PERBANDINGAN *WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT* MENGGUNAKAN *DEEP LEARNING* PADA ULASAN KONDISI PENGGUNA OBAT KESEHATAN

(Skripsi)

Oleh
FIQIH AULIA PRADANA

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023

ABSTRAK

Judul

PERBANDINGAN *WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT* MENGGUNAKAN *DEEP LEARNING* PADA ULASAN KONDISI PENGGUNA OBAT KESEHATAN

Oleh

Fiqih Aulia Pradana

Produk kesehatan seperti obat merupakan salah satu aspek penting dalam kesehatan. Sebagai masyarakat awam perlu rekomendasi obat dari penyintas penyakit. Rekomendasi tersebut berupa ulasan yang dapat dijadikan bahan pengetahuan untuk masyarakat dalam memilih obat sesuai kondisi yang dialami. Dalam statistika khususnya dalam metode klasifikasi, data berbentuk ulasan diproses melalui NLP. *Text mining* merupakan salah satu bagian dari NLP. Dalam penerapannya *text mining* harus berdampingan dengan *machine learning* untuk mengolah algoritma untuk mendapat hasil yang baik. *Word embedding* merupakan teknik analisis kebahasaan terbaru, dimana metode ini memberikan informasi struktur, urutan, semantik dan konteks di sekitar kata. *Word Embedding* yang dipakai adalah *Word2Vec, Glove,* dan *FastText*. Metode yang digunakan untuk mengklasifikasikan yaitu *deep learning* LSTM. LSTM bisa digunakan untuk menyimpan memori jangka panjang. Perpaduan *word embedding* dan LSTM dapat melihat hasil kinerja masing-masing *word embedding* dan keunggulan dari *word embedding* yang bisa dimaksimalkan. Hasil dari perbandingan word embedding *Word2Vec, Glove,* dan *FastText* digabungkan dengan metode LSTM mampu mendapatkan hasil nilai akurasi berturut-turut 85.20%, 84.19%, 86.22%. Sedangkan nilai *F1-Score word embedding Word2Vec, Glove,* dan *FastText* berturut-turut 85%, 84%, 86%.

Kata Kunci : **Ulasan, Kondisi, Klasifikasi, *Word2Vec, Glove, FastText,* LSTM**

ABSTRACT

Judul

COMPARISON OF WORD EMBEDDING WORD2VEC, GLOVE, AND FASTTEXT USING DEEP LEARNING IN REVIEWS ON CONDITIONS OF HEALTH MEDICINE USERS

By

Fiqih Aulia Pradana

Health products such as medicines are one of the important aspects of health. As ordinary people, we need drug recommendations from disease survivors. These recommendations are in the form of reviews that can be used as material for knowledge for the public in choosing drugs according to the conditions experienced. In statistics, especially in classification methods, data in the form of reviews are processed through NLP. Text mining is a part of NLP. In its application, text mining must coexist with machine learning to process algorithms to get good results. Word embedding is the latest linguistic analysis technique, where this method provides structure, sequence, semantic and context information around words. The Word Embedding used are Word2Vec, Glove, and FasText. The method used to classify is deep learning LSTM. LSTM can be used to store long-term memory. The combination of word embedding and LSTM can see the performance results of each word embedding and the advantages of word embedding that can be maximized. The results of the word embedding comparison Word2Vec, Glove, and FasText combined with the LSTM method were able to obtain accuracy values of 85.20%, 84.19%, 86.22%, respectively. While the F1-Score value of word embedding Word2Vec, Glove, and FasText respectively 85%, 84%, 86%.

Key Words : **Reviews, Conditions, Classification, Word2Vec, Glove, FastText, LSTM**

PERBANDINGAN *WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT* MENGGUNAKAN *DEEP LEARNING* PADA ULASAN KONDISI PENGGUNA OBAT KESEHATAN

Oleh
FIQIH AULIA PRADANA
1917031099

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023

Judul Skripsi	:	**PERBANDINGAN *WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT* MENGGUNAKAN *DEEP LEARNING* PADA ULASAN KONDISI PENGGUNA OBAT KESEHATAN**
Nama Mahasiswa	:	*Fiqih Aulia Pradana*
Nomor Pokok Mahasiswa	:	**1917031099**
Jurusan	:	**Matematika**
Fakultas	:	**Matematika dan Ilmu Pengetahuan Alam**

Bandar Lampung, 8 Mei 2023

MENYETUJUI

1. Komisi Pembimbing

Dian Kurniasari, S.Si.,M.Sc.	**Prof. Dra. Wamiliana, MA., Ph.D.**
NIP. 196903051996032001	NIP. 19631108 198902 2 001

2. Ketua Jurusan Matematika

Dr. Aang Nuryaman, S.Si., M.Si.
NIP. 197403162005011001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dian Kurniasari, S.Si.,M.Sc.** .....

Sekretaris : **Prof. Dra. Wamiliana, MA., Ph.D.** .....

Penguji
Bukan Pembimbing : **Ir. Warsono, M.S., Ph.D.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: **8 Mei 2023**

PERNYATAAN SKRIPSI MAHASISWA

Saya yang bertanda tangan di bawah ini:

Nama	: **Fiqih Aulia Pradana**
Nomor Pokok Mahasiswa	: **1917031099**
Jurusan	: **Matematika**
Judul Skripsi	: **PERBANDINGAN *WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT* MENGGUNAKAN *DEEP LEARNING* PADA ULASAN KONDISI PENGGUNA OBAT KESEHATAN**

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri dan semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah karya penulisan ilmiah Universitas Lampung.

Bandar Lampung 8 Mei 2023

Penulis

Fiqih Aulia Pradana

NPM. 1917031099

RIWAYAT HIDUP

Penulis bernama lengkap Fiqih Aulia Pradana, anak pertama dari tiga bersaudara yang lahir di Pringsewu pada tanggal 9 Juni 2001 dari pasangan Bapak Suyadi dan Ibu Wariyati. Penulis menyelesaikan pendidikan sekolah dasar di SD Negeri 4 Wates pada tahun 2007 s.d 2010 lalu pindah di SD Negeri 1 Wates pada tahun 2010 s.d 2013, sekolah menengah pertama di SMP Negeri 3 Pringsewu pada tahun 2013 s.d 2016, dan sekolah menengah atas di SMA Negeri 1 Pringsewu pada tahun 2016 s.d 2019.

Pada tahun 2019 penulis diterima sebagai mahasiswa S1 di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SBMPTN. Selama aktif menjadi mahasiswa, penulis ikut serta dalam Himpunan Mahasiswa Jurusan Matematika (HIMATIKA) sebagai anggota Bidang Keilmuan pada tahun 2020.

Pada tahun 2022, sebagai bentuk penerapan bidang ilmu di dunia kerja, penulis melaksanakan Kerja Praktek (KP) di PT. Taspen Persero Kantor Cabang Kota Bandar Lampung dan sebagai bentuk pengabdian kepada masyarakat penulis melaksanakan Kuliah Kerja Nyata (KKN) di Kelurahan Sumber Mulyo, Kecamatan Sumber Rejo, Kabupaten Tanggamus.

PERSEMBAHAN

Alahmdulillahirabbil'alamin.
Puji dan syukur kepada Allah SWT atas nikmat serta hidayahnya sehingga skripsi ini dapat terselesaikan dengan baik dan tepat pada waktunya.

Oleh karena itu, dengan rasa syukur dan bahagia saya persembahkan rasa terima-kasih saya kepada

Diri Sendiri

Terima kasih untuk tetap selalu berusaha di segala keadaan untuk mengembangkan diri dan menjadi pribadi yang lebih baik.

Orang Tersayang

Tidak ada kata yang dapat aku sampaikan untuk kalian kecuali ucapan terimakasih atas semua yang telah kalian berikan untukku. Cinta, kasih sayang, motivasi, waktu, pengorbanan yang belum bisa aku balas, serta doa dan sujud yang selalu menantikan keberhasilanku dengan sabar dan penuh pengertian. Terima- kasih karena selalu mendoakan dan mendukung setiap langkah yang aku pilih. Karena atas doa dan ridho kalian, Allah memudahkan setiap perjalanan hidup ini.

Dosen Pembimbing dan Pembahas

Terimakasih kepada dosen pembimbing dan pembahas yang sudah sangat mem-bantu, memberikan motivasi, memberikan arahan serta ilmu yang berharga.

Almamater Tercinta Universitas Lampung

KATA INSPIRASI

".....dan jangan kamu berputus asa dari rahmat Allah. Sesungguhnya tiada berputus asa dari rahmat Allah, melainkan kaum yang kafir."

(*Q.S. Yusuf: 87*)

"Janganlah kamu bersikap lemah dan janganlah pula kamu bersedih hati, padahal kamulah orang orang yang paling tinggi derajatnya jika kamu beriman."

(Q.S. Ali Imran : 189)

"Sistem pendidikan yang bijaksana setidaknya akan mengajarkan kita betapa sedikitnya yang belum diketahui oleh manusia, seberapa banyak yang masih harus ia pelajari."

(Sir John Lubbock)

"Musuh utama manusia manusia untuk sukses bukan dari orang lain, melainkan dari kemalasan dan rasa ingin menunda pekerjaan untuk kesuksesan"

(Penulis)

SANWACANA

Puji dan syukur penulis ucapkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Perbandingan *Word Embedding Word*2vec, *GloVe*, dan *FastText* Dalam Menggunakan *deep learning* Pada Ulasan Kondisi Pengguna Obat Kesehatan".

Penulis menyadari bahwa dalam penulisan skripsi ini tidak terlepas dari bimbingan, dukungan, bantuan dan doa dari berbagai pihak. Oleh karena itu, dengan ketulusan hati penulis ingin menyampaikan terimakasih kepada:

1. Ibu Dian Kurniasari, S.Si., M.Sc. selaku Dosen Pembimbing I yang senantiasa selalu membimbing dan memberikan arahan, kritik, dan saran serta dukungan kepada penulis selama proses perkuliahan dan pembuatan skripsi ini.
2. Ibu Prof. Dra. Wamiliana, MA., Ph.D., selaku Dosen Pembimbing II yang telah memberikan bimbingan serta saran yang membantu kepada penulis dalam proses penyelesaian skripsi ini..
3. Bapak Ir. Warsono, M.S., Ph.D., selaku Dosen Pembahas atas ketersediaannya untuk membahas serta memberikan kritik dan saran serta evaluasi kepada penulis dalam penyelesaian skripsi ini.
4. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Dr. Eng. Heri Satria, S.Si., M.Si., selaku dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Seluruh dosen, staf, dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

7. Ayah, ibu, adik dan seluruh keluarga besar yang selalu memberikan kasih sayang, dukungan, nasihat, motivasi doa, serta segala kebutuhan kepada penulis untuk menyelesaikan skripsi ini.
8. Anindya yang telah menemani dan mendukung penulis untuk tetap konsisten untuk menyelesaikan dunia perkuliahan.
9. Azza, Alfira, Silvi, Aris, Ikhsan, Thoif yang telah mendoakan, mendukung, dan memberikan kenangan indah selama menjalani masa perkuliahan.
10. Teman-teman Matematika 2019, terima kasih atas kebersamaannya.
11. Almamater tercinta, Universitas Lampung.
12. Seluruh pihak yang telah membantu yang tidak dapat disebutkan satu persatu.

