

**REPRESENTASI MATAN JAUHARAH TAUHID
MENGUNAKAN METODE *RECURRENT NEURAL
NETWORK* DENGAN MODEL ARSITEKTUR
*LONG SHORT-TERM MEMORY***

TUGAS AKHIR

Diajukan oleh:

NOVI AYU IRHAMI

NIM. 190705010

**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2023 M/1444 H**

LEMBAR PERSETUJUAN

REPRESENTASI MATAN *JAUHARAH TAUHID* MENGUNAKAN METODE *RECURRENT NEURAL* *NETWORK* DENGAN MODEL ARSITEKTUR *LONG SHORT-TERM MEMORY*

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana
pada Prodi Teknologi Informasi

Oleh:

NOVI AYU IRHAMI
NIM. 190705010


Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui untuk Dimunakaqasyahkan Oleh:

Pembimbing I,


Bustami, M.Sc
NIP. 198604082014031001

Pembimbing II,


Ghufuran Ibnu Yasa', M.T
NIP. 198409262014031005

A R - R A N I R Y

Mengetahui,
Ketua Program Studi Teknologi Informasi



Ima Dwitawati, MBA
NIP. 198210132014032002

LEMBAR PENGESAHAN

REPRESENTASI MATAN JAUHARAH TAUHID MENGUNAKAN METODE *RECURRENT NEURAL NETWORK* DENGAN MODEL ARSITEKTUR *LONG SHORT-TERM MEMORY*

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Pada Prodi Teknologi Informasi

Pada Hari/Tanggal: Selasa, 13 Juni 2023
24 Zulkaidah 1444 H
di Darussalam, Banda Aceh
Panitia Ujian Munaqasah Tugas Akhir

Ketua,



Bustami, M.Sc
NIP. 198604082014031001

Sekretaris,



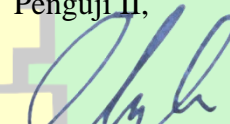
Ghufraan/Ibnu Yasa', M.T
NIP. 198409262014031005

Penguji I,



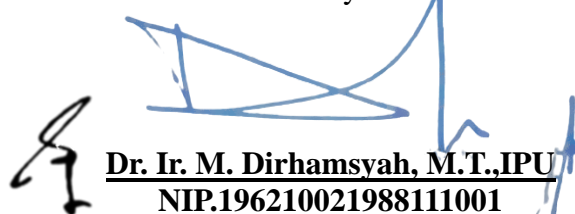
Hendri Ahmadian M.I.M
NIP. 198301042014031002

Penguji II,



Malahayati, M.T
NIP. 198301272015032003

Mengetahui,
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh



Dr. Ir. M. Dirhamsyah, M.T., IPU
NIP.196210021988111001

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Novi Ayu Irhami

NIM : 190705010

Program Studi : Teknologi Informasi

Fakultas : Sains dan Teknologi

Judul Tugas Akhir : Representasi *Matan Jauharah Tauhid* Menggunakan Metode *Recurrent Neural Network* dengan Model Arsitektur *Long Short-Term Memory*

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu mempertanggungjawab atas karya ini;

Bila kemudian hari ini ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat mempertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenakan sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh,
Yang Menyatakan



(Novi Ayu Irhami)

ABSTRAK

Nama : Novi Ayu Irhami
NIM : 190705010
Program Studi : Teknologi Informasi
Judul : Representasi *Matan Jauharah Tauhid* Menggunakan Metode *Recurrent Neural Network* dengan Model Arsitektur *Long Short-Term Memory*
Tanggal Sidang : 13 Juni 2023
Jumlah Halaman : 89 Halaman
Pembimbing I : Bustami, M.Sc
Pembimbing II : Ghufran Ibnu Yasa', M.T

Perkembangan teknologi informasi ini memberikan dampak dan manfaat yang dapat diimplementasikan pada bidang keagamaan. Manfaat teknologi informasi pada bidang keagamaan adalah untuk memudahkan umat Islam mempelajari hukum dan fatwa dalam Islam. Mempelajari ilmu tauhid merupakan kewajiban bagi umat Islam. Salah satu kitab yang mempelajari tentang ilmu tauhid adalah *Matan Jauharah Tauhid*. Mempelajari *Matan Jauharah Tauhid* memerlukan kemampuan pemahaman terhadap makna matan. Upaya pengintegrasian teknologi informasi untuk membantu pembaca memahami makna *Matan Jauharah Tauhid* yaitu dengan merepresentasikan matan kitab melalui proses *text summarization*.

Penelitian ini menggunakan metode *Recurrent Neural Network* (RNN) dengan model arsitektur *Long-Short Term Memory* (LSTM) untuk *text summarization*. Dataset penelitian ini berjumlah 1012 baris data. Dataset terdiri dari terjemahan *Matan Jauharah Tauhid* dan sampel ringkasan dari ahli. Metode RNN dengan model arsitektur LSTM pada penelitian ini menerapkan lima skenario percobaan dengan menggunakan bahasa pemrograman *python* dan *framework keras*. Evaluasi terhadap hasil *text summarization* metode RNN dengan model arsitektur LSTM menggunakan metode ROUGE. Hasil evaluasi dengan metode ROUGE memiliki nilai akurasi tertinggi sebesar 70%. Secara keseluruhan hasil *text summarization* *Matan Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM dapat dikategorikan baik.

A R - R A N I R Y

Kata Kunci: *Text Summarization*, RNN, LSTM, ROUGE

ABSTRACT

Name : Novi Ayu Irhami
Student Number : 190705010
Department : Information Technology
Title : Representation of *Matan Jauharah Tauhid* Using the Recurrent Neural Network Method with Long Short-Term Memory Architecture Model
Date : 13 Juny 2023
Number of Pages : 89 Pages
Supervisor I : Bustami, M.Sc
Supervisor II : Ghufran Ibnu Yasa', M.T

The development of information technology has provided impacts and benefits that can be implemented in the field of religion. The benefit of information technology in the religious field is to facilitate Muslims in studying *Islamic* laws and fatwas. Studying the knowledge of Tauhid is an obligation for Muslims. One of the books that discusses the knowledge of Tauhid is *Matan Jauharah Tauhid*. Studying *Matan Jauharah Tauhid* requires the ability to comprehend the meaning of the text (matan). The effort to integrate information technology to assist readers in understanding the meaning of *Matan Jauharah Tauhid* is by representing the text of the book through the process of text summarization.

This research utilizes the Recurrent Neural Network (RNN) method and Long-Short Term Memory (LSTM) architecture model for text summarization. The research dataset consists of 1012 lines of data, including translations of *Matan Jauharah Tauhid* and expert-generated sample summaries. The RNN method and LSTM architecture in this research apply five experimental scenarios using the Python programming language and the Keras framework. The evaluation of the text summarization results using the RNN method with LSTM architecture model is conducted using the ROUGE method. The evaluation results with the ROUGE method achieved the highest accuracy score of 70%. Overall, the text summarization results of *Matan Jauharah Tauhid* using the RNN method with LSTM architecture model can be categorized as good.

A R - R A N I R Y

Keywords: Text Summarization, RNN, LSTM, ROUGE

KATA PENGANTAR

Bismillahirrahmanirrahim.

Puji syukur penulis panjatkan atas kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul “**Representasi Matan Jauharah Tauhid Menggunakan Metode Recurrent Neural Network dengan Model Arsitektur Long Short-Term Memory**”. Shalawat beserta salam semoga tersampaikan kepada Rasulullah SAW beserta keluarga dan sahabat beliau sekalian yang telah memperjuangkan umat Islam kepada jalan kebenaran dengan dibekali ilmu yang bermanfaat untuk dunia dan akhirat.

Penyusunan tugas akhir ini diajukan sebagai salah satu syarat untuk menyelesaikan tugas akhir pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, UIN Ar-Raniry. Dalam penulisan tugas akhir ini, penulis dengan segala kerendahan hati ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibunda Hajidah dan Ayahanda Amiruddin serta keluarga tercinta yang selalu memberikan doa dan dukungan yang tiada hentinya.
2. Bapak Bustami, M.Sc selaku Pembimbing I dan Bapak Ghufran Ibnu Yasa' M.T selaku Pembimbing II yang senantiasa memberikan arahan dan bimbingan dalam penyusunan tugas akhir ini.
3. Ibu Ima Dwitawati, MBA dan Bapak Khairan AR, M.Kom selaku Ketua dan Sekretaris Prodi Teknologi Informasi, yang telah memberikan bimbingan dan arahan dalam menyelesaikan tugas akhir ini.
4. Ibu Malahayati, M.T. sebagai Penasehat Akademik yang senantiasa memberikan arahan dan motivasi.
5. Ibu Cut Ida Rahmadiana, S.Si selaku *Staff* Prodi Teknologi Informasi, yang senantiasa membantu penulis dalam pemberkasan administrasi.
6. Bapak Dr. Ir. M. Dirhamsyah, M.T.,IPU selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry.

7. Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah membekali penulis dengan ilmu pengetahuan di bidang Teknologi Informasi.
8. Sahabat dan teman-teman yang selalu memberikan dukungan moral dalam menyelesaikan penyusunan tugas akhir ini.
9. Pihak-pihak terkait yang lainnya yang membantu penulis dalam penyusunan tugas akhir ini.

Penulis menyadari sepenuhnya bahwa dalam penyusunan tugas akhir ini tidak cukup dikategorikan sempurna, untuk itu penulis dengan segala kerendahan hati menerima saran dan kritikan guna menyempurnakan penyusunan tugas akhir ini. Akhir kata, semoga tugas akhir ini dapat bermanfaat bagi penulis dan pembaca dan semoga dicatat sebagai sebuah amal kebaikan oleh Allah SWT.

Amiin Ya Rabbal A'lamin.

Banda Aceh, 04 April 2023

Penulis

Novi Ayu Irhami

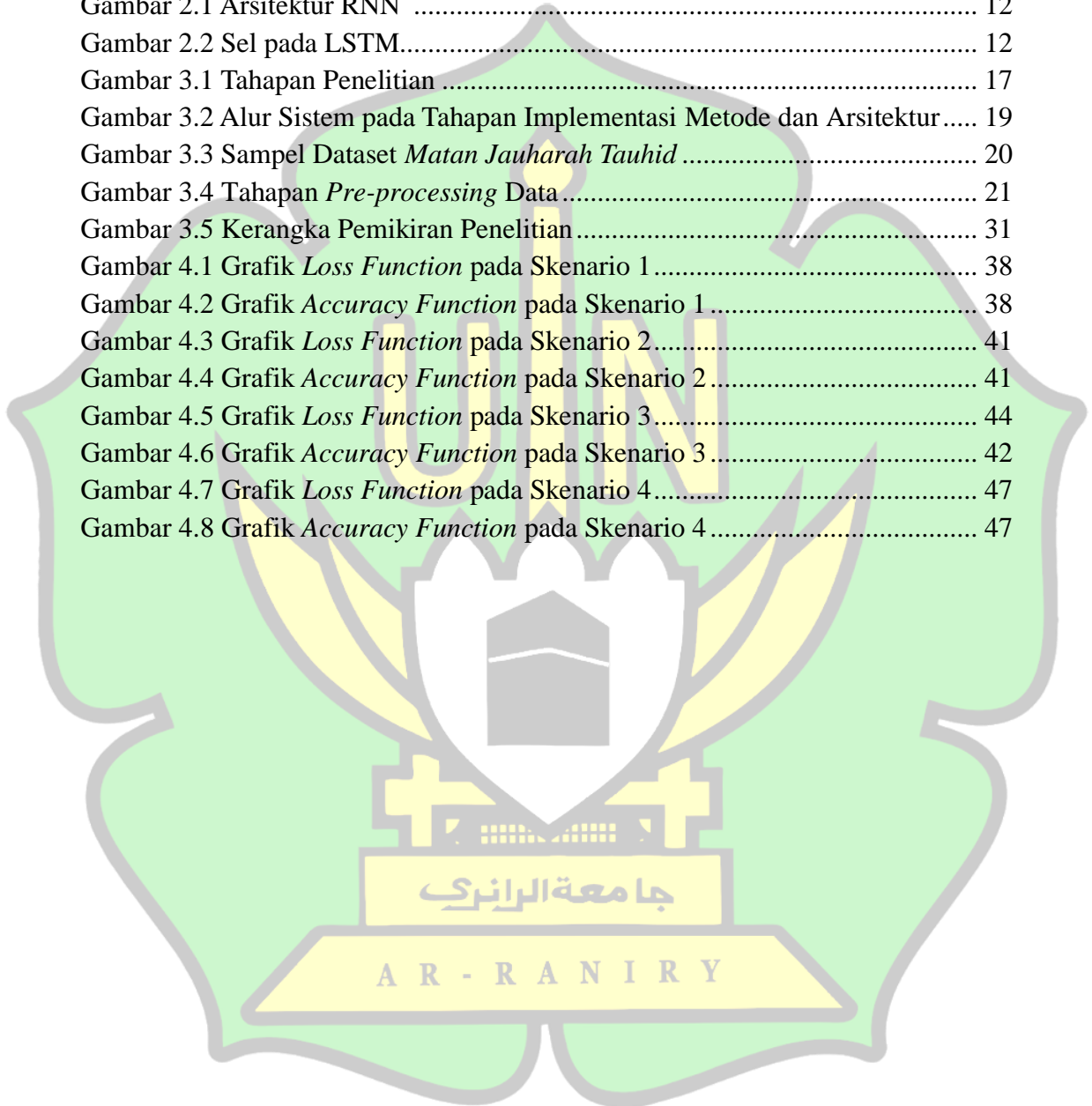
DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Terdahulu	5
2.2 Definisi Kitab <i>Matan Jauharah Tauhid</i>	8
2.3 Definisi dan Hukum Mempelajari Ilmu Tauhid.....	9
2.4 <i>Text Summarization</i>	9
2.5 <i>Natural Language Processing (NLP)</i>	10
2.6 <i>Deep Learning</i>	11
2.7 <i>Recurrent Neural Network (RNN)</i>	12
2.8 <i>Long Short-Term Memory (LSTM)</i>	13
2.9 <i>Tools</i>	14
2.9.1 <i>Python</i>	14
2.9.2 <i>Keras</i>	14
2.9.3 <i>Google Colaboratory</i>	15
2.10 Model Evaluasi	15

BAB III METODOLOGI PENELITIAN	17
3.1 Jenis Penelitian	17
3.2 Tahapan Penelitian	17
3.3 Metode Pengumpulan Data.....	18
3.3.1 Identifikasi Masalah.....	18
3.3.2 Studi Literatur	18
3.4 Metode Simulasi	18
3.4.1 Perumusan Konsep Penelitian.....	19
3.4.2 Implementasi Metode dan Arsitektur.....	19
3.5 Metode Analisis Hasil.....	28
3.5.1 Evaluasi.....	28
3.6 Alat Bantu Penelitian.....	30
3.7 Kerangka Pemikiran Penelitian	30
BAB IV HASIL DAN PEMBAHASAN.....	32
4.1 Hasil <i>Ground Truth Summarization</i>	32
4.2 Hasil Pengujian Terhadap Implementasi Metode dan Arsitektur.....	34
4.2.1 Hasil Pengujian Skenario 1.....	37
4.2.2 Hasil Pengujian Skenario 2.....	41
4.2.3 Hasil Pengujian Skenario 3.....	43
4.2.4 Hasil Pengujian Skenario 4.....	45
4.2.5 Hasil Pengujian Skenario 5.....	49
4.2.6 Evaluasi Hasil Pengujian Oleh Ahli.....	51
4.2.7 Evaluasi Hasil Pengujian dengan ROUGE.....	51
BAB V KESIMPULAN DAN SARAN	55
5.1 Kesimpulan	55
5.2 Saran	55
DAFTAR PUSTAKA.....	50
LAMPIRAN.....	53
RIWAYAT HIDUP.....	71

DAFTAR GAMBAR

Gambar 1.1 <i>Matan Jauharah Tauhid Bait 9-10</i>	2
Gambar 2.1 Arsitektur RNN	12
Gambar 2.2 Sel pada LSTM.....	12
Gambar 3.1 Tahapan Penelitian	17
Gambar 3.2 Alur Sistem pada Tahapan Implementasi Metode dan Arsitektur	19
Gambar 3.3 Sampel Dataset <i>Matan Jauharah Tauhid</i>	20
Gambar 3.4 Tahapan <i>Pre-processing</i> Data	21
Gambar 3.5 Kerangka Pemikiran Penelitian	31
Gambar 4.1 Grafik <i>Loss Function</i> pada Skenario 1	38
Gambar 4.2 Grafik <i>Accuracy Function</i> pada Skenario 1	38
Gambar 4.3 Grafik <i>Loss Function</i> pada Skenario 2	41
Gambar 4.4 Grafik <i>Accuracy Function</i> pada Skenario 2	41
Gambar 4.5 Grafik <i>Loss Function</i> pada Skenario 3	44
Gambar 4.6 Grafik <i>Accuracy Function</i> pada Skenario 3	42
Gambar 4.7 Grafik <i>Loss Function</i> pada Skenario 4	47
Gambar 4.8 Grafik <i>Accuracy Function</i> pada Skenario 4	47



DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Sejenis	6
Tabel 3.1 <i>Case Folding</i> pada Dataset <i>Matan Jauharah Tauhid</i>	22
Tabel 3.2 <i>Stopword Removal</i> pada Dataset <i>Matan Jauharah Tauhid</i>	22
Tabel 3.3 <i>Lemmatization</i> pada Dataset <i>Matan Jauharah Tauhid</i>	23
Tabel 3.4 <i>Tokenizing</i> pada Dataset <i>Matan Jauharah Tauhid</i>	23
Tabel 3.5 Pembagian Jumlah Data <i>Training</i> dan <i>Testing</i>	24
Tabel 3.6 Sampel Teks Proses Perhitungan <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score</i> pada Perhitungan ROUGE	29
Tabel 3.7 Proses Perhitungan <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score</i> pada Perhitungan ROUGE	29
Tabel 4.1 Ground Truth Summarization	32
Tabel 4.2 Pembagian Skenario Percobaan pada Pengujian terhadap Implementasi Metode dan Arsitektur	34
Tabel 4.3 Hasil ROUGE Skenario 1	39
Tabel 4.4 Sampel Hasil Skenario 1 yang Tidak Bisa Dilakukan Summarization .	39
Tabel 4.5 Sampel Hasil Summarization pada Skenario 1	40
Tabel 4.6 Hasil ROUGE Skenario 2	43
Tabel 4.7 Sampel Hasil Skenario 2 yang Tidak bisa Dilakukan Summarization..	43
Tabel 4.8 Sampel Hasil Summarization pada Skenario 2	48
Tabel 4.9 Hasil ROUGE Skenario 3	48
Tabel 4.10 Sampel Hasil Skenario 3 yang Tidak Bisa Dilakukan Summarization	48
Tabel 4.11 Sampel Hasil Summarization pada Skenario 3	48
Tabel 4.12 Hasil ROUGE Skenario 4	50
Tabel 4.13 Sampel Hasil Skenario 4 yang Tidak Bisa Dilakukan Summarization	50
Tabel 4.14 Sampel Hasil Summarization pada Skenario 4	50
Tabel 4.15 Hasil ROUGE Skenario 5	50
Tabel 4.16 Sampel Hasil Summarization pada Skenario 5	50
Tabel 4.17 Evaluasi Hasil Pengujian Oleh Ahli	50
Tabel 4.18 Evaluasi Hasil Pengujian dengan ROUGE	53

BAB I

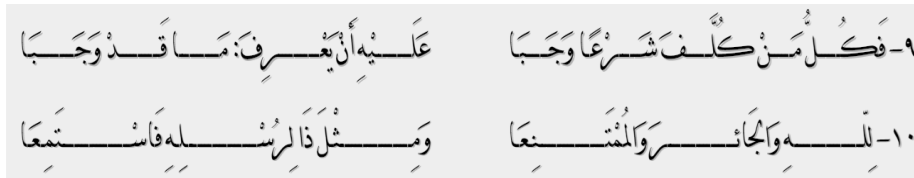
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin canggih memberikan pengaruh yang sangat besar dalam segala bidang keilmuan. Pengaruh teknologi informasi ini memberikan dampak dan manfaat yang dapat diimplementasikan pada bidang keilmuan lainnya, seperti kesehatan, sosial, ekonomi, pendidikan dan keagamaan. Salah satu manfaat teknologi informasi pada bidang keagamaan adalah untuk memudahkan umat Islam mempelajari hukum dan fatwa-fatwa dalam Islam. Manfaat teknologi informasi pada bidang keagamaan ini linear dengan studi yang dikembangkan oleh Nadirsyah Hosen yang dikutip oleh (Ja'far, 2019), mengemukakan bahwa kebutuhan umat muslim akan fatwa (petunjuk keagamaan) ini dapat dilakukan dengan memanfaatkan teknologi melalui reduksi relasi interpersonal keagamaan.

Cabang ilmu agama dalam Islam yang paling mendasar adalah ilmu tauhid. Nilai-nilai keilmuan tauhid sebagai landasan dasar seseorang bertindak, berpikir dan berbuat dalam segala bidang (Hamidah Dedeh, 2021). Syaifullah menuturkan, semakin hari semakin berkurangnya pembelajaran dan pemahaman tentang ilmu tauhid yang berdampak pada masalah *aqidah* dan moral umat (Syaifullah dkk., 2021). Oleh karena itu, pentingnya mempelajari ilmu tauhid untuk diterapkan dalam kehidupan umat Islam sehari-hari.

Mempelajari ilmu tauhid merupakan suatu kewajiban bagi umat Islam sebagaimana visi utama Nabi Muhammad diutus Allah untuk memperbaiki *aqidah*. Salah satu kitab yang mempelajari tentang ketauhidan yaitu *Matan Jauharah Tauhid* yang merupakan kitab yang lumrah dijadikan sebagai salah satu panduan untuk mempelajari ilmu tauhid. Berikut ini adalah penggalan isi dari kitab *Matan Jauharah Tauhid*.



