

LAPORAN PROYEK

11S4037 – Pemrosesan Bahasa Alami

Fake News Detection Using Bidirectional LSTM



Disusun Oleh:

12S17029	Silvany Lumbangaol
12S17040	Yeni Chintya Panjaitan
12S17058	Juanda Antonius Pakpahan
12S17064	Melani Basaria Pakpahan

**PROGRAM STUDI SARJANA SISTEM INFORMASI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL
JANUARI 2020**

DAFTAR ISI

DAFTAR GAMBAR.....	4
DAFTAR TABEL	5
BAB I.....	6
PENDAHULUAN	6
1.1 Latar Belakang.....	6
1.2 Tujuan.....	7
1.3 Manfaat.....	8
1.4 Ruang Lingkup.....	8
1.5 Sistematika Penyajian.....	8
BAB II ISI.....	9
2.1 Analisis.....	9
2.1.1 Analisis Sumber Data	9
2.1.2 Analisis Bentuk Data	9
2.1.3 Analisis Metode	11
2.1.4 Analisis Pra-pemrosesan Data	14
2.1.5 Analisis Metode <i>Word Embedding</i>	16
2.1.6 Analisis Metode Klasifikasi	17
3.1.7 Analisis Metode Bidirectional Long Short Term Memory.....	17
2.2 Desain.....	18
2.2.1 Desain Umum Sistem	18
2.2.2 Desain Prapemrosesan Data	20
2.2.2.1 <i>Data Transformation</i>	23
2.2.2.2 Data Reduction.....	24
2.2.2.3 <i>Punctuation Removal</i>	25
2.2.2.4 <i>Tokenization</i>	25
2.2.2.6 <i>Stopword Removal</i>	26
2.2.2.7 <i>Lemmatization</i>	27
2.3 Implementasi	28
2.3.1 Implementasi Prapemrosesan data	29
2.3.2 Implementasi Word2Vec.....	31
2.3.3 Implementasi Bidirectional LSTM.....	33

2.4	Hasil.....	34
2.4.1	Hasil Pra pemrosesan data.....	35
2.4.2	Hasil Word2Vec	39
2.4.3	Hasil Bidirectional LSTM	40
REFERENSI.....		43

DAFTAR GAMBAR

Gambar 1 Metode Pengerjaan proyek.....	12
Gambar 2 Desain Umum Sistem.....	19
Gambar 3 Prapemrosesan Data.....	21
Gambar 4 Proses Pembersihan Data	22
Gambar 5 Proses Transformasi Nilai pada Atribut	23
Gambar 6 Proses Reduksi Data	24
Gambar 7 Proses <i>Punctuation Removal</i>	25
Gambar 8 Proses <i>Tokenization</i>	26
Gambar 9 Proses <i>Stopword Removal</i>	27
Gambar 10 Proses <i>Lemmatization</i>	28

DAFTAR TABEL

Tabel 1. Atribut pada <i>fake news dataset</i>	9
Tabel 2. Analisis untuk proses <i>Tranformasi</i>	14
Tabel 3. Analisis untuk proses <i>punctuation removal</i>	15
Tabel 4. Analisis untuk proses <i>tokenization</i>	15
Tabel 5. Analisis untuk proses <i>stopword removal</i>	16
Tabel 6. Analisis untuk proses <i>lemmatization</i>	16

BAB I

PENDAHULUAN

Bab pendahuluan berisi penjelasan terkait latar belakang pemilihan topik, tujuan, manfaat yang ingin dicapai, ruang lingkup, dan sistematika penyajian laporan proyek.

1.1 Latar Belakang

Teknologi komunikasi dan informasi (TIK) berkembang pesat mengikuti perkembangan zaman, bersamaan dengan adanya beragam media termasuk media *online* yang memainkan peran penting dalam penyebaran informasi tentang suatu peristiwa kepada publik. Berbagai media *online* menawarkan kemudahan bagi pengguna untuk menerima artikel – artikel terbaru mengenai berita. Banyak informasi yang bermanfaat yang bisa didapatkan melalui membaca berita dari jejaring sosial atau media *online*. Penyebaran informasi atau berita melalui media *online* tidak hanya dilakukan oleh situs berita yang sudah dikenal oleh masyarakat, namun juga dapat dilakukan oleh seluruh pengguna internet tanpa melalui pemeriksaan apapun. Namun, beberapa informasi yang disebarluaskan secara individu atau berkelompok tersebut tidak dapat dipertanggungjawabkan kebenaran atau terindikasi berita palsu dan menjangkau ribuan pengguna. Berita palsu merupakan informasi atau berita yang berisi hal-hal yang tidak diketahui kebenarannya atau belum pasti. [1]

Untuk mencegah penyebaran berita palsu perlu dibangun suatu model yang berfungsi untuk mengidentifikasi suatu informasi agar informasi tersebut tidak mempengaruhi banyak orang. Dengan melakukan pencegahan, maka dampak negatif yang ditimbulkan oleh penyebaran berita atau konten palsu ini bisa diminimalisir. Penelitian mengenai deteksi berita palsu atau *Fake News Detection* telah dilakukan sebelumnya oleh Sheng How Kong, Li Mei Tan, Keng Hoon Gan, dan Nur Hana Samsudin (2019). Dalam proses *data preprocessing*, penelitian ini menggunakan 2 metode vektorisasi, yakni *N-Gram Vector* dengan *TF-IDF Encoding* dan metode *Sequence Vector* dengan *One-Hot Encoding*. Proses *Model training* pada penelitian ini digunakan 4 model, yaitu 2 model menggunakan *N-*

Gram Vector dengan *TF-IDF Encoding* untuk judul dan konten berita dan 2 model lainnya menggunakan *Sequence Vector* dengan *One-Hot Encoding* untuk judul dan konten berita. Dalam penelitian ini diperoleh hasil bahwa penggunaan metode *N-Gram Vector* dengan *TF-IDF Encoding* menghasilkan akurasi dan *recall* lebih baik jika dibandingkan dengan penggunaan *Sequence Vector* dengan *One-Hot Encoding*. [2] Sebagai pengembangan selanjutnya, Sheng How Kong et al. menyarankan untuk mengkombinasikan RNN dengan LSTM dengan tujuan untuk meningkatkan akurasi. [2]

Berdasarkan pemaparan diatas, penulis tertarik untuk mengerjakan sebuah proyek dengan pendekatan *deep learning* menggunakan metode *Bidirectional LSTM* yang merupakan variasi RNN (*Recurrent Neural Network*) dan *Word2vec* sebagai metode vektorisasi kata sebelum diproses pada *Bidirectional LSTM*. *Bidirectional LSTM* digunakan dalam memproses data secara sekuensial dan dari dua arah dengan menggunakan *forward layer* untuk mempelajari informasi mendatang dan *backward layer* untuk informasi masa lalu, sehingga lebih mudah memahami keterkaitan kedua *layer* tersebut dan memprediksi apa kata selanjutnya yang akan muncul pada berita. Penerapan *Bidirectional LSTM* juga dapat menangani masalah *vanishing gradient*, karena LSTM didesain untuk mengatasi permasalahan tersebut, dengan menggunakan mekanisme gerbang (*gate*) untuk memilih informasi penting pada berita yang akan disimpan di *memory cell*, sehingga dapat mengurangi penumpukan *layer* yang dapat memperlambat proses *training*. Pemilihan *Word2vec* sebagai metode vektorisasi dikarenakan *Word2vec* menghasilkan vektorisasi padat (*dense*), pendek serta tidak memakan banyak memori dan sumber daya. [3]. Ada 1 model yang akan dibangun di dalam penelitian dengan menggunakan *Bidirectional LSTM* yakni penerapan *Bidirectional LSTM* pada judul berita.

1.2 Tujuan

Adapun tujuan proyek adalah sebagai berikut:

- a. Untuk mengetahui proses kerja dari penggunaan metode *Bidirectional LSTM* dengan CBOW sebagai *word embedding* dalam mengidentifikasi berita palsu.

- b. Untuk mengukur seberapa besar akurasi dan *recall* dari penggunaan metode *Bidirectional LSTM* dengan CBOW sebagai *word embedding* dalam mendeteksi berita palsu.

1.3 Manfaat

Adapun yang menjadi manfaat proyek adalah sebagai berikut:

- a. Penulis dapat menambah wawasan dan pengalaman secara langsung dalam membangun sebuah model untuk mengidentifikasi berita.
- b. Sebagai referensi, jika ingin dilakukan pengembangan terhadap proyek.

1.4 Ruang Lingkup

Pada sub-bab ini dijelaskan mengenai batasan penelitian yang akan dilakukan.

Adapun batasan ruang lingkup penelitian yang akan dilakukan yaitu:

1. Variabel yang akan digunakan pada proyek sebagai pertimbangan dalam melakukan mengidentifikasi berita palsu adalah title atau judul dari berita.
2. Dataset yang berisi fake news dan real news yang diperoleh dalam rentang waktu 3 tahun terakhir yang bersumber dari *IEEE Dataport*. Dimana *dataset* yang digunakan adalah file dokumen berekstensi .csv yang mengandung teks judul berita yang unik di setiap barisnya.
3. Word Embedding yang digunakan merupakan salah satu arsitektur *Word2vec* yakni *Continious Bag of Word* (CBOW) dengan *window size* sebesar 5.
4. Ukuran yang dibandingkan dalam Penggunaan *Bidirectional LSTM* adalah akurasi dan *recall*.

1.5 Sistematika Penyajian

Sistematika penyajian yang digunakan pada penulisan dokumen tugas akhir dibagi menjadi tujuh (7) bab, yaitu:

- BAB 1 PENDAHULUAN. Pada bab ini dijelaskan latar belakang proyek, tujuan, manfaat, ruang lingkup dan sistematika penyajian dalam pelaksanaan proyek.
- BAB 2 ISI. Pada bab ini dijelaskan mengenai tahapan serta penjabaran pemrosesan bahasa alami yang akan diterapkan pada proyek.
- BAB 3 PENUTUP. Pada bab ini dijelaskan mengenai kesimpulan dan saran dari proyek yang akan dikerjakan.

BAB II

ISI

Pada bab ini dijelaskan mengenai tahapan analisis data dan metode, desain pemrosesan bahasa alami, implementasi dan hasil yang berupa evaluasi dari implementasi yang telah dikerjakan.

2.1 Analisis

Pada bagian ini dijelaskan mengenai berbagai metode yang akan digunakan dan analisis terhadap data. Analisis dilakukan untuk mengenali atau mengetahui struktur dari data dan metode yang menjadi acuan pada tahap implementasi. Analisis yang dilakukan terdiri dari analisis sumber data, analisis bentuk data, analisis metode, analisis metode *text preprocessing*, analisis metode *word embedding* dengan CBOW, analisis metode *bidirectional LSTM*, dan analisis metode evaluasi penelitian.

2.1.1 Analisis Sumber Data

Dataset berita yang didapatkan dari IEEE *Dataport* yaitu *fake news dataset*. IEEE *dataport* adalah *platform* standard untuk *dataset*, dimana *platform* ini bersifat *open access* (dapat diakses oleh semua orang) dan biasanya merupakan wadah untuk kompetisi pembuatan *dataset*. Tahapan Analisis sumber data digunakan untuk mengetahui atribut data relevan yang akan digunakan dalam mendeteksi berita palsu. Data judul berita yang berhasil terkumpul dari *fake news dataset* sebanyak 16.268 judul, dimana judul merupakan atribut yang dipertimbangan dalam pembangunan model.

