

PROPOSAL
IMPLEMENTASI ALGORITMA *NEURAL MACHINE TRANSLATION*
PENERJEMAH BAHASA DAERAH MUNA KE BAHASA
INDONESIA BERBASIS ANDROID

Diajukan Untuk Memenuhi
Salah Satu Syarat Memperoleh Gelar Sarjana Teknik



HADIKUL JABIL
E1E120009

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HALU OLEO
KENDARI
2024

INTISARI

Hadikul Jabil, E1E120009

IMPLEMENTASI ALGORITMA *NEURAL MACHINE TRANSLATION* PENERJEMAH BAHASA DAERAH MUNA KE BAHASA INDONESIA BERBASIS ANDROID

Proposal, Fakultas Teknik, 2024

Kata Kunci: Aplikasi Penerjemah, *Android*, Bahasa Daerah Muna, *BLEU Score*, *Neural Machine Translation*.

Seiring dengan perkembangan zaman dan modernisasi di lingkungan masyarakat di Kota Raha, Kabupaten Muna terkhususnya anak-anak dan remaja yang masih jarang dibiasakan atau mendapatkan pembelajaran mengenai bahasa daeran Muna baik secara formal ataupun tidak formal dari lingkungan keluarga dan sekolahnya.

Seiring dengan kemajuan teknologi, aplikasi penerjemah atau *translator* otomatis telah menjadi sarana efektif untuk mengatasi hambatan komunikasi lintas bahasa. Salah satu pendekatan terkini yang memperlihatkan kemajuan signifikan adalah penggunaan algoritma *Neural Machine Translation* (NMT). *Neural Machine Translation* menggunakan jaringan saraf untuk memahami dan menerjemahkan teks dari satu bahasa ke bahasa lain dengan tingkat akurasi yang lebih tinggi dibandingkan dengan metode tradisional.

Penelitian ini akan menggunakan dua jenis pengujian, yaitu pengujian *black box* dan *BLEU Score*. Pengujian *black box* akan digunakan untuk menguji sistem perangkat lunak *android* dan pengujian *BLEU Score* akan digunakan untuk melihat Tingkat akurasi dari hasil terjemahan algoritma *Neural Machine Translation*.

ABSTRAK

Hadikul Jabil, E1E120009

IMPLEMENTASI ALGORITMA *NEURAL MACHINE TRANSLATION* PENERJEMAH BAHASA DAERAH MUNA KE BAHASA INDONESIA BERBASIS ANDROID

Proposal, Fakultas Teknik, 2024

Kata Kunci: *Translator Application, Android, BLUE Score, Muna Regional Language, Neural Machine Translation.*

In line with the times and modernization in the community environment in Raha City, Muna Regency, especially children and teenagers who are still rarely accustomed to or receive learning about the Muna regional language either formally or informally from their family and school environment.

Along with advances in technology, automatic translator applications have become an effective means of overcoming cross-language communication barriers. One of the latest approaches that shows significant progress is the use of the Neural Machine Translation (NMT) algorithm. Neural Machine Translation uses neural networks to understand and translate text from one language to another with a higher degree of accuracy compared to traditional methods.

This research will use two types of testing, namely black box testing and BLEU Score. Black box testing will be used to test the Android software system and BLEU Score testing will be used to see the level of accuracy of the translation results of the Neural Machine Translation algorithm.

DAFTAR ISI

INTISARI.....	ii
ABSTRAK	iii
DAFTAR ISI	iv
DAFTAR TABEL.....	vi
DAFTAR GAMBAR	vii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan.....	4
1.7 Tinjauan Pustaka	4
BAB II LANDASAN TEORI.....	9
2.1 Bahasa Daerah.....	9
2.2 Bahasa Muna.....	11
2.2.1 Dialek Bahasa Muna	12
2.3 <i>Preprocessing</i>	13
2.4 <i>Deep Learning</i>	14
2.5 <i>Natural Language Processing</i> (NLP)	15
2.6 <i>Recurrent Neural Network</i> (RNN)	16
2.7 <i>Neural Machine Translation</i> (NMT).....	19
2.7.1 Rumus Matematika Model NMT	22
2.7.2 Perhitungan Manual NMT	23
2.8 <i>Android</i>	26
2.9 <i>Flowchart</i>	26
2.10 <i>Unified Modeling Language</i> (UML).....	27
2.11 <i>Use Case diagram</i>	28
2.12 <i>Activity Diagram</i>	29
2.13 <i>Sequence Diagram</i>	30
2.14 <i>Class Diagram</i>	31
2.15 Metodologi Pengembangan Sistem.....	32

2.16	Pengujian <i>Black Box</i>	33
2.17	Pengujian <i>Bilingual Evaluation Understudy</i> (BLEU) Score.....	34
BAB III METODOLOGI PENELITIAN.....		35
3.1	Metode Pengumpulan Data	35
3.2	Metode Pengembangan Sistem	35
3.3	Waktu dan Tempat Penelitian.....	36
3.3.1	Waktu Penelitian	36
3.3.2	Tempat Penelitian.....	36
3.4	Analisis Kebutuhan Sistem	37
3.5	<i>Hardware</i> dan <i>Software</i>	37
3.5.1	<i>Hardware</i>	37
3.5.2	<i>Software</i>	37
3.6	Perancangan Sistem	38
3.6.1	Pengumpulan dan Persiapan Data.....	38
3.6.2	<i>Preprocessing</i> Data	38
3.6.3	Pelatihan Model	39
3.6.4	Evaluasi dan Penyesuaian	39
3.6.5	Pengembangan Aplikasi.....	39
3.6.6	Optimasi dan Penyempurnaan	40
3.7	Perancangan <i>Flowchart</i>	40
3.7.1	<i>Flowchart</i> Aplikasi	40
3.7.2	<i>Flowchart</i> Model	41
3.8	Perancangan <i>Unified Modeling Language</i> (UML)	42
3.8.1	<i>Use Case Diagram</i>	42
3.8.2	<i>Activity Diagram</i>	42
3.8.3	<i>Sequance Diagram</i>	43
3.8.4	<i>Class Diagram</i>	44
3.9	Perancangan <i>Interface</i>	44
3.10	Pengujian <i>Black Box</i>	47
3.11	Pengujian <i>BLEU Score</i>	47
DAFTAR PUSTAKA		50

DAFTAR TABEL

Tabel 1.1 Tinjauan Pustaka	5
Tabel 2.1 Simbol-Simbol <i>Flowchart</i>	26
Tabel 2.2 Simbol-Simbol <i>Use Case Diagram</i>	28
Tabel 2.3 Simbol-Simbol <i>Activity Diagram</i>	29
Tabel 2.4 Simbol-Simbol <i>Sequence Diagram</i>	30
Tabel 2.5 Simbol-Simbol <i>Class Diagram</i>	31
Tabel 3. 1 Gantt Chart Waktu Penelitian	36
Tabel 3.2 Kebutuhan Perangkat Keras	37
Tabel 3.3 Kebutuhan Perangkat Keras	38
Tabel 3.4 Pengujian <i>Black Box</i>	47

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi <i>Deep Learning</i> (Razsiah., 2023).....	15
Gambar 2.2 <i>Recurrent Neural Network</i> (RNN)	17
Gambar 2.3 <i>Unfolding Recurrent Neural Network</i>	17
Gambar 2.4 Pemrosesan RNN	18
Gambar 2.5 Struktur RNN	19
Gambar 2.6 Model <i>Encoder-Decoder</i>	20
Gambar 3.1 <i>Flowchart</i> Aplikasi.....	41
Gambar 3.2 <i>Flowchart</i> Model.....	41
Gambar 3.3 <i>Use Case Diagram</i>	42
Gambar 3.4 <i>Activity Diagram</i>	43
Gambar 3.5 <i>Sequence Diagram</i>	43
Gambar 3.6 <i>Class Diagram</i>	44
Gambar 3.7 <i>Splash Screen</i>	45
Gambar 3.8 <i>Dashboard</i>	45
Gambar 3.9 <i>Translation Language</i>	46
Gambar 3.10 <i>About Me</i>	46

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia, dengan keberagaman budaya dan linguistiknya, memiliki berbagai bahasa daerah yang berfungsi sebagai sarana komunikasi dalam lingkungan masyarakat setempat. Sekitar 652 bahasa daerah di Indonesia telah diverifikasi oleh Badan Pengembangan dan Pembinaan Bahasa Kementerian Pendidikan dan Kebudayaan (Badan Bahasa Kemendikbud). Berdasarkan sumber dari Summer Institute of Linguistik terdapat 719 bahasa daerah di Indonesia, dimana dari jumlah tersebut, sekitar 707 bahasa daerah masih aktif digunakan (Abdurrahman dkk., 2023)

Salah satu bahasa daerah yaitu bahasa Muna, digunakan oleh masyarakat di daerah Muna, Sulawesi Tenggara. Meskipun bahasa Muna memegang peranan penting dalam identitas budaya dan warisan lokal, terbatasnya sumber daya untuk mendukung pembelajaran dan pemahaman bahasa ini dapat menimbulkan masalah berkelanjutan.

Seiring dengan perkembangan zaman dan modernisasi di lingkungan masyarakat di Kota Raha, Kabupaten Muna terkhususnya anak-anak dan remaja yang masih jarang dibiasakan atau mendapatkan pembelajaran mengenai bahasa daerah Muna baik secara formal ataupun tidak formal dari lingkungan keluarga dan sekolahnya, mengakibatkan kurangnya penutur bahasa daerah Muna saat ini (Abdurrahman dkk., 2023)

Kurangnya dialog antara orang tua dan anak menjadi salah satu faktor yang menyebabkan bahasa daerah ini tidak terwariskan ke generasi muda di Kabupaten Muna khususnya di daerah Kota Raha. Menurut Kepala Kantor Bahasa Sulawesi Tenggara, tujuh dari sembilan bahasa daerah di provinsi Sulawesi Tenggara terancam punah akibat sejumlah faktor. Tujuh bahasa daerah tersebut yakni Bahasa Ciacia, Bahasa Culambacu, Bahasa Lasalimu Kamaru, Bahasa Kulisusu, Bahasa Moronene, Bahasa Muna, dan Bahasa Tolaki.

Seiring dengan kemajuan teknologi, aplikasi penerjemah atau *translator* otomatis telah menjadi sarana efektif untuk mengatasi hambatan komunikasi lintas bahasa. Salah satu pendekatan terkini yang memperlihatkan kemajuan signifikan adalah penggunaan algoritma *Neural Machine Translation* (NMT). *Neural Machine Translation* menggunakan jaringan saraf untuk memahami dan menerjemahkan teks dari satu bahasa ke bahasa lain dengan tingkat akurasi yang lebih tinggi dibandingkan dengan metode tradisional.

NMT bergantung pada banyak komponen dan metode, namun yang paling banyak digunakan adalah *Recurrent Neural Network* (RNN). RNN memiliki kemampuan untuk mengolah rangkaian data, seperti teks, yang pada dasarnya berupa rangkaian kata. Model RNN digunakan dalam berbagai tugas *Natural Language Processing* (NLP) seperti teks klasifikasi, pengenalan suara, sistem tanya jawab, dan terjemahan teks. Ada dua jenis RNN yang berbeda: *Long Short-Term Memory* (LSTM) dan *Gate Recurrent Unit* (GRU), perbedaan di antara keduanya terletak pada struktur unit memorinya (Fauziyah dkk., 2022).

Penelitian ini mengusulkan pendekatan NMT pada mesin penterjemahkan kalimat Bahasa Muna ke Bahasa Indonesia menggunakan arsitektur *Encoder-Decoder* untuk memproses kata masukan pada mesin penterjemah dan mekanisme tambahan yaitu *Attention Mechanism* (Al-Ibrahim dan Duwairi, 2020).

Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi translator menggunakan algoritma *Neural Machine Translation* untuk menerjemahkan teks dari Bahasa Muna ke Bahasa Indonesia, karena dengan sumber daya terjemahan yang terbatas seperti bahasa Muna ini akan efektif bila menggunakan algoritma *Neural Machine Translation* (Al-Ibrahim dan Duwairi, 2020).

Aplikasi ini diharapkan dapat menjadi solusi yang efisien dan efektif untuk memfasilitasi komunikasi lintas bahasa, mendukung pelestarian bahasa daerah, dan memberikan akses informasi yang lebih luas bagi masyarakat yang berbicara bahasa Muna. Dengan demikian, penelitian ini mendukung upaya pelestarian keanekaragaman bahasa dan budaya di Indonesia melalui pemanfaatan teknologi terkini.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah diuraikan di atas, dapat dirumuskan permasalahan antara lain:

1. Bagaimana merancang dan membangun aplikasi *translate* bahasa daerah Muna ke bahasa Indonesia berbasis *Android*.
2. Bagaimana mengimplementasikan algoritma *Neural Machine Translation* pada aplikasi *translate* bahasa daerah Muna ke bahasa Indonesia berbasis *Android*.