Gambar 1.1 *Matan Jauharah Tauhid* Bait 9-10

Terjemahan bait 9-0 *Matan Jauharah Tauhid* adalah “ Maka bagi setiap orang yang di-*taklif*-kan akan ia pada *syara*’, wajib atas ia mengenal akan barang yang wajib, bagi Allah dan yang harus dan mustahil, dan misal bagi Rasul, maka dengarkan oleh mu ” (Basri, 2015). Ringkasan dari terjemahan *Matan Jauharah Tauhid* pada bait 9 - 10 yaitu, setiap umat Islam wajib memahami hal yang wajib, harus, dan mustahil bagi Allah dan Rasul.

Mempelajari *Matan Jauharah Tauhid* ini memerlukan kemampuan pemahaman terhadap makna matan dalam kitab. Mengingat rentan terjadinya persoalan-persoalan penyimpangan *aqidah* tauhid disebabkan kesalahan dalam memahami makna dan isi dari matan dalam kitab tauhid (Rikza, 2016). Berdasarkan permasalahan tersebut, dibutuhkan upaya pengintegrasian teknologi yang dapat membantu pembaca untuk dapat memahami makna *Matan Jauharah Tauhid* dengan proses *text summarization* untuk merepresentasikan matan kitab dengan cara membuat ringkasan dari terjemahan matan. Teknologi yang dapat diimplementasikan untuk menyelesaikan masalah *text summarization* yaitu *Artificial Intelligence*.

Artificial Intelligence (AI) adalah kecerdasan buatan yang ditambahkan kepada suatu sistem yang dapat diatur dalam konteks ilmiah (Siahaan dkk, 2020). Salah satu metode AI yang dapat digunakan adalah *deep learning*. Metode *deep learning* yang dapat diimplementasikan dalam proses *text summarization* pada suatu teks adalah *Recurrent Neural Network* (RNN). RNN memiliki beberapa modifikasi arsitektur untuk membantu meningkatkan kinerja proses pada *deep learning*. Salah satu arsitektur dari RNN yang sering digunakan untuk *text summarization* adalah *Long Short-Term Memory* (LSTM). Selain itu, penelitian ini juga melibatkan *tool ChatGPT* untuk *text summarization* pada *Matan Jauharah Tauhid*.

ChatGPT (*Generative Pre-Trained Transformer*) adalah *chatbot* yang memanfaatkan kecerdasan buatan atau *artificial intelegent* yang dapat melakukan

interaksi dan membantu manusia dalam menyelesaikan berbagai tugas (Misnawati, 2023). Salah satu keuntungan penggunaan *ChatGPT* adalah kemampuannya untuk memahami dan menghasilkan teks yang kompleks. Dalam konteks penelitian ini, penggunaan *ChatGPT* sebagai *tool* pembandingan terhadap hasil *text summarization* *Matan Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM.

Text summarization dengan model arsitektur LSTM dapat digunakan untuk membuat ringkasan teks secara abstraktif (Elsaid dkk., 2022). *Text summarization* secara abstraktif yaitu teknik ekstraksi teks dari gabungan kalimat yang memungkinkan untuk diparafrasa dengan menampilkan kata baru yang belum ada dalam teks tanpa merubah maksud dari kalimat tersebut (Ivanedra & Mustikasari, 2019).

Berdasarkan uraian di atas, penulis berinisiatif untuk melakukan penelitian dengan judul “Representasi *Matan Jauharah Tauhid* Menggunakan Metode *Recurrent Neural Network* dengan Model Arsitektur *Long Short-Term Memory*”. Penulis berharap dengan adanya penelitian ini dapat menjadi pionir untuk penelitian-penelitian selanjutnya dalam mengintegrasikan teknologi informasi pada pembelajaran *matan kitab*.

1.2 Rumusan Masalah

Adapun rumusan masalah pada penelitian ini sebagai berikut:

1. Skenario apa saja yang digunakan dalam proses *text summarization* *Matan Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM?
2. Bagaimana nilai akurasi hasil *text summarization* *Matan Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM?

1.3 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah diuraikan, tujuan dari penelitian ini, yaitu:

1. Mengetahui skenario percobaan yang digunakan dalam proses *text summarization* *Matan Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM.
2. Mengetahui nilai akurasi hasil *text summarization* *Matan Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM.

1.4 Batasan Masalah

Adapun batasan masalah pada penelitian ini, yaitu:

1. Data yang digunakan adalah kitab *Matan Jauharah Tauhid*.
2. Metode yang digunakan adalah RNN dengan model arsitektur LSTM.
3. Bahasa pemrograman yang digunakan adalah *python* dengan *framework keras*, dan *google colaboratory* sebagai media untuk editor teks.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini, sebagai berikut:

1. Memberikan pengetahuan tentang implementasi *deep learning* menggunakan metode RNN dengan model arsitektur LSTM untuk membantu menyelesaikan permasalahan *text summarization* *Matan Jauharah Tauhid*.
2. Mengetahui nilai akurasi *text summarization* *Matan Jauharah Tauhid* dari implementasi *deep learning* menggunakan metode RNN dengan model arsitektur LSTM untuk *text summarization* *Matan Jauharah Tauhid*.

جامعة الرانيري

A R - R A N I R Y

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu yang relevan dengan penelitian yang dilakukan penulis sangat dibutuhkan sebagai referensi dalam mengembangkan penelitian yang akan dilakukan penulis. Penelitian terdahulu yang menjadi acuan pada penelitian ini yaitu berkaitan dengan penggunaan teknologi *deep learning* untuk proses *text summarization*.

Penelitian dengan topik “*Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM*” (Pawade dkk., 2018), dataset yang digunakan pada penelitian ini berupa inputan *file*. Hasil akurasi pada penelitian ini mencapai 62.86 – 77.14%. Terdapat pula penelitian dengan judul “*Sentence-State LSTM for Text Representation*” (Zhang dkk., 2018), penelitian ini menggunakan dataset tentang *review movie* dengan jumlah dataset sebanyak 8527, dan hasil akurasi pada penelitian ini sebesar 91.93%.

Penelitian tentang penerapan RNN untuk proses *text summarization* dengan tema “*Implementasi Metode Recurrent Neural Network pada Text summarization dengan Teknik Abstraktif*” (Ivanedra & Mustikasari, 2019). Ivanedra dan Mustikasari pada penelitian ini menggunakan dataset yang terdiri dari 4515 artikel beserta ringkasan teks sebagai target output. Hasil akurasi dari penelitian ini mencapai 88%. Selain itu, penelitian yang dilakukan oleh peneliti asal Bangladesh dengan judul “*Sequence-to-sequence Bangla Sentence Generation with LSTM Recurrent Neural Networks*” (Md. Sanzidul dkk., 2019). Penelitian ini menggunakan dataset dari *prothom alo online* yang berjumlah 917 berita dengan 4500 *sentence contain*.

Penelitian dengan studi kasus berita di Indonesia yang berjudul “*Peringkasan Teks Otomatis Bahasa Indonesia Secara Abstraktif Menggunakan Metode Long Short-Term Memory*” (Saputra dkk, 2021). Penelitian ini menggunakan dataset

artikel berita dari INDOSUM dengan jumlah 10.000 sampel data dengan hasil *F1-score* sebesar 0.51429.

Penelitian mengenai *text summarization* dengan menggunakan bahasa Arab yaitu, “*Arabic Language Opinion Mining Based on Long Short-Term Memory (LSTM)*” (Setyanto dkk., 2022). Penelitian ini melakukan pengujian terhadap beberapa dataset. Pengujian pada dataset *Corpus on Arabic Egyptian* menghasilkan akurasi 81.3%. Pengujian menggunakan dataset LABR hasil akurasinya mencapai 87.98 – 90.75%. Hasil akurasi pengujian menggunakan dataset Arabic SemEval mencapai 82.6%.

Penelitian dengan topik “*A Comprehensive Review of Arabic Text Summarization*” (Elsaid dkk., 2022). Penelitian ini berfokus pada pengujian terhadap beberapa algoritma yang dapat digunakan untuk *text summarization*, salah satu algoritma yang diuji adalah LSTM dengan menggunakan dataset *Reuters* dan *Aljazeera*. Hasil akurasi *text summarization* dengan LSTM pada penelitian ini mencapai 68.75 – 88%.

Tabel 2.1 Perbandingan Penelitian Sejenis

No.	PENELITI	METODE	KASUS	DATASET	AKURASI ROUGE
1.	Pawade dkk, 2018	Metode RNN dengan Model Arsitektur LSTM	<i>Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM</i>	Input File Story	62.86 %.
2.	Zhang dkk, 2018	Metode RNN dengan Model Arsitektur LSTM	<i>Sentence-State LSTM for Text Representation</i>	Dataset tentang review movie dengan jumlah dataset	91.93%.

				sebanyak 8527	
3.	Ivanedra & Mustikasari, 2019	Metode RNN dengan Model Arsitektur LSTM	Implementasi Metode <i>Recurrent Neural Network</i> pada <i>Text Summarization</i> dengan Teknik Abstraktif	4515 artikel berita	88%.
4.	Saputra, 2021	Metode RNN dengan Model Arsitektur LSTM	Peringkasan Teks Otomatis Bahasa Indonesia Secara Abstraktif Menggunakan Metode <i>Long Short-Term Memory</i>	INDOSUM dengan jumlah 10.000.	<i>F1-score</i> sebesar 0.51429.
5.	Elsaid dkk, 2022	Metode RNN dengan Model Arsitektur LSTM	<i>A Comprehensive Review of Arabic Text Summarization</i>	<i>Reuteurs</i> dan <i>aljazeera</i>	68.75 %.
6.	Setyanto dkk, 2022	Metode RNN dengan Model Arsitektur LSTM	<i>Arabic Language Opinion Mining Based on Long Short-Term Memory (LSTM)</i>	<i>Corpus on Arabic Egyptian, Corpus on Arabic Egyptian, dan LABR.</i>	81.3 %

Berdasarkan studi literatur terkait perbandingan penelitian terdahulu mengenai implementasi metode RNN dengan model arsitektur LSTM, dapat disimpulkan bahwa:

- Jumlah dataset yang digunakan untuk *training* sangat mempengaruhi tingkat akurasi pengujian. Semakin banyak dataset yang digunakan, maka perbendaharaan kata akan lebih banyak, sehingga tingkat akurasi dari hasil pengujian *text summarization* akan lebih tinggi.
- Data yang digunakan pada proses *text summarization* harus melewati tahapan *pre-processing* dengan baik untuk menjaga keseimbangan data pada saat pengujian. Data yang tidak melewati tahapan *pre-processing* dengan baik, akan mengurangi hasil akurasi.

2.2 Definisi Kitab *Matan Jauharah Tauhid*

Matan adalah teks asli yang ditulis oleh seorang pengarang kitab yang bukan merupakan penjelasan rinci dari kitab yang ditulis oleh ulama. Penulisan kitab dalam bentuk matan ini sudah berlangsung sejak abad pertama *Hijriah* sampai abad ketujuh *Hijriah* dan berlanjut sampai sekarang (Suwarjin, 2018).

Matan Jauharah Tauhid merupakan salah satu contoh matan kitab. *Matan Jauharah Tauhid* adalah karangan Syeikh Ibrahim Al-Laqani. Syeikh Ibrahim Al-Laqani adalah seorang ulama besar di bidang ilmu tauhid dan hadis. Beliau meninggal di umur 70 tahun, bertepatan pada tanggal 3 *Safar* 1041 *Hijriah*. Syeikh Ibrahim Al-Laqani dimakamkan di kaki bukit Ailah (Tgh. Mujiburrahman, 2010).

Kitab *Matan Jauharah Tauhid* berisi pembahasan tentang ilmu tauhid. Naskah kalimat dalam kitab *Matan Jauharah Tauhid* ditulis dalam bentuk bait. Jumlah keseluruhan naskah kalimat pada kitab *Matan Jauharah Tauhid* adalah 144 bait, dengan 1 bait terdiri dari 2 kalimat.

Kandungan isi kitab *Matan Jauharah Tauhid* ini disampaikan oleh pengarang secara implisit melalui bait matan kitab. Makna implisit disini bermaksud bahwa, isi bait kitab *Matan Jauharah Tauhid* tidak dijelaskan secara langsung, melainkan banyak menggunakan kata kiasan. Oleh karena itu, dalam

mengamalkan isi kitab ini sangat penting untuk dapat memahami makna implisit dari bait kata yang terkandung dalam matan.

2.3 Definisi dan Hukum Mempelajari Ilmu Tauhid

Kata tauhid berasal dari bahasa Arab, masdar dari kata *wahhaada* – *yuwahhidu* yang berarti esa, sedangkan menurut istilah tauhid adalah ilmu yang membahas tentang keesaan Allah. Secara istilah *syar'i*, ilmu tauhid berarti mengesakan Allah dalam hal mencipta, menguasai, mengatur dan memurnikan (mengikhlaskan) peribadahan hanya kepada-Nya, meninggalkan penyembahan kepada selain-Nya serta menetapkan *asmaul husna* dan sifat *al-ulya* bagi-Nya dan mensucikan-Nya dari kekurangan dan cacat (Amin, 2019).

Ilmu tauhid merupakan ilmu dasar dalam Islam yang wajib dipelajari oleh umat Islam. Inti dari pembelajaran ilmu tauhid adalah tentang akidah yaitu keyakinan yang tertanam pada diri setiap umat Islam tentang keesaan Allah dengan segala kuasa-Nya.

Hukum mempelajari ilmu tauhid secara umum adalah *fardhu 'ain*, sedangkan mempelajari ilmu tauhid dengan terperinci hukumnya *fardhu kifayah*, sebagaimana firman Allah dalam Quran Surat (Q.S) Al-Ikhlâs ayat 1 - 4. Allah berfirman, Katakanlah (Muhammad): Dialah Allah Yang Maha Esa, Allah tempat meminta segala hal. Allah tidak beranak dan tidak pula diperanakkan. Dan tidak ada yang setara dengan Allah.

2.4 Text Summarization

Text summarization adalah sebuah proses peringkasan teks pada metode *deep learning*. Menurut Joshi dkk, *text summarization* adalah cabang ilmu dalam *Natural Language Processing* (NLP) yang bertujuan untuk merepresentasikan dokumen yang panjang dengan cara mengompres informasi sehingga dapat dengan cepat dipahami dan dibaca oleh pengguna (Joshi dkk., 2019).

Secara umum, terdapat dua teknik dalam proses *text summarization*, yaitu teknik ekstraktif dan abstraktif. Teknik ekstraktif adalah teknik peringkasan teks yang menghasilkan ringkasan dengan cara memberikan hierarki pada tiap kalimat berdasarkan kelas tertentu. Proses ekstraksi yang dilakukan menggunakan matriks

yang telah ditentukan tanpa melakukan perubahan terhadap kata dalam teks (Mubarok, 2021). Teknik ekstraktif dalam proses peringkasan teks tidak memerhatikan tata bahasa dan kecocokan kata dalam kalimat, sehingga hasil ringkasannya terkesan lebih kaku.

Teknik abstraktif merupakan teknik peringkasan teks yang menghasilkan ringkasan dengan cara memahami konteks teks menggunakan NLP untuk memproduksi teks baru yang lebih pendek dari teks asli (Mubarok, 2021). Peringkasan teks dengan teknik abstraktif memungkinkan munculnya kata-kata yang tidak ada pada kalimat asalnya. Menurut Reztaputra & Khodra yang dikutip oleh (Mubarok, 2021), teknik abstraktif merupakan teknik yang paling ideal dan memiliki potensi besar untuk menghasilkan ringkasan teks yang tepat secara tata bahasa, sehingga lebih mudah dipahami oleh pembaca.

2.5 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah pengolahan bahasa alami yang merupakan cabang dari ilmu komputer yang membahas tentang interaksi antara manusia dengan komputer menggunakan bahasa manusia (Migunani & Kevin Aditama, 2020). Sistem NLP bekerja dengan cara mengekstrak makna yang terkandung dalam setiap kalimat, kemudian data dari kalimat tersebut dikumpulkan. Terkadang mesin akan gagal memahami makna dari suatu kalimat dengan jelas, karena sejatinya bahasa dari kalimat terkadang memiliki makna yang beragam dan dapat berubah sepanjang waktu (Mubarok, 2021). Tujuan utama NLP adalah untuk konversi bahasa manusia menjadi data yang terstruktur (Sorin dkk., 2020).

Beberapa teknik yang digunakan dalam proses penyelesaian masalah menggunakan NLP, yaitu:

- *Sentence segmentation*

Sentence segmentation merupakan teknik NLP untuk menentukan unit proses yang terdiri dari beberapa kata. Teknik ini digunakan untuk melakukan identifikasi terhadap batas kalimat.

- *Tokenization*

Teknik merupakan teknik dasar pada NLP. Teknik ini bekerja dengan cara memisahkan kalimat menjadi token-token yang berisi kata.

- *Stopword removal*

Stopword removal merupakan teknik pada NLP yang digunakan untuk menghapus kata yang bersifat non-informatif untuk meningkatkan tingkat akurasi hasil.

2.6 Deep Learning

Deep learning pada dasarnya adalah bagian dari *machine learning*, dimana komputer dapat belajar dari atribut tanpa harus melalui campur tangan manusia secara manual (Sorin dkk., 2020). *Deep learning* diadaptasi dari cara kerja *neuron* otak manusia dan telah terbukti mempunyai fleksibilitas yang besar dan dapat membangun model yang akurat dibandingkan *machine learning* (Aravindh dkk., 2020).

Deep learning disebut juga dengan *deep neural network* yang tersusun dari beberapa lapisan, yang setiap lapisannya terdapat beberapa *neuron*. *Deep learning* dapat menyelesaikan beberapa masalah pada saat yang bersamaan. Algoritma *deep learning* merupakan bagian dari jaringan saraf tiruan. Algoritma *deep learning* ini dibagi menjadi 4 bagian, yaitu:

- Algoritma *Deep Neural Network* (DNN)

Algoritma DNN merupakan algoritma yang diimplementasikan pada proses pengambilan keputusan. Struktur algoritma ini menggunakan model sirkuit saraf kranial makhluk hidup untuk mengenali pola.

- Algoritma *Artificial Neural Network* (ANN)

ANN merupakan algoritma yang mengadaptasi kerja otak manusia yang terdiri dari *neuron*. Cara kerja algoritma ini yaitu dengan menerima data dari nodes pada sebuah *layer* yang kemudian akan ditransfer dan diolah pada *hidden layer*.

- Algoritma *Convolutional Neural Network* (CNN)

CNN merupakan algoritma yang dikembangkan dari *Multi Layer Perceptron* (MLP). Algoritma CNN digunakan untuk menyelesaikan masalah yang melibatkan data dua dimensi.

- Algoritma *Recurrent Neural Network* (RNN)

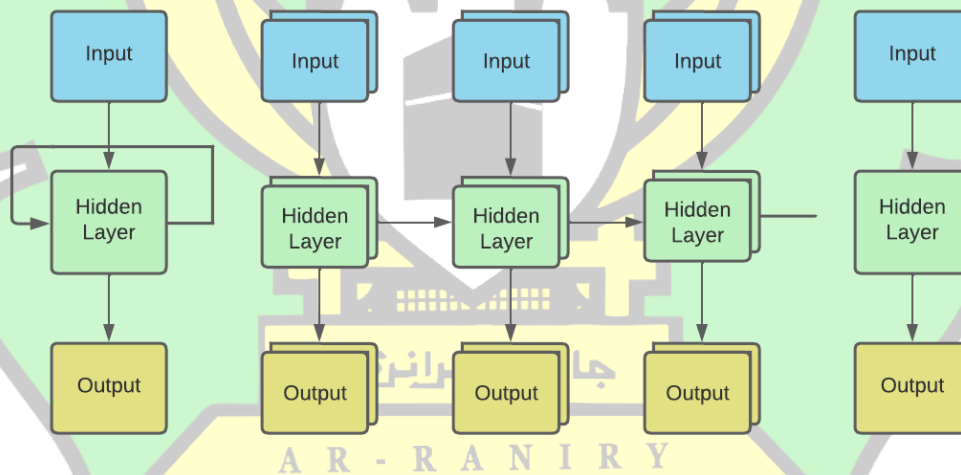
RNN merupakan algoritma *deep learning* yang digunakan untuk memproses data berurutan. RNN merupakan algoritma RNN yang dapat digunakan untuk proses NLP, *speech recognition*, dan *character level language modeling*.

2.7 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) adalah jenis arsitektur dari jaringan saraf tiruan yang melakukan pemrosesan berulang untuk memanggil input data sekuensial (Firmansyah dkk., 2020). RNN juga disebut jaringan umpan balik, dimana pada RNN terdapat *loop* sebagai koneksi umpan balik pada jaringan (Tarkus dkk., 2020).

RNN digunakan untuk menyelesaikan masalah yang berkaitan dengan *time series*. RNN dapat menggunakan informasi yang telah direkam sebelumnya. RNN akan menyimpan semua rekam proses pada *layer* sekuensial untuk dapat ditampilkan kembali jika diperlukan.

Arsitektur RNN dibangun dari 3 layer utama yaitu *output*, *hidden layer*, dan *input*. Arsitektur RNN dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur RNN (Tarkus dkk., 2020)

Gambar 2.1 adalah diagram arsitektur RNN, dimana *input layer* berwarna biru. *Input layer* merupakan jalur masuknya data yang akan diproses dengan metode RNN. *Hidden Layer* dengan persegi berwarna hijau. *Hidden layer* merupakan tempat terjadinya proses *looping* yang memungkinkan RNN untuk menyimpan informasi sementara yang akan digunakan pada pemrosesan data

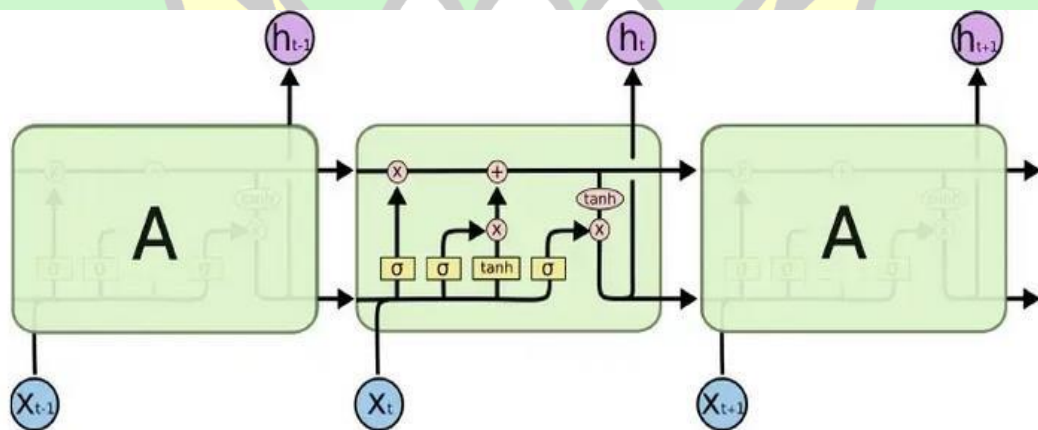
berikutnya. *Output* dengan persegi berwarna kuning, *output layer* merupakan hasil akhir dari pemrosesan RNN.

