

**APLIKASI PENERJEMAH BAHASA BANGKA – INDONESIA  
– INGGRIS BERBASIS WEBSITE DENGAN NEURAL  
MACHINE TRANSLATION (NMT)**

**PROYEK AKHIR**

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan  
Diploma IV Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

FADEL RAZSIAH

NIM: 1061911

**POLITEKNIK MANUFAKTUR NEGERI  
BANGKA BELITUNG  
TAHUN 2023**

## LEMBAR PENGESAHAN

### APLIKASI PENERJEMAH BAHASA BANGKA – INDONESIA – INGGRIS BERBASIS WEBSITE DENGAN NEURAL MACHINE TRANSLATION (NMT)

Oleh:

FADEL RAZSIAH /1061911

Laporan ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1

Pembimbing 2



Ahmat Josi, M.Kom  
NIP. 198908202019031015



Sari Mubaroh, M.Pd  
NIP. 198501122019032015

Penguji 1

Penguji 2



Linda Fujiyanti, M.Ti  
NIP. 198109262014042001



Indah Riezky Pratiwi, M.Pd  
NIP. 199010082019032018

## **PERNYATAAN BUKAN PLAGIAT**

Yang bertanda tangan di bawah ini:

Nama Mahasiswa : FADEL RAZSIAH  
NIM : 1061911  
Dengan Judul : APLIKASI PENERJEMAH BAHASA BANGKA –  
INDONESIA – INGGRIS BERBASIS WEBSITE  
DENGAN NEURAL MACHINE TRANSLATION  
(NMT)

Menyatakan bahwa laporan akhir ini adalah hasil kerja saya sendiri dan bukan merupakan plagiat. Pernyataan ini saya buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, saya bersedia menerima sanksi yang berlaku.

Sungailiat, 18 Januari 2023

Nama Mahasiswa

Tanda Tangan

Fadel Razsiah



## ABSTRAK

*Pada saat ini mesin penerjemah merupakan salah satu pilihan untuk melakukan terjemahan bahasa, terutama bagi orang-orang yang ingin mempelajari suatu bahasa. Bahasa Bangka merupakan Bahasa yang sering digunakan masyarakat Bangka untuk keseharian. Telah banyak pengembangan mesin penerjemah, tetapi belum ada yang mengembangkan mesin penerjemah untuk bahasa Bangka. Pengembangan mesin penerjemah dengan Neural Machine Translation (NMT) dan implementasinya dengan menggunakan microframework Flask merupakan langkah awal untuk pengembangan mesin penerjemah bahasa Bangka. Penelitian ini bertujuan untuk membuat dan mengetahui hasil terjemahan aplikasi penerjemah bahasa Bangka – bahasa Indonesia – bahasa Inggris dengan model penerjemah RNN. Pengembangan dimulai dari membuat korpus paralel bahasa Bangka – Indonesia, bahasa Bangka – Inggris dan bahasa Inggris – Indonesia dan arsitektur mesin penerjemah dengan model RNN. Metode penelitian dimulai dari pengumpulan dataset, preprocessing data, pemodelan dan pelatihan, evaluasi dan implementasi sistem. Dari hasil evaluasi BLEU Scores didapatkan nilai sebesar 55,3% untuk bahasa Bangka – Indonesia, bahasa Bangka – bahasa Inggris didapatkan akurasi sebesar 64,7%, bahasa Indonesia – Bangka didapatkan akurasi sebesar 54,5%, bahasa Indonesia – Inggris didapatkan akurasi sebesar 55,8%, bahasa Inggris – Indonesia didapatkan akurasi sebesar 40,2% dan bahasa Inggris – Bangka didapatkan akurasi sebesar 37,3%. Selanjutnya model diimplementasikan ke bentuk website dengan memanfaatkan framework Flask, sehingga pengguna dapat dengan mudah dalam melakukan penerjemahan.*

**Kata kunci :** BLEU Score, Flask, Korpus Paralel, Mesin Penerjemah, RNN;

## **ABSTRACT**

*At this time machine translation is one of the options for translating languages, especially for people who want to learn a language. Bangka language is a language that is often used by the Bangka people for everyday life. There have been many developments for machine translation, but no one has yet developed a translation machine for the Bangka language. The development of a machine translation using Neural Machine Translation (NMT) and its implementation using the Flask microframework is the first step for the development of a machine translation for the Bangka language. This study aims to make and find out the translation results of the Bangka language translator application – Indonesia – English with the RNN translator model. The development starts with creating a parallel corpus of Bangka-Indonesian, Bangka-English, and English – Indonesian languages and machine translation architecture with the RNN model. The research method starts with dataset collection, data preprocessing, modeling and training, system evaluation, and implementation. From the results of the BLEU Scores evaluation, a value of 55.3% was obtained for Bangka - Indonesia, Bangka – English obtained an accuracy of 64.7%, Indonesian - Bangka obtained an accuracy of 54.5%, Indonesian - English obtained an accuracy of 55.8%, English – Indonesia obtained an accuracy of 40.2% and English – Bangka obtained an accuracy of 37.3%. Furthermore, the model is implemented in the form of a website by utilizing the Flask framework, so that users can easily translate.*

**Keywords :** *BLUE Score, Corpus Parallel, Flask, Machine Translate, RNN;*

## KATA PENGANTAR

*Assalamualaikum warahmatullahi wabarakatuh,*

Puji dan syukur penulis panjatkan kehadiran Allah SWT karena atas berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan penyusunan laporan proyek akhir dengan judul “**Aplikasi Penerjemah Bahasa Bangka – Indonesia – Inggris Berbasis Website dengan Neural Machine Translation (NMT)**”.

Tidak lupa penulis mengucapkan terima kasih kepada berbagai pihak yang telah memberikan doa, dorongan dan motivasi sehingga Proyek Akhir ini dapat terlaksana dengan baik. Oleh karena itu pada kesempatan kali ini penulis ingin menyampaikan terima kasih kepada:

1. Bapak I Made Andik Setiawan, M.Eng., Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
2. Bapak Irwan, M.Sc., Ph.D selaku Wakil Direktur I Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Muhammad Subhan, M.T selaku Wakil Direktur II Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Eko Sulistyono, M.T selaku Wakil Direktur III Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak M. Iqbal Nugraha, M.Eng selaku Kepala Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
6. Bapak Ahmat Josi, M.Kom selaku Kepala Prodi D-IV Teknologi Rekayasa Perangkat Lunak sekaligus Dosen Pembimbing I pada Proyek Akhir Politeknik Manufaktur Negeri Bangka Belitung.
7. Ibu Sari Mubaroh, M.Pd selaku Dosen Pembimbing II pada Proyek Akhir Politeknik Manufaktur Negeri Bangka Belitung.
8. Terkhusus dan Istimewa kepada Alm. Ayah, Ibu, dan Keluarga yang telah memberikan dukungan materil dan moril.

9. Kepada teman – teman penulis (Apri, Zuhri, Maula, Anjas, Ego, Rio, Monic Hansen, Diego, Rizki, Gaspar, Trializa dan Tri) yang selalu memberikan bantuan sehingga mengantarkan penulis untuk menyelesaikan Laporan Proyek Akhir ini.

Penulis menyadari bahwa penyusunan laporan akhir ini masih terdapat banyak kekurangan dan jauh dari kata sempurna, maka dari itu penulis memohon maaf apabila dalam penulisan terdapat kesalahan kata, pengetikan dan kekeliruan sehingga membingungkan para pembaca dalam memahami maksud dari isi laporan. Oleh karena itu, penulis mengharapkan kritik serta saran yang bersifat membangun sehingga dapat menyempurnakan laporan ini dan menjadi acuan dalam menyusun laporan selanjutnya.

Doa terbaik penulis sampaikan untuk siapapun yang saat ini sedang berjuang di bangku kuliah demi membahagiakan kedua orang tua, keluarga, dan yang ingin menggapai cita – cita. “Tidak ada mimpi yang gagal, yang ada hanyalah mimpi yang tertunda. Sekiranya teman – teman merasa gagal dalam menggapai mimpi, jangan khawatir, mimpi – mimpi lain bisa diciptakan. Jangan menyerah, tetaplah berjuang, bangkit dari keterpurukan, karena penulis yakin kita semua disini adalah petarung untuk kehidupan yang keras ini.” Akhir kata penulis ucapkan terimakasih.

*Wassalamualaikum warahmatullahi wabarakatuh.*

Sungailiat, 18 Januari 2023



Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN .....	ii
PERNYATAAN BUKAN PLAGIAT .....	iii
ABSTRAK.....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR.....	vi
DAFTAR ISI .....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL .....	xi
DAFTAR LAMPIRAN .....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	2
1.3 Tujuan Proyek Akhir .....	3
BAB II DASAR TEORI .....	4
2.1 Kajian Teori .....	4
2.1.1 <i>Neural Machine Translation</i> (NMT) .....	4
2.1.2 <i>Natural Language Processing</i> (NLP).....	4
2.1.3 <i>Deep Learning</i> .....	6
2.1.4 <i>Reccurent Neural Network</i> (RNN) .....	7
2.1.5 <i>Long Short Term Memory</i> (LSTM) .....	11
2.1.6 <i>Bilingual Evaluation Understudy</i> (BLEU).....	15
2.1.7 Tensorflow .....	16
2.1.8 Flask.....	17
2.2 Penelitian Terkait.....	17
BAB III METODE PELAKSANAAN.....	21
3.1 Alur Penelitian .....	21
3.2 Uraian Penelitian .....	22
3.2.1 Pengumpulan Data.....	22



3.2.2 <i>Preprocessing</i> Data.....	23
3.2.3 Pemodelan dan Pelatihan .....	23
3.2.4 Evaluasi.....	23
3.2.5 Implementasi Sistem.....	24
BAB IV HASIL DAN PEMBAHASAN.....	25
4.1 Hasil Pengumpulan Data .....	25
4.2 <i>Preprocessing</i> Data .....	26
4.2.1 <i>Case Folding</i> .....	27
4.2.2 <i>Removing Punctuation</i> .....	29
4.2.3 <i>Removing Digits</i> .....	31
4.2.4 <i>Tokenization</i> .....	34
4.3 Pemodelan dan Pelatihan .....	36
4.3.1 Pembuatan Model .....	37
4.3.2 Optimasi.....	37
4.3.3 Pelatihan Model .....	38
4.4 Evaluasi.....	39
4.4.1 Aktifitas Grafik <i>Loss</i> .....	39
4.4.2 BLEU Scores .....	44
4.5 Implementasi Sistem.....	46
4.5.1 Pembuatan <i>User Interface</i> .....	47
4.5.2 Tampilan Aplikasi .....	48
BAB V KESIMPULAN DAN SARAN .....	49
5.1 Kesimpulan .....	49
5.2 Saran .....	49
DAFTAR PUSTAKA.....	50

## DAFTAR GAMBAR

Gambar 2. 1 Model <i>Encoder – Decoder</i> .....	4
Gambar 2. 2 Ilustrasi <i>Deep Learning</i> .....	7
Gambar 2. 3 <i>Recurrent Neural Network (RNN)</i> .....	8
Gambar 2. 4 <i>Unfolding Recurrent Neural Network</i> .....	8
Gambar 3. 1 Alur Penelitian .....	21
Gambar 3. 2 Buku "Kamus Bahasa Melayu Bangka - Indonesia .....	22
Gambar 4. 1 <i>Flowchart Preprocessing Data</i> .....	27
Gambar 4. 2 Arsitektur Model RNN .....	36
Gambar 4. 3 Struktur Model pada Pelatihan Model .....	38
Gambar 4. 4 <i>Loss dan Validation Loss</i> Model Penerjemah Bangka - Indonesia ..	40
Gambar 4. 5 <i>Loss dan Validation Loss</i> Model Penerjemah Bangka – Inggris .....	41
Gambar 4. 6 <i>Loss dan Validation Loss</i> Model Penerjemah Indonesia – Bangka .	41
Gambar 4. 7 <i>Loss dan Validation Loss</i> Model Penerjemah Indonesia – Inggris ..	42
Gambar 4. 8 <i>Loss dan Validation Loss</i> Model Penerjemah Inggris - Indonesia...	43
Gambar 4. 9 <i>Loss dan Validation Loss</i> Model Penerjemah Inggris – Bangka .....	43
Gambar 4. 10 Arsitektur Aplikasi .....	46
Gambar 4. 11 <i>User Interface</i> .....	47
Gambar 4. 12 Tampilan <i>Website</i> .....	48

## DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait.....	18
Tabel 4. 1 Korpus Paralel Bahasa Bangka - Bahasa Indonesia .....	25
Tabel 4. 2 Korpus Paralel Bahasa Bangka - Bahasa Inggris.....	26
Tabel 4. 3 Korpus Paralel Bahasa Indonesia - Bahasa Inggris .....	26
Tabel 4. 4 Hasil <i>Case Folding</i> Korpus Paralel Bangka - Indonesia .....	28
Tabel 4. 5 Hasil <i>Case Folding</i> Korpus Paralel Bangka - Inggris.....	28
Tabel 4. 6 Hasil <i>Case Folding</i> Korpus Paralel Indonesia - Inggris .....	29
Tabel 4. 7 Hasil <i>Removing Punctuation</i> Korpus Paralel Bangka - Indonesia .....	30
Tabel 4. 8 Hasil <i>Removing Punctuation</i> Korpus Paralel Bangka - Inggris.....	30
Tabel 4. 9 Hasil <i>Removing Punctuation</i> Korpus Paralel Indonesia - Inggris .....	31
Tabel 4. 10 Hasil <i>Removing Digits</i> Korpus Paralel Bangka - Indonesia .....	32
Tabel 4. 11 Hasil <i>Removing Digits</i> Korpus Paralel Bangka - Inggris .....	33
Tabel 4. 12 Hasil <i>Removing Digits</i> Korpus Paralel Indonesia - Inggris .....	33
Tabel 4. 13 Hasil <i>Tokenization</i> Korpus Paralel Bangka - Indonesia .....	34
Tabel 4. 14 Hasil <i>Tokenization</i> Korpus Paralel Bangka - Inggris.....	35
Tabel 4. 15 Hasil <i>Tokenization</i> Korpus Paralel Indonesia - Inggris .....	35
Tabel 4. 16 Hasil Aktifitas Grafik <i>Loss</i> .....	40
Tabel 4. 17 Hasil Evaluasi BLEU <i>Scores</i> .....	44
Tabel 4. 18 Hasil Evaluasi BLEU <i>Scores</i> Penelitian Terkait.....	45

## DAFTAR LAMPIRAN

Lampiran 1: Daftar Riwayat Hidup

Lampiran 2: Form Monitoring

Lampiran 3: Form Bimbingan-1

Lampiran 4: Form Bimbingan-2

Lampiran 5: Plagiasi Turnitin

Lampiran 6: Bukti Publikasi

Lampiran 7: *Source Code*



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Penggunaan mesin penerjemah bahasa sangat sering kita gunakan dalam keseharian, selain untuk pembelajaran bahasa mesin penerjemah juga berperan penting dalam pariwisata maupun perdagangan. Mesin penerjemah seperti, *google translate* juga mampu membantu para pelajar untuk belajar bahasa asing dengan lebih efisien. Namun, dengan banyaknya bahasa daerah yang terdapat di Indonesia tidak mampu untuk dikelola oleh *Google Translate* itu sendiri termasuk bahasa Bangka. Indonesia mempunyai keragaman bahasa sebanyak 726 bahasa daerah dan 719 diantaranya masih sering digunakan dalam keseharian masyarakat Indonesia [1].

Mesin penerjemah merupakan pilihan utama setiap orang untuk sekedar mengetahui ataupun mempelajari suatu bahasa di luar bahasa keseharian mereka. Bahasa Bangka merupakan bahasa yang digunakan dalam aktivitas seluruh masyarakat Kepulauan Bangka Belitung [2]. Asal-usul bahasa ini masih sering diperdebatkan antara berasal dari Sumatera yang bermuara dari kerajaan Sriwijaya atau berasal dari Kalimantan yang berkembang dan bermigrasi hingga ke Pulau Sumatera. Meskipun demikian, bahasa Melayu Bangka tetap merupakan salah satu asset Negara yang berjasa besar dalam membangun peradaban masyarakat Melayu dan Indonesia. Pengembangan model mesin penerjemah dari bahasa Bangka ke dalam bahasa Indonesia dan bahasa Inggris ataupun sebaliknya dapat menjadi langkah awal dalam proses meningkatkan kemudahan untuk belajar bahasa Bangka.

Terdapat beberapa penelitian dalam pengembangan model mesin penerjemah, seperti mesin penerjemah dengan *neural network* [3]. Mesin penerjemah dengan *neural network* dan mesin penerjemah dengan statis, dengan hasil penelitian bahwa model *recurrent neural network* (RNN) memiliki akurasi

dan performa yang lebih baik dibandingkan dengan mesin penerjemah dengan statis [4].

Penelitian lainnya juga mengatakan bahwa mesin penerjemah dengan model RNN memiliki performa yang lebih baik dari model penerjemah dengan model statis, ketika menggunakan sumber bahasa yang sedikit (*low resource language*) [5]. Hal ini membuktikan bahwa mesin penerjemah dengan model RNN dapat digunakan dalam mesin penerjemah Bahasa Bangka yang dapat dikategorikan memiliki sumber data bahasa yang sedikit.

