MODEL LONG SHORT-TERM MEMORY DALAM PENERJEMAHAN MESIN BAHASA LAMPUNG DIALEK API KE BAHASA INDONESIA

Mohammed Raihan Akbar
Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Ilmu Komputer
Universitas Lampung
Email: raihanakbar@protonmail.com

Kurnia Muludi
Program Studi Magister Teknik Informatika
Institut Informatika dan Bisnis Darmajaya
Email: kurnia@darmajaya.ac.id

Akmal Junaidi
Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Ilmu Komputer
Universitas Lampung
Email: akmal.junaidi@fmipa.unila.ac.id

Favorisen Rosyking Lumbanraja
Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Ilmu Komputer
Universitas Lampung
Email: favorisen.lumbanraja@fmipa.unila.ac.id

ABSTRAK

Penelitian ini mengkaji penerjemahan bahasa Lampung dialek Api ke bahasa Indonesia dengan menggunakan model *Long Short-Term Memory* (LSTM). Bahasa Lampung, salah satu bahasa daerah Indonesia, menghadapi ancaman kepunahan sehingga perlu dilestarikan dan dioptimalkan penggunaannya. Penelitian ini bertujuan untuk membangun model penerjemahan otomatis yang dapat memfasilitasi pembelajaran serta penggunaan bahasa Lampung secara lebih luas. *Dataset* yang digunakan berisi pasangan kalimat paralel dalam bahasa Lampung dan bahasa Indonesia, yang dipisahkan menjadi data latih dan data uji. Model LSTM terdiri atas lapisan *embedding*, LSTM, dan *dense*, yang dilatih menggunakan *optimizer* Adam dan fungsi *loss categorical crossentropy*. Evaluasi model dengan metrik BLEU menunjukkan hasil terjemahan yang cukup baik, dengan skor tertinggi mencapai sekitar 0,8 pada 1-grams, 0,75 pada 1-2-grams, 0,65 pada 1-3-grams, dan 0,55 pada 1-4-grams untuk data latih. Sementara pada data uji, skor BLEU tercatat 0,5 untuk 1-grams, 0,35 untuk 1-2-grams, 0,25 untuk 1-3-grams, dan 0,2 untuk 1-4-grams. Hasil ini mengindikasikan bahwa model mampu memahami konteks bahasa dengan baik selama pelatihan dan tetap mempertahankan performa yang layak saat diuji dengan data baru. Selama pelatihan, penurunan *loss* yang konsisten menunjukkan peningkatan kemampuan model dalam memprediksi terjemahan dengan akurat. Penelitian ini membuktikan bahwa model LSTM dapat dimanfaatkan untuk penerjemahan bahasa Lampung ke bahasa Indonesia serta berpotensi diterapkan pada bahasa daerah lainnya yang terancam punah. Pengembangan lebih lanjut dapat mencakup penggunaan arsitektur model yang lebih kompleks, *dataset* yang lebih besar dan beragam, serta teknik augmentasi data guna meningkatkan kinerja model. Diharapkan hasil penelitian ini menjadi acuan dalam pengembangan lebih lanjut di bidang penerjemahan mesin dan pelestarian bahasa.

Kata kunci: penerjemahan mesin, bahasa lampung, bahasa indonesia, *long short-term memory*, pemrosesan bahasa alami, pelestarian bahasa

ABSTRACT

This study examines the translation of the Lampung Api dialect into Indonesian using the Long Short-Term Memory (LSTM) model. Lampung, one of Indonesia's regional languages, is facing the threat of extinction, making its preservation and optimal use essential. The study aims to develop an automatic translation model that can facilitate broader learning and use of the Lampung language. The dataset consists of parallel sentence pairs in Lampung and Indonesian, divided into training and testing data. The LSTM model comprises embedding, LSTM, and dense layers, trained using the Adam optimizer and categorical crossentropy loss function. Model evaluation using the BLEU metric demonstrated satisfactory translation results, with the highest scores reaching approximately 0.8 for 1-grams, 0.75 for 1-2-grams, 0.65 for 1-3-grams, and 0.55 for 1-4-grams on the training set. On the test set, BLEU scores recorded 0.5 for 1-grams, 0.35 for 1-2-grams, 0.25 for 1-3-grams, and 0.2 for 1-4-grams. These results indicate that the model effectively captures linguistic context during training and maintains adequate performance when tested with new data. A consistent decrease in loss during training indicates the model's improved ability to predict accurate translations. This study demonstrates that the LSTM model can be utilized for translating Lampung into Indonesian and holds potential for application to other endangered regional languages. Further development could involve using more complex model architectures, larger and more diverse datasets, and data augmentation techniques to enhance model performance. It is hoped that this study will serve as a reference for further development in the fields of machine translation and language preservation.

Keywords: *machine translation, lampung language, indonesian language, long short-term memory, natural language processing, language preservation*

1. PENDAHULUAN

Indonesia, sebagai negara kepulauan di Asia Tenggara dengan lebih dari 17.000 pulau dan populasi lebih dari 275 juta jiwa, memiliki keragaman etnis dan bahasa yang sangat beragam. Bahasa daerah, termasuk Bahasa Lampung, memiliki peran vital dalam memperkuat budaya dan tradisi lokal. Bahasa Lampung, terutama dialek Lampung Api, memiliki ciri khas yang unik dan memainkan peran krusial dalam budaya masyarakat Lampung. Namun, sejak era Presiden Soeharto, perkembangan globalisasi dan kebijakan transmigrasi menyebabkan penurunan jumlah penutur bahasa Lampung, sehingga bahasa ini berisiko punah dalam waktu 40 hingga 50 tahun mendatang [1].

Untuk melindungi kelangsungan bahasa daerah, perlu dikembangkan sistem penerjemahan antara bahasa daerah dan bahasa nasional yang dapat memperlancar pemahaman antar-penutur [2]. *Machine translation* adalah perangkat lunak yang dirancang untuk mengubah teks dari satu bahasa ke bahasa lainnya secara otomatis [3]. Dalam upaya pelestarian bahasa, pengembangan mesin penerjemah Bahasa Lampung menjadi hal yang sangat mendesak.

Machine translation mampu secara otomatis menerjemahkan kalimat dari bahasa sumber ke bahasa target [4], membantu mengonversi bahasa asing ke bahasa Indonesia atau sebaliknya [5]. Meski demikian, penerjemahan, baik oleh manusia maupun mesin, menghadapi berbagai tantangan, terutama dalam menjaga konteks. Tantangan ini sering disebabkan oleh perbedaan dalam aspek geografis, budaya, kebiasaan, serta leksikon [6]. Teknologi penerjemahan ini telah berevolusi pesat dari metode berbasis aturan seperti *Rule-Based Machine Translation* (RBMT) [7], [8], [9], ke metode berbasis statistik yang dikenal sebagai *Statistical Machine Translation* (SMT) [10], [11], [12]. Saat ini, pendekatan paling mutakhir adalah *Neural Machine Translation* (NMT), yang menggunakan jaringan saraf buatan [13], [14], [15].

Pendekatan NMT mengandalkan struktur model dan proses yang lebih kompleks. Model ini menggunakan arsitektur *Sequence to Sequence* (Seq2seq) [16], yang terdiri atas dua bagian utama: *Encoder* dan *Decoder* [17]. *Encoder* memproses bahasa sumber, sementara *Decoder* menghasilkan terjemahan berdasarkan keluaran dari *Encoder* [18]. Proses ini didukung oleh jaringan saraf *Recurrent Neural Network* (RNN) [19].

