

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam sebuah hubungan yang ada pada individu manusia, membutuhkan komunikasi atau sarana yang dapat menghubungkan antara individu dan kelompok pada manusia. Salah satunya menggunakan sebuah sarana komunikasi[1]. Dalam sebuah proses komunikasi, dikenal dengan adanya beberapa unsur komunikasi dan komunikator[1]. Hubungan yang terjalin antara komunikasi dan komunikator terjadi karena sebuah keinginan untuk menginteraksikan sebuah hal atau ingin menyampaikan sebuah pesan. Dengan adanya hubungan antara komunikasi dan komunikator, membutuhkan sebuah media yang unik sehingga menciptakan sebuah bahasa[1]. Bahasa menunjukkan sebuah ciri khas bangsa dan warga yang hidup dalam suatu negara atau bahkan daerah. Bahasa merupakan sebuah sistem simbol komunikasi yang bersifat arbitrer, memungkinkan individu atau seseorang dalam sebuah kebudayaan tertentu, atau orang lain yang telah mempelajari dan mendalami sebuah sistem kebudayaan tersebut untuk melakukan interaksi pada budaya tersebut untuk melakukan sebuah komunikasi[1].

Berbagai macam teori muncul darimana bahasa berasal dan mengapa bahasa bisa menjadi sebuah ciri khas sebuah kebudayaan sehingga manusia bisa mengerti tentang bahasa dari kebudayaan yang ada pada lingkungannya[2]. Dengan adanya banyak bahasa pada setiap negara atau bahkan daerah pada negara tersebut, menjadikan ciri khas dari daerah yang di tempati oleh sebuah individu maupun berkelompok[2]. Pada sebuah negara, terdapat berbagai macam bahasa daerah sehingga menjadi sebuah ciri khas dari daerah tersebut. Dalam suatu wilayah memungkinkan hidup dalam sebuah variasi bahasa secara berdampingan sehingga membentuk sebuah interaksi yang cenderung bersifat kode sehingga bahasa tersebut bersifat multilingual. Bahasa yang dimiliki oleh suatu masyarakat dalam tutur bahasanya selalu memiliki berbagai macam variasi[2].

Semua hal itu disebabkan oleh sebuah kenyataan bahwa sebuah bahasa yang hidup dalam masyarakat selalu digunakan dalam peran sosial oleh leluhur daerah tersebut[3]. Bahasa daerah dapat dikatakan sebagai sebuah citra suatu masyarakat yang berada dalam kehidupan[3]. Dalam sebuah bahasa daerah, terdapat kearifan lokal kebudayaan dalam daerah tersebut. Oleh sebab itu, bahasa daerah dapat diartikan sebagai cerminan suatu masyarakat[3].

Negara Indonesia sebagai bangsa dengan negara kepulauan memiliki banyak sekali bahasa daerah. Tercatat dalam petabahasa.kemdikbud.go.id , kurang lebih terdapat 748 bahasa daerah yang ada di Indonesia[3]. Dari sekian banyak bahasa daerah yang ada di Indonesia, tidak mungkin bisa menguasai semua bahasa daerah yang ada di Indonesia[3]. Salah satu bahasa daerah yang terdapat di Indonesia ialah bahasa Madura. Bahasa Madura merupakan suatu bahasa ibu bagi daerah Madura yang terdapat di bagian timur pada provinsi Jawa Timur[3]. Bahasa Madura juga merupakan bahasa ibu yang lumayan rumit dikarenakan setiap kata yang ada pada bahasa Madura ada simbol khusus dalam kata. Terkadang, warga lokal asli Madura kesulitan menggunakan bahasa Madura. Dan, para remaja saat ini lebih mengutamakan bahasa gaul dibandingkan dengan menggunakan bahasa daerah-nya.