Maka dari itu dilakukannya penelitian dengan judul “**Aplikasi Penerjemah Bahasa Bangka – Indonesia – Inggris dengan *Neural Machine Translation* Berbasis *Website***”. Penelitian ini bertujuan untuk merancang model mesin penerjemah berbasis RNN untuk melakukan penerjemahan dari Bahasa Bangka – Bahasa Indonesia – Bahasa Inggris. Pada penelitian ini selain menghasilkan model mesin penerjemah juga menghasilkan *dataset* berupa korpus paralel Bahasa Bangka – Bahasa Indonesia, Bahasa Bangka – Bahasa Inggris dan juga korpus paralel Bahasa Indonesia – Bahasa Inggris. Pada akhir penelitian model akan diimplementasikan ke dalam aplikasi berbentuk *website* menggunakan *microframework* Flask.

## 1.2 Perumusan Masalah

Berdasarkan beberapa penjelasan yang telah disampaikan pada latar belakang, dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana membuat aplikasi penerjemah dari model penerjemah RNN untuk terjemahan Bahasa Bangka – Bahasa Indonesia – Bahasa Inggris menggunakan?
2. Bagaimana hasil terjemahan dari model penerjemah RNN yang digunakan pada mesin penerjemah untuk hasil terjemahan Bahasa Bangka – Bahasa Indonesia – Bahasa Inggris?

### **1.3 Tujuan Proyek Akhir**

Berdasarkan permasalahan yang telah dirumuskan, adapun tujuan yang ingin dicapai pada proyek akhir ini sebagai berikut:

1. Mengetahui hasil dari pembuatan aplikasi penerjemah Bahasa Bangka – Bahasa Indonesia – Bahasa Inggris dengan model penerjemah RNN.
2. Mengetahui hasil terjemahan dari mesin penerjemahan Bahasa Bangka – Bahasa Indonesia – Bahasa Inggris pada model penerjemah RNN.



## BAB II

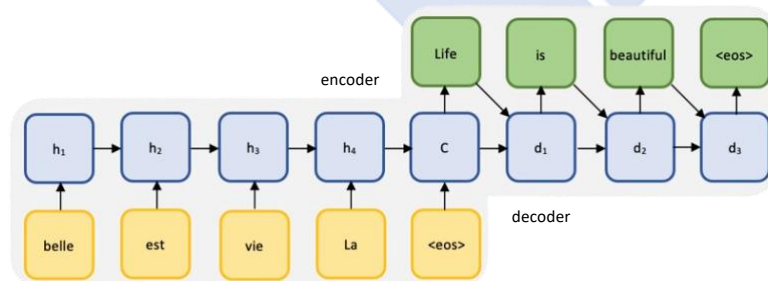
### DASAR TEORI

#### 2.1 Kajian Teori

##### 2.1.1 *Neural Machine Translation* (NMT)

*Neural Machine Translation* (NMT) merupakan metode translasi yang menggunakan RNN dengan *encoder* dan *decoder* [6]. Pada saat ini metode yang sering digunakan yaitu metode *attention* atau perhatian [3]. Pada saat melakukan proses terjemahan menggunakan RNN terlebih dahulu terhubung dengan *encoder* dan *decoder*.

Pada *encoder* yang diterapkan RNN akan memproses kata ataupun frasa pada sumber bahasa menjadi sebuah kalimat yang lebih mudah dimengerti oleh mesin. Sedangkan untuk *decoder* RNN akan mengambil *output* dari *encoder* pada RNN untuk dijadikan sebagai sebuah *input*-an, kemudian dapat menghasilkan hasil terjemahan dari kata ke kata [7]. Arsitektur pada model mesin penerjemah juga dapat dikatakan model *encoder-decoder* [8]. Pada (Gambar 2.1) menampilkan bagaimana sebuah model *encoder* dan *decoder* saling terhubung.



Gambar 2. 1 Model *Encoder – Decoder*

##### 2.1.2 *Natural Language Processing* (NLP)

*Natural Language Processing* (NLP) adalah bidang studi yang mempelajari atau mengembangkan algoritma atau sistem yang dapat dipahami oleh komputer untuk melakukan berbagai perintah yang berhubungan dengan bahasa manusia [9].



Linguistik komputasi, pemrosesan bahasa, dan teknologi yang terkait dengan bahasa manusia semuanya adalah nama untuk NLP. NLP sering digunakan untuk analisis bahasa manusia tertulis dan lisan.

NLP sering digunakan dalam membangun beberapa aplikasi seperti mesin penerjemah, *chatbot* dan beberapa aplikasi lain yang berkaitan dengan pemrosesan bahasa manusia. NLP sangat membutuhkan mesin yang dapat membaca, memahami ataupun mengambil sebuah informasi dari bahasa manusia yang tidak terstruktur dan abstrak bagi mesin.

NLP bertujuan untuk mengubah bahasa manusia menjadi data yang lebih terstruktur [10]. Berikut merupakan beberapa teknik dasar NLP dalam memecahkan berbagai permasalahan:

1. Segmentasi Kalimat

Segmentasi kalimat merupakan sebuah teknik yang digunakan untuk mendefinisikan sebuah unit pada pemrosesan yang terdiri dari satu atau lebih kata. Biasanya teknik ini melibatkan proses yang mengidentifikasi batas-batas dari sebuah kalimat antar kata dalam sebuah kalimat yang berbeda [11].

2. Tokenisasi

Tokenisasi adalah teknik dasar NLP. Metode ini memecah kalimat menjadi klausa yang mengandung kata atau frasa [12].

3. *Stemming* dan Lemmatisasi

Tujuan mendasar lemmatisasi dan derivasi adalah setara, kedua teknik tersebut mengurangi variasi kata dengan menemukan akar kata [13]. Derivasi menempatkan akar dengan mengikuti aturan seperti menghilangkan imbuhan, tetapi lemmatisasi menempatkan akar dengan memahami konteks kata dan menemukan bentuk standarnya.

4. Penghapusan Kata Henti

Penghapusan kata berhenti meningkatkan akurasi dengan menghilangkan kata-kata yang mengandung sedikit informasi [14]. Kata-kata berhenti biasanya disebut sebagai kata-kata non-informatif. Dalam bahasa Indonesia,

misalnya, kata berhenti biasanya disebut dengan konjungsi seperti "dan", "atau", "yang", dan "kepada". Karena hanya berfokus pada informasi penting, mengurangi jumlah kata yang tidak diperlukan dapat mempercepat pemrosesan dan meningkatkan nilainya.

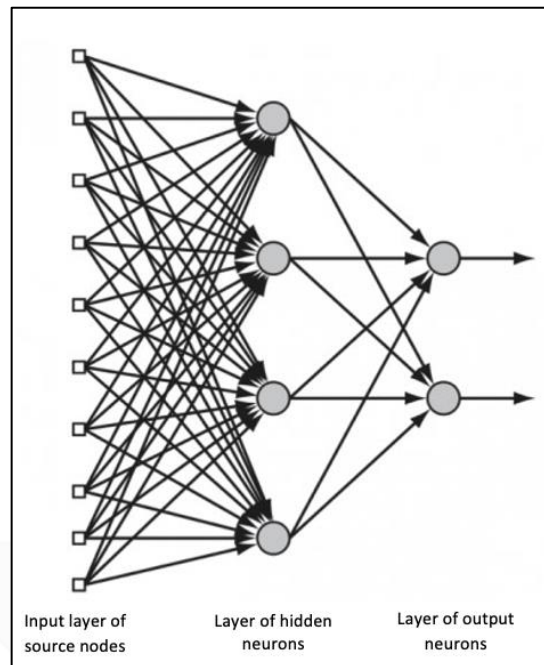
### **2.1.3 Deep Learning**

Bidang penelitian ilmu komputer yang dikenal sebagai deep learning baru saja muncul belakangan ini. Pembelajaran mendalam pada dasarnya adalah subbidang pembelajaran mesin yang memungkinkan komputer mempelajari atribut tanpa campur tangan manusia [10]. Metode yang memungkinkan komputer memperoleh pengalaman dan pengetahuan untuk meningkatkan keterampilannya [15]. *Deep learning* terbukti lebih mudah beradaptasi dan mampu menghasilkan model yang lebih akurat daripada pembelajaran mesin tradisional [16]. Ini didasarkan pada neuron otak. Jaringan saraf yang dalam, yang memiliki banyak lapisan dan banyak neuron di masing-masingnya, adalah nama lain untuk pembelajaran mendalam. Setiap lapisan mungkin juga mengandung ribuan neuron, dan jumlah lapisan dapat berkisar dari beberapa hingga ribuan.

Jaringan saraf merupakan teori pembelajaran yang mendalam. *Deep learning* dan jaringan saraf reguler pada dasarnya berbeda karena pembelajaran mendalam menggunakan lebih banyak neuron dan lapisan tersembunyi [17]. Karena memiliki lebih banyak lapisan transformasi, banyak orang percaya bahwa algoritma pembelajaran mendalam lebih unggul daripada jaringan saraf biasa. Data juga dapat diubah dari non-linier menjadi linier menggunakan deep learning.

Setiap lapisan tersembunyi dalam jaringan saraf bertugas melatih fitur-fitur khusus berdasarkan hasil jaringan sebelumnya [15]. Semakin banyak lapisan tersembunyi, semakin abstrak dan rumit algoritma ini. Hirarki berlapis-lapis sederhana pada tingkat tinggi atau banyak lapisan membentuk jaringan saraf pembelajaran yang mendalam. Karena merepresentasikan data sebagai konsep hierarki bersarang, di mana setiap konsep yang didefinisikan memiliki hubungan dengan konsep yang lebih sederhana, pembelajaran mendalam memiliki banyak

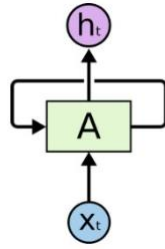
kekuatan dan fleksibilitas [15]. Ilustrasi bagaimana deep learning menggunakan diagram konsep untuk merepresentasikan suatu masalah ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Ilustrasi *Deep Learning*

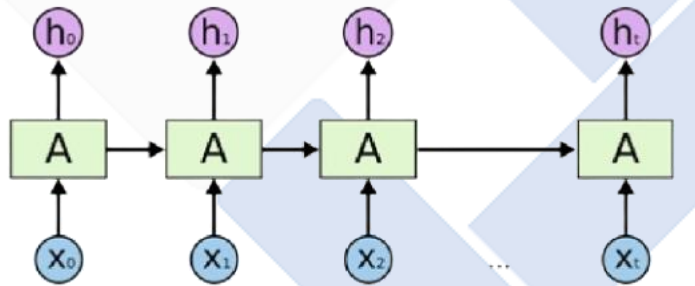
#### **2.1.4 Recurrent Neural Network (RNN)**

Algoritma *recurrent neural network* merupakan salah satu kategori dari deep learning. *Recurrent Neural Network* merupakan model yang meniru cara berpikir dalam pengambilan keputusan tersebut dimana RNN tidak membuang begitu saja informasi dari masa lalu dalam proses pembelajarannya yang secara otomatis informasi dari masa lalu tetap tersimpan. RNN memproses input secara sekuensial. Data sekuensial mempunyai karakteristik dimana sampel diproses dengan suatu urutan, dan suatu sampel dalam urutan mempunyai hubungan yang erat satu dengan yang lain [18]. Arsitektur *Recurrent Neural Network* dapat dilihat pada Gambar 2.3 *Recurrent Neural Network*.



Gambar 2. 3 *Recurrent Neural Network (RNN)*

Pada gambar diatas,  $X_t$  merupakan *input*-an terhadap  $t$  (waktu/urutan *input*-an berdasarkan waktu/*input*-an data ke- $t$ )  $h_t$  merupakan hasil *output*. Gambar diatas juga menggambarkan bahwa proses pada RNN dilakukan secara berulang-ulang sehingga data pada *input*-an sebelumnya tersimpan ke memori RNN. *Unfolding Recurrent Neural Network* dapat dilihat pada Gambar 2.4 *Unfolding Recurrent Neural Network*.



Gambar 2. 4 *Unfolding Recurrent Neural Network*

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

1. *Input Layer*

Merupakan sebuah lapisan yang menerima beberapa *input* yang kemudian akan diteruskan ke neuron lainnya pada jaringan.

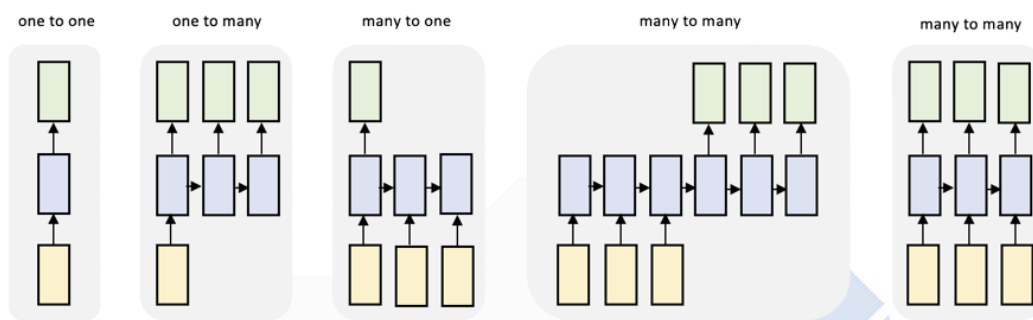
2. *Hidden Layer*

Merupakan sebuah lapisan yang tersembunyi, *hidden layer* sendiri berfungsi untuk meningkatkan kemampuan jaringan dalam memecahkan beberapa masalah.

### 3. *Output Layer*

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

Gambar 2.4 memeperlihatkan bahwa arsitektur RNN memproses data dari *input* secara sekuensial atau satu persatu, pada waktu berikutnya *hidden layer* akan mengirim data menuju *hidden layer* lainnya. Proses ini akan berjalan terus secara sekuensial. Berikut Gambar 2.5 yang menggambarkan beberapa proses pada RNN dalam melakukan pemrosesan.



Gambar 2. 5 Pemrosesan RNN

Berikut adalah penjelasan dari gambar 2.3 sebagai berikut :

#### 1. *One to One*

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

#### 2. *One to Many*

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

#### 3. *Many to One*

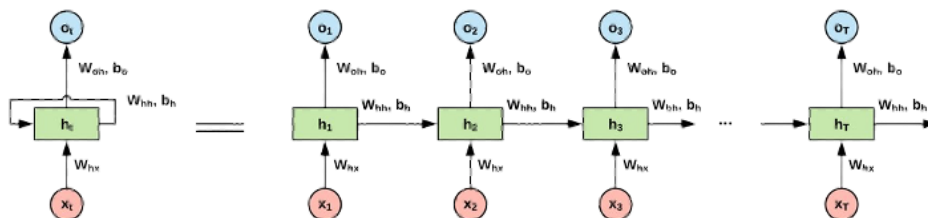
Berbeda dengan sebelumnya model ini menerima banyak data *input* tetapi hanya menghasilkan satu keluaran atau *output*.

#### 4. *Many to Many*

Model ini akan memproses data *input* dan *output* dengan banyak dan diproses secara berurutan atau sekuensial, model ini hanya akan

menghasilkan keluaran atau *output* ketika semua data pada *input* selesai diproses.

Pada proses melakukan *training* pada RNN hampir sama dengan proses *training* pada *neural network* lainnya, yaitu terdapat langkah-langkah utama dalam melakukan proses *training*. Pertama, proses untuk membuat prediksi atau sering disebut dengan proses *forward pass*. Proses ini dilakukan dengan melakukan perhitungan untuk setiap nilai *hidden state* ( $h_t$ ) yang didapatkan dari setiap masukkan pada ( $x_t$ ) dan nilai bobot yang telah ditentukan sebelumnya. Setelah menentukan nilai untuk *hidden state*, selanjutnya akan melakukan perhitungan untuk keluaran atau *output* dari hasil prediksi ( $z_t$ ). Kedua, melakukan perbandingan untuk hasil prediksi ( $z_t$ ) dengan nilai *output* yang telah ditargetkan dengan menggunakan *loss function* yang menunjukkan hasil dari nilai prediksi yang jauh dari *output* yang telah ditargetkan. *Loss Function* juga dapat memberikan kesimpulan seberapa baik atau buruknya kinerja yang dihasilkan oleh RNN. Ketiga, setelah berhasil mendapatkan nilai dari *loss function* maka, akan dilakukan proses *Backpropagation Through Time* (BPTT) yang bertujuan untuk menghitung nilai gradien untuk setiap waktu yang dibutuhkan dalam proses pada jaringan. Proses ini juga dilakukan untuk mencari nilai pada bobot-bobot dan juga bias yang dihasilkan lebih baik dari proses yang dilakukan sebelumnya. Setelah proses ini selesai, selanjutnya akan dilakukan pembaruan bobot dan juga bias dengan menggunakan metode *Stochastic Gradient Descent* (SGD). Berikut Gambar 2.4 yang menggambarkan struktur pada RNN.



