

**IDENTIFIKASI *CYBERBULLYING* PADA MEDIA SOSIAL
TWITTER MENGGUNAKAN LSTM DAN
BILSTM**



Disusun Oleh:

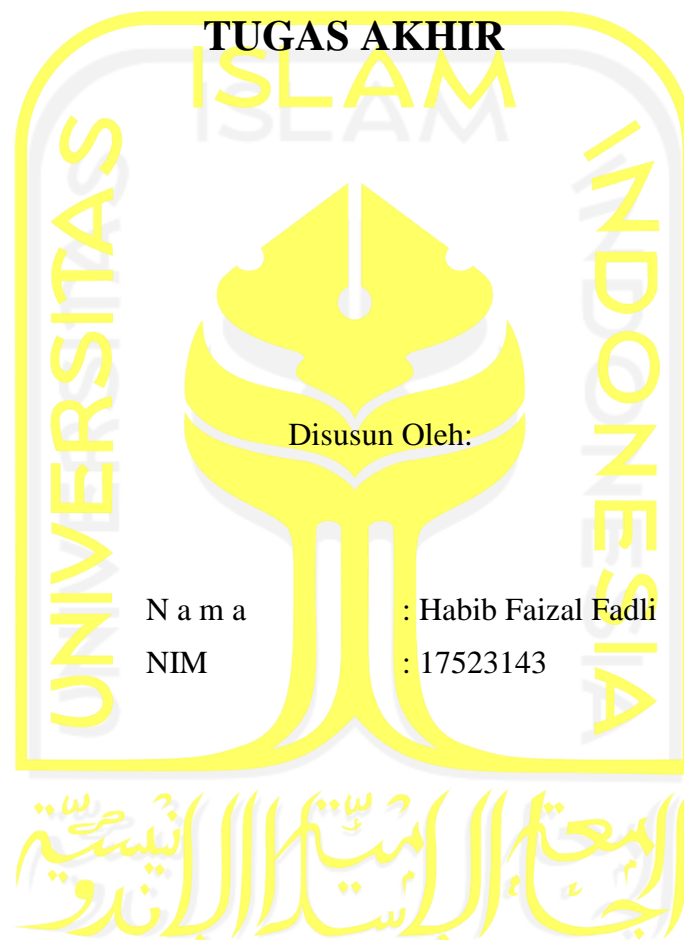
N a m a : Habib Faizal Fadli
NIM : 17523143

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2020

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**IDENTIFIKASI *CYBERBULLYING* PADA MEDIA SOSIAL
TWITTER MENGGUNAKAN LSTM DAN
BILSTM**



Yogyakarta, 11 Januari 2021

Pembimbing,

(Ahmad Fathan Hidayatullah, S.T., M.Cs.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**IDENTIFIKASI *CYBERBULLYING* PADA MEDIA SOSIAL
TWITTER MENGGUNAKAN LSTM DAN
 BILSTM**

TUGAS AKHIR

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

Yogyakarta, 11 Januari 2021

Tim Penguji

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

Anggota 1

Yudi Prayudi, S.Si., M.Kom.

Anggota 2

Fayruz Rahma, S.T., M.Eng.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



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

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Habib Faizal Fadli

NIM : 17523143

Tugas akhir dengan judul:

**IDENTIFIKASI *CYBERBULLYING* PADA MEDIA SÖSIAL
TWITTER MENGGUNAKAN LSTM DAN
BILSTM**

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

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

Yogyakarta, 11 Januari 2021



(Habib Faizal Fadli)

HALAMAN PERSEMBAHAN

Alhamdulillah robbil'alamin, puja dan puji syukur ke hadirat Allah *Subhanahu wata'ala* atas segala nikmat yang diberikan kepada kita semua. Shalawat serta salam kepada junjungan kita Nabi Muhammad *Sallallahu 'alaihi wasallam* yang telah membimbing kita menuju Islam yang rahmatan lil 'alamin, dan yang kita nantikan syafa'atnya kelak di hari akhir.

Terima kasih yang tidak terhingga saya haturkan kepada kedua orangtua saya, yang telah mengasuh dan mendidik saya dengan tulus hingga saat ini.

Terima kasih kepada Bapak Ahmad Fathan Hidayatullah, S.T., M.Cs. yang telah membimbing, mengajar, dan memberikan saya motivasi dalam pembuatan laporan skripsi ini.

Terima kasih kepada semua pihak yang tidak bisa saya sebutkan satu-satu, atas segala do'a dan semangat yang telah diberikan kepada saya.



HALAMAN MOTO

“Rencana Allah selalu menjadi rencana yang sebaik-baiknya”

“Tidak ada sesuatu yang tidak bisa diselesaikan”

“Jangan takut dengan masalah besar, karena kita hamba dari Allah Yang Maha Besar”



KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Alhamdulillahirobbil'alamin, penulis memanjatkan rasa syukur yang setinggi-tingginya kepada Allah *Subhanahu wata'ala* atas segala nikmat yang telah diberikan, sehingga penulis bisa menyelesaikan laporan skripsi dengan lancar tanpa hambatan yang berat. Shalawat serta Salam terhaturkan kepada junjungan Nabi Muhammad *Sallallahu 'alaihi wasallam* yang telah membawa cahaya Islam ke muka bumi, dan yang dinantikan syafa'atnya di *Yaumul Qiyamah*.

Banyak pihak terlibat dalam penyelesaian tugas akhir ini. Oleh karena itu, penulis hendak menyampaikan apresiasi dan terima kasih setinggi-tingginya kepada pihak-pihak yang disebutkan di bawah ini:

1. Kedua orang tua penulis yang selalu ada, mendukung, dan mendo'akan anak-anaknya
2. Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia
3. Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Program Studi Informatika – Program Sarjana
4. Ahmad Fathan Hidayatullah, S.T., M.Cs. selaku Dosen Pembimbing
5. Bapak dan Ibu dosen Jurusan Informatika, yang telah memberikan ilmu yang bermanfaat kepada penulis, semoga bapak dan ibu selalu diberikan kesehatan serta lindungan dari Allah SWT.
6. Semua keluarga, yang telah memberikan semangat dan motivasi
7. Dhinda Ayu Rasitta, yang selalu membantu dan memberikan semangat positif
8. Teman-teman Halghib, yang selalu memberikan semangat positif
9. Semua orang yang telah memberikan semangat dan do'a kepada penulis.

Tugas akhir ini jauh dari kata sempurna, namun penulis berharap dengan disusunnya laporan skripsi ini bisa bermanfaat kepada semua orang dan bisa dikembangkan menjadi hal yang lebih besar lagi dan bisa bermanfaat bagi agama, nusa, dan bangsa.

Yogyakarta, 11 Januari 2021

(Habib Faizal Fadli)

SARI

Cyberbullying menjadi sebuah masalah yang harus mendapat perhatian serius oleh semua pihak. Di samping tindakan ini merupakan kebiasaan yang buruk, *cyberbullying* juga memberikan dampak yang mengerikan, mulai dari gangguan psikis korban, hingga berujung pada kasus bunuh diri.

Tujuan dari penelitian ini adalah membuat sebuah aplikasi untuk mengidentifikasi konten yang mengandung makna perundungan secara daring (*cyberbullying*) pada media sosial. Media sosial yang dipilih adalah *Twitter*. Setidaknya, ada 6835 baris data yang telah dikumpulkan. Data-data tersebut berisi dua jenis cuitan baik cuitan yang memiliki kecenderungan *cyberbullying* dan *non-cyberbullying*.

Untuk mencapai tujuan penelitian, peneliti menggunakan lima langkah penelitian, yaitu pengumpulan data, preprocessing, klasifikasi, evaluasi, dan deteksi konten. Algoritma *Deep Learning* yang diimplementasikan dalam penelitian ini adalah *Bidirectional Long Short Term Memory*. Setelah model terbentuk, kemudian dibuat sebuah fitur untuk deteksi konten.

Klasifikasi menggunakan algoritma *Bidirectional Long Short Term Memory* terbukti memiliki performa yang lebih baik dibandingkan algoritma *Long Short Term Memory*. Model tersebut menjadi inti dari fitur identifikasi *cyberbullying* yang diimplementasikan dalam sebuah fitur deteksi konten.

Kata kunci: *twitter*, *cyberbullying*, *deep learning*, klasifikasi.

GLOSARIUM

<i>Batch Size</i>	jumlah sampel yang dilatih pada setiap <i>epoch</i> .
<i>Data training</i>	data pelatihan untuk dijadikan model.
<i>Dataset</i>	kumpulan data dalam pembentukan model dan fitur.
<i>Deep Learning</i>	pembangunan sistem berdasarkan data.
<i>Epoch</i>	jumlah tahapan pelatihan untuk seluruh data
Klasifikasi	penentuan kelas.
<i>Pre-processing</i>	pembersihan data hingga menjadi data terstruktur untuk dijadikan data <i>training</i> .