Maka dari itu, dibutuhkan sebuah alat mesin penerjemah atau *translator machine* untuk menerjemahkan bahasa Madura ke bahasa Indonesia. Hal ini bertujuan untuk melestarikan kembali bahasa Madura sehingga para golongan anak muda lebih menyukai bahasa daerahnya dibandingkan dengan menggunakan bahasa prokem atau bahasa gaul. *Translator machine* atau mesin translator merupakan sebuah pembelajaran mesin untuk menerjemahkan teks atau sebuah ucapan dari bahasa alami ke bahasa lainnya[4]. Pada sebuah tingkat dasar, mesin penerjemah melakukan sebuah pertukaran atau substitusi sederhana terhadap kata-kata dari sebuah bahasa alami ke bahasa lainnya[4]. Namun, dengan cara sederhana ini tidak dapat menghasilkan bahasa yang baik dan sebuah frasa yang utuh. Maka, dengan menggunakan sebuah teknik korpus, statistik dan pembelajaran mesin untuk menghasilkan sebuah mesin penerjemah yang baik sehingga menciptakan mesin penerjemah dengan akurasi dan frasa yang baik[4].

Mesin penerjemah yang banyak dikenal oleh masyarakat awam ialah mesin penerjemah *google*. *Google translate* memang mudah untuk mengaksesnya, namun masih terdapat 2 bahasa daerah yang digunakan dalam *google translate*, yang terdiri dari bahasa Jawa dan bahasa Sunda[5]. Bahasa Madura masih belum tersedia dalam *google translate*[5].

Dalam pendekatan metode dalam mesin penerjemah, terdapat 2 metode yang sering digunakan dalam pengimplementasiannya. 2 metode tersebut terdiri dari SMT (*Statistical Machine Translator*) dan NMT (*Neural Machine Translator*)[4]. *Statistical Machine Translator* atau SMT merupakan sebuah pendekatan mesin penerjemah yang berdasarkan pada model statistik pada sebuah parameter yang parameternya berasal dari analisa corpora teks (badan teks) sebuah dwibahasa, yaitu sebuah teks sumber yang belum diterjemahkan[4].

Pendekatan SMT dimulai dengan mengumpulkan data dari beberapa terjemahan sebelumnya yang sangat besar dan disetujui, dapat dikenal dengan korpus teks (merupakan bentuk jamak dari corpus teks)[4]. Lalu, model ini diterapkan pada sebuah teks tujuan yang tidak diterjemahkan untuk membuat sebuah kecocokan berdasarkan dari probabilitas untuk menghasilkan sebuah terjemahan yang masuk akal sehingga membentuk frasa yang baik.

Neural Machine Translator atau NMT adalah pendekatan dalam mesin penerjemah yang masih baru dengan menggunakan sebuah arsitektur RNN pada bagian *encoder* dan *decoder*-nya[5]. *Encoder* berfungsi sebagai mengkodekan sebuah data bilangan dengan beberapa *Format* tertentu. *Encoder* merupakan rangkaian sebuah kombinasi dalam gerbang digital yang memiliki *output* sedikit dalam *Format* bilangan biner[5]. *Encoder* akan mengkodekan sebuah jalur *Input* menjadikannya bilangan biner[5]. Sedangkan *decoder* merupakan mengubah kembali data hasil dari *encoder* menjadi bahasa yang dapat dikenali oleh manusia[5]. Dalam NMT terdapat arsitektur RNN memiliki sebuah kapabilitas dalam merepresentasikan sebuah *history words* atau mengingat kata-kata yang telah dipelari-nya dalam arsitektur RNN sehingga menjadikan RNN memiliki kelebihan dibandingkan dengan pendekatan mesin penerjemah yang lain[5].

Model pada RNN bergantung pada model *encoder* dan *decoder*. Pada pengembangan mesin penerjemah RNN menggunakan LSTM, model *encoder* dan *decoder* disebut juga *sequence to sequence*[5]. RNN juga dinilai mampu menghadapi sebuah data yang berurutan atau sequential, seperti contoh video (sequence dari beberapa frame), sebuah kalimat dan lain-lain. Seperti contoh, jika di dalam kalimat terdapat 7 kata, maka RNN akan melakukan 7 kali setiap *timestep*. *Encoder* dalam RNN berfungsi sebagai membaca *Input* dari beberapa *sequence*, kata demi kata sehingga dapat menghasilkan sebuah *context* yang lalu akan mengambil esensinya dari hasil *Input* tersebut[5]. Dari hasil *context* ini, *decoder* akan menghasilkan sebuah *output sequence*. Setiap kata setiap *timestep*nya dan kata yang akan keluar berikutnya akan dipengaruhi oleh kata sebelumnya[5].