Gambar 2. 6 Struktur RNN

Rumus-rumus untuk perhitungan yang terjadi dalam proses pelatihan RNN dapat dilihat pada rumus (2.1), (2.2), (2.3) dan (2.4), yaitu sebagai berikut:

1. Perhitungan nilai *hidden state* untuk waktu  $t$ , digunakan rumus (2.1) dengan sebuah fungsi aktivasi *tanh* yang memiliki rumus (2.2).

$$h_t = \tanh(W_{hh} \cdot [h_{t-1}, x_t] + W_{xh} \cdot x_t + b_h) \quad (2.1)$$

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (2.2)$$

Keterangan:

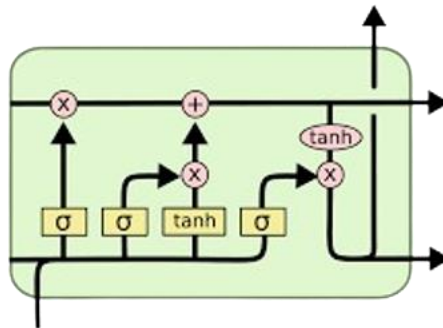
- $f_t$  = *Forget gate*
- $\sigma$  = Fungsi sigmoid
- $W_f$  = Nilai *weight* untuk *forget gate*
- $h_{t-1}$  = Nilai *output* sebelum orde ke  $t$
- $x_t$  = Nilai *input* pada orde ke  $t$
- $b_f$  = Nilai bias pada *forget gate*

2. Rumus pada persamaan (2.3) merupakan rumus untuk menghitung hasil dari prediksi, rumus ini berfungsi untuk menghitung nilai probabilitas dari setiap label dengan fungsi *softmax*. Fungsi ini memiliki kelebihan dengan keluaran nilai dengan probabilitas nilai 0 sampai 1, jika setiap hasil dari fungsi dijumlahkan akan bernilai 1.
3. Hasil dari prediksi atau *output* yang dihasilkan akan dibandingkan dengan nilai target atau nilai sebenarnya, untuk membandingkan hal tersebut menggunakan rumus dari persamaan (2.4) atau rumus *Cost Function*. Rumus yang digunakan pada penelitian ini yaitu *Mean Squared Error* (MSE). MSE merupakan fungsi yang digunakan untuk mengetahui besar dari nilai *error* yang dihasilkan dari mesin ketika telah melewati proses pelatihan.

### 2.1.5 Long Short Term Memory (LSTM)

*Long Short-Term Memory* (LSTM) merupakan salah satu dari beberapa arsitektur pada *Recurrent Neural Network* (RNN). Arsitektur RNN biasanya dirancang dengan khusus untuk melakukan pengolahan data secara sekuensial

dengan melakukan pembelajaran pada *internal state* melalui *layer-layer* dengan berurutan atau dengan sekuensial. Sehingga, RNN dapat melakukan pembelajaran informasi pada sebuah data *input* menggunakan informasi data *input* sekuensial sebelumnya.



Gambar 2. 7 Long Short Term Memory (LSTM)

RNN memiliki kelemahan yaitu pada kemampuan dalam mengingat, sehingga pada saat dihadapkan dengan data yang panjang, maka RNN akan kehilangan beberapa informasi pada data *input*. Hal ini disebabkan karena pada RNN terjadi *blow up gradient* atau *vanishing gradient* pada saat proses *backpropagation* yang mengakibatkan nilai *weight* pada sebuah data menjadi sangat besar ataupun sangat kecil, sehingga memperlambat proses *training* pada RNN [19]. Maka dari itu diperlukannya sebuah *state* yang dapat mengingat data, yang bernama *long short-term memory* atau lebih sering disebut dengan LSTM yang telah diilustrasikan pada Gambar 2.7.

LSTM telah menjadi solusi para RNN dengan menyediakan *cell state* sebagai *back-up* dari *hidden state* untuk melakukan pemrosesan memori secara sekuensial. Kedua *state* ini dapat menambah kemampuan RNN dalam mengingat. Pada langkah  $t$ , *Cell state*  $C_t$  akan berinteraksi dengan data dan *hidden state* melalui tahapan empat gerbang. Gerbang- gerbang tersebut adalah:

- a. Gerbang *forget*  $f_t$  menentukan informasi mana yang tidak lagi diperlukan di *cell state* yang lalu akan direduksi atau dibuang. Variabel pembelajaran  $W_f$



akan lakukan kalkulasi dengan hidden state  $h_{t-1}$  yang di gabungkan dengan data input  $x_t$ .  $b_f$  ditambahkan sebagai variabel pembelajaran bias. Hasil akhir perhitungan diproses dengan fungsi sigmoid  $\sigma$ . Perhitungan dapat dilihat pada persamaan (2.5).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.5)$$

Keterangan:

- $f_t$  = *Forget gate*
- $\sigma$  = Fungsi sigmoid
- $W_f$  = Nilai *weight* untuk *forget gate*
- $h_{t-1}$  = Nilai *output* sebelum orde ke t
- $x_t$  = Nilai *input* pada orde ke t
- $b_f$  = Nilai bias pada *forget gate*

- b. Gerbang *input*  $i_t$  menentukan informasi baru yang akan disimpan ke dalam *cell state* dengan perhitungan yang dapat dilihat pada persamaan (2.6) dan (2.7).  $W_i$ ,  $b_i$ ,  $W_c$ , dan  $b_c$  sebagai variabel pembelajaran.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.6)$$

Keterangan:

- $i_t$  = *Input gate*
- $\sigma$  = Fungsi sigmoid
- $W_i$  = Nilai *weight* untuk *input gate*
- $h_{t-1}$  = Nilai *output* sebelum orde ke t
- $x_t$  = Nilai *input* pada orde ke t
- $b_i$  = Nilai bias pada *input gate*

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.7)$$

Keterangan:

- $\tilde{C}_t$  = Nilai baru yang dapat ditambahkan ke *cell state*
- $\tanh$  = Fungsi *tanh*
- $W_c$  = Nilai *weight* untuk *cell state*
- $h_{t-1}$  = Nilai *output* sebelum orde ke t
- $x_t$  = Nilai *input* pada orde ke t
- $b_c$  = Nilai bias untuk *cell state*

- c. Gerbang *update* menyimpan informasi yang telah dipersiapkan sebelumnya ke dalam *cell state*. Perhitungan pada persamaan (2.8).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (2.8)$$

Keterangan:

- $C_t$  = *Cell state*
- $f_t$  = *Forget gate*
- $C_{t-1}$  = *Cell state* sebelum orde ke t
- $i_t$  = Nilai *input* pada orde ke t
- $\tilde{C}_t$  = Nilai baru yang dapat ditambahkan ke *cell state*

- d. Gerbang *output*  $o_t$  akan dimasukan bersama data *cell state* terbaru untuk dijadikan *hidden state* terbaru yang akan digunakan pada sekuens berikutnya. Perhitungan gerbang ini dapat dilihat pada persamaan (2.9) dan (2.10).  $W_o$  dan  $b_o$  adalah variabel pembelajaran.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.9)$$

Keterangan:

- $o_t$  = *Output gate*
- $\sigma$  = Fungsi sigmoid
- $W_o$  = Nilai *weight* untuk *output gate*

- $h_{t-1}$  = Nilai *output* sebelum orde ke t
- $x_t$  = Nilai *input* pada orde ke t
- $b_o$  = Nilai bias pada output gate

$$\tilde{C}_t = (W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.10)$$

Keterangan:

- $\tilde{C}_t$  = Nilai baru yang dapat ditambahkan ke *cell state*
- $W_c$  = Nilai *weight* untuk *cell state*
- $h_{t-1}$  = Nilai *output* sebelum orde ke t
- $x_t$  = Nilai *input* pada orde ke t
- $b_c$  = Nilai bias untuk *cell state*

### 2.1.6 Bilingual Evaluation Understudy (BLEU)

*Bilingual Evaluation Understudy* (BLEU) merupakan sebuah metode yang digunakan untuk melakukan evaluasi secara otomatis [8]. *BLEU score* memiliki cara kerja dengan membandingkan n-gram pada kalimat *candidate* yang dihasilkan mesin dengan kalimat referensi, unigram atau 1-gram akan dijadikan token dan bigram adalah sebuah pasangan kata dari sebuah kalimat [20]. *BLEU score* memiliki rentang nilai dari nilai 0 – 1, yang dimana nilai 1 akan didapatkan jika hasil terjemahan mendekati kalimat referensi, pada beberapa penelitian biasanya *BLEU score* dikalikan dengan 100 sehingga memiliki rentang nilai 0 – 100.

*BLEU Score* memiliki *precision* yang didapatkan dari perhitungan jumlah kata pada *candidate* yang memiliki kecocokan dengan *reference* dan dibagi dengan total jumlah kata pada unigram pada hasil *candidate*. Pada persamaan (2.11) *Brevity Penalty* pada *BLEU score* bertugas untuk melakukan perhitungan agar kalimat yang sedikit pada *candidate* tidak memiliki skor yang tinggi. Kemudian hasil akhir *BLEU Score* didapatkan dari hasil perkalian antara *brevity penalty* dengan hasil rata-rata geometrik *modified precision score* seperti pada persamaan (2.12).

Rumus menghitung skor BLEU adalah sebagai berikut:

$$BP = \begin{cases} 1, & \text{if } c > r \\ \exp^{(1-\frac{r}{c})}, & \text{if } c \leq r \end{cases} \quad (2.11)$$

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^n w_n \log p_n\right) \quad (2.12)$$

Keterangan:

$BP$  = *Brevity Penalty*

$c$  = Jumlah kata dari hasil terjemahan otomatis (*candidate*)

$r$  = Jumlah kata rujukan (*reference*)

$w_n$  =  $1/N$  (standar nilai  $N$  untuk BLEU adalah 4)

$p_n$  = Jumlah  $n$ -gram hasil terjemahan yang sesuai dengan rujukan dibagi jumlah  $n$ -gram hasil terjemahan.

### 2.1.7 Tensorflow

Tensorflow merupakan *framework* yang bersifat *open source*, biasanya tensorflow digunakan untuk melatih, mengembangkan, dan menggunakan model deteksi objek yang telah banyak digunakan pada beberapa produk *google* seperti *google street view*, *google assistant*, dan lain-lain.

Tensorflow bekerja secara *computational* dalam membuat model *machine learning* ataupun model *deep learning*. Tensorflow juga menyediakan beberapa *toolkit* yang berguna untuk membuat model dengan menentukan serangkaian matematis. Penggunaan tensorflow juga dapat menggunakan API, seperti (tf.estimator) untuk menentukan yang telah ditetapkan, seperti regresi linear ataupun *neural network*.

Tensorflow juga biasa digunakan pada proses pembuatan sistem deteksi objek, sehingga memudahkan implementasi pada algoritma dan penggunaan bahasa pemrograman, juga terdapat GPU yang berguna untuk mempercepat proses

*training*. Tensorflow juga digunakan untuk mengolah data atau untuk membuat kecerdasan buatan (*artificial Intellegence*). Arsitekturnya yang fleksibel memungkinkan penyebaran komputasi dengan mudah di berbagai *platform* (CPU, GPU, TPU), dan dari *desktop* ke *cluster server* hingga perangkat seluler [21].

Tensorflow dikembangkan oleh para peneliti ahli dari tim *google brain* pada divisi AI *Google*, tensorflow memiliki perhitungan numerik yang fleksibel dan digunakan pada banyak domain ilmiah lainnya.

### **2.1.8 Flask**

Flask adalah *framework web* berbasis Python yang termasuk dalam kategori *microframework*. Flask adalah kerangka aplikasi dan tampilan berbasis *web*. Pengembang dapat lebih mudah mengelola perilaku *web* dan membuat *web* terstruktur dengan memanfaatkan Flask dan bahasa pemrograman Python. Flask adalah *microframework* karena tidak memerlukan alat atau pustaka khusus untuk bekerja. Sebagian besar fungsi dan komponen umum, seperti validasi *form* dan *database*, tidak ter-*install* secara *default* di Flask. Ini karena Flask dapat menggunakan ekstensi untuk membuat fitur dan komponen ini tampak diimplementasikan oleh Flask sendiri karena disediakan oleh pihak ketiga. Selain itu, hanya karena Flask disebut sebagai *microframework* tidak berarti Flask tidak memiliki fungsionalitas. *Microframework* ini mengacu untuk membuat inti aplikasi berpotensi sambil tetap membuat instalasi *add-on* menjadi sederhana. Dengan cara ini, Flask mengungguli *framework* lain dalam hal skalabilitas dan fleksibilitas [22].

### **2.2 Penelitian Terkait**

Penelitian terkait merupakan penelitian yang telah dilakukan sebelumnya dan terkait dengan topik yang dibahas. Pada Tabel 2.1 menampilkan penelitian-penelitian terdahulu yang sesuai dengan penelitian ini. Penelitian sejenis digunakan oleh penulis sebagai perbandingan dan acuan dalam melakukan penelitian ini.

Tabel 2. 1 Penelitian Terkait

No	Judul	Hasil
1	Pengembangan Model Mesin Penerjemah Inggris Indonesia Menggunakan <i>Reccurent Neural Network LSTM</i> dan <i>Attention</i> [23].	Hasil evaluasi terhadap nilai akurasi model bahasa menunjukkan bahwa dengan menggunakan model LSTM dan <i>attention</i> dapat menghasilkan hasil akurasi yang lebih baik. Model LSTM dan <i>attention</i> memiliki skor BLEU sebesar 68,04, yang lebih baik dari model LSTM tanpa <i>attention</i> yang memiliki skor BLUE sebesar 60,08.
2	Penerapan <i>Neural Machine Translation</i> untuk Eksperimen Penerjemahan secara Otomatis pada Bahasa Lampung – Indonesia [24].	Penerjemahan bahasa Lampung ke bahasa Indonesia secara otomatis dengan menggunakan <i>Neural Machine Translation</i> memberikan hasil penerjemahan bahasa Lampung-Indonesia pada 25 kalimat tunggal tanpa OOV diperoleh nilai <i>Bilingual Evaluation Under Study</i> (BLEU) sebesar 41.79 dan 25 kalimat majemuk tanpa OOV diperoleh nilai <i>Bilingual Evaluation Under Study</i> (BLEU) sebesar 37.5.

- 
- 3 Mesin Penterjemah Bahasa Indonesia – Bahasa Sunda Menggunakan *Recurrent Neural Networks* [25]. Hasil yang didapatkan pada penelitian ini pada pengujian pertama arsitektur model variasi dari RNN didapatkan nilai optimal oleh GRU akurasi sebesar 99.17%. Pengujian kedua menghasilkan nilai optimal oleh model dengan menggunakan *Attention* dengan nilai akurasi 99.94%. Pengujian ketiga dalam perbandingan model optimasi dimana Adam mendapat hasil optimal dengan nilai akurasi 99.35%. Pengujian terakhir yaitu *BLEU Score* dimana menghasilkan nilai optimal bleu 92.63% dengan *brevity penalty* 0.929.
- 
- 4 Perancangan Mesin Translasi berbasis *Neural* dari Bahasa Kawi ke dalam Bahasa Indonesia menggunakan *Microframework Flask* [26]. MT yang dibuat dengan model RNN ini masih jauh dari berhasil melihat *BLEU scores* yang masih dibawah rata-rata nilai translasi yang konteksnya dapat dipahami, seperti pada translasi dari ahli manusia maupun mesin translasi seperti *google translate* dengan *BLEU score* di 60.
- 
- 5 Model Bahasa *Recurrent Neural Network* untuk Mesin Penterjemah Inggris - Indonesia [27]. Hasil evaluasi terhadap nilai *perplexity* kedua model bahasa menunjukkan bahwa dengan menggunakan model *Recurrent Neural Network* dapat menghasilkan hasil yang lebih baik. Untuk rata-rata penurunan nilai *perplexity* sebesar 10,6 persen untuk keseluruhan *order* pada korpus Bahasa
-

---

Inggris dan 4 persen pada korpus Bahasa Indonesia. Sementara itu, nilai BLEU dan nilai RIBES mengalami peningkatan 0,3 dan 0,79 lebih tinggi daripada hasil yang dihasilkan dengan menggunakan model berbasis statistik.

