahasiswa mungkin merasa lebih nyaman untuk mengevaluasi kondisi kesehatan mental mereka tanpa takut dihakimi. Selain itu, sistem ini dapat menyediakan informasi dan sumber daya yang relevan bagi mahasiswa yang membutuhkan bantuan, sehingga mereka dapat mengambil langkah proaktif dalam menjaga kesehatan mental mereka.

Universitas Sumatera Utara, khususnya Fakultas Ilmu Komputer dan Teknologi Informasi, memiliki tanggung jawab untuk menyediakan lingkungan yang mendukung kesehatan mental mahasiswanya. Dengan adanya sistem deteksi dini berbasis GRU, universitas dapat lebih proaktif dalam menangani masalah kesehatan mental. Hal ini juga sejalan dengan upaya global untuk meningkatkan kesadaran dan penanganan kesehatan mental di kalangan remaja dan dewasa muda.

Oleh karena itu, penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan model GRU dalam sistem prediksi gangguan kesehatan mental mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Dengan memanfaatkan data survei dan teknologi machine learning, diharapkan sistem ini dapat memberikan solusi yang efektif dan efisien untuk mendeteksi gangguan mental secara dini, serta membantu mahasiswa mendapatkan bantuan yang mereka butuhkan tepat waktu. Penelitian ini juga diharapkan dapat mendorong lingkungan akademik yang lebih sehat dan suportif, di mana kesehatan mental menjadi prioritas yang diakui dan dihargai..

1.2 Rumusan Masalah

Mahasiswa termasuk dalam kelompok yang rentan terhadap masalah kesehatan mental, terutama dalam hal menjaga keseimbangan emosi. Hal ini karena mahasiswa berada dalam rentang usia 17 hingga 25 tahun, yaitu periode transisi dari masa remaja ke dewasa awal, yang memiliki risiko tinggi mengalami gangguan emosional. Gangguan kesehatan mental yang tidak ditangani dapat berujung pada risiko bunuh diri. Di Indonesia, kasus bunuh diri di kalangan mahasiswa cukup banyak terjadi. Mereka biasanya merasa tertekan oleh berbagai faktor seperti tekanan akademis, masalah keluarga, sosial, dan lainnya. Oleh karena itu, diperlukan sistem untuk memprediksi gangguan kesehatan mental Mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi USU dengan menerapkan algoritma Gated Recurrent Unit.

1.3 Batasan Masalah

Dalam penelitian ini, peneliti memiliki batasan terhadap penyelesaian masalah. Adapun batasan masalah dalam pelaksanaan penelitian ini adalah:

1. Program dibuat untuk memprediksi gangguan Kesehatan mental

mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

2. Penyakit kesehatan mental mahasiswa yang dituju hanya berfokus pada *Anxiety* (Kecemasan) dan Depresi.
3. Ulasan Pembuatan sistem prediksi mengimplementasikan menggunakan algoritma *Gated Recurrent Unit* (GRU).
4. Data yang digunakan menggunakan data yang bersumber dari lapangan langsung yaitu mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi USU dengan melakukan distribusi kuisioner.
5. Hasil dari proses pengujian akan di validasi oleh seorang pakar psikologi yaitu Dr. Meutia Naully S.Psi., M.si., Psikolog.
6. Program akan diimplementasikan menggunakan *Google Colab* menggunakan Bahasa *python* untuk *training data* dan menghasilkan model dari algoritma GRU.
7. Implementasi model kedalam Sistem dan Sistem yang akan dibuat merupakan sistem prediksi berbasis *web*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini untuk melakukan prediksi gangguan kesehatan mental mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi USU menggunakan algoritma GRU (*Gated Recurrent Unit*) dengan model yang tetap agar mampu memberikan hasil prediksi yang akurat dan optimal berdasarkan validasi Ahli Psikologi.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah:

1. Mengetahui cara mengimplementasikan algoritma *Gated Recurrent Unit* dalam memprediksi gangguan mental mahasiswa di Ilmu Komputer USU.
2. Dapat memberikan kontribusi dalam pemahaman dan pengetahuan

gangguan kesehatan mental pada mahasiswa melalui implementasi *Gated Recurrent Unit* (GRU).

3. Melalui sistem yang dibuat dapat membantu pengembangan strategi pencegahan dan *intervensi* dini bagi mahasiswa dalam menjaga kesejahteraan mental mereka baik di lingkungan akademis maupun non akademis.
4. Menjadi referensi pada penelitian berikutnya khususnya di bidang Kesehatan mental mahasiswa yang menggunakan algoritma *Gated Recurrent Unit*.

1.6 Metodologi Penelitian

Metodologi penelitian yang digunakan adalah sebagai berikut:

1. Studi Pustaka

Dalam metode ini, peneliti mengumpulkan dan menganalisis bahan-bahan pustaka yang relevan, seperti jurnal, buku, artikel, atau makalah, untuk mendapatkan pemahaman yang baik tentang topik penelitian yang ingin diteliti. Tujuan utama dari studi pustaka adalah untuk menyusun kerangka pemikiran yang kuat dan mendasari penelitian yang akan dilakukan, serta memperoleh pemahaman yang komprehensif tentang perkembangan penelitian terkini, temuan-temuan sebelumnya, dan pendekatan yang digunakan oleh peneliti lain dalam bidang yang sama.

2. Identifikasi Masalah

Dalam metode ini, peneliti melakukan analisis menyeluruh terhadap lingkungan atau domain tertentu untuk mengidentifikasi masalah yang ada, celah pengetahuan yang perlu diisi, atau situasi yang membutuhkan pemecahan masalah. Tujuan utama dari metode ini adalah untuk menentukan tujuan penelitian yang jelas dan relevan, serta merumuskan pertanyaan penelitian yang spesifik dan terarah.

3. Pengumpulan Data

Pada tahap ini, dilakukan proses pengumpulan dataset melalui persebaran kuisioner kepada mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi USU kemudian data nya di bagi menjadi 2 untuk

training data dan testing data yang dimana masing-masing pembagian dibagi 80% dan 20%.

4. Perancangan dan Implementasi Sistem

Metode penelitian Perancangan dan Implementasi Sistem berfokus pada pengembangan solusi teknologi yang dapat diimplementasikan untuk memecahkan masalah yang relevan dengan topik penelitian. Hasil penelitian ini berupa sebuah program yang akan melakukan *training* terhadap ribuan data dan menghasilkan model yang cerdas.

5. Pengujian Sistem

Metode penelitian Pengujian Sistem adalah pendekatan yang digunakan untuk menguji dan mengevaluasi sistem yang telah dirancang dan diimplementasikan dalam suatu penelitian. Tujuan dari metode ini adalah untuk memverifikasi dan memvalidasi fungsionalitas, kualitas, dan performa sistem yang dikembangkan. Proses pengujian sistem melibatkan serangkaian langkah yang sistematis dan terencana. Pertama, peneliti akan merancang rencana pengujian yang mencakup tujuan pengujian, skenario pengujian, data uji, dan kriteria keberhasilan. Selanjutnya, peneliti akan melakukan pengujian fungsionalitas sistem untuk memastikan bahwa sistem dapat beroperasi sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan.

6. Analisis Hasil

Dalam sub bab analisis hasil, data yang telah dikumpulkan dari penelitian akan diurai secara sistematis untuk mendapatkan pemahaman yang mendalam tentang masalah yang diteliti. Analisis dimulai dengan penyajian data secara deskriptif, baik melalui tabel, grafik, atau statistik ringkasan, yang membantu menggambarkan karakteristik dan distribusi variabel yang diteliti. Selanjutnya, analisis dilakukan lebih mendalam menggunakan metode statistik atau teknik analisis lainnya yang sesuai dengan tujuan penelitian. Misalnya, penelitian kuantitatif dapat melibatkan uji hipotesis atau analisis regresi, sementara penelitian kualitatif dapat mengadopsi analisis isi atau analisis tematik.

7. Dokumentasi

Dokumentasi adalah pendekatan yang menggunakan analisis dan penelaahan dokumen atau sumber data yang telah ada sebagai sumber informasi utama dalam penelitian. Metode ini melibatkan pengumpulan, evaluasi, dan interpretasi dokumen-dokumen yang relevan dengan topik penelitian untuk memperoleh pemahaman yang lebih baik tentang masalah atau kejadian yang diteliti.

1.7 Sistematika Penyusunan

Adapun tahap-tahap dalam penelitian ini terdiri atas:

BAB 1: Pendahuluan

Pendahuluan adalah bagian awal dari suatu laporan, makalah, atau penelitian yang bertujuan untuk memperkenalkan pembaca dengan topik yang akan dibahas. Dalam pendahuluan, pembaca diberikan gambaran umum tentang latar belakang topik, termasuk konteks dan perkembangan terbaru dalam bidang tersebut. Selain itu, pendahuluan juga mengidentifikasi masalah atau pertanyaan penelitian yang akan dijawab, menyoroti relevansi dan pentingnya topik tersebut untuk penelitian lebih lanjut.

Bagian pendahuluan juga mencakup tujuan penelitian, di mana penulis menjelaskan secara jelas apa yang ingin dicapai melalui penelitian tersebut. Tujuan penelitian membantu menetapkan fokus dan arah penelitian, serta memberikan pemahaman kepada pembaca tentang manfaat yang diharapkan dari penelitian tersebut. Dengan menyajikan pendahuluan yang baik, pembaca akan memiliki pemahaman yang kuat tentang topik yang akan dibahas dan tujuan dari penelitian atau pembahasan yang dilakukan.

BAB 2: Landasan Teori

Sub bab landasan teori dari penelitian tersebut akan menjelaskan konsep-konsep utama terkait prediksi gangguan kesehatan mental pada mahasiswa dan penggunaan *Gated Recurrent Unit* (GRU) dalam konteks ini. Ini termasuk pemahaman mendalam tentang gangguan kesehatan mental yang umum dialami oleh mahasiswa, seperti stres, kecemasan, dan depresi, serta faktor-faktor yang memengaruhinya di lingkungan pendidikan tinggi.

Selanjutnya, sub bab ini akan membahas secara rinci tentang *Gated Recurrent Unit* (GRU) sebagai algoritma pembelajaran mesin yang efektif untuk memodelkan data deret waktu, termasuk arsitektur, prinsip kerja, dan keunggulan dalam memprediksi pola kompleks dalam data mental health. Penjelasan ini akan memberikan landasan teoritis yang kokoh bagi penelitian dalam mengimplementasikan GRU dalam prediksi gangguan kesehatan mental pada mahasiswa.

BAB 3: Analisis dan Perancangan

Bab ini mencakup evaluasi prediksi *mental health* di Fasilkom-TI Universitas Sumatera Utara menggunakan model GRU (*Gated Recurrent Unit*) dengan tujuan mengategorikan menjadi tiga kategori yaitu anxiety, depresi dan belum terdiagnosis.

BAB 4: Implementasi dan Pengujian

Bab ini adalah bagian penting dalam sebuah penelitian yang fokus pada penerapan dan evaluasi praktis dari sistem atau solusi yang telah dirancang. Dalam bab ini, peneliti menjelaskan langkah-langkah yang diambil untuk menerapkan solusi yang telah dirancang ke dalam lingkungan nyata atau simulasi.

BAB 5: Kesimpulan dan Saran

Bab Kesimpulan dan Saran merangkum temuan penelitian, menjawab pertanyaan penelitian, dan mengemukakan kesimpulan yang didapat dari hasil penelitian.

BAB 2

LANDASAN TEORI

2.1 Prediksi

Prediksi adalah proses memperkirakan hasil atau kejadian di masa depan berdasarkan data dan informasi yang tersedia saat ini. Dalam konteks teknologi dan analisis data, prediksi sering dilakukan menggunakan algoritma dan model statistik atau pembelajaran mesin, yang mampu mengidentifikasi pola dari data historis untuk menghasilkan estimasi yang akurat. Prediksi digunakan dalam berbagai bidang seperti cuaca, ekonomi, kesehatan, dan teknologi untuk membantu dalam pengambilan keputusan dan perencanaan. Misalnya, dalam bidang kesehatan mental, prediksi dapat membantu mengidentifikasi individu yang berisiko mengalami gangguan, sehingga intervensi dapat dilakukan lebih awal untuk mencegah dampak yang lebih serius. (Manalu, 2016).

Prediksi merupakan hasil dari kegiatan perkiraan, ramalan, atau proyeksi suatu kejadian untuk menilai nilainya di masa yang akan datang dengan menggunakan data yang terkait dengan masa lalu. Karena sifat prediktifnya, hasil ramalan tidak dapat mencapai akurasi 100%. Dalam prediksi yang bersifat obyektif, semakin banyak data masa lalu yang digunakan, semakin tinggi kemungkinan untuk mendapatkan hasil prediksi yang konsisten dan lebih baik. Kesimpulan dari konsep prediksi adalah bahwa perhitungan dilakukan untuk menentukan kemiripan kejadian di masa depan dengan referensi data historis yang relevan (Pratama, 2022).

Ada berbagai metode yang digunakan dalam prediksi, termasuk statistik, machine learning, dan deep learning. Metode statistik seperti regresi linier digunakan untuk memodelkan hubungan antara variabel independen dan dependen dalam data. Metode machine learning, seperti decision trees, random forests, dan support vector machines, digunakan untuk mengidentifikasi pola kompleks dalam data. Sedangkan metode deep learning, seperti neural networks, digunakan untuk memodelkan pola yang sangat kompleks dan non-linear dalam data.

Prediksi memiliki banyak keunggulan, termasuk membantu organisasi dan individu untuk merencanakan dan mengantisipasi perubahan di masa depan, memperbaiki kinerja dan efisiensi operasional, mengurangi risiko dan

ketidakpastian, serta memperoleh wawasan yang lebih dalam tentang tren dan pola dalam data.

2.2 Kesehatan Mental

Kesehatan mental adalah kondisi kesejahteraan psikologis seseorang yang melibatkan berbagai aspek kehidupan emosional, sosial, dan psikologis. Kesehatan mental mencakup kemampuan seseorang untuk mengelola stres, berinteraksi secara positif dengan orang lain, membuat keputusan, menangani emosi, dan mengatasi tantangan kehidupan sehari-hari.

Beberapa gangguan kesehatan mental yang umum mencakup depresi, gangguan kecemasan, gangguan bipolar, skizofrenia, dan gejala stres pasca-trauma. Masing-masing penyakit ini memiliki gejala dan tingkat keparahan yang berbeda-beda. Misalnya, depresi sering ditandai dengan perasaan sedih yang berkelanjutan dan hilangnya minat pada aktivitas yang biasa dinikmati, sementara gangguan kecemasan bisa menyebabkan perasaan cemas yang berlebihan dan ketakutan yang tidak masuk akal.

Kesehatan mental yang buruk dapat berdampak signifikan pada kehidupan sehari-hari seseorang. Ini dapat memengaruhi kemampuan mereka untuk bekerja, belajar, dan menjaga hubungan sosial yang sehat. Masalah kesehatan mental juga dapat mempengaruhi kebiasaan tidur, nafsu makan, dan tingkat energi. Jika tidak ditangani, gangguan kesehatan mental dapat menyebabkan penurunan kualitas hidup dan bahkan meningkatkan risiko bunuh diri.

