

PERBANDINGAN KINERJA SISTEM NEURAL MACHINE TRANSLATION OPENNMT DAN THUMT DALAM EKSPERIMEN MENERJEMAHKAN BAHASA JAWA NGOKO – KRAMA

Leni Purwaningsih; Endang Wahyu Pamungkas, S.Kom., M.Kom., Ph.D
Informatika, Fakultas Komunikasi dan Informatika,
Universitas Muhammadiyah Surakarta

Abstrak

Penelitian ini dibuat untuk menguji pendekatan NMT dalam menerjemahkan bahasa Jawa ngoko – krama menggunakan OpenNMT dan THUMT. Kedua sistem tersebut dibandingkan melalui aspek ketepatan hasil terjemahan yang akan dinilai melalui pengukuran skor BLEU (Bilingual Evaluation Understudy). Langkah-langkah penerjemahan bahasa Jawa ngoko – krama menggunakan OpenNMT dan THUMT antara lain membuat *dataset*, membangun kosakata, membuat model NMT, menerjemahkan data uji, mengevaluasi hasil terjemahan menggunakan skor BLEU, dan membuat aplikasi penerjemahan menggunakan model NMT. Hasil eksperimen ini menunjukkan *dataset* yang terdiri dari sepasang data latih yang berisikan 1000 kalimat dan sepasang data validasi yang berisikan 500 kalimat, mampu menghasilkan skor BLEU untuk OpenNMT sebesar 98,55 dan THUMT sebesar 99,47 dari terjemahan data uji yang memiliki kalimat terdapat di data latih. Sementara itu untuk terjemahan dari data uji yang memiliki kalimat tidak terdapat di data latih, mampu menghasilkan skor BLEU untuk OpenNMT sebesar 2,01 dan THUMT sebesar 1,6. Aplikasi penerjemahan bahasa Jawa ngoko - krama berhasil dibuat menggunakan bahasa pemrograman Python dan diberi nama “Nerjemahake”.

Kata Kunci: bahasa Jawa, NMT, penerjemahan, OpenNMT, THUMT.

Abstract

This research was conducted to test the NMT approach in translating Javanese language from ngoko to krama using OpenNMT and THUMT. Both systems were compared based on the accuracy of translation results, which were evaluated using BLEU (Bilingual Evaluation Understudy) scores. The steps of translating Javanese language from ngoko to krama using OpenNMT and THUMT include creating a dataset, building a vocabulary, creating an NMT model, translating test data, evaluating the translation results using BLEU scores, and creating a translation application using the NMT model. The experimental results show that a dataset consisting of a pair of training data containing 1000 sentences and a pair of validation data containing 500 sentences, can produce BLEU scores of 98,55 for OpenNMT and 99,47 for THUMT for the translation of test data that contains sentences found in the training data. Meanwhile, for the translation of test data that contains sentences not found in the training data, the BLEU scores are 2,01 for OpenNMT and 1,6 for THUMT. A Javanese language translation application from ngoko to krama was successfully created using Python programming language and named "Nerjemahake".

Keywords: Javanese language, NMT, translation, OpenNMT, THUMT.

1. PENDAHULUAN

1.1. Latar Belakang

Bahasa daerah merupakan salah satu kekayaan bangsa yang berfungsi sebagai alat komunikasi bagi masyarakat suatu suku di Indonesia. Perkembangan Bahasa Indonesia sebagai bahasa persatuan dan sebagai bahasa negara tidak akan dapat dipisahkan dari perkembangan bahasa-bahasa daerah. Bahasa daerah diyakini menjadi penyumbang kosakata terbesar dalam Bahasa Indonesia (Sukriayu dkk., 2017). Bahasa daerah merupakan salah satu unsur penting dari keberlanjutan lestariannya suatu suku. Apabila selalu dilestarikan, bahasa daerah dapat terus digunakan kepada generasi selanjutnya sehingga terhindar dari kepunahan.

Salah satu bahasa daerah yang memiliki penutur terbanyak di Indonesia adalah Bahasa Jawa. Bahasa Jawa memiliki tingkatan yang berbeda pada penggunaannya, dua yang sering digunakan adalah bahasa Jawa ngoko dan Jawa krama. Bahasa Jawa ngoko digunakan ketika seseorang berbicara kepada seseorang lain yang sudah akrab atau seumuran. Sedangkan bahasa Jawa krama atau juga disebut bahasa Jawa halus digunakan ketika seseorang berbicara kepada orang tua atau orang yang lebih tua. Kaidah ini diartikan sebagai tata krama dan sopan santun dalam berbahasa Jawa yang harus ditaati atau dinamakan *unggah-ungguhing basa*. Jika seseorang berbahasa Jawa dengan orang lain dengan tidak tepat tataran yang digunakan, maka pergaulan dengan orang lain menjadi terganggu, tidak serasi, dan tidak harmonis (Purwadi, 2020). Menurut penelitian yang dilakukan oleh Hartanti dan Kaltsum (2019), setidaknya terdapat 3 dimensi nilai kearifan lokal budaya Jawa yang perlu diterapkan pada buku pembelajaran bahasa Jawa di sekolah dasar, yaitu nilai keberagaman, nilai kemandirian, serta nilai kesusilaan. Sebab, dengan memahami dan menerapkan nilai kearifan lokal tersebut, siswa dapat lebih memahami budaya Jawa dan dapat menghormati norma dan etika dalam berbahasa Jawa, yang tercermin dalam *unggah-ungguhing basa*. Dari pernyataan tersebut, dapat diketahui bahwa *unggah-ungguhing basa* juga merupakan hal penting yang harus dipahami dan diterapkan pada pembelajaran bahasa Jawa kepada siswa sekolah.

