Deteksi Ujaran Kebencian pada Teks Bahasa Indonesia Menggunakan Bidirectional Long Short Term Memory (Bi-LSTM)

Aditya Perwira Joan Dwitama
20917035

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Sains Data
Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia
2023

Lembar Pengesahan Pembimbing

Deteksi Ujaran Kebencian pada Teks Bahasa Indonesia Menggunakan Bidirectional Long Short Term Memory (Bi-LSTM)

Aditya Perwira Joan Dwitama
20917035

Yogyakarta, Maret 2023

{Jika terdapat dua pembimbing yang telah bergelar doktor, tuliskan pembimbing pertama di sebelah kiri, Anda dapat menggunakan tabel yang tersedia sebagai pemisah bantu. Pembimbing yang belum bergelar doktor tidak perlu ditulis di sini}

Pembimbing 1	Pembimbing 2
Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.	Dr. Syarif Hidayat, S.Kom., M.I.T.

Lembar Pengesahan Penguji

Deteksi Ujaran Kebencian pada Teks Bahasa Indonesia Menggunakan Bidirectional Long Short Term Memory (Bi-LSTM)

Aditya Perwira Joan Dwitama

20917035

Yogyakarta, Maret 2023

Tim Penguji,

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.
Ketua

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

Chandra Kusuma Dewa, S.Kom., M.Cs., Ph.D.
Anggota II

Mengetahui,
Ketua Program Studi Informatika Program Magister

Universitas Islam Indonesia

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

Abstrak

Deteksi Ujaran Kebencian pada Teks Bahasa Indonesia Menggunakan Bidirectional Long Short Term Memory (Bi-LSTM)

Media sosial memberikan wadah bagi pengguna untuk bebas berekspresi termasuk menyebarkan konten ujaran kebencian yang dapat menimbulkan konflik sosial. Pemerintah Indonesia telah menerbitkan UU ITE sebagai upaya penanganan serta membentuk satu departemen khusus yaitu *virtual police*. Dari sisi teknologi, penelitian dilakukan menggunakan LSTM untuk mendeteksi ujaran kebencian pada teks media sosial. Penelitian tersebut berhasil mendapatkan akurasi yang sangat baik yakni 94,66%. Akan tetapi, penelitian tersebut memiliki batasan dengan *output* hanya satu label saja. Penelitian lain kemudian dilakukan untuk mendeteksi ujaran kebencian dengan *output multilabel* menggunakan Bi-GRU. Namun, akurasi yang didapatkan masih lebih rendah dari penelitian dengan LSTM yakni 86,44%. Oleh karena itu, penelitian terkait ujaran kebencian *multilabel* dilakukan pada penelitian ini. Penelitian dilakukan dengan menggunakan algoritma Bi-LSTM. Dataset yang digunakan dalam penelitian diambil dari dataset publik yang dapat diakses melalui github. Dataset tersebut berisikan data teks yang berasal dari twitter dengan jumlah 13ribu data. Percobaan dalam penelitian dimulai dari eksplorasi data dan *pre-processing*. Kemudian dilanjutkan dengan tokenisasi pada teks menggunakan model *pre-train* dari IndoBERT. Percobaan-percobaan untuk menggunakan variasi nilai parameter dilakukan untuk mendapatkan model dengan performa terbaik dalam mendeteksi ujaran kebencian *multilabel*. Penelitian juga dilakukan terhadap beberapa model IndoBERT guna mendapatkan hasil tokenisasi yang menunjang performa dari Bi-LSTM dalam melakukan klasifikasi. Hasilnya, model terbaik yang diusulkan dalam penelitian ini adalah dengan menggunakan 20 *epoch*, 192 *batch size*, 1 layer Bi-LSTM dengan 40 *node*, dan menerapkan *class weighing* dalam proses optimasinya. *Pre-train* model dari IndoBERT yang digunakan untuk mendukung kinerja dari model dalam melakukan klasifikasi adalah "indobenchmark/indobert-large-p2". Performa yang diberikan model sangat baik dengan berhasil mendapatkan akurasi yang lebih tinggi dari penelitian sebelumnya yakni 97,66%.

Kata kunci

Bi-LSTM, ujaran kebencian, multilabel, teks, twitter

Abstract

Indonesian Hate Speech Detection using Bidirectional Long Short-Term Memory (Bi-LSTM)

Social media provides a platform for users to express themselves freely including spreading hate speech content that can lead to social conflict. Indonesian government has issued UU ITE to handle hate speech and formed a special department, namely the virtual police. Research was also conducted using LSTM to classify the text into hate speech or not. The study managed to get a very good accuracy of 94.66%. However, it has limitations with the output of only one label. Another research was purposed to detect hate speech with multilabel output using Bi-GRU. However, the accuracy obtained is still lower than research with LSTM, which is 86.44%. Therefore, research related to multilabel hate speech was carried out in this study. The dataset used in the study is taken from a public dataset that can be accessed via github. The dataset contains text data from Twitter with a total of 13k data. Experiments in this study started from data exploration and pre-processing. Then, the process is continued with the tokenization of the text using the pre-train model from IndoBERT. Experiments to use a variety of parameter values were carried out to obtain a model with the best performance in detecting multilabel hate speech. Research was also conducted on several IndoBERT models to obtain tokenization results that support the performance of Bi-LSTM in classifying. As a result, the best model proposed in this study is to use 20 epochs, 192 batch sizes, 1 layer Bi-LSTM with 40 nodes, and apply class weighing in the optimization process. The pre-train model from IndoBERT used to support the performance of the model in classifying is "indobenchmark/indobert-large-p2". the model managed to provide a very good performance of 97.66%. The accuracy is higher than previous studies.

Keywords

Bi-LSTM, hate speech, multilabel, text, twitter.

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Maret 2023

Aditya Perwira Joan Dwitama, S.Kom

Daftar Publikasi

Publikasi yang menjadi bagian dari tesis

Dwitama, A. P. J., & Hidayat, S. (2021). Identifikasi Ujaran Kebencian Multilabel Pada Teks Twitter Berbahasa Indonesia Menggunakan Convolution Neural Network. Jurnal Sistem Komputer Dan Informatika (JSON), 3(2), 117. https://doi.org/10.30865/json.v3i2.3610.

Publikasi berikut menjadi bagian dari Bab 3

Sitasi publikasi 1

Kontributor	Jenis Kontribusi
Aditya Perwira Joan Dwitama	Mendesain eksperimen (70%) Menulis *paper* (70%)
Syarif Hidayat	Mendesain eksperimen (30%) Menulis dan mengedit *paper* (30%)

Dwitama, A.P.J., Fudholi, D.H., Hidayat, S. (2023). Indonesian Hate Speech Detection Using Bidirectional Long Short-Term Memory (Bi-LSTM). Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), (approved).

Publikasi berikut meliputi keseluruhan laporan tesis kecuali bagian penelitian tambahan pada Bab 4

Sitasi publikasi 2

Kontributor	Jenis Kontribusi
Aditya Perwira Joan Dwitama	Mendesain eksperimen (60%) Menulis *paper* (60%)
Dhomas Hatta Fudholi	Mendesain eksperimen (30%) Menulis *paper* (25%)
Syarif Hidayat	Mendesain eksperimen (10%) Menulis dan mengedit *paper* (15%)

Halaman Kontribusi

Tidak ada kontribusi dari pihak lain

Halaman Persembahan

Dengan menyebut nama Allah yang Maha Pengasih dan Maha Penyayang. Puji sykur saya ucapkan kepada-Nya sebagai rasa bentuk syukur karena telah berhasil menyelesaikan tessis dengan judul "Deteksi Ujaran Kebencian pada Teks Bahasa Indonesia Menggunakan Bidirectional Long Short Term Memory (Bi-LSTM)". Tesis ini saya persembahkan untuk:

1. Diri saya sendiri sebagai bentuk apresiasi karena telah berhasil menggapai salah satu cita-cita untuk mendapatkan gelar magister.
2. Kedua orang tua saya, H. Joko Wiyono dan Hj. Muawwanah yang selalu memberikan support untuk segala pilihan dalam perjalanan hidup.
3. Kakak satu-satunya, Okky Angriawan Joan Pratama yang telah mendukung adek bungsunya serta memberikan pembelajaran dari pengalaman hidupnya.

Kata Pengantar

Assalamu'alaikum warahmatullahi wabarakatuh.

Alhamdulillah, puji syukur dipanjatkan kepada Allah SWT yang telah melimpahkan limpahan rahmatnya sehingga laporan tesis dengan judul "Deteksi Ujaran Kebencian pada Teks Bahasa Indonesia Menggunakan Bidirectional Long Short Term Memory (Bi-LSTM)" bisa terselesaikan di waktu yang tepat. Tesis ini menjadi bukti bahwa saya sebagai penulis telah berhasil menyelesaikan masa studi program magister di Program Studi Magister Informatika di Universitas Islam Indonesia dan lebih tepatnya pada konsentrasi Sains Data. Penulis menyadari sepenuhnya jika tanpa adanya bantuan dari berbagai pihak, maka penulis tidak akan menyelesaikan tesis ini dengan baik. Oleh karena itu, selain rasa syukur yang tiada henti tercurah, izinkan penulis dengan tulisan ini untuk menyampaikan segenap terimakasih kepada:

1. Keluarga tercinta yakni orang tua dan saudara yang selalu mensupport selama proses studi.
2. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku dosen pembimbing 1 yang sabar memberikan arahan dan masukan sejak proposal tesis saya ajukan.
3. Bapak Dr. Syarif Hidayat, S.Kom., M.I.T. selaku doesn pembimbing 2 yang selalu memotivasi dan membimbing untuk saya terus mengerti tentang apa yang dilakukan selama tesis.
4. Bapak dan Ibu dosen yang telah memberikan bekal pengetahuan untuk saya menjalani tesis.
5. Staff Program Studi Magister Informatika Universitas Islam Indonesia yang telah membantu menjawab segala pertanyaan terkait administrasi.
6. Sahabat dan seperjuangan yang tidak dapat disebutkan satu persatu.

Sekali lagi, penulis mengucapkan terimakasih juga kepada pembaca. Semoga tesis ini bisa bermanfaat untuk pembaca dan perkembangan ilmu pengetahuan dikemudian hari.

Wassalamu'alaikum warahmatullahi wabarakatuh.

Aditya Perwira Joan Dwitama

Daftar Isi

Daftar Tabel

Daftar Gambar

BAB 1
Pendahuluan

1.1. Latar Belakang

Survei yang dilakukan oleh Hootsuite menunjukkan hasil bahwa pengguna internet di Indonesia pada tahun 2021 meningkat lebih dari 15,5% dari tahun sebelumnya yakni sebanyak 202,6 juta pengguna. Dimana, 83,9% diantaranya menggunakan internet untuk beraktivitas di media sosial [1].

Media sosial memberikan wadah kepada penggunanya untuk mengekspresikan apa yang ada di dalam pikirannya. Di sisi lain, kebebasan tersebut juga memberikan peluang terhadap kemunculan konten-konten yang mengandung ujaran kebencian [2]. Konten yang mengandung ujaran kebencian dapat berdampak negatif bagi sosial masyarakat karena dapat menimbulkan konflik sosial, diskriminasi, sampai pembunuhan, bahkan berakhir pada hukuman penjara bagi pelaku sesuai dengan UU yang berlaku [3].

Ujaran kebencian merupakan suatu ungkapan baik secara tulisan maupun tindakan yang ditujukan untuk merendahkan bahkan mendiskriminasi suatu individu dan/atau kelompok berdasarkan ras, suku, jenis kelamin, agama, dan aspek-aspek lainnya [4]. Misalkan pada teks yang berbunyi “Kristen halal da- rahnya! Bunuh mereka! Jangan biarkan mereka mendirikan gereja di tanah kita!”. Kata “halal darahnya”, “Bunuh mereka” sudah bukan lagi menjadi sebuah teks diskriminasi lagi, tapi menunjukkan ancaman yang sangat kuat kepada kelompok masyarakat Kristen. Bahaya dari hal tersebut adalah, jika dari pihak yang menjadi target ujaran kebencian tidak terima atas kalimat tersebut, maka kemungkinan konflik perpecahan antar agama akan terjadi.

Pemerintah Indonesia dalam hal ini tidak tinggal diam. Upaya pencegahan dan penanganan ujaran kebencian sudah diupayakan dengan dimuat dalam Undang-Undang tentang Informasi dan Transaksi Elektronik (UU ITE) pasal 28 ayat 2. Dalam peraturan tersebut disebutkan bahwa setiap warga negara dilarang untuk menyebarkan informasi dapat menimbulkan rasa kebencian atau permusuhan individu dan/atau kelompok masyarakat tertentu berdasarkan atas suku, agama, ras, dan antar golongan (SARA) [5]. Akan tetapi, aturan ini tidak cukup untuk mencegah dari kemunculan kasus ujaran kebencian di Indonesia. Pihak kepolisian pada periode Februari sampai Maret 2021 melalui Virtual Police

bentukan Direktorat Siber Bareskrim Polri telah melakukan teguran kepada 125 akun media sosial yang terindikasi membuat konten yang mengandung ujaran kebencian [6].

Dalam praktiknya, suatu konten atau teks digolongkan sebagai ujaran kebencian atau tidak masih dilakukan secara manual [7]. Hal ini kemudian bisa jadi membuat analisa atau penafsiran dari ujaran kebencian menjadi bias. Hasil analisa akan kembali lagi pada perspektif individu yang menganalisis ujaran kebencian tersebut [8]. Apalagi aktivitas media sosial yang sangat padat mengharuskan pihak yang berwenang untuk melakukan pengawasan dan tindakan secara cepat.

Dari sisi teknologi, penelitian tentang ujaran kebencian telah dilakukan menggunakan pendekatan *Artificial Intelligent* terlebih khusus *Machine Learning*. Penelitian dilakukan sebagai upaya untuk melakukan otomatisasi terhadap pendeteksian ujaran kebencian. Salah satu dari penelitian tersebut adalah penelitian untuk membangun model pendeteksi ujaran kebencian menggunakan algoritma Bidirctional Long Short-Term Memory (Bi-LSTM). Penelitian ini menghasilkan model dengan *output* yang dapat mengenali apakah suatu teks tergolong sebagai ujaran kebencian atau tidak. Performa dari model pun sangat baik dengan mendapatkan akurasi sebesar 94,66% [9].

Tidak hanya itu, penelitian tentang ujaran kebencian juga pernah dilakukan untuk menghasilkan model yang dapat mendeteksi suatu teks tergolong sebagai ujaran kebencian berikut dengan aspek yang disinggung didalamnya. Dengan kata lain, model yang dihasilkan memiliki output label lebih dari satu atau bisa disebut sebagai multilabel. Penelitian tersebut dilakukan dengan menggunakan algoritma Bidirectional Gated Recurrent Unit (Bi-GRU). Hal yang menarik dari penelitian ini juga adalah terletak pada teknik tokenisasi teks yang digunakan. Tokenisasi dilakukan dengan menggunakan *pre-train* model dari IndoBERT. Model yang diusulkan kemudian mampu memberikan performa akurasi sebesar 84,77% [2].

Akan tetapi, performa 84,77% dari penelitian ujaran kebencian *multilabel* [2] masih berada di bawah penelitian [9] dengan model *single* labelnya yang memperoleh performa akurasi sebesar 94,66%. Penulis kemudian melihat adanya peluang untuk melakukan penelitian kembali terhadap ujaran kebencian multilabel pada teks Bahasa Indonesia karena melihat performa dari penelitian *single* label menggunakan Bi-LSTM mampu mencapai akurasi yang sangat tinggi dengan menembus angka 90%.

Oleh karena itu, penelitian ini akan melakukan pemodelan untuk mendeteksi ujaran kebencian *multilabel* pada teks Bahasa Indonesia. Pemodelan dilakukan dengan menggunakan algoritma Bi-LSTM dan *pre-train* dari IndoBERT sebagai model untuk

melakukan tokenisasi pada teks. Dengan demikian, penelitian ini diharapkan mampu memberikan performa yang lebih baik dari pada penelitian sebelumnya yang menggunakan Bi-GRU dalam melakukan deteksi ujaran kebencian *multilabel*.

1.2. Rumusan Masalah

Penelitian tentang ujaran kebencian dengan *output multilabel* telah dilakukan menggunakan algoritma Bi-GRU dengan akurasi sebesar 84,66%. Penelitian tersebut masih mendapatkan akurasi yang lebih rendah dibandingkan dengan penelitian dengan *output* satu label yakni 94,66%.

1.3. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini untuk dapat meningkatkan akurasi dari model dalam mendeteksi ujaran kebencian dengan *output multilabel* menggunakan algoritma Bi-LSTM.

1.4. Manfaat Penelitian

Beberapa manfaat yang bisa diambil dari penelitian ini adalah antara lain.

a. Model yang dihasilkan dapat digunakan sebagai bahan untuk *transfer learning* guna menghasilkan sistem otomatisasi dalam memfilter teks-teks yang mengandung ujaran kebencian serta aspek yang disinggungnya.
b. Performa model dalam mendeteksi teks ujaran kebencian dan aspek yang ada disinggung dapat membantu pihak terkait dalam mencegah ataupun memproses tindakan ujaran kebencian dalam teks terutama di media sosial.
c. Menjadi acuan bagi penelitian selanjutnya untuk melakukan pengembangan sistem ataupun peningkatan performa.

1.5. Batasan Masalah

Beberapa batasan masalah diterapkan dalam penelitian ini yaitu antara lain.

a. Data yang digunakan adalah data yang dihasilkan dari penelitian [10].
b. Label yang akan digunakan dari dataset yang tersedia adalah *hate speech*, *abisuve*, *individual*, *group*, *religion*, *race*, *physical*, *gender*, dan *others*.
c. Model dibangun dengan algoritma Bi-LSTM dengan bahasa pemrograman python.

d. Performa model akan dianalisis menggunakan matriks klasifikasi yaitu akurasi, presisi, dan *recall*.

BAB 2
Tinjauan Pustaka

2.1. Ujaran Kebencian

Ujaran kebencian dalam pemerintahan Negara Kesatuan Republik Indonesia dijelaskan dalam UU ITE Pasal 28 ayat (2) yang berbunyi "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 antragolongan (SARA)". Penjelasan mengenai penanganan kasus ujaran kebencian juga pernah tersematkan dalam Surat Edaran KAPOLRI Nomor SE/6/X/2015. Salah satu hal yang dimuat dalam surat edaran tersebut adalah mengenai aspek yang ditujukan dalam suatu ujaran kebencian. Aspek-aspek yang dimaksud antara lain suku, agama, aliran keagamaan, keyakinan/kepercayaan, ras, antragolongan, warna kulit, etnis, gender, kaum difabel (cacat), dan orientasi seksual.