Neural Machine Translation (NMT), yang menggunakan arsitektur RNN, dikenal karena keunggulannya dalam menangani urutan teks secara kontekstual dan berulang, menjadikannya solusi efektif dalam penerjemahan bahasa [20]. *Long Short-Term Memory* (LSTM) merupakan pengembangan dari

arsitektur RNN. Penelitian ini berfokus pada penerapan arsitektur LSTM untuk penerjemahan Bahasa Lampung Api ke Bahasa Indonesia dan mengevaluasi hasil terjemahannya. Fokus penelitian ini dibatasi pada penggunaan arsitektur LSTM untuk penerjemahan spesifik antara Bahasa Lampung Api dan Bahasa Indonesia.

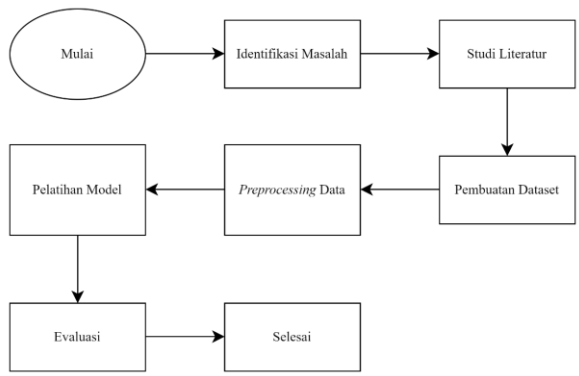
Diharapkan penelitian ini dapat berkontribusi pada pelestarian Bahasa Lampung, memperlancar komunikasi antarbudaya, memperkuat identitas budaya masyarakat Lampung, meningkatkan akses informasi, serta menjadi referensi bagi penelitian serupa di masa mendatang.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental yang terdiri dari beberapa tahapan utama. Tahap pertama melibatkan pengumpulan data dengan menyusun korpus paralel berisi pasangan kalimat dalam bahasa Lampung dialek Api dan bahasa Indonesia. Selanjutnya, dilakukan proses *preprocessing* untuk memastikan data memiliki kualitas dan konsistensi yang memadai sebelum digunakan dalam pelatihan model.

Model yang diterapkan dalam penelitian ini adalah *Long Short-Term Memory* (LSTM), salah satu jenis *Recurrent Neural Network* (RNN) yang dikenal karena kemampuannya dalam mengolah data sekuensial yang panjang dan kompleks. Setelah pelatihan menggunakan korpus yang telah melalui proses *preprocessing*, model tersebut diterapkan untuk menerjemahkan kalimat dari bahasa Lampung dialek Api ke bahasa Indonesia.

Tahap terakhir penelitian adalah evaluasi hasil terjemahan. Evaluasi dilakukan dengan membandingkan hasil terjemahan model dengan hasil yang dihasilkan oleh penerjemah manusia. Kualitas terjemahan diukur menggunakan metrik BLEU (*Bilingual Evaluation Understudy*), yang merupakan standar umum dalam penilaian kualitas terjemahan mesin. Alur penelitian ini dapat dilihat pada ilustrasi berikut.



Gambar 1. Alur Penelitian

Penjelasan dari gambar 1 di atas adalah sebagai berikut.

2.1 *Identifikasi Masalah*

Bahasa Lampung, salah satu bahasa daerah di Indonesia, terancam punah karena semakin jarang digunakan dalam percakapan sehari-hari. Faktor-faktor seperti urbanisasi, migrasi, dan dominasi bahasa Indonesia sebagai bahasa nasional berkontribusi pada penurunan jumlah penuturnya. Kondisi ini diperparah oleh minimnya dukungan teknologi yang dapat membantu mempertahankan dan memfasilitasi penggunaan bahasa Lampung.

Penurunan penggunaan bahasa ini di kalangan generasi muda mempercepat risiko kepunahan. Untuk mengatasi tantangan ini, diperlukan pendekatan inovatif berupa pengembangan teknologi penerjemahan otomatis yang dapat mendukung penggunaan bahasa Lampung di berbagai bidang, termasuk pendidikan dan kebudayaan.

Penelitian ini bertujuan untuk mengembangkan mesin penerjemahan berbasis *Long Short-Term Memory* (LSTM) yang mampu menerjemahkan bahasa Lampung ke bahasa Indonesia secara efektif. Diharapkan hasil dari penelitian ini dapat berkontribusi pada upaya pelestarian bahasa Lampung serta mempermudah pemahaman bahasa ini bagi masyarakat secara lebih luas.

2.2 Studi Literatur

Langkah kedua dalam penelitian ini adalah melakukan studi literatur, di mana peneliti mengumpulkan berbagai referensi yang relevan dengan permasalahan yang sedang dikaji. Tujuan utama dari tahap ini adalah memperoleh pemahaman mendalam mengenai penelitian-penelitian sebelumnya yang berkaitan dengan topik dan metode yang digunakan. Informasi yang diperoleh dari teori dan hasil penelitian terdahulu menjadi dasar penting untuk menganalisis masalah secara akurat sesuai dengan kerangka berpikir ilmiah yang telah ditentukan.

Dalam konteks penelitian ini, telah banyak dilakukan studi terkait penerjemahan mesin dan metode *Long Short-Term Memory* (LSTM). Studi literatur ini bertujuan untuk memberikan wawasan yang lebih luas dan mendalam guna memperkuat pemahaman peneliti terhadap topik yang tengah diteliti.

2.3 Pembuatan Dataset

Langkah awal pengumpulan data dalam penelitian ini dimulai dengan pencarian berbagai teks dalam bahasa Indonesia yang kemudian diterjemahkan secara manual ke dalam bahasa Lampung oleh peneliti. Tujuan dari proses ini adalah untuk memastikan bahwa data yang diperoleh mencakup beragam konteks serta variasi bahasa yang relevan.

Penerjemahan manual dari bahasa Indonesia ke bahasa Lampung dilakukan guna menyusun korpus paralel, yang menjadi fondasi utama penelitian ini. Data dalam bahasa Lampung dialek Api diperoleh dari berbagai sumber, termasuk kamus, cerita rakyat, dan teks sastra. Data ini digunakan dalam proses pelatihan dan pengujian model *Long Short-Term Memory* (*LSTM*) untuk penerjemahan otomatis dari bahasa Lampung ke bahasa Indonesia.

Dengan memastikan kualitas dan keragaman data yang diperoleh melalui penerjemahan manual, diharapkan model *LSTM* yang dibangun dapat menghasilkan terjemahan yang akurat dan konsisten.

2.4 *Preprocessing Data*

Preprocessing data merupakan tahap krusial untuk memastikan bahwa data yang digunakan dalam pelatihan dan evaluasi model *Long Short-Term Memory* (LSTM) memiliki kebersihan, konsistensi, dan format yang sesuai. Tahap pertama dalam proses ini adalah membagi *dataset* menjadi dua bagian, yaitu data latih (80%) dan data uji (20%). Pembagian 80-20 dipilih karena merupakan praktik standar yang memberikan cukup data untuk pelatihan sekaligus menyisakan data yang memadai untuk evaluasi, sehingga memungkinkan estimasi performa model pada data baru secara lebih akurat.

