# DETEKSI KALIMAT MENGGUNAKAN METODE BIDIRECTIONAL LSTM -- STUDI KASUS: INDONESIA DAN MALAYSIA


Disusun Oleh:

Nama : Ananda Dirgantara R.S.

NIM : 15523199



**PROGRAM STUDI INFORMATIKA -- PROGRAM SARJANA**

**FAKULTAS TEKNOLOGI INDUSTRI UNIVERSITAS ISLAM INDONESIA**



**2022**

## HALAMAN PENGESAHAN DOSEN PEMBIMBING

# DETEKSI KALIMAT MENGGUNAKAN METODE BIDIRECTIONAL LSTM – STUDI KASUS: INDONESIA DAN MALAYSIA

## TUGASAKHIR

Disusun Oleh:


Nama : Ananda Dirgantara R.S.

NIM : 15523199


Yogyakarta, 22 Juli 2022

Pembimbing,

( Ahmad Fathan Hidayatullah, ST., M.Cs. )

**HALAMAN PENGESAHAN DOSEN PENGUJI**

# DETEKSI KALIMAT MENGGUNAKAN METODE BIDIRECTIONAL LSTM – STUDI KASUS: INDONESIA DAN MALAYSIA

## TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 22 Juli 2022

Tim Penguji

Ahmad Fathan Hidayatullah, S.T., M.Cs. _____

**Anggota 1**

Syarif Hidayat, Dr., S.Kom., M.I.T. _____

**Anggota 2**

Irving Vitra Paputungan, S.T., M.Sc., Ph.D. _____

Mengetahui,

Ketua Program Studi Informatika Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

( Dr. Raden Teduh Dirgahayu, S.T., M.Sc. )

## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama: Ananda Dirgantara R.S.

NIM : 15523199

Tugas akhir dengan judul:

# DETEKSI KALIMAT MENGGUNAKAN METODE BIDIRECTIONAL LSTM – STUDI KASUS: INDONESIA DAN MALAYSIA

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 Juli 2022


( Ananda Dirgantara Rama Syahputra )

# HALAMAN PERSEMBAHAN

Puji dan syukur kepada Allah *Subhanallahu Wa Ta'ala* atas segala rahmat dan karunia-Nya, sehingga tugas akhir ini dapat terselesaikan dengan baik.

Tugas akhir ini saya persembahkan kepada:

**Keluarga**

Terima kasih untuk doa, dukungan, cinta dan kasih sayang kalian.

## HALAMAN MOTO

Aku yang sedang rnenulis ini sekarang, dan kau yang sedang rnernbacanya. Benarkah ini diriku dan dirirnu yang sebenamya?

# KATA PENGANTAR

Assalamu' alaikum warahmatullahi wabarakatuh

Alhamdulillahi Rabbil'alamin, puji syukur kepada Allah Subhanallahu Wa Ta'ala atas rahmat, hidayah dan karunia-Nya yang diberikan, sehingga penulis diberikan kesempatan untuk menyelesaikan laporan penelitian tugas akhir ini. Tak lupa shalawat dan salam penulis panjatkan kepada junjungan kita Nabi Muhammad SAW, yang telah membawa kita dari zaman jahiliah menuju zaman yang terang benderang. Keberhasilan dan kelancaran yang didapatkan dalam proses menyelesaikan tugas akhir ini semata-mata karena kemudahan dan pertolongan yang diberikan oleh Allah Subhanallahu Wa Ta'ala.

Tugas akhir yang berjudul "**Deteksi Kalimat Menggunakan Metode Bidirectional LSTM - Studi Kasus: Indonesia dan Malaysia**" merupakan salah satu syarat yang harus dipenuhi untuk mendapatkan gelar sarjana (S1) Jurusan Teknik Informatika Universitas Islam Indonesia. Tugas akhir ini dapat terselesaikan berkat bantuan, bimbingan dan dukungan dari berbagai pihak. Sehingga penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah membantu serta mendukung dalam menyelesaikan tugas akhir ini kepada:

1. Allah subhanahu wa ta'ala atas segala nikmat dan pertolongannya.
2. Rasulullah shallallahu'alaihi wasallam yang menjadi tauladan.
3. Bapak Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
4. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., selaku dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Hendrik, S.T., M.Eng., selaku Ketua Jurusan Teknik Informatika Universitas Islam Indonesia.
6. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Jurusan Teknik Informatika (Program Sarjana), Fakultas Teknologi Industri Universitas Islam Indonesia.
7. Bapak Ahmad Fathan Hidayatullah, S.T., M.Cs., selaku dosen pembimbing, terimakasih banyak atas ilmu, bimbingan, motivasi, dukungan, waktu, tenaga, kesabaran dan masukan-masukan, sehingga penulis bisa menyelesaikan tugas akhir ini dengan baik.
8. Kepada keluarga tercinta yang selalu mendoakan, memberikan nasihat dan memberikan dukungan dari awal hingga akhir.
9. Kepada teman-teman Teknik Informatika angkatan 2015 yang memberikan dukungan.

10. Seluruh pihak yang rnernbantu dalarn bentuk apapun secara langsung atau tidak langsung yang tidak dapat saya sebutkan satu per satu.

Yogyakarta, 22 Juli 2022

( Ananda Dirgantara Rama Syahputra )

# SARI

Bahasa Indonesia dan Bahasa Malaysia merupakan dua bahasa yang memiliki kemiripan atau kedekatan terutama dalam hal penulisan, sehingga berdampak pada proses deteksi kedua bahasa tersebut. Oleh karena itu, penelitian ini bertujuan untuk membangun model deteksi kalimat menggunakan metode *Bidirectional Long Short-Term Memory* (Bi-LSTM) untuk melakukan deteksi kedua bahasa tersebut. Model Bi-LSTM yang digunakan menggunakan *word embedding* Word2Vec, dengan empat mode yaitu *Concatenation*, *Multiplication*, *Average* dan *Sum*. Data yang digunakan berupa konten atau isi dari surat kabar elektronik kedua negara tersebut. Berdasarkan hasil yang diperoleh, metode Bi-LSTM bekerja dengan baik dalam melakukan proses deteksi kedua bahasa tersebut dengan akurasi sebesar 99.71% pada mode *Sum*.

Kata kunci: Bi-LSTM, Deteksi kalimat, Mode, Model, *Word embedding*.

# GLOSARIUM

| | |
|---|---|
| *Batch size* | Ukuran atau jumlah sampel yang diproses sebelum model diperbarui. |
| *Callback* | Sebuah fungsi yang akan dijalankan apabila suatu argumen dari fungsi lain terpenuhi. |
| *Dropout* | Teknik untuk melepas beberapa node jaringan. |
| *Epoch* | Jumlah tahapan pelatihan untuk seluruh data. |
| *Input* | Unit luar yang digunakan untuk memasukkan data dari luar ke dalam yang nantinya diproses lebih lanjut. |
| *Library* | Kumpulan kode yang terkumpul dalam sebuah modul yang dapat digunakan ke program lain |
| *Output* | Menghasilkan atau menampilkan keluaran dari pengolahan data. |
| *Sequence* | Metode dimana penyimpanan dan pembacaan data yang dilakukan secara berurutan. |
| *Word embedding* | Teknik untuk mengubah sebuah kata menjadi sebuah vektor atau array yang terdiri dari kumpulan angka. |

# DAFTARISI

...

# DAFTAR GAMBAR

# BABI
# PENDAHULUAN

## 1.1 Latar Belakang

Bahasa merupakan alat yang digunakan manusia untuk saling berkomunikasi. Sebagai cara untuk menyampaikan informasi, manusia saling berkomunikasi dengan cara lisan maupun tulisan agar dapat saling berinteraksi satu sama lain. Bahasa mempunyai fungsi dan peranan yang penting dalam kehidupan manusia, dengan bahasa manusia dapat menyampaikan pikiran, perasaan, dan tujuan. Setiap bahasa memiliki variasi yang berbeda-beda, variasi tersebut disebabkan oleh faktor geografis dan faktor sosial (Waridah, 2015). Negara Indonesia dan Negara Malaysia masih dalam satu rumpun yang sama yaitu rumpun *Austronesia*, sehingga terdapat persamaan dan perbedaan makna bahasa dari masing-masing negara. Adapun contoh kalimat dalam Bahasa Malaysia yaitu "Kami sungguh seronok dapat melihat beberapa spesies ikan laut dengan lebih dekat.". Kata "Kami" dalam Bahasa Indonesia dan Bahasa Malaysia memiliki makna yang sama yaitu kelompok, kata "Seronok" dalam Bahasa Malaysia memiliki makna bahagia atau senang, sedangkan dalam Bahasa Indonesia pemakaian kata "Seronok" jarang digunakan di dalam kalimat biasanya didahului oleh kata "Tidak" dan memiliki makna negatif.

Deteksi kalimat sangat penting dalam kasus *text mining* terutama untuk mengklasifikasi data yang memiliki kemiripan bahasa yang digunakan. Adapun penelitian dalam kasus text mining yang telah dilakukan oleh (Adani, 2018) menyebutkan bahwa terdapat masalah dalam melakukan pengklasifikasian data tweet yang Berbahasa Indonesia di media sosial Twitter. Dataset yang seharusnya berisi tweet Berbahasa Indonesia tercampur dengan tweet yang Berbahasa Malaysia.

Berdasarkan permasalahan yang telah dijelaskan sebelumnya tentang kemiripan Bahasa Indonesia dan Bahasa Malaysia. Maka, akan dilakukan penelitian tentang Deteksi Kalimat Menggunakan *Metode Bidirectional LSTM* Studi Kasus: Indonesia dan Malaysia dengan menggunakan empat mode yaitu *Concatenation*, *Multiplication*, *Average* dan *Sum*. pemilihan Metode Bidirectional LSTM pada penelitian ini karena menurut (Liu & Guo, 2019) metode BiLSTM dapat memproses data secara dua arah yaitu *forward* dan *backward* untuk memahami makna setiap kata di dalam suatu kalimat.

Hasil dari penelitian ini adalah sebuah prototipe yang dapat rnendeteksi kalirnat yang digunakan (Bahasa Indonesia atau Bahasa Malaysia) dari suatu kalirnat. Dari penelitian ini, diharapkan kedepannya dapat rneningkatkan kinerja aplikasi-aplikasi yang berkaitan dengan deteksi bahasa yang rnerniliki kerniripan.

## 1.2 Rumusan Masalah

Rurnusan rnasalah yang diangkat berdasarkan latar belakang rnasalah diatas yaitu bagairnana rnelakukan deteksi kalirnat antara Bahasa Indonesia dan Bahasa Malaysia dengan rnenggunakan rnetode *Bidirectional LSTM Network*?

## 1.3 Batasan Masalah

Batasan rnasalah dari penelitian ini adalah sebagai berikut:

a. Data yang digunakan berasal dari surat kabar elektronik.
b. Surat kabar elektronik yang digunakan berasal dari Negara Indonesia dan Negara Malaysia.
c. Bagian isi dari surat kabar elektronik yang hanya akan digunakan sebagai data deteksi.
d. Sistern hanya akan rnelakukan deteksi dari rnasukan data yang berupa kalirnat dan rnernberikan keluaran yaitu bahasa yang digunakan (Indonesia atau Malaysia) dari kalirnat tersebut.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini yaitu rnengaplikasikan rnetode *Bidirectional LSTM Network* untuk rnelakukan deteksi kalirnat antara Bahasa Indonesia dan Bahasa Malaysia.

## 1.5 Manfaat Penelitian

Manfaat yang didapatkan dari penelitian ini adalah rnengetahui rnetode Bidirectional LSTM Network dalarn rnelakukan deteksi kalirnat yang rnerniliki kerniripan dapat diterapkan dengan baik atau tidak.

## 1.6 Sistematika Penulisan

Sisternatika penulisan pada penelitian ini adalah sebagai berikut:

a. **BAB I PENDAHULUAN**

   Bab ini berisi tentang latar belakang dari permasalahan yang dihadapi, rurnusan rnasalah, batasan rnasalah, tujuan penelitian, rnanfaat penelitian, dan sisternatika penelitian.

**b. BAB II LANDASAN TEORI**

Bab ini memberikan penjelasan mengenai penelitian yang sudah dilakukan sebelumnya dan juga dasar teori.

c. **BAB III METODOLOGI PENELITIAN**

Bab ini berisi tentang seluruh tahapan-tahapan yang dilakukan dalam melakukan penelitian deteksi kalimat Bahasa Indonesia dan Bahasa Malaysia.

d. **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil penelitian yang telah dilakukan, serta memuat pembahasan dari hasil penelitian tersebut.

e. **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan yang didapatkan dari penelitian yang telah dilakukan, serta saran untuk perbaikan pada penelitian selanjutnya.

# BAB II LANDASAN TEORI

## 2.1 Penelitian Terkait

Tabel 2.1 menampilkan daftar penelitian-penelitian yang telah dilakukan sebelumnya dan menjadi acuan pada penelitian ini.

Tabel 2. 1 Daftar penelitian sebelumnya

| Judul Penelitian | Metode | Hasil |
|---|---|---|
| *Automatic Identification of Close Languages - Case study: Malay and Indonesian (Ranaivo-Malancon, Bali. 2006)* | N-Gram | Penelitian ini membangun sebuah sistem identifikasi bahasa untuk menentukan sebuah teks dituliskan dalam Bahasa Indonesia atau dengan Bahasa Malaysia. Metode yang digunakan dalam penelitian menggunakan N-Gram. Hasil dari penelitian ini adalah mengidentifikasi teks yang memiliki panjang 12 hingga 90 kalimat dan 147 hingga 180 kalimat. Jumlah error yang diperoleh untuk Bahasa Indonesia yaitu 35 error sedangkan untuk Bahasa Malaysia error yang diperoleh sebesar 49 error. |
| *Automatic Identification of Closely-related Indian Languages: Resources and Experiments (Kumar, Ritesh; Lahiri, Bornini. 2018)* | N-Gram | Penelitian ini bertujuan untuk membangun sebuah sistem identifikasi bahasa otomatis yang memiliki kemiripan atau kedekatan pada 5 bahasa yang digunakan di India yaitu Awadhi, Bhojpuri, Braj, Hindi dan Magahi. Metode yang digunakan pada penelitian ini menggunakan N-Gram dengan tiga fitur yaitu *Character n-gram Features, Word n-gram features* dan *Combined features*. Hasil akurasi tertinggi yang diperoleh yaitu lebih dari 96% pada fitur *combined*. |
| *A Deep Neural Network Sentence Level Classification Method with Context Information (Xingyi, Song; Johann, Petrak. 2018)* | LSTM | Penelitian ini mengklasifikasikan teks dialog di setiap kalimatnya (level kalimatnya) untuk menentukan satu di antara empat label yaitu *Anger, Excitement, Neutral* dan *Sadness*. Metode yang digunakan pada penelitian ini adalah *Long Short-Term Memory* (LSTM). Hasil akurasi tertinggi sebesar 79% untuk label *Anger* dan untuk akurasi terendah sebesar 66% untuk label Sadness. |
| *Densely Connected Bidirectional LSTM with Applications* | Bi-LSTM | Penelitian ini menggunakan metode *Densely Connected Bidirectional Long Short-Term Memory* (DC-Bi-LSTM) untuk mengklasifikasi kalimat. DC-Bi-LSTM merupakan tumpukan LSTM dimana *layer* pertama membaca *input*, |

| | | |
|---|---|---|
| *to Sentence Classification (Ding, Zixiang; Kia, Rui; Yu, Jianfei; Li, Xiang; Yang, Jian. 2018)* | | *layer* kedua, ketiga dan seterusnya membaca rangkaian memori. Penelitian ini menggunakan lima data yang digunakan yaitu *Movie Review, Stanford Sentiment Treebank, Stanford Sentiment Treebank Binary Mode, Subjectivity Dataset* dan *TREC*. Hasil akurasi yang diperoleh yaitu *Movie Review* memperoleh 82.8%, *Stanford Sentiment Treebank memperoleh 51.9%, Stanford Sentiment Treebank Binary Mode* memperoleh 89.7%, *Subjectivity Dataset* memperoleh 94.5% dan *TREC* memperoleh 95.6%. |
| *When Sparse Traditional Models Outperform Dense Neural Networks: The Curious Case of Discriminating between Similar Languages (Medvedeva, Maria; Kroon, Martin; Plank, Barbara. 2017)* | SVM | Penelitian ini mengimplementasikan metode *Support Vector Machines* (SVM) dalam proses identifikasi bahasa yang memiliki kemiripan. Data yang digunakan yaitu 14 bahasa yang dikelompokkan dalam 6 grup, salah satu grupnya adalah Bahasa Indonesia dan Malaysia. Hasil akurasi identifikasi Bahasa Indonesia dan Bahasa Malaysia yang didapatkan untuk metode SVM yaitu 98% |

Berbeda dari penelitian-penelitian yang telah dilakukan sebelumnya, penelitian deteksi kalimat antara Bahasa Indonesia dan Bahasa Malaysia ini menggunakan data dari isi surat kabar elektronik kedua negara tersebut, pemilihan data isi surat kabar karena sesuai dengan kaidah tata bahasa kedua negara tersebut. Metode yang digunakan adalah Bi-LSTM dengan fitur *Word Embedding Word2Vec* yang digunakan dalam proses *embedding* untuk merepresentasikan kata menjadi vektor dan juga empat mode untuk mengkombinasikan nilai *dari forward output* dan *backward outputs* pada Bi-LSTM yaitu *Concatenation, Multiplication, Average* serta *Sum*. Berbeda dengan LSTM biasa yang hanya bisa mengakses informasi dari sebelumnya saja, Bi-LSTM dapat mengakses informasi sebelum dan setelahnya, karena fitur pada Bi-LSTM lebih panjang, maka informasi yang diproses akan lebih detail pada proses *feed forward neural* nya.