---

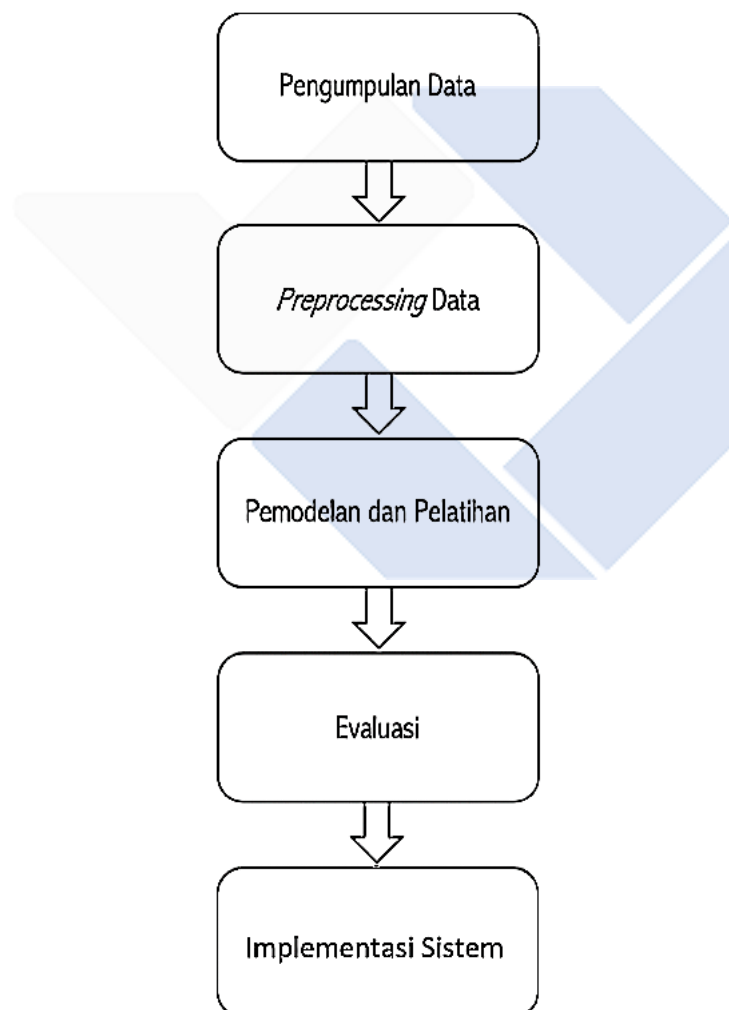




## BAB III METODE PELAKSANAAN

### 3.1 Alur Penelitian

Pada penelitian ini terdapat beberapa tahapan yang terdiri dari pengumpulan data, *preprocessing* data, pemodelan dan pelatihan, evaluasi, pengujian dan implementasi sistem. Gambar 3.1 berikut yang mengilustrasikan dari alur pada penelitian ini.



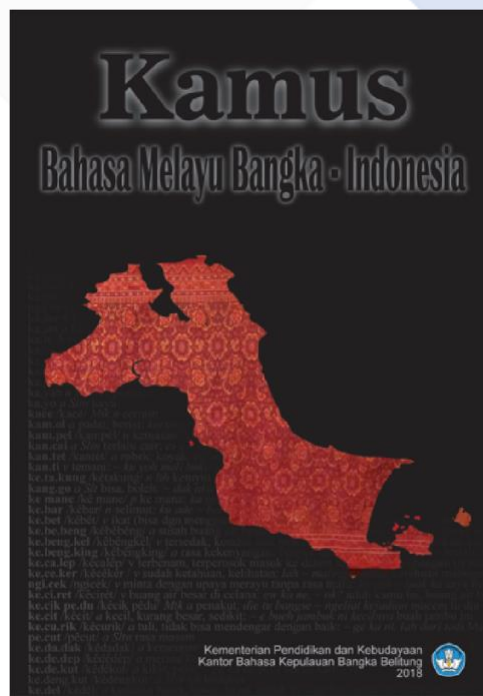
Gambar 3. 1 Alur Penelitian

## 3.2 Uraian Penelitian

Berikut merupakan uraian yang menjelaskan alur penelitian tentang tahapan pada penelitian ini.

### 3.2.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini memiliki banyak sumber dalam pengambilan data untuk data korpus paralel bahasa Inggris – bahasa Indonesia didapatkan dari Tatoeba, untuk data korpus paralel bahasa Indonesia – bahasa Bangka bersumber dari buku dengan judul “Kamus Bahasa Melayu Bangka – Indonesia” yang diterbitkan Kementerian Pendidikan dan Kebudayaan pada tahun 2018 dan untuk data korpus paralel bahasa Inggris – bahasa Bangka didapatkan dari terjemahan pada bahasa Indonesia pada korpus paralel bahasa Indonesia – Bangka. Berikut tampilan buku Kamus Bahasa Melayu Bangka – Indonesia Gambar 3.2.



Gambar 3. 2 Buku "Kamus Bahasa Melayu Bangka - Indonesia

### 3.2.2 Preprocessing Data

Proses ini bertujuan untuk membersihkan data dari berbagai komponen atau fitur yang tidak diinginkan sehingga memaksimalkan pembelajaran mesin. Berdasarkan target keluaran alur *preprocessing* disesuaikan oleh peneliti. *preprocessing* data dilakukan untuk mempersiapkan data untuk digunakan selama proses pembelajaran. Kumpulan data yang digunakan untuk pemilihan data telah diproses sebelumnya agar lebih terstruktur dengan menghapus tanda baca, angka, dan tahapan teks *preprocessing* lainnya.

### 3.2.3 Pemodelan dan Pelatihan

Pemodelan dilakukan dengan menentukan arsitektur dari varian model yang akan dipakai. Untuk penelitian ini telah ditentukan enam model yang akan dikembangkan dan dievaluasi. Ke-enam model ini menggunakan RNN dan LSTM pada bagian *decoder*. Untuk komponen pemrograman dalam pengembangan ini adalah Bahasa python versi 3.10 dan *library* Tensorflow. *Hyperparameter* setiap model berupa jumlah unit LSTM 50. Nilai batch size berupa 25. Jenis *optimizer* *Root-Mean-Square Propagation* (RMSprop). Fungsi perhitungan *Loss* menggunakan *sparse categorical entropy*. Strategi *early stopping* juga diterapkan dengan *patience* 10. Penerapan *early stopping* tersebut akan memberi kesempatan model untuk terus melakukan proses latihan sebanyak 10 kali secara beruntun tidak menghasilkan progres *loss* melewati batas nilai nol.

### 3.2.4 Evaluasi

Evaluasi pada penelitian bertujuan untuk mengukur kinerja dari model yang telah dibuat. Hasil dari terjemahan akan dinilai dengan skor BLEU. Skor BLEU yang semakin tinggi mengindikasikan bahwa terjemahan yang telah dibuat semakin menyamai referensi. Pada tahap evaluasi ini akan dibandingkan hasil terjemahan yang dihasilkan sistem dengan terjemahan dari korpus paralel yang telah dibuat sebelumnya.

### 3.2.5 Implementasi Sistem

Pada tahapan ini akan mengimplementasikan model ke dalam aplikasi berbasis *website* dengan menggunakan *microframework* Flask. Perancangan *user interface* juga diterapkan agar pengerjaan aplikasi lebih terarah.



## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil Pengumpulan Data

Pada penelitian ini menggunakan data berupa korpus paralel terdapat beberapa korpus paralel yang digunakan yaitu korpus paralel Bahasa Bangka – Bahasa Indonesia, Bahasa Bangka – Bahasa Inggris dan Bahasa Indonesia – Bahasa Inggris. Untuk korpus paralel Bahasa Bangka – Bahasa Indonesia berjumlah 3.080 data yang bersumber dari buku “Kamus Bahasa Melayu Bangka – Indonesia” yang diterbitkan Kementerian Pendidikan dan Kebudayaan pada tahun 2018 dan terdapat beberapa data yang di tambahkan oleh peneliti. Pada Tabel 4.1 merupakan contoh korpus paralel Bahasa Bangka – Bahasa Indonesia.

Tabel 4. 1 Korpus Paralel Bahasa Bangka - Bahasa Indonesia

<b>Korpus Paralel</b>	<b>Bahasa Bangka</b>	<b>Bahasa Indonesia</b>
	Ka nek minum ape?	Kamu mau minum apa?
	Ku yang pertama	Aku yang pertama
	Yo kita cube!	Mari kita coba!
	Ka dak ape-ape?	Kamu tidak apa-apa?
	Jangan berbicara!	Jangan ngomong!
<b>Total Dataset</b>	<b>3080</b>	
<b>Sumber</b>	<b>Kamus Bahasa Melayu Bangka – Indonesia</b>	

Korpus paralel Bahasa Bangka – Bahasa Inggris dengan jumlah 3.080 data yang didapatkan dari hasil dari korpus sebelumnya dengan menerjemahkan bahasa indonesia menjadi bahasa inggris terlebih dahulu Tabel 4.2.

Tabel 4. 2 Korpus Paralel Bahasa Bangka - Bahasa Inggris

<b>Korpus Paralel</b>	<b>Bahasa Bangka</b>	<b>Bahasa Indonesia</b>
	Ka nek minum ape?	Kamu mau minum apa?
	Ku yang pertama	Aku yang pertama
	Yo kita cube!	Mari kita coba!
	Ka dak ape-ape?	Kamu tidak apa-apa?
	Jangan berbicara!	Jangan ngomong!
<b>Total Dataset</b>	<b>3080</b>	
<b>Sumber</b>	<b>Kamus Bahasa Melayu Bangka – Indonesia</b>	

Lalu untuk korpus paralel Bahasa Indonesia – Bahasa Inggris di dapatkan dari Tatoeba. Berikut contoh data korpus paralel Tabel 4.3.

Tabel 4. 3 Korpus Paralel Bahasa Indonesia - Bahasa Inggris

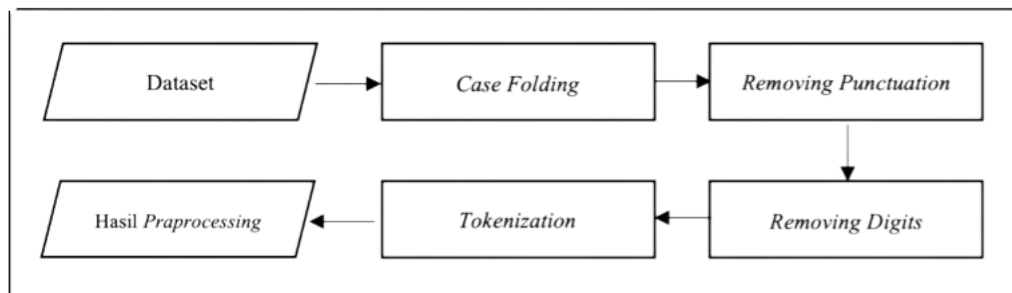
<b>Korpus Paralel</b>	<b>Bahasa Indonesia</b>	<b>Bahasa Inggris</b>
	Buku itu punya saya	That book is mine
	Tas itu milik siapa?	Whose bag is that?
	Apakah di luar dingin?	Is it cold outside?
	Saya sangat lelah	I feel very tired
	Waktu adalah uang	Time is money
<b>Total Dataset</b>	<b>44030</b>	
<b>Sumber</b>	<b>Tatoeba</b>	
<b>URL</b>	<b><a href="https://tatoeba.org/en/downloads">https://tatoeba.org/en/downloads</a></b>	

*Dataset* dibentuk menggunakan format .txt, yaitu sebuah file dokumen teks standar yang berisi teks yang tidak diformat. *Dataset* ini nantinya akan di proses dan disimpan kedalam file dengan format .csv untuk mendapatkan kumpulan data yang lebih terstruktur.

#### 4.2 Preprocessing Data

Tahapan *preprocessing* terhadap data yaitu proses untuk menjadikan data yang sebelumnya tidak terstruktur menjadi struktur. Melalui proses *preprocessing*

data mentah yang tersedia dalam *dataset* diolah dan di standarkan sehingga dapat diproses oleh algoritma yang akan digunakan. Sebelum data dapat diproses data harus dapat melalui beberapa tahapan seperti, *case folding*, *removing punctuation*, *removing digits* dan *tokenization*. Gambar 4.1 merupakan *flowchart* pada tahapan *preprocessing* data.



Gambar 4. 1 *Flowchart Preprocessing Data*

#### 4.2.1 *Case Folding*

Dalam penelitian ini, standarnya adalah menggunakan huruf kecil saat memproses kasus *dataset* untuk proses pelatihan. Hal ini dilakukan untuk membakukan data dengan kata yang sama tetapi kasus yang berbeda. Karena *dataset* berisi huruf besar dan huruf kecil, langkah ini dilakukan. Misalnya, frasa "Saya", "SaYa", atau "Saya" dapat ditemukan dalam sebuah *dataset*. "Aku" memiliki arti yang sama bagi semua orang. Komputer, di sisi lain, mengartikan beberapa dari kata-kata ini sebagai kata yang berbeda meskipun memiliki arti yang sama karena kasus karakter yang berbeda. Menggunakan satu standar atau, seperti dalam penelitian ini, huruf kecil dapat membakukan kata yang sama dengan kasus yang berbeda menggunakan fungsi pelipatan kasus. Sehingga untuk menghindari hal tersebut pada penelitian ini menggunakan satu standar yaitu dengan menyamakan semua data dengan menjadikan data menjadi huruf kecil. Berikut *code* untuk melakukan *case folding*.

```
source_text_ = [x.lower() for x in source_text]
target_text_ = [x.lower() for x in target_text]
```

Setelah *code* sudah dijalankan maka akan menghasilkan data yang sesuai standarisasi yaitu *lowercase* atau huruf kecil. Berikut contoh hasil dari tahapan *case folding* pada Tabel 4.4.

Tabel 4. 4 Hasil *Case Folding* Korpus Paralel Bangka - Indonesia

<b>Sebelum</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Indonesia</b>
Ka nek minum ape?	Kamu mau minum apa?
Ku yang pertama	Aku yang pertama
Yo kita cube!	Mari kita coba!
Ka dak ape-ape?	Kamu tidak apa-apa?
Jangan berbicara!	Jangan ngomong!
<b>Sesudah</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Indonesia</b>
ka nek minum ape?	kamu mau minum apa?
ku yang pertama	aku yang pertama
yo kita cube!	mari kita coba!
ka dak ape-ape?	kamu tidak apa-apa?
jangnan berbicara!	jangnan ngomong!

Selanjutnya pada Tabel 4.5 merupakan contoh dari hasil *case folding* pada Bahasa Bangka – Bahasa Inggris.

Tabel 4. 5 Hasil *Case Folding* Korpus Paralel Bangka - Inggris

<b>Sebelum</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Inggris</b>
Kami kenal die	We know him
Siapa yang ngambar tu?	Who drew it?
Ape ka yakin?	Are you sure?
Ku penasaran	I am curious
Ku dak pacak bace	I can't read
<b>Sesudah</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Inggris</b>



kami kenal die	we know him
siapa yang ngambar tu?	who drew it?
ape ka yakin?	are you sure?
ku penasaran	i am curious
ku dak pacak bace	i can't read

Selanjutnya pada Tabel 4.6 merupakan contoh dari hasil *case folding* pada Bahasa Indonesia – Bahasa Inggris.

Tabel 4. 6 Hasil *Case Folding* Korpus Paralel Indonesia - Inggris

<b>Sebelum</b>	
<b>Bahasa Indonesia</b>	<b>Bahasa Inggris</b>
Buku itu punya saya	That book is mine
Tas itu milik siapa?	Whose bag is that?
Apakah di luar dingin?	Is it cold outside?
Saya sangat lelah	I feel very tired
Waktu adalah uang	Time is money
<b>Sesudah</b>	
<b>Bahasa Indonesia</b>	<b>Bahasa Inggris</b>
buku itu punya saya	that book is mine
tas itu milik siapa?	whose bag is that?
apakah di luar dingin?	is it cold outside?
saya sangat lelah	i feel very tired
waktu adalah uang	time is money

#### 4.2.2 *Removing Punctuation*

Tujuan dari penghapusan tanda baca atau tanda baca adalah untuk menghilangkan karakter yang tidak akan mempengaruhi bagaimana terjemahan dilakukan. Akibatnya, proses penyaringan dilakukan pada titik ini untuk mencari tanda baca yang tidak berdampak nyata pada kalimat dan tanda baca yang dapat merusak makna jika ditempatkan secara tidak tepat. Tanda baca berikut disaring oleh peneliti dalam penelitian ini: ",&%? +-.". Kode perintah untuk menghilangkan tanda baca adalah sebagai berikut.

```
def remove_punc(text_list):
    table = str.maketrans('', '', string.punctuation)
    removed_punc_text = []
```

Setelah *code* sudah dijalankan maka akan menghasilkan data yang tidak terdapat tanda baca. Berikut contoh hasil dari tahapan *removing punctuation* pada Tabel 4.7.

Tabel 4. 7 Hasil *Removing Punctuation* Korpus Paralel Bangka - Indonesia

<b>Sebelum</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Indonesia</b>
ka nek minum ape?	kamu mau minum apa?
ku yang pertama	aku yang pertama
yo kita cube!	mari kita coba!
ka dak ape-ape?	kamu tidak apa-apa?
jangan berbicara!	jangan ngomong!
<b>Sesudah</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Indonesia</b>
ka nek minum ape	kamu mau minum apa
ku yang pertama	aku yang pertama
yo kita cube	mari kita coba
ka dak apeape	kamu tidak apaapa
jangan berbicara	jangan ngomong

Berikut contoh hasil dari tahapan *removing punctuation* untuk Bahasa Bangka – Bahasa Inggris pada Tabel 4.8.

Tabel 4. 8 Hasil *Removing Punctuation* Korpus Paralel Bangka - Inggris

<b>Sebelum</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Inggris</b>
kami kenal die	we know him
siapa yang ngambar tu?	who drew it?
ape ka yakin?	are you sure?

ku penasaran	i am curious
ku dak pacak bace	i can't read
<b>Sesudah</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Inggris</b>
kami kenal die	we know him
siape yang ngambar tu	who drew it
ape ka yakin	are you sure
ku penasaran	i am curious
ku dak pacak bace	i cant read

Berikut contoh hasil dari tahapan *removing punctuation* untunk Bahasa Bangka – Bahasa Inggris pada Tabel 4.9.