2.2. Dataset

Dataset merupakan kumpulan data yang memiliki struktur tertentu sesuai dengan kebutuhannya. Struktur tersebut sering disebut sebagai sebuah *dataframe* memiliki sejumlah baris dan atribut (kolom) berikut dengan *header*-nya. Atribut dalam dataset dapat berupa teks yang sering digunakan dalam membangun model untuk analisis pada teks [11]. Salah satu platform yang dapat sering digunakan untuk membangun dataset dengan atribut berupa teks adalah Twitter.

Twitter merupakan salah satu platform media sosial yang masuk ke dalam 10 besar pengguna terbanyak di Indonesia. Hal ini mengacu pada laporan hasil survei yang telah dirilis oleh Hootsuite yang bertajuk "Digital 2021: Indonesia". Dalam laporan tersebut, tercatat bahwa Twitter menempati peringkat 5 pada rangking media sosial yang paling sering digunakan oleh masyarakat Indonesia [1].

Twitter merupakan suatu platform sosial media yang memberikan ruang bagi penggunanya untuk bebas berekspresi di dalamnya. Pengguna dapat mem-*post* tulisan (twit) ke dalam Twitter dengan maksimal 280 karakter. Fasilitas tersebut dirasa sangat cukup pagi pengguna Twitter untuk bebas mengekspresikan apa yang sedang mereka pikirkan [12].

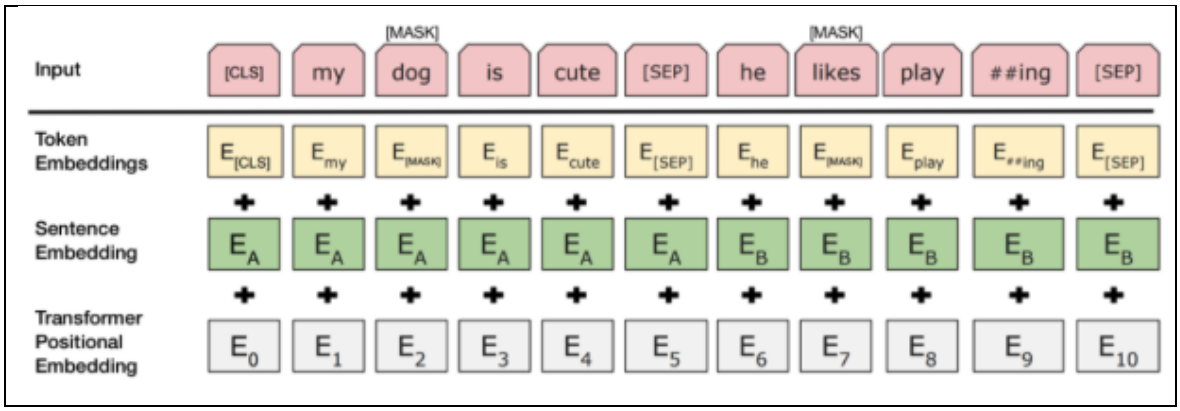
2.3. IndoBERT dan Tokenisasi

Komputasi pemodelan yang dilakukan oleh komputer sejatinya adalah merupakan hitung-hitungan aritmetika. Untuk itu, dalam proses Natural Language Processing (NLP) perlu dilakukan proses untuk mentransformasi teks dalam NLP menjadi sebuah angka. Selanjutnya proses tersebut dikenal dengan istilah *word embedding* [13].

Word embedding telah digunakan dalam penelitian-penelitian untuk membantu meningkatkan performa dari proyek penelitian NLP. *Output* yang diberikan oleh *word embedding* adalah berupa vektor yang merepresentasikan kumpulan kata pada suatu kalimat [14]. Salah satu algoritma yang bisa digunakan untuk melakukan proses *word embedding* adalah IndoBERT.

IndoBERT merupakan salah satu *pre-train* dengan basis model menggunakan *Bidirectional Encoder Representations from Transformers* (BERT). IndoBERT dibangun dan dikembangkan secara khusus untuk menagani kasus teks dalam Bahasa Indonesia. IndoBERT dikatakan mampu melakukan *embedding* pada level *contextual word embedding*. Jika satu kata digunakan dalam 2 kalimat yang berbeda, maka BERT akan mengkonversi kata tersebut menjadi nilai yang berbeda juga sesuai dengan konteks kalimatnya [15].

BERT melakukan tokenisasi pada level kata dan menghasilkan sebanyak 3 *output* sebagai id dari tiap token. *Output-output* tersebut adalah *token embeddings*, *segment embeddings*, dan *position embedding*. *Token embeddings* berfungsi sebagai id dari kata dalam *bag of words* dalam pretrain modelnya. *Segment embeddings* berfungsi sebagai id yang memisahkan antara kalimat satu dengan kalimat selanjutnya. Terakhir, *posisitonal embedding* yang digunakan untuk menandai mana kalimat yang sebenarnya dan mana *token padding* untuk menutup kekurangan jumlah kata pada suatu teks. Gambaran mengenai proses tokenisasi pada BERT dapat dilihat pada Gambar 2.1.



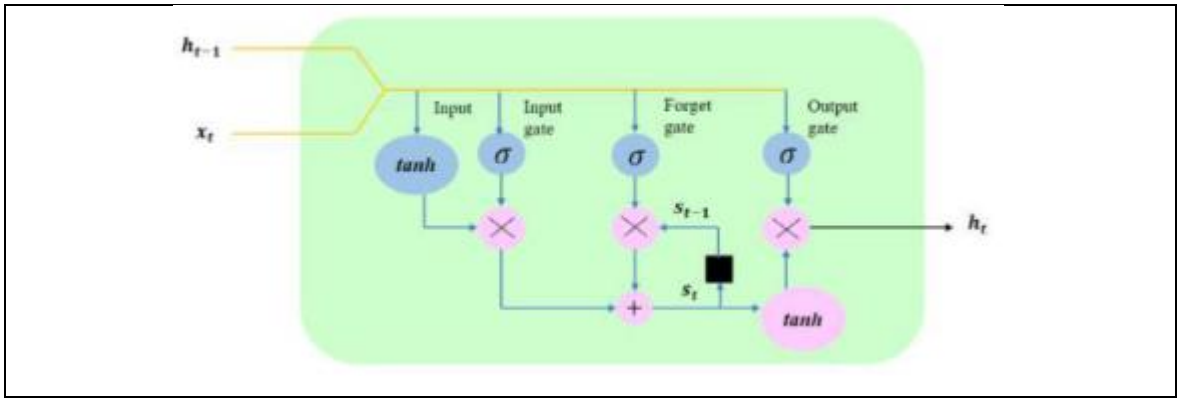
Gambar 2.1. *input/output* tokenisasi pada BERT [16].

Proses tokenisasi pada BERT diawali dengan dilakukannya *word tokenization* pada teks. BERT memiliki *default token* yang akan ditambahkan secara otomatis pada hasil tokenisasi. Token-token yang dimaksud antara lain [CLS], [PAD], dan [SEP]. Token [CLS] akan ditambahkan pada awal teks untuk menandai titik awal dari sebuah teks. [PAD] muncul apabila jumlah kata pada teks atau *token* yang dihasilkan dari proses tokenisasi kurang dari panjang token yang sudah didefinisikan pada model BERT. Jika terdefinisi panjang *output* token dari BERT adalah 10 sedangkan token pada teks hanya berjumlah 6 maka 4 token lainnya akan diisi oleh [PAD]. Kemudian token [SEP] digunakan apabila terdapat 2 atau lebih kalimat sebagai *input*-an dan token ini digunakan sebagai pemisah antar keduanya [16].

2.4. LSTM

Long Short Term Memory (LSTM) sebenarnya merupakan salah satu hasil dari pengembangan algoritma sebelumnya yaitu Recurrent Neural Netwotrk (RNN). LSTM dikembangkan untuk menjadi solusi atas masalah *vanishing gradient* yang terjadi pada RNN [17].

LSTM menawarkan struktur yang terdiri dari 3 buah *gate* (gerbang) yaitu *input gate, output gate*, dan *forget gate*. Keberadaan *gate* tersebut bertujuan agar LSTM mampu menyimpan informasi-informasi dari data yang masuk ke dalam arsitekturnya (Chauhan & Palivela, 2021).



Gambar 2.2 Arsitektur pada layer LSTM.

Pada Gambar 2.2, arsitektur diawali dengan variable x_t dan h_{t-1} sebagi *input* dari arsitektur. x_t merepresentasikan vektor *embedding* dari *token* yang akan di proses oleh LSTM. Sedangkan h_{t-1} adalah *hidden state* yang dihasilkan dari *state* atau *node* LSTM sebelumnya.

Pada bagian awal arsitektur, x_t dan h_{t-1} digabungkan menggunakan fungsi tanh (2.1) sebagai suatu informasi baru pada arsitektur LSTM.

$$g = \tanh\left(b^g + x_t U^g + h_{t-1} V^g\right) \tag{2.1}$$

Dimana g bertindak sebagai *input* pada LSTM, U^g dan V^g masing-masing adalah *weight* dari *model*, serta b^g bertindak sebagai *bias*. Selanjutnya, x_t dan h_{t-1} juga dibawa ke *input gate* yang merupakan fungsi sigmoid dari x_t dan h_{t-1} (2.2).

$$i = \sigma\left(b^i + x_t U^i + h_{t-1} V^i\right) \tag{2.2}$$

Dimana i merupakan representasi dari *input gate*. *Input gate* pada LSTM berperan sebagai gerbang yang akan menyortir sebanyak apa elemen dari *input* g yang boleh lanjut ke tahap selanjutnya. Elemen-elemen dihilangkan karena keberadaan *sigmoid* pada i yang menghasilkan nilai antara 0 sampai 1. Jika hasil kali element g terhadap i menghasilkan nilai yang mendekati 0 maka elemen dari g dianggap tidak penting dan tidak diizinkan untuk melanjutkan k ke dalam LSTM *cell state* (h_t). Gate selanjutnya yang menyusun LSTM adalah *forget gate*. Secara matematis, *forget gate* tersusun atas fungsi sigmoid seperti persamaan (2.3).

$$f = \sigma(b^f + x_t U^f + h_{t-1} V^f) \tag{2.3}$$

Dimana f merepresentasikan fungsi *forget gate* pada LSTM. f bertugas untuk menentukan mana saja state dari *cell* sebelumnya (s_{t-1}) yang perlu dihilangkan. Terakhir, LSTM memiliki *output gate* yang berfungsi sebagai *gate* terakhir pada LSTM untuk menghasilkan *hidden state*. *Output gate* lagi lagi tersusun atas fungsi sigmoid seperti persamaan (2.4).

$$o = \sigma(b^o + x_t U^o + h_{t-1} V^o) \tag{2.4}$$

Dimana o merupakan representasi dari *output gate*. *Cell state* pada LSTM (s_t) di filter kembali pada *output gate* dengan operator perkalian. Hasilnya, sebuah layer LSTM akan menghasilkan *hidden state* h_t yang akan menjadi input bagi node LSTM slanjutnya atau menjadi output bagi layer LSTM [18].

LSTM kemudian berkembang dari segi arsitektur pemodelannya dengan penambahan konsep *bidirectional*. Satu layer LSTM akan memuat 2 buat LSTM sehingga menjadi sebuah layer Bi-LSTM. Arsitektur terebut memungkinkan 2 buah LSTM saling terhubung satu sama lain. Layer pertama akan meneruskan hasil dari *hidden layer*-nya kepada layer kedua. Begitu juga sebaliknya pada layer kedua, pada arus baliknya, hasil dari *hidden layer* kedua akan menjadi *input*-an untuk layer pertama tadi [9].

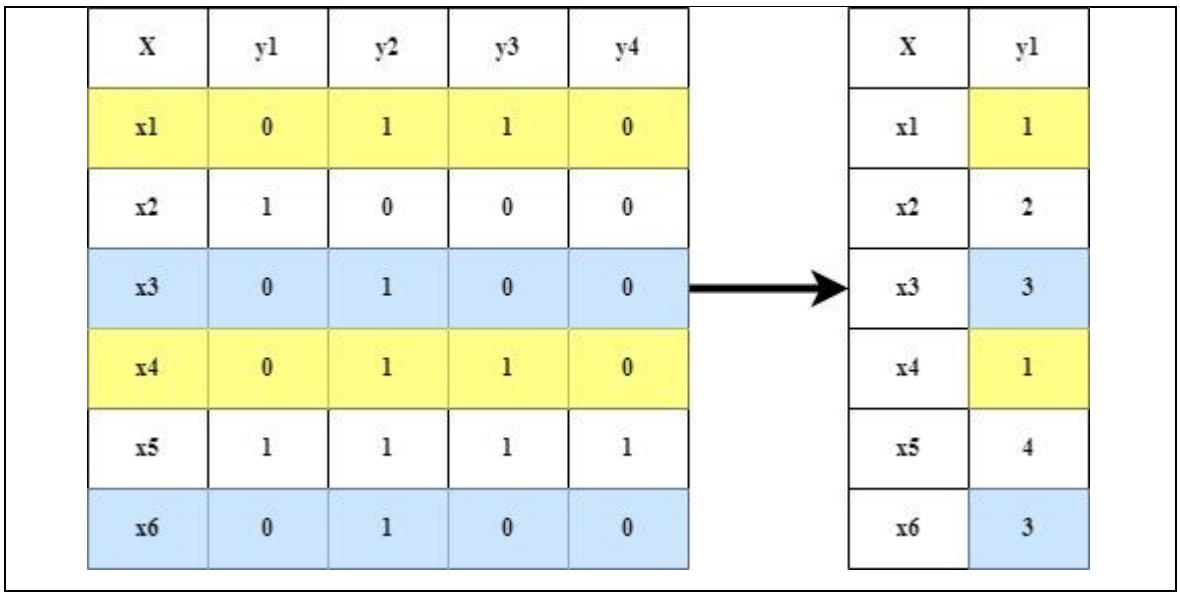
2.5. MultiLabel

Perbedaan yang paling menonjol antara klasifikasi *single* label dan *multilabel* adalah terletak pada keluaran dari model klasifikasi. Klasifikasi *single* label memberikan *output* berupa satu buah kelas, sedangkan *multilabel* memberikan keluaran dengan kelas lebih satu. Pembangunan model klasifikasi untuk permasalahan *multilabel* memiliki tingkat kesulitan yang lebih tinggi jika dibandingkan dengan *single* label. Permasalahan terkait klasifikasi *multilabel* menuntut model untuk dapat memprediksi atau mengategorikan data ke dalam beberapa kelas sekaligus [19].

Salah satu pendekatan yang bisa dilakukan untuk melakukan pemodelan terhadap permasalahan multilabel adalah dengan mentransformasi data. Terdapat beberapa metode yang bisa diterapkan dalam transformasi data pada kasus dataset *multilabel*. Metode-metode tersebut antara lain Binary Relevance (BR), Label Powerset (LP), dan Classifier Chain (CC). ketiga metode tersebut pernah diterapkan dalam penelitian [10]. Penelitian berhasil memperoleh hasil terbaiknya ketika menggunakan metode LP.

Metode LP diterapkan dengan melakukan transformasi dataset *multilabel* menjadi *multiclass*. LP mentransformasi kombinasi dari tiap label menjadi kelas. Kelas-kelas tersebut kemudian di klasifikasi dalam satu label dengan karakteristik *multiclass classification* [20]. Lebih jelas mengenai transformasi oleh LP bisa dilihat pada ilustrasi Gambar 2.3.

Pada Gambar 2.3, *y1*, *y2*, *y3*, dan *y4* merupakan representasi dari 4 label dalam dataset. Tiap kombinasi dari label tersebut menghasilkan satu *class* masing-masing. Klasifikasi dilakukan menggunakan satu model *multiclass* dengan *output* sebanyak 4 *node*. Jumlah *node output* dari model mengikuti jumlah kombinasi dari *class* yang terlibat. Tiap *node* dari model merepresentasi nilai dari tiap label sesuai urutannya. Sebagai contoh, misalkan model memberikan *output* {0,1,0,0}. *Output* tersebut memiliki arti bahwa model mengklasifikasi data sebagai *class* 3 atau dengan kata lain model mengklasifikasi data sebagai positif label *y2* dan negatif untuk label *y1*, *y3*, dan *y4*.



X	y1	y2	y3	y4
x1	0	1	1	0
x2	1	0	0	0
x3	0	1	0	0
x4	0	1	1	0
x5	1	1	1	1
x6	0	1	0	0

X	y1
x1	1
x2	2
x3	3
x4	1
x5	4
x6	3

Gambar 2.3. Ilustrasi transformasi Label Powerset (LP).

2.6. Batch Size, Epoch, Learning Rate

Batch Size merupakan parameter dalam pemodelan *machine learning* untuk mendefinisikan berapa jumlah data yang digunakan dalam sekali proses *training* sebelum model melakukan optimasi *weight*. Semakin kecil *batch size* yang digunakan dalam pemodelan, maka semakin sering model melakukan optimasi *weight*. Hal ini dikarenakan model akan memiliki jumlah iterasi yang lebih banyak dalam satu *epoch training* [21].

Epoch merupakan parameter dalam pemodelan *machine learning* untuk mendefinisikan berapa kali model harus melakukan pembelajaran terhadap keseluruhan dataset. Satu *epoch* berarti bahwa model melakukan pembelajaran terhadap dataset hanya satu kali saja. Jadi, semakin banyak *epoch* yang digunakan maka semakin sering model mempelajari dataset atau semakin sering model melakukan optimasi *weight* [21].

Learning rate merupakan parameter dalam pemodelan yang digunakan untuk mendefinisikan seberapa besar perubahan *weight* model dalam sekali proses optimasi. *Learning rate* juga berperan untuk mendapatkan model yang paling optimal serta waktu komputasi dari proses pemodelan. Semakin kecil *learning rate* yang digunakan maka semakin lama proses *training* dari model karena membutuhkan *epoch* yang banyak untuk mencapai posisi optimalnya. Sedangkan nilai *learning rate* yang terlalu besar memungkinkan model untuk melewati posisi optimalnya [21].

2.7. Peneltian terkait

Penelitian tentang ujaran kebencian *multilabel* dalam 3 tahun belakangan masih ditemukan dalam penelitian terkait dengan *machine learning atau deep learning*. Terlebih khusus lagi penelitian yang menggunakan dataset berbahasa Indonesia. Penulis menemukan sebanyak 5 penelitian terdahulu yang membahas mengenai ujaran kebencian dalam Bahasa Indonesia. Ulasan kritis masing-masing penelitian disajikan Tabel 1.1 .

Seorang peneliti bernama Okky Ibrahim melakukan penelitian terhadap ujaran kebencian Bahasa Indonesia sebanyak 3 kali pada tahun 2019. Penelitian pertamanya menghasilkan satu dataset ujaran kebencian yang digunakan oleh penelitian-penelitian berikutnya [2], [4], [22], [23]. Dataset tersebut didapatkan dari hasil *crawling* menggunakan Twitter API. Tiap data dalam dataset memiliki sifat *multilabel* dimana satu data dianotasi ke dalam 12 label dengan masing-masing label bersifat biner [10].