Namun di kehidupan sehari-hari, sering dijumpai seseorang penutur bahasa Jawa yang kesulitan menggunakan bahasa Jawa krama ketika melakukan percakapan. Mereka yang kesulitan acapkali mencampurkan kosakata bahasa Jawa krama dengan Jawa ngoko, atau bahkan bahasa Indonesia ketika berbicara, meskipun lawan bicaranya adalah orang yang lebih tua. Hal tersebut tentu tidak sesuai dengan kaidah *unggah-ungguhing basa*. Kesulitan penutur bisa disebabkan oleh penggunaan bahasa Jawa krama yang tidak sering digunakan atau tidak dibiasakan (Fardani & Wiranti, 2019). Kalau terus tidak dibiasakan, maka *unggah-ungguhing basa* dapat berakhir kepunahan dan hilang di kehidupan sehari-hari, sehingga tata krama dan

kesopanan yang diterapkan di dalam bahasa Jawa akan senantiasa hilang perlahan dan tidak dapat diteruskan ke generasi selanjutnya.

Kepunahan bahasa dapat dicegah melalui beragam cara dan beragam pula metodenya. Teknologi bisa menjadi opsi dalam melestarikan bahasa karena merupakan gaya hidup seluruh masyarakat masa sekarang dan masa depan yang mana tidak akan terlepas dari perkembangan teknologi. Salah satu pemanfaatan teknologi dalam melestarikan bahasa adalah menggunakan mesin penerjemahan bahasa otomatis. Mesin penerjemahan bahasa otomatis sejatinya sudah banyak dipasarkan dan ramai digunakan oleh masyarakat untuk membantu percakapan secara cepat antara seseorang dengan seseorang lain yang memiliki pemahaman bahasa berbeda. Sebagai contoh, Google memiliki aplikasi bernama Google Translate yang menyimpan berbagai paralel antar bahasa di dunia dan menjadi pilihan utama bagi masyarakat untuk menerjemahkan bahasa. Masyarakat umum, khususnya di Indonesia sangat terbantu dengan adanya penerjemahan bahasa otomatis dari Google Translate, sesuai survei yang dilakukan oleh Thamrin (2017) yang menyatakan sejumlah responden mereka terkesan dengan performa dari Google Translate. Google menggunakan pendekatan *Neural Machine Translation* (NMT) untuk meningkatkan performa Google Translate di tahun 2016, yang mana mereka perkenalkan dengan nama Google Neural Machine Translation (GNMT) (Wu dkk., 2016).

NMT adalah pendekatan penerjemahan otomatis yang menjanjikan potensi dalam mengatasi banyak kekurangan sistem terjemahan mesin tradisional. Kekuatan NMT terletak pada kemampuannya untuk belajar secara langsung, secara *end-to-end* memetakan dari teks masukan untuk terhubung ke teks luaran. Arsitektur NMT biasanya terdiri dari dua jaringan saraf berulang atau disebut *Recurrent Neural Networks* (RNNs), satu untuk mengkonsumsi urutan teks masukan dan satu lagi untuk menghasilkan teks keluaran yang diterjemahkan (Wu dkk., 2016). Selain GNMT dari Google yang menggunakan pendekatan NMT dalam melakukan penerjemahan otomatis, terdapat pula sistem NMT berbasis *open source* yang dapat mempermudah penelitian serta pengujian performa NMT itu sendiri. Sistem NMT tersebut diantaranya adalah OpenNMT dan THUMT. OpenNMT diciptakan oleh kerja sama antara *Natural Language Processing Group* (perkumpulan mahasiswa, dosen, atau peneliti lainnya yang membahas berkaitan dengan pemrosesan bahasa alami) di Harvard University dan SYSTRAN sebagai implementasi NMT terlengkap. Di luar dari menyediakan kode untuk tugas penerjemahan inti, OpenNMT dirancang dengan tiga tujuan: (a) memprioritaskan pelatihan pertama dan efisiensi pengujian, (b) mempertahankan modularitas dan keterbacaan model, dan (c) mendukung ekstensibilitas penelitian yang signifikan (Klein dkk., 2017). THUMT dikembangkan oleh *Natural Language Processing Group* di Tsinghua University. THUMT

mengimplementasikan *encoder-decoder* berbasis perhatian standar kerangka kerja melalui pustaka Python bernama Theano dan mendukung tiga kriteria pelatihan: maksimum estimasi kemungkinan, pelatihan risiko minimum, dan pelatihan semi-supervisi (Zhang dkk., 2017). Penelitian oleh Abidin (2017) terkait penerapan NMT untuk menerjemahkan bahasa Lampung – Indonesia juga menggunakan THUMT sebagai sistem NMT yang digunakan dalam melakukan eksperimen penerjemahan.