Arsitektur RNN menunjukkan bagaimana RNN bekerja dengan layer perulangan hidden state sebanyak data waktu yang disediakan $x_t = \text{input}$, $s_t = \text{hidden layer}$, $O_t = \text{output}$. Maka dalam *hidden layer* dapat menyimpan memori dari hasil kalkulasi (Tarkus dkk., 2020).

2.8 Long Short-Term Memory (LSTM)

LSTM dibangun oleh Hochreiter dan Schmidhuber. LSTM terdiri dari beberapa modul dengan pemrosesan berulang (Tarkus dkk., 2020). LSTM merupakan arsitektur dari RNN yang dapat digunakan untuk menyelesaikan masalah mengenai urutan dalam spasial waktu tertentu.

LSTM berfungsi untuk menyimpan informasi berupa pola data. LSTM dapat mempelajari data yang akan disimpan dan data apa yang akan dihapus, karena pada setiap neuron LSTM mempunyai *gate* yang mengatur memori pada setiap neuron (Mubarok, 2021).



Gambar 2.2 Sel pada LSTM (Mubarok, 2021)

Gambar 2.2 merupakan ilustrasi isi sel yang terdapat pada LSTM. LSTM memiliki sel yang lengkap. Sel pada LSTM terdiri dari *layer neuron* yang dilambangkan dengan persegi panjang. Elemen dilambangkan dengan lingkaran. Aliran data yang ada di dalam sel dan antar sel maupun dari sel keluaran (*output h*) dilambangkan dengan panah hitam. LSTM memiliki tiga jenis gates yaitu, *input gate* (X_{t-1}), *forget gate* (X_t), dan *output gate* (X_{t+1}). *Input gate* berfungsi untuk memutuskan nilai dari input untuk diperbaharui pada memori, hasil *input*

gate diinisiasikan dengan $ht-1$. *Forget gate* berfungsi untuk memutuskan informasi yang dihapus dari sel dengan menggunakan hasil dari perhitungan *sigmoid layer* (σ) dan *tanh layer*. Hasil dari *forget gate* diinisiasikan dengan ht . *Output gate* berfungsi untuk memutuskan hasil dari *output*. *Output* harus sesuai dengan hasil dari *input gate* dan *forget gate* yang telah diproses, hasil dari *output gate* ditulis dengan variabel $ht+1$.

2.9 Tools

Tools yang digunakan penulis dalam penelitian ini adalah bahasa pemrograman *python*, *framework keras*, dan *google colab*. *Tools* tersebut akan digunakan penulis dalam proses representasi matan kitab *Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM. *Tools* yang digunakan penulis dapat digunakan dengan cara mengunduh dan instalasi semua *tools* yang dibutuhkan.

2.9.1 Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat kemampuan pemahaman terhadap kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* juga didukung oleh komunitas yang besar (Syahrudin & Kurniawan, 2018).

Python memiliki banyak fungsi yang dapat diaplikasikan pada pemrograman yang berbasis *deep learning*. Bahasa pemrograman *python* mempunyai beberapa *library* yang dapat digunakan untuk proses yang berbasis *deep learning*.

2.9.2 Keras

Keras merupakan *framework* untuk proses *deep learning* tingkat tinggi yang ditulis dengan bahasa *python* dan mampu berjalan di atas *tensorflow*, *CNTK* dan *Theano* (Rahutomo & Sari, 2020). *Keras* dikembangkan untuk dapat digunakan pada pengujian pada *Central Processing Unit* (CPU) dan *Graphic Processing Unit* (GPU).

Keras memiliki fitur *Application Programming Interface* (API) yang dapat membantu pengguna dalam mengembangkan metode RNN. *Keras* juga memiliki model khusus untuk *deep learning* yaitu *Keras Applications*. *Keras Applications* merupakan model keras yang dapat digunakan untuk proses pengambilan pola pada suatu objek dengan metode analisis.

2.9.3 Google Colaboratory

Google colaboratory atau biasa disebut dengan “*Colab*” merupakan produk *open source* dari *Google*. *Colab* menyediakan layanan bagi user untuk menulis dan mengeksekusi kode *python* melalui browser. *Colab* efisien digunakan untuk proses *deep learning* dan analisis data (Soen dkk., 2022).

Google colaboratory pada dasarnya merupakan *platform google* yang mengadaptasi teks editor *Jupyter Notebook*. *Colab* mengimplementasikan Fitur sharing yang berpotensi menjadi solusi untuk pembatasan akses (Febrian Sengkey dkk., 2020). *Google colaburatory* dapat terkoneksi dengan *google drive* untuk mencegah hilangnya kode yang sudah dikerjakan user dikarenakan *error* atau *bug* pada sistem.

2.10 Model Evaluasi

Model evaluasi yang digunakan penulis pada penelitian ini menggunakan evaluasi pengujian oleh ahli dalam skala *likert* dan *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE). Skala *likert* adalah skala penelitian yang digunakan untuk mengukur hasil penelitian yang bersifat subjektif. Dengan skala *likert* ini, validator diminta untuk melengkapi kuesioner yang mengharuskan mereka untuk menunjukkan tingkat penilaiannya terhadap serangkaian pertanyaan. Pertanyaan atau pernyataan yang digunakan dalam penelitian ini biasanya disebut dengan variabel penelitian dan ditetapkan secara spesifik oleh peneliti atau penulis. Tingkat penilaian yang dimaksud dalam skala *likert* ini terdiri dari 5 pilihan skala yang mempunyai gradasi dari tingkat penilaian terendah hingga tertinggi (Hanafiah dkk., 2020).

Metode ROUGE digunakan untuk melakukan evaluasi berbasis sistem untuk mengukur tingkat akurasi hasil pengujian. Menurut Lin yang dikutip oleh (Mubarok, 2021), ROUGE adalah model evaluasi dengan serangkaian matriks

yang digunakan untuk mengevaluasi *automatic text summarization* dan *machine translation* dalam NLP. Cara kerja ROUGE menggunakan metode intrinsik, yaitu dengan membandingkan hasil ringkasan mesin dengan hasil ringkasan ahli. Ringkasan ahli ini disebut juga dengan *ground truth summarization*.

Hasil ringkasan bisa dikategorikan baik, jika memuat informasi utama yang terdapat pada teks asal. Selain itu, jumlah kata hasil dari ringkasan sistem harus dibandingkan dengan jumlah kata pada teks. ROUGE dalam proses perbandingan hasil ringkasan ahli dengan hasil ringkasan sistem, menggunakan persamaan 2.1 (Saputra, 2021).

$$ROUGE = \frac{\sum_{i=1}^n x \in y}{y} \times 100 \% \quad (2.1)$$

Keterangan:

x = Jumlah kata relevan yang dihasilkan sistem.

y = Total kata dalam referensi asal.

ROUGE terdiri dari berbagai matriks untuk menjadi acuan standarisasi hasil ringkasan. Berikut beberapa matriks yang digunakan pada model evaluasi ROUGE (Mubarok, 2021):

- ROUGE-1 menggunakan n-gram (kata) tunggal untuk evaluasi. ROUGE-1 menghitung jumlah n-gram tunggal antara ringkasan sistem dan matan asli.
- ROUGE-2 menggunakan pasangan n-gram untuk menghitung jumlah pasangan n-gram antara ringkasan sistem dengan matan asli.
- ROUGE-L untuk perbandingan yang berbasis *Longest Common Subsequence* (LCS). LCS merupakan pebandingan berdasarkan urutan terpanjang untuk menghitung kesamaan struktur level kalimat secara alami dan mengidentifikasi kombinasi terpanjang dalam urutan n-gram secara otomatis.

BAB III

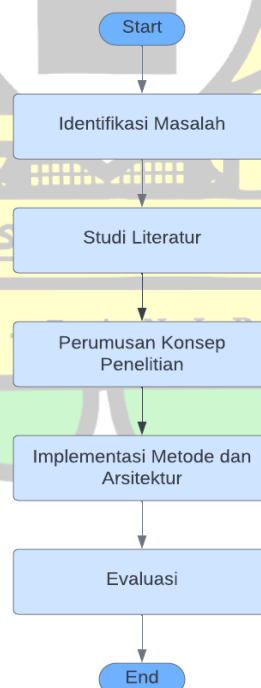
METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini merupakan jenis penelitian kuantitatif. Penelitian kuantitatif merupakan jenis penelitian yang didasarkan pada analisa dengan menggunakan metode numerik untuk memberikan jawaban atas pertanyaan penelitian. Tujuan penelitian kuantitatif yaitu untuk mengembangkan dan menggunakan model matematis atau teori-teori yang berkaitan dengan fenomena yang terjadi. Pemilihan jenis penelitian kuantitatif berdasarkan penyelesaian masalah dalam penelitian ini yang menggunakan pendekatan numerik dalam mengimplementasikan metode RNN dengan model arsitektur LSTM untuk representasi *Matan Jauharah Tauhid* dengan proses *text summarization*.

3.2 Tahapan Penelitian

Tahapan penelitian ini menjelaskan alur penelitian yang digunakan oleh penulis pada penelitian ini. Alur tahapan penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.3 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan penulis pada penelitian ini berdasarkan dari identifikasi masalah dan studi literatur. Identifikasi masalah dilakukan oleh penulis melalui observasi. Studi literatur dilakukan oleh penulis dengan membandingkan penelitian-penelitian sebelumnya untuk dapat diadaptasi pada penelitian ini.

3.3.1 Identifikasi Masalah

Tahapan ini dilakukan penulis untuk menemukan suatu permasalahan untuk dilakukan penelitian sesuai dengan bidang ilmu yang dipelajari penulis. Masalah yang diidentifikasi oleh penulis adalah kesulitan memahami makna implisit dari *Matan Jauharah Tauhid*. Penelitian ini dilakukan penulis sebagai upaya untuk menyelesaikan permasalahan tersebut dengan mengintegrasikan metode RNN dengan model arsitektur LSTM untuk proses *text summarization* *Matan Jauharah Tauhid*.

3.3.2 Studi Literatur

Studi literatur merupakan metode yang digunakan penulis untuk mendapatkan gambaran rinci tentang permasalahan yang ada pada proses *text summarization* *Matan Jauharah Tauhid*. Penulis melakukan studi dengan mengkaji penelitian-penelitian sebelumnya yang relevan dengan penelitian yang dilakukan penulis. Referensi yang diperoleh dari tahap studi literatur ini mencakup penelitian yang berhubungan dengan implementasi *deep learning* dengan metode RNN dengan model arsitektur LSTM untuk proses *text summarization*.

3.4 Metode Simulasi

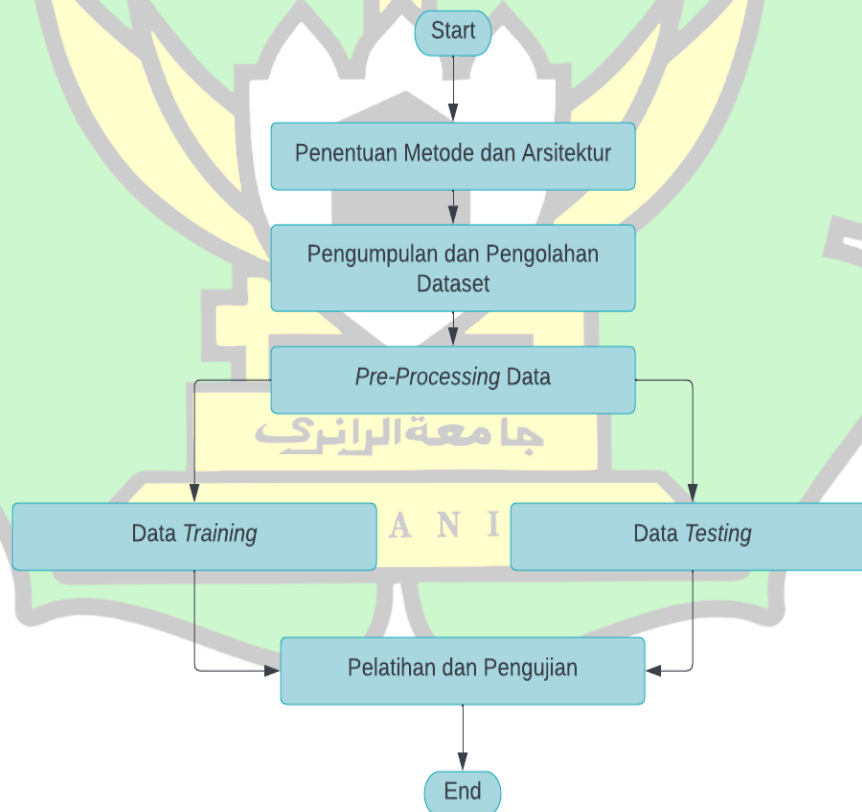
Metode simulasi merupakan skenario penelitian yang akan diimplementasikan pada sistem. Metode simulasi ini bertujuan untuk memberikan gambaran serta pengimplementasian metode yang akan digunakan. Adapun tahapan pada metode simulasi penelitian ini terdiri dari dua tahapan utama yaitu, perumusan konsep penelitian dan implementasi metode dan arsitektur.

3.4.1 Perumusan Konsep Penelitian

Konsep penelitian yang akan dilakukan penulis dirumuskan berdasarkan hasil dari identifikasi masalah dan studi literatur. Perumusan konsep penelitian merupakan tahapan penelitian yang dilakukan penulis untuk menentukan metode, arsitektur dan *tools* apa yang dapat diaplikasikan dalam studi kasus yang penulis akan teliti. Berdasarkan studi dan observasi yang dilakukan penulis, penelitian ini menggunakan metode RNN dengan model arsitektur LSTM untuk proses *text summarization*.

3.4.2 Implementasi Metode dan Arsitektur

Tahap implementasi metode dan arsitektur merupakan tahapan untuk mentransformasikan rumusan konsep penelitian menjadi implementasi sistem yang dibangun oleh penulis. Implementasi metode dan arsitektur terdiri dari alur sistem yang dikembangkan pada penelitian. Berikut ini penulis mengilustrasikan alur sistem proses *text summarization* *Matan Jauharah Tauhid*.



Gambar 3.2 Alur Sistem pada Tahapan Implementasi Metode dan Arsitektur

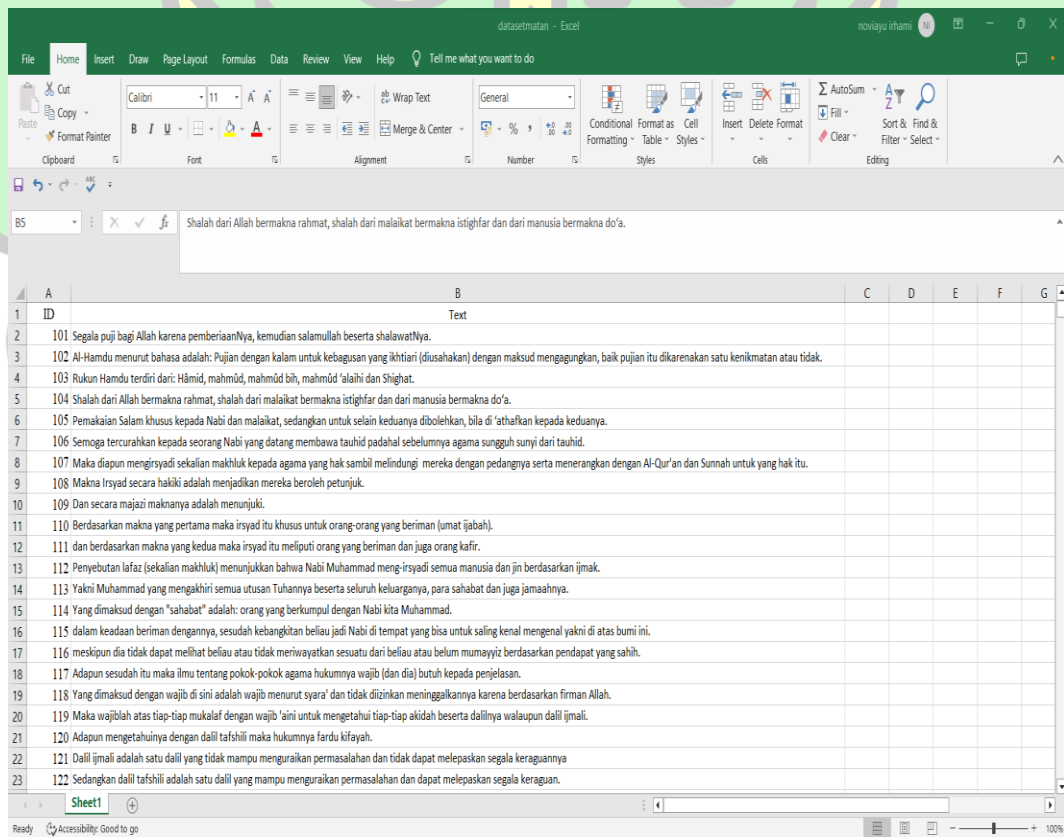
Rincian alur sistem pada tahapan implementasi adalah sebagai berikut:

1) Penentuan Metode dan Arsitektur

Alur sistem pada tahapan ini merupakan proses penentuan metode dan arsitektur dari *deep learning* yang akan digunakan pada penelitian. Setelah mempelajari masalah dan studi kasus penelitian, penulis memilih metode RNN dengan model arsitektur LSTM yang dapat digunakan untuk *text summarization*.

2) Pengumpulan dan Pengolahan Dataset

Dataset terdiri dari 1012 baris data yang diambil dari terjemahan *Matan Jauharah Tauhid*. Setiap baris data mewakili satu kalimat yang berasal dari bait matan dan setiap satu baris data dilengkapi dengan penjelasan yang terkait dengan baris data tersebut. Dataset matan terdiri dari teks asal terjemahan dan prediksi ringkasan. Dataset ini dikumpulkan dari syarah *Matan Jauharah Tauhid* (Tgh. Mujiburrahman, 2010) dan (Basri, 2015). Proses pengumpulan dataset menggunakan *Microsoft Office Excel*. Sedangkan untuk pengolahan dataset matan ini menggunakan bahasa pemrograman *python* dengan library *pandas*.