Bandar Lampung, 8 Mei 2023

Penulis

Fiqih Aulia Pradana
1917031099

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

I.PENDAHULUAN

1.1 Latar Belakang dan Masalah

Produk kesehatan seperti obat-obatan pada masa kini sangat mudah dijual bebas terutama obat standar yang tidak berbahaya, yang banyak dijual apotik dan toko online di beberapa *e-commerce*. Karena kemudahan itu, sebagai orang awam perlu rekomendasi obat dari penyakit yang diderita, rekomendasi tersebut salah satunya ulasan orang yang sudah menggunakan obat itu untuk dirinya sendiri (Ariyulinda, 2018). Dengan adanya ulasan orang lain dapat tahu penyakit apa yang diderita, yang selanjutnya bisa ditentukan obat yang cocok untuk kondisi yang terjadi. Oleh karena itu penting bagi pihak tenaga kesehatan, farmasi, dan pihak terkait untuk menganalisis produk obat yang mereka gunakan, agar pemberian obat tepat sasaran sesuai kondisi yang diderita pasien. Responden memberikan ulasan dalam dua bentuk, ulasan dan kondisi responden. Kondisi menjadi label dapat dengan mudah diklasifikasikan menjadi beberapa kelas berbeda , mulai dari dua kelas atau lebih (Cheon and Kyaing, 2015).

Klasifikasi adalah proses pengkategorian terhadap kumpulan data. Klasifikasi sangat penting dalam hal kemudahan untuk membedakan data satu dengan yang lain. Dalam kasus ini klasifikasi digunakan untuk membedakan kondisi pasien berdasakan ulasan dari pengguna terhadap penggunaan obat maupun sakit yang dialami (Indriani, 2014).

Ulasan adalah wujud ekspresi pengguna atau pihak terkait mengenai produk yang dirasakan, tetapi setiap orang memiliki persepsi dan skala penilaian yang berbeda-beda. Ulasan tidak bisa langsung diklasifikasikan menjadi pengetahuan untuk menentukan kondisi penyakit responden yang diinginkan manusia. Karena ulasan memiliki persepsi yang berbeda maka dilakukan ekstrasi. Ekstrasi sentimen dapat dilakukan dengan pemrosesan data teks / *Natural Language Processing* (NLP) (Cheon and Kyaing, 2015).

Pemrosesan teks / *Natural Language Processing* (NLP) merupakan bagian dari *Artifisial intelegent* (AI) yaitu salah satu metode pemrosesan data yang secara teoritis didasari dari teknik komputasi untuk menganalisis dan menilai suatu teks pada tingkat analisis linguistik yang tinggi untuk tujuan mencapai pemikiran manusia/*Artifisial intelegent* (Liddy, 2001). *Feature engineering* data sangat penting dalam pemrosesan data tekstual, hal ini akan berdampak signifikan terhadap nilai akurasi atau model yang dibangun (Nurdin, 2020). *Feature engineering* adalah langkah preprocessing data *machine learning* yang bertujuan meningkatkan peforma model (Nagersian, *et al*, 2017). *Feature engineering* pada pemrosesan teks dalam *machine learning* memiliki hambatan tersendiri karena karakter teks yang berbentuk data tidak terstruktur (*unstructure data*).

Feature engineering dari masa ke masa sangat berkembang dimulai dari *one hot encoding* yaitu teknik untuk menyeragamkan data kategorik atau numerik. Cara kerja *one hot encoding* adalah dengan membuat suatu array 1 dimensi dengan panjang sebanyak jenis fitur yang ada dan mempunyai isi biner 0 dan 1. *One hot encoding* merepresentasikan data bertipe kategorik menjadi lebih ekspresif (Silviana, dkk, 2022). Setelah itu berkembang model *Bag of Words* (BOW), yang terdiri dari *n-grams, term frequencies* (TF), hingga *term frequency-inverse document frequency* (TF-IDF). *Bag of Words* merepresantasikan kata didasari oleh frekuensi kemunculan dalam sebuah kalimat pada data tekstual (Nurdin, 2020). Kelemahan BOW ini tidak bisa memberikan informasi struktur, urutan, semantik dan konteks di sekitar kata dalam kalimat pada data tekstual (Nurdin, 2020).

Kemudian perkembangan lebih lanjut dari BOW yaitu *word embedding* yang dikembangkan dalam penelitian oleh Mikolov dkk (Mikolov, *et al*, 2013).

Word embedding memetakan setiap kata dalam dokumen ke dalam *dense vector*, di mana sebuah vektor merepresentasikan proyeksi kata di dalam ruang vektor. Posisi kata tersebut dipelajari dari teks atau berdasarkan kata-kata di dekatnya. *Word embedding* ini dapat menangkap makna semantik dan sintaktik kata. Pada tahun 2013 Mikolov memperkenalkan *Word2Vec* dengan dua model nya yaitu *Skip-gram* dan *Continuous Bag of Words* (CBOW)(Mikolov, *et al*, 2013). Pada tahun 2013 diperkenalkan *word embedding* baru oleh Penington dkk dari Stanford University, yaitu *GloVe* yang memakai rasio *co-occurrence probability* antarkata (Pennington, *et al*, 2014). Kemudian pada tahun 2017, Facebook memperkenalkan *FastText* model *word vector* mirip dengan *Word2Vec* namun *FastText* menggunakan informasi sub-kata(Bojanowski, *et al*, 2017). *Word embedding Word2Vec, GloVe, dan FastText* dapat diimplementasikan dalam *machine learning* dengan konsep NLP metode *Deep Learning* yaitu RNN dengan pengembangannya yaitu *Long Short-TermMemory* (LSTM).

Salah satu metode yang sedang berkembang di *Deep Learning* yaitu perkembangan dari *Recurrent Neural Network* (RNN) yaitu *Long Short-TermMemory* (LSTM) telah terbukti cukup efektif dalam mengatasi pemrosesan teks yang melibatkan urutan teks (Rao and Spasojevic, 2016). Karena pada dasarnya LSTM sangat baik untuk menyimpab memori jangka panjang. Di penelitian ini menerapkan penyematan kata (*word embedding*) dan metode *Deep Learning* yaitu LSTM untuk masalah klasifikasi teks, dengan *word embedding Word2Vec, GloVe, dan FastText*.

Adapun penelitian sebelumnya yang pernah dilakukan terkait klasifikasi data teks yaitu penelitian yang dilakukan yaitu membandingkan kinerja dari *word embedding* seperti *Word2Vec, GloVe* dan *FastText* dan diklasifikasikan dengan algoritma *Convolutional Neural Network* dan menghasilkan nilai F-measure

berturut-turut 0.925, 0.958, dan 0.979 (Nurdin, dkk, 2020). Penelitian kedua dengan membandingkan akurasi model analisis sentimen dari review hotel menggunakan *Word2Vec* dan *FastText* digabungkan dengan *ensemble learning: Random Forest, Extra Tree,* dan *AdaBoost,* menghasilkan nilai akurasi yang terbaik yaitu *FastText* digabungkan dengan *Random Forest* dan *Extra Tree* yaitu masing-masing 93% (Khomsah, dkk, 2021).

Penelitian ketiga yaitu sentimen analisis menggunakan *supervised machine learning algorithms yang* menggunakan *dataset Drugs Review* label digabungkan menjadi 3 rating, yaitu 1-4 negatif, 5-6 netral, 7-10 positif. Penelitian ini memakai *Count Vectorize* dan TFI-DF sebagai *features engineering* untuk kebahasaan, dengan menggunakan beberapa algoritma klasifikasi *supervised learning* seperti *Random Forest, Artifisial Neural Network, Long Short-Term Memory, Support Vector Machine, Gated Recurrent Units,* dan *Logistic Regression,* menghasilkan nilai rata-rata akurasi tertinggi menggunakan *count vectorize dan* TFIDF pada variabel kondisi *Birth Control* (Vijayaraghavan, 2020). Penelitian keempat yaitu membandingkan berbagai arsitektur *Deep Learning* untuk analisis sentimen pada data *drugs reivew*. Penelitian ini mirip dengan penelitian ketiga dengan mereduksi label menjadi 3. Dengan menggunakan *word embedding Word*2Vec menggunakan metode seperti CNN, LSTM, hybrid LSTM-CNN, CNN-LSTM, BERT-LSTM, menghasilkan nilai akurasi tertinggi pada model BERT-LSTM dengan nilai *micro F1-Score* 90,46% (Ruiz, 2020).

Penelitian kelima yaitu klasifikasi kejadian *multiclass* pada data teks, dengan menggunakan TFI-DF serta algoritma yang dipakai yaitu CNN, *Recurent Neural Network* (RNN) *Deep Neural Network* (DNN), dan algoritma yang memiliki hasil tingkat akurasi tertinggi yaitu DNN dengan 84% (Ali, *et al* 2021). Penelitian keenam yaitu mengenai sentimen analisis review obat menggunakan berbagai algoritma boosting dengan algoritma boosting seperti *Light Gradien Boosting Machine* (LGBM), XGBoost, dan CatBoost. Label di reduksi menjadi 2 yaitu positif dan negatif, menghasilkan nilai akurasi 88,8% LGBM, 76,68% XGBoost, 88,2% CatBoost (Mishra, 2021).

Berdasarkan beberapa penelitian diatas banyak literatur tentang analisis teks menggunakan *word embedding* dan algoritma *neural network* atau pengembangannya. Dari berbagai penelitian diatas sangat populernya penggunaan *word embedding* digunakan dalam bidang NLP terutama penerapan pada algortima pengembangan *neural network* seperti ANN, CNN, LSTM dll. Hal itu menjadi motivasi untuk melakukan penelitian mengenai perbandingan kinerja dari setiap model *word embedding* yang ada dan digabungkan dengan metode *deep learning* yang juga sedang berkembang yaitu LSTM. Maka penelitian ini akan membahas "Perbandingan *Word Embedding Word2*vec, *GloVe*, dan *FastText* Dalam Menggunakan *deep learning* Pada Ulasan Kondisi Pengguna Obat Kesehatan".