Mengatasi masalah kesehatan mental memerlukan pendekatan yang holistik, termasuk terapi, obat-obatan, dan dukungan sosial. Terapi, seperti psikoterapi atau konseling, dapat membantu individu memahami dan mengelola perasaan mereka. Obat-obatan mungkin diperlukan untuk mengatasi ketidakseimbangan kimia di otak. Selain itu, dukungan dari keluarga, teman, dan komunitas sangat penting dalam proses pemulihan. Pendidikan dan kesadaran tentang kesehatan mental juga penting untuk mengurangi stigma dan mendorong individu untuk mencari bantuan ketika diperlukan.

2.3 *Machine Learning*

Machine learning merupakan bagian dari bidang kecerdasan buatan yang memungkinkan komputer untuk memperoleh pengetahuan dari data tanpa perlu program yang eksplisit. Dalam machine learning, digunakan algoritma dan model statistik untuk menganalisis serta menemukan pola dari data, yang selanjutnya dimanfaatkan untuk melakukan prediksi atau pengambilan keputusan yang cerdas. Terdapat beberapa jenis utama dari machine learning, termasuk supervised learning di mana model dipelajari dari data yang berlabel, unsupervised learning di mana model dipelajari dari data tanpa label, dan reinforcement learning di mana model belajar melalui interaksi dengan lingkungan. Machine learning sering digunakan pada macam-macam aplikasi, mulai dari identifikasi wajah dan suara, pemrosesan bahasa baku, hingga analisis data dan prediksi di berbagai industri seperti finansial, kesehatan, dan teknologi. Secara rinci, *Machine Learning* melibatkan beberapa konsep dan komponen utama, termasuk:

1. *Data Training*: Proses di mana model *Machine Learning* dilatih menggunakan data yang telah diberi label. *Data training* ini berfungsi sebagai "pelajaran" bagi model untuk mengenali pola dan menggeneralisasikan pengetahuan dari data tersebut.
2. *Algoritma*: Algoritma *Machine Learning* adalah metode matematis dan statistik yang digunakan untuk menggali informasi dari data. Ini termasuk metode seperti regresi, pohon keputusan, naive bayes, support vector machines, dan neural networks.
3. *Fitur dan Vektor*: *Data input* pada *Machine Learning* direpresentasikan sebagai fitur atau atribut numerik yang relevan. Fitur ini membantu model dalam mempelajari pola dari data. *Data input* ini sering diwakili dalam bentuk vektor.
4. *Proses Pelatihan (Training Process)*: Proses di mana model *Machine Learning* disesuaikan dengan data *training*. Ini melibatkan memperbarui parameter model untuk mengoptimalkan prediksi model sesuai dengan data *training*.
5. *Evaluasi dan Validasi*: Setelah model dilatih, model tersebut dievaluasi dan divalidasi menggunakan data yang terpisah, yang disebut sebagai

data validasi atau data pengujian. Evaluasi ini membantu untuk mengukur sejauh mana model dapat melakukan prediksi yang akurat dan mampu menggeneralisasi pengetahuan dari data yang belum dilihat sebelumnya.

6. **Prediksi dan Inferensi:** Setelah model dilatih dan divalidasi, model tersebut dapat digunakan untuk melakukan prediksi atau inferensi pada data baru yang sebelumnya tidak pernah dilihat.

Machine learning memiliki beberapa jenis yang berbeda, tergantung pada cara model belajar dari data dan jenis masalah yang dihadapi. Pertama, *Supervised Learning* adalah jenis di mana model dilatih menggunakan data yang berpasangan dengan label. Model belajar untuk membuat prediksi atau membuat keputusan berdasarkan contoh-contoh yang diberikan. Kedua, *Unsupervised Learning* adalah ketika model mengidentifikasi pola atau struktur dalam data tanpa label. Ini dapat digunakan untuk pengelompokan data (*clustering*) atau reduksi dimensi. Ketiga, *Reinforcement Learning* adalah jenis di mana agen belajar untuk membuat keputusan melalui interaksi dengan lingkungan, menerima umpan balik positif atau negatif dari tindakan-tindakan yang diambilnya. Keempat, *Semi-supervised Learning* adalah gabungan dari *supervised* dan *unsupervised learning*, di mana model menggunakan sebagian data yang berlabel dan sebagian data yang tidak berlabel untuk melakukan prediksi. Terakhir, ada juga *Transfer Learning*, yang melibatkan penggunaan pengetahuan yang diperoleh dari suatu tugas untuk membantu kinerja pada tugas terkait yang berbeda. Setiap jenis *machine learning* memiliki kegunaan dan aplikasi yang unik tergantung pada jenis data dan masalah yang dihadapi.

Penerapan *Machine Learning* sangat besar dan dapat ditemukan di berbagai bidang, termasuk pengenalan suara dan gambar, analisis data, rekomendasi produk, deteksi penipuan, pengenalan pola, pemrosesan bahasa alami, dan banyak lagi. Dengan kemajuan teknologi dan ketersediaan data yang melimpah, *Machine Learning* terus berkembang dan menjadi salah satu bidang yang paling penting dalam ilmu komputer.

2.4 *Deep Learning*

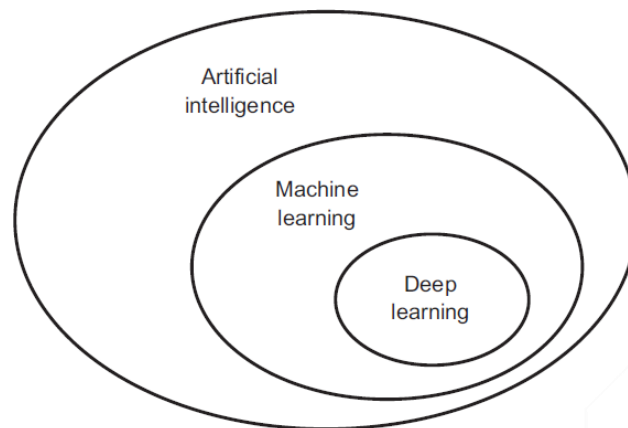
Deep learning adalah sub-bidang dari *machine learning* yang menggunakan jaringan neural artificial (ANNs) dengan banyak lapisan (disebut *layer*) untuk memahami dan memproses data yang kompleks. *Deep Learning* adalah sebuah metode pembelajaran mesin yang menggunakan jaringan neural yang sangat dalam, terdiri dari banyak lapisan (*layer*), untuk mempelajari representasi yang semakin kompleks dari data yang diberikan (Goodfellow, 2016). Pada *Deep Learning*, jaringan saraf yang terdiri dari multiple *layers* (*layer-layer* yang mendalam) digunakan untuk memproses data. Setiap *layer* di dalam jaringan saraf terdiri dari banyak unit komputasi yang disebut neuron atau node. Setiap neuron menerima *input* dari neuron di *layer* sebelumnya, melakukan komputasi, dan mengirimkan *output* ke neuron di *layer* berikutnya. Proses ini dilakukan secara berulang-ulang melalui *layer-layer* yang mendalam hingga mencapai *layer output*, di mana hasil prediksi atau pengklasifikasian diperoleh.

Deep Learning memungkinkan pembelajaran yang otomatis dan hierarkis dari fitur-fitur yang lebih kompleks dan abstrak dalam data. Dengan memperluas jumlah *layer* dan unit di dalam jaringan saraf, *Deep Learning* dapat mengekstraksi representasi yang lebih dalam dan kompleks dari data yang berdampak pada kinerja yang lebih baik dalam tugas-tugas seperti pengenalan pola, wajah, suara, pemrosesan bahasa baku atau alami, dan lain-lain.

Beberapa jenis arsitektur *Deep Learning* yang umum digunakan antara lain *Convolutional Neural Networks* (CNN) yang efektif dalam pengolahan gambar dan pengenalan pola spasial, *Recurrent Neural Networks* (RNN) yang cocok untuk data berurutan seperti teks dan suara, dan *Generative Adversarial Networks* (GAN) yang digunakan dalam pembuatan dan rekonstruksi data baru.

Deep Learning menjadi sangat populer dan sukses karena kemampuannya untuk mempelajari representasi yang kompleks dan abstrak dari data secara end-to-end, tanpa perlu ekstraksi fitur manual. Keberhasilannya dalam berbagai bidang seperti pengenalan wajah, terjemahan mesin, kendaraan otonom, dan banyak lagi, telah membawa kemajuan yang signifikan dalam bidang kecerdasan buatan. Pada Gambar 2.2 akan menunjukkan hubungan antara *Deep Learning*, *Machine Learning*

dan kecerdasan buatan.



Gambar 2. 1 Hubungan antara *Deep Learning*, *Machine Learning* dan *Kecerdasan Buatan* (Chollet, 2018)

2.5 ANN (*Artificial Neural Network*)

Artificial Neural Network (ANN) adalah model komputasi terinspirasi dari struktur dan fungsi jaringan saraf biologis manusia. ANN terdiri dari sejumlah besar unit pemrosesan sederhana yang disebut neuron atau node, yang terorganisir dalam lapisan-lapisan. Setiap neuron menerima input, melakukan operasi matematika pada input tersebut, dan menghasilkan output yang kemudian disampaikan ke neuron-neuron pada lapisan berikutnya. Neuron-neuron dalam lapisan pertama disebut lapisan input, sementara neuron-neuron dalam lapisan terakhir disebut lapisan output. Lapisan-lapisan antara keduanya disebut lapisan tersembunyi, di mana transformasi dan pembelajaran terjadi.

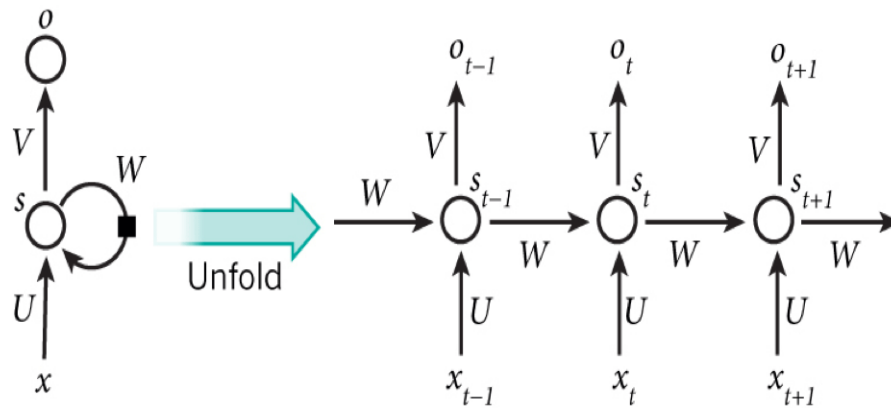
ANN digunakan untuk memodelkan hubungan kompleks antara input dan output, dan dapat digunakan untuk berbagai tugas seperti klasifikasi, regresi, pengenalan pola, dan prediksi. Keunggulan utama dari ANN adalah kemampuannya untuk belajar dari data latih, sehingga ANN dapat menyesuaikan diri dengan pola-pola yang ada dalam data dan melakukan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya. Pelatihan ANN dilakukan melalui proses iteratif yang disebut pembelajaran, di mana parameter-parameter jaringan (misalnya, bobot dan bias) diubah sedemikian rupa sehingga meminimalkan kesalahan antara output yang dihasilkan oleh jaringan dan output yang diharapkan.

Terdapat beberapa jenis arsitektur ANN, termasuk jaringan feedforward dan jaringan rekurensi. Jaringan feedforward adalah jenis paling sederhana dari ANN, di mana informasi mengalir dari lapisan input ke lapisan output tanpa siklus atau umpan balik. Sebaliknya, jaringan rekurensi memiliki struktur yang lebih kompleks, dengan kemampuan untuk menangani data berurutan atau berbasis waktu, karena memiliki siklus atau umpan balik yang mengizinkan informasi untuk beredar kembali ke lapisan sebelumnya. Kedua jenis jaringan ini memiliki kelebihan dan kekurangan masing-masing, dan dipilih tergantung pada tugas yang ingin diselesaikan dan karakteristik data yang ada.

Penerapan ANN telah merambah ke berbagai bidang, termasuk pengenalan wajah, suara, pemrosesan bahasa, pengenalan tulisan tangan, pengenalan pola, dan banyak lagi. ANN telah menjadi salah satu alat yang paling kuat dalam analisis data dan kecerdasan buatan, karena kemampuannya untuk memodelkan hubungan yang rumit dalam data dan melakukan prediksi yang akurat. Dengan perkembangan teknologi dan metode pembelajaran mesin, ANN terus menjadi area penelitian yang sangat aktif, dengan upaya untuk meningkatkan kinerja, efisiensi, dan skalabilitasnya dalam berbagai aplikasi yang beragam.

2.6 RNN (*Recurrent Neural Network*)

Recurrent Neural Network (RNN) adalah jenis arsitektur dalam machine learning yang dirancang khusus untuk menangani data berurutan atau data berbasis urutan, seperti teks, suara, atau rangkaian waktu. RNN mempunyai kemampuan mengingat informasi sebelumnya dalam urutan data, membuatnya sangat cocok untuk tugas-tugas yang melibatkan konteks temporal. Struktur dasar dari RNN terdiri dari serangkaian unit atau sel yang terhubung satu sama lain dalam urutan linier, dengan setiap unit menerima input dari unit sebelumnya dan menghasilkan output yang kembali ke unit berikutnya. Hal ini memungkinkan RNN untuk mempertahankan pemahaman tentang konteks dalam data berurutan dan membuat prediksi berbasis pada informasi sebelumnya. Pada **Gambar 2.2** menunjukkan arsitektur kerja dari metode RNN (*Recurrent Neural Network*)



Gambar 2.2 arsitektur kerja dari metode RNN

Salah satu keunggulan utama dari RNN adalah kemampuannya untuk memproses data berurutan dengan panjang yang bervariasi. Hal ini membuatnya sangat berguna dalam berbagai tugas seperti pemrosesan bahasa alami, pemodelan bahasa, dan pemrosesan sinyal waktu. Namun, RNN juga memiliki beberapa kelemahan, terutama dalam menangani masalah yang melibatkan dependensi jarak jauh antara informasi dalam urutan data. Masalah ini dikenal sebagai masalah vanishing gradients, di mana informasi yang relevan dengan prediksi saat ini dapat hilang atau "lenyap" saat melewati urutan panjang, menghasilkan kinerja yang kurang memuaskan dalam beberapa kasus.

Untuk mengatasi masalah vanishing gradients, berbagai varian dari RNN telah dikembangkan, salah satunya adalah Long Short-Term Memory (LSTM). LSTM adalah jenis arsitektur RNN yang dilengkapi dengan mekanisme pintar untuk mengontrol aliran informasi dalam sel-selnya. Dengan adanya gate kontrol, LSTM dapat memilih untuk menyimpan atau mengabaikan informasi berdasarkan pada kebutuhan, memungkinkannya agar mempertahankan informasi yang relevan dalam karang waktu yang lama. Seiring dengan LSTM, Gated Recurrent Unit (GRU) adalah varian lain dari RNN yang menawarkan efisiensi yang tinggi dalam pelatihan dan penggunaan, dengan mekanisme gating yang lebih sederhana dibandingkan dengan LSTM.