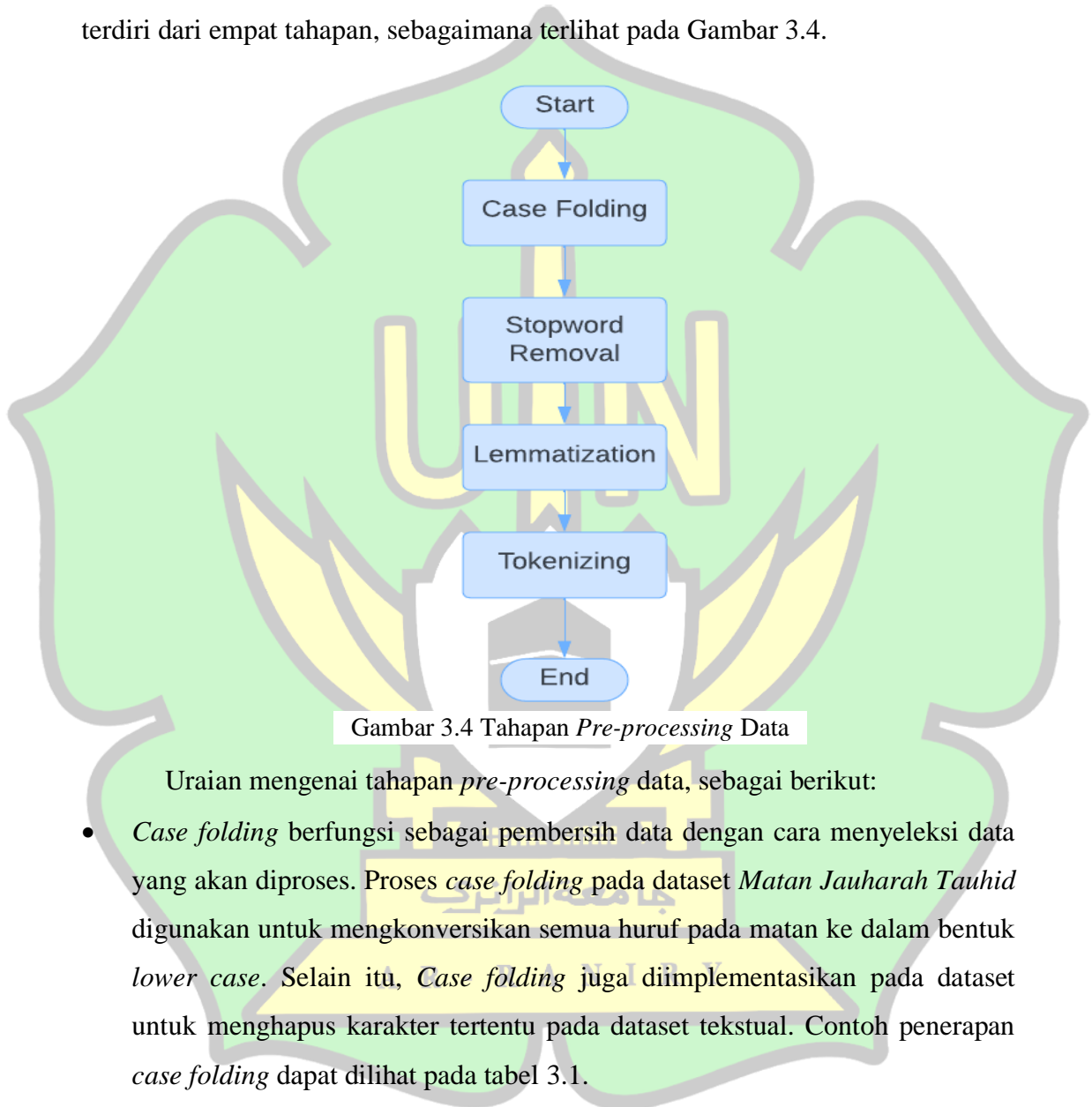


ID	Text
101	Segala puji bagi Allah karena pemberiannya, kemudian salamullah beserta shalawatNya.
102	Al-Hamdu menurut bahasa adalah: Pujian dengan kalam untuk kebagusan yang ikhtari (diusahakan) dengan maksud mengagungkan, baik pujian itu dikarenakan satu kenikmatan atau tidak.
103	Rukun Hamdu terdiri dari: Hämüd, mahmüd, mahmüd bih, mahmüd 'alâhi dan Shighat.
104	Shalah dari Allah bermakna rahmat, shalah dari malaikat bermakna istighfar dan dari manusia bermakna do'a.
105	Pemakaian Salam khusus kepada Nabi dan malaikat, sedangkan untuk selain keduanya dibolehkan, bila di 'athafkan kepada keduanya.
106	Semoga tercurahkan kepada seorang Nabi yang datang membawa tauhid padahal sebelumnya agama sungguh sunyi dari tauhid.
107	Maka diapun mengirsyadi sekalian makhluk kepada agama yang hak sambil melindungi mereka dengan pedangnya serta menerangkan dengan Al-Qur'an dan Sunnah untuk yang hak itu.
108	Makna Irsyad secara hakiki adalah menjadikan mereka beroleh petunjuk.
109	Dan secara majazi maknanya adalah menunjuki.
110	Berdasarkan makna yang pertama maka irsyad itu khusus untuk orang-orang yang beriman (umat ijabah).
111	dan berdasarkan makna yang kedua maka irsyad itu meliputi orang yang beriman dan juga orang kafir.
112	Penyebutan lafaz (sekalian makhluk) menunjukkan bahwa Nabi Muhammad meng-irsyadi semua manusia dan jin berdasarkan ijmak.
113	Yakni Muhammad yang mengakhiri semua utusan Tuhannya beserta seluruh keluarganya, para sahabat dan juga jamaahnya.
114	Yang dimaksud dengan "sahabat" adalah: orang yang berkumpul dengan Nabi kita Muhammad.
115	dalam keadaan beriman dengannya, sesudah kebangkitan beliau jadi Nabi di tempat yang bisa untuk saling kenal mengenal yakni di atas bumi ini.
116	meskipun dia tidak dapat melihat beliau atau tidak meriwayatkan sesuatu dari beliau atau belum mumayyiz berdasarkan pendapat yang sah.
117	Adapun sesudah itu maka ilmu tentang pokok-pokok agama hukumnya wajib (dan dia) butuh kepada penjelasan.
118	Yang dimaksud dengan wajib di sini adalah wajib menurut syara' dan tidak diizinkan meninggalkannya karena berdasarkan firman Allah.
119	Maka wajiblah atas tiap-tiap mukalaf dengan wajib 'aini untuk mengetahui tiap-tiap akidah beserta dalilnya walaupun dari ijmal.
120	Adapun mengetahuinya dengan dalil tafshili maka hukumnya fardu kifayah.
121	Dalil ijmal adalah satu dalil yang tidak mampu menguraikan permasalahan dan tidak dapat melepaskan segala keraguan
122	Sedangkan dalil tafshili adalah satu dalil yang mampu menguraikan permasalahan dan dapat melepaskan segala keraguan.

Gambar 3.3 Sampel Dataset *Matan Jauharah Tauhid*

3) *Pre-processing* Data

Pre-processing data adalah tahapan alur sistem untuk proses penyesuaian format dataset dengan sistem yang akan digunakan. *Pre-processing* data merupakan proses untuk menghilangkan permasalahan yang dapat mengganggu hasil dari pemrosesan dataset pada implementasi sistem. Tahapan *pre-processing* terdiri dari empat tahapan, sebagaimana terlihat pada Gambar 3.4.



Gambar 3.4 Tahapan *Pre-processing* Data

Uraian mengenai tahapan *pre-processing* data, sebagai berikut:

- *Case folding* berfungsi sebagai pembersih data dengan cara menyeleksi data yang akan diproses. Proses *case folding* pada dataset *Matan Jauharah Tauhid* digunakan untuk mengkonversikan semua huruf pada matan ke dalam bentuk *lower case*. Selain itu, *Case folding* juga diimplementasikan pada dataset untuk menghapus karakter tertentu pada dataset tekstual. Contoh penerapan *case folding* dapat dilihat pada tabel 3.1.

Tabel 3.1 *Case Folding* pada Dataset *Matan Jauharah Tauhid*

ID	text	summary	case_folding
101	Segala puji bagi Allah karena pemberiaannya, kemudian salamullah beserta shalawatNya.	Segala puji bagi Allah karena pemberiaannya, salamullah beserta shalawatNya.	segala puji bagi allah karena pemberiaannya kemudian salamullah beserta shalawatnya
102	Al-Hamdu menurut bahasa adalah: Pujian dengan kalam untuk kebagusan yang ikhtiar (diusahakan) dengan maksud mengagungkan, baik pujian itu dikarenakan satu kenikmatan atau tidak.	Al-Hamdu menurut bahasa adalah Pujian kalam untuk kebagusan yang ikhtiar (diusahakan) dengan maksud mengagungkan, pujian karena satu kenikmatan atau tidak.	al hamdu menurut bahasa adalah pujian dengan kalam untuk kebagusan yang ikhtiar diusahakan dengan maksud mengagungkan baik pujian itu dikarenakan satu kenikmatan atau tidak

- *Stopword removal* merupakan proses untuk melakukan penghapusan beberapa kata-kata yang dianggap kurang penting. *Stopword removal* ini akan menghapus frekuensi kata yang sering muncul, tetapi tidak memiliki arti yang signifikan pada teks. Kata tersebut seperti kata 'dan', 'yaitu', 'sebaliknya', 'begitu', 'lain', dan sebagainya (Ma'riah dkk., 2020).

Tabel 3.2 *Stopword removal* pada Dataset *Matan Jauharah Tauhid*

ID	text	summary	stopwords_text	st_join
101	Segala puji bagi Allah karena pemberiaannya, kemudian salamullah beserta shalawatNya.	Segala puji bagi Allah karena pemberiaannya, salamullah beserta shalawatNya.	['puji', 'allah', 'pemberiaannya', 'salamullah', 'beserta', 'shalawatnya']	puji allah pemberiaannya salamullah beserta shalawatnya
102	Al-Hamdu menurut bahasa adalah: Pujian dengan kalam untuk kebagusan yang ikhtiar (diusahakan) dengan maksud mengagungkan, baik pujian itu dikarenakan satu kenikmatan atau tidak.	Al-Hamdu menurut bahasa adalah Pujian kalam untuk kebagusan yang ikhtiar (diusahakan) dengan maksud mengagungkan, pujian karena satu kenikmatan atau tidak.	['al', 'hamdu', 'bahasa', 'pujian', 'kalam', 'kebagusan', 'ikhtiar', 'diusahakan', 'maksud', 'mengagungkan', 'pujian', 'kenikmatan']	al hamdu bahasa pujian kalam kebagusan ikhtiar diusahakan maksud mengagungkan pujian kenikmatan

- *Lemmatization* adalah subproses dari *pre-processing* data lanjutan setelah *stopword removal*. *Lemmatization* digunakan untuk menemukan kata dasar dari kata berimbuhan (Arief dkk., 2023). Proses *lemmatization* hanya akan merubah kata berimbuhan yang tidak bisa diproses pada model *sequence* dari RNN.

Tabel 3.3 *Lemmatization* pada Dataset *Matan Jauharah Tauhid*

ID	text	summary	lemmatization	lemmatized_join
101	Segala puji bagi Allah karena pemberiaannya, kemudian salamullah beserta shalawatNya.	Segala puji bagi Allah karena pemberiaannya, salamullah beserta shalawatNya.	['puji', 'allah', 'pemberiaannya', 'salamullah', 'beserta', 'shalawatnya']	puji allah pemberiaannya salamullah beserta shalawatnya
102	Al-Hamdu menurut bahasa adalah: Pujian dengan kalam untuk kebagusan yang ikhtiar (diusahakan) dengan maksud mengagungkan, baik pujian itu dikarenakan satu kenikmatan atau tidak.	Al-Hamdu menurut bahasa adalah Pujian kalam untuk kebagusan yang ikhtiar (diusahakan) dengan maksud mengagungkan, pujian karena satu kenikmatan atau tidak.	['al', 'hamdu', 'bahasa', 'pujian', 'kalam', 'kebagusan', 'ikhtiar', 'diusahakan', 'maksud', 'mengagungkan', 'pujian', 'kenikmatan']	al hamdu bahasa pujian kalam kebagusan ikhtiar diusahakan maksud mengagungkan pujian kenikmatan

- *Tokenizing* merupakan tahapan *pre-processing* data untuk melakukan pemenggalan pada setiap kata dalam kalimat. *Tokenizing* dibutuhkan untuk perulangan setiap kata pada dokumen saat pengimplementasian metode RNN.

Tabel 3.4 *Tokenizing* pada Dataset *Matan Jauharah Tauhid*

ID	text	summary	token
101	Segala puji bagi Allah karena pemberiaannya, kemudian salamullah beserta shalawatNya.	Segala puji bagi Allah karena pemberiaannya, salamullah beserta shalawatNya.	['segala', 'puji', 'bagi', 'allah', 'karena', 'pemberiaannya', 'kemudian', 'salamullah', 'beserta', 'shalawatnya']
102	Al-Hamdu menurut bahasa adalah: Pujian dengan kalam untuk kebagusan yang ikhtiar (diusahakan) dengan maksud mengagungkan, baik pujian itu dikarenakan satu kenikmatan atau tidak.	Al-Hamdu menurut bahasa adalah Pujian kalam untuk kebagusan yang ikhtiar (diusahakan) dengan maksud mengagungkan, pujian karena satu kenikmatan atau tidak.	['al', 'hamdu', 'menurut', 'bahasa', 'adalah', 'pujian', 'dengan', 'kalam', 'untuk', 'kebagusan', 'yang', 'ikhtiar', 'diusahakan', 'dengan', 'maksud', 'mengagungkan', 'baik', 'pujian', 'itu', 'dikarenakan', 'satu', 'kenikmatan', 'atau', 'tidak']

4) Pembagian Data *Training* dan *Testing*

Dataset yang sudah dilakukan *pre-processing* data akan dibagi menjadi dua, yaitu data *training* dan *testing*. Data *training* merupakan dataset yang digunakan untuk menghasilkan model sistem dari metode RNN dengan model arsitektur LSTM. Sedangkan data *testing* adalah data yang digunakan untuk menghasilkan *value* evaluasi.

Rasio pembagian data training dan testing yang digunakan pada penelitian ini adalah 80:20 untuk data training dan testing. Rasio untuk data *training* harus lebih besar daripada data *testing* untuk membangun model *text summarization* yang kompleks. Semakin banyak data training yang digunakan akan membantu meningkatkan akurasi hasil dari pelatihan metode RNN dengan model arsitektur LSTM.

Tabel 3.5 Pembagian Jumlah Data *Training* dan *Testing*

No.	Pembagian Data	Jumlah Data yang Dipakai
1.	Data <i>Training</i>	809
2.	Data <i>Testing</i>	203
3.	Total Dataset	1012

5) Pelatihan dan Pengujian

Proses pelatihan adalah proses pembuatan *pattern* dari metode RNN dengan model arsitektur LSTM. Tujuan alur sistem pada tahap ini untuk menciptakan sebuah metode yang efisien berdasarkan suatu fungsi yang diharapkan. Proses pengujian merupakan hasil metode yang sudah dilatih sebelumnya yang disimpan pada direktori untuk digunakan sebagai analisis kinerja metode. Proses pelatihan dan pengujian pada penelitian ini menggunakan metode RNN dengan model arsitektur LSTM.

Proses metode RNN dengan model arsitektur LSTM terdiri dari tiga *gate*, yaitu :

- *Input gate*

Input gate berfungsi sebagai pengontrol informasi yang disimpak pada sel LSTM. Berikut ini merupakan rumus dari *input gate* (Wiranda & Sadikin, 2019).

$$i_{t=\sigma}(W_i * S_{t-1} + W_i * X_t) \quad (3.1)$$

Keterangan:

I_t = *Input gate*.

W_i = Bobot *input gate*.

S_{t-1} = *State sebelumnya*.

X_t = *input pada saat waktu t*.

σ = *fungsi aktivasi sigmoid*.

Perhitungan untuk nilai variabel keterangan pada rumus *input gate* memerlukan informasi vektor dari setiap inputan yang dihasilkan oleh sistem yang telah dirancang. Jika nilai ini tidak diketahui, maka tidak dapat menghasilkan perhitungan manual yang akurat. Namun sebagai penggambaran dari perhitungan variabel dalam rumus *input gate*, penulis membuat asumsi terhadap nilai dari setiap variabel keterangan dari rumus tersebut.

Contoh perhitungan manual dari *input gate* berdasarkan persamaan 3.1, sebagai berikut:

Diketahui :

Teks inputan : namun ada dikatakan bahwa fitnah kubur adalah keraguan dalam menjawab.

Misalkan:

$$W_i = 0,5$$

$$S_{t-1} = 0,3$$

$$X_t = 0,8$$

Dari persamaan 3.1 dapat diuraikan dalam beberapa langkah perhitungan. Langkah – langkah perhitungannya sebagai berikut:

- $W_i * S_{t-1} = 0,5 * 0,3 = 0,15$
- $W_i * X_t = 0,5 * 0,8 = 0,4$
- $W_i * S_{t-1} + W_i * X_t = 0,15 + 0,4 = 0,55$
- $\sigma (W_i * S_{t-1} + W_i * X_t) = \sigma(0,55)$

- $\sigma(0,55) = 1/(1 + \exp(-0,55)) = 1/ (1+0,576949) = 0,634135$

Maka, nilai *input gate* (i_t) adalah 0,634.

- *Forget gate*

Forget gate berfungsi untuk mengontrol nilai tetap dalam sel memori. Rumus untuk *forget gate* sebagai berikut:

$$f_t = \sigma (W_f s_{t-1} + W_f x_t) \quad (3.2)$$

Keterangan:

F_t = *Forget gate*.

W_f = Bobot *forget gate*.

S_{t-1} = *State* sebelumnya.

X_t = *input* pada saat waktu t .

σ = *fungsi aktivasi sigmoid*.

Perhitungan untuk nilai variabel keterangan dari rumus *forget gate* sama halnya dengan perhitungan nilai variabel keterangan pada rumus *input gate*, yakni memerlukan informasi vektor dari setiap inputan yang dihasilkan oleh sistem. Berikut contoh perhitungan manual untuk *forget gate* dengan nilai variabel permisalan.

Diketahui :

Teks inputan : namun ada dikatakan bahwa fitnah kubur adalah keraguan dalam menjawab.

Misalkan:

$$W_f = 0,7$$

$$S_{t-1} = 0,6$$

$$X_t = 0,8$$

Langkah – langkah perhitungannya sebagai berikut:

- $W_f * S_{t-1} = 0,7 * 0,6 = 0,42$
- $W_f * X_t = 0,7 * 0,8 = 0,56$
- $W_f * S_{t-1} + W_f * X_t = 0,42 + 0,56 = 0,98$
- $\sigma (W_f * S_{t-1} + W_f * X_t) = \sigma(0,98)$
- $\sigma(0,55) = 1/(1 + \exp(-0,98)) = 1/ (1+0,373383) = 0,729416$

Maka, nilai *forget gate* (f_i) adalah 0,729.

- *Output gate*

Output gate digunakan untuk mengontrol banyaknya state yang melalui output.

Rumus menghitung *output gate*:

$$O_t = \sigma (W_o S_{t-1} + W_o X_t) \quad (3.3)$$

Keterangan:

O_t = *Output gate*

W_o = Bobot output gate.

S_{t-1} = *State sebelumnya*.

X_t = *input pada saat waktu t*.

σ = *fungsi aktivasi sigmoid*.

Perhitungan untuk nilai variabel dari keterangan pada rumus *output gate* sama dengan perhitungan nilai variabel dari keterangan pada rumus *input* dan *forget gate*.

Contoh perhitungan manual untuk *output gate*, sebagai berikut:

Teks inputan : namun ada dikatakan bahwa fitnah kubur adalah keraguan dalam menjawab.

Misalkan:

$$W_o = 0,9$$

$$S_{t-1} = 0,7$$

$$X_t = 0,8$$

Langkah – langkah perhitungannya sebagai berikut:

- $W_o * S_{t-1} = 0,9 * 0,7 = 0,63$
- $W_o * X_t = 0,9 * 0,8 = 0,72$
- $W_o * S_{t-1} + W_o * X_t = 0,63 + 0,72 = 1,35$
- $\sigma (W_o * S_{t-1} + W_o * X_t) = \sigma(1,35)$
- $\sigma(0,55) = 1/(1 + \exp(-1,35)) = 1/ (1+1,26) = 0,7937$

Maka, nilai *output gate* (o_t) adalah 0,793.

Metode RNN dengan model arsitektur LSTM dalam proses *text summarization* menggunakan beberapa *main layer* untuk rekonstruksi arsitektur pelatihan model. *Main layer* tersebut terdiri dari:

- *Embedding layer*

Embedding layer merupakan sebuah *layer* yang berfungsi dalam membentuk vektorisasi teks. Pengimplementasian *embedding layer* pada metode RNN dengan model arsitektur LSTM untuk mengkonversi teks yang akan dilakukan *summarization* menjadi representasi numerik. Representasi numerik digunakan untuk mempelajari *pattern* dari data pada teks inputan.

- *LSTM layer*

Layer ini digunakan untuk menginisiasi unit *layer* pada *sequence* yang ditetapkan untuk data pelatihan dan mengembalikan urutan *output* dari setiap *timestep* pada *layer* sebelumnya. *LSTM layer* juga bertugas untuk mengatur sekumpulan informasi yang akan diambil dari input baru dan menggabungkan dengan informasi yang telah ada pada sel sebelumnya.

- *Dense Layer*

Dense Layer adalah layer neural network dari metode RNN untuk mengubah *output* dari *LSTM layer* menjadi probabilitas pada kata yang akan muncul pada dataset (Arief dkk., 2023). Penggunaan *dense layer* berfungsi untuk mempelajari hubungan antara kata-kata dalam teks secara lebih kompleks, sehingga dapat menghasilkan *summarization* yang akurat.

3.5 Metode Analisis Hasil

Metode analisis hasil merupakan tahapan terakhir pada penelitian yang dikembangkan oleh penulis. Analisis hasil dilakukan dengan melakukan evaluasi terhadap sistem yang telah diterapkan penulis pada penelitian ini.

3.5.1 Evaluasi

Evaluasi merupakan tahapan penelitian yang sangat penting untuk menyelesaikan suatu penelitian. Penulis menggunakan dua metode evaluasi pengujian yaitu evaluasi pengujian oleh ahli dalam skala *likert* dan metode ROUGE. Evaluasi pengujian oleh ahli dalam skala *likert* bertujuan untuk memberikan penilaian yang dilakukan oleh ahli dalam skala *likert* terhadap hasil *text summarization* *Matan Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM. Sedangkan metode ROUGE merupakan metode evaluasi terhadap hasil *text summarization* *Matan Jauharah Tauhid* berbasis sistem dalam

bentuk matriks. ROUGE digunakan untuk analisis terhadap hasil sistem dengan cara membandingkan hasil ringkasan teks oleh sistem dengan ringkasan asli yang dibuat oleh ahli secara konvensional.

Hasil evaluasi dengan ROUGE akan menentukan tingkat akurasi yang dihasilkan dari implementasi metode dan arsitektur. Perhitungan ROUGE menggunakan nilai dari matriks *precision* (p), *recall* (r) dan *f1-score*. *Precision* digunakan untuk mengukur jumlah kalimat relevan dalam hasil ringkasan yang termasuk dalam teks asli. Sedangkan *recall* digunakan untuk menghitung jumlah kalimat relevan pada teks asli dan termasuk juga pada hasil ringkasan. *F1-score* digunakan untuk menghitung nilai gabungan dari *precision* dan *recall*. Dalam menghitung *precision*, *recall*, dan *f1-score*, diperlukan empat nilai yaitu:

- True positif (TP) merupakan jumlah pasangan frase yang dianggap relevan dengan hasil ringkasan manual dan ringkasan sistem.
- True negatif (TN), jumlah pasangan frase yang tidak relevan pada hasil ringkasan manual dan ringkasan sistem.
- False positif (FP) adalah jumlah pasangan frase yang relevan pada ringkasan sistem tetapi tidak relevan pada ringkasan manual.
- False negatif (FN) merupakan jumlah pasangan frase yang tidak relevan berdasarkan hasil ringkasan sistem tetapi relevan pada ringkasan manual.

Berikut contoh perhitungan *precision*, *recall*, dan *f1-score* pada perhitungan ROUGE secara manual.

Tabel 3.6 Sampel Teks Proses Perhitungan *Precision*, *Recall*, dan *F1-Score* pada Perhitungan ROUGE

Teks asli setelah proses <i>case folding</i>	: namun ada dikatakan bahwa fitnah kubur adalah talajjuj keraguan menjawab.
Teks hasil ringkasan sistem	: namun ada dikatakan bahwa fitnah kubur adalah keraguan dalam menjawab.

Tabel 3.7 Proses Perhitungan *Precision*, *Recall*, dan *F1-Score* pada Perhitungan ROUGE

TP = 9 , Namun Ada Dikatakan Kubur Adalah Keraguan Menjawab	TN = 0 , karena tidak ada kata yang memenuhi kriteria nilai TN.
FP = 1 , Dalam .	FN = 2 , Talajjuj Dalam

- *Precision (p)*

$$p = TP / (TP + FP) = 9 / (9+1) = 0,9 \quad (3.4)$$

- *Recall (r)*

$$r = TP / (TP + FN) = 9 / (9+2) = 0,81 \quad (3.5)$$

- *F1-score*

$$\begin{aligned} F1\text{-score} &= 2 * (p*r) / (p+r) \quad (3.6) \\ &= 2 * (0,9 * 0,81) / (0,9 + 0,81) \\ &= 2 * (0,729) / (1,71) \\ &= 2 * 0,426 = 0,852 \end{aligned}$$

Dengan demikian, nilai akurasi dari contoh perhitungan ROUGE di atas untuk *text summarization* adalah 85% yang diambil dari nilai *f1-score* = 0,852.