## 2.2 Dasar Teori

### 2.2.1 *Text Mining*

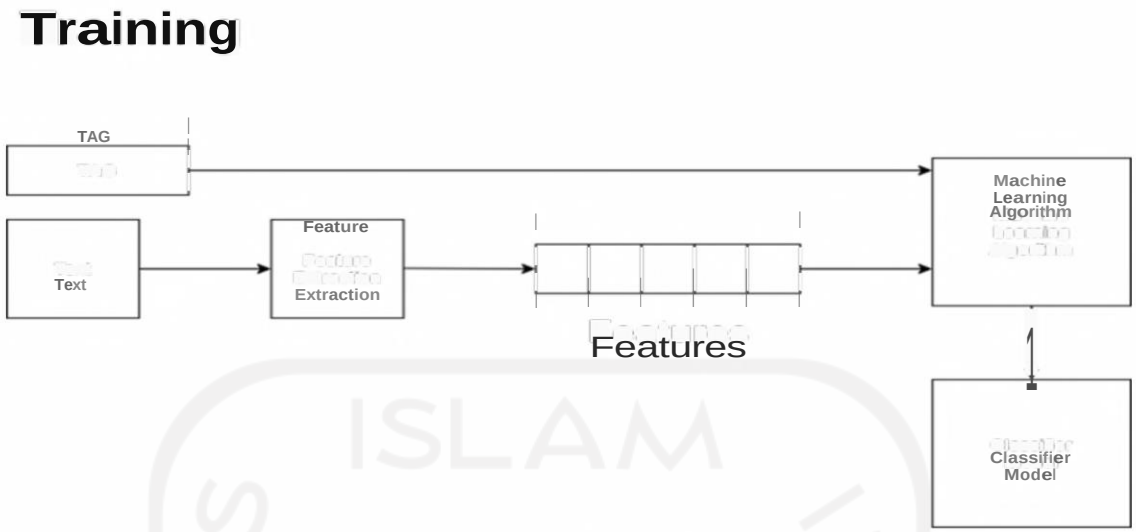
*Text mining*, juga dikenal sebagai *data mining* atau *knowledge discovery* dari *textual database*, mengacu pada proses mencari dan menemukan pola penting dari dokumen teks (Tan, 1999). *Text mining* telah menjadi bidang penelitian yang menarik karena mencoba untuk menemukan informasi berharga dari text yang tidak terstruktur.

Text tidak terstruktur yang berisi sejumlah besar informasi tidak bisa begitu saja digunakan digunakan untuk proses lebih lanjut oleh komputer, oleh karena, metode pemrosesan, algoritma dan teknik yang tepat sangat penting untuk memperoleh informasi berharga (Dang & Ahmad, 2014). *Text mining* telah menjadi area penting dalam penelitian, sejumlah besar informasi disimpan di tempat yang berbeda dan tidak terstruktur (Vidhya & Aghila, 2010).

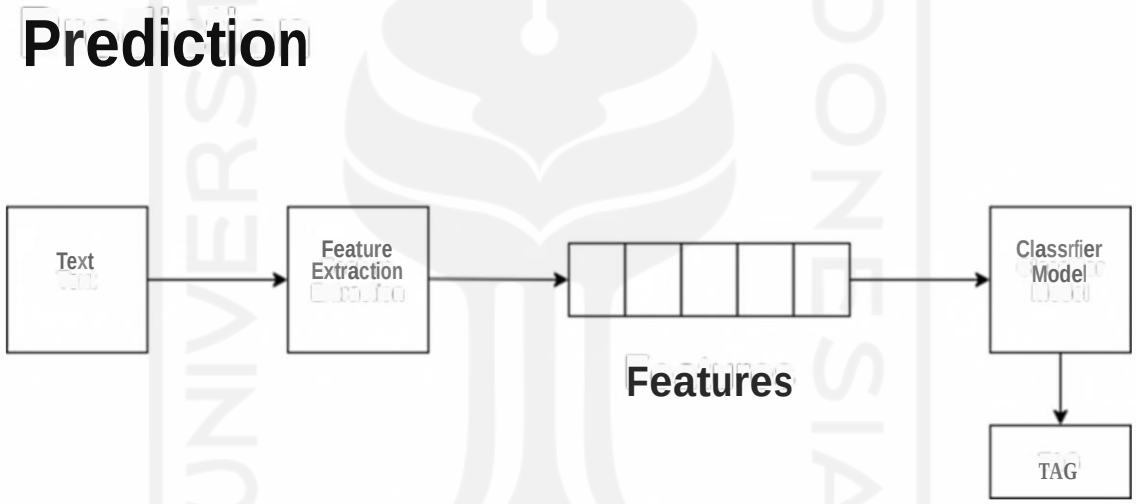
### 2.2.2 *Text Classification*

Klasifikasi teks merupakan fungsi penting dalam *Natural Language Processing* yang bertujuan untuk menentukan sebuah dokumen ke dalam satu atau beberapa kategori seperti sentimen analisis, deteksi spam dan klasifikasi dokumen (Zhou et al., 2016). Klasifikasi teks dapat dilakukan dengan dua cara yaitu manual dan otomatis, klasifikasi secara otomatis dapat dilakukan dengan pendekatan *Machine Learning Base System*.

Klasifikasi teks dengan menggunakan *machine learning* membuat klasifikasi berdasarkan pengamatan sebelumnya, dengan menggunakan data yang telah diberi label. Langkah yang dilakukan dalam klasifikasi teks adalah *feature extraction*, yaitu metode yang digunakan untuk mengubah setiap teks menjadi angka dalam bentuk vektor, lalu algoritma *machine learning* dilengkapi dengan *data training* yang terdiri *dari feature sets* (vektor dari setiap teks) dan *tag* (label: politik, olahraga, hiburan) untuk membangun model klasifikasi (Jambukia et al., 2018). Proses ini disebut sebagai *training*. Gambar 2.1 menunjukkan proses *training* klasifikasi teks.




Gambar 2. 1 Proses *training* klasifikasi teks

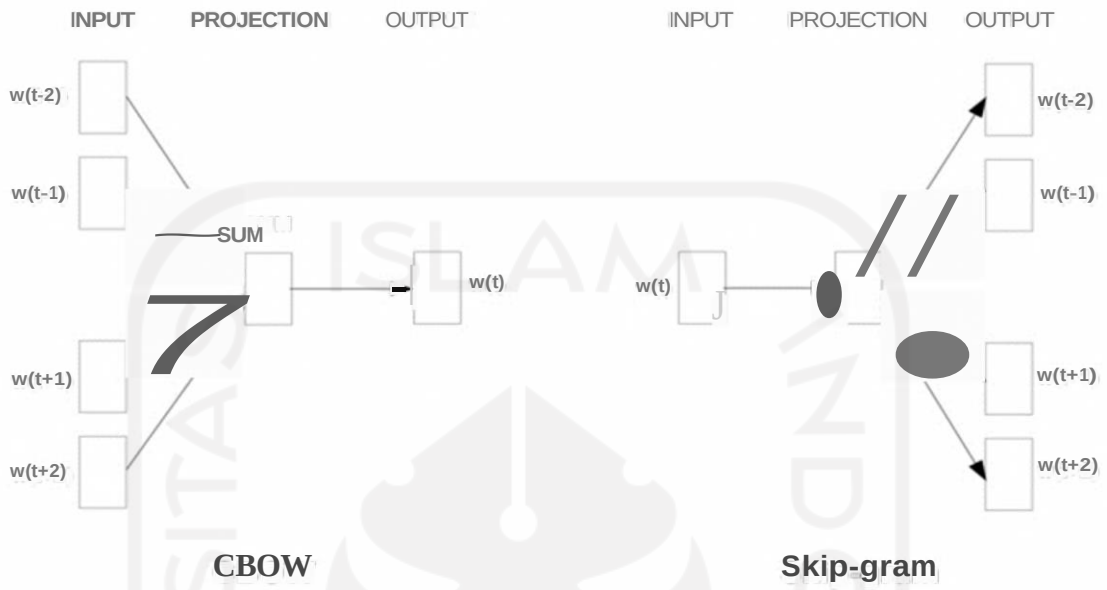



Gambar 2. 2 Proses *prediction* klasifikasi teks

### 2.2.3 Word2Vec

Word2Vec merupakan model yang digunakan dalam proses *word embeddings* atau merepresentasikan kata menjadi vektor. Representasi vektor tersebut memiliki hubungan terhadap kata-kata yang berkaitan melalui proses *training*. Sebagai contoh vektor representasi kata dari "Indonesia" akan berdekatan dengan vektor representasi kata "Jakarta" dan juga vektor representasi kata dari "Malaysia" akan berdekatan dengan vektor representasi kata "Kuala Lumpur". Model Word2Vec akan memahami bahwa "Indonesia" dan "Jakarta" memiliki hubungan yang sama dengan "Malaysia" dan "Kuala Lumpur" yaitu negara dan ibukotanya.

Terdapat dua model arsitektur yang dapat digunakan pada Word2Vec, yaitu *CBOW* dan *Skip-Gram*. Kedua model tersebut dapat dilihat pada gambar 2.3 berikut.

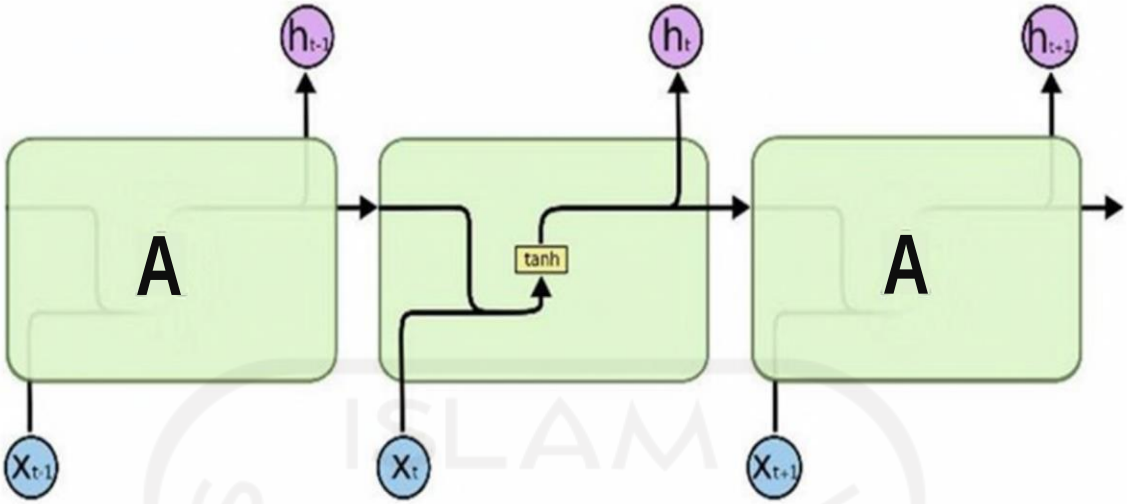



Gambar 2.3 Arsitektur *Word2Vec*. *CBOW* dan *Skip-gram*

Pada model CBOW, word2vec menggunakan kata-kata yang ada di sebelah kiri dan kanan kata target dan dibatasi dengan *window* (jarak maksimum antara kata saat ini dan kata yang diprediksi dalam sebuah kalimat) untuk memprediksi kata target tersebut. Sedangkan *skip-gram* menggunakan sebuah kata untuk memprediksi kata-kata yang ada di sebelah kiri dan kanan kata tersebut yang dibatasi oleh *window*.

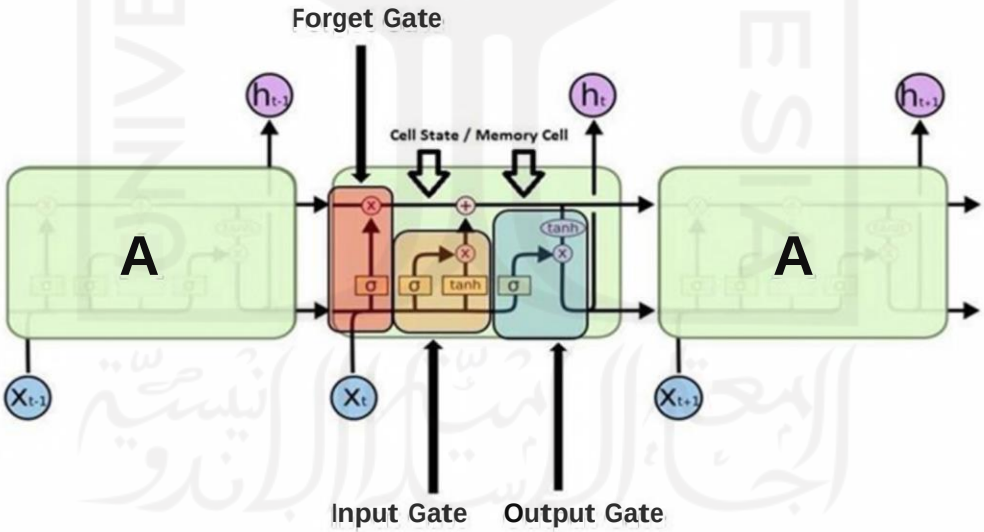
### 2.2.4 *Long Short-Term Memory* (LSTM)

*Long Short-Term Memory* (LSTM) merupakan variasi dari *Recurrent Neural Network (RNN)* (Hochreiter & Schmidhuber, 1997). Prediksi pada RNN dilakukan secara sequence, diman *output* dari hasil komputasi sebelumnya menjadi *input* untuk proses komputasi berikutnya. Gambar 2.4 menunjukan alur pada RNN.