Meskipun RNN memiliki kelebihan dalam menangani data berurutan, ada beberapa tantangan dalam pelatihan dan penggunaannya. Salah satu masalah umum adalah kecenderungan RNN untuk menghasilkan output yang berulang-ulang dalam beberapa kasus, yang dikenal sebagai masalah mode collapse. Selain itu,

pelatihan RNN juga bisa menjadi rumit dan memakan waktu, terutama dalam kasus dataset yang besar atau kompleks. Meskipun demikian, dengan pengembangan terus-menerus dalam bidang machine learning, RNN tetap menjadi alat yang sangat kuat dan relevan dalam pemrosesan data berurutan dan terus digunakan dalam berbagai aplikasi yang berkaitan dengan urutan data.

2.7 HTML (*Hypertext Markup Language*)

HTML (*Hypertext Markup Language*) merupakan bahasa pemrograman yang digunakan untuk membuat dan mendesain halaman *web*. Dengan HTML, para pengembang web dapat membuat struktur dasar dari sebuah halaman *web*, termasuk teks, gambar, tautan, dan elemen-elemen lainnya. HTML menggunakan tag-tag yang ditempatkan di dalam tanda kurung sudut (<>) untuk menandai elemen-elemen ini. Setiap tag memiliki fungsi tertentu dalam menentukan bagaimana sebuah elemen akan ditampilkan di *browser*. Misalnya, tag <p> digunakan untuk menandai sebuah paragraf, sementara digunakan untuk menyisipkan gambar. Dengan kombinasi tag-tag ini, pengembang *web* dapat membuat halaman *web* yang interaktif dan menarik bagi pengguna. HTML juga sering digunakan bersama dengan CSS (*Cascading Style Sheets*) dan JavaScript untuk memberikan tampilan dan interaktivitas yang lebih kompleks pada halaman *web*. Dengan demikian, HTML menjadi fondasi utama dalam pembangunan web modern.

HTML juga memiliki peran penting dalam membangun struktur dan hierarki sebuah halaman *web*. Setiap elemen HTML dapat disusun secara hierarkis, di mana elemen-elemen tersebut dapat menjadi anak, induk, atau saudara satu sama lain, membentuk struktur yang terorganisir dengan baik. Hal ini memungkinkan pengembang untuk membuat halaman *web* yang mudah dibaca, dimengerti, dan diakses oleh pengguna serta mesin pencari. Selain itu, HTML juga memungkinkan integrasi dengan berbagai teknologi *web* lainnya seperti CSS untuk styling dan *layout*, JavaScript untuk interaktivitas dan dinamisme, serta API (*Application Programming Interface*) untuk mengakses berbagai fitur dan layanan *web*. Dengan kombinasi ini, HTML memberikan dasar yang kuat bagi pengembangan aplikasi *web* yang kompleks dan inovatif.

2.8 CSS (*Cascading Style Sheet*)

Cascading Style Sheets (CSS) merupakan bahasa pemrograman yang digunakan untuk mengendalikan tampilan dan gaya dari elemen-elemen HTML pada sebuah halaman *web*. Dengan CSS, pengembang web dapat mengatur berbagai properti seperti warna, ukuran, jenis huruf, tata letak, dan banyak lagi, untuk menciptakan tampilan yang konsisten dan menarik. CSS bekerja dengan cara yang terpusat dan berskala, di mana aturan-aturan gaya dapat didefinisikan secara terpisah dalam file terpisah, dan diterapkan ke elemen-elemen HTML dalam halaman *web*. Konsep kaskade dalam CSS memungkinkan pengembang untuk menetapkan prioritas dan menimpa aturan gaya sesuai dengan hierarki dokumen. Selain itu, CSS juga mendukung konsep pemisahan antara struktur (HTML), presentasi (CSS), dan perilaku (JavaScript), yang memungkinkan pemeliharaan dan pengembangan halaman web yang lebih efisien dan mudah dimengerti. Dengan demikian, CSS menjadi landasan penting dalam pembangunan halaman web modern, memungkinkan pengembang untuk menciptakan tampilan yang menarik dan responsif bagi pengguna.

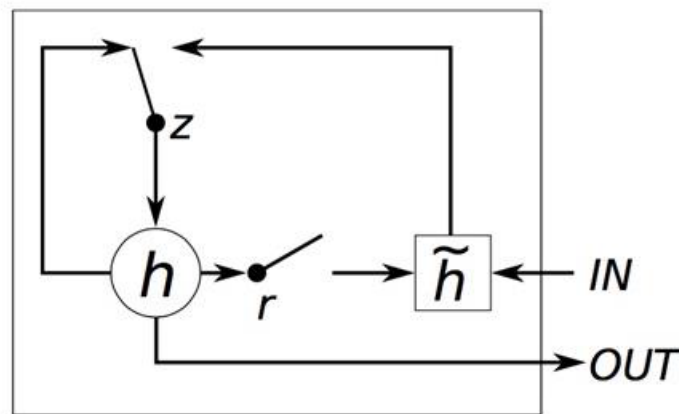
Dalam konteks penelitian, pemahaman yang mendalam tentang CSS sangat penting karena CSS memiliki dampak yang signifikan terhadap pengalaman pengguna pada sebuah situs *web*. Penerapan CSS yang baik dapat meningkatkan kejelasan, keterbacaan, dan estetika sebuah situs *web*, sementara kesalahan atau kekurangan dalam penggunaan CSS dapat mengakibatkan tampilan yang tidak sesuai, lambatnya waktu muat, atau masalah aksesibilitas.

2.9 GRU (*Gated Recurrent Unit*)

Gated Recurrent Unit (GRU) adalah jenis jaringan saraf berulang (*recurrent neural network*, RNN) yang diolah untuk memodelkan data urutan seperti teks atau sinyal waktu. GRU memperbaiki kelemahan RNN tradisional dengan menggunakan mekanisme *gating* yang mengatur aliran informasi, sehingga mampu menangani masalah *vanishing gradient* dan lebih efektif dalam menangkap dependensi jangka panjang dalam data sekuensial. Dibandingkan dengan *Long Short-Term Memory* (LSTM), GRU memiliki struktur yang lebih sederhana dengan lebih sedikit parameter karena menggabungkan sel memori dan mekanisme

pengaturan dalam satu unit. Ini membuat GRU lebih efisien dan sering kali lebih cepat untuk dilatih, sementara tetap memberikan performa yang kompetitif dalam banyak aplikasi, termasuk pemrosesan bahasa alami, prediksi deret waktu, dan pengenalan pola.

Arsitektur dari algoritma *Gated Recurrent Unit* (GRU) melibatkan penggunaan *gate*, yaitu *reset gate* dan *update gate*, untuk mengontrol aliran informasi di dalam model. *Reset gate* menentukan bagaimana cara menggabungkan *input* yang baru dengan informasi sebelumnya, sementara *update gate* menentukan seberapa banyak informasi dari langkah sebelumnya akan disimpan. GRU tidak memiliki *internal memory* yang dibedakan dari *hidden state* dan tidak memiliki *output gate*. Gambaran arsitektur GRU dapat dilihat pada Gambar 2.3 berikut :

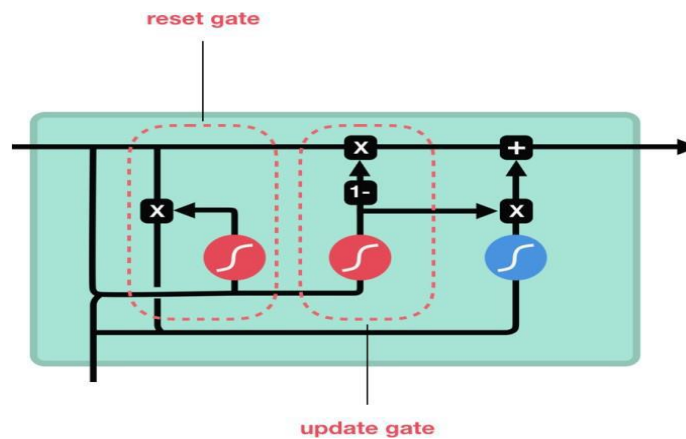


Gambar 2.3 Arsitektur Sistem GRU (Aryo & Derwin, 2022)

Dalam gambar tersebut, r mewakili reset gate, dan z mewakili update gate. Sementara itu, h dan \tilde{h} adalah activation dan candidate activation. Kedua istilah tersebut mengacu pada fungsi aktivasi. GRU (*Gated Recurrent Unit*) belum dieksplorasi secara mendalam karena relatif baru. Masih banyak peluang untuk mengembangkan GRU agar menjadi jenis RNN yang lebih efisien dan memiliki akurasi tinggi..

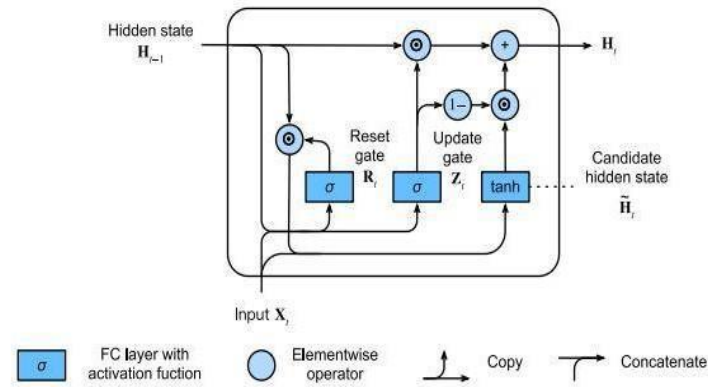
Gated Recurrent Unit (GRU) terdiri dari dua vektor yang mengatur aliran informasi yang diteruskan ke output, dan dasar kerja jaringan GRU memiliki kesamaan dengan Recurrent Neural Network (RNN) dalam ilustrasinya. Perbedaan utama antara keduanya adalah setiap unit berulang dalam GRU adalah jaringan unit

gated yang terdiri dari gerbang untuk mengontrol aliran informasi input dan status tersembunyi sebelumnya. GRU memiliki dua gerbang untuk mengatur informasi yang diabaikan dan yang disimpan. Pada gambar 2.4 adalah gambaran visual dari sel memori beserta gerbang dalam struktur GRU.:



Gambar 2.4 Cell Memory dan Gates GRU (Reyhan Bonovan, 2023)

Seperti yang ditunjukkan dengan gambar 2.4 dalam *Gated Recurrent Unit* (GRU), konsep "*cell memory*" digantikan oleh hidden state yang berperan sebagai memori tunggal yang menyimpan informasi dari langkah waktu sebelumnya dalam urutan data. Interaksi antara *reset gate* (r) dan *update gate* (z) mengatur pengelolaan dan pembaruan informasi dalam hidden state. Reset gate mengontrol seberapa banyak informasi dari hidden state sebelumnya yang akan dipertahankan atau diabaikan dalam perhitungan *candidate activation*, sementara *update gate* mengatur campuran antara informasi baru dan lama dalam pembaruan *hidden state*. Ini memungkinkan GRU untuk secara adaptif mengontrol informasi yang disimpan dalam memori sekuensialnya, menjaga ketergantungan temporal yang diperlukan untuk tugas-tugas prediktif.



Gambar 2.5 Ilustrasi *Cell Memory* pada GRU (Reyhan Bonovan, 2023)

Seperti yang ditunjukkan pada gambar 2.5. Ilustrasi Sel memori ini menggambarkan bagaimana GRU mengatur aliran informasi melalui *reset gate*, *update gate*, dan kandidat konteks. Sel memori GRU memungkinkan model untuk mengirimkan dan mengatur informasi sepanjang melalui dua *gate*, yaitu *reset gate* dan *update gate*, serta menghasilkan *output* berdasarkan informasi yang telah diproses. Ilustrasi ini memberikan gambaran visual tentang bagaimana GRU bekerja dalam mengelola informasi sepanjang sekuens.

$$R_t = \sigma (W_r [H_{t-1}, X_t] + b) \quad (1)$$

$$Z_t = \sigma (W_u [H_{t-1}, X_t] + b) \quad (2)$$

$$\tilde{H}_t = \tanh (W_c [R_t * H_{t-1}, X_t] + b) \quad (3)$$

$$H_t = Z_t * H_{t-1} + (1 - Z_t) * \tilde{H}_t \quad (4)$$

Keterangan :

R_t = Reset Gate

X_t = Input saat ini

Z_t = Update Gate

H_t = New state

\tilde{H}_t = Candidate hidden state

H_{t-1} = Hidden state sebelumnya

Tahapan Algoritma GRU

Tahapan dalam metode GRU adalah :

1. Reset Gate (r):

$$\text{Rumus: } r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

Penjelasan:

- r_t adalah output dari reset gate pada langkah waktu t .
- W_r adalah matriks bobot yang diterapkan pada vektor gabungan dari hidden state sebelumnya (h_{t-1}) dan input saat ini (x_t).
- b_r adalah bias.
- σ adalah fungsi aktivasi sigmoid yang menghasilkan nilai antara 0 dan 1, menentukan seberapa banyak informasi masa lalu yang harus dilupakan atau diabaikan dalam langkah saat ini.

2. Update Gate (z):

$$\text{Rumus: } z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

Penjelasan:

- z_t adalah output dari update gate pada langkah waktu t .
- W_z adalah matriks bobot yang diterapkan pada vektor gabungan dari hidden state sebelumnya (h_{t-1}) dan input saat ini (x_t).
- b_z adalah bias.
- σ adalah fungsi aktivasi sigmoid yang menghasilkan nilai antara 0 dan 1, menentukan seberapa banyak informasi baru yang harus disertakan dalam langkah saat ini.

3. Candidate Activation (h^{\wedge}):

$$\text{Rumus: } h^{\wedge}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b)$$

Penjelasan:

- h^{\wedge}_t adalah kandidat untuk nilai aktivasi pada langkah waktu t .
- W adalah matriks bobot.
- \odot adalah operasi perkalian elemen-wise (element-wise multiplication).

- \tanh adalah fungsi aktivasi tangen hiperbolik yang menghasilkan nilai antara -1 dan 1, mengubah hasil dari kombinasi linier vektor gabungan dari reset gate dan hidden state sebelumnya ($r_t \odot h_{t-1}$) dengan input saat ini (x_t).

4. Hidden State (h):

$$\text{Rumus: } h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h^t$$

Penjelasan:

- h_t adalah hidden state aktual pada langkah waktu t .
- \odot adalah operasi perkalian elemen-wise.
- $(1 - z_t)$ adalah gate untuk memutuskan seberapa banyak informasi dari langkah sebelumnya yang harus dipertahankan.
- z_t adalah gate untuk memutuskan seberapa banyak informasi dari kandidat aktivasi yang harus diperbarui.
- h^t adalah kandidat untuk nilai aktivasi yang telah dihitung.

2.10 Website

Website merupakan sebuah halaman atau kumpulan halaman yang dapat diakses melalui internet. Secara umum, website digunakan untuk menyajikan informasi, layanan, atau konten digital kepada pengguna internet. Dalam konteks bisnis dan komunikasi modern, website menjadi salah satu sarana yang paling penting untuk membangun kehadiran online, berinteraksi dengan audiens, dan menyampaikan pesan atau produk kepada pengguna.