Pada *machine translation*, terdapat berbagai macam penggabungan metode pada *neural machine translation*, seperti contoh RNN-*Attention*, dan RNN-LSTM. RNN-*Attention* merupakan upaya dalam mempelajari sebuah perepresentasian vektor tunggal pada kalimat dan mengarahkan pada sebuah fokus dan faktor penting hingga dapat memprediksi nilai *output* dalam model *sequential to sequential*[5]. Kelebihan model RNN-*Attention* adalah kemampuannya dalam menangkap sebuah konteks, peningkatan kinerja untuk menunjang hasil akurasi pada RNN tradisional, dan dapat digabungkan dengan LSTM maupun *TransFormer*[5]. Kekurangan model RNN-*Attention* adalah kompleksnya perhitungan pada mekanisme *Attention*, penggunaan *resource* atau sumber daya yang besar pada model RNN-*Attention* karena harus menyimpan informasi pada nilai *Input*, beberapa pengaturan parameter tambahan pada model RNN-*Attention* seperti nilai bobot dan jumlah data latih yang besar[5].

LSTM atau *Long Short Term Memory* merupakan bagian dari RNN dalam pembuatan mesin penerjemah dan merupakan tipe spesial dalam *neural network*[6]. LSTM terdiri dari neuron yang akan berfungsi sebagai sel. LSTM disebut sebagai arsitektur alternatif untuk menangani sebuah masalah dalam RNN konvensional. LSTM pertama kali diperkenalkan pada tahun 1997, namun telah disempurnakan sebagai penggunaan skala praktis besar pada tahun 2014[6]. Dalam pengimplementasian LSTM, menggunakan bahasa pemrograman python.

Penggunaan bahasa python merupakan bahasa pemrograman yang mudah dipahami dan dalam *module* python terdapat banyak sekali *library* sehingga memudahkan dalam proses pembelajaran mesin[6].

Kelebihan RNN-LSTM adalah mengatasi masalah *vanishing* dan *exploding gradient*, yaitu apabila terjadi sebuah perubahan pada nilai jangkauan dari satu lapisan menuju lapisan berikutnya pada sebuah arsitektur RNN, nilai gradien dapat mengecil pada saat propagasi mundur, sehingga informasi dilupakan[7]. Sehingga LSTM menggunakan gerbang atau *gate* khusus untuk mengatur aliran informasi dan mencegah gradien yang menghilang[7].

Pada penanganan data sekuensial, LSTM dapat membantu kinerja pada arsitektur RNN seperti dalam bentuk data teks, audio maupun deret waktu[7]. Kemampuan LSTM untuk mengingat sebuah informasi yang berasal pada nilai *Input*, membuat sebuah struktur pada data sekuensial memungkinkan LSTM dapat menangkap struktur dan konteks yang ada pada *Dataset* yang digunakan pada mesin penerjemah[7].

Dan, RNN-LSTM sangat minim sumber daya pada saat melakukan pemrosesan dalam mesin penerjemah sehingga pada saat melakukan komputasi dan ini membuat fitur pada RNN-LSTM lebih baik dibandingkan dengan menggunakan RNN-Attention[5]. RNN-LSTM lebih baik dalam pemahaman konteks dan fleksibilitas pada data panjang sehingga dapat menangani data pada jumlah waktu yang berbeda tanpa perlu mengubah perubahan besar pada setiap arsitektur[7].

Arsitektur yang terdapat pada sebuah rancangan LSTM terdiri dari lapisan *Input*, lapisan *output*, dan lapisan tersembunyi. Lapisan tersembunyi terdiri dari sel memori, satu sel memori memiliki tiga *gate* yaitu *Input gate*, *forget gate*, *output gate*[7]. *Input gate* berfungsi mengontrol berapa banyak informasi yang harus disimpan dalam keadaan sel[7]. Ini mencegah sel dari menyimpan data yang tidak perlu. *Forget gate* berfungsi mengontrol sejauh mana nilai tetap di dalam sel memori[7]. *Output gate* berfungsi untuk memutuskan berapa banyak konten atau nilai dalam sel memori, digunakan untuk menghitung *output*.