1.3 Batasan Masalah

Adapun batasan masalah penelitian ini yaitu:

1. Pengembangan ini difokuskan untuk menterjemahkan bahasa daerah Muna ke bahasa Indonesia.
2. Penelitian ini terbatas pada penggunaan algoritma *Neural Machine Translation* (NMT) sebagai basis aplikasi translator.
3. Penelitian ini akan membatasi evaluasi terhadap akurasi terjemahan antara bahasa Muna dan bahasa Indonesia.
4. Bahasa daerah Muna yang di terjemahkan menggunakan dialog bahasa Muna standar.
5. Jumlah kalimat yang digunakan dalam data penelitian ini menggunakan 1500 kalimat yang terdiri dari 750 bahasa Indonesia dan 750 bahasa daerah Muna.
6. Minimal versi *Android* yang digunakan adalah versi 6.0.1 (*Marshmallow*).

1.4 Tujuan Penelitian

Adapun tujuan dalam penelitian ini, yaitu:

1. Untuk merancang dan membangun sebuah aplikasi *translate* bahasa daerah Muna ke bahasa Indonesia berbasis *Android*.

2. Untuk membuat media belajar bahasa daerah Muna untuk masyarakat khususnya pelajar.

1.5 Manfaat Penelitian

Adapun manfaat dalam penelitian ini, yaitu:

1. Dapat merancang dan membangun sebuah aplikasi *transalate* bahasa daerah Muna ke bahasa Indonesia berbasis *Android*.
2. Dapat menyediakan media belajar bahasa daerah muna untuk masyarakat khususnya pelajar.

1.6 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

BAB II LANDASAN TEORI

Bab ini berisi pengertian-pengertian dan teori-teori yang menjadi acuan dalam pembuatan Analisa dan pemecahan dari permasalahan yang dibahas meliputi *Neural Machine Transalation* (NMT), *Android*, *Flowchart*, *Unified Modeling Language* (UML) dan pendukung lainnya.

BAB III METODE PENELITIAN

Bab ini memuat metodologi penelitian yang meliputi metode pengumpulan data, uraian metode *Neural Machine Translation* (NMT) untuk pengembangan sistem dan uraian waktu penelitian.

1.7 Tinjauan Pustaka

Ada beberapa jurnal yang digunakan untuk menyusun proposal penelitian ini yang membahas mengenai bahasa daerah muna, algoritma *Neural Machine Translation*, arsitektur *Recurrent Neural Network* dan pengujian *Bilingual Evaluation Understudy* (BLEU) score.

Tabel 1.1 Tinjauan Pustaka

Judul	Hasil
<p><i>Neural Machine Translation from Jordanian Dialect to Modern Standard Arabic</i></p>	<p>(Al-Ibrahim dan Duwairi, 2020) Pada penelitian ini menggunakan <i>Deep Learning</i> dengan model <i>encoder decoder</i> RNN. Penelitian yang dilakukan dengan menggunakan model ini dibagi menjadi dua bagian: level kata dan level kalimat, dan hasilnya sebagai berikut: nilai <i>loss</i> atau tingkat kesalahan sama dengan 0,8 dan akurasi sama dengan 91,3% saat menggunakan model penerjemahan kata ke kata; dan nilai <i>loss</i> sama dengan 3,33 dan akurasi sama dengan 63,2% bila menggunakan model penerjemahan kalimat mendapatkan tingkat akurasi secara spesifik untuk dataset kata (W2W) adalah 91,3%, dan untuk dataset kalimat (S2S) sebesar 63,2%.</p>
<p>Penerjemah Kalimat Bahasa Lampung-Indonesia Dengan Pendekatan <i>Neural Machine Translation</i> Berbasis <i>Attention</i></p>	<p>(Abidin dkk., 2018) Penerjemahan bahasa Lampung-Indonesia dengan pendekatan NMT <i>attention</i>. Penelitian yang dilakukan menggunakan 3000 kalimat sebagai <i>corpus</i> paralel bahasa Lampung dan bahasa Indonesia dan dilanjutkan dengan penentuan parameter model NMT. Pengujian dengan pendekatan NMT menggunakan 25 kalimat tunggal tanpa out-of-vocabulary (OOV), 25 kalimat tunggal dengan OOV, 25 kalimat majemuk tanpa OOV dan 25 kalimat majemuk dengan OOV. Hasil pengujian penerjemahan kalimat bahasa Lampung ke bahasa Indonesia menunjukkan nilai rata-rata bilingual evaluation understudy (BLEU) yang diperoleh dengan pendekatan NMT adalah 51.96.</p>

Tabel 1.1 Tinjauan Pustaka (Lanjutan)

Judul	Hasil
Mesin Penerjemah Bahasa Indonesia-Bahasa sunda Menggunakan <i>Recurrent Neurall Network</i>	(Fauziyah dkk., 2022) Penelitian ini membangun sebuah penerjemah bahasa Indonesia ke bahasa Sunda. Adapun tahapan yang digunakan dimulai dari pra proses menggunakan <i>text preprocessing</i> dan <i>word embedding</i> <i>Word2Vec</i> dan pendekatan yang digunakan yaitu <i>Neural Machine Translation</i> (NMT) dengan arsitektur <i>Encoder Decoder</i> yang di dalamnya terdapat sebuah <i>Recurrent Neural Network</i> (RNN). Pengujian pada penelitian menghasilkan nilai optimal oleh GRU sebesar 99,17%. Model dengan menggunakan <i>Attention</i> mendapat 99,94%. Penggunaan model optimasi mendapat hasil optimal oleh Adam 99,35% dan hasil BLEU Score dengan optimal bleu 92,63% dan <i>brevity penalty</i> 0,929.
Aplikasi Penerjemah Bahasa Bangka Ke Bahasa Indonesia Menggunakan Neural Machine Translation Berbasis Website	(F. Razsiah dkk., 2023) Penelitian ini bertujuan untuk membuat dan mengetahui hasil terjemahan aplikasi penerjemah bahasa Bangka ke bahasa Indonesia dengan model penerjemah RNN. Pengembangan dimulai dari membuat <i>corpus</i> paralel bahasa Bangka ke bahasa Indonesia dan arsitektur mesin penerjemah dengan model RNN. Metode penelitian dimulai dari pengumpulan <i>dataset</i> , <i>preprocessing</i> data, pemodelan dan pelatihan, evaluasi dan implementasi sistem. Pada model akan menggunakan <i>Hyperparameter</i> dengan jumlah <i>batch size</i> sebanyak 25, jenis <i>optimizer</i> yang digunakan adalah <i>Root-Mean-Square Propagation</i> (RMSprop). Untuk perhitungan nilai <i>loss</i> menggunakan <i>sparse categorical entropy</i> . Selain itu penerapan <i>early stopping</i> juga

Tabel 1.1 Tinjauan Pustaka (Lanjutan)

JUDUL	HASIL
	<p>digunakan dengan nilai <i>patience</i> sebesar 10 agar model akan terus melakukan pelatihan sebesar 10 kali secara beruntuk dan tidak menghasilkan progress nilai <i>loss</i> melewati batas nilai nol. Dari hasil evaluasi BLEU Scores didapatkan nilai sebesar 55,3% untuk bahasa Bangka ke bahasa Indonesia.</p>
<p>Analisis Perbandingan Nilai Akurasi Mekanisme Attention Bahdanau dan Luong pada Neural Machine Translation Bahasa Indonesia ke Bahasa Melayu Ketapang dengan Arsitektur Recurrent Neural Network</p>	<p>(Gunawan dkk., 2021) Pada penelitian ini dilakukan pengujian berdasarkan metode penambahan secara konsisten dengan jumlah <i>epoch</i> didapatkan nilai skor BLEU yaitu pada <i>attention</i> Bahdanau menghasilkan akurasi 35,96% tanpa <i>out-of-vocabulary</i> (OOV) dengan menggunakan jumlah <i>epoch</i> 40, sedangkan pada <i>attention Luong</i> menghasilkan akurasi 26,19% tanpa OOV menggunakan jumlah 30 <i>epoch</i>. Hasil pengujian berdasarkan <i>k-fold cross validation</i> didapatkan nilai rata-rata akurasi tertinggi sebesar 40,25% tanpa OOV untuk <i>attention</i> Bahdanau dan 30,38% tanpa OOV untuk <i>attention Luong</i>, sedangkan pengujian manual oleh dua orang ahli bahasa memperoleh nilai akurasi sebesar 78,17% dan 72,53%.</p>

Tabel 1.1 Tinjauan Pustaka (Lanjutan)

JUDUL	HASIL
Implementasi Algoritma Knuth Morris Pratt pada Aplikasi Kamus Bahasa Muna	(Abdurrahman dkk., 2023) Pada penelitian ini mengembangkan aplikasi kamus Bahasa Muna dengan algoritma <i>Knuth Morris Pratt</i> dengan <i>string matching</i> dalam proses pencocokan kata yang akan dikerjakan, dimana perancangan <i>database</i> menggunakan <i>Firestore</i> yang merupakan sebuah <i>database online</i> yang berasal dari Google. Hasil penelitian ini diuji menggunakan teknik pengujian <i>blackbox</i> dan aplikasi kamus digital ini diuji dengan menggunakan 30 kata mencari nilai <i>precision</i> dan nilai <i>recall</i> dan didapatkan hasil <i>precision</i> sebesar 92,4%, hasil <i>recall</i> sebesar 100% dan keakuratan sebesar 100%

Berdasarkan keenam penelitian yang telah dikaji, terbukti bahwa aplikasi penerjemah dapat dibuat menggunakan NMT (*Neural Machine Translation*). Hal ini membuka peluang dan potensi yang besar untuk mengembangkan aplikasi penerjemah yang lebih efektif dan efisien dengan pengembangan aplikasi penerjemah menggunakan arsitektur RNN (*Recurrent Neural Network*) berbasis Android.

BAB II

LANDASAN TEORI

2.1 Bahasa Daerah

Pengaturan tentang bahasa daerah dalam peraturan perundang-undangan bukanlah hal utama, kecuali dalam beberapa perda. Pengaturan penggunaan Bahasa daerah menjadi pelengkap pengaturan tentang bahasa Indonesia atau bahasa negara. Dalam Undang-Undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional – termasuk Undang-Undang Nomor 4 Tahun 1950 Undang-Undang Nomor 12 Tahun 1954 dan Undang-Undang Nomor 2 Tahun 1989 yang menjadi cikal bakal Undang-Undang Nomor 20 Tahun 2003 penggunaan bahasa daerah diatur sebagai pelengkap penggunaan bahasa Indonesia yang diwajibkan dalam penyelenggaraan pendidikan nasional di Indonesia. Bahasa daerah boleh digunakan pada tahap awal pendidikan untuk menyampaikan pengetahuan dan keterampilan tertentu. Senada dengan itu, bahasa asing dapat pula digunakan sebagai bahasa pengantar untuk mendukung pemerolehan kemahiran berbahasa asing peserta didik. Baik bahasa daerah maupun bahasa asing mempunyai fungsi pendukung bahasa Indonesia sebagai bahasa pengantar utama dalam sistem pendidikan nasional (Pemerintah Indonesia, 2003).

Pelindungan terhadap bahasa daerah didasarkan pada amanat Pasal 32 Ayat 2 UUD 1945, yang menyatakan bahwa negara menghormati dan memelihara bahasa daerah sebagai kekayaan budaya nasional. Dengan ayat itu, negara memberi kesempatan dan keleluasaan kepada masyarakat untuk melestarikan dan mengembangkan bahasanya sebagai bagian dari kebudayaannya masing-masing. Selain itu, negara memajukan kebudayaan nasional Indonesia di tengah peradaban dunia dengan menjamin kebebasan masyarakat dalam memelihara dan mengembangkan nilai-nilai budayanya. Kebebasan yang diberikan UUD 1945 bukan berarti kebebasan yang tanpa pembatasan karena hingga pada batas tertentu pengembangan dan penggunaan bahasa daerah pasti akan berbenturan dengan ketentuan lain. Untuk keperluan bernegara, kebebasan penggunaan bahasa daerah

yang diamanatkan itu akan terbentur dengan batas penggunaan bahasa negara. Untuk keperluan hidup dan pergaulan sosial, keleluasaan penggunaan satu bahasa daerah harus juga menghormati penggunaan bahasa daerah lain. Dengan kata lain, keleluasaan penggunaan dan pengembangan bahasa daerah dalam banyak hal juga tidak boleh melanggar norma “sosial” dan norma perundang-undangan yang ada (Pemerintah Indonesia, 2003).

Untuk menjamin hubungan harmonis masyarakat Indonesia atas penggunaan bahasanya, Pasal 36C UUD 1945 mengamanatkan bahwa perihal bendera, bahasa, dan lambang negara, serta lagu kebangsaan harus diatur dalam sebuah undang-undang. Amanat pasal itulah yang melahirkan Undang-Undang Nomor 24 Tahun 2009 tentang Bendera, Bahasa, dan Lambang Negara, serta Lagu Kebangsaan. Khusus tentang bahasa negara, pengaturannya dituangkan dalam Bab III, mulai Pasal 25 sampai dengan Pasal 45 dalam undang-undang tersebut. Ibarat sisi mata uang, pengaturan tentang bahasa negara, tentu berkaitan dengan pengaturan bahasa yang bukan bahasa negara, yang dalam hal itu berupa bahasa daerah dan bahasa asing

Undang-Undang Nomor 24 Tahun 2009 yang disahkan berlakunya pada tanggal 9 Juli 2009 mengatur empat substansi pokok, yaitu bendera negara, bahasa negara, lambang negara, dan lagu kebangsaan. Dalam undang-undang itu, bahasa Indonesia dibatasi sebagai bahasa yang dinyatakan sebagai bahasa resmi negara dalam Pasal 36 Undang-Undang Dasar Negara Kesatuan Republik Indonesia Tahun 1945 dan yang diikrarkan dalam Sumpah Pemuda tanggal 28 Oktober 1928 sebagai bahasa persatuan yang dikembangkan sesuai dengan dinamika peradaban bangsa (Pemerintah Indonesia, 2019).