Website memiliki berbagai fungsi tergantung pada tujuan dan kebutuhan penggunaannya. Fungsi utama dari sebuah website adalah menyajikan informasi kepada pengguna. Ini bisa berupa informasi tentang perusahaan, produk, layanan, atau topik tertentu. Selain itu, website juga digunakan untuk berkomunikasi dengan pengguna melalui formulir kontak, forum, atau chat. Di samping itu, website juga dapat digunakan sebagai platform untuk menjual produk atau layanan secara online melalui *e-commerce*.

Ada berbagai jenis *website* yang dapat dibuat sesuai dengan kebutuhan dan tujuan pengguna. Misalnya, website perusahaan atau korporat digunakan untuk

menyajikan informasi tentang perusahaan kepada pengguna, sedangkan website pribadi digunakan untuk berbagi informasi atau cerita pribadi. Selain itu, ada juga website e-commerce yang digunakan untuk menjual produk atau layanan secara online, serta website blog yang digunakan untuk membagikan konten atau informasi secara berkala.

Sebuah website terdiri dari beberapa komponen utama yang bekerja bersama-sama untuk menyajikan konten kepada pengguna. Salah satu komponen utama adalah halaman web, yang merupakan unit dasar dari sebuah website dan berisi konten atau informasi tertentu. Selain itu, website juga memiliki struktur navigasi yang memungkinkan pengguna untuk berpindah antara halaman-halaman yang berbeda. Komponen lainnya termasuk desain grafis, elemen interaktif seperti formulir atau tombol, serta backend yang mengatur penyimpanan dan pengelolaan data.

Dalam era digital saat ini, memiliki website menjadi sangat penting bagi berbagai jenis organisasi atau individu. *Website* merupakan wajah perusahaan atau organisasi di dunia online dan dapat meningkatkan visibilitas serta kredibilitas mereka. Selain itu, website juga dapat menjadi alat yang efektif untuk berinteraksi dengan pengguna, membangun hubungan dengan pelanggan, dan memperluas jangkauan pasar. Dengan memiliki website yang baik, sebuah organisasi dapat mencapai tujuannya lebih efektif dalam dunia digital yang terus berkembang pesat.

2.11 Flask

Text Flask adalah sebuah *framework web* mikro yang ditulis dalam bahasa pemrograman *Python*. Dibuat oleh Armin Ronacher, Flask dirancang untuk menjadi sederhana, fleksibel, dan mudah dipahami, sehingga cocok untuk pengembangan aplikasi *web* yang ringan dan skalabilitas rendah hingga menengah. Flask menyediakan seperangkat alat dan fitur dasar yang menguatkan pengembang untuk membangun aplikasi *web* dengan cepat dan efisien. Salah satu kegunaan utama Flask adalah memudahkan pengembangan aplikasi *web* dengan *Python*. Dengan menggunakan Flask, pengembang dapat membuat berbagai jenis aplikasi *web*, mulai dari *blog* sederhana hingga aplikasi *e-commerce* yang kompleks.

Flask menyediakan kemudahan dalam menangani *routing* URL, mengelola permintaan HTTP, dan menyediakan templating untuk merender halaman *web*. Selain itu, Flask juga memiliki fleksibilitas yang tinggi. Pengembang dapat menyesuaikan struktur aplikasi sesuai dengan kebutuhan mereka, dengan kemampuan untuk menambahkan ekstensi atau paket eksternal untuk fungsi tambahan seperti otentikasi pengguna, manajemen database, atau integrasi dengan layanan pihak ketiga. Keunggulan lain dari Flask adalah skalabilitasnya. Karena sifatnya yang ringan, Flask cocok digunakan untuk membangun aplikasi yang memerlukan kinerja tinggi namun memiliki tingkat kompleksitas yang rendah. Dengan desain yang modular, pengembang dapat menambahkan atau menghapus fitur sesuai dengan kebutuhan aplikasi mereka tanpa mempengaruhi keseluruhan struktur. Dengan demikian, Flask memberikan platform yang kuat dan mudah digunakan bagi pengembang *Python* untuk membangun berbagai jenis aplikasi *web* dengan cepat, efisien, dan dapat diandalkan.

2.12 Penelitian yang Relevan

Berikut beberapa penelitian yang relevan:

1. Studi sebelumnya (Rafika Puspa Wardana, 2020) yang berjudul "Penerapan Model Gated Recurrent Unit untuk Peramalan Jumlah Penumpang Kereta Api di PT. KAI (Persero)" mengevaluasi kinerja algoritma Gated Recurrent Unit dalam memprediksi jumlah penumpang kereta api. Penelitian menggunakan data bulanan tentang jumlah penumpang kereta api dari PT Kereta Api Indonesia (persero). Evaluasi akurasi dan ketepatan prediksi dilakukan dengan menggunakan *Mean Absolute Percentage Error* (MAPE), yang hasilnya menunjukkan nilai sebesar 4,84%. Hasil tersebut mengindikasikan bahwa kemampuan prediksi penumpang PT.KAI telah terbukti baik. Hasil penelitian menunjukkan bahwa algoritma *Gated Recurrent Unit* menghasilkan nilai kesesuaian yang lebih rendah, yang menandakan kinerja yang lebih baik dalam peramalan.
2. Studi sebelumnya (AD. Yulianto, 2021) membahas evaluasi performa algoritma *Gated Recurrent Unit* (GRU) dalam memprediksi harga saham di Indeks Saham Syariah Indonesia. Penelitian ini menggunakan data training

sebesar 80% dan data testing sebesar 20%, dengan berbagai konfigurasi batch size (10, 100, dan 100) dan jumlah epoch (50, 150, dan 250). Hasil analisis menunjukkan bahwa model terbaik, yang memiliki nilai loss function terendah, diperoleh dengan batch size 10 dan 250 epoch. Tingkat akurasi *Mean Absolute Percentage Error* (MAPE) yang diperoleh adalah sebesar 0.802%. Hasil ini menyiratkan bahwa kemampuan prediksi harga saham di Indeks Saham Syariah Indonesia dengan menggunakan metode *Gated Recurrent Unit* terbukti sangat baik.

3. Penelitian sebelumnya (Nugroho Wahyu, 2023) membuktikan efektivitas model *Gated Recurrent Unit* dalam memprediksi harga minyak mentah dunia. Studi ini menghasilkan model yang mampu melakukan prediksi dengan tingkat akurasi yang tinggi. Dari 72 model yang dieksplorasi, model optimal yang menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) terendah adalah yang menggunakan hyperparameter 30 *lookback*, 50 unit GRU, dan 256 *batch size*, dengan MAPE sebesar 2,25%. Penggunaan model ini menunjukkan kemampuan peramalan yang sangat baik. Selain itu, analisis rata-rata dari berbagai model menunjukkan bahwa penggunaan *lookback* sebanyak 30 memberikan nilai rata-rata MAPE terendah, yaitu 3,29%. Hal ini menunjukkan bahwa menggunakan 30 *lookback* mampu menangkap tren atau pola pada dataset dengan sangat baik.
4. Studi sebelumnya (Junta, 2024) menginvestigasi penggunaan *algoritma Long Short-Term Memory* (LSTM) dalam pengembangan chatbot untuk masalah kesehatan mental. Hasil penelitian menunjukkan bahwa chatbot yang dikembangkan menunjukkan kinerja yang baik, dengan tingkat akurasi sebesar 93%, val_akurasi sebesar 83%, dan loss sekitar 0.3%. Meskipun demikian, terdapat sedikit overfitting yang terjadi pada Validasi Loss karena keterbatasan data yang digunakan. Implementasi chatbot ini ke dalam aplikasi web menggunakan kerangka kerja Flask, HTML, CSS, dan JavaScript memungkinkan akses melalui server railway secara online. Saran yang diajukan dalam penelitian ini mencakup penggunaan dataset yang lebih besar untuk meningkatkan kehandalan dan akurasi chatbot berbasis LSTM, serta memberikan pemahaman yang lebih dalam mengenai efektivitas algoritma

yang diterapkan dalam membantu individu mengatasi masalah kesehatan mental sehari-hari.

5. Penelitian sebelumnya (Arunseto, Siti & Bagus, 2021) mengulas tentang pengembangan aplikasi mobile untuk mendeteksi gangguan mental pada mahasiswa menggunakan metode *Forward Chaining*. Aplikasi ini berhasil menghasilkan sebuah *platform* mobile yang mampu mengenali gangguan mental pada mahasiswa dengan memanfaatkan teknik *forward chaining*. Output dari aplikasi ini adalah persentase dari setiap gangguan yang terdeteksi dalam sistem, yang kemudian digunakan sebagai diagnosis awal. Selain itu, aplikasi ini juga dikemas dengan informasi yang informatif. Harapannya, kehadiran aplikasi ini dapat meningkatkan pemahaman pengguna tentang pentingnya menjaga kesehatan mental secara menyeluruh serta meningkatkan kesadaran akan bahayanya gangguan mental

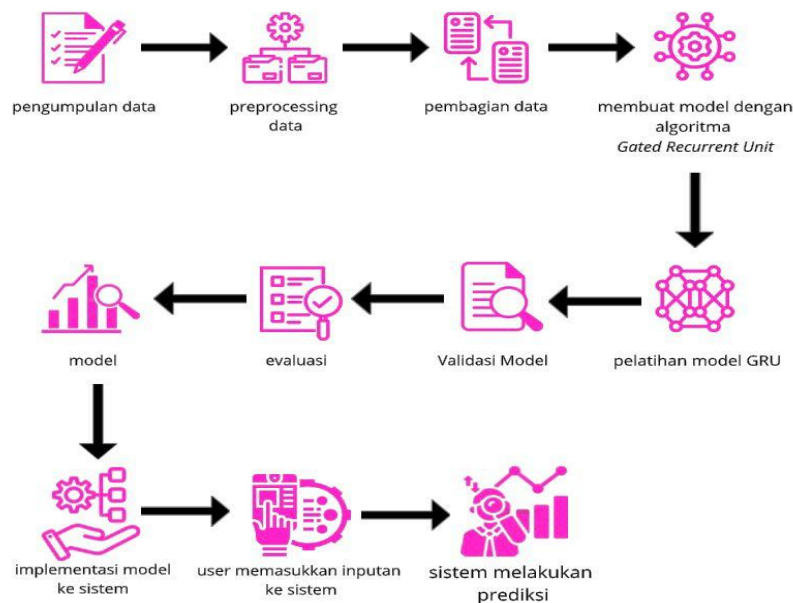
BAB 3

ANALISIS DAN PERANCANGAN

Bab analisis dan perancangan sistem dengan algoritma GRU membahas langkah-langkah yang terlibat dalam menerapkan GRU dalam sistem prediksi. Pada bab ini, pertama-tama dilakukan analisis terhadap kebutuhan dan masalah yang ingin diselesaikan dalam sistem prediksi penyakit *mental health*. Hal ini meliputi pemahaman terhadap data yang akan diproses, tujuan analisis, serta kendala dan batasan yang ada. Setelah itu, dilakukan perancangan sistem dengan mempertimbangkan langkah-langkah yang diperlukan dalam penggunaan GRU.

3.1 Arsitektur Umum

Tahap ini diajukan dalam sistem prediksi penyakit *mental health* dengan menggunakan algoritma GRU (*Gated Recurrent Unit*) terdiri dari beberapa tahapan yang tertera pada Gambar 3.1.



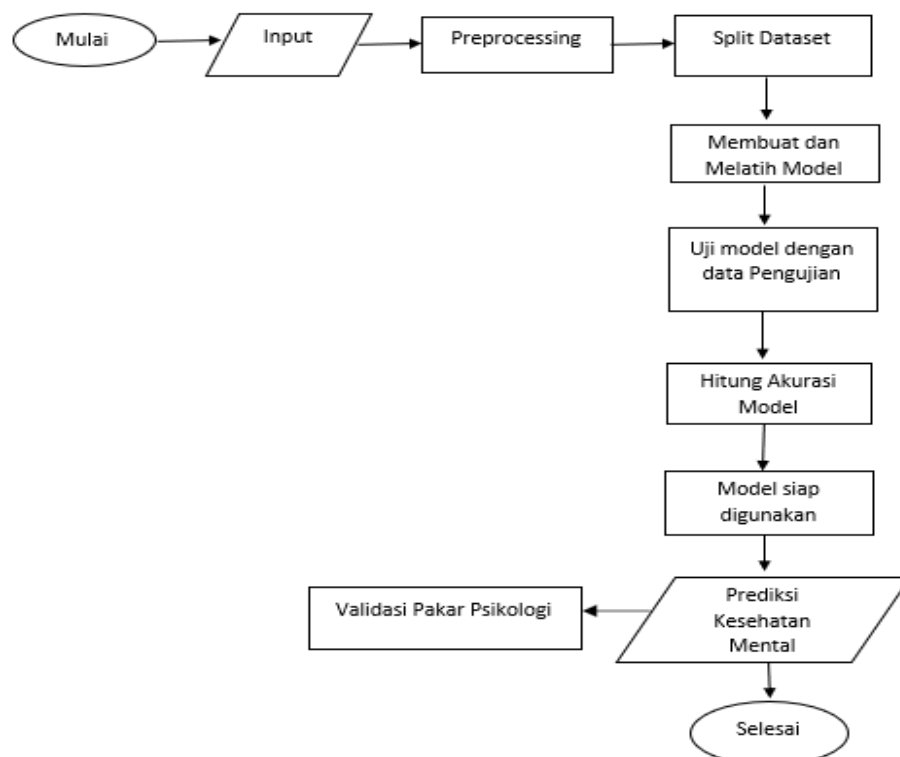
Gambar 3. 1 Arsitektur Umum

Arsitektur umum pada gambar 3.1 dapat dilihat langkah awalnya yaitu melakukan pengumpulan data melalui persebaran distribusi kuesioner kepada mahasiswa fakultas ilmu komputer dan teknologi informasi universitas sumatera utara, lalu jika data yang sudah dikumpulkan langsung di *preprocessing* yaitu

dibersihkan data dari nilai-nilai yang hilang atau tidak valid. Jika diperlukan data akan dinormalisasi. Setelah dibersihkan data akan dibagi menjadi 2 data training dan data *testing*. Lalu Langkah selanjutnya melakukan model dengan algoritma *Gated Recurrent Unit*. Lalu setelah itu uji model menggunakan data pengujian yang tidak dilihat sebelumnya dan hitung akurasi. Lalu validasi model dan evaluasi. Kemudian model siap digunakan jika sudah melewati tahap evaluasi. Setelah itu model yang telah evaluasi diimplementasikan ke dalam sistem lalu user memberikan inputan jawaban ke sistem lalu sistem akan memprediksi sesuai kerja algoritma *gated recurrent unit* dan menghasilkan hasil prediksi yang sudah divalidasi oleh seorang pakar psikologi. Sistem ini dapat membantu dan memberikan wawasan dan informasi seputar Kesehatan mental mahasiswa.