Tabel 4. 9 Hasil *Removing Punctuation* Korpus Paralel Indonesia - Inggris

<b>Sebelum</b>	
<b>Bahasa Indonesia</b>	<b>Bahasa Inggris</b>
buku itu punya saya	that book is mine
tas itu milik siapa?	whose bag is that?
apakah di luar dingin?	is it cold outside?
saya sangat lelah	i feel very tired
waktu adalah uang	time is money
<b>Sesudah</b>	
<b>Bahasa Indonesia</b>	<b>Bahasa Inggris</b>
buku itu punya saya	that book is mine
tas itu milik siapa	whose bag is that
apakah di luar dingin	is it cold outside
saya sangat lelah	i feel very tired
waktu adalah uang	time is money

#### 4.2.3 *Removing Digits*

Proses *removing digits* atau menghapus angka memiliki tujuan hampir sama dengan proses menghapus *punctuation*, namun proses menghilangkan angka

lebih pada menghapus angka atau digit. Berikut adalah perintah *code* untuk melakukan *removing digits*.

```
remove_digits = str.maketrans('', '', digits)
removed_digits_text = []
```

Setelah *code* sudah dijalankan maka akan menghasilkan data yang tidak terdapat digit atau angka. Berikut contoh hasil dari tahapan *removing digits* pada Tabel 4.10.

Tabel 4. 10 Hasil *Removing Digits* Korpus Paralel Bangka - Indonesia

<b>Sebelum</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Indonesia</b>
ka nek minum ape?	kamu mau minum apa?
ku yang pertama	aku yang pertama
yo kita cube!	mari kita coba!
ka dak ape-ape?	kamu tidak apa-apa?
jangan berbicara!	jangan ngomong!
<b>Sesudah</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Indonesia</b>
ka nek minum ape	kamu mau minum apa
ku yang pertama	aku yang pertama
yo kita cube	mari kita coba
ka dak apeape	kamu tidak apaapa
jangan berbicara	jangan ngomong

Berikut contoh hasil dari tahapan *removing digits* untuk Bahasa Bangka – Bahasa Inggris pada Tabel 4.11.

Tabel 4. 11 Hasil *Removing Digits* Korpus Paralel Bangka - Inggris

<b>Sebelum</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Inggris</b>
kami kenal die	we know him
siapa yang ngambar tu?	who drew it?
ape ka yakin?	are you sure?
ku penasaran	i am curious
ku dak pacak bace	i can't read
<b>Sesudah</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Inggris</b>
kami kenal die	we know him
siapa yang ngambar tu	who drew it
ape ka yakin	are you sure
ku penasaran	i am curious
ku dak pacak bace	i cant read

Berikut contoh hasil dari tahapan *removing digits* untuk Bahasa Indonesia – Bahasa Inggris pada Tabel 4.12.

Tabel 4. 12 Hasil *Removing Digits* Korpus Paralel Indonesia - Inggris

<b>Sebelum</b>	
<b>Bahasa Indonesia</b>	<b>Bahasa Inggris</b>
buku itu punya saya	that book is mine
tas itu milik siapa?	whose bag is that?
apakah di luar dingin?	is it cold outside?
saya sangat lelah	i feel very tired
waktu adalah uang	time is money
<b>Sesudah</b>	
<b>Bahasa Indonesia</b>	<b>Bahasa Inggris</b>
buku itu punya saya	that book is mine
tas itu milik siapa	whose bag is that
apakah di luar dingin	is it cold outside
saya sangat lelah	i feel very tired
waktu adalah uang	time is money

#### 4.2.4 Tokenization

Proses *tokenization* adalah operasi memisahkan teks menjadi potongan-potongan berupa token, bisa berupa potongan huruf, kata, atau kalimat, sebelum dianalisis lebih lanjut. Entitas yang bisa disebut sebagai token misalnya kata, angka, simbol, tanda baca, dan lain sebagainya. Dalam proses *tokenizer* menggunakan *word tokenizer* yaitu untuk memecah kalimat dalam bentuk kata per kata. Berikut adalah perintah *code* untuk melakukan *tokenizer*.

```
def tokenizer_(text_data):
    tokenizer = Tokenizer ()
    tokenizer.fit_on_texts(text_data)
    return tokenizer
```

Setelah *code* sudah dijalankan maka akan menghasilkan data akan berubah menjadi token-token. Berikut contoh hasil dari tahapan *tokenization* pada Tabel 4.13.

Tabel 4. 13 Hasil *Tokenization* Korpus Paralel Bangka - Indonesia

Sebelum	
Bahasa Bangka	Bahasa Indonesia
jangan ngolokku	jangan mengejekku
dakde yang ngeliet kite	tidak ada yang melihat kita
ku leteh bener	aku sangat capek
ape tu tembikar	apa itu tembikar
die orang yang tepat waktu	dia orang yang tepat waktu
Sesudah	
Bahasa Bangka	Bahasa Indonesia
[24, 567]	[20, 632]
[195, 5, 33, 12]	[10, 21, 6, 48, 11]
[1, 95, 18]	[4, 38, 132]
[3, 6, 376]	[3, 8, 811]
[10, 59, 5, 265, 73]	[14, 71, 6, 148, 67]

Berikut contoh hasil dari tahapan *tokenization* untuk Bahasa Bangka – Bahasa Inggris pada Tabel 4.14.

Tabel 4. 14 Hasil *Tokenization* Korpus Paralel Bangka - Inggris

<b>Sebelum</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Inggris</b>
buku tu warna merah	the book is red
di mane dorang	where were they
ku harus bantu	i have to help
kami berangkat	were leaving
ape tu tembikar	what is pottery
<b>Sesudah</b>	
<b>Bahasa Bangka</b>	<b>Bahasa Inggris</b>
[1, 200]	[8, 346]
[11, 26, 25]	[40, 32, 55]
[11, 26, 25]	[40, 32, 55]
[14, 527]	[32, 232]
[5, 4, 344]	[46, 4, 384]

Berikut contoh hasil dari tahapan *tokenization* untunk Bahasa Indonesia – Bahasa Inggris pada Tabel 4.15.

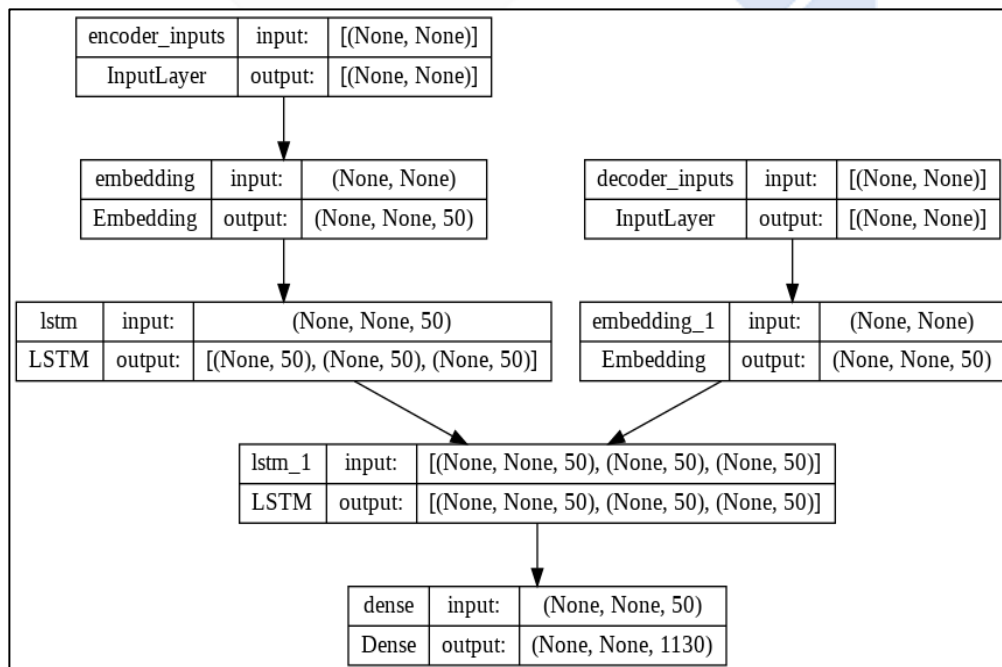
Tabel 4. 15 Hasil *Tokenization* Korpus Paralel Indonesia - Inggris

<b>Sebelum</b>	
<b>Bahasa Indonesia</b>	<b>Bahasa Inggris</b>
dia bilang dia tidak tersinggung	he said that he wasnt offended
aku tidak bercanda	im not joking
aku harus pergi ke kamar kecil	i have to use the restroom
itu adalah satu hal yang belum aku pikirkan	thats the one thing i hadnt thought of
sedang turun hujan	its raining
<b>Sesudah</b>	
<b>Bahasa Indonesia</b>	<b>Bahasa Inggris</b>
[14, 494, 14, 4, 4043]	[4, 86, 13, 14, 315, 4261]

[5, 4, 640]	[48, 30, 1269]
[5, 42, 37, 21, 397, 175]	[8, 20, 4, 215, 3, 3255]
[12, 24, 74, 52, 1, 149, 5, 1154]	[155, 3, 54, 411, 8, 1810, 250, 5]
[54, 435, 344]	[53, 1042]

### 4.3 Pemodelan dan Pelatihan

Dalam pembuatan model atau pemodelan terdiri atas beberapa proses, untuk model penerjemah bahasa membutuhkan input korpus paralel, sebagai sumber data pembelajaran. Pada data korpus terjadi proses *preprocessing* data untuk menghasilkan hasil terjemahan dengan kinerja yang baik dan menghindari *overfitting*. Tahapan selanjutnya adalah perancangan model, pada penelitian ini menggunakan model RNN. Setelah tahap perancangan model, kemudian dilakukan *training* atau pelatihan untuk menghasilkan model yang dapat menerjemahkan bahasa. Berikut merupakan arsitektur model yang digunakan pada penelitian ini Gambar 4.2.



Gambar 4. 2 Arsitektur Model RNN



### 4.3.1 Pembuatan Model

Model akan di program terdiri dari beberapa bagian yang memiliki fleksibilitas untuk menyesuaikan keperluan tiap varian model. Eksekusi program akan berfokus pada satu modul yang menampung pilihan tiap *hyperparameter* dan metode yang diinginkan. Keseluruhan model akan otomatis menyesuaikan. program ini akan memanfaatkan metode *subclasssing* tensorflow dalam pengembangan program. Berikut tiap kelas untuk *layer* pada model.

#### a. Encoder Layer

```
encoder_inputs = Input(shape=(None, ),
name="encoder_inputs")
emb_layer_encoder = Embedding(vocab_size_source,
latent_dim, mask_zero=True)(encoder_inputs)
encoder = LSTM(latent_dim, return_state=True)
encoder_outputs, state_h, state_c =
encoder(emb_layer_encoder)
encoder_states = [state_h, state_c]
decoder_inputs =
Input(shape=(None, ), name="decoder_inputs")
```

#### b. Decoder Layer

```
emb_layer_decoder =
Embedding(vocab_size_target, latent_dim,
mask_zero=True)(decoder_inputs)
decoder_lstm = LSTM(latent_dim, return_sequences=True,
, return_state=True)
decoder_outputs, _, _ = decoder_lstm(emb_layer_decoder,
initial_state=encoder_states)
decoder_dense = Dense(vocab_size_target,
, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)
```

### 4.3.2 Optimasi

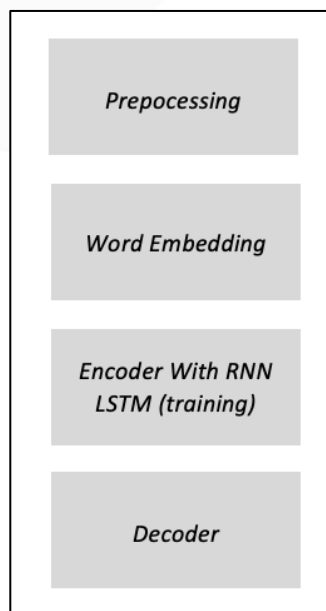
Proses pelatihan memerlukan beberapa pengaturan, penggunaan *loss function categorical crossentropy* karena data pelatihan berbentuk kategori, sehingga pengevaluasian *loss* dilakukan berdasarkan kategori data. Penelitian ini

menggunakan optimasi *Root-Mean-Square Propagation* (RMSprop), penggunaan optimasi RMSprop dipilih karena optimasi ini lebih efisien dan efektif serta tidak menggunakan banyak sumber daya. Metrik yang digunakan dalam penelitian ini adalah akurasi.

```
model.compile(loss='categorical_crossentropy',  
optimizer='rmsprop',  
metrics=['accuracy'])
```

### 4.3.3 Pelatihan Model

Pada penelitian ini menggunakan arsitektur model RNN. Pada beberapa kasus, RNN menghasilkan *output* yang sama panjang dengan *input*-nya. Sehingga, mekanisme *encoder-decoder* digunakan karena panjang ringkasan *output* harus berbeda dengan *input*. Model yang digunakan dalam penelitian ini digambarkan pada Gambar 4.3.



Gambar 4. 3 Struktur Model pada Pelatihan Model

Untuk mengatur jalannya proses pelatihan diperlukan pengaturan pelatihan. Pengaturan pelatihan bertujuan untuk mendapatkan model terbaik secara efisien dan efektif.

```
train_samples = len(X_train)
val_samples = len(X_test)
batch_size = 25
epochs = 100
```

Tujuan `train_samples` untuk menentukan jumlah data training dan `val_samples` untuk menentukan jumlah data untuk validasi. Pada proses pelatihan terhadap *dataset* terjadi pengulangan (*epoch*) beberapa kali untuk mendapatkan kinerja model yang paling maksimal.

```
model.fit_generator(generator =
generator_batch(X_train, y_train,
batch_size = batch_size),
steps_per_epoch = train_samples,
epochs=epochs)
```

Proses terakhir adalah melakukan pengaturan proses jalannya pelatihan. *Fit generator* dipilih untuk dapat melakukan pelatihan secara paralel, sehingga proses pelatihan menjadi lebih efisien. Penggunaan *fit generator* memungkinkan augmentasi data pada CPU dan training pada GPU dapat dilaksanakan paralel secara *real-time*.

## 4.4 Evaluasi

### 4.4.1 Aktifitas Grafik *Loss*

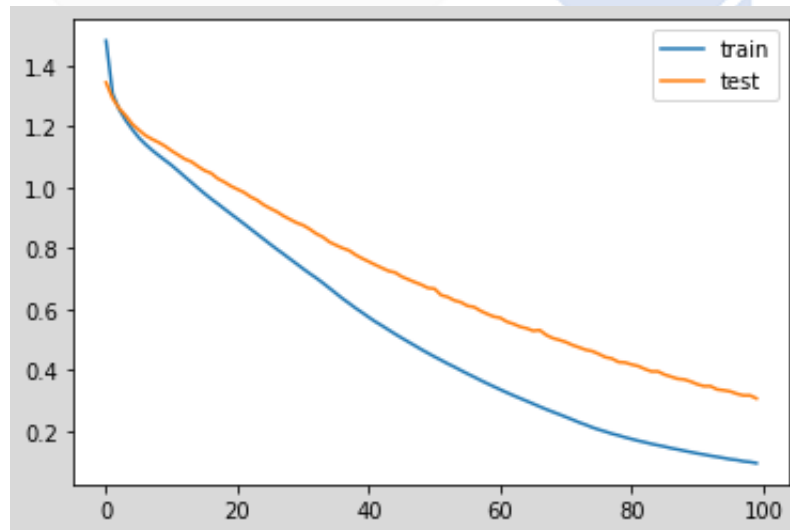
Pada saat proses *training* selesai, didapatkan nilai *loss* dari masing-masing model yang telah dibuat. Nilai tersebut ditampilkan pada Tabel 4.1. Berdasarkan tabel tersebut Model BN – ID dan Model EN - ID memiliki nilai *training loss* paling kecil yaitu sebesar 0.4, lebih kecil daripada model – model lainnya yang memiliki nilai *training loss* diatas kedua model tersebut. *Validation loss* Model BN – ID dan Model ID – BN memiliki nilai yang cenderung lebih kecil daripada model lainnya yaitu sebesar 0.1. Grafik nilai *loss* dan *validation loss* dapat dilihat pada Tabel 4.16.

Tabel 4. 16 Hasil Aktifitas Grafik *Loss*

	Model BN-ID	Model BN-EN	Model ID-BN	Model ID-EN	Model EN-BN	Model EN-ID
<i>Training Loss</i>	0.4	3.9	0.5	0.4	1.4	0.5
<i>Validaton Loss</i>	0.1	2.6	0.1	0.2	0.5	0.2

### 1. Model Penerjemah Bangka – Indonesia

Hasil yang didapatkan untuk proses pelatihan terhadap *dataset* Bahasa Bangka – Bahasa Indonesia menghasilkan nilai *loss* pada validasi yang fluktuatif, dengan nilai *loss* terendah 0.1 pada nilai *loss training* hingga 0.4 pada nilai *loss* validasi. Semakin kecil nilai *loss* maka semakin baik kinerja model yang didapatkan. Berikut grafik nilai *loss* dan *validation loss* model penerjemah Bangka – Indonesia Gambar 4.4.