Bahasa daerah diberi batasan sebagai bahasa yang digunakan secara turun-temurun oleh warga negara Indonesia di daerah di wilayah Negara Kesatuan Republik Indonesia. Sementara itu, bahasa asing diberi batasan sebagai bahasa di Indonesia selain bahasa Indonesia dan bahasa daerah. Dalam Undang-Undang Nomor 24 Tahun 2009, baik bahasa daerah maupun bahasa asing, memegang fungsi pendukung bagi bahasa Indonesia. Sebagai pendukung, bahasa daerah dan

bahasa asing dapat digunakan apabila fungsi bahasa Indonesia tidak dapat dijalankan secara efektif (Pemerintah Indonesia, 2019).

2.2 Bahasa Muna

Bahasa Daerah Muna merupakan rumpun dari Bahasa Austronesia yang banyak digunakan di pulau Muna dan Buton, Sulawesi Tenggara. Selain sebagai alat berkomunikasi, penggunaan Bahasa Daerah Muna mencakup sebagian besar kegiatan kemasyarakatan, seperti upacara adat, kegiatan kebudayaan, keagamaan, dan juga menjadi bahasa pengantar di kelas-kelas tingkat permulaan sekolah dasar (Abdurrahman dkk., 2023).

Bahasa Muna memiliki dua puluh dialek, yaitu dialek Lohia dituturkan oleh masyarakat yang berada di Desa Lohia, Kecamatan Lohia, Kabupaten Muna dialek Sidamangura dituturkan oleh masyarakat yang berada di Desa Sidamangura, Kecamatan Kusambi, Kabupaten Muna Barat dialek Lasiwa dituturkan oleh masyarakat yang berada di Desa Lasiwa, Kecamatan Wakorumba Utara, Kabupaten Buton Utara dialek Labora dituturkan oleh masyarakat yang berada di Desa Lambelu, Kecamatan Pasi Kolaga, Kabupaten Muna dialek Lapadaku dituturkan oleh masyarakat yang berada di Kelurahan Lapadaku, Kecamatan Lawa, Kabupaten Muna Barat dialek Bente dituturkan oleh masyarakat yang berada di Desa Bente, Kecamatan Kabawo, Kabupaten Muna dialek Bone Tondo dituturkan oleh masyarakat yang berada di Desa Bone Tondo, Kecamatan Bone, Kabupaten Muna dialek Gala dituturkan oleh masyarakat yang berada di Desa Gala, Kecamatan Maginti, Kabupaten Muna Barat dialek Lambiku dituturkan oleh masyarakat yang berada di Desa Lambiku, Kecamatan Napabalano, Kabupaten Muna dialek Wasilomata dituturkan oleh masyarakat yang berada di Desa Wakambangura, Kecamatan Mawasangka, Kabupaten Buton Tengah dialek Lombe dituturkan oleh masyarakat yang berada di Kelurahan Bombonawulu, Kecamatan Gu, Kabupaten Buton Tengah dialek Siompu dituturkan oleh masyarakat yang berada di Kelurahan Talaga Satu, Kecamatan Talaga Raya, Kabupaten Buton Tengah dialek Todanga dituturkan oleh masyarakat yang berada di Desa Todanga, Kecamatan Kapontori, Kabupaten Buton dialek Gu-

Mawasangka dituturkan oleh masyarakat yang berada di Desa Nepa Mekar, Kecamatan Lakudo, Kabupaten Buton Tengah dialek Pancana dituturkan oleh masyarakat yang berada di Kelurahan Watumotobe, Kecamatan Kapontori, Kabupaten Buton dialek Lipu dituturkan oleh masyarakat yang berada di Kelurahan Lipu, Kecamatan Betoambari, Kabupaten Kota Bau Bau dialek Boneoge dituturkan oleh masyarakat yang berada di Kelurahan Boneoge, Kecamatan Lakudo, Kabupaten Buton Tengah dialek Kioko dituturkan oleh masyarakat yang berada di Kelurahan Lipu, Kecamatan Kulisusu, Kabupaten Buton Utara dialek Waara dituturkan oleh masyarakat yang berada di Desa Waara, Kecamatan Lohia, Kabupaten Muna dialek Oempu dituturkan oleh masyarakat yang berada di Desa Oempu, Kecamatan Tongkuno, Kabupaten Muna. Persentase perbedaan antardialek tersebut berkisar antara 51%-78 % (Sarmawati dkk., 2016).

Sementara itu, hasil penghitungan dialektometri menunjukkan bahwa isolek Muna merupakan bahasa dengan persentase perbedaan berkisar antara 81%-100% jika dibandingkan dengan bahasa-bahasa lainnya di Sulawesi Tenggara, misalnya dibandingkan dengan bahasa Kulisusu, Wolio, dan Moronene (Sarmawati dkk., 2016).

2.2.1 Dialek Bahasa Muna

Dialek bahasa daerah adalah variasi bahasa yang dituturkan oleh sekelompok masyarakat penutur bahasa tersebut dalam suatu wilayah tertentu. Faktor geografis dapat menyebabkan perbedaan dialek bahasa daerah karena adanya perbedaan lingkungan alam dan budaya di suatu wilayah. Misalnya, dialek bahasa daerah yang dituturkan di wilayah pegunungan akan berbeda dengan dialek bahasa daerah yang dituturkan di wilayah (Sarmawati dkk., 2016).

Faktor sosial juga dapat menyebabkan perbedaan dialek bahasa daerah. Misalnya, dialek bahasa daerah yang dituturkan oleh masyarakat pedesaan akan berbeda dengan dialek bahasa daerah yang dituturkan oleh masyarakat perkotaan. Faktor budaya juga dapat menyebabkan perbedaan dialek bahasa daerah. Misalnya, dialek bahasa daerah yang dituturkan oleh masyarakat suku A akan

berbeda dengan dialek bahasa daerah yang dituturkan oleh masyarakat suku B (Sarmawati dkk., 2016).

Dialek bahasa daerah Muna adalah variasi bahasa Muna yang dituturkan oleh masyarakat penutur bahasa Muna dalam suatu wilayah tertentu di Pulau Muna dan pantai barat Pulau Buton, Sulawesi Tenggara. Perbedaan dialek bahasa daerah Muna dapat menyebabkan kebingungan atau kesalahpahaman antar penutur bahasa Muna. Oleh karena itu, penting untuk mempelajari perbedaan dialek bahasa daerah Muna agar dapat berkomunikasi dengan lebih efektif (Non, 2019).

Dialek Muna Barat dituturkan di wilayah barat Pulau Muna, yaitu di Kecamatan Soroako, Kecamatan Tongkaina, Kecamatan Wawonii Barat, Kecamatan Wakorumba Utara, dan Kecamatan Wakorumba Tengah. Dialek ini memiliki ciri khas yaitu pelafalan huruf "a" yang diucapkan dengan lafal "e". (Yunus dkk., 2022).

2.3 *Preprocessing*

Preprocessing adalah salah satu langkah utama dalam *Neural Machine Translation* (NMT). Jika *preprocessing* tidak diterapkan pada *corpus* paralel, maka kualitas hasil terjemahan akan berkurang. *Preprocessing* dapat melakukan tokenisasi, menghilangkan kata-kata yang tidak penting, dan lain sebagainya. Persiapan teknik-teknik ini akan bergantung pada situasinya (Narasimha Raju dkk., 2021).

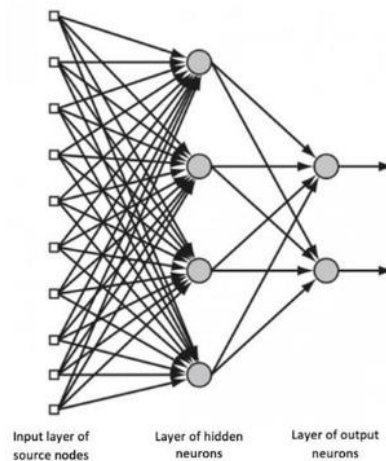
Preprocessing dapat menangani masalah replikasi data dalam terjemahan bahasa Indonesia ke bahasa Muna. Replikasi data akan membingungkan model saat melakukan terjemahan, sehingga menurunkan performa. Jadi, hapus data yang direplikasi dari *corpus* paralel. Untuk mengatasi masalah *Out of Vocabulary* (OOV), pada proses *preprocessing* menggunakan skema tokenisasi yang disebut *byte-pair coding* (BPE) selama *preprocessing*. BPE umumnya digunakan untuk kompresi data. Ini adalah jenis model subkata yang mencapai akurasi terjemahan yang lebih baik. BPE memungkinkan untuk menerjemahkan beberapa kata yang tidak ada pada waktu pelatihan (Zhang dkk., 2019).

2.4 *Deep Learning*

Bidang penelitian ilmu komputer yang dikenal sebagai deep learning baru saja muncul belakangan ini. Pembelajaran mendalam pada dasarnya adalah subbidang pembelajaran mesin yang memungkinkan komputer mempelajari atribut tanpa campur tangan manusia (Sorin dkk., 2020). Metode yang memungkinkan komputer memperoleh pengalaman dan pengetahuan untuk meningkatkan keterampilannya (Ian dkk., 2016). Deep learning terbukti lebih mudah beradaptasi dan mampu menghasilkan model yang lebih akurat daripada pembelajaran mesin tradisional (Sharma dkk., 2019). Ini didasarkan pada neuron otak. Jaringan saraf yang dalam, yang memiliki banyak lapisan dan banyak neuron di masing-masingnya, adalah nama lain untuk pembelajaran mendalam. Setiap lapisan mungkin juga mengandung ribuan neuron, dan jumlah lapisan dapat berkisar dari beberapa hingga ribuan.

Jaringan saraf merupakan teori pembelajaran yang mendalam. Deep learning dan jaringan saraf reguler pada dasarnya berbeda karena pembelajaran mendalam menggunakan lebih banyak neuron dan lapisan tersembunyi (Ravi dkk., 2017). Karena memiliki lebih banyak lapisan transformasi, banyak orang percaya bahwa algoritma pembelajaran mendalam lebih unggul daripada jaringan saraf biasa. Data juga dapat diubah dari non-linier menjadi linier menggunakan deep learning.

Setiap lapisan tersembunyi dalam jaringan saraf bertugas melatih fitur-fitur khusus berdasarkan hasil jaringan sebelumnya (Ian dkk., 2016). Semakin banyak lapisan tersembunyi, semakin abstrak dan rumit algoritma ini. Hirarki berlapis-lapis sederhana pada tingkat tinggi atau banyak lapisan membentuk jaringan saraf pembelajaran yang mendalam. Karena merepresentasikan data sebagai konsep hierarki bersarang, di mana setiap konsep yang didefinisikan memiliki hubungan dengan konsep yang lebih sederhana, pembelajaran mendalam memiliki banyak 7 kekuatan dan fleksibilitas (Ian dkk., 2016). Ilustrasi bagaimana deep learning menggunakan diagram konsep untuk merepresentasikan suatu masalah ditunjukkan pada Gambar 2.1.



Gambar 2.1 Ilustrasi *Deep Learning* (Razsiah., 2023)

2.5 *Natural Language Processing (NLP)*

Natural Language Processing (NLP) adalah bidang studi yang mempelajari atau mengembangkan algoritma atau sistem yang dapat dipahami oleh komputer untuk melakukan berbagai perintah yang berhubungan dengan bahasa manusia (Sintoris dan Vergidis, 2017).

Linguistik komputasi, pemrosesan bahasa, dan teknologi yang terkait dengan bahasa manusia semuanya adalah nama untuk NLP. NLP sering digunakan untuk analisis bahasa manusia tertulis dan lisan. NLP sering digunakan dalam membangun beberapa aplikasi seperti mesin penerjemah, *chatbot* dan beberapa aplikasi lain yang berkaitan dengan pemrosesan bahasa manusia. NLP sangat membutuhkan mesin yang dapat membaca, memahami ataupun mengambil sebuah informasi dari bahasa manusia yang tidak terstruktur dan abstrak bagi mesin.

NLP bertujuan untuk mengubah bahasa manusia menjadi data yang lebih terstruktur (Sorin dkk., 2020). Berikut merupakan beberapa teknik dasar NLP dalam memecahkan berbagai permasalahan:

1. Segmentasi

Kalimat Segmentasi kalimat merupakan sebuah teknik yang digunakan untuk mendefinisikan sebuah unit pada pemrosesan yang terdiri dari satu

atau lebih kata. Biasanya teknik ini melibatkan proses yang mengidentifikasi batas-batas dari sebuah kalimat antar kata dalam sebuah kalimat yang berbeda (Nitin dan Fred, 2010).