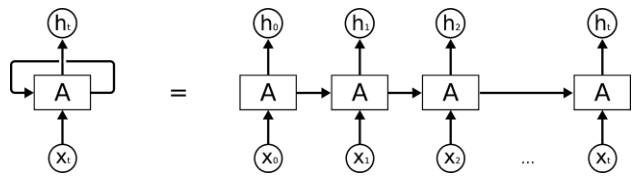
Setelah pembagian *dataset*, dilakukan proses normalisasi teks untuk menghilangkan variasi yang tidak relevan. Langkah ini meliputi perubahan semua huruf menjadi huruf kecil, penghapusan karakter khusus, serta penghilangan spasi berlebih.

Tahap terakhir adalah tokenisasi teks, yaitu proses memecah teks menjadi unit-unit kecil yang disebut token. Tokenisasi ini dapat memecah setiap kalimat menjadi deretan kata, karakter, atau sub-kata, sesuai dengan kebutuhan model.

Dengan menerapkan pembagian *dataset*, normalisasi teks, dan tokenisasi secara sistematis, diharapkan model LSTM dapat dilatih dengan data yang bersih dan terstruktur, yang pada akhirnya dapat meningkatkan kinerja dan akurasi dalam menerjemahkan bahasa Lampung ke bahasa Indonesia.

2.5 Pelatihan Model

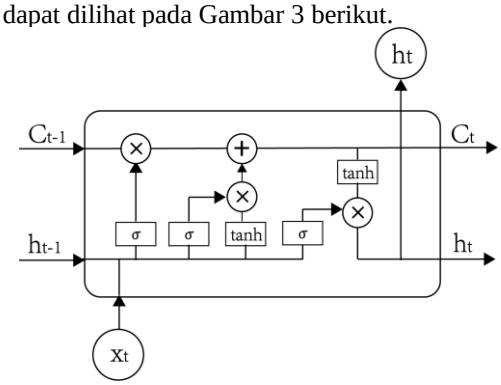
Pelatihan model merupakan tahap kunci dalam penelitian ini. *Dataset* berupa korpus paralel digunakan untuk melatih model *Long Short-Term Memory* (LSTM). LSTM adalah varian dari *Recurrent Neural Network* (RNN) yang banyak diaplikasikan pada pemrosesan data berurutan seperti teks dan sejenisnya [21]. *Recurrent Neural Network* (RNN) dalam *Neural Machine Translation* (NMT) terdiri dari tiga lapisan utama: lapisan pertama mengonversi setiap kata menjadi vektor melalui metode seperti *word embedding* atau *one-hot word index*, lapisan tersembunyi berulang yang terus memperbarui status tersembunyi saat setiap kata diproses, dan lapisan terakhir yang memprediksi kata berikutnya sambil mempertahankan status tersembunyi [22]. Arsitektur RNN ditunjukkan pada Gambar 2 berikut.



Gambar 2. Arsitektur RNN

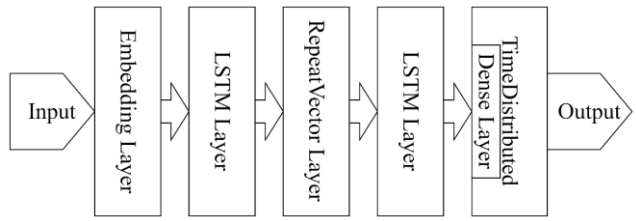
Long Short-Term Memory (LSTM), diperkenalkan oleh Sepp Hochreiter dan Juergen Schmidhuber pada tahun 1997, merupakan pengembangan dari RNN yang dirancang untuk mengatasi masalah *vanishing gradient* dalam pemrosesan data sekuensial jangka panjang [23]. Masalah *vanishing gradient* pada RNN muncul ketika informasi lama menjadi kurang relevan dan digantikan oleh informasi baru, yang menyebabkan hilangnya jejak informasi penting. LSTM mengatasi masalah ini dengan menggunakan *memory cell* yang menyimpan informasi dalam jangka waktu tertentu serta memanfaatkan gerbang untuk mengontrol aliran informasi masuk dan keluar dari *memory cell*, sehingga lebih efektif dalam mengelola ketergantungan jangka panjang.

LSTM memiliki rangkaian *memory cell* yang menggantikan neuron pada lapisan tersembunyi RNN [24]. Setiap *memory cell* dilengkapi dengan tiga gerbang utama: *forget gate* yang menentukan informasi mana yang harus dihapus dari *cell state*, *input gate* yang mengatur informasi mana yang akan diperbarui, dan *output gate* yang menghasilkan keluaran berdasarkan *cell state*. Arsitektur LSTM dapat dilihat pada Gambar 3 berikut.



Gambar 3. Arsitektur LSTM [24]

Model *Encoder-Decoder* adalah metode *Sequence to Sequence* (*Seq2seq*) yang terdiri dari dua komponen utama. Komponen pertama, *Encoder*, mengubah kata-kata dari kalimat masukan menjadi vektor. Komponen kedua, *Decoder*, memprediksi kata-kata terjemahan berdasarkan vektor yang dihasilkan oleh *Encoder* [25]. Skema *Encoder-Decoder* yang digunakan dalam penelitian ini dapat dilihat pada Gambar 4 berikut.



Gambar 4. Model LSTM Seq2seq *Encoder-Decoder*

Lapisan sebelum *RepeatVector* berfungsi sebagai *encoder*, sementara lapisan setelah *RepeatVector* berfungsi sebagai *decoder*. *Layer RepeatVector* berperan sebagai adaptor untuk menghubungkan bagian *encoder* dan *decoder*. Parameter pelatihan yang digunakan pada model ini dirangkum dalam Tabel 1 di bawah.

Tabel 1. Pengaturan Parameter

No	*Parameter*	*Nilai*
1	*Unit LSTM*	*256*
2	*Fungsi Aktivasi*	*Softmax*
3	*Fungsi Loss*	*Categorical Cross-Entropy*
4	*Persentase Validation*	*10*
5	*Epoch*	*60 dengan EarlyStopping*
6	*Ukuran Batch*	*64*
7	*Optimizer*	*Adam*

Konfigurasi parameter dalam tabel di atas dipilih berdasarkan pertimbangan keseimbangan antara efisiensi komputasi dan akurasi model. Berikut adalah justifikasi dari setiap parameter yang digunakan dalam penelitian ini. Pemilihan parameter dalam penelitian ini dilakukan dengan mempertimbangkan keseimbangan antara performa model dan efisiensi komputasi. Model *Long Short-Term Memory* (LSTM) menggunakan 256 unit karena ukuran ini dianggap cukup untuk menangkap pola dalam data sekuensial dengan kompleksitas sedang tanpa meningkatkan risiko *overfitting* secara signifikan. Unit yang lebih kecil ditemukan kurang mampu mempertahankan konteks dalam kalimat panjang, sementara unit yang lebih besar meningkatkan kebutuhan komputasi tanpa memberikan peningkatan kinerja yang sepadan. Pelatihan model dilakukan selama 60 *epoch* dengan penerapan *Early Stopping*, yang memungkinkan proses berhenti lebih awal jika tidak ada peningkatan performa pada data validasi setelah 20 *epoch* berturut-turut. Pemilihan jumlah *epoch* ini didasarkan pada hasil eksperimen awal yang menunjukkan bahwa model mulai mengalami konvergensi setelah sekitar 40 *epoch* dan mencapai titik optimal sebelum 60 *epoch*.

