

MESIN PENERJEMAH BAHASA PALEMBANG KE BAHASA INDONESIA MENGUNAKAN ALGORITMA ENCODER-DECODER

¹⁾Laode Hengky Pandrawan, ²⁾Yesi novaria Kunang, ³⁾Ilman Zuhri Yadi, ⁴⁾Deni Erlansyah,
⁵⁾Rahayu Amalia

Sistem Informasi, Sains Dan Teknologi, Universitas Bina Darma
Jl. Jenderal Ahmad Yani No.3, 9/10 Ulu, Kecamatan Seberang Ulu I, Kota Palembang, Sumatera Selatan-
Indonesia

E-mail : laodehengky2000@gmail.com, yesinovaria@binadarma.ac.id
ilmanzuhriyadi@binadarma.ac.id

ABSTRAK

Bahasa Palembang adalah bahasa yang populer di Sumatera Selatan, Ini memungkinkan untuk dikembangkan menjadi mesin penerjemah untuk memudahkan komunikasi antara penutur bahasa tersebut dengan penutur bahasa Indonesia. Hal ini penting karena bahasa Palembang merupakan salah satu kekayaan budaya Indonesia yang perlu dilestarikan. Teknologi berkembang pesat dan berkelanjutan seiring perkembangan zaman. Teknologi dibutuhkan oleh semua orang untuk mempercepat kemajuan dan pembangunan. Salah satu contoh teknologi yang dapat digunakan adalah kecerdasan buatan (AI). Machine learning adalah cabang dari AI yang dapat dimanfaatkan untuk membuat mesin penerjemah. Untuk menerjemahkan bahasa Palembang ke bahasa Indonesia, peneliti menggunakan algoritma encoder-decoder. Algoritma ini bekerja dengan cara mempelajari dokumen input dan menghasilkan terjemahan untuk tiap kata bahasa Palembang. Kemudian, hasil terjemahan tersebut digunakan untuk menghasilkan terjemahan ke bahasa Indonesia.

Kata Kunci: Bahasa Palembang, Algoritma Encoder-Decoder, Mesin Penerjemah.

ABSTRACT

Palembang language is a popular language in South Sumatra. It allows for the development of a translation machine to facilitate communication between speakers of that language and Indonesian speakers. This is important because the Palembang language is one of Indonesia's cultural treasures that needs to be preserved. Technology is developing rapidly and continuously along with the progress of time. Technology is needed by everyone to accelerate advancement and development. One example of technology that can be used is artificial intelligence (AI). Machine learning is a branch of AI that can be utilized to create a translation machine. To translate Palembang language into Indonesian, researchers use an encoder-decoder algorithm. This algorithm works by studying input documents and generating translations for each Palembang language word. Then, the translation results are used to generate translations into Indonesian.

Keyword: Palembang Language, Encoder-Decoder Algorithm, Translation Machine.

PENDAHULUAN

Bahasa adalah ciri khas suatu bangsa yang memungkinkan pengidentifikasian kelompok, komunitas dan identitas individu. Dalam konteks identitas, bahasa menjadi esensi dari jati diri suatu bangsa, mencerminkan eksistensinya dalam perbandingan dengan bangsa lain. Sebagai bagian dari warisan budaya tak benda, bahasa memiliki nilai penting yang membutuhkan usaha pelestarian. Jika bahasa dianggap sebagai warisan budaya, upaya penyelamatan dan pemeliharaannya dapat diartikan sebagai upaya pelestarian kebudayaan secara keseluruhan. (Afria dkk.,

2020)

Keragaman bahasa di setiap daerahnya, seperti bahasa Palembang di Sumatera Selatan. Palembang yang merupakan ibukota provinsi Sumatera Selatan di bagian selatan Sumatera memiliki bahasa sehari-hari yang berbeda. Untuk menjaga keberlanjutan bahasa Palembang masyarakat Sumatera Selatan menggunakan bahasa tersebut sebagai bagian dari komunikasi sehari-hari, selain bahasa Indonesia. (Andri, 2019)

Pemeliharaan bahasa daerah seperti bahasa Palembang memiliki budaya dan sejarah yang sangat penting. Mesin penerjemah dapat membantu dalam memelihara dan melestarikan

bahasa daerah ini dengan memudahkan akses dan pemahaman terhadap dokumen-dokumen dalam bahasa Palembang. Pengembangan mesin penerjemah bahasa Palembang ke bahasa Indonesia juga dapat mendukung prinsip multikulturalisme dengan memungkinkan dialog yang lebih baik antara berbagai kelompok budaya dan etnis.

Mesin penerjemah bahasa Palembang ke bahasa Indonesia dan sebaliknya dapat menjadi jembatan komunikasi antarbudaya dan dapat memfasilitasi pertukaran informasi dan ide antara masyarakat yang berbicara dalam kedua bahasa tersebut.