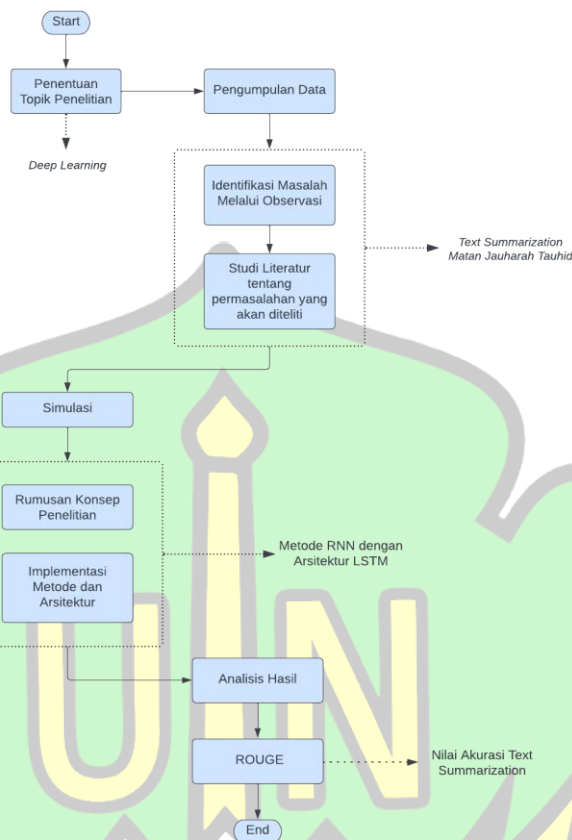
3.6 Alat Bantu Penelitian

Alat bantu pada penelitian ini adalah perangkat keras dan perangkat lunak komputer. Perangkat keras yang digunakan adalah satu unit laptop Asus TUF F15 FX506LU dengan spesifikasi sebagai berikut:

- *Processor* Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz 2.21 GHz.
- *RAM* 8,00 GB (7,84 GB usable).
- *Storage* 512 GB M.2 NVMe PCIe® 3.0 SSD.
- *Sistem operasi* Microsoft Windows 11 Home Single Language Version 22H2

3.7 Kerangka Pemikiran Penelitian

Kerangka pemikiran disebut juga dengan alur logika yang digunakan penulis pada suatu penelitian. Kerangka pemikiran menggambarkan alur logika yang digunakan penulis dalam mengembangkan penelitian. Kerangka pemikiran pada penelitian ini mencakup penentuan topik, runutan dan pengembangan sistem penelitian. Kerangka pemikiran yang digunakan penulis pada penelitian ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 Kerangka Pemikiran Penelitian

Uraian dari kerangka pemikiran pada penelitian ini adalah sebagai berikut:

- Penelitian dimulai dari penentuan topik yang dipilih penulis. Penulis memilih topik yang berkaitan dengan bidang pemanfaatan *deep learning*.
- Pengumpulan data penelitian berdasarkan dengan masalah yang diteliti, yaitu tentang *text summarization Matan Jauharah Tauhid*.
- Simulasi ini merupakan perumusan dan implementasi metode yang digunakan pada penelitian ini, yaitu metode RNN dengan model arsitektur LSTM.
- Proses analisis hasil untuk mendapatkan kesimpulan akhir dari penelitian dengan cara evaluasi menggunakan ROUGE untuk mengukur tingkat akurasi hasil ringkasan sistem.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil *Ground Truth Summarization*

Ground truth summarization merupakan hasil *summarization* teks dari ahli. Pada penelitian ini, penulis melibatkan seorang ahli yang bernama Tgk. Daryani, seorang guru pengajian di dayah Malikussaleh, Aceh Utara untuk membuat *summarization* terhadap 15 bait *Matan Jauharah Tauhid* yang akan dijadikan sebagai sampel *ground truth summarization*. Sampel ini akan menjadi acuan untuk mengevaluasi hasil *summarization* dari sistem dengan metode ROUGE. Sampel *summarization* *Matan Jauharah Tauhid* yang dilakukan oleh ahli dapat dilihat pada Tabel 4.1.

Tabel 4.1 *Ground Truth Summarization*

No.	Teks <i>Matan Jauharah Tauhid</i>	<i>Ground Truth Summarization</i>
1.	Azab kubur itu adalah bagi orang kafir, orang munafik dan orang-orang mukmin yang berbuat maksiat.	Azab kubur bagi orang kafir, orang munafik dan orang-orang mukmin yang bermaksiat.
2.	Kedua, nasakh sunnah dengan sunnah.	Kedua, nasakh sunnah dengan sunnah.
3.	Lauh adalah satu jisim yang bercahaya dimana Pena mencatat padanya,	Lauh adalah benda yang bercahaya yang menggunakan pena untuk mencatat,
4.	Yakni mereka yang ringan kemaksiatannya azab itu akan terputus dan mereka diazab sesuai derajat kemaksiatannya.	Jika mereka ringan maksiatnya, maka azab akan segera diputuskan, dan mereka diazab sesuai kadar kemaksiatannya.
5.	Dan dalam hal fana atau (kebinasaan) ruh ketika terjadi tiupan sangkakala adanya	fana ruh ketika ditiup sangkakala, namun ada perbedaan pendapat ulama tentang ini.

	perbedaan pendapat ulama.	
6.	Maka Bukanlah (pelantikan imam itu) sebagai satu rukun yang menjadi I'tikad dalam agama.	Pelantikan imam bukan sebagai rukun yang menjadi I'tikad dalam agama.
7.	Dan tujuh puluh orang pemuka mereka (kaum musyrikin) mati terbunuh,	tujuh puluh orang pemuka kaum musyrikin terbunuh,
8.	Karena pada jihat ulwi itu terdapat kemuliaan dan ketinggian martabat.	Pada jihat ulwi itu terdapat kemuliaan dan ketinggian martabat.
9.	Maksud dari Allah itu bersifat dengan wahdaniyah adalah bahwa Allah itu Esa, baik pada zat, sifat maupun af'al.	Allah bersifat dengan wahdaniyah maknanya bahwa Allah itu Esa, baik pada zat, sifat maupun 'afal.
10.	Dan sesuatu yang mengumpulkan makna dari apa-apa yang telah ditetapkan yaitu dua kalimah Syahadat yang menunjukkan kepada jalan keislaman, Dan jangan pernah ragu terhadap kalimah tersebut.	Sesuatu yang mengumpulkan makna tentang apa yang ditetapkan yaitu dua kalimah Syahadat yang menunjukkan kepada jalan keislaman, jangan ragu mengenai kalimah tersebut.
11.	diketika Israfil meniup terompet pada tiupan yang pertama.	Israfil meniup terompet pada tiupan yang pertama.
12.	Dan memelihara agama, kemudian jiwa, harta benda, nasab dan yang seumpamanya yakni akal dan kehormatan adalah hukumnya wajib.	memelihara agama, jiwa, harta benda, nasab dan seumpamanya termasuk akal dan kehormatan hukumnya wajib.
13.	dengan cara yang mendalam sampai-sampai menyibukkan	Dengan cara menyibukkan diri dalam menerangkan hakikatnya.

	diri menerangkan hakikatnya.	
14.	Ahlu perang uhud yakni orang-orang yang terlibat dalam perang uhud,	Ahlu perang uhud adalah orang-orang yang terlibat dalam perang uhud,
	Untuk membela utsman ketika ada informasi bahwa beliau dibunuh oleh kaum kafir Quraisy.	Membela utsman ketika ada informasi bahwa beliau dibunuh oleh kaum kafir Quraisy.

4.2 Hasil Pengujian Terhadap Implementasi Metode dan Arsitektur

Bagian ini penulis akan menjelaskan mengenai hasil dari implementasi metode RNN dengan model arsitektur LSTM untuk *text summarization* *Matan Jauharah Tauhid*. Pada bagian ini akan dijabarkan bagaimana hasil implementasi metode dan arsitektur yang telah dibangun dengan menerapkan lima skenario percobaan. Terdapat empat skenario yang dibedakan berdasarkan perbedaan variasi pada tahapan *pre-processing* data. Dalam penelitian ini juga dilakukan pengujian yang melibatkan sebuah *tools* yang sangat fenomenal saat ini, yaitu *ChatGPT*. Dikarenakan kemampuan *ChatGPT* untuk melakukan berbagai tugas, penulis menggunakan *ChatGPT* untuk melakukan *text summarization* terhadap sampel *Matan Jauharah Tauhid* sebagai pembanding dengan hasil *text summarization* dengan metode RNN dengan model arsitektur LSTM. Penjabaran lima skenario percobaan ini dapat dilihat pada Tabel 4.2.

Tabel 4.2 Pembagian Skenario Percobaan pada Pengujian Terhadap Implementasi Metode dan Arsitektur

No.	Skenario	Keterangan
1.	Skenario 1	<i>Text summarization</i> metode RNN dengan model arsitektur LSTM menggunakan teks asal (belum dilakukan <i>pre-processing</i> data).

2.	Skenario 2	<i>Text summarization</i> metode RNN dengan model arsitektur LSTM menggunakan teks hasil <i>case folding</i> .
3.	Skenario 3	<i>Text summarization</i> metode RNN dengan model arsitektur LSTM menggunakan teks hasil <i>stopword removal</i> .
4.	Skenario 4	<i>Text summarization</i> metode RNN dengan model arsitektur LSTM menggunakan teks hasil dari <i>lemmatization</i> dan <i>stopword removal</i> .
5.	Skenario 5	<i>Text summarization</i> menggunakan <i>ChatGPT</i> .

Tujuan dilakukan pembagian skenario percobaan untuk membandingkan pengaruh dari tahapan *pre-processing* data terhadap akurasi hasil *text summarization* yang dihasilkan. Hasil dari kelima skenario percobaan akan dievaluasi menggunakan ROUGE dengan menggunakan perhitungan *recall*, *precision*, dan *f1-score*. Hasil dari skenario percobaan akan dibandingkan dengan melihat nilai akurasi konstan pada *train model*, nilai maksimum *f1-score* dan nilai rata-rata *f1-score*. Nilai-nilai tersebut merupakan representatif dari pengujian implementasi metode dan arsitektur dan keseluruhan data uji.

Semua skenario percobaan implementasi metode RNN dan model arsitektur LSTM menggunakan *platform text editor*, bahasa pemrograman, library dan perangkat keras yang sama. Selain itu, kelima skenario dilatih dengan variabel dan parameter pelatihan yang sama, sehingga tidak terdapat perbedaan *train model* pengujian untuk semua skenario. Berikut merupakan variabel dan parameter yang digunakan untuk pengujian kelima skenario tersebut.

$$1. \text{Vocab_size} = \text{len}(\text{tokenizer.word_index}) + 1$$

Vocab_size adalah sebuah variabel yang menyimpan jumlah keseluruhan kata yang ada di dalam kamus kata. parameter dari variabel ini diambil dari *tokenizer.word_index* yang berisi jumlah kata unik dalam teks.

$$2. \text{Max_seq_len} = \max(\text{len}(\text{seq}) \text{ for } \text{seq} \text{ in } \text{X_train})$$

Max_seq_len merupakan variabel penampung nilai maksimum dari *sequence* yang terdapat pada data *training*.

3. *Embedding_layer = embedding(input_dim=vocab_size, output_dim=512)*

Embedding_layer adalah variabel yang digunakan untuk menyimpan objek *layer embedding*. *Layer embedding* merupakan *layer* yang digunakan untuk memetakan setiap kata pada suatu teks ke dalam bentuk vektor. *Input_dim* adalah parameter untuk menentukan jumlah kata pada kamus yang digunakan untuk memetakan kata menjadi vektor. *Output_dim* adalah parameter yang menentukan jumlah dimensi pada vektor yang merepresentasikan kata yang terdapat dalam teks. Banyaknya informasi yang dapat direpresentasikan vektor tergantung kepada besar-kecilnya nilai *output_dim*. Pengujian ini menggunakan nilai parameter *output_dim* sebesar 512.

4. *LSTM_layer = 512*

LSTM_layer merupakan variabel yang menampung jumlah unit pada *layer LSTM*. *Layer LSTM* pada percobaan ini digunakan untuk mengatasi masalah mengenai *vanishing gradient*. *Vanishing Gradient* adalah *error function* yang dihitung pada *layer* awal menjadi sangat kecil sampai mendekati nol, yang mengakibatkan optimasi pada *train model* menjadi lambat. Nilai parameter *LSTM_layer* adalah 512.

5. *Activation = softmax*

Activation softmax merupakan sebuah *function* yang digunakan untuk mengambil nilai dari input vektor dan menghasilkan output vektor yang berukuran sama.

6. *Epoch = 50*

Epoch merupakan iterasi *train model* terhadap keseluruhan data *training*. Pada pengujian ini *epoch* dilakukan maksimal sebanyak 50 kali. Namun pada iterasi *train model*, *epoch* ini dapat berhenti sebelum mencapai *epoch* 50, dikarenakan pemakaian parameter *callbacks = [early_stopping]* yang akan menghentikan proses *train model* saat mencapai nilai akurasi tertinggi. Penghentian iterasi *epoch* ini agar tidak mengakibatkan *overfitting* pada proses *train model*.

7. *Batch_size = 32*

Batch_size adalah variabel untuk menentukan jumlah sampel data yang di proses pada setiap *batch* saat *train model*. Pada pengujian ini, proses *train*

model sebesar 32. Dengan kata lain, setiap iterasi *train* model akan melakukan proses terhadap 32 sampel data.

8. *Validation_split* = 0,2

Validation_split merupakan variabel untuk menetapkan persentase data *training* yang digunakan sebagai data validasi. Pengujian ini menggunakan 20% dari data *training* untuk dijadikan data validasi.

9. *Optimizer* = adam

Optimizer adam adalah variabel untuk mengatur jenis *optimizer* yang digunakan pada proses *train model*. *Optimizer* merupakan algoritma untuk memperbarui parameter model selama *train model* untuk mengoptimalkan kinerja model. Parameter *optimizer* yang dipakai pada pengujian ini adalah adam. Adam merupakan singkatan dari *adaptive moment estimation*. Adam berfungsi dalam optimalisasi laju *train model*.

10. *Loss* = *sparse_categorical_crossentropy*

Variabel ini digunakan untuk menentukan jenis loss function pada *train model*. *Sparse_categorical_crossentropy* berfungsi untuk mengukur prediksi model.

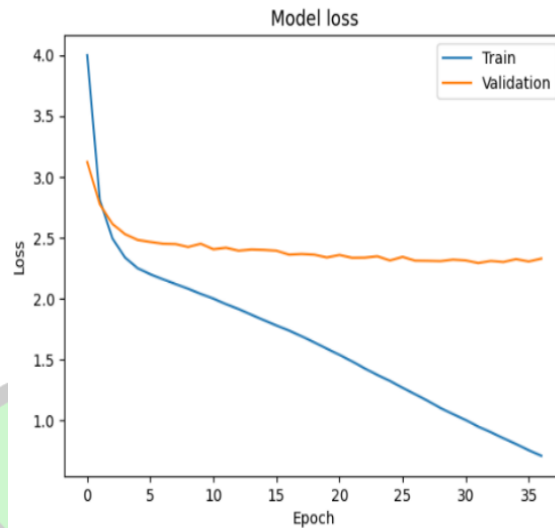
11. *Callbacks* = [*early_stopping*]

Callbacks adalah variabel untuk menetapkan jenis *callbacks* yang digunakan dalam *train model*. *Early_stopping* merupakan jenis *callbacks* untuk analisa performa model selama proses *training*.

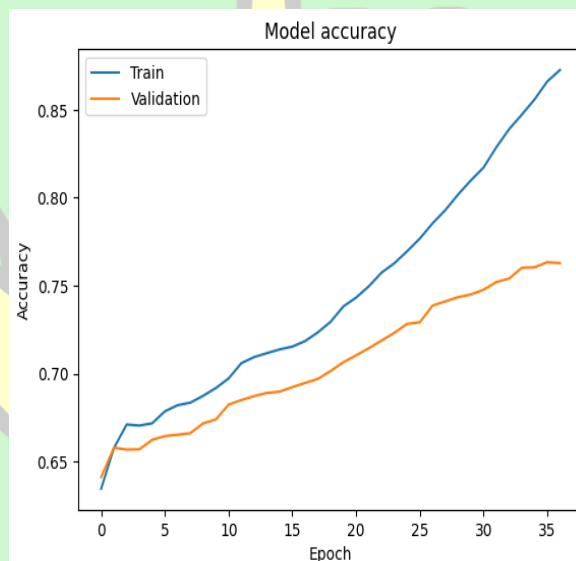
Hasil finalisasi dari proses *text summarization* pada semua skenario percobaan dengan implementasi metode RNN dengan model arsitektur LSTM akan ditampilkan dalam bentuk tabel. Tabel hasil ini terdiri dari 3 kolom yaitu, kolom *text* yang menampilkan teks yang akan dijadikan objek *train model*, kolom *predict_summary* untuk menampilkan hasil prediksi teks yang akan diringkas, dan kolom *true_summary* untuk menampilkan hasil akhir dari proses *text summarization* dengan implementasi RNN dan arsitektur LSTM.

4.2.1 Hasil Pengujian Skenario 1

Skenario pertama ini adalah skenario *text summarization* dengan implementasi metode RNN dengan model arsitektur LSTM menggunakan teks asal (belum dilakukan *pre-processing* data). Hasil pengujian untuk skenario 1 dapat dilihat pada Gambar 4.1 dan 4.2.



Gambar 4.1 Grafik *Loss Function* pada Skenario 1



Gambar 4.2 Grafik *Accuracy Function* pada Skenario 1

Gambar 4.1 menunjukkan bahwa selama iterasi *train model*, nilai loss untuk data *training* dan data *testing* semakin berkurang. Nilai loss tertinggi dari data *training* pada skenario 1 sebesar 3,99 berhasil turun menjadi 0,70. Sedangkan nilai loss tertinggi dari data validasi sebesar 3,12 berhasil turun menjadi 2,32. Pada Gambar 4.2 grafik *accuracy function* pada skenario 1, dapat diketahui bahwa seiring berjalannya proses pelatihan nilai akurasi terhadap data *training* dan data validasi semakin tinggi. Nilai akurasi *train model* dari data *training* dan validasi berkisar 0,87 dan 0,76. Dengan kata lain, data *training* yang berhasil di latih yaitu sebesar 87% dan data validasi 76 %.

Setelah proses *train model* akan dilakukan pengujian terhadap data testing untuk menghasilkan *summarization*. Hasil *summarization* akan dievaluasi dengan perhitungan ROUGE. Perhitungan ROUGE ini akan menghasilkan tiga matriks ROUGE yang terdiri dari nilai *precision*, *recall*, dan *f1-score*. Ketiga ROUGE tersebut yaitu ROUGE-1, ROUGE-2, dan ROUGE-L. Nilai akurasi hasil *summarization* dilihat dari nilai maksimum *f1-score* dan nilai rata-rata *f1-score*. Nilai maksimum *f1-score* dan nilai rata-rata *f1-score* merupakan representatif dari pengujian implementasi metode dan model arsitektur pada keseluruhan data uji.

Tabel 4.3 Hasil ROUGE Skenario 1

No.	Skenario 1	ROUGE (Average)	ROUGE (Maximum)
1.	<i>Precision</i>	0,5021	0,6008
2.	<i>Recall</i>	0,4565	0,5386
3.	<i>F1-score</i>	0,4173	0,5612

Berdasarkan hasil akurasi ROUGE pada Tabel 4.3, diketahui bahwa skenario 1 mendapatkan hasil nilai maksimum *f1-score* sebesar 0,5612 yang berarti nilai akurasi tertinggi dari *summarization* pada skenario 1 sebesar 56%. Nilai akurasi tersebut dapat dikategorikan cukup. Namun untuk mendapatkan hasil akurasi yang lebih baik untuk *text summarization*, nilai *f1-score* ini dapat ditingkatkan dengan menggunakan data *training* yang telah melalui *pre-processing* data.

Hasil dari proses *text summarization* ini terkadang menghasilkan teks *summary* yang sama dengan teks asal. Hal ini disebabkan adanya kompleksitas terhadap makna pada matan tersebut sehingga jika dilakukan *summarization* akan menghilangkan makna dari teks asal.