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	viii
GLOSARIUM.....	ix
DAFTAR ISI	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI.....	4
2.1 Dasar Teori	4
2.1.1 <i>Cyberbullying</i>	4
2.1.2 Pengolahan Bahasa Alami	4
2.1.3 <i>Text Classification</i>	5
2.1.4 <i>Long Short-Term Memory (LSTM) Network</i>	6
2.1.5 <i>Bidirectional Long Short-Term Memory (BiLSTM) Network</i>	7
2.1.6 <i>Performance Evaluation Measure</i>	8
2.2 Penelitian Sebelumnya.....	11
BAB III METODOLOGI PENELITIAN	14
3.1 Langkah-langkah Penelitian	14
3.2 Uraian Penelitian	15
3.2.1 Pengumpulan Data	15
3.2.2 <i>Preprocessing</i>	16
3.2.3 Klasifikasi	19
3.2.4 Evaluasi.....	22
3.2.5 Deteksi Konten.....	23
BAB IV HASIL DAN PEMBAHASAN	24
4.1 Pengumpulan Data.....	24
4.2 <i>Preprocessing</i>	25
4.2.1 <i>Cleaning</i>	25
4.2.2 <i>Normalize</i>	26
4.2.3 <i>Stemming</i>	27
4.2.4 <i>Remove Stopwords</i>	27
4.3 Klasifikasi.....	28
4.3.1 <i>Splitting Dataset</i>	28
4.3.2 <i>Tokenizing</i>	29
4.3.3 <i>Create Model</i>	30

	xi
4.3.4 <i>Training Model</i>	32
4.4 Evaluasi	32
4.4.1 Skenario Model	32
4.4.2 Analisis Skenario	33
4.5 Deteksi Konten	39
BAB V KESIMPULAN DAN SARAN	41
5.1 Kesimpulan	41
5.2 Saran	42
DAFTAR PUSTAKA	43
LAMPIRAN	45



DAFTAR TABEL

Tabel 2.1 Daftar perbandingan penelitian sebelumnya.....	12
Tabel 3.1 Contoh data yang dihimpunkan	15
Tabel 3.2 Implementasi <i>cleaning</i> pada cuitan.....	17
Tabel 3.3 <i>Normalize slangword</i> pada cuitan.....	18
Tabel 3.4 <i>Stemming</i> pada cuitan	18
Tabel 3.5 <i>Removing stopwords</i> pada cuitan.....	19
Tabel 3.6 Skenario model LSTM.....	22
Tabel 3.7 Skenario model BiLSTM.....	22
Tabel 4.1 Contoh penerapan <i>preprocessing</i>	28
Tabel 4.2 <i>Confusion matrix</i> LSTM.....	33
Tabel 4.3 <i>Confusion matrix</i> LSTM+ <i>Dropout</i>	33
Tabel 4.4 <i>Confusion matrix</i> BiLSTM	33
Tabel 4.5 <i>Confusion matrix</i> BiLSTM+ <i>Dropout</i>	33
Tabel 4.6 <i>Accuracy, Precision, Recall, dan F1-Score</i> LSTM.....	34
Tabel 4.7 <i>Accuracy, Precision, Recall, dan F1-Score</i> BiLSTM.....	34

DAFTAR GAMBAR

Gambar 2.1 Alur pengolahan bahan alami	5
Gambar 2.2 Proses klasifikasi teks	6
Gambar 2.3 Arsitektur LSTM.....	7
Gambar 2.4 Arsitektur BiLSTM	8
Gambar 2.5 <i>Confussion matrix</i>	10
Gambar 3.1 Langkah-langkah penelitian	14
Gambar 3.2 Tahapan <i>preprocessing</i>	16
Gambar 3.3 Tahapan klasifikasi	19
Gambar 3.4 Perbedaan penggunaan <i>dropout</i>	20
Gambar 3.5 Alur kerja model	21
Gambar 3.6 Deteksi konten.....	23
Gambar 4.1 Cacah data	24
Gambar 4.2 Kode tahapan <i>cleaning</i>	26
Gambar 4.3 Kode tahapan <i>normalize</i>	27
Gambar 4.4 Kode tahapan <i>stemming</i>	27
Gambar 4.5 Kode tahapan <i>remove stopwords</i>	28
Gambar 4.6 <i>Splitting data train, data test, dan data validation</i>	29
Gambar 4.7 <i>Tokenizing</i>	29
Gambar 4.8 Model LSTM.....	31
Gambar 4.9 Model BiLSTM.....	31
Gambar 4.10 <i>Training LSTM</i>	32
Gambar 4.11 <i>Training BiLSTM</i>	32
Gambar 4.12 Grafik LSTM.....	35
Gambar 4.13 Grafik LSTM+ <i>dropout</i>	36
Gambar 4.14 Grafik BiLSTM.....	37
Gambar 4.15 Grafik BiLSTM+ <i>dropout</i>	37
Gambar 4.16 Grafik gabungan LSTM	38
Gambar 4.17 Grafik gabungan BiLSTM	38
Gambar 4.18 Contoh hasil klasifikasi kalimat <i>cyberbullying</i>	39
Gambar 4.19 Contoh hasil klasifikasi kalimat <i>non-cyberbullying</i>	40

BAB I PENDAHULUAN

1.1 Latar Belakang

Media sosial merupakan tempat di mana penggunanya melakukan interaksi dengan pengguna lain secara daring tanpa mengenal waktu dan tempat. Indonesia sendiri merupakan negara dengan jumlah pengguna media sosial tertinggi di dunia. Kementerian Komunikasi dan Informatika (Kemenkominfo) menyatakan bahwa 95% dari sekitar 63 juta pengguna internet adalah pengguna media sosial (Kominfo, 2013). Menurut *Country Industry Head Twitter Indonesia* mengklaim bahwa Indonesia termasuk negara dengan pertumbuhan pengguna aktif harian *Twitter*-nya paling besar (Clinten, 2019).

Twitter sendiri merupakan aplikasi yang sering digunakan oleh masyarakat Indonesia. Penggunaan *Twitter* di Indonesia tidak hanya ditujukan untuk perorangan. Akan tetapi, *Twitter* juga digunakan oleh lembaga-lembaga negara, komunitas, hingga toko *online*. Namun, tidak semua pengguna menggunakan teknologi ini dengan bijak. Tindakan negatif seperti penipuan, penyebaran *hoax*, menyebarkan opini yang cenderung mengandung ujaran kebencian, hingga perundungan secara daring (*cyberbullying*) banyak dilakukan oleh pengguna *Twitter* (Bohang, 2017). Selama tahun 2019 masyarakat telah melaporkan setidaknya 244.738 jumlah konten pornografi. Lalu, terdapat sejumlah 19.970 konten kategori perjudian. Ada pula konten penipuan sejumlah 18.845, dan konten informasi hoaks, serta konten mengandung SARA, terorisme, radikalisme, pelanggaran HAK, serta kekerasan terhadap anak sejumlah 15.361 (Redaksi WE Online, 2020). Hal ini tentu menjadi dampak negatif penggunaan media sosial. Dampak-dampak yang dapat ditimbulkan seperti kerusuhan karena menyebarnya berita bohong, kerugian materiil karena penipuan, hingga berbagai kasus yang ditimbulkan karena *cyberbullying* (Hinduja & Patchin, 2010).

Cyberbullying atau perundungan secara daring adalah tindakan penyerangan, penghinaan, atau menyakiti orang lain secara sengaja dan berulang-ulang pada sosial media, pesan, atau dengan cara lainnya (Hinduja & Patchin, 2010). *Cyberbullying* di media sosial *Twitter* dilakukan dengan menulis cuitan yang mengandung kata-kata hinaan atau kata-kata kasar, bahkan kata-kata yang menjurus kepada penghinaan terhadap SARA. Pemerintah mengumumkan setidaknya 84% remaja berusia 12 sampai 17 tahun di Indonesia menjadi korban tindakan perundungan (*bullying*) dan kebanyakan kasus *bullying* yang ditemukan

merupakan *cyberbullying* (Laksana, 2017). *Cyberbullying* menjadi kekhawatiran publik karena banyak kasus *cyberbullying* sering dikaitkan dengan tindakan bunuh diri. Salah satu organisasi non-profit, *Cyber Bullying Research Center* mengungkapkan bahwa kebiasaan *bullying* baik secara langsung maupun secara daring di kalangan remaja dapat mengakibatkan depresi, tindakan bunuh diri dan percobaan pembunuhan (Hinduja & Patchin, 2010). Oleh karena efek berbahaya yang ditimbulkan *cyberbullying*, tindakan pencegahan atau deteksi perlu dilakukan agar tidak membahayakan korban.

Penelitian dilakukan dengan melakukan lima tahapan proses. Langkah pertama yaitu pengumpulan data, lalu *preprocessing*, kemudian ekstraksi fitur, setelah itu klasifikasi, dan terakhir evaluasi. Metode yang digunakan yaitu *deep learning* dan algoritma yang digunakan yaitu *Long Short-Term Memory (LSTM) Network* dan *Bidirectional Long Short-Term Memory (BiLSTM) Network*. Penelitian ini penting dilakukan karena belum ditemukan model yang dibuat khusus untuk mendeteksi dan mengklasifikasikan cuitan yang berbahasa Indonesia dan bermakna *cyberbullying* dengan metode *deep learning* terutama dengan 2 algoritma tersebut. Kedua algoritma tersebut dinilai sangat tepat dalam melakukan deteksi dan klasifikasi dibandingkan algoritma- algoritma yang lainnya.

Penelitian ini mengklasifikasikan data untuk mendeteksi *cyberbullying* di media sosial *Twitter* menjadi dua kelas yaitu *cyberbullying* dan bukan *cyberbullying*. Kelas *cyberbullying* berisi cuitan berupa kata-kata yang mengandung unsur *cyberbullying*, sementara kelas bukan *cyberbullying* berisi cuitan berupa kata-kata yang tidak mengandung unsur *cyberbullying*. Dengan dikembangkannya penelitian ini, diharapkan akan dapat membantu orang tua, pemerintah, dan negara untuk melindungi generasi muda dari perundungan secara daring (*cyberbullying*) dan menekan jumlah para pelaku *cyberbullying*.

12 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah sebagai berikut:

- a. Bagaimana melakukan klasifikasi untuk mendeteksi cuitan yang mengandung *cyberbullying* pada *Twitter* dengan model LSTM dan BiLSTM?
- b. Bagaimana performa model model LSTM dan BiLSTM dalam melakukan klasifikasi untuk mendeteksi cuitan yang mengandung *cyberbullying* pada *Twitter*?