Gambar 2. 4 Alur RNN *network*

LSTM memberi RNN fitur lebih dalam pengendalian memori, aspek ini mengendalikan seberapa penting *input* yang digunakan untuk membentuk memori baru (Hassan & Mahmood, 2017).



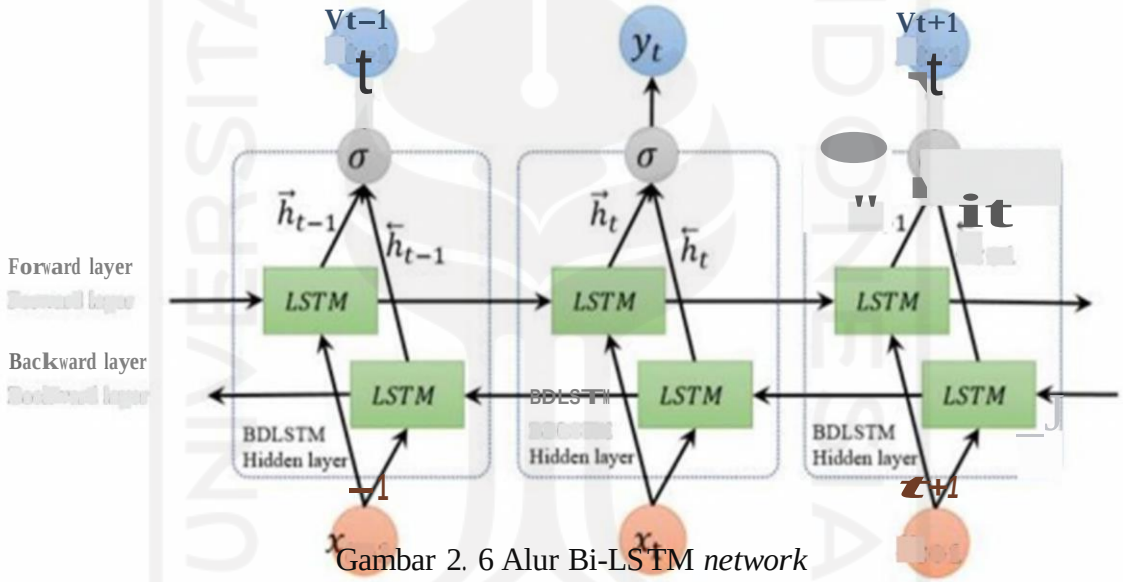

Gambar 2. 5 Alur LSTM *network*

Dapat dilihat Gambar 2.5, LSTM memiliki *cell state* atau *memory cell* dan *gate* unit yaitu *input gate*, *forget gate* dan *output gate* yang terdapat pada setiap cell. Cell state atau *memory cell* digunakan untuk menyimpan informasi yang akan diteruskan ke proses berikutnya tanpa

melewati fungsi aktivasi. *Forget gate* dengan fungsi aktivasi sigmoid (nilai 0 hingga 1) menentukan apakah input dari informasi neuron sebelumnya diteruskan atau tidak. *Input gate* berfungsi untuk melakukan pembaruan pada *memory cell* dengan melewati dua fungsi aktivasi sigmoid dan tanh. *Output gate* merupakan nilai output dari cell yang akan menjadi input untuk cell berikutnya.

### 2.2.5 *Bidirectional Long Short-Tearm Memory* (Bi-LSTM)

*Bidirectional Long Short-Term Memory* (BiLSTM) memproses data *sequence* secara *forward* dan *backward directions* dengan dua *hidden layers* yang terpisah (Cui et al., 2018). BiLSTM menggabungkan dua *hidden layer* ke dalam *output layer* yang sama.



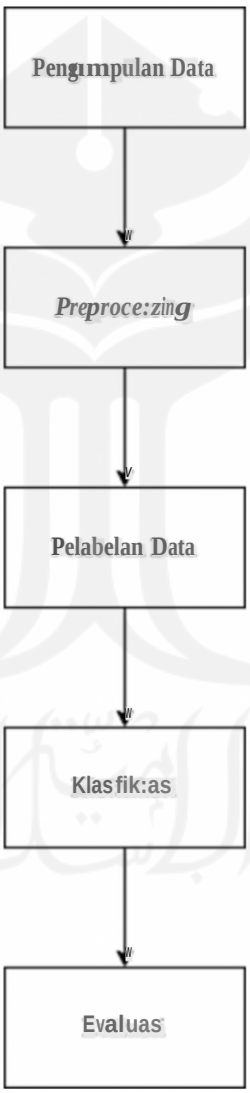

Gambar 2. 6 Alur Bi-LSTM *network*

Struktur lapisan BiLSTM berisi *layer* LSTM *forward* dan *backward* yang diilustrasikan Gambar 2.6. Dengan layer arsitektur tersebut model BiLSTM dapat mempelajari data yang ada di masa lalu *(past)* dan data di masa mendatang *(future)*. *Forward layer* yang berfungsi untuk merepresentasikan informasi sebelumnya sedangkan *backward layer* berfungsi untuk merepresentasikan informasi setelahnya. *Output* dari *forward* dan *backward layer* dihitung dengan empat mode berupa *concatenating function, summation function, average function* atau *multiplication function*.

# BABIII
# METODOLOGI PENELITIAN

## 3.1 Langkah-Langkah Penelitian

Langkah-langkah penelitian yang dilakukan dalarn deteksi kalirnat Bahasa Indonesia dan Bahasa Malaysia dengan rnenggunakan rnetode Bi-LSTM adalah pengurnpulan data, *preprocessing*, pelabelan data, klasifikasi dan evaluasi. Garnbar 3.1 rnenunjukan langkah• langkah rnetodologi penelitian yang dilakukan.



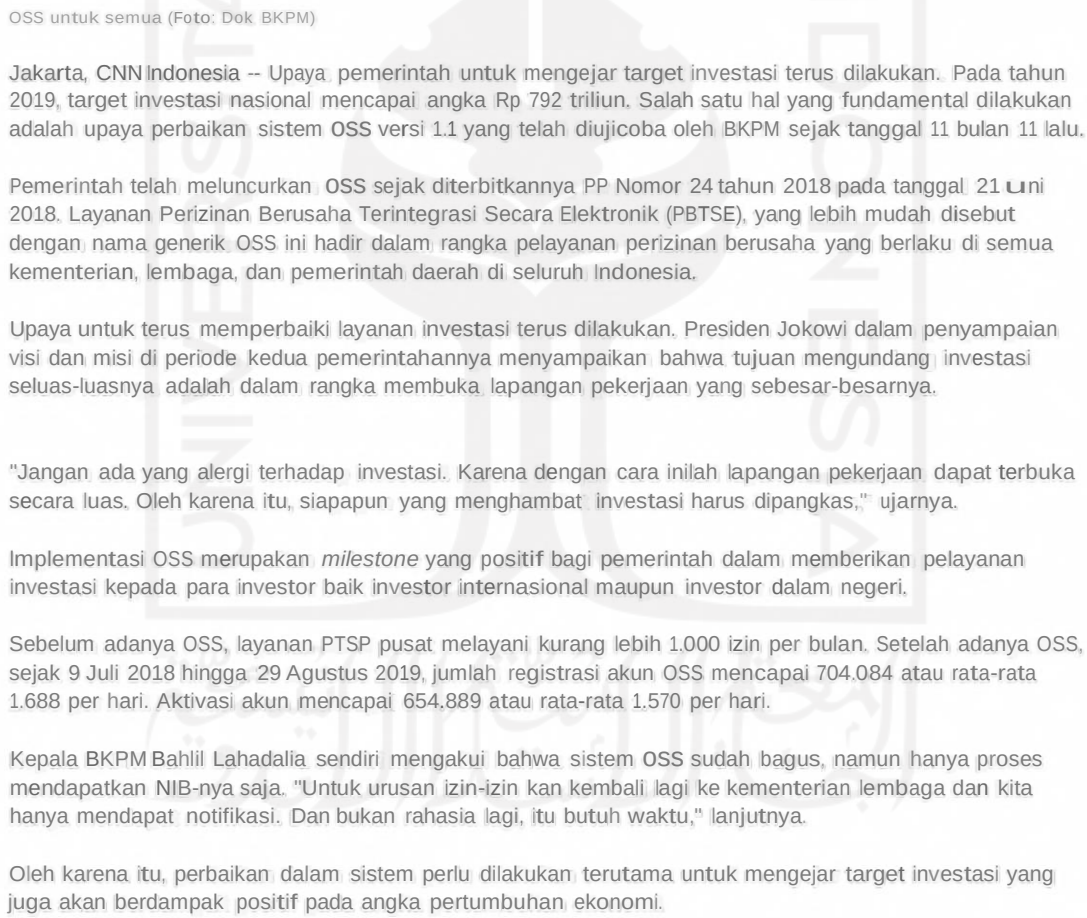

Garnbar 3. 1 Langkah-langkah penelitian

## 3.2 Uraian Metodologi

Berdasarkan langkah-langkah penelitian yang telah diperlihatkan Gambar 3.1 berikut merupakan penjelasan setiap langkahnya:

### 3.2.1 Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan dengan menggunakan data dari isi surat kabar elektronik Negara Indonesia dan Malaysia. Surat kabar elektronik untuk Negara Indonesia yaitu CNN, Kompas dan Tribun. Surat kabar elektronik Negara Malaysia yaitu Harian Metro dan Sinar Harian. Gambar 3.2 merupakan contoh isi surat kabar CNN Indonesia yang datanya akan digunakan untuk deteksi kalimat.



OSS untuk semua (Foto: Dok. BKPM)

Jakarta, CNN Indonesia -- Upaya pemerintah untuk mengejar target investasi terus dilakukan. Pada tahun 2019, target investasi nasional mencapai angka Rp 792 triliun. Salah satu hal yang fundamental dilakukan adalah upaya perbaikan sistem OSS versi 1.1 yang telah diujicoba oleh BKPM sejak tanggal 11 bulan 11 lalu.

Pemerintah telah meluncurkan OSS sejak diterbitkannya PP Nomor 24 tahun 2018 pada tanggal 21 Juni 2018. Layanan Perizinan Berusaha Terintegrasi Secara Elektronik (PBTSE), yang lebih mudah disebut dengan nama generik OSS ini hadir dalam rangka pelayanan perizinan berusaha yang berlaku di semua kementerian, lembaga, dan pemerintah daerah di seluruh Indonesia.

Upaya untuk terus memperbaiki layanan investasi terus dilakukan. Presiden Jokowi dalam penyampaian visi dan misi di periode kedua pemerintahannya menyampaikan bahwa tujuan mengundang investasi seluas-luasnya adalah dalam rangka membuka lapangan pekerjaan yang sebesar-besarnya.

"Jangan ada yang alergi terhadap investasi. Karena dengan cara inilah lapangan pekerjaan dapat terbuka secara luas. Oleh karena itu, siapapun yang menghambat investasi harus dipangkas," ujarnya.

Implementasi OSS merupakan *milestone* yang positif bagi pemerintah dalam memberikan pelayanan investasi kepada para investor baik investor internasional maupun investor dalam negeri.

Sebelum adanya OSS, layanan PTSP pusat melayani kurang lebih 1.000 izin per bulan. Setelah adanya OSS, sejak 9 Juli 2018 hingga 29 Agustus 2019, jumlah registrasi akun OSS mencapai 704.084 atau rata-rata 1.688 per hari. Aktivasi akun mencapai 654.889 atau rata-rata 1.570 per hari.

Kepala BKPM Bahlil Lahadalia sendiri mengakui bahwa sistem OSS sudah bagus, namun hanya proses mendapatkan NIB-nya saja. "Untuk urusan izin-izin kan kembali lagi ke kementerian lembaga dan kita hanya mendapat notifikasi. Dan bukan rahasia lagi, itu butuh waktu," lanjutnya.

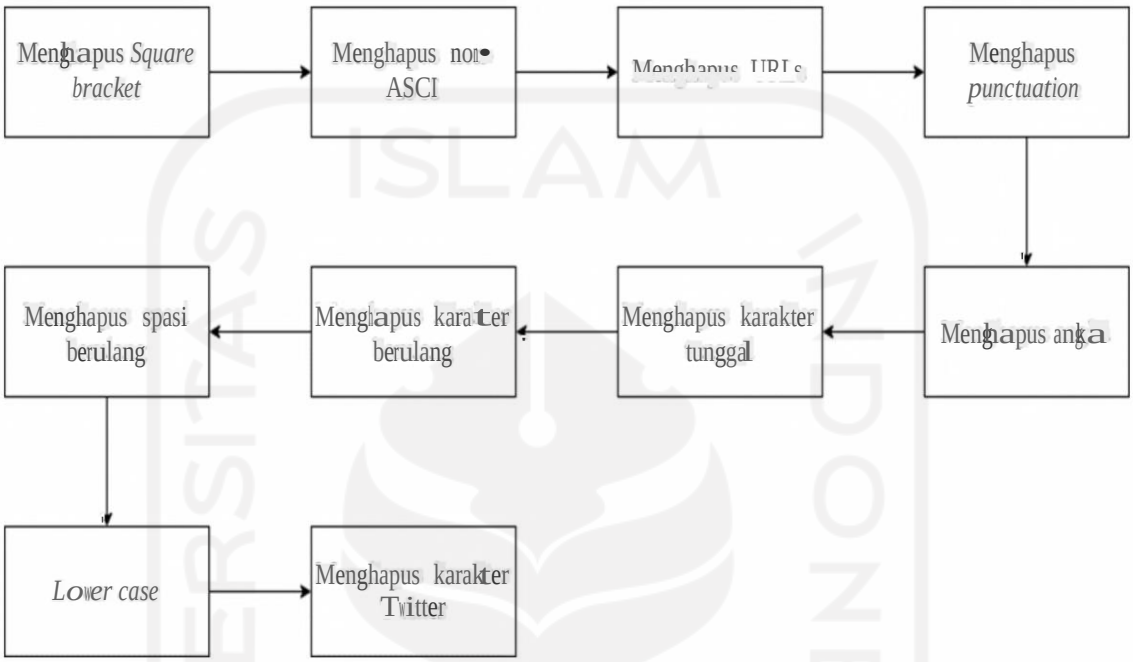
Oleh karena itu, perbaikan dalam sistem perlu dilakukan terutama untuk mengejar target investasi yang juga akan berdampak positif pada angka pertumbuhan ekonomi.

Gambar 3. 2 Isi berita CNN Indonesia

### 3.2.2 *Preprocessing*

Preprocessing merupakan langkah yang dilakukan untuk membersihkan data yang dinilai tidak memberikan pengaruh besar terhadap proses klasifikasi. Langkah *preprocessing* yang

dilakukan yaitu: menghapus *square bracket*, menghapus non-ASCII, menghapus URL, menghapus *punctuation*, menghapus angka, menghapus karakter tunggal, menghapus karakter berulang, menghapus spasi berulang, menghapus *lower case* dan karakter Twitter. Gambar 3.3 merupakan proses *preprocessing* yang dilakukan.




Gambar 3. 3 *Tahapan preprocessing*

a. Menghapus *Square bracket*

Menghapus *square bracket* atau kurung siku pada surat kabar elektronik berfungsi sebagai label atau tag isi suatu berita yang dimuat seperti label gambar atau video. Tabel 3.1 merupakan contoh menghapus *square bracket*.