Deteksi ujaran kebencian *multilabel* dalam Bahasa Indonesia telah dilakukan menggunakan pemodelan *machine learning* seperti Support Vector Machine (SVM) [10], [22], [23], Random Forest Decision Tree (RFDT) [10], [22], [23], Logistic Regression [22]. Setiap penelitian tersebut menggunakan dataset yang sama yaitu dataset yang dihasilkan dari penelitian [10]. Namun masing-masing penelitian tersebut menggunakan hanya beberapa label saja dari total 12 label yang disediakan dalam dataset. Pada penelitian [4], label yang digunakan sebagai *output* dari model hanya 2 yakni *hate speech* dan *abusive* saja. Penelitian tersebut berhasil mendapatkan akurasi sebesar 86,2%.

Pada tahun 2021, Bi-GRU diusulkan sebagai algoritma untuk melakukan klasifikasi terhadap ujaran kebencian *multilabel* Bahasa Indonesia. Model Bi-GRU dikombinasikan dengan *pre-train* model dari IndoBERT untuk melakukan tokenisasi pada teks. untuk melakukan tokenisasi pada teks. Penelitian tersebut mendapatkan hasil yang baik dengan akurasi sebesar 86,44%. Namun, akurasi yang baik tersebut masih memiliki catatan dimana model masih sulit untuk mendeteksi ujaran kebencian pada label yang didominasi oleh nilai negatif.

Algoritma Bi-GRU pada penelitian [2] pada dasarnya merupakan algoritma GRU dengan menerapkan konsep *bidirectional* di dalam arsitekturnya. GRU merupakan algoritma yang kembangkan dengan tujuan yang sama dengan LSTM untuk mengatasi *vanishing gradient* dari RNN. Penelitian-penelitian yang menggunakan LSTM ataupun Bi-LSTM dalam 3 tahun belakangan masih ditemukan untuk menangani kasus pada analisis teks. Ulasan kritis dari penelitian-penelitian tersebut bisa dilihat pada Tabel 1.2.

Terdapat 4 dari 5 penelitian memanfaatkan LSTM sebagai algoritma untuk menyelesaikan kasus ujaran kebencian [8], [9], [24], [25]. Dari keempat penelitian tersebut, terdapat 4 bahasa yang muncul sebagai bahan penelitian antara lain bahasa Indonesia [8], [9], [25], arab [24]. LSTM sebagai algoritma klasifikasi berhasil bekerja dengan sangat baik di setiap penelitian dengan akurasi di atas 90% kecuali pada penelitian Bahasa arab yang mendapatkan akurasi sebesar 86,4%. Namun, yang perlu di garis bawahi dari bagusnya akurasi yang diperoleh tiap penelitian adalah Batasan masalah yang digunakan adalah *single* label.

Penelitian pada Tabel 1.2 juga ada yang menerapkan konsep *bidirectional* dalam arsitektur LSTM. Masing-masing penelitian membahasa tentang ujaran kebencian [9] dan *sentiment analysis* [26]. Bi-LSTM dalam dua penelitian tersebut berhasil memberikan performa yang sangat baik dengan akurasi di atas 90%. Menariknya, pada penelitian [9] dilakukan pemodelan untuk ujaran kebencian menggunakan dataset yang sama dengan penelitian menggunakan Bi-GRU [2]. Hanya saja, penelitian tersebut berfokus pada satu label saja sebagai *output* dari model yakni *hate speech*.

Tabel 1.1 Ulasan Penelitian Ujaran Kebencian.

No.	Sub Tema	Keywords	Ulasan Kritis	Pustaka
1	*Identification of hate speech and abusive language on Indonesian twitter using theword2vec, part of speech and emoji features*	*Hate Speech, Abusive Language, Twitter, Machine Learning*	Penelitian dilakukan untuk membandingkan performa beberapa algoritma *machine learning* seperti Logistic Regression, RFDT, dan SVM. Dari sisi data, penelitian ini berfokus untuk melakukan deteksi terhadap teks dengan label berjumlah 2 yaitu label *hate speech* dan label *abusive*. Performa terbaik didapatkan pada algoritma Logistic Regression dengan akurasi sebesar 79,85%	[22]
2	*Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter*	*Multilabel, data transformation, machine learning, twitter*	Penelitian ini dilakukan untuk melakukan deteksi ujaran kebencian pada teks Twitter dengan *multilabel*. Ada beberapa eksperimen yang dilakukan pada penelitian ini untuk mencari arsitektur model deteksi yang paling baik. eksperimen yang dimaksud adalah dari sisi fitur ekstraksi (Orthography, lexicon, unigram), algoritma (NB, SVM, FRDT. Model terbaik diperoleh dengan kombinasi unigram dan RFDT dengan rata-rata akurasi tiap label sebesar 66.12 %	[10]
3	*Hierarcical Multi-label Classification to Identify Hate Speech and Abusive Language on Indonesian Twitter*	*Multilabel, data transformation, machine*	Penelitian ini memiliki 5 skenario untuk kombinasi label yang digunakan dari 12 label yang tersedia sebagai target dari model. Dari sisi algoritma, penelitian ini menggunakan algoritma *machine learning* sebagai pembanding yaitu NB,	[23]

		learning, twitter	RFDT, dan SVM. Hasil terbaik didapatkan ketika menggunakan kombinasi label pada skenario 2 dengan jumlah label sebanyak 9 label dan algoritma SVM dengan akurasi sebesar 68.43%	
4	*Hate Speech Detection in Indonesian Twitter Texts using Bidirectional Gated Recurrent Unit*	*Hatespeech, text classification, BiGRU, Word2Vec, Fasttext*	Penelitian ini melakukan deteksi terhadap beberapa label sekaligus seperti etnis, agama, individu, grup, serta ras. Model dibangun menggunakan algoritma Bi-GRU dan berhasil mendapat akurasi terbaik sebesar 84,77% dengan teknik *word embedding* menggunakan Indo-BERT	[2]
5	*Hate Speech and Abusive Language Classification using fastText*	*abusive language, hate speech, continous bag ofwords, text classification, fastText*	Penelitian ini berfokus untuk melihat performa dari fastext *classification* dengan arsitektur model deteksi berupa *neural network*. Penelitian ini dilakukan melakukan deteksi teks *multilabel* dengan rincian tabel yaitu *hate speech* dan *abusive*. Model terbaik diperoleh menggunakan *pre-train* model fasttext Wikipedia dengan nilai F1 sebesar 86,2%.	[4]

Tabel 1.2 Ulasan Penelitian LSTM.

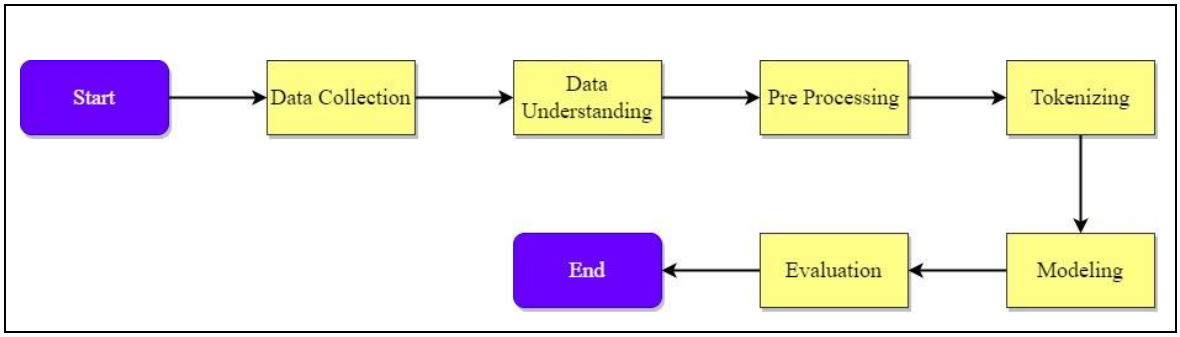
No.	Sub Tema	Keywords	Ulasan Kritis	Pustaka
1	*Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection*	*Hate Speech, LSTM, BiLSTM, Word2vec, CBOW, Skipgram, Twitter*	Penelitian dilakukan untuk membangun model yang bertujuan untuk mendeteksi suatu teks adalah ujaran kebencian atau tidak. Penelitian ini menggunakan algoritma Bi-LSTM dengan performa model sebesar 94,66%	[9]
2	*Sentiment classification using attention mechanism and bidirectional long short-term memory network*	*Attention mechanism, Bidirectional long short-term memory, Sentiment classification, Social media, Word embedding*	Penelitian ini adalah ditujukan untuk kasus *sentiment analysis* pada Bahasa China dan inggris. Salah satu metode yang digunakan dalam penelitian ini adalah dengan Bi-LSTM. algoritma mampu memberikan hasil akurasi sebesar 94,10% untuk dataset berbahasa China dan 93,92% untuk dataset berbahasa inggris.	[26]
3	*Detecting Offensive Language on Arabic Social*	*offensive language*	Penelitian ini bertujuan untuk mendeteksi kata *offensive* dalam komentar di Youtube yang ditulis dalam aksara arab.	[24]

	Media using Deep Learning	*detection, Arabic language, social media, deep learning, convolutional neural network, long short-term memory, attention model*	Bahasa arab memiliki word embeddingnya sendiri dalam teks analisis. Salah satunya adalah yang digunakan dalam penelitian ini yaitu aravec. Untuk algoritma pemodelan, penelitian ini salah satunya menggunakan model dengan arsitektur Bi-LSTM. performa yang diberikan oleh model cukup baik dengan menyentuh angka akurasi sebesar 86,4%	
4	*Analysis Text of Hate Speech Detection Using Recurrent Neural Network*	*Hate Speech, Analysis text, Deep Learning, Recurrent Neural Network, RNN, LSTM*	Penelitian dilakukan untuk mendeteksi apakah suatu teks berbahasa Indonesia tergolong sebagai ujaran kebencian atau tidak. Teknik *embedding* yang digunakan pada penelitian ini adalah word2vec. Hasilnya, penelitian mampu melakukan mendeteksi teks yang tergolong sebagai ujaran kebencian dengan akurasi sebesar 91%.	[25]
5	*Deep Learning-Based Implementation of Hate Speech Identification on Texts in Indonesian: Preliminary Study*	*Deep Learning-Based Implementation*	Penelitian dilakukan untuk mendeteksi suatu teks dalam bahasa Indonesia tergolong sebagai ujaran kebencian atau tidak. *Embedding* dari penelitian ini membandingkan 2 teknik yaitu word2vec dan fasttext. Dari segi hasil, model berhasil	[8]

		of Hate Speech Identification on Texts in Indonesian: Preliminary Study	menunjukkan performa yang baik terhadap 2 dataset yang digunakan dengan *embeeding* menggunakan fastext. Dataset pertama berhasil mendapatkan akurasi sebesar 95,93% dan dataset kedua berhasil mendapatkan akurasi sebesar 97,39%.	

BAB 3
Metodologi

Penelitian ini akan dilakukan dengan beberapa tahapan proses. Tahapan pertama yakni pengambilan dataset yang diikuti oleh eksplorasi terhadap data tersebut. Tahap selanjutnya adalah melakukan *preprocessing* pada tiap teks dalam dataset. Teks-teks yang sudah standar setelah melalui *pre-processing* kemudian ditokenisasi sehingga siap digunakan sebagai *input* dari model Bi-LSTM. Proses pemodelan menjadi tahapan terakhir dalam penelitian ini dengan diikuti oleh evaluasi dari hasil pemodelan. Lebih jelas mengenai tahapan-tahapan tersebut dapat dilihat pada Gambar 3.1.



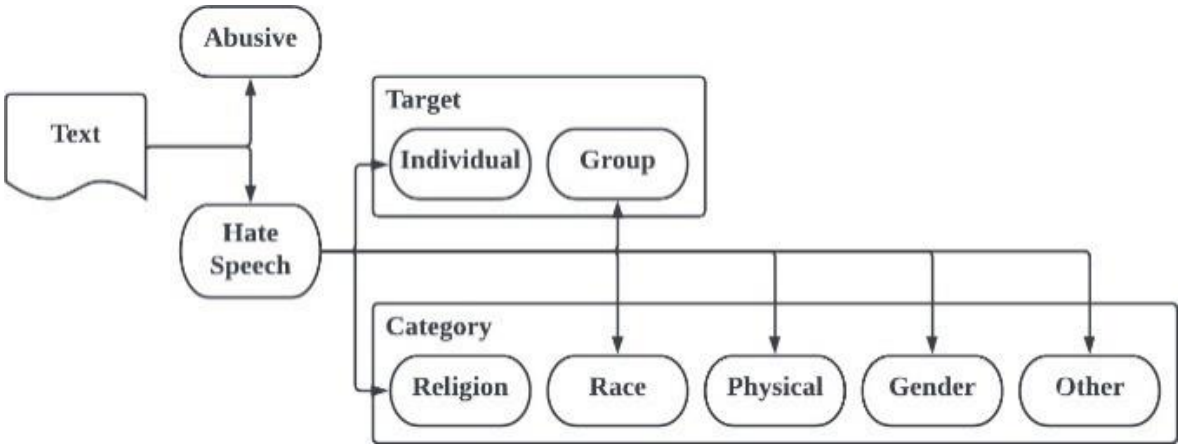
Gambar 3.1. Metodologi penelitian.

3.1. Data Collection

Dataset yang akan digunakan dalam penelitian ini adalah dataset ujaran kebencian berbahasa Indonesia yang dihasilkan dari penelitian (Ibrohim & Budi, 2019). Dataset tersebut merupakan dataset yang dapat diakses secara publik melalui github (https://github.com/okkyibrohim/id-multi-label-hate-speech-and-abusive-language-detection). Dataset dibangun dari teks (*twit*) yang diperoleh dari hasil *crawling* menggunakan Twitter API. *Twit-twit* tersebut kemudian dianotasi oleh 30 orang annotator ke dalam 12 label. Proses anotasi dari dataset juga melibatkan ahli dalam *Forum Group Discussion* (FGD). Ahli yang dimaksud adalah staf Direktorat Tindak Pidana Siber Bareskrim Polri dan seorang ahli Bahasa bernama Muzaina Nurasijah. Pada tahap akhir, dataset dirilis dengan jumlah data sebanyak 13.169 twit dengan masing-masing twit diberikan label berjumlah 12 yakni *hate speech, abusive, individual, group, race, religion, physical, gender, others, strong, moderate,* dan *weak*. Tiap label bersifat *binary* dengan nilai

1 atau 0. Nilai 1 (positif) menandakan bahwa twit tergolong ke dalam label. Sedangkan nilai 0 (negatif) menandakan bahwa twit tidak tergolong ke dalam label.

Pada penelitian ini, label yang digunakan untuk penelitian hanya 9 dari 12 label yang tersedia. Hal ini didasarkan pada hasil penelitian yang dilakukan (Ibrohim, Setiadi & Budi, 2019). Penelitian tersebut melakukan eksperimen terhadap beberapa skenario terkait label-label yang digunakan dalam pembangunan model klasifikasi ujaran kebencian. Penelitian mendapatkan performa terbaiknya pada skenario dengan menggunakan 9 label dengan mencapai akurasi sebesar 68,43%. Kesembilan label yang dimaksud antara lain *hate speech*, *abusive*, *individual*, *group*, *race*, *religion*, *physical*, *gender*, dan *others.*



Gambar 3.2. Hirarki label pada dataset.

Gambar 3.2 menunjukkan ilustrasi dari hierarki tiap label dalam dataset. Aspek ujaran kebencian dapat digolongkan menjadi 2 garis besar yaitu berdasarkan target dan kategorinya. Ujaran kebencian dalam dataset memiliki 2 jenis target yang diarahkan kepada individu atau kelompok. Sedangkan untuk aspek lainnya dapat ditujukan untuk menyerang target berdasarkan *religion* (agama), *race* (suku), *physical* (fisik), *gender* (jenis kelamin), dan *others* (lainnya) [10].

- *Religion*, teks yang digolongkan ke dalam label ini adalah teks-teks yang ditujukan untuk menyerang suatu agama (Islam, Kristen, Katolik, Hindu, Budha, dan lainnya) ataupun suatu organisasi keagamaan/kepercayaan tertentu [10].
- *Race*, teks yang digolongkan ke dalam label ini adalah teks-teks yang ditujukan untuk menyerang ras/suku tertentu [10].

- *Physical*, teks yang digolongkan ke dalam label ini adalah teks-teks yang ditujukan untuk menyinggung target menggunakan kekurangan/perbedaan fisik (wajah, mata, dan bagian tubuh lainnya) [10].
- *Gender*, teks yang digolongkan ke dalam label ini adalah teks-teks yang menyerang target dengan merendahkan gender (jalang, gigolo, dan lainnya) atau berorientasi seksual (banci, lesbi, homo, dan lainnya) [10].
- *Other*, teks yang digolongkan ke dalam label ini adalah teks-teks yang menyerang target dengan kata-kata yang tidak berkaitan dengan empat kategori di atas [10].

Adapun sampel dari dataset yang akan digunakan dalam penelitian kali ini disajikan dalam Tabel 3.1.

.

Tabel 3.1. Sampel teks pada dataset.

Teks	Label								
	Hate Speech	**Abusive**	**Individual**	**Group**	**Religion**	**Race**	**Physical**	**Gender**	**Others**
Cina perusak bangsa!! Usir !! Stuju??	Positif	Negatif	Negatif	Positif	Negatif	Positif	Negatif	Negatif	Negatif
USER lebih baik pilih ahok drpd pilih yg se iman tp koruptor dn munafik !! Bodoh kau	Positif	Positif	Positif	Negatif	Positif	Negatif	Negatif	Negatif	Negatif
Dibalik Insiden Heli Jatuh, PT IMIP Pekerjakan 3.000 WN China	Negatif	Negatif	Negatif	Negatif	Negatif	Negatif	Negatif	Negatif	Negatif
Siapapun gubernur dan presidennya, rakyatnya, ya kita² juga...	Negatif	Negatif	Negatif	Negatif	Negatif	Negatif	Negatif	Negatif	Negatif

3.2. Data understanding

Data understanding dilakukan dengan melakukan eksplorasi terhadap data. Hal ini diperlukan untuk mengetahui bagaimana kondisi dari dataset yang digunakan dalam penelitian. *Data understanding* akan sangat membantu dalam menentukan tindakan apa yang akan dilakukan terhadap data pada tahap *preprocessing*

3.3. Pre-processing

Setiap orang memiliki gaya menulis yang berbeda-beda walaupun kata yang disebutkannya adalah sama. Contoh kecil saja bisa dilihat pada penulisan kata “tunggu”. Seseorang bisa saja menulis kata tersebut dalam bentuk yang tidak formal misalkan dengan menulis “tg” atau “tgu”. Kondisi tersebut mengharuskan proses modeling pada NLP untuk melakukan memproses data tersebut terlebih dahulu untuk menjadi bentuk yang lebih standar. Proses tersebut selanjutnya bisa dikatakan sebagai *pre-processing.*

Pada penelitian ini, direncanakan untuk menggunakan 4 teknik dari *pre*-processing yaitu *case folding*, *cleaning*, dan *normalization.* Pada proses *cleaning*, teks-teks yang tidak diperlukan seperti *username*, link, *punctuation*, angka, dan *stopwords* akan dihapus dari teks. Kemudian, proses dilakukan untuk melakukan *normalization*. *Normalization* akan dilakukan untuk mengonversi kata-kata slang menjadi satu kata bakunya. Proses normalisasi dilakukan dengan menggunakan suatu *dictionary* yang sudah dibangun oleh (Ibrohim & Budi, 2019).