1.2 Tujuan Penelitian

Berdasarkan latar belakang tujuan penelitian ini adalah sebagai berikut:

1. Mengkaji kinerja *deep learning* klasifikasi text dengan *word embedding Word2Vec*.
2. Mengkaji kinerja *deep learning* klasifikasi text dengan *word embedding Glove.*
3. Mengkaji kinerja *deep learning* klasifikasi text dengan *word embedding FasText.*
4. Membandingkan kinerja *deep learning* klasifikasi text dengan *word embedding Word2Vec, GloVe, dan FastText.*

1.3 Manfaat Penelitian

Hasil penelitian ini dapat menjadi landasan dalam pengembangan klasifikasi teks menggunakan LSTM dengan fitur *word embedding* secara lebih lanjut. Selain itu manfaat dari penelitian ini yaitu sebagai berikut:

1. Hasil dari penelitian ini dapat berguna oleh konsumen, tenaga kesehatan, maupun pihak farmasi untuk memutuskan obat mana yang harus dipakai.
2. Hasil dari penelitian ini dapat berguna untuk meminimalisir kesalahpahaman pasien dan perbedaan pendapat tentang obat yang dipakai berdasarkan kondisi pasien.
3. Pengembangan LSTM dengan *embedding Word2vec*, *GloVe, dan FastText* dapat menghasilkan kinerja *confusion matrix*, kurva *accuracy* dan *loss* dari proses *training* dan *testing* yang lebih optimal.
4. Hasil penelitian ini dapat menjadi referensi untuk meningkatkan nilai performa akurasi, presisi, recall, *F1-Score* dalam klasifikasi teks yang menerapkan LSTM dengan *word embedding Word2vec*, *GloVe, dan FastText*.

II.TINJAUAN PUSTAKA

2.1 *Data mining*

Data mining atau penambangan data adalah proses menggali jauh kedalam data yang besar (*big data*) untuk menemukan hubungan, fitur, dan pola tersirat yang sebelumnya tidak diketahui dan tidak dimengerti yang dapat berguna untuk masa depan. Banyak orang sering menganggap *machine learning* identik dengan *data mining*, tapi secara teknis keduanya berbeda. *Data mining* dimasa kini sangat perlu dilakukan terutama belakangan ini karena banyaknya data dalam jumlah sangat besar dan semakin besarnya kebutuhan untuk ekstraksi data menjadi informasi dan *knowlegde* yang berguna yaitu dengan cara melakukan kegiatan mengekstrasi atau menambang pengetahuan dari data yang berukuran/berjumlah besar, informasi inilah yang nantinya sangat berguna untuk pengembangan suatu kegiatan manusia (Thoomkuzhhy, 2020). Beberapa metode *data mining* yang sering digunakan adalah

a. *Classification*

   Classification merupakan metode *data mining* dengan memberikan kelompok terhadap setiap keadaan. Setiap keadaan berisi sekelompok atribut, yaitu atribut label (*supervised learning*).

b. *Clustering*

   Clustering merupakan metode *data mining* dengan mengelompokan data menjadi beberapa kelompok sesuai kemiripan dari masing-masing data sehingga antar kelompok memiliki kemiripan jauh dan satu kelompok memiliki kemiripan dekat. Metode ini tidak memerlukan atribut untuk mengarahkan pola (*unsupervised learning*)

c. *Association*

Metode *Association* juga disebut sebagai *Market Basket Analysis*. Seperti namanya, metode ini mengasosiasikan suatu data tertentu hingga mendapatkan informasi, contohnya adalah menganalisa transaksi jual beli produk-produk yang seringkali dibeli bersamaan oleh pembeli, misalnya apabila orang membeli *smartphone*, biasanya juga dia membeli *case* nya.

d. *Regression*

Metode *regression* mirip dengan *classification*, namun perbedaan mendasar yaitu *regression* tidak mencari pola tertentu yang dijabarkkan oleh kelas tertentu, namun mencari pola dan menentukan sebuah nilai numerik.

e. *Forecasting*

Metode *Forecasting* atau Peramalan yaitu menhubungkan nilai masa depan dengan teknik-teknik *machine learning* dan stastistik yang berhubungan dengan waktu, musim, *trend* pada data.

2.2 *Machine Learning*

Ditengah berkembangnya kecerdasan buatan/*Artifisial Intelegent* (AI), tak lepas dari *machine learning* (Sihombing dan Arsani, 2021). Bidang *machine learning* sangat erat kaitannya dengan bagaimana membangun algoritma dan program komputer secara otomatis meningkat dengan *training data* (Mitchell, 1997). *Machine learning* dikembangkan berdasarkan beberapa disiplin ilmu utama seperti ilmu statistika, matematika dan penambangan data. Beberapa metode *machine learning* yang digunakan untuk mengklasifikasikan dalam kasus penelitian diantaranya, *Naive Bayes* (NB), *Random Forest* (RF), *K-Nearest Neighbor* (KNN), *Artificial Neural Network* (ANN), *Support Vector Machine* (SVM), *Decision Tree* (DT), dan lainnya (Sihombing dan Arsani, 2021).

Machine learning merupakan fokus dari pengembangan sebuah sistem yang mampu belajar secara mandiri untuk membuat keputusan tanpa perlu pemrograman manusia berulang kali. Metode ini menciptakan sebuah mesin yang dapat menemukan aturan perilaku pengambilan keputusan yang optimal dan beradaptasi ketika terjadi perubahan. Mesin melakukan analisis pada kumpulan data besar untuk menemukan pola spesifik dalam data.

Machine learning mempunyai 3 jenis algoritma pembelajaran

a. *Supervised Learning*

Supervised Learning adalah algoritma *machine learning* yang menggunakan kumpulan data yang diketahui (*dataset* pelatihan/*data train*) untuk membangun *classifier* atau *regressor* yang dapat memperkirakan nilai *output* untuk *input* yang sebelumnya tidak terlihat. *Output* dari *supervised learning* telah ditentukan sebelumnya dan proses pembelajaran berakhir ketika algoritma mencapai hasil yang diinginkan. Teknik yang digunakan dalam *supervised learning* adalah metode klasifikasi yang sepenuhnya melabeli *dataset* untuk mengklasifikasikan kelas yang tidak diketahui.

b. *Unsupervised learning*

Unsupervised learning tidak memberikan *output* nilai spesifik. Sebaliknya, ia mencoba untuk mendapatkan struktur yang mendasarinya dari *input* tanpa label atau *output* dari *unsupervised learning* belum ditentukan sebelumnya. Misalnya, dalam *cluster unsupervised learning* tujuannya adalah untuk memperoleh pemetaan dan pola dari *input* yang diberikan (seperti vektor bilangan real) ke grup sehingga *input* yang serupa dipetakan ke grup yang sama. Teknik ini cocok untuk *cluster* dan *association rule* (Engelen and Hoss, 2019).

c. *Semi-supervised learning*

Semi-supervised learning yaitu bagaimana menggabungkan *supervised learning* dan *unsupervised learning* dengan mengelompokkan kumpulan data dengan atau tanpa label kelas ke dalam beberapa *cluster* (Jain, *et al*, 2018).

2.3 Natural Language Processing (NLP)

Natural Language Processing (NLP)/ Pemrosesan teks merupakan bagian dari *Artifisial intelegent* (AI) yang merupakan salah satu teknik komputasi untuk menganalisis dan menilai suatu teks atau memahami bahasa natural dari manusia pada tingkat analisis linguistik yang tinggi untuk mencapai *Artifisial intelegent* (AI) (Liddy, 2001). NLP dengan tujuan mencapai *Artifisial intelegent* dikatakan tercapai jika alat bantu dari *machine learning* yaitu komputer bisa memahami *natural language* yang diberikan dari *natural language* itu sendiri, baik dari segi kata yang dipakai, arti kata, kegunaan kata dari sebuah kalimat dan pengetahuan mengenai bagaimana kata-kata tersebut bergabung dan akhirnya membentuk kalimat dan memiliki suatu makna. *Natural language* disini yaitu berupa pesan yang ingin disampaikan dan dikomunikasikan baik secara lisan maupun tulisan (Migunani dan Aditama, 2020). Penelitian yang berkaitan dengan NLP memiliki stuktur data *supervised learning*.

Alasan mengapa NLP akan sangat penting di masa depan adalah karena NLP membantu mengembangkan model dan proses yang mengambil informasi sebagai *input* dalam bentuk ucapan dan/atau teks dan memanipulasinya sesuai dengan algoritma komputer. Oleh karena itu, *input* dapat berupa ucapan, teks, atau gambar, dan *output* dari sistem NLP dapat berupa ucapan atau teks.

2.4 *Text Mining*

Text mining adalah proses ekstraksi informasi dalam jumlah besar untuk dijadikan *knowledge*. Text mining menganalisis data teks yang tidak terstruktur dari data teks tersebut setelah diolah menjadi pemahaman baru yang sebelumnya tidak diketahui (Thoomkuzhhy, 2020). Text mining berbeda dalam menganalisis yang memerlukan hal-hal khusus dibandingkan menganalisis data numerik biasa, yang perlu

dilakukan adalah *prepocessing text*. *Prepocessing text* sendiri sangat berbeda daripada *prepocessing* data numerik.

Preprocessing text merupakan proses pembersihan data teks untuk menangani sejumlah besar teks dengan kata-kata yang tidak terstruktur seperti singkatan, *emoticon*, simbol, dan angka. Hal Ini membutuhkan teknik *preprocessing*. *Preprocessing* membantu mengekstrak informasi dari evaluasi dan mengubah kata-kata tidak terstruktur ini menjadi bentuk vektor. *Preprocessing* sangat penting dan penting dalam *data mining* karena dapat mempengaruhi nilai akurasi dari model (Hermanto, dkk, 2021).

Tahap prepocessing pada data teks yaitu,

1. *Case Folding*

Case Folding merupakan proses mengubah semua huruf kapital menjadi huruf kecil (*lower case*). Hal ini perlu dilakukan untuk menjadikan teks seragam.

2. *Filtering/Cleansing*

Filtering/Cleansing merupakan proses menghilangkan karakter khusus yang tidak diinginkan seperti tanda baca, angka dan simbol lain yang tidak diperlukan dalam proses *training*.

3. *StopWords Removal*

StopWords Removal merupakan proses menghilangkan kosakata dan kata-kata gaul/slang dan tidak memiliki makna yang tidak ada dalam kamus bahasa yang digunakan seperti kamus bahasa inggris atau indonesia (sastrawi).