13 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

- a. Melakukan klasifikasi untuk mendeteksi pada *Twitter* cuitan yang mengandung *cyberbullying* dengan metode LSTM dan BiLSTM.
- b. Melakukan pengujian performa model dalam klasifikasi untuk melihat hasil dari *Accuracy* dan *F1-Score* yang dihasilkan dengan model LSTM dan BiLSTM.

14 Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut:

- a. Data bersifat statis yang diperoleh melalui Twitter API.
- b. Data cuitan yang digunakan hanya yang berbahasa Indonesia.
- c. Data diambil berdasarkan kata kunci, baik yang mengandung *cyberbullying* atau tidak mengandung *cyberbullying*.

15 Manfaat

Manfaat dari penelitian ini adalah untuk membantu pemerintah, orang tua, dan negara dalam melindungi generasi muda dari perundungan secara daring (*cyberbullying*) dan untuk membantu mengetahui kalimat *cyberbullying* dan kalimat *non-cyberbullying* menggunakan metode LSTM dan BiLSTM.

16 Sistematika Penulisan

Laporan skripsi ini disusun atas lima bagian, yang terdiri dari:

BAB I Pendahuluan, Bagian pendahuluan membahas latar belakang dan sebab penelitian ini dilakukan.

BAB II Landasan Teori, Bagian ini membahas tentang teori-teori yang berkaitan dalam proses identifikasi *cyberbullying* dan hal-hal lain yang mendukung.

BAB III Metodologi Penelitian, Metodologi penelitian membahas langkah-langkah yang dilakukan dalam identifikasi *cyberbullying*.

BAB IV Hasil dan Pembahasan, Bagian ini membahas performa model dalam melakukan klasifikasi *cyberbullying* mulai dari hasil pengumpulan data, *preprocessing*, pembuatan model, menguji model, evaluasi model, serta kelebihan dan kekurangan model yang digunakan.

BAB V Kesimpulan dan Saran, Bagian ini berisi rangkuman dari segala tahap yang telah dilakukan dalam penelitian, dan berisi saran-saran untuk penelitian selanjutnya.

BAB II LANDASAN TEORI

21 Dasar Teori

2.1.1 Cyberbullying

Cyberbullying adalah perlakuan yang ditujukan untuk mempermalukan, menakut-nakuti, melukai, atau menyebabkan kerugian bagi pihak yang lemah dengan menggunakan sarana komunikasi Teknologi Informasi (Rahayu, 2012). Dengan kalimat lain, *cyberbullying* adalah tindakan perundungan yang dilakukan melalui internet dan kemajuan teknologi komunikasi.

Ada tiga hal yang membedakan *bullying* tradisional dengan *cyberbullying* (Akbar & Utari, 2014). Pertama, *bullying* tradisional merupakan tindakan yang dilakukan secara langsung bertatap muka (*face-to-face*), namun *cyberbullying* tidak. Kedua, *cyberbullying* tidak memberi dampak secara fisik, melainkan dampak yang ditimbulkan menyerang psikis korban. Ketiga, tidak seperti tradisional *bullying*, *cyberbullying* dapat muncul kapan saja tidak dibatasi oleh ruang dan waktu.

Bentuk perilaku *cyberbullying* yang dilakukan pelaku kepada korban seperti (Akbar & Utari, 2014).

- a. Pelaku mengirimkan komentar atau pesan *cyberbullying* berulang kali
- b. Pelaku mengirim komentar atau pesan *cyberbullying* dengan kata kasar
- c. Pelaku ikut bereaksi menambahkan pesan *cyberbullying* dari pelaku lainnya.

2.1.2 Pengolahan Bahasa Alami

Pengolahan Bahasa Alami atau *Natural Language Processing* (NLP) adalah sebuah subbidang dari kecerdasan buatan (*Artificial Intelligence*) dan merupakan hal penting dalam *Text Mining*. Tujuan dilakukan NLP pada sistem adalah agar sistem bisa mengerti bahasa manusia dengan menuliskannya dalam bentuk teks atau dokumen kemudian menjadikannya kalimat yang baku dan terstruktur sehingga sistem bisa dengan mudah memanipulasi dan memberikan respon yang semestinya. Adapun komponen-komponen utama dari PBA sebagai berikut (Soyusiawaty & Haspiyan, 2015).

Parser

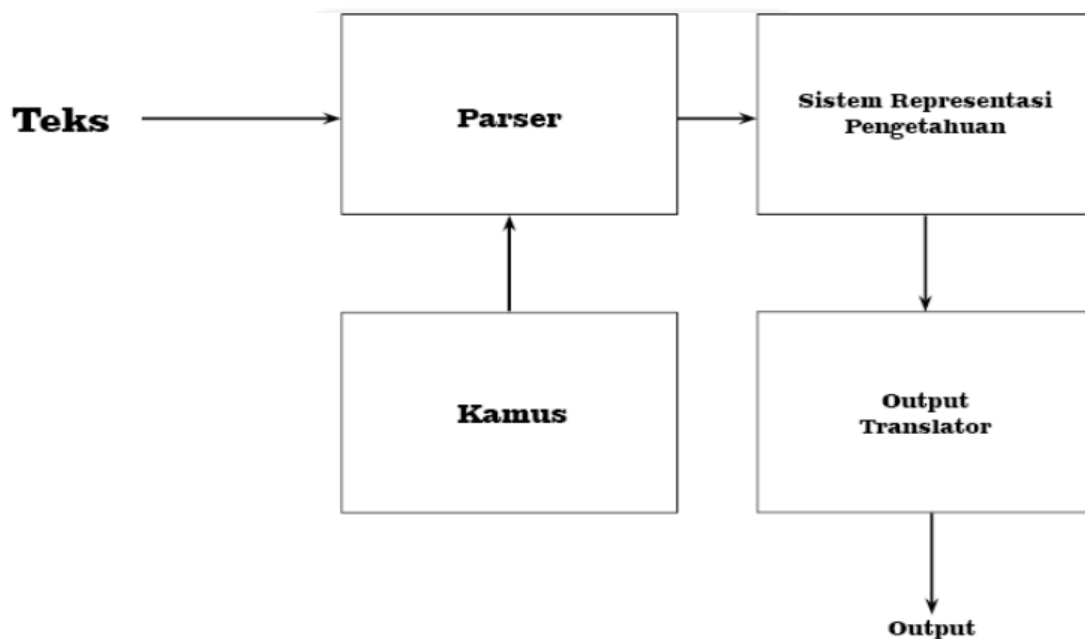
Bagian ini berfungsi untuk menerima masukan berupa teks kemudian menguraikannya dalam bagian-bagian gramatikal seperti subyek, kata kerja, kata benda, obyek, dan lain-lain.

Sistem Representasi Pengetahuan

Bagian ini berfungsi untuk menganalisis keluaran dari bagian *parser* untuk menentukan maknanya. Sistem representasi pengetahuan berupa kamus yang berisikan kata-kata bahasa alami.

Output Translator

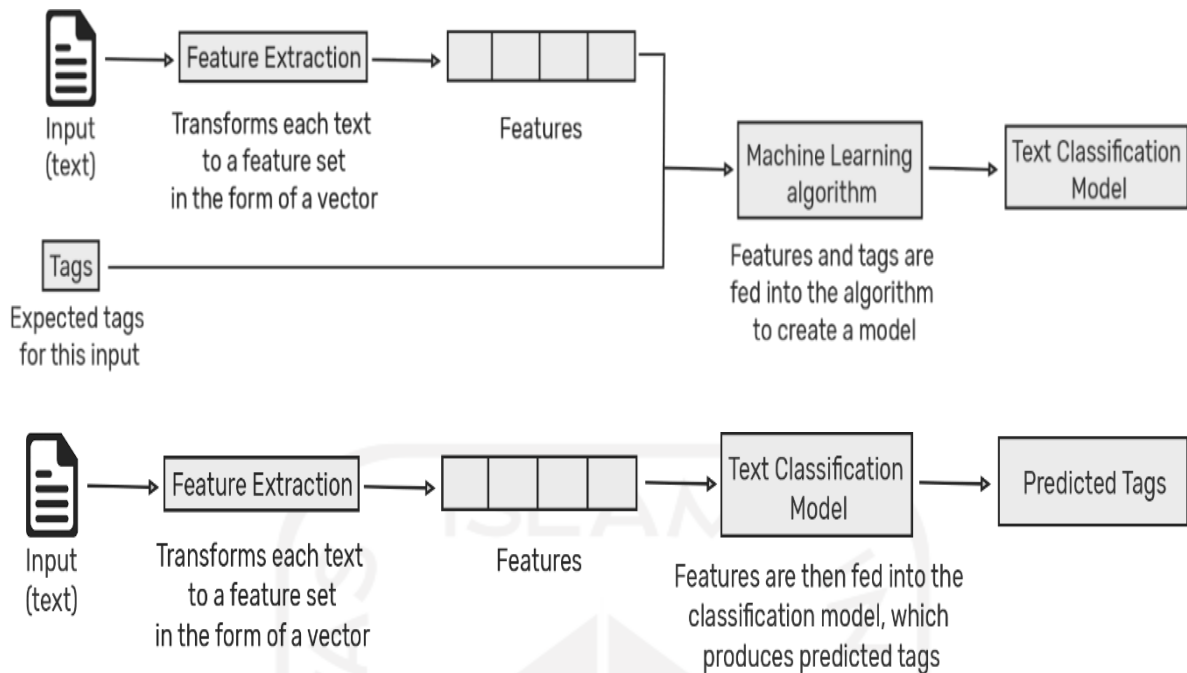
Bagian ini merepresentasikan sistem pengetahuan berupa hasil terjemahan masukan bahasa alami. Alur pengolahan bahasa alami dapat dilihat pada Gambar 2.1.