2.1.2 Analisis Bentuk Data

Analisis terhadap bentuk data yang digunakan pada penelitian mengacu pada data yang telah diambil dari IEEE *Dataport*. Analisis ini dilakukan dengan tujuan untuk mengetahui karakteristik *dataset* yang akan digunakan pada penelitian guna memberikan gambaran terkait atribut apa saja yang dibutuhkan pada penelitian. Berikut adalah penjelasan lebih rinci terkait bentuk data pada penelitian. Penjelasan atribut, tipe atribut, keterangan dan contoh nilai setiap atribut pada *fake news dataset* dapat dilihat pada table di bawah ini.

Tabel 1. Atribut pada *fake news dataset*

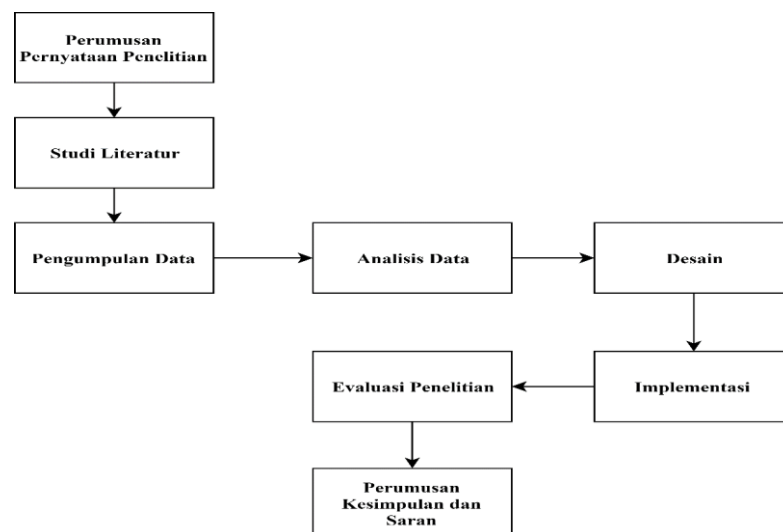
No	Atribut	Tipe Atribut	Keterangan Atribut	Contoh Nilai Atribut
1	Id	Numerik	Kode unik berita	3106
2	<i>Date</i>	Numerik	Waktu <i>publish</i> berita	2011-01-25
3	<i>speaker</i>	Kategorikal	Pembicara berita	Joe Wilkinson
4	<i>statement</i>	Kategorikal	Isi singkat berita	<i>A national organization says Georgia has one of America toughest ethics laws.</i>
5	<i>Sources</i>	Kategorikal	Sumber berita dalam bentuk halaman web	' http://www.politifact.com/georgia/statements/2011/apr/11/saxby-chambliss/senators-ring-alarm-about-4-billion-debt-problem/
6	<i>Paragraph_based_content</i>	Kategorikal	Paragraph ringkas sebuah berita	Before that, from 1998 to 2001, for a brief period at the end of President Bill Clinton's term and the start of President Barack Obama's term, Instagram users are pinning the 2019 coronavirus outbreak on carnivores
7	<i>FullText_based_content</i>	Kategorikal	Isi detail konten berita	In a post published March 15, the account for "'Cowspiracy,'" a film about the environmental impact of meat production, claimed that COVID-19 started "'because we eat animals.'", "'COVID-19 would not exist if the world was vegan,'" reads the image.'. 'The post was flagged as part of Facebook's efforts to combat false news and misinformation on its News Feed. (Read more about our partnership with Facebook , which owns Instagram.) It had more than 27,000 likes as of March 18.', '(Screenshot from Instagram)', 'Early on in the coronavirus

				pandemic, which had infected more than 179,000 people worldwide as of March 17, we saw a lot of baseless claims pinning the disease on the consumption of bats in China. Cowspiracyâ€™s Instagram post piggybacks on that misinformation.', 'As evidence, it refers to the"
8	Label_fnn / label-liar	Kategorikal	Penggolongan berita antara berita palsu atau asli	<i>Fake, Real</i>

Berdasarkan deskripsi di atas, maka atribut yang dibutuhkan untuk pengklasifikasian berita palsu adalah *statement* sebagai judul berita dan label sebagai pengklasifikasian berita. Statement dipilih sebagai objek yang akan di prediksi dan label pada berita akan digunakan untuk melatih model.

2.1.3 Analisis Metode

Subbab ini menjelaskan tentang metodologi penelitian yang digunakan. Metodologi penelitian tersebut adalah rumusan pernyataan masalah, studi literatur, analisis desain, implementasi, evaluasi dan perumusan kesimpulan dan saran. Secara umum, langkah-langkah pengerjaan yang dilakukan dapat dilihat pada gambar berikut ini.



Gambar 1 Metode Pengerjaan proyek

Gambar 1 merupakan metodologi penelitian yang direpresentasikan dalam bentuk bagan. Berikut adalah penjelasan bagan terkait metode penelitian.

1. Perumusan Pernyataan Penelitian

Pada tahap ini dijelaskan mengenai hal yang melatar belakangi permasalahan terkait dengan berita palsu (*fake news*). Hal yang menjadi latar belakang penelitian ini diambil dari permasalahan para pembaca yang hendak membaca berita, tetapi sering sekali berita yang mereka konsumsi merupakan berita palsu. Berita tersebut dibuat semenarik mungkin menyerupai berita asli yang disertai dengan tambahan informasi – informasi lainnya dan disajikan secara berlebihan. Hal tersebut akan mempengaruhi pandangan, pola pikir dan tindakan masyarakat terhadap objek yang terdapat pada berita palsu. Untuk menghindari hal tersebut dibutuhkan suatu metode yang dapat membedakan antara berita yang nyata dan palsu. Beberapa faktor yang dipertimbangkan dalam penelitian ini adalah judul dan konten dari suatu berita.

2. Studi Literatur

Pada tahap ini akan mencari berbagai studi literatur yang akan menjadi informasi untuk penelitian. Studi literatur didapatkan dari berbagai sumber seperti buku dan jurnal penelitian terdahulu yang terkait dengan penelitian yang akan dilakukan. Studi literatur tersebut akan digunakan untuk menjawab pertanyaan penelitian yang akan menghasilkan landasan teoritis untuk penelitian yang akan dilakukan. Literatur yang digunakan pada penelitian ini sebagian besar menggunakan bahasa Inggris dan merupakan *paper* yang berasal dari luar Indonesia. Hal tersebut disebabkan karena literatur yang membahas terkait *Fake News Detection* dalam bahasa Indonesia masih berjumlah sangat sedikit.

3. Pengumpulan Data

Pada tahap ini akan dibahas mengenai persiapan data berupa pencarian dan pengumpulan data. Data yang dikumpulkan untuk penelitian ini bersumber dari *IEEE Dataport*. Data yang digunakan adalah data berita asli dan palsu yang menggunakan bahasa Inggris. Penggunaan *dataset* dalam bahasa Inggris dikarenakan ketersediaan *dataset* berita dalam bahasa Indonesia sangat sedikit.

Melalui tahapan ini telah dikumpulkan sebanyak 334.784 untuk data train dan 15.845 untuk data test.

4. Analisis

Pada tahap ini akan dilakukan analisis yang dimulai dari analisis data yang mencakup sumber dan bentuk data, pra-proses data, analisis metode *bidirectional LSTM*, analisis metode representasi kata menggunakan *CBow* dan analisis metode evaluasi terhadap pengerjaan proyek. Tujuan akhir dari analisis adalah untuk mendapatkan informasi dan gambaran mengenai penelitian yang akan dikerjakan.

5. Desain

Pada tahap ini dilakukan perancangan sistem yang tersusun dari beberapa komponen yang terstruktur sebelum masuk ke tahap implementasi. Perancangan dilakukan berdasarkan hasil analisis yang sudah dilakukan dan diperoleh sebelumnya.

6. Implementasi

Pada tahap ini dilakukan pengimplementasian komponen-komponen yang telah dirancang atau didesain pada tahap sebelumnya, untuk menguji ketepatan rancangan yang dibuat. Pada penelitian ini juga dilakukan eksperimen yang mana diharapkan dapat menghasilkan keakurasian yang cukup tinggi melalui pendekatan *bidirectional LSTM*.

7. Evaluasi Penelitian

Pada tahap ini dilakukan evaluasi dan pembahasan hasil eksperimen pendeteksi berita palsu (*fake news*). Mengukur kinerja suatu sistem klasifikasi merupakan hal yang penting, sehingga dapat menggambarkan seberapa baik sistem tersebut untuk mengklasifikasikan data. *Confusion Matrix* merupakan pengukur kinerja klasifikasi yang paling umum. Evaluasi dan pembahasan hasil eksperimen dilakukan untuk mengetahui apakah eksperimen memiliki performansi yang cukup baik untuk menentukan apakah suatu berita dikategorikan palsu atau tidak, berdasarkan jumlah Akurasi dan *Recall* yang diharapkan tinggi. Apabila masih terdapat *error* maka dilakukan perbaikan pada proyek.

8. Perumusan Kesimpulan dan Saran

Pada akhir penelitian, dilakukan perumusan kesimpulan dan saran terkait hasil penelitian pendeteksi berita palsu yang dicapai serta saran yang diberikan untuk pengembangan proyek selanjutnya.

2.1.4 Analisis Pra-pemrosesan Data

Pada tahap ini akan membahas metode atau teknik apa saja yang digunakan untuk mempersiapkan data sebelum masuk ke dalam tahap vektorisasi. Pra-pemrosesan merupakan tahapan yang penting untuk membersihkan data yang akan digunakan untuk melatih model pengklasifikasian. Adapun yang menjadi teknik atau metode pemrosesan data pada penelitian adalah sebagai berikut.

1. *Data Cleaning*

Tahapan *data cleaning* diterapkan untuk menangani nilai *null* pada data. Metode yang digunakan adalah dengan menggunakan *dropna()* untuk membersihkan data dari nilai *null*.

2. *Data Transformation*

Tahapan ini dilakukan dengan tujuan untuk mentransformasikan label berita ke dalam bentuk *binary*. Isi atribut label terdiri dari dua bagian yakni *real* sebagai berita asli dan *fake* sebagai berita palsu. Setelah dilakukan tahapan *data transformation*, dapat dilanjutkan dengan menerapkan proses *Exploratory Data Analysis*. Bentuk transformasi yang dilakukan pada tahapan ini dapat dilihat pada tabel di bawah ini.

Tabel 2. Analisis untuk proses *Transformasi*

Label berita	Ditransformasi menjadi
<i>Fake</i>	"0"
<i>Real</i>	"1"

3. *Data Reduction*

Tahapan *data reduction* yang diterapkan dalam penelitian ini berfungsi untuk mengurangi dimensi dari data. Dimensi yang akan digunakan pada penelitian terdiri *statement* dan label, yang artinya selain dari dimensi yang telah disebutkan sebelumnya harus dihapus. Untuk menghapus beberapa atribut yang tidak relevan dalam penelitian, digunakan *data reduction* dengan strategi *feature selection*. Cara

yang diterapkan pada *feature selection* adalah melakukan *filter* terhadap atribut yang dibutuhkan, dimana atribut yang akan *filter* merupakan atribut bertipe kategorikal.