Kedua sistem NMT tersebut berstatus aktif pada saat penelitian ini dibuat. Maka dari itu, dalam membangun sistem penerjemahan otomatis bahasa Jawa ngoko – krama, penelitian ini dibuat untuk menguji pendekatan NMT dalam menerjemahkan bahasa Jawa ngoko – krama menggunakan OpenNMT dan THUMT. Keduanya akan dibandingkan melalui aspek ketepatan hasil terjemahan yang akan dinilai melalui pengukuran skor BLEU (Bilingual Evaluation Understudy). BLEU merupakan algoritma yang digunakan untuk mengevaluasi kualitas dari hasil terjemahan bahasa yang telah diterjemahkan oleh mesin. BLEU menghitung skor presisi statistik yang dikoreksi antara hasil terjemahan otomatis dan terjemahan rujukan menggunakan konstanta yang dinamakan *brevity penalty* (Abidin, 2017).

Hipotesis dari penelitian ini adalah untuk mengidentifikasi kelebihan dan kekurangan dari sistem NMT dalam melakukan penerjemahan otomatis serta berkontribusi dalam upaya melestarikan bahasa Jawa. Sehingga bahasa daerah ini terhindar dari kepunahan dan memudahkan masyarakat dalam berkomunikasi sesuai dengan *unggah-ungguhing basa*.

1.2. Rumusan Masalah

Melalui pemaparan latar belakang di atas, rumusan masalah penelitian ini akan membahas beberapa hal sebagai berikut:

1. Apa yang perlu diuji untuk menerjemahkan kalimat bahasa Jawa ngoko – krama?
2. Bagaimana perbandingan proses kerja antara sistem OpenNMT dan THUMT?
3. Bagaimana performa yang dihasilkan dari pendekatan NMT dalam melakukan penerjemahan?

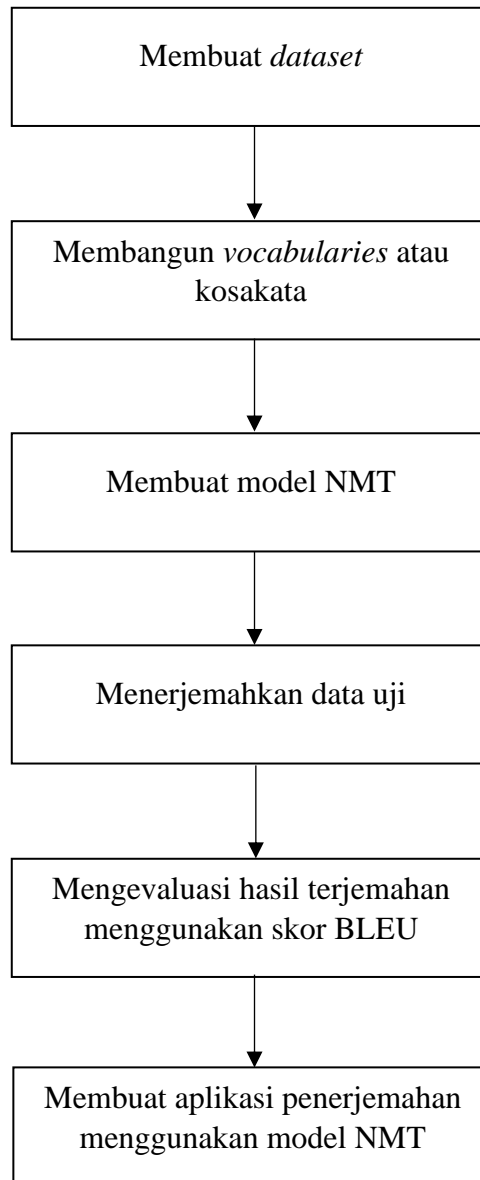
1.3. Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

1. Membuat dan menguji *dataset* penerjemahan bahasa Jawa ngoko – krama.
2. Membandingkan kinerja sistem OpenNMT dan THUMT dalam menerjemahkan kalimat bahasa Jawa ngoko – krama.
3. Mengukur performa dari hasil terjemahan menggunakan NMT.

2. METODE

Metode penelitian yang digunakan adalah metode kuantitatif dengan membandingkan kinerja dua sistem NMT yaitu OpenNMT dan THUMT melalui hasil evaluasi menggunakan skor BLEU. Langkah-langkah penerjemahan bahasa Jawa ngoko – krama menggunakan OpenNMT dan THUMT disajikan pada *flowchart* atau diagram alir di bawah ini:



Gambar 1. Tahapan menerjemahkan bahasa Jawa ngoko – krama

2.1. Membuat *dataset*

Penelitian ini menggunakan *dataset* yang berisikan kalimat berbahasa Jawa ngoko sebagai *source* atau sumber masukan dan kalimat terjemahannya berbahasa Jawa krama sebagai *target* atau target luaran. *Dataset* tersebut dibagi menjadi 3 komponen, yaitu data latih, data validasi,

dan data uji. Data latih digunakan untuk melatih model NMT, sedangkan data validasi digunakan untuk menyeleksi dan mengoptimisasi algoritma pembelajaran dari model NMT. Berbeda dengan data latih dan validasi, data uji digunakan untuk menguji serta mengevaluasi performa dari data yang telah dilatih sebelumnya.