3.2 Flowchart

Perancangan sistem dalam penelitian ini akan direpresentasikan dalam bentuk diagram alir atau *flowchart*. *Flowchart* ini bertujuan untuk memberikan gambaran visual mengenai alur kerja atau proses yang sedang diidentifikasi atau dirancang seperti gambar 3.2 dibawah ini :



Gambar 3. 2 Flowchart Sistem

3.3 Pengumpulan *Dataset*

Pengumpulan data dalam penelitian ini dilakukan melalui *Google Form*, yang berisi 12 pertanyaan dirancang untuk mengukur berbagai aspek kesehatan mental mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi (Fasilkom-TI) Universitas Sumatera Utara (USU). Pertanyaan-pertanyaan tersebut mencakup identitas responden, frekuensi stres dan kecemasan, kualitas dan durasi tidur, aktivitas fisik, interaksi sosial, kehadiran di kelas, produktivitas akademik, pola makan, penggunaan media sosial, dan perasaan umum selama seminggu terakhir. Kuesioner ini didistribusikan secara online melalui grup WhatsApp, *email* fakultas, dan platform media sosial untuk mencapai partisipasi yang luas. Partisipasi bersifat sukarela dengan jaminan kerahasiaan data.

Setelah pengumpulan data selesai, hasil dari *Google Form* diekspor ke dalam format *spreadsheet* untuk dianalisis lebih lanjut. Data ini akan dibersihkan untuk menghapus entri yang tidak valid atau tidak lengkap, dan dinormalisasi untuk konsistensi dalam proses pembelajaran mesin. Data yang diperoleh akan digunakan sebagai input untuk model *Gated Recurrent Unit* (GRU), yang diharapkan dapat memprediksi gangguan kesehatan mental dengan akurat. Dengan demikian, penelitian ini bertujuan untuk memberikan gambaran yang komprehensif mengenai kondisi kesehatan mental mahasiswa Fasilkom-TI USU dan mengembangkan model prediksi yang dapat membantu dalam upaya peningkatan kesejahteraan mental mereka.

3.4 Labelisasi *Dataset*

Labelisasi *dataset anxiety*, depresi, dan belum terdiagnosis merujuk pada proses memberikan label pada data yang menggambarkan klasifikasi atau evaluasi terhadap suatu entitas atau topik. Data yang diberikan label dapat diklasifikasikan ke dalam tiga kategori tersebut:

1. *Anxiety* : Responden yang menunjukkan skor tinggi pada pertanyaan terkait kecemasan akan diberi label "*Anxiety*". Kriteria tambahan mencakup pola tidur yang buruk, frekuensi aktivitas fisik yang rendah, dan laporan perasaan umum yang negatif.
2. Depresi : Responden yang menunjukkan tanda-tanda depresi berdasarkan

jawaban mereka pada kuesioner akan diberi label "Depresi". Indikator lain termasuk kualitas tidur yang sangat buruk, rendahnya produktivitas akademik, dan perasaan umum yang sangat negatif.

3. Normal : Responden yang tidak menunjukkan tanda-tanda signifikan dari kecemasan atau depresi akan diberi label "Normal". Ini mencakup responden dengan skor rendah pada pertanyaan terkait kecemasan dan depresi, serta laporan perasaan umum yang positif atau netral

Proses labelisasi *dataset anxiety*, depresi, dan netral dapat dilakukan secara manual oleh anotator manusia yang mengkategorikan data berdasarkan konteks dan skor yang terdapat pada jawaban *Google Form*. Anotasi pada *dataset* dilakukan oleh tim *annotator* sebanyak 2 orang. Label data *anxiety* akan diubah menjadi angka 1, label data depresi akan diubah menjadi angka 2 dan label data normal akan diubah menjadi angka 0. Perubahan label data yang awalnya dari teks menjadi angka dimaksudkan agar mesin lebih mudah untuk mengelola *dataset* yang sudah dibuat. Contoh *dataset* yang sudah dilabelisasi tertera pada Tabel 3.3.

Tabel 3.3 *Dataset* Hasil Labelisasi

Pertanyaan	jawaban
1. Apakah anda merasa gelisah dan khawatir akan sesuatu hal?	Sangat Sering
2. Apakah anda mudah lelah setelah melakukan aktivitas?	Tidak Sama Sekali
3. Apakah anda mengalami gemeteran pada tangan?	Kadang-kadang
4. Apakah anda sulit berkonsentrasi atau pikiran menjadi blank?	Sering Sekali
5. Apakah anda mudah marah perkara hal sepele?	Kadang-kadang
6. Apakah anda kesulitan untuk tidur malam di waktu yang tepat?	Kadang-kadang
7. Apakah suasana hati anda tertekan dan mudah tersinggung hampir setiap harinya?	Kadang-kadang
8. Apakah anda tidak dapat merasakan perasaan yang positif?	Tidak Sama Sekali
9. Apakah anda mengalami keputusasaan?	Kadang-Kadang
10. Apakah anda merasa tidak memiliki masa depan?	Tidak Sama Sekali

11. Apakah anda merasakan perasaan tidak berharga atau rasa bersalah yang berlebihan terhadap sesuatu? (yang mungkin bersifat delusional) hampir setiap hari (bukan sekadar menyalahkan diri sendiri atau merasa bersalah karena sakit).	Kadang-Kadang
12. Apakah anda sering merasa keinginan bunuh diri yang berulang, atau percobaan bunuh diri atau rencana khusus untuk bunuh diri?	Tidak Sama Sekali
Termasuk Ke :	Anxiety
Label	1

3.5 *Pre-processing Dataset*

Pre-processing adalah tahap awal dalam pengolahan data yang melibatkan serangkaian teknik untuk membersihkan dan mempersiapkan data mentah sebelum diproses lebih lanjut, seperti penghapusan data duplikat, penghapusan nilai yang hilang, dan transformasi data (Ahmed dan Kamal, 2020). Dengan melakukan preprocessing data dengan baik, kita dapat mengurangi risiko overfitting dan underfitting dalam model, meningkatkan kinerja model, dan menghasilkan hasil analisis yang lebih akurat. Selain itu, preprocessing juga membantu menghemat waktu dan upaya dalam tahap analisis selanjutnya, dengan menyediakan dataset yang lebih bersih dan mudah untuk dimengerti. Dengan demikian, preprocessing menjadi tahap penting dalam siklus analisis data yang memastikan bahwa data yang digunakan untuk pengambilan keputusan atau pemodelan memiliki kualitas yang baik dan dapat diandalkan. Tahapan - tahapan yang terdapat pada *pre-processing* pada penelitian ini adalah konversi kolom ke tipe numerik dan menghapus nilai kosong.

3.5.1 Konversi Kolom Ke Tipe Numerik

Langkah konversi kolom dataset ke tipe data numerik merupakan tahap penting dalam proses praproses data. Hal ini dilakukan untuk meyakinkan bahwa semua data yang digunakan dalam analisis dan pelatihan model berada dalam format yang sesuai dan dapat diolah oleh algoritma pembelajaran mesin. Pada penelitian ini, semua kolom kecuali kolom "Nama_lengkap" dikonversi menjadi tipe numerik menggunakan fungsi `pd.to_numeric` dari

pustaka *pandas*. Fungsi ini mencoba mengubah setiap nilai dalam kolom menjadi angka dan menggantikan nilai yang tidak bisa dikonversi dengan NaN (*Not a Number*). Penggunaan `errors='coerce'` dalam fungsi ini memastikan bahwa nilai-nilai yang tidak dapat dikonversi secara langsung diubah menjadi NaN tanpa menyebabkan error, sehingga mempermudah proses pembersihan data berikutnya.

Setelah konversi, langkah selanjutnya adalah menghapus baris-baris yang mengandung nilai NaN menggunakan metode `dropna()`. Langkah ini dilakukan untuk menjaga integritas dan kualitas data, memastikan bahwa hanya baris-baris dengan informasi lengkap yang digunakan dalam analisis dan pelatihan model. Data yang tidak lengkap dapat menyebabkan kesalahan atau bias dalam hasil analisis, sehingga menghilangkannya adalah langkah yang diperlukan. Dengan demikian, konversi kolom ke tipe numerik dan penghapusan nilai NaN merupakan langkah awal yang krusial dalam mempersiapkan dataset agar siap digunakan dalam proses machine learning selanjutnya, memastikan data bersih, konsisten, dan dalam format yang sesuai untuk pemodelan.

3.5.2 Menghapus Nilai Kosong

Menghapus nilai kosong (*missing values*) dari dataset merupakan langkah penting dalam proses pembersihan data untuk memastikan kualitas dan konsistensi data yang akan digunakan dalam analisis lebih lanjut. Dalam penelitian ini, setelah konversi semua kolom menjadi tipe numerik, terdapat kemungkinan adanya nilai-nilai NaN yang muncul akibat kegagalan konversi beberapa data. Nilai-nilai NaN ini perlu dihapus karena dapat menyebabkan kesalahan dalam proses analisis dan pelatihan model machine learning. Penghapusan dilakukan menggunakan metode `dropna()` dari pustaka *pandas*, yang secara efektif menghapus semua baris yang mengandung nilai NaN.

Proses penghapusan nilai kosong ini memastikan bahwa hanya data yang lengkap dan valid yang digunakan dalam analisis selanjutnya. Data yang mengandung nilai NaN dapat mengganggu perhitungan statistik dan

mengurangi keakuratan model prediksi, karena algoritma *machine learning* umumnya tidak dapat menangani nilai kosong dengan baik. Dengan menghapus baris yang memiliki nilai NaN, kita memastikan bahwa dataset yang digunakan bebas dari data yang tidak lengkap, sehingga dapat meningkatkan kualitas model yang dibangun. Langkah ini juga mempermudah proses praproses data selanjutnya, seperti normalisasi dan pembagian data menjadi *set* pelatihan dan pengujian, dengan memastikan semua data yang digunakan berada dalam kondisi optimal untuk analisis dan pemodelan.

3.6 Split Dataset

Sub bab split dataset dalam penelitian ini memegang peranan penting dalam proses pemodelan dan evaluasi model prediksi gangguan kesehatan mental pada mahasiswa. Proses ini dimulai dengan membagi dataset menjadi subset yang berbeda untuk melatih dan menguji model. Bagian terbesar dari dataset, biasanya sekitar 70-80%, akan dialokasikan untuk pelatihan model. Ini berarti model akan diperkenalkan pada sebagian besar data yang tersedia untuk mempelajari pola yang ada. Selanjutnya, sebagian kecil dari dataset, umumnya sekitar 20-30%, akan dialokasikan untuk pengujian model. Data uji ini tidak diperkenalkan kepada model selama proses pelatihan dan digunakan untuk menguji kinerja model yang dilatih. Pembagian dataset ini memastikan bahwa model diuji pada data yang belum pernah dilihat sebelumnya, sehingga memberikan gambaran yang lebih akurat tentang kemampuan model untuk melakukan prediksi pada data baru. Pada penelitian ini akan digunakan split dataset 20:80.

Pembagian dataset biasanya dilakukan secara acak untuk menghindari bias yang mungkin muncul. Sebagian besar dataset akan dialokasikan untuk data pelatihan, yang digunakan untuk melatih model. Sementara itu, sebagian kecil dataset akan dialokasikan untuk data pengujian, yang digunakan untuk menguji seberapa baik model dapat melakukan prediksi pada data baru. Pembagian ini harus dilakukan dengan hati-hati untuk memastikan bahwa proporsi kelas atau pola yang ada dalam data asli juga terjaga dalam kedua subset.

Tujuan dari pembagian dataset adalah untuk mengevaluasi kinerja model dengan menggunakan data yang belum pernah dilihat sebelumnya. Dengan cara ini, kita dapat memastikan bahwa model tidak hanya mempelajari pola yang spesifik pada dataset pelatihan, tetapi juga mampu melakukan generalisasi dengan baik pada data baru. Evaluasi ini penting untuk mengukur seberapa handal dan akurat model dalam memprediksi gangguan mental health pada data yang belum pernah dikenal sebelumnya.

Pada penelitian ini, pengaturan pembagian dataset perlu diperhatikan dengan cermat. Proporsi antara data pelatihan dan data pengujian harus dipilih dengan bijaksana, tergantung pada jumlah total data yang tersedia dan kompleksitas model yang digunakan. Selain itu, teknik validasi silang atau cross-validation juga dapat digunakan untuk memastikan bahwa hasil evaluasi model lebih stabil dan dapat dipercaya.

Setelah pembagian dataset selesai dilakukan, tahap selanjutnya adalah melatih model menggunakan data pelatihan dan menguji performanya menggunakan data pengujian. Analisis hasil dari pengujian ini akan memberikan gambaran tentang seberapa baik model GRU dapat memprediksi gangguan mental health pada data baru. Evaluasi ini akan membantu menentukan apakah model yang dikembangkan memiliki kinerja yang memadai atau perlu ditingkatkan sebelum diterapkan dalam konteks klinis atau praktik yang lebih luas.

3.7 Implementasi GRU (*Gated Recurrent Unit*)

Dalam penelitian ini penulis menggunakan model *Gated Recurrent Unit* (GRU) yang merupakan salah satu jenis dari *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi masalah *vanishing gradient* yang sering terjadi pada RNN konvensional. GRU memperkenalkan mekanisme gating yang mengontrol aliran informasi dan memungkinkan model untuk menangkap dependensi jangka panjang dalam data sequential, seperti data deret waktu atau teks. GRU memiliki dua gerbang utama: *Update Gate* dan *Reset Gate*. Kedua gerbang ini bekerja sama untuk menentukan informasi mana yang harus diteruskan ke status berikutnya dan informasi mana yang harus dilupakan.

- *Update Gate (z_t)*: Gerbang ini memutuskan seberapa banyak informasi dari status sebelumnya (h_{t-1}) yang akan diteruskan ke status saat ini (h_t).
- *Reset Gate (r_t)*: Gerbang ini menentukan seberapa banyak informasi dari status sebelumnya yang akan digunakan untuk calon status saat ini (h).

3.7.1 Hyperparameter Model GRU

Hyperparameter pada model *Gated Recurrent Unit* (GRU) adalah parameter yang harus diatur sebelum proses pelatihan dimulai dan tidak dioptimalkan selama pelatihan. Beberapa *hyperparameter* kunci dalam model GRU meliputi jumlah unit dalam layer GRU, *learning rate*, *batch size*, jumlah *epoch*, dan *dropout rate*. Berikut beberapa *hyperparameter* yang digunakan dalam algoritma GRU :

1. *Learning rate* : *Hyperparameter* ini mengontrol seberapa besar langkah pembelajaran yang diambil selama pelatihan. *Learning rate* yang terlalu tinggi dapat menyebabkan model tidak konvergen, sedangkan *learning rate* yang terlalu rendah dapat membuat pelatihan menjadi lambat. Biasanya, *learning rate* diatur dengan nilai awal yang relatif kecil, misalnya $1e-5$ atau $2e-5$, dan dapat disesuaikan selama pelatihan jika diperlukan.
2. *Batch Size* : *Batch size* adalah jumlah sampel yang diproses secara bersamaan sebelum melakukan update parameter pada model. *Batch size* yang lebih besar dapat meningkatkan efisiensi komputasi tetapi membutuhkan lebih banyak memori. Biasanya, *batch size* yang digunakan dalam *training* GRU adalah 16, 32, atau 64, tergantung pada sumber daya yang tersedia.
3. *Epochs* : *Epochs* menentukan berapa kali seluruh *dataset* akan dilalui selama pelatihan. Jumlah *epochs* dapat bervariasi tergantung pada kompleksitas tugas dan ukuran *dataset*. Untuk pelatihan GRU, biasanya beberapa *epochs* (misalnya 50) sudah cukup untuk mencapai konvergensi.
4. *Dropout* : *Dropout* adalah teknik regularisasi yang digunakan untuk

mencegah *overfitting* dalam jaringan. *Hyperparameter Dropout* mengontrol proporsi neuron yang dinonaktifkan secara acak selama pelatihan. Nilai umum untuk *Dropout* dalam pelatihan GRU adalah sekitar 0,1 hingga 0,3.