4. *Punctuation Removal*

Tahapan ini diterapkan dalam judul dan konten berfungsi untuk menghapus simbol – simbol yang terdapat pada berita. Untuk dapat menghapus simbol-simbol tersebut dibutuhkan program yang dapat menemukan dan kemudian menghapus simbol atau tanda baca tersebut menggunakan *regular expression*. Hal yang ingin diselesaikan pada bagian ini adalah data yang berpotensi menghasilkan kesalahan, ketidakakuratan dalam pengklasifikasian berita. Dalam tahap ini dilakukan penghapusan karakter tanda baca dari setiap kata di dalam teks baik pada judul maupun konten berita dengan tujuan mendapatkan inti dari judul dan konten berita. Contoh *remove punctuation* pada data dapat dilihat pada tabel di bawah ini.

Tabel 3. Analisis untuk proses *punctuation removal*

Teks Berita	Hasil <i>Remove Punctuation</i>
<i>Employers and schools have no right to conduct "surveillance of a dorm room or a workerâ€™s cubicle."</i>	<i>Employers and schools have no right to conduct surveillance of a dorm room or a workers cubicle</i>

5. *Tokenization*

Tujuan penggunaan teknik tokenisasi sebagai *data transformation* adalah untuk memecah *text* berita baik judul maupun konten menjadi bagian-bagian kata yang disebut token. Penggunaan teknik ini bertujuan untuk membantu dalam menafsirkan makna teks dengan menganalisis urutan kata. Contoh *tokenization* pada data dapat dilihat pada tabel di bawah ini.

Tabel 4. Analisis untuk proses *tokenization*

Teks Berita	Hasil <i>Tokenization</i>
<i>employer and school have no right to conduct surveillance of a dorm room or a worker cubicl</i>	<i>['employers', 'and', 'schools', 'have', 'no', 'right', 'to', 'conduct', 'surveillance', 'of', 'a', 'dorm', 'room', 'or', 'a', 'workers', 'cubicl']</i>

6. *Stopword Removal*

Teknik data transformasi selanjutnya yang akan digunakan pada penelitian ini adalah *stopword removal*. Tujuan penggunaan teknik ini sebagai *data transformation* adalah untuk menghapus kata yang tidak memberikan banyak dampak terhadap sebuah kalimat, atau dengan kata lain menghapus informasi yang kurang berguna di dalam sebuah kalimat. *Stopword removal* menghapus kata yang bertindak sebagai penghubung atau konjungsi di dalam sebuah kalimat.

Tabel 5. Analisis untuk proses *stopword removal*

Teks Berita	Hasil <i>Tokenization</i>
<i>employer and school have no right to conduct surveillance of a dorm room or a worker cubicl</i>	['employer', 'school', 'right', 'conduct', 'surveillance', 'dorm', 'room', 'worker', 'cubicl']

7. *Lemmatization*

Tujuan penggunaan teknik lematisasi sebagai *data transformation* adalah untuk mengubah kata dalam sebuah kalimat menjadi bentuk dasarnya tanpa menghilangkan makna dari kalimat itu sendiri. Tahap ini juga berguna untuk mengurangi variasi kata dalam bentuk yang berbeda, dimana hal tersebut dapat mempengaruhi proses klasifikasi. Dalam proses lematisasi, dibutuhkan *WordNet corpus* yang berfungsi untuk membangun hubungan semantik terstruktur antar kata. Contoh *lemmatization* dapat dilihat pada tabel di bawah ini.

Tabel 6. Analisis untuk proses *lemmatization*

Teks Berita	Hasil <i>Lemmatization</i>
<i>Employers and schools have no right to conduct surveillance of a dorm room or a workers cubicle</i>	<i>employers and schools have no right to conduct surveillance of a dorm room or a workers cubicl</i>

2.1.5 Analisis Metode *Word Embedding*

Data berita yang sudah dipisahkan menjadi data *training* dan data *test* akan diubah menjadi sebuah vektor atau *array* yang terdiri dari kumpulan angka. Metode *word embedding* yang digunakan untuk merepresentasikan teks berita menjadi sebuah vektor yaitu menggunakan *word2vec* dengan arsitektur CBOW. CBOW akan bekerja pada model dengan memprediksi kata (target) yang diberikan konteks (kata sekitarnya), sehingga CBOW akan menghitung kedekatan kata sebelum dan kata sesudah untuk memberikan nilai bobot. Ukuran *windows* yang digunakan

pada saat melakukan *word embedding* harus diperhatikan, karena dengan memilih ukuran *windows* yang tepat maka hasil vektor yang akan dihasilkan oleh *CBOW* akan semakin baik. Hasil vektor pada proses *word embedding* ini akan menjadi masukan pada *bidirectional long short-term memory* (LSTM) sebelum mengklasifikasikan berita.

2.1.6 Analisis Metode Klasifikasi

Pengklasifikasian merupakan suatu proses untuk mengelompokkan suatu objek ke dalam suatu kelas atau kategori yang telah ditentukan sebelumnya. Pengklasifikasian dalam proyek ini menggunakan *bidirectional long short-term memory* (LSTM). Struktur dari LSTM adalah *sequence* yang dapat bekerja secara berkelanjutan dimana suatu kesatuan dianggap utuh atau tidak dapat dipotong. Dokumen teks berita yang dipotong atau dipisah akan merubah makna dari suatu kalimat yang terdapat di dalam berita tersebut. Data uji berupa berita yang sudah dilakukan tahap pra-proses akan digunakan sebagai inputan pengujian terhadap model yang sudah dibuat. Setiap berita yang sudah melalui tahap *training* dan *test* akan diklasifikasikan apakah berita tersebut masuk ke dalam kelas *fake* atau *real*.

3.1.7 Analisis Metode Bidirectional Long Short Term Memory

Klasifikasi berita palsu dengan *neural network* dilakukan dengan menggunakan metode *bidirectional long short term memory*. Metode *bidirectional lstm* memiliki arsitektur yang berbeda dengan *recurrent neural network* dalam mengelola setiap informasi yang diterima. Arsitektur *bidirectional lstm* tersusun dari gabungan arsitektur *long short term memory* dengan *bidirectional*. Pada *lstm* memiliki koneksi umpan balik atau *feedback connection* yang menghubungkan informasi yang sebelumnya dengan yang sedang diproses, dalam hal ini informasi tersebut adalah kata-kata yang diwakilkan dengan vektor. Selain itu, *lstm* memiliki *memory cell* yang dapat menyimpan informasi dalam memori untuk jangka waktu yang lama dan terdapat tiga *gates*, yaitu *forget gate*, *input gate*, *output gate*. Setiap informasi yang masuk akan melawati setiap *gate* dengan komputasi yang berbeda-beda pada setiap *gate*. Pada setiap *gate* terdapat *activation sigmoid* yang bertugas menyeleksi informasi yang masuk dengan mengeluarkan angka antara nol dan satu, dimana nilai nol memiliki arti bahwa informasi tidak diperbolehkan masuk sedangkan nilai satu berarti bahwa informasi diperbolehkan masuk. Setiap

hasil komputasi *gate* akan diteruskan ke dalam *cell memory* atau *cell state* dengan melakukan komputasi atau penjumlahan vector pada semua hasil setiap *gate*.

Pada penelitian ini menerapkan dua *hidden layer* dalam mengelola informasi yang diterima disertai penggunaan *forward layer* dan *backward layer* yang merupakan arsitektur *bidirectional recurrent neural network*. *Forward* melakukan proses maju pada setiap informasi yang ada atau dari masukkan awal sampai akhir, sedangkan *backward* akan melakukan proses mundur dari. Tujuan peneliti menggunakan arsitektur *bidirectional* untuk memperoleh informasi yang lebih akurat dengan melakukan perulangan komputasi terhadap setiap informasi yang telah diproses sebelumnya.

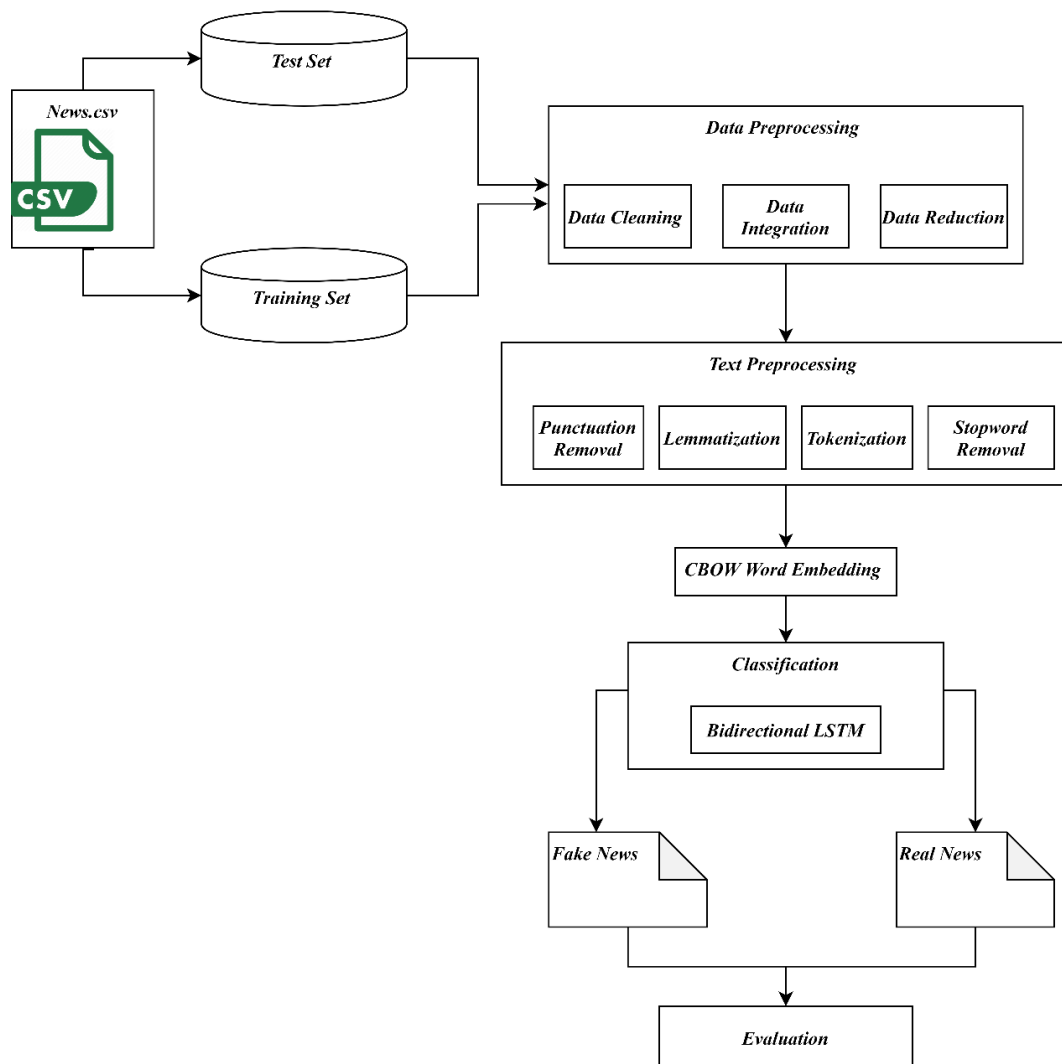
2.2 Desain

Pada bab ini dijelaskan desain yang akan di implementasikan oleh penulis. Desain yang dimaksud terdiri dari desain umum sistem, desain pengumpulan data, desain pemrosesan data, desain algoritma, desain klasifikasi berita palsu, desain evaluasi performa *Bidirectional LSTM*, dan desain eksperimen.