Pada proses pemrosesan teks, dalam NLP atau (*Natural Language Processing*) terdapat berbagai macam proses, diantaranya *tokenization* merupakan sebuah proses pemecahan teks menjadi sebuah kata-kata dalam sebuah kalimat, *stopword removal* merupakan tahapan menghilangkan sebuah kata umum sehingga hanya menyisakan kata khusus yang mengandung informasi, *lowercase* merupakan proses yang dimana mengubah semua kata-kata huruf kapital atau huruf besar menjadi huruf kecil, *lemmatization and stemming* merupakan sebuah proses pemotongan atau penghilangan sebuah kata imbuhan sehingga kembali pada bentuk kata tanpa sebuah imbuhan, dan *part of speech tagging* merupakan proses untuk menambahkan sebuah kata atau tanda tulisan sesuai dengan bagian yang diperoleh, seperti contoh pada kata sifat, kata kerja, kata benda, dan lain-lain[8].

Pada bahasa Madura, terdapat sebuah kata atau kalimat yang mempunyai huruf simbol. Menurut Pedoman Umum Ejaan Bahasa Madura Yang Disempurnakan pada tahun 2012 edisi revisi, beberapa huruf vokal yang terdapat yaitu: â halus, a tajam, a dungu, i halus, i miring, u halus, u miring, e bisu, è miring, e halus, è tajam, è dungu, o tajam dan o miring[9]. Dimana hal ini menunjukkan penggunaan huruf vokal yang ada pada bahasa Madura mempunyai beberapa huruf simbol sehingga memerlukan perlakuan khusus pada saat pemrosesan teks.

Pada saat melakukan pemrosesan teks untuk huruf simbol atau *Accented Characters*, mempunyai beberapa metode. Metode yang digunakan untuk *Accented Characters* adalah menggunakan *Unicode Normalization*[10]. *Unicode Normalization* merupakan proses standarisasi pada sebuah huruf berkarakter dan huruf dengan tanda aksent sehingga melalui proses standarisasi dan didekomposisi sehingga dapat direpresentasikan dalam bentuk huruf yang diinginkan tanpa menghapus huruf berkarakter dan huruf yang memiliki tanda aksent[10].

Pada *Unicode Normalization* terdapat 4 metode, diantaranya NFD (*Normalization Form Canonical Decomposition*) merupakan mendekomposisi sebuah karakter yang telah terkomposisi menjadi karakter dasar dan komponen

tanda aksen dan NFD digunakan dalam proses seperti pencarian dan pengindeksan teks[10]. NFC (*Normalization Form Canonical Composition*) merupakan sebuah bentuk normalisasi paling umum yang digunakan pada teks yang ditujukan sebagai tampilan dan pertukaran, sebagai contoh “è” akan direpresentasikan sebagai “e”[10]. NFKD (*Normalization Form Compatibility Decomposition*) merupakan bentuk normalisasi *Unicode* dengan mengganti sebuah karakter-karakter dengan sebuah ekivalen kompatibilitas dengan mendekomposisikan karakter-karakter khusus, Dalam bentuk normalisasi NFKD, karakter-karakter dengan tanda aksen, karakter lebar penuh, dan karakter khusus serupa akan diubah menjadi bentuk dasar atau karakter-karakter kompatibel sesuai dengan standar *Unicode*[10]. Dan NFKC (*Normalization Form Compatibility Composition*) merupakan normalisasi yang menggantikan karakter dengan karakter kompatibilitas dan mengkomposisikan karakter-karakter yang seharusnya menjadi sebuah satu karakter, sebagai contoh "C" akan diubah menjadi karakter lebar setengah yang setara dengan "C." Demikian pula, karakter dengan tanda aksen, karakter gantung, dan karakter kompatibel lainnya akan diperlakukan agar sesuai dengan karakter dasar yang diwakili[10].