4. *Lemmatization*

StopWords Removal merupakan proses penghilangan afiks/imbuhan dari sebuah kata menjadi kata dasar dengan mengetahui konteks dari kata tersebut.

5. *Tokenization*

StopWords Removal merupakan proses penguraian kalimat menjadi kata demi kata dengan cara mengurai kalimat per spasi.

6. *Word Tokenization*

Word Tokenization merupakan proses pengubahan kata dalam kalimat menjadi nilai indeks dalam bentuk numerik. Proses ini mengambil nilai indeks dari kosa kata yang dibuat, menetapkan nomor indeks berdasarkan kata yang sering muncul dalam *dataset.*

7. *Padding data*

Padding adalah proses yang terjadi karena setiap pesan teks dalam *dataset* memiliki jumlah kata yang berbeda, sehingga vektor harus memiliki panjang yang sama. Oleh karena itu, *padding* data atau entri data dilakukan jika panjang pesan teks kurang dari panjang maksimal. Vektor diisi dengan 0 angka dan *string* kata kosong sehingga vektor memiliki panjang yang sama (Pratama, 2022).

2.5 *Balancing Data*

Imbalance data merupakan kondisi dimana suatu kelompok target/label yang memiliki jumlah yang berjarak cukup jauh dibanding target/label lainnya. Label data dengan jumlah lebih banyak disebut *majority* dan yang lebih sedikit disebut *minority*. Karena hal ini perlu proses *balancing* data sangat berpengaruh kepada akurasi model yang dihitung karena jika data masih *imbalance*, dalam proses nya data *majority* akan diutamakan dibandingkan *minority* dari pengolahan algoritma yang dipakai (Akbar dan Hayaty, 2020).

Balancing data adalah proses menyeimbangkan jumlah dari label data *majority* atau *minority*. Salah satu algoritma yang paling terkenal adalah *Synthetic Minority Oversampling Technique* (SMOTE). SMOTE merupakan salah satu metode

balancing data dengan meningkatkan data label *minority* agar setara dengan label *mayority*. SMOTE berbeda dengan metode *oversampling* lain yang hanya memperbanyak pengamatan secara acak, namun SMOTE membangkitkan data buatan berdasarkan k-tetangga terdekat (*k-nearest neighbor*) yang mempertimbangkan kemudahan dalam memprosesnya (Akbar dan Hayaty, 2020).

2.6 Representasi Kata (*Word Embedding*)

Setelah teks diproses dalam tahap prepocessing dan berbentuk vektor akan dilakukan proses penyematan kata / *word embedding. Word Embeding* adalah kumpulan nama dari pemodelan bahasa dan teknik ekstraksi fitur dalam *Natural Language Processing* (NLP), di mana setiap kata atau frasa kosakata dipetakan ke vektor dalam bentuk bilangan real. Penyematan kata banyak digunakan dalam *neural networks*, reduksi dimensi pada matriks kemunculan kata, model *probabilistik*, dll.

Metode *word embedding* ini juga digunakan sebagai masukan untuk meningkatkan kinerja *Natural Language Processing* (NLP) seperti *parsing* sintaktik (penguraian kata berdasarkan kamus) dan analisis sentimen.
One hot encoding yaitu metode standarisasi data kategorikal atau numerik. Cara kerja *one hot encoding* adalah dengan membuat sebuah array 1 dimensi dengan panjang sebanyak jenis fitur yang mempunyai isi biner. *One hot encoding* dapat menjelaskan data bertipe kategorikal secara lebih ekspresif. Kemudian dikembangkan model *Bag of Words* (BOW), yang terdiri dari *n-grams, term frequencies* (TF), dan *term frequency-inverse document frequency* (TF-IDF). *Bag of Words* dapat menjelaskan kata berdasarkan frekuensi kemunculan di sebuah kalimat pada data tekstual, Namun kelemahan keduanya adalah tidak dapat menyediakan struktur, urutan, informasi semantik dan konteks dari kata-kata dalam kalimat pada data tekstual, Maka dari itu untuk mengatasi masalah ini digunakan *Word Embedding* yang merupakan perkembangan dari BOW. Jenis *word*

embedding yang digunakan dalam menggantikan *one hot encoding* dan BOW adalah *Word2Vec, GloVe*, dan *FastText*.

2.6.1 *Word2Vec*

Word2Vec adalah algoritma *word embedding* yang memetakan setiap kata kedalam bentuk vektor. Algoritma *Word2Vec* dikembangkan oleh peneliti Google yaitu Mikolov *et al* pada tahun 2013. Model *word embedding* ini telah banyak digunakan oleh peneliti khususnya di bidang NLP (Nurdin, 2020).

Word2Vec mewakili kata-kata dalam vektor yang dapat membawa makna semantik kata-kata. Model *word embedding* ini merupakan aplikasi *unsupervised learning* yang menggunakan jaringan saraf (*neural network*) yang terdiri dari *hidden layer* dan *fully connected layer*. Dimensi matriks bobot untuk setiap lapisan adalah jumlah kata dalam korpus (kumpulan teks) dikalikan dengan jumlah sel/neuron tersembunyi pada lapisan tersembunyi. Matriks bobot pada *hidden layer* dari model yang telah dilatih digunakan untuk mentransformasikan kata kedalam vektor. Matriks bobot ini seprti *lookup table,* dimana setiap baris mewakili setiap kata dan kolom mewakili vektor dari kata tersebut (Nurdin, 2020).

Wod2vec mengandalkan informasi internal data dari bahasa.semantik yang akan dipelajari dari kata tertentu dan dipengaruhi oleh kata-kata sekitarnya. Model ini menjelaskan kemampuan untuk mempelajari pola linguistik hubungan linear antar-vektor kata (Nurdin, 2020).

Terdapat dua algoritma *Word2Vec* yang dikembangkan Mikolov *et al* yaitu *Continuous Bagof-Word* (CBOW) dan *Skip-gram* (Sutskever, *et al*, 2013).

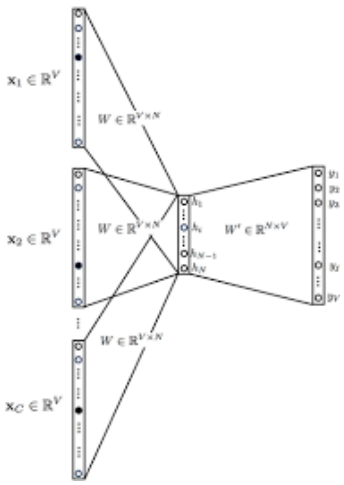
a. *Continuous Bagof-Word* (CBOW)

Model ini menggunakan konteks untuk memprediksi target kata. CBOW memiliki waktu *training* lebih cepat dan memiliki akurasi yang sedikit lebih baik untuk *frequent words* atau kata-kata umum. Model ini juga mempunyai tujuan untuk memprediksi kata (*output*) ketika diberikan konteks disekitar kata tersebut (*input*).

Sebagai contoh, data *training* yang dimiliki adalah kalimat yang sudah di *prepocessing* "**Ayah pulang kerja naik mobil malam hari**" dengan *window* size = 2

Ayah pulang kerja naik mobil malam hari
Ayah pulang kerja naik mobil malam hari
Ayah pulang kerja naik mobil malam hari
Ayah **pulang kerja naik mobil malam** hari
Ayah pulang **kerja naik mobil malam hari**
Ayah pulang kerja **naik mobil malam hari**
Ayah pulang kerja naik **mobil malam hari**

Dengan warna merah adalah target dan hitam *bold* adalah *input*



Gambar 1. Arsitektur CBOW

(sumber: Rong, 2016)

b. *Skip-Gram*

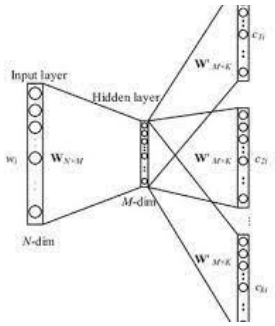
Model ini menggunakan sebuah kata untuk memprediksi sasaran konteks. *Skip-gram* bekerja dengan baik menggunakan data pelatihan yang jumlahnya sedikit serta bisa merepresentasikan istilah-kata yang disebut langka.

Arsitektur *Word2Vec Skip-gram* adalah kebalikan dari *Word2Vec* CBOW. Tujuan dari arsitektur *skip-gram* adalah untuk memprediksi konteks (*output*) disekitar kata tersebut (*input*).

Sebagai contoh, data *training* yang dimiliki adalah kalimat yang sudah di *prepocessing* "**Ayah pulang kerja naik mobil malam hari**" dengan *window* size = 2

Ayah pulang kerja naik mobil malam hari
Ayah pulang kerja naik mobil malam hari
Ayah pulang kerja naik mobil malam hari
Ayah **pulang kerja naik mobil malam** hari
Ayah pulang **kerja naik mobil malam hari**
Ayah pulang kerja **naik mobil malam hari**
Ayah pulang kerja naik **mobil malam hari**

Dengan warna merah adalah *input* dan hitam *bold* adalah target



Gambar 2. Arsitektur Skip-gram
(sumber: Rong, 2016)

Cara kerja *Word2Vec*

Contoh kalimat :

“Manusia melihat seekor Ular, Manusia mengejar ular, Ular berlari keluar Selokan”

Misalkan jaringan akan mempelajari hubungan antara kata “keluar” sebagai konteks dengan kata “melihat” sebagai target.