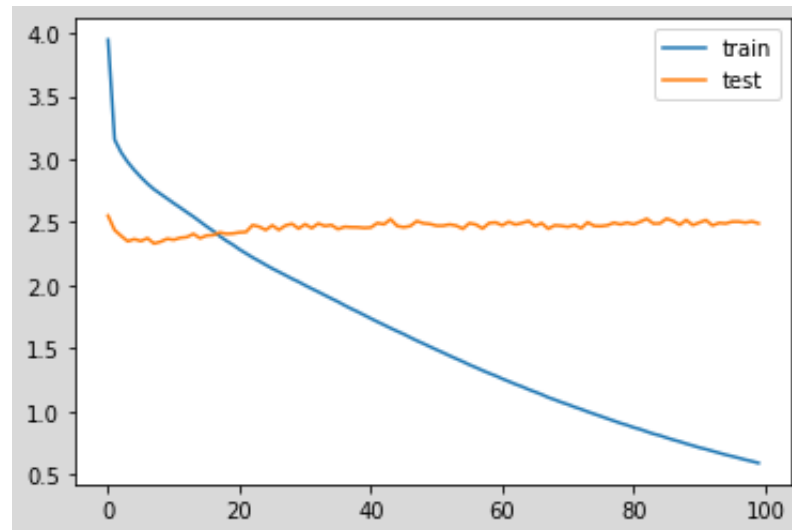


Gambar 4. 4 *Loss* dan *Validation Loss* Model Penerjemah Bangka - Indonesia

### 2. Model Penerjemah Bangka – Inggris

Hasil yang didapatkan untuk proses pelatihan terhadap Bahasa Bangka – Bahasa Inggris menghasilkan nilai *loss* pada validasi yang fluktuatif, dengan nilai *loss* terendah 3.9 pada nilai *loss training* hingga 2.6 pada nilai *loss* validasi.

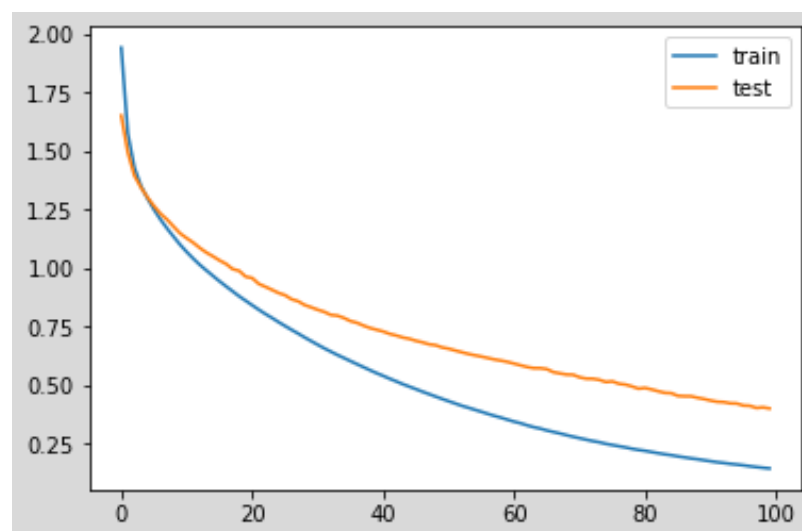
Berikut grafik nilai *loss* dan *validation loss* model penerjemah Bangka – Indonesia Gambar 4.5.



Gambar 4. 5 *Loss* dan *Validation Loss* Model Penerjemah Bangka – Inggris

### 3. Model Penerjemah Indonesia – Bangka

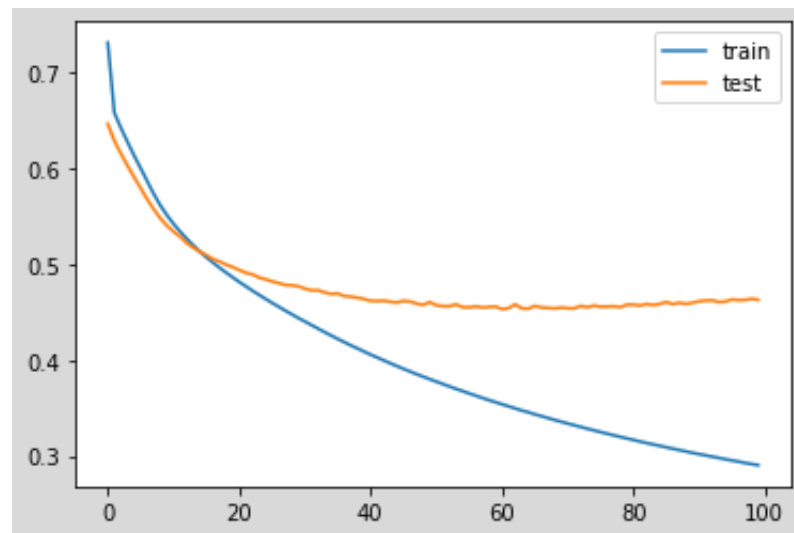
Hasil yang didapatkan untuk proses pelatihan terhadap Bahasa Indonesia – Bahasa Bangka menghasilkan nilai *loss* pada validasi yang fluktuatif, dengan nilai *loss* terendah 0.1 pada nilai *loss* training hingga 0.5 pada nilai *loss* validasi. Berikut grafik nilai *loss* dan *validation loss* model penerjemah Bangka – Indonesia Gambar 4.6.



Gambar 4. 6 *Loss* dan *Validation Loss* Model Penerjemah Indonesia – Bangka

#### 4. Model Penerjemah Indonesia – Inggris

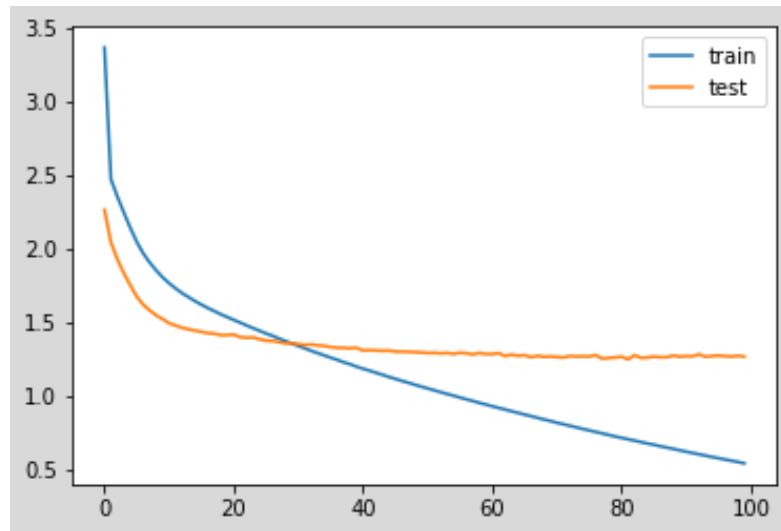
Hasil yang didapatkan untuk proses pelatihan terhadap Bahasa Indonesia – Bahasa Inggris menghasilkan nilai *loss* pada validasi yang fluktuatif, dengan nilai *loss* terendah 0.2 pada nilai *loss training* hingga 0.4 pada nilai *loss* validasi. Berikut grafik nilai *loss* dan *validation loss* model penerjemah Bangka – Indonesia Gambar 4.7.



Gambar 4. 7 *Loss* dan *Validation Loss* Model Penerjemah Indonesia – Inggris

#### 5. Model Penerjemah Inggris – Bangka

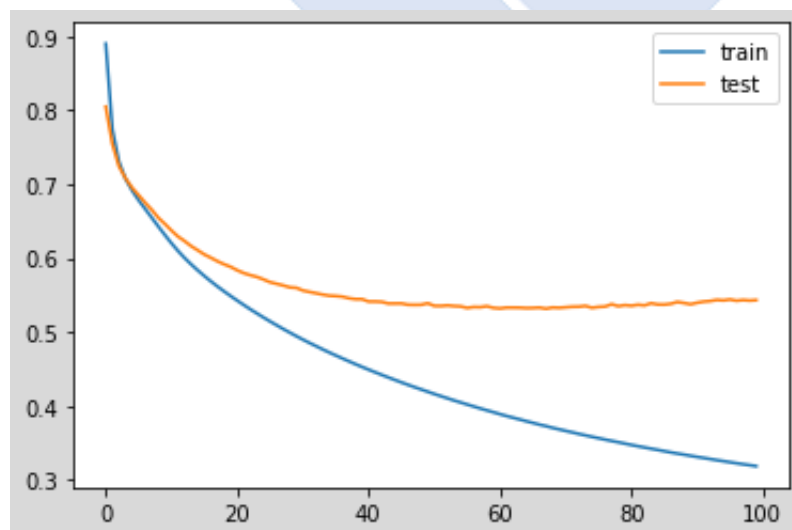
Hasil yang didapatkan untuk proses pelatihan terhadap Bahasa Inggris – Bahasa Bangka menghasilkan nilai *loss* pada validasi yang fluktuatif, dengan nilai *loss* terendah 0.5 pada nilai *loss training* hingga 1.4 pada nilai *loss* validasi. Berikut grafik nilai *loss* dan *validation loss* model penerjemah Bangka – Indonesia Gambar 4.8.



Gambar 4. 8 *Loss* dan *Validation Loss* Model Penerjemah Inggris - Indonesia

## 6. Model Penerjemah Inggris – Indonesia

Hasil yang didapatkan untuk proses pelatihan terhadap Bahasa Inggris – Bahasa Indonesia menghasilkan nilai *loss* pada validasi yang fluktuatif, dengan nilai *loss* terendah 0.2 pada nilai *loss training* hingga 0.58 pada nilai *loss* validasi. Berikut grafik nilai *loss* dan *validation loss* model penerjemah Bangka – Indonesia Gambar 4.9.



Gambar 4. 9 *Loss* dan *Validation Loss* Model Penerjemah Inggris – Bangka

#### 4.4.2 BLEU Scores

Pada tahap evaluasi menggunakan BLEU Score bertujuan untuk menilai seberapa akurat hasil terjemahan yang dihasilkan oleh model mesin penerjemah dengan menggunakan BLEU Score. Hasil terjemahan akan dibandingkan dengan kalimat referensi yang terdapat pada data korpus paralel. Hasil evaluasi membandingkan kalimat referensi dibandingkan dengan kalimat hasil terjemahan. Rentang penilaian BLEU Score adalah dari 0 – 1. Hasil terjemahan yang mendekati kalimat referensi akan di nilai dengan skor 1, kalimat referensi yang dimaksudkan adalah kalimat yang terdapat pada korpus paralel. Pada penelitian ini penentuan nilai BLEU Score menggunakan library NLTK (*Natural Language Toolkit Library*) dengan bahasa *python*. Berikut Tabel 7. yang menampilkan hasil dari evaluasi BLEU Score.

```
from nltk.translate.bleu_score import sentence_bleu
reference = [['this', 'is', 'a', 'test'], ['this', 'is', 'a', 'test']]
candidate = ['this', 'is', 'a', 'test']
score = sentence_bleu(reference, candidate)
```

Berikut hasil evaluasi skor BLEU pada masing – masing model yang telah melalui tahapan pengujian dapat dilihat pada Tabel 4.17.

Tabel 4. 17 Hasil Evaluasi BLEU Scores

	<b>Model BN-ID</b>	<b>Model BN-EN</b>	<b>Model ID-BN</b>	<b>Model ID-EN</b>	<b>Model EN-BN</b>	<b>Model EN-ID</b>
<b>Jumlah Dataset</b>	3.008	3.008	3.008	88.000	3.008	88.000
<b>Epoch</b>	100	100	100	100	100	100
<b>Batch Size</b>	28	28	28	256	28	256
<b>BLEU Scores</b>	<b>55,3%</b>	<b>64,7%</b>	<b>54,5%</b>	<b>55,8%</b>	<b>40,2%</b>	<b>37,3%</b>

Tabel 4.17 menampilkan hasil rata-rata skor BLEU pada semua model. Dilihat dari Tabel 4.17 terlihat bahwa Model BN – EN memiliki skor yang lebih tinggi dibandingkan dengan model lainnya. Percobaan lain pada saat proses



terjemahan, peneliti menemukan beberapa kali skor 0 pada semua BLEU dari semua model yang dibuat, yang artinya terjemahan yang dihasilkan sama sekali berbeda dari referensi.

Tabel 4. 18 Hasil Evaluasi BLEU Scores Penelitian Terkait

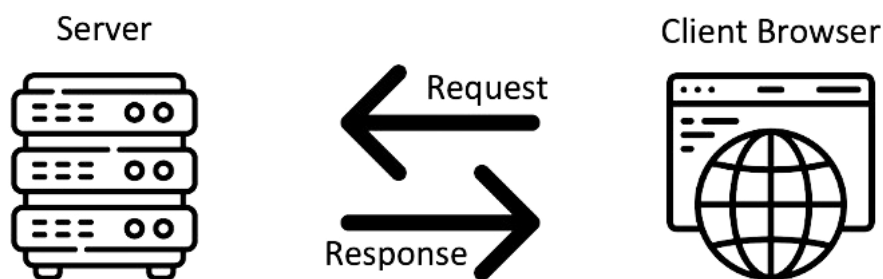
Model	Judul	BLEU Scores	
LSTM + Attention	Pengembangan Model Mesin Penerjemah Inggris Indonesia Menggunakan <i>Reccurent Neural Network</i> LSTM dan <i>Attention</i> [23].	68,04%	
RNN	Penerapan <i>Neural Machine Translation</i> untuk Eksperimen Penerjemahan secara Otomatis pada Bahasa Lampung – Indonesia [24].	41,79%	
RNN	Mesin Penterjemah Bahasa Indonesia – Bahasa Sunda Menggunakan <i>Recurrent Neural Networks</i> [25].	92,63%	
Simple RNN	Perancangan Mesin Translasi berbasis <i>Neural</i> dari Bahasa Kawi ke dalam Bahasa Indonesia menggunakan <i>Microframework</i> Flask [26].	20,43%	
RNN	Peneliti	Bangka – Indonesia	55,32%
		Bangka – Inggris	64,73%
		Indonesia – Bangka	54,5%
		Indonesia – Inggris	55,83%
		Inggris – Bangka	40,2%
		Inggris – Indonesia	37,32%

Penjelasan dari hasil perbandingan penelitian terkait adalah untuk penelitian mesin penerjemah masih belum terlalu banyak. Peneliti menggunakan model RNN

seperti [24] sebagai referensi karena sama-sama menggunakan model RNN. Dari beberapa model yang berhasil peneliti buat terdapat beberapa model yang memiliki nilai BLEU Scores yang lebih tinggi seperti pada Model penerjemah Bangka – Indonesia, Bangka – Inggris, Indonesia – Bangka dan Indonesia – Inggris, namun terdapat beberapa model yang masih memiliki nilai BLEU Scores dibawah rata-rata seperti Inggris – Bangka dan Inggris – Indonesia.

#### 4.5 Implementasi Sistem

Pada tahap implementasi sistem peneliti menggunakan bahasa pemrograman *python* dengan menggunakan *microframework* Flask. Untuk arsitekur pada aplikasi dibagi menjadi dua yaitu, *client* dan *server*, *client* akan melakukan *request* dan menerima *response* dari *server*, pada hal ini *server* bertugas sebagai pemberi layanan pada aplikasi. Komunikasi yang terjadi antara *client* dan *server* menggunakan metode REST API, sehingga memungkinkan *user* untuk melakukan *request* ke *server* melalui *url* yang telah disediakan oleh *server*, *server* kemudian mengirimkan *response* kepada *user* berdasarkan apa yang diminta oleh *user*. Pemilihan Flask menjadi *tools* dalam penelitian ini adalah karena Flask merupakan *microframework* berbasis *python* yang tepat untuk arsitektur REST API. Gambar 4.10 adalah ilustrasi arsitektur aplikasi yang digunakan dalam penelitian ini.

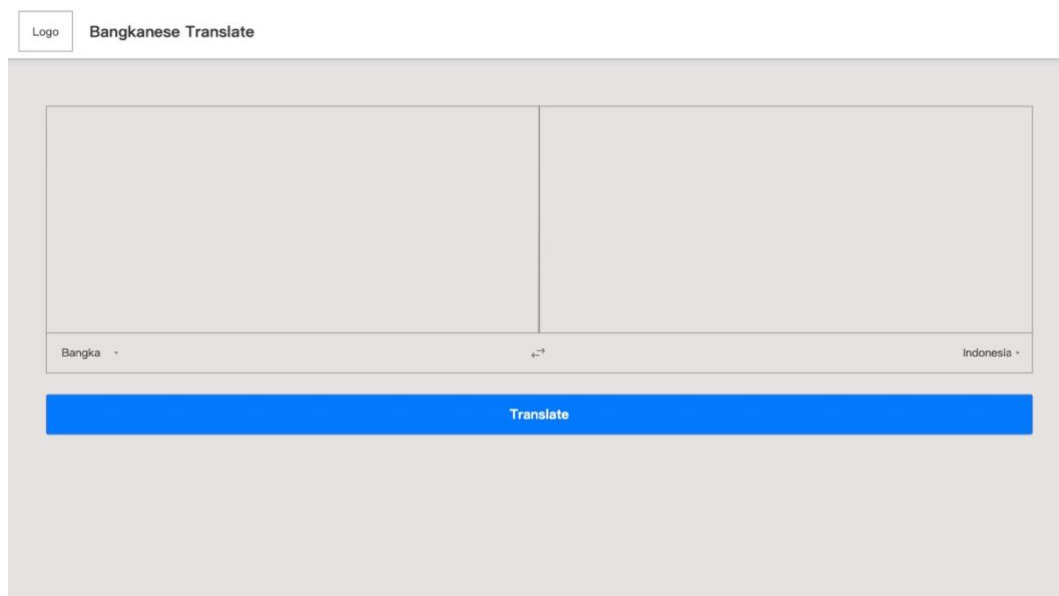


Gambar 4. 10 Arsitektur Aplikasi

Sisi *client* bertugas untuk mempermudah *user* melakukan *request* pada aplikasi, oleh karena itu diperlukan *design interface* yang menarik dan mudah

dipahami oleh *user* agar *client* dapat dengan mudah untuk mendapatkan apa yang mereka inginkan. Dalam penelitian ini aplikasi dibuat dengan *single page application* sederhana karena memang fitur dalam aplikasi kali ini adalah hanya untuk menerjemahkan kalimat dari inputan *user*, sehingga diperlukan *user interface* yang sesederhana mungkin.