Pemanfaatan perangkat berbasis komputer dapat menjadi langkah untuk melestarikan bahasa Palembang secara digital sesuai dengan kemajuan teknologi saat ini. Salah satu caranya adalah melalui pembangunan kamus digital dan mesin penerjemah. Sebagai contoh, mesin penerjemah Google, yang dapat diakses melalui situs <https://translate.google.com>, merupakan salah satu alat yang bisa dimanfaatkan. Mesin ini mudah diakses, namun disayangkan, hanya menyediakan dukungan untuk dua bahasa daerah, yaitu bahasa Jawa dan bahasa Sunda, sedangkan untuk bahasa Palembang tidak tersedia pada mesin penerjemah Google (Abidin, 2018)

Para pengguna Google Translate paling aktif. Lebih lanjut, Macduff menambahkan bahwa pengguna Google Translate di Indonesia mengalami pertumbuhan yang sangat signifikan secara tahunan, mencapai 50% pada perangkat Android, 94% pada desktop komputer, dan bahkan mencapai 354% pada penerjemahan melalui perangkat mobile. (Amalia Yunia Rahmawati, 2020)

Mesin penerjemah merupakan perangkat lunak penerjemah bahasa yang proses penerjemahannya dilakukan oleh mesin (komputer) secara otomatis dan hasilnya dapat

langsung disajikan kepada pengguna. Opsi penggunaan mesin penerjemah lebih umum dipilih karena cepat dan praktis dalam melakukan penerjemahan teks, berbanding terbalik dengan penerjemahan oleh manusia yang memerlukan waktu lama. Meskipun penerjemah manusia yang berkompeten akan menghasilkan penerjemahan yang lebih baik dan dalam beberapa keadaan tak tergantikan oleh penerjemah otomatis, namun ada banyak kasus di mana cukup untuk diterjemahkan menggunakan mesin penerjemah. Selain itu, hasil penerjemahan oleh mesin pun tidak serta merta digunakan begitu saja, namun disunting kembali oleh manusia. Sehingga penerjemah manusia berkualifikasi tinggi pun juga menggunakan mesin penerjemah untuk mempercepat proses penerjemahan (Wismoyo, 2018)

Kelebihan menggunakan Model *encoder decoder* adalah untuk memperincis teks menggunakan *pointer-generator* dan *coverage* untuk mengatasi dua masalah utama: ketergantungan terhadap kosakata dan repetisi kata. *Pointer-generator* memungkinkan model untuk meniru kata dari dokumen masukan, sehingga mengurangi ketergantungan terhadap kosakata. *Coverage* membantu mengurangi repetisi kata dengan melacak kata-kata yang telah digunakan dalam ringkasan (AKBAR, 2021)

Kekurangan dari algoritma *encoder decoder*, terletak pada peningkatan kemampuan *encoder* dan model secara keseluruhan untuk memahami konteks dokumen input harus menerapkan penggunaan *pointer-generator* dan *coverage*. *Pointer-generator* mengurangi ketergantungan model pada kosakata dengan memungkinkan model untuk meniru kata-kata langsung dari dokumen input (AKBAR, 2021)

Oleh karena itu, untuk menyelesaikan

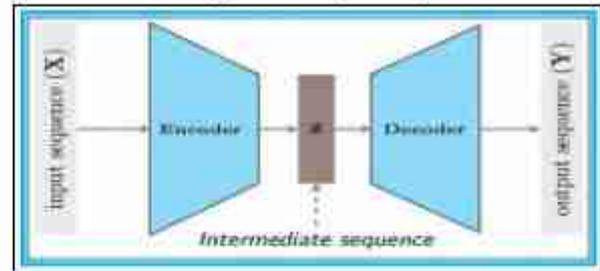
permasalahan yang ada maka diperlukan algoritma *encoder decoder*, merupakan algoritma yang dapat membantu pengembangan mesin penerjemah bahasa daerah. Pada bahasa daerah Palembang, pengembangan mesin penerjemah dengan memanfaatkan teknologi informasi belum terlaksana dengan baik. Untuk itu diperlukan teknologi penerjemahan otomatis dari Palembang ke Bahasa Indonesia agar bahasa Palembang semakin dikenal dan tidak ketinggalan dengan budaya bahasa asing. Melihat penjelasan masalah, Peneliti melakukan penelitian dengan judul “Mesin Penerjemah Bahasa Palembang Ke Bahasa Indonesia Menggunakan Algoritma *Encoder-Decoder*”.

Algoritma *Encoder-Decoder*

Encoder-Decoder adalah salah satu arsitektur dalam *Machine Learning* yang umumnya digunakan untuk pembelajaran *unsupervised* yaitu data yang tidak memiliki label. Bagian ini akan dibagi menjadi dua bagian utama. Bagian pertama adalah *encoder*, bagian ini berfungsi untuk mengolah data masukan menjadi sebuah *state* atau vektor representasi yang mewakili informasi-informasi dari data masukan tersebut. Bagian kedua adalah *decoder*, bagian ini bertugas untuk mengambil *state* atau vektor representasi yang dihasilkan *encoder*, lalu mengolahnya menjadi keluaran yang diinginkan (AKBAR, 2021).

