



TESIS - IS185401

**Deteksi Hate Speech Menggunakan Long-Short  
Term Memory (LSTM) dan Gate Recurrent Unit  
(GRU) di Twitter Berbahasa Indonesia**

**Akhmad Irsyad**  
NRP. 05211850010014

Dosen Pembimbing  
Nur Aini Rakhmawati, M.Sc.Eng., Ph.D.

Departemen Sistem Informasi  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
2019

*(Halaman sengaja dikosongkan)*

# LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

**Magister Komputer (M.Kom)**

di

**Institut Teknologi Sepuluh Nopember**

Oleh:

**Akhmad Irsyad**

**NRP: 05211850010014**

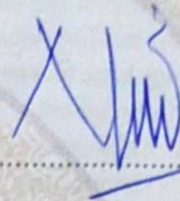
Tanggal Ujian: 02 Januari 2020

Periode Wisuda: Maret 2020

Disetujui oleh:

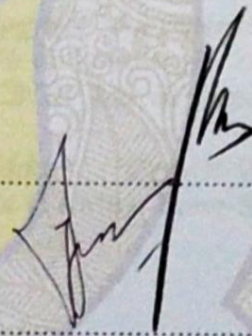
**Pembimbing:**

1. Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D.  
NIP: 198201202005012001



**Penguji:**

1. Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom  
NIP: 197302191998021001
2. Dr. Apol Pribadi Subriadi, S.T., M.T.  
NIP: 197002252009121003



Kepala Departemen Sistem Informasi  
Fakultas Teknologi Elektro dan Informatika Cerdas



**Dr. Mudjahidin, S.T., M.T**  
NIP: 197010102003121001

*(Halaman sengaja dikosongkan)*

## Judul

Nama Mahasiswa : Akhmad Irsyad  
NRP : 05211850010018  
Dosen Pembimbing : Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D

## ABSTRACT

*Hate speech is the use of aggressive, abusive, or offensive language that targets certain groups or people who attack gender (sexism), ethnic or racial groups (racism) or beliefs, and religion. Social media technology allows messages to be sent quickly, to be broad, and even viral if the topic attracts public attention. Unfortunately, this also means that Hate speech can spread easily and quickly so that it can cause conflict between groups in society. Therefore researchers try to detect Indonesian hate speeches on social media, especially Twitter. This research was conducted using Twitter as the data source. In this study, researchers tried to compare existing word embedding methods such as continuous Bag-Of-Word (CBOW), skip-gram, and Embeddings from Language Models (ELMo). The purpose of this research is to be able to detect hate speech using the Indonesian language by using deep learning methods such as Long-Short Term Memory (LSTM) and Gate Recurrent Unit (GRU). From the research conducted in the results obtained from all experiments conducted GRU-Skipgram experiments are experiments with the largest F1-score with a value of 94.67%. and the largest recall is GRU-Skipgram with a recall value of 99.09%. GRU-CBOW has the greatest precision with a value of 90.96%.*

*Keywords: Hate Speech; Deep Learning; Word Embedding ; Bag-Of-Word (CBOW); skip-gram; Embeddings from Language Models (ELMo); Long-Short Term Memory (LSTM) ; Gate Recurrent Unit (GRU)*

*(Halaman sengaja dikosongkan)*

## Judul

Nama Mahasiswa : Akhmad Irsyad  
NRP : 05211850010018  
Dosen Pembimbing : Nur Aini Rakhmawati, S.Kom., M.Sc.Eng., Ph.D

## ABSTRAK

*Hate speech* adalah penggunaan bahasa yang agresif, kasar atau ofensif yang menargetkan kelompok atau orang tertentu, yang menyerang jenis kelamin (seksisme), kelompok etnis atau ras (rasisme) atau keyakinan dan agama. Teknologi media sosial memungkinkan pesan untuk dikirim dengan cepat, menjadi luas dan bahkan viral jika topik tersebut menarik perhatian publik. Sayangnya, ini juga berarti bahwa *Hate speech* dapat menyebar dengan mudah dan cepat sehingga dapat menyebabkan konflik antar kelompok dalam masyarakat, maka dari itu peneliti mencoba mendeteksi *hate speech* berbahasa Indonesia yang ada di media sosial khususnya Twitter. Penelitian ini dilakukan dengan menggunakan twitter sebagai sumber datanya. Dalam penelitian ini peneliti mencoba membandingkan metode *word embedding* yang telah ada seperti continuous Bag-Of-Word (CBOW), skip-gram dan *Embeddings from Language Models* (ELMo). Tujuan dari penelitian ini adalah dapat mendeteksi *hate speech* yang menggunakan bahasa Indonesia dengan menggunakan metode *deep learning* seperti *Long-Short Term Memory* (LSTM) dan *Gate Recurrent Unit* (GRU). Dari penelitian yang dilakukan di peroleh hasil Dari semua percobaan yang dilakukan percobaan GRU-Skipgram adalah percobaan dengan F1-score terbesar dengan nilai 94.67%. dan recall terbesar adalah GRU-Skipgram dengan nilai recall 99.09%. GRU-CBOW memiliki precision terbesar dengan nilai 90.96%.

Kata kunci : *Hate Speech*; *Deep Learning*; *Word Embedding* ; Bag-Of-Word (CBOW); skip-gram; *Embeddings from Language Models* (ELMo); Long-Short Term Memory (LSTM) ; Gate Recurrent Unit (GRU)

*(Halaman sengaja dikosongkan)*



## DAFTAR ISI

|                                 |     |
|---------------------------------|-----|
| LEMBAR PENGESAHAN .....         | i   |
| ABSTRACT .....                  | iii |
| ABSTRAK .....                   | v   |
| DAFTAR ISI.....                 | vii |
| DAFTAR GAMBAR .....             | xi  |
| DAFTAR TABEL.....               | xv  |
| BAB 1 PENDAHULUAN .....         | 1   |
| 1.1 Latar Belakang .....        | 1   |
| 1.2 Rumusan Masalah .....       | 4   |
| 1.3 Tujuan Penelitian.....      | 4   |
| 1.4 Manfaat Penelitian.....     | 4   |
| 1.5 Kontribusi Penelitian ..... | 4   |
| 1.5.1 Kontribusi Teoritis ..... | 5   |
| 1.5.2 Kontribusi Praktis .....  | 5   |
| 1.6 Keterbaruan (Novelty).....  | 5   |
| 1.7 Batasan Penelitan .....     | 6   |
| BAB 2 KAJIAN PUSTAKA.....       | 7   |
| 2.1 <i>Kajian Teori</i> .....   | 7   |
| 2.1.1 Media Sosial.....         | 7   |
| 2.1.2 Hate Speech.....          | 8   |
| 2.1.4 Deep Learning.....        | 10  |
| 2.1.5 Praproses .....           | 19  |
| 2.1.6 Word Embedding .....      | 24  |

|                                   |   |    |
|-----------------------------------|---|----|
| 2.1.8                             | <i>Part-of-speech Tagging</i> .....     | 27 |
| 2.1.9                             | Precision, Recall, dan F-Measure.....   | 28 |
| 2.2                               | Literatur Review .....                  | 29 |
| 2.3                               | Penelitian Terkait .....                | 31 |
| 2.3.1                             | Hate Speech Indonesia .....             | 32 |
| 2.3.2                             | LSTM .....                              | 33 |
| 2.3.3                             | GRU .....                               | 34 |
| BAB 3 METODOLOGI PENELITIAN.....  |   | 37 |
| 3.1                               | Tahapan Penelitian .....                | 37 |
| 3.1.1                             | Identifikasi Masalah.....               | 38 |
| 3.1.2                             | Studi Literatur .....                   | 38 |
| 3.1.3                             | Perancangan Pembuatan Model.....        | 38 |
| 3.1.4                             | Menyiapkan Data .....                   | 39 |
| 3.1.5                             | Praproses Data .....                    | 39 |
| 3.1.6                             | Tokenisasi .....                        | 40 |
| 3.1.7                             | Proses <i>labeling</i> .....            | 40 |
| 3.1.8                             | <i>Split Data</i> .....                 | 40 |
| 3.1.9                             | <i>Word Embedding</i> .....             | 40 |
| 3.1.10                            | <i>Embedding Layer</i> .....            | 41 |
| 3.1.11                            | <i>Deep Learning</i> .....              | 42 |
| 3.1.12                            | Pola <i>hate Speech</i> .....           | 43 |
| 3.1.13                            | Evaluasi Model Deteksi Hate Speech..... | 43 |
| 3.1.14                            | Penyusunan Laporan.....                 | 44 |
| 3.2                               | Rencana Penelitian .....                | 44 |
| BAB IV HASIL DAN PEMBAHASAN ..... |   | 45 |

|                        |   |    |
|------------------------|---|----|
| 4.1                    | Pengumpulan Data .....                  | 45 |
| 4.2                    | Hapus Duplicate Sebelum Praproses.....  | 46 |
| 4.3                    | Praproses.....                          | 46 |
| 4.3.1                  | <i>Cleansing</i> .....                  | 46 |
| 4.3.2                  | <i>Case Folding</i> .....               | 47 |
| 4.3.3                  | <i>Stopwords</i> .....                  | 47 |
| 4.3.4                  | <i>Stemming</i> .....                   | 48 |
| 4.4                    | Hapus Duplicate Setelah Praproses ..... | 48 |
| 4.5                    | Labeling.....                           | 49 |
| 4.7                    | <i>Word Embedding</i> .....             | 49 |
| 4.8                    | <i>Deep Learning</i> .....              | 52 |
| BAB V Kesimpulan ..... |   | 83 |
| 5.1                    | Kesimpulan.....                         | 83 |
| 5.2                    | Saran.....                              | 84 |
| DAFTAR PUSTAKA .....   |   | 85 |
| LAMPIRAN.....          |   | 93 |

*(Halaman sengaja dikosongkan)*

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 2. 1 Proses perulangan informasi pada RNN (Olah, 2015).....                       | 11 |
| Gambar 2. 2 RNN terdiri dari banyak salinan jaringan yang sama (Olah, 2015) .            | 11 |
| Gambar 2. 3 Layer tanh pada RNN (Olah, 2015) .....                                       | 11 |
| Gambar 2. 4 Perulangan dengan empat layer pada LSTM (Olah, 2015).....                    | 12 |
| Gambar 2. 5 Cell state pada LSTM (Olah, 2015) .....                                      | 13 |
| Gambar 2. 6 Sigmoid layer pada LSTM (Olah, 2015).....                                    | 13 |
| Gambar 2. 7 Langkah pertama metode LSTM "Forget gate layer" (Olah, 2015) .               | 14 |
| Gambar 2. 8 Langkah kedua metode LSTM "Input gate layer & tanh layer" (Olah, 2015) ..... | 15 |
| Gambar 2. 9 Langkah ketiga metode LSTM "Membuat cell state baru" (Olah, 2015) .....      | 15 |
| Gambar 2. 10 Langkah keempat metode LSTM "menentukan output" (Olah, 2015) .....          | 16 |
| Gambar 2. 11 Perulangan dengan empat layer pada GRU (Simeon Kostadinov, 2017) .....      | 16 |
| Gambar 2. 12 tahap pertama GRU ( <i>Update Gate</i> ) (Simeon Kostadinov, 2017) ..       | 17 |
| Gambar 2. 13 tahap kedua pada GRU ( <i>Reset Gate</i> ) (Simeon Kostadinov, 2017)        | 18 |
| Gambar 2. 14 tahapan Konten memori saat ini (Simeon Kostadinov, 2017) .....              | 18 |
| Gambar 2. 15 proses terakhir dari GRU(Simeon Kostadinov, 2017) .....                     | 19 |
| Gambar 2. 16 Arsitektur (A) Skip-gram dan (b) CBOW (Meyer, 2016).....                    | 25 |
| Gambar 2. 17 <i>Confussion Matrix</i> .....  | 28 |
| Gambar 2. 18 main map metode klasifikasi pada <i>hate speech</i> .....                   | 31 |
| Gambar 3. 1 Tahapan Penelitian.....  | 37 |
| Gambar 3. 2 contoh 2D tensor .....   | 42 |
| Gambar 4. 1 Contoh Vector Skipgram.....  | 50 |
| Gambar 4. 2 Contoh Kedekatan kata menggunakan Skipgram .....                             | 50 |
| Gambar 4. 3 Contoh hasil Vector CBOW .....   | 51 |
| Gambar 4. 4 Contoh kedekatan kata menggunakan CBOW .....                                 | 51 |
| Gambar 4. 5 Contoh Kalimat .....   | 52 |

|  |    |
|--|----|
| Gambar 4. 6 Contoh Visualisasi kalimat ELMo .....                                    | 52 |
| Gambar 4. 7 Model GRU-CBOW .....   | 54 |
| Gambar 4. 8 akurasi dan loss model GRU-CBOW neuron 66 dan dropout 0 .....            | 54 |
| Gambar 4. 9 akurasi dan loss model GRU-CBOW neuron 66 dan dropout 0.2 ...            | 54 |
| Gambar 4. 10 akurasi dan loss model GRU-CBOW neuron 66 dan dropout 0.5 .             | 54 |
| Gambar 4. 11 akurasi dan loss model GRU-CBOW neuron 130 dan dropout 0 ..             | 55 |
| Gambar 4. 12 akurasi dan loss model GRU-CBOW neuron 130 dan dropout 0.2              | 55 |
| Gambar 4. 13 akurasi dan loss model GRU-CBOW neuron 130 dan dropout 0.5              | 55 |
| Gambar 4. 14 akurasi dan loss model GRU-CBOW neuron 190 dan dropout 0 ..             | 55 |
| Gambar 4. 15 akurasi dan loss model GRU-CBOW neuron 190 dan dropout 0.2              | 56 |
| Gambar 4. 16 akurasi dan loss model GRU-CBOW neuron 190 dan dropout 0.5              | 56 |
| Gambar 4. 17 Receiver Operating Characteristic GRU-CBOW .....                        | 56 |
| Gambar 4. 18 Model GRU-Skipgram .....  | 58 |
| Gambar 4. 19 akurasi dan loss model GRU-Skipgram neuron 66 dan dropout 0.            | 58 |
| Gambar 4. 20 akurasi dan loss model GRU-Skipgram neuron 66 dan dropout 0.2<br>.....  | 58 |
| Gambar 4. 21 akurasi dan loss model GRU-Skipgram neuron 66 dan dropout 0.5<br>.....  | 58 |
| Gambar 4. 22 akurasi dan loss model GRU-Skipgram neuron 130 dan dropout 0.5          | 59 |
| Gambar 4. 23 akurasi dan loss model GRU-Skipgram neuron 130 dan dropout 0.2<br>..... | 59 |
| Gambar 4. 24 akurasi dan loss model GRU-Skipgram neuron 130 dan dropout 0.5<br>..... | 59 |
| Gambar 4. 25 akurasi dan loss model GRU-Skipgram neuron 190 dan dropout 0.5          | 59 |
| Gambar 4. 26 akurasi dan loss model GRU-Skipgram neuron 190 dan dropout 0.2<br>..... | 60 |
| Gambar 4. 27 akurasi dan loss model GRU-Skipgram neuron 190 dan dropout 0.5<br>..... | 60 |
| Gambar 4. 28 Receiver Operating Characteristic model GRU-Skipgram .....              | 60 |
| Gambar 4. 29 hasil uji coba pengaturan model GRU-ELMo.....                           | 61 |
| Gambar 4. 30 model GRU ELMo .....  | 62 |

|  |    |
|--|----|
| Gambar 4. 31 akurasi dan loss model GRU-ELMo neuron 60 dan dropout 0.....            | 62 |
| Gambar 4. 32 akurasi dan loss model GRU-ELMo neuron 60 dan dropout 0.2...            | 62 |
| Gambar 4. 33 akurasi dan loss model GRU-ELMo neuron 60 dan dropout 0.5...            | 62 |
| Gambar 4. 34 akurasi dan loss model GRU-ELMo neuron 130 dan dropout 0....            | 63 |
| Gambar 4. 35 akurasi dan loss model GRU-ELMo neuron 130 dan dropout 0.2.             | 63 |
| Gambar 4. 36 akurasi dan loss model GRU-ELMo neuron 130 dan dropout 0.5.             | 63 |
| Gambar 4. 37 akurasi dan loss model GRU-ELMo neuron 190 dan dropout 0....            | 63 |
| Gambar 4. 38 akurasi dan loss model GRU-ELMo neuron 190 dan dropout 0.2.             | 64 |
| Gambar 4. 39 akurasi dan loss model GRU-ELMo neuron 190 dan dropout 0.5.             | 64 |
| Gambar 4. 40 Receiver Operating Characteristic GRU-ELMo .....                        | 64 |
| Gambar 4. 41 Model LSTM-CBOW .....   | 66 |
| Gambar 4. 42 akurasi dan loss model LSTM-CBOW neuron 66 dan dropout 0..              | 66 |
| Gambar 4. 43 akurasi dan loss model LSTM-CBOW neuron 66 dan dropout 0.2              | 66 |
| Gambar 4. 44 akurasi dan loss model LSTM-CBOW neuron 66 dan dropout 0.5              | 66 |
| Gambar 4. 45 akurasi dan loss model LSTM-CBOW neuron 130 dan dropout 0               | 67 |
| Gambar 4. 46 akurasi dan loss model LSTM-CBOW neuron 130 dan dropout 0.2<br>.....    | 67 |
| Gambar 4. 47 akurasi dan loss model LSTM-CBOW neuron 130 dan dropout 0.5<br>.....    | 67 |
| Gambar 4. 48 akurasi dan loss model LSTM-CBOW neuron 190 dan dropout 0               | 67 |
| Gambar 4. 49 akurasi dan loss model LSTM-CBOW neuron 190 dan dropout 0.2<br>.....    | 68 |
| Gambar 4. 50 akurasi dan loss model LSTM-CBOW neuron 190 dan dropout 0.5<br>.....    | 68 |
| Gambar 4. 51 Model LSTM Skipgram .....   | 70 |
| Gambar 4. 52 akurasi dan loss model LSTM-Skipgram neuron 66 dan dropout 0<br>.....   | 70 |
| Gambar 4. 53 akurasi dan loss model LSTM-Skipgram neuron 66 dan dropout 0.2<br>..... | 70 |
| Gambar 4. 54 akurasi dan loss model LSTM-Skipgram neuron 66 dan dropout 0.5<br>..... | 71 |

|  |    |
|--|----|
| Gambar 4. 55 akurasi dan loss model LSTM-Skipgram neuron 130 dan dropout 0 .....   | 71 |
| Gambar 4. 56 akurasi dan loss model LSTM-Skipgram neuron 130 dan dropout 0.2 ..... | 71 |
| Gambar 4. 57 akurasi dan loss model LSTM-Skipgram neuron 130 dan dropout 0.5 ..... | 71 |
| Gambar 4. 58 akurasi dan loss model LSTM-Skipgram neuron 190 dan dropout 0 .....   | 72 |
| Gambar 4. 59 akurasi dan loss model LSTM-Skipgram neuron 190 dan dropout 0.2 ..... | 72 |
| Gambar 4. 60 akurasi dan loss model LSTM-Skipgram neuron 190 dan dropout 0.5 ..... | 72 |
| Gambar 4. 61 Receiver Operating Characteristic LSTM-Skipgram .....                 | 72 |
| Gambar 4. 62 Model LSTM ELMo .....   | 74 |
| Gambar 4. 63 akurasi dan loss model LSTM-ELMo neuron 66 dan dropout 0....          | 74 |
| Gambar 4. 64 akurasi dan loss model LSTM-ELMo neuron 66 dan dropout 0.2.           | 74 |
| Gambar 4. 65 akurasi dan loss model LSTM-ELMo neuron 66 dan dropout 0.5.           | 74 |
| Gambar 4. 66 akurasi dan loss model LSTM-ELMo neuron 130 dan dropout 0.            | 75 |
| Gambar 4. 67 akurasi dan loss model LSTM-ELMo neuron 130 dan dropout 0.275 .....   | 75 |
| Gambar 4. 68 akurasi dan loss model LSTM-ELMo neuron 130 dan dropout 0.5 .....     | 75 |
| Gambar 4. 69 akurasi dan loss model LSTM-ELMo neuron 190 dan dropout 0..           | 75 |
| Gambar 4. 70 akurasi dan loss model LSTM-ELMo neuron 190 dan dropout 0.2 = .....   | 76 |
| Gambar 4. 71 akurasi dan loss model LSTM-ELMo neuron 190 dan dropout 0.576 .....   | 76 |
| Gambar 4. 72 Receiver Operating Characteristic LSTM_ELMo.....                      | 76 |
| Gambar 4. 73 Contoh perbandingan hasil (a) Skipgram dan (b) CBOW .....             | 79 |
| Gambar 4. 74 persebaran kata kasar pada CBOW .....                                 | 81 |
| Gambar 4. 75 persebaran kata kasar pada Skipgram .....                             | 81 |



## DAFTAR TABEL

|  |    |
|--|----|
| Tabel 2. 1 Kombinasi Awalan Akhiran Yang Tidak Diiijinkan .....                    | 21 |
| Tabel 2. 2 Cara Menentukan Tipe Awalan Kata Diawali Dengan “te-” .....             | 21 |
| Tabel 2. 3 Jenis Awalan Berdasarkan Tipe Awalannya.....                            | 22 |
| Tabel 2. 4 Aturan Awalan Pembentuk Kata Dasar .....                                | 23 |
| Tabel 2. 5 metode klasifikasi yang digunakan dalam deteksi hate speech .....       | 29 |
| Tabel 2. 6 metode <i>word embedding</i> yang digunakan dalam deteksi hate speech . | 30 |
| Tabel 2. 7 metode tokenisasi yang digunakan dalam deteksi hate speech.....         | 31 |
| Tabel 3. 1 contoh praproses .....  | 39 |
| Tabel 3. 2 contoh tokenisasi.....  | 40 |
| Tabel 3. 3 Rencana penelitian .....  | 44 |
| Tabel 4. 1 Contoh status twitter .....   | 45 |
| Tabel 4. 2 contoh status twitter yang sama .....                                   | 46 |
| Tabel 4. 3 contoh hasil cleansing .....  | 46 |
| Tabel 4. 4 contoh hasil <i>case folding</i> .....                                  | 47 |
| Tabel 4. 5 contoh hasil <i>stopword</i> .....                                      | 47 |
| Tabel 4. 6 contoh hasil <i>stemming</i> .....                                      | 48 |
| Tabel 4. 7 contoh status yang sama setelah praproses .....                         | 48 |
| Tabel 4. 8 hasil uji coba pengaturan model GRU-CBOW.....                           | 53 |
| Tabel 4. 9 confusion matrix .....  | 57 |
| Tabel 4. 10 hasil uji coba pengaturan model GRU-Skipgram .....                     | 57 |
| Tabel 4. 11 confusion matrix .....   | 61 |
| Tabel 4. 12 confusion matrix .....   | 65 |
| Tabel 4. 13 hasil uji coba pengaturan model LSTM-CBOW.....                         | 65 |
| Tabel 4. 14 confusion matrix .....   | 69 |
| Tabel 4. 15 hasil uji coba pengaturan model LSTM-Skipgram.....                     | 69 |
| Tabel 4. 16 confusion matrix .....   | 73 |
| Tabel 4. 17 hasil uji coba pengaturan model LSTM-ELMo .....                        | 73 |
| Tabel 4. 18 confusion matrix .....   | 77 |
| Tabel 4. 19 Contoh kesalahan penulisan kata .....                                  | 77 |

|   |    |
|---|----|
| Tabel 4. 20 Tabel Hasil pelatihan Model ..... | 79 |
| Tabel 4. 21 contoh klasifikasi benar .....    | 79 |
| Tabel 4. 22 contoh klasifikasi salah .....    | 80 |
| Tabel 4. 23 Pola <i>Hate speech</i> .....     | 82 |
| Tabel 4. 24 Pola <i>Non Hate Speech</i> ..... | 82 |

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Saat ini, jumlah pengguna media sosial meningkat pesat. Diperkirakan bahwa jumlah pengguna Twitter yang terdaftar pada 2016 telah mencapai 317 juta pengguna (Fatahillah, Suryati, & Haryawan, 2017). Facebook pada Juni 2017 memiliki 2 miliar pengguna aktif bulanan, yang lebih dari seperempat populasi manusia di bumi (Alfina, Mulia, Fanany, & Ekanata, 2017), ini menunjukkan bahwa media sosial telah menjadi media komunikasi yang penting saat ini. Teknologi media sosial memungkinkan pesan untuk dikirim dengan cepat, menjadi luas dan bahkan viral jika topik tersebut menarik perhatian publik. Sayangnya, ini juga berarti bahwa Hate speech dapat menyebar dengan mudah dan cepat sehingga dapat menyebabkan konflik antar kelompok dalam masyarakat.

*Hate speech* adalah salah satu informasi berbahaya atau negatif yang menyebar di media sosial. Penyebaran *hate speech* dilakukan oleh satu orang, atau kelompok, atau organisasi. Sementara itu, korban *hate speech* juga bisa satu orang, atau kelompok. Secara definisi *Hate speech* adalah penggunaan bahasa agresif, kasar atau ofensif, menargetkan kelompok atau orang tertentu, yang menyerang jenis kelamin (seksisme), kelompok etnis atau ras (rasisme) atau keyakinan dan agama (Watanabe, Bouazizi, & Ohtsuki, 2018). Penyebaran *hate speech* sangat berbahaya dalam kehidupan sosial manusia, seperti orang berpikir akan menjadi lebih bias, lebih banyak orang akan mendiskriminasi orang lain, orang cenderung kurang empati dan juga melanggar hak asasi manusia.

Tahun 2019 bagi Indonesia merupakan tahun yang sangat penting, hal ini dikarenakan bahwa pada tahun 2019 dilaksanakan berbagai pemilihan umum yang diselenggarakan oleh KPU, seperti pemilihan kepala daerah, pemilihan anggota dewan baik itu anggota Dewan Perwakilan Daerah (DPD), anggota Dewan Perwakilan Rakyat (DPR) baik nasional ataupun anggota dewan di tingkat wilayah baik provinsi, kabupaten atau kota dan juga yang tidak kalah pentingnya adalah

pemilihan Presiden dan Wakil Presiden RI periode masa bakti 2019 – 2024. Berjalannya mekanisme roda pemerintahan sebuah negara sangat bergantung oleh keberadaan presiden, maka posisi ini tak pernah luput dari perebutan posisi, mengingat sentralnya kedudukan ini. Pada tahun 2019 terdapat dua calon Presiden 2019 yaitu “Joko Widodo” dan “Prabowo Subianto”. Pembahasan mengenai pemilihan presiden tidak lepas dari pembahasan di media sosial, hal ini juga mengundang terjadinya hate speech.

Dari penelitian yang dilakukan yang membahas community detection dalam media sosial twitter (Irsyad & Rakhmawati, 2019) pembahasan tentang politik sering dilakukan orang-orang bahkan ketika menggunakan hastag yang tidak berhubungan dengan politik. Terlebih lagi saat bertepatan pemilihan presiden, pembahasan politik dapat menyebabkan terbentuknya komunitas baik itu komunitas yang melakukan dukungan, mengkritik, dan menyerang salah satu pasangan calon presiden, dalam pembahasan yang dilakukan tidak sedikit pembahasan yang mengandung hate speech.

*Sentiment analysis* dan *Hate speech* terkadang dianggap sama. Namun, dalam mendeteksi hate speech membutuhkan teknik yang lebih kompleks. Dalam *Sentiment analysis*, tugas utamanya adalah mendeteksi polaritas sentimen suatu kalimat, yang dimana outputnya berupa deteksi kata atau ekspresi termasuk kedalam sentiment yang positif atau negatif. Ini membuatnya mudah untuk mengandalkan makna langsung dari kata-kata, kata-kata biasanya memiliki polaritas sentimen yang sama terlepas dari konteks atau makna sebenarnya dengan sangat sedikit pengecualian (misalnya, kata "buruk" tidak dapat diartikan secara positif, dalam keadaan apa pun). Namun, dalam kasus hate speech, beberapa kata mungkin negatif, bahkan mungkin memiliki arti kebencian, tetapi konteksnya membuat mereka tidak membenci ucapan yang terkait (Martins, Gomes, Almeida, Novais, & Henriques, 2018).

Dalam studi yang didukung UNESCO (Gagliardone, Gal, Alves, Martinez, & Unesco., 2015) mengulas masalah meningkatnya kebencian-. Mereka menyarankan bahwa platform seperti Facebook dan Twitter tidak hanya mengadopsi pendekatan reaktif untuk menangani hate speech yang dilaporkan oleh

pengguna mereka, tetapi mereka seharusnya bisa melakukan yang lebih baik dengan data yang mereka miliki. Di Indonesia *hate speech* terutama mengenai ras dan agama, menjadi bentuk kejahatan *online* paling banyak dilaporkan pada tahun 2016, petugas polisi di Indonesia mengklaim sedikitnya 5 kasus dilaporkan setiap hari, yang berarti ada sekitar 150 kasus setiap bulan. Polisi juga mengatakan bahwa menangani kejahatan dunia maya tidak mudah karena dibutuhkan fasilitas dan sumber daya manusia (Alfina et al., 2017). Di Indonesia sendiri sudah terdapat undang-undang yang membahas tentang hate speech yaitu Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (UU ITE) pasal 27 dan 28, namun pihak kepolisian belum memiliki kemampuan untuk mendeteksi hate speech secara otomatis, polisi mengandalkan laporan yang dilakukan oleh korban *hate speech*.

Berdasarkan literature review yang dilakukan terhadap 30 penelitian yang membahas tentang klasifikasi hate speech di media sosial terdapat 12 penelitian yang menggunakan Deep Learning. Dari penelitian yang membandingkan antara Deep Learning dan Machine Learning dapat ditarik kesimpulan akurasi Deep learning lebih baik dari machine learning selain itu jumlah dataset yang digunakan dalam penelitian berpengaruh, ketika menggunakan dataset yang besar lebih baik menggunakan Deep learning.

Di Indonesia telah ada beberapa penelitian mengenai *Hate Speech*. Dimana peneliti menggunakan Facebook (Sutejo & Lestari, 2018), Instagram (Pratiwi, Budi, & Alfina, 2018) dan Twitter (Fatahillah et al., 2017; Pratiwi et al., 2018; Sutejo & Lestari, 2018) sebagai sumber dataset untuk mendeteksi *Hate speech*. Dalam mendeteksi *hate speech* telah banyak fitur digunakan seperti tokenisasi menggunakan n-gram, *bag of words* (BOW) (Alfina et al., 2017; Sutejo & Lestari, 2018), *Term Frequency-Inverse Document Frequency* (TF-IDF) (Sutejo & Lestari, 2018) dan fitur *word embedding* seperti *Continous Bag-Of-Words* (CBOW) (Sutejo & Lestari, 2018) dan *Skip-gram* (Pratiwi et al., 2018; Sutejo & Lestari, 2018).

Metode *Machine Learning* telah digunakan untuk mendeteksi *Hate Speech* di Indonesia seperti *Support Vector Machine* (SVM) (Pratiwi et al., 2018), *Decision Tree* (Pratiwi et al., 2018), *Naïve Bayes* (Fatahillah et al., 2017; Pratiwi et al.,

2018), *Logistic Regression* (Alfina et al., 2017; Pratiwi et al., 2018), dan *Random Forest Decision Tree* (Alfina et al., 2017; Pratiwi et al., 2018) adalah beberapa algoritma yang digunakan sebagai klasifikasi. Metode *Deep Learning* untuk deteksi *Hate Speech* telah mulai digunakan seperti menggunakan *Long-Short Term Memory* (LSTM) (Sutejo & Lestari, 2018). Pada penelitian ini akan mencoba membandingkan metode LSTM dan GRU untuk mendeteksi *hate speech*, selain itu penelitian ini juga akan membandingkan pengaruh *word embedding* CBOW, *Skip-gram* dan *Embeddings from Language Models* (ELMo) dalam mendeteksi *hate speech*.

## **1.2 Rumusan Masalah**

Berdasarkan penjelasan latar belakang, maka rumusan permasalahan yang menjadi fokus dan akan diselesaikan dalam penelitian ini antara lain:

1. Bagaimana membuat model *Deep Learning* untuk mengidentifikasi *hate speech* yang berhubungan dengan Jokowi dan Prabowo selama pemilihan presiden 2019 di Twitter berbahasa Indonesia ?
2. Bagaimana pengaruh *word embedding* terhadap tingkat performa identifikasi *hate speech* selama pemilihan presiden 2019?

## **1.3 Tujuan Penelitian**

Tujuan yang akan dicapai dari penelitian ini adalah untuk membuat model *deep learning* yang dapat mendeteksi *hate speech* selama pemilihan presiden 2019 yang ada di Twitter berbahasa Indonesia dan melihat pengaruh dari *word embedding* terhadap identifikasi *hate speech*.

## **1.4 Manfaat Penelitian**

Manfaat yang diharapkan dari penelitian ini adalah dengan menggunakan *deep learning* dan *word embedding* dalam identifikasi *hate speech* pada media sosial Indonesia diharapkan dapat meningkatkan tingkat akurasi dari identifikasi *hate speech*. Sehingga dapat menjadi kajian lebih lanjut pada penelitian – penelitian lain terkait dengan *Hate Speech*.

## **1.5 Kontribusi Penelitian**

Kontribusi dapat diberikan kepada bidang penelitian yang terkait yaitu berupa kontribusi teoritis serta kontribusi praktis yang dijabarkan sebagai berikut.

### **1.5.1 Kontribusi Teoritis**

Kontribusi secara teoritis yang diperoleh dari penelitian ini adalah membandingkan *word embedding* CBOW, Skipgram, dan ELMo sebagai fitur klasifikasi *hate speech* pada kampanye pemilihan presiden 2019 di media sosial twitter. Metode deep learning yang digunakan dalam klasifikasi *hate speech* adalah LSTM dan GRU. Penelitian ini mencoba menemukan pola *hate speech* pada kampanye pemilihan presiden 2019 di media sosial twitter dan memberikan pertimbangan dan masukan dalam memilih metode yang tepat berdasarkan kinerja klasifikasi. Hasil analisis ini dapat digunakan sebagai acuan, yaitu penentuan *word embedding* dan metode klasifikasi dalam mendeteksi *hate speech*.

### **1.5.2 Kontribusi Praktis**

Kontribusi praktis dari penelitian ini adalah (1) penggunaan Embeddings from Language Models (ELMo) pada data twitter berbahasa indonesia, dan (2) dapat mengklasifikasi *hate speech* yang terjadi selama pemilihan presiden Indonesia 2019.

### **1.6 Keterbaruan (Novelty)**

Berdasarkan penyusunan penelitian dari pendahuluan, perumusan masalah, tujuan penelitian dan manfaat penelitian dapat ditentukan keterbaruan (*novelty*) penelitian ini. Pada Penelitian sebelumnya mengenai *hate speech* umumnya adalah menggunakan fitur word2vec sebagai *word embedding*. Pada penelitian ini akan diusulkan menggunakan ELMo sebagai word embedding, yang dimana ELMo baru digunakan pada penelitian sentimen, selain itu metode pendekatan yang umumnya digunakan adalah *machine learning* namun dalam penelitian ini akan menggunakan *deep learning*. Pada penelitian mengenai identifikasi *hate speech* peneliti menggunakan fitur Embeddings from Language Models (ELMo) untuk melakukan *word embedding* dan membandingkan CBOW dan Skipgram. Dimana dataset yang digunakan twitter Bahasa indonesia hasil dari pencarian

berdasarkan kueri “jokowi” dan “prabowo” selama kampanye pemilihan presiden 2019.