#### 4.5.1 Pembuatan *User Interface*



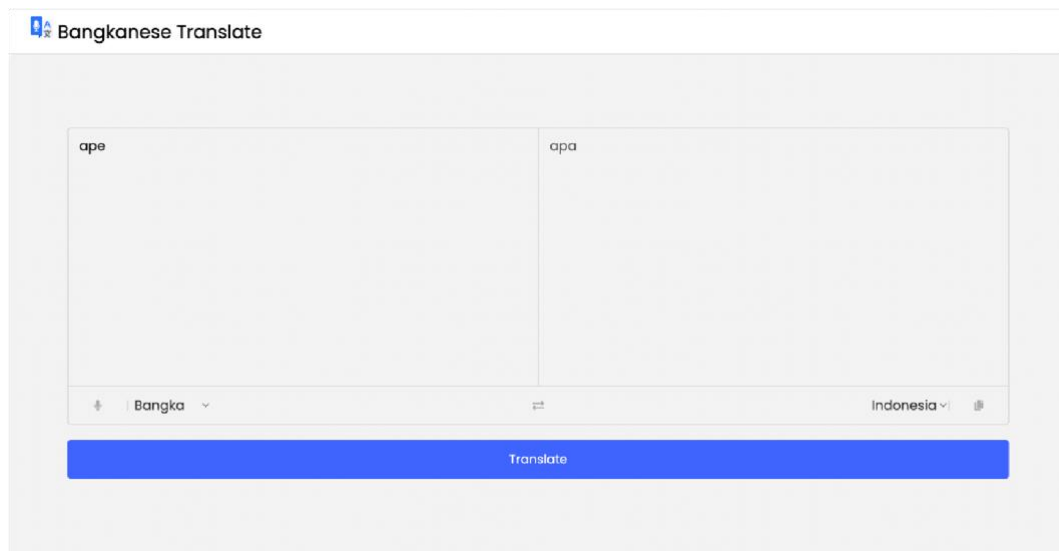
Gambar 4. 11 *User Interface*

Penjelasan dari *user interface* seperti ditunjukkan pada Gambar 4.11 adalah aplikasi *client* adalah berupa *one page application* atau aplikasi yang terdiri dari hanya satu halaman. Dalam halaman tersebut terbagi menjadi dua bagian penting yaitu *input text* dan *output text*, berikut penjelasan dari *user interface* tersebut.

1. *Input text*, merupakan teks area yang digunakan *user* untuk memasukkan kalimat yang akan diterjemahkan.
2. *Output text*, merupakan teks area yang akan menampilkan hasil terjemahan dari kalimat yang telah dimasukkan *user* pada *input text*.
3. *Select language*, merupakan tempat *user* untuk memilih bahasa yang di input dan bahasa yang akan menjadi *output* atau hasil dari terjemahan.

4. *Button translate*, merupakan tombol untuk user akan menampilkan hasil dari terjemahan.

#### 4.5.2 Tampilan Aplikasi



Gambar 4. 12 Tampilan *Website*

Setelah *user* memasukkan kalimat kedalam kotak *input* dan memilih bahasa *user* bisa menekan tombol *translate* untuk melakukan proses penerjemahan. Ketika proses penerjemahan berjalan akan ada proses tampilan *loading* dari aplikasi yang menandakan proses sedang berjalan. Kemudian setelah proses selesai *output* dari terjemahan tersebut akan langsung ditampilkan tepat di kotak *output*. Gambar 4.12 adalah ilustrasi hasil dari *output* terjemahan.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan dari hasil pembuatan dan pengujian model mesin penerjemah *deep learning* dengan *Recurrent Neural Network* (RNN) maka dapat ditarik kesimpulan sebagai berikut:

1. Pada penelitian ini menghasilkan enam model *deep learning* dengan *Recurrent Neural Network* untuk mesin penerjemah. Model ini akan menghasilkan terjemahan dari Bahasa Bangka – Indonesia – Inggris.
2. Pada penelitian ini menghasilkan aplikasi berbasis website untuk melakukan proses penterjemahan berdasarkan model. Evaluasi pada model menggunakan *BLEU Score*. Dari proses evaluasi mendapatkan hasil 55,32% untuk model Bangka – Indonesia, Bangka – Inggris sebesar 64,73%, 54,5% untuk model Indonesia – Bangka, model Indonesia – Inggris sebesar 55,83%, 40,2% untuk model Inggris – Bangka dan model Inggris – Indonesia sebesar 37,32%. Dari hasil tersebut didapatkan bahwa nilai *BLEU Score* tertinggi didapatkan oleh model Bangka – Inggris yaitu sebesar 64,73% dan nilai *BLEU Score* terendah didapatkan oleh model Inggris – Indonesia 37,32%.

#### **5.2 Saran**

Berdasarkan dari hasil penelitian yang telah dilakukan, maka peneliti memberikan beberapa saran untuk penelitian selanjutnya, agar dapat menambah data korpus paralel dan membandingkan model mesin penerjemah dengan arsitektur model lainnya seperti *Attention Mechanism* ataupun BERT.

## DAFTAR PUSTAKA

- [1] A. Ginting and N. AZ, "Penerjemah Dua Arah Bahasa Indinonesia Ke Bahasa Daerah", *Jurnal Telematika MKOM*, 2012.
- [2] P. P. Saputra and M. Afifulloh, "Pemetaan Penggunaan Bahasa Melayu Bangka," *Jurnal Ilmiah Bahasa dan Sastra Terakreditasi Sinta 4 berdasarkan Keputusan Direktorat Jenderal Penguatan Riset dan Pengembangan, Kementerian Riset, Teknologi dan Pendidikan Tinggi Republik Indonesia*, vol. 3, 2020.
- [3] D. Bahdanau, K. Cho and B. Yoshua, "Neural machine translation by jointly learning to align and translate," *International Conference on Learning Representations*, 2015.
- [4] S. Yang, Y. Wang and X. Chu, "A Survey of Deep Learning Techniques for Neural Machine Translation," *ArXiv*, 2020.
- [5] R. Sennrich and B. Zhang, "Revisiting Low-Resource Neural Machine Translation: A Case Study," *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics*, pp. 211-221, 2020.
- [6] S. Hunston, "*Corpora in Applied Linguistics*," Cambridge: Cambridge University Press, pp. 71-72, 2012.
- [7] Z. Abidin, "Penerapan Neural Machine Translation untuk Eksperimen Penerjemahan secara Otomatis pada Bahasa Lampung -Indonesia," *Prosiding Seminar Nasional Metode Kuantitatif*, No. 1, 2017.
- [8] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 10 2002.

- [9] K. Sintoris and K. Vergidis, "Extracting business process models using natural language processing (NLP) techniques," in *2017 IEEE 19th conference on business informatics (CBI)*, 2017.
- [10] V. Sorin, Y. Barash, E. Konen and E. Klang, "Deep Learning for Natural Language Processing in Radiology—Fundamentals and a Systematic Review," *Journal of the American College of Radiology*, vol. 17, 2020.
- [11] N. Indurkha and F. J. Damerau, *Handbook of Natural Language Processing (second edition)*, Chapman and Hall/CRC, 2010.
- [12] S. Sun, C. Luo and J. Chen, "A review of natural language processing techniques for opinion mining systems," *Information Fusion*, vol. 36, pp. 1025, 2017.
- [13] G. A. Jivani, "A Comparative Study of Stemming Algorithms," *International Journal of Control Theory and Applications*, vol. 2, pp. 1930-1938, 2012.
- [14] J. Méndez Reboledo, E. Iglesias, F. Fdez-Riverola, F. Díaz and J. Corchado Rodríguez, "Tokenising, Stemming and Stopword Removal on Anti-spam Filtering Domain," *Current Topics in Artificial Intelligence*, vol. 4177, pp. 449-458, 2005.
- [15] I. Goodfellow, Y. Bengio and A. Courville, "*Deep Learning*," MIT Press, 2016.
- [16] A. Sharma, E. Vans, D. Shigemizu, K. Boroevich and T. Tsunoda, "DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture," *Scientific Reports*, vol. 9, 2019.
- [17] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo and G.-Z. Yang, "Deep Learning for Health Informatics," *EEE journal of biomedical and health informatics*, vol. PP, 2016.
- [18] A. Karpathy, "The Unreasonable Effectiveness of Recurrent Neural Networks," 2015. [Online]. Available: <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>.

- [19] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, pp. 1735-80, 1997.
- [20] J. Brownlee, "Deep Learning For Natural Language Processing," Machine Learning Mastery, 2017.
- [21] N. H. Harani,, C. Prianto and M. Hasanah, "Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python," *Jurnal Teknik Informatika*, vol. 11, no. 3, pp. 47-53, 2019.
- [22] R. Irsyad, "Penggunaan Python Web Framework Flask Untuk Pemula," 2018.
- [23] M. R. S. Huda, "Pengembangan Model Mesin Penerjemah Inggris Indonesia Menggunakan Recurrent Neural Network Lstm Dan Attention," *Doctoral dissertation, Universitas Gadjah Mada*, 2020.
- [24] Z. Abidin, "Penerapan Neural Machine Translation untuk Eksperimen Penerjemahan secara Otomatis pada Bahasa Lampung – Indonesia," *Prosiding Seminar Nasional Metode Kuantitatif*, 2017.
- [25] Y. Fauziyah, R. Ilyas and F. Kasyidi, "Mesin Penerjemah Bahasa Indonesia-Bahasa Sunda Menggunakan Recurrent Neural Networks," *Jurnal TEKNOINFO*, vol. 16, pp. 313-322, 2022.
- [26] I. G. B. A. Budaya, M. W. A. Kesiman and I. M. G. Sunarya, "Perancangan Mesin Translasi berbasis Neural dari Bahasa Kawi ke dalam Bahasa Indonesia menggunakan Microframework Flask," *Jurnal Sistem Dan Informatika (JSI)*, vol. 16, pp. 94-103, 2022.
- [27] V. Sorin, Y. Barash, E. Konen and E. Klang, "Deep Learning for Natural Language Processing in Radiology—Fundamentals and a Systematic Review," *Journal of the American College of Radiology*, vol. 17, 2020.





**LAMPIRAN**

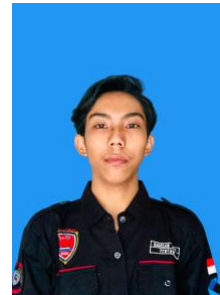


**Lampiran 1: Daftar Riwayat Hidup**

## DAFTAR RIWAYAT HIDUP

### 1. Data Pribadi

Nama Lengkap : Fadel Razsiah  
Tempat & tanggal lahir : Pangkalpinang, 30 September 2000  
Alamat : Jl. Prajurit KKO Harun No. 49A  
Jenis Kelamin : Laki – laki  
Agama : Islam  
No. Handphone : 0812-739-2112  
Email : [fadelrzsh@gmail.com](mailto:fadelrzsh@gmail.com)



### 2. Riwayat Pendidikan

a. 2006 – 2012 : SD Negeri 48 Pangkalpinang  
b. 2012 – 2015 : SMP Negeri 1 Pangkalpinang  
c. 2015 – 2018 : SMA Negeri 3 Pangkalpinang  
d. 2019 – Sekarang : Politeknik Manufaktur Negeri Bangka Belitung

Sungailiat, 18 Januari 2023

Tanda Tangan

Fadel Razsiah



**Lampiran 2: Form Monitoring**

TRPL

FORM-PPR-3-6: Foto Monitoring Proyel Akhir

FORM MONITORING PROYEL AKHIR TAJRIK AR-RUMI 2022 / 2023			
JUDUL		Aplikasi Penerjemah Bahasa Bangka - Indonesia - Inggris dengan Algoritma Heural Machine Tranlaison (HMT)	
Nama Mahasiswa		1. Fadel Ratsiah /NIRN: 1061911 2. .... /NIRN: ..... 3. .... /NIRN: ..... 4. .... /NIRN: ..... 5. .... /NIRN: .....	
Monitoring ke	Tanggal	Progress Alat	Paraf Pembimbing
1	01/07 2022	15% (Pengumpulan sumber)	
2	09/10 2022	55% (Aplikasi)	
3	05/12/2022	60%	
4	18/11 2023	100%	
5	18/1/23	100% ✓	


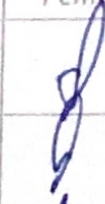






KESIAPAN ALAT UNTUK SIDANG: SIAP / ~~SIAP~~ (coret salah satu)

Mengetahui		
Pembimbing 1	Pembimbing 2	Pembimbing 3
 (.....)	 (SARI MUBARROT)	(.....)



**Lampiran 3: Form Bimbingan-1**

FORM-PTR-3-4: Bimbingan Proyek Akhir

 <p style="text-align: center;">FORM BIMBINGAN PROYEK AKHIR TAHUN AKADEMIK 2021 / 2022</p>			
JUDUL		Aplikasi Penerjemah Bahasa Bangka - Indonesia - Inggris dengan Algoritma Neural Machine Translation (NMT)	
Nama Mahasiswa		Fadel Pazziah NIRM : 1061911	
Nama Pembimbing		1. <u>Ahmat Josi, M.Kom</u> 2. <u>Sari Mubarrh, M.Pd.</u> 3. _____	
Pertemuan Ke	Tanggal	Topik Bimbingan	Paraf dan nama Pembimbing
1	17/05/2022	Bab I Latar Belakang	
2	17/06/2022	Bab I Rumusan Masalah	
3	4/07/2022	Bab I Tujuan	
4	11/07/2022	Bab 2,3 metode pelaksanaan	
5	18/07/2022	Bab 2,3 Rincian metode pelaksanaan	
6	29/07/2022	Bab 2,3 landasan teori	
7	14/10/2022	Bab 3	
8	6/12/2022	Program Program GUY	
9	15/12/2022	program Aplikasi	
10			

Catatan :


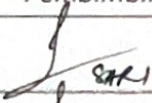
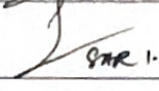
- Jika pertemuan bimbingan lebih dari sepuluh kali, dapat mengambil Form kembali ke komisi



**Lampiran 4: Form Bimbingan-2**



FORM-PPR-3- 4: Bimbingan Proyek Akhir

		FORM BIMBINGAN PROYEK AKHIR TAHUN AKADEMIK 2021 / 2022	
JUDUL		Aplikasi Penerjemah Bahasa Bangka - Indonesia - Inggris dengan Neural Machine Translation (NMT)	
Nama Mahasiswa		Fadel Patsiah ..... NIRM: 1061911	
Nama Pembimbing		1. _____ 2. _____ 3. _____	
Pertemuan Ke	Tanggal	Topik Bimbingan	Paraf dan nama Pembimbing
1	12/8/22	Bab 1-3	 STK 1
2	18/01/23	Bab 1,2,3,4,5 / dat 75%	 STK 1
3			
4			
5			
6			
7			
8			
9			
10			

Catatan:

- Jika pertemuan bimbingan lebih dari sepuluh kali, dapat mengambil Form kembali di Panitia/Komisi Proyek Akhir



**Lampiran 5: Plagiasi Turnitin**

# FIXED

---

## ORIGINALITY REPORT

---

16%

SIMILARITY INDEX

16%

INTERNET SOURCES

1%

PUBLICATIONS

9%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

1	<a href="http://rocoderes.com">rocoderes.com</a> Internet Source	5%
2	<a href="http://dspace.uii.ac.id">dspace.uii.ac.id</a> Internet Source	3%
3	<a href="http://lib.unnes.ac.id">lib.unnes.ac.id</a> Internet Source	2%
4	<a href="http://vdocuments.pub">vdocuments.pub</a> Internet Source	2%
5	<a href="http://www.routech.ro">www.routech.ro</a> Internet Source	2%
6	Submitted to Politeknik Manufaktur Negeri Bangka Belitung Student Paper	1%
7	<a href="http://medium.com">medium.com</a> Internet Source	1%
8	<a href="http://repository.polman-babel.ac.id">repository.polman-babel.ac.id</a> Internet Source	1%

---

---

Exclude quotes      On

Exclude matches      < 1%

Exclude bibliography      On



**Lampiran 6: Bukti Publikasi**



**JITT :**

**JURNAL INOVASI TEKNOLOGI TERAPAN  
POLITEKNIK MANUFAKTUR NEGERI BANGKA BELITUNG**

Kawasan Industri Air Kantung Sungailiat – Bangka 33211, Telp (0717)93586, Fax (0717)93585  
website : <https://jitt.polman-babel.ac.id>

e-ISSN : xxxx-xxxx

---

**SURAT KETERANGAN**

Nomor : 038/PL.28.C/PB/2023

Dengan ini menerangkan bahwa artikel yang berjudul :

**“APLIKASI PENERJEMAH BAHASA BANGKA KE BAHASA  
INDONESIA MENGGUNAKAN NEURAL MACHINE  
TRANSLATION BERBASIS WEBSITE”**

Atas nama :

Penulis : **FADEL RAZSIAH, AHMAT JOSI, SARI MUBAROH**

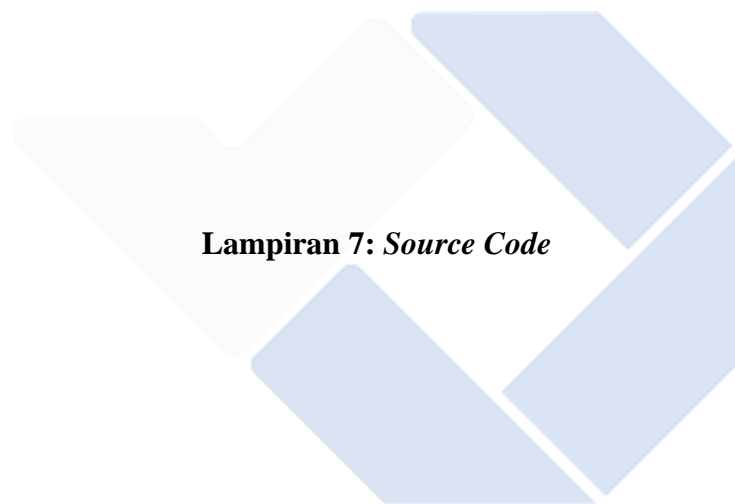
Afiliasi : **POLITEKNIK MANUFAKTUR NEGERI BANGKA BELITUNG**

Telah mengirimkan artikel dengan status *Submit* di Jurnal Inovasi Teknologi Terapan (JITT) Politeknik Manufaktur Negeri Bangka Belitung pada Tanggal 17 Januari 2023.