Pada bagian *encoder* dan *decoder* dapat terdiri dari beberapa *sub-layer*. Tergantung kebutuhan layer *neural machine translation* yang biasa digunakan. Arsitektur *encoder decoder* telah banyak dimanfaatkan untuk beberapa tugas, diantaranya adalah untuk kompresi gambar, denoising gambar, deteksi anomali, dimensionality reduction, machine

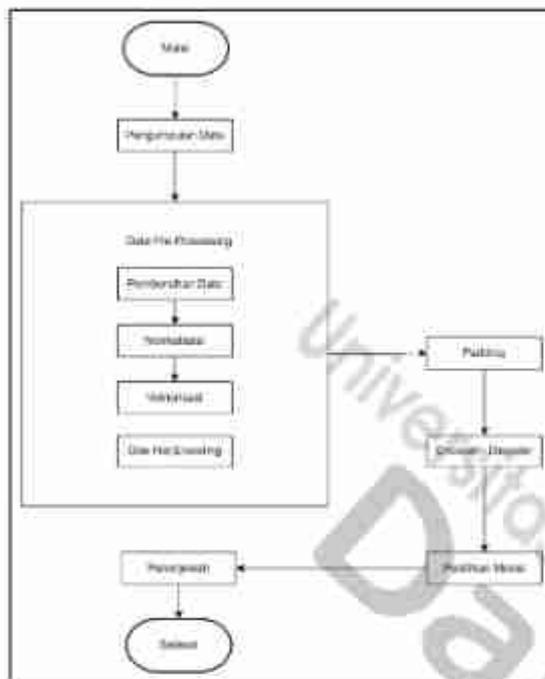
translation hingga automatic text summarization. (AKBAR, 2021)



Gambar 2. 1 Arsitektur *Encoder-Decoder*

Selanjutnya kelebihan dari Algoritma *encoder-decoder* dapat menghasilkan terjemahan yang lebih akurat karena sistem memproses seluruh input sebelum menerjemahkan. Hal ini memungkinkan *decoder* untuk menggunakan informasi dari seluruh kalimat sumber saat menghasilkan kata berikutnya dalam terjemahan.

Dalam pengembangan Mesin Penerjemah Bahasa Palembang Ke Bahasa Indonesia menggunakan Algoritma *Encoder-Decoder*. Proses dimulai dengan pengumpulan data dan pemrosesan teks dalam Bahasa Palembang untuk memastikan konsistensinya. Pada algoritma ini terdapat dua buah tahapan yaitu *Encoder* dan *Decoder*, dimana *Encoder* merupakan sebuah lapisan proses yang akan menjadi masuknya bahasa sumber dan *Decoder* merupakan lapisan proses hasil terjemahan dari lapisan *Encoder* yang diubah menjadi sebuah bahasa target atau bahasa terjemah. Lapisan *Encoder-Decoder* terdiri dari sebuah jaringan proses pembelajaran dengan metode *Recurrent Neural Network*(RNN). Akibatnya, dasar-dasar jaringan saraf akan dibahas pertama kali dalam bagian latar belakang ini. Hal ini juga akan membahas perbedaan utama antara jaringan saraf berulang dan jaringan saraf normal. Dari sini akan melihat teknik pemrosesan data One Hot Encoding di bagian ini sebelum mempelajari struktur pengkodean. (Lanners & Laurent, 2019)



Gambar 3. 1 Tahapan Penelitian

a. Pengumpulan data

Tahap awal penelitian ini melibatkan pengumpulan data, yaitu dokumen mesin terjemah dan bahasa Palembang yang telah diterjemahkan ke dalam bahasa Indonesia.

b. Pembersihan data

Pembersihan data adalah langkah krusial dalam persiapan dataset untuk model penerjemah bahasa. Dengan memastikan data bersih, terstruktur, dan relevan, dapat diharapkan bahwa model penerjemah yang dikembangkan akan memiliki kualitas yang baik dalam menghasilkan terjemahan antara bahasa Palembang dan bahasa Indonesia menggunakan algoritma *encoder-decoder*.

c. Normalisasi

Prosedur untuk menstandarisasi data teks, seperti menghapus tanda baca dan mengubah semua huruf menjadi huruf kecil, atau mengurangi bentuk kata ke bentuk aslinya untuk menjaga konsistensi dan meningkatkan akurasi model penerjemah.

d. Vektorisasi

Vektorisasi adalah langkah awal yang penting dalam mempersiapkan data untuk pengembangan model penerjemah bahasa menggunakan algoritma *encoder-decoder*. Dengan menerapkan proses ini dengan hati-hati, diharapkan bahwa model penerjemah akan mampu belajar dengan baik dari data yang diberikan dan menghasilkan penerjemahan yang akurat antara bahasa Palembang dan bahasa Indonesia.