### **1.7 Batasan Penelitian**

Batasan penelitian ditentukan agar penyelesaian penelitian dapat dikerjakan secara terfokus. Maka studi kasus batasan penelitian ini adalah sebagai berikut:

1. Objek penelitian adalah media sosial twitter yang menggunakan Bahasa Indonesia.
2. Dataset yang digunakan adalah hasil dari pencarian berdasarkan kueri “jokowi” dan “prabowo” selama kampanye pemilihan presiden 2019.
3. Hanya menggunakan Unigram.

Sistematika Penulisan Laporan Penulisan laporan penelitian dibuat secara sistematis dengan dilakukan sebagai berikut:

#### **Bab 1: Pendahuluan**

Bab ini berisi penjelasan terkait latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, kontribusi penelitian, dan sistematika penulisan laporan. Pada bab ini diharapkan mampu memberikan gambaran yang bersifat umum penelitian.

#### **Bab 2 : Kajian Pustaka**

Bab ini berisi pembahasan kajian teori dan penelitian terkait. Pada bab ini akan digunakan sebagai landasan penelitian.

#### **Bab 3 : Metodologi Penelitian**

Bab ini berisi pembahasan tahapan penelitian dan rencana penelitian. Pada bab ini akan menjelaskan detail dari tahapan penelitian beserta jadwal kegiatan pengerjaan penelitian. Selain itu, bab ini digunakan sebagai panduan dalam penyusunan penelitian agar terarah dan sistematis.



## **BAB 2**

### **KAJIAN PUSTAKA**

#### **2.1 *Kajian Teori***

Bab ini menjelaskan mengenai teori-teori yang digunakan dalam penyusunan thesis yang diambil dari penelitian-penelitian sebelumnya yang relevan. Kajian teori ini selanjutnya akan dibangun sebagai landasan dalam melakukan penelitian.

##### **2.1.1 Media Sosial**

Menurut laporan yang dikeluarkan oleh we are social, 2019 mengenai data jumlah pengguna website, mobile dan situs jejaring sosial di Indonesia, dengan rincian sejumlah 150 juta pengguna aktif internet, 150 juta pengguna aktif situs jejaring sosial dimana 130 juta penggunanya mengakses situs jejaring sosial menggunakan perangkat mobile. Pada laporan tersebut juga dijelaskan bahwa Twitter merupakan situs jejaring sosial kelima yang paling banyak digunakan oleh masyarakat Indonesia (we are social, 2019).

Twitter adalah media sosial yang dapat menghubungkan semua orang di dunia dan memungkinkan pengguna untuk berkomunikasi satu sama lain dalam pesan singkat 140 karakter yang disebut tweet. Twitter sendiri memungkinkan pengguna untuk mengikuti berita terbaru mengenai topik yang mereka minati. Twitter menyediakan akses ke data twitter melalui API. Setiap API menyajikan beberapa aspek Twitter dan memungkinkan pengembang untuk membangun dan memperluas aplikasi mereka dengan cara baru berdasarkan kreativitas mereka sendiri. Twitter terus mengembangkan API Twitter sehingga dapat terus mengalami perubahan (Azali 2016). Jumlah pengguna yang terus meningkat menjadikan Twitter menjadi media informasi yang banyak digunakan oleh pengguna internet untuk berbagi informasi. Twitter sendiri tidak memiliki sistem penyaringan mengenai apakah tweet yang ditulis positif atau negatif (Fatahillah, Suryati, and Haryawan 2017).

### 2.1.2 Hate Speech

*Hate speech* adalah penggunaan bahasa agresif, kasar atau ofensif, menargetkan kelompok atau orang tertentu, yang menyerang jenis kelamin (seksisme), kelompok etnis atau ras (rasisme) atau keyakinan dan agama (Watanabe et al., 2018a).

Di Indonesia pemerintah sudah memiliki peraturan mengenai hate speech yaitu terdapat pada Surat Edaran Kepala Kepolisian Negara Nomor SE/6/X/2015 Tahun 2015 tentang Penanganan Ujaran Kebencian (*Hate Speech*) dan Undang-Undang Nomor 11 Tahun 2008. Dalam Surat Edaran Kepala Kepolisian Negara Nomor SE/6/X/2015 Tahun 2015 dijelaskan bahwa ujaran kebencian dapat dikenakan tindak pidana yang diatur dalam KUHP dan ketentuan pidana lainnya di luar KUHP, bahwa ujaran kebencian sebagaimana dimaksud bertujuan untuk menghasut dan menyulut kebencian terhadap individu dan/atau kelompok masyarakat dalam berbagai komunitas yang dibedakan dari aspek: suku;agama; aliran keagamaan; keyakinan/kepercayaan; ras; antargolongan; warna kulit; etnis; gender; kaum difabel (cacat); orientasi seksual (Kepolisian Negara Republik Indonesia, 2015).

D dalam Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (UU ITE) juga mengatur tentang hate speech. Hukum yang mengatur tercantum dalam Pasal 27 ayat (3) dan Pasal 28 ayat (2) UU ITE , berbunyi : Setiap Orang dengan sengaja dan tanpa hak mendistribusikan dan/atau mentransmisikan dan/atau membuat dapat diaksesnya Informasi Elektronik dan/atau Dokumen Elektronik yang memiliki muatan penghinaan dan/atau pencemaran nama baik (Arsip, 2008).

Bunyi Pasal 28 ayat (2) UU ITE adalah sebagai berikut: Setiap Orang dengan sengaja dan tanpa hak menyebarkan informasi yang ditujukan untuk menimbulkan rasa kebencian atau permusuhan individu dan/atau kelompok masyarakat tertentu berdasarkan atas suku, agama, ras, dan antargolongan (SARA) (Arsip, 2008).

Menurut Handono (2019) terdapat beberapa jenis gaya bahasa yang sering digunakan dalam sosial media Indonesia, yaitu: Persamaan atau simile adalah gaya

bahasa yang membandingkan suatu hal dengan hal lain secara langsung. Contoh: “mimi kok mirip kodok si mukanya”; Eponim adalah gaya yang menghubungkan nama suatu karakter dengan sifat tertentu untuk menyatakan sifat itu sehingga nama itu dipakai untuk menyatakan sifat itu. Contoh: “Jempol kakinya kayak Hulk”; Sarkasme adalah gaya yang mengandung kepahitan dan celaan yang kasar, contoh: ”SAMPAH NI ORANG”; Satire adalah ungkapan yang menertawakan atau menolak sesuatu. Satire biasanya mengandung kritik tentang kelemahan manusia, contoh: “idungnya bisa dijadiin google tuh”. Menurut Handono (2019) jenis gaya bahasa sarkasme yang ada di sosial media termasuk dalam kategori *hate speech* karena menggunakan kata-kata kasar yang mana termasuk dalam unsur dari hate speech berupa penghinaan serta perbuatan tidak menyenangkan.

### **2.1.3 Pemilihan Presiden**

Presiden dalam bentuk negara republik dipilih untuk setiap periode tertentu. Dengan demikian, pergantian Presiden di negara berbentuk republik merupakan sebuah proses yang harus dilakukan dan umum terjadi. Kedudukan lembaga Presiden merupakan kedudukan yang paling sentral dalam sebuah sistem pemerintahan republik. Presiden merupakan eksekutor terhadap seluruh kebijakan ekonomi, politik, kebudayaan, dan pertahanan yang diambil. Berjalannya mekanisme roda pemerintahan sebuah negara sangat bergantung oleh lembaga ini, maka lembaga ini tak pernah luput dari perebutan posisi, mengingat sentralnya kedudukan ini. Pemilihan Presiden dan Wakil Presiden untuk menghasilkan Presiden dan Wakil Presiden Republik Indonesia yang terpilih adalah pemimpin bangsa, bukan hanya pemimpin golongan atau kelompok tertentu (Amalia & Andi, 2019).

Dalam pemilihan presiden terdapat proses kampanye. kampanye sebagai kegiatan yang dilakukan secara sadar untuk menunjang dan meningkatkan proses pelaksanaan yang terencana pada periode tertentu yang bertujuan mempengaruhi masyarakat dalam sasaran tertentu (Nuril, 2019). Keputusan Komisi Pemilihan Umum (KPU) No.35 tahun 2004 tentang kampanye Pemilihan Umum Presiden dan Wakil Presiden mengatur semua jenis atau bentuk kampanye, yaitu: Pertemuan terbatas; Tatap muka dan dialog; Penyebaran melalui media cetak dan media

elektronik; Penyiaran melalui radio dan atau televisi; Penyebaran bahan kampanye kepada umum; Pemasangan alat peraga di tempat umum; Rapat umum; Debat publik / debat terbuka antar calon; Kegiatan lain yang tidak melanggar peraturan perundang-undangan (Sa'diyah, 2017).

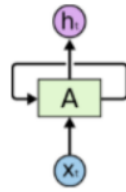
Selain kampanye secara resmi terdapat jenis kampanye lain seperti kampanye negatif yaitu menyerang pihak lain melalui sejumlah data atau fakta yang bisa diverifikasi dan diperdebatkan. kampanye hitam adalah kampanye yang merusak, menyindir, bahkan sampai memfitnah seseorang atau organisasi tertentu, yang dilakukan untuk menimbulkan persepsi yang tidak etis dan tidak benar. Kampanye hitam dilakukan untuk menyerang lawan politik, sehingga para pemilik suara akhirnya terpengaruh dan menurunkan elektabilitasnya. Menurut Febriyan (2017) bahwa terdapat beberapa jenis black campaign yang terjadi selama pemilu seperti Fitnah, Menghasut (mengadu domba) dan Penghinaan terhadap lawan politik

#### **2.1.4 Deep Learning**

*Deep learning* merupakan pengembangan dari Jaringan Saraf Tiruan (*Artificial Neural Network*) yang memiliki lebih banyak lapisan (layer). Dengan Lapisan yang lebih banyak, *Deep learning* diharapkan untuk dapat mengenali proses yang lebih kompleks. Dengan kata lain, *Deep Learning* merepresentasikan pengetahuan yang lebih detil (Putra, Djamal, & Ilyas, 2018)

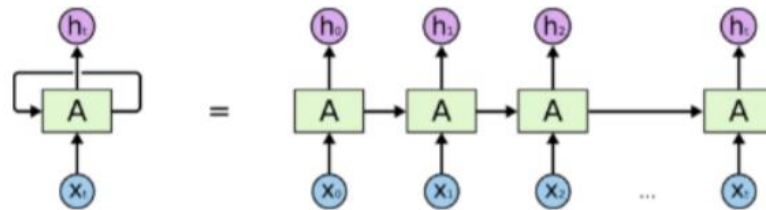
##### **A. Recurrent neural network (RNN)**

*Recurrent neural network* adalah bagian dari *neural network* yang memproses sejumlah data yang bersambung (*data sequential*). Secara umum, manusia tidak membuat keputusan dari awal setiap saat. Manusia selalu memperhitungkan informasi masa lalu dalam membuat keputusan. Seperti halnya manusia, *Recurrent Neural Network* menyimpan informasi dari masa lalu dengan melakukan *looping* dalam arsitekturnya seperti yang terlihat dalam gambar 2.1 yang secara otomatis membuat informasi dari masa lalu tetap tersimpan



Gambar 2. 1 Proses perulangan informasi pada RNN (Olah, 2015)

Pada gambar 2.1,  $X_t$  sebagai *input*,  $H_t$  sebagai *output* dan terdapat alur perulangan yang memungkinkan informasi dilewatkan dari satu langkah jaringan ke langkah berikutnya. *Recurrent Neural Network* dapat dianggap sebagai banyak salinan dari jaringan yang sama. Masing-masing jaringan mengirimkan pesan kepada jaringan berikutnya seperti terlihat pada gambar 2.3.

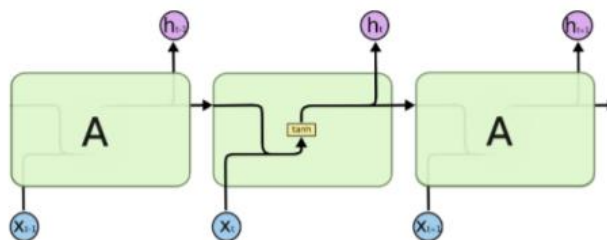


Gambar 2. 2 RNN terdiri dari banyak salinan jaringan yang sama (Olah, 2015)

Pada RNN terdapat beberapa *unit gate* seperti *Gate Recurrent Unit* (GRU), dan *Long Short Term Memory* (LSTM).

### B. Long Short Term Memory (LSTM)

*Long Short Term Memory* (LSTM) adalah salah satu variasi dari *Recurrent Neural Network* yang dibuat untuk menghindari masalah ketergantungan jangka Panjang pada *Recurrent Nueral Network* (RNN). LSTM dapat mengingat informasi jangka panjang. Pada RNN perulangan jaringan hanya menggunakan satu layer sederhana, yaitu layer tanh seperti pada gambar 2.3.



Gambar 2. 3 Layer tanh pada RNN (Olah, 2015)

Persamaan tanh diuraikan pada persamaan 1.

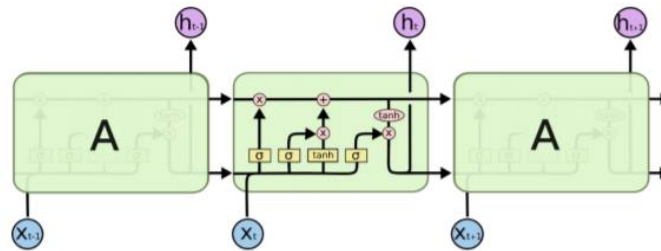
$$\tanh(x) = 2\sigma(2x) - 1 \quad (1)$$

Dimana:

$\sigma$  = fungsi aktivasi sigmoid

$x$  = data input

Sedangkan, LSTM memiliki empat *layer* pada perulangan modelnya seperti pada gambar 2.4.



Gambar 2. 4 Perulangan dengan empat layer pada LSTM (Olah, 2015)

Persamaan metode LSTM menurut Hochreiter & Schmidhber (1997)

diuraikan pada persamaan berikut:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$c\bar{t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * C\bar{t} \quad (5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad h_t = o_t * \tanh(C_t) \quad (6)$$

Dimana:

$f_t$  = forget gate

$W_f$  = nilai weight untuk forget gate

$i_t$  = input gate

$\sigma$  = fungsi sigmoid

$W_i$  = nilai weight untuk input gate

$h_{t-1}$  = nilai output sebelum orde ke t

$x_t$  = nilai input pada orde ke t

$b_i$  = nilai bias pada input gate

$h_t$  = nilai output orde t

$C\bar{t}$  = nilai baru yang dapat ditambahkan ke cell state

$\tanh$  = fungsi tanh

$W_C$  = nilai weight untuk cell state

$b_f$  = nilai bias untuk cell state

$C_t$  = Cell state

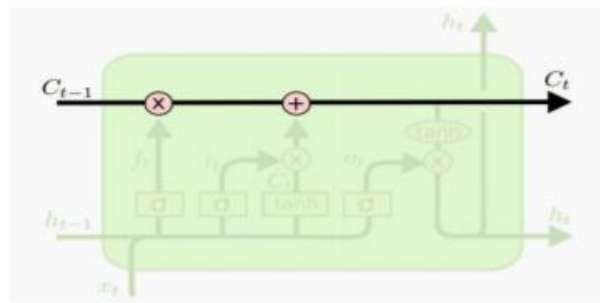
$C_{t-1}$  = Cell state sebelum orde ke t

$o_t$  = output gate

$W_o$  = nilai weight untuk output gate

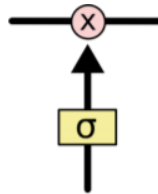
$b_o$  = nilai bias pada output gate

Kunci utama pada LSTM adalah *cell state*. *Cell state* adalah garis horizontal yang menghubungkan semua output layer pada LSTM seperti terlihat pada gambar 2.5.



Gambar 2. 5 Cell state pada LSTM (Olah, 2015)

LSTM memiliki kemampuan untuk menambah dan menghapus informasi dari *cell state*. Kemampuan ini disebut dengan *gates*. *Gates* sebagai pengatur apakah informasi akan diteruskan atau dihentikan. *Gates* terdiri dari *sigmoid layer* dan *pointwise multiplication operation* seperti yang terlihat pada gambar 2.6.



Gambar 2. 6 Sigmoid layer pada LSTM (Olah, 2015)

*Output* dari *sigmoid layer* adalah angka 1 atau 0 yang menunjukkan apakah informasi tersebut akan diteruskan atau dihentikan. Angka 0 menunjukkan bahwa tidak ada informasi yang akan diteruskan, sedangkan angka 1 menunjukkan bahwa semua informasi akan diteruskan. *Output gate*, *forget gate* dan *input gate* memerlukan nilai *sigmoid*, Persamaan *sigmoid* diuraikan sebagai berikut:

$$\sigma(x) = 1/(1 + e^{-x}) \quad (7)$$

Dimana:

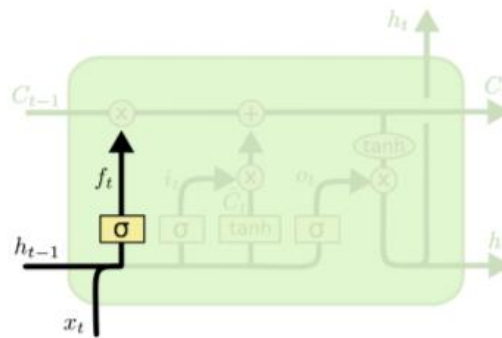
X = data input

e = konstanta matematika (2,71828 18284 59045 23536 02874 71352)

LSTM memiliki 3 jenis *gates* diantaranya adalah *forget gate*, *input gate* dan *output gate*. *Forget gate* adalah *gate* yang memutuskan informasi mana yang akan dihapus dari *cell*. *Input gate* adalah *gate* yang memutuskan nilai dari *input* untuk di diperbarui pada *state memori*. *Output gate* adalah *gate* yang memutuskan apa yang

akan dihasilkan *output* sesuai dengan input dan memori pada *cell* (Ibnu Habibie, 2018). Langkah - langkah panduan jalannya metode LSTM:

Langkah pertama adalah LSTM memutuskan informasi apa yang akan dihapus dari *cell state*. Keputusan ini dibuat oleh *sigmoid layer* yang bernama “*forget gate layer*”. *Forget gate layer* akan memproses  $h_{t-1}$  dan  $x_t$  sebagai *input*, dan menghasilkan *output* berupa angka 0 atau 1 pada *cell state*  $C_{t-1}$  seperti yang terlihat pada gambar 2.8.



Gambar 2. 7 Langkah pertama metode LSTM "Forget gate layer" (Olah, 2015)

Persamaan *forget gate* diuraikan pada persamaan 2. Persamaan *forget gate* memerlukan nilai *weight*, Untuk memperoleh nilai *weight* diuraikan pada persamaan:

$$w = -\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}} \quad (8)$$

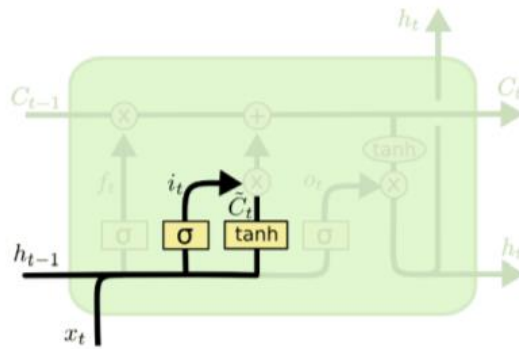
Dimana:

W = weight

d = jumlah data

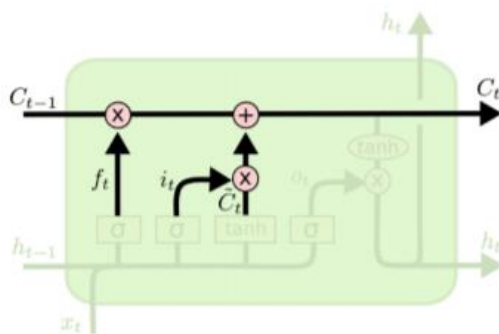
Langkah kedua adalah memutuskan informasi apa yang akan disimpan di *cell state*. Untuk langkah ini terdapat dua bagian. Bagian pertama, *sigmoid layer* yang bernama “*input gate layer*” memutuskan nilai mana yang akan diperbarui. Selanjutnya, *tanh layer* membuat satu kandidat dengan nilai baru,  $C \sim t$ , yang dapat ditambahkan ke *cell state*. Tahap selanjutnya adalah *output* dari *input gate layer* dan *tanh layer* akan digabungkan untuk memperbarui *cell state*. Langkah kedua digambarkan pada gambar 2.8. Persamaan input gate diuraikan pada persamaan 3. Dan Persamaan kandidat baru diuraikan pada persamaan 4.





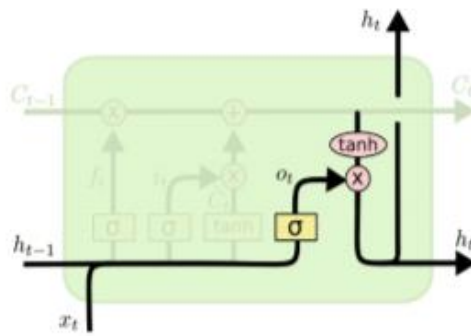
Gambar 2. 8 Langkah kedua metode LSTM "Input gate layer & tanh layer" (Olah, 2015)

Langkah ketiga adalah memperbarui *cell state* yang lama,  $C_{t-1}$ , menjadi *cell state* baru,  $C_t$  seperti digambarkan pada gambar 2.9. Dengan mengkalikan *state* lama dengan  $f_t$ , untuk menghapus informasi yang sudah ditentukan sebelumnya pada langkah *forget gate layer*. Selanjutnya, ditambahkan dengan  $i_t * C_{t-1}$ , yang merupakan nilai baru dan digunakan untuk memperbarui *state*. Persamaan *cell state* diuraikan pada persamaan 7.



Gambar 2. 9 Langkah ketiga metode LSTM "Membuat cell state baru" (Olah, 2015)

Langkah keempat adalah langkah terakhir dalam metode LSTM yang bertujuan untuk memutuskan hasil *output* digambarkan pada gambar 2.10. *Output* harus sesuai dengan *cell state* yang telah diproses terlebih dahulu. Pertama *sigmoid layer* memutuskan bagian dari *cell state* yang menjadi *output*. Selanjutnya, *output* dari *cell state* dimasukkan kedalam tanh layer (untuk mengganti nilai menjadi diantara -1 dan 1) dan dikalikan dengan *sigmoid gate*, agar *output* yang dihasilkan sesuai dengan apa yang kita putuskan sebelumnya, Persamaan output gate diuraikan pada persamaan 6.



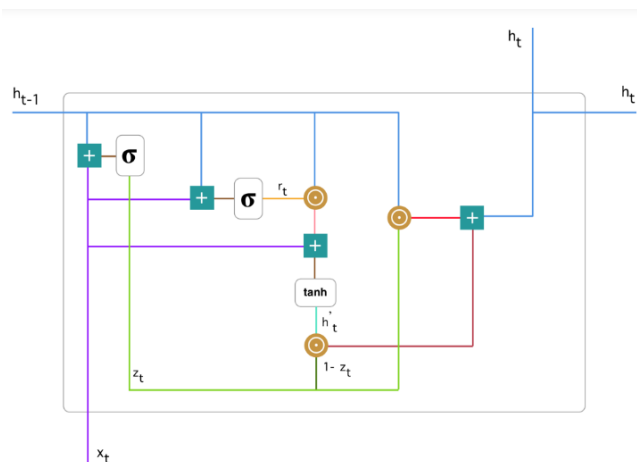
Gambar 2. 10 Langkah keempat metode LSTM "menentukan output" (Olah, 2015)

Untuk memperoleh nilai *output gate* diperlukan nilai *output* orde t, Persamaan nilai *output* orde t diuraikan sebagai berikut:

$$h_t = o_t * \tanh (C_t) \quad (9)$$

### C. Gate Recurrent Unit (GRU)

*Gate Recurrent Unit* (GRU) dikembangkan oleh Cho et al. (2014). Seperti LSTM, GRU dirancang untuk mengatur ulang atau memperbarui konten memorinya secara adaptif. Setiap GRU memiliki *reset gate* dan *update gate*. Gambar 2.11 adalah model dari layer GRU.



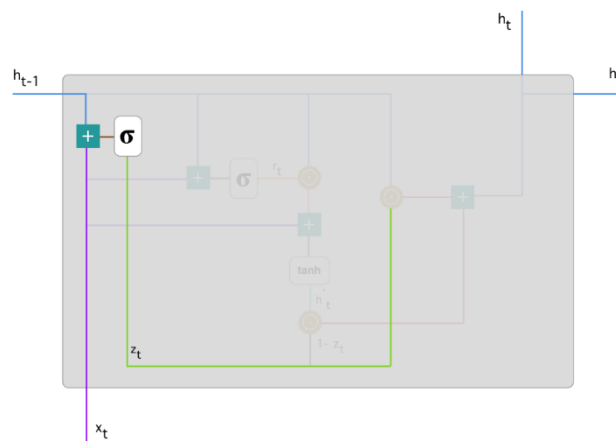
Gambar 2. 11 Perulangan dengan empat layer pada GRU (Simeon Kostadinov, 2017)

GRU memiliki 2 gate yaitu *update gate* dan *reset gate*. *update gate* bertindak mirip dengan *forget* dan *input gate* LSTM. *update gate* memutuskan informasi apa yang harus dibuang dan informasi baru apa yang akan ditambahkan. *Reset gate* adalah *gate* yang digunakan untuk memutuskan berapa banyak informasi masa lalu yang dilupakan (Michael Nguyen, 2018). Terdapat beberapa tahap dalam

proses GRU. GRU dimulai menghitung update gate ( $z_t$ ) pada waktu ke  $t$  menggunakan rumus:

$$z_t = \sigma(W^z \cdot X_t + U^z \cdot h_{t-1}) \quad (10)$$

Ketika nilai *input* pada orde ke  $t$  ( $x_t$ ) dimasukkan ke unit jaringan, ia dikalikan dengan bobotnya sendiri ( $W^z$ ). Hal yang sama berlaku untuk nilai output sebelum orde ke  $t$  ( $h_{(t-1)}$ ) yang menyimpan informasi untuk unit  $t-1$  sebelumnya dan dikalikan dengan bobotnya sendiri ( $U^z$ ). Kedua hasil ditambahkan bersama-sama dan fungsi aktivasi sigmoid diterapkan untuk mendapatkan hasil antara 0 dan 1. Gambar 2.13 adalah gambaran dari skema proses *update gate*.

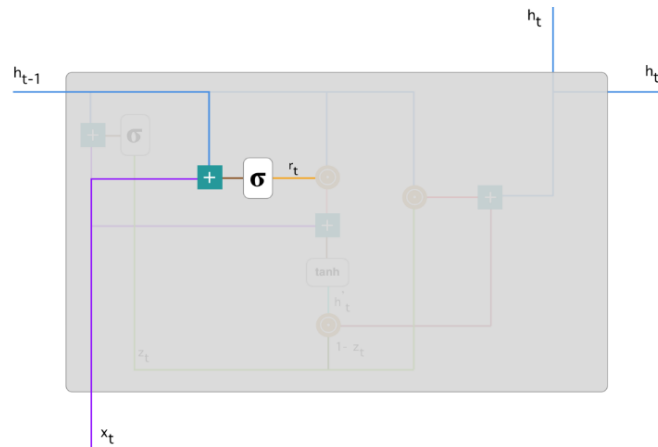


Gambar 2. 12 tahap pertama GRU (*Update Gate*) (Simeon Kostadinov, 2017)

Tahap selanjutnya adalah pada *reset gate*. Pada dasarnya gate ini digunakan dari model untuk memutuskan berapa banyak informasi masa lalu yang dilupakan. Untuk menghitungnya, menggunakan:

$$r_t = \sigma(W^r \cdot X_t + U^r \cdot h_{t-1}) \quad (11)$$

Rumus ini sama dengan rumus untuk *update gate*. Perbedaannya terletak pada bobot dan penggunaan *gate*. Skema di bawah ini menunjukkan di mana *reset gate* berada.

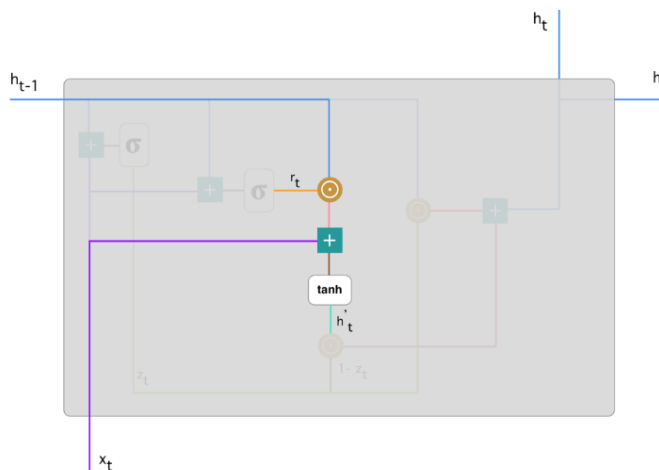


Gambar 2. 13 tahap kedua pada GRU (Reset Gate) (Simeon Kostadinov, 2017)

Selanjutnya adalah *current memory content*, dimulai dengan penggunaan *reset gate*. Lalu *current memory content* yang akan menggunakan *reset gate* untuk menyimpan informasi yang relevan dari masa lalu. Persamaannya diurai sebagai berikut:

$$\hat{h}_t = \tanh(W \cdot x_t + r_t \odot U h_{t-1}) \quad (12)$$

Proses ini bertujuan untuk menentukan apa yang harus dihapus dari langkah waktu sebelumnya. Dalam hal ini ketika jaringan saraf mendekati akhir dari inputan, ia akan belajar untuk menetapkan vektor  $r_t$  apakah mendekati 0 atau 1. Gambaran pada proses ini dapat dilihat pada gambar 2.14

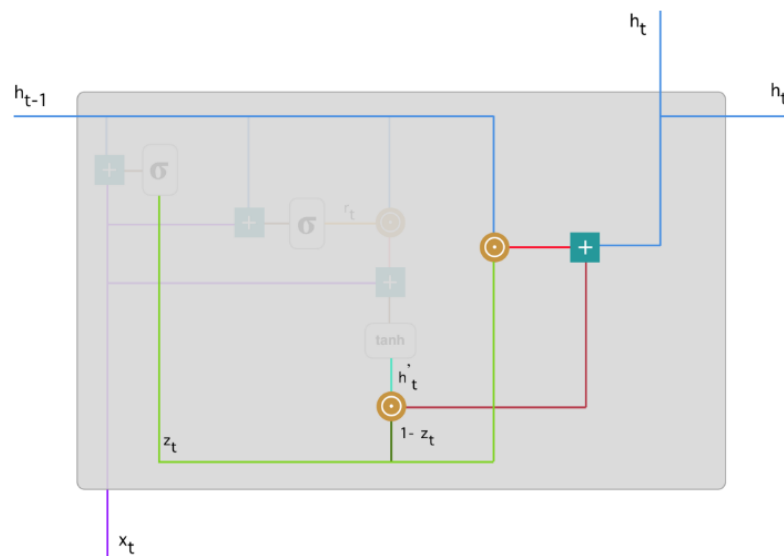


Gambar 2. 14 tahapan Konten memori saat ini (Simeon Kostadinov, 2017)

Sebagai langkah terakhir, jaringan perlu menghitung  $h_t$  yang menyimpan informasi untuk unit saat ini dan meneruskannya ke jaringan. Untuk melakukan itu dibutuhkan *update gate*. Ini menentukan apa yang diambil dari konten memori saat ini dan dari langkah sebelumnya. Hal itu dirumuskan sebagai berikut:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t \quad (13)$$

proses ini akan menjumlahkan konten saat ini dan informasi yang disimpan oleh *update gate* sebelumnya, skema dari proses ini dapat dilihat pada gambar 2.15.



Gambar 2. 15 proses terakhir dari GRU(Simeon Kostadinov, 2017)

### 2.1.5 Praproses

Praproses diimplementasikan untuk menghindari data yang tidak lengkap, gangguan data, dan data yang tidak konsisten. Tahapan praproses dalam penelitian ini meliputi:

1. Menghapus URL (<http://www.situs.com>) dan email ([nama@situs.com](mailto:nama@situs.com)).
2. Menghapus Karakter Twitter Khusus, Proses ini dilakukan dengan menghapus karakter Twitter khusus seperti hashtag, nama pengguna (@namapengguna), dan karakter khusus (misalnya RT, yang menunjukkan bahwa pengguna meretweet sesuatu).

3. Menghapus Simbol. Langkah ini dilakukan untuk menghapus simbol dan tanda baca di tweet.
4. Menghapus *Stopwords*. *Stopwords* adalah kata-kata yang tidak mempengaruhi proses klasifikasi.
5. *Stemming*, Tujuan dari *stemming* adalah untuk mendapatkan kata dasar dari sebuah kata.
6. Tokenisasi, Proses tokenisasi adalah proses pemotongan *string input* berdasarkan tiap kata yang menyusunnya.

#### **A. Stemming**

Tujuan dari stemming adalah untuk mendapatkan kata dasar dari sebuah kata. Karena kata yang berasal dari semantik mirip dengan kata dasarnya, Proses Algoritma stemming seringkali digunakan sesuai dengan bahasa yang akan diteliti (Uysal & Gunal, 2014) Stemming bertujuan mengurangi variasi kata yang memiliki kata dasar sama. Proses stemming dalam Bahasa Indonesia dilakukan dengan menggunakan bantuan Kamus Besar Bahasa Indonesia (KBBI) (Yusuf Nur dan Diaz Santika, 2011).

Algoritma Nazief dan Adriani adalah algoritma stemming Bahasa Indonesia, Algoritma yang dibuat oleh (Adriani, Asian, Nazief, Tahaghoghi, & Williams, 2007) memiliki tahap – tahap sebagai berikut (Arif, 2016):

1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tersebut adalah *root word*.
2. Inflection Suffixes (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa particles (“-lah”, “-kah”, “-tah”, atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Passive Pronouns* (-ku, -mu, atau -nya) jika ada.
3. *Hapus Derivation Suffixes* (“-i”, “-an”, atau “-kan”). Jika kata ditemukan dalam kamus, maka algoritma berhenti. Jika tidak maka lanjut ke langkah 3a:
  - a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lanjut ke langkah 3b.
  - b. Akhiran yang dihapus (“-i”, “-an”, atau “-kan”) dikembalikan, lanjut ke langkah 4.

4. Hapus *Derivation Prefix*. Jika pada langkah 3 ada *sufiks* yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
  - a. Periksa tabel kombinasi awalan dan akhiran yang tidak diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.
  - b. For  $i=1$  to 3, tentukan tipe awalan kemudian hapus awalan. Jika root word belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Jika awalan kedua sama dengan awalan pertama algoritma berhenti.
5. Melakukan *recording*
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.  
Tipe awalan ditentukan melalui langkah – langkah sebagai berikut:
  1. Jika awalannya (“di-”, “ke-”, atau “se-”) maka tipe awalannya secara berturut – turut adalah (“di-”, “-ke”, atau “se-”).
  2. Jika awalannya adalah (“te-”, “me-”, “be-”, atau “pe-”) maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
  3. Jika dua karakter pertama bukan (“di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, atau “pe-”) maka berhenti.
  4. Jika tipe awalan adalah “none” maka berhenti. Jika tipe awalan adalah bukan “none” maka awalan dapat dilihat pada tabel 2.2. Hapus awalan jika ditemukan.