Demikian surat keterangan ini dibuat untuk dapat digunakan sebagaimana mestinya.

Sungailiat, 18 Januari 2023  
Kepala P3KM,

**Dr. Parulian Silalahi, M.Pd**  
NIK. 1901010201640006



**Lampiran 7: *Source Code***

## a. app.py

```
from flask import Flask, render_template, request, jsonify
from keras.models import Model
import numpy as np
from tensorflow.keras.layers import Input, Dense, Embedding
from tensorflow.keras.preprocessing.sequence import pad_sequences
import tensorflow as tf
import pickle
import keras as k

app = Flask(__name__, static_url_path='/static')

# Load the models and tokenizers
models = {
    'bn-id': {'model': 'models/model_bn2id.h5',
             'src_tokenizer': 'tokenizer/source_bn2id.pkl',
             'tgt_tokenizer': 'tokenizer/target_bn2id.pkl',
             },
    'bn-en': {'model': 'models/model_bn2en.h5',
             'src_tokenizer': 'tokenizer/source_bn2en.pkl',
             'tgt_tokenizer': 'tokenizer/target_bn2en.pkl',
             },
    'id-bn': {'model': 'models/model_id2bn.h5',
             'src_tokenizer': 'tokenizer/source_id2bn.pkl',
             'tgt_tokenizer': 'tokenizer/target_id2bn.pkl',
             },
    'id-en': {'model': 'models/model_id2en.h5',
             'src_tokenizer': 'tokenizer/source_id2en.pkl',
             'tgt_tokenizer': 'tokenizer/target_id2en.pkl',
             },
    'en-id': {'model': 'models/model_en2id.h5',
             'src_tokenizer': 'tokenizer/source_en2id.pkl',
             'tgt_tokenizer': 'tokenizer/target_en2id.pkl',
             },
    'en-bn': {'model': 'models/model_en2bn.h5',
```



```

        'src_tokenizer': 'tokenizer/source_en2bn.pkl',
        'tgt_tokenizer': 'tokenizer/target_en2bn.pkl',
    },
}

@app.route("/")
def index():
    return render_template("index.html")

# Load the tokenizers and models
for lang_pair in models.keys():
    with open(models[lang_pair]['src_tokenizer'], 'rb') as handle:
        models[lang_pair]['src_tokenizer'] = pickle.load(handle)
    with open(models[lang_pair]['tgt_tokenizer'], 'rb') as handle:
        models[lang_pair]['tgt_tokenizer'] = pickle.load(handle)
    models[lang_pair]['model'] = tf.keras.models.load_model(models[lang_pair]['model'])

@app.route('/translate', methods=['POST'])
def translate():
    src_input_text = request.form['source_text']
    src_lang = request.form['source_language']
    tgt_lang = request.form['target_language']
    lang_pair = src_lang + '-' + tgt_lang

    latent_dim = 50

    # inference encoder
    encoder_inputs_inf = models[lang_pair]['model'].input[0]
    encoder_outputs_inf, inf_state_h, inf_state_c = models[lang_pair]['model'].layers[4].output
    encoder_inf_states = [inf_state_h, inf_state_c]
    encoder = Model(encoder_inputs_inf, encoder_inf_states)

    # inference decoder
    decoder_state_h_input = Input(shape = (latent_dim,))
    decoder_state_c_input = Input(shape = (latent_dim,))

```

```

decoder_state_input = [decoder_state_h_input, decoder_state_c_input]

decoder_input_inf = models[lang_pair]['model'].input[1]
decoder_emb_inf = models[lang_pair]['model'].layers[3](decoder_input_inf)
decoder_lstm_inf = models[lang_pair]['model'].layers[5]
decoder_output_inf, decoder_state_h_inf, decoder_state_c_inf =
decoder_lstm_inf(decoder_emb_inf, initial_state= decoder_state_input)
decoder_state_inf = [decoder_state_h_inf, decoder_state_c_inf]

# inference dense layer
dense_inf = models[lang_pair]['model'].layers[6]
decoder_output_final = dense_inf(decoder_output_inf)
decoder = Model([decoder_input_inf] + decoder_state_input, [decoder_output_final] +
decoder_state_inf)

reverse_word_map_target = dict(map(reversed,
models[lang_pair]['tgt_tokenizer'].word_index.items()))

def decode_seq(input_seq):
    state_values_encoder = encoder.predict([input_seq])
    target_seq = np.zeros((1, 1))
    target_seq[0, 0] = models[lang_pair]['tgt_tokenizer'].word_index['start']
    stop_condition = False
    decoder_sentence = ""
    while not stop_condition:
        sentence, decoder_h, decoder_c = decoder.predict([target_seq] + state_values_encoder)
        sample_word_index = np.argmax(sentence[0,-1,:])
        decoder_word = reverse_word_map_target[sample_word_index]
        decoder_sentence += ' ' + decoder_word
        if (decoder_word == 'end' or
            len(decoder_sentence) > 70):
            stop_condition = True
        target_seq[0, 0] = sample_word_index
        state_values_encoder = [decoder_h, decoder_c]
    return decoder_sentence

```

```

sentence = request.form['source_text']

input_seq = models[lang_pair]['src_tokenizer'].texts_to_sequences([sentence])
max_len = len(input_seq[0])
pad_sequence = pad_sequences(input_seq, maxlen = max_len, padding = 'post')

translation = decode_seq(pad_sequence)

k.backend.clear_session()

return render_template("index.html", translate = "{}".format(translation[:-3]), sentence =
"{}".format(sentence))

if __name__ == '__main__':
    app.run(debug=True)

```

## b. index.html

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
  <meta charset="utf-8">
  <title>Bangkanese Translate</title>
  <link rel="icon" href="//ssl.gstatic.com/translate/favicon.ico" sizes="64x64">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='css/styles.css') }}">
</head>
<body>
  <nav class="navbar">

```

```

    <div class="logo"> Bangkanese Translate</div>
</nav>
<form action="{{url_for('translate')}}" method="post">
    <div class="container">
        <div class="wrapper">
            <div class="text-input">
                <textarea id="source_text" name="source_text" spellcheck="false" class="from-text"
placeholder="Enter text">{{sentance}}</textarea>
                <textarea id="target_text" name="target_text" spellcheck="false" class="to-text"
disabled readonly placeholder="Translation">{{translate}}</textarea>
            </div>
            <ul class="controls">
                <li class="row from">
                    <select id="source_language" name="source_language">
                        <option value="bn">Bangka</option>
                        <option value="id">Indonesia</option>
                        <option value="en">English</option>
                    </select>
                </li>
                <li class="exchange"><i class="fas fa-exchange-alt"></i></li>
                <li class="row to">
                    <select id="target_language" name="target_language">
                        <option value="id">Indonesia</option>
                        <option value="bn">Bangka</option>
                        <option value="en">English</option>
                    </select>
                </li>
            </ul>
        </div>
        <button type="submit" value="Translate">Translate</button>
    </div>
</form>
</body>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>window.jQuery || document.write('<script src="path/to/jquery.min.js"></script>')</script>

```

```
<script type = "text/javascript" src="{{ url_for('static', filename='js/main.js') }}"></script>
<script type = "text/javascript" src="{{ url_for('static', filename='js/script.js')}}"></script>
<script type = "text/javascript" src="{{ url_for('static', filename='js/countries.js')}}"></script>
</html>
```

### c. styles.css

```
/* Import Google Font - Poppins */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
body{
  background: #fff;
}
.navbar{
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 10px;
  background-color: #fff;
  color: #000;
  border-width: 0 0 1px 0;
  border-color: rgb(197, 197, 197);
  border-style: solid;
}
.logo{
  justify-content: center;
  align-content: center;
  vertical-align: center;
  padding-left: 20px;
```

```
font-size: 25px;
}
.container{
align-items: center;
justify-content: center;
max-width: 1500px;
width: 100%;
padding: 100px 80px 150px 80px;
background: rgb(243, 243, 243);
border-radius: 7px;
box-shadow: 0 10px 20px rgba(0,0,0,0.01);
}
.wrapper{
border-radius: 5px;
border: 1px solid #ccc;
}
.wrapper .text-input{
display: flex;
border-bottom: 1px solid #ccc;
}
.text-input .to-text{
border-left: 1px solid #ccc;
}
.text-input textarea{
height: 350px;
width: 100%;
border: none;
outline: none;
resize: none;
background: none;
font-size: 18px;
padding: 10px 15px;
}
.text-input textarea::placeholder{
color: #b7b6b6;
}
```

```
.controls, li, .icons, .icons i{
  display: flex;
  align-items: center;
  justify-content: space-between;
}
.controls{
  list-style: none;
  padding: 12px 15px;
}
.controls .row .icons{
  width: 38%;
}
.controls .row .icons i{
  width: 50px;
  color: #adadad;
  font-size: 14px;
  cursor: pointer;
  transition: transform 0.2s ease;
  justify-content: center;
}
.controls .row.from .icons{
  padding-right: 15px;
  border-right: 1px solid #ccc;
}
.controls .row.to .icons{
  padding-left: 15px;
  border-left: 1px solid #ccc;
}
.controls .row select{
  color: #333;
  border: none;
  outline: none;
  font-size: 18px;
  background: none;
  padding-left: 5px;
}
```

```
.text-input textarea::-webkit-scrollbar{
  width: 4px;
}

.controls .row select::-webkit-scrollbar{
  width: 8px;
}

.text-input textarea::-webkit-scrollbar-track,
.controls .row select::-webkit-scrollbar-track{
  background: #fff;
}

.text-input textarea::-webkit-scrollbar-thumb{
  background: #ddd;
  border-radius: 8px;
}

.controls .row select::-webkit-scrollbar-thumb{
  background: #999;
  border-radius: 8px;
  border-right: 2px solid #ffffff;
}

.controls .exchange{
  color: #adadad;
  cursor: pointer;
  font-size: 16px;
  transition: transform 0.2s ease;
}

.controls i:active{
  transform: scale(0.9);
}

.container button{
  width: 100%;
  padding: 14px;
  outline: none;
  border: none;
  color: #fff;
  cursor: pointer;
  margin-top: 20px;
}
```



```
font-size: 17px;
border-radius: 5px;
background: #3e64fe;
}
@media (max-width: 660px){
.container{
padding: 25px;
padding-bottom: 65px;
}
.wrapper .text-input{
flex-direction: column;
}
.text-input .to-text{
border-left: 0px;
border-top: 1px solid #ccc;
}
.text-input textarea{
height: 200px;
}
.controls .row .icons{
display: none;
}
.container button{
padding: 13px;
font-size: 16px;
}
.controls .row select{
font-size: 16px;
}
.controls .exchange{
font-size: 14px;
}
}
```

#### d. main.js

```

$(document).ready(function() {
  $('#translate_button').click(function() {
    var src_input_text = $('#source_text').val();
    var src_lang = $('#source_language').val();
    var tgt_lang = $('#target_language').val();

    $.post('/translate', {source_text: src_input_text, source_language: src_lang,
target_language: tgt_lang}, function(data) {
      $('#target_text').val(data.translation);
    });
  });

  document.getElementById("source_language").addEventListener("change", function() {
    var source_language = this.value;
    var target_language = document.getElementById("target_language");
    for (var i = 0; i < target_language.options.length; i++) {
      target_language.options[i].style.display = "block";
      if (target_language.options[i].value === source_language) {
        target_language.options[i].style.display = "none";
      }
    }
  });
});

```

### e. script.js

```

window.SpeechRecognition = window.webkitSpeechRecognition ||
window.SpeechRecognition;
const synth = window.speechSynthesis;
const recognition = new SpeechRecognition();
const fromText = document.querySelector(".from-text"),

```

```

toText = document.querySelector(".to-text"),
exchagelcon = document.querySelector(".exchange"),
selectTag = document.querySelectorAll("select"),
icons = document.querySelectorAll(".row i");
selectTag.forEach((tag, id) => {
  for (let country_code in countries) {
    let selected = id == 0 ? country_code == "id" ? "selected" : "" : country_code == "en" ?
"selected" : "";
    let option = `<option ${selected}
value="${country_code}">${countries[country_code]}</option>`;
    tag.insertAdjacentHTML("beforeend", option);
  }
});
exchagelcon.addEventListener("click", () => {
  let tempText = fromText.value,
  tempLang = selectTag[0].value;
  fromText.value = toText.value;
  toText.value = tempText;
  selectTag[0].value = selectTag[1].value;
  selectTag[1].value = tempLang;
});
fromText.addEventListener("keyup", () => {
  if(!fromText.value) {
    toText.value = "";
  }
});
icons.forEach(icon => {
  icon.addEventListener("click", ({target}) => {
    if(!fromText.value || !toText.value) return;
    if(target.classList.contains("fa-microphone")) {
      let paragraph = document.createElement('p');
      let container = document.querySelector('.text-box');
      container.appendChild(paragraph);
      sound.play();
      dictate();
    } else if (target.classList.contains("fa-copy")) {

```

```

        if(target.id == "from") {
            navigator.clipboard.writeText(fromText.value);
        } else {
            navigator.clipboard.writeText(toText.value);
        }
    } else {
        let utterance;
        if(target.id == "from") {
            utterance = new SpeechSynthesisUtterance(fromText.value);
            utterance.lang = selectTag[0].value;
        } else {
            utterance = new SpeechSynthesisUtterance(toText.value);
            utterance.lang = selectTag[1].value;
        }
        speechSynthesis.speak(utterance);
    }
});
});
const dictate = () => {
    recognition.start();
    recognition.onresult = (event) => {
        const speechToText = event.results[0][0].transcript;

        paragraph.textContent = speechToText;

        if (event.results[0].isFinal) {

            if (speechToText.includes('what is the time')) {
                speak(getTime);
            };

            if (speechToText.includes('what is today's date')) {
                speak(getDate);
            };

            if (speechToText.includes('what is the weather in')) {

```

```

    getTheWeather(speechToText);
  };
}
}
}
const speak = (action) => {
  utterThis = new SpeechSynthesisUtterance(action());
  synth.speak(utterThis);
};

const getTime = () => {
  const time = new Date(Date.now());
  return `the time is ${time.toLocaleString('en-US', { hour: 'numeric', minute: 'numeric', hour12: true })}`;
};

const getDate = () => {
  const time = new Date(Date.now());
  return `today is ${time.toLocaleDateString()}`;
};

const getTheWeather = (speech) => {
  fetch(`http://api.openweathermap.org/data/2.5/weather?q=${speech.split('')[5]}&appid=58b6f7c78582bffab3936dac99c31b25&units=metric`)
    .then(function(response){
      return response.json();
    })
    .then(function(weather){
      if (weather.cod === '404') {
        utterThis = new SpeechSynthesisUtterance(`I cannot find the weather for ${speech.split('')[5]}`);
        synth.speak(utterThis);
        return;
      }
    })
};

```

```
utterThis = new SpeechSynthesisUtterance(`the weather condition in ${weather.name} is
mostly full of ${weather.weather[0].description} at a temperature of ${weather.main.temp}
degrees Celcius`);
synth.speak(utterThis);
});
};
```

#### f. countries.js

```
const countries = {
  "bn": "Bangka",
  "en": "English",
  "id": "Indonesian"
}
```