Ukuran *batch* yang digunakan adalah 64, karena nilai ini memberikan keseimbangan antara stabilitas pembaruan parameter dan efisiensi pelatihan. Ukuran *batch* yang lebih kecil meningkatkan kestabilan model tetapi memperlambat pelatihan, sedangkan ukuran yang lebih besar dapat menyebabkan model kehilangan detail penting dalam pola bahasa. *Optimizer* yang digunakan adalah Adam, karena dikenal memiliki kecepatan konvergensi yang baik dengan menggabungkan momentum dan *adaptive learning rate*, sehingga membantu model belajar lebih cepat dan stabil. Pada lapisan *output*, digunakan fungsi aktivasi *Softmax* karena tugas yang dilakukan merupakan klasifikasi multi-kelas, di mana model harus memilih kata yang paling mungkin sebagai hasil terjemahan berdasarkan distribusi probabilitas. Fungsi *loss Categorical Cross-Entropy* digunakan untuk mengukur sejauh mana model dapat memprediksi kata-kata dalam bahasa Indonesia berdasarkan *input* dalam bahasa Lampung, mengingat *output* yang dihasilkan berupa distribusi probabilitas atas kosakata target.

Dengan pemilihan parameter ini, model diharapkan dapat menghasilkan terjemahan yang optimal tanpa mengalami *overfitting* atau *underfitting* yang signifikan. Pendekatan ini juga memastikan bahwa pelatihan berlangsung secara efisien tanpa mengorbankan kualitas terjemahan yang dihasilkan.

2.6 Evaluasi

Setelah model menyelesaikan proses penerjemahan otomatis, langkah berikutnya adalah mengevaluasi kualitas hasil terjemahan dengan membandingkannya terhadap hasil terjemahan manual yang dilakukan oleh penerjemah manusia.

Penelitian ini menggunakan metode evaluasi *Bilingual Evaluation Understudy* (*BLEU*) yang diperkenalkan oleh Papineni et al. [26] untuk mengukur seberapa dekat hasil terjemahan mesin dengan hasil terjemahan manusia. Metode *BLEU* mengevaluasi kualitas terjemahan dengan membandingkan kesamaan antara *n-grams* dalam hasil terjemahan mesin dan *n-grams* yang terdapat dalam terjemahan referensi yang dibuat secara manual. Rumus-rumus penghitungan skor *BLEU* disajikan pada bagian berikut.

2.6.1 *Rumus Penghitungan BLEU*

Perhitungan skor *BLEU* didasarkan pada penggabungan antara *n-gram precision* dan *brevity penalty*. Rumus dasar untuk menghitung *n-gram precision* pada ukuran n tertentu adalah sebagai berikut:

$$\text{Precision}_n = \frac{\text{Jumlah n-gram yang cocok}}{\text{Jumlah n-gram dalam terjemahan mesin}} \tag{1}$$

Papineni et al. [26] memperkenalkan konsep *modified precision n-gram* untuk mengatasi masalah *over-counting*. Pada setiap ukuran n, precision dimodifikasi dengan mengurangi jumlah *n-grams* yang berlebihan. Jenis precision yang telah dimodifikasi inilah yang digunakan dalam perhitungan skor *BLEU*.

$$\text{Modified Precision}_n = \frac{\text{Jumlah n-gram yang cocok}}{\text{Jumlah n-gram yang seharusnya}} \tag{2}$$

Selanjutnya, *modified precision* digabungkan menggunakan rumus berikut, yang memperhitungkan berbagai nilai N dengan bobot yang berbeda. Biasanya, N = 4 digunakan dengan bobot seragam $w_n = N / 4$ (*four-gram*).

$$\begin{aligned} \textit{Geometric Average Precision } (N) &= exp(\sum_{n=1}^{N} w_n \; log \; p_n) \\ &= \prod_{n=1}^{N} {p_n}^{w_n} \\ &= (p_1)^{\frac{1}{4}} \cdot (p_2)^{\frac{1}{4}} \cdot (p_3)^{\frac{1}{4}} \cdot (p_4)^{\frac{1}{4}} \end{aligned} \tag{3}$$

Langkah berikutnya adalah menghitung *brevity penalty*, yang berfungsi untuk mencegah terjemahan yang terlalu pendek mendapatkan skor tinggi. Jika panjang terjemahan mesin lebih pendek atau sama dengan panjang referensi, maka penalti diterapkan menggunakan rumus berikut:

$$\text{BP} = \begin{cases} 1 & \text{jika } c > r \\ e^{(1-\frac{r}{c})} & \text{jika } c \leq r \end{cases} \tag{4}$$

Di mana r adalah panjang total korpus referensi dan c adalah panjang total korpus yang diprediksi. Nilai *brevity penalty* dijaga agar tidak melebihi 1, bahkan jika panjang kalimat prediksi jauh lebih besar dibandingkan dengan referensi. Jika kalimat prediksi terlalu pendek, nilai penalti ini akan menjadi lebih kecil.

Terakhir, skor *BLEU* dihitung dengan mengalikan *brevity penalty* dengan rata-rata geometrik dari semua nilai *n-gram precision*:

$$Bleu\ (N) = Brevity\ Penalty \cdot Geometric\ Average\ Precision\ Scores\ (N) \quad (5)$$

Penting untuk diperhatikan bahwa skor *BLEU* dihitung berdasarkan keseluruhan korpus, bukan berdasarkan kalimat individual. Skor ini memperhitungkan semua teks dalam korpus yang diprediksi secara keseluruhan, sehingga tidak dapat dihitung dengan mengukur skor *BLEU* untuk setiap kalimat secara terpisah dan mengambil rata-rata dari skor-skor tersebut.

3. HASIL DAN PEMBAHASAN

3.1 *Dataset*

Proses penelitian penerjemahan dari bahasa Lampung ke bahasa Indonesia dimulai dengan pengumpulan korpus bahasa Indonesia. Data diperoleh dari situs *manythings.org*, terutama dari laman *Tab-delimited Bilingual Sentence Pairs* yang menyediakan korpus paralel antara bahasa Inggris dan bahasa Indonesia. Korpus tersebut berformat teks, dengan kolom-kolom yang memisahkan kalimat bahasa Inggris dan bahasa Indonesia serta disertai atribut lisensi CC-BY. Data ini digunakan untuk membangun korpus yang representatif dan komprehensif guna mendukung analisis dalam penelitian penerjemahan.

Setelah memperoleh korpus bahasa Indonesia, data diterjemahkan ke dalam bahasa Lampung secara manual untuk memastikan keakuratan dan kesesuaian dengan konteks serta struktur bahasa Lampung yang autentik. Proses penerjemahan manual ini melibatkan penggunaan berbagai referensi, termasuk kamus bahasa Indonesia-Lampung dan bahan budaya Lampung. Data yang telah diterjemahkan kemudian divalidasi oleh Ibu Nurul Fitrianisa, S.Pd., guna memastikan bahwa kosakata yang digunakan sesuai dengan standar kebahasaan yang berlaku.

Selama proses pengolahan data, beberapa pasangan kalimat yang dinilai kurang relevan dihapus, menghasilkan total 11.933 pasangan kalimat yang tersisa. Dataset ini kemudian digunakan untuk melatih dan menguji model *Long Short-Term Memory* (*LSTM*) dengan harapan model mampu mempelajari data secara efektif dan menghasilkan terjemahan yang akurat antara bahasa Lampung dan bahasa Indonesia. Tabel berikut menampilkan beberapa contoh pasangan kalimat dalam dataset tersebut.