3.4. Tokenizing

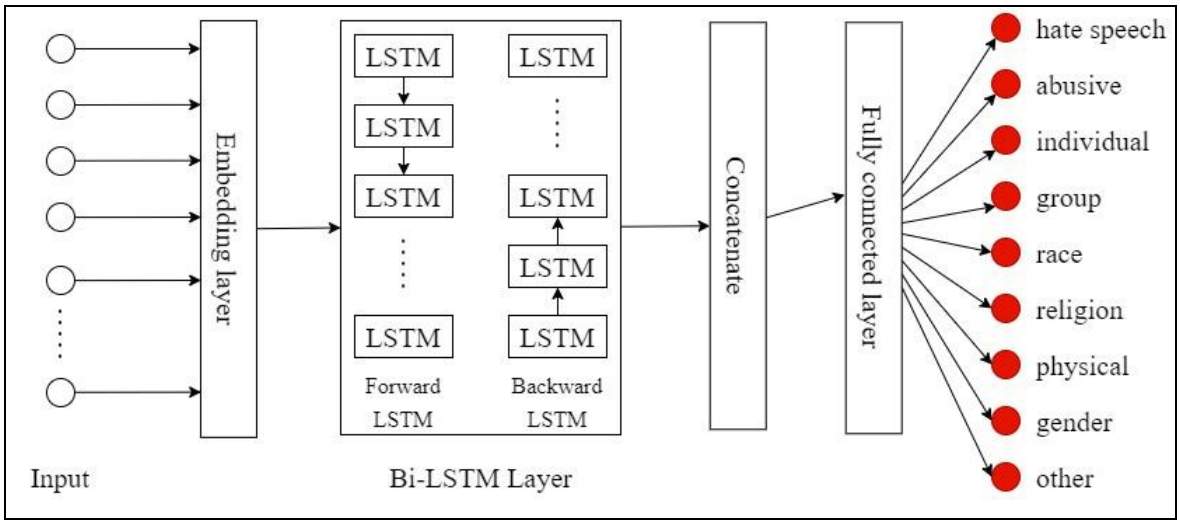
Pada penelitian ini, algoritma *embedding* yang direncakan adalah menggunakan model *pretrain* dari BERT (*Bidirectional Encoder Representations from Transformers*). Salah satu *pre-train* model dari BERT yang menangani kasus Bahasa Indonesia adalah IndoBERT [27]. *Pretrain* IndoBERT sudah pernah digunakan dalam penelitian tentang ujaran kebencian Bahasa Indonesia dengan performa akurasi paling tinggi yaitu 84,77% dibandingkan dengan 2 algoritma *embedding* lainnya yaitu word2vec dan fasttext [2].

Model IndoBERT yang akan digunakan dalam penelitian ini adalah "cahya/bert-base-indonesian-522M". Model tersebut dibangun menggunakan dataset yang berasal dari Wikipedia dengan total *corpus* sebanyak 32.000 kata. Selain model tersebut, beberapa model *pre-train* juga akan diujikan pada penelitian. Hal ini bertujuan untuk melihat *pre-train* model terbaik untuk menagani kasus ujaran kebencian *multilabel* berbahasa Indonesia. Model-model yang dimaksud antara lain “indobenchmark/indobert-base-p1”,

“indobenchmark/indobert-large-p2”, “indolem/indobertweet-base-uncased”, “ayameRushia/indobert-base-uncased-finetuned-indonlu-smsa”, dan “afbudiman/indobert-classification”.

3.5. Modeling

Pada penelitian ini, pemodelan akan dilakukan menggunakan Bi-LSTM. Dalam penelitian mengenai ujaran kebencian Bahasa Indonesia, Bi-LSTM sendiri telah tercatat mampu memberikan performa akurasi yang sangat baik dengan nilai akurasi mencapai 94% untuk mendeteksi teks sebagai ujaran kebencian atau tidak [9]. Adapun detail arsitektur dari model Bi-LSTM dapat dilihat pada Gambar 3.3.



Gambar 3.3. Arsitektur pemodelan dengan Bi-LSTM.

Gambar 3.3 menyajikan arsitektur dari model Bi-LSTM yang digunakan dalam penelitian. *Input* dari model Bi-LSTM adalah list dari token yang dihasilkan pada tahap tokenisasi. Tiap *scalar* dalam *input* tersebut kemudian dikonversi menjadi vektor pada layer *embedding.* Proses tersebut dibutuhkan karena layer Bi-LSTM membutuhkan *input* berupa vektor untuk proses mendapatkan *hidden state* disetiap *node*-nya.

. Pada layer Bi-LSTM, terdapat 2 baris dari *node-node* LSTM yang merepresentasikan proses *forward* dan *backward*. Proses tersebut merupkan bentuk dari penerapan *bidirectional* pada model Bi-LSTM. *Output* dari layer Bi-LSTM kemudian akan di *concatenate* dan masuk ke layer terakhir yaitu *fully connected layer* (FCL).

FCL memiliki *node* sebanyak 9 yang merupakan representasi 9 label pada dataset ujaran kebencian. Layer ini memiliki *output* berupa list dengan elemen biner. Tiap index dari list akan merepresentasikan hasil analisa model terhadap teks untuk tiap label. Sebagai

contoh, jika model memberikan *output* 1 pada element k 0,2 dan 5 maka teks tergolong ke dalam label *hate speech, individual,* dan *religion*.

Dalam proses pemodelan, penelitian perlu melakukan beberapa kali percobaan guna mendapatkan arsitektur terbaik untuk mendeteksi ujaran kebencian dengan *output multilabel*. Percobaan-percobaan dilakukan dengan memvariasikan nilai dari beberapa parameter seperti *epoch, node* pada layer Bi-LSTM, *learning rate* dan *batch size*. Tidak lupa juga dilakuakn percobaan untuk menggunakan variasi *pre-train* dari model IndoBERT sebagai model tokenisasi guna menunjang hasil analisa dari model Bi-LSTM.

1. Jumlah *epoch*

   Initial value dari *epoch* dalam pemodelan adalah 10. Proses *tunning* kemudian dilakukan dengan menambah jumlah *epoch* sebanyak 10 hingga mencapai 100 *epoch*.

2. Jumlah *node* layer Bi-LSTM

   Model pada awalnya menggunakan arsitektur dengan 10 *node* pada layer Bi-LSTM. Selanjutnya, parameter ini di-*tunning* dengan menambah *node* sebanyak 10 hingga mencapai 100 *node.*

3. *Learning rate*

   Initial value dari *learning rate* dalam pemodelan adalah 1e-1 atau 0,1. Proses *tunning* kemudian dilakukan dengan menurunkan *learning rate* sebesar sepersepuluh sehingga menjadi 1e-2 , 1e-3, 1e-4, dan 1e-5.

4. *Batch size*

   Initial value dari *batch size* dalam pemodelan adalah 128. Selanjutnya, proses *tunning* dilakukan dengan mencoba menurunkan *batch size* menjadi 96, 64 dan 32. Selain itu, *batch size* juga akan coba dinaikkan dari *initial value* menjadi 160, 192, 224, dan 256.

5. *Pre-train* model IndoBERT

   Guna mendukung performa dari model Bi-LSTM, variasi model IndoBERT juga digunakan dalam pemodelan. Model pertama yang digunakan adalah “cahya/bert-base-indonesian-522M”. Kemudian, dilanjutkan dengan mencoba beberapa *pre-train* model lain seperti

   - indobenchmark/indobert-base-p1,
   - indobenchmark/indobert-large-p2,
   - indolem/indobertweet-base-uncased,
   - ayameRushia/indobert-base-uncased-finetuned-indonlu-smsa, dan
   - afbudiman/indobert-classification

3.6. Evaluasi

Evaluasi dibutuhkan untuk melihat seberapa baik performa model dalam melakukan deteksi ujaran kebencian pada teks. Evaluasi dilakukan di setiap proses *tunning* parameter pemodelan. Penentuan nilai terbaik dari tiap parameter akan mempertimbangkan hasil dari matriks klasifikasi. Ketiga matriks yang dimaksud adalah presisi (1), *recall* (2), dan akurasi (3).

$$presisi = \frac{TP}{TP+FP} \quad (1)$$

$$recall = \frac{TP}{TP+FN} \quad (2)$$

$$akurasi = \frac{TP+FP}{TP+TN+TP+FP} \quad (3)$$

TP (*True Positive*) pada persamaan (1), (2), dan (3) merepresentasikan seberapa banyak data positif yang berhasil diprediksi benar sebagai data positif oleh model. Sedangkan TN (*True Negative*) merepresentasikan seberapa banyak data negatif yang diprediksi dengan oleh model sebagai data negatif. Kemudian FP (*False Positive*) merepresentasikan banyak data negatif yang hasil prediksinya salah karena diprediksi sebagai data positif. Sebaliknya, FN (*False Negative*) merepresentasi jumlah data positif yang salah diprediksi oleh model menjadi data negatif.

Dalam implementasinya, proses evaluasi model pendeteksi ujaran kebencian *multilabel* menerapkan teknik k-fold validation. Dataset dalam penelitian akan dibagi menjadi sejumlah *k*. Jika nilai *k* yang digunakan adalah 5 maka dataset akan dibagi menjadi 5 kelompok data. Ketika pemodelan berlangsung, satu kelompok data akan bertindak sebagai data *test* sedangkan 4 kelompok data lainnya menjadi data *training*. Proses pemodelan akan terus dilakukan sampai semua kelompok data berperan sebagai data *test*.

Fold	K1	K2	K3	K4	K5
1					
2					
3					
4					
5					

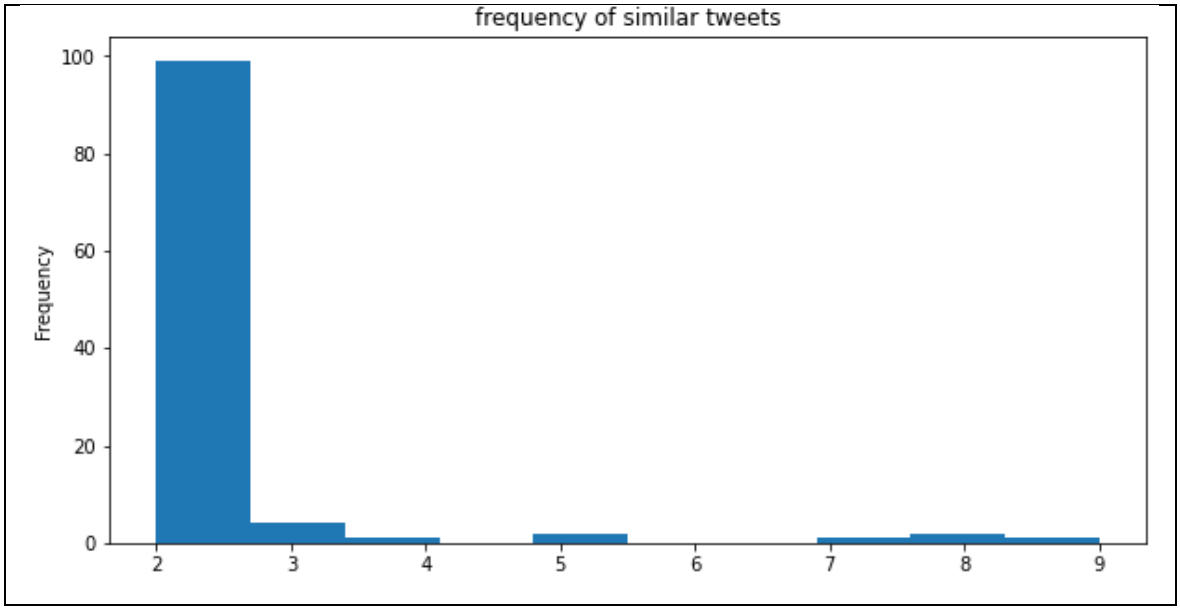
Gambar 3.4. Ilustrasi k-fold validation.

Pada Gambar 3.4 merupakan ilustrasi dari *k-fold validation* menggunakan nilai *k* sebesar 5. Header dari kolom K1 sampai K5 merepresentasikan urutan dari kelompok dataset yang terbentuk. Pada gambar tersebut bisa dilihat terdapat *cell* yang berwarna kuning. Warna kuning tersebut menandakan bahwa kelompok dataset tersebut akan digunakan sebagai data *test* saat pemodelan dan kelompok lainnya berperan sebagai data *test*.

BAB 4
Hasil dan Pembahasan

4.1. Eksplorasi data dan pre-processing

Penelitian diawali dengan eksplorasi dataset yang sudah didapatkan pada Github. Eksplorasi data bertujuan untuk mengetahui bagaimana kondisi awal dari dataset. Eksplorasi pada menemukan fakta bahwa terdapat sebanyak 146 teks terduplikasi dalam dataset. Pada Gambar 4.1, hampir 100 teks terduplikasi sebanyak 2 kali. Bahkan, ada teks yang terduplikasi sebanyak sampai 9 kali. Teks yang duplikat tersebut kemudian dieliminasi dari dataset. Dengan ini, dataset masih menyisakan sebanyak 13.023 teks.



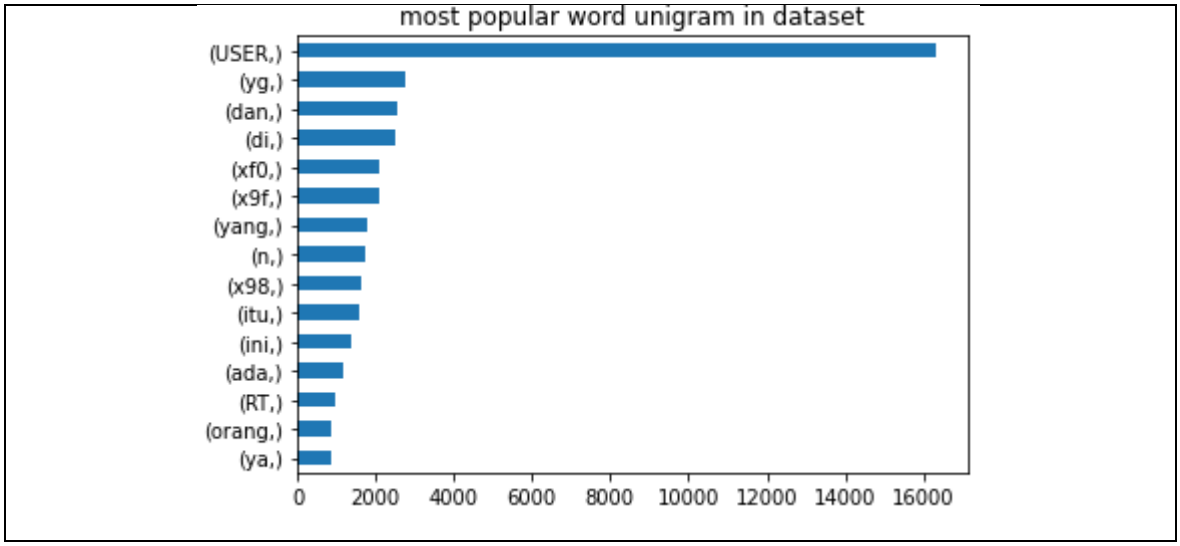
Gambar 4.1. Histogram frekuensi teks duplikat.

Proses persiapan data kemudian berlanjut ke tahap *pre-processing.* Tahap ini dimulai dengan mengubah *case* pada teks menjadi huruf kecil atau biasa disebut sebagai *case folding*. *Case folding* bertujuan untuk mempermudah proses pencarian kata pada saat proses *cleaning* dan *normalization*. Selain itu, *case folding* juga diperlukan ketika proses tokenisasi. Kata yang sama namun memiliki *case* yang berbeda bisa saja dianggap sebagai 2 indeks yang berbeda pada saat tokenisasi karena adanya perbedaan nilai dari sisi ASCII.

Pre-processing selanjutnya memasuki tahap *cleaning*. Pada Gambar 4.4 terlihat bahwa beberapa kata memiliki porsi yang besar seperti kata “USER”, “yg”, dan emoji (xf0, x9f, dan lainnya). Kata “USER” bahkan muncul 2 kali dengan porsi besar dalam *wordcloud*. Jika diperhatikan lebih detail lagi, kata “USER” juga kembali muncul dengan porsi yang

lebih kecil lagi. Dominasi tersebut terjadi karena tingkat kemunculan kata “USER” sangat tinggi dalam *corpus*. Dilihat dari Gambar 4.3, kata tersebut muncul sebanyak lebih dari 16ribu kali. Jumlah tersebut sangat jauh dari kata “yg” diurutkan kedua yang tingkat kemunculannya tidak lebih dari 4ribu kali.

Gambar 4.2. Dataset *wordcloud* sebelum *pre-processing.*



Gambar 4.3. Tingkat kemunculan kata pada dataset.

Intensitas kemunculan yang tinggi tidak lantas menjadi patokan suatu kata menjadi diperlukan dalam proses pemodelan. Kata “USER”, “yg”, ataupun emoji (xf0, x9f, x9f, dan lainnya) yang memiliki intensitas yang tinggi dirasa kurang memiliki makna sebagai penentu

suatu teks tergolong sebagai ujaran kebencian atau tidak. Oleh karena itu, kata-kata tersebut dihapus pada proses *cleaning*. Selain kata-kata tersebut, beberapa kata dan karakter yang juga turut dihapus antara lain tanda baca, angka, *hashtag*, *link*, *single character*, *multiple space*, dan *stopwords*.

Output dari proses *cleaning* kemudian dibawa ke tahap *normalization. Normalization* dilakukan dengan untuk mentransformasi kata-kata "alay" atau kata yang ditulis dengan bentuk tidak formal menjadi bentuk yang lebih standar. Proses ini memanfaatkan kamus "alay" dari [10] yang berisi daftar kata yang perlu ditransformasi berikut dengan hasil transformasinya.