2. Tokenisasi

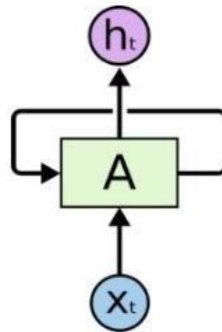
Tokenisasi adalah teknik dasar NLP. Metode ini memecah kalimat menjadi klausa yang mengandung kata atau frasa (Sun dkk., 2017).

3. Penghapusan

Kata Henti Penghapusan kata berhenti meningkatkan akurasi dengan menghilangkan kata-kata yang mengandung sedikit informasi (Méndez dkk., 2006) Kata-kata berhenti biasanya disebut sebagai kata-kata non-informatif. Dalam bahasa Indonesia, 6 misalnya, kata berhenti biasanya disebut dengan konjungsi seperti "dan", "atau", "yang", dan "kepada". Karena hanya berfokus pada informasi penting, mengurangi jumlah kata yang tidak diperlukan dapat mempercepat pemrosesan dan meningkatkan nilainya.

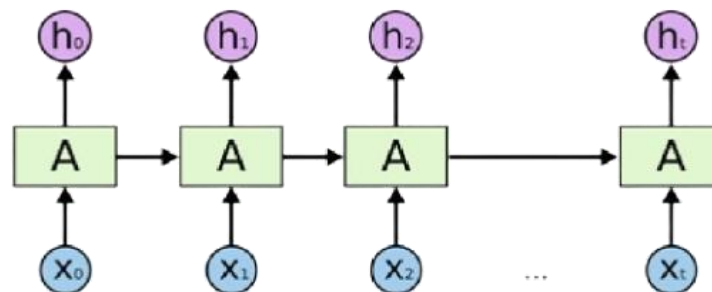
2.6 *Recurrent Neural Network (RNN)*

Algoritma *Recurrent Neural Network* merupakan salah satu kategori dari *deep learning*. *Recurrent Neural Network* merupakan model yang meniru cara berpikir dalam pengambilan keputusan tersebut dimana RNN tidak membuang begitu saja informasi dari masa lalu dalam proses pembelajarannya yang secara otomatis informasi dari masa lalu tetap tersimpan. RNN memproses input secara sekuensial. Data sekuensial mempunyai karakteristik dimana sampel diproses dengan suatu urutan, dan suatu sampel dalam urutan mempunyai hubungan yang erat satu dengan yang lain (Razsiah, 2023).



Gambar 2.2 Recurrent Neural Network (RNN)

Pada gambar diatas, x_t merupakan *input*-an terhadap t (waktu/urutan *input*an berdasarkan waktu/*input*-an data ke- t) h_t merupakan hasil *output*. Gambar diatas juga menggambarkan bahwa proses pada RNN dilakukan secara berulang-ulang sehingga data pada *input*-an sebelumnya tersimpan ke memori RNN. *Unfolding Recurrent Neural Network* dapat dilihat pada Gambar 2.3 *Unfolding Recurrent Neural Network*.



Gambar 2.3 Unfolding Recurrent Neural Network

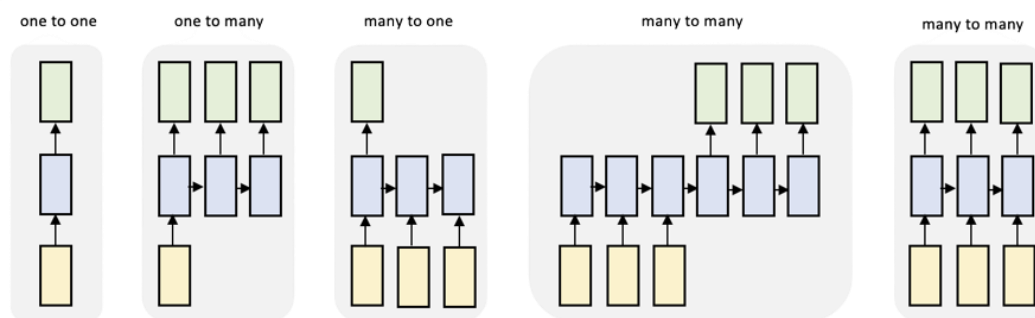
RNN merupakan bagian dari *Neural Network* sehingga lapisan-lapisan dari RNN dibagi menjadi beberapa bagian yaitu:

1. *Input Layer*
Merupakan sebuah lapisan yang menerima beberapa *input* yang kemudian akan diteruskan ke neuron lainnya pada jaringan.
2. *Hidden Layer*
Merupakan sebuah lapisan yang tersembunyi, *hidden layer* sendiri berfungsi untuk meningkatkan kemampuan jaringan dalam memecahkan beberapa masalah.

3. *Output Layer*

Output layer merupakan lapisan yang menghasilkan *output* dari hasil proses.

Gambar 2.3 memperlihatkan bahwa arsitektur RNN memproses data dari input secara satu persatu, pada waktu berikutnya *hidden layer* akan mengirim data menuju *hidden layer* lainnya. Proses ini akan berjalan terus secara satu persatu. Berikut Gambar 2.4 yang menggambarkan beberapa proses pada RNN dalam melakukan pemrosesan.



Gambar 2.4 Pemrosesan RNN

Berikut adalah penjelasan dari Gambar 2.4 sebagai berikut:

1. *One to One*

Model ini akan memproses data pada *input* dan *output* dengan cara satu persatu.

2. *One to Many*

Model ini akan memproses satu data *input* tetapi mampu menghasilkan banyak keluaran *output*.

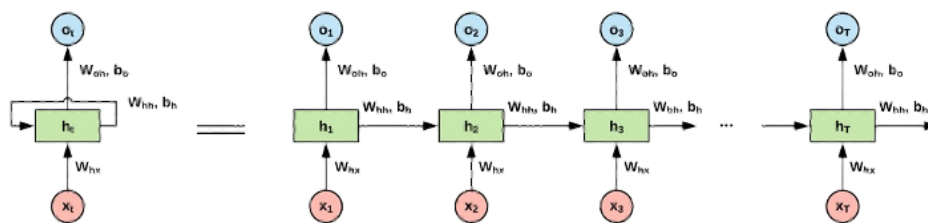
3. *Many to One*

Berbeda dengan sebelumnya model ini menerima banyak data *input* tetapi hanya menghasilkan satu keluaran atau *output*.

4. *Many to Many*

Model ini akan memproses data *input* dan *output* dengan banyak dan diproses secara berurutan atau sekuensial, model ini hanya akan menghasilkan keluaran atau *output* ketika semua data selesai di *input*.

Pada proses melakukan *training* pada RNN hampir sama dengan proses *training* pada *Neural Network* lainnya, yaitu terdapat langkah-langkah utama dalam melakukan proses *training*. Pertama, proses untuk membuat prediksi atau sering disebut dengan proses *forward pass*. Proses ini dilakukan dengan melakukan perhitungan untuk setiap nilai *hidden state* (h_t) yang didapatkan dari setiap masukkan pada (x_t) dan nilai bobot yang telah ditentukan sebelumnya. Setelah menentukan nilai untuk *hidden state*, selanjutnya akan melakukan perhitungan untuk keluaran atau *output* dari hasil prediksi (z_t). Kedua, melakukan perbandingan untuk hasil prediksi (z_t) dengan nilai *output* yang telah ditargetkan dengan menggunakan *loss function* yang menunjukkan hasil dari nilai prediksi yang jauh dari *output* yang telah ditargetkan. *loss function* juga dapat memberikan kesimpulan seberapa baik atau buruknya kinerja yang dihasilkan oleh RNN. Ketiga, setelah berhasil mendapatkan nilai dari *loss function* maka, akan dilakukan proses *Backpropagation Through Time* (BPTT) yang bertujuan untuk menghitung nilai *gradien* untuk setiap waktu yang dibutuhkan dalam proses pada jaringan. Proses ini juga dilakukan untuk mencari nilai pada bobot-bobot dan juga bias yang dihasilkan lebih baik dari proses yang dilakukan sebelumnya. Setelah proses ini selesai, selanjutnya akan dilakukan pembaruan bobot dan juga bias dengan menggunakan metode *Stochastic Gradient Descent* (SGD). Berikut Gambar 2.5 yang menggambarkan struktur pada RNN.



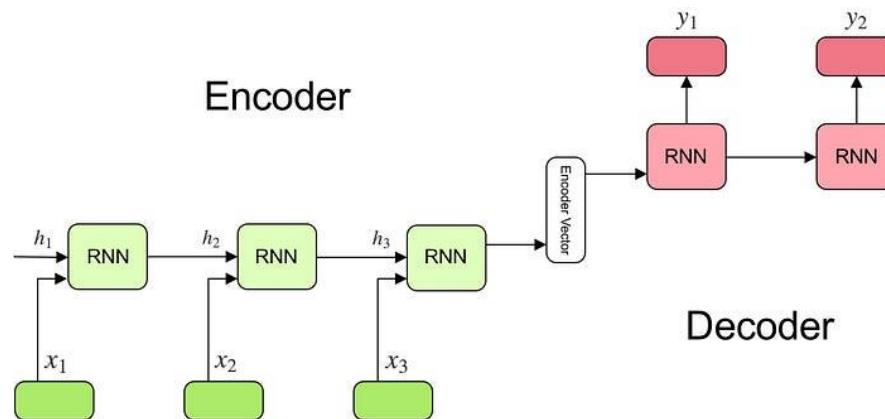
Gambar 2.5 Struktur RNN

2.7 Neural Machine Translation (NMT)

Neural Machine Translation (NMT) merupakan metode translasi yang menggunakan RNN dengan *encoder* dan *decoder*. Pada saat ini metode yang sering digunakan yaitu metode *attention* atau perhatian (Bahdanau dkk., 2015).

Pada saat melakukan proses terjemahan menggunakan RNN terlebih dahulu terhubung dengan *encoder* dan *decoder*.

Pada *encoder* yang diterapkan RNN akan memproses kata ataupun frasa pada sumber bahasa menjadi sebuah kalimat yang lebih mudah dimengerti oleh mesin. Sedangkan untuk *decoder* RNN akan mengambil *output* dari *encoder* pada RNN untuk dijadikan sebagai sebuah *input*-an, kemudian dapat menghasilkan hasil terjemahan dari kata ke kata (Abidin., 2017). Arsitektur pada model mesin penerjemah juga dapat dikatakan model *encoder-decoder* (Papineni dkk., 1922). Pada Gambar 2.1 menampilkan bagaimana sebuah model *encoder* dan *decoder* saling terhubung.



Gambar 2.6 Model *Encoder-Decoder*

Model ini terdiri dari 3 bagian *encoder*, *vektor encoder* dan *decoder*.

1. *Encoder*

Tumpukan beberapa unit berulang (sel LSTM atau GRU untuk kinerja yang lebih baik) di mana masing-masing menerima satu elemen dari urutan *input*, mengumpulkan informasi untuk elemen itu dan menyebarkannya ke depan. Dalam masalah menjawab pertanyaan, urutan *input* adalah kumpulan semua kata dari pertanyaan. Setiap kata direpresentasikan sebagai x_i di mana i adalah urutan kata tersebut. Status tersembunyi h_i dihitung menggunakan rumus:

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t) \quad (2.1)$$

Keterangan:

- h_t = Keadaan tersembunyi pada waktu t
- $W^{(hh)}$ = Matriks untuk koneksi ke *recurrent*
- h_{t-1} = Keadaan tersembunyi pada langkah waktu sebelumnya
- $W^{(hx)}$ = Matriks untuk koneksi input
- x_t = Input pada waktu t

Rumus sederhana ini mewakili hasil dari jaringan saraf berulang biasa. Seperti yang Anda lihat, kami hanya menerapkan bobot yang sesuai ke keadaan tersembunyi sebelumnya h_{t-1} dan *vector input* x_t .

2. *Vector Encoder*

Vector Encoder adalah keadaan tersembunyi terakhir yang dihasilkan dari bagian *encoder* model. Itu dihitung menggunakan rumus di atas. *Vector* ini bertujuan untuk merangkum informasi untuk semua elemen input untuk membantu *decoder* membuat prediksi yang akurat. Ini bertindak sebagai keadaan tersembunyi awal dari bagian *decoder* model.

4. Decoder

Tumpukan beberapa unit berulang di mana masing-masing memprediksi *output* y_t pada langkah waktu t . Setiap unit berulang menerima keadaan tersembunyi dari unit sebelumnya dan menghasilkan dan menghasilkan serta keadaan tersembunyinya sendiri. Dalam soal tanya jawab, urutan keluaran adalah kumpulan semua kata dari jawabannya. Setiap kata direpresentasikan sebagai y_i di mana i adalah urutan kata itu. Setiap h_i status tersembunyi dihitung menggunakan rumus:

$$h_t = f(W^{(hh)}h_{t-1}) \quad (2.2)$$

Keterangan:

- h_t = Keadaan tersembunyi pada waktu t
- $W^{(hh)}$ = Matriks untuk koneksi ke *recurrent*
- h_{t-1} = Keadaan tersembunyi pada langkah waktu sebelumnya
- f = Fungsi aktivitas untuk menghitung nilai tersembunyi

2.7.1 Rumus Matematika Model NMT

Rumus perhitungan matematika dari model NMT ini menggunakan arsitektur RNN yang didalamnya mencakup *encoder* dan *decoder*.