5. Jumlah Unit GRU : Jumlah unit dalam layer GRU menentukan kapasitas dan kompleksitas model. Setiap unit di dalam GRU bertanggung jawab untuk menangkap pola dalam data sequential. Lebih banyak unit memungkinkan model untuk menangkap pola yang lebih kompleks, namun juga meningkatkan risiko *overfitting* dan memerlukan lebih banyak daya komputasi.

3.7.2 Algoritma Optimasi

Algoritma optimasi yang digunakan pada penelitian ini adalah *Adaptive Moment Estimation* (Adam). Algoritma optimasi adalah kunci dalam melatih model jaringan saraf seperti *Gated Recurrent Unit* (GRU). Fungsi utamanya adalah meminimalkan fungsi *loss* dengan menyesuaikan bobot dan bias dalam jaringan. Salah satu algoritma optimasi yang paling populer dan sering digunakan adalah Adam (*Adaptive Moment Estimation*). Adam menggabungkan keuntungan dari dua algoritma optimasi lain, yaitu AdaGrad dan RMSProp, dengan menggunakan rata-rata momentum gradien pertama dan kedua untuk menyesuaikan *learning rate* dari setiap parameter. Keunggulan Adam terletak pada kemampuannya untuk secara adaptif mengubah *learning rate* sepanjang pelatihan, yang mempercepat konvergensi dan menghindari jebakan dalam minima lokal.

Selain Adam, ada beberapa algoritma optimasi lain yang sering digunakan, seperti SGD (*Stochastic Gradient Descent*), AdaGrad, dan RMSProp. SGD adalah metode paling dasar, di mana bobot diperbarui untuk setiap sampel data, yang dapat mempercepat pelatihan tetapi juga bisa menghasilkan pembaruan yang berisik. AdaGrad menyesuaikan *learning rate* untuk setiap parameter berdasarkan gradien sebelumnya, cocok untuk data jarang tetapi bisa menyebabkan *learning rate* menjadi terlalu kecil. RMSProp mengatasi kelemahan AdaGrad dengan membatasi pembaruan gradien

menggunakan rata-rata eksponensial, menjaga *learning rate* tetap stabil. Pemilihan algoritma optimasi yang tepat sangat penting untuk memastikan model GRU dilatih secara efisien dan mencapai performa optimal.

3.7.3 Mekanisme *Update Gate*

Update Gate (z_t) bertanggung jawab untuk memutuskan apakah status sebelumnya harus dipertahankan atau diperbarui dengan informasi baru dari *input* saat ini. Ketika z_t mendekati 1, status sebelumnya lebih dominan; sebaliknya, ketika z_t mendekati 0, *input* saat ini lebih dominan. Ini memberikan fleksibilitas pada GRU untuk menangani informasi jangka panjang dan jangka pendek.

3.7.4 Mekanisme *Reset Gate*

Reset Gate (r_t) mengatur seberapa banyak informasi dari masa lalu yang perlu dilupakan dalam perhitungan calon status saat ini (h_t). Ketika r_t mendekati 0, jaringan melupakan hampir semua status sebelumnya dan hanya menggunakan *input* saat ini; ketika r_t mendekati 1, informasi dari masa lalu lebih banyak digunakan. Ini memungkinkan GRU untuk secara dinamis memilih berapa banyak dari sejarah yang relevan untuk setiap langkah waktu.

3.7.5 Perhitungan Calon Status

Calon status (h_t) dihitung menggunakan kombinasi dari *input* saat ini dan status sebelumnya yang telah dimodifikasi oleh *Reset Gate*. Fungsi aktivasi \tanh digunakan untuk memastikan bahwa nilai-nilai dalam calon status tetap berada dalam rentang yang dapat diatur. Ini membantu mengendalikan perbedaan besar dalam data yang bisa menyebabkan ketidakstabilan dalam pelatihan.

3.7.6 Pembaruan Status

Status saat ini (h_t) kemudian dihitung sebagai kombinasi dari status sebelumnya (h_{t-1}) dan calon status (h_t), yang dikendalikan oleh *Update Gate*. Ini memberikan kemampuan kepada GRU untuk memutuskan seberapa

banyak informasi baru yang perlu diterima dan seberapa banyak informasi lama yang perlu dipertahankan. Hasil akhirnya adalah model yang mampu menangkap dependensi jangka panjang tanpa mengalami masalah *vanishing gradient* yang signifikan.

3.8 Deploy Model Ke Server

Deploying model ke *server* menggunakan Flask adalah cara yang populer dan relatif sederhana untuk menyediakan API untuk model yang telah dibuat. Berikut adalah langkah-langkah yang diperlukan untuk melakukan *deploy model* ke *server* menggunakan Flask.

3.8.1 Persiapan Model dan Lingkungan Flask

Langkah pertama adalah mempersiapkan model dan lingkungan Flask. Pastikan model telah dilatih dan disimpan dalam format yang sesuai, seperti HDF5 untuk model Keras. Selanjutnya, penulis melakukan instalasi Flask dan dependensi lainnya di server. Penulis menggunakan pip untuk menginstal Flask

3.8.2 Membuat API dengan Flask

Setelah lingkungan terpasang, penulis membuat API menggunakan Flask dengan nama `app.py` dan mengimpor Flask serta model yang akan digunakan. Kemudian, menentukan *endpoint* API yang akan menerima *input* dan mengembalikan hasil prediksi.

3.8.3 Menjalankan API Flask

Setelah API selesai dibuat, selanjutnya adalah menjalankan aplikasi Flask dengan menjalankan *file Python* yang telah dibuat. Setelah *server* berjalan dengan baik dan API dapat diakses dari luar. Lalu menavigasi ke alamat IP *server* dan *port* yang ditentukan untuk melihat API berfungsi dengan benar.

3.9 Evaluasi

Tujuan tahap evaluasi dalam GRU adalah untuk mengukur kinerja dan keefektifan model dalam melakukan analisis sentimen atau tugas lainnya. Evaluasi dilakukan untuk memperoleh pemahaman yang lebih baik tentang seberapa baik model GRU dalam memprediksi sentimen atau mengklasifikasikan teks. Beberapa tujuan utama dari tahap evaluasi dalam GRU adalah:

1. Mengukur akurasi: Evaluasi dilakukan untuk mengukur sejauh mana model GRU dapat memprediksi dengan benar. Akurasi adalah salah satu metrik yang umum digunakan untuk mengukur performa model, yaitu seberapa sering model memberikan prediksi yang benar.

$$Accuracy = \frac{Total\ Number\ of\ Predictions}{Number\ of\ Correct\ Predictions}$$

2. Mengidentifikasi kelemahan model: Evaluasi membantu mengidentifikasi kelemahan atau keterbatasan model GRU dalam memprediksi. Dengan mengevaluasi model, kita dapat melihat apakah ada pola kesalahan tertentu yang sulit dikenali oleh model.
3. Mengoptimalkan *hyperparameter*: Evaluasi membantu dalam penentuan *hyperparameter* yang optimal untuk model GRU. Dengan menguji model dengan berbagai kombinasi *hyperparameter*, kita dapat menemukan konfigurasi yang memberikan hasil terbaik.

BAB 4

IMPLEMENTASI DAN PENGUJIAN

Bab ini dalam konteks implementasi GRU dalam memprediksi *mental health* membahas langkah-langkah yang dilakukan untuk menerapkan model GRU dalam memprediksi gejala pada penyakit *mental health*.

4.1 Implementasi Sistem

4.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat lunak adalah dokumen yang berisi rincian tentang kebutuhan dan fungsionalitas yang diharapkan dari sebuah perangkat lunak. Spesifikasi ini menjadi landasan bagi pengembang perangkat lunak untuk merancang, mengembangkan, dan menguji produk yang memenuhi kebutuhan pengguna dengan baik. Untuk dapat menerapkan dan menguji implementasi GRU dalam memprediksi *mental health*, diperlukan perangkat keras dengan spesifikasi sebagai berikut.:

1. Processor AMD® Rhyzen 7 series 5700U
2. Memory RAM 8 GB
3. SSD dengan kapasitas 500 GB
4. GPU AMD Radheon Graphic

Untuk mendukung persyaratan perangkat keras yang diperlukan, penulis juga memanfaatkan layanan cloud gratis dari Google yang disebut *Google Colab*. Dalam *Google Colab*.

4.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Sistem Operasi Windows 11 Home 64 *bit operating system*.
2. *Google Colab*
3. *Python3*
4. *Library*: numpy, matplotlib, sklearn, keras ,.

5. Visual Studio Code

4.2 Pengumpulan Dataset

Tahap pertama untuk melakukan pengumpulan dataset penulis membuat sebuah *Google Form* sebagai sarana untuk mengumpulkan jawaban responden yang dapat dilihat pada Gambar 4.1. Setelah membuat *Google Form* kemudian penulis meletakkan 12 pertanyaan di dalam *Google Form* tersebut yang mana pertanyaan-pertanyaan yang ada di dalam *Google Form* bersumber dari pakar penulis yang telah memberikan saran dan masukan terkait pertanyaan dan bobot dari masing-masing pertanyaan tersebut. lalu setelah *Google Form* siap untuk dibagikan penulis membagikan link *Google Form* melalui grup Whatsapp dan media sosial lainnya. Jumlah data yang diambil berjumlah 645 data.

Gambar 4. 1 *Google Form* Sebagai Sarana Pengumpulan Data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Timestamp	Nama lengkap	Jenis Kelamin	Usia	1. Apakah anda	2. Apakah anda	3. Apakah anda	4. Apakah anda	5. Apakah anda	6. Apakah anda	7. Apakah anda	8. Apakah anda	9. Apakah anda	10. Apakah anda	11. Apakah anda	12. Apakah anda
2	2024/03/19	Syabrina ramad	Perempuan	21	kadang-kadang	kadang-kadang	tidak sama se	kadang-kadang	kadang-kadang	tidak sama se	tidak sama se	tidak sama se	tidak sama se	tidak sama se	tidak sama se	tidak sama se
3	2024/03/19	Tridinda Permat	Perempuan	21	kadang-kadang	tidak sama se	kadang-kadang	sering sekali	kadang-kadang	tidak sama se	kadang-kadang	tidak sama se	kadang-kadang	tidak sama se	kadang-kadang	tidak sama se
4	2024/03/19	Ayu Prasinta	Perempuan	21	kadang-kadang	tidak sama se	tidak sama se	kadang-kadang	sering sekali	sering sekali	kadang-kadang	tidak sama se	kadang-kadang	tidak sama se	tidak sama se	tidak sama se
5	2024/03/19	Delpiero	Laki-Laki	22	kadang-kadang	kadang-kadang	tidak sama se	kadang-kadang	tidak sama se	sering sekali	tidak sama se	tidak sama se	tidak sama se	tidak sama se	kadang-kadang	tidak sama se
6	2024/03/19	Wina Octaria S	Perempuan	22	sering sekali	sering sekali	sangat sering	kadang-kadang	kadang-kadang	sangat sering	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	sangat sering	tidak sama se
7	2024/03/19	rani syafitri	Perempuan	21	sangat sering	kadang-kadang	sering sekali	kadang-kadang	tidak sama se	sangat sering	kadang-kadang	tidak sama se	tidak sama se	tidak sama se	kadang-kadang	kadang-kadang
8	2024/03/19	Ainun	Perempuan	21	kadang-kadang	sangat sering	kadang-kadang	tidak sama se	tidak sama se	sangat sering	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	sering sekali	tidak sama se
9	2024/03/20	Haryanda fidi	Laki-Laki	21	tidak sama se	kadang-kadang	tidak sama se	tidak sama se	tidak sama se	kadang-kadang	tidak sama se	tidak sama se	tidak sama se	tidak sama se	tidak sama se	tidak sama se
10	2024/03/20	Ewaldo	Laki-Laki	22	kadang-kadang	kadang-kadang	tidak sama se	sering sekali	tidak sama se	sering sekali	kadang-kadang	kadang-kadang	sering sekali	sering sekali	kadang-kadang	tidak sama se
11	2024/03/20	Muhammad Alii Laki-Laki		19	kadang-kadang	kadang-kadang	tidak sama se	sering sekali	kadang-kadang	tidak sama se	tidak sama se	tidak sama se	tidak sama se	tidak sama se	kadang-kadang	tidak sama se
12	2024/03/20	Hanifah Yumna	Perempuan	19	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	tidak sama se	tidak sama se	tidak sama se	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang

Gambar 4. 2 Dataset Jawaban Responden

4.3 Labelisasi Dataset

Tahap labelisasi *dataset* dalam algoritma GRU adalah mengganti label anxiety, depresi, dan normal dengan angka yang sesuai. Dalam kasus ini, label anxiety akan diganti menjadi angka 1, label depresi menjadi angka 2, dan label normal menjadi angka 0. Jumlah data dalam *dataset* yang digunakan adalah 645 data.

Dalam tahap labelisasi ini, setiap data pada *dataset* akan diperiksa labelnya. Jika data memiliki label *anxiety*, labelnya akan diganti menjadi 1. Jika data memiliki label depresi, labelnya akan diganti menjadi 2. Sedangkan jika data memiliki label normal, labelnya akan diganti menjadi 0. Proses ini dilakukan untuk setiap data dalam *dataset* dapat dilihat pada Gambar 4.3. Setelah labelisasi *dataset* selesai, *dataset* siap digunakan untuk proses *preprocessing dataset*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q		
1	Timestamp	Nama lengkap	Jenis Kelamin	Usia	1. Apakah anda 2. Apakah anda 3. Apakah anda 4. Apakah anda 5. Apakah anda 6. Apakah anda 7. Apakah anda 8. Apakah anda 9. Apakah anda 10. Apakah anda 11. Apakah anda 12. Apakah anda Disease														
2	2024/03/19	Syabrina ramad	Perempuan	21	kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	0	
3	2024/03/19	Tridinda Permat	Perempuan	21	kadang-kadang	tidak sama	se kadang-kadang	sering sekali	kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	1
4	2024/03/19	Ayu Prasinta	Perempuan	21	kadang-kadang	tidak sama	se kadang-kadang	sering sekali	sering sekali	kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	1
5	2024/03/19	Delpiero	Laki-Laki	22	kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	tidak sama	se kadang-kadang	sering sekali	tidak sama	se kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	1
6	2024/03/19	Wina Octaria Si	Perempuan	22	sering sekali	sering sekali	sangat sering	kadang-kadang	kadang-kadang	sangat sering	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	sangat sering	tidak sama	se kadang-kadang	1	
7	2024/03/19	rani syafitri	Perempuan	21	sangat sering	kadang-kadang	sering sekali	kadang-kadang	tidak sama	se kadang-kadang	sering sekali	kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	1	
8	2024/03/19	Ainun	Perempuan	21	kadang-kadang	sangat sering	kadang-kadang	tidak sama	se kadang-kadang	sering sekali	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	sering sekali	tidak sama	se kadang-kadang	1	
9	2024/03/20	Haryanda fidi	Laki-Laki	21	tidak sama	se kadang-kadang	tidak sama	se kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	tidak sama	se kadang-kadang	tidak sama	se kadang-kadang	0	
10	2024/03/20	Ewaldo	Laki-Laki	22	kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	sering sekali	tidak sama	se kadang-kadang	sering sekali	sering sekali	kadang-kadang	tidak sama	se kadang-kadang	1		
11	2024/03/20	Muhammad Alii	Laki-Laki	19	kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	sering sekali	kadang-kadang	tidak sama	se kadang-kadang	tidak sama	se kadang-kadang	tidak sama	se kadang-kadang	1		
12	2024/03/20	Hanifah Yumna	Perempuan	19	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	tidak sama	se kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	kadang-kadang	0		

Gambar 4.3 Dataset yang Sudah dilabelisasi

4.4 Preprocessing Dataset

Pada penelitian ini ada dua tahap dalam proses pembersihan data atau *preprocessing data* yaitu konversi kolom ke tipe numerik yang tertera pada Gambar 4.4.1 dan menghapus nilai kosong yang tertera pada gambar 4.4.2.