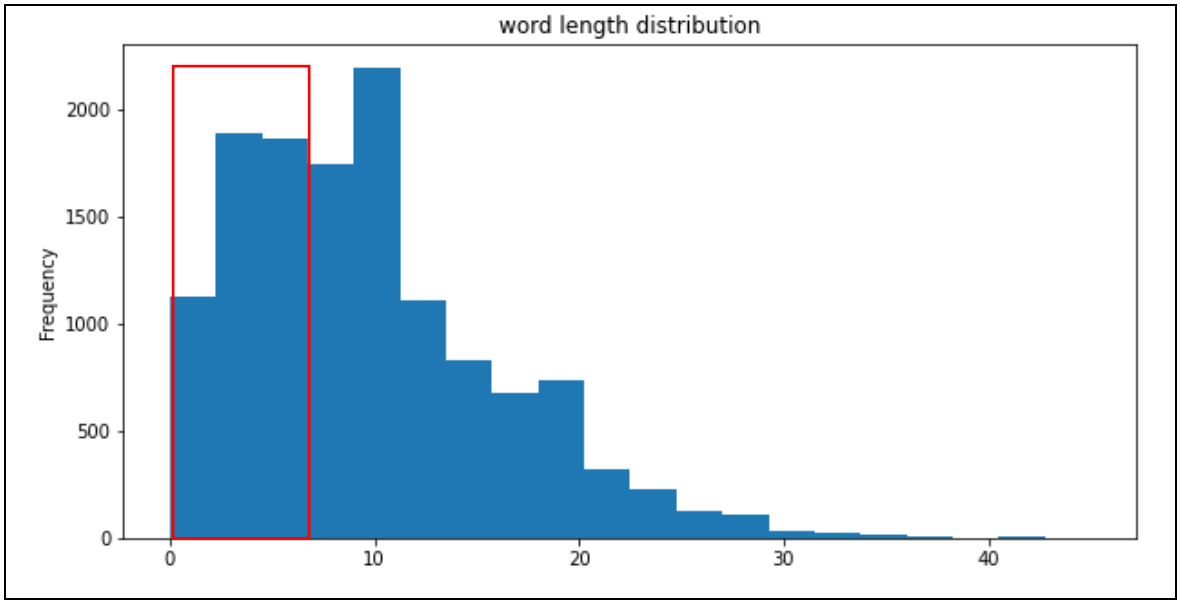
Normalization pada teks berguna untuk meminimalisir *variance* dalam dataset terkait dengan perbedaan penulisan pada kata yang memiliki makna sama. Sebagai contoh, seseorang mungkin saja menulis kata "tunggu" dengan tulisan yang formal. Tapi di lain sisi, seseorang lainnya menuliskan kata tersebut dengan "tgu". Kedua penulisan tersebut dalam proses tokenisasi akan dianggap sebagai 2 buat kata yang berbeda. Padahal, pada kenyataannya kedua kata tersebut adalah kata yang sama.

Gambar 4.4 *Wordcloud* dari dataset setelah preprocessing.

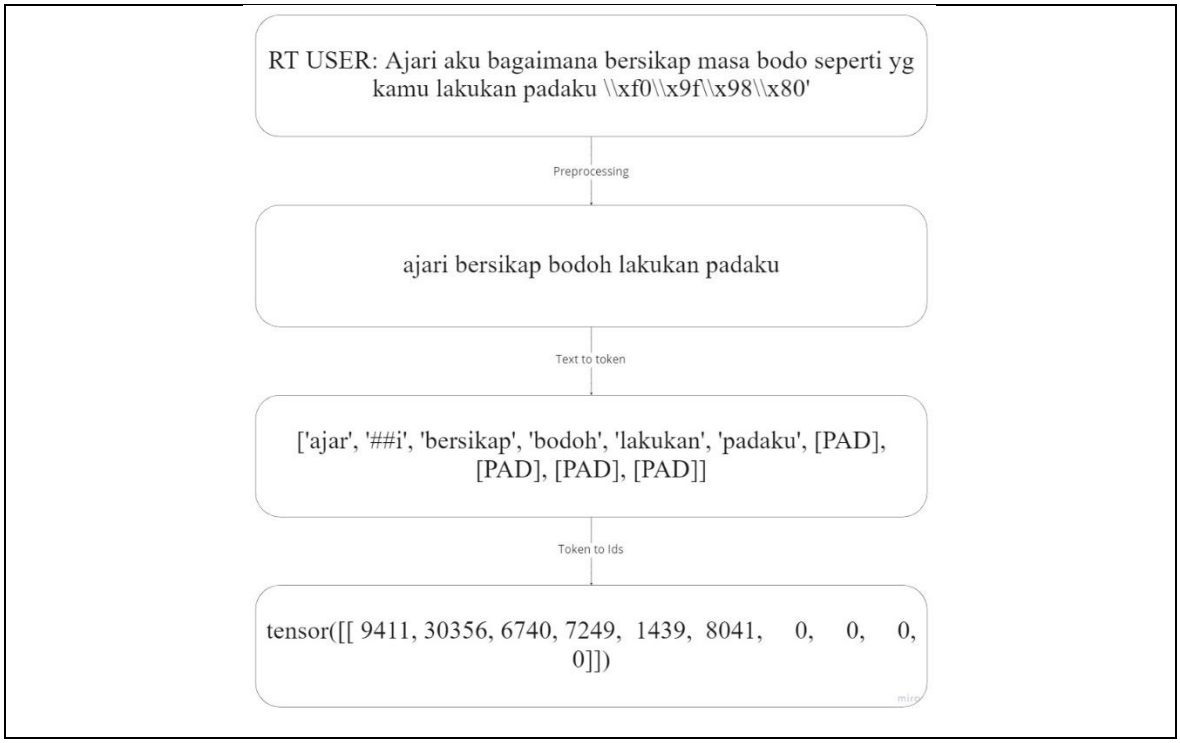
Pre-processing berhasil mereduksi dominansi dari kata-kata yang sebelumnya memiliki intensitas kemunculan yang tinggi. Pada Gambar 4.4, terlihat *wordcloud* menghasilkan bentuk yang baru setelah *pre-processing.* Kata "USER" dan emoji sepenuhnya sudah dieliminasi dari *corpus*. Begitu pula dengan kata "yg" yang dinormalisasi menjadi "yang". Kata tersebut juga hilang karena masuk dalam list *stopwords* Bahasa Indonesia.

Hilangnya karakter atau kata dalam teks setelah *pre-processing* tentu berimbas pada berkurangnya panjang kata pada teks tersebut. Jumlah kata pada teks berpengaruh pada panjang token yang dihasilkan pada proses tokenisasi oleh BERT. Dari Gambar 4.5, dapat

dilihat bahwa panjang kata pada teks setelah *pre-processing* dominan terletak pada angka 10. Oleh karena itu, panjang token yang akan dihasilkan pada tokenisasi BERT didefinisikan sebanyak 10. Sepuluh token tersebut menjadi *input* bagi model model Bi-LSTM.



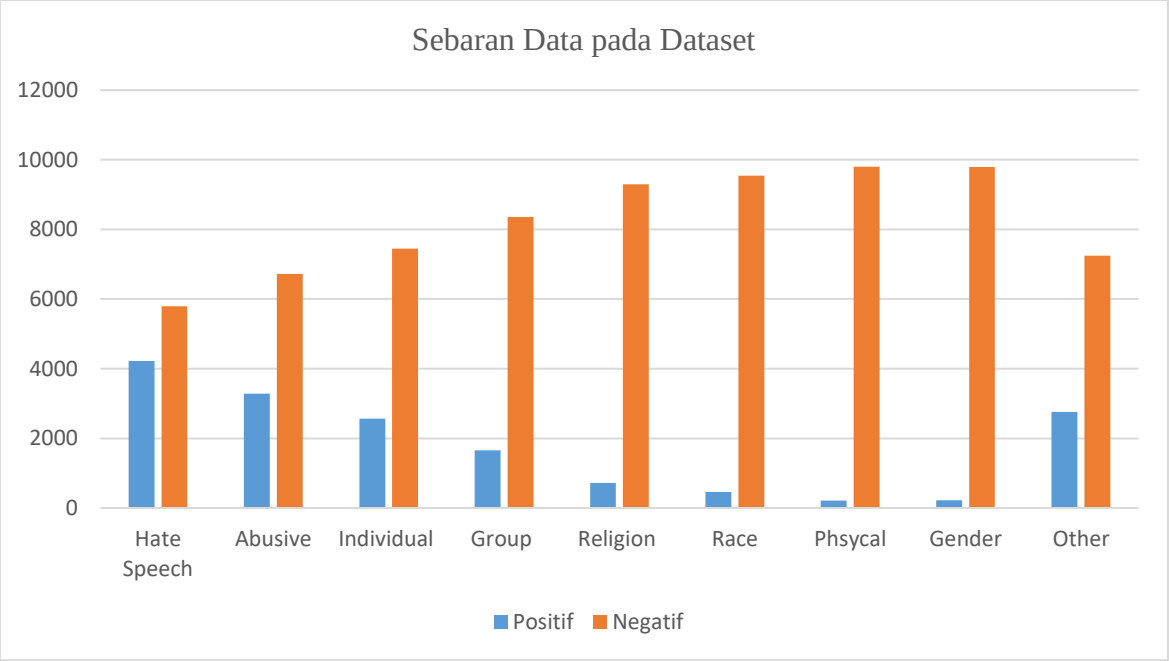
Gambar 4.5 Distribusi Panjang kata tiap teks pada dataset.



Gambar 4.6. Ilustrasi proses tokenisasi oleh IndoBERT dengan panjang maksimum token sebanyak 10.

Di sisi lain, Gambar 4.5 menunjukkan bahwa tidak sedikit teks memiliki panjang kata di bawah 10. Teks dengan panjang kata sama dengan di bawah 5 terlihat memiliki histogram yang hampir setara dengan panjang teks 10. Hal ini memutuskan peneliti untuk mereduksi data dengan mengeliminasi teks yang memiliki panjang kata kurang dari sama dengan 5. Keputusan tersebut dilakukan dengan tujuan untuk mengurangi data yang mengandung token [PAD. Token [PAD] muncul sebagai *padding* untuk menutupi token yang belum terisi apabila token yang dihasilkan kurang dari 10. Lebih jelas mengenai bagaimana hasil tokenisasi oleh BERT dapat dilihat pada ilustrasi Gambar 4.6.

Jumlah teks yang memiliki panjang kata di bawah 5 tercatat ada sebanyak 3.012 data. Sejumlah data tersebut kemudian dieliminasi dan menyisakan 10.011 teks dalam dataset. Teks-teks tersebut kemudian akan menjadi dataset untuk melakukan pemodelan dalam penelitian ini. Adapun mengenai sebaran data positif dan negatif pada tiap label disajikan pada Gambar 4.7.



Gambar 4.7. Sebaran data pada dataset setelah *pre-processing*.

Pada Gambar 4.7, terlihat bahwa dataset berada dalam kondisi yang tidak seimbang antara nilai positif dan negatif pada tiap label. Untuk itu, penelitian ini menerapkan teknik *class weighting* dalam pemodelan. *Class weighting* adalah salah satu teknik yang bisa diterapkan untuk menangani kasus data *imbalance* . Cara kerja dari teknik ini adalah dengan memberikan *weight* kepada tiap label ketika proses optimasi model [28]. Adapun cara yang digunakan untuk menghitung *weight* tiap label adalah menggunakan persamaan 4.1.

$$CW_i = \frac{1}{\sqrt{N_i}} \quad (4.1)$$

CW_i merupakan representasi dari *class weight* pada label ke-i. Nilai i menunjukkan indeks dari tiap label. Misalkan untuk label "hate speech" memiliki indeks ke-0 atau i bernilai 0. Kemudian N_i merepresentasikan jumlah nilai positif pada label dengan indeks ke- i.

Class weight memberikan *weight* yang lebih besar pada label yang memiliki jumlah data positif sangat kecil seperti *physical* dan *gender*. Besarnya *weight* tersebut akan memberikan fokus pada proses optimasi oleh *loss function* untuk memperbaiki tingkat kesalahan pada kelas dengan data positif [29].

Dengan demikian, proses bisa dilanjutkan ke tahap pemodelan. Proses pemodelan akan menerapkan *class weighting* dan akan mengolah dataset yang sudah dihasilkan pada tahap *pre-processing*.

4.2. Pemodelan

Proses pemodelan dilakukan dengan melakukan *tunnig* terhadap parameter *epoch*, *node* layer Bi-LSTM, *learning rate*, *batch size*, serta *pre-train* model IndoBERT. Sebelum masuk ke proses *tunning* parameter, penelitian memberikan *initial value* terhadap parameter-parameter tersebut. *Initial value* yang diberikan adalah 10 *epoch*, 10 *node* layer Bi-LSTM, 1e-1 *learning rate*, 128 *batch size*, dan *pre-train* model IndoBERT dari "cahya/bert-base-indonesian-522M" untuk tokenisasi.

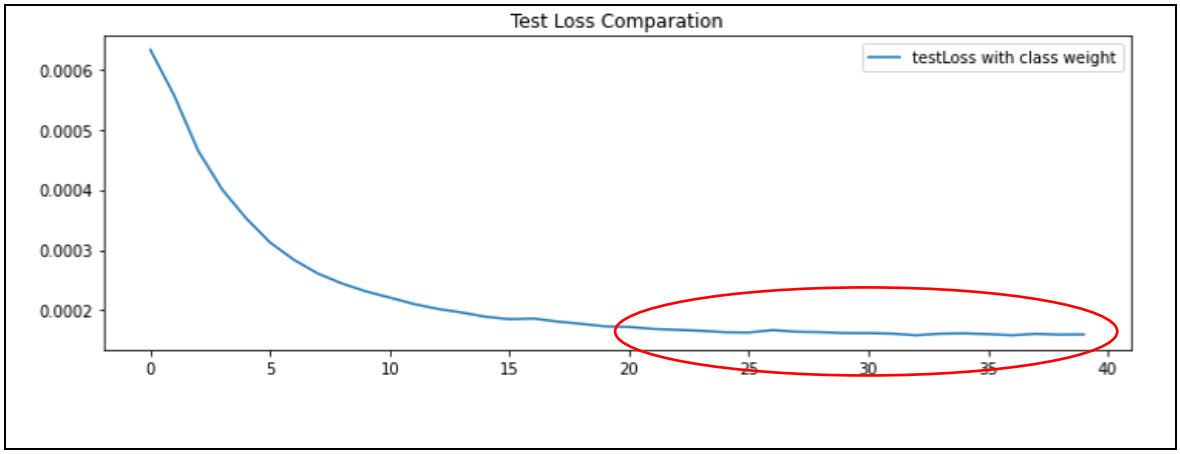
Proses pemodelan divalidasi dengan menerapkan *cross validation (K-Fold).* Model terbaik dari tiap proses *tunning* parameter akan ditentukan dengan mempertimbangkan performa dari model dalam mendeteksi ujaran kebencian dengan *output multilabel*. Performa tersebut diukur menggunakan matriks akurasi, presisi, dan *recall.*

4.2.1. Hasil percobaan terhadap parameter *epoch*

Proses *tunning* pertama kali dilakukan terhadap parameter *epoch*. Percobaan untuk melakukan *tunning* terhadap parameter *epoch* dilakukan untuk mengetahui berapa nilai *epoch* terbaik bagi model untuk melakukan mempelajari dataset. Proses *tunning* dimulai dengan *epoch* sebanyak 10. Selanjutnya, jumlah *epoch* ditambahkan sebanyak 10 dan berhenti pada 40 *epoch*.

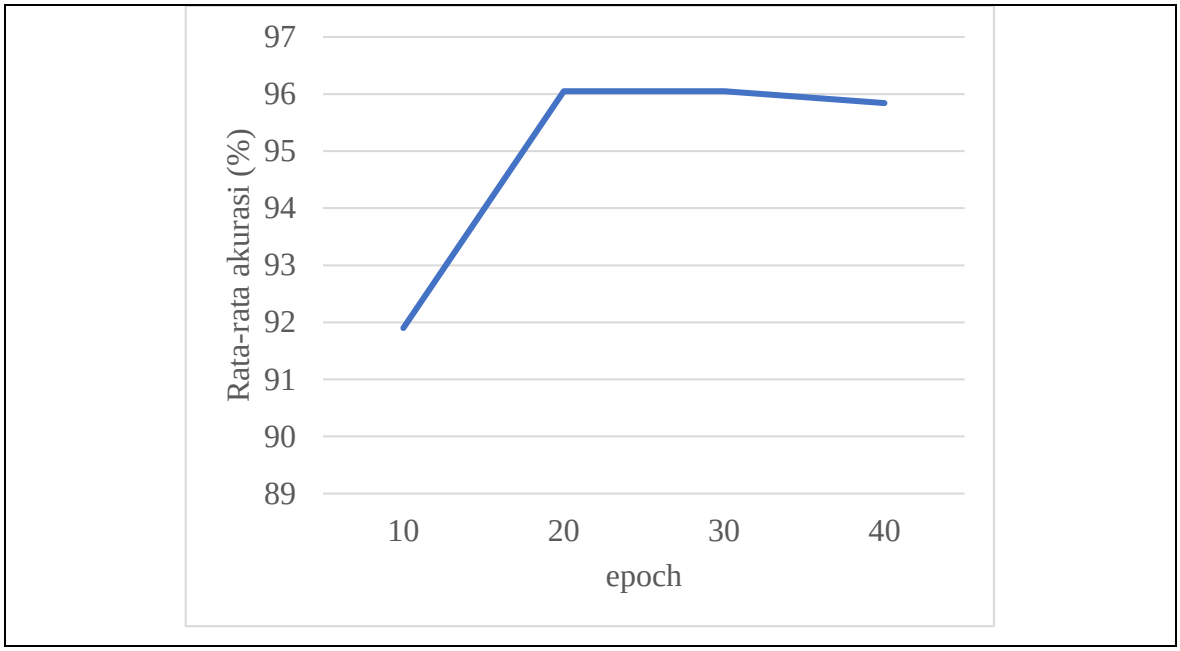
Percobaan dihentikan ketika menggunakan 40 *epoch* karena mempertimbangkan nilai *loss* pada Gambar 4.8. Pada awalnya, *loss* mengalami penurunan yang signifikan ketika

masuk ke *epoch 20*. Namun, pada *epoch* setelahnya nilai *loss* tidak mengalami perubahan yang signifikan lagi. Perubahan *loss* terkesan datar ketika pemodelan mencapai *epoch* 30 dan 40.



Gambar 4.8 *Loss* dari pemodelan dengan 40 epoch.