Tabel 3. 1 Menghapus *square bracket*

| Sebelum | Sesudah |
|---|---|
| [Gambas: Video CNN] Catatan Redaksi: Judul artikel ini diubah menjadi Pertamina Tawarkan Skema Kilang Balikpapan ke Saudi Aramco terkait dengan klarifikasi dari pihak terkait. | Catatan Redaksi: Judul artikel ini diubah menjadi Pertamina Tawarkan Skema Kilang Balikpapan ke Saudi Aramco terkait dengan klarifikasi dari pihak terkait. |
| [Gambas: Video CNN] Faisal mengatakan sinkronisasi kebijakan saat ini sudah mendapatkan dukungan dari BI. | Faisal mengatakan sinkronisasi kebijakan saat ini sudah mendapatkan dukungan dari BI. |

b. Menghapus non-ASCII

ASCII atau *American Standard Code for Information Interchange* rnerupakan standar intemasional dalarn pengkodean huruf dan sirnbol. Menghapus non-ASCII bertujuan untuk rnenghapus karakter yang tidak terdapat pada standar ASCII dan rnengurangi terjadinya *error* pada *script* Python.

c. Menghapus URL

Menghapus URL dilakukan karena penggunaan URL tidak rnernberikan darnpak yang besar dalarn proses klasifikasi. Contoh rnenghapus URL ditunjukkan Tabel 3.2.

Tabel 3. 2 Menghapus URL

| Sebelum | Sesudah |
|---|---|
| Website www.bni-life.co.id juga difungsikan sebagai sarana sosialisasi produk maupun pelatihan. | Website juga difungsikan sebagai sarana sosialisasi produk maupun pelatihan. |
| Untuk maklumat lanjut bolehlah singgah di www.zootaiping.gov.my supaya kunjungan anda ke Taiping yang baru• baru ini dinamakan sebagai Destinasi Hijau oleh badan bukan kerajaan (NGO) pelancongan akan lebih bermakna. | Untuk maklumat lanjut bolehlah singgah di supaya kunjungan anda ke Taiping yang baru-baru ini dinamakan sebagai Destinasi Hijau oleh badan bukan kerajaan (NGO) pelancongan akan lebih bermakna. |

d. Menghapus *punctuation*

*Punctuation* atau tanda baca pada suatu kalirnat tidak rnernberikan darnpak yang besar dalarn proses klasifikasi. Oleh karena itu punctuation perlu dihapus. Tabel 3.3 rnerupakan contoh rnenghapus *punctuation*.

Tabel 3. 3 Menghapus *punctuation*

| Sebelum | Sesudah |
|---|---|
| Kita juga sudah kerja sama dengan e• commerce yaitu LinkAja untuk penjualan," ujar Shadiq.Selain itu, BNI Life terus berupaya meningkatkan kinerja serta memberikan pelayanan melalui layanan Eazy Claim. | Kita juga sudah kerja sama dengan e commerce yaitu LinkAja untuk penjualan ujar Shadiq Selain itu BNI Life terus berupaya meningkatkan kinerja serta memberikan pelayanan melalui layanan Eazy Claim |
| Antara tujuh permainan baru diperkenalkan, Power Surge, Disco! Sky Tower, Spin Crazy, Bumper, Boo Boo Bump, Tea Cup dan Balloon Racem katanya. | Antara tujuh permainan baru diperkenalkan Power Surge Disco Sky Tower Spin Crazy Bumper Boo Boo Bump Tea Cup dan Balloon Racem katanya |

e. Menghapusangka

Digit atau angka dihapuskan karena tidak rnerniliki pengaruh yang besar terhadap klasifikasi. Tabel 3.4 rnerupakan contoh rnenghapus angka.

Tabel 3. 4 Menghapus angka

| Sebelum | Sesudah |
|---|---|
| Setelah adanya OSS, sejak 9 Juli 2018 hingga 29 Agustus 2019, jumlah registrasi akun OSS mencapai 704.084 atau rata-rata 1.688 per hari. | Setelah adanya OSS, sejak Juli hingga Agustus jumlah registrasi akun OSS mencapai atau rata-rata per hari. |
| Aktivasi akun mencapai 654.889 atau rata-rata 1.570 per hari. Kepala BKPM Bahlil Lahadalia sendiri mengakui bahwa sistem OSS sudah bagus, namun hanya proses mendapatkan NIB-nya saja. | Aktivasi akun mencapai atau rata-rata per hari. Kepala BKPM Bahlil Lahadalia sendiri mengakui bahwa sistem OSS sudah bagus, namun hanya proses mendapatkan NIB-nya saja. |

f. Menghapus karakter tunggal

Karakter tunggal disebabkan karena kesalahan penulisan atau penerapan dari regular expression. Tabel 3.5 rnerupakan contoh rnenghapus karakter tunggal.

Tabel 3. 5 Menghapus karakter tunggal

| Sebelum | Sesudah |
|---|---|
| Namun dari tahun k ke tahun, pengumpulan zakat Baznas tak pernah bisa mencapai atau bahkan mendekati angka tersebut. | Namun dari tahun ke tahun, pengumpulan zakat Baznas tak pernah bisa mencapai atau bahkan mendekati angka tersebut. |
| Kemudian nama-nama itu a akan diserahkan ke Presiden Joko Widodo untuk dipilih bersama DPR | Kemudian nama-nama itu akan diserahkan ke Presiden Joko Widodo untuk dipilih bersama DPR |

g. Menghapus karakter berulang

Karakter berulang disebabkan kesalahan penulisan atau penerapan dari *regular expression*. Tabel 3.6 rnerupakan contoh rnenghapus karakter berulang.

Tabel 3. 6 Menghapus karakter berulang

| Sebelum | Sesudah |
|---|---|
| Kita ingin buktikan dengan zzzzakat ini bisa perbaiki ekonomi umat Islam yang ada di Indonesia ini. | Kita ingin buktikan dengan zakat ini bisa perbaiki ekonomi umat Islam yang ada di Indonesia ini. |
| Menteri Yasonna pun memberiiii tenggat 14 hari bagi kedua pihak untuk | Menteri Yasonna pun memberi tenggat 14 hari bagi kedua pihak untuk |

| | |
|---|---|
| menyelesaikan persoalan tersebut secara business to business. | menyelesaikan persoalan tersebut secara business to business. |

h. Menghapus spasi berulang

Spasi berulang disebabkan kesalahan penulisan atau penerapan dari *regular expression*. Tabel 3.7 rnerupakan contoh rnenghapus spasi berulang.

Tabel 3. 7 Menghapus spasi berulang

| Sebelum | Sesudah |
|---|---|
| Jakarta, CNN Indonesia<br><br>Menteri Keuanganmeminta Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi menginisiasi revisi Peraturan Pemerintah (PP) Nomor 53 Tahun 2010 tentang Disiplin Pegawai Negeri Sipil (). Perubahan perlu dilakukan karena ia memandang beleid itu hanya berisi hukuman bagi PNS yang melanggar aturan. | Jakarta, CNNindonesia Menteri Keuanganmeminta Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi menginisiasi revisi Peraturan Pemerintah **(PP)** Nomor 53 Tahun 2010 tentang Disiplin Pegawai Negeri Sipil (). Perubahan perlu dilakukan karena ia memandang beleid itu hanya berisi hukuman bagi PNS yang melanggar aturan. |
| ~~Dalarn revisi~~<br><br>nantinya perlu diatur PNS yang rnerniliki kinerja baik perlu diberikan penghargaan lebih. | ~~Dalarn revisi nantinya perlu diatur~~ PNS yang rnerniliki kinerja baik perlu diberikan penghargaan lebih. |

1. *Lower case*

*Lower case* adalah proses untuk rnengubah seluruh teks rnenjadi huruf kecil. Tabel 3.8 rnerupakan contoh rnengubah seluruh teks rnenjadi huruf kecil.

Tabel 3. 8 *Lower case*

| Sebelum | Sesudah |
|---|---|
| ~~Bila pemerintah memberikan apresiasi~~ yang pantas, Sri Mulyani optimistis bisa meminimalisir jumlah pegawai yang melakukan korupsi. | ~~bila pemerintah memberikan apresiasi~~ yang pantas, sri mulyani optimistis bisa meminimalisir jumlah pegawai yang melakukan korupsi. |
| ~~Kemudian, hukuman sedang terdiri dari~~ penundaan kenaikan gaji selama satu tahun, penundaan kenaikan pangkat selama satu tahun, dan penurunan pangkat setingkat lebih rendah selama satu tahun. Sementara, PNS yang mendapatkan hukuman berat bisa saja diturunkan | ~~kemudian, hukuman sedang terdiri dari~~ penundaan kenaikan gaji selama satu tahun, penundaan kenaikan pangkat selama satu tahun, dan penurunan pangkat setingkat lebih rendah selama satu tahun. sementara, pns yang mendapatkan hukuman berat bisa saja diturunkan |

| | |
|---|---|
| pangkatnya setingkat lebih rendah selama tiga tahun. | pangkatnya setingkat lebih rendah selama tiga tahun. |

j. Menghapus karakter Twitter

Karakter Twitter atau Instagram seperti *mention* (@usemame), *hashtag* (#hastag) dan *retweet* (RT) muncul pada berita dengan kategori hiburan. Karena tidak memiliki dampak yang besar terhadap proses klasifikasi maka dihapus. Tabel 3.9 merupakan contoh menghapus karakter khusus yang ada pada twitter atau di Instagram.

Tabel 3. 9 Menghapus karakter Twitter

| Sebelum | Sesudah |
|---|---|
| Ia percaya diri untuk langsung menggelar konser. Melalui akun Instagram @wordfangs, Hindia memberikan kisi-kisi konser yang diadakan di Studio Palem, Kemang, Jakarta Selatan. | Ia percaya diri untuk langsung menggelar konser. Melalui akun Instagram Hindia memberikan kisi-kisi konser yang diadakan di Studio Palem, Kemang, Jakarta Selatan. |
| #MnetApologizeToEXO punya maknalain, tulis akun@zkaiytlin. Seorang fan menyebut acara penghargaan musik Mnet itu bak seekor ular. | punya maknalain, tulis seorang akun fan menyebut acara penghargaan musik Mnet itu bak seekor ular. |

### 3.2.3 Pelabelan Data

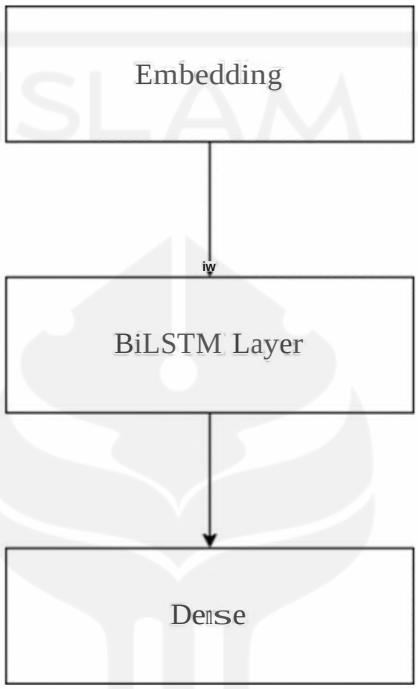
Proses pelabelan data dilakukan setelah data yang dikumpulkan berupa paragraf kemudian diubah ke bentuk kalimat. Kalimat-kalimat yang telah dikelompokan sesuai dengan bahasa yang digunakan kemudian diberi label dengan bantuan suatu fungsi yang dibangun dengan menggunakan Python.

### 3.2.4 Klasifikasi

Proses klasifikasi Bahasa Indonesia dan Malaysia dilakukan menggunakan metode Bi• LSTM dengan *mode sum, average, multiplication* dan *concatenation (default).* Mode pada Bi• LSTM berfungsi untuk mengkombinasikan forward dan *backward* output sebelum diteruskan pada layer berikutnya. Fungsi mode *sum* yaitu menjumlahkan output, fungsi mode *average* yaitu mengambil rata-rata output, fungsi mode *multiplication* yaitu mengalikan output dan fungsi mode *concatenation* yaitu output digabungkan bersamaan, memberikan dua kali lipat jumlah output untuk layer berikutnya.

Proses membangun model Bi-LSTM menggunakan tiga layer yaitu embedding layer, Bi• LSTM layer dan dense layer. Embedding layer merupakan layer yang berfungsi mengubah kata menjadi sebuah vektor atau array yang terdiri dari kumpulan angka. Word embedding yang

digunakan pada model Bi-LSTM ini adalah Word2Vec. Bi-LSTM layer merupakan layer yang digunakan untuk menjalankan metode klasifikasi Bi-LSTM. Dense layer merupakan layer yang berfungsi sebagai output layer. Gambar 3.4 menunjukkan model arsitektur dengan menggunakan metode klasifikasi Bi-LSTM.




Gambar 3. 4 Model arsitektur Bi-LSTM

### 3.2.5 Evaluasi

Evaluasi bertujuan untuk mengukur kinerja dari model yang telah dibangun. Metode yang digunakan dalam evaluasi model Bi-LSTM adalah confusion matrix. Confusion matrix berisi informasi tentang klasifikasi aktual dan prediksi yang telah dilakukan oleh sistem klasifikasi. Confusion matrix memiliki dua dimensi, satu dimensi diindeks oleh kelas sebenamya dari suatu objek, yang lain diindeks oleh kelas yang diprediksi oleh pengklasifikasi (Deng et al., 2016).

Evaluasi confusion matrix dilakukan untuk memperoleh nilai accuracy, precision, recall, dan FI-Score. Accuracy merupakan tingkat prediksi benar dengan keseluruhan data. Precision merupakan tingkat keakuratan data yang diinginkan dengan hasil prediksi yang diberikan. Recall merupakan tingkat keberhasilan model dalam menemukan kembali sebuah informasi.

FI-Score merupakan perbandingan rata-rata precision dan recall. Confusion matrix ditunjukkan Tabel 3.10.

Tabel 3. 10 Confusion matrix

| | | *Prediction* | |
|---|---|---|---|
| | | *Positive* | *Negative* |
| *Actual* | *Positive* | TP | FP |
| | *Negative* | FN | TN |

Nilai *accuracy*, *precision*, *recall*, dan FI-*Score* diperoleh dengan persamaan sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \quad (3.1)$$

$$Precision = \frac{TP}{TP + FP} * 100\% \quad (3.2)$$

$$Recall = \frac{TP}{TP + FN} * 100\% \quad (3.3)$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} * 100\% \quad (3.4)$$

Keterangan:

*True Positive* (TP) : jumlah data positif yang terklasifikasi dengan benar oleh sistem.

*True Negative (TN)* : jumlah data negatif yang terklasifikasi dengan benar oleh sistem.

*False Positive (FP)* : jumlah data positif namun terklasifikasi salah oleh sistem.

*False Negative (FN)* : jumlah data negatif namun terklasifikasi salah oleh sistem.