**Tabel 2. 1** Kombinasi Awalan Akhiran Yang Tidak Diiijinkan

| Awalan | Akhiran yang tidak diijinkan |
|--------|------------------------------|
| Be-    | -i                           |
| Di-    | -an                          |
| Ke-    | -i, -kan                     |
| Me-    | -an                          |
| Se-    | -i, -kan                     |

**Tabel 2. 2** Cara Menentukan Tipe Awalan Kata Diawali Dengan “te-”

| Following Characters |       |       |       | Tipe Awalan |
|----------------------|-------|-------|-------|-------------|
| Set 1                | Set 2 | Set 3 | Set 4 |             |
| “-r-”                | “-r-” | -     | -     | none        |
| “-r-”                | Vowel | -     | -     | ter-luluh   |

|                      |                      |           |           |      |
|----------------------|----------------------|-----------|-----------|------|
| “-r-”                | not (vowel or “-r-”) | “-er”     | vowel     | ter- |
| “-r-”                | not (vowel or “-r-”) | “-er”     | not vowel | ter- |
| “-r-”                | not (vowel or “-r-”) | not “-er” | -         | ter- |
| not (vowel or “-r-”) | “-er”                | vowel     | -         | none |
| not (vowel or “-r-”) | “-er”                | not vowel | -         | te   |

**Tabel 2. 3 Jenis Awalan Berdasarkan Tipe Awalannya**

| Tipe Awalan | Awalan Yang Harus Dihapus |
|-------------|---------------------------|
| di-         | di-                       |
| ke-         | ke-                       |
| se-         | se-                       |
| te-         | te-                       |
| ter-        | ter-                      |
| ter-luluh   | ter-                      |

Untuk mengatasi keterbatasan pada algoritma di atas, maka ditambahkan aturan aturan di bawah ini:

1. Aturan duplikasi.
  - a. Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka *root word* adalah bentuk tunggalnya, contoh: “buku – buku” *root word*-nya adalah “buku”.
  - b. Kata lain, misalnya “bolak – balik”, “berbalas – balasan”, dan “seolah – olah”. Untuk mendapatkan *root word*-nya, kedua kata diartikan secara terpisah. Jika keduanya memiliki *root word* yang sama maka diubah menjadi bentuk tunggal, contoh: kata “berbalas – balasan”, “berbalas” dan “balasan” memiliki *root word* yang sama yaitu “balas”, maka *root word* “berbalas – balasan” adalah “balas”. Sebaliknya, pada kata “bolak – balik”, “bolak” dan “balik” memiliki *root word* yang berbeda, maka *root word*-nya adalah “bolak – balik”.

Tambahan bentuk awalan dan akhiran serta aturannya.

- a. Untuk tipe awalan “mem-”, kata yang diawali dengan awalan “memp-” memiliki tipe awalan “mem-”.



- b. Tipe awalan “meng-”, kata yang diawali dengan awalan “mengk-” memiliki tipe awalan “meng-”.

Pada tabel 2.4 akan dijelaskan contoh – contoh aturan yang terdapat pada awalan sebagai pembentuk kata dasar. Simbol C merupakan huruf konsonan, simbol V merupakan huruf vokal, simbol A merupakan vokal atau konsonan, dan simbol P merupakan partikel suatu kata, contohnya “er”.

**Tabel 2. 4 Aturan Awalan Pembentuk Kata Dasar**

| <b>Format Kata</b> | <b>Pemenggalan</b>        | <b>Aturan</b>                          |
|--------------------|---------------------------|--|
| berV               | ber-V                     |  |
| berCAP             | ber-CAP                   | dimana C!="r" & P!="er"                |
| berCAerV           | ber-CaerV                 | dimana C!="r"                          |
| belajar            | bel-ajar                  |  |
| beCR1RerCR2R       | beCR1RerCR2R              | dimana CR1R!={"r"} "l" }               |
| terV               | ter-V   te-rV             |  |
| terCerV            | Ter-CerV                  | dimana C!="-r"                         |
| terCP              | ter-CP                    | dimana C!="r" dan P!="er"              |
| terCR1RerCR2R      | te-CR1RerCR2R             | dimana CR1R!="r"                       |
| Me{l r w y}V       | me- {l r w y}V            |  |
| Mem{b f v}         | mem- {b f v}              |  |
| Mempe{r l}         | mem-pe                    |  |
| Mem{rV V}          | me-m{rV v} me-<br>p{rV V} |  |
| Men{c d j z}       | Men- {c d j z}            |  |
| menV               | me-nV me-tV               |  |
| Meng{g h q}        | meng- {g h q}             |  |
| mengV              | meng-V meng-kV            |  |
| menyV              | meny-sV                   |  |
| mempV              | mempV                     | dimana V!="e"                          |
| Pe{w y}V           | pe- {w y V}               |  |
| perV               | per-V pe-rV               |  |
| perCAP             | per-CAP                   | dimana C!="r" dan P!="er"              |
| perCAerV           | per-CAerV                 | dimana C!="r" dan P!="er"              |
| Pem{b f V}         | pem- {b f V}              |  |
| Pem{rV V}          | pe-m{rV V} pe-<br>p{rV V} |  |
| Pen {c d j z}      | pen- {c d j z}            |  |
| penV               | pe-nV pe-IV               |  |
| Peng{g h q}        | peng- {g h q}             |  |
| pengV              | Peng -V peng-kV           |  |
| penyV              | peny-sV                   |  |
| pelV               | pe-IV                     | kecuali "pelajar"<br>menghasilkan ajar |
| perCerV            | pe-erV                    | dimana C!={r w y l m n}                |

|       |       |  |
|-------|-------|--|
| perCP | perCP | dimana $C! = \{r w y l m n\}$<br>dan $P! = "er"$ |
|-------|-------|--|

## B. Tokenisasi

Proses tokenisasi adalah proses pemotongan string input berdasarkan tiap kata yang menyusunnya. *Token* yang dipakai dalam penelitian ini ini menggunakan. Penelitian ini mengimplementasikan tokenisasi N-Gram yang tidak terikat dengan satu aturan bahasa apapun, Tokenisasi menggunakan N-Gram adalah tahap pemrosesan dimana teks input dibagi menjadi unit-unit kecil yang disebut token sepanjang n karakter . Dalam bahasa Indonesia, frasa dengan satu kesatuan arti memiliki maksimal 3 kata, pembagian token dibagi menjadi Unigram, Bigram, Trigram dan N-Gram (Saputra, Bharata Adji, & Erna Permanasari, 2015), berikut contoh pemecahan pada kalimat “Hate Speeh media social”.

Unigram: yaitu token yang terdiri dari hanya satu kata, menghasilkan: “hate”, “speech”, “media”, “social”.

Bigram: yaitu token yang terdiri dari dua kata, menghasilkan: “hate speech”, “speech media”, “media sosial”.

Trigram: yaitu token yang terdiri dari tiga kata, menghasilkan: “hate speech media”, “speech media social”.

N-Gram, yaitu membagi menjadi Unigram, Bigram, dan Trigram, menghasilkan : “hate”, “hate speech”, “hate speech media”

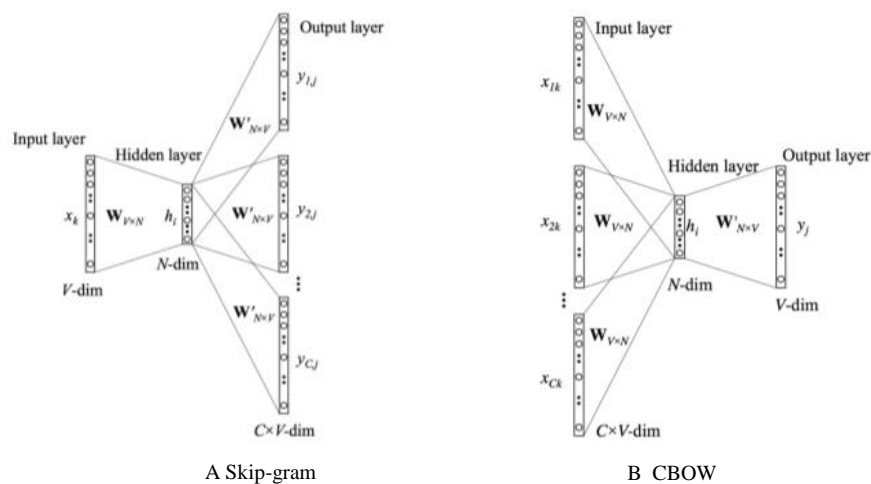
### 2.1.6 Word Embedding

*Word embedding* bekerja dengan cara melatih satu *set vector* yang mempunyai panjang tetap secara terus menerus. Secara visual, *word embedding* dapat digambarkan bahwa setiap kata diwakili oleh sebuah titik di dalam luasan bidang tertentu, titik-titik ini kemudian akan dipelajari oleh perhitungan *word embedding* dan satu titik akan dipindahkan menjauh atau mendekati titik yang lainnya, berdasarkan kata-kata lain yang mengelilingi titik tersebut. Hal ini akan dilakukan terus menerus hingga sampai pada sebuah kondisi dimana semua titik tidak dapat dipindahkan lagi mendekati (atau menjauhi) titik lainnya. Sehingga hasil akhir dari iterasi ini dapat memberikan sebuah gambaran dimana kata-kata dengan makna yang serupa akan cenderung berada dalam satu area yang sama

dalam bidang tersebut atau dengan kata lain katakata yang ada dalam satu area pada bidang tersebut dan mempunyai jarak kedekatan yang kecil cenderung mempunyai kesamaan *semantic* (Prabowo, Marselino, & Suryawiguna, 2019).

### A. Word2Vec

*Word2Vec* dikembangkan oleh Mikolov 2012. *Word2Vec* merupakan sebuah algoritma untuk mempelajari posisi kedekatan semantik antar kata dari sebuah teks masukan, Cara kerja alat ini yaitu dengan mengambil korpus *teks* sebagai *input*, lalu menghasilkan representasi vektor setiap kata yang ada pada korpus teks tersebut sebagai output . *File* vektor yang dihasilkan dapat digunakan untuk penelitian pada pemrosesan bahasa alami dan aplikasi *machine learning*. Vektor kata tersebut juga dapat digunakan untuk mengukur jarak kedekatan antar vektor kata yang lain (Prabowo et al., 2019). metode word2vec ini terdiri dari dua buah algoritma utama *word embedding* didalamnya, yaitu: Continuous-Bag-Of-Word (CBOW) dan skip-gram. Algoritma CBOW digunakan untuk melihat panjang tertentu dari sebuah kata pada dokumen masukan. Sedangkan algoritma skip-gram digunakan untuk memprediksi konteks kata dengan cara melihat kedekatan sebuah kata dengan kata lain yang posisinya sebelum atau sesudah kata tersebut.



Gambar 2. 16 Arsitektur (A) Skip-gram dan (b) CBOW (Meyer, 2016)

Arsitektur Skip-gram dan CBOW bekerja dengan tiga *layer*, *layer* yang pertama yaitu *layer input*, *layer* yang kedua yaitu *hidden layer*, dan *layer* yang ketiga yaitu *output layer*. Di antara *input layer* dengan *hidden layer* terdapat matriks *weight* (W) yang didapatkan dari nilai random, *layer* ini berfungsi untuk mengaktifkan *hidden layer*. *Output layer* dihasilkan dari perhitungan *hidden layer* dengan matriks *weight* (W'). Setelah itu menggunakan *Softmax function* pada setiap *neuron* yang dihasilkan oleh *layer output* untuk mengubah total jumlah *output* menjadi bernilai antara 0 - 1, dan jumlah dari semua nilai *output* akan bertambah sehingga nilainya 1 (Widyastuti, Bijaksana, & Sardi, 2018). Persamaan 1 *Softmax function* adalah sebagai berikut :

$$\text{SoftmaxFunction} = \frac{e^x}{\sum e^x} \quad (13)$$

Dimana x adalah nilai *ouput* yang dihasilkan oleh pemodelan Skip-gram dan CBOW

### **B. Embeddings from Language Models (ELMo)**

Embedding from Language Models (ELMo) yang dikembangkan oleh (Peters et al., 2018). ELMo menggunakan representasi dari bi-directional LSTM yang dilatih dengan tujuan *language model* (LM) pada dataset teks yang besar (Perone, Silveira, & Paula, 2018). Representasi ELMo adalah fungsi dari lapisan internal bi-directional Language Model (biLM), ELMo menghasilkan multi word embeddings per kata untuk skenario yang berbeda. Lapisan yang lebih tinggi menangkap aspek tergantung konteks dari word embeddings sementara lapisan yang lebih rendah menangkap aspek model sintaksis (Gardner et al., 2018). Berikut adalah persamaan dari ELMo (Peters et al., 2018):

$$R_k = \left\{ x_k^{LM}, \vec{h}_{kj}^{LM}, \overleftarrow{h}_{kj}^{LM} \mid j = 1, \dots, L \right\} \quad (14)$$

$$= \left\{ x h_{kj}^{LM} \mid j = 1, \dots, L \right\}$$

Dimana

x = token(word embedding)

k = posisi dari token

j = #j *layer* dari biLSTM

h = mengarahkan ke biLSTM layer( yang pertama adalah *forward layer* dan yang kedua adalah *backward layer* )

### 2.1.7 Menentukan Jumlah Hidden Layer dan Jumlah Neuron

Dalam Deep learning diperlukan menentukan jumlah Node dan neuron. Ada banyak metode aturan praktis untuk menentukan jumlah neuron yang dapat diterima untuk digunakan dalam lapisan tersembunyi (Jeff, 2015), seperti berikut:

- a. Jumlah neuron tersembunyi harus antara ukuran lapisan input dan ukuran lapisan output.
- b. Jumlah neuron tersembunyi harus  $2/3$  ukuran lapisan input, ditambah ukuran lapisan output.
- c. Jumlah neuron yang tersembunyi harus kurang dari dua kali ukuran lapisan input.

Sedangkan dalam menentukan jumlah node Tabel berikut merangkum kemampuan beberapa arsitektur lapisan umum:

| Jumlah Hidden Layer | Hasil   |
|---------------------|---|
| None                | Hanya mampu mewakili fungsi atau keputusan linier yang dapat dipisah.   |
| 1                   | Dapat memperkirakan fungsi apa pun yang berisi pemetaan berkelanjutan dari satu ruang terbatas ke ruang lainnya.  |
| 2                   | Dapat mewakili batas keputusan arbitrer untuk akurasi sewenang-wenang dengan fungsi aktivasi rasional dan dapat memperkirakan pemetaan yang mulus untuk akurasi apa pun |
| >2                  | Lapisan tambahan dapat mempelajari representasi kompleks (semacam rekayasa fitur otomatis) untuk lapisan lapisan.   |

### 2.1.8 Part-of-speech Tagging

*Part-of-speech Tagging* atau yang sering disebut Tagging atau POS Tagging, merupakan proses pemberian atau penentuan sebuah part-of-speech tag terhadap suatu kata dalam suatu kalimat. Sedangkan part-of-speech merupakan kategori kata ditinjau dari sudut pandang kebahasaan (gramatikal), seperti kata benda, kata kerja, kata keterangan, kata sifat dan sebagainya. POS Tagging merupakan komponen proses yang sangat penting dan sering digunakan dalam rangkaian pemrosesan Bahasa alamiah (Natural Language Processing)(Rozi, 2013)

### 2.1.9 Precision, Recall, dan F-Measure

*Precision*, *Recall*, dan *F-Measure* merupakan parameter yang digunakan dalam pengujian hasil klasifikasi. *Precision* adalah proporsi kasus Prediksi Positif yang benar-benar Positif Sedangkan *Recall* adalah proporsi kasus *Real Positive* yang diprediksi dengan Positif dengan benar (Powers, 2011). Secara umum, *precision* dan *recall* dapat dirumuskan sesuai dengan persamaan 15 dan 16 dan detail pembahasan pada gambar 2.17.

|           |          | Actual   |          |
|-----------|----------|----------|----------|
|           |          | Positive | Negative |
| Predicted | Positive | TP       | FP       |
|           | Negative | FN       | TN       |

Gambar 2. 17 Confussion Matrix

$$Precision = \frac{TP}{TP + FP} \quad (18)$$

$$Recall = \frac{TP}{TP + FN} \quad (19)$$

Dari dua persamaan diatas, yang dimaksud dengan TP, TN, FP, dan FN adalah:

- *True Positives* (TP), data yang diklasifikasikan oleh model sebagai positif dan label sesungguhnya adalah memang positif (prediksi benar).
- *True Negatives* (TN), data yang diklasifikasikan oleh model sebagai negatif dan label sesungguhnya adalah memang negatif (prediksi benar).
- *False Positives* (FP), data yang diklasifikasikan oleh model sebagai positif dan label sesungguhnya adalah negatif (prediksi salah).
- *False Negative* (FN), data yang diklasifikasikan oleh model sebagai negatif dan label sesungguhnya adalah positif (prediksi salah).

Untuk mengoptimalkan hasil yang didapatkan, diperlukan parameter pengukuran lainnya yaitu *F-measure*. *F-measure* secara efektif mereferensikan *True Positive* ke rata-rata dari Prediksi Positif dan *Real Positive*. *F-measure* dapat

dirumuskan sesuai dengan persamaan 17. Semakin besar nilai *F-measure*, maka semakin baik pula hasil dari klasifikasi tersebut.

$$F - measure = 2x \frac{Precision \times Recall}{Precision + Recall} \quad (10)$$

## 2.2 Literatur Review

Berdasarkan Literatur review yang dilakukan terhadap 30 artikel yang dipublikasi antara tahun 2015 sampai 2019 dimana dari paper yang digunakan untuk *literature review* 20 paper diterbitkan pada tahun 2018. Selain itu, kami mengamati bahwa 23 dari penelitian diterbitkan dalam conference, sedangkan 7 sisanya di jurnal internasional. untuk lebih jelas dapat lihat Lampiran 1. Dari data ini dapat disimpulkan bahwa paper yang diterbitkan sejauh ini sebagian besar melaporkan landasan awal tentang bagaimana teknik klasifikasi dapat dikembangkan untuk hate speech.

**Tabel 2. 5 metode klasifikasi yang digunakan dalam deteksi hate speech**

| classification      | paper   |
|---------------------|---|
| Random Forest       | 3,7,8,10,11,13,14,21,24,27                        |
| J48                 | 5,7   |
| GBDTs               | 16,29   |
| Naïve Bayes         | 5,6,8,12,15,23,27                                 |
| Logistic regression | 3,5,13,14,17,20,24,27,28,29                       |
| SVM                 | 1 ,2,5,7,8,10,12,13,14,16,17,18,20,24,26,27,29,30 |
| RNN                 | 2,16,22   |
| CNNs                | 2,11,22,29  |
| GRU                 | 18,19   |
| LSTM                | 2,4,9,18,25,28                                    |

*Support Vector Machines* (SVM) digunakan oleh 18 paper. Berdasarkan penelitian sebelumnya klasifikasi dengan SVM dapat mencapai kinerja yang sangat tinggi (Martins, Gomes et al., 2018). Random Forest Decision Tree Digunakan dalam 10 Paper. Selain itu, 6 penelitian menggunakan Naive Bayes. Yang menyebabkan peneliti menggunakan beberapa algoritma diatas adalah performa yang dihasilkan sangat baik apabila dibandingkan dengan beberapa algoritma yang lain. Algoritma klasifikasi lain adalah Gradient Boosted Decision Trees (GBDTs) dan J48 digunakan untuk dalam 2 penelitian, apabila dibandingkan performa Gradient Boosted Decision Trees (GBDTs) dan J48 masih dibawah tiga metode

sebelumnya. Sedangkan untuk *Neural Network*, metode yang cukup terkenal adalah *Recurrent Neural Network* (RNN) dengan 3 penelitian dan *Convolutional Neural Networks* (CNNs) dengan 4 penelitian.

Apabila dibanding dengan dua *Neural Network* sebelumnya metode Long Short-Term Memory (LSTM) digunakan oleh 6 studi; LSTM adalah pengembangan algoritma RNN. Algoritma ini cocok untuk penelitian yang berhubungan dengan word embedding (Sutejo & Lestari, 2018). Gated Recurrent Unit (GRU) digunakan dalam 2 studi, yang menyebabkan penelitian menggunakan GRU lebih rendah dibandingkan dengan metode lain adalah metode ini masih baru dimana metode ini dikembangkan oleh Cho et al. (2014)..

SVM adalah algoritma yang paling banyak digunakan untuk deteksi *hate speech* karena SVM lebih baik berurusan dengan fitur numerik (Alfina et al., 2017). SVM cenderung berkinerja secara signifikan lebih baik daripada model machine learning lain (Burnap & Williams, 2015; Davidson, Warmsley, Macy, & Weber, 2017; Sahi, Kilic, & Saglam, 2018; Salminen et al., 2018; Zampieri et al., 2019). Tetapi kami menemukan bahwa kinerja *Neural Network* lebih baik dari pada SVM (Badjatiya et al., 2017; Gröndahlet et al., 2018; Mishra, Del Tredici et al., 2019; Pitsilis et al., 2018; Santosh et al., 2019; Zampieri et al., 2019; Zhang et al., 2018).

**Tabel 2. 6 metode word embedding yang digunakan dalam deteksi hate speech**

| Word Embedding | paper   |
|----------------|---|
| BOW            | 2,4,6,7,9,10,11,12,13,16,17,18,20,21,22,23,24,27,28,29,30 |
| CBOW           | 4,8,25,30   |
| skip-gram      | 2,4,15,25,26  |
| GloVe          | 16  |
| FastText       | 2,3,25,29   |

Bags-Of-Word (BOW) digunakan dalam 21 studi. BOW digunakan karena selain mudah, metode ini adalah dasar dari *word embedding* lainnya. Metode antara CBOW, Skip Gram dan GLoVe memiliki beberapa kesamaan. Tetapi peneliti menggunakan studi CBOW dan Skip gram berjumlah 5 dan FastText digunakan oleh 3 studi sedangkan GloVe digunakan dalam 1 penelitian. *Word embedding* memiliki efek signifikan dalam pendeteksian *Hate speech*, seperti yang dapat kita

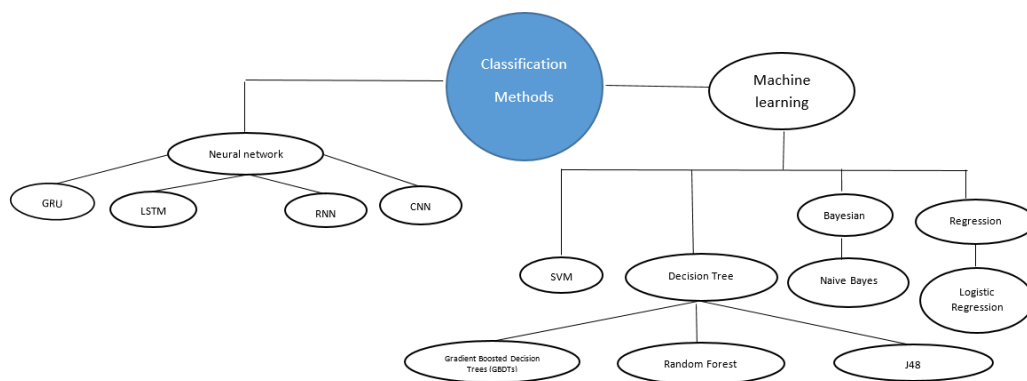


lihat dalam penelitian Badjatiya et al. ( 2017) ketika mereka menggunakan *word embedding* performa yang dihasilkan lebih baik dalam hal precision, recall and F1-scores.

**Tabel 2. 7 metode tokenisasi yang digunakan dalam deteksi hate speech**

| Tokenization | paper  |
|--------------|--|
| Unigram      | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 |
| Bigram       | 1,2,3,4,5,8,9,10,12,13,14,15,16,20,21,22,24,27,28,29,30                          |
| Trigram      | 1,2,3,4,5,8,9,10,12,13,14,15,16,20,21,22,24,27,28,29,30                          |
| n-gram       | 10,12,13,15,16,20,21,22,29,30  |

Dalam proses tokenisasi, unigram digunakan dalam semua penelitian, bigram digunakan oleh 22 studi, trigram digunakan oleh 22 studi, dan n-gram digunakan oleh 17 studi. Fitur Tokenisasi memiliki efek yang sangat signifikan dalam deteksi hate speech dan jika paper menggunakan bigram otomatis menggunakan trigram. Unigram cenderung berkinerja lebih baik daripada fitur lainnya (Alfina et al., 2017), dalam penelitian ini juga menunjukkan bahwa menggabungkan fitur unigram dengan fitur lain menghasilkan performa yang lebih baik (Alfina et al., 2017; Pitsilis et al., 2018; Sutejo & Lestari, 2018; Watanabe et al., 2018).



**Gambar 2. 18 main map metode klasifikasi pada hate speech**

### 2.3 Penelitian Terkait

Pada bagian ini menjelaskan tentang referensi penelitian yang berkaitan dengan tugas akhir. Pada bagian ini memaparkan acuan penelitian sebelumnya yang digunakan oleh penulis dalam melakukan penelitiannya.

### 2.3.1 Hate Speech Indonesia

Penelitian yang dilakukan oleh Fatahillah et al. (2017) dilakukan dengan menggunakan dataset dari twitter dan menggunakan metode naïve bayes, pada penelitian ini peneliti tidak menggunakan metode *word embedding*. Proses pelatihan dimulai dari membaca data dari database kemudian melakukan proses *stopword*. Selanjutnya melakukan proses *checkwordIndoc*, *checkwordIndoc* adalah proses pengelompokan kata-kata positif dan kata-kata negatif. Prosedur *checkwordInDoc* adalah proses memeriksa apakah kalimat pada pelatihan data positif atau negatif. Proses perhitungan probabilitas klasifikasi tweet menggunakan metode klasifikasi naïve bayes. Proses klasifikasi memiliki tiga proses yaitu perhitungan sebelumnya, kemungkinan, dan probabilitas. Proses perhitungan sebelumnya adalah perhitungan tingkat kemunculan label pada data pelatihan sebelumnya. Proses perhitungan kemungkinan adalah perhitungan kemungkinan kata muncul pada label tertentu. Proses perhitungan probabilitas adalah perhitungan yang mengukur kemunculan label pada suatu kalimat.

Dalam penelitian Alfina et al. (2017), mereka mengadopsi beberapa pendekatan NB, BLR, RFDT, dan SVM. peneliti memilih topik umum sebagai dataset. peneliti menggunakan cara menganotasi dataset yang melibatkan *annotator* dari latar belakang yang berbeda untuk meningkatkan objektivitas. Sebagai fitur peneliti menggunakan unigram, bigram, trigram karakter (karakter 3-gram) dan quadragram karakter (karakter 4-gram), peneliti membandingkan kinerja empat algoritma klasifikasi.

Penelitian yang dilakukan oleh Pratiwi et al. (2018) menggunakan FastText sebagai pengklasifikasi dan menggunakan Instagram sebagai sumber data. Kontribusi penelitian ada dua: membangun *dataset* hate speech baru dari komentar Instagram dan mengevaluasi pemanfaatan FastText sebagai pengklasifikasi. Beberapa fitur yang digunakan seperti unigram, bigram, dan *word n-gram*. Karakter *n-gram* juga dapat digunakan sebagai fitur untuk mengurangi variasi kata yang tidak lengkap. Fitur *word embedding* juga digunakan untuk deteksi *hate speech*. menggunakan *Continous Bag-Of-Words* (CBOW) dan *embedding paragraf2vec*. peneliti menggunakan pendekatan FastText sebagai *classifier* untuk mendeteksi

*hate speech* dalam bahasa Indonesia. FastText dibandingkan dengan dua algoritma lain yaitu *Random Forest Decision Tree* dan *Logistic Regression*. Proporsi *dataset* yang digunakan untuk data pelatihan adalah 80% dan untuk *dataset* pengujian adalah 20% sisanya.

Penelitian yang dilakukan oleh Sutejo & Lestari (2018). Dalam penelitian ini mereka membuat *dataset* baru dari beberapa media sosial seperti Facebook, Twitter, dan LineToday untuk melatih model *deep learning* untuk mendeteksi *hate speech* dalam Bahasa Indonesia. Mereka mendeteksi *hate speech* untuk bahasa Indonesia menggunakan fitur *word embedding* Continuous Bag-of-Words (CBOW) and skipgram, dan melatih menggunakan LSTM untuk mendeteksi *hate speech*.

### 2.3.2 LSTM

Pitsilis et al. (2018) melakukan penelitian dengan menggunakan pendekatan LSTM, peneliti pertama-tama menguraikan detail fitur yang diturunkan untuk menggambarkan kecenderungan setiap pengguna terhadap setiap kelas (Netral, Rasisme, atau Seksisme), sebagaimana ditangkap dalam riwayat tweet mereka. Secara total, peneliti mendefinisikan tiga fitur yang mewakili kecenderungan pengguna untuk memposting konten Netral, Rasis dan Seksis, setelah itu dilakukan proses klasifikasi menggunakan. dalam melakukan evaluasi peneliti terhadap skema yang diusulkan, setiap classifier diimplementasikan sebagai model *deep learning* yang memiliki empat lapisan, dan dijelaskan sebagai berikut: dimulai dengan *Input Layer (word Embedding)*. Pada penelitian ini LSTM melakukan iterasi sebanyak 200. Output dari LSTM dijalankan melalui *dense layer* untuk meningkatkan pembelajaran dan mendapatkan output yang lebih stabil. Ukurannya *output layer* sama dengan ukuran *input layer*. *Output layer* memiliki tiga kelas yaitu Netral, Rasisme, dan Seksisme.

Badjatiya, Gupta, Gupta, & Varma (2017) melakukan penelitian menggunakan LSTM yang dilatih menggunakan data yang sudah dilabel dengan *back-propagation*. Setelah jaringan dilatih, tweet baru diuji terhadap jaringan yang mengklasifikasikannya sebagai rasis, seksis atau tidak. metode ini juga

menggunakan GLoVe dan *random embedding* sebagai *word embedding* yang disesuaikan dengan label *hate speech*.

### 2.3.3 GRU

Penelitian ini dilakukan oleh Zhang, Robinson, & Tepper (2018), mereka menggunakan fitur *word embedding* skip-gram dan metode *deep learning* GRU, Lapisan pertama adalah *word embedding* layer, yang memetakan setiap kata ke dalam vektor. Fitur yang diekstraksi ini kemudian dimasukkan ke dalam lapisan GRU, yang memperlakukan dimensi fitur sebagai *timesteps* dan menghasilkan 100 hidden unit per timestep.

**Tabel 2.1 Rangkuman penelitian terkait**

| No | Penelitian                              | fitur   | Metode  | dataset                                |
|----|---|---|---|--|
| 1  | Fatahillah et al. (2017)                | -   | naïve bayes   | twitter                                |
| 2  | Alfina et al. (2017)                    | bag of words (BOW), word n-grams , char n-grams   | Naïve Bayes, Support Vector Machine, Bayesian Logistic Regression, and Random Forest Decision | twitter                                |
| 3  | Pratiwi et al. (2018)                   | Skip-gram , word n-grams , char n-grams   | FastText, Random Forest Decision Tree dan Logistic Regression                                 | Instagram                              |
| 4  | Sutejo & Lestari (2018)                 | word n-grams, bag of words (BOW), TF-IDF (Term Frequency-Inverse Document Frequency), CBOW (Continuous Bag-of-Words), Skip-gram | LSTM  | Facebook, Twitter, Line Today, YouTube |
| 5  | Badjatiya, Gupta, Gupta, & Varma (2017) | char n-gram, GloVe  | Logistic Regression, Random Forest, SVMs, Gradient Boosted Decision Trees                     | twitter                                |

|   |                        |                    |                                   |         |
|---|------------------------|--------------------|-----------------------------------|---------|
|   |                        |                    | (GBDTs), FastText, CNNs dan LSTMs |         |
| 6 | Pitsilis et al. (2018) | bag of words (BOW) | LSTM                              | twitter |
| 7 | Zhang et al., (2018)   | skip-gram          | CNN,GRU                           | twitter |

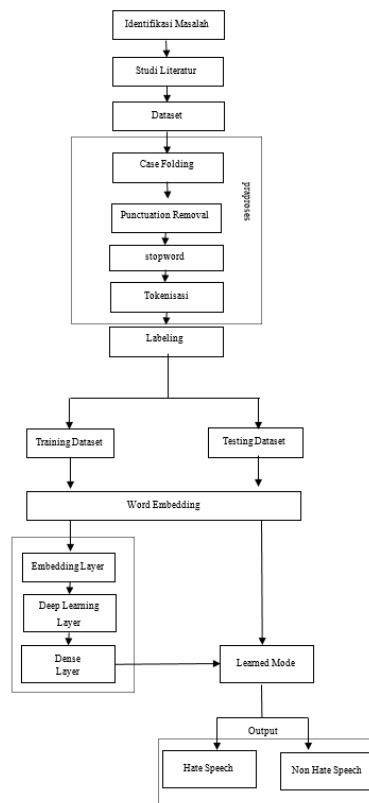
(Halaman sengaja dikosongkan)

## BAB 3

### METODOLOGI PENELITIAN

Bab ini berisi pembahasan tahapan penelitian dan rencana penelitian. Pada bab ini akan menjelaskan detail dari tahapan penelitian beserta jadwal kegiatan pengerjaan penelitian. Selain itu, bab ini digunakan sebagai panduan dalam penyusunan penelitian agar terarah dan sistematis.

#### 3.1 Tahapan Penelitian



**Gambar 3. 1 Tahapan Penelitian**

Tahapan penelitian utama dalam metodologi ini adalah (1) Identifikasi masalah; (2) Studi literatur; (3) Perancangan pembuatan model dimulai dari mengumpulkan dataset sampai mendapatkan learned model deep learning; (4) Pengujian; (5) Penyusunan laporan. Tahapan penelitian dapat dilihat pada gambar 3.1.

### 3.1.1 Identifikasi Masalah

Identifikasian masalah bertujuan untuk menemukan *research question* yang ingin diselesaikan pada penelitian ini. Identifikasi permasalahan permasalahan penelitian ini menggunakan pendekatan studi kasus telah dibahas pada bagian sebelumnya. Berdasarkan permasalahan yang telah teridentifikasi, tujuan yang ingin dicapai pada penelitian ini yaitu menghasilkan solusi dalam permasalahan mengidentifikasi hate speech di media sosial indonesia.

### 3.1.2 Studi Literatur

Studi literatur dilakukan dengan mempelajari literatur-literatur terkait berbagai konsep yang digunakan dalam beberapa penelitian sebelumnya. Studi literatur dalam penelitian ini bersumber dari buku, media, ataupun dari hasil penelitian orang lain. Tahap ini bertujuan untuk mempelajari konsep, metode dan teknik yang mampu diimplementasikan untuk memecahkan solusi terkait permasalahan utama dalam penelitian. Studi literatur yang digunakan pada penelitian ini adalah literatur yang berkaitan dengan *deep learning*, *word embedding* dan *hate speech*.