Encoder (RNN):

1. *Embedding Layer*:

- Representasi *embedding* untuk setiap kata dalam urutan sumber.

$$H_i = \text{Embed}(x_i), \text{ untuk } i = 1, 2, \dots, T_x \quad (2.3)$$

2. *Recurrent Layer*:

- Meneruskan representasi melalui lapisan *recurrent*.

$$h_i = \text{RNN}(H_i, h_{i-1}) \quad (2.4)$$

Decoder (RNN):

1. *Embedding Layer*:

- Representasi *embedding* untuk setiap kata dalam ukuran target.

$$S_i = \text{Embed}(y_i), \text{ untuk } i = 1, 2, \dots, T_y \quad (2.5)$$

2. *Recurrent Layer*:

- Meneruskan representasi melalui lapisan *recurrent*.

$$s_i = \text{DecoderRNN}(S_i, s_{i-1}) \quad (2.6)$$

3. *Attention Mechanism*:

- Menggunakan *attention mechanism* untuk memperhatikan konteks dari urutan sumber.

$$C_i = \text{Attention}(s_i H) \quad (2.7)$$

4. *Linear and Softmax*:

- Menggunakan *layer linear* dan fungsi *softmax* untuk menghasilkan distribusi probabilitas.

$$P_i = \text{Softmax}(\text{Linear}(s_i, C_i)) \quad (2.8)$$

2.7.2 Perhitungan Manual NMT

Contoh kasus dari perhitungan manual *Neural Machine Translation* (NMT) dengan dengan dua bahasa, yaitu sumber bahasa dan target bahasa. Dalam contoh kasus ini akan menggunakan kalimat sederhana sebagai contoh:

1. Data Training

Bahasa Sumber (*Source Language*): "Hello, how are you?"

Bahasa Target (*Target Language*): "Halo, apa kabar?"

Pada contoh kasus ini akan dilakukan perhitungan manual untuk beberapa parameter model NMT, seperti bobot (*weight*) dan nilai aktivasi (*activation value*) pada lapisan tersembunyi (*hidden layer*).

2. Parameter Bobot NMT

- Bobot (Weight) antara input dan hidden layer (W1):

$$W1 = \begin{bmatrix} 0.2 & 0.4 & 0.1 \\ 0.1 & 0.3 & 0.6 \\ 0.4 & 0.2 & 0.2 \\ 0.3 & 0.5 & 0.7 \\ 0.5 & 0.2 & 0.8 \end{bmatrix}$$

- Bias pada hidden layer (b1):

$$[0.1, 0.2, 0.3]$$

- Bobot antara hidden layer dan output layer (W2):

$$W2 = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.5 & 0.1 & 0.7 \\ 0.2 & 0.4 & 0.6 \end{bmatrix}$$

- Bias pada output layer (b2):

$$[0.2, 0.1, 0.3]$$

3. Langkah-Langkah Perhitungan

- *Embedding Input:*

$$x = \begin{bmatrix} 0.4 & 0.3 & 0.2 & 0.8 \\ 0.7 & 0.5 & 0.6 & 0.4 \\ 0.2 & 0.1 & 0.7 & 0.3 \\ 0.6 & 0.9 & 0.1 & 0.7 \\ 0.8 & 0.2 & 0.4 & 0.5 \end{bmatrix}$$

- *Hidden Layer:*

$$H = \sigma(X \cdot W_1 + b_1), \text{ Di mana } \sigma \text{ adalah fungsi aktivitas seperti } \textit{sigmoid}.$$

$$H = \sigma(X \cdot W_1 + b_1) =$$

$$\sigma \left(\begin{bmatrix} 0.4 & 0.3 & 0.2 & 0.8 \\ 0.7 & 0.5 & 0.6 & 0.4 \\ 0.2 & 0.1 & 0.7 & 0.3 \\ 0.6 & 0.9 & 0.1 & 0.7 \\ 0.8 & 0.2 & 0.4 & 0.5 \end{bmatrix} \cdot \begin{bmatrix} 0.2 & 0.1 & 0.4 \\ 0.1 & 0.3 & 0.6 \\ 0.4 & 0.2 & 0.2 \\ 0.3 & 0.5 & 0.7 \\ 0.5 & 0.2 & 0.8 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix} \right)$$

- *Output Layer:*

$O = \sigma(H \cdot W_2 + b_2)$, di mana σ adalah fungsi aktivitas pada *output layer*.

$$O = \sigma(H \cdot W_2 + b_2) =$$

$$\sigma = \left(\begin{bmatrix} 1.71 & 1.55 & 1.89 \\ 1.18 & 0.98 & 1.41 \\ 1.15 & 1.2 & 1.36 \\ 1.17 & 1.55 & 1.89 \end{bmatrix} \cdot \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.5 & 0.1 & 0.7 \\ 0.2 & 0.2 & 0.6 \end{bmatrix} + \begin{bmatrix} 0.2 \\ 0.1 \\ 0.3 \end{bmatrix} \right)$$

- *Softmax:*

Probabilitas = *softmax*(0), di mana *Softmax*(\cdot) menghitung probabilitas kelas.

$$\text{Probabilitas} = \text{softmax}(0) =$$

$$\begin{bmatrix} \frac{\exp(1.486)}{\exp(1.486) + \exp(1.293) + \exp(2.155)} \\ \frac{\exp(1.293)}{\exp(1.486) + \exp(1.293) + \exp(2.155)} \\ \frac{\exp(2.155)}{\exp(1.486) + \exp(1.293) + \exp(2.155)} \end{bmatrix}$$

Keterangan:

- X : Matriks *input embedding* untuk setiap kata dalam kalimat.
- W_1 : Matriks bobot antara *input* dan *hidden layer*.
- b_1 : Vektor bias pada *hidden layer*.
- H : Matriks aktivasi pada *hidden layer*.
- W_2 : Matriks bobot antara *hidden layer* dan *output layer*.
- b_2 : Vektor bias pada *output layer*.
- O : Matriks aktivasi pada *output layer*.
- *softmax*(\cdot)/*softmax*(\cdot): Fungsi *softmax* untuk menghitung probabilitas.

Perhitungan manual di atas merupakan perhitungan manual *Neural Machine Translation* dengan contoh kasus menerjemahkan kalimat “*Hello, how are you?*” menjadi “Halo, apa kabar?”. Kata-kata dalam kalimat “*Hello, how are you?*” diubah menjadi vektor *embedding* menggunakan model *embedding*. Pada contoh kasus ini mendapatkan matriks *embedding* berikut untuk setiap kata:

Embedding *Hallo, how are you?*:

- “*Hello*”: [0.4, 0.7, 0.2, 0.6, 0.8]
- “*how*”: [0.3, 0.5, 0.1, 0.9, 0.2]
- “*are*”: [0.2, 0.6, 0.7, 0.1, 0.4]
- “*you*”: [0.8, 0.4, 0.3, 0.7, 0.5]

Embedding *Halo, apa kabar?*:

- “*Halo*”: [0.6, 0.8, 0.4, 0.9, 0.7]
- “*apa*”: [0.2, 0.5, 0.3, 0.6, 0.1]
- “*kabar*”: [0.3, 0.4, 0.7, 0.5, 0.2]

Pada *hidden layer* matriks *embedding input* dengan matriks bobot $W1$ antara *input* dan *hidden layer*, dan menambahkan vektor bias $b1$. Kemudian, aktivasi dari *hidden layer* dihitung menggunakan fungsi aktivasi tertentu misalnya, *sigmoid* atau ReLU, kemudian menggunakan *sigmoid* untuk fungsi aktivasi. Hasil perhitungan aktivasi pada *hidden layer* adalah sebagai berikut:

- Aktivasi pada *hidden layer* 1: $[0.4, 0.7, 0.2, 0.6, 0.8] * [0.2, 0.1, 0.4, 0.3, 0.5] + [0.1, 0.2, 0.3] = [0.87, 0.78, 0.88]$
- Aktivasi pada *hidden layer* 2: $[0.3, 0.5, 0.1, 0.9, 0.2] * [0.4, 0.3, 0.2, 0.5, 0.2] + [0.1, 0.2, 0.3] = [0.82, 0.84, 0.89]$
- Aktivasi pada *hidden layer* 3: $[0.2, 0.6, 0.7, 0.1, 0.4] * [0.1, 0.6, 0.2, 0.7, 0.8] + [0.1, 0.2, 0.3] = [0.75, 0.85, 0.91]$
- Aktivasi pada *hidden layer* 4: $[0.8, 0.4, 0.3, 0.7, 0.5] * [0.3, 0.5, 0.7, 0.2, 0.8] + [0.1, 0.2, 0.3] = [0.95, 0.94, 0.97]$

Ketika probabilitas dihitung, hasilnya akan memberikan prediksi kelas untuk kalimat terjemahan. Dalam contoh ini, kata "Halo" memiliki probabilitas tertinggi, sehingga menjadi kata pertama dalam terjemahan, dan seterusnya. Dengan demikian, kita mendapatkan terjemahan "Halo, apa kabar?" dari kalimat "*Hello, how are you?*".


2.8 *Android*

Android adalah sistem operasi untuk perangkat *mobile* berbasis Linux yang menawarkan aplikasi, sistem operasi, *middleware*, dan *framework*. Selain itu *Android* menyediakan *platform* bagi pengembang untuk membuat aplikasi mereka sendiri dan menawarkan semua *tools* dan *framework* yang diperlukan untuk proses pengembangan aplikasi yang cepat dan mudah. Dengan bantuan *Android Software Development Kit* (SDK), pengembang aplikasi dapat menggunakan bahasa pemrograman untuk membuat aplikasi pada *platform Android* (Utami dkk., 2016).

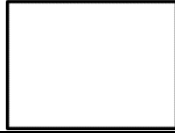

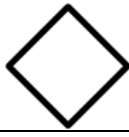
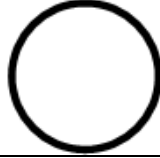




2.9 *Flowchart*

Ada dua *tool* yang sering digunakan untuk membantu menyusun dokumen logika pemrograman, yaitu *flowchart* dan *pseudocode* (kode semu). *Flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang sering terhubung. Jadi, setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya. Simbol-simbol *flowchart* adalah standar yang ditentukan oleh *American National Standard Institute Inc* (Karenina dan Ela, 2021).

Tabel 2.1 Simbol-Simbol *Flowchart*

Nama Komponen	Deskripsi	Simbol
<i>Start/End</i>	Simbol yang mendefinisikan awal atau akhir suatu <i>flowchart</i> .	

Tabel 2.1 Simbol-Simbol *Flowchart* (Lanjutan)

Nama Komponen	Deskripsi	Simbol
<i>Process</i>	Simbol pemrosesan yang terjadi pada sebuah alur kerja.	
<i>Input/Output</i>	Simbol yang mendefinisikan masukan dan keluaran proses.	
<i>Decision</i>	Simbol untuk memutuskan proses lanjutan dari kondisi tertentu.	
<i>On-Page Reference</i>	Simbol konektor untuk menyambung proses pada lembar kerja yang sama.	
<i>Off-Page Reference</i>	Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda.	
<i>Flowline</i>	Simbol untuk menghubungkan antar proses atau antar simbol.	
<i>Database</i>	Simbol <i>database</i> atau basis data.	
<i>Preparation</i>	Simbol yang digunakan untuk memberi nilai awal suatu variabel.	

2.10 *Unified Modeling Language* (UML)





Unified Modeling Language (UML) adalah kerangka kerja dan metodologi terpadu untuk memodelkan desain program berorientasi objek (OOP) dan aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok alat yang mendukung pengembangan sistem tersebut. UML diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980. Saat ini,

UML mulai banyak digunakan oleh para praktisi OOP. UML adalah dasar dari alat desain berorientasi objek IBM (Pakaya dkk., 2020).

2.11 Use Case diagram

Use Case diagram atau *use case* adalah salah satu dari berbagai jenis diagram UML yang menggambarkan hubungan interaksi antara sistem dan *actor*. *Use case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dan sistemnya (Pakaya dkk., 2020).

Tabel 2.2 Simbol-Simbol Use Case Diagram

Nama Komponen	Deskripsi	Simbol
<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i>	
<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri	
<i>Association</i>	Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki hubungan interaksi dengan <i>actor</i>	
<i>Generalization</i>	Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>use case</i>	



Tabel 2.2 Simbol-Simbol *Use Case Diagram* (Lanjutan)

Nama Komponen	Deskripsi	Simbol
<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya	— <<include>> →
<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi	— <<extend>> →




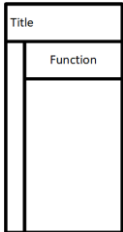
2.12 *Activity Diagram*

Activity diagram atau diagram aktivitas adalah diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem, Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *use case* yang memiliki alur aktivitas. Alur atau aktivitas berupa runtutan menu-menu atau proses bisnis yang terdapat di dalam sistem tersebut (Pakaya dkk., 2020).