Vektor konteks = 0 1 0 0 0 0 0 0

Vektor Target = 0 0 0 1 0 0 0 0

Pertama, buat indeks kata dari contoh kalimat

Tabel 1. Contoh Indeks Korpus Per-kata *Word2Vec*

Kosakata	**Indeks**
Berlari	1
Keluar	2
Manusia	3
Melihat	4
Mengejar	5
Seekor	6
Selokan	7
Ular	8

Kedua, Inisiasi bobot kata dengan 3 *neuron hidden layer* (3 *hidden layer* sebagai contoh)

Tabel 2. Contoh Bobot *Hidden Layer* W_1

W1 =	0,2	0,3	0,4
	0,1	0,5	0,8
	0,3	0,2	0,6
	0,5	0,3	0,4
	0,8	0,6	0,2
	0,2	0,3	0,4
	0,1	0,6	0,5
	0,3	0,4	0,7

Tabel 3. Contoh Bobot *Hidden Layer* W_0

W0 =	0,4	0,6	0,1	0,5	0,8	0,4	0,5	0,2
	0,9	0,2	0,1	0,5	0,7	0,3	0,2	0,4
	0,4	0,2	0,5	0,1	0,3	0,6	0,1	0,8

Ketiga, hitung nilai *hidden layer.*

$$H = X_{konteks} * W_1 \quad (2.5.1)$$

Dengan, H : nilai *hidden layer*

$X_{konteks}$: vektor konteks

W_1 : Bobot $layer$

Tabel 4. Nilai *Hidden Layer* H

X								
$X_{konteks}$	0	1	0	0	0	0	0	0
H =	0,1	0,5	0,8					

Keempat, hitung nilai *output hidden layer*

$$H_{out} = H * W_0 \quad (2.5.2)$$

Dengan, H_{out} : nilai *hidden layer*

X_{target} : vektor target

W_0 : Bobot $layer$

Tabel 5. Nilai *Output Hidden Layer* H_{out}

$$H_{out} = H * W_0$$

H_{out}	0.81	0.32	0.46	0.38	0.67	0.67	0.23	0.86

Kelima, hitung *output* dengan fungsi *softmax* [0,1]

$$y_k = P_r(kata_k | kata_{konteks}) = \frac{\exp(k)}{\sum_n exp(n)} \quad (2.5.2)$$

Tabel 6. Nilai *Output* Dengan Nilai *Softmax*

exp(n) =	2,247	1,377	1,584	1,462	1,954	1,954	1,258	2,363
S exp(n) =	14,201							
yk =	0,158	0,096	0,112	0,103	0,137	0,137	0,088	0,166

Karena target memiliki vektor target (0 0 0 1 0 0 0 0) maka nilai *output* adalah 0,102966

bobot nilai matriks W_1 dan W_0 dapat diperbarui menggunakan *backpropagation*. Setelah ini, pelatihan dapat diteruskan terhadap berbagai pasangan konteks-target kata dari korpus data.

2.6.2 *Global Vector* (*GloVe*)

Global Vector for Word Representation (*GloVe*) adalah representasi kata untuk menghasilkan *word embedding* yang digunakan untuk menangani kesamaan/analogi kata (semantik kata), dan pengenalan entitas kata (Pennington, *et al*, 2014). *GloVe* adalah teknik *unsupervised learning* di mana proses pembelajaran representasi kata dari kemunculan kata dalam korpus tertentu dilakukan. *Unsupervised learning* merupakan metode yang tidak menggunakan data *training*, sehingga memperoleh data dengan mengklasifikasikan data yang ada menjadi beberapa bagian. *GloVe* memeriksa hubungan kata dengan menghitung berapa kali kata muncul bersamaan dalam korpus tertentu. Rasio peluang kejadian kata untuk mengkodekan berbagai bentuk makna dan dapat meningkatkan kinerja dalam masalah analogi kata (Pennington, *et al*, 2014).

Cara kerja *GloVe*

Pada prinsipnya *GloVe* memperoleh hubungan semantik antar kata didasarkan *co-occurrence* matriks (matrik kemunculan kata secara bersamaan). Contoh diberikan korpus kata-kata pada V, *co-occurance* matriks sebagai X yang membentuk matrik

V x V, dimana i adalah baris dan j adalah kolom dari X, X_{ij} menunjukkan kata i muncul bersamaan dengan kata j.

Contoh *co-occurrence* matriks ditunjukkan sebagai berikut.

Kalimat
"aku cinta matematika"
"aku cinta orang tua"
Tabel 7. Nilai Matriks *Co-occurrance*

	Aku	Cinta	Matematika	Orang	Tua
Aku	0	2	0	0	0
Cinta	2	0	1	1	0
matematika	0	1	0	0	0
Orang	0	1	0	0	1
Tua	0	0	0	1	0

Untuk mengukur kesamaan semantik antar kata, dibutuhkan tiga kata yang terlibat. Dari perhitungan matriks di atas bahwa i (aku) dengan j (cinta) adalah 2 yang mana dua kata itu muncul secara bersamaan dalam ukuran *window* =1.

Tabel 8. Contoh Probabilitas dan Rasio *GloVe*

Probability and Ratio	k=solid	k=gas	k=*water*	k=fashion
P(k|*ice*)	$1.9\ x\ 10^{-4}$	$6.6\ x\ 10^{-5}$	$3.0\ x\ 10^{-3}$	$1.7\ x\ 10^{-5}$
P(k|*steam*)	$2.2\ x\ 10^{-5}$	$7.8\ x\ 10^{-4}$	$2.2\ x\ 10^{-3}$	$1.8\ x\ 10^{-5}$
P(k|*ice*)/ P(k|*steam*)	8.9	$8.5\ x\ 10^{-2}$	1.36	0.96

Dengan memperhatikan entitas berikut :

P_{ik}/P_{jk} dimana

$$P_{ik} = X_{ik}/X_i \qquad (2.5.4)$$

P_{ik} menunjukkan probabilitas terjadinya kata i dan k secara bersamaan yang dihitung dengan membagi berapa kali i dan k muncul bersama X_{ik} dengan jumlah total i muncul dalam korpus X_i.

Sebagaimana contoh pada dua kata yaitu "*ice*" dan "*steam*". Kata dalam k dapat disebut "*probe word*".

a. Misal. k = *solid* sangat serupa dengan *ice* namun tidak serupa dengan *steam*, maka probabilitas P_{ik}/P_{jk} akan sangat tinggi (>1).
b. Mis. k = gas sangat serupa dengan *steam* namun tidak serupa dengan *ice*, maka probabilitas P_{ik}/P_{jk} akan sangat rendah (<1).
c. Mis. k = *water* mirip dengan *ice* dan *steam* atau Mis. k = *fashion* tidak mirip dengan *ice* dan *steam*, maka dari itu probabilitas P_{ik}/P_{jk} akan mendekati 1.

Maka apabila ingin menyatukan P_{ik}/P_{jk} ke pada hitungan vektor kata diperlukan statistik global ketika melakukan pembelajaran vektor kata.

2.6.3 *FastText*

FastText adalah library yang dikembangkan oleh Facebook yang dapat digunakan untuk *word embedding*. *FastText* merupakan metode yang merupakan pengembangan dari *Word2Vec*. Metode ini mempelajari representasi kata dengan mempertimbangkan informasi sub-kata. Setiap kata direpresentasikan sebagai sekumpulan karakter *n-gram* (Bojanowski, *et al*, 2017). *FastText* dibentuk dari pengembangan dari library *Word2Vec* yang telah cukup populer sebagai library *word embedding*. *FastText* juga memberikan pilihan algoritma *training* CBOW dan Skipgram, seperti yang ada pada *Word2Vec* (Nurdin, 2020).

Perbedaan *FastText* dan *Word2Vec* yaitu karena *FastText* merepresentasikan setiap kata dalam karakter *n-gram*, maka jika menggunakan *custom embedding* proses *training* akan memakan waktu lebih lama daripada menggunakan *Word2Vec*.

Word2Vec hanya melihat kata-kata disekitarnya atau secara makna (semantik), tapi tidak melihat seberapa penting tidaknya kata dengan mempertimbangkan seberapa intens kata itu muncul. Sementara *FastText* mampu menangani kata baik secara semantik dan kemunculan kata. Sehingga *FastText* bekerja secara *co-occurance* (kata yang muncul secara berurutan, misalnya. makan pagi, malam hari). Kelebihan lainnya yaitu *FastText* dapat menangani kata yang tidak ada dalam kamus kata (*out of vocabulary*). Misalnya kata yang mengandung imbuhan dan akhiran seperti "menggambarkan" vs "gambar" yang sebenarnya secara semantik mirip hanya beda dalam imbuhannya.

Secara umum, metode yang mengkaji representasi kata dalam vektor mengabaikan jenis, bentuk kata, dan setiap kata memiliki vektor yang berbeda. Ini adalah kendala untuk merepresentasikan kata-kata dalam bahasa dengan kosa kata yang banyak dan banyak kata yang tidak biasa. *FastText* bekerja dengan baik, dapat melatih model dengan cepat pada kumpulan data besar, dan dapat memberikan representasi untuk kata-kata yang tidak ada dalam data pelatihan. Jika sebuah kata tidak muncul selama pelatihan model, Anda dapat membaginya menjadi *n-gram* untuk mendapatkan *embedding* vektornya (Nurdin, 2020).

Berbeda dari *Word2Vec*, *FastText* tidak menggunakan hanya satu kata secara utuh untuk diproses, tapi *FastText* menggunakan *n-gram* didalam prosesnya.

Cara kerja *FastText*

Dalam dokumentasi yang berkaitan dengan *FastText* tidak terlalu dijelaskan secara rinci mengenai langkah-langkah algoritma ini bekerja. Namun singkatnya berikut implementasi dari algoritma ini bekerja (Joulin, *et al*., 2017).

a. Misal jika menggunakan trigram atau n=3
   Dengan kata "kecerdasan"
   Maka *FastText* merepresentasikan kata diatas dalam bentuk sebagai berikut (<ke, kec, ece, cer, erd, rda, das, asa, san, an>)

Sehingga *word* vector bisa diperoleh dengan rata-rata *word embedding* kata / *n-gram*

b. Kemudian proses klasifikasi dilakukan secara *linear clasification* (*multinomial logistic regression*), di mana proses ini menangani setiap kalimat maupun dokumen dalam fitur.

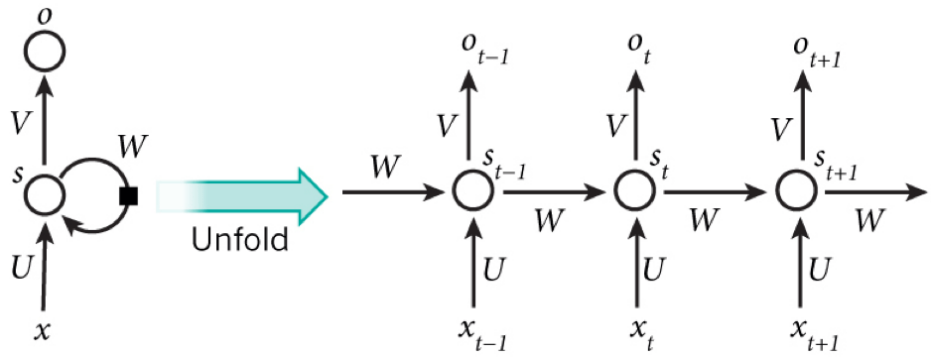
Selanjutnya lapisan *softmax* berguna untuk mendapatkan distribusi probabilitas melalui class yang telah ditentukan. *FastText* digunakan untuk menangani masalah dalam jumlah class yang besar. *FastText* memiliki hirarki *softmax* untuk mempercepat proses komputasi. *Hierarchial Softmax* bekerja berdasarkan *Huffman Coding Tree* yang digunakan untuk mengurangi kompleksitas komputasi O(kh) menjadi O(h log_2(k)), di mana k adalah jumlah kelas dan h adalah dimensi representasi teks (Joulin, A, *et al*., 2017).