Tabel 2. Korpus Paralel

Baris	*Bahasa Indonesia*	*Bahasa Lampung*
732	*Susu berwarna putih.*	*Susu bewaghna handak.*
11764	*Buku ini diperuntukkan bagi siswa yang bahasa ibunya bukan bahasa Jepang.*	*Buku sinji dipeguwaiko bagi siswa sai bahasa makni ayin bahasa Jepang.*
11931	*Januari, Februari, Maret, April, Mei, Juni, Juli, Agustus, September, Oktober, November, dan Desember adalah dua belas bulan dalam setahun.*	*Januaghi, Pebghuaghi, Maghet, Apghil, Mei, Juni, Juli, Agustus, Septembegh, Oktobegh, Nopembegh, Desembegh iyulah wa belas bulan lom setahun.*
11932	*Irene Pepperberg, seorang peneliti di Universitas Northwestern menemukan bahwa burung beo tidak hanya bisa menirukan perkataan manusia tetapi juga bisa memahami arti dari perkataan tersebut.*	*Irene Pepperberg, seulun peneliti di Univeghsitas Northwestern ngepehaluko bahwa putik beo mak cuma dapok ngenighuko pecawaan manusiya kidang juga dapok ngemahhomi aghti anjak pecawaan tesebut.*
11933	*Jika seseorang tidak berkesempatan untuk menguasai bahasa yang diinginkannya ketika menginjak dewasa, maka kecil kemungkinan ia*	*Ki seseulun mak bekesempatan guwai nguasai bahasa sai dihagakoni waktu ngilik dewasa, mula lunik kehalokan ia*

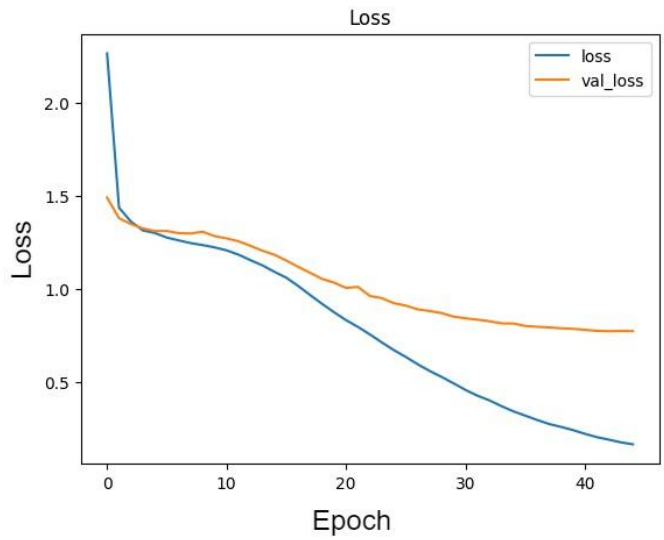
akan bisa mencapai tingkatan penutur asli dalam bahasa tersebut.	*bakal dapok tigoh tingkatan penutugh asli lom bahasa tesebut.*

Diketahui bahwa dalam korpus bahasa Indonesia terdapat 5.556 kosakata dengan panjang kalimat maksimal 26 kata dalam satu korpus. Sementara itu, korpus bahasa Lampung memiliki 6.703 kosakata dengan panjang kalimat maksimal 25 kata dalam satu korpus. Informasi ini menunjukkan perbedaan jumlah kosakata dan panjang kalimat antara kedua bahasa, yang dapat memengaruhi proses pelatihan dan akurasi model penerjemahan.

3.2 *Pelatihan Model*

Pelatihan model menggunakan korpus paralel yang disebutkan dilakukan dengan memanfaatkan Google Colab Pro, yang dilengkapi dengan spesifikasi GPU A100. Bahasa pemrograman Python 3 digunakan karena fleksibilitasnya serta dukungan yang luas dalam implementasi tugas-tugas pembelajaran mesin.

Selama proses pelatihan, nilai *loss* dicatat pada setiap *epoch* dan divisualisasikan dalam Gambar 5. Grafik ini memberikan gambaran mengenai perkembangan pembelajaran model, yang dijelaskan lebih lanjut pada bagian berikut.



Gambar 5. Grafik Loss

Grafik Loss di atas menunjukkan perubahan nilai *loss* selama proses pelatihan model LSTM dari *epoch* pertama hingga epoch terakhir. Sumbu horizontal (X) mewakili jumlah *epoch*, sedangkan sumbu vertikal (Y) menunjukkan nilai *loss*. Dua kurva dalam grafik ini merepresentasikan *loss* pada data latih dan *loss* pada data validasi.

Selama pelatihan, terlihat bahwa nilai *loss* mengalami penurunan secara bertahap, yang menunjukkan bahwa model semakin baik dalam menyesuaikan bobotnya untuk menghasilkan terjemahan yang lebih akurat. Nilai *loss* pada data validasi juga menurun, meskipun dengan fluktuasi yang lebih besar dibandingkan data latih. *Early Stopping* diterapkan untuk menghentikan pelatihan sebelum terjadi *overfitting*, yaitu saat *loss* pada data latih terus menurun tetapi *loss* pada data validasi mulai meningkat.

Hasil dari grafik ini menunjukkan bahwa model berhasil belajar secara efektif tanpa mengalami *overfitting* yang signifikan. Penurunan *loss* yang konsisten mengindikasikan bahwa model mampu memahami pola dalam data latih dan mempertahankan kinerja yang stabil pada data validasi.

Beberapa contoh hasil terjemahan per kalimat yang dihasilkan oleh model pada set latih dan set uji ditampilkan dalam Tabel 3 dan Tabel 4 di bawah. Contoh-contoh ini memberikan gambaran tentang kemampuan model dalam menerjemahkan bahasa Lampung ke bahasa Indonesia, baik pada data pelatihan maupun data pengujian, yang mencerminkan tingkat akurasi dan konsistensi model dalam menghasilkan terjemahan.

Tabel 3. Contoh Penerjemahan pada Set Latih

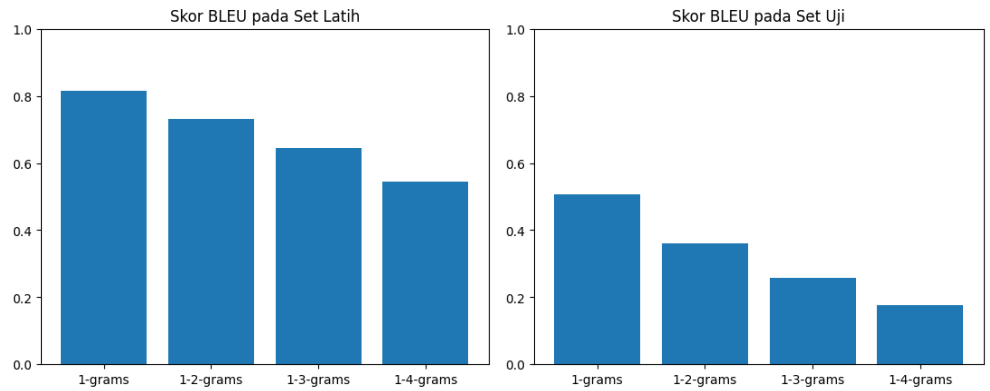
BAHASA LAMPUNG (SUMBER)	*BAHASA INDONESIA (SASARAN)*	*TERJEMAHAN OTOMATIS DALAM BAHASA INDONESIA*
akuk baghangmu	*ambil barangmu*	*ambil barangmu*
gham mak demon matematika	*kita tidak suka matematika*	*kita tidak suka matematika*
sinji keghen nihan	*ini sangat keren*	*ini sangat keren*

Tabel 4. Contoh Penerjemahan pada Set Uji

BAHASA LAMPUNG (SUMBER)	*BAHASA INDONESIA (SASARAN)*	*TERJEMAHAN OTOMATIS DALAM BAHASA INDONESIA*
sikam ghadu ngelakuko heno	*kami sudah melakukan itu*	*kami telah melakukan itu*
tomas mak demon dicubil	*tomas tidak suka disentuh*	*tomas tidak suka kegelapan*
gham kalah	*kita kalah*	*kita*

3.3 Evaluasi

Tahap akhir dari penelitian ini adalah melakukan evaluasi model dengan menggunakan metrik *Bilingual Evaluation Understudy* (*BLEU*). Hasil evaluasi ini ditampilkan pada Gambar 6 di bawah, yang memberikan gambaran mengenai performa model dalam menerjemahkan bahasa Lampung ke bahasa Indonesia berdasarkan berbagai tingkat *n-grams*.