Tabel 2.3 Simbol-Simbol *Activity Diagram*

Nama Komponen	Deskripsi	Simbol
Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal	
Aktivitas	Aktivitas yang dilakukan sistem, aktivitasi biasanya diawali dengan kata kerja	


Tabel 2.3 Simbol-Simbol *Activity Diagram* (Lanjutan)

Nama Komponen	Deskripsi	Simbol
<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu	
<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu	
Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir	
<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi	







2.13 *Sequence Diagram*

Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu, *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim beserta waktu pelaksanaannya (Pakaya dkk., 2020).

Tabel 2.4 Simbol-Simbol *Sequence Diagram*

Nama Komponen	Deskripsi	Simbol
Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri	

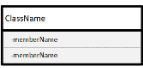

Tabel 2.4 Simbol-Simbol *Sequence Diagram* (Lanjutan)

Nama Komponen	Deskripsi	Simbol
<i>Lifeline</i>	Menyatakan kehidupan suatu objek	
Objek	Menyatakan objek yang berinteraksi pesan	
Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi	
Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya	
Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi	
Pesan tipe <i>self</i>	Menyatakan bahwa suatu objek mengirimkan pesan ke objek itu sendiri	




2.14 *Class Diagram*

Class diagram atau diagram kelas adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode dan hubungan dari setiap objek (Pakaya dkk., 2020).

Tabel 2.5 Simbol-Simbol *Class Diagram*

Nama Komponen	Deskripsi	Simbol
Kelas	Kelas pada struktur sistem	
Asosiasi/ <i>association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>	

Tabel 2.5 Simbol-Simbol *Class Diagram* (Lanjutan)

Nama Komponen	Deskripsi	Simbol
Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)	
Kebergantungan/ <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas	
Agregasi/ <i>aggregation</i>	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)	

2.15 Metodologi Pengembangan Sistem

Rational Unified Process (RUP) merupakan salah satu metode pengembangan perangkat lunak dengan menggunakan pendekatan yang disiplin dalam melakukan setiap tugas dan tanggung jawabnya dalam sebuah organisasi. Tujuan dari RUP adalah untuk dapat menjamin produksi berkualitas tinggi dan memenuhi semua kebutuhan pihak yang berkepentingan, termasuk waktu dan biaya sesuai dengan rencana yang telah disepakati (Yordan dkk., 2019). Adapun 4 tahapan kerja dari RUP, yaitu:

1. Fase *Inception* (Permulaan)

Fase *inception* adalah tahap pertama dalam pengembangan RUP. *Inception* dilakukan bertujuan untuk dilakukannya serangkaian analisis kebutuhan dalam sistem yang nantinya akan dikembangkan. Hasil dari fase ini pada pengembangan sistem yang dilakukan adalah pemodel proses bisnis *as-is* dan *to-be*, pemodelan *use case*, dan analisis persyaratan.

2. Fase *Elaboration* (Perluasan/Perencanaan)

Fase *elaboration* adalah fase melakukan perancangan atau arsitektur desain dari sistem yang akan dikembangkan. Hasil dari fase ini nantinya berupa *class diagram*, *sequence diagram*, perancangan antarmuka dan *physical data model*.

3. Fase *Construction* (Konstruksi)

Fase *construction* adalah fase melakukan pengimplementasian sistem berdasarkan perancangan yang sudah dilakukan sebelumnya dalam fase *elaboration* dan dilakukan pengujian terhadap sistem.

4. Fase *Transition* (Transisi)

Fase *transition* adalah fase melakukan *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

2.16 Pengujian *Black Box*

Pengujian *black box* adalah pengujian aspek fundamental dari suatu sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan baik. Pengujian *black box* merupakan metode perancangan data pengujian berdasarkan spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi dalam perangkat lunak dan kemudian keluaran perangkat lunak diperiksa apakah sesuai dengan yang diharapkan. Pengujian *black box* mencoba untuk menemukan kesalahan dalam kategori (Susilawati dkk., 2022):

1. Fungsi yang salah atau hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan terminasi.

Pengujian *black box* memiliki beberapa tipe, salah satunya adalah tipe pengujian fungsional. Basis uji dari pengujian fungsional ini adalah pada spesifikasi dari komponen perangkat lunak yang akan diuji. Pengujian fungsional berkonsentrasi pada hasil dari proses, bukan bagaimana prosesnya terjadi (Susilawat dkk., 2022).

2.17 Pengujian *Bilingual Evaluation Understudy* (BLEU) Score

Bilingual Evaluation Understudy (BLEU) merupakan sebuah pengujian yang banyak digunakan pada sebuah mesin penterjemah. Matriks BLEU dirancang untuk mengukur seberapa dekat keluaran yang dihasilkan dengan melakukan pencocokan panjang frasa variabel keluaran dari mesin penterjemah dengan referensi terjemahan. Matriks dasar memerlukan sebuah kalkulasi *brevity penalty*.

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan untuk mengumpulkan data berupa dalam penyusunan tugas akhir ini. Penulisan tugas akhir ini menggunakan beberapa metode pengumpulan data yaitu:

1. Kepustakaan, dilakukan dengan cara mengumpulkan referensi yang berkaitan dengan cara menggunakan *Neural Machine Translation* (NMT) dan cara mengembangkan modelnya. Selain itu dilakukan juga pengumpulan referensi mengenai arsitektur Recurrent Neural Network.
2. Studi literatur, digunakan untuk pengumpulan data. Data yang digunakan berupa *corpus* paralel yang berisikan kalimat bahasa Indonesia dan bahasa Muna yang di ambil dari kamus bahasa Muna-Indonesia.
3. Wawancara, dilakukan untuk menguji keakuratan hasil terjemahan model yang telah dikembangkan baik secara penulisan maupun penyusunan katanya.

3.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam sistem ini adalah metode *Rational Unified Process* (RUP). Tahap ini memiliki empat fase dalam pengembangan sistem, yaitu:

1. *Inception* (Permulaan)

Pada fase ini dilakukan proses pengidentifikasian aplikasi, dilakukan dengan analisis kebutuhan akan aplikasi, melakukan kajian terhadap penelitian yang meliputi pengenalan *Bilingual Evaluation Understudy* (BLEU) Score dan algoritma *Neural Machine Translation* (NMT).

2. *Elaboration* (Perluasan/perencanaan)

Pada tahap *elaboration* dilakukan proses analisis yaitu tahap ditentukan *Unified Modeling Language* (UML) yang terdiri dari *use case diagram*, *sequence diagram* dan *activity diagram* serta membuat *flowchart* dan analisis algoritma

yang digunakan yaitu *Neural Machine Translation* (NMT). Pada tahap ini dilakukan juga desain arsitektur aplikasi.

3. *Construction* (Konstruksi)

Proses yang dilakukan dalam tahap ini ialah membangun perangkat lunak secara utuh, mulai dari interface aplikasi dan *coding* mulai dari *preprocessing*, *rescale*, *recognize* hingga menanamkan model kedalam aplikasi *android* yang telah dibuat.

4. *Transition* (Transisi)

Fase *transition* difokuskan untuk melakukan proses *deployment*, untuk memastikan sistem sudah bekerja dengan baik di lingkungan pengguna. Dalam penelitian ini, sistem dipakai untuk melakukan pengujian terhadap aplikasi serta memperbaiki segala masalah yang muncul selama pengujian.

3.3 Waktu dan Tempat Penelitian

3.3.1 Waktu Penelitian

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Maret sampai dengan Juni 2024. Rincian kegiatan penelitian hingga pengembangan aplikasi dapat dilihat pada Tabel 3.1 berikut:

Tabel 3. 1 Gantt Chart Waktu Penelitian

No	Uraian	Waktu (2024)															
		Maret				April				Mei				Juni			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	<i>Inception</i>																
2	<i>Elaboration</i>																
3	<i>Construction</i>																
4	<i>Transition</i>																

3.3.2 Tempat Penelitian

Penelitian tugas akhir ini dilakukan di Laboratorium *Computer Scinence & Artificial Intelligence*, Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo.

3.4 Analisis Kebutuhan Sistem

Analisis sistem merupakan tahapan yang bertujuan untuk mengetahui dan mengamati semua komponen pada suatu sistem. Pada analisis sistem ini yaitu analisis kebutuhan system melakukan analisis yang meliputi kebutuhan perangkat keras dan perangkat lunak yang digunakan serta bahasa pemodelan UML, perancangan *interface* atau tampilan.

3.5 Hardware dan Software

3.5.1 Hardware

Hardware atau perangkat keras yang digunakan untuk mengembangkan model dan aplikasi penerjemah bahasa Muna ke bahasa Indonesai, dibutuhkan beberapa perangkat keras sebagai sarana untuk merealisasikan model dan aplikasi yang dibangun. Perangkat keras yang digunakan ditunjukkan pada Table 3.2.

Tabel 3.2 Kebutuhan Perangkat Keras

No.	Nama Perangkat	Spesifikasi
1.	Laptop	Acer Aspire 5
	➤ <i>Processor</i>	Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz
	➤ <i>Monitor</i>	15.6 Inch
	➤ <i>Memori</i>	Ram 4 GB, DDR4 Memory
	➤ <i>Harddisk</i>	500 GB HDD
2.	<i>Android Smartphone</i>	Versi OS Funtouch OS 12 Global

3.5.2 Software

Software atau perangkat lunak yang digunakan untuk mengembangkan model dan aplikasi yang telah dirancang sebelumnya dapat dilihat pada Tabel 3.3.

Tabel 3.3 Kebutuhan Perangkat Keras

No.	Nama Perangkat	Fungsi	Spesifikasi
1	<i>Windows</i>	<i>Operating System</i>	<i>Windows 11</i>
2	<i>Android Studio</i>	<i>Integrated Development Environment</i>	Versi 2022.3
3	<i>Visual Studio Code</i>	<i>Integrated Development Environment</i>	Versi 1.85.2
4	<i>Microsoft Edge</i>	<i>Web Browser</i>	Versi 122.2365.52
5	<i>Python</i>	Bahasa Pemrograman	Versi 3.10.4
6	<i>Tensorflow</i>	<i>Framework Python</i>	Versi 2.12.0
7	<i>Java</i>	Bahasa Pemrograman	Versi 1.8.20

3.6 Perancangan Sistem

3.6.1 Pengumpulan dan Persiapan Data

Untuk memastikan kualitas terjemahan yang optimal, langkah awal dalam perancangan sistem ini adalah mengumpulkan dataset terjemahan bahasa Muna ke bahasa Indonesia yang teranotasi dengan baik. Dataset ini kemudian dibagi menjadi set pelatihan, validasi, dan uji untuk melatih dan mengukur performa model. Keakuratan dan keragaman dataset sangat penting untuk hasil yang baik.

3.6.2 Preprocessing Data

Dalam tahap ini, setiap kalimat pada dataset dianalisis lebih lanjut. Kalimat-kalimat tersebut diterjemahkan menjadi token atau kata-kata terpisah atau *tokenisasi*, panjang kalimat disesuaikan untuk menciptakan homogenitas atau *padding*, dan kata-kata diubah menjadi representasi *vector* atau *embedding*.

Preprocessing aata ini membantu model memahami dan memproses informasi dengan lebih baik.

3.6.3 Pelatihan Model

Fase pelatihan model merupakan langkah penting dalam pengembangan sistem terjemahan ini. Dalam penelitian ini digunakan *categorical cross-entropy* untuk meminimalkan perbedaan antara distribusi probabilitas prediksi dan distribusi probabilitas sebenarnya. Pengoptimalan model menggunakan *Adaptive Moment Estimation* (Adam) sebagai *optimizer* dengan *learning rate* yang dioptimalkan agar dapat mengarahkan model untuk mencapai konvergensi dengan cepat. Selama pelatihan, model diuji dengan data validasi untuk memantau performanya. Hal ini diperlukan untuk mencegah *overfitting* yang dapat mengakibatkan model kurang mampu menggeneralisasi ke data baru. Monitoring performa juga melibatkan penyelidikan terhadap kurva pembelajaran untuk mengidentifikasi tingkat kekonvergenan yang optimal.

3.6.4 Evaluasi dan Penyesuaian

Setelah model dilatih, langkah selanjutnya adalah mengukur kualitas terjemahan menggunakan *BLEU score*. Pengukuran ini memberikan gambaran tentang seberapa baik model mampu menghasilkan terjemahan yang sesuai dengan terjemahan referensi. Hasil evaluasi digunakan untuk penyesuaian model, yang mencakup *fine-tuning* parameter dan mungkin penerapan teknik augmentasi data untuk meningkatkan keragaman. Proses ini bertujuan untuk menjamin bahwa model memiliki kemampuan generalisasi yang optimal dan dapat mengatasi variasi dalam data uji yang mungkin dihadapi.