Dari sisi akurasi, performa model berbanding lurus dengan nilai *loss* pada Gambar 4.8. Pada Gambar 4.9, rata-rata akurasi pada 20 *epoch* berhasil meningkat sebanyak 3,15% dari 91.90% menjadi 96.05%. Sedangkan, akurasi tidak mengalami perubahan yang signifikan pada *epoch* 30 dan 40. Rata-rata akurasi yang didapatkan secara berturut-turut sebesar 96.05% dan 95.84% untuk *epoch* 30 dan 40. Dengan demikian, percobaan ini memutuskan bahwa *epoch* terbaik untuk proses pemodelan adalah 20.



Gambar 4.9. Komparasi hasil percobaan pada jumlah *epoch*.

Namun, performa pemodelan dengan 20 *epoch* masih memiliki catatan dari sisi *recall*. Pada Tabel 4.1, label dari "physical" dan "gender" masih menunjukkan performa *recall* yang rendah. *Recall* dari kedua label tersebut berada pada level 50%. Untuk itu, pada percobaan selanjutnya perlu diperhatikan kembali performa *recall* yang diberikan model.

Tabel 4.1. Performa model dengan 20 *epochs*.

Label	Akurasi	Presisi	Recall
Hate Speech	97,75%	97,39%	97,27%
Abusive	90,14%	90,49%	78,18%
Individual	93,90%	92,77%	82,61%
Group	95,24%	91,36%	78,61%
Religion	96,76%	89,49%	61,96%
Race	98,03%	93,18%	61,99%
Physical	98,87%	90,68%	**51,20%**
Gender	98,96%	94,57%	**55,71%**
Other	94,83%	94,70%	86,06%

4.2.2. Hasil tunning pada node layer Bi-LSTM

Proses *tunning* selanjutnya dilakukan terhadap jumlah *node* di layer Bi-LSTM. Jumlah *node* akan berpengaruh pada kedalaman dari layer Bi-LSTM dalam melakukan proses perhitungan *hidden state*. Semakin banyak jumlah *node* yang diterapkan pada maka proses komputasi akan semakin panjang.

Pada percobaan sebelumnya, model berhasil memberikan performa akurasi sebesar 96,05%. Model tersebut dibangun dengan menggunakan 10 *node* layer Bi-LSTM. Selanjutnya, *tunning* parameter dicoba dengan menambah 10 *node* hingga mencapai 60 *node*. Adapun hasil yang didapatkan tersaji pada Tabel 4.2.

Tabel 4.2. Komparasi jumlah *node* pada layer Bi-LSTM.

No	Bi-LSTM Node	Akurasi
1	10	96.05%
2	20	96,08%
3	30	**96,42%**
4	40	**96,39%**
5	50	95,86%
6	60	94,96%

Pada Tabel 4.2, performa model terus mengalami peningkatan hingga mencapai puncak rata-rata akurasi pada model dengan 30 *node* Bi-LSTM. Rata-rata akurasi dari model

berhasil mencapai 96,42%. Setelahnya, performa model terus mengalami penurunan hingga menjadi 94,96% pada percobaan terakhir yaitu dengan 60 *node* Bi-LSTM. Hasil tersebut menunjukkan bahwa kompleksitas model ternyata tidak berbanding lurus dengan performa yang diberikan oleh model.

Namun, jika diperhatikan lagi ternyata model dengan 30 dan 40 *node* Bi-LST tidak memiliki margin yang besar. Rata-rata akurasi dari kedua model hanya terpaut 0,03% saja. Untuk itu, Analisa lebih lanjut dilakukan terhadap performa kedua model dilihat dari sisi *recall*. Lebih jelas mengenai perbandingan *recall* dari kedua model bisa mengacu pada (Tabel 4.3).

Tabel 4.3 Komparasi performa *recall* model dengan 30 dan 40 *nodes*

Labels	30 nodes	40 nodes
Hate Speech	**97,23%**	97,18%
Abusive	**81,95%**	78,00%
Individual	83,50%	**85,14%**
Group	78,61%	**79,34%**
Religion	68,39%	**71,33%**
Race	73,87%	**75,81%**
Physical	59,81%	**68,42%**
Gender	59,36%	**67,12%**
Other	86,78%	**88,75%**

Pada Tabel 4.3, Model dengan 40 *node* Bi-LSTM berhasil mengungguli model dengan 30 *node* di sebagian besar label. Terlebih lagi pada label *physical* dan *gender*, model berhasil meningkatkan nilai *recall* dari percobaan sebelumnya (Tabel 4.1). *Recall* dari kedua label tersebut berhasil melewati angka 65% dari yang sebelumnya hanya 50%. Oleh karena itu, percobaan ini memutuskan bahwa jumlah *node* terbaik untuk diterapkan ke dalam layer Bi-LSTM adalah 40 *node*. Hasil ini kemudian akan dibawa pada penelitian selanjutnya untuk menguji parameter *learning rate*.

4.2.3. Hasil tunning pada parameter learning rate

Proses *tunning* terhadap parameter *learning rate* bertujuan mengetahui besar *learning* rate yang paling optimal untuk diterapkan ke dalam pemodelan. Besarnya nilai *learning* rate akan berdampak pada besar perubahan *weight* yang pada proses pemodelan. Semakin besar *learning rate* yang digunakan maka semakin besar perubahan *weight* dari model di tiap

optimasinya. Begitu juga sebaliknya, semakin kecil *learning rate* yang digunakan, maka optimasi *weight* dari model akan semakin pendek.

Pada percobaan sebelumnya, performa model untuk *learning rate* sebesar 1e-1 sudah didapatkan yaitu 96,39%. Proses kemudian dilanjutkan dengan memperkecil dan memperbesar nilai dari *learning rate* menjadi 5e-1, 1e-2, dan 1e-3. Hasil dari percobaan ini disajikan pada Tabel 4.4.

Tabel 4.4. Komparasi performa dari percobaan *learning rate.*

No	Learning rate	Akurasi
1	0,1	**96.05%**
2	0,01	94,47%
3	0,001	82,14%
4	0,5	83,65%

Pada Tabel 4.4, performa dari model terlihat menurun seiring dengan diperkecilnya *learning rate*. Hingga percobaan ketiga, performa akurasi dari model turun sebanyak 14,36% dari arsitektur awal yaitu dengan *learning rate* 1e-1. Hasil ini menunjukkan bahwa model membutuhkan perubahan yang cukup besar atau *learning rate* yang lebih besar untuk mencapai konvergen.

Percobaan kemudian dilanjutkan dengan memperbesar *learning* rate menjadi 5e-1. Namun, performa model juga turun menjadi 83,65%. Hasil tersebut menjadi pertimbangan untuk kemudian menghentikan pengujian terhadap *learning rate.* Ketika *learning rate* diperkecil ataupun diperbesar dari kondisi awal yaitu 1e-1, performa model mengalami penurunan dalam melakukan klasifikasi. Oleh karena itu, pada pengujian ini diambil kesimpulan bahwa nilai terbaik untuk *learning rate* 1e-1 dengan rata-rata akurasi sebesar 96,05%. Hasil ini kemudian digunakan pada pengujian selanjutnya yaitu pengujian terhadap *batch size*.

4.2.4. Hasil tunning pada parameter batch size

Proses *tunning* parameter *batch size* bertujuan untuk mengetahui nilai *batch size* paling optimal bagi model untuk melakukan pembelajaran. Semakin banyak jumlah *batch* memiliki arti bahwa model akan mempelajari data lebih banyak dalam sekali proses pembelajaran. Begitu pula sebaliknya, jika *batch size* yang digunakan sedikit maka iterasi akan bertambah karena model mempelajari lebih sedikit data dalam satu iterasi.

Proses *tunning* menggunakan *batch size* 128 sebagai *initial value* dan mendapatkan akurasi sebesar 96,05% (Tabel 4.4). Selanjutnya, proses *tunning* dilanjutkan dengan menggunakan variasi *batch size* sebesar 96, 64, 32, 160, 192, 224, dan 256. Hasil dari percobaan disajikan pada Tabel 4.5.

Tabel 4.5 Komparasi rata-rata akurasi dari percobaan penggunaan variasi *batch size*.

No	Batch Size	Akurasi
1	128	96.05%
2	96	95,08%
3	64	93,15%
4	32	89,15%
5	160	**97,27%**
6	192	**97,21%**
7	224	**97,62%**
8	256	**97,13%**

Ketika *batch size* diperkecil menjadi 96, 64, dan 32, performa model terus mengalami penurunan. Rata-rata akurasi yang didapatkan secara berturut-turut adalah 95,08%, 93,15%, dan 89,15%. Berbeda halnya ketika *batch size* diperbesar dari 128. Rata-rata akurasi yang diberikan oleh model mengalami peningkatan dengan angka di atas 97%. Analisa kemudian dilanjutkan dengan melihat performa model dari sisi *recall*. Keputusan untuk melakukan analisa tersebut karena melihat perbedaan nilai akurasi yang tidak begitu signifikan antara *batch size* 160, 192, 224, dan 256.

Tabel 4.6 Komparasi *recall* pada pengujian *batch size*.

Batch Size **Label**	**160 (%)**	**192 (%)**	**224 (%)**	**256 (%)**
Hate Speech	94,29	97,06	99,05	97,63
Abusive	71,58	81,38	85,03	83,38
Individual	76,33	88,18	90,02	83,39
Group	70,82	83,14	86,16	79,52
Religion	58,14	78,04	79,02	80,56
Race	64,8	78,62	81,64	84,88
Physical	55,98	**65,17**	69,38	68,42
Gender	48,4	76,71	66,67	66,67
Other	85,44	92,47	94,75	93,15
Rata-rata	**69,53**	**82,31**	**83,52**	**81,96**

Pada Tabel 4.6, *batch size* 224 mendapat rata-rata nilai *recall* paling tinggi. Rata-rata *recall* yang didapat terpaut 1,21% dari *batch size* 192. Namun, jika melihat lebih detail lagi pada nilai di tiap label, *batch size* 192 hanya memiliki satu label yang bernilai di bawah 70%. Label tersebut adalah *physical* dengan nilai *recall* sebesar 65,17%. Penggunaan *batch size* 192 berhasil membawa *recall* dari label *gender* menembus angka di atas 75%. Label ini pada pengujian-pengujian sebelumnya memiliki nilai *recall* yang rendah dibandingkan dengan label-label lain. Oleh karena itu, pada pengujian terhadap *batch size* diputuskan untuk nilai terbaik dari *batch size* dalam pemodelan adalah sebesar 192.

4.2.5. Hasil percobaan menggunakan variasi pretrain model IndoBERT

Pre-train model dari IndoBERT berperan dalam melakukan tokenisasi dan *embedding* pada teks. Berbagai model telah dibangun dan ter-*publish* sehingga dapat digunakan kembali dalam penelitian ini. Beberapa *pre-train* dari IndoBERT kemudian diterapkan ke dalam pengujian untuk mendapatkan *pre-train* mana yang terbaik sebagai model untuk melakukan tokenisasi dari teks. Adapun model-model dari IndoBERT yang digunakan tersaji dalam Tabel 4.7.

Tabel 4.7 Komparasi performa *pre-train* IndoBert dalam tokenisasi teks.

No	Pre-train BERT	Akurasi
1	cahya/bert-base-indonesian-522M	96.05%
2	indobenchmark/indobert-base-p1	97,53%
3	indobenchmark/indobert-large-p2	**97,66%**
4	indolem/indobertweet-base-uncased	**97,84%**
5	ayameRushia/indobert-base-uncased-finetuned-indonlu-smsa	97,46%
6	afbudiman/indobert-classification	96,89%

Pada Tabel 4.7, terlihat bahwa terdapat 2 model *pre*-train IndoBERT yang membantu model dalam memberikan performa terbaiknya. Kedua *pre-train* tersebut adalah “indobenchmark/indobert-large-p2” dan “indolem/indobertweet-base-uncased”. Rata-rata akurasi kedua model tersebut hanya terpaut 0,18% yang diungguli oleh “indolem/indobertweet-base-uncased”. Hal ini menjadi menarik jika performa model di komparasi kembali dari metrik *recall* yang pada penelitian kali ini cukup sulit untuk ditingkatkan.

Pada Tabel 4.8, terlihat bahwa nilai *recall* untuk label *pyshical* dan *gender* mengalami perbaikan. Namun, “indobenchmark/indobert-large-p2” mengungguli model

“indolem/indobertweet-base-uncased” dari sisi rata-rata *recall* tiap label yaitu sebesar 85,25%. Selain itu, jika melihat lebih detail lagi ke nilai tiap label, “indolem/indobertweet-base-uncased” hanya mendapatkan satu label dengan *recall* di bawah 80%. Label tersebut adalah *gender* dengan nilai *recall* sebesar 72,15%. Oleh karena itu, pada pengujian terhadap *pre*-train IndoBERT diambil kesimpulan bahwa performa terbaik model didapatkan ketika menggunakan tokenisasi dari *pre-train* model “indobenchmark/indobert-large-p2”.

Tabel 4.8 Perbandingan recall dari 2 IndoBERT terbaik.

Labels	**indobenchmark/indobert-large-p2**	**indolem/indobertweet-base-uncased**
Hate Speech	98,13%	98,77%
Abusive	84,08%	85,40%
Individual	88,73%	90,60%
Group	84,83%	85,56%
Religion	81,96%	80,84%
Race	80,78%	82,72%
Physical	82,78%	74,64%
Gender	**72,15%**	73,97%
Other	93,77%	94,17%
Rata-rata	**85,25%**	85,19%

4.2.6. Percobaan tambahan

Peneliti memutuskan untuk melakukan percobaan tambahan dalam penelitian. Hal ini dilakukan untuk memastikan argumen-argumen yang didapatkan selama proses pengujian berlangsung. Misalnya ketika melakukan *tunning* terhadap jumlah *node* pada layer Bi-LSTM. Hasil percobaan mendapatkan bahwa model yang kompleks akan menurunkan performa model dalam mendeteksi ujaran kebencian. Sehingga perlu dilakukan validasi terhadap hasil tersebut yakni salah satunya dengan menambah layer Bi-LSTM. Begitu juga halnya dengan penerapan teknik *class weighting*, peneliti memutuskan untuk mengetahui bagaimana performa model jika tidak menerapkan teknik tersebut dalam model Bi-LSTM.

1. **Hasil percobaan terhadap model dengan 2 layer Bi-LSTM**

   Percobaan ini bertujuan untuk memperkuat argumen bahwa peningkatan kompleksitas dari model tidak lantas sejalan dengan performa model dalam melakukan klasifikasi

ujaran kebencian (Tabel 4.2). Adapun kompleksitas model yang diuji dalam percobaan ini adalah dengan menambah layer Bi-LSTM menjadi 2 layer.

Pada Tabel 4.9, performa model dengan 2 layer Bi-LSTM mengalami penurunan. Rata-rata akurasi yang didapatkan adalah sebesar 91,16%. Nilai tersebut lebih rendah 6,5% dari model dengan 1 layer Bi-LSTM. Oleh karena itu, percobaan ini membuktikan bahwa peningkatan kompleksitas pada model dapat menurunkan performa model dalam mengklasifikasi ujaran kebencian pada teks.

Tabel 4.9 Performa model dengan 2 buah layer Bi-LSTM.

Labels	**Model Terbaik**			**2 Layer Bi-LSTM**		
	Akurasi	**Presisi**	**Recall**	**Akurasi**	**Presisi**	**Recall**
Hate Speech	98,59%	98,52%	98,13%	96,00%	95,65%	94,83%
Abusive	93,12%	94,33%	84,08%	73,55%	78,38%	**26,81%**
Individual	96,18%	96,07%	88,73%	85,46%	85,42%	**52,11%**
Group	96,90%	95,97%	84,83%	87,65%	75,90%	**37,10%**
Religion	98,43%	95,44%	81,96%	93,27%	83,61%	**7,13%**
Race	98,99%	96,90%	80,78%	95,76%	85,45%	**10,15%**
Physical	99,56%	95,59%	82,78%	97,95%	70,00%	**3,35%**
Gender	99,35%	97,53%	72,15%	97,87%	80,00%	**3,65%**
Other	97,79%	98,14%	93,77%	92,95%	91,35%	82,22%
Rata-rata	97,66%	96,50%	85,25%	91,16%	82,86%	**35,26%**

2. Hasil percobaan terhadap pemodelan yang tidak menerapkan class weight

Percobaan lainnya dilakukan untuk menguji performa model tanpa *class weight.* Percobaan ini dilakukan untuk melihat besar pengaruh yang diberikan oleh penerapan *class weight* terhadap performa model dalam mengklasifiksi teks ke dalam label-label ujaran kebencian.

Pada Tabel 4.10, terlihat rata-rata akurasi dari model meningkat menjadi 98,06%. Namun, akurasi yang sangat baik tidak lantas diimbangi dengan performa *recall* dari model. Rata-rata *recall* yang didapatkan turun sebanyak 4,9% menjadi 80,35%. Turunnya performa model terlihat sangat singnifikan pada label *gender*. Label tersebut turun menjadi 47,95% dari yang sebelumnya berhasil mencapai 72,15% ketika model menerapkan *class weighting*. Oleh karena itu, keputusan untuk menerapkan *class weighting* pada model adalah keputusan yang tepat untuk menangani kasus *imbalance* pada dataset multilabel.

Tabel 4.10 Performa model tanpa menerapkan *class weighting*.

Labels	Model Terbaik			Model tanpa Class Weight		
	Akurasi	**Presisi**	**Recall**	**Akurasi**	**Presisi**	**Recall**
Hate Speech	98,59%	98,52%	98,13%	99,06%	98,77%	99,00%
Abusive	93,12%	94,33%	84,08%	96,29%	96,17%	92,39%
Individual	96,18%	96,07%	88,73%	97,28%	96,14%	93,14%
Group	96,90%	95,97%	84,83%	97,65%	96,83%	88,70%
Religion	98,43%	95,44%	81,96%	98,15%	94,46%	78,74%
Race	98,99%	96,90%	80,78%	98,75%	94,24%	77,75%
Physical	99,56%	95,59%	82,78%	98,88%	91,45%	**51,20%**
Gender	99,35%	97,53%	72,15%	98,78%	92,92%	**47,95%**
Other	97,79%	98,14%	93,77%	97,68%	97,27%	94,24%
Rata-rata	97,66%	96,50%	85,25%	98,06%	95,36%	**80,35%**

3. Hasil percobaan menggunakan arsitektur pemodelan menggunakan LSTM (tanpa biderectional)

Percobaan ini bertujuan untuk melihat apakah konsep penerapan *bidirectional* pada model berpengaruh secara signifikan terhadap performa model dalam mendeteksi ujaran kebencian pada teks berbahasa Indonesia. Hal ini dilakukan karena mengingat karakteristik dari Bahasa Indonesia yang bisa dikatakan sederhana. Bahasa Indonesia tidak ada ragam *tenses* dalam penggunaan ataupun penulisannya seperti Bahasa Inggris.