2.2. Membangun *vocabularies* atau kosakata

Kosakata diperlukan untuk membuat model NMT. Berkas ini dibangun menggunakan data latih dan data validasi dari *dataset* yang telah dibuat sebelumnya. Pada tahap ini, kalimat-kalimat akan mengalami proses tokenisasi atau teknik memecah kalimat ke dalam beberapa kata (Thalib, 2019). Selanjutnya kata-kata tersebut akan dimasukkan ke dalam berkas kosakata dan diteruskan ke tahap pembuatan model NMT.

2.3. Membuat model NMT

Langkah ini akan melibatkan berkas kosakata untuk diproses menjadi model NMT. Dalam proses pembuatan model tersebut membutuhkan satu unit pemrosesan grafis agar pembuatan model oleh sistem berjalan lebih cepat. Dari proses ini akan menghasilkan berkas model NMT yang berikutnya digunakan sebagai landasan sistem NMT dalam menerjemahkan data uji.

2.4. Menerjemahkan data uji

Data uji yang akan diterjemahkan adalah data uji berbahasa Jawa ngoko. Sistem NMT akan memprediksi hasil terjemahan kalimat pada data uji berdasarkan model NMT yang telah dibuat sebelumnya. Setelah proses penerjemahan selesai, luaran yang dihasilkan adalah berkas yang berisikan hasil terjemahan menggunakan sistem NMT dalam bahasa Jawa krama.

2.5. Mengevaluasi hasil terjemahan menggunakan skor BLEU

Hasil terjemahan yang dihasilkan NMT akan dibandingkan kecocokannya dengan data uji berbahasa Jawa krama, yang mana merupakan terjemahan sebenarnya dari data uji berbahasa Jawa ngoko. Evaluasi ini akan menggunakan pengukuran skor BLEU, dari 0 sampai 100. Taraf penilaian dari performa penerjemahan menggunakan OpenNMT dan THUMT didasarkan oleh tinggi atau rendahnya skor yang dihasilkan. Jika skor BLEU tinggi atau mendekati 100, maka kualitas terjemahan yang dihasilkan lebih baik dari pada skor BLEU yang lebih rendah.

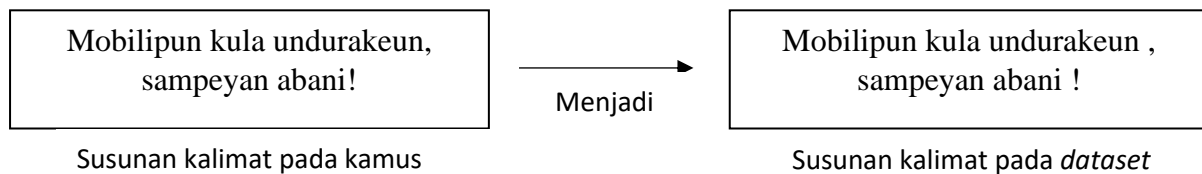
2.6. Membuat aplikasi penerjemahan menggunakan model NMT

Model NMT yang telah dilatih, baik dari OpenNMT atau THUMT akan dipilih salah satu untuk dibangun sebuah aplikasi penerjemahan bahasa Jawa ngoko – krama. Aplikasi tersebut akan dibuat menggunakan bahasa pemrograman Python. Selain itu, aplikasi yang dibuat berbasis situs web dan dapat diakses melalui beragam perangkat, seperti telepon seluler maupun komputer.

3. HASIL DAN PEMBAHASAN

3.1. Pembuatan *dataset*

Dataset bahasa Jawa ngoko – krama dibuat dalam berkas berformat teks (.txt). Kalimat dari *dataset* tersebut diambil dari kalimat yang terdapat pada Kamus Unggah-Ungguh Basa Jawa yang ditulis oleh Harjawiyana (2001). Tiap barisnya disusun oleh kalimat berbeda dan kata yang menempel dengan tanda baca akan dipisah agar mengurangi kata yang ambigu saat membangun kosa kata. Susunan kalimat tersebut dapat dilihat pada **Gambar 2**.



Gambar 2. Susunan kalimat pada kamus dan *dataset*.

Terdapat 4 berkas untuk *dataset* bahasa Jawa ngoko, yaitu ‘ngoko-train.txt’ sebagai data latih, ‘ngoko-val.txt’ sebagai data validasi, ‘ngoko-test1.txt’ sebagai data uji dengan kalimat terdapat di data latih, dan ‘ngoko-test2.txt’ sebagai data uji dengan kalimat tidak terdapat di data latih. Sama halnya dengan Jawa ngoko, berkas untuk *dataset* bahasa Jawa krama juga ada 4, yaitu ‘krama-train.txt’ sebagai data latih, ‘krama-val.txt’ sebagai data validasi, ‘krama-test1.txt’ sebagai data uji dengan kalimat terdapat di data latih, dan ‘krama-test2.txt’ sebagai data uji dengan kalimat tidak terdapat di data latih. Untuk skema pembagian *dataset* dapat terlihat pada **Tabel 1**.