2.2.1 Desain Umum Sistem

Pada subbab ini akan dijelaskan mengenai gambaran umum dari sistem yang akan diimplementasikan. Sistem yang akan dibangun pada penelitian ini adalah sebuah system yang akan mendeteksi sebuah berita dan mengklasifikasikan berita tersebut ke dalam berita fakta atau berita palsu. Sistem ini akan dibangun menggunakan pendekatan *bidirectional lstm* yang dapat mendeteksi berita melalui dua arah, *forward* dan *backward*. Secara umum sistem yang akan diimplementasikan dapat dilihat pada gambar berikut.

Desain Umum Sistem



Gambar 2 Desain Umum Sistem

Adapun proses yang akan dilakukan pada gambaran umum sistem diatas, adalah sebagai berikut:

1. *Split Data*

Split data merupakan proses membagi data. Salah satu keputusan pertama yang harus diambil saat memulai untuk membuat pemodelan yaitu bagaimana memanfaatkan data yang ada. Data berita yang sudah dikumpulkan akan dibagi menjadi *training data* dan *test data*.

2. *Data Preprocessing*

Data Preprocessing merupakan tahapan yang penting untuk membersihkan data yang akan digunakan untuk melatih model pengklasifikasian. Tahap yang akan dilakukan dalam *data preprocessing* yaitu *data cleaning*, *data integration*, dan *reduction*.

3. *Text Processing*

Text Processing merupakan tahapan awal untuk mempersiapkan teks berita yang akan digunakan menjadi data yang siap untuk diolah lebih lanjut. Tahap yang akan dilakukan dalam *text processing* yaitu *punctuation removal*, *lemmatization*, *tokenization*, *stopword removal*.

4. *Word Embedding*

Word Embedding merupakan tahapan untuk konversi sebuah teks menjadi angka.

5. *Classification*

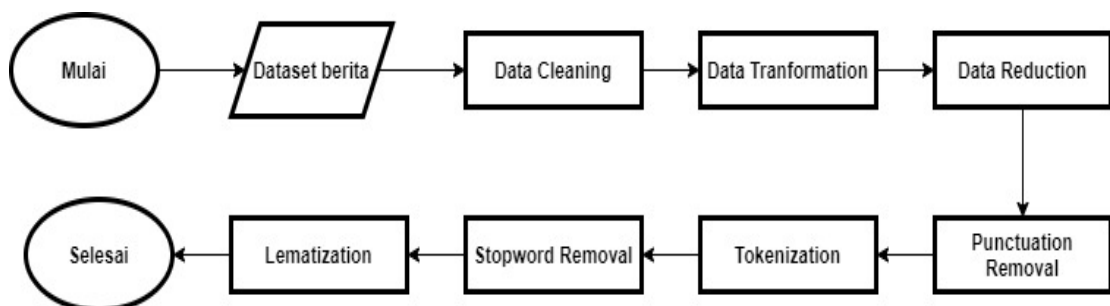
Mengklasifikasikan data ke dalam label kelas yang sesuai dengan ketentuan yang telah dibuat.

6. *Evaluation*

Pada tahap ini dilakukan evaluasi dan pembahasan hasil eksperimen pendeteksi berita palsu (*fake news*). Mengukur kinerja suatu sistem, sehingga dapat menggambarkan seberapa baik sistem tersebut untuk mengklasifikasikan data.

2.2.2 Desain Prampemrosesan Data

Setelah *dataset* berita sudah diperoleh, maka *dataset* tersebut akan diolah atau diproses. Adapun tahapan pemrosesan data pada penelitian ini terdiri dari *data cleaning*, *data transformation*, *data reduction*, *punctuation removal*, *lemmatization*, *Tokenization* dan *Stopword Removal*. Berikut adalah penjelasan lebih rinci terkait pemrosesan data.

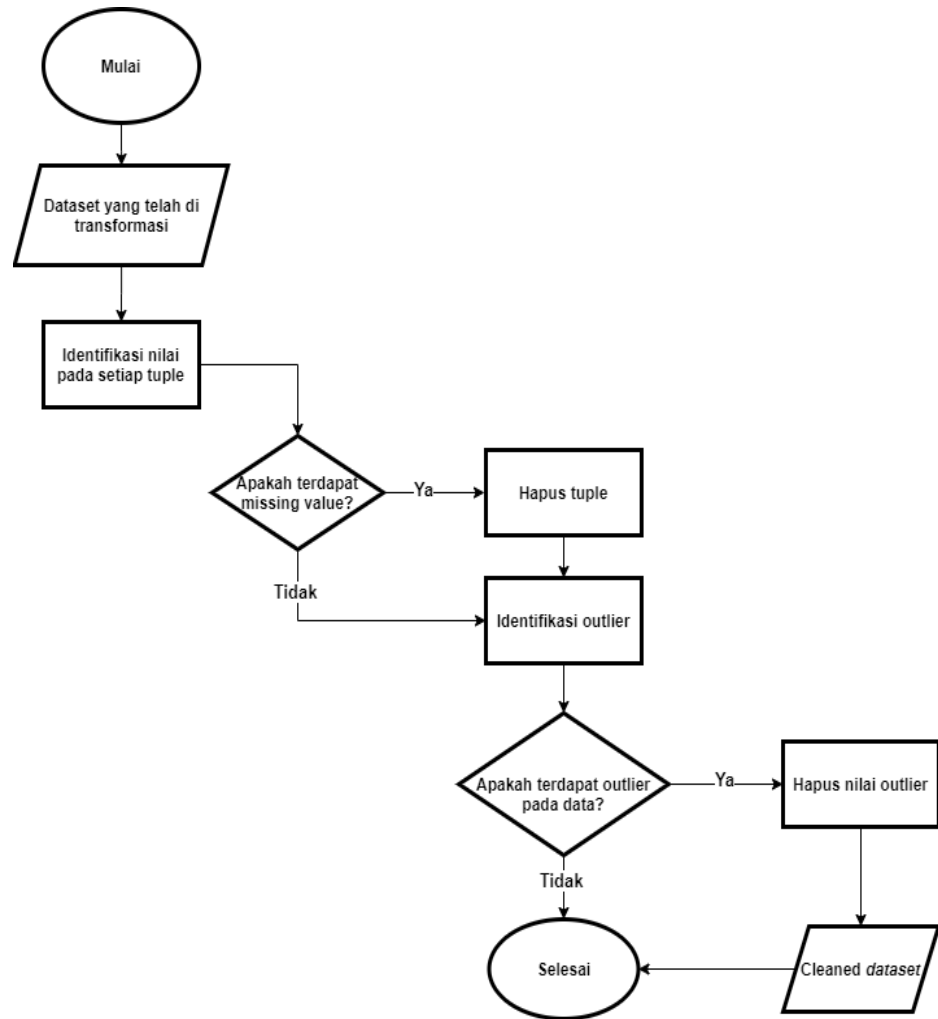


Gambar 3 Prapemrosesan Data

1. *Data Cleaning* digunakan untuk memeriksa dan memastikan tidak ada *missing value* pada *dataset*.
2. *Data Transformation* digunakan untuk menormalisasi nilai pada atribut label dengan metode *encoding categorical value*.
3. *Data Reduction* digunakan untuk mengurangi jumlah dimensi pada data.
4. *Punctuation Removal* digunakan untuk menghapus semua tanda baca dan simbol pada kalimat di *dataset*.
5. *Tokenization* digunakan untuk memecah kalimat menjadi token, agar lebih mudah untuk di proses di tahapan selanjutnya.
6. *Stopword removal* digunakan untuk menghilangkan kata pada kalimat, dimana kata tersebut biasanya merupakan kata penghubung.
7. *Lemmatization* digunakan untuk mengubah semua kata pada kalimat yang ada pada *dataset*, dimana kata tersebut terbentuk secara gramatikal. Kata tersebut diubah menjadi bentuk kata dasarnya tanpa mengubah makna dari kalimat itu sendiri.

2.2.2.1 Data Cleaning

Gambaran kerja proses *data cleaning* dalam penelitian ini ditunjukkan pada gambar di bawah ini.

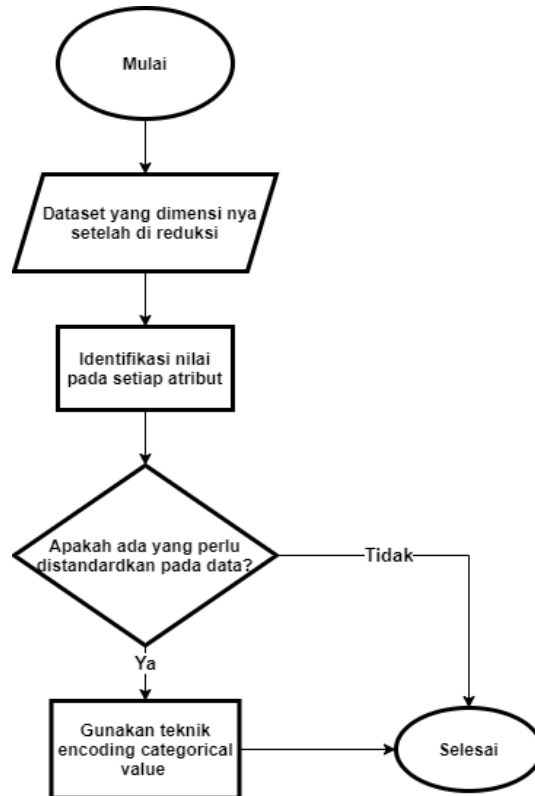


Gambar 4 Proses Pembersihan Data

1. Proses *data cleaning* yang dilakukan pada penelitian ini adalah melakukan pemeriksaan terhadap *missing value* pada data. *Data cleaning* menggunakan *data tranformatted* sebagai *inputan* dalam proses.
2. Selanjutnya dilakukan identifikasi nilai dari setiap *tuple* di *dataset*.
3. Jika terdapat *missing value* pada *tuple* maka baris akan dihapus. Hal ini bertujuan untuk menghindari bias.
4. Jika tidak terdapat *missing value*, maka selanjutnya dilakukan pemeriksaan *outlier* pada data. Jika ditemukan *outlier* maka *outlier* akan dihapus agar dapat di proses pada tahap pemrosesan selanjutnya. Namun sebaliknya, jika tidak terdapat *outlier* maka proses *data cleaning* selesai dilakukan.

2.2.2.1 Data Transformation

Gambaran kerja proses *data transformation* dalam penelitian ini ditunjukkan pada gambar di bawah ini.