Tabel 4.4 Sampel Hasil Skenario 1 yang Tidak Bisa Dilakukan *Summarization*

No.	Teks Asal	Ground Truth Summarization	Predict Summary	True Summary
1.	Azab kubur itu adalah bagi orang kafir, orang	Azab kubur itu bagi orang kafir, orang	dan adalah adalah adalah bagi orang kafir	azab kubur adalah bagi orang kafir,

	munafik dan orang-orang mukmin yang berbuat maksiat.	munafik dan orang-orang mukmin yang bermaksiat.	orang orang	orang munafik, dan orang-orang mukmin yang berbuat maksiat
2.	Kedua, Nasakh Sunnah dengan Sunnah.	Kedua, nasakh sunnah dengan sunnah.	barangsiapa nasakh dengan dan	kedua, nasakh sunnah dengan sunnah
3.	Lauh adalah satu jisim yang bercahaya dimana Pena mencatat padanya,	Lauh adalah benda yang bercahaya yang menggunakan pena untuk mencatat,	hamdu adalah satu jisim yang diambil sebagaimana suka pada keduanya	lauh adalah satu jisim yang bercahaya dimana pena mencatat padanya

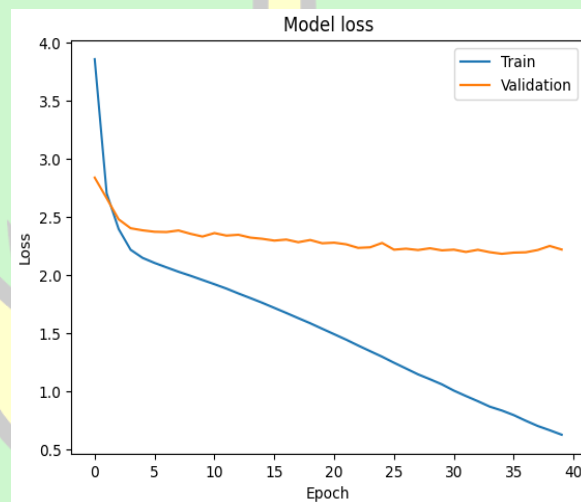
Tabel 4.5 Sampel Hasil *Summarization* pada Skenario 1

No.	Teks Asal	<i>Ground Truth Summarization</i>	<i>Predict Summary</i>	<i>True Summary</i>
1.	yakni mereka yang ringan kemaksiatannya azab itu akan terputus dan mereka diazab sesuai derajat kemaksiatannya.	Jika mereka ringan maksiatnya, maka azab akan putus dan mereka diazab sesuai kemaksiatannya.	dan mereka yang adil seperti adalah akan dan mereka dengan bidang	yakni mereka yang ringan azab akan dan mereka sesuai derajat
2.	Dan dalam hal fana atau (kebinasaan) ruh ketika terjadi tiupan sangkakala adanya perbedaan pendapat ulama.	Dan fana ruh itu ketika ditiup sangkakala, namun ada perbedaan pendapat ulama tentang ini.	dan dalam hal diterimanya atau dirinya jahat pendapat	dan dalam hal fana atau kebinasaan ruh ketika terjadi tiupan adanya perbedaan pendapat ulama
3.	Maka Bukanlah (pelantikan imam itu) sebagai satu rukun yang menjadi I'tikad dalam agama.	Pelantikan imam bukan sebagai rukun yang menjadi I'tikad dalam agama.	maka akan yang adalah sebagai satu nabi yang kembali agama dalam dengan	maka bukanlah imam sebagai satu rukun yang menjadi i'tikad dalam agama

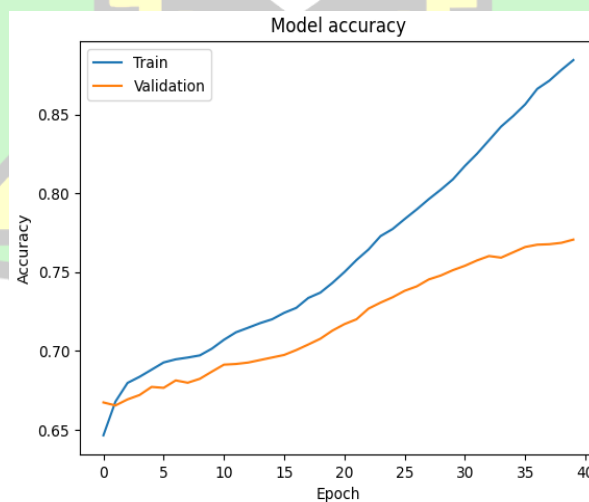
Tabel 4.4 dan 4.5 terdiri dari tiga kolom, yaitu kolom teks asal berisi teks asli matan, kolom *ground truth summarization* berisi ringkasan yang dibuat oleh ahli, kolom *predict summary* berisi prediksi ringkasan dari proses *train model* awal (masih dalam kondisi awal pelatihan model pada data uji), dan kolom *true summary* berisi hasil *summarization* dari keseluruhan hasil iterasi *train model* pada data uji.

4.2.2 Hasil Pengujian Skenario 2

Skenario 2 merupakan hasil *text summarization* dengan Metode RNN dengan model arsitektur LSTM menggunakan teks hasil *case folding*. Berikut ini merupakan hasil pengujian untuk skenario 2.



Gambar 4.3 Grafik *Loss Function* pada Skenario 2



Gambar 4.4 Grafik *Accuracy Function* pada Skenario 2

Grafik *loss accuracy* pada Gambar 4.3 menunjukkan bahwa selama iterasi *train model*, nilai *loss* untuk data *training* dan data *testing* semakin menurun. Hal ini menunjukkan bahwa proses *train model* dapat memahami konteks data *training* yang digunakan. Rentang nilai *loss* dari data *training* pada skenario 2 berkisar antara 3,86-0,62. Sedangkan nilai *loss* dari data validasi rentang nilai antara 2,84-2,27. Pada grafik *accuracy function* skenario 2, diketahui bahwa nilai akurasi dari data *training* dan validasi berkisar 0,88 dan 0,76.

Tabel 4.6 Hasil ROUGE Skenario 2

No.	Skenario 2	ROUGE (Average)	ROUGE (Maximum)
1.	<i>Recall</i>	0,4960	0,5854
2.	<i>Precision</i>	0,5313	0,3490
3.	<i>F1-score</i>	0,5075	0,5882

Hasil akurasi ROUGE pada Tabel 4.6, diketahui bahwa skenario ini mendapatkan hasil rata-rata *f1-score* sebesar 0,5075 dan nilai maksimum *f1-score* sebesar 0,5882. Dengan kata lain, nilai akurasi ROUGE sebesar 58%. Nilai *f1-score* tersebut dapat dikategorikan cukup baik. Namun hasil akurasi ini dapat ditingkatkan dengan menggunakan data *training* dengan tahapan selanjutnya dari *pre-processing* data.

Tabel 4.7 Sampel Hasil Skenario 2 yang Tidak bisa Dilakukan Summarization

No.	<i>Text Case Folding</i>	<i>Ground Truth Summarization</i>	<i>Predict Summary</i>	<i>True Summary</i>
1.	Dan tidak ada perbedaan antara Ahlu Badar yang mati syahid	Tidak ada beda ahlu badar dan yang mati syahid	dan tidak ada dengan antara pelaku sifat yang mati syahid	dan tidak ada perbedaan antara ahlu badar yang mati syahid
2.	Kedua, Nasakh Sunnah dengan Sunnah.	Kedua, nasakh sunnah dengan sunnah.	barangsiapa nasakh dengan dan	kedua, nasakh sunnah dengan sunnah

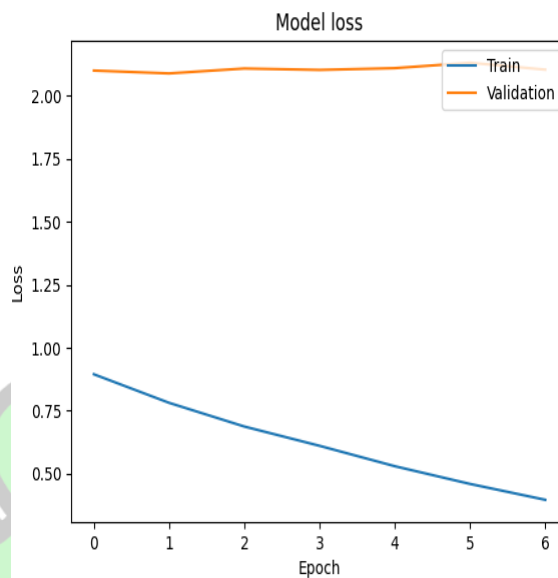
3.	Lauh adalah satu jisim yang bercahaya dimana Pena mencatat padanya,	Lauh adalah benda yang bercahaya yang menggunakan pena untuk mencatat,	hamdu adalah satu jisim yang diambil sebagaimana suka pada keduanya	lauh adalah satu jisim yang bercahaya dimana pena mencatat padanya
----	---	--	---	--

Tabel 4.8 Sampel Hasil Summarization pada Skenario 2

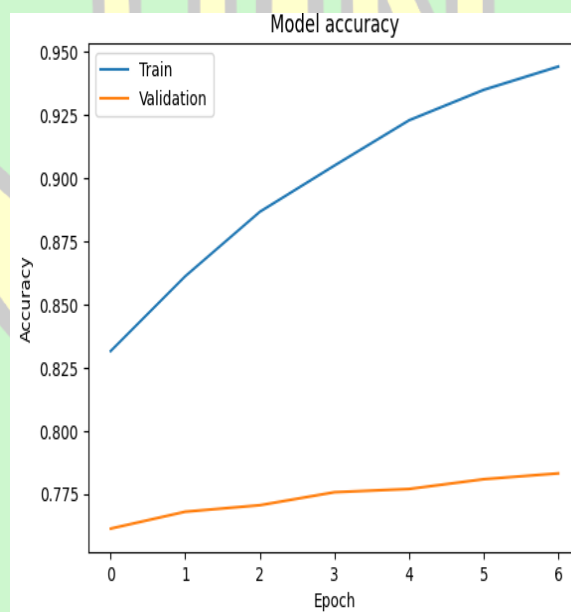
No.	<i>Text Case Folding</i>	<i>Ground Truth Summarization</i>	<i>Predict Summary</i>	<i>True Summary</i>
1.	dan tujuh puluh orang pemuka mereka (kaum musyrikin) mati terbunuh,	tujuh puluh orang pemuka kaum musyrikin terbunuh,	dan mi'raj siapa orang mereka kaum mati diinginkannya	dan tujuh puluh orang mereka kaum mati terbunuh
2.	karena pada jihat ulwi itu terdapat kemuliaan dan ketinggian martabat.	Pada jihat ulwi itu terdapat kemuliaan dan ketinggian martabat.	karena pada tentang nabi adalah terdapat dilakukan dan nabi	karena pada jihat ulwi terdapat kemuliaan dan martabat
3.	Maksud dari Allah itu bersifat dengan wahdaniyah adalah bahwa Allah itu Esa, baik pada zat, sifat maupun af'al.	Allah bersifat dengan wahdaniyah bermaksud bahwa Allah itu Esa, baik pada zat, sifat maupun 'afal.	maksud setiap allah adalah sesudah dengan qadar adalah bahwa allah adalah melakukan baik pada zat sifat dan	maksud allah bersifat dengan wahdaniyah adalah bahwa allah esa baik pada zat sifat maupun

4.2.3 Hasil Pengujian Skenario 3

Skenario ketiga ini merupakan skenario *text summarization* dengan implementasi metode RNN dengan model arsitektur LSTM menggunakan dataset yang sudah dilakukan *pre-processing* data dengan tahapan *stopword removal*. Hasil pengujian untuk skenario 3, sebagaimana terlihat pada Gambar 4.5 dan Gambar 4.6.



Gambar 4.5 Grafik *Loss Function* pada Skenario 3



Gambar 4.6 Grafik *Accuracy Function* pada Skenario 3

Gambar 4.5 menunjukkan bahwa selama iterasi *train model*, nilai *loss* untuk data *training* dan data validasi tidak berkurang secara konstan. Nilai *loss* data *training* pada skenario 3 ini dari 0,89 turun menjadi 0,39. Sedangkan nilai *loss* data validasi berkisar di rentang nilai 2,10-2,13. Pada Gambar 4.6 grafik *accuracy function* pada Skenario 3, dapat diketahui bahwa nilai akurasi dari data *training* dan validasi berkisar 83-94%.

Tabel 4.9 Hasil ROUGE Skenario 3

No.	Skenario 3	ROUGE (Average)	ROUGE (Maximum)
1.	<i>Recall</i>	0,648	0,75
2.	<i>Precision</i>	0,548	0,6
3.	<i>F1-score</i>	0,592	0,666

Hasil akurasi ROUGE pada Tabel 4.9, diketahui bahwa skenario ini mendapatkan hasil rata-rata *f1-score* sebesar 0,592 dan nilai maksimum *f1-score* sebesar 0,666. Dengan kata lain, nilai akurasi ROUGE sebesar 66%. Nilai *f1-score* tersebut dapat dikategorikan cukup baik.

Tabel 4.10 Sampel Hasil Skenario 3 yang Tidak bisa Dilakukan *Summarization*

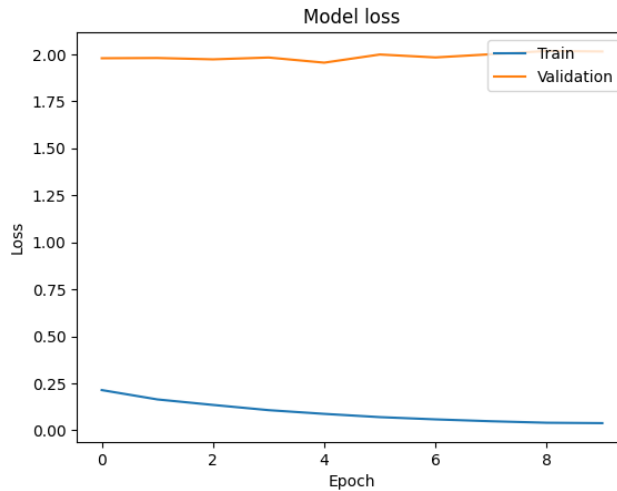
No.	<i>Text Stopword Removal</i>	<i>Ground Truth Summarization</i>	<i>Predict Summary</i>	<i>True Summary</i>
1.	mengumpulkan makna ditetapkan kalimah syahadat jalan keislaman ragu kalimah	Sesuatu yang mengumpulkan makna tentang apa yang ditetapkan yaitu dua kalimah Syahadat yang menunjukkan kepada jalan keislaman, jangan ragu mengenai kalimah tersebut.	orang makna irsyad hafizhun maujud maujud	mengumpulkan makna ditetapkan kalimah syahadat jalan keislaman ragu kalimah
2.	diketika israfil meniup terompet tiupan	Israfil meniup terompet pada tiupan yang pertama.	diketika istikhlaf menimpa tingkatan kebaikan	diketika israfil meniup terompet tiupan
3.	memelihara agama jiwa harta benda nasab seumpamanya akal kehormatan hukumnya wajib	memelihara agama, jiwa, harta benda, nasab dan seumpamanya yakni akal dan kehormatan hukumnya wajib.	diriwayatkan selainnya syahadat manusia dibarengi lauh manusia lauh mahfudh makna wajib	memelihara agama jiwa harta benda nasab seumpamanya akal kehormatan hukumnya wajib

Tabel 4.11 Sampel Hasil *Summarization* pada Skenario 3

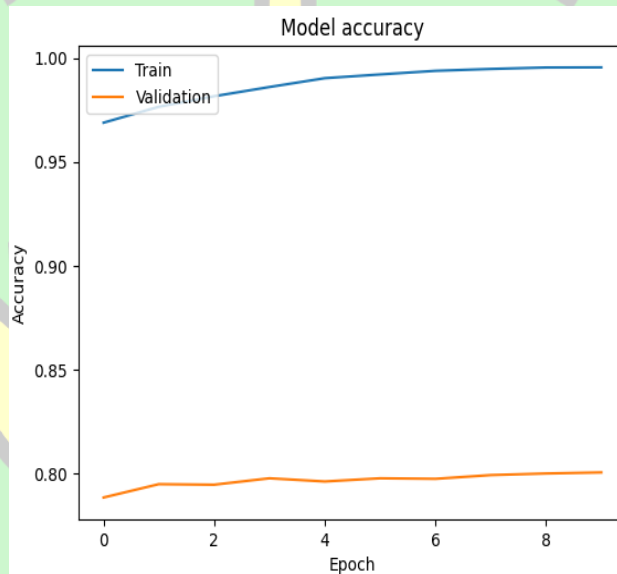
No.	<i>Text Stopword Removal</i>	<i>Ground Truth Summarization</i>	<i>Predict Summary</i>	<i>True Summary</i>
1.	ringan kemaksiatannya azab terputus diazab sesuai derajat kemaksiatannya	ringan azab maksiatnya sesuai derajat kemaksiatannya.	dalil azab azab neraka	ringan azab sesuai derajat
2.	kebenaran maksudnya agama lurus agama islam menaati perintahnya menjauhi larangannya	Kebenaran yang dimaksud adalah agama Islam dengan menaati segala perintahnya dan menjauhi segala larangannya.	kebenaran maksud azab hukumnya maujud imani	kebenaran maksud agama islam perintahnya menjauhi larangannya
3.	mendalam menyibukkan menerangkan hakikatnya	Dengan cara menyibukkan diri dalam menerangkan hakikatnya.	keenam zaman hakikatnya	mendalam menerangkan hakikatnya

4.2.4 Hasil Pengujian Skenario 4

Skenario keempat ini adalah skenario *text summarization* dengan menggunakan dataset yang sudah dilakukan *pre-processing* data dengan variasi tahapan *lemmatization* dan *stopword removal*. Pada metode RNN dengan model arsitektur LSTM, *lemmatization* beroperasi dengan cara mengolah data hasil *stopword removal*. *Lemmatization* akan mengekstrak kata dasar dari kata berimbuhan yang telah di olah sebelumnya pada tahapan *stopword removal*. Tahapan *lemmatization* ini sangat bergantung terhadap hasil dari tahapan *pre-processing* data sebelumnya. *Lemmatization* ini merupakan tahapan *pre-processing* yang sangat berperan penting dalam proses pengolahan bahasa alami atau NLP. Berikut ini merupakan hasil pengujian skenario 4 dengan tahapan *lemmatization* dan *stopword removal*.



Gambar 4.7 Grafik *Loss Function* pada Skenario 4



Gambar 4.8 Grafik *Accuracy Function* pada Skenario

Grafik *loss accuracy* menunjukkan rentang nilai *loss* dari data *training* pada skenario 4 berkisar antara 0,21 – 0,03. Nilai *loss* dari data *training* yang kecil menandakan bahwa *train model* pada data *training* yang menggabungkan proses *stopword removal* dan *lemmatization* ini menghasilkan pelatihan yang sangat baik. Nilai *loss* dari data validasi pada skenario ini juga berkisar 1,97 – 2,01. Pada grafik *accuracy function* pada skenario 4, diketahui bahwa nilai akurasi dari data *training* berada di rentang nilai 0,96 -0,99 dan data validasi 0,80. Nilai akurasi dari *train model* pada data *training* dan data validasi pada skenario ini merupakan nilai tertinggi daripada ketiga skenario sebelumnya.

Tabel 4.12 Hasil ROUGE Skenario 4

No.	Skenario 4	ROUGE (Average)	ROUGE (Maximum)
1.	<i>Recall</i>	0,646	0,75
2.	<i>Precision</i>	0,6106	0,6666
3.	<i>F1-score</i>	0,6273	0,7058

Hasil akurasi ROUGE pada Tabel 4.12, diketahui bahwa skenario ini mendapatkan hasil rata-rata *f1-score* sebesar 0,6273 dan nilai maksimum *f1-score* sebesar 0,7058. Dengan kata lain, nilai akurasi ROUGE sebesar 70%. Nilai *f1-score* tersebut dapat dikategorikan baik.

Tabel 4.13 Sampel Hasil Skenario 4 yang Tidak bisa dilakukan *Summarization*

No.	<i>Text Stopword Removal & Lemmatization</i>	<i>Ground Truth Summarization</i>	<i>Predict Summary</i>	<i>True Summary</i>
1.	mendalam menyibukkan menerangkan hakikatnya	Dengan cara menyibukkan diri dalam menerangkan hakikatnya.	keenam zaman hakikatnya	mendalam menerangkan hakikatnya
2.	ahlu perang uhud orang terlibat dalam perang uhud	Ahlu perang uhud adalah orang-orang yang terlibat dalam perang uhud,	ahlu perang uhud orang orang terlibat perang badar	ahlu perang uhud orang terlibat perang uhud
3.	wajib hukumnya umat islam mengangkat imam adil	Wajib hukumnya bagi umat islam untuk mengangkat imam atau pemimpin yang adil	wajib wajib wajib islam hadist	wajib hukumnya umat islam mengangkat imam adil

Tabel 4.14 Sampel Hasil *Summarization* pada Skenario 4

No.	<i>Text Stopword Removal & Lemmatization</i>	<i>Ground Truth Summarization</i>	<i>Predict Summary</i>	<i>True Summary</i>
1.	membela utsman informasi beliau dibunuh kaum kafir quraisy	Membela utsman ketika ada informasi bahwa beliau dibunuh oleh kaum kafir Quraisy.	utsman beliau turun kaum ulama	utsman beliau dibunuh kaum kafir
2.	jihat ulwi kemuliaan ketinggian martabat	Pada jihat ulwi terdapat kemuliaan dan ketinggian martabat.	diambil mengganti kemuliaan kebaikan	jihat ulwi kemuliaan martabat
3.	orang serius menekuni perkara tujuannya	orang yang serius menekuni satu perkara pada akhirnya sampai kepada tujuannya.	orang mengalami perkara diharamkan	orang menekuni perkara tujuannya

4.2.5 Hasil Pengujian Skenario 5

Skenario kelima ini berbeda dengan keempat skenario percobaan sebelumnya. Pada skenario ini 10 sampel teks dari dataset *Matan Jauharah Tauhid* akan dilakukan *text summarization* menggunakan *ChatGPT*. Hasil *text summarization* dari *ChatGPT* akan dievaluasi dengan matrik ROUGE untuk menghitung nilai akurasi. Tujuan dilakukan pengujian dengan *ChatGPT* untuk menguji hasil dari *ChatGPT* dengan empat skenario sebelumnya yang menggunakan *train model* metode RNN dengan model arsitektur LSTM. Tabel berikut berisi 3 kolom yang terdiri dari kolom teks *Matan Jauharah Tauhid*, referensi ringkasan, dan hasil *text summarization* menggunakan *ChatGPT* dari 10 sampel teks dari dataset *Matan Jauharah Tauhid*.

Hasil *text summarization* menggunakan *ChatGPT* sudah dikumpulkan, tahapan selanjutnya adalah menghitung nilai akurasi dari ringkasan tersebut menggunakan matrik ROUGE. Nilai akurasi *text summarization* menggunakan *ChatGPT* dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil ROUGE Skenario 5

No.	Skenario 5	ROUGE (Average)	ROUGE (Maximum)
1.	<i>Recall</i>	0,3726	0,4705
2.	<i>Precision</i>	0,3530	0,444
3.	<i>F1-score</i>	0,3626	0,4571

Hasil akurasi ROUGE pada Tabel 4.15, diketahui bahwa skenario ini mendapatkan hasil rata-rata *f1-score* sebesar 0,3626 dan nilai maksimum *f1-score* sebesar 0,4571. Dengan kata lain, nilai akurasi ROUGE dari skenario percobaan ini hanya 45%.