### 3.1.3 Perancangan Pembuatan Model

Perancangan pembuatan model dilakukan dengan mendesain alur model yang akan dibuat pada penelitian ini, yang dapat dilihat pada gambar 3.1. Perancangan pembuatan model ini terdiri dari beberapa tahapan utama yaitu : file berekstensi csv yang menyimpan data *tweet hate speech* dan *non hate speech* ditampilkan dalam bentuk *dataframe*. Selanjutnya dilakukan pra-proses yang dimana didalamnya dilakukan casefolding, Punctuation Removal, stopword, dan tokenisasi. Data yang sudah di pra-proses selanjutnya dipecah menjadi data latih dan data uji kemudian dataframe diubah formatnya dari *text* kedalam *2D tensor of integer*. Kemudian pelatihan model menggunakan algoritma *Deep learning* dengan pengujian model. Apabila tweet dianggap non hate speech, tweet selanjutnya dimasukan ke proses deteksi sarkasme untuk memastikan kalimat tersebut bukan hate speech.



### 3.1.4 Menyiapkan Data

Penelitian ini menggunakan data Twitter sebagai sumber dataset dan mengumpulkan tweet menggunakan Twitter *Streaming* API. Tweet dikumpulkan berdasarkan pencarian menggunakan kueri, kueri yang digunakan adalah yang berhubungan dengan pemilihan presiden seperti "jokowi" dan "prabowo". Setelah tweet terkumpul dilakukan labeling terhadap tweet dengan 0 apabila tweet *non hate speech* dan 1 untuk tweet *hate speech*.

### 3.1.5 Praproses Data

Tahap selanjutnya adalah pra-proses, setelah melakukan proses dilakukan ekstraksi fitur yang terdapat pada proses *labeling*, terdapat beberapa tahap yang harus dilakukan sebelum data teks dapat diklasifikasikan antara lain:

- 1 Menghapus URL (misalnya <http://www.situs.com>) dan email (misalnya nama@situs.com).
- 2 Menghapus Karakter Twitter khusus, Proses ini dilakukan dengan menghapus karakter Twitter khusus seperti *hashtag*, nama pengguna (misalnya @namapengguna), dan karakter khusus (misalnya RT, yang menunjukkan bahwa pengguna me-retweet sesuatu).
- 3 Menghapus Simbol. Langkah ini dilakukan untuk menghapus simbol dan tanda baca di tweet.
- 4 Menghapus *Stopwords*. *Stopwords* adalah kata-kata yang tidak mempengaruhi proses klasifikasi.
- 5 Stemming, Tujuan dari *stemming* adalah untuk mendapatkan kata dasar dari sebuah kata. Metode stemming yang digunakan adalah algoritma Algoritma Nazief dan Adriani

Tabel 3. 1 contoh praproses

| Sebelum Praproses  | Sesudah Praproses  |
|--|--|
| RT @kumparan: kumparan menghitung bagaimana Prabowo-Sandi dapat memperoleh kemenangan 54,91 persen.<br><a href="https://t.co/xkE0yE1xLD">https://t.co/xkE0yE1xLD</a> | kumparan hitung bagaimana prabowosandi dapat peroleh menang 54 91 persen |

### 3.1.6 Tokenisasi

Tokenisasi, Proses tokenisasi adalah proses pemotongan string input berdasarkan tiap kata yang menyusunnya, dalam penelitian ini tokenisasi yang digunakan adalah unigram. Tahap ini untuk memisahkan tiap kata di seluruh dokumen menjadi bentuk token.

Tabel 3. 2 contoh tokenisasi

| Tokenisasi   |
|--|
| “kumparan”, ”hitung”, “bagaimana”, “PrabowoSandi”, “dapat”, “peroleh”, “menang”, “54”, “91 ”, “persen” |

### 3.1.7 Proses labeling

Proses labeling dilakukan dengan tiga tahap. Pertama dilakukan pelabelan secara manual, proses ini dibantu oleh beberapa orang dan akan dikerjakan dalam waktu 30 hari. kedua melakukan labeling dengan dengan mengumpulkan kata-kata yang dianggap sering muncul dalam hate speech, data yang mengandung kata-kata tersebut akan dilabelkan sebagai hate speech.

### 3.1.8 Split Data

Disini akan dilakukan splitting data menjadi 3 bagian yaitu *train* digunakan untuk melatih model yang akan dibuat, *validation* data untuk memastikan proses training berjalan dengan baik, dan *test* digunakan untuk melihat hasil dari model yang dibuat. Proporsinya sebesar 60% untuk train, 20% untuk *validation* dan sisanya 20% di partisi untuk data testing.

*Data Train* adalah data yang akan kita gunakan untuk melatih model (menyesuaikan *weight* dan *bias*). *Data Validation* untuk evaluasi tuning *hyperparameter* di model (mengataur *hidden layer*, *optimizer*, *learning rate*, dll). Sedangkan data test sebagai *evaluator* dari *model* yang kita buat. *Dataset* yang digunakan adalah data *tweet* yang masih dalam bentuk Bahasa Indonesia.

### 3.1.9 Word Embedding

Sebelum diubah menjadi vector yang mempresentasikan setiap kata tahapan yang harus dilakukan adalah melakukan tokenisasi dan membentuk sebuah kamus akan berisi list seluruh kata yang unique yang ada dalam dataset. Setiap kata

nantinya di ubah Menggunakan metode one-hot encoding akan menghasilkan vektor dimana 1 merepresentasikan posisi kata tersebut berada, dan 0 untuk kata lainnya. Misal diberikan kalimat berikut: “Hate Speech Berbahasa Indonesia media sosial” Sehingga kamus yang terbentuk adalah: [“hate”, “speech”, “berbahasa”, “indonesia”, “media”, “sosial”]. Vektor representasi dari kata “sosial” mengacu pada format kamus diatas adalah [0,0,0,0,0,1] dan kata “indonesia” adalah [0,0,0,1,0,0], nantinya vector representasi ini yang akan di input untuk proses training model word2vec dan ELM. Setiap data nantinya akan dilatih dengan model word embedding sebanyak 10 iterasi. hasil dari word embedding menghasilkan vector dengan panjang tertentu untuk merepresentasikan masing-masing kata.

#### **3.1.10 *Embedding Layer***

Setelah dataset dipisah menjadi data latih dan data uji kemudian data diubah formatnya menjadi tensor. Embedding layer merupakan layer yang menerima input berupa *2D tensor integer*. Sebelum diubah menjadi tensor tahapan yang harus dilakukan adalah menentukan jumlah kata maksimal yang paling sering muncul didalam dataset yang akan dijadikan kamus untuk proses training model. Misalkan pada penelitian ini menggunakan 82597 kata yang muncul di dataset. Kemudian ditentukan maksimal jumlah kata dari masing-masing tweet sebanyak 44 kata (berdasarkan 183.006 kamus yang ditentukan diawal). Kemudian 82597 kata tersebut dikonversi menjadi sequence of integer (urutan kata yang paling sering muncul dalam bentuk integer). Setelah format menjadi sequence of integer kemudian diubah menjadi tensor 2-Dimensi dengan ketentuan (183.006, 98) artinya 183.006 sequence dengan masing-masing panjang 98.

Semua sequences yang akan menjadi input di embedding layer harus mempunyai panjang yang sama. Karena tweet sangat berbeda dalam hal panjangnya, Kita perlu memiliki sampel teks dengan panjang yang sama untuk memasukkannya ke jaringan saraf kita. Jika tweet lebih pendek dari 98 kata, kami akan menambahkannya dengan nol. Setelah itu diatur jumlah maksimum kata yang akan digunakan, di sort berdasarkan urutan frekuensi yang terbesar. kata yang jarang muncul akan dihilangkan. Contoh konversi format text tweet menjadi 2D tensor yang nantinya akan menjadi input pada embedding layer dapat dilihat pada gambar 3.4.

```
[ 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 1 162 458 1307 216 25 2 166 206 216
  216 289 121 101 628 817 216 24]
```

Gambar 3. 2 contoh 2D tensor

### 3.1.11 Deep Learning

model Deep Learning yang akan digunakan pada penelitian ini sebagai :

1. Membuat layer Input. *Layer* pertama yang dibuat adalah Embedding Layer yang menggunakan vector dengan panjang 100 untuk merepresentasikan masing-masing kata. Dalam menentukan nilai vector akan menggunakan hasil dari word2vec dan ELMo. Dan penelitian ini jumlah inputan sebanyak 98. Nilai 98 berasal dari jumlah kata terbanyak dalam 1 tweet.
2. *Layer* berikutnya adalah LSTM atau GRU layer. Pada penelitian ini menggunakan 1 node. LSTM atau GRU menggunakan backpropagation dimana pembelajaran untuk memperkecil tingkat error dengan cara menyesuaikan bobotnya berdasarkan perbedaan output dan target yang diinginkan. Dalam proses ini nilai error ditunjukkan dalam loss.
3. *Layer* terakhir adalah *Dense Output Layer* dengan 1 unit neuran dan fungsi aktivasi menggunakan sigmoid, dimana apabila nilai sigmoid mendekati 0 maka tweet dianggap sebagai *non hate speech* dan apabila mendekati 1 maka tweet dianggap *hate speech*.

Langkah-langkah yang dilakukan dalam proses Deep learning adalah sebagai berikut.

1. Dalam proses Deep learning dilakukan proses membandingkan hasil dari tiap pengaturan. Berikut adalah proses pengaturan dalam deep learning. Pertama adalah menentukan metode yang akan digunakan. Dalam penelitian ini pertama-tama metode yang digunakan adalah GRU dengan word embedding CBOW.
2. Tahap selanjutnya adalah menentukan jumlah neuron. Dalam menentukan jumlah neuron penelitian ini menggunakan rule dari (Jeff, 2015) dimana neuron berada di antara 2/3 dari jumlah input ditambah jumlah output dan jumlah dua kali dari input. Karena jumlah input sebesar 98 maka neuron berada diantara 66 dan 196. Sehingga neuron yang dicoba adalah 66,130,190.

3. Tahap selanjutnya adalah tuning untuk menentukan nilai dropout. Dropout yang digunakan adalah 0.0, 0.2, 0.5.
4. Setelah dibandingkan semua model kemudian diambil parameter yang menghasilkan akurasi tertinggi lalu menyimpan model tersebut.
5. selanjutnya adalah mengganti word embedding dengan Skipgram dan ELMo. Lalu ulangi proses dari proses 2.
6. setelah proses semua model GRU dilatih selanjutnya menggunakan model LSTM.lalu ulangi proses 2-5.
7. Tahap berikutnya menggabungkan LSTM dan GRU menggunakan pengaturan terbaik masing-masing untuk *word embedding*. Lalu latih dan simpan model.
8. Menguji masing-masing model yang sudah disimpan dengan data uji

### **3.1.12 Pola *hate Speech***

Untuk Mendapatkann pola *hate speech* yang pertama dilakukan adalah mengaubah tweet menjadi bentuk POS Tagger. Selanjutnya melihat pola kalimat yang terbentuk dari data hate speech. Setelah itu pola kalimat hate speech dicoba pada data non hate speech untuk melihat apakah terdapat tweet yang terbentuk dari pola dan kondisi *hate speech*.

### **3.1.13 Evaluasi Model Deteksi Hate Speech**

Data testing yang sebelumnya sudah dilakukan tahap preprosesing, selanjutnya digunakan sebagai pengujian terhadap model yang sudah dilatih. Pengujian hasil identifikasi *hate speech* pada penelitian ini dilakukan untuk mengetahui bagaimana kinerja yang dimiliki oleh model yang dibuat pada penelitian ini terutama dalam mengenai akurasi dalam mengidentifikasi *hate speech*. Pengujian akan dilakukan berberapa kali dengan setiap pengujian memiliki konfigurasi yang berbeda satu sama lain, yaitu:

- a. Pengujian menggunakan LSTM dan *word embedding* CBOW.
- b. Pengujian menggunakan LSTM dan *word embedding* skip-gram
- c. Pengujian menggunakan LSTM dan *word embedding* ELMo
- d. Pengujian menggunakan GRU dan *word embedding* CBOW
- e. Pengujian menggunakan GRU dan *word embedding* skip-gram
- f. Pengujian menggunakan GRU dan *word embedding* ELMo

- g. Pengujian menggunakan LSTM+GRU dan *word embedding* CBOW
- h. Pengujian menggunakan LSTM+GRU dan *word embedding* skip-gram
- i. Pengujian menggunakan LSTM+GRU dan *word embedding* ELMo

Pengujian model dilakukan dengan menggunakan parameter *precision*, *recall*, dan *F-measure*. Pengujian menggunakan data test yang sudah disiapkan, dimana data test sudah ada label yang benar, lalu dibandingkan dengan hasil dari pengujian model.

### 3.1.14 Penyusunan Laporan

Penyusunan laporan dilakukan sebagai bentuk dokumentasi atas terlaksananya penelitian dari awal hingga berakhirnya penelitian. Penyusunan laporan dibuat sesuai dengan format yang telah ditentukan.

## 3.2 Rencana Penelitian

Berdasarkan metode penelitian, aktivitas penelitian ini direncanakan berlangsung selama kurang lebih 5 bulan dimulai dari juni 2019 sampai dengan oktober 2019 dengan rincian jadwal pelaksanaan seperti pada tabel 3.1.

Tabel 3. 3 Rencana penelitian

| Kegiatan             | juni 2019 |   | juli 2019 |   | agustus 2019 |   | september 2019 |   | oktober 2019 |   | November 2019 |   |   |   | Desember 2019 |   |   |   |
|----------------------|-----------|---|-----------|---|--------------|---|----------------|---|--------------|---|---------------|---|---|---|---------------|---|---|---|
|                      | 3         | 4 | 1         | 2 | 3            | 4 | 1              | 2 | 3            | 4 | 1             | 2 | 3 | 4 | 1             | 2 | 3 | 4 |
| Identifikasi Masalah |           |   |           |   |              |   |                |   |              |   |               |   |   |   |               |   |   |   |
| Studi Literatur      |           |   |           |   |              |   |                |   |              |   |               |   |   |   |               |   |   |   |
| Perancangan          |           |   |           |   |              |   |                |   |              |   |               |   |   |   |               |   |   |   |
| Pembuatan sistem     |           |   |           |   |              |   |                |   |              |   |               |   |   |   |               |   |   |   |
| Pengujian            |           |   |           |   |              |   |                |   |              |   |               |   |   |   |               |   |   |   |
| Penyusunan Laporan   |           |   |           |   |              |   |                |   |              |   |               |   |   |   |               |   |   |   |

## BAB IV

### HASIL DAN PEMBAHASAN

Bab ini berisi hasil dan pembahasan penelitian. Pada bab ini akan menjelaskan hasil penelitian secara detail dimulai dari proses pengumpulan data, penghapusan status Twitter yang persis sama, praproses, penghapusan status Twitter yang sama hasil praproses, pembagian data *training* dan *test*, *word embedding*, dan hasil model *deep learning*.

#### 4.1 Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan Bahasa pemrograman python dan menggunakan *library* Twint. Data yang dikumpulkan adalah status twitter yang mengandung kata “Jokowi” atau “Prabowo” baik itu *mention*, *reply*, *hashtag* ataupun isi Status Twitter tersebut. Data yang berhasil dikumpulkan adalah sebanyak 1,271,272 tweet.

Hasil dari pengumpulan status twitter menggunakan python disimpan dengan format csv dan terdiri dari beberapa fitur antara lain: *id*, *conversation\_id*, *created\_at*, *date*, *time*, *timezone*, *user\_id*, *username*, *name*, *place*, *tweet*, *mentions*, *urls*, *photos*, *replies\_count*, *retweets\_count*, *likes\_count*, *hashtags*, *cashtags*, *link*, *retweet*, *quote\_url*, *video*, *near*, *geo*, *source*, *user\_rt\_id*, *user\_rt*, *retweet\_id*, *reply\_to*, *retweet\_date*. data yang digunakan pada penelitian ini adalah fitur *tweet*. Status twitter yang berhasil dikumpulkan adalah status yang buat antara tanggal 13-30-20019 sampai 17-04-2019. Pada tabel 4.1 adalah contoh Status Twitter yang berhasil dikumpulkan.

**Tabel 4. 1 Contoh status twitter**

| username        | Status Twitter  |
|-----------------|---|
| akhira_zamana   | Besok kl pak prabowo jadi presiden,kita kawal agar mentrinya disiplin ya pak @Dahnilarzar @sandiuno @hnurwahid @rockygerung<br><a href="https://twitter.com/CNNIndonesia/status/1117382535435051013">https://twitter.com/CNNIndonesia/status/1117382535435051013</a><br>!&#x201c; |
| ijazz40         | ORANG MABOK KOK DIDENGAR  |
| badjanuswantara | Capresnya tdk percaya KPK, bela temennya yg dicokok KPK. Cawapresnya berkali2 diperiksa KPK. Tokoh2 pendukungnya sering serang KPK bahkan getol bubarkan KPK. Mau ngasih pensiun  |

|              |  |
|--------------|--|
|              | koruptor. Berantas korupsi dr hongkong.<br>pic.twitter.com/Uke42UYIMI  |
| ariaramdhana | Dugaan saya kok sepertinya para ketua KPPS KPU adalah para militan prabowo sandi baik yg diluar negeri maupun yg didalam negeri... " at Halte TransJakarta Museum Fatahillah |

## 4.2 Hapus Duplicate Sebelum Praproses

Data yang berhasil dikumpulkan akan di periksa apakah Status Twitter terdapat *duplicate*. Sehingga nantinya diperoleh data yang unik, proses ini dilakukan sebanyak dua kali, yaitu saat dataset selesai dikumpulkan dan setelah proposes. Setelah proses penghapusan duplicate diperoleh 1.114.358 tweet.

Tabel 4. 2 contoh status twitter yang sama

| Status twitter 1  | Status twitter 2   |
|---|--|
| saepuloh27: Pengacara Setya Novanto: JK Harus Seperti Jokowi<br><a href="https://news.detik.com/berita/3729035/pengacara-setya-novanto-jk-harus-seperti-jokowi">https://news.detik.com/berita/3729035/pengacara-setya-novanto-jk-harus-seperti-jokowi</a> | maulmaulana22: Pengacara Setya Novanto: JK Harus Seperti Jokowi<br><a href="https://news.detik.com/berita/3729035/pengacara-setya-novanto-jk-harus-seperti-jokowi">https://news.detik.com/berita/3729035/pengacara-setya-novanto-jk-harus-seperti-jokowi</a> |

## 4.3 Praproses

Tahap praproses teks merupakan tahap awal dari pemrosesan klasifikasi *hate speech*. Tahap praproses teks sentimen terdiri dari *cleansing*, *case folding*, dan *parsing*. Secara detail tahap praproses teks *hate speech* akan dijelaskan pada tahap selanjutnya.

### 4.3.1 *Cleansing*

Tahap pertama praproses yang dilakukan dalam penelitian ini adalah *cleansing*, yaitu proses membersihkan dokumen dari kata yang tidak diperlukan untuk mengurangi *noise*. Kata yang dihilangkan adalah *hashtag* (#), *username* (@username), url (<http://>) karakter khusus dan Simbol Pada tabel 4.3 akan dijelaskan data sebelum dan sesudah proses *cleansing*.

Tabel 4. 3 contoh hasil cleansing

| Status Twitter  | Setelah <i>cleansing</i>   |
|---|--|
| Sebenarnya pak @jokowi ini terbuat dari apa sih? Kmrin sore di GBK, malamnya Debat Capres, paginya sudah tiba di Arab, Ini betul2 seorang pemimpin yg tanpa | Sebenarnya pak ini terbuat dari apa sih Kmrin sore di GBK malamnya Debat Capres paginya sudah tiba di Arab Ini betul2 seorang pemimpin |



|  |  |
|--|--|
| mengenal lelah, pekerja keras sejati. Beruntung Indonesia memilikinya. #JokowiDILANjutkanLagi pic.twitter.com/G9n2Ju2VbN | yg tanpa mengenal lelah pekerja keras sejati Beruntung Indonesia memilikinya |
| Hi, #Senin. #PagiPagi masih sulit pilih Prabowo karena harus pake Sandi. â~1   | Hi masih sulit pilih Prabowo karena harus pake Sandi                         |

### 4.3.2 Case Folding

Tahap kedua dari tahap praproses pada penelitian ini adalah *case folding*, yaitu proses penyeragaman bentuk huruf serta penghapusan angka, yang digunakan adalah huruf abjad dari a sampai z. Tabel 4.4 adalah contoh data sebelum dan sesudah proses *case folding*.

Tabel 4. 4 contoh hasil *case folding*

| Status Twitter   | Setelah <i>case folding</i>  |
|--|--|
| Sebenarnya pak @jokowi ini terbuat dari apa sih? Kmrin sore di GBK, malamnya Debat Capres, paginya sudah tiba di Arab, Ini betul2 seorang pemimpin yg tanpa mengenal lelah, pekerja keras sejati. Beruntung Indonesia memilikinya. #JokowiDILANjutkanLagi pic.twitter.com/G9n2Ju2VbN | sebenarnya pak ini terbuat dari apa sih kmrin sore di gbk malamnya debat capres paginya sudah tiba di arab ini betul seorang pemimpin yg tanpa mengenal lelah pekerja keras sejati beruntung indonesia memilikinya |
| Hi, #Senin. #PagiPagi masih sulit pilih Prabowo karena harus pake Sandi. â~1   | hi masih sulit pilih prabowo karena harus pake sandi   |

### 4.3.3 Stopwords

Tahap selanjutnya dari tahap praproses pada penelitian ini adalah menghapus *Stopwords*. *Stopwords* adalah proses penghapusan kata-kata yang tidak mempengaruhi proses klasifikasi.

Tabel 4. 5 contoh hasil *stopword*

| Status Twitter  | Setelah proses <i>stopword</i>   |
|---|--|
| Sebenarnya pak @jokowi ini terbuat dari apa sih? Kmrin sore di GBK, malamnya Debat Capres, paginya sudah tiba di Arab, Ini betul2 seorang pemimpin yg tanpa mengenal lelah, pekerja keras sejati. | terbuat kmrin sore gbk malamnya debat capres paginya arab betul pemimpin mengenal lelah pekerja keras sejati |

|  |                                 |
|--|---------------------------------|
| Beruntung Indonesia memilikinya.<br>#JokowiDILANjutkanLagi<br>pic.twitter.com/G9n2Ju2VbN | beruntung indonesia memilikinya |
| Hi, #Senin. #PagiPagi masih sulit pilih Prabowo karena harus pake Sandi. â~1             | sulit pilih prabowo harus sandi |

#### 4.3.4 Stemmping

Tahap selanjutnya dari tahap praproses pada penelitian ini adalah stemming. Tujuan dari *stemming* adalah untuk mendapatkan kata dasar dari sebuah kata. Metode stemming yang digunakan adalah algoritma Algoritma Nazief dan Adriani (Adriani et al., 2007).

Tabel 4. 6 contoh hasil *stemming*

| Status twitter   | Setelah <i>stemming</i>   |
|--|---|
| Sebenarnya pak @jokowi ini terbuat dari apa sih? Kmrin sore di GBK, malamnya Debat Capres, paginya sudah tiba di Arab, Ini betul2 seorang pemimpin yg tanpa mengenal lelah, pekerja keras sejati. Beruntung Indonesia memilikinya.<br>#JokowiDILANjutkanLagi<br>pic.twitter.com/G9n2Ju2VbN | buat kmrin sore gbk malam debat capres pagi arab pimpin kenal lelah kerja keras sejati untung indonesia milik |
| Hi, #Senin. #PagiPagi masih sulit pilih Prabowo karena harus pake Sandi. â~1   | sulit pilih prabowo pake sandi  |

#### 4.4 Hapus Duplicate Setelah Praproses

setelah melakukan praproses, dilakukan lagi proses penghapusan status twitter yang mempunyai *duplicate*, status yang dihapus adalah status yang memiliki kesamaan setelah dilakukan praproses. Sisa status setelah penghapusan *duplicate* ini sebanyak 702,191 status twitter.

Tabel 4. 7 contoh status yang sama setelah praproses

| Status twitter 1  | Status twitter 2   | praproses   |
|---|--|---|
| budi_telco:Presiden @Jokowi, Koruptor Jangan Digaji, Pecat PNS Koruptor! #KoruptorKokDigaji<br>http://chnng.it/7mVWWhrNtÂ lewat @ChangeOrg_ID | jcapaldi59: Presiden @Jokowi, Koruptor Jangan Digaji, Pecat PNS Koruptor! #KoruptorKokDigaji - Tandatangani Petisi!<br>http://chnng.it/sbmnxB5RÂ lewat @ChangeOrg_ID | presiden koruptor gaji pecat pns koruptor tandatangani petisi |

|  |  |                      |
|--|--|----------------------|
| Hanya kecurangan yang bisa mengalahkan Prabowo<br>#GembiraJokowiJuara<br><a href="https://twitter.com/ElgarLouee/status/1117300307040489472">https://twitter.com/ElgarLouee/status/1117300307040489472</a> | Hanya kecurangan yg bisa mengalahkan Prabowo<br>#17April2019PeoplePower<br><a href="https://twitter.com/wahyususilo/status/1117343770394021888">https://twitter.com/wahyususilo/status/1117343770394021888</a> | curang kalah prabowo |
|--|--|----------------------|

#### 4.5 Labeling

Proses labeling membutuhkan waktu selama 4 bulan. Proses labeling dilakukan secara manual dengan dibantu sebanyak 5 orang, Pertama orang-orang yang akan melakukan labeling diberikan materi oleh para ahli mengenai apa itu hate speech dan ciri-ciri hate speech, selanjutnya hasil labeling diverifikasi oleh ahli sebanyak 4 orang dengan pengecekan secara acak. Dalam menentukan jumlah sampel yang akan diperiksa dengan menggunakan Rumus Slovin (Riduwan, 2005). Dimana :

$$n = 702,191/702,191 (0.05)^2 + 1$$

dimana n adalah sampel; 702,191 adalah populasi; nilai presisi 95% atau sig. = 0,05. Sehingga dari 702,191 tweet yang ada maka total tweet yang akan dicek adalah 400.

#### 4.6 Split data

Setelah proses labeling diperoleh sebanyak 91.292 data yang terlabel hate speech, dan 610.899 non hate speech. Dari data yang diperoleh diketahui bahwa dataset tidak seimbang, sehingga perlu diseimbangkan dengan metode Under sampling. Yaitu menyamakan jumlah hate speech dan non hate speech dengan mengurangi jumlah sampel yang lebih banyak sehingga jumlahnya sama dengan data sampel yang rendah, dalam hal ini data yang dikurangi adalah data non hate speech sehingga jumlahnya sama dengan jumlah data hate speech. sehingga data yang digunakan 91.292 hate speech dan 91.292 non hate speech. Total data yang digunakan sebanyak 182.584. Data akan dibagi menjadi 60% data train. 20% Validation dan 20% test.

#### 4.7 Word Embedding

Word embedding adalah metode mengubah sebuah kata menjadi sebuah vektor atau array yang terdiri dari kumpulan angka. Dalam penelitian ini menggunakan tiga metode *word embedding* yaitu skip-gram, CBOW dan ELMo

### 4.7.1 Skip Gram

Metode Skip-gram adalah metode word embedding dimana input-nya adalah yang dijadikan sebagai target dan output-nya adalah kata-kata yang ada disekitarnya. Menurut Mikolov bekerja dengan baik dengan data pelatihan yang sedikit, mampu mewakili kata-kata atau frasa yang langka sekalipun (Mikolov, Chen, Corrado, & Dean, 2013). Dari 82604 kata yang digunakan memerlukan waktu selama 40689.0 detik untuk melakukan pelatihan dengan jumlah iterasi sebanyak 10.

Hasil pelatihan word embedding Skip-gram menghasilkan 100 vector untuk tiap kata. *Vector* ini nantinya akan digunakan sebagai inputan untuk *deep learning*. Berikut adalah contoh *vector* dari suatu kata.

```
In [10]: word_model.wv['provinsi']
Out[10]: array([-0.79331595, -0.51064235, -0.3532942,  0.38652435, -0.09543271,
  0.19974715,  0.04244421, -0.47684938, -0.2465619 , -0.36055934,
  0.4203582 , -1.1418825 , -0.38360666, -0.0115733 ,  0.4253231 ,
  0.25647417,  0.22999474, -0.18991849, -0.02805961, -0.1834607 ,
  0.8775792 ,  0.47231266,  0.3184127 , -0.6249331 , -0.42777675,
 -0.27804366, -0.35605505,  0.0537332 , -0.6263052 , -0.24674638,
 -0.4173288 ,  0.24783973,  0.37210327,  0.21408945,  0.35068557,
 -1.0312191 ,  1.1036856 , -0.18216407, -0.61603546, -0.45630726,
  0.43530554, -0.8506527 , -0.1305835 , -0.22213499, -0.29080603,
 -0.32044944, -0.2529463 ,  0.09053288, -0.5137673 ,  0.37100255,
 -0.09182149, -0.07708646,  0.3230341 ,  1.0007744 ,  0.51724946,
  0.66908437, -0.49277 , -0.88482326,  0.27638984,  0.18816455,
 -1.2386593 ,  0.4924027 , -0.9925637 ,  0.3909275 ,  0.10715275,
 -0.03706896, -0.4418084 ,  0.54195726,  0.07403405, -0.31571472,
 -0.3013994 ,  0.04490258, -0.62382126, -0.33947933, -0.37755257,
 -0.22749843, -0.70814025, -0.01241176, -0.35595196,  0.20222448,
 -0.36397606,  1.0392772 ,  0.7406588 , -0.59220576,  0.31765458,
  0.73339105, -0.32193902,  0.65910494, -0.23186791, -0.5851433 ,
 -0.37856388,  0.14155884, -0.32325238, -0.3081643 ,  0.17671132,
 -0.46158382, -0.29372254, -0.13034625, -0.60778815, -0.48406172],
 dtype=float32)
```

Gambar 4. 1 Contoh Vector Skipgram

Dari model Skip-gram kita dapat mengetahui kata-kata yang mempunyai kemiripan makna atau kata-kata yang sering digunakan bersamaan. Berikut adalah contoh kata-kata yang memiliki kemiripan makna dengan kata “provinsi” hasil dari model CBOW.

```
In [9]: word_model.wv.most_similar_cosmul(positive=['provinsi'])
Out[9]: [(('rohul', 0.8794548511505127),
 ('riau', 0.8770385980606079),
 ('dayak', 0.8753668665885925),
 ('mengatasai', 0.8728542923927307),
 ('konsorsium', 0.8727926015853882),
 ('barnus', 0.8683148622512817),
 ('kampar', 0.867203950881958),
 ('syamsuar', 0.8669190406799316),
 ('babel', 0.8628403544425964),
 ('wandi', 0.8622135519981384)]
```

Gambar 4. 2 Contoh Kedekatan kata menggunakan Skipgram

pada gambar 4.3 dapat dilihat. terdapat beberapa kata yang tidak seharusnya ada disekitar kata “provinsi”. Hal ini disebabkan skip-gram tidak memperhatikan jumlah kemunculan kata tersebut.

#### 4.7.2 CBOW

Metode CBOW adalah kebalikan dari metode Skip-gram dimana input-nya adalah kata-kata yang ada disekitarnya. Menurut Mikolov CBOW beberapa kali lebih cepat untuk berlatih daripada skip-gram, akurasi sedikit lebih baik untuk kata-kata yang sering muncul (Mikolov et. al, 2013). Penelitian ini memperlihatkan bahwa CBOW memang lebih cepat. Dari 82604 kata yang digunakan memerlukan waktu selama 7934.9 detik untuk melakukan pelatihan dengan jumlah iterasi sebanyak 10.

Hasil pelatihan word embedding CBOW menghasilkan 100 *vector* tiap untuk tiap kata. Vector ini nantinya akan digunakan sebagai inputan untuk deep learning. Berikut adalah contoh vector dari suatu kata.

```
In [4]: word_model.wv['provinsi']
Out[4]: array([-0.51998276, -0.48100093, -1.4363524, -0.7418443, 1.277539,
-1.1607317, 0.26723224, 0.97442067, 0.23522137, -0.34880343,
-0.6751489, 2.6053257, 0.33111423, -0.5251498, -0.29192916,
2.0082095, 0.55177224, 1.1927519, -1.0950462, 0.22916877,
-0.5047866, -0.46969026, 0.8287551, -0.20118052, 1.0101523,
0.34894058, 0.06187957, -0.11547926, 0.39255804, 0.2824912,
0.45706537, -3.52975, 2.0341198, -2.1299305, -2.4418967,
-0.32696348, 0.3143793, -0.8388768, -1.7710752, -0.93480706,
-0.71844804, 0.9218632, 0.65082157, 2.9522376, 0.6279074,
2.2781084, 0.25963813, -1.2603404, 0.969272, 1.2603201,
2.367739, -2.1693292, 0.57760745, -0.51827043, -1.0369629,
1.4152619, 2.0400846, 0.36627358, 1.3345404, -0.81281036,
0.3874615, -0.52402264, 0.93243563, 1.169307, -0.35504624,
-0.13731709, -0.39075997, 1.3823107, 0.20920436, -1.1555077,
0.34081852, -1.1002998, -1.2693466, 1.3873265, 3.0797548,
-1.7124152, 0.2642146, -0.27627143, -0.68989956, 0.6667553,
-1.0458335, -2.82181, -1.2788031, 0.906041, -0.38936928,
0.8650678, 0.51980627, -0.16691314, 1.3402674, 0.40380332,
0.44928905, -1.8051108, 0.12443582, -0.09183932, 1.4508667,
-0.9534306, 0.28770766, -0.69237876, 0.5312482, 0.77685785],
dtype=float32)
```

Gambar 4. 3 Contoh hasil Vector CBOW

Dari model CBOW kita dapat mengetahui kata-kata yang mempunyai kemiripan makna atau kata-kata yang sering digunakan bersamaan. Berikut adalah contoh kata-kata yang memiliki kemiripan makna hasil dari model CBOW

```
In [3]: word_model.wv.most_similar_cosmul(positive=['provinsi'])
Out[3]: [('kabupaten', 0.9448193311691284),
('utara', 0.9290441274642944),
('selatan', 0.9243647456169128),
('kab', 0.9204432368278503),
('riau', 0.9193588495254517),
('sumatra', 0.918342888352551),
('kalimantan', 0.915885865688324),
('hasundutan', 0.911843478679657),
('sulawesi', 0.9115939736366272),
('humbang', 0.9106057286262512)]
```

Gambar 4. 4 Contoh kedekatan kata menggunakan CBOW

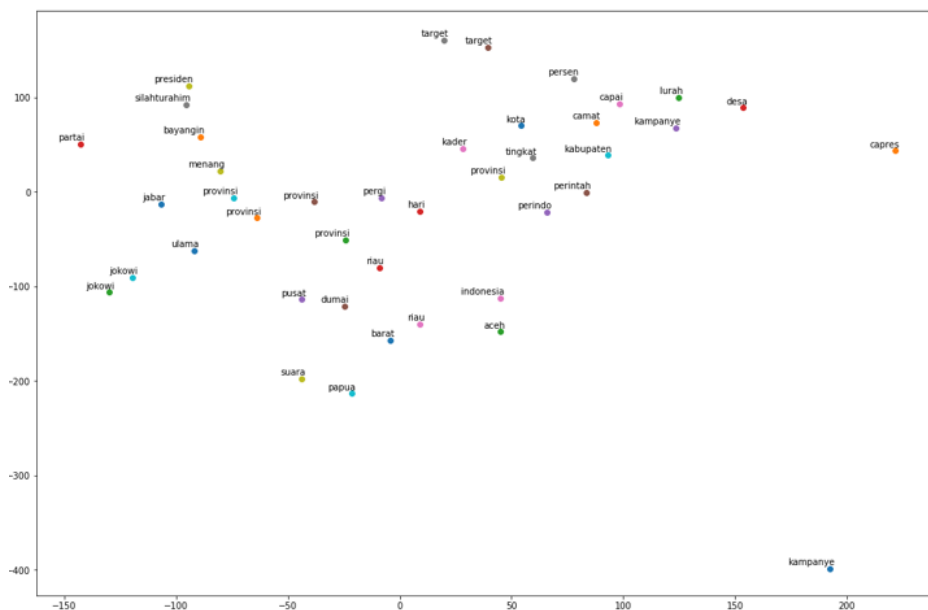
Pada gambar 4.5 dapat dilihat bahwa kata-kata yang ada disekitar kata "provinsi" adalah kata-kata yang mempunyai kedekatan makna. Hal ini disebabkan kata-kata tersebut sering muncul bersamaan dengan kata "provinsi".