# BABIV
# HASIL & PEMBAHASAN

## 4.1 Pengumpulan Data

Data penelitian yang diperoleh berdasarkan pada surat kabar elektronik kedua negara Indonesia dan Malaysia terkumpul dalam penelitian ini sebesar 41.317 kalimat. Surat kabar elektronik untuk Negara Indonesia yaitu CNN, Kompas dan Tribun untuk kategori ekonomi, hiburan, teknologi serta nasional data yang diperoleh sebesar 20.617 Kalimat, sedangkan untuk surat kabar elektronik Negara Malaysia yaitu Harian Metro dan Sinar Harian untuk kategori ekonomi, hiburan, teknologi, nasional serta olahraga data yang diperoleh sebesar 20.700 kalimat. Proses pengumpulan data dilakukan dengan cara membuat list URL surat kabar elektronik kedua negara tersebut menggunakan *Newspaper library* pada *Python*, yang kemudian disimpan kedalam *file* dengan format *txt*. Gambar 4.1 merupakan kode program untuk mengumpulkan data surat kabar elektronik kedua negara.

```
import newspaper
from newspaper import Article
urls= ['https://www.cnnindonesia.com/ekonomi']
for url in urls:
    article= Article(url, language='id')
    article.download()
    article.parse()
    with open('data_cnnekonomi.txt', 'a+') as f:
        txt = article.text
        txt = txt.encode('ascii', 'ignore').decode('utf-8')
        f.write(txt + '\n\n')
        f.close()
```

Gambar 4.1 Kode program mengumpulkan data

## 4.2 Preprocessing

Tahapan yang dilakukan selanjutnya adalah preprocessing, adapun proses preprocessing yang dilakukan dalam penelitian ini adalah sebagai berikut:

a. Menghapus *square bracket*

Kode program Gambar 4.2 akan menghapus tanda kurung persegi serta kata tau kalimat yang ada di dalamnya.

```
def remove_between_square_brackets(str) :
    return re.sub('\[[^]]*\]', '', str)
```

```
text = '[Gambas:Video CNN] Faisal mengatakan sinkronisasi kebijakan saat ini sudah mendapatkan dukungan dari BI.'
x= re.sub('\[[^]]*\]', '', text)

print(x)
```

```
Faisal mengatakan sinkronisasi kebijakan saat ini sudah mendapatkan dukungan dari BI.
```

Gambar 4. 2 Kode program menghapus *square bracket*

b. Menghapus non-ASCII

Implementasi kode program Gambar 4.3 untuk menghapus karakter yang tidak terdapat pada standar ASCII.

```
def remove_non_ascii(str):
    str = unicodedata.normalize('NFKD', str).encode('ascii',
'ignore').decode('utf-8-sig', 'ignore')
    return str
```

```
text ='Jan Peter Alexander «▼_▼ –@ui.ac.id»'

x1 = unicodedata.normalize('NFKD', text).encode('ascii', 'ignore').decode('utf-8-sig', 'ignore')
#6 = re.sub("b'[b\"", "", text)

print(x1)
```

```
Jan Peter Alexander <@ui.ac.id>
```

Gambar 4. 3 Kode program menghapus non-ASCII

c. Menghapus URL

Kode program Gambar 4.4 merupakan implementasi kode program untuk menghapus URL.

```
def remove_urls(str):
    str = re.sub(r'(2i)\b((?:https?:///wwwld(0,3[.]l[a-z0-9.\•
|+[.|[a•

z](2,4)/)(2:[^s()<>]+|\(([^sO<>]+|(\([^sO<>]+\)))^))+(2:\(([^s(
)<>]+1(\([^s0)<>]+\)))^)/^s!(\[\]{};:\'".,<>?«»w71))'
str)
    return str
```

```
text = 'Untuk maklumat lanjut bolehlah singgah di www.zootaiping.gov.my supaya kunjungan anda ke Taiping yang baru-baru ini

x2 = re.sub(r'(i)\b((2:https?:///a\d(0,3[.]/[a-z0-9.\-+[.J[a-2](2,4}/)02:(^5()<]+/\(([^s()O]+|(\([^s()O]+\)))^))

print(x2)
```

```
Untuk maklumat lanjut bolehlah singgah di  supaya kunjungan anda ke Taiping yang baru-baru ini dinamakan sebagai Destinasi Hija
u oleh badan bukan kerajaan (NGO) pelancongan akan lebih bermakna.
```

Gambar 4. 4 Kode program menghapus URL

**d. Menghapus *punctuation***

**Gambar 4.5 merupakan implementasi kode program untuk menghapus *punctuation* atau tanda baca.**

```
def      remove_punctuation(str):
    str = re.sub(r'[\w]]_',' ',str)
    return str
```

```
text = 'Antara tujuh permainan baru diperkenalkan, Power Surge, Disco!, Sky Tower, Spin Crazy, Bumper, Boo Boo Bump, Tea Cup

x3 = re.sub(r'[\w][_',' ',text)

print(x3)
```

```
Antara tujuh permainan baru diperkenalkan  Power Surge  Disco  Sky Tower  Spin Crazy  Bumper  Boo Boo Bump  Tea Cup dan Balloo
n Racem katanya
```

**Gambar 4. 5 Kode program menghapus *punctuation***

**e. Menghapusangka**

**Gambar 4.6 merupakan kode program untuk menghapus angka atau digit.**

```
def      remove_digit(str):
    str = re.sub(r"\b\d+\b", " ", str)
    return str
```

```
text = 'Aktivasi akun mencapai 654.889 atau rata-rata 1.570 per hari.Kepala BKPM Bahlil Lahadalia sendiri mengakui bahwa sis

x5 = re.sub(r"\b\d+\b", " ", text)

print(x5)
```

```
Aktivasi akun mencapai  .  atau rata-rata  .  per hari.Kepala BKPM Bahlil Lahadalia sendiri mengakui bahwa sistem OSS sudah bag
us, namun hanya proses mendapatkan NIB-nya saja.
```

**Gambar 4. 6 Kode program menghapus angka**

**f. Menghapus karakter tunggal**

**Gambar 4.7 merupakan kode program untuk menghapus karakter tunggal.**

```
def      remove_single character(str):
    str = re.sub(r"\s+[a-zA-Z]\s+", " ", str)
    return str
```

```
text = 'Kemudian nama-nama itu a akan diserahkan ke Presiden Joko Widodo untuk dipilih bersama DPR'

x6 = re.sub(r"\s+[a-zA-Z]\s+", " ", text)

print(x6)
```

```
Kemudian nama-nama itu akan diserahkan ke Presiden Joko Widodo untuk dipilih bersama DPR
```

Gambar 4. 7 Kode program menghapus karakter tunggal

g. Menghapus karakter berulang

Kode program Gambar 4.8 merupakan implementasi untuk menghapus karakter berulang.

```
def remove_repeated_character(str):
    str = re.sub(r'(.)\1{2,}', r'\1', str)
    return str
```

```
text = 'Menteri Yasonna pun memberiiii tenggat 14 hari bagi kedua pihak untuk menyelesaikan persoalan tersebut secara busine

x7 = re.sub(r'(.)\M{2,}', r'\', text)

print(x7)
```

```
Menteri Yasonna pun memberi tenggat 14 hari bagi kedua pihak untuk menyelesaikan persoalan tersebut secara business to busines
s.
```

Gambar 4. 8 Kode program menghapus karakter berulang

h. Menghapus spasi berulang

Gambar 4.9 merupakan implementasi kode program untuk menghapus spasi yang berulang.

```
def remove_additional_white_spaces(str):
    str = re.sub(r'(.)\1{2,}', r'\1', str)
    return str
```

```
text ='Dalam revisi nantinya perlu diatur PNS yang memiliki kinerja baik perlu diberikan penghargaan lebih.'

x8 = re.sub('[\s]+', ' ', text)

print(x8)
```

```
Dalam revisi nantinya perlu diatur PNS yang memiliki kinerja baik perlu diberikan penghargaan lebih.
```

Gambar 4. 9 Kode program menghapus spasi berulang

1. *Lower case*

Gambar 4.10 merupakan implementasi kode program untuk mengubah teks menjadi huruf kecil atau *lower case*.

```
def     to_lowercase(str) :
    str = str.lower()
    return str
```

```
text ='Kemudian, hukuman sedang terdiri dari penundaan kenaikan gaji selama satu tahun, penundaan kenaikan pangkat selama sa

x9 = text.lower()

print(x9)
```

```
kemudian, hukuman sedang terdiri dari penundaan kenaikan gaji selama satu tahun, penundaan kenaikan pangkat selama satu tahun,
dan penurunan pangkat setingkat lebih rendah selama satu tahun.sementara, pns yang mendapatkan hukuman berat bisa saja diturunk
an pangkatnya setingkat lebih rendah selama tiga tahun,
```

Gambar 4. 10 Kode program *lower case*

j. Menghapus karakter Twitter

Gambar 4.11 merupakan implementasi kode program untuk menghapus karakter yang ada di Twitter seperti *hashtag*, *mention* dan *retweet*.

```
def     remove_Twitter_Symbols(str) :
    #remove RT
    str = re.sub('RT', '', str)
    # hashtag
    str = re.sub(r"(?:\#+[\w_]+[\w\'_\-]*[\w_]+)", " ", str)
    #remove @username
    str = re.sub('@[\s]+','',str)
    # mention
    str = re.sub(r'(?:@[\w_]+)', ' ', str)
    return str
```

```
def removeTwitterSymbols(str):
    #remove RT
    str = re.sub('RT', '', str)
    # hashtag
    str = re.sub(r"(?:\#+[\w_]+[\w\'_\-]*[\w_]+)",      str)
    #remove @username
    str = re.sub('@[\s]+','',str)                        ",
     # mention
    str = re.sub(r'(?:@[\w_]+)',    ', str)



    return str

removeTwitterSymbols('#MnetApologizeToEXO punya maknalain, tulis akun @zkaiytlin. Seorang fan menyebut acara penghargaan mus
```

```
punya maknalain, tulis akun  Seorang fan menyebut acara penghargaan musik Mnet itu bak seeker ular.'
```

Gambar 4. 11 Kode program menghapus symbol Twitter

## 4.3 Pelabelan Data

Gambar 4.12 merupakan implementasi kode program untuk pelabelan data Bahasa Indonesia.

```
def     applyPreprocessing(file):
        df = pd.read_excel(file)
    df['CleanText'] = df['Kalimat'].apply(preprocessing)
        df.assign(cleanText=df['CleanText'])
        label= 'Indonesia'
        dfl = df.assign(Label=label)
        dfl.dropna()
                    writer= pd.ExcelWriter('%s_Clean.xlsx'%(label))
       #save to excel
            dfl.to_excel(writer, 'Sheetl')
            writer.save()
```

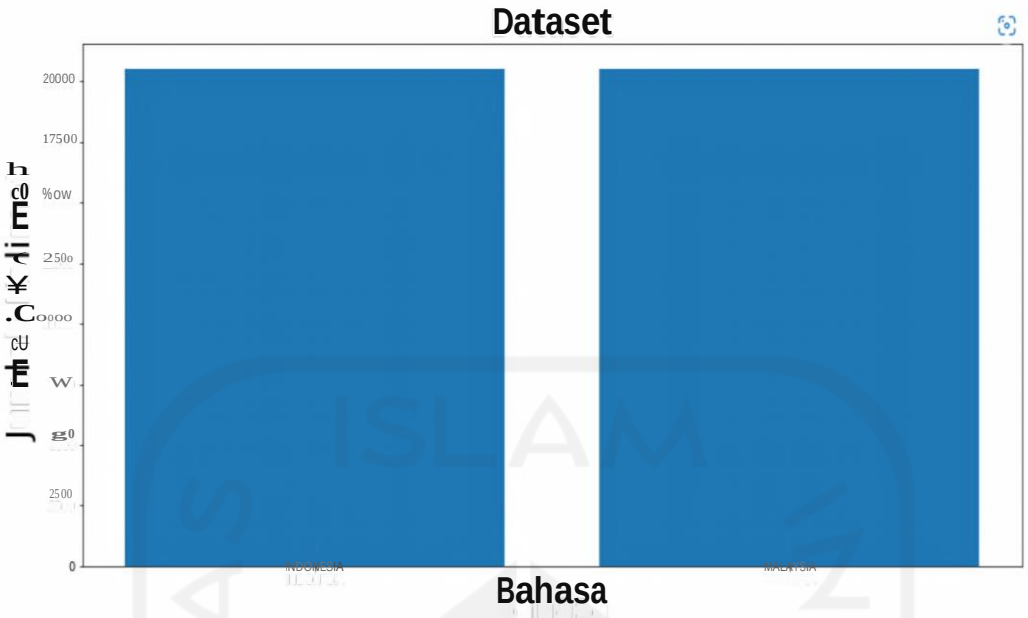
**Gambar 4. 12 Kode program pelabelan data Indonesia**

**Gambar 4.13 merupakan implementasi kode program untuk pelabelan data Bahasa Malaysia.**

```
def     applyPreprocessing(file):
        df = pd.read_excel(file)
    df['CleanText'] = df['Kalimat'].apply(preprocessing)
        df.assign(cleanText=df['CleanText'])
        label= 'Malaysia'
        dfl = df.assign(Label=label)
        dfl.dropna()
                    writer= pd.ExcelWriter('%s_Clean.xlsx'%(label))
       #save to excel
            dfl.to_excel(writer, 'Sheetl')
            writer.save()
```

**Gambar 4. 13 Kode program pelabelan data Malaysia**

**Gambar 4.14 merupakan total data yang diperoleh sebesar 41.317 kalimat untuk Bahasa Indonesia sebanyak 20.617 dan Malaysia sebanyak 20.700.**




Gambar 4. 14 Grafik data Indonesia dan Malaysia

## 4.4 *Embedding* Word2Vec

Word2Vec merupakan *word embedding* model yang digunakan untuk merepresentasikan kata yang memungkinkan memiliki arti atau makna serupa dengan kata-kata lain agar dipahami oleh *machine learning*. Secara teknis adalah pemetaan kata menjadi vektor. Tahapan dalam membangun model Word2Vec adalah sebagai berikut:

a. *Split* data *train* dan data *Test*

Pada proses *embedding* dengan menggunakan *library* Word2Vec Total data sebesar 41.3117 dibagi kedalam dua data *train* dan data *test*. Persentase untuk data *train* sebesar 75% diperoleh sebanyak 30.775 Kalimat, sedangkan untuk data *test* 25% diperoleh sebanyak 10.259 kalimat. Pada Gambar 4.15 merupakan kode program untuk membagi antara data *train* dan data *test*.

```
df_train, df_test = train_test_split(df, test_size=1-TRAIN_SIZE,
random_state=42)
print("TRAIN size:", len(df_train))
print("TEST size:", len(df_test))
```

Gambar 4. 15 *Split* data *train* dan *test*

**b. *Split list* data *train***

**Proses berikutnya adalah memisahkan kalimat menjadi kata pada data *train* untuk membuat kamus atau list item kata-kata *unique* pada proses berikutnya. Pada Gambar 4.16 merupakan kode program untuk *split* data *train*.**

```
%%time
documents = [_text.split() for _text in df_train.CleanText]
```

Gambar 4. 16 *Split list* data *train*

**c. Membangun model Word2Vec**

**Proses berikutnya adalah membangun model Word2Vec untuk membuat kamus atau *list* item kata *unique* ke dalam bentuk vektor. Parameter yang digunakan yaitu *size* untuk menentukan panjang vektor, *window* untuk menentukan jarak maksimum antara kata saat ini dan kata yang diprediksi dalam sebuah kalimat, *min_count* untuk menentukan jumlah minimum frekuensi kata yang sering muncul, jika kurang dari jumlah minimum maka kata tersebut akan diabaikan sedangkan parameter *workers* merupakan jumlah *thread* CPU yang akan digunakan. Gambar 4.17 merupakan kode program untuk model Word2Vec.**

```
w2v_model = gensim.models.word2vec.Word2Vec(size=W2V_SIZE,
window=W2V_WINDOW,
min_count=W2V_MIN_COUNT,
workers=6)
    w2v_model.build_vocab(documents)
    words = w2v_model.wv.vocab.keys()
    vocab_size= len(words)
    print("Vocab size", vocab_size)
```

**Gambar 4. 17 Model Word2Vec**

**d. *Training* model Word2Vec**

**Proses selanjutnya adalah *training list* item data *train* dengan menggunakan model yang sudah dibangun. Gambar 4.18 merupakan kode program untuk *training* dengan menggunakan model yang telah dibangun.**

```
%%time
w2v_model.train(documents,total_examples=len(documents),
epochs=W2V_EPOCH)
```

**Gambar 4. 18 *Training* model Word2Vec**

## 4.5 Klasifikasi

**Klasifikasi merupakan tahapan yang dilakukan untuk memperoleh model terbaik untuk mendeteksi kalimat Indonesia dan Malaysia. Tahapan yang dilakukan dalam membangun model klasifikasi Bi-LSTM *networks* yaitu:**