Tabel 4.16 Sampel Hasil *Summarization* pada Skenario

No.	Text Asal	Ground Truth Summary	ChatGPT Summary
1.	Azab kubur itu adalah bagi orang kafir, orang munafik dan orang-orang mukmin yang berbuat maksiat.	Azab kubur adalah bagi orang kafir, orang munafik dan orang-orang mukmin yang berbuat maksiat.	Azab kubur hanya akan dialami oleh tiga kelompok orang, yaitu kafir, munafik, dan mukmin yang berdosa.
2.	Dan memelihara agama, kemudian jiwa, harta benda, nasab dan yang seumpamanya yakni akal dan kehormatan adalah hukumnya wajib	Dan memelihara agama, jiwa, harta benda, nasab dan yang seumpamanya yakni akal dan kehormatan adalah hukumnya wajib.	Menjaga agama, jiwa, harta, keturunan, serta martabat dan akal adalah kewajiban yang harus dilakukan.
3.	Namun ada dikatakan bahwa fitnah kubur itu adalah talajjuj (keraguan) dalam menjawab.	Namun ada dikatakan bahwa fitnah kubur adalah talajjuj (keraguan) dalam menjawab.	Ada yang mengatakan bahwa fitnah kubur adalah suatu keraguan dalam menjawab pertanyaan.

Tabel 4.16 sampel *text summarization* menggunakan *ChatGPT* berisi tiga kolom yang sedikit berbeda dengan skenario sebelumnya. Tabel sampel hasil skenario 5 ini, terdiri dari kolom teks asal yang akan diringkas, kolom *ground*

truth summary berisi ringkasan dari ahli, dan *ChatGPT summary* berisi hasil *summarization* menggunakan *ChatGPT*.

4.2.6 Evaluasi Hasil Pengujian oleh Ahli

Evaluasi hasil pengujian oleh ahli ini melibatkan dua orang santri dan satu orang dewan guru *salafiyah* (pengajian) yang bertindak sebagai validator. Evaluasi ini bertujuan untuk memberikan nilai terhadap hasil ringkasan *Matan Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM dan *tool ChatGPT* dalam skala *likert*. Dalam evaluasi ini, validator akan mengisi formulir dari penulis, dengan memberikan nilai terhadap hasil *text summarization Matan Jauharah Tauhid* dari kelima skenario percobaan dalam skala *likert*. Berikut ini merupakan keterangan validator dan bobot penilaian pada evaluasi pengujian oleh ahli.

- Validator I : Tgk. Daryani, dewan guru dayah Malikussaleh, Tanah Jambo Aye, Aceh Utara.
- Validator II : Raudhatul Ula, santriwati tingkat *a'liyah* dayah Putri Muslimat, Samalanga, Bireuen.
- Validator III : Muna Diyal Jannah, santriwati tingkat *a'liyah* dayah Putri Muslimat, Samalanga, Bireuen.
- Bobot Nilai:
 - 1 = Sangat Buruk
 - 2 = Buruk
 - 3 = Cukup Baik
 - 4 = Baik
 - 5 = Sangat Baik

Tabel 4.17 Evaluasi Hasil Pengujian oleh Ahli

NO	Skenario Percobaan	Validator			Jumlah	Nilai Rata-Rata
		I	II	III		
1.	Metode RNN dengan Model Arsitektur LSTM					
	Skenario 1	2	3	2	10	2,33
	Skenario 2	3	3	3	9	3,0
	Skenario 3	3	4	4	11	3,66
	Skenario 4	4	4	5	13	4,33
2.	ChatGPT					
	Skenario 5	4	5	5	14	4,66

Berdasarkan Tabel 4.17, dapat dilihat bahwa dari semua skenario percobaan nilai rata-rata hasil *text summarization* *Matan Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM berada direntang nilai 2,3-4,33. Sedangkan nilai rata-rata hasil *text summarization* menggunakan *ChatGPT* memiliki nilai rata-rata yaitu 4,66. Dapat disimpulkan bahwa hasil evaluasi penilaian oleh ahli nilai rata-rata tertinggi dari skenario percobaan yang menggunakan metode RNN dengan model arsitektur LSTM sedikit lebih rendah dibandingkan dengan nilai rata-rata skenario percobaan *ChatGPT* dengan selisih nilai rata-rata sebesar 0,33.

Jika diperhatikan lebih lanjut pada Tabel 4.17, penilaian validator terhadap *text summarization* *Matan Jauharah Tauhid* terdapat perbedaan dalam nilai penilaian yang diberikan. Penilaian terhadap hasil *text summarization* bersifat subjektif dan dapat dipengaruhi oleh berbagai aspek, seperti tatanan bahasa yang dihasilkan. Penilaian terhadap hasil *text summarization* bersifat subjektif dan dapat dipengaruhi oleh berbagai aspek. Meskipun nilai rata-rata dari hasil *text summarization* menggunakan *ChatGPT* lebih tinggi, perlu diketahui bahwa terdapat kekurangan dalam hasil *text summarization* dari *ChatGPT* terkait makna *matan*. Hal ini disebabkan oleh kemungkinan adanya penambahan kata-kata baru yang mengubah makna dari *matan* hasil *text summarization* *Matan Jauharah Tauhid* oleh *ChatGPT*.

4.2.7 Evaluasi Hasil Pengujian Dengan ROUGE

Evaluasi hasil dari kelima skenario percobaan untuk pengujian implementasi metode dan arsitektur dengan ROUGE dilakukan dengan membandingkan nilai *f1-score*. Perhitungan akurasi dengan ROUGE menggunakan dua indikator, yaitu *f1-score (average)* untuk menghitung jumlah nilai rata-rata *f1-score* dari seluruh data, dan *f1-score (maximum)* untuk menghitung *f1-score* tertinggi dari data yang digunakan. *F1-score* digunakan sebagai indikator evaluasi ROUGE karena bersifat representatif. Perhitungan *f1-score* melibatkan nilai *recall* dan *precision* secara bersamaan.

Tabel 4.18 Evaluasi Hasil dengan ROUGE

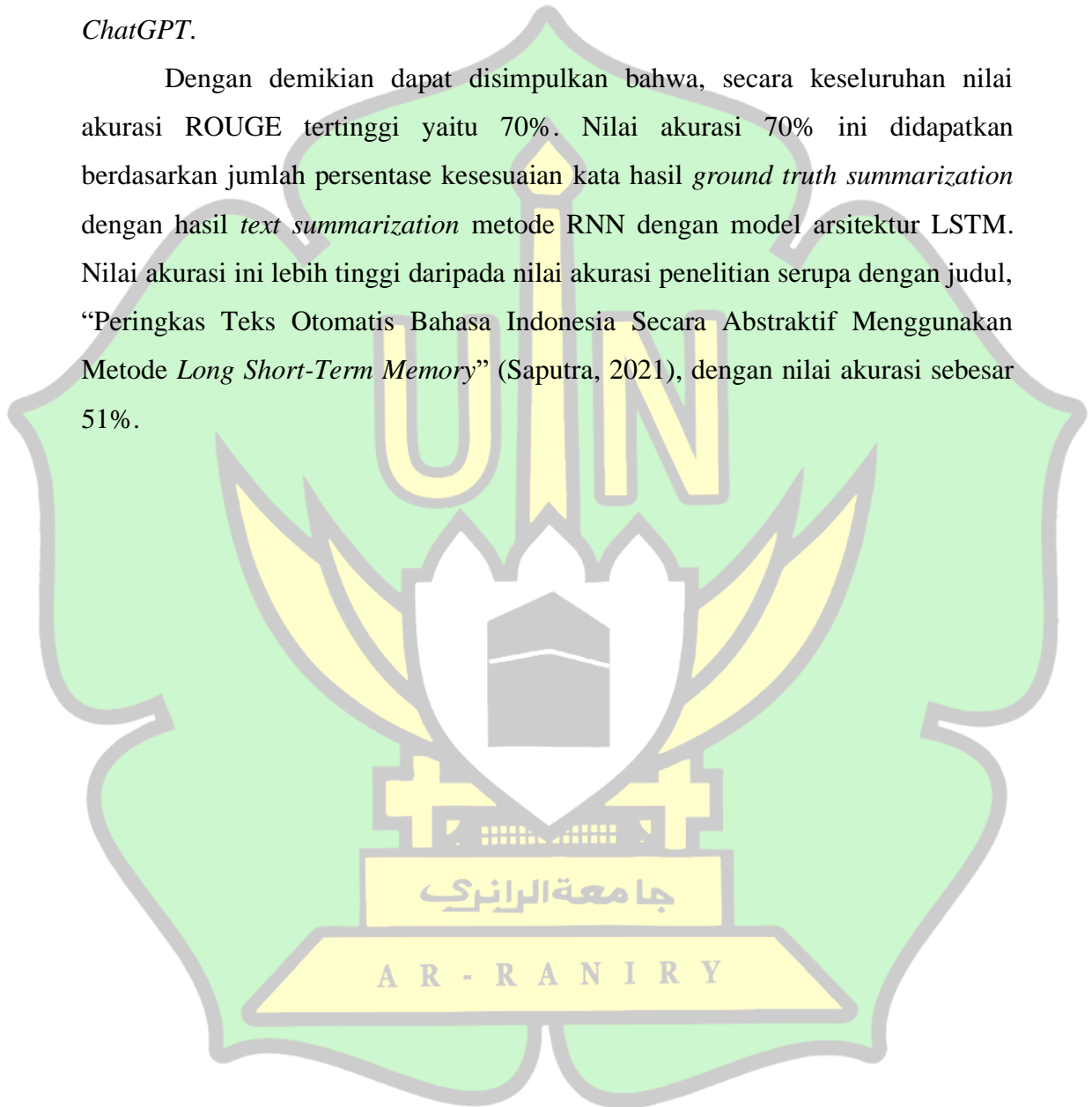
NO	Skenario Percobaan	<i>F1-Score</i> ROUGE (<i>Average</i>)	<i>F1-Score</i> ROUGE (<i>Maximum</i>)	Nilai Akurasi ROUGE
1.	Metode RNN dengan Model Arsitektur LSTM			
	Skenario 1	0,4173	0,5612	56%
	Skenario 2	0,5075	0,5882	58%
	Skenario 3	0,592	0,666	66%
	Skenario 4	0,6273	0,7058	70%
2.	ChatGPT			
	Skenario 5	0,3626	0,4571	45%

Tabel 4.18 merupakan tabel perbandingan hasil ROUGE dari kelima skenario percobaan untuk evaluasi nilai akurasi. Berdasarkan Tabel 4.18, dapat diidentifikasi bahwa pada skenario 1 sampai 4 dengan implementasi metode RNN dengan model arsitektur LSTM memiliki nilai akurasi yang lebih tinggi dibandingkan dengan skenario 5 yang menggunakan *text summarization* dari *ChatGPT*.

Jika dilakukan analisis lebih lanjut, dapat diidentifikasi bahwa skenario percobaan dengan nilai akurasi ROUGE terbaik adalah skenario 4, baik berdasarkan nilai *f1-score* rata-rata maupun nilai *f1-score* maksimum dari skenarionya. Penggabungan dua tahapan *pre-processing* data yaitu tahapan

stopword removal dan *lemmatization* memberikan pengaruh yang besar untuk nilai akurasi dari hasil *text summarization* *Matan Jauharah Tauhid*. Hasil nilai akurasi ROUGE untuk *text summarization* *Matan Jauharah Tauhid* dengan implementasi metode RNN dengan model arsitektur LSTM lebih unggul dibandingkan dengan *text summarization* *Matan Jauharah Tauhid* dengan *ChatGPT*.

Dengan demikian dapat disimpulkan bahwa, secara keseluruhan nilai akurasi ROUGE tertinggi yaitu 70%. Nilai akurasi 70% ini didapatkan berdasarkan jumlah persentase kesesuaian kata hasil *ground truth summarization* dengan hasil *text summarization* metode RNN dengan model arsitektur LSTM. Nilai akurasi ini lebih tinggi daripada nilai akurasi penelitian serupa dengan judul, “Peringkasan Teks Otomatis Bahasa Indonesia Secara Abstraktif Menggunakan Metode *Long Short-Term Memory*” (Saputra, 2021), dengan nilai akurasi sebesar 51%.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pembahasan dan penelitian yang telah diuraikan pada bab sebelumnya, dapat disimpulkan bahwa:

1. Representasi *Matan Jauharah Tauhid* dengan proses *text summarization* menggunakan metode RNN dengan model arsitektur LSTM dilakukan dengan lima skenario percobaan. Terdapat empat skenario yang dibedakan berdasarkan perbedaan variasi pada tahapan *pre-processing* data dan satu skenario melibatkan *ChatGPT*.
2. Hasil akurasi ROUGE menunjukkan bahwa nilai akurasi tertinggi hasil *text summarization* *Matan Jauharah Tauhid* menggunakan metode RNN dengan model arsitektur LSTM adalah 70%.

5.2 Saran

Berdasarkan kesimpulan dari hasil dan pembahasan pada penelitian ini, maka penulis mengemukakan beberapa saran yang dapat digunakan untuk pengembangan lanjutan dari penelitian ini, sebagai berikut:

1. Penelitian ini hanya memfokuskan satu kitab tauhid saja yaitu kitab *Matan Jauharah Tauhid*, untuk penelitian selanjutnya bisa digabungkan beberapa matan kitab tauhid untuk diterapkan *text summarization* menggunakan metode RNN dengan model arsitektur LSTM.
2. Skenario percobaan pada penelitian ini menggunakan tiga *layer* dari LSTM, yaitu *embedding layer*, *LSTM layer*, dan *dense Layer*, penelitian selanjutnya dapat dilakukan penambahan *layer* dari arsitektur LSTM untuk melihat ada-tidaknya pengaruh banyaknya lapisan *layer* terhadap nilai akurasi ROUGE.

DAFTAR PUSTAKA

- Amin, S. (2019). *EKSISTENSI KAJIAN TAUHID DALAM KEILMUAN USHULUDDIN*. 22(1), 71–83.
- Aravinth, T. S., Arunkumar, E., & Archana, P. (2020). Deep learning associated with deep challenges. *International Journal of Advanced Science and Technology*, 29(8 Special Issue), 515–523.
- Arief, B., Kholifatullah, H., & Prihanto, A. (2023). *Penerapan Metode Long Short Term Memory Untuk Klasifikasi Pada Hate Speech*. 04, 292–297.
- Basri, A. S. H. (2015). *I'tikad Mukminin (Syarah Matan Jauhar Tauhid)*.
- Dkk, V. S. (2020). *Deep Learning for Natural Language Processing in Radiology—Fundamentals and a Systematic Review*. <https://doi.org/10.1016/j.jacr.2019.12.026>
- Dr. Hj. Naili Anafah, S.HI, M.Ag.Dr. Hj. Naili Anafah, S.HI, M.Ag.Mukhamad Rikza, M. (2016). *KITAB SABIIL AL-'ABIID 'ALAA JAUHAROT ALTAUHID KIAI SALEH DARAT (TELAAH KRITIS TERHADAP KITAB TAUHID BERBAHASA JAWA ABAD-19)*.
- Elsaid, A., Mohammed, A., Ibrahim, L. F., & Sakre, M. M. (2022). A Comprehensive Review of Arabic Text Summarization. *IEEE Access*, 10, 38012–38030. <https://doi.org/10.1109/ACCESS.2022.3163292>
- Febrian Sengkey, D., Diane Kambey, F., Paulus Lengkong, S., Reynaldo Joshua, S., & Valentino Florensus Kainde, H. (2020). Pemanfaatan Platform Pemrograman Daring dalam Pembelajaran Probabilitas dan Statistika di Masa Pandemi CoVID-19. *Jurnal Informatika*, 15(4), 257–264.
- Firmansyah, M. R., Ilyas, R., & Kasyidi, F. (2020). Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network. *Prosiding The 11th Industrial Research Workshop and National Seminar*, 11(1), 488–495.
- Hamidah Dedeh. (2021). Pendidikan Islam Berbasis Nilai Tauhid. *Tsamratul - Fikri*, 15(1), 183–194.
- Hanafiah., Adang, S., Iskandar, A., (2020). *Pengantar Statistika*.
- Ivanedra, K., & Mustikasari, M. (2019). Implementasi Metode Recurrent Neural Network Pada Text the Implementation of Text Summarization With Abstractive. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 6(4). <https://doi.org/10.25126/jtiik.201961067>
- Ja'far, A. (2019). Literasi Digital Pesantren: Perubahan Dan Kontestasi. *Islamic Review: Jurnal Riset Dan Kajian Keislaman*, 8(1), 17–35. <https://doi.org/10.35878/islamicreview.v8i1.156>
- Joshi, A., Fidalgo, E., Alegre, E., & Fernández-Robles, L. (2019). SummCoder:

- An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications*, 129, 200–215. <https://doi.org/10.1016/j.eswa.2019.03.045>
- Ma'rifah, H., Wibawa, A. P., & Akbar, M. I. (2020). Klasifikasi Artikel Ilmiah Dengan Berbagai Skenario Preprocessing. *Sains, Aplikasi, Komputasi Dan Teknologi Informasi*, 2(2), 70. <https://doi.org/10.30872/jsakti.v2i2.2681>
- Md. Sanzidul, I., Sadia Sultana, S. M., Abujar, S., & Hossain, S. A. (2019). Sequence-to-sequence Bangla sentence generation with LSTM recurrent neural networks. *Procedia Computer Science*, 152, 51–58. <https://doi.org/10.1016/j.procs.2019.05.026>
- Migunani, & Kevin Aditama. (2020). Pemanfaatan Natural Language Processing Dan Pattern Matching Dalam Pembelajaran Melalui Guru Virtual. *Elkom : Jurnal Elektronika Dan Komputer*, 13(1), 121–133. <https://doi.org/10.51903/elkom.v13i1.187>
- Mubarok, M. M. (2021). *Indonesian Abstractive News Summarization Berbasis Deep Learning Dengan Metode Sequence-To-Sequence Long Short-Term Memory*.
- Pawade, D., Sakhapara, A., Jain, M., Jain, N., & Gada, K. (2018). Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM. *International Journal of Information Technology and Computer Science*, 10(6), 44–53. <https://doi.org/10.5815/ijitcs.2018.06.05>
- Rahutomo, F., & Sari, D. N. (2020). *IMPLEMENTASI LIBRARY DEEP LEARNING KERAS PADA SISTEM Program Studi Teknik Informatika , Jurusan Teknologi Informasi , Politeknik Negeri*. 73–79.
- Raya, U. P., & Tengah, K. (2023). *ChatGPT : Keuntungan , Risiko , Dan Penggunaan Bijak Dalam Era Kecerdasan Buatan*. 2(1).
- Saputra, M. A. (2021). Peringkat Teks Otomatis Bahasa Indonesia secara Abstraktif Menggunakan Metode Long Short-Term Memory. *E-Proceeding of Engineering : Vol.8, No.2 April 2021* |, 8(2), 3474–3488.
- Setyanto, A., Laksito, A., Alarfaj, F., Alreshoodi, M., Kusri, Oyong, I., Hayaty, M., Alomair, A., Almusallam, N., & Kurniasari, L. (2022). Arabic Language Opinion Mining Based on Long Short-Term Memory (LSTM). *Applied Sciences (Switzerland)*, 12(9), 1–18. <https://doi.org/10.3390/app12094140>
- Siahaan, M., Jasa, C. H., Anderson, K., & Valentino, M. (2020). Penerapan Artificial Intelligence (AI) Terhadap Seorang Penyandang Disabilitas Tunanetra. *Information System and Technology*, 01(02), 186–193.
- Soen, G. I. E., Marlina, & Renny. (2022). Implementasi Cloud Computing dengan Google Colaboratory Pada Aplikasi Pengolah Data Zoom Participants. *Journal Informatic Technology And Communication*, 6(1), 24–30.
- Suwarjin, S. (2018). Kitab Syarah Dan Tradisi Intelektual Pesantren. *Jurnal Ilmiah Mizani: Wacana Hukum, Ekonomi Dan Keagamaan*, 4(2), 85.