e. *One Hot Encoding*

Teknik yang mengubah kata-kata menjadi vektor biner. Setiap kata diwakili oleh vektor yang panjangnya sama dengan ukuran kosakata, dengan satu elemen bernilai 1 (menunjukkan posisi kata tersebut) dan sisanya bernilai 0. Teknik ini memungkinkan model memproses teks secara numerik.

f. Padding

Padding yaitu metode menambahkan elemen khusus, biasanya nol, ke urutan data untuk memastikan bahwa semua input dan output memiliki panjang yang sama. Teknik ini sangat penting untuk memastikan bahwa model dapat memproses batch data secara efisien dan konsisten, meskipun panjang kalimat dapat berbeda.

g. *Encoder - Decoder*

Tahapan ini, *decoder* memberikan informasi dari setiap keadaan tersembunyi *encoder*. Informasi ini memungkinkan model untuk memusatkan perhatian pada elemen-elemen penting dalam urutan input dan mempelajari hubungan di antara elemen-elemen tersebut. Mekanisme perhatian dapat menggabungkan informasi dari setiap langkah waktu *encoder* menjadi representasi vektor tunggal. Representasi vektor ini membantu model untuk menangani kalimat input yang panjang dan meningkatkan akurasi terjemahan.

h. Pelatihan Model

Pelatihan model adalah proses di mana

model penerjemah dilatih untuk menerjemahkan dengan benar dengan menggunakan pasangan kalimat dalam Bahasa Palembang dan Bahasa Indonesia. Proses ini memungkinkan model untuk memahami pola dan hubungan antara kedua bahasa tersebut, sehingga mereka dapat melakukan penerjemahan yang akurat. Dalam proses ini, parameter model dioptimalkan untuk mengurangi kesalahan terjemahan.

i. Evaluasi Model

Proses mengevaluasi kinerja model penerjemah dengan data uji yang tidak digunakan selama pelatihan. Ini mencakup menilai kemampuan model untuk menerjemahkan teks dari Palembang ke Bahasa Indonesia dengan menggunakan metrik ketepatan akurasi.

j. Penerjemah

Penerjemah adalah sistem atau perangkat lunak yang secara otomatis dan akurat mengonversi teks dari Bahasa Palembang ke Bahasa Indonesia dengan menggunakan algoritma encoder-decoder.

METODE

Jelaskan metode yang digunakan, metode penelitian maupun metode pengolahan data, jika terdapat gambar bagan atau sejenisnya gunakan gambar yang dapat di baca dan jelas. Misalnya :

a. Pengumpulan Data

Pada penelitian yang dilakukan, data yang dikumpulkan dan akan digunakan bersumber dari data hasil scan dari KAMUS SEDERHANA BAHASA PALEMBANG, dan KAMUS PALEMBANG-INDONESIA. Berdasarkan hasil Scan yang dilakukan, data yang didapatkan berjumlah 6343 data yang terdiri dari 5062 kata dan 1.281 kalimat. 6343 data yang dikumpulkan tersebut disimpan di dalam file excel.



Gambar 4. 1 Pengumpulan Data

b. Pembuatan Dataset

Data dikumpulkan dengan titik di ujung kata atau kalimat dan kutip di akhir kata atau kalimat. Kemudian data dipecah dari file Excel sebelumnya dan disimpan dalam format file .txt. Tujuan pemecahan ini adalah untuk mempermudah pemrosesan data dalam bahasa Indonesia dan Palembang untuk penerjemahan yang akan dilakukan nantinya.

c. Preprocessing

Pada tahap ini, dataset yang telah dibentuk akan mengalami pra-pemrosesan terlebih dahulu. Pra-pemrosesan ini mencakup empat tahap yaitu Pembersihan data, normalisasi, vektorisasi, dan *one hot encoding*.

d. Pembersihan Data

Tahapan pembersihan ini mencakup beberapa langkah penting, yang meliputi namun tidak terbatas pada, penghapusan atau perbaikan entri yang tidak lengkap atau rusak, penghilangan noise seperti simbol atau karakter yang tidak relevan, penyesuaian format teks seperti kapitalisasi atau pemisahan kata yang tepat, serta pengelolaan kata-kata yang mungkin memiliki makna ganda atau yang tidak memiliki padanan yang jelas antara kedua bahasa.