Gambar 6. Hasil Evaluasi Skor BLEU

Pada set latih, skor *BLEU* tertinggi tercatat pada 1-grams dengan nilai sekitar 0,8, yang menunjukkan kemampuan model dalam menerjemahkan kata-kata individu dengan baik. Namun, seiring bertambahnya panjang *n-grams*, skor BLEU mengalami penurunan: sekitar 0,75 untuk 1-2-grams, 0,65 untuk 1-3-grams, dan 0,55 untuk 1-4-grams. Penurunan ini menunjukkan bahwa model mengalami kesulitan dalam mempertahankan urutan kata yang lebih panjang, yang penting untuk menghasilkan terjemahan yang alami dan koheren.

Pada set uji, skor BLEU umumnya lebih rendah dibandingkan dengan set latih di semua jenis *n-grams*. Skor tertinggi juga dicapai pada 1-grams, dengan nilai sekitar 0,5, yang menunjukkan bahwa model masih mampu menerjemahkan kata-kata individu pada data baru, tetapi dengan akurasi yang lebih rendah dibandingkan data pelatihan. Penurunan yang lebih tajam pada *n-grams* yang lebih panjang—sekitar 0,35 untuk 1-2-grams, 0,25 untuk 1-3-grams, dan 0,2 untuk 1-4-grams—mengindikasikan bahwa model menghadapi tantangan yang lebih besar dalam mempertahankan urutan kata pada data baru.

Analisis hasil BLEU menyoroti beberapa poin penting tentang kinerja model. Pada level unigram, model menunjukkan hasil yang sangat baik dalam menerjemahkan kata-kata individu di kedua set, meskipun performa pada set uji lebih rendah. Penurunan skor BLEU seiring bertambahnya panjang *n-grams* menunjukkan bahwa model kesulitan mempertahankan konteks dalam frasa atau kalimat yang lebih panjang. Perbedaan signifikan antara skor pada set latih dan set uji menunjukkan potensi *overfitting*, di

mana model terlalu menyesuaikan diri dengan data latih dan kurang mampu menggeneralisasi pada data baru.

Skor BLEU yang lebih rendah pada set uji menunjukkan keterbatasan model dalam menangani variasi yang mungkin muncul dalam teks baru yang tidak terdapat dalam data latih. Secara keseluruhan, meskipun model menunjukkan akurasi yang baik, terutama untuk kata-kata individu dan urutan pendek, efektivitasnya menurun saat menghadapi urutan kata yang lebih panjang dan kompleks. Skor BLEU dan analisis *n-grams* menunjukkan bahwa perbaikan lebih lanjut diperlukan untuk meningkatkan akurasi penerjemahan pada struktur teks yang lebih kompleks dan memastikan kesesuaian yang lebih baik dengan konteks dan isi teks asli.

Hasil penelitian ini sejalan dengan beberapa studi sebelumnya yang menggunakan model LSTM dalam penerjemahan otomatis. Beberapa penelitian menunjukkan bahwa LSTM memiliki keunggulan dalam menangani penerjemahan kata-kata individu tetapi mengalami kesulitan dalam mempertahankan konteks pada kalimat panjang. Model LSTM telah diterapkan dalam berbagai penelitian penerjemahan otomatis, termasuk penelitian oleh Fauziyah et al. [2], yang mengembangkan sistem penerjemahan bahasa Indonesia ke bahasa Sunda menggunakan *Recurrent Neural Networks* (RNN). Hasil penelitian tersebut menunjukkan bahwa model berbasis LSTM mampu menangani konteks lebih baik dibandingkan metode berbasis aturan atau statistik. Namun, mereka menemukan bahwa kualitas terjemahan masih bergantung pada ketersediaan *dataset* yang besar dan bervariasi. Hal ini sejalan dengan penelitian ini, di mana model menunjukkan skor BLEU yang cukup tinggi pada 1-grams (0,8 pada data latih dan 0,5 pada data uji), tetapi mengalami penurunan pada *n-grams* yang lebih panjang, yang mengindikasikan bahwa model masih memiliki keterbatasan dalam menangkap struktur kalimat yang lebih kompleks.

Penelitian lain oleh Ortega et al. [15] yang mengembangkan model penerjemahan untuk bahasa *polysynthetic* menggunakan *Neural Machine Translation* (NMT) berbasis LSTM juga menemukan bahwa meskipun model ini cukup efektif dalam menangani struktur bahasa yang unik, LSTM masih mengalami keterbatasan dalam menangkap ketergantungan kata dalam kalimat panjang. Fenomena serupa terlihat dalam penelitian ini, di mana model mengalami kesulitan dalam mempertahankan urutan kata yang panjang pada data uji, sebagaimana ditunjukkan oleh penurunan skor BLEU pada *n-grams* lebih tinggi. Sutskever et al. [16] yang pertama kali memperkenalkan arsitektur *Sequence-to-Sequence* (Seq2Seq) dengan LSTM, juga menunjukkan bahwa LSTM mampu menangani penerjemahan antar bahasa dengan cukup baik, tetapi masih menghadapi tantangan dalam menangkap konteks jangka panjang.

Meskipun hasil penelitian ini menunjukkan bahwa LSTM mampu menghasilkan terjemahan dengan akurasi yang cukup baik, studi terbaru seperti yang dilakukan oleh Vaswani et al. [27] menunjukkan bahwa pendekatan *Transformer*, yang diperkenalkan melalui model *Transformer-based* NMT, memiliki keunggulan dalam memahami konteks kalimat panjang dengan lebih baik dibandingkan LSTM. Model *Transformer* menggunakan mekanisme *self-attention*, yang memungkinkan model untuk menangkap hubungan antar kata dalam satu urutan tanpa mengalami kendala seperti yang dialami oleh model berbasis LSTM. Dengan demikian, meskipun LSTM masih menjadi pilihan yang solid untuk penerjemahan bahasa Lampung ke bahasa Indonesia, peningkatan model dengan arsitektur *Transformer* dapat menjadi langkah penelitian lanjutan untuk meningkatkan akurasi dan kelancaran terjemahan.