<https://doi.org/10.29300/mzn.v4i2.1013>

- Syahrudin, A. N., & Kurniawan, T. (2018). Input dan Output pada Bahasa Pemrograman Python. *Jurnal Dasar Pemrograman Python STMIK*, June 2018, 1–7. <https://www.researchgate.net/publication/338385483>
- Syaifullah, Kamalludin, & Triwoelandari, R. (2021). Efektivitas Dakwah Dalam Meningkatkan Pemahaman Tauhid. *Rayah Al-Islam*, 5(02), 388–402. <https://doi.org/10.37274/rais.v5i02.470>
- Tarkus, D., Sompie, S. R. U. A., & Jacobus, A. (2020). Implementasi Metode Recurrent Neural Network pada Pengklasifikasian Kualitas Telur Puyuh. *Jurnal Teknik Informatika*, 15(2), 137–144.
- Tgh. Mujiburrahman. (2010). *Permata IlmuTauhid*. 394.
- Wiranda, L., & Sadikin, M. (2019). Penerapan Long Short Term Memory Pada Data Time Series Untuk Memprediksi Penjualan Produk Pt. Metiska Farma. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 8(3), 184–196.
- Zhang, Y., Liu, Q., & Song, L. (2018). Sentence-state LSTM for text representation. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1, 317–327. <https://doi.org/10.18653/v1/p18-1030>



LAMPIRAN

Adapun lampiran lengkap hasil dari penelitian ini dapat diakses pada link *google drive* berikut:
bit.ly/LampiranPenelitianRepresentasiMatanJauharahTauhiddenganRNN-LSTM

Lampiran 1. Source Code Untuk Pre-processing Dataset

Code berikut merupakan program *python* untuk mengimport *library* dan modul yang dibutuhkan untuk *pre-processing* dataset matan.

```
import pandas as pd
import nltk
import re
import numpy as np
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
from keras.utils import plot_model
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
```

Code ini berfungsi untuk membaca *file* dataset yang akan diproses. Kemudian dilanjutkan dengan *pre-processing* data dengan tahapan *case folding, tokenizing, stopword removal, lemmatization*. Hasil dari proses ini akan disimpan pada file baru dengan format CSV.

```
data = pd.read_excel('dataset_matan.xlsx')

# Case Folding
data['case_folding'] = data['text'].apply(lambda x: x.lower())
data['case_folding'] = data['case_folding'].apply(lambda x: re
.sub(r'[\W_]+', ' ', x))

# Tokenisasi
nltk.download('punkt')
data['token'] = data['case_folding'].apply(lambda x: nltk.word
_tokenize(x))

# Stopword Removal
nltk.download('stopwords')
```

```

words_to_remove = ['di', 'sini', 'ada', 'itu', 'ini', 'adapun',
, 'kita', 'juga', 'dari', 'sungguh', 'pun', 'para', ':', 'seka
lian', 'adapaun', 'sekalian', 'disini', 'kepada', 'niscaya', '
hanya', '?', '()', '^', 'satu', '()']
stop_words = set(words_to_remove).union(set(stopwords.words('i
ndonesian')))
data['stopwords_text'] = data['token'].apply(lambda x: [word f
or word in x if word not in stop_words])
data['st_join'] = data['stopwords_text'].apply(lambda x: ' '.j
oin(x))

# Lemmatization
nltk.download('wordnet')
lemmatizer = WordNetLemmatizer()
data['lemmatization'] = data['st_join'].apply(lambda x: [lemma
tizer.lemmatize(word) for word in x.split()])
data['lemmatized_join'] = data['lemmatization'].apply(lambda x
: ' '.join(x))

# Simpan hasil
data.to_csv('hasil_preprocessingdatamatan.csv', index=False)

```

Code berikut akan membagi dataset yang sudah dilakukan *pre-processing* menjadi data *training* dan data *testing* dengan rasio 80:20. Selain itu, *code* ini juga akan menampilkan jumlah data *training* dan *testing*.

```

train_data, test_data = train_test_split(data, test_size=0.2,
random_state=42)

```

```

print('Jumlah data train:', len(train_data))
print('Jumlah data test:', len(test_data))

```

Data *training* dan data *testing* yang sudah dibagi akan disimpan ke dalam file baru dengan format CSV.

```

train_data.to_csv('data_train.csv', index=False)
test_data.to_csv('data_test.csv', index=False)

```


Lampiran 2. Source Code untuk Summarization Skenario 1

Code ini untuk instal ROUGE dan *library* yang dibutuhkan.

```
!pip install rouge
import keras
import numpy as np
import pandas as pd
import re
import matplotlib.pyplot as plt
from keras.layers import Input, LSTM, Dense, Embedding, Reshape
from keras.models import Model
from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences
from keras.utils import to_categorical
from keras.callbacks import EarlyStopping
from keras.models import Sequential
from keras.models import load_model
from rouge import Rouge
```

Code berikut untuk input file data *training* dan data *testing* dalam bentuk *dataframe*.

```
#data training
train_df = pd.read_csv("data_train.csv")

# data testing
test_df = pd.read_csv("data_test.csv")
```

Code yang digunakan untuk membuat tokenizer dan menghitung ukuran kamus kata.

```
# tokenizer
tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_df['text'])
vocab_size = len(tokenizer.word_index) + 1
```

Code ini berfungsi untuk mengubah teks menjadi *integer sequence*.

```
# text sequences to int
X_train = tokenizer.texts_to_sequences(train_df['text'])
y_train = tokenizer.texts_to_sequences(train_df['summary'])
X_test = tokenizer.texts_to_sequences(test_df['text'])
y_test = tokenizer.texts_to_sequences(test_df['summary'])
```

Selanjutnya, *code* untuk menentukan panjang *integer sequence* yang sama dengan *padding*.

```

#padding sequences
max_seq_len = max(len(seq) for seq in X_train)
X_train = pad_sequences(X_train, maxlen=max_seq_len, padding='
post')
y_train = pad_sequences(y_train, maxlen=max_seq_len, padding='
post')
X_test = pad_sequences(X_test, maxlen=max_seq_len, padding='po
st')
y_test = pad_sequences(y_test, maxlen=max_seq_len, padding='po
st')
print("X_train:")
print(X_train)
print("\ny_train:")
print(y_train)
print("\nX_test:")
print(X_test)
print("\ny_test:")
print(y_test)

```

Code untuk mengubah variabel ke dalam bentuk yang dapat dilatih dengan metode RNN dengan model arsitektur LSTM.

```
y_train = to_categorical(y_train)
```

Code untuk definisi model RNN-LSTM dengan 3 layer utama yaitu *embedding*, *LSTM*, dan *dense*.

```

# input sequence
inputs = Input(shape=(max_seq_len,))

# embedding layer
embedding_layer = Embedding(input_dim=vocab_size, output_dim=5
12)(inputs)

# LSTM layer
lstm_layer = LSTM(512, return_sequences=True)(embedding_layer)

# A output layer
outputs = Dense(vocab_size, activation='softmax')(lstm_layer)

```

Code untuk menyimpan model yang sudah didefinisikan sebelumnya.

```

# Modelling
model = Model(inputs=inputs, outputs=outputs)
model.summary()

```

Code untuk *train model* yang telah didefinisikan.

```
# Create model
model = Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
# Train model
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
history = model.fit(X_train, y_train.argmax(axis=-1), batch_size=32, epochs=50, validation_split=0.2, callbacks=[early_stopping])
```

Code untuk melihat grafik akurasi dan *loss*.

```
# Plot training and validation loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()

# Plot training and validation accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```

Code ini digunakan untuk melakukan prediksi pada data uji dengan *train model* awal.

```
# Predict on test set
y_pred = model.predict(X_test)
```

Code dibawah ini untuk mengkonversi hasil *train model* dari bentuk angka ke teks untuk membangun *text summary*.

```
# Convert predicted sequences to text
predicted_summaries = []
for prediction in y_pred:
    summary = ""
    for index in prediction.argmax(axis=-1):
        word = tokenizer.index_word.get(index, "")
```

```

        if word:
            summary += " " + word
    predicted_summaries.append(summary.strip())
# Convert ground truth sequences to text
true_summaries = []
for summary in y_test:
    summary_text = ""
    for index in summary:
        word = tokenizer.index_word.get(index, "")
        if word:
            summary_text += " " + word
    true_summaries.append(summary_text.strip())

```

Selanjutnya proses *export* hasil *summarization* ke dalam file baru dengan format CSV.

```

import csv
with open('summaries_textasal.csv', mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['Sample', 'Text', 'Predicted Summary', 'True Summary'])
    for i in range(203):
        text = test_df['text'].iloc[i]
        predicted_summary = predicted_summaries[i]
        true_summary = true_summaries[i]
        writer.writerow([i+1, text, predicted_summary, true_summary])

```

Code untuk menghitung nilai akurasi ROUGE dan menampilkan nilai matriks ROUGE.

```

# Compute ROUGE scores
rouge = Rouge()
scores = rouge.get_scores(predicted_summaries, true_summaries,
    avg=True)
print('ROUGE-1: ', scores['rouge-1'])
print('ROUGE-2: ', scores['rouge-2'])
print('ROUGE-L: ', scores['rouge-l'])

```

Lampiran 3. Source Code Skenario 2

```
!pip install rouge
import keras
import numpy as np
import pandas as pd
import re
import matplotlib.pyplot as plt
from keras.layers import Input, LSTM, Dense, Embedding, Reshape
from keras.models import Model
from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences
from keras.utils import to_categorical
from keras.callbacks import EarlyStopping
from keras.models import Sequential
from keras.models import load_model
from rouge import Rouge

# data training
train_df = pd.read_csv("data_train.csv")

# data testing
test_df = pd.read_csv("data_test.csv")

# tokenizer
tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_df['case_folding'])
vocab_size = len(tokenizer.word_index) + 1

#text sequences to int
X_train = tokenizer.texts_to_sequences(train_df['case_folding']
])
y_train = tokenizer.texts_to_sequences(train_df['summary'])
X_test = tokenizer.texts_to_sequences(test_df['case_folding'])
y_test = tokenizer.texts_to_sequences(test_df['summary'])

# padding sequences
max_seq_len = max(len(seq) for seq in X_train)
X_train = pad_sequences(X_train, maxlen=max_seq_len, padding='post')
y_train = pad_sequences(y_train, maxlen=max_seq_len, padding='post')
X_test = pad_sequences(X_test, maxlen=max_seq_len, padding='post')
y_test = pad_sequences(y_test, maxlen=max_seq_len, padding='post')

print("X_train:")
print(X_train)
print("\ny_train:")
print(y_train)
```

```

print("\nX_test:")
print(X_test)
print("\ny_test:")
print(y_test)

y_train = to_categorical(y_train)
# input sequence
inputs = Input(shape=(max_seq_len,))

# embedding layer
embedding_layer = Embedding(input_dim=vocab_size, output_dim=512)(inputs)

# LSTM layer
lstm_layer = LSTM(512, return_sequences=True)(embedding_layer)

# output layer
outputs = Dense(vocab_size, activation='softmax')(lstm_layer)
# Modelling
model = Model(inputs=inputs, outputs=outputs)
model.summary()
# Create model
model = Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
# Train model
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
history = model.fit(X_train, y_train.argmax(axis=-1), batch_size=32, epochs=50, validation_split=0.2, callbacks=[early_stopping])
# Plot training and validation loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()

# Plot training and validation accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

```

```

# Predict on test set
y_pred = model.predict(X_test)
# predicted
predicted_summaries = []
for prediction in y_pred:
    summary = ""
    for index in prediction.argmax(axis=-1):
        word = tokenizer.index_word.get(index, "")
        if word:
            summary += " " + word
    predicted_summaries.append(summary.strip())
# gt sequences to text
true_summaries = []
for summary in y_test:
    summary_text = ""
    for index in summary:
        word = tokenizer.index_word.get(index, "")
        if word:
            summary_text += " " + word
    true_summaries.append(summary_text.strip())
import csv
with open('summaries_casefolding.csv', mode='w', newline='') as
s file:
    writer = csv.writer(file)
    writer.writerow(['Sample', 'Text Case Folding', 'Predicted
Summary', 'True Summary'])
    for i in range(203):
        text = test_df['case_folding'].iloc[i]
        predicted_summary = predicted_summaries[i]
        true_summary = true_summaries[i]
        writer.writerow([i+1, text, predicted_summary, true_su
mmary])
# ROUGE scores
rouge = Rouge()
scores = rouge.get_scores(predicted_summaries, true_summaries,
    avg=True)
print('ROUGE-1: ', scores['rouge-1'])
print('ROUGE-2: ', scores['rouge-2'])
print('ROUGE-L: ', scores['rouge-1'])

```

Lampiran 4. Source Code Skenario 3

```
!pip install rouge
import keras
import numpy as np
import pandas as pd
import re
import matplotlib.pyplot as plt
from keras.layers import Input, LSTM, Dense, Embedding, Reshape
from keras.models import Model
from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences
from keras.utils import to_categorical
from keras.callbacks import EarlyStopping
from keras.models import Sequential
from keras.models import load_model
from rouge import Rouge

#data train
train_df = pd.read_csv("data_train.csv")

# data test
test_df = pd.read_csv("data_test.csv")
# tokenizer
tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_df['st_join'])
vocab_size = len(tokenizer.word_index) + 1
# text sequences to int
X_train = tokenizer.texts_to_sequences(train_df['st_join'])
y_train = tokenizer.texts_to_sequences(train_df['summary'])
X_test = tokenizer.texts_to_sequences(test_df['st_join'])
y_test = tokenizer.texts_to_sequences(test_df['summary'])
# padding sequences
max_seq_len = max(len(seq) for seq in X_train)
X_train = pad_sequences(X_train, maxlen=max_seq_len, padding='post')
y_train = pad_sequences(y_train, maxlen=max_seq_len, padding='post')
X_test = pad_sequences(X_test, maxlen=max_seq_len, padding='post')
y_test = pad_sequences(y_test, maxlen=max_seq_len, padding='post')
print("X_train:")
print(X_train)
print("\ny_train:")
print(y_train)
print("\nX_test:")
print(X_test)
```



```

print("\ny_test:")
print(y_test)
y_train = to_categorical(y_train)
# input sequence
inputs = Input(shape=(max_seq_len,))

# embedding layer
embedding_layer = Embedding(input_dim=vocab_size, output_dim=512)(inputs)

# LSTM layer
lstm_layer = LSTM(512, return_sequences=True)(embedding_layer)

# output layer
outputs = Dense(vocab_size, activation='softmax')(lstm_layer)
# Modelling
model = Model(inputs=inputs, outputs=outputs)
model.summary()
# Create model
model = Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
# Train model
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
history = model.fit(X_train, y_train.argmax(axis=-1), batch_size=32, epochs=50, validation_split=0.2, callbacks=[early_stopping])
# Plot training and validation loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()

# Plot training and validation accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
# Predict on test set
y_pred = model.predict(X_test)
# input sequence

```

```

inputs = Input(shape=(max_seq_len,))

# embedding layer
embedding_layer = Embedding(input_dim=vocab_size, output_dim=512)(inputs)

# LSTM layer
lstm_layer = LSTM(512, return_sequences=True)(embedding_layer)

# output layer
outputs = Dense(vocab_size, activation='softmax')(lstm_layer)
# Modelling
model = Model(inputs=inputs, outputs=outputs)
model.summary()
# Create model
model = Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
# Train model
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
history = model.fit(X_train, y_train.argmax(axis=-1), batch_size=32, epochs=50, validation_split=0.2, callbacks=[early_stopping])
# Plot training and validation loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()

# Plot training and validation accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
# Predict on test set
y_pred = model.predict(X_test)
# Convert predicted sequences to text
predicted_summaries = []
for prediction in y_pred:
    summary = ""
    for index in prediction.argmax(axis=-1):

```

```

        word = tokenizer.index_word.get(index, "")
        if word:
            summary += " " + word
        predicted_summaries.append(summary.strip())
# Convert ground truth sequences to text
true_summaries = []
for summary in y_test:
    summary_text = ""
    for index in summary:
        word = tokenizer.index_word.get(index, "")
        if word:
            summary_text += " " + word
    true_summaries.append(summary_text.strip())
import csv
with open('summaries_stopword.csv', mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['Sample', 'Text Stopword', 'Predicted Summary', 'True Summary'])
    for i in range(203):
        text = test_df['st_join'].iloc[i]
        predicted_summary = predicted_summaries[i]
        true_summary = true_summaries[i]
        writer.writerow([i+1, text, predicted_summary, true_summary])
# Compute ROUGE scores
rouge = Rouge()
scores = rouge.get_scores(predicted_summary, true_summary, avg=True)
print('ROUGE-1: ', scores['rouge-1'])
print('ROUGE-2: ', scores['rouge-2'])
print('ROUGE-L: ', scores['rouge-1'])

```

جامعة الرانيري

A R - R A N I R Y

Lampiran 5. Source Code Skenario 4

```
!pip install rouge
import keras
import numpy as np
import pandas as pd
import re
import matplotlib.pyplot as plt
from keras.layers import Input, LSTM, Dense, Embedding, Reshape
from keras.models import Model
from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences
from keras.utils import to_categorical
from keras.callbacks import EarlyStopping
from keras.models import Sequential
from keras.models import load_model
from rouge import Rouge

# data train
train_df = pd.read_csv("data_train.csv")

# data test
test_df = pd.read_csv("data_test.csv")

# tokenizer
tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_df['lemmatized_join'])
vocab_size = len(tokenizer.word_index) + 1

# text sequences to int
X_train = tokenizer.texts_to_sequences(train_df['lemmatized_jo
in'])
y_train = tokenizer.texts_to_sequences(train_df['summary'])
X_test = tokenizer.texts_to_sequences(test_df['lemmatized_join
'])
y_test = tokenizer.texts_to_sequences(test_df['summary'])

#padding sequences
max_seq_len = max(len(seq) for seq in X_train)
X_train = pad_sequences(X_train, maxlen=max_seq_len, padding='
post')
y_train = pad_sequences(y_train, maxlen=max_seq_len, padding='
post')
X_test = pad_sequences(X_test, maxlen=max_seq_len, padding='po
st')
y_test = pad_sequences(y_test, maxlen=max_seq_len, padding='po
st')
print("X_train:")
print(X_train)
print("\ny_train:")
```

```

print(y_train)
print("\nX_test:")
print(X_test)
print("\ny_test:")
print(y_test)
y_train = to_categorical(y_train)
# input sequence
inputs = Input(shape=(max_seq_len,))

# embedding layer
embedding_layer = Embedding(input_dim=vocab_size, output_dim=512)(inputs)

# LSTM layer
lstm_layer = LSTM(512, return_sequences=True)(embedding_layer)

# output layer
outputs = Dense(vocab_size, activation='softmax')(lstm_layer)
# Modelling
model = Model(inputs=inputs, outputs=outputs)
model.summary()
# Create model
model = Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
# Train model
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
history = model.fit(X_train, y_train.argmax(axis=-1), batch_size=32, epochs=50, validation_split=0.2, callbacks=[early_stopping])
# Plot training and validation loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()

# Plot training and validation accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

```

```

# Predict on test set
y_pred = model.predict(X_test)
# predicted sequences to text
predicted_summaries = []
for prediction in y_pred:
    summary = ""
    for index in prediction.argmax(axis=-1):
        word = tokenizer.index_word.get(index, "")
        if word:
            summary += " " + word
    predicted_summaries.append(summary.strip())
# gt sequences to text
true_summaries = []
for summary in y_test:
    summary_text = ""
    for index in summary:
        word = tokenizer.index_word.get(index, "")
        if word:
            summary_text += " " + word
    true_summaries.append(summary_text.strip())
import csv
with open('summaries_lemmatization.csv', mode='w', newline='')
    as file:
        writer = csv.writer(file)
        writer.writerow(['Sample', 'Text Lemmatization', 'Predicted
Summary', 'True Summary'])
        for i in range(203):
            text_lemmatization = test_df['lemmatized_join'].iloc[i]

            predicted_summary = predicted_summaries[i]
            true_summary = true_summaries[i]
            writer.writerow([i+1, text_lemmatization, predicted_summary, true_summary])
# ROUGE scores
rouge = Rouge()
scores = rouge.get_scores(predicted_summary, true_summary, avg
=True)
print('ROUGE-1: ', scores['rouge-1'])
print('ROUGE-2: ', scores['rouge-2'])
print('ROUGE-L: ', scores['rouge-1'])

```

Lampiran 6. Source Code Skenario 5

```
!pip install rouge
import keras
from rouge import Rouge
# Initialize ROUGE
rouge = Rouge()

# ground truth summary and predicted summary
summary_chatgpt = "Azab kubur adalah bagi orang kafir, orang m
unafik dan orang-
orang mukmin yang berbuat maksiat.Menjaga agama, jiwa, harta,
keturunan, serta martabat dan akal adalah kewajiban yang harus
dilakukan.Ada yang mengatakan bahwa fitnah kubur adalah suatu
keraguan dalam menjawab pertanyaan.Qadar adalah penentuan All
ah atas segala sesuatu berdasarkan ketetapan yang telah ditent
ukan, termasuk bentuk dan kehendak-
Nya.Sunnah Nabi Muhammad yang dijelaskan atau ditunjukkan mela
lui kata-
kata, tindakan, dan keyakinan adalah lebih kuat dalam memberik
an petunjuk daripada sumber lainnya.Selain mengucapkan dua kal
imat syahadat, melakukan amal kebaikan juga merupakan syarat u
ntuk memiliki iman.Kehidupan yang sempurna mencakup keseimbang
an antara tubuh dan jiwa, sehingga hidup dapat dinikmati secar
a menyeluruh.Ajbuz zanab juga menjadi perdebatan di antara par
a ulama tentang fana (paham bahwa segala sesuatu bersifat fana
) atau baqa (segala sesuatu memiliki kelangsungan hidup).Konse
p wahdaniyah dalam agama Islam menggambarkan bahwa Allah adala
h satu, baik dalam zat, sifat, dan perbuatan-
Nya.Perincian tentang usaha dan tawakal menunjukkan bahwa kedu
anya bervariasi tergantung pada keadaan dan kondisi individu."
summary = "Azab kubur itu adalah bagi orang kafir, orang munaf
ik dan orang-
orang mukmin yang berbuat maksiat.Dan memelihara agama, kemudi
an jiwa, harta benda, nasab dan yang seumpamanya yakni akal da
n kehormatan adalah hukumnya wajib.Namun ada dikatakan bahwa f
itnah kubur itu adalah talajjuj (keragu-
raguan) dalam menjawab.Sedangkan qadar adalah Penjadian Allah
terhadap segala sesuatu berdasarkan ketetapan yang sudah dikhu
suskan dan bentuk yang sudah ditentukan serta sudah dikehendak
i oleh Allah.Tiap-
tiap petunjuk yang ada penjelasan atau contohnya dari Nabi Muh
ammad berupa perkataan, perbuatan dan iktikad lebih kuat ketim
bang selainnya.seperti halnya mengucapkan dua kalimah syahadat
maka melakukan, amal-
amal kebaikan juga merupakan syarat bagi iman.Hidup yang dimak
```

sudkan itu adalah untuk zat dan ruh, maka dia adalah hidup yang sempurna. Ajbuz zanab juga dikhilafkan oleh para ulama mengenai ai fana atau baqanya. Maksud dari Allah itu bersifat dengan wah daniyah adalah bahwa Allah itu Esa, baik pada zat, sifat maupun af'al. Perincian dimaksud adalah bahwa usaha dan tawakal itu berbeda-beda sesuai perbedaan ihwal manusia."

```
# Calculate ROUGE scores
```

```
scores = rouge.get_scores(summary_chatgpt, summary, avg=True)
```

```
# Print ROUGE scores
```

```
print('ROUGE-1: ', scores['rouge-1'])
```

```
print('ROUGE-2: ', scores['rouge-2'])
```

```
print('ROUGE-L: ', scores['rouge-l'])
```



RIWAYAT HIDUP



Novi Ayu Irhami, lahir di Seunuddon, Aceh Utara pada tanggal 22 November 2001. Anak kedua dari tiga bersaudara, dari pasangan Ibu Hajidah, S.Pd dan Bapak Amiruddin, S.Pd. Penulis menyelesaikan pendidikan Sekolah Dasar di SD Negeri 5 Seunuddon. Kemudian melanjutkan Pendidikan di jenjang Sekolah Menengah Pertama di SMP Negeri 1 Seunuddon dan lulus tahun 2016. Penulis menempuh pendidikan jenjang Sekolah Menengah Atas di SMA Swasta Muslimat Samalanga dan lulus pada tahun 2019. Pada tahun yang sama penulis terdaftar sebagai mahasiswa pada program studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry, Banda Aceh melalui jalur seleksi SNMPTN.