Parameter yang terdapat pada RNN-LSTM sesuai dengan penelitian terjemahan bahasa Inggris ke bahasa Sunda terdapat *Optimizer* menggunakan *Adam*, *Loss* menggunakan *Categorical Crossentropy*, *Epochs* menggunakan nilai sebesar 200, *validation split* menggunakan nilai sebesar 0.1 dimana nilai *training* dan *testing* sebesar 90:10, dan *callback* menggunakan *early stopping*[11]. Nilai parameter berfungsi sebagai pemodelan pada RNN-LSTM dan nilai sebuah parameter dapat berubah sesuai dengan penelitian yang dilakukan, sehingga memiliki hasil dan nilai akurasi yang baik. Hasil dari nilai akurasi terjemahan menggunakan *Bleu-score*[11].

Dalam penelitian ini, juga dilakukan percobaan dengan menambahkan *cross validation* sebagai metodologi evaluasi. *Cross-validation* adalah metode evaluasi yang penting dalam konteks pembelajaran mesin dan pemodelan statistik, termasuk dalam penerjemahan mesin neural (Neural Machine Translation, NMT). Metode ini digunakan untuk memastikan bahwa model tidak hanya menunjukkan

kinerja yang baik pada data training, tetapi juga memiliki kemampuan generalisasi yang kuat pada data yang tidak terlihat sebelumnya.

Jumlah *cross-validation* dalam penelitian ini berjumlah 5-fold, yang dimana berfungsi sebagai teknik evaluasi untuk mengukur kinerja model secara menyeluruh dan mengurangi kemungkinan *overfitting*. Dalam 5-fold *cross-validation*, *dataset* dibagi menjadi lima subset (*folds*) yang sama besar. Setiap fold secara bergantian digunakan sebagai data *testing*, sementara empat *fold* lainnya digunakan sebagai data *training*. Proses ini diulang sebanyak lima kali sehingga setiap fold digunakan sekali sebagai data *testing* dan empat kali sebagai data *training*.

Fungsi utama dari 5-fold *cross-validation* adalah untuk memberikan estimasi kinerja model yang lebih akurat dan dapat diandalkan. Dengan membagi *dataset* menjadi beberapa *fold*, model diuji pada berbagai subset data yang berbeda, sehingga hasil evaluasi tidak bergantung pada satu subset data tertentu. Ini membantu memastikan bahwa model memiliki kemampuan generalisasi yang baik dan tidak hanya bekerja baik pada data *training* tetapi juga pada data yang belum pernah dilihat sebelumnya.

Selain itu, 5-fold *cross-validation* membantu dalam mengidentifikasi kombinasi parameter pelatihan yang optimal. Dalam konteks penelitian ini, variasi parameter seperti *Batch Size* dan *Epochs* dievaluasi menggunakan *cross-validation* untuk menemukan kombinasi yang memberikan kinerja terbaik berdasarkan BLEU score. Dengan melakukan evaluasi pada beberapa *fold*, peneliti dapat mengukur bagaimana setiap kombinasi parameter mempengaruhi kinerja model pada data yang berbeda, memberikan wawasan yang lebih mendalam tentang keefektifan parameter tersebut.

Masalah penelitian ini adalah bagaimana cara membedakan kalimat dalam bahasa Madura yang memiliki huruf yang berbeda, sehingga mesin penerjemah bisa membedakan antara kata yang memiliki simbol yang berbeda dengan kata yang tidak memiliki simbol berbeda. Karena, dalam bahasa Madura berbeda huruf akan mengubah arti atau makna dari kata tersebut. Dan, bagaimana cara

membedakan tingkat bahasa pada bahasa Madura yang memiliki 3 tingkat, yaitu bahasa “*Engghi Bunten*”, bahasa “*Engghi Enten*”, dan bahasa “*Enje’ iye*”.

1.2 Rumusan Masalah

Berdasarkan uraian dalam latar belakang, maka permasalahan yang menjadi perhatian dalam penelitian ini adalah bagaimana membuat sebuah mesin penerjemah dari bahasa Madura ke bahasa Indonesia menggunakan rancangan LSTM atau (*Long Short Term Memory*). Pada saat pembuatan mesin penerjemah, dan bagaimana merepresentasikan sebuah data teks menjadi data model sehingga bisa dilakukan proses *preprocessing*. Bagaimana hasil implementasi mesin penerjemah dengan menggunakan *deep learning* dan LSTM, dan apa kelebihan mesin penerjemah menggunakan *neural netowrk* dibandingkan mesin penerjemah menggunakan metode *statistical*.