3.6.5 Pengembangan Aplikasi

Setelah memastikan kualitas model terjemahan, berikutnya melakukan pengembangan aplikasi. Langkah ini melibatkan pengembangan antarmuka pengguna yang intuitif untuk meningkatkan pengalaman pengguna. Aplikasi dirancang untuk memberikan akses mudah dan cepat terhadap layanan

terjemahan. Model NMT kemudian diintegrasikan ke dalam aplikasi, memungkinkan pengguna untuk mengakses fungsi terjemahan langsung. Proses ini melibatkan pemilihan teknologi dan alat pengembangan yang optimal untuk memastikan aplikasi dapat berjalan dengan lancar dan responsif.

3.6.6 Optimasi dan Penyempurnaan

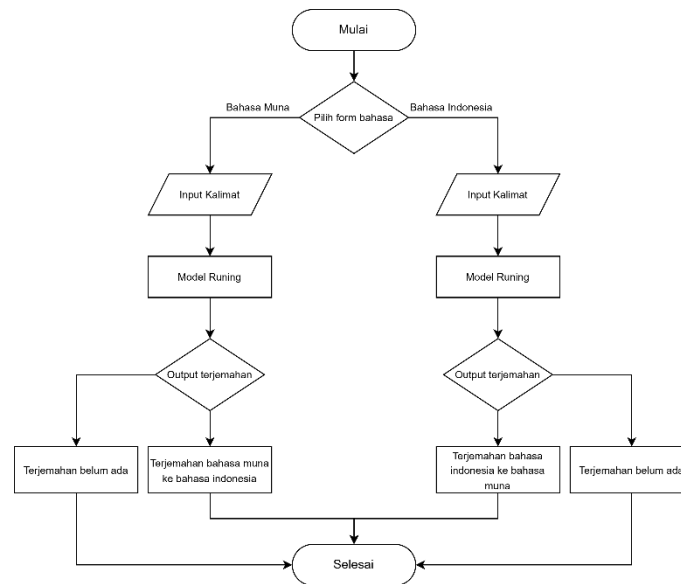
Setelah memastikan model dan sistem pada aplikasi penerjemah bahasa Muna ke bahasa Indonesia bekerja dengan baik, langkah selanjutnya adalah mengoptimalkan model dan aplikasi dengan melakukan pengujian *BLUE score* dan *Black Box testing* yang digunakan sebagai panduan untuk melakukan penyesuaian parameter model yang lebih lanjut dan menyempurnakan desain aplikasi. Proses ini melibatkan literasi dan eksperimen berkelanjutan untuk memastikan sistem terus berkembang dan memenuhi standar kualitas yang tinggi.

3.7 Perancangan *Flowchart*

Perancangan *flowchart* ini terdiri dari 2 *flowchart* yaitu *flowchart* aplikasi dan juga *flowchart* sistem atau model. Perancangan *flowchart* ini dapat dilihat pada Gambar 3.1 dan Gambar 3.2.

3.7.1 *Flowchart* Aplikasi

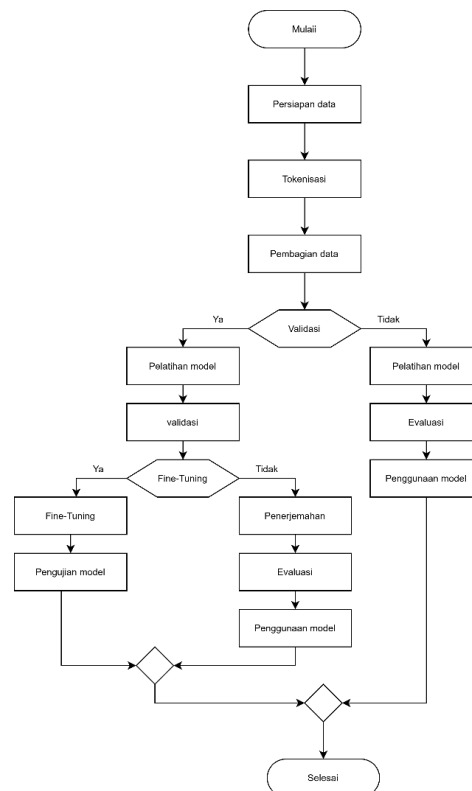
Flowchart aplikasi ini menunjukkan aktivitas pengguna dalam menjalankan aplikasi penerjemah bahasa Muna ke bahasa Indonesia yang menggunakan algoritma *Neural Machine Translation* dan menunjukkan beberapa kondisi tertentu saat menjalankan aplikasi.



Gambar 3.1 Flowchart Aplikasi

3.7.2 Flowchart Model

Flowchart model menunjukkan proses pelatihan data pada model terjemahan dengan algoritma *Neural machine translation*.



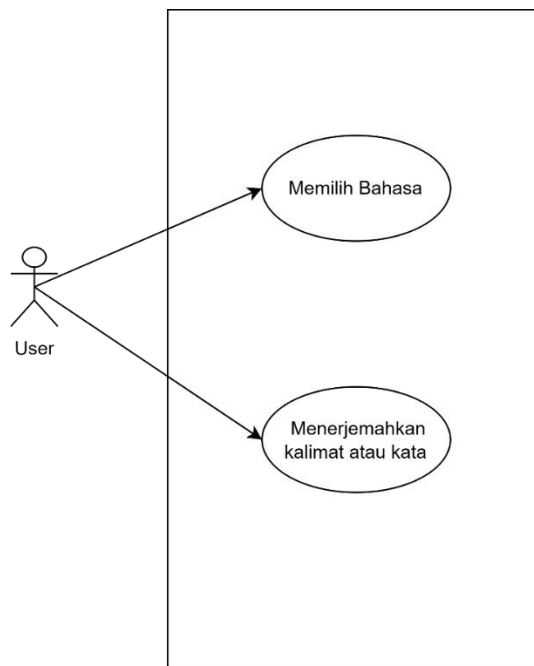
Gambar 3.2 Flowchart Model

3.8 Perancangan *Unified Modeling Language* (UML)

Perancangan UML ini menjelaskan berbagai macam aktifitas tertentu di dalam aplikasi yang di bangun dalam bentuk diagram, *Unified modeling language* ini terdiri dari *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*.

3.8.1 *Use Case Diagram*

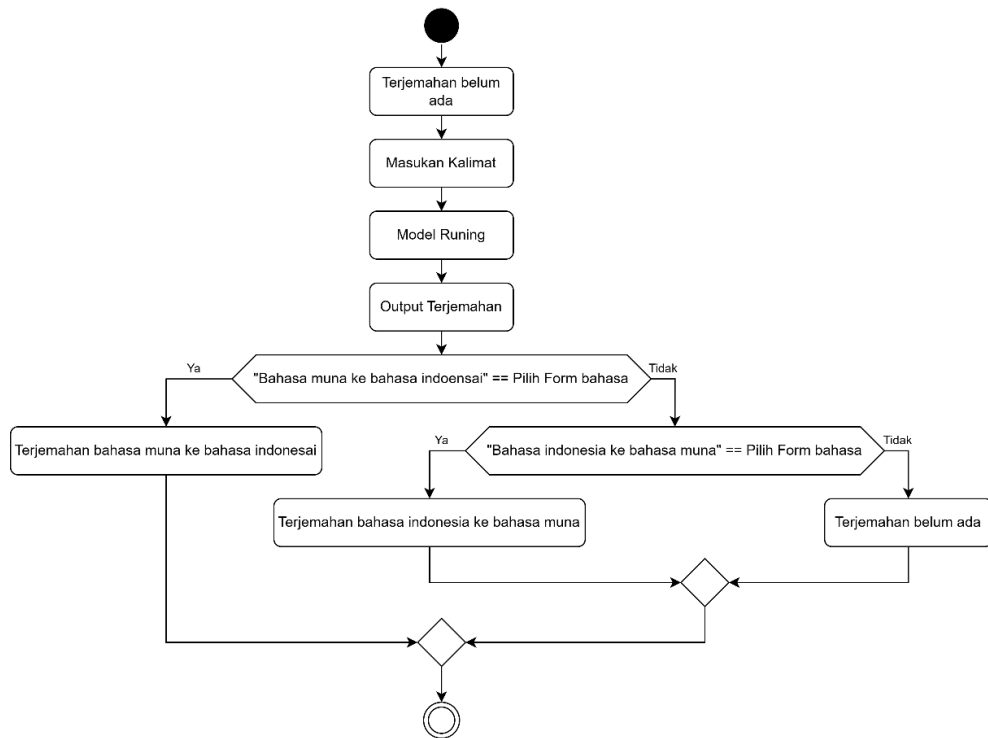
Use case diagram ini menunjukan aktivitas pengguna dalam menerjemahkan kalimat pada aplikasi penerjemah bahasa menggunakan *Neural Machine Translation*.



Gambar 3.3 *Use Case Diagram*

3.8.2 *Activity Diagram*

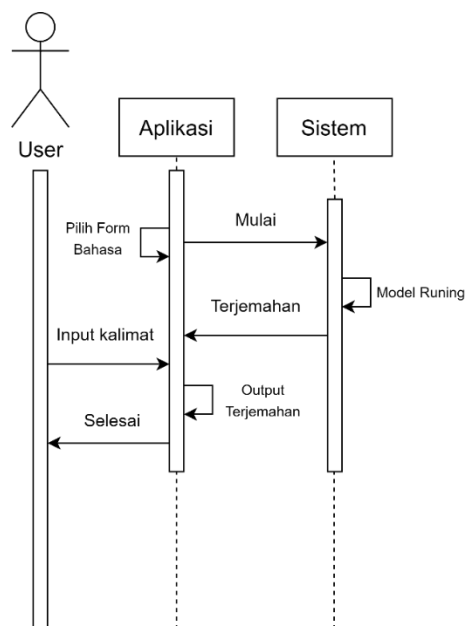
Activity diagram ini menunjukkan proses penerjemahan bahasa Muna ke bahasa Indonesia di dalam aplikasi mulai dari memasukan kalimat, proses penerjemahan yang dilakukan sistem, memberikan hasil terhemahan, hingga apa yang terjadi jika terjemahan belum tersedia.



Gambar 3.4 Activity Diagram

3.8.3 Sequence Diagram

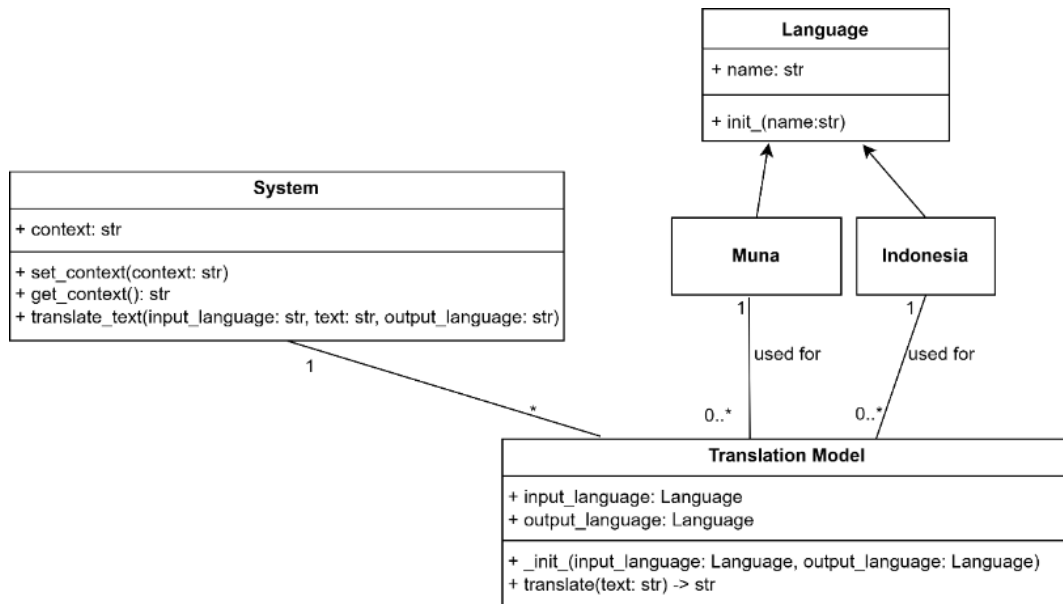
Sequence diagram ini menunjukkan urutan proses penerjemahan kalimat bahasa Muna ke bahasa Inonesia.



Gambar 3.5 Sequence Diagram

3.8.4 Class Diagram

Class diagram ini bertujuan untuk menggambarkan struktur dan hubungan antar kelas dalam sistem terjemahan.



Gambar 3.6 Class Diagram

3.9 Perancangan Interface

Desain *interface* merupakan antarmuka yang menghubungkan antara user dengan sistem. Setelah melakukan proses perancangan sistem, maka dilakukan perancangan *interface* dari aplikasi penerjemah bahasa Muna ke bahasa Indonesia ini. Adapun rancangan *interface* dari sistem ini adalah sebagai berikut:

1. Halaman *Splash Screen*

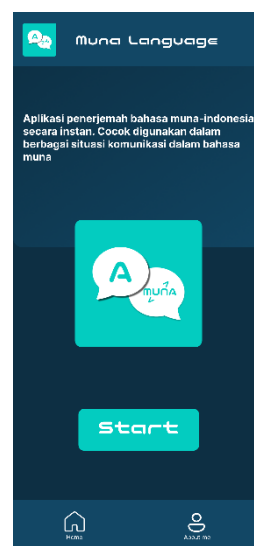
Halaman ini digunakan untuk memberikan informasi pendahuluan kepada pengguna saat mereka memulai aplikasi. Halaman ini akan tampil selama 2 detik dan akan langsung masuk ke halaman *dashboard*.