**a. *Tokenizing***

***Tokenizing* merupakan operasi untuk memisahkan teks menjadi potongan-potongan token sebelum dianalisis lebih lanjut. Proses ini bertujuan agar mendapatkan kata-kata unik yang akan digunakan sebagai kamus bahasa atau *vocabulary*. Fungsi *fit_on_texts* bertujuan untuk membuat index pada suatu kata atau karakter berdasarkan frekuensi kemunculannya, semakin sering muncul maka indexnya akan semakin kecil. Fungsi *text_to_sequences* bertujuan mengurutkan setiap kata berdasarkan urutan indexnya. Fungsi *pad_sequences* bertujuan untuk menyamakan ukuran dimensi pada seluruh teks sesuai dengan nilai *maxlen*, *maxlen* merupakan panjang maximum semua *sequences*. Gambar 4.19 merupakan kode program untuk tokenizing pada klasifikasi Bi-LS TM.**

```
%%time
tokenizer = Tokenizer()
tokenizer.fit_on_texts(df_train.CleanText)

vocab_size= len(tokenizer.word_index) + 1
print("Total words", vocab_size)

%%time
x_train =
pad_sequences(tokenizer.texts_to_sequences(df_train.CleanText),
maxlen=SEQUENCE_LENGTH)
x_test=
pad_sequences(tokenizer.texts_to_sequences(df_test.CleanText),
maxlen=SEQUENCE_LENGTH)
```

**Gambar 4. 19 *Tokenizing***

**b. *Encoding* label**

**Label *encoder* bertujuan untuk normalisai label pada data *test* dan data *train* untuk mengubah *non-numeric* label ke *numeric* label. Fungsi *fit_transform* bertujuan untuk**

**mencocokkan label dan mengembalikan label yang *di-encode* sedangkan fungsi *reshape* bertujuan untuk memberikan bentuk pada *array*. Gambar 4.20 merupakan kode program untuk *encode* label pada data *train* dan data *test*.**

```
encoder= LabelEncoder()
encoder.fit(df_train.Label.tolist())

y_train = encoder.transform(df_train.Label.tolist())
y_test = encoder.transform(df_test.Label.tolist())

y_train = y_train.reshape(-1,1)
y_test = y_test.reshape(-1,1)
```

**Gambar 4. 20 *Encoding* label**

**c. *Embedding***

**Pada Gambar 4.21 menunjukan kode program untuk implementasi hasil dari model Word2Vec yang telah dibangun, yang nantinya akan digunakan sebagai *embedding layer* untuk model klasifikasi. *Embedding_matrix* berisi kamus daftar vector untuk setiap kata, argumen *trainable=False* bertujuan untuk mencegah bobot *Embedding_matrix* diperbarui saat proses *training.***

```
embedding_matrix = np.zeros((vocab_size, W2V_SIZE))
for word, i in tokenizer.word_index.items():
  if word in w2v_model.wv:
    embedding_matrix[i] = w2v_model.wv[word]
print(embedding_matrix.shape)

embedding_layer = Embedding(vocab_size, W2V_SIZE,
weights=[embedding_matrix], input_length=SEQUENCE_LENGTH,
trainable=False)
```

**Gambar 4. 21 *Embedding layer***

**d. Model**

**Pada Gambar 4.22 menunjukkan kode program model Bi-LSTM. Tipe model *sequential* menunjukkan bahwa model dibuat berdasarkan *layer-by-layer*, fungsi *add* digunakan untuk menambahkan *layer* pada model yang sedang dibangun. Model dengan tipe ini memungkinkan untuk membangun sebuah model dengan tumpukan *layer* secara berurutan. Hal ini menunjukan bahwa data masuk dari satu *layer* ke *layer* lainnya sesuai dengan urutan *layer*. Kasus untuk**

klasifikasi dua kelas kategori model yang dibangun menggunakan fungsi aktivasi *sigmoid* dan fungsi *loss binaary_crossentropy*.

Langkah berikutnya setelah konfigurasi *layer* selesai yaitu *compile*, *compile* bertujuan agar konfigurasi *layer* yang telah dibuat dapat digunakan untuk proses *training*. Pada proses *compile* parameter yang digunakan yaitu *loss*, *optimizer dan metrics*. Fungsi *loss* bertujuan untuk menghitung kuantitas yang hams diminimalkan oleh model selama pelatihan. Fungsi optimizer bertujuan untuk mengatur respon model untuk estimasi error setiap kali bobot model diperbarui. Fungsi *matric* bertujuan untuk menilai kinerja dari model. Untuk klasifikasi dua kelas kategori fungsi *loss* yang digunakan pada model Bi-LSTM adalah *binary_crossentropy*. Untuk kasus *text classification* fungsi *optimizer* yang digunakan adalah *adam*. Fungsi metrics menggunakan *accuracy*, maka kinerja model dinilai berdasarkan akurasinya.

```
def blstm (mode):
    model= Sequential() model.add(embedding_layer)
    model.add(Dropout(0.5))
    model.add(Bidirectional(LSTM(lOO, dropout=0.2,
recurrent_dropout=0.2), merge_mode = mode))
    model.add(Dense(l, activation='sigmoid'))
    model.compile(loss='binary_crossentropy',
                  optimizer="adam",
                  metrics=['accuracy'])
    model.summary()
    return model
```

Gambar 4. 22 Model Bi-LSTM

e. *Training* model

*Training* model Bi-LSTM dilakukan dengan menggunakan fungsi *fit*. Parameter yang digunakan adalah *x* untuk *input* data, *y* untuk target data, *batch_size* untuk merupakan jumlah sampel yang akan dilatih setiap *epoch*, *epoch* merupakan jumlah untuk melatih seluruh data, *validation_split* berfungsi untuk membagi data *train* secara acak untuk digunakan sebagai data validasi, verbose berfungsi untuk menampilkan *progress bar* dari setiap *epoch*, *callbacks* yang merupakan fungsi yang akan dijalankan apabila fungsi lainya terpenuhi. Fungsi *callbacks* yang digunakan antara lain *ReduceLROnPlateau dan EarlyStopping*. *Fungsi ReduceLROnPlateau* bertujuan untuk mengurangi kecepatan proses training saat nilai yang dimonitoring tidak mengalami perkembangan. Fungsi *EarlyStopping* bertujuan untuk menghentikan proses *training saat nilai yang dimonitoring tidak mengalami perkembangan*. Pada Gambar 4.23 menunjukan kode program untuk *callbacks*.

```
callbacks= [ ReduceLROnPlateau(monitor='val_loss', patience=4,
cooldown=1),
                EarlyStopping(monitor='val_acc', min_delta=1e-3,
patience=5)]
```

**Gambar 4. 23 *Callback***

**Pada Gambar 4.24 menunjukan kode program untuk *training* model Bi-LSTM dengan menggunakan mode *sum*, *average*, *multiplication* dan *concatenation (default)*.**

```
history_sum = blstm('sum').model.fit(x_train, y_train,
                    batch_size=BATCH_SIZE,
                    epochs=EPOCHS,
                    validation_split=0.1,
                    verbose=1,
                    callbacks=callbacks)

history_ave = blstm('ave').model.fit(x_train, y_train,
                    batch_size=BATCH_SIZE,
                    epochs=EPOCHS,
                    validation_split=0.1,
                    verbose=1,
                    callbacks=callbacks)


history_mul= blstm('mul').model.fit(x_train, y_train,
                    batch_size=BATCH_SIZE,
                    epochs=EPOCHS,
                    validation_split=0.1,
                    verbose=1,
                    callbacks=callbacks)

history_cnc= blstm('concat').model.fit(x_train, y_train,
                    batch_size=BATCH_SIZE,
                    epochs=EPOCHS,
                    validation_split=0.1,
                    verbose=1,
                    callbacks=callbacks)
```

**Gambar 4. 24 Mode Bi-LSTM**

## 4.6 Evaluasi

Tahap evaluasi bertujuan untuk mengukur kinerja dari model Bi-LSTM yang dibuat. Pada proses evaluasi untuk mengukur kinerja model data *test* yang digunakan sebanyak 10.259. Hasil evaluasi adalah sebagai berikut:

a. Mode

Tabel 4.1 merupakan tabel model Bi-LSTM dengan mode *Concatenation, Multiplication, Average* dan *Sum. Epoch* untuk setiap modenya melalui proses *training*

sebanyak lima kali putaran yang dibagi kedalam satuan kecil (batch) sebesar 128 sampel data pertama dari 10.259 data. Fungsi dari dropout yaitu secara acak menonaktifkan neuron untuk mencegah overfitting.

Tabel 4. 1 Mode

| Model | Mode | Dropout | Batch Size | Epoch |
|---|---|---|---|---|
| Ml | Sum | 0.5 | 128 | 5 |
| M2 | Average | 0.5 | 128 | 5 |
| M3 | Multiplication | 0.5 | 128 | 5 |
| M4 | Concatenation | 0.5 | 128 | 5 |

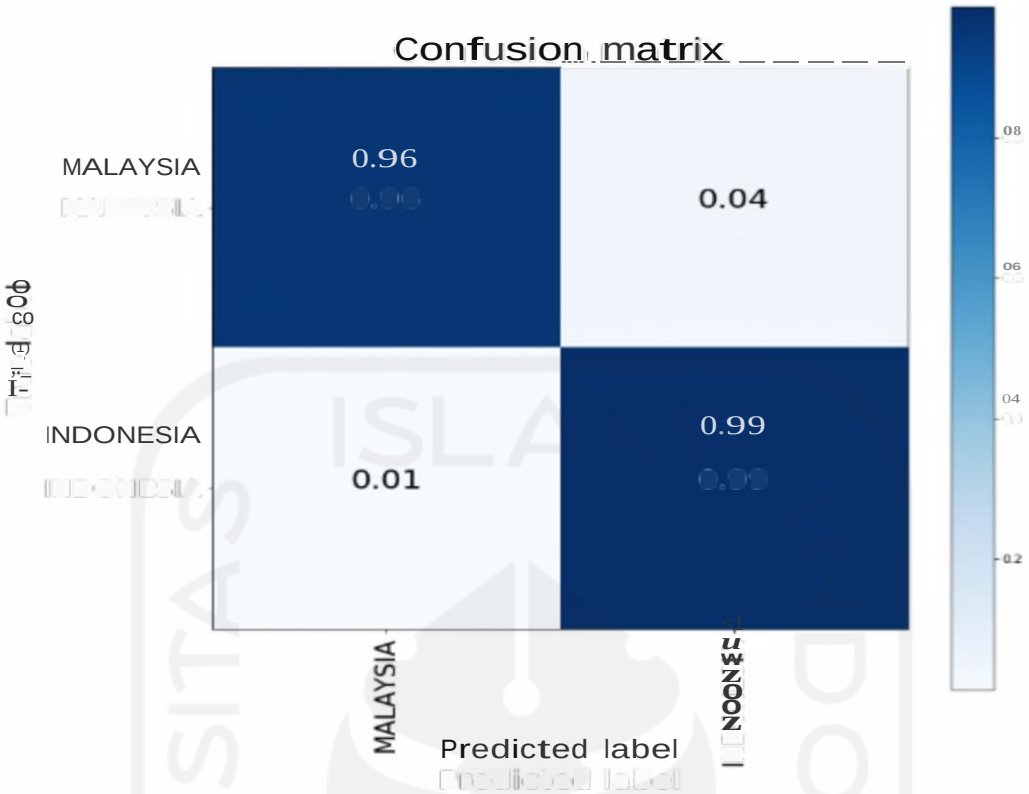
b. Confusion Matrix

Tabel 4.2 merupakan tabel hasil evaluasi model Bi-LSTM dengan menggunakan mode *Sum*. Nilai *precision* terbesar diperoleh Bahasa Indonesia yaitu 99%, untuk nilai *Recall* terbesar diperoleh Bahasa Malaysia yaitu 99%, nilai F1 *score* dan *Accuracy* Bahasa Indonesia dan Malaysia mendapatkan nilai yang sama yaitu 97%.

Tabel 4. 2 Hasil evaluasi mode *Sum*

| Bahasa | Precision | Recall | F1 Score | Accuracy |
|---|---|---|---|---|
| Indonesia | 99% | 96% | 97% | 97% |
| Malaysia | 96% | 99% | 97% | 97% |

Gambar 4.25 merupakan pengujian *Confusion Matrix* mode *Sum* dimana 96% data yang berbahasa Malaysia terklasifikasi benar berbahasa Malaysia, dan 99% data yang berbahasa Indonesia terklasifikasi benar berbahasa Indonesia.



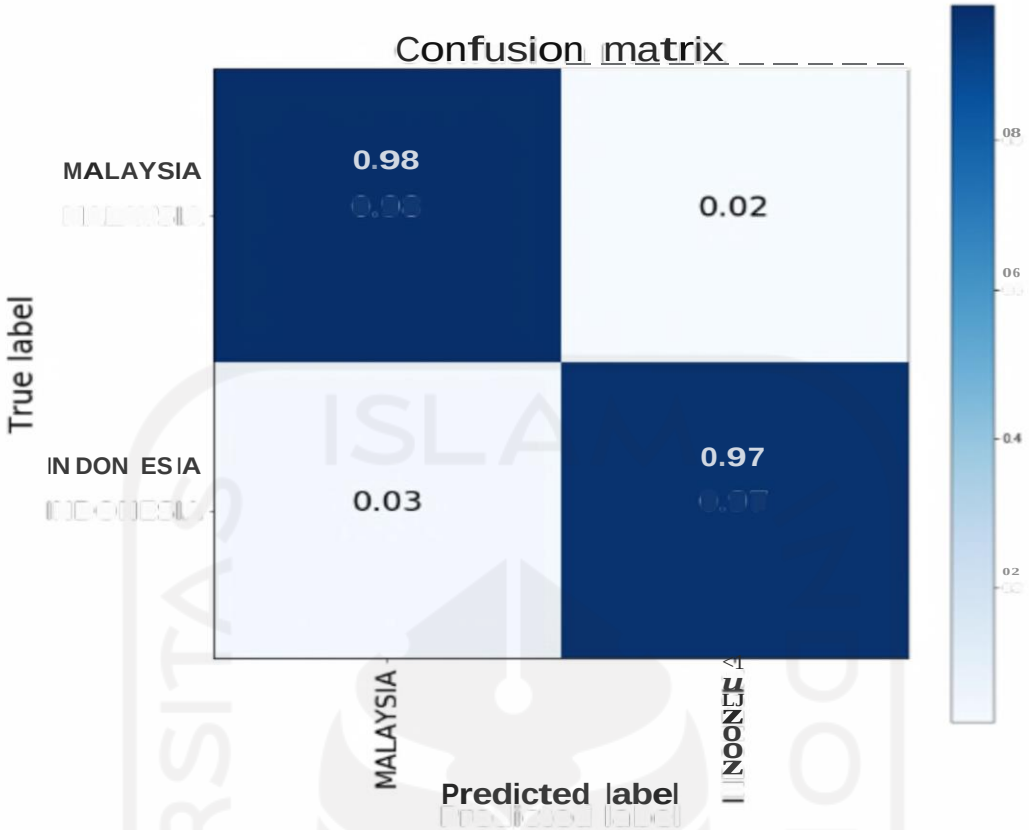

Gambar 4. 25 *Confusion Matrix* mode *Sum*

Tabel 4.3 merupakan tabel hasil evaluasi model Bi-LSTM dengan menggunakan mode *Average*. Nilai *precision* terbesar diperoleh Bahasa Malaysia yaitu 98%, untuk nilai *Recall* terbesar diperoleh Bahasa Indonesia yaitu 98%, sedangkan untuk nilai F1 *score* dan *Accuracy* baik Bahasa Indonesia dan Malaysia mendapatkan nilai yang sama yaitu 97%.