Gambar 2.1 Alur pengolahan bahan alami

2.1.3 Text Classification

Text Classification atau klasifikasi teks merupakan teknologi utama dalam *text mining*, yang bertujuan menetapkan sebuah dokumen teks ke dalam satu atau beberapa kategori yang telah ditentukan (Dai & Liu, 2014). Klasifikasi teks dapat dilakukan dengan memanfaatkan pembelajaran mesin. Dengan menggunakan data latih, pembelajaran mesin mampu membangun pengetahuan berdasarkan data tersebut



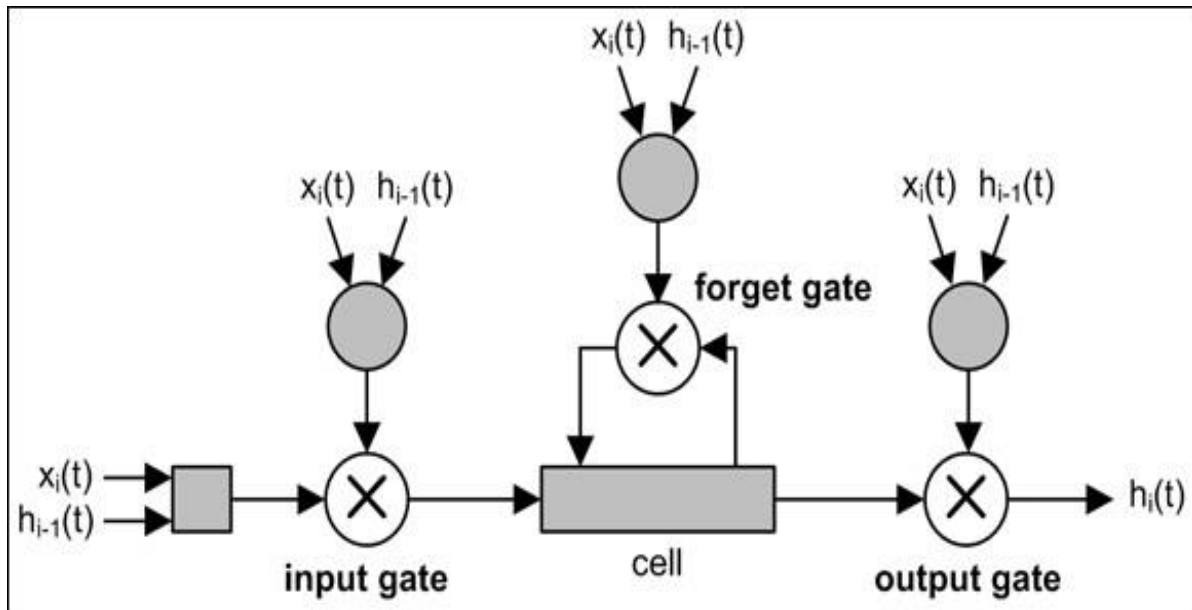
Gambar 2.2 Proses klasifikasi teks

Sumber: devopedia.org

Gambar 2.2 menunjukkan proses klasifikasi teks. Langkah yang perlu dilakukan dalam klasifikasi teks adalah mengubah teks menjadi numerik dalam bentuk vektor menggunakan *feature extraction*. Setelah itu, *feature set* dan *tags* digunakan pada algoritma pembelajaran mesin untuk membangun model klasifikasi. Proses ini dapat disebut sebagai proses pelatihan (*training*). Selanjutnya, model klasifikasi yang telah dihasilkan digunakan untuk melakukan prediksi teks. Proses ini dapat disebut sebagai proses pengujian (*testing*).

2.1.4 Long Short-Term Memory (LSTM) Network

LSTM adalah modifikasi dari *Recurrent Neural Network* (RNN) dengan adanya penambahan *memory cell* yang digunakan untuk menyimpan informasi dengan jangka waktu yang panjang, serta LSTM juga dapat menangani masalah *vanishing gradient* yang terdapat pada RNN saat memproses data sekuensial yang panjang dengan menggunakan satu set gerbang yang digunakan untuk mengontrol informasi yang masuk ke memori (Manaswi, 2018).



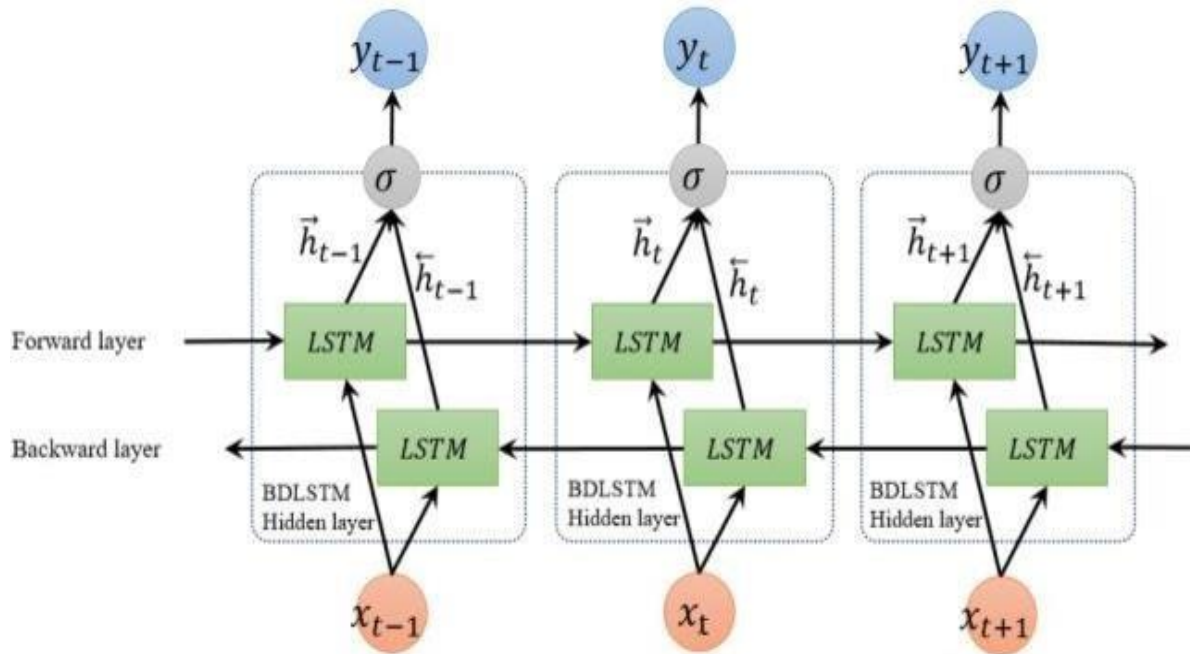
Gambar 2.3 Arsitektur LSTM

Sumber: indoml.com

Gambar 2.3 menunjukkan arsitektur dari LSTM. *Cell state* merupakan tempat untuk menyimpan informasi yang diberikan dari satu langkah waktu ke langkah waktu berikutnya. *Gate units* berperan dalam memproses informasi yang dibutuhkan dan dibuang. *Gate units* terdiri dari *input gate*, *forget gate*, dan *output gate*. *Input gate* merupakan *gate* yang berfungsi untuk memutuskan nilai *input* yang akan diteruskan pada *cell state* untuk diperbaharui. *Forget gate* merupakan *gate* yang memutuskan informasi mana yang perlu dibuang dari *cell state*. *Output gate* merupakan *gate* yang memutuskan *output* yang akan dihasilkan.

2.1.5 Bidirectional Long Short-Term Memory (BiLSTM) Network

BiLSTM adalah perkembangan dari model LSTM di mana terdapat dua lapisan yang prosesnya saling berkebalikan arah, model ini sangat baik untuk mengenali pola dalam kalimat karena setiap kata dalam dokumen diproses secara sekuensial, karena cuitan dapat dipahami bila pembelajaran secara berurut setiap kata. Lapisan di bawahnya bergerak maju (*forward*), yaitu memahami dan memproses dari kata pertama menuju kata terakhir sedangkan lapisan di atasnya bergerak mundur (*backward*), yaitu memahami dan memproses dari kata terakhir menuju kata pertama. Dengan adanya lapisan dua arah yang saling berlawanan ini maka model dapat memahami dan mengambil perspektif dari kata terdahulu dan kata terdepan, sehingga proses pembelajaran akan semakin dalam yang berdampak pada model akan lebih memahami konteks pada cuitan tersebut.



Gambar 2.4 Arsitektur BiLSTM

Sumber: gabormelli.com

Gambar 2.4 dapat dilihat bahwa setiap *hidden unit* keluaran unit pada lapisan bawah dan atas digabungkan membentuk nilai fitur kata tersebut dengan ukuran lebih panjang daripada menggunakan LSTM biasa. Karena lebih panjang nilai fitur, maka informasi yang akan diproses pada tahap selanjutnya yaitu *feed forward neural* akan mengklasifikasikan dengan lebih akurat.

BiLSTM akan sangat bermanfaat dalam hal pelabelan sekuensial apabila memiliki akses terhadap kedua informasi dari sebelum dan sesudahnya. Namun *hidden state* pada LSTM hanya mengambil informasi dari sebelumnya (masa lalu), sedangkan untuk informasi yang ada setelahnya tidak diketahui. Permasalahan tersebut dapat dipecahkan dengan menggunakan BiLSTM (Ma & Hovy, 2016). Pada dasarnya BiLSTM terdiri dari dua LSTM, *forwardLSTM* dan *backward LSTM*, sehingga gabungan tersebut akan menangkap informasi dari kedua arah.