Gambar 3.7 Splash Screen

2. Halaman *Dashboard*

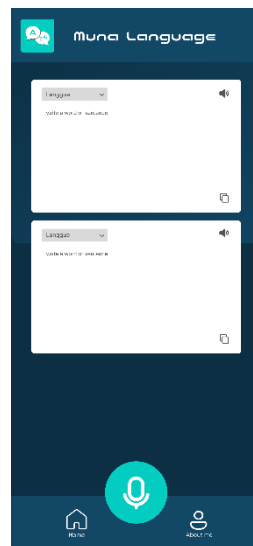
Halaman ini digunakan untuk pengguna untuk memulai fungsi utama aplikasih yaitu untuk menerjemahkan bahasa Muna dan bahasa Indonesia.



Gambar 3.8 Dashboard

3. Halaman *Translation language*

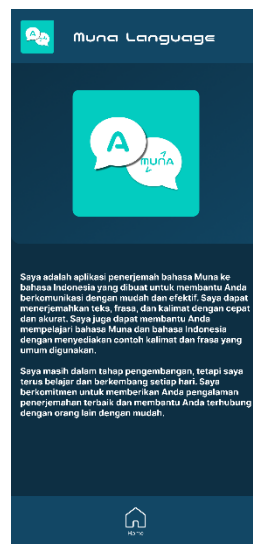
Halaman ini digunakan untuk melakukan proses penerjemahan bahasa Muna ke bahasa Indonesia ataupun bahasa Indonesia ke bahasa Muna



Gambar 3.9 Translation Language

4. Halaman *About Me*

Halaman ini berisikan penjelasan dan pengenalan mengenai aplikasi penerjemah bahasa Muna ke bahasa Indonesia.



Gambar 3.10 About Me

3.10 Pengujian *Black Box*

Pengujian perangkat lunak digunakan untuk mengetahui apakah perangkat lunak dapat berjalan dengan baik atau tidak. Salah satu pengujian perangkat lunak yang digunakan dalam penelitian ini adalah *black box*. *Black box testing* digunakan untuk mengetahui apakah program yang dibuat sudah sesuai dengan kebutuhan fungsionalnya atau tidak.

Tabel 3.4 Pengujian *Black Box*

No	Halaman yang Diuji	Aksi	Reaksi Sistem Yang Diharapkan
1.	<i>Splash Screen</i>	Membuka aplikasi	<i>Splash Screen</i> tampil \pm 2 detik, lalu masuk ke halaman <i>dashboard</i>
2.	<i>Dashboard</i>	Menekan tombol “Strat”	Berpidah ke halaman <i>Translation Language</i> untuk melakukan penerjemahan kalimat
3	<i>Translation Language</i>	Memasukan kalimat atau kata yang diinginkan	Model menampilkan hasil terjemahan kalimat atau kata yang diinginkan
4	<i>Dashboard</i>	Menekan tombol “About me”	Menampilkan informasi mengenai aplikasi
5	<i>About Me</i>	Menekan tombol “Home”	Berpindah ke halaman <i>Dashboard</i>

3.11 Pengujian *BLEU Score*

Pengujian *Bilingual Evaluation Understudy* (BLEU) *score* adalah metode evaluasi otomatis yang umum digunakan dalam penelitian terjemahan mesin untuk mengukur sejauh mana hasil terjemahan mendekati teks referensi manusia.

BLEU menghitung skor berdasarkan n-gram presisi dan memberikan bobot lebih tinggi pada n-gram yang lebih panjang.

Rumus *Bilingual Evaluation Understudy* (BLEU) *score* yang digunakan pada pengujian BLEU *score* ini dapat dilihat pada tabel berikut.

$$BP = \begin{cases} 1 & \text{if } \text{len}(\text{translation}) > \text{len}(\text{reference}) \\ e^{\left(1 - \frac{\text{len}(\text{reference})}{\text{len}(\text{translation})}\right)} & \text{if } \text{len}(\text{translation}) \leq \text{len}(\text{reference}) \end{cases} \quad (3.1)$$

$$n - \text{gram precision}_n = \frac{\text{count}(n\text{-grams terjemahan sesuai } n\text{-gram referensi})}{\text{count}(n\text{-grams pada translation})} \quad (3.2)$$

$$BLEU = BP \times \exp\left(\sum_{n=1}^N n - \text{gram precision}_n \times \log(n - \text{gram precision}_n)\right) \quad (3.3)$$

Keterangan:

2. Brevity Penalty (BP):

- $\text{len}(\text{translation})$: Panjang teks terjemahan.
- $\text{len}(\text{reference})$: Panjang teks referensi.

3. Presisi n -gram:

- $\text{count}(n\text{-grams in translation that match } n\text{-grams in reference})$: Jumlah n -gram dalam teks terjemahan yang cocok dengan n -gram dalam teks referensi.
- $\text{count}(n\text{-grams in translation})$: Jumlah total n -gram dalam teks terjemahan.

4. BLEU *Score*:

- precision_n : Presisi n -gram.
- $\sum_{n=1}^N n$: Penjumlahan dari $n = 1$ hingga N , di mana N adalah orde maksimum n -gram yang dihitung.
- $\text{precision}_n \times \log(n - \text{gram precision}_n)$: Kontribusi presisi n -gram pada BLEU *score* dengan memberikan bobot eksponensial.

- $\exp(\sum_{n=1}^N n - \text{gram precision}_n \times \log(n - \text{gram precision}_n))$:
Menggabungkan semua kontribusi presisi n -gram dengan eksponensial untuk menghasilkan BLEU *score* akhir.

Dalam penelitian ini, BLEU Score digunakan sebagai metrik evaluasi utama. Brevity Penalty memperhitungkan perbedaan panjang antara teks terjemahan dan referensi, sementara presisi n -gram mengukur sejauh mana n -gram dalam teks terjemahan cocok dengan n -gram dalam teks referensi. Skor BLEU kemudian dihitung dengan memberikan bobot eksponensial pada presisi n -gram untuk mendapatkan gambaran keseluruhan kualitas terjemahan. Metode ini memberikan evaluasi kuantitatif yang berguna dalam mengevaluasi performa model terjemahan mesin.

DAFTAR PUSTAKA

- Abdurrahman, L.O., Anraeni, S., Gaffar, A.W.M., 2023, Implementasi Algoritma Knuth Morris Pratt pada Aplikasi Kamus Bahasa Muna, *Jurnal Minfo Polgan*, 12, 1, 1330–1339.
- Abidin, Z., 2017, Penerapan Neural Machine Translation untuk Eksperimen Penerjemahan secara Otomatis pada Bahasa Lampung – Indonesia, *Prosiding Seminar Nasional Metode Kuantitatif* 2017, 978, 53–68.
- Abidin, Z., 2018, Translation of Sentence Lampung-Indonesian Languages with Neural Machine Translation Attention Based Approach, *Inovasi Pembangunan: Jurnal Kelitbangan*, 6, 02, 191–206.
- Bahdanau, D., Cho, K.H., Bengio, Y., 2015, Neural machine translation by jointly learning to align and translate, *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.
- Fauziyah, Y., Ilyas, R., Kasyidi, F., 2022, Mesin Penterjemah Bahasa Indonesia-Bahasa Sunda Menggunakan Recurrent Neural Networks, *Jurnal Teknoinfo*, 16, 2, 313.
- Susilawati, F.E., Suparman, Andi. M.G.P., 2022, Pengujian Black Box Aplikasi Penjualan Pupuk Bersubsidi Menggunakan Teknik Equivalavence Partitioning, *Jurnal Ilmiah Information Technology*, 12, 8.5.2017, 2003–2005.
- Gunawan, W., Sujaini, H., Tursina, T., 2021, Analisis Perbandingan Nilai Akurasi Mekanisme Attention Bahdanau dan Luong pada Neural Machine Translation Bahasa Indonesia ke Bahasa Melayu Ketapang dengan Arsitektur Recurrent Neural Network, *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 7, 3, 488.
- Ian, G., Yoshua, B., Aaron, C., 2016, *Deep Learning*. MIT Press Cambridge.

- Karenina, E., Ela, R.T., 2021, Analisis dan Perancangan Sistem Informasi Akuntansi Kas Rumah Sakit Menggunakan Metode SDLC dan Framework BABOK Versi 3.0, *Indonesian Accounting Literacy Journal*, 1, 2, 411–420.
- Kishore, P., Salim, R., Todd, W., Wei-Jing, Z., 1922, BLEU: a Method for Automatic Evaluation of Machine Translation, *Annalen Der Physik*, 371, 23, 437–461.
- Méndez, J.R., Iglesias, E.L., Fdez-Riverola, F., Díaz, F., Corchado, J.M., 2006, Tokenising, stemming and stopword removal on anti-spam filtering domain, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4177 LNAI, 449–458.
- Nitin, I., Fred, D., 2010, *Handbook of Natural Language Processing second edition*, CRC Press.
- Narasimha, R.B.N., Bhadri, R.M.S.V.S., Satyanarayana, K.V.V., 2021, Effective preprocessing based neural machine translation for english to telugu cross-language information retrieval, *IAES International Journal of Artificial Intelligence*, 10, 2, 306–315.
- Non, A., 2019, Upaya Masyarakat dan Pemerintah dalam Mencegah Kepunahan Bahasa Daerah untuk Menghadapi Tantangan Revolusi Industri di Era 4.0, *Jurnal Akademika*, 1, 3, 61–64.
- Pakaya, R., Tapate, A.R., Suleman, S., 2020, Perancangan Aplikasi Penjualan Hewan Ternak Untuk Qurban Dan Aqiqah Dengan Metode Unified Modeling Language (Uml), *Jurnal Technopreneur (JTech)*, 8, 1, 31–40.
- Pemerintah Indonesia, 2003, *Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional*, Issue Kolisch 1996, Sekretaris Negara Republik Indonesia.

- Pemerintah Indonesia, 2019, UU RI No. 24 Tahun 2009 tentang Bendera, Bahasa, dan Lambang Negara serta Lagu Kebangsaan, *Journal of Chemical Information and Modeling*, 53, 9, 1689–1699.
- Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., Yang, G.Z., 2017, Deep Learning for Health Informatics, *IEEE Journal of Biomedical and Health Informatics*, 21, 1, 4–21.
- Razsiah, F., Josi, A., Mubaroh, S., 2023, Aplikasi Penerjemah Bahasa Bangka Ke Bahasa Indonesia Menggunakan Neural Machine Translation Berbasis Website, *JITT : Jurnal Inovasi Teknologi Terapan*, 01, 1, 68–76.
- Al-Ibrahim, R., Duwairi, R.M., 2020, Terjemahan Mesin Neural dari bahasa Yordania Dialek ke Bahasa Arab Standar Modern Machine Translated by Google, *International Conference on Information and Communication Systems (ICICS)*, 173–178.
- Sarmawati, Sri, S.D., Ino, L., 2016, Inteferesi Fonologis Bahasa Muna Dalam Penggunaan Bahasa Indonesia Lisan Siswa MTs Karolembo Di Kabupaten Muna, *Revista Brasileira de Linguística Aplicada*, 5, 1, 10
- Sharma, A., Vans, E., Shigemizu, D., Boroevich, K.A., & Tsunoda, T., 2019, DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture, *Scientific Reports*, 9, 1, 1–7.
- Sintoris, K., Vergidis, K., 2017, Extracting business process models using natural language processing (NLP) techniques, *Proceedings - 2017 IEEE 19th Conference on Business Informatics, CBI 2017*, 1, 135–139.
- Sorin, V., Barash, Y., Konen, E., Klang, E., 2020, Deep Learning for Natural Language Processing in Radiology—Fundamentals and a Systematic Review, *Journal of the American College of Radiology*, 17, 5, 639–648.

- Sun, S., Luo, C., Chen, J., 2017, A review of natural language processing techniques for opinion mining systems, In *Information Fusion* Vol. 36. Elsevier B.V.
- Utami, A.E., Nurhayati, O.D., Martono, K.T., 2016, Aplikasi Penerjemah Bahasa Inggris – Indonesia dengan Optical Character Recognition Berbasis Android, *Jurnal Teknologi Dan Sistem Komputer*, 4, 1, 167.
- Yordan, A.B., Mursityo, Y.T., Pramono, D., 2019, *Pengembangan Sistem Informasi Layanan Pemeriksaan Kesehatan Tenaga Kerja Dengan Metode Rational Unified Process (Studi Kasus Klinik Argaraya Medika Malang)*, 3, 4, 3461–3468.
- Yunus, Balawa, L.O., Harmin, Tike, L., 2022, Kebertahanan kosakata bahasa muna pada kalangan remaja di Kota Raha Kabupaten Muna. *Jurnal Bastra*, 7, 1, 165–176.
- Zhang, W., Lin, F., Wang, X., Liang, Z., Huang, Z., 2019, *SubCharacter Chinese-English Neural Machine Translation with Wubi encoding*, Lunde 2009.