Tabel 1. Skema pembagian kalimat *dataset* bahasa Jawa ngoko – krama

	Total kalimat	
	Jawa ngoko (Sumber)	Jawa krama (Target)
Data Latih	1000	1000
Data Validasi	500	500
Data Uji 1 (Kalimat terdapat di data latih)	100	100
Data Uji 2 (Kalimat tidak terdapat di data latih)	50	50

3.2. Pembuatan *vocabularies* atau kosakata

Membuat *vocabularies* atau kosakata memiliki langkah yang berbeda untuk masing-masing sistem OpenNMT maupun THUMT. Sistem OpenNMT membutuhkan berkas konfigurasi yang digunakan untuk membangun kosakata serta membuat model NMT. Berkas ini diberi nama ‘config.yaml’. Baris kode di dalam berkas ‘config.yaml’ tertera di **Kode 1**.

Kode 1. Baris kode di dalam berkas ‘config.yaml’

```
1. ## lokasi sampel ditulis
2. save_data: data/run/example
3. ## tempat simpan berkas vocab
4. src_vocab: data/run/example.vocab.src
5. tgt_vocab: data/run/example.vocab.tgt
6. overwrite: False
7. # pembuatan korpus
8. data:
9.     corpus_1:
10.         path_src: data/ngoko-train.txt
11.         path_tgt: data/krama-train.txt
12.     valid:
13.         path_src: data/ngoko-val.txt
14.         path_tgt: data/krama-val.txt
```



```
15. # tempat simpan model dan langkah pelatihan
16. save_model: data/run/model
17. save_checkpoint_steps: 500
18. valid_steps: 500
19. train_steps: 2000
20.
21. # Batching
22. bucket_size: 64
23. world_size: 1
24. gpu_ranks: [0]
25. num_workers: 2
26. batch_type: "tokens"
27. batch_size: 128
28. valid_batch_size: 64
29. accum_count: [4]
30. accum_steps: [0]
31.
32. # Optimisasi
33. model_dtype: "fp16"
34. optim: "adam"
35. learning_rate: 2
36. warmup_steps: 8000
37. decay_method: "noam"
38. adam_beta2: 0.998
39. max_grad_norm: 0
40. label_smoothing: 0.1
41. param_init: 0
42. param_init_glorot: true
43. normalization: "tokens"
44.
45. # Pembuatan model dengan tipe Transformer
46. encoder_type: transformer
47. decoder_type: transformer
48. position_encoding: true
49. enc_layers: 6
50. dec_layers: 6
51. heads: 8
52. hidden_size: 512
53. word_vec_size: 512
54. transformer_ff: 2048
55. dropout_steps: [0]
56. dropout: [0.1]
57. attention_dropout: [0.1]
```

Pembuatan kosakata di atas memanfaatkan berkas ‘ngoko-train.txt’ dan ‘krama-train.txt’ sebagai korpus dan ‘ngoko-val.txt’ dan ‘krama-val.txt’ sebagai validasi korpus. Dari proses pembuatan kosakata ini akan menghasilkan berkas kosakata untuk Jawa ngoko bernama ‘example.vocab.src’ dan berkas kosakata untuk Jawa krama bernama ‘example.vocab.tgt’. Kedua berkas ini selanjutnya digunakan untuk membuat model NMT OpenNMT.

THUMT tidak memerlukan berkas konfigurasi khusus untuk membuat kosakata layaknya di OpenNMT. Untuk membuat kosakata di THUMT, baris kode yang dijalankan tertera di **Kode 2**.

Kode 2. Baris kode untuk membuat kosakata di THUMT

```
build_vocab.py data/ngoko-train.txt data/ngoko.vocab
build_vocab.py data/krama-train.txt data/krama.vocab
```

Dari proses tersebut, menghasilkan berkas ‘ngoko.vocab.txt’ dan ‘krama.vocab.txt’ yang mana kedua berkas tersebut digunakan untuk membuat model NMT THUMT.

3.3. Pembuatan model NMT

Model NMT yang dibuat di OpenNMT kembali membutuhkan berkas ‘config.yaml’ untuk melalui proses pelatihan data. Sementara itu di THUMT membutuhkan enam berkas yang diantaranya adalah dua berkas data latih, dua berkas kosakata, dan dua berkas data validasi. Baris kode untuk membuat model NMT di kedua sistem tersebut dapat dilihat di **Kode 3** dan **Kode 4**.

Kode 3. Baris kode untuk membuat model NMT di OpenNMT

```
onmt_train -config data/config.yaml
```

Kode 4. Baris kode untuk membuat model NMT di THUMT

```
thumt-trainer \  
  --input data/ngoko-train.txt data/krama-train.txt \  
  --vocabulary data/ngoko.vocab.txt data/krama.vocab.txt \  
  --model transformer \  
  --validation data/ngoko-val.txt \  
  --references data/krama-val.txt \  
  --parameters=batch_size=128,device_list=[0],update_cycle=10 \  
  --hparam_set base
```

Model NMT yang dihasilkan sistem OpenNMT adalah berkas bernama ‘model_step_2000.pt’, yang berarti model NMT yang dihasilkan dengan 2000 langkah pelatihan. Sedangkan sistem THUMT menghasilkan berkas model NMT bernama ‘model-2.pt’ yang berarti model NMT dengan langkah pelatihan pada *checkpoint* ke-2 (2000 langkah).