Sebagai contoh, dalam pembersihan data untuk penerjemah bahasa Palembang ke bahasa Indonesia, penting untuk memastikan bahwa setiap kalimat atau frasa dalam bahasa Palembang memiliki terjemahan yang akurat dan tepat dalam bahasa Indonesia, tanpa adanya

ketidakkonsistenan atau ambiguitas. Selain itu, proses pembersihan ini juga bisa melibatkan normalisasi data, seperti penyeragaman ejaan kata-kata tertentu atau mengganti kata-kata slang dengan istilah yang lebih umum dan mudah dipahami dalam bahasa target. Semua langkah ini dilakukan untuk meningkatkan kualitas data pelatihan sehingga model *encoder-decoder* yang digunakan dapat mempelajari pola penerjemahan dengan lebih baik dan menghasilkan terjemahan yang lebih akurat.

e. Normalisasi

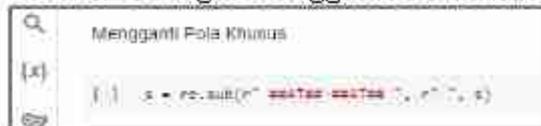
Proses ini mencakup langkah-langkah untuk menyederhanakan teks menjadi bentuk standar. Normalisasi melibatkan tindakan seperti mengubah semua huruf menjadi huruf kecil dan menghapus spasi yang berlebihan. Berikut adalah penjelasan langkah demi langkah untuk setiap bagian dalam fungsi koding yang ditunjukkan pada gambar di bawah ini :



Gambar 4.3. 1 Normalisasi

Penjelasan langkah demi langkah pada kodingan tersebut :

1. Kode Codingan Mengganti Pola Khusus

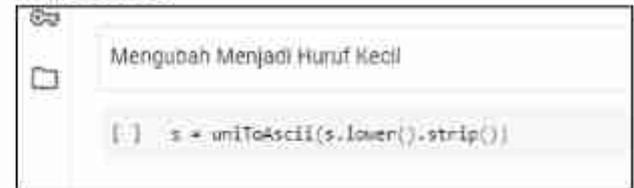


Gambar 4.3. 2 Mengganti Pola Khusus

Langkah ini menggantikan pola teks spesifik "##AT## ##AT## " dengan satu spasi. Ini

berguna jika teks masukan memiliki pola ini dan ingin menggantinya dengan spasi.

2. Kode Codingan Mengubah Huruf menjadi Huruf Kecil

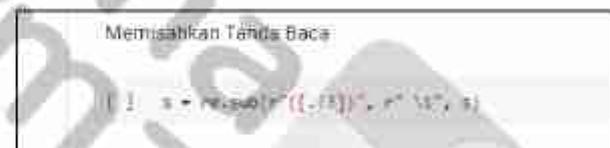


Gambar 4.3. 3 Mengubah Huruf Menjadi Huruf Kecil

s.lower() mengubah semua huruf menjadi huruf kecil.

s.strip() menghapus spasi kosong di awal dan akhir string.

3. Kode Codingan Memisahkan Tanda Baca dari Kata



Gambar 4.3. 4 Memisahkan Tanda Baca

Langkah ini menambahkan satu spasi sebelum setiap tanda baca seperti titik (.), tanda seru (!), dan tanda tanya (?). Misalnya, "Hello!" akan menjadi "Hello".

4. Kode Codingan Menghapus Karakter Non-Huruf



Gambar 4. 2. 5 Menghapus Karakter Non-Huruf

z, A-Z), titik (.), tanda seru (!), atau tanda tanya (?) dengan spasi. Karakter non huruf seperti angka, tanda baca lain, dan karakter khusus lainnya akan dihapus.

f. Vektorisasi

Vektorisasi adalah langkah awal yang penting dalam mempersiapkan data untuk pengembangan model penerjemah bahasa

menggunakan algoritma *encoder-decoder*. Dengan menerapkan proses ini dengan hati-hati, diharapkan bahwa model penerjemah akan mampu belajar dengan baik dari data yang diberikan dan menghasilkan penerjemahan yang akurat antara bahasa Palembang dan bahasa Indonesia.

g. One Hot Encoding

Teknik yang mengubah kata-kata menjadi vektor biner. Setiap kata diwakili oleh vektor yang panjangnya sama dengan ukuran kosakata, dengan satu elemen bernilai 1 (menunjukkan posisi kata tersebut) dan sisanya bernilai 0. Teknik ini memungkinkan model memproses teks secara numerik.

```
One Hot encoding
[ ] """converts a sentence to one hot encoding matrix - 2d matrix where 1 is 1 and 0 is 0"""
def one_hot_encoding(sentence, vocab):
    """converts a sentence to one hot encoding matrix for each word"""
    vocab_size = len(vocab.get_vocab('tokens'))
    one_hot = torch.zeros((len(sentence), vocab_size))
    for word in sentence.split():
        one_hot[word.get_vocab_index('tokens')] = 1
    return one_hot

def one_hot_encoding(sentence, vocab):
    """converts a sentence to one hot encoding matrix for each word"""
    vocab_size = len(vocab.get_vocab('tokens'))
    one_hot = torch.zeros((len(sentence), vocab_size))
    for word in sentence.split():
        one_hot[word.get_vocab_index('tokens')] = 1
    return one_hot

"""converts a pair of sentences (source and target) to a pair of tensors"""
def one_hot_encoding(source, target, vocab):
    """converts a pair of sentences (source and target) to a pair of tensors"""
    source_vocab_size = len(vocab.get_vocab('tokens'))
    target_vocab_size = len(vocab.get_vocab('tokens'))
    source_one_hot = torch.zeros((len(source), source_vocab_size))
    target_one_hot = torch.zeros((len(target), target_vocab_size))
    for word in source.split():
        source_one_hot[word.get_vocab_index('tokens')] = 1
    for word in target.split():
        target_one_hot[word.get_vocab_index('tokens')] = 1
    return source_one_hot, target_one_hot
```