### 4.7.3 ELMo

Metode ELMo adalah word embedding yang menggunakan metode biLSTM sehingga suatu kata bisa memiliki lebih dari satu array vector. Karena metode ini juga memperhatikan posisi kata saat berada didalam kalimat. Seperti yang bisa dilihat pada gambar 4.6 terdapat beberapa kalimat, dari kalimat-kalimat tersebut akan dibentuk menjadi word embedding. Hasil dari word embedding dapat dilihat pada gambar 4.5 dimana kata setiap kata memiliki vector yang berbeda tergantung posisinya dalam kalimat.

```
sent = ["kampanye", "capres", "provinsi", "riau", "pusat", "dumai", "riau"]
sent1 = ["silaturahmi", "presiden", "jokowi", "ulama", "provinsi", "aceh"]
sent2 = ["partai", "perindo", "perintah", "kader", "tingkat", "provinsi", "kabupaten",
        "kota", "camat", "lurah", "desa", "kampanye", "target", "capai", "persen", "suara", "papua", "barat"]
sent3 = ["bayangin", "jokowi", "hari", "pergi", "provinsi", "indonesia"]
sent4 = ["target", "menang", "provinsi", "jabar"]
```

Gambar 4. 5 Contoh Kalimat



Gambar 4. 6 Contoh Visualisasi kalimat ELMo

## 4.8 Deep Learning

Setelah mendapatkan hasil word embedding selanjutnya dilakukan proses pelatihan dengan *deep learning*. Penelitian ini menggunakan dua metode deep learning yaitu LSTM dan GRU. Setiap model deep learning akan dilatih dengan

nilai inputan dari tiap *word embedding*, sehingga terdapat 6 kali percobaan yaitu LSTM dengan *word embedding* CBOW, LSTM dengan *word embedding* Skipgram. LSTM dengan *word embedding* ELMo GRU dengan *word embedding* CBOW, GRU dengan *word embedding* Skipgram, GRU dengan *word embedding* ELMo.

#### 4.8.1 GRU-CBOW

Berikut adalah model GRU yang digunakan, dimana inputanya adalah matriks 98x100, 98 adalah panjang kalimat terpanjang yang terdapat dalam data yang digunakan, 100 adalah panjang vector CBOW untuk setiap kata. Selanjutnya dilanjutkan pelatihan dengan beberapa pengaturan neuron dan dropout yang nantinya akan menghasilkan output menentukan apakah kalimat tersebut termasuk hate speech atau tidak. Hasil validasi tertinggi selanjutnya akan digunakan uji coba menggunakan data test.

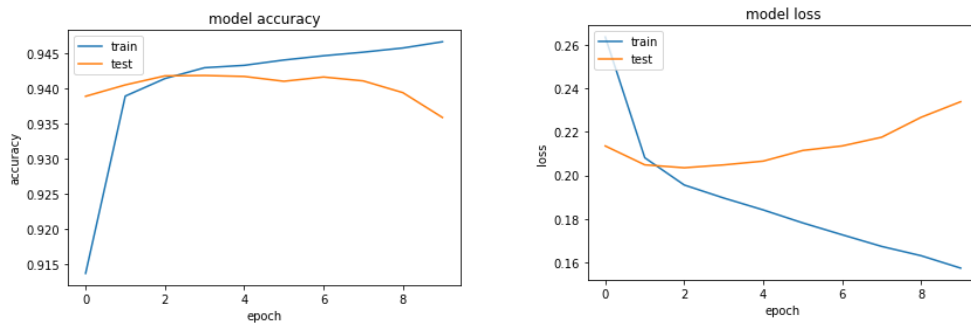
**Tabel 4. 8 hasil uji coba pengaturan model GRU-CBOW**

| neuron | dropout | Akurasi validation | Loss validasi |
|--------|---------|--------------------|---------------|
| 66     | 0       | 0.9358             | 0.2339        |
| 66     | 0.2     | 0.9432             | 0.2012        |
| 66     | 0.5     | 0.9404             | 0.2115        |
| 130    | 0       | 0.9339             | 0.2384        |
| 130    | 0.2     | 0.9435             | 0.2024        |
| 130    | 0.5     | 0.9406             | 0.2077        |
| 190    | 0       | 0.4656             | 0.2071        |
| 190    | 0.2     | 0.9426             | 0.8355        |
| 190    | 0.5     | 0.9414             | 0.2156        |

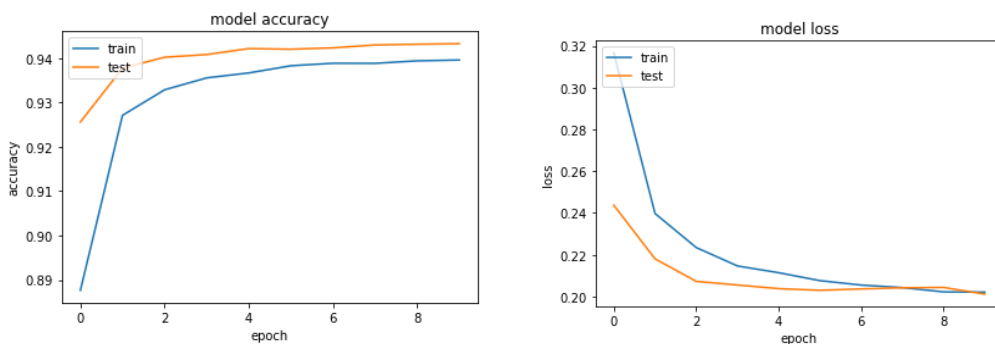
Seperti yang dapat dilihat pengaturan dengan neuron 130 dan dropout 0.2 menghasilkan akurasi dan loss tertinggi. Selain itu apabila diperhatikan terdapat beberapa pengaturan yang menghasilkan overfitting terutama untuk model yang menggunakan dropout 0, hal ini dapat dilihat dari loss dan akurasi selama pelatihan.

| Layer (type)            | Output Shape    |
|-------------------------|-----------------|
| embedding_3 (Embedding) | (None, 98, 100) |
| dropout_3 (Dropout)     | (None, 98, 100) |
| gru_3 (GRU)             | (None, 130)     |
| dense_3 (Dense)         | (None, 1)       |

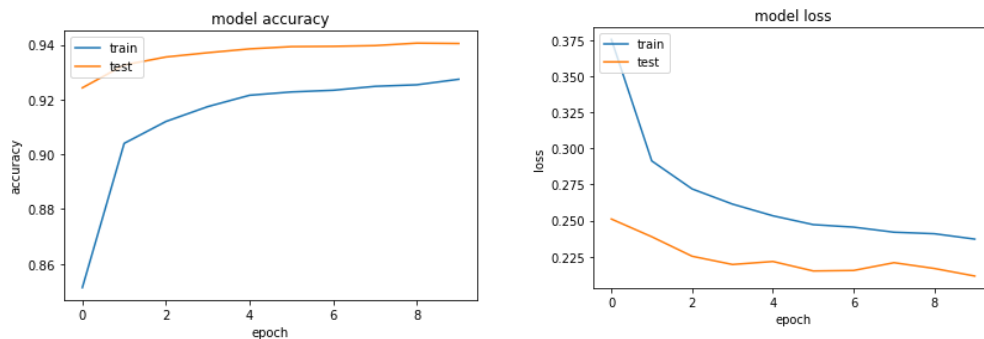
**Gambar 4. 7 Model GRU-CBOW**



**Gambar 4. 8 akurasi dan loss model GRU-CBOW neuron 66 dan dropout 0**

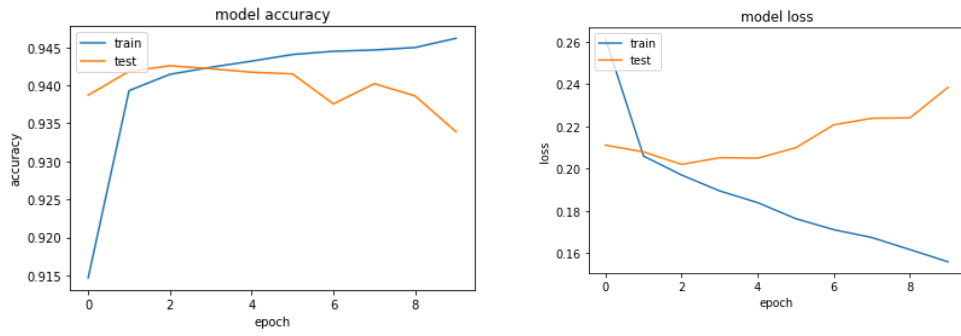


**Gambar 4. 9 akurasi dan loss model GRU-CBOW neuron 66 dan dropout 0.2**

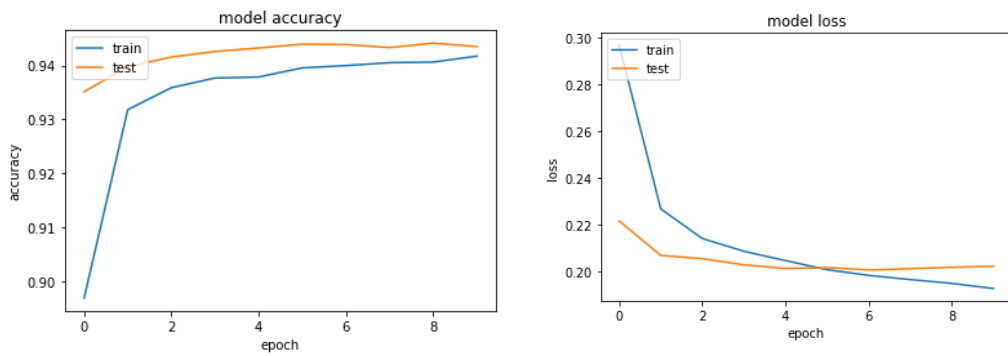


**Gambar 4. 10 akurasi dan loss model GRU-CBOW neuron 66 dan dropout 0.5**

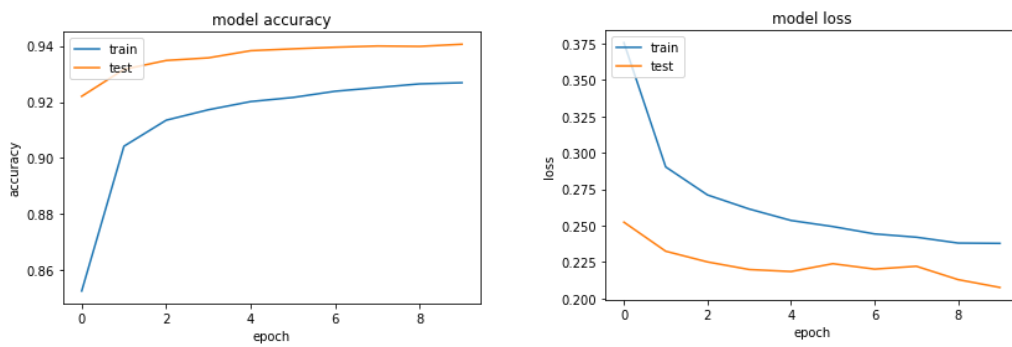




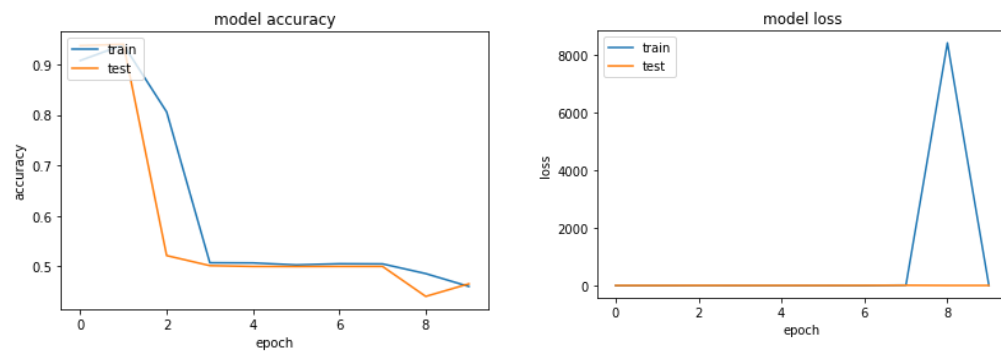
Gambar 4.11 akurasi dan loss model GRU-CBOW neuron 130 dan dropout 0



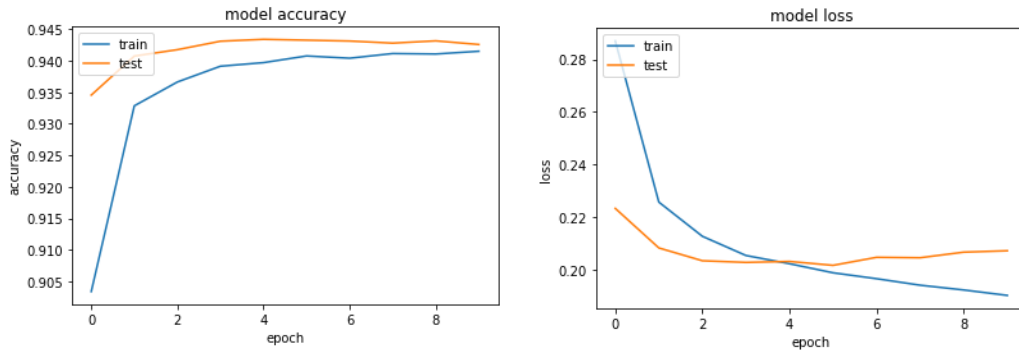
Gambar 4.12 akurasi dan loss model GRU-CBOW neuron 130 dan dropout 0.2



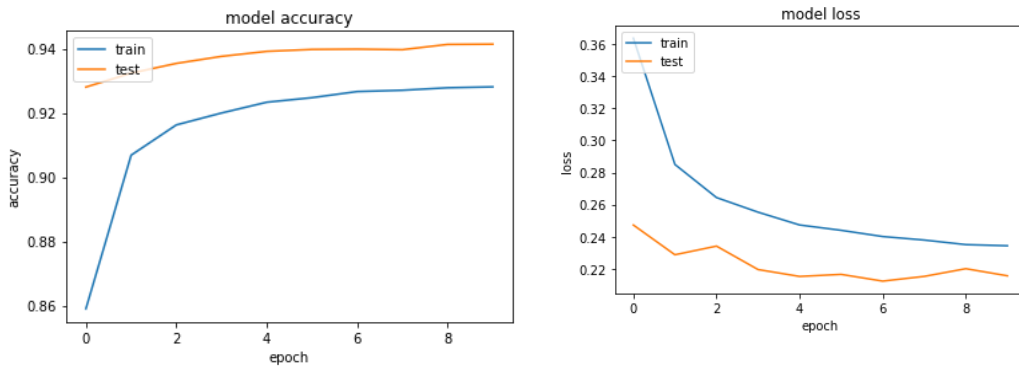
Gambar 4.13 akurasi dan loss model GRU-CBOW neuron 130 dan dropout 0.5



Gambar 4.14 akurasi dan loss model GRU-CBOW neuron 190 dan dropout 0

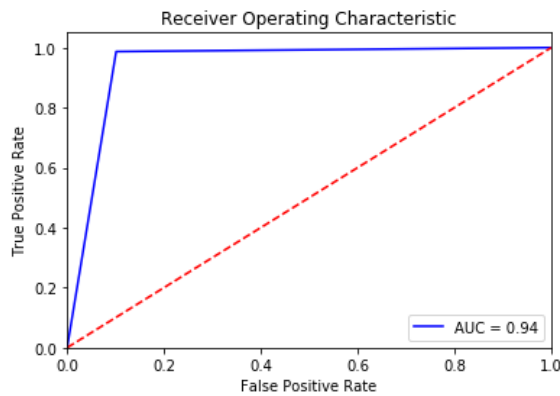


**Gambar 4. 15 akurasi dan loss model GRU-CBOW neuron 190 dan dropout 0.2**



**Gambar 4. 16 akurasi dan loss model GRU-CBOW neuron 190 dan dropout 0.5**

Setelah memperoleh pengaturan yang sesuai selanjutnya dilakukan pengujian dengan menggunakan data test sehingga diperoleh Accuracy: 0.942766, Precision: 0.906972, Recall: 0.986746, F1 score: 0.945179. selain itu dilakukan juga Receiver Operating Characteristic sebesar 0.94 sehingga menghasilkan grafik seperti gambar 4.16.



**Gambar 4. 17 Receiver Operating Characteristic GRU-CBOW**

Berikut adalah hasil confusion matrix dari pengujian model dengan menggunakan data test.

**Tabel 4. 9 confusion matrix**

|                               |                               |
|-------------------------------|-------------------------------|
| <b>True Positive</b><br>16410 | <b>False Positive</b><br>1848 |
| <b>False Negative</b><br>242  | <b>True Negatif</b><br>18017  |

#### 4.8.2 GRU-Skipgram

Berikut adalah model GRU yang digunakan, dimana inputnya adalah matriks 98x100, 98 adalah panjang kalimat terpanjang yang terdapat dalam data yang digunakan, 100 adalah panjang vector Skipgram untuk setiap kata. Selanjutnya dilanjutkan pelatihan dengan beberapa pengaturan neuron dan dropout yang nantinya akan menghasilkan output menentukan apakah kalimat tersebut termasuk hate speech atau tidak. Hasil validasi tertinggi selanjutnya akan digunakan uji coba menggunakan data test.

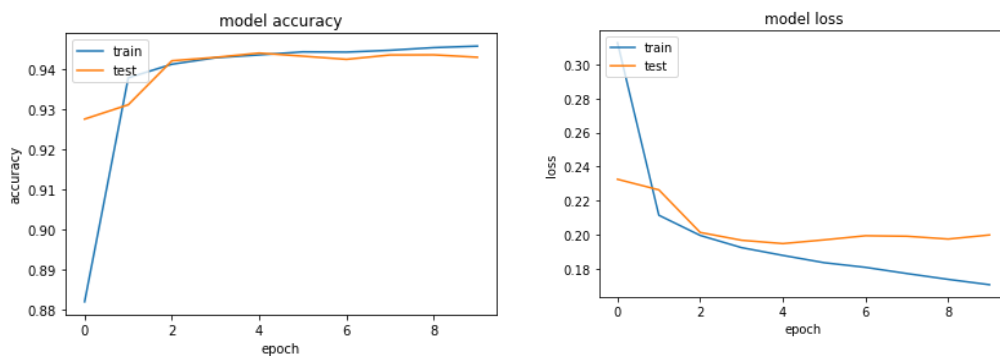
**Tabel 4. 10 hasil uji coba pengaturan model GRU-Skipgram**

| neuron | Dropout | akurasi | loss      |
|--------|---------|---------|-----------|
| 66     | 0       | 0.9429  | 0.1999    |
| 66     | 0.2     | 0.9444  | 0.1966    |
| 66     | 0.5     | 0.9407  | 0.2117    |
| 130    | 0       | 0.9418  | 0.2152    |
| 130    | 0.2     | 0.9462  | 0.1933    |
| 130    | 0.5     | 0.9420  | 0.2138    |
| 190    | 0       | 0.5000  | 1781.9612 |
| 190    | 0.2     | 0.9458  | 0.1919    |
| 190    | 0.5     | 0.9431  | 0.2304    |

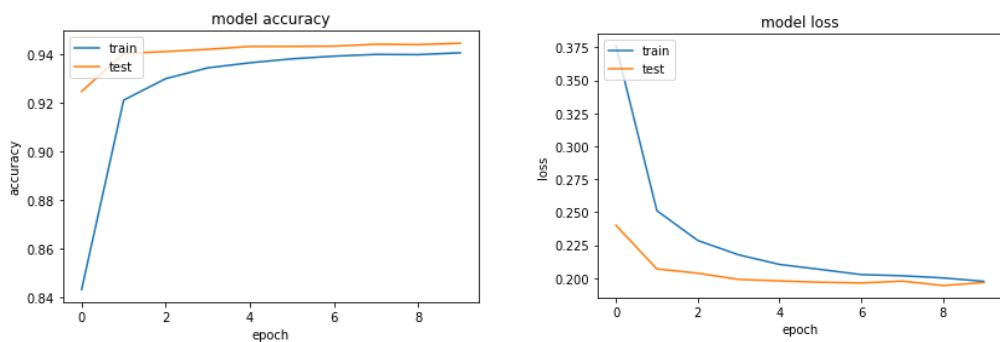
Seperti yang dapat dilihat pengaturan dengan neuron 130 dan dropout 0.2 menghasilkan akurasi dan loss tertinggi. Selain itu apabila diperhatikan terdapat beberapa pengaturan yang menghasilkan overfitting terutama untuk model yang menggunakan dropout 0, hal ini dapat dilihat dari loss dan akurasi selama pelatihan.

| Layer (type)            | Output Shape    |
|-------------------------|-----------------|
| embedding_3 (Embedding) | (None, 98, 100) |
| dropout_3 (Dropout)     | (None, 98, 100) |
| gru_3 (GRU)             | (None, 130)     |
| dense_3 (Dense)         | (None, 1)       |

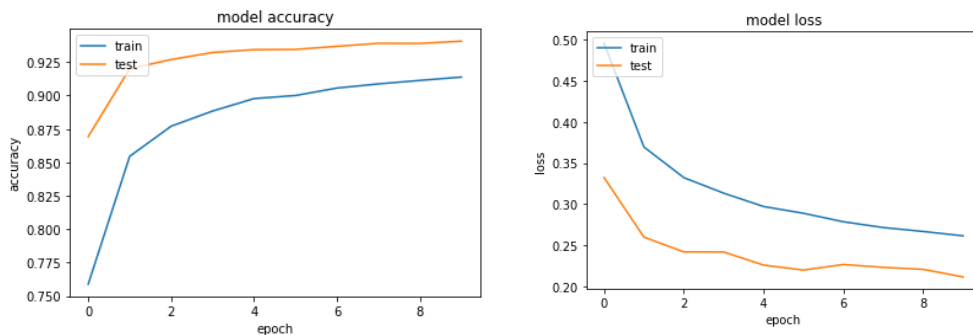
Gambar 4. 18 Model GRU-Skipgram



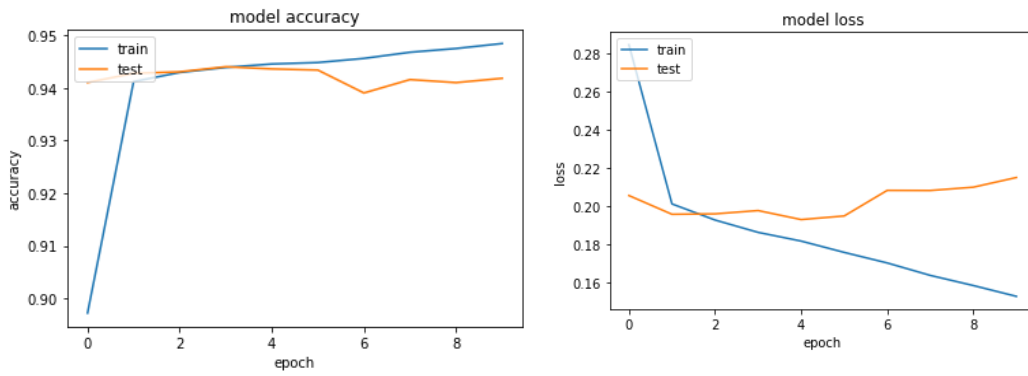
Gambar 4. 19 akurasi dan loss model GRU-Skipgram neuron 66 dan dropout 0



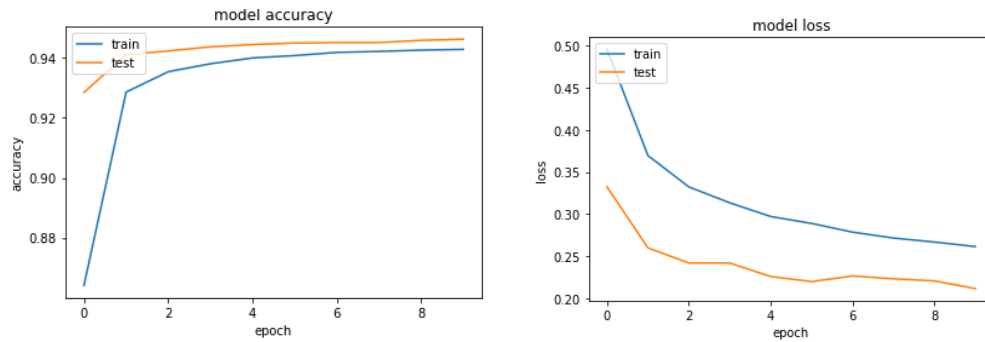
Gambar 4. 20 akurasi dan loss model GRU-Skipgram neuron 66 dan dropout 0.2



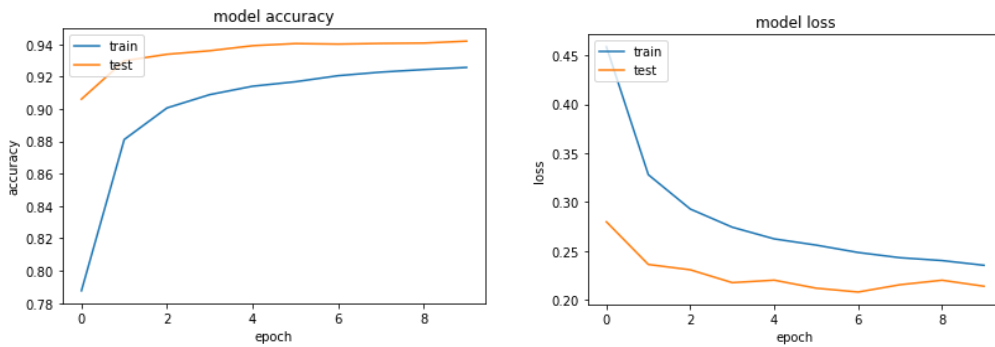
Gambar 4. 21 akurasi dan loss model GRU-Skipgram neuron 66 dan dropout 0.5



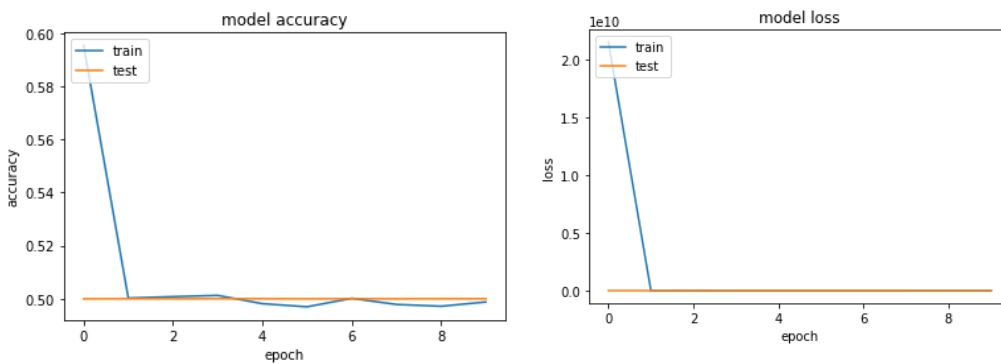
**Gambar 4. 22 akurasi dan loss model GRU-Skipgram neuron 130 dan dropout 0**



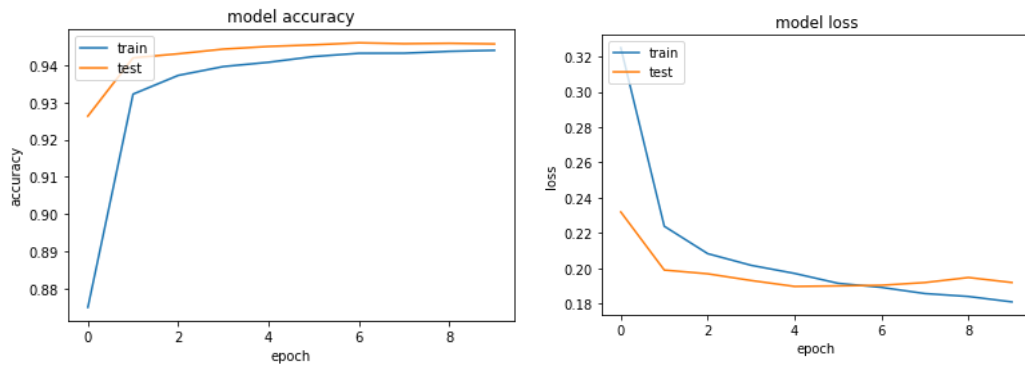
**Gambar 4. 23 akurasi dan loss model GRU-Skipgram neuron 130 dan dropout 0.2**



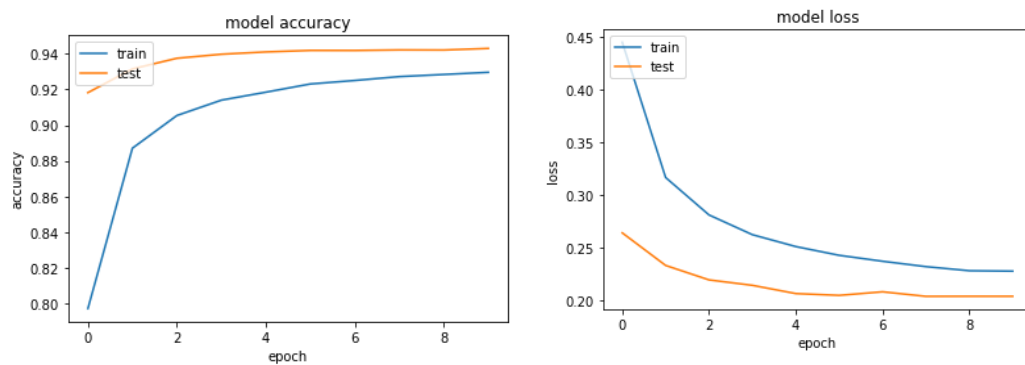
**Gambar 4. 24 akurasi dan loss model GRU-Skipgram neuron 130 dan dropout 0.5**



**Gambar 4. 25 akurasi dan loss model GRU-Skipgram neuron 190 dan dropout 0**

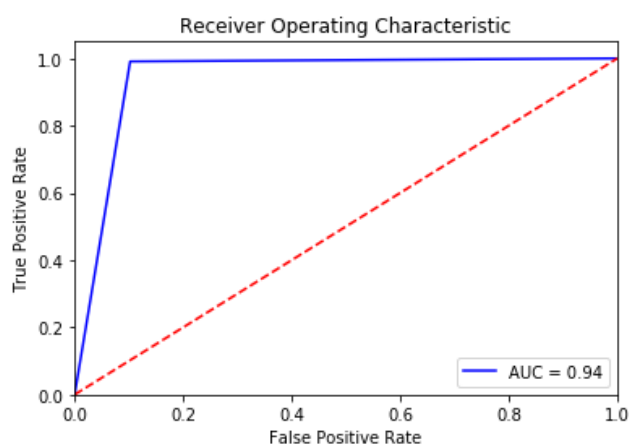


**Gambar 4. 26 akurasi dan loss model GRU-Skipgram neuron 190 dan dropout 0.2**



**Gambar 4. 27 akurasi dan loss model GRU-Skipgram neuron 190 dan dropout 0.5**

Setelah memperoleh pengaturan yang sesuai selanjutnya dilakukan pengujian dengan menggunakan data test sehingga diperoleh Accuracy: 0.944300, Precision: 0.906418, Recall: 0.990909, F1 score: 0.946782. selain itu dilakukan juga Receiver Operating Characteristic sebesar 0.94 sehingga menghasilkan grafik seperti gambar 4.28.



**Gambar 4. 28 Receiver Operating Characteristic model GRU-Skipgram**

Berikut adalah hasil confusion matrix dari pengujian model dengan menggunakan data test.