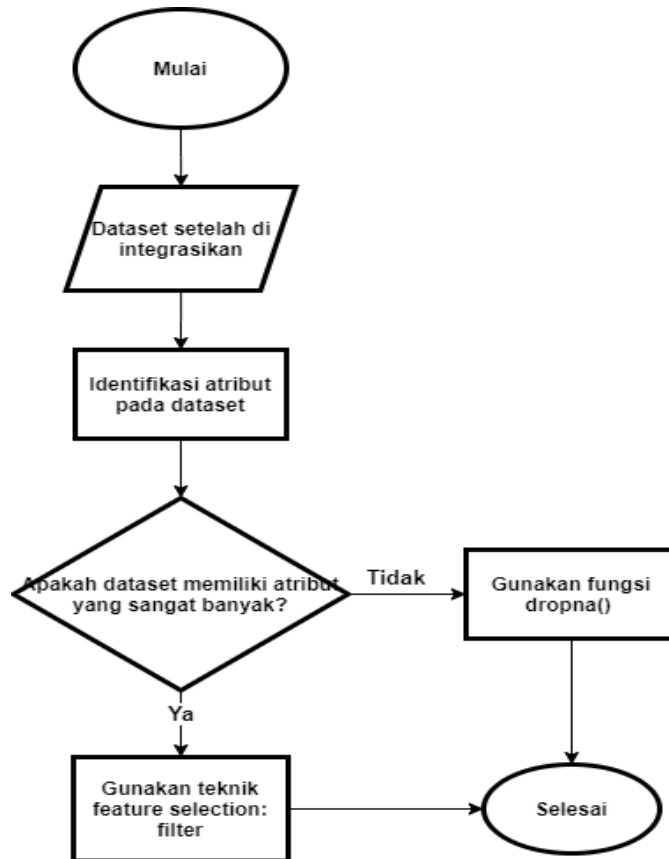
Gambar 5 Proses Transformasi Nilai pada Atribut

1. Jenis *data transformation* yang dilakukan pada penelitian ini adalah mentransformasi nilai atribut kategorikal yakni label menjadi bentuk standard yang disepakati. Proses *data transformation* menggunakan *data reduced* sebagai *inputan* dalam proses.
2. Kemudian identifikasi nilai dari atribut label, apakah setelah melalui proses integrasi terdapat ketidak konsistenan nilai pada atribut. Jika terdapat ketidak konsistenan pelabelan, maka tentukan standard pelabelan berita. Contoh standard pelabelan dapat berupa biner (0 sebagai berita palsu dan 1 sebagai berita fakta) atau (*Real* dan *Fake*).
3. Setelah standar pelabelan disepakati, maka gunakan teknik *encoding categorical value* untuk melakukan standarisasi pada label. Jenis dari *encoding categorical value* yang akan diterapkan pada penelitian ini adalah *label encoding*.

4. Jika nilai atribut label konsisten maka tidak perlu dilakukan transformasi.

2.2.2.2 Data Reduction

Gambaran kerja proses *data reduction* dalam penelitian, ditunjukkan pada gambar di bawah ini.

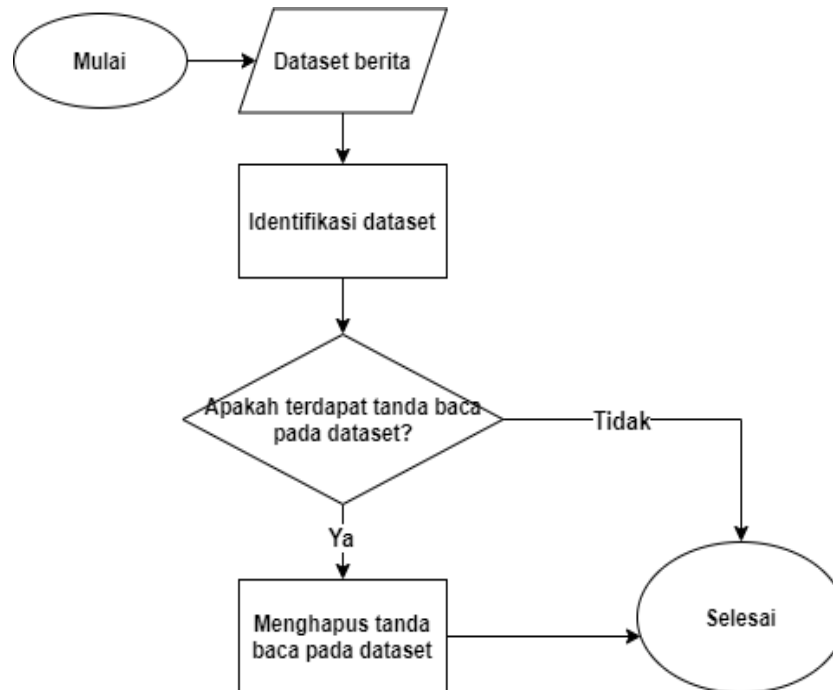


Gambar 6 Proses Reduksi Data

1. Jenis *data reduction* yang dilakukan pada penelitian ini adalah *dimensionality reduction*. *Data reduction* menggunakan *data integrated* sebagai *inputan* dalam proses.
2. Kemudian dilakukan identifikasi atribut pada *dataset* yang telah diintegrasikan.
3. Jika ditemukan banyak atribut yang tidak diperlukan dalam penelitian, maka digunakan teknik *dimensionality reduction* yakni *feature selection (filter)* untuk menghapus atribut tersebut.
4. Sebaliknya, jika sedikit jumlah atribut yang tidak diperlukan pada penelitian maka gunakan fungsi *dropna()* untuk menghapus atribut tersebut.

2.2.2.3 Punctuation Removal

Gambaran kerja proses *punctuation removal* dalam penelitian ini ditunjukkan pada gambar di bawah ini.



Gambar 7 Proses *Punctuation Removal*

1. Pada bagian ini akan dilakukan penghapusan tanda baca dan juga simbol pada *dataset*. Data yang merupakan hasil proses *data cleaning* akan menjadi *input* untuk proses *punctuation removal*.
2. Kemudian lakukan identifikasi *dataset* berita untuk mengetahui apakah di dalam *dataset* terdapat tanda baca dan juga simbol.
3. Jika terdapat tanda baca dan juga simbol pada *dataset* maka hal yang perlu dilakukan adalah menghapusnya, namun jika sebaliknya maka tidak perlu dilakukan proses *punctuation removal*.

2.2.2.4 Tokenization

Gambaran kerja proses *tokenization* dalam penelitian ini ditunjukkan pada gambar di bawah ini.

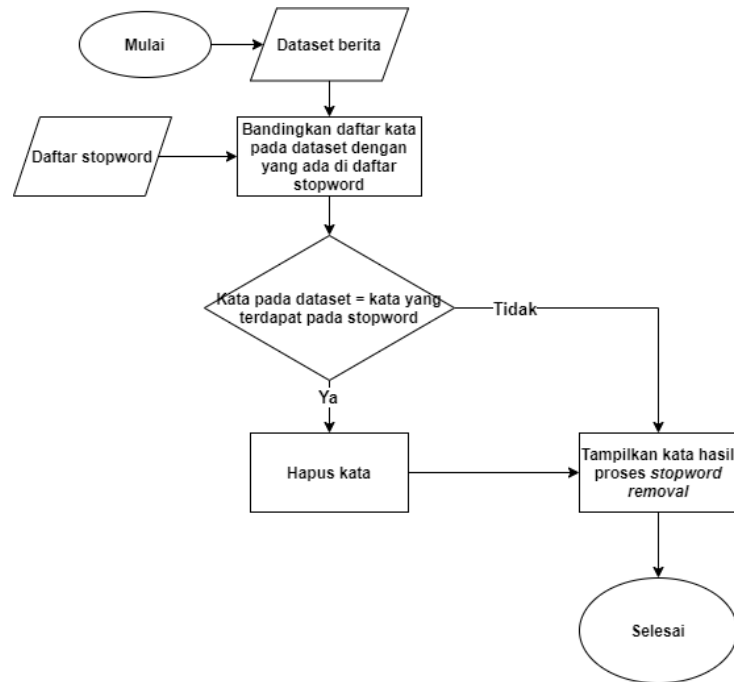


Gambar 8 Proses *Tokenization*

1. *Tokenization* dilakukan untuk mempartisi kalimat menjadi token. *Tokenization* menggunakan data yang telah melalui proses lematisasi sebagai *input* dalam proses.
2. Hal pertama yang dilakukan dalam proses ini adalah menyiapkan data berupa judul berita yang akan digunakan , dan kemudian lakukan proses *tokenization* pada setiap kalimat pada teks.

2.2.2.6 *Stopword Removal*

Gambaran kerja proses *stopword removal* dalam penelitian ini ditunjukkan pada gambar di bawah ini.

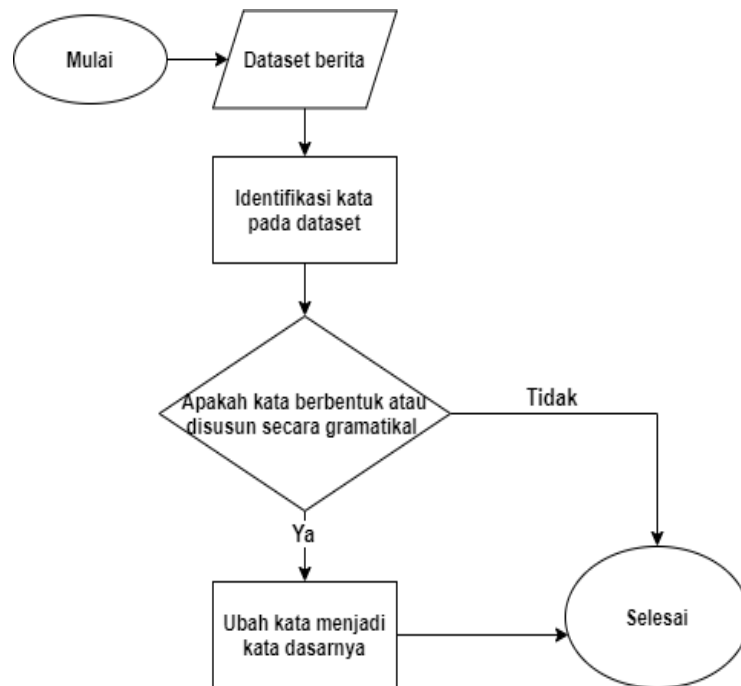


Gambar 9 Proses *Stopword Removal*

1. *Stopword removal* dilakukan untuk menyederhanakan kalimat, dimana proses ini menyederhanakan kalimat dengan menghapus setiap kata penghubung yang ada pada kalimat.
2. Hal pertama yang dilakukan adalah menyiapkan *dataset* dan kemudian dibandingkan dengan daftar kata dalam *stopword*. Apabila kata dalam kalimat sama dengan kata yang termasuk di dalam daftar *stopword*, maka kata tersebut akan dihapus.
3. Selanjutnya tampilkan kalimat yang merupakan hasil dari proses *stopword removal*.

2.2.2.7 Lemmatization

Gambaran kerja proses *lemmatization* dalam penelitian ini ditunjukkan pada gambar di bawah ini.



Gambar 10 Proses Lemmatization

1. Pada tahap ini dilakukan lemmatisasi dengan tujuan mengubah sebuah kata dalam kalimat, menjadi kata dasarnya tanpa mengubah makna dari kalimat. Hal ini dilakukan untuk mengambil inti yang penting dari sebuah kalimat.
2. Identifikasi kata dalam kalimat yang terdapat pada *dataset*, dilakukan untuk mengetahui apakah kata disusun secara gramatikal (*grammar*) atau tidak. Hal ini perlu dilakukan karena kalimat yang digunakan merupakan kalimat dalam bahasa inggris dan juga memiliki aspek *grammar* di dalamnya, sehingga perlu ditransformasikan ke dalam bentuk kata dasarnya.
3. Jika kata dalam kalimat tidak disusun secara gramatikal di dalam bahasa inggris, maka tidak perlu dilakukan tahapan *lemmatization*.