Gambar 4. 3. 6 One Hot Encoding

h. Padding

Padding yaitu metode menambahkan elemen khusus, biasanya nol, ke urutan data untuk memastikan bahwa semua input dan output memiliki panjang yang sama. Teknik ini sangat penting untuk memastikan bahwa model dapat memproses batch data secara efisien dan konsisten, meskipun panjang kalimat dapat berbeda.

i. Encoder-Decoder

Dalam penelitian ini, algoritma *Encoder-Decoder* diterapkan untuk membuat mesin penerjemah dari bahasa Besemah ke bahasa Indonesia. Algoritma ini memiliki dua komponen utama, yaitu encoder dan decoder. *Encoder* bertugas mengubah input teks dalam bahasa Palembang menjadi representasi vektor yang lebih abstrak. Sedangkan *decoder* menerima representasi vektor tersebut dan menghasilkan teks dalam bahasa Indonesia sebagai hasil terjemahan. Berikut adalah rumus dasar algoritma *Encoder-Decoder*.

```
Encoder-Decoder
class EncoderDecoder:
    def __init__(self, vocab, embedding_dim, hidden_dim):
        self.vocab = vocab
        self.embedding_dim = embedding_dim
        self.hidden_dim = hidden_dim
        self.encoder = nn.LSTM(embedding_dim, hidden_dim, batch_first=True)
        self.decoder = nn.LSTM(hidden_dim, hidden_dim, batch_first=True)
        self.hidden_state = None
        self.output = None

    def encode(self, source):
        """converts a sentence to one hot encoding matrix for each word"""
        source_vocab_size = len(self.vocab.get_vocab('tokens'))
        source_one_hot = torch.zeros((len(source), source_vocab_size))
        for word in source.split():
            source_one_hot[word.get_vocab_index('tokens')] = 1
        return source_one_hot

    def decode(self, hidden_state):
        """converts a hidden state to a sentence"""
        decoder_vocab_size = len(self.vocab.get_vocab('tokens'))
        decoder_one_hot = torch.zeros((1, decoder_vocab_size))
        decoder_one_hot[0, self.vocab.get_vocab_index('tokens')] = 1
        decoder_output, decoder_hidden_state = self.decoder(decoder_one_hot, hidden_state)
        return decoder_output, decoder_hidden_state

    def encode_decode(self, source):
        """converts a sentence to one hot encoding matrix for each word"""
        source_one_hot = self.encode(source)
        hidden_state, _ = self.encoder(source_one_hot)
        decoder_output, decoder_hidden_state = self.decode(hidden_state)
        return decoder_output
```

Gambar 4. 5. 1 Class Encoder RNN

```
class Decoder(nn.Module):
    def __init__(self, hidden_size, vocab_size, num_layers, dropout):
        super(Decoder, self).__init__()
        self.hidden_size = hidden_size
        self.vocab_size = vocab_size
        self.num_layers = num_layers
        self.dropout = dropout
        self.embedding = nn.Embedding(vocab_size, hidden_size)
        self.rnn = nn.LSTM(hidden_size, hidden_size, num_layers, batch_first=True, dropout=dropout)
        self.decoder_input = nn.Embedding(vocab_size, hidden_size)
        self.decoder_output = nn.Linear(hidden_size, vocab_size)

    def forward(self, encoder_outputs, decoder_input):
        # Flatten encoder outputs
        encoder_outputs = encoder_outputs.view(-1, self.hidden_size)
        # Initialize decoder hidden state
        decoder_hidden, decoder_cell_state = self.rnn.reset_hidden()
        # Decode
        decoder_outputs = []
        for i in range(decoder_input.size()[0]):
            decoder_input_embed = self.decoder_input(decoder_input[i])
            decoder_hidden, decoder_cell_state = self.rnn(decoder_input_embed, decoder_hidden)
            decoder_output = self.decoder_output(decoder_hidden)
            decoder_outputs.append(decoder_output)
        decoder_outputs = torch.nn.functional.softmax(torch.stack(decoder_outputs, dim=0))
        return decoder_outputs
```