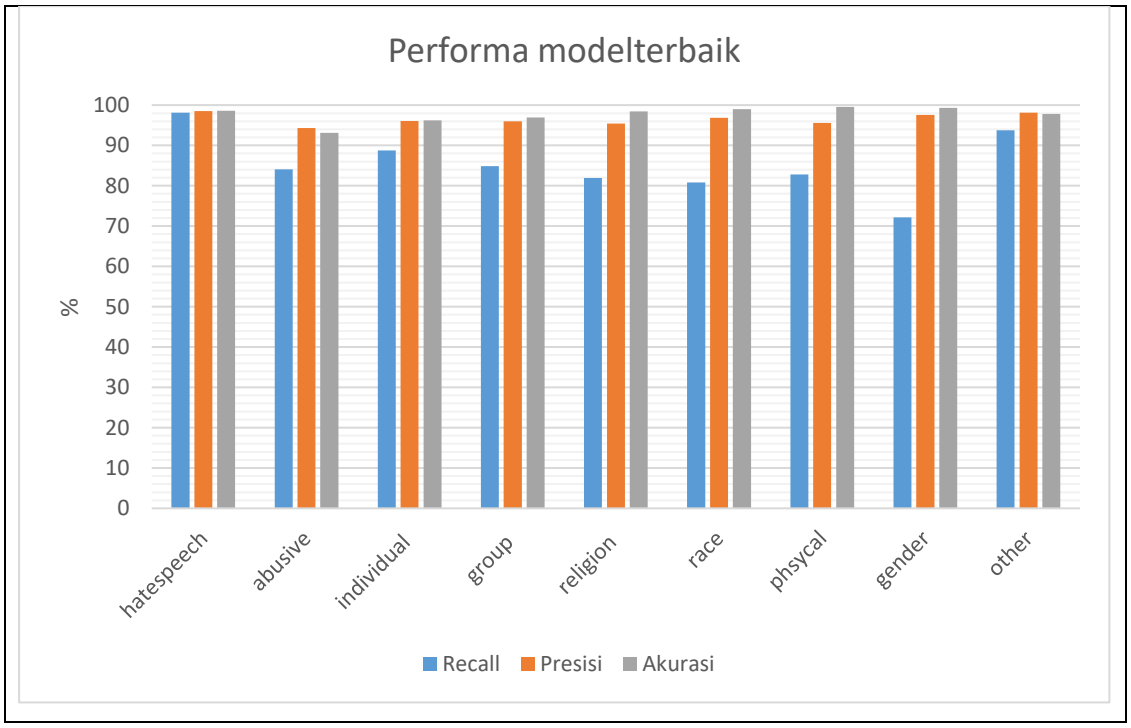
Tabel 4.11 Performa model menggunakan LSTM layer.

Labels	Model Terbaik			LSTM (tanpa Bidirectional)		
	Akurasi	**Presisi**	**Recall**	**Akurasi**	**Presisi**	**Recall**
Hate Speech	98,59%	98,52%	98,13%	85,75%	89,13%	75,37%
Abusive	93,12%	94,33%	84,08%	86,66%	88,12%	68,62%
Individual	96,18%	96,07%	88,73%	87,14%	86,67%	58,85%
Group	96,90%	95,97%	84,83%	91,62%	86,89%	58,07%
Religion	98,43%	95,44%	81,96%	95,81%	89,15%	47,13%
Race	98,99%	96,90%	80,78%	97,35%	88,37%	49,24%
Physical	99,56%	95,59%	82,78%	98,29%	93,18%	**19,62%**
Gender	99,35%	97,53%	72,15%	98,32%	80,00%	**31,05%**
Other	97,79%	98,14%	93,77%	87,65%	88,76%	63,24%
Rat-rata	97,66%	96,50%	85,25%	92,07%	87,81%	**52,35%**

Pada Tabel 4.11, penerapan *bidirectional* terlihat sangat berpengaruh terhadap performa model dalam mengklasifikasi teks Bahasa Indonesia ke dalam label-label

ujaran kebencian. Model LSTM (tanpa *bidirectional*) memberikan performa yang lebih rendah dibandingkan dengan model arsitektur Bi-LSTM. Rata-rata akurasi yang diperoleh turun sebesar 5,59% menjadi 92,07%. Penurunan performa juga terjadi secara signifikan pada matriks *recall*. Terdapat 4 label yang memiliki nilai di bawah 50% bahkan ada yang mencapai 19% yaitu label *pyshical*.

Turunnya performa dari model LSTM ini tidak lepas dari kondisi data yang *imbalance*. Jika kita melihat kembali histogram pada Gambar 4.1, keempat label dengan nilai *recall* terendah (*gender*, *physical*, *race*, *religion*) merupakan label-label dengan kondisi imbalance yang ekstrim. Pada label *pyshical*, hanya terdapat 2,09% teks dengan nilai positif dari total 10.011 teks yang tersedia. Kondisi tersebut kemungkinan menjadi aspek yang membuat performa model mengalami penurunan. Sehingga, pada penelitian ini model terbaik untuk melakukan deteksi ujaran kebencian *multilabel* pada teks adalah menggunakan Bi-LSTM. Adapun mengenai parameter pemodelannya yaitu menggunakan 20 *epoch*, *learning rate* 1e-1, 192 *batch size*, 1 layer Bi-LSTM dengan 40 node, dan terakhir menggunakan tokenisasi teks menggunakan *pre-train* model dari "indobenchmark/indobert-large-p2". Performa dari model tersebut tersaji dalam Gambar 4.10.



Gambar 4.10. Histogram performa model terbaik

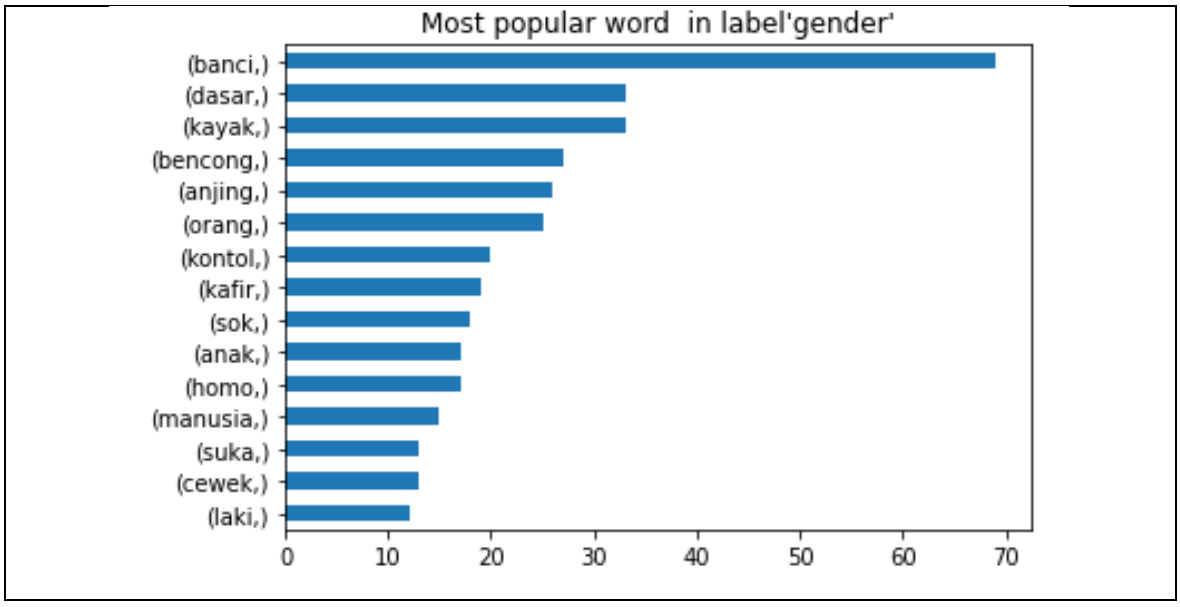
4. Hasil percobaan terhadap variasi ukuran token IndobBERT

Analisa lebih lanjut kemudian dilakukan terhadap performa model terbaik untuk mendeteksi ujaran kebencian *multilabel* pada teks berbahasa Indonesia. Gambar 4.10 menunjukkan bahwa terdapat satu nilai yang memiliki nilai *recall* di bawah 80%. Nilai tersebut adalah *recall* untuk label *gender*. Beberapa contoh teks yang gagal diklasifikasi sebagai label *gender* disajikan pada Tabel 4.12.

Tabel 4.12 Contoh teks berlabel *gender* yang tidak berhasil diklasifikasi dengan benar.

No	Teks Asli	Hasil Preprocessing	Hasil Prediksi
1	USER Tangkap aja itu jendral banci kayak *** yg bodoh	tangkap jenderal banci kayak *** bodoh	Label: *Hate speech, Abusive, Individual, Gender* Prediksi: *Non-Hate Speech*
2	Jadi cowo itu harus Gantle kalo ga Gantle itu namanya BANCI	cowok gantle gantle namanya banci	Label: *Hate speech, abusive, Gender* Prediksi: *Non-Hate Speech*

Pada Tabel 4.12, terlihat bahwa kedua teks mengandung kata “banci” yang identik dengan sebuah kata untuk menyerang *gender*. Kata tersebut merupakan kata yang paling sering muncul pada teks yang positif label “gender” (Gambar 4.11). Kata “banci” muncul sebanyak hampir 70 kali dalam dataset diikuti oleh kata “dasar”. Namun, kedua teks tersebut gagal diklasifikasikan dengan benar oleh model. Dari Gambar 4.11 juga bisa dievaluasi proses *pre-processing* karena terdapat kata bencong yang muncul dalam 4 besar kata paling sering muncul pada label positif *gender*. Kata “bencong” pada dasarnya memilki makna yang sama dengan “banci” sehingga perlu ditambahkan dalam kamus normalisasi.



Gambar 4.11 Kata yang paling muncul dalam ujaran kebencian "gender".

Jika dilihat lagi pada hasil *pre-processing* (Tabel 4.12), teks nomor 1 dan 3 memiliki jumlah token yang kurang dari 10. Jadi, kekurangan token dari kedua teks tersebut harus diisi dengan token [PAD]. Hal ini terjadi karena panjang token yang didefinisikan pada model BERT adalah 10. Keberadaan token [PAD] pada hasil tokenisasi kemungkinan yang menyebabkan model gagal dalam melakukan klasifikasi teks ujaran kebencian. Untuk itu, penulis memutuskan untuk melanjutkan pengujian terkait dengan panjang token dalam proses tokenisasi. Percobaan mencoba untuk melakukan penambahan dan pengurangan panjang token menjadi 5, 15, dan 20. Adapun hasil dari pengujian tersaji dalam Tabel 4.13.

Tabel 4.13 Perbandingan performa akurasi model pada percobaan variasi panjang token.

Label	5	10	15	20
Hate Speech	96,49%	98,59%	98,34%	98,74%
Abusive	90,41%	93,12%	94,74%	95,53%
Individual	93,60%	96,18%	96,21%	96,96%
Group	95,18%	96,90%	97,38%	96,91%
Religion	97,96%	98,43%	98,65%	98,77%
Race	98,64%	98,99%	99,31%	99,10%
Physical	99,33%	99,56%	99,73%	99,37%
Gender	99,22%	99,35%	99,51%	99,33%
Other	96,35%	97,79%	97,88%	97,58%
Rata-rata	96,35%	97,66%	97,97%	98,03%

Pada Tabel 4.13, terlihat bahwa rata-rata akurasi yang didapatkan oleh model tidak mengalami perubahan yang cukup signifikan ketika panjang token diubah. Penurunan performa didapatkan ketika panjang token diperkecil menjadi 5. Rata-rata akurasi yang didapat model ketika menggunakan 5 token adalah sebesar 96,35%. Penurunan terjadi sebanyak 1,31% dari performa awal model ketika menggunakan 10 token. Sebaliknya, model mengalami peningkatan performa ketika token di perpanjang menjadi 15 dan 20. Ketika menggunakan 15 token, performa model meningkat sebanyak 0,31% menjadi 97,97%. Peningkatan performa kembali terjadi ketika token semakin diperpanjang menjadi 20. Namun, peningkatan yang terjadi tidak begitu banyak yakni hanya 0,06% saja dari performa model ketika menggunakan 15 token.

Analisa lebih lanjut kemudian dilakukan terhadap performa *recall* dari model. Hal ini mengingat perbedaan performa akurasi yang tidak begitu signifikan antara penggunaan token dengan panjang 10, 15, dan 20. Komparasi dari performa *recall* tiap label tersaji dalam Tabel 4.14.

Pada Tabel 4.14, terlihat bahwa performa model model mampu meningkatkan performa *recall* dengan rata-rata 88,74% ketika menggunakan token dengan panjang 15. Perbaikan performa juga terlihat dari nilai *recall* tiap label. *Recall* dari label "gender" meningkat dengan berhasil menembus angka 80% atau lebih tepatnya 80,82%. Hasil tersebut menunjukkan bahwa model dengan token sepanjang 15 memberikan informasi yang lebih lengkap dibandingkan dengan token dengan panjang 10. Kelengkapan informasi tersebut membuat model menjadi lebih maksimal dalam melakukan klasifikasi. Sehingga performa yang dihasilkan ketika menggunakan 15 token menjadi lebih baik.

Tabel 4.14. Komparasi *recall* pada variasi panjang token.

Label	10	15	20
Hate Speech	98,13%	97,39%	98,65%
Abusive	84,08%	88,50%	90,69%
Individual	88,73%	88,65%	92,55%
Group	84,83%	88,34%	85,38%
Religion	81,96%	85,73%	84,76%
Race	80,78%	87,26%	82,29%
Physical	82,78%	88,04%	74,64%
Gender	72,15%	80,82%	72,60%
Other	93,77%	93,95%	93,77%
Rata-rata	85,25%	88,74%	86,15%

Akan tetapi, ketika token kembali diperpanjang menjadi 20, performa *recall* menurun dengan rata-rata sebesar 86,15%. Penurunan yang cukup signifikan juga terjadi pada label label “pyshical” dan “gender”. Nilai *recall* dari kedua label tersebut turun menjadi 74,64% dan 72,60%. Padahal, sejak awal percobaan kedua label tersebut mengalami kesulitan untuk meningkatkan nilai *recall*.

Penentuan panjang token pada proses tokenisasi menjadi tantangan tersendiri pada penelitian ini. Sepanjang percobaan yang dilakukan, model terbaik didapatkan dengan menggunakan token sepanjang 15. Namun, performa model dari sisi *recall* masih tertinggal dari 2 metrik lainnya yaitu akurasi dan presisi. Rata-rata nilai *recall* yang didapatkan oleh model adalah 88,74%. Sedangkan rata-rata akurasi diperoleh adalah 97,97%. Rendahnya performa *recall* tersebut kemungkinan terjadi karena adanya informasi yang hilang terkait dengan pemotongan token ketika proses tokenisasi.

Sebagai contoh, teks “*Provokasi mayat, politisasi agama" penyebab kekalahan pilkada DKI, beginilah cara cebong mendeskripsikan kekalahan junjungannya. ; FYI ahog blm pernah ikut pemilihan apapun (kec jd wakil) dan dia bukanlah etnis mayorita*”. Teks tersebut akan menjadi “*'provokasi mayat politisasi agama penyebab kekalahan pilihan kepala daerah daerah khusus ibukota cebong mendeskripsikan kekalahan junjungannya for your information ahok pemilihan apapun kecamatan wakil etnis mayoritas*” setelah mengalami *pre-processing* dengan panjang kata sebanyak 26. Jumlah kata yang banyak terjadi karena teks dibangun oleh lebih dari satu kalimat. Oleh karena itu, diputuskan untuk melakukan percobaan sekali lagi untuk mereanotasi dataset. Reanotasi dilakukan terhadap teks yang memiliki 2 kalimat atau lebih.

5. Hasil percobaan terhadap model mengguanakan dataset yang telah di-reanotasi

Proses reanotasi dataset dimulai dengan memfilter teks-teks yang memiliki panjang kalimat lebih dari satu. Proses filterisasi tersebut mendapatkan sebanyak 1.039 teks akan direanotasi. Tiap kalimat dalam teks tersebut akan membentuk satu data baru. Tercatat ada sebanyak 2.341 data yang siap dianotasi ulang untuk menjadi data tambahan untuk dataset.

Proses anotasi dilakukan oleh 3 orang relawan yang memiliki latar belakang pendidikan informatika. Kesimpulan nilai label dari tiap data diambil apabila minimal 2 annotator yang memberikan nilai yang sama baik nilai positif ataupun negatif. Data yang dihasilkan dari anotasi ini kemudian digabungkan dengan dataset induk. Dengan demikian total data pada dataset yang baru bertambah menjadi 14.468 data.

Selanjutnya, percobaan terhadap data baru dilakukan mulai dari *pre-processing*. *Pre-processing* menerapkan perlakuan yang sama terhadap dataset yang baru. Begitu juga dengan model IndoBERT yang digunakan beserta arsitektur dari model Bi-LSTM.. percobaan ini menggunakan arsitektur terbaik yang sudah diperoleh pada penelitian sebelumnya yakni 20 *epoch*, *learning rate* 1e-1, 192 *batch size*, 1 layer Bi-LSTM dengan 40 node, dan model tokenisasi dari "indobenchmark/indobert-large-p2".

Pada proses pengujian, diterapkan beberapa variasi panjang token hasil tokenisasi. Pengujian tersebut dilakukan karena mengingat reanotasi terjadi karena adanya masalah hilangnya data pada hasil token karena teks yang panjang. Adapun variasi panjang token yang diterapkan dalam percobaan ini antara lain 5, 10, 15, 20, dan 25 dengan hasil tersaji pada Tabel 4.15.

Tabel 4.15. Komparasi rata-rata akurasi model menggunakan dataset hasil reanotasi.

Label	10	15	20
Hate Speech	96,88%	98,81%	98,18%
Abusive	94,22%	95,78%	95,87%
Individual	95,59%	97,21%	96,70%
Group	97,04%	98,10%	97,40%
Religion	98,96%	98,94%	98,93%
Race	99,38%	99,50%	99,57%
Physical	99,60%	99,56%	99,67%
Gender	99,63%	99,56%	99,57%
Other	98,15%	98,28%	98,32%
Rata-rata	**97,72**%	**98,42**%	**98,25**%

Pada Tabel 4.15, terlihat bahwa rata-rata akurasi tiap percobaan mengalami peningkatan setelah dataset di reanotasi. Namun, peningkatan performa yang terjadi tidak begitu signifikan. Sebagai contoh, ketika menggunakan 10 token, performa model hanya meningkat 0,06% dari yang sebelumnya sebesar 97,66% menjadi 97,72%. Begitu juga dengan model yang menggunakan 15 token. Performa model meningkat sebesar 0,45% yakni menjadi 98,42% setelah reanotasi dataset dilakukan.