3.4. Penerjemahan data uji

Model NMT selanjutnya digunakan untuk membuat sistem penerjemahan. Pada tahap ini, semua kalimat dari data latih yaitu ‘ngoko-test1.txt’ dan ‘ngoko-test2.txt’ diterjemahkan dan hasil terjemahan tersebut dimasukkan ke dalam berkas baru bernama ‘pred.txt’. Baris kode untuk menerjemahkan data uji di OpenNMT dan THUMT tertera di **Kode 5** dan **Kode 6**.

Kode 5. Baris kode untuk menerjemahkan data uji di OpenNMT

```
onmt_translate -model data/run/model_step_2000.pt -src data/ngoko-  
test.txt -output data/run/pred.txt -gpu 0 -verbose
```

Kode 6. Baris kode untuk menerjemahkan data uji di THUMT

```
thumt-translator \  
  --models transformer \  
  --input data/ngoko-test.txt \  
  --output data/pred.txt \  
  --vocabulary data/ngoko.vocab.txt data/krama.vocab.txt \  
  --checkpoints train/eval \  
  --parameters=device_list=[0],decode_alpha=1.2
```

3.5. Hasil skor BLEU

Penggunaan skor BLEU untuk evaluasi dapat digunakan dengan memanfaatkan pustaka SacreBLEU (Post, 2018). Berkas ‘pred.txt’ akan dibandingkan kesamaannya dengan hasil terjemahan data uji yang benar yaitu ‘krama-test.txt’. Baris kode yang digunakan untuk mengevaluasi hasil terjemahan di OpenNMT dan THUMT dapat dilihat di **Kode 7**.

Kode 7. Baris kode untuk mengevaluasi hasil terjemahan OpenNMT dan THUMT

```
import sacrebleu
# Baca isi file prediksi
with open('data/pred.txt', 'r', encoding='utf-8') as f:
    pred = [line.strip() for line in f]
# Baca isi file data uji
with open('data/krama-test.txt', 'r', encoding='utf-8') as f:
    tgt = [line.strip() for line in f]
bleu_score = sacrebleu.corpus_bleu(pred, [tgt])
print("Skor BLEU:", bleu_score.score)
```

Hasilnya, skor BLEU untuk masing-masing sistem tercantum di **Tabel 2** yang tertera sebagai berikut:

Tabel 2. Hasil skor BLEU

	Skor BLEU	
	Data Uji 1	Data Uji 2
OpenNMT	98,55	2,01
THUMT	99,47	1,6

Sistem NMT mampu menerjemahkan dengan tepat kalimat yang sama dengan yang terdapat di data latih atau kalimat yang sudah tersimpan di model NMT. Hal ini dibuktikan dengan skor yang didapatkan OpenNMT dan THUMT sama-sama mendekati 100. Salah satu perbedaan hasil terjemahan antara OpenNMT dan THUMT dari data uji dengan kalimat terdapat di data latih tertera di **Tabel 3**.

Tabel 3. Perbandingan terjemahan OpenNMT dan THUMT dari kalimat Data Uji 1

Kalimat	Bapak lagi wae teka andang arep lunga maneh !
Terjemahan	Bapak saweg kemawon dhateng andang (enggal-enggal) badhe kesah malih !
Terjemahan OpenNMT	Bapak saweg kemawon dhateng enggal-enggal badhe kesah malih !
Terjemahan THUMT	Bapak saweg kemawon dhateng andang (enggal-enggal) badhe kesah malih !

Dari tabel tersebut, terlihat bahwa THUMT menerjemahkan sama persis dengan terjemahan yang seharusnya. Sementara OpenNMT fokus menghasilkan terjemahan kata yang terdapat di dalam tanda kurung yaitu ‘enggal-enggal’ dan menghilangkan kata sebelumnya di luar tanda kurung yaitu ‘andang’.

Namun, untuk Data Uji 2 yang memiliki kalimat berbeda dengan yang terdapat di data latih mendapatkan skor BLEU yang sangat rendah untuk kedua sistem NMT. Hal ini dapat terjadi dikarenakan sejumlah kata dan struktur kalimat yang belum dimengerti oleh model NMT, serta jumlah kalimat di data latih yang masih terlalu sedikit. Untuk perbandingan terjemahan dari data uji 2 dapat dilihat di **Tabel 4**.