2.3 Implementasi

Pada tahap ini dilakukan pengimplementasian komponen-komponen yang telah dirancang atau didesain pada tahap sebelumnya, untuk menguji ketepatan rancangan yang dibuat. Pada penelitian ini juga dilakukan eksperimen yang mana diharapkan dapat menghasilkan keakurasian yang cukup tinggi melalui pendekatan *bidirectional long short term memory*.

2.3.1 Implementasi Prapemrosesan data

Prapemrosesan data terdiri dari beberapa proses. Hasil implementasi tahapan - tahapan prapemrosesan data dapat di lihat pada gambar di bawah ini.

2.3.1.1 Implementasi Data Cleaning

Pada tahap ini akan dilakukan beberapa pendekatan umum untuk menangani data yang hilang, dengan melakukan penghapusan baris terhadap nilai yang bernilai *null*.

```
for col in data_train.columns:
    print(col, data_train[col].isnull().sum())

for col in data_test.columns:
    print(col, data_test[col].isnull().sum())
```

2.3.1.2 Implementasi Data Transformation

Implementasi *data Transformation* dilakukan untuk mentransformasikan isi dari atribut label berita yaitu : *Fake* menjadi 0 dan *Real* menjadi 1, dengan tujuan untuk mengkonsistenkan semua label pada berita, untuk mengetahui sebaran berita *fake* maupun *real*. Teknik tranformasi yang digunakan adalah *encoding categorical value*.

```
data_train.label_fnn.replace({"fake":0}, inplace= True)
data_train.label_fnn.replace({"real":1}, inplace= True)

data_test.label_fnn.replace({"fake":0}, inplace= True)
data_test.label_fnn.replace({"real":1}, inplace= True)
```

Untuk menampilkan data yang telah di transformasi dilakukan

```
data_train.head()
```

```
data_test.head()
```

2.3.1.3 Implementasi Data Reduction

Atribut pada data terdiri dari *id*, *date*, *speaker*, *sources*, *paragraph_based_content*, *fullText_based_content*, *statement* dan *label_fnn*, data *reduction* dilakukan untuk mengurangi atribut yang dimiliki data *train* dan data *test*. Syntax yang digunakan adalah *drop*.

```
#dimensional reduction atau pengurangan dimensi pada data train
data_train =
data_train.drop(['id', 'date', 'speaker', 'sources', 'paragraph_based_content', 'fullText based content'], axis = 1)
```

```
#dimensional reduction atau pengurangan dimensi pada data test
data_test =
data_test.drop(['id', 'date', 'speaker', 'sources', 'paragraph_based_content', 'fullText based content'], axis = 1)
```

Untuk menampilkan data, dilakukan:

```
data_train.head()
```

```
data_test.head()
```

2.3.1.4 Implementasi Punctuation Removal

Tahapan pra-pemrosesan *Punctuation Removal*, dilakukan untuk menghapus semua tanda baca pada data *train* maupun data *test*.

```
data_train["statement"]=data_train['statement'].str.replace(r'[\^\\w\\s]+', '')
data_test["statement"]=data_test['statement'].str.replace(r'[\^\\w\\s]+', '')
```

2.3.1.5 Implementasi Tokenization

Menggunakan *tokenizer* yang disediakan oleh *library* NLTK, yaitu ‘punkt’, menjadikan setiap data teks terbagi menjadi daftar kalimat yang terdiri dari token - token kata.

```
data_train['statement']=data_train['statement'].apply(nltk.tokenize.WhitespaceTokenizer().tokenize)
data_test['statement']=data_test['statement'].apply(nltk.tokenize.WhitespaceTokenizer().tokenize)
```

2.3.1.6 Implementasi Stopword Removal

Implementasi *Stopword removal* (Filtering) pada tahapan Pra-pemrosesan data dilakukan untuk mengambil kata-kata yang penting dari hasil token (*wordlist*) dan membuang kata-kata yang kurang penting (*stoplist*),

```
stop = stopwords.words('english')
data_train['statement'] = data_train['statement'].apply(lambda x:
[item for item in x if item not in stop])
data_test['statement'] = data_test['statement'].apply(lambda x: [item
```

Yang sebelumnya telah di *set* dalam bahasa inggris.

```
from nltk.corpus import stopwords
stop=set(stopwords.words('english'))
```

2.3.1.7 Implementasi Lemmatization

Implementasi *Lematization* akan mengembalikan bentuk kamus dari kata, dengan mereduksi bentuk kata agar menjadi suatu kata yang secara linguistik *valid*.

```
from nltk.stem.wordnet import WordNetLemmatizer
def lema_words(text):
    wnl=WordNetLemmatizer()
```

```
lemmatizer = WordNetLemmatizer()

data_train['statement'] = data_train['statement'].apply(lambda x:
[lemmatizer.lemmatize(y) for y in x])
data_test['statement'] = data_test['statement'].apply(lambda x:
```

2.3.2 Implementasi Word2Vec

Pada tahapan implementasi *Word2Vec*, pembentukan model *Word2Vec* dapat dilihat pada kode program berikut

```
w2v_model = gensim.models.Word2Vec(data_train['statement'],
size=100, sg=0, min_count=1, window=5, iter =10)
w2v_weights = w2v_model.wv.vectors
vocab_size, embedding_size = w2v_weights.shape
```

Pada kode program tersebut, ditunjukkan bahwa tahapan implementasi *Word2Vec* diawali dengan melakukan *import Word2Vec* dari *library* gensim. Model yang dibangun memiliki 6 parameter, diantaranya

Parameter	Keterangan
<i>data_train</i>	merupakan kumpulan kata yang dijadikan sebagai bahan pemodelan

<i>size</i>	merupakan ukuran dimensi yang akan digunakan dalam membuat vektor, yaitu 100
<i>sg</i>	nilai <i>sg</i> = 0, menyatakan bahwa arsitektur yang digunakan adalah <i>CBOW</i>
<i>min_count</i>	merupakan jumlah minimum kemunculan kata dalam koleksi teks. <i>min_count</i> di set = 1. Apabila kemunculan kata kurang dari jumlah minimum, maka kata tersebut akan diabaikan.
<i>window</i>	merupakan jarak maksimum antara target dengan kata-kata yang berada di sekitar target.
<i>iter</i>	merupakan jumlah <i>iterasi</i> atau <i>epoch</i> untuk melatih model <i>Word2Vec</i>

Setelah membangun model, akan di definisikan fungsi untuk menentukan *average sentence vector* dari *list of sentence*.

```
# definisikan fungsi untuk membuat averaged sentence vector dari list of
sentence tokens
def buildWordVector(tokens, size):
    vec = np.zeros(size).reshape((1, size))
    count = 0.
    for word in tokens:
        try:
            # jumlahkan
            vec += w2v_model.wv.__getitem__(word).reshape((1, size))
            count += 1.
        except KeyError: # handling the case where the token is not
                        # in the corpus. useful for testing.
            continue
    # bagikan dengan total word dalam sentence
    if count != 0:
        vec /= count
    return vec
```

Dan dilakukan konversi terhadap ‘*statement*’ pada *data-train* dan *data_test* kedalam *list of vectors*.

```
# definisikan fungsi untuk membuat averaged sentence vector dari list of
sentence tokens

# konversi data_train['statement'] dan data_test['statement'] ke dalam
list of vectors
X_train_w2v = np.concatenate([buildWordVector(z, 100) for z in map(lambda
x: x, data_train['statement'])])

def word_token(word):
    try:
        return w2v_model.wv.vocab[word].index

    except KeyError:
        return 0
def token_word(token):
    return w2v_model.wv.index2word[token]
```

2.3.3 Implementasi Bidirectional LSTM

Dalam membuat test Bidirectional LSTM, model dibagi menjadi 2, dengan label berjumlah 2, yaitu 1 dan 0

```
VALID_PER = 0.2
total_samples = set_x.shape[0]
n_val = int(VALID_PER * total_samples)
n_train = total_samples - n_val

random_i = random.sample(range(total_samples), total_samples)
train_x = set_x[random_i[:n_train]]
train_y = set_y[random_i[:n_train]]
val_x = set_x[random_i[n_train:n_train+n_val]]
val_y = set_y[random_i[n_train:n_train+n_val]]

print("Train Shapes - X: {} - Y: {}".format(train_x.shape,
train_y.shape))
print("Val Shapes - X: {} - Y: {}".format(val_x.shape, val_y.shape))

categories, ccount = np.unique(train_y, return_counts=True)
#jumlah label yang digunakan yaitu 1 dan 0 (berjumlah 2)
```

Juga dapat membuat sebuah model *sequential* secara bertahap melalui metode *add()* juga menentukan konfigurasinya (optimizer, loss, metrics), dan setelah model dibangun, kita dapat memanggil metode *summary()* untuk menampilkan isinya seperti yang ditunjukkan pada kode program berikut

```

model = Sequential()
model.add(Embedding(input_dim=vocab_size,
                    output_dim=embedding_size,
                    weights=[w2v_weights],
                    input_length=MAX_SEQUENCE_LENGTH,
                    mask_zero=True,
                    trainable=False))

model.add(Bidirectional(LSTM(100)))
model.add(Dense(n_categories, activation='softmax'))

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy', metrics=['accuracy'])
print(model.summary())

```

Fit() dipanggil untuk melatih model dengan membagi data menjadi sekumpulan *batch size* dan berulang kali melakukan iterasi pada seluruh kumpulan data untuk *5 epoch*. *Fit()* dilakukan terhadap model 1 dengan kode program berikut

```

epochs = 5
batch_size = 64
historys = model.fit(train_x, train_y, epochs=epochs,
                    batch_size=batch_size,
                    validation_data=(val x, val y), verbose=1)

```

Dan dilakukan terhadap model 2 dengan kode program sebagai berikut

```

epochs = 5
batch_size = 64
train =model2.fit(X_train, Y_train, epochs=epochs,
                batch_size=batch_size,validation_split=0.1,callbacks=[EarlyStopping
                (monitor='val_loss', patience=3, min_delta=0.0001)])

```

```

test =model2.fit(test_X_train, test_Y_train, epochs=epochs,
                batch_size=batch_size,validation_split=0.1,callbacks=[EarlyStopping
                (monitor='val_loss', patience=3, min_delta=0.0001)])

```

2.4 Hasil

Pada tahap ini dilakukan evaluasi dan pembahasan hasil eksperimen pendeteksi berita palsu (*fake news*). Mengukur kinerja suatu system klasifikasi merupakan hal yang penting, sehingga dapat menggambarkan seberapa baik sistem tersebut untuk mengklasifikasikan data. Confusion Matrix merupakan pengukur kinerja klasifikasi yang paling umum. Evaluasi dan pembahasan hasil eksperimen

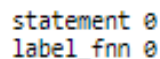
dilakukan untuk mengetahui apakah eksperimen memiliki performansi yang cukup baik untuk menentukan apakah suatu berita dikategorikan palsu atau tidak, berdasarkan jumlah (*number of neuron Accuracy*), *Precision*, *Recall* dan *Computational time* yang diharapkan tinggi. Apabila masih terdapat *error* maka dilakukan perbaikan pada penelitian.