Gambar 4. 5. 2 Class Decoder

HASIL

Dalam tahap ini, akan dilakukan pelatihan model *Encoder-Decoder*. Setelah model dibuat sebelumnya, akan dilakukan pelatihan agar model tersebut dapat melakukan penerjemahan bahasa Indonesia dan bahasa Palembang. Pelatihan yang dilakukan menggunakan 100 epoch.

```
training_data = torch.utils.data.TensorDataset(train_loader.get_data_loader(), train_loader.get_data_loader())
train_loader = torch.utils.data.DataLoader(training_data)
vocab = Vocabulary.from_instances(train_loader.get_data_loader(), min_count={'tokens': 3})
encoder = Encoder(vocab.get_vocab('tokens'), hidden_size, num_layers, dropout)
decoder = Decoder(hidden_size, vocab.get_vocab('tokens'), num_layers, dropout)
optimizer = optim.Adam(encoder.parameters() + decoder.parameters())
trainer = GradientDescentTrainer(encoder, decoder, train_loader, vocab, optimizer)
```

Gambar 4. 6. 1 Pelatihan Model

Dari hasil keluaran pelatihan model yang diberikan, maka hasil dari model *Encoder-Decoder* ini sebagai berikut:

a. Membaca dan Memproses Data

Membaca 6341 pasangan kalimat, dan

setelah dipangkas menjadi 5753 pasangan kalimat

b. Menghitung Kosakata

Menghitung kata dalam korpus, kosakata bahasa Indonesia adalah 5742 sedangkan kosakata bahasa Palembang adalah 5782, termasuk tag EOS dan SOS

c. Pelatihan Model

Menggunakan 6341 pasangan kalimat untuk pelatihan tanpa data pengujian. *Encoder* dan *Decoder* telah dibuat menggunakan arsitektur *Seq2Seq*. Pelatihan menggunakan GPU menggunakan CUDA selama 100 epoch.

d. Evaluasi Model

Evaluasi model penerjemah bahasa Palembang ke bahasa Indonesia dimulai dengan persiapan data set yang terdiri dari pasangan kalimat atau teks yang maknanya setara dalam kedua bahasa. Data ini digunakan untuk melatih model agar dapat mengenali pola-pola bahasa yang relevan dan menggunakannya dalam terjemahan dalam menangani berbagai aspek penerjemahan, seperti akurasi, kelancaran, dan konsistensi. Selain itu, perplexity digunakan untuk mengevaluasi seberapa baik model dalam memprediksi rangkaian kata atau kalimat dalam bahasa Indonesia atau bahasa Palembang ada pun beberapa tantangan spesifik. Salah satunya adalah keunikan bahasa Palembang yang mungkin memiliki idiom, kosakata, atau struktur kalimat yang berbeda dari bahasa Indonesia standar, sehingga model perlu dirancang untuk mampu menangani variasi ini dengan baik.

```

Train Pairs #
5753
Current mem usage:
0.0
Current mem usage:
0.0
Encoder and Decoder Created
Current mem usage:
0.0
Cuda being used
Number of epochs: 100
Current mem usage:
10.1770752
Iter: 1
Learning Rate: 1
Time: 0h 0m 9s
Train Loss: 1.9840219055063079

> seisi sele .
= seii .
< . <EOS>

> tiang rumah panggung .
= becagak .
< . <EOS>

> bau makanan yang lama tersimpan .
= tengik .
< . <EOS>

```

Gambar 4. 7. 1 Evaluasi Model Iter 1

```

Iter: 100
Learning Rate: 1
Time: 0h 11m 53s
Train Loss: 0.03358078065077321

> raba .
= rabo .
< rabo . <EOS>

> adakala .
= adokalo .
< adokalo . <EOS>

> mengajak .
= ngajak .
< ngajak . <EOS>

> akil baigh .
= baler .
< baler . <EOS>

```

Gambar 4. 7. 2 Evaluasi Model Iter 100

Dari gambar diatas. Menggambarkan proses melatih kalimat dan kata. Hal ini dilakukan dengan membuat kalimat sesuai dengan penerjemahan yang sudah dibuat sebelumnya dengan dataset. Agar kalimat atau kata yang dipanggil sesuai dengan yang dimaksud. Proses ini membutuhkan waktu 11 menit sampai semua kalimat atau kata sudah benar-benar bisa dijalankan atau dipanggil.

e. Penerjemah

Pada proses hasil penerjemahan ini dapat menghasilkan proses perubahan terjemahan teks dari bahasa Palembang ke bahasa Indonesia dan sebaliknya. Penerjemahan dari bahasa inti agar pesan dan makna dapat dipahami dengan baik oleh pembaca yang menerjemahkan dari bahasa tersebut.