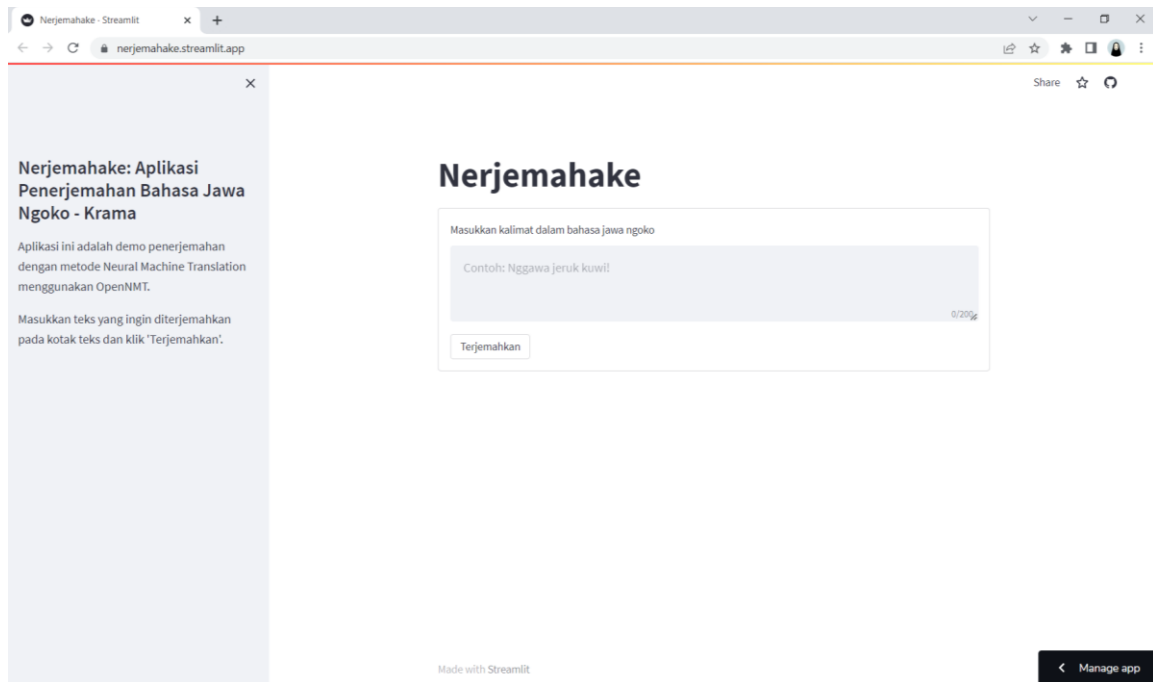
Tabel 4. Perbandingan terjemahan OpenNMT dan THUMT dari kalimat Data Uji 2

Kalimat	Aku duwe utang marang Paman sapuluh ewu rupiyah .
Terjemahan	Kula gadhah sambutan dhateng Paman sadasa ewu rupiyah .
Terjemahan OpenNMT	Kula gadhah atur dhateng Pak Camat bab Karang Taruna .
Terjemahan THUMT	Kula gadhah pelem , menapa sampeyan ajeng .

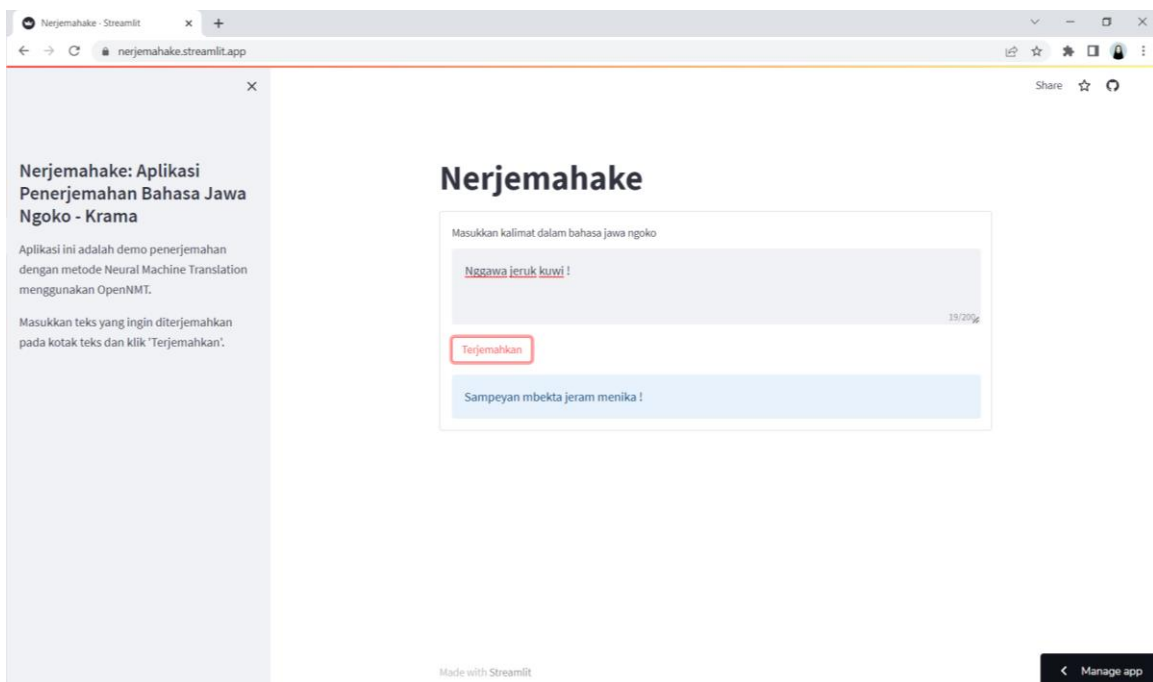
Terlihat pada tabel tersebut bahwa OpenNMT mampu menerjemahkan beberapa kata yang telah tersedia di model NMT, yaitu ‘Aku duwe’ menjadi ‘Kula gadhah’, serta ‘marang’ menjadi ‘dhateng’. Sedangkan THUMT hanya mampu mengidentifikasi kata yang berada di depan kalimat yaitu ‘Aku duwe’ menjadi ‘Kula gadhah’ dan seterusnya hanya mengikuti kalimat yang telah tersedia di model NMT.