**Tabel 4. 11 confusion matrix**

|                               |                               |
|-------------------------------|-------------------------------|
| <b>True Positive</b><br>16406 | <b>False Positive</b><br>1868 |
| <b>False Negative</b><br>166  | <b>True Negatif</b><br>18093  |

### 4.8.3 GRU-ELMo

Berikut adalah model GRU yang digunakan, dimana inputnya adalah matriks 1x1024, 1 adalah kalimat yang akan di input kedalam model sehingga setiap kalimat dimasukan satu per satu, 1024 adalah panjang vector 1024 untuk setiap kata. Selanjutnya dilanjutkan pelatihan dengan beberapa pengaturan neuron dan dropout yang nantinya akan menghasilkan output menentukan apakah kalimat tersebut termasuk hate speech atau tidak. Hasil validasi tertinggi selanjutnya akan digunakan uji coba menggunakan data test

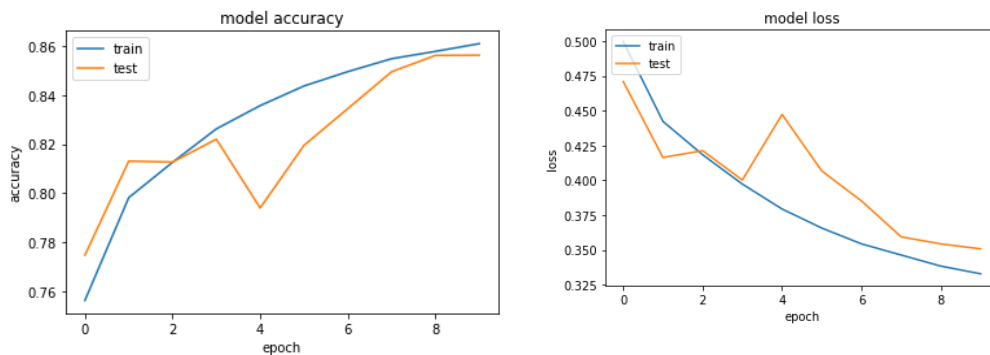
**.Gambar 4. 29 hasil uji coba pengaturan model GRU-ELMo**

| neuron | dropout | akurasi | loss   |
|--------|---------|---------|--------|
| 66     | 0       | 0.8562  | 0.3507 |
| 66     | 0.2     | 0.8217  | 0.3986 |
| 66     | 0.5     | 0.7961  | 0.4462 |
| 130    | 0       | 0.8322  | 0.3915 |
| 130    | 0.2     | 0.8352  | 0.3857 |
| 130    | 0.5     | 0.8029  | 0.4427 |
| 190    | 01      | 0.8535  | 0.3564 |
| 190    | 02      | 0.8270  | 0.3974 |
| 190    | 05      | 0.8146  | 0.4248 |

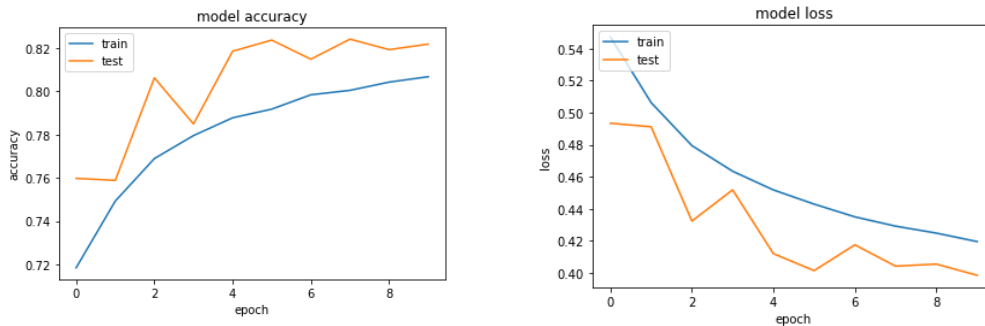
Seperti yang dapat dilihat pengaturan dengan neuron 130 dan dropout 0.2 menghasilkan akurasi dan loss tertinggi. Selain itu apabila diperhatikan terdapat beberapa pengaturan yang menghasilkan overfitting terutama untuk model yang menggunakan dropout 0, hal ini dapat dilihat dari loss dan akurasi selama pelatihan.

| Layer (type)         | Output Shape    |
|----------------------|-----------------|
| input_1 (InputLayer) | (None, 1)       |
| lambda_1 (Lambda)    | (None, 1024)    |
| reshape_1 (Reshape)  | (None, 1, 1024) |
| dropout_1 (Dropout)  | (None, 1, 1024) |
| gru_1 (GRU)          | (None, 130)     |
| dense_1 (Dense)      | (None, 2)       |

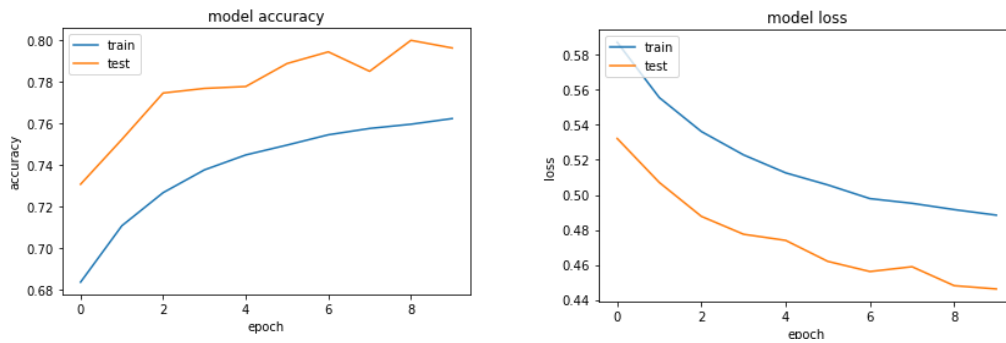
**Gambar 4. 30 model GRU ELMo**



**Gambar 4. 31 akurasi dan loss model GRU-ELMo neuron 60 dan dropout 0**

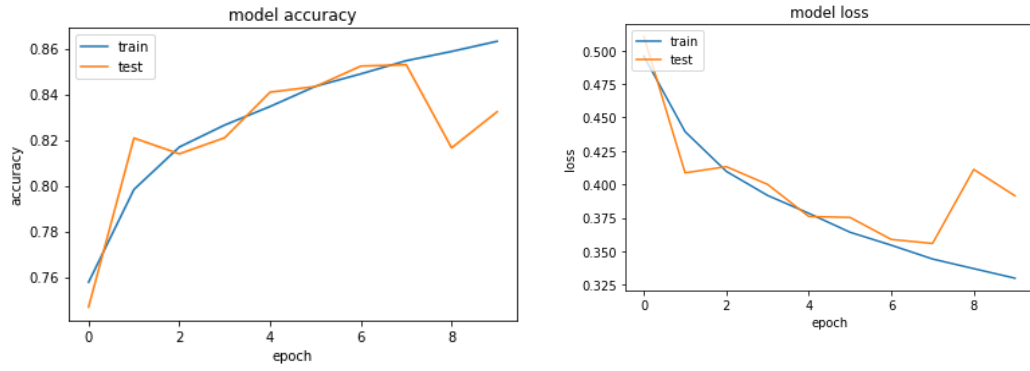


**Gambar 4. 32 akurasi dan loss model GRU-ELMo neuron 60 dan dropout 0.2**

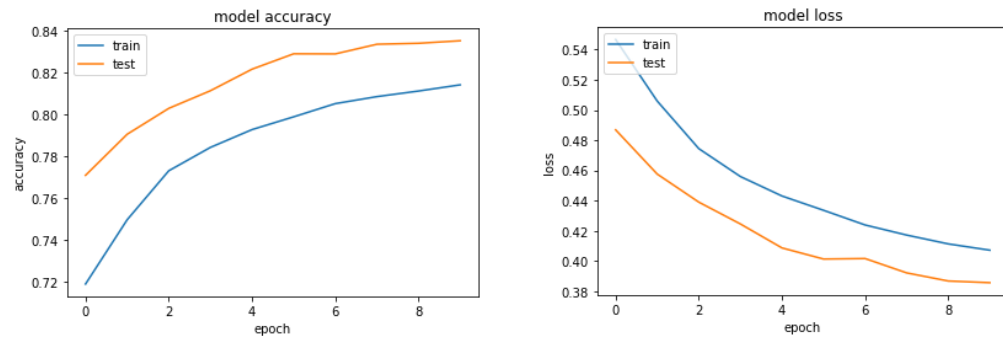


**Gambar 4. 33 akurasi dan loss model GRU-ELMo neuron 60 dan dropout 0.5**

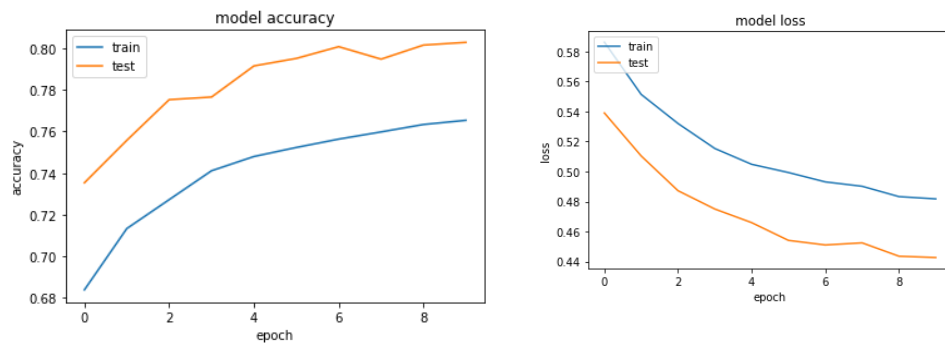




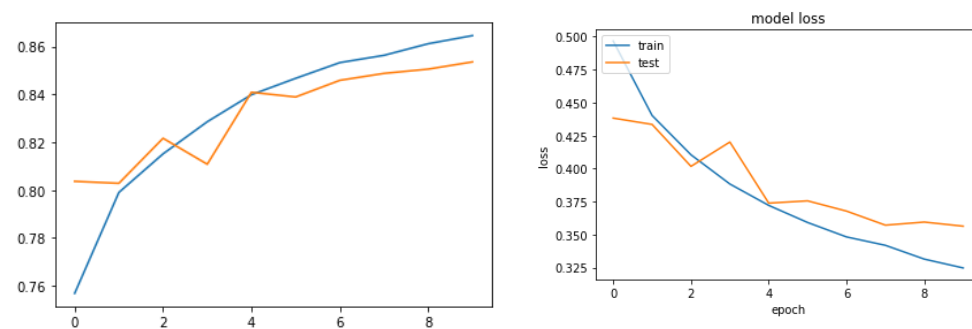
**Gambar 4. 34 akurasi dan loss model GRU-ELMo neuron 130 dan dropout 0**



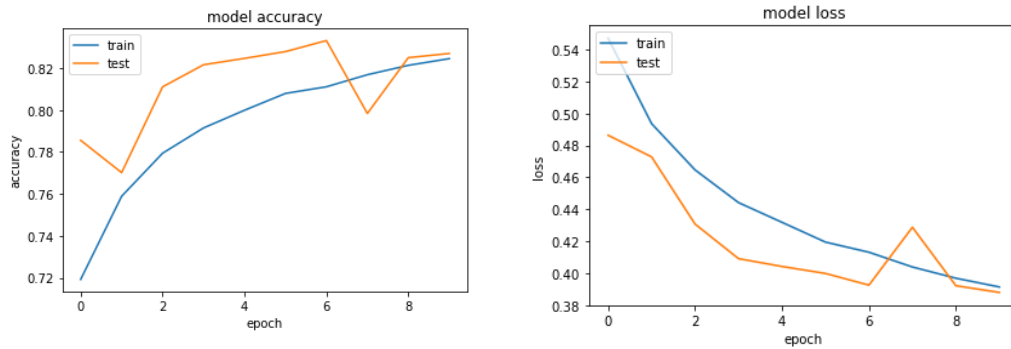
**Gambar 4. 35 akurasi dan loss model GRU-ELMo neuron 130 dan dropout 0.2**



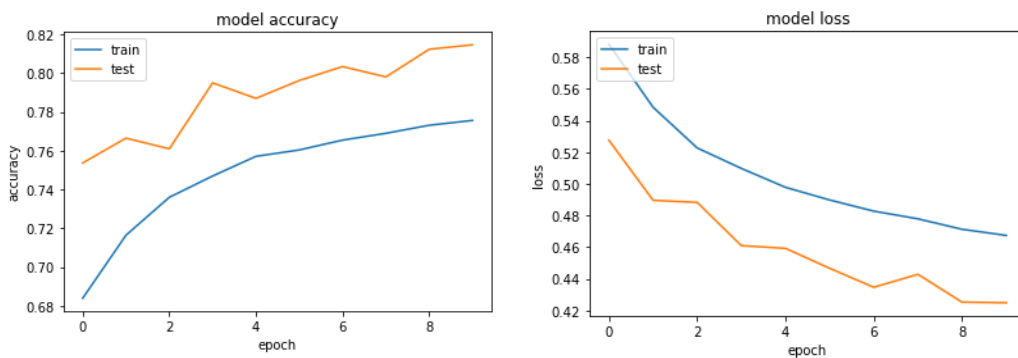
**Gambar 4. 36 akurasi dan loss model GRU-ELMo neuron 130 dan dropout 0.5**



**Gambar 4. 37 akurasi dan loss model GRU-ELMo neuron 190 dan dropout 0**

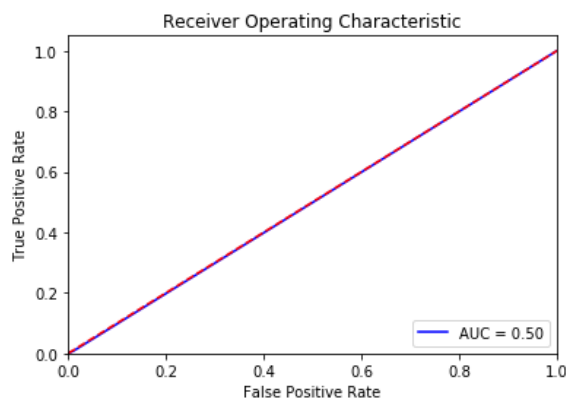


**Gambar 4. 38 akurasi dan loss model GRU-ELMo neuron 190 dan dropout 0.2**



**Gambar 4. 39 akurasi dan loss model GRU-ELMo neuron 190 dan dropout 0.5**

Setelah memperoleh pengaturan yang sesuai selanjutnya dilakukan pengujian dengan menggunakan data test sehingga diperoleh Accuracy: 0.498508, Precision: 0.431472, Recall: 0.009310, F1 score: 0.0946782. selain itu dilakukan juga Receiver Operating Characteristic sebesar 0.94 sehingga menghasilkan grafik seperti gambar 4.40.



**Gambar 4. 40 Receiver Operating Characteristic GRU-ELMo**

Berikut adalah hasil confusion matrix dari pengujian model dengan menggunakan data test.

**Tabel 4. 12 confusion matrix**

|                                |                              |
|--------------------------------|------------------------------|
| <b>True Positive</b><br>18034  | <b>False Positive</b><br>224 |
| <b>False Negative</b><br>18089 | <b>True Negatif</b><br>170   |

#### 4.8.4 LSTM-CBOW

Berikut adalah model GRU yang digunakan, dimana inputnya adalah matriks 98x100, 98 adalah panjang kalimat terpanjang yang terdapat dalam data yang digunakan, 100 adalah panjang *vector* CBOW untuk setiap kata. Selanjutnya dilanjutkan pelatihan dengan beberapa pengaturan neuron dan dropout yang nantinya akan menghasilkan output menentukan apakah kalimat tersebut termasuk hate speech atau tidak. Hasil validasi tertinggi selanjutnya akan digunakan uji coba menggunakan data test.

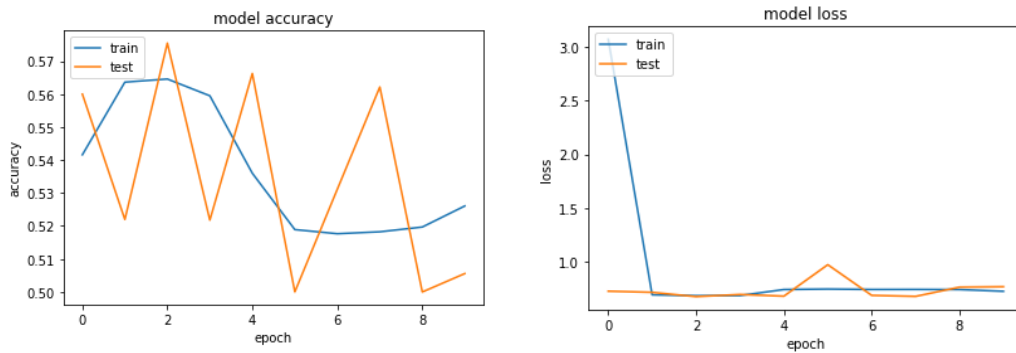
**Tabel 4. 13 hasil uji coba pengaturan model LSTM-CBOW**

| neuron | dropout | akurasi | loss      |
|--------|---------|---------|-----------|
| 66     | 0       | 0.5056  | 0.7706    |
| 66     | 0.2     | 0.5000  | 7759.4938 |
| 66     | 0.5     | 0.9241  | 0.2501    |
| 130    | 0       | 0.8552  | 0.3809    |
| 130    | 0.2     | 0.9408  | 0.2115    |
| 130    | 0.5     | 0.9412  | 0.2104    |
| 190    | 0       | 0.5000  | 0.6932    |
| 190    | 0.2     | 0.5000  | 0.6931    |
| 190    | 0.5     | 0.9414  | 0.2123    |

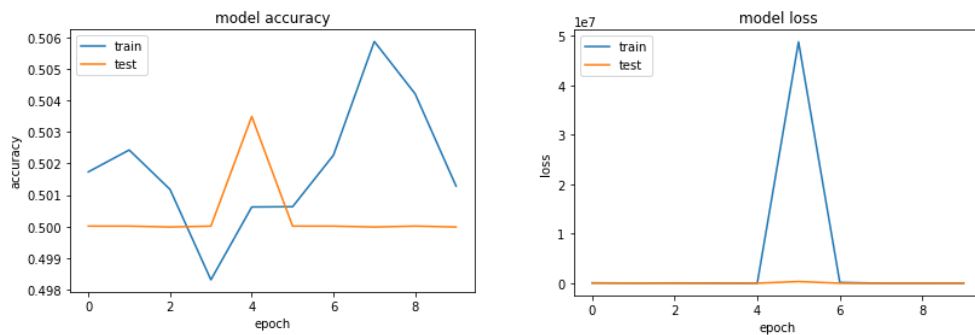
Seperti yang dapat dilihat pengaturan dengan neuron 130 dan dropout 0.5 menghasilkan akurasi dan loss tertinggi. Selain itu apabila diperhatikan terdapat beberapa pengaturan yang menghasilkan overfitting terutama untuk model yang menggunakan dropout 0, hal ini dapat dilihat dari loss dan akurasi selama pelatihan

| Layer (type)          | Output Shape    |
|-----------------------|-----------------|
| embedding (Embedding) | (None, 98, 100) |
| dropout (Dropout)     | (None, 98, 100) |
| lstm (LSTM)           | (None, 130)     |
| dense (Dense)         | (None, 1)       |

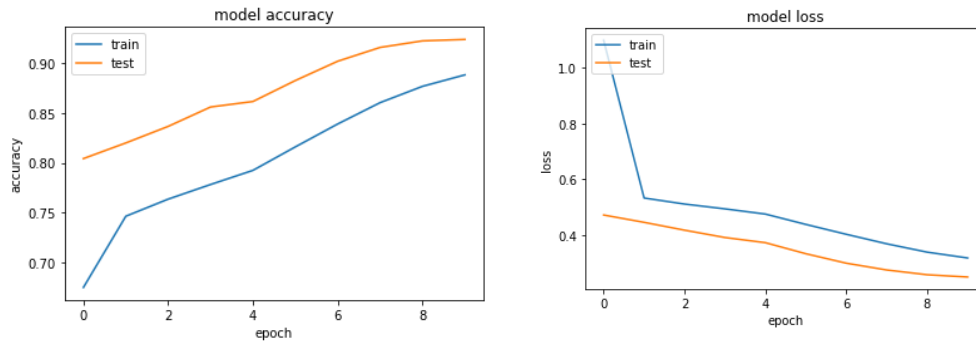
**Gambar 4. 41 Model LSTM-CBOW**



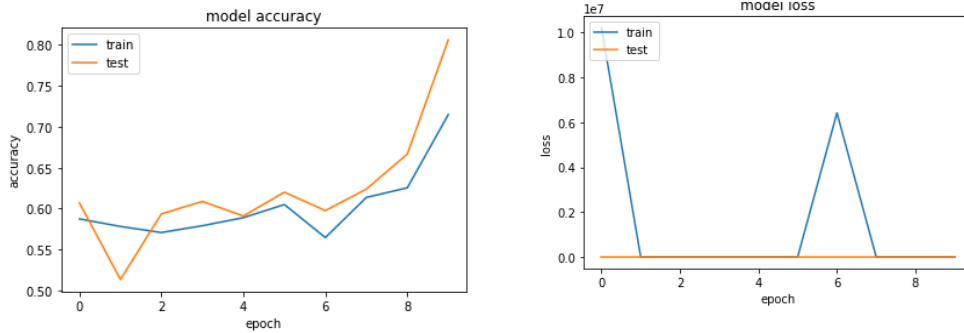
**Gambar 4. 42 akurasi dan loss model LSTM-CBOW neuron 66 dan dropout 0**



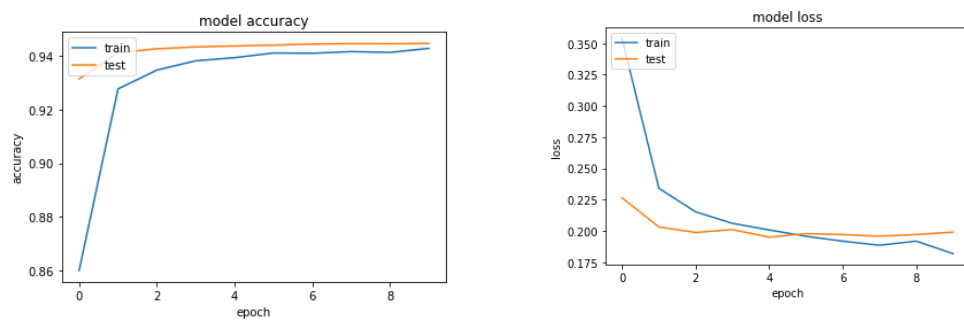
**Gambar 4. 43 akurasi dan loss model LSTM-CBOW neuron 66 dan dropout 0.2**



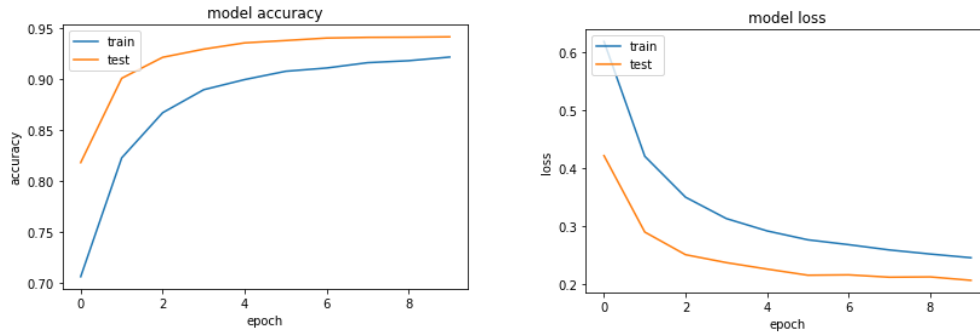
**Gambar 4. 44 akurasi dan loss model LSTM-CBOW neuron 66 dan dropout 0.5**



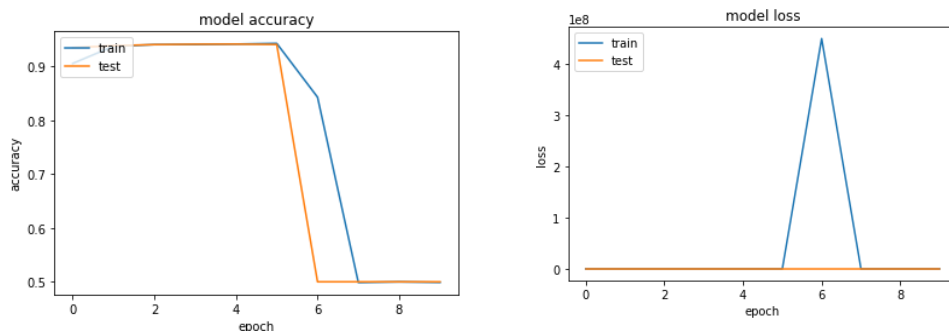
**Gambar 4. 45 akurasi dan loss model LSTM-CBOW neuron 130 dan dropout 0**



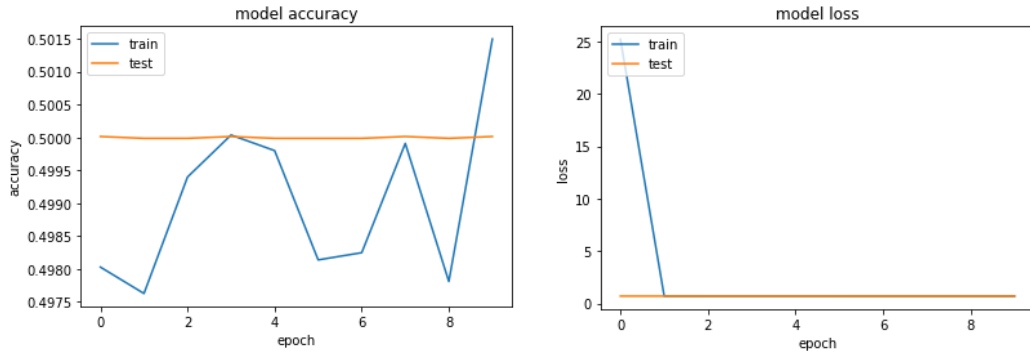
**Gambar 4. 46 akurasi dan loss model LSTM-CBOW neuron 130 dan dropout 0.2**



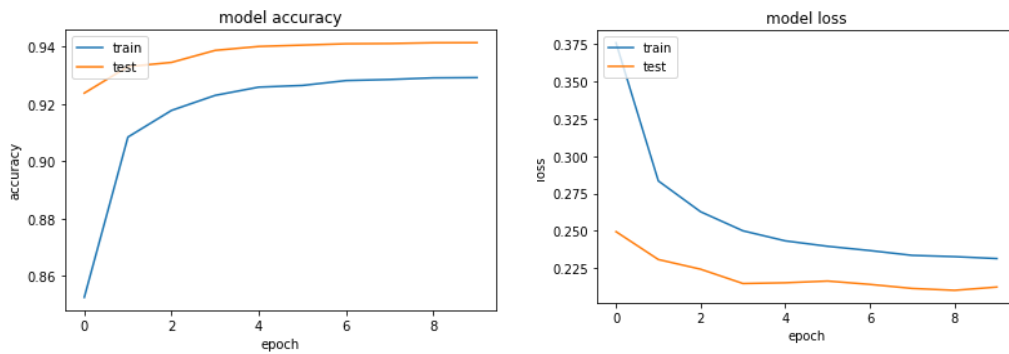
**Gambar 4. 47 akurasi dan loss model LSTM-CBOW neuron 130 dan dropout 0.5**



**Gambar 4. 48 akurasi dan loss model LSTM-CBOW neuron 190 dan dropout 0**

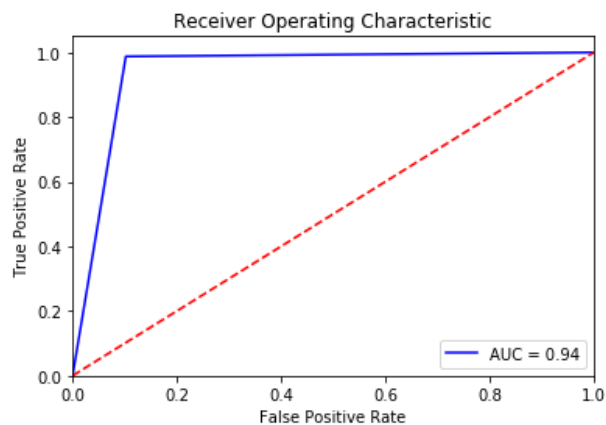


**Gambar 4. 49** akurasi dan loss model LSTM-CBOW neuron 190 dan dropout 0.2



**Gambar 4. 50** akurasi dan loss model LSTM-CBOW neuron 190 dan dropout 0.5

Setelah memperoleh pengaturan yang sesuai selanjutnya dilakukan pengujian dengan menggunakan data test sehingga diperoleh Accuracy: 0. 942684, Precision: 0. 906303, Recall: 0. 987458, F1 score: 0. 945142. selain itu dilakukan juga Receiver Operating Characteristic sebesar 0.94 sehingga menghasilkan grafik seperti gambar 4.40.



**confusion matrix**

Berikut adalah hasil confusion matrix dari pengujian model dengan menggunakan data test.

**Tabel 4. 14 confusion matrix**

|                               |                               |
|-------------------------------|-------------------------------|
| <b>True Positive</b><br>16394 | <b>False Positive</b><br>1864 |
| <b>False Negative</b><br>229  | <b>True Negatif</b><br>18030  |

#### 4.8.5 LSTM\_Skipgram

Berikut adalah model LSTM yang digunakan, dimana inputnya adalah matriks 98x100, 98 adalah panjang kalimat terpanjang yang terdapat dalam data yang digunakan, 100 adalah panjang *vector* Skipgram untuk setiap kata. Selanjutnya dilanjutkan pelatihan dengan beberapa pengaturan neuron dan dropout yang nantinya akan menghasilkan output menentukan apakah kalimat tersebut termasuk hate speech atau tidak. Hasil validasi tertinggi selanjutnya akan digunakan uji coba menggunakan data test.

**Tabel 4. 15 hasil uji coba pengaturan model LSTM-Skipgram**

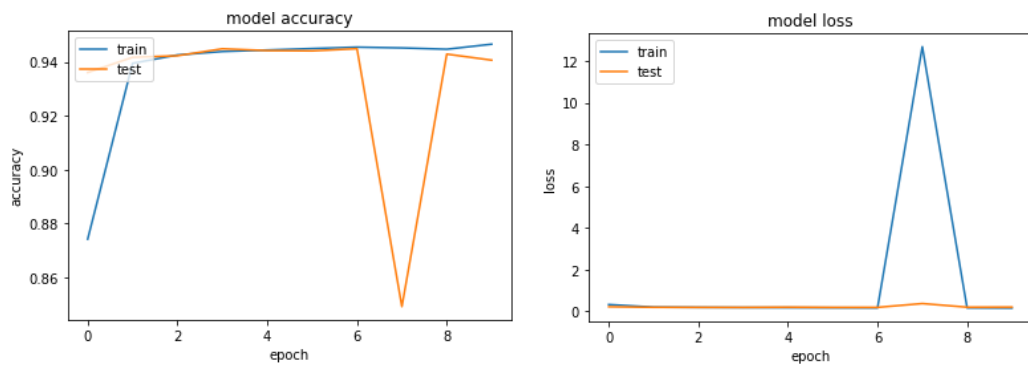
| neuron | dropout | akurasi  | loss   |
|--------|---------|----------|--------|
| 66     | 0       | 0.9372   | 0.1868 |
| 66     | 0.2     | 0.9437   | 0.1968 |
| 66     | 0.5     | 0.9392   | 0.2183 |
| 130    | 0       | 0.8059   | 0.4603 |
| 130    | 0.2     | 0.9448   | 0.1990 |
| 130    | 0.5     | 0.9413   | 0.2066 |
| 190    | 0       | 0.9391   | 0.2164 |
| 190    | 0.2     | 212.0967 | 0.5000 |
| 190    | 0.5     | 0.9422   | 0.2096 |

Seperti yang dapat dilihat pengaturan dengan neuron 130 dan dropout 0.2 menghasilkan akurasi dan loss tertinggi. Selain itu apabila diperhatikan terdapat

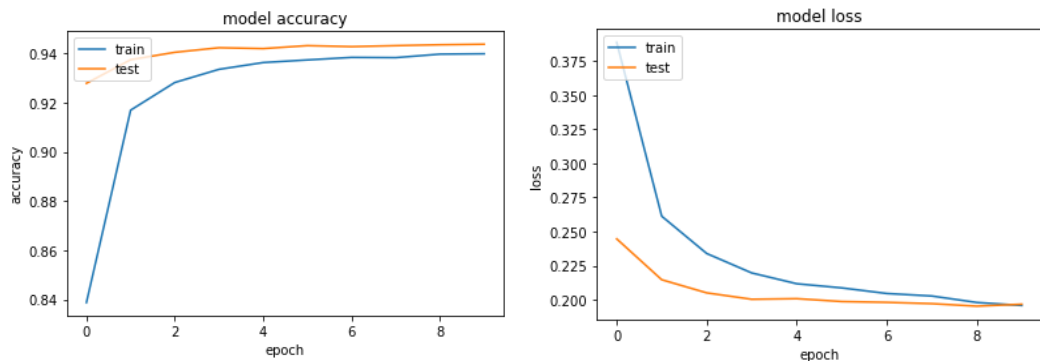
beberapa pengaturan yang menghasilkan overfitting terutama untuk model yang menggunakan dropout 0, hal ini dapat dilihat dari loss dan akurasi selama pelatihan

| Layer (type)          | Output Shape    |
|-----------------------|-----------------|
| embedding (Embedding) | (None, 98, 100) |
| dropout (Dropout)     | (None, 98, 100) |
| lstm (LSTM)           | (None, 130)     |
| dense (Dense)         | (None, 1)       |

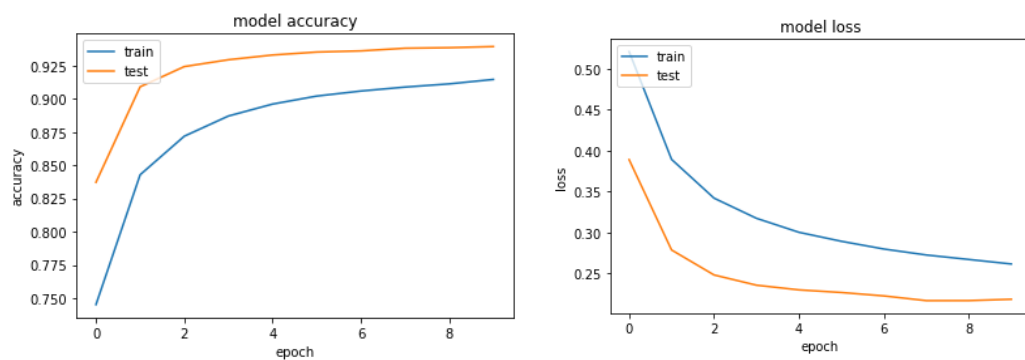
**Gambar 4. 51 Model LSTM Skipgram**



**Gambar 4. 52 akurasi dan loss model LSTM-Skipgram neuron 66 dan dropout 0**

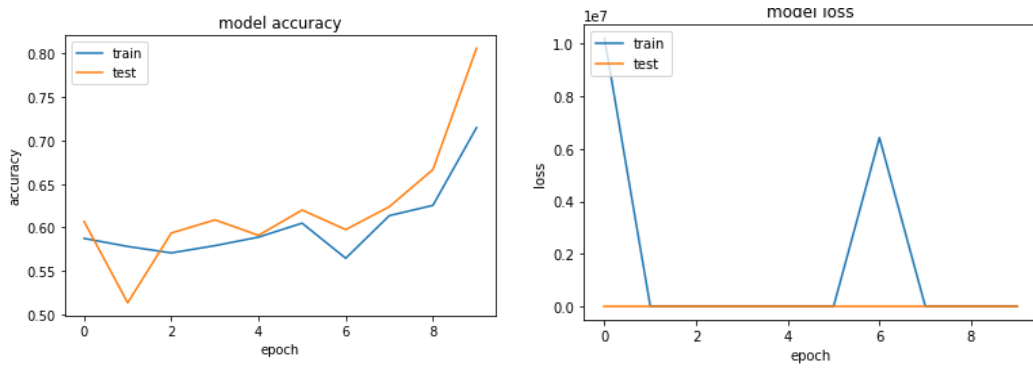


**Gambar 4. 53 akurasi dan loss model LSTM-Skipgram neuron 66 dan dropout 0.2**

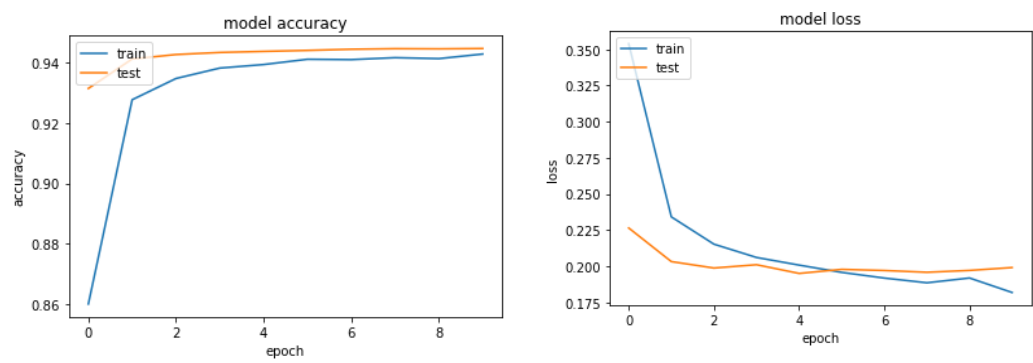




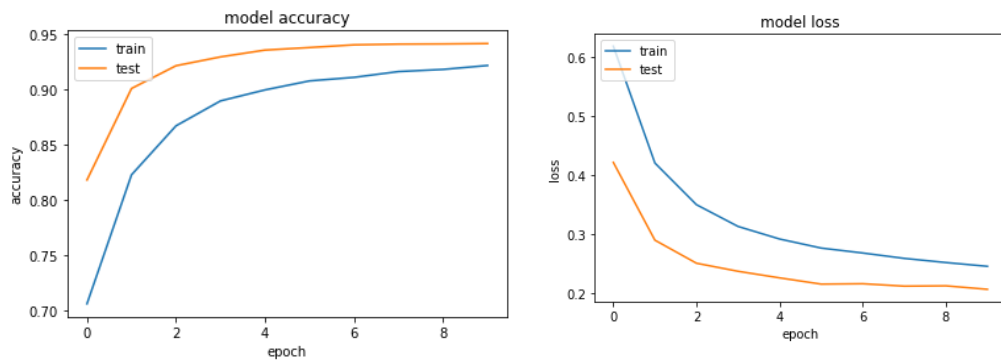
**Gambar 4. 54 akurasi dan loss model LSTM-Skipgram neuron 66 dan dropout 0.5**