2.7 *Recurrent Neural Network* (RNN)

Recurrent Neural Network adalah salah satu bentuk arsitektur *Artificial Neural Networks* (ANN). Seperti namanya RNN beroperasi secara berulang, dengan *output*nya tergantung pada *input* saat ini dan operasi sebelumnya (Wang, *et al*., 2021). *Recurrent Neural Network* merupakan salah satu bentuk arsitektur ANN yang dirancang khusus untuk memproses data yang bersambung/ berurutan (*sequential* data) (Raharjo, 2022).

Recurrent Neural Network adalah salah satu jaringan saraf tiruan yang paling cocok untuk mengenali pola urutan data, seperti teks, video, ucapan, bahasa, genom, dan data deret waktu. *Recurrent Neural Network* adalah algoritma yang sangat kuat dan dapat mengklasifikasikan, mengelompokkan, dan membuat prediksi tentang data, terutama deret waktu dan teks. Maka dari itu RNN dapat di implementasikan termasuk kedalam *Natural Language Processing*, pengenalan suara (*speech recognition*), terjemahan mesin (*machine translation*), pemodelan bahasa tingkat

karakter (*character-level language modeling*), klasifikasi gambar (*image classification*), keterangan gambar (*image captioning*), prediksi saham (*stock prediction*), dan rekayasa keuangan (*financial engineering*) (Raharjo, 2022).



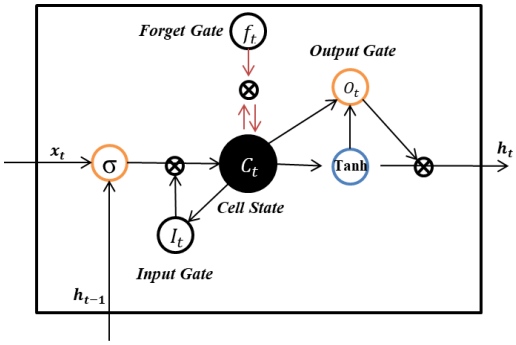
Gambar 3. Arsitektur RNN
(Fimansyah dkk, 2020)

2.8 *Long Short-Term Memory* (LSTM)

Jaringan LSTM adalah salah satu perkembangan dari RNN dengan kemampuan ingatan/memori yang diperluas secara eksplisit cocok untuk menangani perbedaan data jangka panjang dan sebagai solusi untuk mengurangi masalah gradien yang hilang (Pratama, 2022).

2.8.1 Arsitektur LSTM

Berikut gambar arsitektur LSTM lengkap.



Gambar 4. Arsitektur LSTM
(Chung and Shin, 2018)

LSTM menggunakan tiga *gate* yaitu *input gate, forget gate* dan *output gate*. Dalam LSTM satu komponen gate digunakan saat mengontrol informasi yang masuk kedalam memori yang bertugas memecahkan masalah dari gradien hilang. Koneksi yang berulang menambah keadaan atau memori ke jaringan dan memungkinkanya untuk memanfaatkan pengamatan yang terurut (Hermanto, dkk, 2021).

a. Forget Gate

Pada lapisan pertama disebut *forget gate* yaitu memutuskan fitur yang akan dihapus bertugas atau melupakan sebagian informasi dalam fitur yang tidak relevan dan sudah tidak dibutuhkan oleh sebuah sistem. Maka dari itu , LSTM dapat menyajikan informasi yang lengkap, tetapi tetap aktual sesuai kebutuhan. Cara kerja dalam *forget gate* ini data x_t adalah *input* data (vektor *input* x dalam *timestep* t) dan h_{t-1} adalah vektor *hidden state* dalam *timestep* sebelumnya t-1. Berikut perhitungan nilai *forget gate* (Pratama, 2022).

$$f_t = \sigma\left(W_f . [h_{t-1}, x_t] + b_i\right) \qquad (2.7.1)$$

b. *Input Gate*

Pada lapisan kedua yaitu *input gate* yang berfungsi untuk memasukkan informasi berguna untuk mendukung keakuratan data. Fungsi *input gate* adalah untuk menambahkan informasi sebelumnya telah diseleksi melewati gerbang *forget gate*. Pada *gate* ini membuat calon vektor baru menggunakan fungsi aktivasi tanh yang ditambahkan dibagian *cell state* $\widetilde{C_t}$. Berikut perhitungan *input gate* (Pratama, 2022).

$$i_t = \sigma\,(W_f.[h_{t-1}, x_t] + b_i) \tag{2.7.2}$$

$$\widetilde{C_t} = tanh\,(W_c.[h_{t-1}, x_t] + b_c) \tag{2.7.3}$$

Selanjutnya memperbaharui nilai *cell state* lama menjadi *cell state* yang baru menggunakan fungsi berikut.

$$C_t = f_t * C_{t-1} + i_t + \widetilde{C_t}) \tag{2.7.4}$$

c. Output Gate

Pada lapisan terakhir yaitu *output gate* yang berfungsi menjalankan aktivasi yang ditentukan untuk menghasilkan nilai *output* pada *hidden state* dan menempatkan *cell state* pada tanh. Setelah menghasilkan nilai *output hidden state* dan nilai *output* tanh, kedua hasil aktivasi tersebut dilakukan perkalian sebelum pergi ke langkah selanjutnya. Berikut perhitungan kedua *output* nya.

$$o_t = \sigma\,(W_o.[h_{t-1}, x_t] + b_o) \tag{2.7.5}$$

$$h_t = o_c * \tanh\,(C_t) \tag{2.7.6}$$

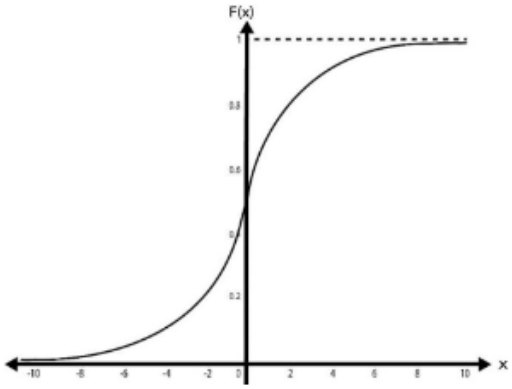
2.8.2 Fungsi Aktivasi

Fungsi aktivasi pada *neural network* adalah persamaan matematika yang menentukan *output* dari *neural network*. Fungsi aktivasi diimplementasikan pada setiap neuron dalam sebuah jaringan, dan dapat memutuskan apakah neuron

tersebut perlu diaktifkan atau tidak berdasarkan relevansi dari setiap *input* neuron untuk memprediksi/mengklasifikasikan model (Sagala, 2022).

Fungsi aktivasi harus efisien secara komputasi dikarenakan harus menghitung neuron yang banyak untuk setiap sampel data. Fungsi aktivasi juga dapat menormalisasi *output* pada setiap neuron agar mempunyai nilai dalam *range* antara 1 dan 0, atau antara -1 dan 1. Terdapat berbagai jenis fungsi aktivasi, sebagai berikut

a) Fungsi sigmoid (fungsi *logistic*)



Gambar 5. Fungsi Aktivasi *Sigmoid*
(Akbar dkk, 2022)

Fungsi aktivasi sigmoid digunakan untuk mengatur seberapa banyak informasi bisa lewat. Fungsi sigmoid menghasilkan *output* yang merupakan kisaran nilai antara 0 dan 1.

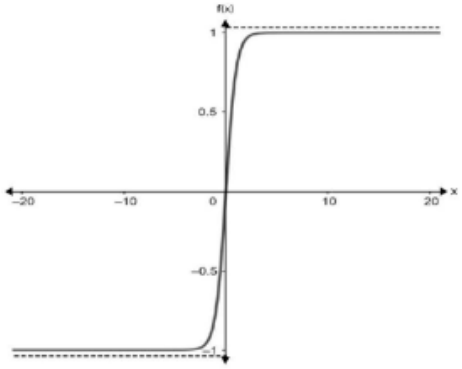
$$f(x) = (1 - e^{-x})^{-1} \qquad (2.7.7)$$

dengan :

e : bilangan *Euler*

x : data

b) Fungsi tanh (tangen hiperbolik)



Gambar 6. Fungsi Aktivasi Tangen Hiperbolik (Tanh)
(Akbar dkk, 2022)

Fungsi tanh merupakan pengembangan dari fungsi sigmoid. Fungsi aktivasi ini merupakan fungsi non linear. *Input* fungsi aktivasi ini adalah bilangan real dan *output* dari fungsi tersebut memiliki range antara -1 sampai 1.

$$\frac{dy}{dx} f(x) = \frac{dy}{dx} \frac{\sinh(x)}{\cosh(x)} \tag{2.7.8}$$

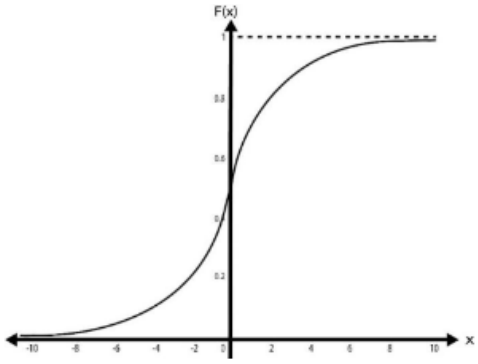
$$f(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})} \tag{2.7.9}$$

dengan :

e : bilangan *Euler*

x : data

c) Fungsi *Softmax*



Gambar 7. Fungsi Aktivasi *Softmax*
(Akbar dkk, 2022)

Fungsi *softmax* digunakan pada lapisan akhir dari model berbasis jaringan saraf. Jaringan seperti itu biasanya dilatih di bawah *log loss* (*cross-entropy*). Secara matematis *softmax* adalah fungsi berikut dengan x adalah vektor *input* ke lapisan *output* dan n mengindeks unit *output* dari 1,2, 3 …. n.

$$f(x) = \frac{\exp(x)}{\sum_n exp(n)} \tag{2.7.10}$$

dengan :

x : data *input*

2.8.3 Dropout

Dropout merupakan salah satu teknik untuk mencegah terjadinya *overfitting* dalam proses *machine learning*. Istilah *dropout* merujuk kepada penghentian penggunaan *nodes* secara sementara. Nilai *dropout* sebesar 20% sampai 50% dari jumlah *nodes* merupakan hal yang optimal untuk mencegah terjadinya *overfitting* (Srivastava,*et al*, 2014).