Tabel 4. 3 Hasil evaluasi mode *Average*

| **Bahasa** | **Precision** | **Recall** | **F1 Score** | **Accuracy** |
|---|---|---|---|---|
| Indonesia | 97% | 98% | 97% | 97% |
| Malaysia | 98% | 97% | 97% | 97% |
| | | | | |

Gambar 4.26 merupakan pengujian *Confusion Matrix* mode *Average* dimana 98% data yang berbahasa Malaysia terklasifikasi benar berbahasa Malaysia, dan 97% data yang berbahasa Indonesia terklasifikasi benar berbahasa Indonesia.



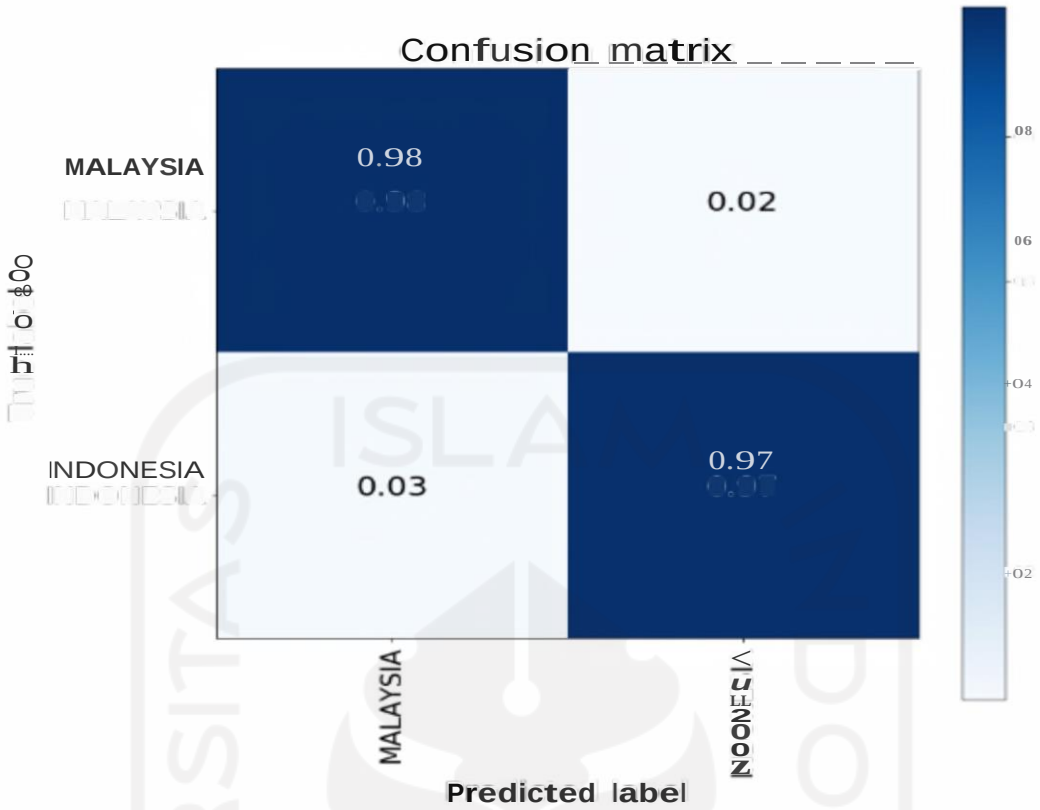

Gambar 4. 26 *Confusion Matrix* mode *Average*

Tabel 4.4 merupakan tabel hasil evaluasi model Bi-LSTM dengan menggunakan mode *Multiplication*. Nilai *precision* terbesar diperoleh Bahasa Malaysia yaitu 98%, untuk nilai *Recall* terbesar diperoleh Bahasa Indonesia yaitu 98%, sedangkan untuk nilai Fl *score* dan *Accuracy* baik Bahasa Indonesia dan Malaysia mendapatkan nilai yang sama yaitu 98%.

Tabel 4. 4 Hasil evaluasi mode *Multiplication*

| Bahasa | Precision | Recall | Fl Score | Accuracy |
|---|---|---|---|---|
| Indonesia | 97% | 98% | 98% | 98% |
| Malaysia | 98% | 97% | 98% | 98% |

Gambar 4.27 merupakan pengujian *Confusion Matrix* mode *Multiplication* dimana 98% data yang berbahasa Malaysia terklasifikasi benar berbahasa Malaysia, dan 97% data yang berbahasa Indonesia terklasifikasi benar berbahasa Indonesia.



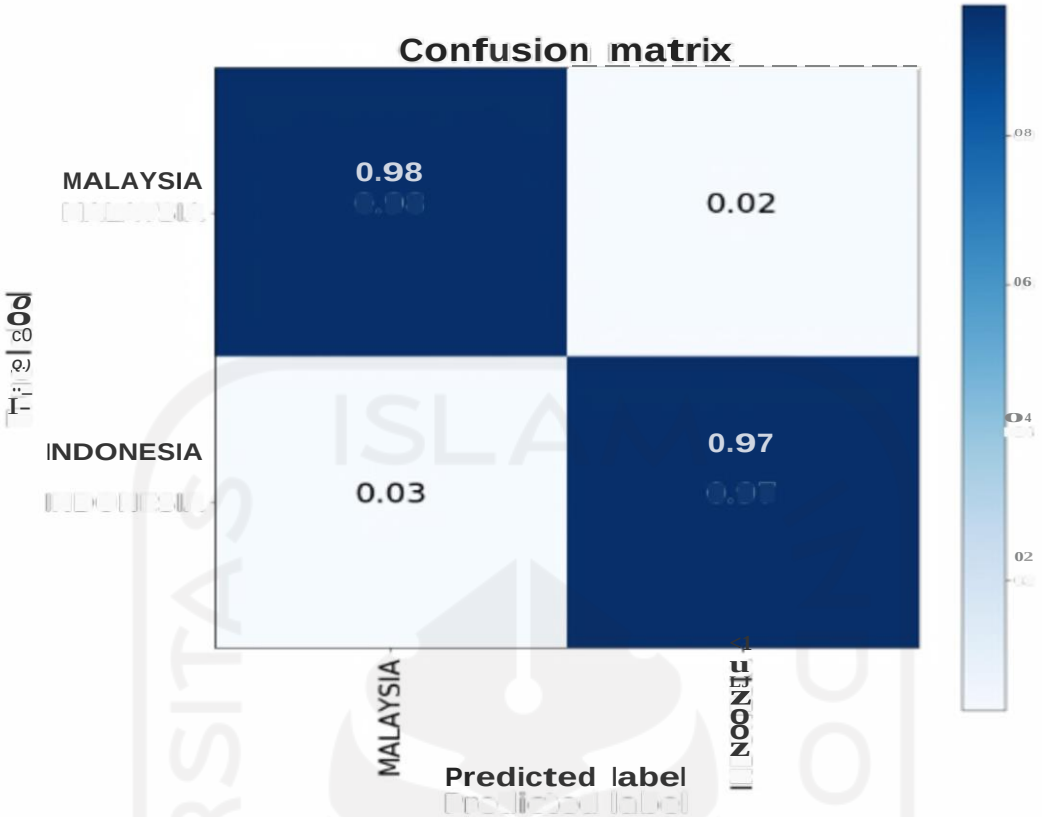

Gambar 4. 27 *Confusion Matrix* mode *Multiplication*

Tabel 4.5 merupakan tabel hasil evaluasi model Bi-LSTM dengan menggunakan mode *Concatenation*. Nilai *precision* terbesar diperoleh Bahasa Malaysia yaitu 98%, untuk nilai *Recall* terbesar diperoleh Bahasa Indonesia yaitu 98%, nilai Fl *score* dan *Accuracy* kedua bahasa mendapatkan nilai yang sama yaitu 98%.

Tabel 4. 5 Hasil evaluasi mode *Concatenation*

| Bahasa | Precision | Recall | Fl Score | Accuracy |
|---|---|---|---|---|
| Indonesia | 97% | 98% | 98% | 98% |
| Malaysia | 98% | 97% | 98% | 98% |

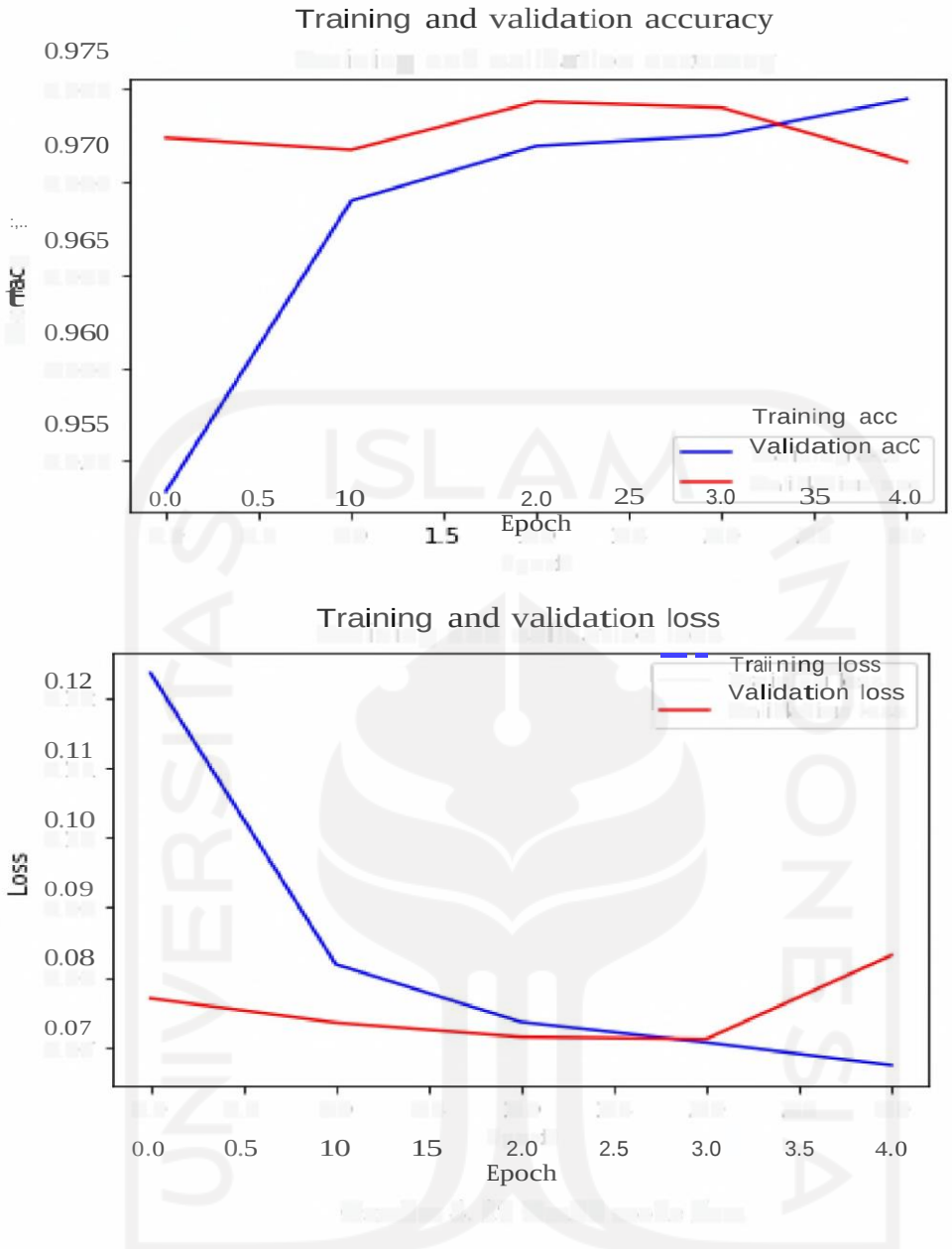
Gambar 4.28 merupakan pengujian *Confusion Matrix* mode *Concatenation* dimana 98% data yang berbahasa Malaysia terklasifikasi benar berbahasa Malaysia, dan 97% data yang berbahasa Indonesia terklasifikasi benar berbahasa Indonesia.




Gambar 4. 28 *Confusion Matrix* mode *Concatenation*

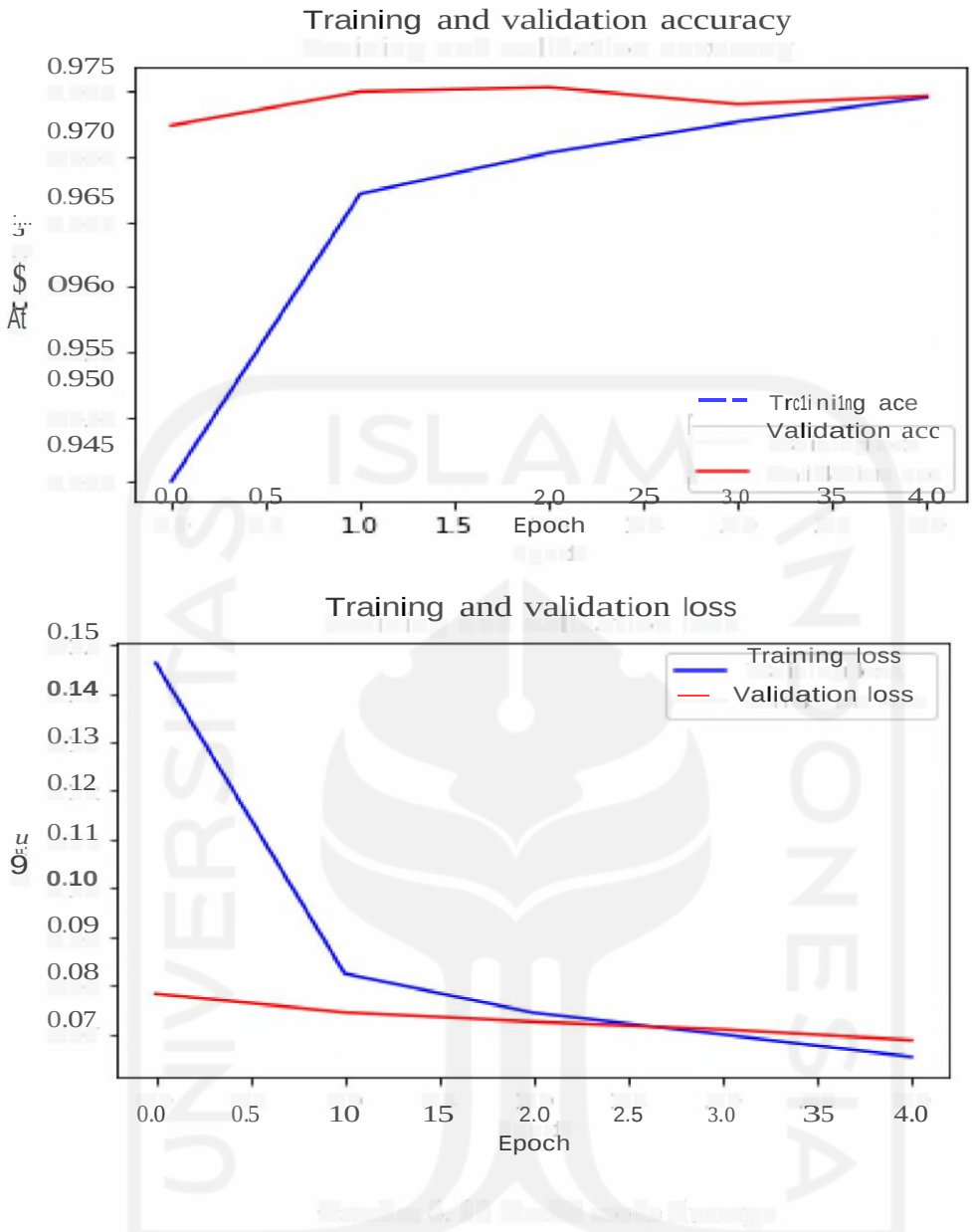
c. Grafik *train* dan *validation*

Pada Gambar 4.29 menunjukkan grafik akurasi *training* dan *validation* model Bi-LSTM dengan menggunakan mode *Sum*.