KESIMPULAN

Penelitian ini berhasil membuktikan bahwa penerapan algoritma *encoder-decoder* dalam pengembangan mesin penerjemah bahasa Palembang ke bahasa Indonesia dapat menghasilkan sistem yang mampu melakukan terjemahan dengan tingkat akurasi yang tinggi. Melalui serangkaian uji coba dan evaluasi, ditemukan bahwa algoritma *encoder-decoder* mampu menangani perbedaan struktur kalimat dan kosakata antara bahasa Palembang dan bahasa Indonesia dengan cukup baik, menghasilkan terjemahan yang natural dan sesuai konteks. Penggunaan model ini memungkinkan pengolahan data bahasa Palembang yang kompleks dan beragam menjadi bentuk bahasa Indonesia yang lebih mudah dipahami. Keberhasilan penelitian ini tidak terlepas dari pemilihan data latih yang representatif dan penggunaan teknik *pre-processing* yang tepat, yang berkontribusi terhadap performa model yang optimal. Meskipun begitu, diperlukan analisis yang lebih mendalam dan evaluasi lebih lanjut untuk meningkatkan kualitas terjemahan yang dihasilkan, sehingga sistem penerjemahan bahasa di masa depan dapat menjadi lebih baik.

DAFTAR PUSTAKA

- [1]Abidin, Z (2018). Translation of Sentence Lampung-Indonesian Languages with Neural Machine Translation Attention Based Approach. *Inovasi Pembangunan: Jurnal Kelitbangan*, 6(02), 191-206. <https://doi.org/10.35450/jip.v6i02.97>

- [2]Afria, R., Sanjaya, D., & Tiara, M. (2020). Leksikostatistik dan Grotokronologi Bahasa Melayu Palembang, Basemah Lahat, Basemah Pagaralam, dan Kayu Agung: Kajian Linguistik Historis Komparatif. *Madah*, 11(1), 27-42. <https://doi.org/10.31503/madah.v11i1.223>
- [3]AKBAR, G. (2021). *Automatic Text Summarization Berita BahasIndonesia Menggunakan Metode Attentional Encoder Decoder*.
- [4]Amalia Yunia Rahmawati. (2020). *BIAS GENDER DALAM MESIN PENERJEMAHAN (GOOGLE TRANSLATE) DARI BAHASA INDONESIA KE BAHASA INGGRIS*. July, 1-23.
- [5]Andri, A. (2019). Penerapan Algoritma Pencarian Binary Search dan QuickSort pada Aplikasi Kamus Bahasa Palembang Berbasis Web. *Jurnal Informatika: Jurnal Pengembangan IT*, 4(1), 70-74. <https://doi.org/10.30591/jpit.v4i1.1104>
- [6]Lanners, Q. M., & Laurent, T. (2019). *Neural Machine Translation*.
- [7]Wismoyo, P. A. (2018). Mesin penerjemah bahasa inggris - indonesia berbasis jaringan saraf tiruan dengan mekanisme attention menggunakan arsitektur transformer skripsi. *Undip E-Journal System*.

The screenshot shows the submission interface for the journal RABIT. The article title is "MESIN PENERJEMAH BAHASA PALEMBANG KE BAHASA INDONESIA MENGGUNAKAN ALGORITMA ENCODER-DECODER" by Luthi Hengky Panthawan. The submission status is "Submitted". The submission files section shows a single file named "19021 - luthihengkypanthawan_jurnal Luthi Hengky Panthawan.docx" with a size of 1000 bytes. The pre-review discussions section is currently empty.

Submission Files				Search
19021 - luthihengkypanthawan_jurnal Luthi Hengky Panthawan.docx	1000 bytes			
				Download All Files

Pre-Review Discussions				Add discussion
Name	Post	Last Post	Replies	Search



SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama: **Laode Hengky Randawan**
NIM: **20190038**
Program Studi: **Sistem Informasi**
Jurusan: **Sains Teknologi**
No. WA: **0821-5090-2900**
Nama Pembimbing: **Uman Zuhri, Tbk, S.kom, M.P, M.kom**
Judul Artikel: **Mesin Penerjemah Fajembang ke Bahasa Indonesia Menggunakan Algoritma Encoder-Decoder**

Menyatakan memang benar belum mendapatkan Letter of Acceptance (LoA) dan mesin telah submit/terunggah proses review dari pihak penerbit jurnal. Mengingat keberadaan artikel sedang berlangsung, untuk itu saya mohon dapat diizinkan untuk melakukan update keupload, belum mendapatkan LoA, dengan konsekuensi tidak mendapatkan Transkrip Akademik karya. Saya sudah sadar tidak akan mendapat Transkrip Akademik saya sebelum saya mendapatkan LoA dan pengumpulan ke Pusat Pelayanan Mahasiswa (PPM).

Demikian surat pernyataan ini saya buat dengan sebenarnya untuk dipergunakan sebagaimana mestinya. Terima kasih.

Mengetahui,
Ketua Program Studi


Nao Roco Damayanti, M.kom, Ph.D.

Pembimbing: **Septor 148 2022**
Harianit 2024




Laode Hengky Randawan

Lampiran:
Bukti submit artikel