2.1.6 Performance Evaluation Measure

Performance Evaluation Measure (PEM) adalah tahapan yang digunakan untuk mengetahui performa model. Ada banyak persamaan untuk menghitung nilai PEM. Beberapa persamaan dalam PEM antara lain:

Accuracy

Accuracy adalah perbandingan antara informasi benar yang dijawab sistem dengan keseluruhan data, dapat dilihat pada persamaan (2.1).

$$Accuracy = \frac{TN + TP}{TN + FN + FP + TP} \cdot 100\% \quad (2.1)$$

Precision

Precision adalah ketepatan nilai antara permintaan pengguna dengan respon sistem, dapat dilihat pada persamaan (2.2).

$$Precision = \frac{TP}{FP + TP} \cdot 100\% \quad (2.2)$$

Recall

Recall adalah ketepatan antara informasi yang sama dengan informasi yang pernah dipanggil sebelumnya, dapat dilihat pada persamaan (2.3).

$$Recall = \frac{TP}{FN + TP} \cdot 100\% \quad (2.3)$$

F1-Score

F1-Score adalah perbandingan rata-rata *precision* dan *recall* yang dibobotkan, dapat dilihat pada persamaan (2.4).

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \cdot 100\% \quad (2.4)$$

PEM biasanya direpresentasikan dengan sebuah *confusion matrix*, yaitu matriks yang sangat intuitif dalam memberi informasi performa model yang dapat dilihat pada Gambar 2.5.

		Predicted Values	
		Positive (0)	Negative (1)
Actual Values	Positive (0)	TP	FP
	Negative (1)	FN	TN

Gambar 2.5 *Confusion matrix*

Keterangan:

True Positive (TP) : jumlah dokumen dari kelas positif yang benar diklasifikasikan sebagai kelas positif.

True Negative (TN) : jumlah dokumen dari kelas negatif yang benar diklasifikasikan sebagai kelas negatif.

False Positive (FP) : jumlah dokumen dari kelas positif yang salah diklasifikasikan sebagai kelas negatif.

False Negative (FN) : jumlah dokumen dari kelas negatif yang salah diklasifikasikan sebagai kelas positif.

22 Penelitian Sebelumnya

Penelitian mengenai klasifikasi teks sudah banyak dilakukan oleh peneliti-peneliti sebelumnya. Konten kasar yang diklasifikasikan berdasarkan cuitan media sosial *Twitter* dengan kamus bahasa Indonesia juga pernah dilakukan sebelumnya. (Hidayatullah *et al.*, 2019) melakukan klasifikasi berdasarkan cuitan menjadi dua kelas dan melakukan perbandingan terhadap performa algoritma NBC dan SVM dalam melakukan klasifikasi. Berikut adalah hasil *Accuracy*, *Precision*, *Recall*, dan *F1-Score* dari masing-masing model yang digunakan di mana NBC mempunyai nilai 0.9834; 0.9912; 0.9762; 0.9836 dan SVM mempunyai nilai 0.9928; 0.9914; 0.9946; 0.9930.

Penelitian tentang deteksi *cyberbullying* pada media sosial *Twitter* berbahasa Indonesia pernah dilakukan sebelumnya. (Abdulloh & Hidayatullah, 2019) melakukan deteksi cuitan pada media sosial *Twitter*. Data yang berupa cuitan tersebut nantinya akan dimasukkan ke dalam kelas *cyberbullying* dan kelas *non-cyberbullying*. Deteksi cuitan tersebut menggunakan metode Machine Learning dengan algoritma NBC, SVM, Logistic Regression, dan KNN. Berikut adalah hasil *Accuracy*, *Precision*, *Recall*, dan *F1-Score* dari masing-masing algoritma yang digunakan di mana algoritma NBC mempunyai nilai 0.961; 0.96; 0.96; 0.96, SVM mempunyai nilai 0.994; 0.99; 0.99; 0.99, Logistic Regression mempunyai nilai 0.997; 1.00; 1.00; 1.00, dan KNN mempunyai nilai 0.918; 0.93; 0.92; 0.92.

Penelitian tentang deteksi *cyberbullying* pada media sosial selain *Twitter* juga sudah pernah dilakukan sebelumnya. (Hosseinmardi *et al.*, 2015) melakukan penelitian tentang klasifikasi *cyberbullying* menggunakan algoritma NBC dan SVM pada media sosial *Instagram* berdasarkan foto dan komentar. Berikut adalah hasil *Accuracy*, *Precision*, dan *Recall* dari masing-masing algoritma yang digunakan di mana algoritma SVM mempunyai nilai 0.74; 0.74; 0.78 dan SVM mempunyai nilai 0.87; 0.88; 0.87.

Ada pula penelitian tentang deteksi *cyberbullying* yang digabungkan dengan metode pendekatan psikologi yang pernah dilakukan sebelumnya. (Balakrishnan *et al.*, 2019) melakukan penelitian ini menggunakan model *Big Five* and *Triad* dan algoritma *Random Forest*. Berikut adalah hasil *Precision*, *Recall*, dan *F-Measure* dari algoritma yang digunakan di mana algoritma *Random Forest* mempunyai nilai 0.960; 0.952; 0.929.

Penelitian tentang *bullying* yang mengambil data lebih dari satu media sosial juga pernah dilakukan sebelumnya. (Agrawal & Awekar, 2018) melakukan penelitian tentang *cyberbullying* di tiga media sosial yaitu *Formspring*, *Twitter*, dan *Wikipedia* dengan membagi 4 kelas *bullying* yaitu *bully*, *racism*, *sexism*, dan *attack* kemudian menguji data dengan mengklasifikasikannya dengan algoritma CNN, LSTM, BiLSTM, dan BiLSTM *with attention*. Akan tetapi, pada penelitian tersebut masih menggunakan korpus bahasa Inggris. Berikut adalah hasil *Precision*, *Recall*, dan *F1-Score* dari masing-masing algoritma yang digunakan di mana CNN mempunyai nilai 0.93; 0.90; 0.91, LSTM mempunyai nilai 0.91; 0.85; 0.88, BiLSTM mempunyai nilai 0.91; 0.81; 0.86, dan BiLSTM *with attention* mempunyai nilai 0.90; 0.91; 0.91. Tabel 2.1 merupakan daftar perbandingan peneliti sebelumnya.

Tabel 2.1 Daftar perbandingan penelitian sebelumnya

Penelitian	Metode	Nilai
Identifikasi Konten Kasar pada <i>Tweet</i> Bahasa Indonesia (Hidayatullah <i>et al.</i> , 2019)	NBC dan SVM	<i>Accuracy</i> , <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score</i> : NBC: 0.9834; 0.9912; 0.9762; 0.9836 SVM: 0.9928; 0.9914; 0.9946; 0.9930
Deteksi <i>Cyberbullying</i> pada Cuitan Media Sosial <i>Twitter</i> (Abdulloh & Hidayatullah, 2019)	NBC, SVM, <i>Logistic Regression</i> , dan KNN	<i>Accuracy</i> , <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score</i> NBC: 0.961; 0.96; 0.96; 0.96 SVM: 0.994; 0.99; 0.99; 0.99 <i>Logistic Regression</i> : 0.997; 1.00; 1.00; 1.00 KNN: 0.918; 0.93; 0.92; 0.92

Penelitian	Metode	Nilai
<i>Detection of Cyberbullying Incidents on the Instagram Social Network</i> (Hosseinmardi et al., 2015)	NBC dan SVM	<i>Accuracy, Precision, dan Recall</i> NBC: 0.74; 0.74; 0.78 SVM: 0.87; 0.88; 0.87
<i>Cyberbullying detection on twitter using Big Five and Dark Triad features</i> (Balakrishnan et al., 2019)	<i>Random Forest</i>	<i>Precision, Recall, dan F-Measure</i> <i>Random Forest</i> : 0.960; 0.952; 0.929
<i>Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms</i> (Agrawal & Awekar, 2018)	CNN, LSTM, BiLSTM, dan BiLSTM with attention	<i>Precision, Recall, dan F1-Score</i> CNN: 0.93; 0.90; 0.91 LSTM: 0.91; 0.85; 0.88 BiLSTM: 0.91; 0.81; 0.86 BiLSTM with attention: 0.90; 0.91; 0.91