Untuk memahami lebih lanjut mengapa model dalam penelitian ini mengalami penurunan skor BLEU pada *n-grams* yang lebih panjang, diperlukan analisis terhadap faktor-faktor yang memengaruhi kinerja model. Salah satu faktor utama adalah perbedaan struktur morfologi dan sintaksis antara bahasa Lampung dan bahasa Indonesia. Dalam beberapa kasus, kata atau frasa dalam bahasa Lampung tidak memiliki padanan langsung dalam bahasa Indonesia, sehingga model mengalami kesulitan dalam menghasilkan terjemahan yang akurat. Selain itu, fleksibilitas dalam susunan kata bahasa Lampung sering kali menyebabkan model menghasilkan terjemahan yang kurang natural dibandingkan dengan terjemahan manusia.

Selain keterbatasan struktural, jumlah dan variasi *dataset* juga berperan dalam memengaruhi hasil model. Dengan total 11.933 pasangan kalimat, *dataset* yang digunakan masih terbatas dalam menangkap seluruh variasi sintaksis dan semantik dalam bahasa Lampung. Meskipun model dapat belajar dari pola dalam data latih, keberagaman kalimat dalam data uji menyebabkan model mengalami kesulitan dalam menangani pola yang jarang muncul atau belum pernah dilihat sebelumnya. Hal ini dapat menjelaskan mengapa skor BLEU lebih rendah pada data uji dibandingkan dengan data latih, yang mengindikasikan adanya kemungkinan *overfitting*.

Kelemahan lain dari model berbasis LSTM adalah keterbatasannya dalam menangani ketergantungan kata dalam kalimat panjang. Meskipun LSTM lebih unggul dibandingkan RNN standar dalam mengatasi *vanishing gradient*, model ini masih mengalami kesulitan dalam mempertahankan konteks saat jumlah kata dalam satu kalimat bertambah. Fenomena ini tercermin dalam penurunan tajam skor BLEU pada *n-grams* yang lebih panjang, yang menunjukkan bahwa model belum sepenuhnya mampu mempertahankan urutan kata yang akurat dalam konteks kalimat utuh. Selain itu, proses *preprocessing* juga berpotensi memengaruhi kualitas model. Jika terdapat kesalahan dalam tokenisasi atau normalisasi teks, seperti pemisahan kata yang tidak sesuai dengan pola bahasa Lampung, model dapat mengalami kesulitan dalam memahami struktur bahasa secara akurat. Kesalahan ini berdampak pada prediksi kata berikutnya, terutama dalam kalimat yang mengandung kosakata yang jarang muncul dalam *dataset*.

Selain faktor struktural dan teknis, distribusi frekuensi kosakata dalam *dataset* juga berpengaruh terhadap hasil terjemahan. Kata-kata yang lebih sering muncul dalam data latih cenderung diterjemahkan dengan lebih akurat dibandingkan dengan kata-kata yang jarang ditemukan. Ketidakseimbangan ini dapat menyebabkan model lebih sering memilih kata-kata yang umum digunakan dalam data latih, yang pada akhirnya mengurangi fleksibilitas dan keakuratan terjemahan pada konteks yang lebih luas.

Berdasarkan berbagai tantangan yang dihadapi, beberapa langkah perbaikan dapat dilakukan di masa depan untuk meningkatkan performa model. Penambahan jumlah dan variasi *dataset* dapat membantu model memahami pola bahasa yang lebih kompleks, sementara teknik augmentasi data seperti parafrase otomatis atau penambahan variasi kalimat dapat memperkaya *dataset* tanpa harus mengumpulkan data secara manual dalam jumlah besar. Selain itu, eksplorasi arsitektur yang lebih kompleks seperti *Transformer* dapat menjadi alternatif untuk meningkatkan kemampuan model dalam memahami dependensi kata dalam kalimat panjang. Penggunaan metode *fine-tuning* pada model *pre-trained* berbasis seq2seq dengan *attention* juga dapat menjadi solusi untuk meningkatkan akurasi terjemahan. Dengan memahami penyebab kesalahan model secara lebih mendalam, langkah-langkah ini dapat diterapkan untuk menghasilkan sistem penerjemahan yang lebih akurat dan mampu menangani variasi bahasa Lampung dalam berbagai konteks.

4. KESIMPULAN

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa arsitektur *Long Short-Term Memory* (LSTM) menunjukkan efektivitas dalam menerjemahkan bahasa Lampung ke bahasa Indonesia dengan tingkat akurasi yang memadai. Evaluasi menggunakan metrik *Bilingual Evaluation Understudy* (BLEU) menunjukkan hasil terjemahan yang cukup baik, dengan skor BLEU tertinggi pada 1-grams, yang mengindikasikan bahwa model mampu menerjemahkan kata-kata individu dengan akurat. Namun, skor BLEU mengalami penurunan pada *n-grams* yang lebih panjang, yang menunjukkan bahwa model masih memiliki keterbatasan dalam mempertahankan struktur kalimat yang kompleks. Salah satu faktor utama yang memengaruhi performa model adalah ukuran dan variasi *dataset* yang digunakan. Dengan total 11.933 pasangan kalimat, *dataset* dalam penelitian ini masih terbatas dalam menangkap kompleksitas sintaksis dan semantik bahasa Lampung secara menyeluruh. Kurangnya variasi dalam data latih menyebabkan model kesulitan menangani struktur kalimat yang lebih kompleks, terutama pada data uji. Selain itu, keterbatasan model LSTM dalam menangani ketergantungan jangka panjang dalam kalimat juga berkontribusi terhadap penurunan akurasi pada *n-grams* yang lebih tinggi.

Selain keterbatasan tersebut, penelitian ini juga menghadapi beberapa hambatan selama proses pengembangan model. Salah satu tantangan utama adalah keterbatasan sumber data paralel untuk bahasa Lampung dan bahasa Indonesia. Tidak seperti bahasa yang lebih umum digunakan dalam penelitian *Natural Language Processing* (NLP), bahasa Lampung memiliki jumlah sumber teks paralel yang sangat terbatas, sehingga proses pengumpulan *dataset* memerlukan usaha tambahan, baik melalui pencarian sumber tertulis yang relevan maupun pembuatan *dataset* secara manual. Selain itu, variasi dialek dalam bahasa Lampung juga menjadi kendala dalam proses penerjemahan. Penelitian ini secara khusus menggunakan dialek Api, namun dalam praktiknya, terdapat beberapa variasi bahasa Lampung yang memiliki perbedaan leksikal maupun sintaktis yang cukup signifikan. Hal ini menyebabkan model yang dikembangkan belum dapat digunakan secara universal untuk semua penutur bahasa Lampung. Dari sisi teknis, pelatihan model dengan arsitektur LSTM memerlukan sumber daya komputasi yang cukup besar, terutama dalam menangani *dataset* yang semakin meningkat. Meskipun penelitian ini menggunakan

Google Colab Pro dengan GPU A100, waktu pelatihan tetap menjadi tantangan, terutama dalam eksperimen parameter dan proses validasi model. Hambatan lainnya adalah evaluasi model yang masih terbatas pada metrik otomatis seperti BLEU. Meskipun BLEU merupakan standar dalam evaluasi penerjemahan mesin, metrik ini tidak selalu mencerminkan kualitas terjemahan dari perspektif pengguna manusia. Evaluasi manual oleh penutur asli bahasa Lampung masih terbatas, sehingga dalam penelitian mendatang, keterlibatan penilai manusia dapat membantu memahami aspek kualitas terjemahan yang tidak dapat diukur hanya dengan BLEU.