Performa model ketika menggunakan dataset setelah reanotasi terlihat tidak memiliki perbedaan yang signifikan ketika diuji menggunakan variasi panjang token. Akurasi terkecil diperoleh ketika percobaan dilakuakan terhadap token dengan panjang 10. Sedangkan akurasi terbaiknya diperoleh ketika melakukan percobaan token sepanjang 15 token. Akan tetapi, margin antara kedua perocbaan tersebut tidaklah jauh

yakni sebesar 0,7%. Untuk itu, dilakukan analisa lebih lanjut dari sisi *recall* untuk melihat bagaimana performa model dalam mendeteksi dengan benar teks yang positif ujaran kebencian. Perbandingan performa *recall* terhadap variasi panjang token tersaji pada Tabel 4.16.

Tabel 4.16 perbandingan *recall* dari variasi panjang token dengan dataset setelah reanotasi.

Label	10	15	20
Hate Speech	95,87%	98,43%	98,14%
Abusive	87,44%	90,62%	91,27%
Individual	86,23%	91,65%	90,48%
Group	84,13%	90,57%	87,19%
Religion	86,28%	87,55%	85,56%
Race	88,64%	92,17%	92,68%
Physical	85,16%	82,42%	85,71%
Gender	85,16%	82,42%	81,32%
Other	94,68%	95,11%	95,69%
Rata-rata	**88,18%**	**90,10%**	**89,78%**

Pada Tabel 4.16, terlihat bahwa performa model mengalami perbaikan dari sisi *recall* setelah reanotasi dilakukan. Pada percobaan dengan token sepanjang 10, rata-rata *recall* yang didapatkan meningkat sebanyak 2,93% menjadi 88,18%. Reanotasi pada dataset juga berhasil meningkatkan nilai *recall* pada label "gender". *Recall* pada label tersebut berhasil menembus angka 85,16% dari yang sebelumnya bernilai 72,15%.

Jika melihat lagi hasil yang didapatkan oleh tiap percobaan pada Tabel 4.15, model berhasil memberikan performa *recall* di atas 80% pada tiap label dengan rata-rata terbaik diperoleh pada percobaan menggunakan token dengan panjang 15. Rata-rata *recall* yang diberikan oleh model ketika menggunakan token dengan panjang 15 adalah sebesar 90,10%. Jika dibandingkan dengan rata-rata *recall* dari model sebelum reanotasi dataset, percobaan berhasil meningkatkan performa dengan perbedaan 1,36% dari penggunaan dataset sebelum di reanotasi

4.2.7. Diskusi

Model Bi-LSTM yang dihasilkan dalam penelitian memberikan akurasi sebesar 96,77% untuk mendeteksi ujaran kebencian dengan *output multilabel*. Model berhasil memberikan performa yang lebih tinggi dibandingkan penelitian [2] yang sama-sama memberikan *output*

multilabel. Akurasi pada penelitian ini berhasil unggul sebesar 9,84% dari penelitian [2]. Penelitian ini juga berhasil mendapatkan akurasi yang lebih tinggi dari penelitian yang hanya memberikan *output single* label [9]. Dengan menggunakan dataset yang sama, penelitian ini berhasil mendapatkan akurasi 1,22% lebih tinggi dibanding penelitian [9].

Penelitian mendapatkan performa terbaiknya dengan menggunakan model Bi-LSTM yang dibangun menggunakan 20 *epoch*, 10 *node* layer Bi-LSTM, 1e-1 *learning rate*, dan 192 *batch size.* Kemampuan model untuk mendeteksi ujaran kebencian pada teks juga didukung oleh proses tokenisasi pada teks yang dihasilkan oleh *pre-train* model dari "indobenchmark/indobert-large-p2".

Dalam proses mendapatkan model terbaiknya, hal yang cukup menjadi hambatan adalah masalah data yang *imbalance* yang kerap terjadi pada dataset *multilabel* [19]. *Imbalance* dataset berakibat pada sulitnya model dalam memberikan performa yang baik dari sisi *recall*. Untuk menangani masalah tersebut, penelitian ini menerapkan teknik *class weighting* pada saat pemodelan. Penerapan teknik tersebut terbilang berhasil dalam kasus ini karena ketika pemodelan tidak menerapkan *class weight*, *recall* yang diberikan menurun sebesar 4,9%. Perbedaan yang sangat signifikan bisa dilihat dari penurunan label yang sangat *imbalance* pada label *gender* yang mendapatkan *recall* di bawah 50% (Tabel 4.10).

Selain melakukan percobaan sesuai dengan yang direncanakan pada metodologi penelitian (poin 3.5), penelitian juga melakukan reanotasi terhadap teks-teks yang memiliki kalimat lebih dari satu. Hal tersebut melihat adanya kasus pemotongan kata pada teks yang memiliki token melebihi batas yang ditentukan yaitu 10. Hasilnya, nilai *recall* yang diperoleh oleh model mengalami peningkatan sebesar 2,93%. Peningkatan terbesar dari nilai *recall* didapatkan ketika batas jumlah token diperbesar menjadi 15. Jumlah peningkatan yang diperoleh oleh model adalah sebesar 4,85%.

Performa 97,66% yang didapatkan pada model dengan 10 token ternyata masih lebih rendah dibandingkan dengan penelitian serupa yang pernah dilakukan dengan menggunakan Convolution Neural Network (CNN) [30]. Deteksi ujaran dengan *output multilabel* menggunakan CNN berhasil mendapatkan akurasi 98,07%.

Percobaan dengan CNN memiliki proses *pre-processing* yang berbeda dengan Bi-LSTM. Terlihat pada Gambar 4.12, kedua percobaan menerapkan teknik *cleaning* yang berbeda sehingga menghasilkan *wordcloud* yang berbeda juga. *Pre-processing* pada percobaan menggunakan Bi-LSTM terlihat menghasilkan data yang lebih *clean* dibandingkan dengan percobaan CNN. Pada percobaan dengan CNN, masih terlihat beberapa kata-kata yang bisa tergolong sebagai *stopwords* seperti "nya", "nih", dan "wkwk".

Selain itu masih terdapat kata “uniform”, “resource”, “locator” yang merupakan kepanjangan dari akronim URL.

(a) Percobaan dengan CNN

(b) Percobaan dengan Bi-LSTM

Gambar 4.12. *Wordcloud* hasil *pre-processing* pada dataset.

Penelitian dengan model Bi-LSTM dan CNN juga memiliki perbedaan pada nilai parameter *epoch* dan *learning rate*. Percobaan kemudian dilakukan dengan merubah parameter pemodelan pada Bi-LSTM mengikuti parameter yang digunakan pada CNN yaitu 200 *epoch* dan *learning rate* sebesar 1e-3. Selain itu, dataset hasil *preprocessing* pada penelitian Bi-LSTM juga dicoba untuk digunakan pada arsitektur pemodelan pada CNN.

Tabel 4.17. Perbandingan percobaan pemodelan dengan Bi-LSTM dan CNN.

Labels	**Model Bi-LSTM**			**Model CNN**		
	Akurasi	**Presisi**	**Recall**	**Akurasi**	**Presisi**	**Recall**
Hate Speech	98,73%	99,04%	97,94%	99,89%	99,95%	99,79%
Abusive	89,68%	94,68%	72,64%	94,91%	89,17%	95,01%
Individual	95,88%	97,44%	86,19%	99,69%	99,61%	99,18%
Group	96,89%	97,46%	83,38%	99,85%	99,46%	99,64%
Religion	98,09%	98,34%	74,55%	99,93%	99,16%	99,86%
Race	99,06%	99,46%	80,13%	100%	100%	100%
Physical	99,62%	99,42%	82,30%	99,98%	99,04%	100%
Gender	99,73%	99,48%	88,13%	100%	100%	100%
Other	98,47%	98,69%	95,73%	99,87%	99,93%	99,6%
Rat-rata	97,35%	98,22%	84,55%	99,35%	98,48%	99,23%

Pada

Tabel 4.17, performa model CNN terbilang sangat baik dibandingkan dengan model Bi-LSTM dalam melakukan klasifikasi ujaran kebencian dengan *output multilabel*. Recall dari label *gender* dan *physical* berhasil meningkat menjadi 100% pada pemodelan dengan CNN. Sementara itu, ketika model Bi-LSTM dilakukan dengan menggunakan 200 *epoch* dan *learning rate* 1e-3, performa model terlihat menurun. Nilai *recall* dari model Bi-LSTM tampak menurun dari yang sebelumnya memiliki rata-rata 85,25% menjadi 84,55%.

Dari Analisa yang sudah dilakukan terhadap kedua model CNN dan Bi-LSTM didapatkan bahwa model CNN ternyata lebih *powerful* dalam melakukan klasifikasi ujaran kebencian berbahasa Indonesia. Meskipun Bi-LSTM secara teori lebih diperuntukkan untuk melakukan pekerjaan terkait dengan analisis teks. Proses pada *pre-processing* juga berpengaruh pada performa model. Hal ini dibuktikan dengan berhasilnya model CNN untuk meningkatkan performa ketika di-*training* menggunakan dataset pada penelitian ini.

BAB 5
Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan percobaan-percobaan pada penelitian, didapatkan kesimpulan antara lain:

1. Model Bi-LSTM dalam penelitian ini berhasil mendapatkan akurasi yang sangat baik dalam mendeteksi ujaran kebencian beserta aspek yang disinggungnya yaitu sebesar 97,66%.
2. Akurasi yang dihasilkan oleh model Bi-LSTM lebih tinggi dibandingkan dengan model Bi-GRU yang mendeteksi ujaran kebencian dan aspek yang disinggung. Model Bi-LSTM dalam penelitian ini juga mampu memberikan akurasi yang lebih tinggi dari model LSTM yang mendeteksi hanya ujaran kebencian dengan dataset yang sama.
3. Model terbaik dalam penelitian didapatkan dengan menggunakan 20 *epoch*, *batch size* 192, *learning rate* 1e-1, 1 layer Bi-LSTM dengan 40 *node*, serta *class weighting* pada proses optimasi.
4. Performa model Bi-LSTM didukung oleh proses tokenisasi pada teks dengan menggunakan model *pre-train* IndoBERT dari “indobenchmark/indobert-large-p2”.
5. Reanotasi terhadap teks yang memiliki kalimat lebih dari satu berhasil meningkatkan performa *recall* model Bi-LSTM dari 85,25% menjadi 88,188%.

5.2. Saran

Pada penelitian ini masih terdapat beberapa hal yang perlu diperhatikan. Pertama adalah pada bagian *pre-processing*. Jika penelitian ini dilanjutkan dikemudian hari, penulis menyarankan untuk melakukan eksplorasi lebih dalam lagi terkait dengan ketika melakukan *cleaning* dan normalisasi. Selain itu, bisa dicoba untuk tidak menghilangkan *stopword* pada proses *pre-processing* sebagai bahan perbandingan untuk melihat performa model.

Selanjutnya adalah masalah ukuran teks sebagai *input* model. Anotasi yang dilakukan pada penelitian ini bukanlah penelitian utama. Untuk itu, disarankan pada penelitian selanjutnya untuk melakukan reanotasi dataset dengan proses yang lebih terstruktur lagi. Selain itu, penelitian selanjutnya juga bisa dilakukan untuk melakukan pemodelan *multilabel* tanpa melakukan transformasi data seperti menggunakan *ensemble learning.*

Daftar Pustaka

[1] S. Kemp, "Digital in Indonesia: All the Statistics You Need in 2021 — DataReportal – Global Digital Insights," 2021. https://datareportal.com/reports/digital-2021-indonesia (accessed Feb. 24, 2022).

[2] A. Marpaung, R. Rismala, and H. Nurrahmi, "Hate Speech Detection in Indonesian Twitter Texts using Bidirectional Gated Recurrent Unit," in *KST 2021 - 2021 13th International Conference Knowledge and Smart Technology*, Jan. 2021, pp. 186–190. doi: 10.1109/KST51265.2021.9415760.

[3] C. Febriyani, "The Danger of Hate Speech in Cyberspace is Regulated as a Crime in UU ITE (Bahaya Ujaran Kebencian di Dunia Maya Diatur Sebagai Tindak Pidana di UU ITE)," 2021. https://www.industry.co.id/read/93219/bahaya-ujaran-kebencian-di-dunia-maya-diatur-sebagai-tindak-pidana-di-uu-ite (accessed Feb. 24, 2022).

[4] G. B. Herwanto, A. M. Ningtyas, K. E. Nugraha, and I. N. P. Trisna, "Hate Speech and Abusive Language Classification using fastText," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 69–72. doi: 10.1109/ISRITI48646.2019.9034560.

[5] D. Putri, "Should all hate speech be punished? Notes for revision of UU ITE (Apakah semua ujaran kebencian perlu dipidana? Catatan untuk revisi UU ITE)," 2021. https://theconversation.com/apakah-semua-ujaran-kebencian-perlu-dipidana-catatan-untuk-revisi-uu-ite-156132 (accessed Feb. 24, 2022).

[6] A. P. J. Dwitama, "Hate Speech Detection on Indonesian Twitter using Machine Learning: Review Literature (Deteksi Ujaran Kebencian Pada Twitter Bahasa Indonesia Menggunakan Machine Learning: Reviu Literatur)," *Jurnal SNATi*, vol. 1, pp. 31–39, 2021.

[7] B. Satrio, "To Cyber Police, Fritz Explains Challenges of Detecting Hate Speech and Hoaxes (Kepada Polisi Siber, Fritz Jabarkan Tantangan Deteksi Ujaran Kebencian dan Hoaks)," *BAWASLU*, 2020. https://bawaslu.go.id/en/berita/kepada-polisi-siber-fritz-jabarkan-tantangan-deteksi-ujaran-kebencian-dan-hoaks (accessed Apr. 10, 2022).

[8] E. Sazany and I. Budi, "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)*, Sep. 2018, pp. 114–117. doi: 10.1109/ICAITI.2018.8686725.

[9] A. R. Isnain, A. Sihabuddin, and Y. Suyanto, "Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 2, p. 169, Apr. 2020, doi: 10.22146/ijccs.51743.

[10] M. O. Ibrohim and I. Budi, "Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter," in *Proceedings of the Third Workshop on Abusive Language Online*, 2019, pp. 46–57. [Online]. Available: https://www.komnasham.go.id/index.php/

[11] V. Kotu and B. Deshpande, "Data Science Process," *Data Science*, pp. 19–37, Jan. 2019, doi: 10.1016/B978-0-12-814761-0.00002-2.

[12] K. Gligori´c, G. Epfl, A. Anderson, and R. West, "How Constraints Affect Content: The Case of Twitter's Switch from 140 to 280 Characters *," in *Proceedings of the Twelfth International AAAI Conference on Web and Social Media (ICWSM 2018)*, 2018. [Online]. Available: https://arxiv.org/abs/1804.02318

[13] P. Malik, A. Aggrawal, and D. K. Vishwakarma, "Toxic Speech Detection using Traditional Machine Learning Models and BERT and fastText Embedding with Deep Neural Networks," in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, Apr. 2021, pp. 1254–1259. doi: 10.1109/ICCMC51019.2021.9418395.

[14] V. Indurthi, B. Syed, M. Shrivastava, N. Chakravartula, M. Gupta, and V. Varma, "Fermi at SemEval-2019 Task 5: Using Sentence Embeddings to identify Hate Speech against Immigrants and Women on Twitter," in *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, 2019, pp. 70–74. [Online]. Available: https://sites.google.com/site/alw2018

[15] S. Agarwal and C. R. Chowdary, "Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19," *Expert Syst Appl*, vol. 185, Dec. 2021, doi: 10.1016/j.eswa.2021.115632.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.04805

[17] P. Bahad, P. Saxena, and R. Kamal, "Fake News Detection using Bi-directional LSTM-Recurrent Neural Network," in *Procedia Computer Science*, 2019, vol. 165, pp. 74–82. doi: 10.1016/j.procs.2020.01.072.

[18] T. Chauhan and H. Palivela, “Optimization and improvement of fake news detection using deep learning approaches for societal benefit,” *International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100051, Nov. 2021, doi: 10.1016/j.jjimei.2021.100051.

[19] F. Herrera, F. Charte, A. J. Rivera, and M. J. del Jesus, “Multilabel Classification,” in *Multilabel Classification*, Springer International Publishing, 2016, pp. 17–31. doi: 10.1007/978-3-319-41111-8_2.

[20] M. Pushpa and S. Karpagavalli, “Multi-label Classification: Problem Transformation methods in Tamil Phoneme classification,” in *Procedia Computer Science*, 2017, vol. 115, pp. 572–579. doi: 10.1016/j.procs.2017.09.116.

[21] “Difference Between a Batch and an Epoch in a Neural Network - MachineLearningMastery.com.” https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/ (accessed Jan. 25, 2023).

[22] M. O. Ibrohim, M. A. Setiadi, and I. Budi, “Identification of hate speech and abusive language on Indonesian twitter using theword2vec, part of speech and emoji features,” in *Advanced Information Science and System*, Nov. 2019. doi: 10.1145/3373477.3373495.

[23] F. A. Prabowo, M. O. Ibrohim, I. Budi, and Institute of Electrical and Electronics Engineers, “Hierarchical Multi-label Classification to Identify Hate Speech and Abusive Language on Indonesian Twitter,” in *2019 6th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, 2019. doi: 10.1109/ICITACEE.2019.8904425.

[24] H. Mohaouchane, A. Mourhir, and N. S. Nikolov, “Detecting Offensive Language on Arabic Social Media Using Deep Learning,” in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Oct. 2019, pp. 466–471. doi: 10.1109/SNAMS.2019.8931839.

[25] A. S. Saksesi, M. Nasrun, and C. Setianingsih, “Analysis Text of Hate Speech Detection Using Recurrent Neural Network,” in *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 2018, pp. 242–248. doi: 10.1109/ICCEREC.2018.8712104.

[26] P. Wu, X. Li, C. Ling, S. Ding, and S. Shen, “Sentiment classification using attention mechanism and bidirectional long short-term memory network,” *Appl Soft Comput*, vol. 112, Nov. 2021, doi: 10.1016/j.asoc.2021.107792.

[27] “Models - Hugging Face.” https://huggingface.co/models?search=indobert (accessed Apr. 08, 2022).

[28] M. Zhu *et al.*, “Class weights random forest algorithm for processing class imbalanced medical data,” *IEEE Access*, vol. 6, pp. 4641–4652, Jan. 2018, doi: 10.1109/ACCESS.2018.2789428.

[29] K. Singh, “How To Dealing With Imbalanced Classes in Machine Learning.” https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/ (accessed Mar. 02, 2023).

[30] A. P. J. Dwitama and S. Hidayat, “Identifikasi Ujaran Kebencian Multilabel Pada Teks Twitter Berbahasa Indonesia Menggunakan Convolution Neural Network,” *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 3, no. 2, p. 117, Dec. 2021, doi: 10.30865/json.v3i2.3610.