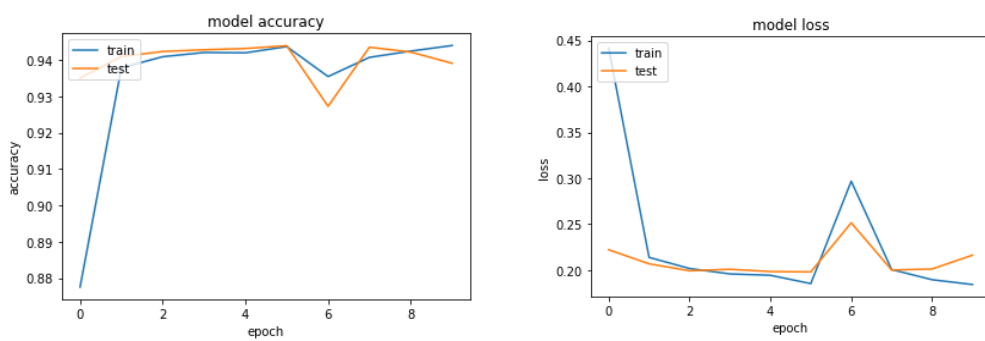
**Gambar 4. 55 akurasi dan loss model LSTM-Skipgram neuron 130 dan dropout 0**



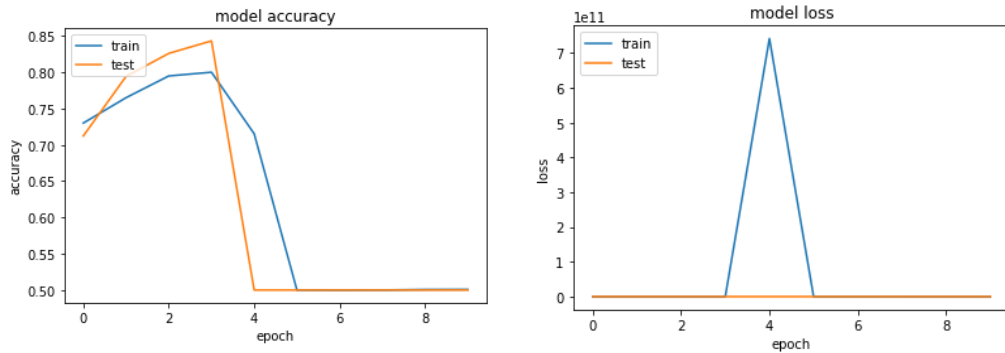
**Gambar 4. 56 akurasi dan loss model LSTM-Skipgram neuron 130 dan dropout 0.2**



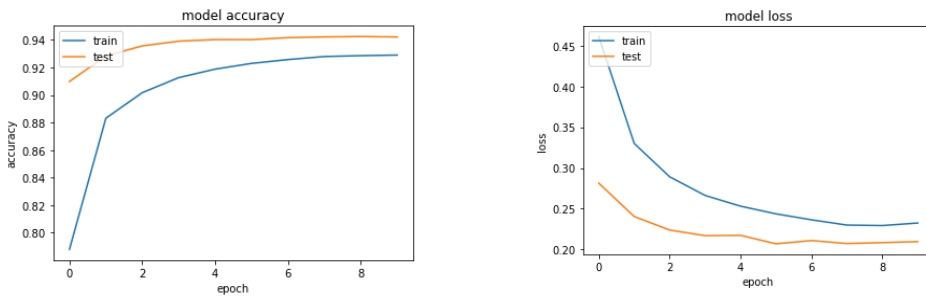
**Gambar 4. 57 akurasi dan loss model LSTM-Skipgram neuron 130 dan dropout 0.5**



**Gambar 4. 58 akurasi dan loss model LSTM-Skipgram neuron 190 dan dropout 0**

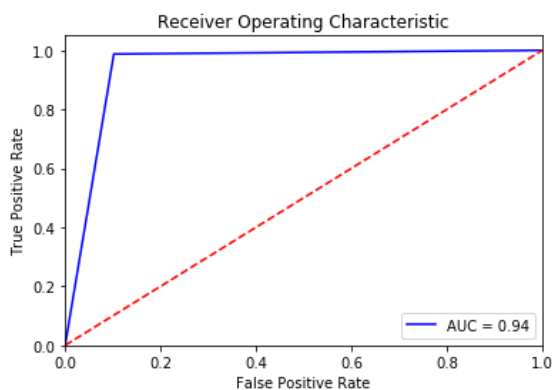


**Gambar 4. 59 akurasi dan loss model LSTM-Skipgram neuron 190 dan dropout 0.2**



**Gambar 4. 60 akurasi dan loss model LSTM-Skipgram neuron 190 dan dropout 0.5**

Setelah memperoleh pengaturan yang sesuai selanjutnya dilakukan pengujian dengan menggunakan data test sehingga diperoleh Accuracy: 0. 943944, Precision: 0. 906153, Recall: 0. 990470, F1 score: 0. 94 64 37. selain itu dilakukan juga Receiver Operating Characteristic sebesar 0.94 sehingga menghasilkan grafik seperti gambar 4.61.



**Gambar 4. 61 Receiver Operating Characteristic LSTM-Skupgram**

Berikut adalah hasil confusion matrix dari pengujian model dengan menggunakan data test

**Tabel 4. 16 confusion matrix**

|                               |                               |
|-------------------------------|-------------------------------|
| <b>True Positive</b><br>16394 | <b>False Positive</b><br>1864 |
| <b>False Negative</b><br>229  | <b>True Negatif</b><br>18030  |

#### 4.8.6 LSTM\_ELMo

Berikut adalah model GRU yang digunakan, dimana inputnya adalah matriks 45x100, 45 adalah panjang kalimat terpanjang yang terdapat dalam data yang digunakan, 100 adalah panjang vector CBOW untuk setiap kata. Selanjutnya dilanjutkan pelatihan dengan beberapa pengaturan neuron dan dropout yang nantinya akan menghasilkan output menentukan apakah kalimat tersebut termasuk hate speech atau tidak. Hasil validasi tertinggi selanjutnya akan digunakan uji coba menggunakan data test.

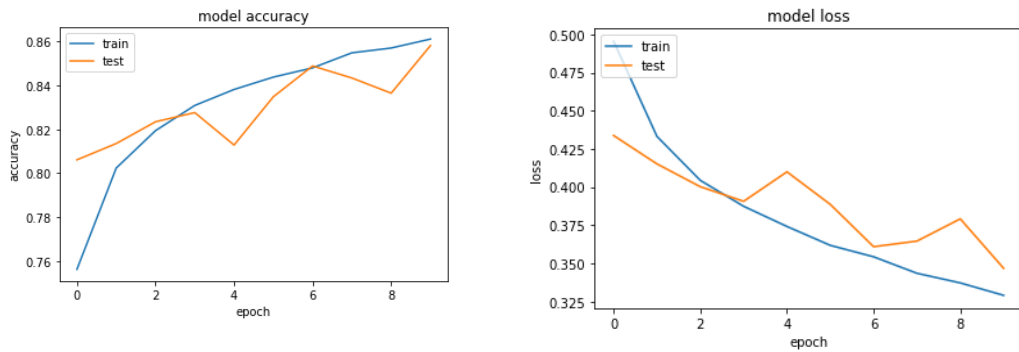
**Tabel 4. 17 hasil uji coba pengaturan model LSTM-ELMo**

| Neuron | dropout | akurasi | loss   |
|--------|---------|---------|--------|
| 66     | 0       | 0.8582  | 0.3467 |
| 66     | 0.2     | 0.8300  | 0.3928 |
| 66     | 0.5     | 0.7950  | 0.4446 |
| 130    | 0       | 0.8579  | 0.3508 |
| 130    | 0.2     | 0.8337  | 0.3890 |
| 130    | 0.5     | 0.8015  | 0.4406 |
| 190    | 0       | 0.8594  | 0.3461 |
| 190    | 0.2     | 0.8442  | 0.3784 |
| 190    | 0.5     | 0.7968  | 0.4494 |

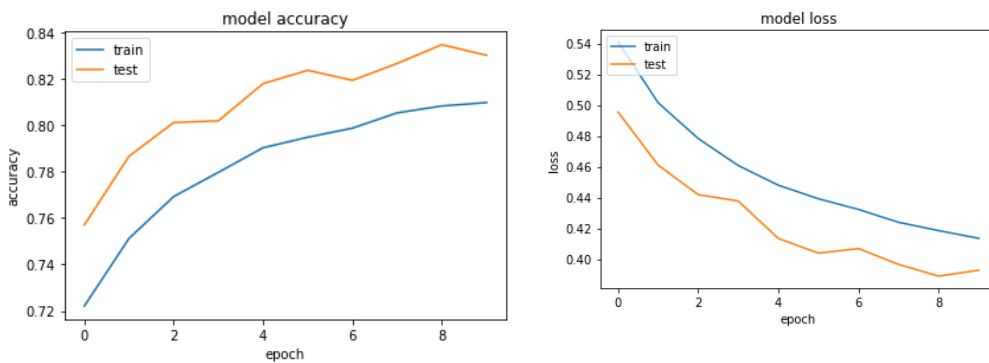
Seperti yang dapat dilihat pengaturan dengan neuron 130 dan dropout 0.2 menghasilkan akurasi dan loss tertinggi. Selain itu apabila diperhatikan terdapat beberapa pengaturan yang menghasilkan overfitting terutama untuk model yang menggunakan dropout 0, hal ini dapat dilihat dari loss dan akurasi selama pelatihan

| Layer (type)         | Output Shape    |
|----------------------|-----------------|
| input_2 (InputLayer) | (None, 1)       |
| lambda_2 (Lambda)    | (None, 1024)    |
| reshape_2 (Reshape)  | (None, 1, 1024) |
| dropout_2 (Dropout)  | (None, 1, 1024) |
| lstm_2 (LSTM)        | (None, 130)     |
| dense_2 (Dense)      | (None, 2)       |

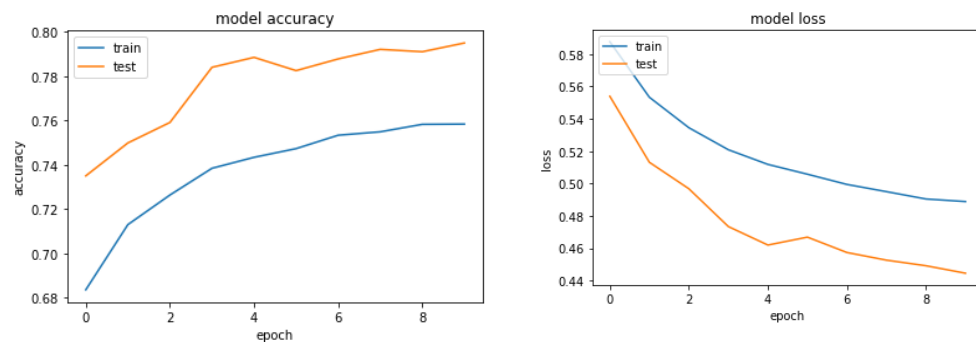
Gambar 4. 62 Model LSTM ELMo



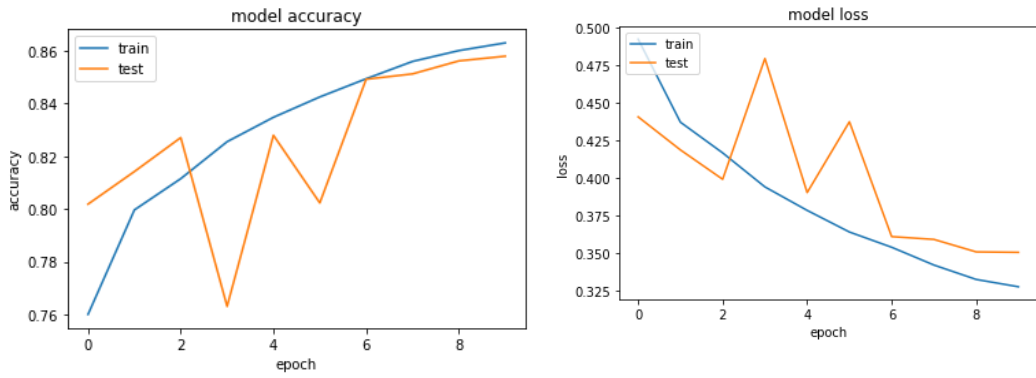
Gambar 4. 63 akurasi dan loss model LSTM-ELMo neuron 66 dan dropout 0



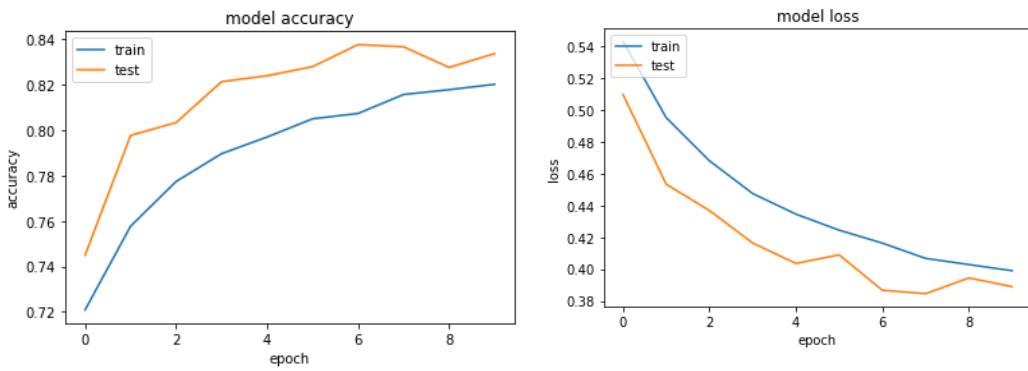
Gambar 4. 64 akurasi dan loss model LSTM-ELMo neuron 66 dan dropout 0.2



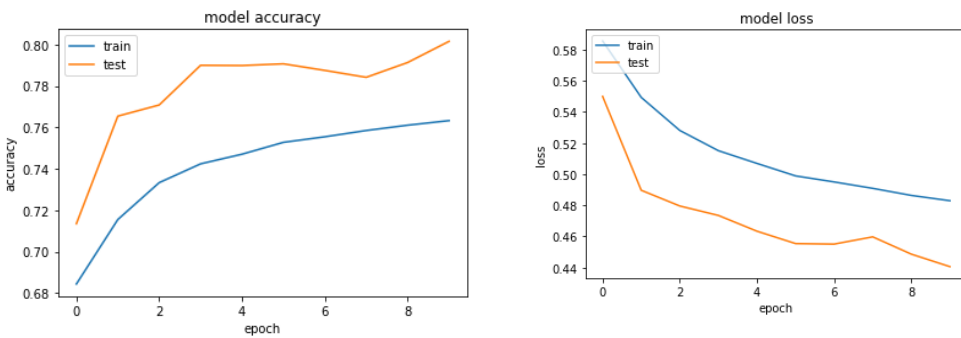
Gambar 4. 65 akurasi dan loss model LSTM-ELMo neuron 66 dan dropout 0.5



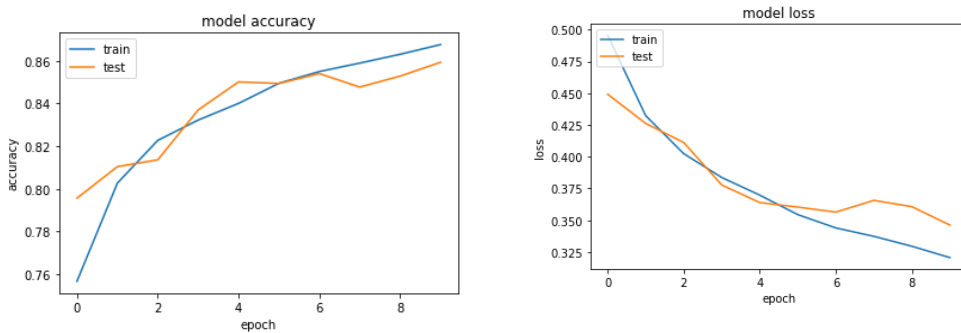
**Gambar 4.66 akurasi dan loss model LSTM-ELMo neuron 130 dan dropout 0**



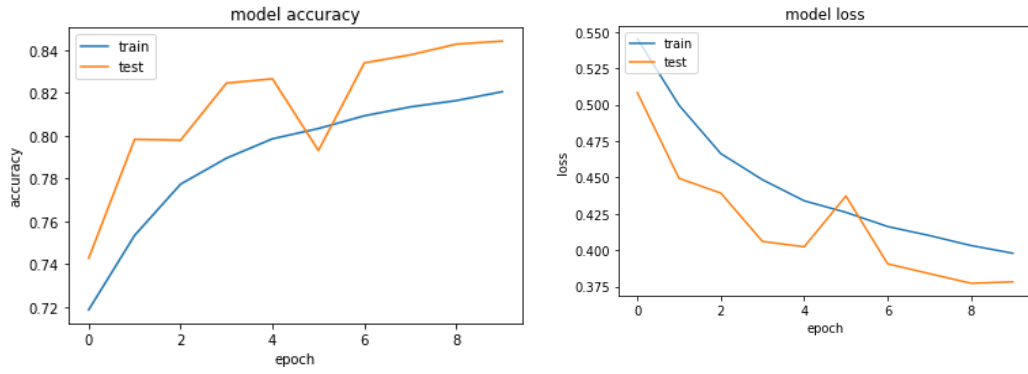
**Gambar 4.67 akurasi dan loss model LSTM-ELMo neuron 130 dan dropout 0.2**



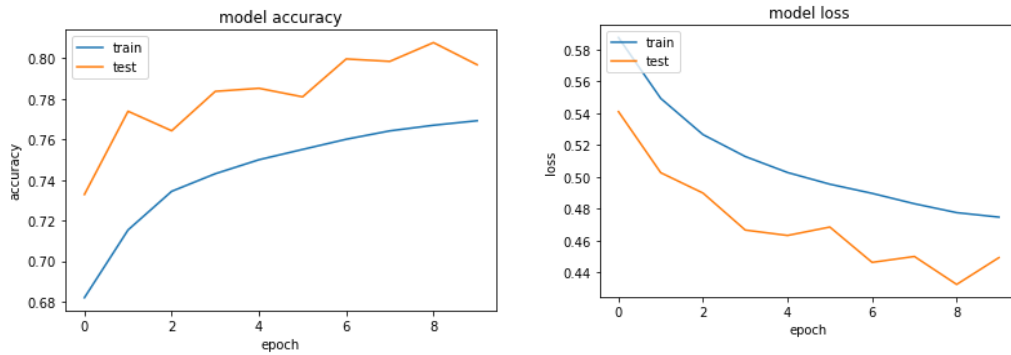
**Gambar 4.68 akurasi dan loss model LSTM-ELMo neuron 130 dan dropout 0.5**



**Gambar 4.69 akurasi dan loss model LSTM-ELMo neuron 190 dan dropout 0**

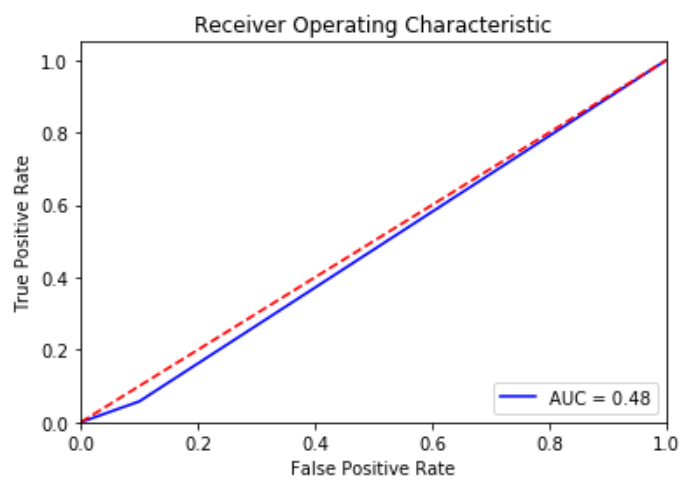


**Gambar 4.70 akurasi dan loss model LSTM-ELMo neuron 190 dan dropout 0.2 =**



**Gambar 4.71 akurasi dan loss model LSTM-ELMo neuron 190 dan dropout 0.5**

Setelah memperoleh pengaturan yang sesuai selanjutnya dilakukan pengujian dengan menggunakan data test sehingga diperoleh Accuracy: 0.478325, Precision: 0.365378, Recall: 0.057395, F1 score: 0.099206. selain itu dilakukan juga Receiver Operating Characteristic sebesar 0.48 sehingga menghasilkan grafik seperti gambar 4.72.



**Gambar 4.72 Receiver Operating Characteristic LSTM-ELMo**

Berikut adalah hasil confusion matrix dari pengujian model dengan menggunakan data test

**Tabel 4. 18 confusion matrix**

|                                |                               |
|--------------------------------|-------------------------------|
| <b>True Positive</b><br>16418  | <b>False Positive</b><br>1822 |
| <b>False Negative</b><br>17228 | <b>True Negatif</b><br>1049   |

#### 4.9 Pembahasan

Terdapat beberapa faktor yang mempengaruhi praproses dalam penelitian ini seperti menyingkat kata, kesalahan penulisan, penggunaan kata istilah yang tidak terdapat dalam kamus Bahasa Indonesia, baik dari Bahasa asing ataupun Bahasa daerah.

**Tabel 4. 19 Contoh kesalahan penulisan kata**

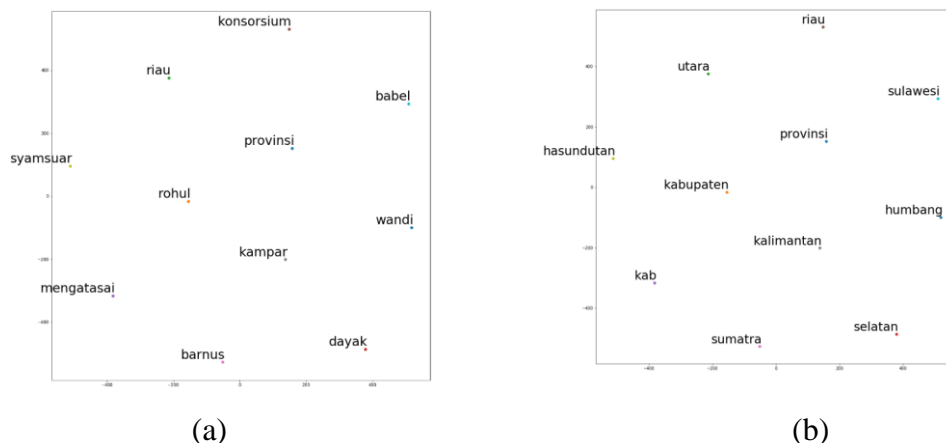
| Kata           |
|----------------|
| sandiyg        |
| payungrevolusi |
| sesatt         |
| kebnran        |
| kebnran        |
| klarga         |

Selain data, hasil proses stemming juga masih memiliki beberapa kekurangan. Nazief dan Adriani menjelaskan bahwa algoritma yang dibuat masih ada kekurangan seperti Word-Sense Ambiguity dimana kesulitan melakukan stemming terhadap kata yang mempunyai beberapa kemungkinan stem contohnya berikan menjadi ber-ikan atau beri-kan, berdesakan menjadi ber-desa-kan atau ber-desak-an. Pengguna Bahasa Indonesia tidak konsisten dalam menentukan stem (secara manual), seperti apakah "adalah" menjadi "ada", atau "sebagai" menjadi "bagai". Soal ini tidak bisa diselesaikan, karena kita tidak sepakat terhadap hasilnya. Akronim dan Proper Noun : seperti pemilu harusnya tidak distem menjadi pe-milu, dan nama orang Abdullah seharusnya tidak distem menjadi abdul. menurut mereka persoalan itu di luar fungsi stemmer. Fungsi untuk mengenali proper noun

berada pada komponen NLP lainnya yaitu NER (Named Entity Recognizer). Sebelum distem, kata tersebut dikenali dulu apakah merupakan Proper Noun. Jika merupakan proper noun, tidak perlu distem (Adriani et al., 2007).

Skipgram dan CBOW memiliki kemiripan hasil, yang membedakan keduanya adalah CBOW dipengaruhi oleh frekuensi kata tersebut muncul. Sedangkan Skipgram tidak dipengaruhi oleh frekuensi kemunculan. seperti yang dapat dilihat pada gambar 4.3 terdapat kata-kata seperti “babel” atau “konsorsium” yang di dalam dataset jarang muncul dalam kalimat yang sama dengan “provinsi”, bahkan kata-kata tersebut hanya muncul sebanyak enam kali. Namun dalam Skipgram kata-kata tersebut dianggap memiliki kedekatan dengan “provinsi”. Sedangkan CBOW hasilnya dipengaruhi oleh frekuensi kemunculan kata. Seperti yang dapat dilihat pada gambar 4.87 semua kata-kata yang muncul disekitar kata “provinsi” adalah kata-kata yang sering muncul dalam dataset dan sering muncul bersamaan dengan kata “provinsi”. Skipgram memiliki kelebihan terhadap kata yang jarang muncul karena tidak terpengaruh oleh frekuensi kemunculan sehingga cocok digunakan untuk dataset yang terbatas. Kekurangannya adalah Skipgram membutuhkan waktu pembelajaran yang lebih lama dari CBOW. CBOW memiliki kelebihan dalam waktu pembelajaran yang lebih cepat. Kekurangannya adalah hasil dari CBOW dipengaruhi oleh frekuensi sehingga ketika kata tersebut jarang muncul akan berpengaruh terhadap nilai vector. ELMo memiliki kelebihan dalam menentukan konteks dari suatu kata karena memiliki lebih dari satu vector. Kekurangannya adalah waktu pembelajaran yang lebih lama dari Skipgram dan CBOW dan dikarenakan ELMo masih baru sehingga masih sedikit *library* yang *support*.





Gambar 4. 73 Contoh perbandingan hasil (a) Skipgram dan (b) CBOW

Dari semua percobaan yang dilakukan percobaan GRU-Skipgram adalah percobaan dengan F1-score terbesar dengan nilai 94.67%. dan recall terbesar adalah GRU-Skipgram dengan nilai recall 99.09%. GRU-CBOW memiliki precision terbesar dengan nilai 90.96%. Selisih hasil tiap percobaan sangat kecil hal ini dapat dilihat di tabel 4.23.

Tabel 4. 20 Tabel Hasil pelatihan Model

| Model         | Precision | recall | F1-score | Accuracy |
|---------------|-----------|--------|----------|----------|
| GRU-CBOW      | 90.69     | 98.67  | 94.51    | 94.27    |
| GRU-Skipgram  | 90.64     | 99.09  | 94.67    | 94.43    |
| GRU_ELMo      | 43.14     | 00.93  | 09.46    | 49.85    |
| LSTM-CBOW     | 90.63     | 98.74  | 94.51    | 94.26    |
| LSTM-Skipgram | 90.61     | 99.04  | 94.64    | 94.39    |
| LSTM-ELMo     | 36.53     | 05.73  | 09.92    | 47.83    |

Hasil test yang dilakukan menunjukkan model cukup baik dalam membedakan kata-kata yang jelas hate speech dan non hate speech . tabel 4.15 contoh hasil test yang berhasil membedakan *hate speech* dan *non hate speech*

Tabel 4. 21 contoh klasifikasi benar

| Status Tweet   | Model     |               |           |             |                |             | Label |
|--|-----------|---------------|-----------|-------------|----------------|-------------|-------|
|  | GRU CBO W | GRU Skipgra m | GRU ELM o | LST M CBO W | LSTM Skipgra m | LST M ELM o |       |
| Kereta MRT Jakarta telah hadir di ibu kota, membawa budaya baru transportasi Indonesia yang cepat, nyaman, dan terintegrasi dengan moda lainnya. Mari, manfaatkan dan jaga bersama-sama. Jangan buang sampah | NH        | NH            | NH        | NH          | NH             | NH          | NH    |

|   |    |    |    |    |    |    |    |
|---|----|----|----|----|----|----|----|
| di MRT dan stasiun, biasakan tertib dan antre, jangan berdesak-desakan.   |    |    |    |    |    |    |    |
| Haini gobloknya nular sampe ke Wakil Ketua DPR.. Ternyata virus bodoh cepet menjalar di kaum akal sehat.  | HS | HS | HS | HS | HS | HS | HS |
| COBLOS 02 PRABOWO-SANDI ikut merayakan kebahagiaan #12HariLagiLebaranAkalSehat  | NH | NH | NH | NH | NH | NH | NH |
| disini tempat sampah..isinya kan sampah pendukungnya juga sampah yg didukung ya mirip sampah masyarakat   | HS | HS | HS | HS | HS | HS | HS |
| Contohnya Jokowi aktif update vlog yang menampilkan aktifitas Jokowi sebagai presiden bersama pemimpin dunia seperti Raja Salman #SelaluOptimis | NH | NH | NH | NH | NH | NH | NH |

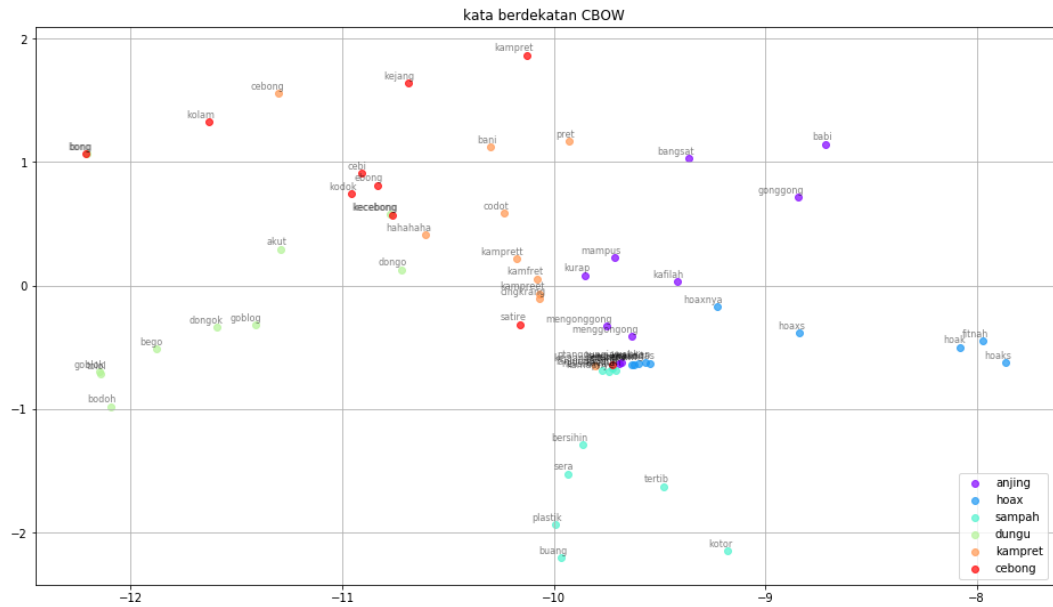
Hasil test yang dilakukan semua model beberapa kali sulit membedakan *non hate speech* yang mengandung kata kasar atau kata-kata yang sering muncul dalam hate speech seperti “sampah”, “bego” atau ”anjing”. Terdapat beberapa kalimat yang seharusnya termasuk hate speech namun ketika di test menghasilkan non hate speech. Berikut adalah contoh prediksi salah yang dilakukan oleh model.

**Tabel 4. 22 contoh klasifikasi salah**

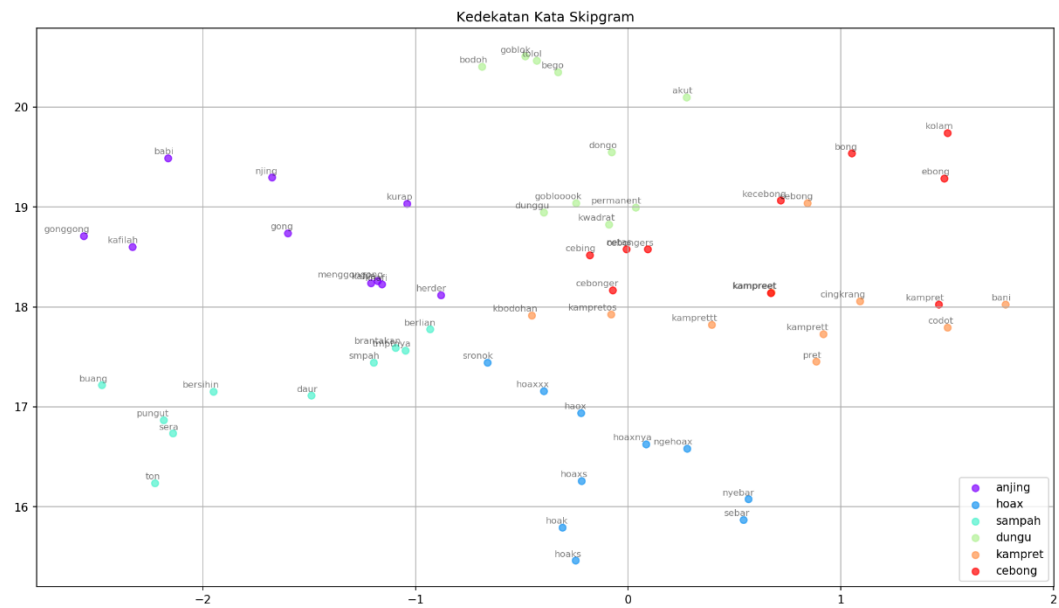
| Status Tweet  | Model    |              |          |           |               |           | Label |
|---|----------|--------------|----------|-----------|---------------|-----------|-------|
|   | GRU CBOw | GRU Skipgram | GRU ELMo | LSTM CBOw | LSTM Skipgram | LSTM ELMo |       |
| Katanya hoaks jgn dirituit.   | HS       | HS           | HS       | HS        | HS            | HS        | NH    |
| Hati yg bersih menjadikan otak cerdas..serta mempunyai akal sehat...Prabowo Sandi.. | HS       | HS           | HS       | HS        | HS            | HS        | NH    |
| Mantap, Akal Sehat pilih No 02.   | HS       | HS           | HS       | HS        | HS            | HS        | NH    |
| Itu yg ngumpul manusia atau monyet sih?   | NH       | NH           | HS       | NH        | NH            | HS        | HS    |
| ada kamprettt kesasar,, wkwkwk  | NH       | NH           | HS       | NH        | NH            | HS        | HS    |

Dalam penelitian juga ditemukan bahwa selama kampanye pemilihan presiden 2019 orang-orang di twitter dalam melakukan hate speech menyamakan orang lain dengan hewan seperti “anjing”, “cebong”, “kampret”, ”babi” ataupun

merendahkan orang lain dengan kata-kata seperti “bego”, “dungu”, “tolol,” dan “sampah”. Dimana dari kata-kata dijadikan sebagai subjek ataupun objek. Berikut adalah enam kata kasar yang memiliki frekuensi muncul tertinggi dalam hate speech dan kata-kata yang ada didekatnya berdasarkan *word embedding*.