Untuk penelitian di masa mendatang, disarankan agar *dataset* diperluas dengan menambahkan lebih banyak teks dalam kedua bahasa guna meningkatkan keanekaragaman dan kualitas data latih. Teknik augmentasi data, seperti parafrase otomatis atau pertukaran sinonim, dapat diterapkan untuk memperkaya *dataset* yang tersedia tanpa perlu mengumpulkan data secara manual dalam jumlah besar. Selain itu, eksplorasi arsitektur model yang lebih kompleks, seperti *Transformer*, dapat dilakukan untuk meningkatkan kemampuan model dalam menangani konteks kalimat panjang melalui mekanisme *self-attention*, yang lebih efektif dibandingkan pendekatan berbasis LSTM. Pendekatan ini juga dapat diperkuat dengan *transfer learning*, menggunakan model *pre-trained* yang telah dilatih pada bahasa dengan karakteristik serupa, serta *fine-tuning* pada model berbasis seq2seq dengan *attention* guna meningkatkan akurasi prediksi kata dalam konteks kalimat.

Selain peningkatan pada model dan *dataset*, penelitian lebih lanjut juga perlu difokuskan pada upaya mengatasi perbedaan struktur bahasa, yang dapat dilakukan melalui teknik *preprocessing* yang lebih canggih, seperti tokenisasi yang lebih presisi dan normalisasi teks yang lebih sesuai dengan karakteristik bahasa Lampung. Pendekatan evaluasi juga dapat diperluas dengan mengintegrasikan metrik tambahan, seperti METEOR atau *Translation Edit Rate* (TER), guna mendapatkan gambaran yang lebih komprehensif mengenai kualitas terjemahan, tidak hanya berdasarkan kesamaan *n-grams*, tetapi juga dari segi kefasihan dan kesesuaian makna. Selain itu, eksplorasi terhadap dialek lain dari bahasa Lampung juga dapat dilakukan untuk memperluas cakupan penelitian dan mendukung pengembangan sistem penerjemahan yang lebih umum bagi berbagai variasi bahasa Lampung.

Dengan menerapkan langkah-langkah tersebut, penelitian di masa depan diharapkan dapat menghasilkan sistem penerjemahan yang lebih akurat, fleksibel, dan mampu menangani variasi bahasa Lampung dalam berbagai konteks. Peningkatan ini tidak hanya akan meningkatkan performa model dalam penerjemahan bahasa Lampung ke bahasa Indonesia, tetapi juga menjadi bagian dari upaya pelestarian dan digitalisasi bahasa daerah yang terancam punah, serta mendukung komunikasi antarbudaya secara lebih luas.

DAFTAR PUSTAKA

[1] Putri, N. W. 2018. "Pergeseran Bahasa Daerah Lampung pada Masyarakat Kota Bandar Lampung." *Prasasti: Journal of Linguistics*, 3. 1, 83–97.

[2] Fauziyah, Y., Ilyas, R., and Kasyidi, F. 2022. "Mesin Penterjemah Bahasa Indonesia-Bahasa Sunda Menggunakan Recurrent Neural Networks." *Jurnal Teknoinfo*, 16. 2, 313.

[3] Suryani, A. A., et al. 2016. "Enriching English into Sundanese and Javanese Translation List Using Pivot Language." *Proceedings of the 2016 International Conference on Information & Communication Technology and Systems (ICTS)*, 167–171.

[4] Sujaini, H. 2018. "Peningkatan Akurasi Penerjemah Bahasa Daerah dengan Optimasi Korpus Paralel." *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 7. 1.

[5] Faqih, A. 2018. "Penggunaan Google Translate Dalam Penerjemahan Teks Bahasa Arab Ke Dalam Bahasa Indonesia." *ALSUNIYAT: Jurnal Penelitian Bahasa, Sastra, dan Budaya Arab*, 1. 2, 88–97.

[6] Shen, J., et al. 2019. "The Source-Target Domain Mismatch Problem in Machine Translation." *ArXiv*.

[7] Alqudsi, A., Omar, N., and Shaker, K. 2019. "A Hybrid Rules and Statistical Method for Arabic to English Machine Translation." *Proceedings of the 2nd International Conference on Computer Applications and Information Security (ICCAIS)*, IEEE, 2019.

[8] Singh, M., Kumar, R., and Chana, I. 2019. "Improving Neural Machine Translation Using Rule-Based Machine Translation." *Proceedings of the 7th International Conference on Smart Computing and Communications (ICSCC)*, 1–5.

[9] Zhang, J., et al. 2020. "Improving Neural Machine Translation Through Phrase-Based Soft Forced Decoding." *Machine Translation*, 34. 1, 21–39.

[10] Li, L., et al. 2016. "Combining Translation Memories and Statistical Machine Translation Using Sparse Features." *Machine Translation*, 30. 3–4, 183–202.

[11] Koehn, P. 2017. *Statistical Machine Translation*.

[12] Cuong, H., and Sima'an, K. 2017. "A Survey of Domain Adaptation for Statistical Machine Translation." *Machine Translation*, 31. 4, 455–481.

[13] Haji Sismat, M. A. 2019. "Neural and Statistical Machine Translation: A Comparative Error Analysis." *Conference: 17th International Conference on Translation*.

[14] Zhang, Y., and Liu, G. 2020. "Paragraph-Parallel Based Neural Machine Translation Model with Hierarchical Attention." *J Phys Conf Ser*, 1453. 1.

[15] Ortega, J. E., Castro Mamani, R., and Cho, K. 2021. "Neural Machine Translation with a Polysynthetic Low Resource Language." *Machine Translation*, 34. 4, 325–346.

[16] Sutskever, I., Vinyals, O., and Le, Q. V. 2014. "Sequence to Sequence Learning with Neural Networks." *Advances in Neural Information Processing Systems*, 3104–3112.

[17] Cho, K., et al. 2014. "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

[18] Garg, S., et al. 2019. "Jointly Learning to Align and Translate with Transformer Models." *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 4453–4462.

[19] Britz, D., et al. 2017. "Massive Exploration of Neural Machine Translation Architectures." *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 1442–1451.

[20] Klein, G., et al. 2017. "OpenNMT: Open-Source Toolkit for Neural Machine Translation." *Proceedings of ACL 2017, System Demonstrations*, 67–72. Vancouver, Canada: Association for Computational Linguistics.

[21] Liu, Y., et al. 2019. "A Simple but Effective Way to Improve the Performance of RNN-Based Encoder in Neural Machine Translation Task." *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, 416–421.

[22] Wang, X., Chen, C., and Xing, Z. 2019. "Domain-Specific Machine Translation with Recurrent Neural Network for Software Localization." *Empirical Software Engineering*, 24. 6, 3514–3545.

[23] Husada, I. N., and Toba, H. 2020. "Pengaruh Metode Penyeimbang Kelas Terhadap Tingkat Akurasi Analisis Sentimen pada Tweets Berbahasa Indonesia." *Jurnal Teknik Informatika dan Sistem Informasi*, 6. 2, 400–413.

[24] Qiu, J., Wang, B., and Zhou, C. 2020. "Forecasting Stock Prices With Long-Short Term Memory Neural Network Based on Attention Mechanism." *PLOS ONE*, 15. 1, 1–15.

[25] Corallo, L., et al. 2022. "A Framework for German-English Machine Translation with GRU RNN." *CEUR Workshop Proceedings*.

[26] Papineni, K., et al. 2002. "BLEU: a Method for Automatic Evaluation of Machine Translation." *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 311–318.

[27] Vaswani, A., et al. 2017 "Attention Is All You Need." *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 6000-6010.