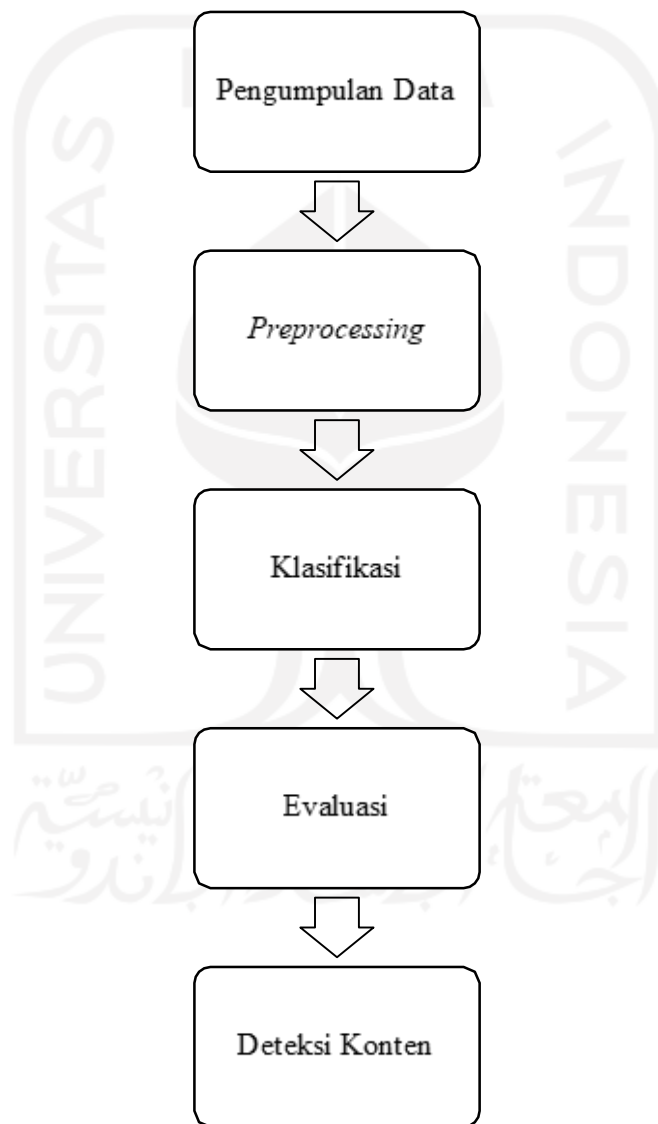
Berdasarkan Tabel 2.1 perbedaan dengan penelitian sebelumnya yaitu pada penelitian ini menggunakan bahasa Indonesia sebagai bahasa yang digunakan untuk melakukan identifikasi *cyberbullying*. Lalu, selain membuat model untuk klasifikasi, penelitian ini juga memiliki luaran berupa sebuah deteksi konten *cyberbullying*.

BAB III

METODOLOGI PENELITIAN

31 Langkah-langkah Penelitian

Langkah-langkah keseluruhan pengerjaan penelitian ini terdiri dari pengumpulan data, *preprocessing*, klasifikasi, evaluasi, dan deteksi konten. Gambar 3.1 menunjukkan alur kerja dari penelitian ini.



Gambar 3.1 Langkah-langkah penelitian

32 Uraian Penelitian

Berdasarkan alur kerja pada Gambar 3.1, berikut merupakan uraian penjelasan pada setiap langkah-langkahnya.

3.2.1 Pengumpulan Data

Langkah pertama yang dilakukan pada penelitian ini adalah pengumpulan data. Data dikumpulkan dari media sosial *Twitter* berupa cuitan-cuitan pengguna berbahasa Indonesia. Kata kunci yang digunakan untuk mengambil data berupa ‘semangat’, ‘ngartis’, ‘iq jongkok’, ‘dungu’, ‘jelek’, ‘kampungan’, ‘norak’, ‘songong’, ‘sombong’, ‘tolol’, ‘gendut’, ‘udik’, dan lain-lain. Data yang didapat kemudian disimpan dengan format *xlsx*. Berikutnya dilakukan proses *labeling* yaitu membagi data cuitan menjadi dua kelas, yaitu kelas *cyberbullying* dengan label nol (1) dan kelas *non-cyberbullying* dengan label satu (0). Proses pelabelan dilakukan sendiri oleh penulis dengan pelabelan manual yang dapat dilihat pada Tabel 3.1.

Tabel 3.1 Contoh data yang dihimpunkan

Cuitan	Label
@Hilmi28 @prabowo Semoga Allah SWT melindungi dan menjaga pak @prabowo bang @sandiuno barakallohu fiikum	0
@humantolol kata papa juga gaboleh sombong	0
@ilovebillies kata mama ga boleh sombong	0
@bilonk Duh belagu amat Wkwk	1
@danillajpr Danilla gendut	1
@bbylibur @seolhyunlibur @sooyaaalibur @tehalibur @holidayjailed beginilah gambaran kalau iq anda jongkok 🙏🙏	1
@beervodkas SONGONG	1
@BEJOS09958347 @ardi_riau @mohmahfudmd Di si dungu	1
@berteman_mari Biar mereka pada makan batu aja jendral...songong, arogan, blegug	1
@BATMANXBABALL YaAllah udik sekali diriku 😭	0
@beauthingy Gapernah beli di olshop (apalagi yg import bangkok dll) krn pasti ngatung jelek gt:(0
@bobbyifaa Kalo di liat dari followers yg udah 4k si keknya doi ngartis 😏	0

3.2.2 *Preprocessing*

Preprocessing merupakan tahapan awal untuk menyiapkan data sebelum dilakukan proses berikutnya (Mujilahwati, 2016). *Preprocessing* berfungsi untuk membersihkan data cuitan yang berupa teks dari kata-kata atau simbol yang kurang penting, sehingga menjadi data teks yang terstruktur. Tahapan *preprocessing* di antaranya yaitu *cleaning*, *stemming*, dan *remove stopwords* yang akan dilihat pada Gambar 3.2.



Gambar 3.2 Tahapan *preprocessing*

Berdasarkan tahapan pada Gambar 3.2, berikut merupakan uraian penjelasan pada setiap tahapannya:

Cleaning

Tidak semua cuitan yang didapatkan merupakan teks yang terstruktur dan sesuai dengan kaidah penulisan yang benar. Bagian ini berfungsi untuk membersihkan cuitan dari URL, karakter NON-ASCII, angka, simbol, tanda baca, *hashtag*, *username*, dan *Re-Tweet*. Proses ini dieksekusi dengan bahasa pemrograman *Python* pada *Google Colab* yang hasilnya disimpan dalam file berformat *xlsx*. Contoh cuitan yang telah melalui proses *cleaning* dapat dilihat pada Tabel 3.2.

Tabel 3.2 Implementasi *cleaning* pada cuitan

Sebelum	Sesudah
@VIVAcoid Alhamdulillah terimakasih ya rob!.orang sombong tak layak jadi wakil rakyat.	alhamdulillah terimakasih iya rob orang sombong tak layak jadi wakil rakyat
@vngnst Nanti aku dikatain sombong	nanti aku dikatakan sombong
@xsehunsi Dih udah pagi 🙄 Kok sehun sombong sama kacy!	dih sudah pagi kok sehun sombong sama kacy
@tubirfess Mereka gak sadar lebih kampungan dari si mc yang mereka sendiri kata katin, bego.	mereka tidak sadar lebih kampungan dari si mc yang mereka sendiri kata katin bego
@viaslsm makan2 bareng temen2. tapi, kok ada yg masih kurang 🙄	makan bareng temen tapi kok ada yg masih kurang

Normalize

Cuitan pada media sosial *Twitter* biasanya ditulis menggunakan penulisan yang tidak sesuai dengan standar penulisan yang benar, mulai dari penyingkatan terhadap kata, menggunakan istilah-istilah kekinian, menggunakan Bahasa daerah dan menggunakan istilah-istilah asing. Pada bagian ini dilakukan normalisasi kata-kata dalam data cuitan yang penulisannya tidak dalam penulisan yang semestinya. Contoh *slangword* seperti ga cepet menjadi tidak, kata aq menjadi saya, dan lain-lain. Contoh penerapan tahap ini dapat dilihat pada Tabel 3.3.

Tabel 3.3 *Normalize slangword* pada cuitan

Sebelum	Sesudah
males followan sm rp sombong	males followan sama rp sombong
Cakep si tp sombong	Cakep si tapi sombong
cuyun jelek kalau ga chaeyon nunak	cuyun jelek kalau tidak chaeyon nunak
iya jdi kaya sombong gt ya kan	iya jadi kaya sombong begitu iya kan
gpp masih iq jongkok mas dari pada iq ndlosor	tidak apa-apa masih iq jongkok mas dari pada iq ndlosor
o yaampun udik bgt ak	o yaampun udik banget saya
iq nya ga kesemutan jongkok terus	iq nya tidak kesemutan jongkok terus
temen yg di ig kadang so ngartis	temen yang di instagram kadang sok ngartis
gue suka gaya elo om	ki suka gaya kamu om
heboh ndiri bantah ndiri klo gak tolol ya bukan cebong	heboh sendiri bantah sendiri kalau tidak tolol iya bukan cebong
loe idiot apa mmg dungu dr kecil ya gk paham jg klo org ngomong	kamu idiot apa memang dungu dari kecil iya tidak paham juga kalau orang bicara

Stemming

Tahap *Stemming* adalah tahap untuk mengubah kata dalam cuitan ke dalam bentuk kata dasarnya dengan menghapus imbuhan. Contoh penerapan *Stemming* dapat dilihat pada Tabel 3.4.

Tabel 3.4 *Stemming* pada cuitan

Sebelum	Sesudah
aku dikatain sombong	aku kata sombong
mereka tidak sadar kampungan mc yang mereka sendiri katain bego	mereka tidak sadar kampung mc yang mereka sendiri katain bego

Removing Stopwords

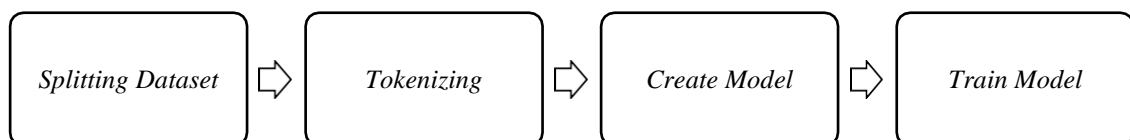
Setelah melalui proses *Cleaning*, teks akan diproses pada tahap *Removing Stopwords*. *Stopwords* berisi kumpulan kata-kata yang kurang penting dalam teks cuitan. Contoh *stopwords* seperti adapun, akankah, bukankah, sebagainya, dan lain-lain. Tahap ini berfungsi untuk menghapus kata-kata yang tidak memiliki makna. Pada tahap ini juga dilakukan proses pemecahan kata-kata pada kalimat cuitan menjadi bagian yang dinamakan token, yang berguna untuk pembobotan di tahap selanjutnya. Contoh penerapan tahap ini dapat dilihat pada Tabel 3.5.