4.4.1 Konversi Kolom ke Tipe Numerik

Pada tahap ini, penulis akan mengubah semua kolom data (kecuali kolom pertama yang biasanya merupakan identitas atau nama yang tidak perlu diubah) ke tipe data numerik. Hal ini dilakukan agar data dapat digunakan untuk analisis lebih lanjut dan sebagai input untuk model machine learning.

```
# Convert columns to numeric
dataset.iloc[:, 1:] = dataset.iloc[:, 1:].apply(pd.to_numeric, errors='coerce')
```

Gambar 4.4.1 Konversi Kolom ke Tipe Numerik

Pada gambar 4.4.1 di atas penulis menggunakan metode `apply` dari `pandas` dengan fungsi `pd.to_numeric` untuk mengonversi semua nilai dalam kolom (mulai dari kolom kedua hingga kolom terakhir) menjadi tipe numerik. Argumen `errors='coerce'` digunakan untuk menggantikan nilai yang tidak dapat dikonversi menjadi NaN (Not a Number). Hal ini penting karena data yang tidak dapat diubah menjadi tipe numerik bisa jadi merupakan kesalahan atau anomali yang harus ditangani sebelum analisis lebih lanjut.

4.4.2 Menghapus Nilai Kosong

Setelah konversi ke tipe numerik, beberapa nilai mungkin telah berubah menjadi NaN akibat ketidakmampuan fungsi `pd.to_numeric` mengonversi nilai-nilai tertentu. Langkah berikutnya adalah menghapus baris yang mengandung nilai NaN untuk memastikan bahwa *dataset* bersih dan siap untuk digunakan dalam pelatihan model.

```
# Drop rows with missing values
dataset.dropna(inplace=True)
```

Gambar 4.4.2 Menghapus Nilai Kosong

Pada gambar 4.4.2 di atas menggunakan metode `dropna` dari `pandas` untuk menghapus semua baris yang memiliki setidaknya satu nilai NaN. Argumen `inplace=True` memastikan bahwa perubahan tersebut diterapkan langsung pada *dataset* tanpa perlu membuat salinan baru. Ini penting untuk memastikan bahwa *dataset* kita tidak memiliki nilai yang hilang, yang dapat menyebabkan masalah selama pelatihan model *machine learning*.

4.5 Split Dataset

Sebelum melakukan *training*, data dibagi menjadi dua bagian, yaitu data latih (*training*) dan data uji (*testing*). Data latih digunakan untuk melatih model, sedangkan data uji digunakan sebagai uji akhir untuk melihat seberapa akurat jaringan yang telah dilatih dengan data latih. Ilustrasi tahap pembagian *dataset* bisa di lihat pada Gambar 4.4.

```
# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)
```

Gambar 4. 4 Proses *Splitting Dataset*

4.6 Implementasi Algoritma *Gated Recurrent Unit* (GRU)

Untuk melatih model, diperlukan beberapa komponen yaitu *optimizer*, *loss function* dan *hyperparameter* agar memastikan proses pelatihan berjalan dengan efektif dan efisien. *Optimizer* merupakan algoritma yang digunakan untuk mengupdate bobot model berdasarkan *gradient descent* pada penelitian ini menggunakan adam optimizer dan menggunakan *Dropout* dengan probabilitas 0.2, mengikuti rekomendasi pada penelitian GRU. Penulis juga melakukan deklarasi terhadap beberapa *hyperparameter* berikut, yang diambil berdasarkan rekomendasi untuk GRU:

1. *Epoch*: 50
2. *Batch Size*: 8
3. *Learning rate*: 0,001

Penulis melakukan pemilihan *hyperparameter* di atas berdasarkan beberapa pertimbangan. *Batch size* dengan nilai 8 dipilih oleh penulis dengan alasan jika semakin besar *batch size* yang dibuat maka waktu untuk menyelesaikan satu *batch* juga akan semakin lama. Hal ini mengacu pada penelitian sebelumnya (Osinga, 2018), sehingga *batch size* 8 yang memang akan memakan waktu lama dalam proses *training* tetapi diharapkan dapat lebih adaptif dan mengurangi proses *overfitting* pada model. Selain itu, penggunaan *learning rate* sebesar 0,001 dipilih karena sering dianggap sebagai nilai yang seimbang antara kecepatan dan stabilitas pelatihan. Nilai ini cukup kecil untuk mencegah pembaruan parameter yang terlalu besar, yang dapat menyebabkan model berosilasi atau bahkan *divergen*. Namun, *learning rate* ini juga cukup besar untuk memastikan konvergensi yang relatif cepat. *Learning rate* yang terlalu besar dapat menyebabkan pembaruan parameter yang terlalu besar, sehingga model gagal untuk mencapai konvergensi dan hasil akhir yang optimal. Dengan *learning rate* 0.001, risiko divergensi ini diminimalkan.

Penulis melakukan eksperimen menggunakan satu variasi jumlah *epoch* karena *epoch* yang dipilih oleh penulis merupakan *epoch* yang stabil untuk metode GRU. Gambar 4.5 menampilkan akurasi yang dihasilkan dengan 50 *epoch* yang digunakan dalam penelitian ini Adapun hasil akurasi nya yaitu 0,9612 %. Dan

gambar 4.6 menampilkan proses *training* yang dilakukan dengan epoch 50 kali.

```
# Calculate and print accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f} %")
```

Accuracy: 0.9612 %

Gambar 4. 5 Akurasi yang Diperoleh dengan 50 Epoch

```
Epoch 1/50
65/65 - 4s - loss: 1.8346 - accuracy: 0.3911 - val_loss: 1.4555 - val_accuracy: 0.4651 - 4s/epoch - 55ms/step
Epoch 2/50
65/65 - 0s - loss: 1.3153 - accuracy: 0.5584 - val_loss: 1.1625 - val_accuracy: 0.6047 - 324ms/epoch - 5ms/step
Epoch 3/50
65/65 - 0s - loss: 1.0678 - accuracy: 0.6498 - val_loss: 0.9227 - val_accuracy: 0.7054 - 286ms/epoch - 4ms/step
Epoch 4/50
65/65 - 0s - loss: 0.8273 - accuracy: 0.7588 - val_loss: 0.7321 - val_accuracy: 0.7364 - 299ms/epoch - 5ms/step
Epoch 5/50
65/65 - 0s - loss: 0.5793 - accuracy: 0.8405 - val_loss: 0.5435 - val_accuracy: 0.8527 - 299ms/epoch - 5ms/step
Epoch 6/50
65/65 - 0s - loss: 0.4573 - accuracy: 0.8891 - val_loss: 0.4325 - val_accuracy: 0.8915 - 352ms/epoch - 5ms/step
Epoch 7/50
65/65 - 0s - loss: 0.3618 - accuracy: 0.9105 - val_loss: 0.3860 - val_accuracy: 0.8992 - 428ms/epoch - 7ms/step
Epoch 8/50
65/65 - 0s - loss: 0.3145 - accuracy: 0.9241 - val_loss: 0.2941 - val_accuracy: 0.8915 - 406ms/epoch - 6ms/step
Epoch 9/50
65/65 - 0s - loss: 0.2582 - accuracy: 0.9416 - val_loss: 0.2798 - val_accuracy: 0.9147 - 420ms/epoch - 6ms/step
Epoch 10/50
65/65 - 0s - loss: 0.2399 - accuracy: 0.9377 - val_loss: 0.2363 - val_accuracy: 0.9225 - 310ms/epoch - 5ms/step
Epoch 11/50
65/65 - 0s - loss: 0.2001 - accuracy: 0.9553 - val_loss: 0.2222 - val_accuracy: 0.9380 - 296ms/epoch - 5ms/step
Epoch 12/50
65/65 - 0s - loss: 0.1904 - accuracy: 0.9533 - val_loss: 0.2363 - val_accuracy: 0.9147 - 321ms/epoch - 5ms/step
Epoch 13/50
65/65 - 0s - loss: 0.1625 - accuracy: 0.9533 - val_loss: 0.2084 - val_accuracy: 0.9225 - 315ms/epoch - 5ms/step
Epoch 14/50
65/65 - 0s - loss: 0.1552 - accuracy: 0.9572 - val_loss: 0.1779 - val_accuracy: 0.9612 - 305ms/epoch - 5ms/step
Epoch 15/50
```

Gambar 4. 6 Proses *Training* dengan 50 epoch.

Setelah dilakukan *training* pada metode GRU di *Google Colab*. Maka model yang sudah dibuat akan di simpan ke dalam ekstensi .h5 yang nantinya akan digunakan untuk memprediksi penyakit yang dialami oleh pengguna di *website* yang telah dibuat oleh penulis menggunakan Flask.

Cek Gejala - MentalCare

Form Data Pasien

Nama Lengkap

Umur

Gender
☒ Laki-laki ☐ Perempuan

Deskripsikan Gejala Anda

Anda dapat memilih gejala yang pernah/sedang Anda alami pada pilihan yang tersedia dibawah ini. Pilihlah semua gejala untuk hasil yang lebih akurat.

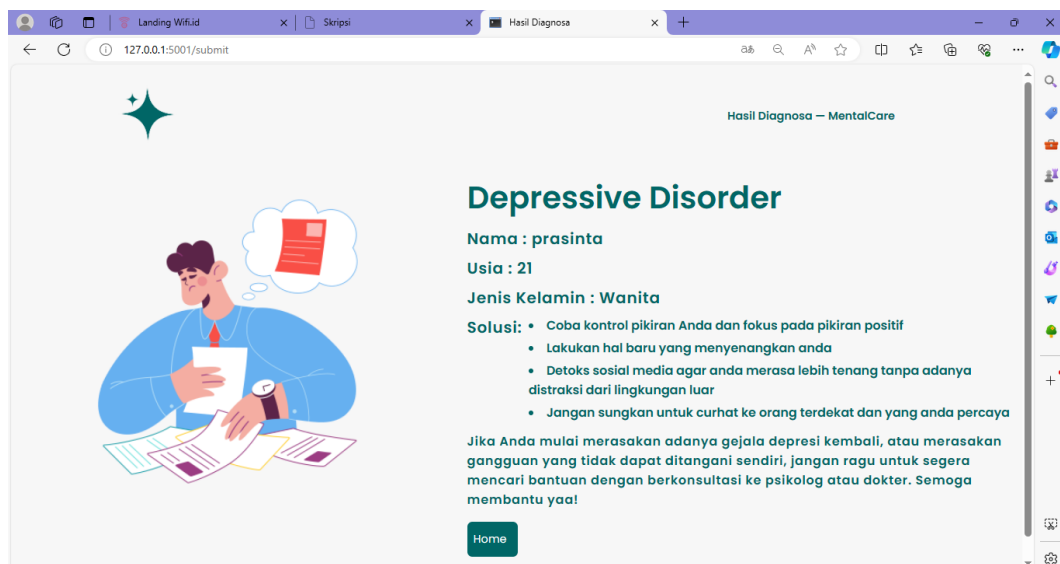
Gejala apa yang pernah/sedang Anda rasakan?

1. Apakah anda merasa gelisah dan khawatir akan sesuatu hal?
2. Apakah anda mudah lelah setelah melakukan aktivitas?
3. Apakah anda mengalami gemeteran pada tangan?
4. Apakah anda sulit berkonsentrasi atau pikiran menjadi blank?
5. Apakah anda mudah marah perkara hal sepele?
6. Apakah anda kesulitan untuk tidur malam di waktu yang tepat?

Gambar 4. 7 Tampilan Halaman Input Gejala

Pada Gambar 4.7 merupakan tampilan *website* untuk menginputkan gejala-gejala yang dirasakan oleh pengguna.

Setelah pengguna menginputkan gejala pada halaman input gejala maka model yang sudah di latih sebelumnya akan memprediksi hasil dari penyakit pengguna. Pada gambar 4.8 merupakan tampilan halaman hasil prediksi penyakit.



Gambar 4. 8 Tampilan Halaman Hasil

4.7 Evaluasi

Setelah model GRU dilatih, langkah penting berikutnya adalah mengevaluasi kinerjanya untuk memastikan bahwa model tersebut dapat memberikan prediksi yang akurat. Salah satu metrik yang digunakan untuk evaluasi ini akurasi. Akurasi adalah salah satu metrik evaluasi yang digunakan untuk mengukur performa model klasifikasi. Akurasi menunjukkan seberapa sering model prediksi benar di antara semua prediksi yang dilakukan. Ini adalah metrik yang paling sederhana dan sering digunakan dalam banyak kasus, terutama ketika kelas-kelas dalam dataset seimbang. Pada gambar 4.9 merupakan evaluasi dengan akurasi.

```
# Calculate and print accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f} %")
```

Accuracy: 0.9612 %

Gambar 4.9 Evaluasi Akurasi

4.7.1 Perhitungan Akurasi

Adapun hasil dataset yang didapat dari persebaran koesiner yang diisi oleh mahasiswa Fakultas Ilmu Komputer dan Teknologi informasi sebanyak 645 dataset. Berdasarkan dataset yang didapat pembagian untuk data latih 80 % dan data uji 20 % sebagai berikut :

Data Pelatihan = $80\% \times 645 = 0.80 \times 645 = 516$ data latih

Data Pengujian = $20\% \times 645 = 0.20 \times 645 = 129$ data uji

Setelah data dibagi kemudian langkah selanjutnya adalah perhitungan manual akurasi dengan rumus jumlah jawaban prediksi yang benar dibagi dengan jawaban yang sebenarnya dikali 100

$$\text{Akurasi} = \frac{\text{Jumlah Data Prediksi Benar}}{\text{Total Data Pengujian}} \times 100\%$$

$$\text{Akurasi} = \frac{124}{129} \times 100 \%$$

$$\text{Akurasi} = 96,12 \%$$

Setelah menghitung akurasi, hasilnya menunjukkan seberapa baik model dalam memprediksi gangguan *mental health* pada mahasiswa. Nilai akurasi yang lebih tinggi menunjukkan bahwa model memiliki kesalahan prediksi yang kecil dan lebih akurat. Dalam penelitian ini, akurasi dihitung untuk memberikan gambaran yang jelas tentang kinerja model GRU yang telah dilatih.