2.9 Akurasi

Akurasi adalah parameter untuk mengukur kualitas dari suatu model yang dibentuk. Terdapat beberapa parameter untuk mengevaluasi klasifikasi LSTM, beberapa parameter akurasi yang digunakan seperti *accuracy, precission, recall*, dan *F1-Score* (Pratama, 2022). Akurasi tersebut dirumuskan sebagai berikut :

a. *Accuracy*

Accuracy adalah nilai perbandingan prediksi *True Positive* (TP) dan *True Negative* (TN) dengan jumlah keseluruhan data.

$$Accuracy = \frac{TP + TN}{(TP + FN) + (FP + TN)} \quad (2.7.11)$$

b. *Precission*

Precission adalah nilai perbandingan prediksi *True Positive* (TP) dengan keseluruhan data yang diprediksi positif. *Precission* digunakan untuk menentukan seberapa sering data benar ketika memprediksi positif.

$$Precission = \frac{TP}{(TP + FP)} \quad (2.7.12)$$

c. *Recall*

Recall adalah nilai perbandingan prediksi *True Positive* (TP) dengan keseluruhan data yang benar positif. *Recall* mendefinisikan bagaimana data benar untuk semua data prediksi.

$$Recall = \frac{TP}{(TP + FN)} \quad (2.7.13)$$

d. *F1-Score*

F1-Score adalah nilai rata-rata *precission* dan *recall* yang dibobotkan.

$$F1 - Score = \frac{2TP}{2TP + FN + FP} \quad (2.7.14)$$

atau jika nilai precission dan recall sudah diketahui bisa menggunakan

$$F1 - Score = \frac{2(Precission \times Recall)}{Precission + Recall} \quad (2.7.15)$$

III. METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada semester ganjil tahun akademik 2022/2023, bertempat di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung

3.2 Spesifikasi Perangkat Penelitian

Komputer laboratorium Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung dengan spesifikasi :

Jenis	: *Personal Computer* (PC)
Merk/Tipe	: HP Pavilion *All-in-One* 24-k0xxx
Prosesor	: Intel(R) Core(TM) i7-10700T CPU @ 2.00GHz 1.99 GHz
Tipe Sistem	: 64-*bit operating system*, x64-*based processor*
RAM	: 8 GB
Software	: *Python* 3.0

3.3 Data Penelitian

Data yang digunakan merupakan data teks yang ada pada situs UCI Repository *https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29*. *Dataset* yang digunakan memiliki 6 attribut/variabel yaitu *drugName*, *condition, review*, *rating, date*, dan *usefulCount. DrugName* merupakan nama obat yang dipakai, *condition* merupakan kondisi yang dialami responden, *review* merupakan ulasan tentang obat yang responden gunakan, *rating* merupakan peringkat tentang obat yang digunakan berdasarkan responden, *date* merupakan tanggal responden mengisi data, dan *usefulCount* merupakan jumlah responden yang merasakan efek dan manfaat yang sama dengan ulasan yang diberikan. Data yang digunakan dalam penelitian ini adalah data kualitatif teks, dengan jumlah data 215063 *review*, dengan data train berjumlah 161297 dan data test berjumlah 53766.

Data yang digunakan adalah *review* dan *condition* untuk diklasifikasi, merupakan jenis data *supervised learning* atau data dengan ciri mempunyai label. *condition* akan digunakan sebagai label dalam klasifikasi teks menggunakan *word embedding Word2Vec*, *GloVe*, dan *FastText* dengan menggunakan metode LSTM. Data yang digunakan untuk *train* dikurangi dalam hal label dengan mempertimbangkan bahwa data yang terlalu banyak label dengan frekuensi data yang kecil akan mempengaruhi peforma model, maka akan digunakan 13 label kondisi penyakit berbeda mempertimbangkan jumlah frekuensi diatas 2000 review dan data selain kondisi sakit responden akan dihilangkan. Data di atas diunduh dalam bentuk CSV.

Berikut contoh sampel dari *dataset*.

Tabel 9. *Dataset Drugs Review*

	uniqueID	*drugName*	*Condition*	*Review*	*rating*	*date*	*usefulCount*
1	206461	Valsartan	*Left Ventricular Dysfunction*	*"It has no side effect, I take it in combination ...."*	9	20-May-12	27
2	95260	Guanfacine	*ADHD*	*"My son is halfway through his fourth week of ...."*	8	27-Apr-10	192
3	92703	Lybrel	*Birth Control*	*"I used to take another oral contraceptive, which had ...."*	5	14-Dec-09	17
..	.....	.....	.....	.....	.....	.....	.....

Tabel 10. Attribut *Condition*

No	*Condition*	Jumlah
1.	*Birth Control*	28788
2.	*Depression*	9069
3.	*Pain*	6145
....	...	.....
910.	*Hydrocephalus*	1

Tabel 11. Label Attribut *Condition*

No	*Condition*	Jumlah
1.	*Depression*	9069
2.	*Pain*	6145
3.	*Anxiety*	5903
4.	*Acne*	5588
5.	*Bipolar Disorder*	4224
6.	*Insomnia*	3673
7.	*Weight Loss*	3609
8.	*Obesity*	3568
9.	*ADHD*	3383
10.	*Diabetes, Type 2*	2554
11.	*High Blood Pressure*	2321
12.	*Vaginal Yeast Infection*	2274
13.	*Abnormal Uterine Bleeding*	2096

Tabel 12. Keterangan Label Attribut *Condition*

No	*Condition*	Keterangan

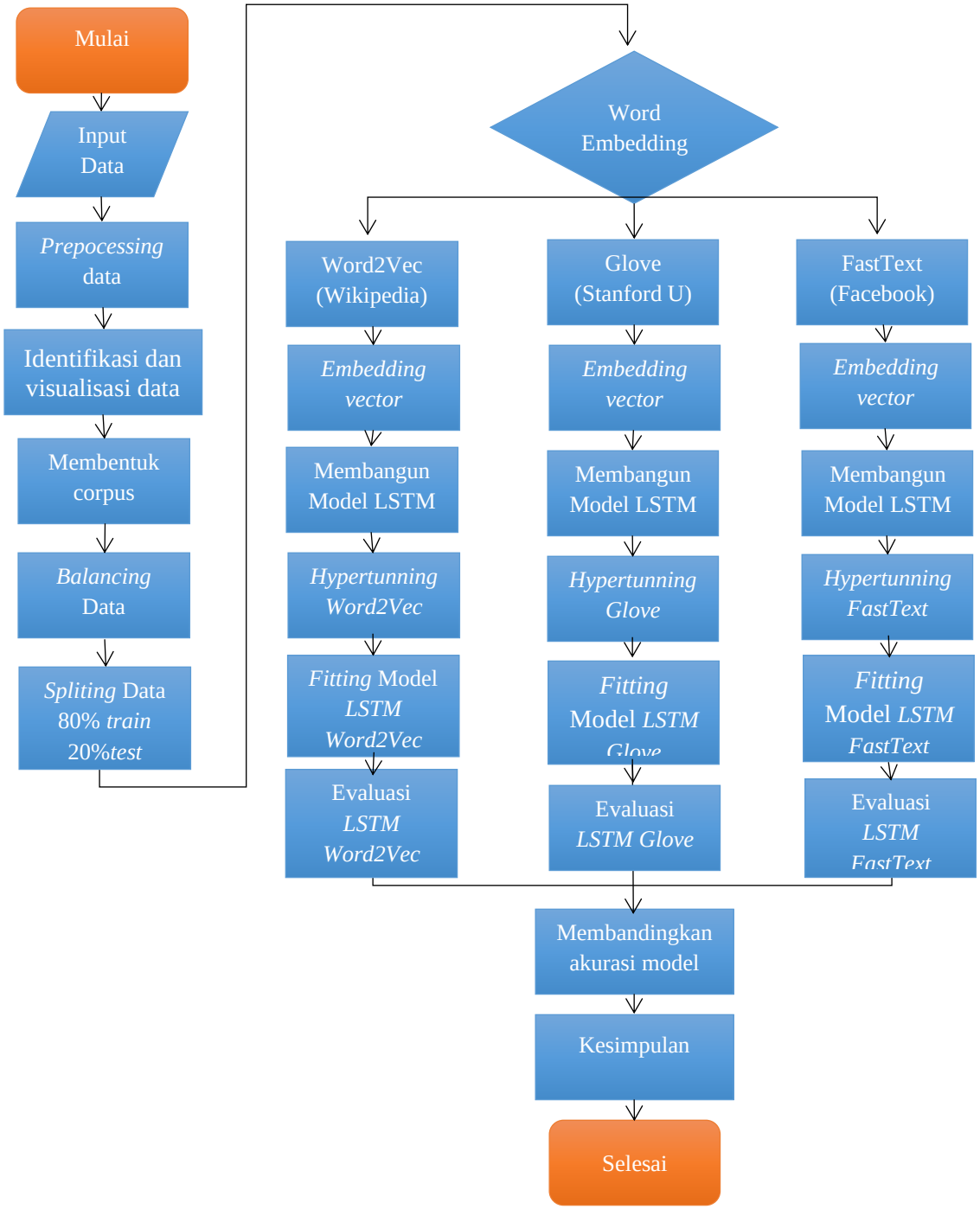
1.	*Depression*	Depresi
2.	*Pain*	Nyeri
3.	*Anxiety*	Kecemasan
4.	*Acne*	Jerawat
5.	*Bipolar Disorder*	Perubahan suasana hati mendadak
6.	*Insomnia*	Sulit tidur
7.	*Weight Loss*	Kehilangan berat badan
8.	*Obesity*	Obesitas
9.	*ADHD*	Hyperaktif yang menyebabkan susah fokus
10.	*Diabetes, Type 2*	Diabetes tipe 2
11.	*High Blood Pressure*	Tekanan darah tinggi
12.	*Vaginal Yeast Infection*	Infeksi jamur pada vagina
13.	*Abnormal Uterine Bleeding*	Pendarahan rahim tidak normal

3.4 Metode Penelitian

Berikut merupakan proses pengerjaan penelitian perbandingan *word embedding Word2Vec, GloVe,* dan *FastText* menggunakan metode *deep learning* (LSTM).

1. Melakukan studi literatur mengenai *word embedding Word2Vec, GloVe, FastText,* serta metode klasifikasi *deep learning* yaitu *Long Short-Term Memory* (LSTM)
2. Mengumpulkan data yang digunakan untuk proses klasifikasi menggunakan *word embedding Word2Vec, GloVe,* dan *FastText* menggunakan metode *deep learning* (LSTM).
3. Melakukan imput data pada *software python,* dengan *jupyter notebook* sebagai tempat menulis *code python.*
4. Melakukan *prepocessing* data agar data dapat diproses sesuai dengan *Prepocessing text* yaitu melewati tahap menghilangkan *missing value*, data duplikat, *cast folding, stopword,* dan *lemmatization*.