Tabel 3.5 *Removing stopwords* pada cuitan

Sebelum	Sesudah
alhamdulillah terimakasih iya rob orang sombong tak layak jadi wakil rakyat	terimakasih orang sombong tak layak wakil rakyat
nanti aku dikatain sombong	aku dikatakan sombong
dih sudah pagi kok sehung sombong sama kacy	dih sudah sehung sombong kacy
mereka tidak sadar lebih kampungan dari si mc yang mereka sendiri kata katain bego	mereka tidak sadar kampungan mc yang mere ka sendiri katain bego
boro balas dm makan teman tidur saja kurang nanti iya dibales jangan ngatain sombong kura ng aktif kamu nya kalau tidak paham anw good morning	boro balas dm makan teman tidur saja dibalas jangan katakan sombong aktif kamu tidak pah am anw good morning

3.2.3 Klasifikasi

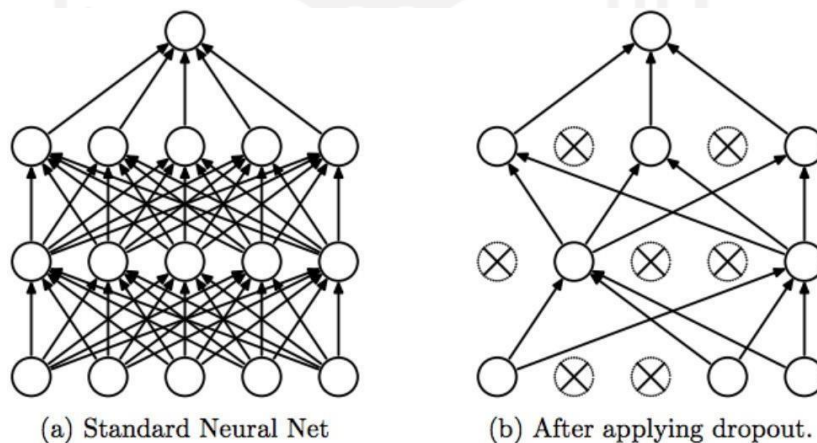
Langkah-langkah pada tahap klasifikasi terdiri dari *splitting dataset*, *tokenizing*, *create model*, dan *train model*. Gambar 3.3 menunjukkan tahapan dari langkah klasifikasi.



Gambar 3.3 Tahapan klasifikasi

Langkah yang pertama yaitu melakukan *splitting dataset* menjadi data *train*, data *test*, dan data *validation*. Data *train* adalah data yang digunakan untuk melatih model untuk belajar memahami data. Data *test* adalah data yang digunakan untuk menguji model untuk mendapat nilai yang berupa *accuracy*, *presicion*, *recall*, dan *f1-score* yang nantinya akan dievaluasi pada tahap selanjutnya. Data *validation* adalah data yang digunakan untuk memvalidasi hasil prediksi model pada saat melakukan proses pengujian.

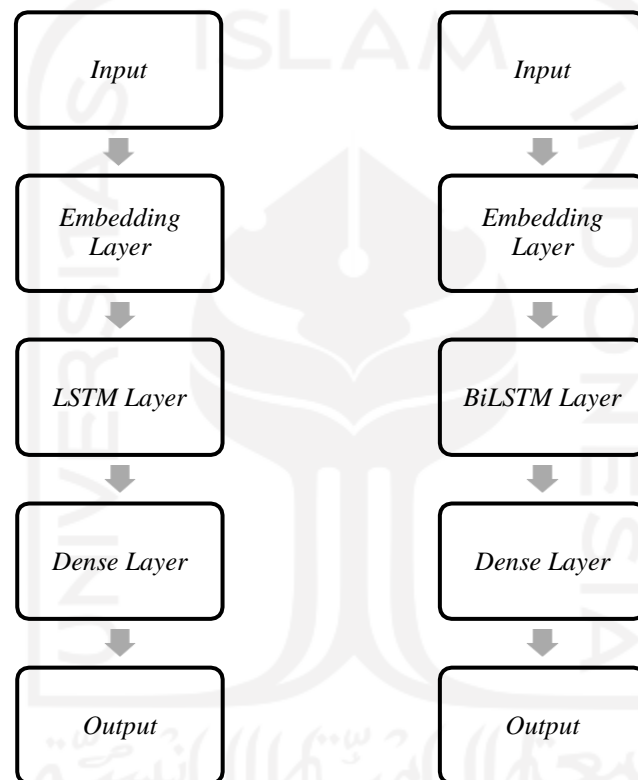
Lalu, langkah kedua yaitu melakukan *tokenizing* yang berfungsi mendapatkan kata-kata unik yang akan digunakan sebagai kamus bahasa atau *vocabulary*. Selanjutnya, langkah ketiga yaitu membuat model LSTM dan BiLSTM. Langkah terakhir yaitu melakukan pengujian model. Dalam melakukan proses pengujian model sering terjadi *overfitting* pada hasil dari model tersebut. *Overfitting* terjadi karena model yang dibuat terlalu fokus pada *training dataset* tertentu, hingga tidak bisa melakukan prediksi dengan tepat jika diberikan *dataset* lain yang serupa, maka dari itu perlu diterapkan metode regularisasi yakni *Dropout*. Teknik *dropout* mengatasi *overfitting* dengan cara menghapus sementara kontribusi terhadap aktivasi *neuron*. Gambar 3.4 menunjukkan bagaimana perbedaan ketika tidak menggunakan teknik *dropout* dan menggunakan teknik *dropout*.



Gambar 3.4 Perbedaan penggunaan *dropout*

Sumber: towardsdatascience.com

Proses klasifikasi dilakukan menggunakan metode LSTM dan BiLSTM. Terdapat 3 *layer* yang menjadi acuan dalam proses membangun model tersebut yaitu *embedding layer*, LSTM *layer* atau BiLSTM *layer*, dan *dense layer*. *Embedding layer* berfungsi untuk mengubah data latih menjadi vektor numerik yang merepresentasikan kedekatan makna tiap kata atau biasa dikenal dengan *word embedding*. LSTM *layer* berfungsi untuk menjalankan fungsi LSTM. BiLSTM *layer* berisi 2 LSTM *layer* yaitu *forward LSTM* dan *backward LSTM* sehingga gabungan tersebut akan menangkap informasi dari kedua arah. *Dense layer* berfungsi sebagai *output layer*. Proses kerja pada model dapat dilihat pada Gambar 3.5.



Gambar 3.5 Alur kerja model

Pada tahapan ini dibuat beberapa skenario untuk memperoleh model dengan nilai akurasi terbaik yang nantinya akan digunakan dalam melakukan deteksian konten *cyberbullying* pada cuitan berbahasa Indonesia. Daftar skenario model pada penelitian ini ditunjukkan pada Tabel 3.6 dan Tabel 3.7.

Tabel 3.6 Skenario model LSTM

No.	Embedding dimension	LSTM unit	Batch size	Epoch	Dropout
1	32	32	32	20	-
2	32	32	32	20	0.5

Tabel 3.7 Skenario model BiLSTM

No.	Embedding dimension	BiLSTM unit	Batch size	Epoch	Dropout
1	32	32	32	20	-
2	32	32	32	20	0.5

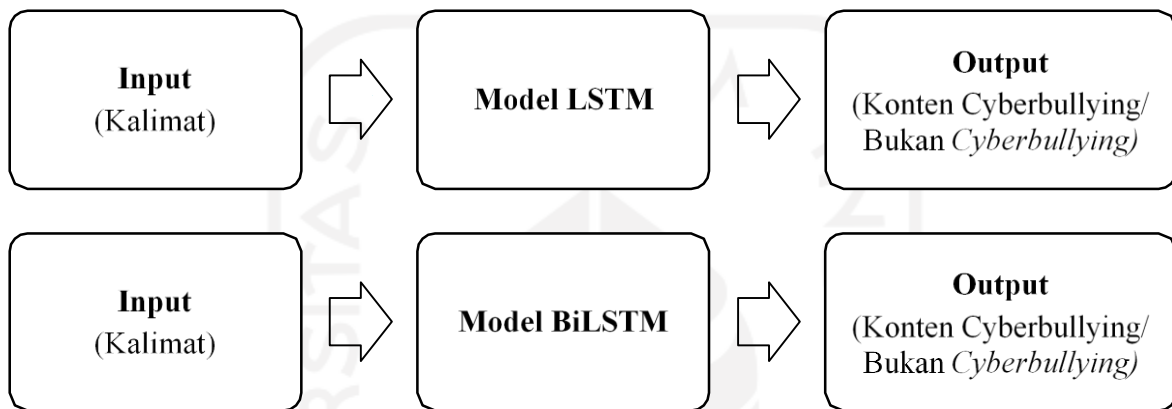
3.2.4 Evaluasi

Evaluasi merupakan tahapan yang bertujuan untuk mengukur kinerja dari model yang telah dibuat. Metode evaluasi yang digunakan dalam penelitian ini adalah *hold-out*. Metode ini membagi data ke dalam tiga bagian yakni data latih, data validasi, dan data uji (Reitermanova, 2010). Data latih digunakan untuk melatih model. Data validasi digunakan untuk evaluasi kinerja model selama proses pelatihan. Data uji digunakan sebagai evaluator performa dari model yang telah dibuat.