3.6. Aplikasi penerjemahan

Aplikasi penerjemahan ini menggunakan model yang dihasilkan sistem OpenNMT. Untuk membuat antar muka aplikasi, bahasa pemrograman yang dipakai adalah bahasa Python serta dibutuhkan pustaka Python yaitu Streamlit. Streamlit dapat digunakan untuk membuat aplikasi web interaktif menggunakan kode sederhana (Parker dkk., 2021). Aplikasi ini kemudian diberi nama “Nerjemahake” dan tangkapan layar dari tampilan aplikasi tersebut dapat dilihat pada **Gambar 3** dan **Gambar 4**.



Gambar 3. Tampilan awal aplikasi



Gambar 4. Tampilan aplikasi setelah menerjemahkan kalimat

4. PENUTUP

Kesimpulannya, penelitian ini telah menghasilkan *dataset* penerjemahan bahasa Jawa ngoko - krama sebanyak 8 berkas, yang terdiri dari total 2000 kalimat data latih, 1000 kalimat data validasi, 200 kalimat data uji 1, dan 100 kalimat data uji 2. Kemudian, penelitian ini juga

menghasilkan skor BLEU dari pengukuran hasil terjemahan OpenNMT sebesar 98,55 untuk data uji 1 dan 2,01 untuk data uji 2, serta untuk hasil terjemahan THUMT sebesar 99,47 untuk data uji 1 dan 1,6 untuk data uji 2. Terakhir, penelitian ini juga menghasilkan aplikasi penerjemahan bahasa Jawa ngoko – krama bernama “Nerjemahake” yang dibuat menggunakan bahasa pemrograman Python dan pustaka Python yaitu Streamlit.

Performa terjemahan yang dihasilkan masih belum sempurna dan perlu adanya perbaikan. Saran dari peneliti, kondisi tersebut bisa diperbaiki dengan menambahkan lebih banyak kalimat di data latih dan data validasi agar kosakata yang dibuat menjadi lebih kaya dan semakin mudah sistem NMT untuk mempelajari kosakata serta struktur kalimatnya. Dalam hal ini, peneliti senantiasa tetap berusaha untuk mempelajari, mengevaluasi, serta memperbaiki performa terjemahan bahasa Jawa ngoko – krama ini agar menjadi aplikasi penerjemahan yang dapat digunakan masyarakat di kehidupan sehari-hari.

DAFTAR PUSTAKA

- Abidin, Z. (2017). Penerapan Neural Machine Translation untuk Eksperimen Penerjemahan secara Otomatis pada Bahasa Lampung – Indonesia. *Prosiding Seminar Nasional Metode Kuantitatif*, 0(1).
- Fardani, M. A., & Wiranti, D. A. (2019). Peran Orang Tua dalam Pengembangan Bahasa Jawa Krama Anak Usia Dini. *Jurnal Penelitian Pendidikan Anak Usia Dini*, 13(1), 21-32.
- Harjawiya, H. (2001). *Kamus Unggah-Ungguh Basa Jawa*. Yogyakarta: Kanisius.
- Haryanti, Y., & Kaltsum, H. U. (2019). Internalisasi Nilai-Nilai Kearifan Lokal pada Pelajaran Bahasa Jawa di Sekolah Dasar. *Jurnal Pendidikan: Teori, Penelitian, dan Pengembangan*, 6(1).
- Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). *OpenNMT: Open-Source Toolkit for Neural Machine Translation*.
- Parker A., Heflin A., Jones L. C. (2021). Analyzing University of Virginia Health publications using open data, Python, and Streamlit. *J Med Libr Assoc.* 2021 Oct 1;109(4):688-689. doi: 10.5195/jmla.2021.1360. PMID: 34858105; PMCID: PMC8608219.
- Post, M. (2018). A Call for Clarity in Reporting BLEU Scores. *Proceedings of the Third Conference on Machine Translation: Research Papers* (pp. 186-191). Association for Computational Linguistics.
- Purwadi, P. (2020). Etika Komunikasi dalam Budaya Jawa. *Jurnal Ilmu Komunikasi*, 9(3), 239–249.

- Sukriayu, R. E., Kridalaksana, A. H., & Hatta, H. R. (2017). Aplikasi Kamus Bahasa Paser-Indonesia Berbasis Android. *Prosiding Seminar Nasional Ilmu Komputer dan Teknologi Informasi*, 2(2).
- Thalib, I. (2019). NLP Preprocessing: Teknik Tokenisasi untuk Memecah Kalimat Menjadi Kata-Kata pada Python. *Medium*. <https://medium.com/@irfandy.thalib/teknik-tokenisasi-untuk-memecah-kalimat-menjadi-kata-kata-pada-python-12f799b74d49>
- Thamrin, H., Ariyanto, G., Pamungkas, E. W., & Sulistyono, Y. (2017). User Participation in Building Language Repository: The Case of Google Translate. *IOP Conference Series: Materials Science and Engineering*, 403(1), 012072. doi: 10.1088/1757-899X/403/1/012072
- Wu, Y., Schuster, M., Chen, Z., Le, Q. v., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., ... Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.
- Zhang, J., Ding, Y., Shen, S., Cheng, Y., Sun, M., Luan, H., & Liu, Y. (2017). THUMT: An Open Source Toolkit for Neural Machine Translation.