5. Berdasarkan hasil *prepocessing* ditinjau kembali dan agar mempermudah memahami sifat data, visualisasikan hasil data *prepocessing* kedalam *histogram* dan *wordcloud.*
6. Membentuk corpus kata sehingga kata-kata dalam data berbentuk vektor kuantitatif (sudah dalam bentuk numerik) dengan tahap *tokenization, word tokenization*, dan *pad sequence*, dari data yang telah ditokenisasi tadi akan dibentuk *word* indeks untuk semua kata yang ada dalam data, lalu diberi nomor untuk masing-masing kata, setelah itu dikembalikan kebentuk token urut sesuai nomor review tadi, dan terakhir bentuk vektor kata yang berisi angka-angka yang merepresentasikan kata-kata tadi.
7. Melakukan *balancing data* yaitu Random Over Sampling (ROS)
8. Melakukan *spliting* data 80% *training* dan 20% *testing*.
9. Melakukan pretrained model *word embedding Word2Vec, GloVe*, dan *FastText.*
10. Melakukan bobot matriks *embedding* dan panjang kata
11. Membangun model LSTM dengan menggunakan *python* dengan *spliting* data *training* yaitu 80% *training* dan 20% *validation*.
12. Menentukan parameter terbaik model LSTM, dengan menggunakan *hypertunning* untuk menentukan batch size, unit LSTM, dan epoch model.
13. Menjalankan model LSTM dengan parameter terbaik untuk masing-masing *word embedding Word2Vec, GloVe,* dan *FastText.*
14. Melakukan evaluasi model terbaik untuk melihat peforma model berupa nilai akurasi, *precision, recall*, dan *F1-Score* untuk melihat *embedding* terbaik.
15. Menampilkan confusion matrix untuk melihat peforma model.



Gambar 8. Flowchart *Word Embedding-LSTM*

V. KESIMPULAN

5.1 Kesimpulan

Penggunaan word embedding *Word2Vec, GloVe* dan *FastText* dengan metode deep learning yaitu LSTM dalam melakukan klasifikasi terhadap data *Drugs Review* (ulasan kondisi pengguna obat kesehatan) mendapat hasil yang baik. Kesimpulan yang didapatkan selama pengerjaan penelitian ulasan kondisi pengguna obat kesehatan dengan word embedding *Word2Vec, GloVe* dan *FastText* dengan metode LSTM sebagai berikut :

1. *Model word embedding pretrained* lebih cocok untuk penggunaan kasus data berbahasa inggris, terkhusus pada *word embedding GloVe*. Untuk model *Word2Vec dan FastText* terdapat fitur salah satunya bahasa indonesia dan terdapat model custom untuk bahasa lainnya dan lebih fleksibel.
2. *Model word embedding Word2Vec, GloVe* dan *FastText* dengan metode LSTM sangat baik dalam mengklasifikasikan kondisi pengguna obat kesehatan dari data *Drugs Review*, dan menghasilkan nilai akurasi berturut-turut 85.20%, 84.19%, 86.22%.
3. *Model word embedding FastText* dengan metode LSTM memiliki nilai akurasi dan *F1-Score* tertinggi dibandingkan kedua *embedding* lainnya dengan nilai akurasi 86.22% dan *F1-Score* 86%

5.2 Saran

Saran yang dianjurkan untuk penelitian selanjutnya adalah penggunaan *limit* kata dalam *word embedding pretrained* dimaksimalkan sehingga diharapkan memiliki hasil lebih optimal. Penelitian selanjutnya dapat pula menambah parameter LSTM yang di *hypertunning* sehingga mendapat pilihan nilai yang menghasilkan nilai terbaik. Terakhir untuk penelitian selanjutnya dapat memperhatikan kombinasi *spliting* data untuk mengetahui jumlah yang cocok untuk menjadi data *training* sehingga mendapat hasil maksimal seperti 80% dan 20%, 70% dan 30%. Terdapat pula berbagai macam *embedding* yang dapat digunakan seperti *Doc2Vec* dan metode *deep learning* lainnya seperti ANN, CNN, dan BERT.

DAFTAR PUSTAKA

Ali, D., Missen M, M, S., Husnain, M. 2021. Multiclass Event Classification from Text. *Hindawi Scientific Programming.* **2021**(1) : 1-15

Ariyulinda, N. 2018. Urgensi Pembentukan Regulasi Penjualan Obat Melalui Media Online. *Jurnal Legislasi Indonesia.* **15**(1) : 37-48

Akbar, K., Hayaty, M. 2020. Data Balancing untuk Mengatasi Imbalance Dataset pada Prediksi Produksi Padi. *Jurnal Ilmiah Intech : Information Technology Journal of UMUS.* **2**(2) : 1-14

Akbar, R., Santoso, R., Warsito, B. 2022. Prediksi Tingkat Temperatur Kota Semarang Menggunakan Metode *Long Short-Term Memory* (LSTM). *Jurnal Gaussian.* **11**(4) : 572-579.

Bojanowski, Grave, E., Joulin, A., Mikolov, T. 2017. Enriching *Word* Vectors with Sub*word* Information. hlm, 135–146. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spanyol.

Cheon Na, J. and Kyaing, W. Y. M. 2015. Sentiment Analysis of User-Generated Content on Drug Review Websites. *Journal of Information Science Theory and Practice*. **3**(1): 6-23.

Chung, H., Shin, K. 2018. Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction. *MDPI.* **10**(10):1-18

Engelen, J, E, v., Hoss, H, H. 2019. A Survey On Semi-Supervised Learning. *Machine Learning*. **109**(1):373-440.

Firmansyah, M, R., Ilyas, R., Kasyidi, F. 2020. Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network. Hlm. 488-495. *Prosiding The 11th Industrial Research Workshop and National Seminar*. Bandung

Hermanto, T. H., Setyanto, A., Luthfi, E. T. 2021. Algoritma LSTM-CNN untuk Sentimen Klasifikasi dengan *Word*2vec pada Media Online. *Citec Journal.* **8**(1): 64-77.

Indriani, A. 2014. Klasifikasi Data Forum dengan menggunakan Metode Naïve Bayes Classifier. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*. pp 5-10

Jain, A., Kulkarni, G., Shah, V. 2018. Natural Language Processing. *International Journal of Computer Sciences and Engineering.* **6**(1):161-167.

Joulin, A., Bojanowski, Grave, E., Mikolov, T. 2017. Bag of Tricks for Efficient Text Classification. hlm, 427–431. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spanyol.

Khomsah, S., Ramadhan, R. D., Wijayanto, S. 2021. The Accuracy Comparison Between *Word*2Vec and FastText On Sentiment Analysis of Hotel Reviews. Jurnal Rekayasa Sistem dan Teknologi Informasi (JURNAL RESTI). **5**(2): 352-358.

Liddy, E. D. 2001. *Natural Language Processing*. Encyclopedia of Library and Information Science. New York.

Migunani, Aditama., K. 2020. Pemanfaatan Natural Language Processing Dan Pattern Matching Dalam Pembelajaran Melalui Guru Virtual. *Elkom*. **13**(1):121-133.

Mikolov, T., Chen, K., Corrado, G., Dean, J. 2013. Efficient Estimation of *Word* Representations in Vector Space. hlm, 1-12. *1st International Conference on Learning Representations, ICLR Workshop Track Proceedings.* USA.

Mishra, S. 2021 Drug Review Sentiment Analysis using Boosting Algorithms. *International Journal of Trend in Scientific Research and Development (IJTSRD).* **5**(4): 937-941

Mitchell, T, M. 1997. *Machine Learning*. McGraw-Hill Science/Engineering/Math. New York.

Nagersian, F., Samulowitz, H., Khurana, U., Khalil, E. B., Turaga, D. 2017. Learning Feature Engineering for Classification. hlm 2529-2535 *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17).* Melbourne, Australia.

Nurdin, A., Aji, B. A. S., Bustamin, A., Abidin, Z. 2020. Perbandingan Kinerja *Word Embedding Word*2vec, *GloVe*, Dan Fasttext Pada Klasifikasi Teks. *Jurnal TEKNOKOMPAK.* **14**(2): 74-79

Pennington, J., Socher, and R., Manning, C. D. 2014. *GloVe*: Global Vectors for *Word* Representation. hlm, 1532-1543 Proceedings of the 2014 *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Qatar.

Pratama, E. D., 2022. Implementasi Model *Long-Short Term* Memory (LSTM) pada Klasifikasi Teks Data SMS Spam Berbahasa Indonesia. *The Journal on Machine Learning and Computating Intelligence (JMLCI)* **1**(2): 38-42

Raharjo, B. 2022. *Deep Learning dengan Python*. Yayasan Prima Agus Teknik. Semarang.

Rao, A. and Spasojevic, N. 2016. *Actionable and Political Text Classification using Word Embeddings and LSTM*. Lithium Technologies. San Francisco.

Rong, X. 2016. *Word*2Vec Parameter Learning Explained. *arXiv*. pp 1-21.

Ruiz, C, C., Bedmar, I, S. 2020. Comparing *Deep Learning* Architectures for Sentiment Analysis on Drug Reviews. *Journal of Biomedical Informatics 110*.

Sagala, L, O, A, S. 2022. Klasifikasi Cats dan Dogs. *Researchgate Publication*. pp 1-6.

Sihombing, P, R., Arsani, A, M. 2021. Comparison Of Machine Learning Methods In Classifying Poverty In Indonesia In 2018. *Jurnal Teknik Informatika (JUTIF)*. **2**(1):51-56.

Silviana, Kurniawan, R., Nazir, A., Budianita, E., Syafria, F., Gusti, S. K. 2022. *Jurnal Informasi dan Komputer*. **10**(1): 154-163.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. **15**(1):1929-1958.

Sutskever, I., Mikolov, T., Chen, K., Corrado, G., Dean, J. 2013. Distributed Representations of *Word*s and Phrases and their Compositionality. hlm, 3111-3119. *NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems*. Curran Associates, Inc. USA.

Thoomkuzhhy, A. M. 2020. *Drug Reviews: Cross-Condition and Cross-source Analysis by Review Quantification Using Regional CNN-LSTM Models* Masters Dissertation. Technological University Dublin.

Vijayaraghavan, S., Basu, D. 2020. Sentiment Analysis in Drug Reviews using Supervised Machine Learning Algorithms. *arXiv*

Wang, Q., Li,W., Jin, Z. 2021. Review of Text Classification in *Deep Learning*. *Open Access Library Journal*. **8**(1):1-8.