2.4.1 Hasil Pra pemrosesan data

Tahapan dalam melakukan *text preprocessing* dilakukan dalam 7 tahapan, yaitu *data cleaning*, *data transformation*, *data reduction*, *punctuatin removal*, *tokenization*, *stopwrđ removal*, dan *lemmatization*

2.4.1.1 Data Cleaning

Tahapan *data cleaning* diimplementasikan untuk mengatasi adanya *missing value*, maupun *noisy value*, dengan menghapus baris data yang memiliki nilai yang kosong. Hasil dari penerapan *data cleaning* pada tahapan pra pemrosesan data, dapat dilihat pada gambar berikut



```
statement 0
label_fnn 0
```

Dari gambar tersebut, dapat dilihat bahwa tidak ada lagi adanya nilai yang kosong pada atribut *state* dan *label_fnn*

2.4.1.2 Data Transformation

Data transformation pada tahapan pra pemrosesan data dilakukan untuk mengkonsistenkan semua label pada beritadengan mentransformasikan isi dari atribut label berita, dimana *fake news* diberi label 0 dan *real news* diberi label 1. Berikut merupakan hasil dari penerapan *data transformation* pada data *train*

	id	date	speaker	statement	sources	paragraph_based_content	fullText_based_content	label_fnn
0	3108	2011-01-25T08:00:00-05:00	Joe Wilkinson	A national organization says Georgia has one o...	[http://www.ajc.com/news/georgia-politics-ele...	[A coalition of government watchdog groups la...	A coalition of government watchdog groups last...	0
1	5655	2012-04-02T11:42:20-04:00	Rick Scott	Says Barack Obama's health care law "will be t...	[http://www.youtube.com/watch?v=TaC0mKApt9Q&f...	[As Supreme Court justices embarked on three ...	As Supreme Court justices embarked on three da...	0
2	3508	2011-04-01T09:49:00-04:00	J.D. Alexander	Says the Southwest Florida Water Management Di...	[http://www.tampabay.com/news/politics/gubern...	[Here's a new one: The Senate budget committe...	Here's a new one: The Senate budget committee ...	0
3	3450	2011-03-21T12:20:02-04:00	Paul Ryan	"The Congressional Budget Office has this econ...	[http://www.cnn.com/2011/POLITICS/03/17/gop.b...	[Recently, House Budget chairman Paul Ryan, R...	Recently, House Budget chairman Paul Ryan, R-W...	1
4	4778	2011-11-13T07:30:00-05:00	Rodney Frelinghuysen	Says the Treasury Department "says 41 percent ...	[http://frelinghuysen.house.gov/index.cfm?sec...	[The millionaires' tax proposal made its late...	The millionaires' tax proposal made its latest...	0

dan hasil dari penerapan *data transformation* pada data *test*

	id	date	speaker	statement	sources	paragraph_based_content	fullText_based_content	label_fnn
0	1678	2010-04-11T16:37:40-04:00	Jon Kyl	"President Obama himself attempted to filibust...	[http://abcnews.go.com/ThisWeek/video/supreme...	[U.S. Supreme Court Justice John Paul Stevens...	U.S. Supreme Court Justice John Paul Stevens a...	1
1	1820	2010-05-23T18:11:09-04:00	Michael Steele	In Hawaii, "they don't have a history of throw...	[http://www.starbulletin.com/news/bulletin/04...	[On ABC's This Week, the chairmen of the Repu...	On ABC's This Week, the chairmen of the Republ...	1
2	1624	2010-03-26T10:24:21-04:00	John Boehner	"Our national debt ... is on track to exceed t...	[http://www.desmoinesregister.com/article/201...	[Ever since Barack Obama became president and...	Ever since Barack Obama became president and b...	1
3	1578	2010-03-12T11:45:14-05:00	America's Health Insurance Plans	"Health insurance companies' costs are only 4 ...	[http://www.youtube.com/watch?v=4O8CxZ1OD58',...	[As the battle over health care reform approa...	As the battle over health care reform approach...	1
4	1770	2010-05-07T11:54:44-04:00	Michael Bloomberg	"We can prevent terror suspects from boarding ...	[http://www.huffingtonpost.com/michael-bloomb...	[In the wake of a foiled car bomb attempt in ...	In the wake of a foiled car bomb attempt in Ti...	1

Hasil dari tahapan transformasi data yaitu, *label_fnn* berubah sesuai dengan aturan transformasi yang sebelumnya telah di definisikan.

2.4.1.3 Data Reduction

Data reduction dilakukan untuk mengurangi atribut dengan cara menghapus atribut yang tidak diperlukan dalam membangun sebuah model. Hasil dari implementasi data reduction terhadap data train dan data test adalah atribut *statement* dan *label_fnn*, yang ditampilkan pada gambar berikut

	statement	label_fnn
0	A national organization says Georgia has one o...	0
1	Says Barack Obama's health care law "will be t...	0
2	Says the Southwest Florida Water Management Di...	0
3	"The Congressional Budget Office has this econ...	1
4	Says the Treasury Department "says 41 percent ...	0

	statement	label_fnn
0	"President Obama himself attempted to filibust...	1
1	In Hawaii, "they don't have a history of throw...	1
2	"Our national debt ... is on track to exceed t...	1
3	"Health insurance companies' costs are only 4 ...	1
4	"We can prevent terror suspects from boarding ...	1

2.4.1.4 Punctuation Removal

Punctuation Removal dilakukan untuk memperoleh data yang bersih dan siap untuk diproses pada tahapan selanjutnya. Hasil yang telah diperoleh dari tahapan implementasi *punctuation removal* terhadap data *train* ditampilkan pada gambar berikut

	statement	label_fnn
0	A national organization says Georgia has one o...	0
1	Says Barack Obamas health care law will be the...	0
2	Says the Southwest Florida Water Management Di...	0
3	The Congressional Budget Office has this econo...	1
4	Says the Treasury Department says 41 percent o...	0

Dan hasil yang telah diperoleh dari tahapan implementasi *punctuation removal* terhadap data *test* ditampilkan pada gambar berikut

	statement	label_fnn
0	President Obama himself attempted to filibuste...	1
1	In Hawaii they dont have a history of throwing...	1
2	Our national debt is on track to exceed the s...	1
3	Health insurance companies costs are only 4 pe...	1
4	We can prevent terror suspects from boarding a...	1

Berdasarkan hasil implementasi yang telah diperoleh, dapat dilihat bahwa penggunaan tanda baca pada data telah dihapus. Berdasarkan hasil implementasi *punctuation removal* dapat diperoleh dengan baik, yang kemudian akan lebih mudah untuk di tokenisasi.

2.4.1.5 Tokenization

Tahapan tokenisasi dilakukan untuk memecah setiap teks berita menjadi token-token agar lebih mudah untuk diproses pada tahapan stopwords removal. Hasil yang diperoleh pada tahapan implementasi tokenisasi terhadap data *train* ditampilkan pada gambar dibawah ini

	statement	label_fnn
0	[A, national, organization, says, Georgia, has...	0
1	[Says, Barack, Obamas, health, care, law, will...	0
2	[Says, the, Southwest, Florida, Water, Managem...	0
3	[The, Congressional, Budget, Office, has, this...	1
4	[Says, the, Treasury, Department, says, percen...	0

Dan hasil yang telah diperoleh dari tahapan implementasi tokenisasi terhadap data *test* ditampilkan pada gambar berikut

	statement	label_fnn
0	[President, Obama, himself, attempted, to, fil...	1
1	[In, Hawaii, they, dont, have, a, history, of,...	1
2	[Our, national, debt, is, on, track, to, excee...	1
3	[Health, insurance, companies, costs, are, onl...	1
4	[We, can, prevent, terror, suspects, from, boa...	1

Berdasarkan hasil implementasi Tokenisasi yang telah dilakukan, dapat dilihat bahwa setiap teks berita yang dimiliki atribut statement telah berubah menjadi token.

2.4.1.6 Stopword Removal

Setelah dilakukan tahapan implementasi *stopwords removal*, data yang tersedia hanyalah kata-kata yang penting (*wordlist*) dari hasil token, karena pada tahapan implementasi, kata-kata yang kurang bermakna (*stoplist*) telah dibuang. Berikut merupakan hasil dari tahapan implementasi *stopword removal* terhadap data *train*

	statement	label_fnn
0	[A, national, organization, says, Georgia, one...	0
1	[Says, Barack, Obamas, health, care, law, bigg...	0
2	[Says, Southwest, Florida, Water, Management, ...	0
3	[The, Congressional, Budget, Office, economic,...	1
4	[Says, Treasury, Department, says, percent, bu...	0

Dan hasil yang telah diperoleh dari tahapan implementasi *stopword removal* terhadap data *test* ditampilkan pada gambar berikut

	statement	label_fnn
0	[President, Obama, attempted, filibuster, Just...	1
1	[In, Hawaii, dont, history, throwing, incumben...	1
2	[Our, national, debt, track, exceed, size, ent...	1
3	[Health, insurance, companies, costs, percent,...	1
4	[We, prevent, terror, suspects, boarding, airp...	1

2.4.1.7 Lemmatization

Pada tahapan implementasi Lemmatisasi, bentuk data akan direduksi sehingga hanya akan menyisakan bentuk dasar dari data. Berikut merupakan tampilan hasil yang diperoleh dari tahapan lematisasi pada data *train*

	statement	label_fnn
0	[A, national, organization, say, Georgia, one,...	0
1	[Says, Barack, Obamas, health, care, law, bigg...	0
2	[Says, Southwest, Florida, Water, Management, ...	0
3	[The, Congressional, Budget, Office, economic,...	1
4	[Says, Treasury, Department, say, percent, bus...	0

Dan hasil yang telah diperoleh dari tahapan implementasi lematisasi terhadap data *test* ditampilkan pada gambar berikut

	statement	label_fnn
0	[President, Obama, attempted, filibuster, Just...	1
1	[In, Hawaii, dont, history, throwing, incumben...	1
2	[Our, national, debt, track, exceed, size, ent...	1
3	[Health, insurance, company, cost, percent, he...	1
4	[We, prevent, terror, suspect, boarding, airpl...	1

Berdasarkan gambar tersebut, dapat dilihat bahwa bentuk data telah berubah menjadi bentuk dasar.

2.4.2 Hasil Word2Vec

Berdasarkan implementasi menggunakan Word2Vec yang telah dilakukan pada subbab 2.3.2, arsitektur yang digunakan adalah Skip-Gram dengan parameter *window* = 5, *min_count* = 1, *dimension* = 100, dan *iter* = 100 dengan jumlah kata yang unik adalah 14.077. jumlah kata yang diolah oleh Word2Vec, tidak sama dengan jumlah *vocabulary* karena banyak dari *vocabulary* memiliki frekuensi kemunculan dengan *min_count* yaitu 1, jika frekuensi kata tertentu urang dari jumlah minimum, maka kata tersebut akan diabaikan.

Found 14077 unique tokens.