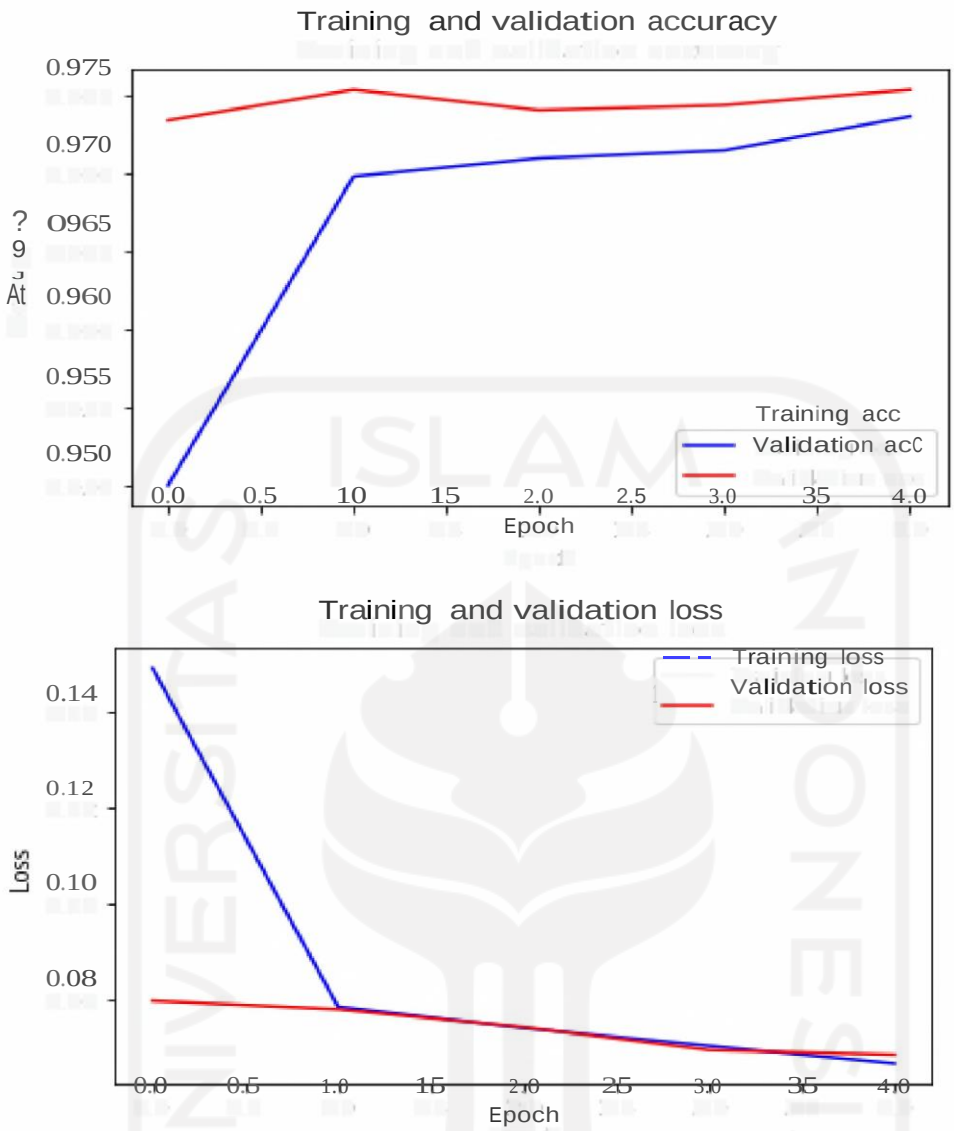

Gambar 4. 29 Grafik mode *Sum*

Pada Gambar 4.30 menunjukkan grafik akurasi *training* dan *validation* model Bi-LSTM dengan menggunakan mode *Average*.



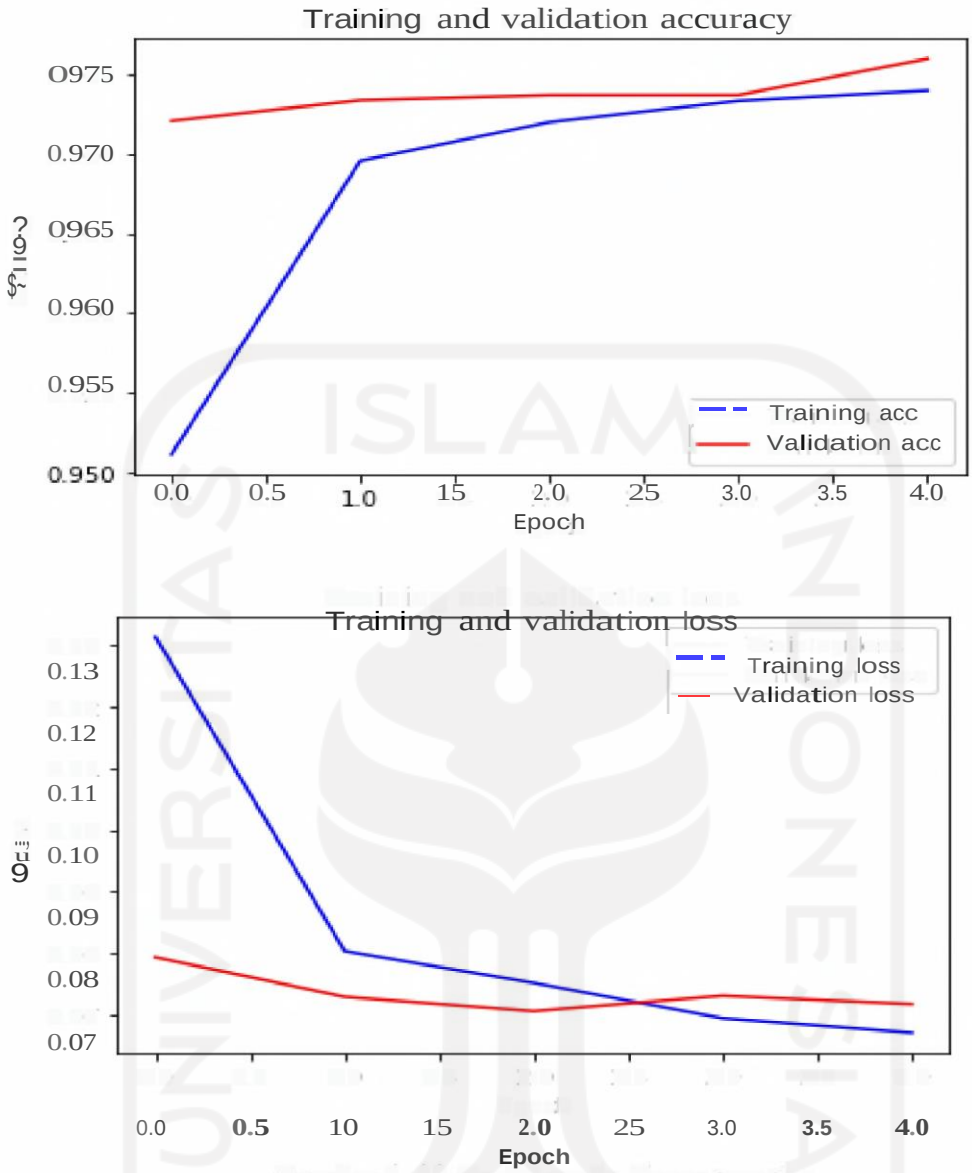

Gambar 4. 30 Grafik mode *Average*

Pada Gambar 4.31 menunjukkan grafik akurasi *training* dan *validation* model Bi-LSTM dengan menggunakan mode *multiplication*.




Gambar 4. 31 Grafik mode *Multiplication*

Pada Gambar 4.32 menunjukkan grafik akurasi *training* dan *validation* model Bi-LSTM dengan menggunakan mode *Concatenation*.




Gambar 4. 32 Grafik mode *Concatenation*

d. Analisis

Metode Bi-LSTM dengan menggunakan mode *Concatenation* dan *Multiplication* memiliki akurasi lebih tinggi dibandingkan dengan mode *Average* dan *Sum* dengan selisih akurasi sebesar 1%. Selisih nilai presisi dan recall pada mode *Sum* sebesar 2%, sedangkan selisih nilai presisi dan recall untuk mode *Concatenation*, *Multiplication* dan *Average* sebesar 1%.

## 4.7 Hasil Deteksi Kalimat

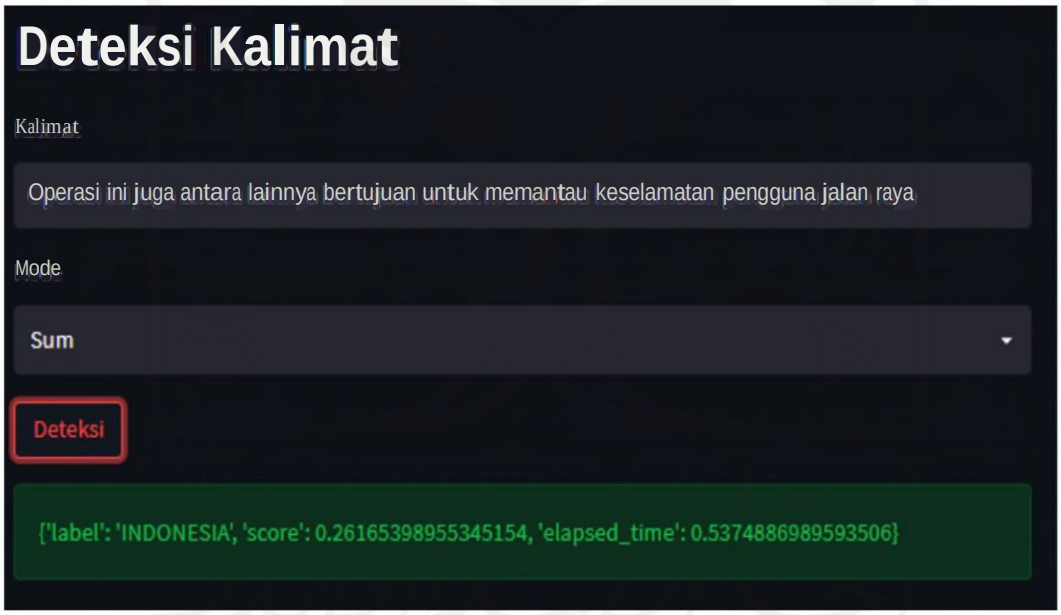
Gambar 4.33, Gambar 4.34, Gambar 4.35 dan Gambar 4.36 merupakan hasil deteksi kalimat Bahasa Indonesia dan Bahasa Malaysia metode Bi-LSTM dengan menggunakan mode

*Concatenation, Multiplication, Average* dan *Sum*. Studi kasus pada kalimat "Operasi ini juga antara lainnya bertujuan untuk memantau keselamatan pengguna jalan raya" yang dikutip dari media berita elektronik Malaysia terdeteksi sebagai Bahasa Indonesia kemungkinan yang terjadi yaitu terlalu banyak kata yang sama dengan Bahasa Indonesia, lalu kurangnya kata unik Malaysia sebagai contoh "lelaki", "tengkujung" dan lainnya lalu kemungkinan yang lain adalah kurangnya data *training*.


Gambar 4. 33 Hasil deteksi kalimat mode *concatenating*


Gambar 4. 34 Hasil deteksi kalimat mode *multiplication*


Gambar 4. 35 Hasil deteksi kalimat mode *average*




Gambar 4. 36 Hasil deteksi kalimat mode *sum*

# BAB V
# KESIMPULAN & SARAN

## 5.1 Kesimpulan

Berdasarkan hasil penelitian deteksi kalimat Bahasa Indonesia dan Malaysia dengan menggunakan metode Bi-LSTM, dapat disimpulkan bahwa:

a. Metode Bi-LSTM terbukti dapat mendeteksi antara Bahasa Indonesia dan Bahasa Malaysia. Hasil akurasi yang diperoleh pada mode *Concatenation* sebesar 98%, mode *Multiplication* sebesar 98%, mode *Average* sebesar 97% dan mode *Sum* sebesar 97%. Dalam melakukan deteksi Bahasa Indonesia dan Malaysia langkah yang perlu dilakukan yaitu *preprocessing* atau normalisasi, *word embedding* dengan menggunakan Word2Vec serta pengklasifikasian dengan Bi-LSTM menggunakan mode *Concatenation* atau *Multiplication* untuk mendapatkan hasil akurasi paling besar yaitu 98%.
b. Mode *Concatenation, Multiplication, Average* dan *Sum* pada model Bi-LSTM memberikan kinerja yang baik dalam melakukan klasifikasi serta tidak memiliki perbedaan yang besar terhadap hasil akurasinya yang hanya berselisih 1%.

## 5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, saran untuk pengembangan selanjutnya yaitu menggunakan sumber data yang berbeda seperti media sosial karena terdapat kalimat yang tidak bakau atau tidak sesuai dengan kaidah bahasa kedua negara tersebut serta menambahkan *hyperparameter* dikarenakan penelitian ini hanya menggunakan *batch size* dan *dropout* dengan nilai yang sama untuk setiap modenya.

# DAFTAR PUSTAKA


Adani, R. B. (2018). *Klasifikasi Bahasa Yang Mirip (Bahasa Indonesia Dan Bahasa Malaysia) Menggunakan Metode Support Vector Machine*. Universitas Islam Negeri Sultan Syarif Kasim Riau.

Cui, Z., Ke, R., & Wang, Y. (2018). *Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction*. 1–11. http://arxiv.org/abs/1801.02143

Dang, S., & Ahmad, P.H. (2014). Text Mining: Techniques and its Application. *International Journal of Engineering & Technology Innovations*, 1(4), 22–25.

Deng, X., Liu, Q., Deng, Y., & Mahadevan, S. (2016). An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences, 340–341*, 250–261. https://doi.org/10.1016/j.ins.2016.01.033

Hassan, A., & Mahmood, A. (2017). Deep learning for sentence classification. *2017 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2017, September*. https://doi.org/10.1109/LISAT.2017.8001979

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jambukia, S. H., Dabhi, V. K., & Prajapati, H. B. (2018). ECG beat classification using machine learning techniques. *International Journal of Biomedical Engineering and Technology*, 26(1), 32–53. https://doi.org/10.1504/IJBET.2018.089255

Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing, 337*, 325–338.

Tan, A.-H. (1999). Text Mining: The state of the art and the challenges. *Proceedings of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases, 8*, 65–70. https://doi.org/10.1.1.38.7672

Vidhya, K.. A., & Aghila, G. (2010). Text Mining Process , Techniques and Tools: an Overview. *International Journal of Information Technology and Knowledge Management, 2(2)*, 613–622.

Waridah, W. (2015). Penggunaan Bahasa dan Variasi Bahasa dalam Berbahasa dan Berbudaya. *JURNAL SIMBOLIKA: Research and Learning in Communication Study (E-Journal)*,

1(1).

Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., & Xu, B. (2016). Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. CO*LING 2016- 26th International Conference on Computational Linguistics, Proceedings of* CO*LING 2016: Technical Papers*, 3485–3495.

LAMPIRA N