Berdasarkan hasil perhitungan tingkat akurasi sebesar 96% dapat disimpulkan bahwa sistem telah menunjukkan kinerja yang baik dalam konteks prediksi gangguan mental mahasiswa. Penentuan akurasi ini

memegang peranan penting dalam mengevaluasi efektivitas dan keandalan sistem dalam memberikan prediksi sesuai dengan pengetahuan pakar Psikologi.

4.9 Pengujian Sistem oleh Pakar

Pendahuluan

Uji validasi oleh pakar merupakan tahapan penting dalam penelitian ini untuk memastikan bahwa model prediksi yang dikembangkan menggunakan Gated Recurrent Unit (GRU) dapat diandalkan dan relevan dalam konteks prediksi gangguan kesehatan mental pada mahasiswa. Validasi ini melibatkan penilaian oleh profesional di bidang kesehatan mental dan akademisi yang berpengalaman dalam penelitian terkait.

Tujuan Uji Validasi

Uji validasi oleh pakar bertujuan untuk:

1. Memastikan bahwa model prediksi GRU yang dikembangkan memiliki tingkat akurasi dan reliabilitas yang tinggi.
2. Mendapatkan masukan konstruktif dari pakar mengenai kelebihan dan kelemahan model.
3. Menilai kesesuaian model dengan realitas klinis dan kebutuhan intervensi pada mahasiswa.
4. Menyempurnakan model berdasarkan saran dan kritik dari pakar.

Hasil Uji Validasi

1. Evaluasi Akurasi dan Reliabilitas

Hasil evaluasi menunjukkan bahwa model GRU memiliki tingkat akurasi 96% dalam memprediksi gangguan kesehatan mental, namun beberapa variabel input dapat ditambahkan atau diubah untuk meningkatkan reliabilitas.

2. Relevansi Klinis

Pakar menyatakan bahwa model ini cukup relevan dengan kondisi nyata mahasiswa, tetapi perlu penyesuaian lebih lanjut untuk menangani variasi individu yang lebih luas.

3. Implementasi Praktis

Hasil dari sistem prediksi ini digunakan untuk mengidentifikasi mahasiswa yang memerlukan dukungan psikologis lebih awal. Hal ini memungkinkan intervensi lebih dini dan lebih efektif, serta perencanaan program kesehatan mental yang lebih baik di kampus.

4. Kesimpulan dan Rekomendasi

Uji validasi oleh pakar menunjukkan bahwa model GRU yang dikembangkan memiliki potensi yang baik untuk digunakan dalam prediksi gangguan kesehatan mental pada mahasiswa yang membutuhkannya. Evaluasi ini memastikan bahwa model prediksi dapat diandalkan dan berguna dalam praktik nyata, membantu meningkatkan kesejahteraan mahasiswa secara keseluruhan.

Penelitian selanjutnya direkomendasikan untuk menguji model pada populasi yang lebih luas dan beragam serta mengevaluasi dampak intervensi berdasarkan prediksi model.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun hasil yang di peroleh dari setiap tahapan analisis, perancangan dan pengujian sistem yang telah dilakukan pada penelitian ini dengan mengimplementasikan algoritma *Gated Recurrent Unit* yang digunakan untuk memprediksi gangguan *mental health* pada mahasiswa Fakultas Ilmu Komputer dan teknologi Informasi Universitas Sumatera Utara :

1. Hasil penelitian menunjukkan bahwa algoritma *Gated Recurrent Unit* (GRU) menunjukkan kinerja dan performa yang baik dalam menangani dataset yang tidak terlalu besar dan fitur kompleks serta kemampuannya dalam memprediksi gangguan *mental health* mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Namun Algoritma *Gated Recurrent Unit* memiliki kelemahan dikarenakan algoritma tersebut sulit untuk memproses dataset non time series.
2. Dalam prediksi kesehatan mental menggunakan algoritma *Gated Recurrent Unit* (GRU) dengan menggunakan beberapa *hyperparameter*. Hasil akurasi dari percobaan tersebut adalah 96%. *Hyperparameter* yang digunakan mencakup *batch size* sebesar 8, *learning rate* sebesar 0,001 dan *epoch* sebanyak 50 kali.
3. Kualitas *dataset* mempengaruhi akurasi yang diperoleh saat mengimplementasikan GRU. Semakin baik kualitas *dataset* maka akan semakin tinggi nilai akurasi yang diperoleh.
4. Berdasarkan hasil pengujian dan validasi dari pakar Psikolog dapat disimpulkan bahwa Sistem Prediksi gangguan mental yang dibuat sudah baik dan sudah bisa di pergunakan bagi mahasiswa yang membutuhkannya.

5.2 Saran

Berikut merupakan saran yang dapat dipertimbangkan dalam pengembangan sistem selanjutnya:

1. Memperhatikan ketidakseimbangan jumlah data dalam *dataset* dengan mengumpulkan atau menyeimbangkan jumlah label *anxiety*, depresi dan belum terdiagnosis. Hal ini dapat membantu meningkatkan kemampuan algoritma GRU dalam memprediksi label *anxiety*, depresi dan belum terdiagnosis dengan lebih akurat.
2. Melakukan eksplorasi lebih lanjut terhadap *hyperparameter* pada algoritma GRU. Pengaturan *hyperparameter* yang optimal dapat membantu meningkatkan performa algoritma.
3. Melakukan analisis lebih lanjut terhadap kesalahan prediksi yang dilakukan oleh algoritma GRU. Ini dapat membantu mengidentifikasi pola atau fitur yang sulit dikenali oleh algoritma dan merumuskan strategi perbaikan yang sesuai.
4. Melakukan perbandingan dengan metode lainnya untuk melihat keunggulan dan kelemahan algoritma GRU dalam memprediksi kesehatan mental.

DAFTAR PUSTAKA

- Abiri, O., Borhani, S., Sellers, E. W., Jiang, Y., & Zhao, X. (2019). A comprehensive review of EEG-based brain-computer interface paradigms. *Journal of Neural Engineering*, 16(1), 011001.
- Adnan, F., & Amelia, I. (2022). *Implementasi Voice Recognition Berbasis Machine Learning*. 11.
- Alfarezy, R., Ermatita, E., & Wadu, R. M. B. (2022, August). Implementasi Algoritma Naïve Bayes Untuk Analisis Klasifikasi Survei Kesehatan Mental (Studi Kasus: Open Sourcing Mental Illness). In *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya* (Vol. 3, No. 2, pp. 825-834).
- Arunseto, P. (2020). *APLIKASI MOBILE PENDETEKSI GANGGUAN MENTAL MAHASISWA DENGAN MENGGUNAKAN METODE FORWARD CHAINING* (Doctoral dissertation, Sekolah Tinggi Informatika dan Komputer Indonesia).
- Auerbach, R. P., Mortier, P., Bruffaerts, R., Alonso, J., Benjet, C., Cuijpers, P., ... & Kessler, R. C. (2018). WHO world mental health surveys international college student project: prevalence and distribution of mental disorders. *Journal of abnormal psychology*, 127(7), 623.
- Bao Y, Sun Y, Meng S, Shi J, Lu L. 2019-nCoV epidemic: address mental health care to empower society. *The Lancet*. 2020; 395 (10224): 37-38.
- Bonovan, R. (2021). *Implementasi Pendekatan Abstraktif Pada Peringkasan Teks Bahasa Indonesia Menggunakan Metode Recurrent Neural Network* (Doctoral dissertation, Universitas Komputer Indonesia).
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

- Du, K., & Liu, M. (2020). Combining GRU and hierarchical attention network for human activity recognition. *IEEE Access*, 8, 133659-133668.
- Fakhriyani, D. V. (2019). Kesehatan mental. *Pamekasan: duta media publishing*.
- Farwati, M., Salsabila, I. T., Navira, K. R., & Sutabri, T. (2023). *ANALISA PENGARUH TEKNOLOGI ARTIFICIAL INTELLIGENCE (AI) DALAM KEHIDUPAN SEHARI-HARI. 11*.
- Halim, J. K., Herwindiati, D. E., & Hendryli, J. (2022). Penerapan gated recurrent unit untuk prediksi zat pencemar udara. *Jurnal Ilmu Komputer dan Sistem Informasi*, 10(2).
- He, H., Wu, D., Wang, H., & Li, H. (2019). Gated recurrent unit based neural network for short-term load forecasting. *Energy*, 170, 1059-1070.
- Islam, R., Rahman, R. M., & Bashar, M. A. (2019). Hybrid deep learning model with GRU and Bi-LSTM networks for human activity recognition in healthcare. In 2019 22nd International Conference on Computer and Information Technology (ICCIT) (pp. 1-6). IEEE.
- Lin, Y., Zhu, X., Yuan, Z., & Li, B. (2020). A novel model based on GRU with emotional learning for stock price prediction. *Applied Soft Computing*, 86, 105934.
- Oni, M., Lauro, M. D., & Handhayani, T. (2023). PREDIKSI HARGA PANGAN KOTA BANDUNG MENGGUNAKAN METODE GATED RECURRENT UNIT. *Jurnal Ilmu Komputer dan Sistem Informasi*, 11(2).
- RECURRENT UNIT (GRU) UNTUK PREDIKSI HARGA SAHAM BANK KONVENSIONAL DI INDONESIA. *JISTech (Journal of Islamic Science and Technology)*, 6(2).
- Saputra, N. W. (2023). Penerapan Deep Learning Menggunakan Gated Recurrent Unit Untuk Memprediksi Harga Minyak Mentah Dunia. *Building of Informatics, Technology and Science (BITS)*, 5(1), 86-94.
- Tang, B., Liu, Z., & Zhang, J. (2021). Predicting depression in the human brain by

combining long short-term memory and convolutional neural network. *IEEE Journal of Biomedical and Health Informatics*, 25(2), 482-490.

Wang, J., Tang, J., Ding, K., & Xu, H. (2021). A GRU-GAN based model for crisis-related Twitter topic extraction. *IEEE Transactions on Computational Social Systems*, 8(1), 134-144.

Zakariya, F., Zeniarja, J., & Winarno, S. (2024). Pengembangan Chatbot Kesehatan Mental Menggunakan Algoritma Long Short-Term Memory. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 8(1), 251-259.

LAMPIRAN

Lampiran 1. *Listing Program*

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense, Dropout
from tensorflow.keras.optimizers import Adam

# Load dataset
dataset = pd.read_csv("dataset_ayu.CSV")

# Convert columns to numeric
dataset.iloc[:, 1:] = dataset.iloc[:, 1:].apply(pd.to_numeric,
errors='coerce')

# Drop rows with missing values
dataset.dropna(inplace=True)

# Split dataset into features and target
X = dataset.drop(columns=["Nama_lengkap"])
y = dataset.iloc[:, 1:]

# Encode target labels membentuk kembali inputan sesuai yg
diinginkan gru
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(np.argmax(y.values,
axis=1))

# Ensure labels are within valid range
num_classes = len(label_encoder.classes_)
y_encoded = np.clip(y_encoded, 0, num_classes - 1) # Ensure
labels are within [0, num_classes)

# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X,
y_encoded, test_size=0.2, random_state=42)

# Reshape features for GRU input (samples, time steps, features)
masukkan ke variabel baru sesuai gru
X_train = X_train.values.reshape(X_train.shape[0], 1,
X_train.shape[1])
X_test = X_test.values.reshape(X_test.shape[0], 1,
X_test.shape[1])

# Define GRU model
model = Sequential()
model.add(GRU(units=64, input_shape=(X_train.shape[1],
X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(units=32, activation='relu'))
model.add(Dense(units=num_classes, activation='softmax')) #
Adjust number of units to match number of classes
```

```

# Compile model
optimizer = Adam(learning_rate=0.001)
model.compile(optimizer=optimizer,
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Train model ditrain sebanyak 50
model.fit(X_train, y_train, epochs=50, batch_size=8,
validation_data=(X_test, y_test), verbose=2)

# Convert X_test from ndarray to DataFrame
X_test_df = pd.DataFrame(X_test.reshape(X_test.shape[0],
X_test.shape[2]), columns=X.columns)

# Predict classes for test data
y_pred_prob = model.predict(X_test)
y_pred = np.argmax(y_pred_prob, axis=1)

# Decode predicted labels
predicted_labels = label_encoder.inverse_transform(y_pred)

# Create a new DataFrame for predicted labels
predicted_df = pd.DataFrame(predicted_labels,
columns=["Predicted_Disease"])

# Concatenate the predicted DataFrame with the original dataset
dataset_with_predictions =
pd.concat([dataset.reset_index(drop=True), predicted_df],
axis=1)

# Save dataset with predicted class column
dataset_with_predictions.to_csv("dataset_with_predictions.csv",
index=False)

# Print first few rows of dataset with predicted class column
mengeluarkan hasil prediksi dari model
print(dataset_with_predictions.head())

```

Lampiran 2. Curriculum Vitae



Ayu Prasinta

082366804871 | prasintaa19@gmail.com | Medan, Sumatera Utara

Tentang Saya

Saya Ayu Prasinta mahasiswa Ilmu Komputer 2020, Saya memiliki jiwa yang berambisi, jujur, disiplin dan mudah beradaptasi. Dengan keterampilan komunikasi yang baik dan semangat belajar dalam hal-hal baru saya mencari kesempatan untuk dapat berkontribusi dan tumbuh dalam perusahaan yang dinamis dan inovatif.

Pendidikan

2014 - 2016 | SMP Swasta Pertiwi Medan

2017 - 2019 | SMA Swasta Yos Sudarso Medan | Jurusan Ilmu Pengetahuan Alam

2020 - 2024 | Universitas Sumatera Utara | Program Studi Ilmu Komputer | IPK 3.77

Jurusan ini mempersiapkan mahasiswa untuk memiliki kemampuan dalam mengelola berbagai aspek dan perkembangan teknologi.

Pengalaman Organisasi

Anggota di Departemen Komunikasi dan Informasi IMILKOM (Ikatan Mahasiswa S-1 Ilmu Komputer)

- Mengedit feed instagram di akun IMILKOM menggunakan canva atau photoshop
- mencari ide dan konsep untuk reels di akun IMILKOM
- membuat script dan mencari talent untuk tiktok dan reels IMILKOM

Pengalaman Kepanitiaan

Bendahara di ILGANI (Ilmu Komputer Gathering Night), 2022

Anggota divisi Kesehatan ISC (Imilkom Sport Competition), 2022

IT Support di Pekan Ilmiah Mahasiswa Nasional (PIMNAS) 34 USU, 2021

Pengalaman Kerja

Internship di unit payment collection pada PT.Telkom Indonesia

Februari 2023 - Juni 2023

- Bertanggung jawab membantu staf admin terhadap segala dokumen, data serta segala hal administratif yang berkepentingan dengan perusahaan.
- Reminding call pelanggan Telkom untuk tagihan yang sudah jatuh tempo
- Visit pelanggan yang sudah jatuh tempo dan dikenakan denda

Hard Skill

- Microsoft Office
- Html/ CSS
- PHP
- Basic Photoshop
- Aplikasi Canva

Soft Skill

- Public Speaking
- Leadership
- Problem Solving
- Team working
- Flexible and Adaptable