Pada model 2, panjang *sequence* yang diperoleh dari atribut *statement* pada data train adalah


```
(13690, 250) (13690, 2)
(1522, 250) (1522, 2)
```

Dan, panjang *sequence* yang diperoleh dari atribut *statement* pada data test adalah

```
(843, 250) (843, 2)
(211, 250) (211, 2)
```

Sementara pada model 1, panjang *padding sequence* yang diperoleh adalah sama, yaitu

```
15212 15212
```

2.4.3 Hasil Bidirectional LSTM

Model yang telah dibangun pada tahapan implementasi, dapat dipanggil dengan metode *summary()* untuk menampilkan isinya. Hasil dari model 1 yang telah dibangun pada tahapan implementasi *bidirectional* LSTM adalah seperti berikut

```
Model: "sequential"

```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 250, 100)	1589600
bidirectional (Bidirectional)	(None, 200)	160800
dense (Dense)	(None, 2)	402

```

Total params: 1,750,802
Trainable params: 161,202
Non-trainable params: 1,589,600
None
```

Dan Hasil dari model 2 yang telah dibangun pada tahapan implementasi *bidirectional* LSTM ditampilkan pada gambar berikut

Model: "sequential_1"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 250, 100)	1589600
spatial_dropout1d (SpatialDr	(None, 250, 100)	0
bidirectional_1 (Bidirection	(None, 200)	160800
dense_1 (Dense)	(None, 2)	402
Total params: 1,750,802		
Trainable params: 1,750,802		
Non-trainable params: 0		
None		

Selanjutnya, pada tahapan implementasi *Bidirectional* LSTM, dipanggil fungsi *fit()* untuk melatih model serta melakukan iterasi terhadap 5 epoch, berikut tampilan yang diperoleh sebagai hasil dari *fit()* terhadap model 1

```
Epoch 1/5
191/191 [=====] - 158s 830ms/step - loss: 0.6748 - accuracy: 0.5850 - val_loss: 0.6661 - val_accuracy:
0.6052
Epoch 2/5
191/191 [=====] - 172s 899ms/step - loss: 0.6664 - accuracy: 0.5945 - val_loss: 0.6685 - val_accuracy:
0.5980
Epoch 3/5
191/191 [=====] - 177s 925ms/step - loss: 0.6628 - accuracy: 0.6025 - val_loss: 0.6592 - val_accuracy:
0.6151
Epoch 4/5
191/191 [=====] - 181s 948ms/step - loss: 0.6603 - accuracy: 0.6017 - val_loss: 0.6604 - val_accuracy:
0.6174
Epoch 5/5
191/191 [=====] - 184s 961ms/step - loss: 0.6569 - accuracy: 0.6085 - val_loss: 0.6580 - val_accuracy:
0.6157
```

Dan berikut tampilan yang diperoleh sebagai hasil dari *fit()* terhadap model 2 pada data *train* dan data *test*.

```
Epoch 1/5
193/193 [=====] - 425s 2s/step - loss: 0.6934 - accuracy: 0.5399 - val_loss: 0.6858 - val_accuracy: 0.
5544
Epoch 2/5
193/193 [=====] - 427s 2s/step - loss: 0.6631 - accuracy: 0.5952 - val_loss: 0.6679 - val_accuracy: 0.
5880
Epoch 3/5
193/193 [=====] - 435s 2s/step - loss: 0.5791 - accuracy: 0.6943 - val_loss: 0.6876 - val_accuracy: 0.
6041
Epoch 4/5
193/193 [=====] - 451s 2s/step - loss: 0.4124 - accuracy: 0.8043 - val_loss: 0.8592 - val_accuracy: 0.
5749
Epoch 5/5
193/193 [=====] - 443s 2s/step - loss: 0.3145 - accuracy: 0.8531 - val_loss: 1.0895 - val_accuracy: 0.
5873
```

```
Epoch 1/5
12/12 [=====] - 26s 2s/step - loss: 0.6909 - accuracy: 0.5910 - val_loss: 0.5785 - val_accuracy: 0.776
5
Epoch 2/5
12/12 [=====] - 26s 2s/step - loss: 0.6061 - accuracy: 0.6570 - val_loss: 0.5458 - val_accuracy: 0.776
5
Epoch 3/5
12/12 [=====] - 26s 2s/step - loss: 0.5706 - accuracy: 0.7058 - val_loss: 0.5337 - val_accuracy: 0.776
5
Epoch 4/5
12/12 [=====] - 26s 2s/step - loss: 0.5565 - accuracy: 0.7177 - val_loss: 0.5182 - val_accuracy: 0.800
0
Epoch 5/5
12/12 [=====] - 26s 2s/step - loss: 0.5320 - accuracy: 0.7375 - val_loss: 0.4988 - val_accuracy: 0.776
5
```

REFERENSI

- [1] C. Juditha, "Interaksi Komunikasi Hoax di Media Sosial serta Antisipasinya," *Pekommas*, vol. 3, pp. 31-44, 2018.
- [2] S. H. Kong, T. L. Mei, G. K. Hoon dan S. N. Hana, *Fake News Detection using Deep Learning*, pp. 1-6, 2019.
- [3] "ProgrammerSought," [Online]. Available: <https://www.programmersought.com/article/27162713146/>. [Diakses 10 11 2020].
- [4] D. Robert dan D. H. Gregory, "Deep learning: RNNs and LSTM," dalam *Handbook of Medical Image Computing and Computer Assisted Intervention*, Baltimore, MD, Johns Hopkins University, Department of Computer Science, 2020, pp. 503-505.
- [5] R. Paul, *Pouring Sequence Prediction using Recurrent Neural*, p. 3, 2018.
- [6] "Bidirectional Recurrent Neural Networks," i2tutorials, 2019 September 2019. [Online]. Available: <https://www.i2tutorials.com/what-is-the-difference-between-bidirectional-rnn-and-rnn/>. [Diakses 03 September 2020].
- [7] A. d. Zhang, "Dive into Deep Learning," September 2020. [Online]. Available: https://d2l.ai/chapter_recurrent-modern/bi-rnn.html. [Diakses 03 September 2020].
- [8] J. Brownlee, *Deep Learning for Natural Language Processing*, 2017.
- [9] R. Herbrich dan T. Graepel, *Handbook of Natural Language Processing* Second edition, Chapman & Hall/CRC (Taylor & Francis group), 2010.
- [10] C. Olah, "Understanding LSTM Networks," 27 Agustus 2015. [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Diakses 4 Oktober 2020].
- [11] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, "NeuralArchitecturesforNamedEntityRecognition," vol. II, pp. 261-262, 2016.
- [12] R. A. Isnain, A. Sihabuddin dan Y. Suyanto, "Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection," *Indonesian Journal of Computing and Cybernetics Systems*, vol. 14, p. 172, 2020.
- [13] "deepai.org," DeepAI, [Online]. Available: <https://deepai.org/machine-learning-glossary-and-terms/bidirectional-recurrent-neural-networks>. [Diakses 3 September 2020].

- [14] T. W. H. & S. P. Perkasa, "RANCANG BANGUN PENDETEKSI GERAK MENGGUNAKAN," *Journal of Control and Network Systems*, vol. 3, no. 2, p. 92, 2014.
- [15] A. C. S. & N. A. Kesarwani, "Fake News Detection on Social Media using," *IEEE Explore*, vol. 978, p. 1, 2020.
- [16] Emily, "TensorFlow," [wikipedia.org](https://en.wikipedia.org/wiki/TensorFlow), Sabtu Januari 2020. [Online]. Available: <https://en.wikipedia.org/wiki/TensorFlow>. [Diakses Senin Oktober 2020].
- [17] Abadi, Martin, (Et all);, TensorFlow: A System for Large-Scale, Savannah: USENIX Association, 2016.
- [18] E. & M. M. Provel, "Using Deep Learning to Detect Rumors," *Springer Nature Switzerland*, vol. 12194, p. 2, 2020.
- [19] A. Ayat, A.-S. Aisha dan A. Malak, *A Closer Look at Fake News Detection: A Deep Learning*, pp. 24-28, 2019.
- [20] e. a. Thota, "Fake News Detection: A Deep Learning Approach," *SMU Data Science*, vol. 1, p. 10, 2018.
- [21] S. d. Vijayaraghavan, "Fake News Detection with Different Models," p. 1, 2020.
- [22] B. Kanani, "Machine Learning Tutorials," 27 September 2019. [Online]. Available: <https://studymachinelearning.com/stemming-and-lemmatization/>. [Diakses 11 10 2020].
- [23] A. P. Yulio, "Devtrik," 18 Januari 2019. [Online]. Available: <https://devtrik.com/python/text-preprocessing-dengan-python-nltk/>. [Diakses 11 September 2020].
- [24] D. & J. H. M. Jurafsky, *Speech and Language Processing*, 2019.
- [25] B. Age, LEXALYTICS, 9 September 2019. [Online]. Available: <https://www.lexalytics.com/lexablog/text-analytics-functions-explained>. [Diakses 10 Oktober 2020].
- [26] S. Chakravarthy, "TowardsDataScience," Juni 2019. [Online]. Available: <https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4>. [Diakses 11 10 2020].
- [27] A. O. e. al, *Fake News Identification on Twitter with Hybrid CNN and RNN Models*, pp. 226-230, 2018.
- [28] P. Bahad, P. Saxena dan R. Kamal, "INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING (ICRTAC) 2019," *Fake News Detection Using Bi-directional LSTM- Recurrent Neural Network*

, pp. 74-82, 2019.

- [29] S. Prabhakaran, “ML+,” 2 10 2018. [Online]. Available: <https://www.machinelearningplus.com/nlp/lemmatization-examples-python/>. [Diakses 10 10 2020].
- [30] D. Sarkar, “Autor99,” 3 April 2018. [Online]. Available: <https://www.guru99.com/word-embedding-word2vec.html>. [Diakses 29 10 2020].
- [31] NSS, “KDNuggets,” 4 Juni 2017. [Online]. Available: <https://www.kdnuggets.com/2018/04/implementing-deep-learning-methods-feature-engineering-text-data-cbow.html>. [Diakses 29 10 2020].
- [32] J. M. K. & J. P. Han, Data Mining Concepts and Techniques, Morgan Kaufmann, 2011.
- [33] R. & J. A. M. George, “Emotion Classification Using Machine Learning and Data Preprocessing Approach on Tulu Speech Data,” *International Journal of Computer Science and Mobile Computing*, p. 10, 2016.
- [34] C. J. C. A. K. Santra, “Genetic Algorithm and Confusion Matrix for Document Clustering,” *International Journal of Computer Science Issues*, p. 3, 2012.
- [35] P. e. a. Bahad, “Fake News Detection using Bi-directional LSTM-Recurrent Neural,” *ScienceDirect*, pp. 74-82, 2019.
- [36] D. d. & M. M. Beer, “Approaches to Identify Fake News: A Systematic Literature Review,” *Springer Nature Switzerland*, vol. 136, p. 13–22, 2020.
- [37] M. M. A. Raffi dan S. A. Ardiayanti, “e-Proceeding of Engineering,” *Analisis Model Word2vec dalam Penyelesaian Soal Analogi pada Bahasa Indonesia*, vol. 6, no. , p. 8513, 2019.
- [38] Hackdeploy, “HackDeploy,” 2 December 2018. [Online]. Available: <https://www.hackdeploy.com/word2vec-explained-easily/>. [Diakses 7 November 2020].
- [39] O. & Y. G. Levy, “Neural Word Embedding as Implicit Matrix Factorization,” *Advances in neural information processing systems*, p. 2177–2185, 2014.
- [40] S. M. Rezaeinia, A. Ghodsi dan R. Rahmani, “Improving the Accuracy of Pre-trained Word Embeddings for,” p. 12, 2017.