Gambar 4. 74 persebaran kata kasar pada CBOW



Untuk mendapatkan pola setiap kata diubah menjadi POS *Tagger*. Dari data praproses berlabel *hate speech* diperoleh pola seperti pada tabel 4.23. Dari penelitian dapat dilihat bahwa pola *hate speech* berisi kombinasi kata benda, kata kerja dan kata sifat. Berikut adalah pola hate speech yang terbentuk.

**Tabel 4. 23 Pola Hate speech**

| Pola   | contoh  |
|--|---|
| ['NN', 'NN', 'NN', 'NN', 'NN', 'NN', 'JJ', 'NN', 'NN', 'VB'] | [('jokowi', 'NN'), ('kelana', 'NN'), ('langgar', 'NN'), ('ham', 'NN'), ('sembunyi', 'NN'), ('malu', 'NN'), ('parah', 'JJ'), ('prabowo', 'NN'), ('buta', 'NN'), ('bayar', 'VB')] |

Ketika pola pada data test *non hate speech* hasilnya menunjukkan tidak ada *tweet* yang mempunyai pola hate speech. Hasil penelitian juga menunjukkan bahwa *non hate speech* memiliki pola yang tidak terdapat pada *hate speech* dari penelitian bahwa hate speech dan non hate speech mempunyai pola yang berbeda. Berikut adalah contoh pola yang hanya terdapat dapa *non hate speech*.

**Tabel 4. 24 Pola Non Hate Speech**

| pola                                 | contoh   |
|--------------------------------------|--|
| ['JJ', 'NN', 'NNP', 'NN', 'JJ']      | [('maju', 'JJ'), ('jokowi', 'NN'), ('indonesia', 'NNP'), ('negara', 'NN'), ('modern', 'JJ')]             |
| ['VB', 'VB', 'NN', 'NN', 'NN', 'JJ'] | [('terima', 'VB'), ('kasih', 'VB'), ('awal', 'NN'), ('realisasi', 'NN'), ('mrt', 'NN'), ('hebat', 'JJ')] |

## **BAB V**

### **Kesimpulan**

Pada bab ini akan dijelaskan kesimpulan dari hasil dan pembahasan yang telah dilakukan sebelumnya. Selain itu pada bab ini juga dijelaskan saran sebagai kajian bagi peneliti selanjutnya untuk pengembangan lebih lanjut.

#### **5.1 Kesimpulan**

Berdasarkan hasil uji coba dan pembahasan, maka dapat ditarik beberapa kesimpulan sebagai berikut. Pengembangan model deteksi hate speech yang diusulkan terbukti dapat digunakan dalam mendeteksi hate speech, khususnya pada situs jejaring sosial Twitter. Terdapat beberapa faktor yang mempengaruhi hasil dalam penelitian ini seperti menyingkat kata, kesalahan penulisan, penggunaan kata istilah yang tidak terdapat dalam kamus Bahasa Indonesia, baik dari Bahasa asing ataupun Bahasa daerah.

Setiap metode metode *word embedding* memiliki kelebihan dan kekurangan masing-masing. Skipgram baik digunakan untuk data yang minim karena tidak memperhatikan frekuensi kemunculan. Namun untuk data yang banyak sebaiknya menggunakan CBOW karena kemampuan yang lebih cepat dalam proses pembelajaran. Dari penelitian yang dilakukan ELMo memiliki kemampuan pemahaman konteks yang baik karena setiap kata dapat memiliki vector lebih dari satu sehingga mampu membedakan fungsi dari kata tersebut berdasarkan posisinya.

Dari penelitian dapat dilihat bahwa pola *hate speech* berisi kombinasi kata benda, kata kerja dan kata sifat. Hasil penelitian juga menunjukkan bahwa *non hate speech* memiliki pola yang tidak terdapat pada *hate speech* dari penelitian bahwa hate speech dan non hate speech mempunyai pola yang berbeda.

Dari semua percobaan yang dilakukan percobaan ROC model yang menggunakan word embedding skip-gram dan CBOW memiliki nilai 0.94 dan model yang menggunakan ELMo memiliki ROC sebesar 0.49. GRU-Skipgram adalah percobaan dengan F1-score terbesar dengan nilai 94.67%. dan recall terbesar adalah GRU-Skipgram dengan nilai recall 99.09%. GRU-CBOW memiliki precision terbesar dengan nilai 90.96%.. ELMo memiliki performa terendah apabila

dibandingkan dengan word embedding yang lain. Dalam klasifikasi hate speech dibutuhkan precision yang tinggi sehingga GRU-CBOW dianggap baik untuk melakukan klasifikasi *hate speech*.

## **5.2 Saran**

Saran yang dapat diberikan berkaitan dengan hasil uji coba dan pembahasan pada penelitian ini adalah data yang digunakan sebaiknya menggunakan data yang lebih umum karena penelitian ini hanya mencakup status twitter selama masa kampanye pemilihan presiden 2019, Sehingga terdapat beberapa istilah yang hanya dipakai saat kampanye.

praproses data lebih dioptimalkan agar mengurangi *noise* data. Dan menggunakan metode stem yang lebih baik karena metode stem yang digunakan saat ini masih memiliki kekurangan.

## DAFTAR PUSTAKA

- Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M. M., & Williams, H. E. (2007). Stemming Indonesian. *ACM Transactions on Asian Language Information Processing*, 6(4), 1–33. <https://doi.org/10.1145/1316457.1316459>
- Alfina, I., Mulia, R., Fanany, M. I., & Ekanata, Y. (2017). Hate speech detection in the Indonesian language: A dataset and preliminary study. In *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* (pp. 233–238). IEEE. <https://doi.org/10.1109/ICACSIS.2017.8355039>
- Alorainy, W., Burnap, P., Liu, H., Javed, A., & Williams, M. L. (2018). Suspended Accounts: A Source of Tweets with Disgust and Anger Emotions for Augmenting Hate Speech Data Sample. In *2018 International Conference on Machine Learning and Cybernetics (ICMLC)* (pp. 581–586). IEEE. <https://doi.org/10.1109/ICMLC.2018.8527001>
- Amalia, N., & Andi, M. (2019). View of Strategi Komisi Pemilihan Umum dalam Mengurangi Angka Golput pada Pemilihan Presiden dan Wakil Presiden. Retrieved January 5, 2020, from <http://journal.ipm2kpe.or.id/index.php/JOPPAS/article/view/801/492>
- Arif, T. (2016). *PREDIKSI PERPINDAHAN PELANGGAN INDUSTRI TELEKOMUNIKASI SELULER MENGGUNAKAN KLASIFIKASI SENTIMEN PADA SITUS JEJARING SOSIAL TWITTER MENGGUNAKAN SUPPORT VECTOR MACHINE*. INSTITUT TEKNOLOGI SEPULUH NOPEMBER.
- Arsip, N. R. I. (2008). *UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 11 TAHUN 2008 TENTANG INFORMASI DAN TRANSAKSI ELEKTRONIK*. Retrieved from <https://www.anri.go.id/assets/download/97UU-Nomor-11-Tahun-2008-Tentang-Informasi-dan-Transaksi-Elektronik.pdf>
- AZALI, F. P. (2016). *KLASIFIKASI PENGADUAN MASYARAKAT BERBASIS SMS DENGAN METODE NAIVE BAYES CLASSIFIER*. Retrieved from

[http://etd.repository.ugm.ac.id/index.php?act=view&buku\\_id=95255&mod=penelitian\\_detail&sub=PenelitianDetail&typ=html](http://etd.repository.ugm.ac.id/index.php?act=view&buku_id=95255&mod=penelitian_detail&sub=PenelitianDetail&typ=html)

- Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017a). Deep Learning for Hate Speech Detection in Tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion* (pp. 759–760). New York, New York, USA: ACM Press. <https://doi.org/10.1145/3041021.3054223>
- Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017b). Deep Learning for Hate Speech Detection in Tweets. <https://doi.org/10.1145/3041021.3054223>
- Burnap, P., & Williams, M. L. (2015). Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making. *Policy & Internet*, 7(2), 223–242. <https://doi.org/10.1002/poi3.85>
- Burnap, P., & Williams, M. L. (2016). Us and them: identifying cyber hate on Twitter across multiple protected characteristics. *EPJ Data Science*, 5(1), 11. <https://doi.org/10.1140/epjds/s13688-016-0072-6>
- Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language, (Icwsn), 512–515. Retrieved from <http://arxiv.org/abs/1703.04009>
- Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., & Bhamidipati, N. (2015). Hate Speech Detection with Comment Embeddings. In *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion* (pp. 29–30). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2740908.2742760>
- Fatahillah, N. R., Suryati, P., & Haryawan, C. (2017a). Implementation of Naive Bayes classifier algorithm on social media (Twitter) to the teaching of Indonesian hate speech. In *2017 International Conference on Sustainable Information Engineering and Technology (SIET)* (pp. 128–131). IEEE. <https://doi.org/10.1109/SIET.2017.8304122>
- Fatahillah, N. R., Suryati, P., & Haryawan, C. (2017b). Implementation of Naive Bayes classifier algorithm on social media (Twitter) to the teaching of



- Indonesian hate speech. In *2017 International Conference on Sustainable Information Engineering and Technology (SIET)* (pp. 128–131). IEEE. <https://doi.org/10.1109/SIET.2017.8304122>
- Febriyan, B. (2017). TINDAK PIDANA KAMPANYE HITAM ( BLACK CAMPAIGN ) DALAM PENYELENGGARAAN PEMILIHAN KEPALA DAERAH WALIKOTA BANDA ACEH TAHUN 2017 PENDAHULUAN Pemilihan umum ( pemilu ) adalah salah satu cara dalam sitem demokrasi untuk memilih wakil-wakil rakyat yang akan dudu. *JIM Bidang Hukum Pidana*, 1(1), 54–62.
- Gagliardone, I., Gal, D., Alves, T., Martinez, G. (Martinez S., & Unesco. (2015). *Countering online hate speech*. United Nations Educational, Scientific and Cultural Organization. Retrieved from <https://unesdoc.unesco.org/ark:/48223/pf0000233231>
- Gao, L., & Huang, R. (2017). Detecting Online Hate Speech Using Context Aware Models. Retrieved from <http://arxiv.org/abs/1710.07395>
- Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., ... Zettlemoyer, L. (2018). *AllenNLP: A Deep Semantic Natural Language Processing Platform*. Retrieved from <http://allennlp.org/>
- Gröndahl, T., Juuti, M., Conti, M., & Asokan, N. (2018). *All You Need is &quot;Love&quot;: Evading Hate Speech Detection*. Retrieved from [www.echr.coe.int/Documents/FS](http://www.echr.coe.int/Documents/FS)
- Habibie, I. (2018). *Identifikasi Judul Berita Clickbait Berbahasa Indonesia dengan Algoritma Long Short Term Memory (LSTM) Recurrent Neural Network* Habibie, Ibnu. Medan. Retrieved from <https://repositori.usu.ac.id/handle/123456789/8874>
- Handono, P. Y. (2019). Gaya Bahasa Komentar Dalam Akun Instagram “Mimi Peri Rapunchelle.” *Linguista: Jurnal Ilmiah Bahasa, Sastra, Dan Pembelajarannya*, 2(2), 97–105. Retrieved from <http://e-journal.unipma.ac.id/index.php/linguista/article/view/3697/1980>
- Ibrohim, M. O., & Budi, I. (2018). A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media. *Procedia Computer Science*,

135, 222–229. <https://doi.org/10.1016/J.PROCS.2018.08.169>

- Irsyad, A., & Rakhmawati, N. A. (2019). Community detection in twitter based on tweets similarities in indonesian using cosine similarity and louvain algorithms. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 6(1), 22. <https://doi.org/10.26594/register.v6i1.1595>
- Jeff, H. (2015). *Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks*:: 9781505714340: Amazon.com: Books (3rd ed.). CreateSpace Independent Publishing Platform. Retrieved from <https://www.amazon.com/Artificial-Intelligence-Humans-Learning-Networks/dp/1505714346>
- KEPOLISIAN NEGARA REPUBLIK INDONESIA. (2015). *Surat Edaran Kepala Kepolisian Negara Nomor SE/6/X/2015 Tahun 2015*. Retrieved from <https://cdn.bidhuan.id/img/2015/11/download-287674212-Se-Hate-Speech-1.pdf>
- Klubička, F., & Fernández, R. (2018). Examining a hate speech corpus for hate speech detection and popularity prediction. Retrieved from <http://arxiv.org/abs/1805.04661>
- Malmasi, S., & Zampieri, M. (2017). Detecting Hate Speech in Social Media. Retrieved from <http://arxiv.org/abs/1712.06427>
- Malmasi, S., & Zampieri, M. (2018). Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2), 187–202. <https://doi.org/10.1080/0952813X.2017.1409284>
- Martins, R., Gomes, M., Almeida, J. J., Novais, P., & Henriques, P. (2018a). Hate Speech Classification in Social Media Using Emotional Analysis. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)* (pp. 61–66). IEEE. <https://doi.org/10.1109/BRACIS.2018.00019>
- Martins, R., Gomes, M., Almeida, J. J., Novais, P., & Henriques, P. (2018b). Hate Speech Classification in Social Media Using Emotional Analysis. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)* (pp. 61–66). IEEE. <https://doi.org/10.1109/BRACIS.2018.00019>
- Meyer, D. (2016). *How exactly does word2vec work?* Retrieved from [88](http://www.1-</a></p></div><div data-bbox=)

4-5.net/~dmm/ml/how\_does\_word2vec\_work.pdf

- Michael Nguyen. (2018). Illustrated Guide to LSTM's and GRU's: A step by step explanation. Retrieved May 4, 2019, from <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Distributed Representations of Words and Phrases and their Compositionality*.
- Miró-Llinares, F., Moneva, A., & Esteve, M. (2018). Hate is in the air! But where? Introducing an algorithm to detect hate speech in digital microenvironments. *Crime Science*, 7(1), 15. <https://doi.org/10.1186/s40163-018-0089-1>
- Mishra, P., Del Tredici, M., Yannakoudakis, H., & Shutova, E. (2019). Author Profiling for Hate Speech Detection. Retrieved from <http://arxiv.org/abs/1902.06734>
- NURIL, I. (2019). *PENCEGAHAN KAMPANYE HITAM (BLACK CAMPAIGN) DALAM PEMILIHAN PRESIDEN 2019 DI KABUPATEN TULUNGAGUNG BERDASARKAN PERSPEKTIF HUKUM POSITIF DAN HUKUM ISLAM*. IAIN Tulungagung.
- Olah, C. (2015). Understanding LSTM Networks. Retrieved May 4, 2019, from <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Pelle, R., Alcântara, C., & Moreira, V. P. (2018). A Classifier Ensemble for Offensive Text Detection. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web - WebMedia '18* (pp. 237–243). New York, New York, USA: ACM Press. <https://doi.org/10.1145/3243082.3243111>
- Perone, C. S., Silveira, R., & Paula, T. S. (2018). *Evaluation of sentence embeddings in downstream and linguistic probing tasks*. Retrieved from <https://arxiv.org/pdf/1806.06259.pdf>
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. Retrieved from <http://arxiv.org/abs/1802.05365>
- Pitsilis, G. K., Ramampiaro, H., & Langseth, H. (2018). Effective hate-speech detection in Twitter data using recurrent neural networks. *Applied Intelligence*,

- 48(12), 4730–4742. <https://doi.org/10.1007/s10489-018-1242-y>
- Powers, D. M. W. (2011). EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION, 2(1), 37–63. Retrieved from <http://www.bioinfo.in/contents.php?id=51>
- Prabowo, Y. D., Marselino, T. L., & Suryawiguna, M. (2019). Pembentukan Vector Space Model Bahasa Indonesia Menggunakan Metode Word to Vector. *Jurnal Buana Informatika*, 10(1), 29–40. Retrieved from <http://ojs.uajy.ac.id/index.php/jbi/article/view/2053/1348>
- Pratiwi, N. I., Budi, I., & Alfina, I. (2018). Hate Speech Detection on Indonesian Instagram Comments using FastText Approach. In *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* (pp. 447–450). IEEE. <https://doi.org/10.1109/ICACSIS.2018.8618182>
- Putra, M. R. A., Djamal, E. C., & Ilyas, R. (2018). Brain Computer Interface untuk Menggerakkan Robot Menggunakan Recurrent Neural Network. *Prosiding Seminar Nasional Rekayasa Teknologi Informasi / SNARTISI*, 1(1). Retrieved from <http://e-jurnal.lppmunsera.org/index.php/snartisi/article/view/829/743>
- Raufi, B., & Xhaferri, I. (2018). Application of Machine Learning Techniques for Hate Speech Detection in Mobile Applications. In *2018 International Conference on Information Technologies (InfoTech)* (pp. 1–4). IEEE. <https://doi.org/10.1109/InfoTech.2018.8510738>
- Rozi, I. F. (2013). Implementasi Rule-Based Document Subjectivity Pada Sistem Opinion Mining. *Jurnal ELTEK*, Vol 11(No 01), 29–41.
- Ruwandika, N. D. T., & Weerasinghe, A. R. (2018). Identification of Hate Speech in Social Media. In *2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer)* (pp. 273–278). IEEE. <https://doi.org/10.1109/ICTER.2018.8615517>
- Sa'diyah, Y. S. H. (2017). KLASIFIKASI DATA TWITTER UNTUK MENDETEKSI TWEET BLACK CAMPAIGN MENGGUNAKAN SUPPORT VECTOR MACHINE. *Undergraduate (S1) Thesis, University of*

*Muhammadiyah Malang.*

- Sahi, H., Kilic, Y., & Saglam, R. B. (2018). Automated Detection of Hate Speech towards Woman on Twitter. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)* (pp. 533–536). IEEE. <https://doi.org/10.1109/UBMK.2018.8566304>
- Salminen, J., Almerexhi, H., Milenković, M., Jung, S.-G., An, J., Kwak, H., & Jansen, B. J. (2018). Anatomy of Online Hate: Developing a Taxonomy and Machine Learning Models for Identifying and Classifying Hate in Online News Media. *Proceedings of the Twelfth International AAAI Conference on Web and Social Media, (Icwsn)*, 330–339. Retrieved from [www.aaai.org](http://www.aaai.org)
- Santosh, T. Y. S. S., & Aravind, K. V. S. (2019). Hate Speech Detection in Hindi-English Code-Mixed Social Media Text. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data - CoDS-COMAD '19* (pp. 310–313). New York, New York, USA: ACM Press. <https://doi.org/10.1145/3297001.3297048>
- Saputra, N., Bharata Adji, T., & Erna Permanasari, A. (2015). ANALISIS SENTIMEN DATA PRESIDEN JOKOWI DENGAN PREPROCESSING NORMALISASI DAN STEMMING MENGGUNAKAN METODE NAIVE BAYES DAN SVM. *Jurnal Dinamika Informatika*, 5(1). Retrieved from <http://ojs.upy.ac.id/ojs/index.php/dinf/article/view/113/82>
- Sazany, E., & Budi, I. (2018). Deep Learning-Based Implementation of Hate Speech Identification on Texts in Indonesian: Preliminary Study. In *2018 International Conference on Applied Information Technology and Innovation (ICAITI)* (pp. 114–117). IEEE. <https://doi.org/10.1109/ICAITI.2018.8686725>
- Simeon Kostadinov. (2017). Understanding GRU Networks. Retrieved May 4, 2019, from <https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be>
- Sutejo, T. L., & Lestari, D. P. (2018). Indonesia Hate Speech Detection Using Deep Learning. In *2018 International Conference on Asian Language Processing (IALP)* (pp. 39–43). IEEE. <https://doi.org/10.1109/IALP.2018.8629154>
- Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text

- classification. *Information Processing & Management*, 50(1), 104–112.  
<https://doi.org/10.1016/j.ipm.2013.08.006>
- Watanabe, H., Bouazizi, M., & Ohtsuki, T. (2018a). Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. *IEEE Access*, 6, 13825–13835.  
<https://doi.org/10.1109/ACCESS.2018.2806394>
- Watanabe, H., Bouazizi, M., & Ohtsuki, T. (2018b). Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. *IEEE Access*, 6, 13825–13835.  
<https://doi.org/10.1109/ACCESS.2018.2806394>
- we are social. (2019). *Digital 2019 Indonesia (January 2019) v01. we are social*. Retrieved from <https://www.slideshare.net/DataReportal/digital-2019-indonesia-january-2019-v01>
- Widyastuti, N. N., Bijaksana, A. A., & Sardi, I. L. (2018). Analisis Word2vec untuk Perhitungan Kesamaan Semantik antar Kata. *EProceedings of Engineering*, 5(3). Retrieved from <https://libraryeproceeding.telkomuniversity.ac.id/index.php/engineering/article/view/7263/7144>
- Yusuf Nur dan Diaz Santika, M. D. (2011). *ANALISIS SENTIMEN PADA DOKUMEN BERBAHASA INDONESIA DENGAN PENDEKATAN SUPPORT VECTOR MACHINE*. *Konferensi Nasional Sistem dan Informatika*. Retrieved from <http://situs.com>
- Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Predicting the Type and Target of Offensive Posts in Social Media. Retrieved from <http://arxiv.org/abs/1902.09666>
- Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network (pp. 745–760). Springer, Cham. [https://doi.org/10.1007/978-3-319-93417-4\\_48](https://doi.org/10.1007/978-3-319-93417-4_48)

## LAMPIRAN

### Lampiran 1.

|    | author                          | title   | publish    | year |
|----|---------------------------------|---|------------|------|
| 1  | Santosh & Aravind               | Hate Speech Detection in Hindi-English Code-Mixed Social Media Text   | Conference | 2019 |
| 2  | Zampieri et al.                 | Predicting the Type and Target of Offensive Posts in Social Media   | Conference | 2019 |
| 3  | Pratiwi et al.                  | Hate Speech Detection on Indonesian Instagram Comments using FastText Approach  | Conference | 2018 |
| 4  | Sutejo & Lestari                | Indonesia Hate Speech Detection using Deep Learning   | Conference | 2018 |
| 5  | Sahi et al.                     | Automated Detection of Hate Speech towards Woman on Twitter   | Conference | 2018 |
| 6  | Ruwandika , Weerasinghe         | Identification of Hate Speech in Social Media   | Conference | 2018 |
| 7  | Watanabe, Bouazizi, & Ohtsuki   | Hate Speech on Twitter A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection          | journal    | 2018 |
| 8  | Martins et al.,                 | Hate speech classification in social media using emotional analysis   | Conference | 2018 |
| 9  | Pitsilis et al.                 | Effective hate-speech detection in Twitter data using recurrent neural networks   | journal    | 2018 |
| 10 | Burnap & Williams               | Us and them: identifying cyber hate on Twitter across multiple protected characteristics  | journal    | 2016 |
| 11 | Miró-Llinares et al.            | Hate is in the air! But where? Introducing an algorithm to detect hate speech in digital microenvironments                          | journal    | 2018 |
| 12 | Ibrohim & Budi                  | A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media   | Conference | 2018 |
| 13 | Pete Burnap Matthew L. Williams | Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making      | Conference | 2018 |
| 14 | Salminen et al.                 | Anatomy of Online Hate: Developing a Taxonomy and Machine Learning Models for Identifying and Classifying Hate in Online News Media | Conference | 2018 |
| 15 | Malmasi & Zampieri              | Detecting Hate Speech in Social Media   | Conference | 2017 |
| 16 | Mishra et al.                   | Author Profiling for Hate Speech Detection  | journal    | 2019 |
| 17 | Pelle et al.                    | A Classifier Ensemble for Offensive Text Detection  | Conference | 2018 |
| 18 | Zhang et al.                    | Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network  | Conference | 2018 |
| 19 | Gröndahl et al.                 | All You Need is "Love": Evading Hate Speech Detection   | Conference | 2018 |
| 20 | Klubička , Fernández            | Examining a hate speech corpus for hate speech detection and popularity prediction  | Conference | 2018 |
| 21 | Alorainy et al.                 | Suspended Accounts: A Source of Tweets with Disgust and Anger Emotions for Augmenting Hate Speech Data Sample                       | Conference | 2018 |

|    |                    |  |            |      |
|----|--------------------|--|------------|------|
| 22 | Raufi , Xhaferri   | Application of Machine Learning Techniques for Hate Speech Detection in Mobile Applications                            | Conference | 2018 |
| 23 | Fatahillah et al.  | Implementation Of Naive Bayes Classifier Algorithm On Social Media (Twitter) To The Teaching Of Indonesian Hate Speech | Conference | 2017 |
| 24 | Alfina et al       | Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study                                      | Conference | 2017 |
| 25 | Sazany & Budi      | Deep Learning-Based Implementation of Hate Speech Identification on Texts in Indonesian: Preliminary Study             | Conference | 2018 |
| 26 | Malmasi & Zampieri | Challenges in discriminating profanity from hate speech  | journal    | 2018 |
| 27 | Davidson et al.    | Automated Hate Speech Detection and the Problem of Offensive Language  | Conference | 2017 |
| 28 | Gao & Huang        | Detecting Online Hate Speech Using Context Aware Models  | Conference | 2017 |
| 29 | Badjatiya et al.   | Deep Learning for Hate Speech Detection in Tweets  | Conference | 2017 |
| 30 | Djuric et al.      | Hate Speech Detection with Comment Embeddings  | Conference | 2015 |



## DAFTAR RIWAYAT HIDUP

Nama Lengkap: Sukmawati, S.Pd., M.Hum.

Pos-el : [sukmagemini78@yahoo.com](mailto:sukmagemini78@yahoo.com)

Alamat Kantor: Kantor Bahasa Sulawesi Tenggara, Jalan Haluoleo, Kompleks  
Bumi Praja, Anduonohu, Kendari

Riwayat Pekerjaan:

- a. Staf Teknis Balai Bahasa Palu (2003—2004)
- b. Staf teknis Kantor Bahasa Sulawesi Tenggara (2004—sekarang)

Riwayat Pendidikan:

- a. S-1 Jurusan Pendidikan Bahasa dan Sastra Indonesia dan Daerah di Universitas Negeri Makassar (2001)
- b. S-2 Jurusan Bahasa Indonesia di Universitas Hasanuddin (2011)

Informasi Lain:

- a. Peneliti bahasa (sosiolingustik)
- b. Penyuluh bahasa
- c. Penyunting bahasa

## DAFTAR RIWAYAT HIDUP

### A. Data Diri

Nama Lengkap : Rizal Adi Saputra  
Gelara : ST, M.Kom  
Tempat / Tanggal Lahir: Kendari, 6 April 1991  
Jenis Kelamin : Laki-Laki  
Agama : Islam  
Status Perkawinan : Belum Menikah  
Alamat : Jalan Tungala IIC No.3, Kel. Anawai, Kec. Wua Wua  
Kota Kendari, Sulawesi Tenggara, 93116  
Nomor Ponsel : 0852 4179 2695  
E-mail : [rizaladisaputra@uho.ac.id](mailto:rizaladisaputra@uho.ac.id)

### Riwayat Pendidikan Formal

2013 – 2015 : Institut Teknologi Sepuluh Nopember (ITS), Surabaya  
Fakultas Teknologi Informasi, Prodi S2 Teknik Informatika  
IPK 3,92  
2008 – 2012 : Universitas Halu Oleo, Kendari Fakultas Teknik, Jurusan  
S1 Teknik Informatika IPK 3,85  
2005 – 2008 : SMA Negeri 4 Kendari, Jurusan IPA  
2002 – 2005 : SMP Negeri 1 Andoolo, Konawe Selatan  
1996 – 2002 : SD Negeri Silea Jaya, Kecamatan Buke, Konawe Selatan

## **(Curriculum Vitae)**

Nama : Eka Kurnia Syawaliyah Subagio  
Tempat/ Tanggal Lahir : Jayapura, 14 Februari 1998  
Alamat : Jalan Anawai, Kendari Sulawesi Tenggara  
Nomor HP : 082396560191  
Jenis Kelamin : Perempuan  
Agama : Islam  
Kewarganegaraan : Indonesia  
Status : Belum Menikah  
Email : ekakurnia662@gmail.com  
Hobi : Jalan-jalan, nonton

### **PENDIDIKAN FORMAL**

- Tahun 2009 SDN 11 Baruga
- Tahun 2009 – 2012 SMPN 4 Kendari
- Tahun 2012 – 2015 SMAN 5 Kendari

### **PENDIDIKAN NON FORMAL**

- 2008 English Course
- 2013-2015 Bimbingan Belajar di Primagama
- 2018 English Course ECC 1-3
- 2018 Magang Kerja di Harian Berita Kota Kendari
- 2019 Pelatihan Entrepreneurship

### **PENGALAMAN ORGANISASI**

- Anggota OSIS SMAN 5 Kendari periode 2012 – 2013 dan berlanjut 2013 - 2014
- Anggota BEM FISIP UHO periode 2017-2018
- Volunteer di Kelas Inspirasi Kendari tahun 2019
- Miss Hijab Sulawesi Tenggara 2018

## DAFTAR RIWAYAT HIDUP

### I. Data Pribadi

1. Nama : Fauqi Nur Ilmih  
2. Tempat dan Tanggal Lahir : Lamongan, 22 Juni 1997  
3. Jenis Kelamin : Perempuan  
4. Agama : Islam  
5. Warga Negara : Indonesia  
6. Alamat KTP : Lingkungan Topa, RT 002/RW 003  
,Kel. Lowu-Lowu, Kec. Lea-Lea, Kota  
Baubau  
7. Alamat Sekarang : Jalan Prof. Dr. Abdurrauf Tarimana  
(Lorong Merpati 2), Kambu, Kota  
Kendari  
8. Nomor Telepon / HP : 0851 4527 2647  
9. e-mail : uqhyki@gmail.com

### II. Pendidikan Formal :

| Periode<br>(Tahun) |   |      | Sekolah / Universitas | Jurusan                         |
|--------------------|---|------|-----------------------|---------------------------------|
| 2003               | - | 2006 | SDN 02 TEMPUR KEMIRI  | -                               |
| 2006               | - | 2009 | SDN 03 LOWU-LOWU      | -                               |
| 2009               | - | 2012 | SMPN 08 BAUBAU        | -                               |
| 2012               | - | 2015 | SMAN 05 BAUBAU        | IPA                             |
| 2015               |   |      | UNIVERSITAS HALU OLEO | S1 Program Studi<br>Jurnalistik |

### III. Pendidikan Non Formal – Seminar/ Organisasi

| Tahun | Seminar/ Organisasi  |
|-------|--|
| 2015  | Peserta Studi Al-Qur'an Intensif VI  |
| 2017  | Peserta Workshop dan Pencarian bakat dalam Media Penyiaran Radio & TV                              |
| 2017  | Finalis 12 Besar Lomba News Presenter & Reporter Communication Day 2017                            |
| 2018  | Anggota BEM FISIP UHO  |
| 2018  | Peserta Youth Town Hall “Discussion, Collaboration, & Action”                                      |
| 2018  | Peserta Seminar Politik Media “Kekuatan Ekonomi Politik Media menjelang Pemilihan Gubernur Sultra” |
| 2019  | Terbaik II Putri Duta Bahasa Sulawesi Tenggara   |

## Curriculum Vitae

Nama Lengkap : Mustika Sari  
Tempat Lahir : Sidodadi  
Tanggal Lahir : 18 Mei 1999  
Email : [mustikasari578@gmail.com](mailto:mustikasari578@gmail.com)

### Pendidikan

| No | Jenjang | Prodi              | Spesialisasi | Nama Sekolah       | Tahun Lulus |
|----|---------|--------------------|--------------|--------------------|-------------|
| 1  | SD      |                    |              | SD Negeri 4 Katobu | 2011        |
| 2  | SMP     |                    |              | SMP Negeri 2 Raha  | 2014        |
| 3  | SMA     |                    |              | SMK 4 Unaaha       | 2017        |
| 4  | UHO     | Teknik Informatika |              | UHO                | -           |

### Pengalaman Kerja

| No | Perusahaan                                     | Tahun                   |
|----|--|-------------------------|
| 1  | Technos Studio - Content Writer                | September 2019-sekarang |
| 2  | Asisten Lab Science and Artificial Inteligence | Oktober 2019-2018       |

### Pengalaman organisasi

| No | Perusahaan/Organisasi   | Tahun |
|----|---|-------|
| 1  | Kelas Inspirasi Kendari (Divisi Sosial Media)                               | 2018  |
| 2  | Himpunan Mahasiswa Teknik Informatika (Kementrian Sumberdaya dan Penalaran) | 2018  |
| 3  | Generasi Baru Indonesia (GenBI)   | 2018  |

## Curriculum Vitae

Nama Lengkap : Anggi Jolanda Limbong  
Tempat Lahir : Kendari  
Tanggal Lahir : 28 Juli 2000  
Email : [jolandaanggi@gmail.com](mailto:jolandaanggi@gmail.com)

### Pendidikan

| No | Jenjang | Prodi              | Spesialisasi | Nama Sekolah         | Tahun Lulus |
|----|---------|--------------------|--------------|----------------------|-------------|
| 1  | SD      |                    |              | SD Negeri 10 baruga  | 2011        |
| 2  | SMP     |                    |              | SMP Negeri 4 Kendari | 2014        |
| 3  | SMA     |                    |              | SMA 5 Kendari        | 2017        |
| 4  | UHO     | Teknik Informatika |              | UHO                  | -           |

### Pengalaman Kerja

| No | Perusahaan/Organisaisi                         | Tahun             |
|----|--|-------------------|
| 1  | Asisten Lab Science and Artificial Inteligence | Oktober 2019-2018 |

### Pengalaman organisasi

| No | Perusahaan/Organisasi   | Tahun |
|----|---|-------|
| 1  | Himpunan Mahasiswa Teknik Informatika (Kementrian Sumberdaya dan Penalaran) | 2018  |

## Curriculum Vitae

Nama Lengkap : Nadhif Hawari Nasrul

Email : [Nadhif.mia4@gmail.com](mailto:Nadhif.mia4@gmail.com)

### Pendidikan

| No | Jenjang | Prodi              | Spesialisasi | Nama Sekolah         | Tahun Lulus |
|----|---------|--------------------|--------------|----------------------|-------------|
| 2  | SMP     |                    |              | SMP Negeri 9 Kendari | 2014        |
| 3  | SMA     |                    |              | SMK 4 Kendari        | 2017        |
| 4  | UHO     | Teknik Informatika |              | UHO                  | -           |

### Pengalaman Kerja

| No | Perusahaan                                     | Tahun             |
|----|--|-------------------|
| 1  | Web Project Leader- Aksarsa Education Centre   | 2018              |
| 2  | Asisten Lab Science and Artificial Inteligence | Oktober 2019-2018 |

### Pengalaman organisasi

| No | Perusahaan/Organisasi                 | Tahun |
|----|---------------------------------------|-------|
| 2  | Himpunan Mahasiswa Teknik Informatika | 2018  |

*(Halaman sengaja dikosongkan)*