1.3 Pertanyaan Penelitian

- a. Apakah nilai parameter pada model mempengaruhi hasil akurasi terjemahan ?
- b. Apakah peningkatan jumlah *epochs* selalu meningkatkan kinerja model terjemahan ?
- c. Seberapa efektif *5-fold cross-validation* dalam memberikan evaluasi kinerja yang akurat pada model terjemahan ?

1.4 Tujuan dan Manfaat Penelitian

Tujuan penelitian ini adalah untuk mengevaluasi kinerja model *Neural Machine Translation* (NMT) dalam menerjemahkan teks dari bahasa Madura ke bahasa Indonesia. Penelitian ini bertujuan untuk menilai dampak variasi parameter pelatihan, seperti *Batch Size* dan jumlah *Epochs*, terhadap kualitas terjemahan yang diukur menggunakan BLEU score. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi efektivitas *5-fold cross-validation* dalam memberikan estimasi kinerja yang akurat dan andal untuk model NMT. Dengan demikian, penelitian ini akan mengidentifikasi kombinasi parameter pelatihan yang optimal untuk meningkatkan performa model dalam konteks penerjemahan bahasa Madura.

Manfaat dari penelitian ini mencakup kontribusi terhadap pengembangan teknologi penerjemahan dengan menyediakan wawasan mengenai parameter pelatihan yang efektif dan meningkatkan kualitas alat penerjemahan otomatis. Penelitian ini juga berperan dalam pelestarian bahasa daerah, khususnya bahasa Madura, dengan mempermudah aksesibilitas dan penggunaan bahasa tersebut dalam kehidupan sehari-hari. Dengan hasil penelitian ini, diharapkan bahasa Madura dapat lebih mudah dipahami dan digunakan oleh generasi muda serta masyarakat lokal, mendukung keberagaman bahasa dan budaya serta meningkatkan komunikasi dalam konteks lokal.

1.5 Batasan Masalah

Batasan masalah yang ada pada penelitian ini adalah:

- a. Mesin penerjemah akan menerjemahkan bahasa Madura ke bahasa Indonesia dengan tingkatan bahasa “*Engghi Bunten*” dan bahasa “*Engghi Enten*” atau tingkatan bahasa halus dan tingkatan bahasa menengah.
- b. Penggunaan pembatasan daerah bahasa Madura menggunakan rumpun daerah pulau Madura, diantaranya daerah Sumenep, Pamekasan, Sampang, Bangkalan

1.6 Sistematika Penulisan

Sistematika penulisan dibuat untuk menjelaskan secara singkat pembahasan yang ada dalam penelitian melalui gambaran isi dari setiap bab. Dalam penelitian ini, terdiri dari 5 bab, diantaranya:

1. BAB I PENDAHULUAN

Bagian ini berisi sub-bab diantaranya, latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, pertanyaan penelitian, dan sistematika penulisan.

2. BAB II TEORI PENUNJANG

Bagian bab ini berisi tentang beberapa teori yang membantu dalam proses penelitian dan uraian beberapa hal yang terkait pada penelitian.

3. BAB III PERANCANGAN SISTEM

Pada bab ini, menjelaskan langkah-langkah yang akan dilakukan pada penelitian serta tahapan pada sistem mesin penerjemah dan menjelaskan tentang pembuatan model pada mesin penerjemah.

4. BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Bab ini memaparkan hasil penelitian dan analisis data yang telah dikumpulkan. Hasil dibandingkan dengan tujuan atau hipotesis awal, dengan interpretasi dan implikasi temuan yang dijelaskan. Bab ini juga membahas keterbatasan penelitian dan faktor yang mempengaruhi hasil.

5. BAB V PENUTUP

Bab ini merangkum kesimpulan dari penelitian, menyatakan apakah tujuan atau hipotesis tercapai. Selain itu, bab ini memberikan rekomendasi atau saran untuk penelitian lebih lanjut dan aplikasi praktis, serta saran untuk perbaikan di masa mendatang.