Pengujian lain dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* memuat informasi klasifikasi aktual dan prediksi dari sistem (Ron & Foster, 1998). *Confusion matrix* ditunjukkan pada Gambar 2.5. Berdasarkan nilai dari *confusion matrix* dapat diperoleh nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Score* dengan menggunakan persamaan (2.1), persamaan (2.2), persamaan (2.3), dan persamaan (2.4). Nilai *accuracy*, *precision*, *recall*, dan *f1-score* dapat dihitung sebagai nilai ukur evaluasi terhadap model. *Accuracy* adalah tingkat kedekatan antara nilai prediksi dengan nilai aktual. *Precision* merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *Recall* merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. *F1-Score* merupakan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. *F1-Score* berguna ketika memiliki jumlah distribusi data yang tidak seimbang.

3.2.5 Deteksi Konten

Deteksi konten merupakan tahapan akhir untuk mendeteksi kalimat apakah merupakan konten dewasa atau bukan. Tahapan ini menggunakan model dengan nilai akurasi terbaik dari skenario yang telah dibuat sebelumnya. Proses deteksi dapat dilihat seperti pada Gambar 3.6 menunjukkan proses deteksi konten.



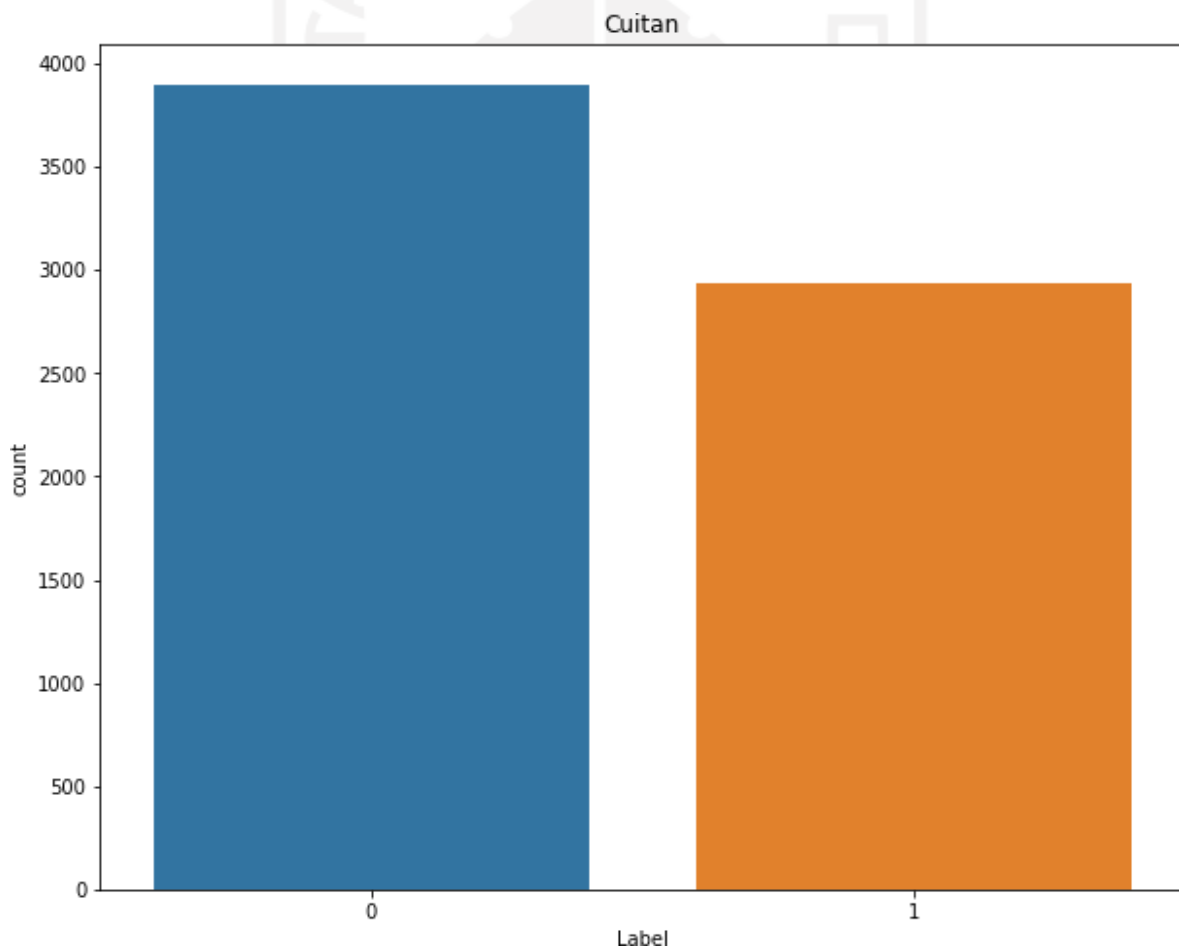
Gambar 3.6 Deteksi konten

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Twitter adalah tempat pengambilan data untuk penelitian ini. Langkah pengumpulan data dilakukan dengan memanfaatkan *Twitter* API. Sebelum menggunakan *Twitter* API, penulis harus mendaftarkan diri sebagai *developer* untuk mendapatkan izin akses berupa *Consumer Key*, *Consumer Secret*, *Access Token*, dan *Access Secret*. Data yang akan diambil dari *Twitter* berupa data cuitan berbahasa Indonesia. Cuitan yang terkumpul untuk membangun model pada penelitian ini adalah 6835 cuitan seperti yang terlihat pada Gambar 4.1 dengan cacah masing-masing yaitu 3900 cuitan *non-cyberbullying* dan 2935 cuitan *cyberbullying*.



Gambar 4.1 Cacah data

Jumlah awal *dataset* yang terkumpul sejumlah 7000 data akan tetapi, pada saat dilihat kembali ternyata banyak data yang berupa cuitan yang diposting berulang kali dalam waktu yang sama. Selanjutnya, penulis melakukan pembersihan data duplikat secara manual. Data yang digunakan pada penelitian ini diperoleh berdasarkan kata kunci pada *Twitter*. Untuk memudahkan memperoleh data yang tidak mengandung konten *cyberbullying*, Cuitan *non-cyberbullying* diambil menggunakan kata kunci yang positif. Sedangkan cuitan *cyberbullying* diambil dengan menggunakan kata kunci yang merupakan kata-kata yang bermakna perundungan. Akan tetapi, cuitan *cyberbullying* yang diambil pada penelitian ini belum memaknai secara penuh dari dasar pengertian *cyberbullying* itu sendiri. *Cyberbullying* yang dimaknai dari cuitan masih berupa *cyberbullying* yang ditujukan untuk mempermalukan dan menakut-nakuti.

4.2 *Preprocessing*

Tahap *preprocessing* memiliki tujuan untuk melakukan pembersihan data cuitan dari kata, simbol, dan hal-hal lain yang kurang bermakna. Tahap *preprocessing* yang dilakukan pada penelitian ini akan dijelaskan secara berurutan.

4.2.1 *Cleaning*

Tahap *cleaning* dilakukan untuk membersihkan data cuitan dari tanda baca, simbol dan sebagainya. Lebih jelas, langkah-langkah *cleaning* yang dilakukan adalah:

- a. Menghilangkan URL.
- b. Menghilangkan karakter NON-ASCII.
- c. Menghilangkan angka, simbol dan tanda baca.
- d. Menghilangkan *hashtag*, *username*, dan RT.
- e. Mengubah huruf ke dalam bentuk *lowercase*.

Adapun kode program yang digunakan untuk implementasi tahap ini dapat dilihat pada Gambar 4.2.

```

1. import re
2. import string
3. import unicodedata
4. import nltk
5.
6. def cleaning(str):
7.     #menghilangkan URL
8.     str = re.sub(r'(?i)\b((?:https?://|www\d{0,3}[.][a-z0-9.\-]+
9.         [a-z]{2,4}/)(?:[^\s()<>+
10.         |\\([^\s()<>+|\\([^\s()<>+\\)))*\\))+(?:\([^\s
11.         ()<>+|\\([^\s()<>+\\)))*\\)|[^\s`!()\\[\]{};:\'".,<>?«»“”‘’]))', '', str)
12.
13.     #menghapus karakter non-ASCII
14.     str = unicodedata.normalize('NFKD',str).encode('ascii', 'ignore').decode('utf8,
15.         'ignore')
16.
17.     #menghapus angka, simbol, tanda baca
18.     str = re.sub(r'^[a-z ]*(.[0-9])*d', '', str)
19.
20.     #menghapus karakter twitter # mention
21.     str = re.sub(r'(?:@[\w_]+)', ' ', str)
22.
23.     # hashtag
24.     str = re.sub(r"(?:\#+[\w_]+[\w\'_\-]*[\w_]+)", " ", str)
25.
26.     # RT/cc
27.     str = str = re.sub('RT', '', str)
28.
29.     #casefolding
30.     str = str.lower()
31.
32.     return str

```

Gambar 4.2 Kode tahapan *cleaning*

4.2.2 *Normalize*

Data cuitan yang didapat bukan merupakan kalimat-kalimat yang sesuai dengan penulisan yang benar. Seringkali ditemukan kata-kata ditulis dengan singkatan dan sebutan-sebutan kekinian sebagai contoh, kata ‘yang’ disingkat menjadi ‘yg’, kata ganti ‘aku’ diubah menjadi ‘ane’ dan masih banyak lagi. Pada tahapan ini, dilakukan proses penyesuaian kata sehingga menjadi kata dengan ejaan yang baku. Kode program yang diimplementasikan pada tahap ini seperti yang terlihat pada Gambar 4.3.

```

1. import pandas as pd
2. import re
3. import string
4. import nltk
5.
6. def normalize_slang_word(str):
7.     # melakukan proses normalize
8.     text_list = str.split(' ') slang_words_raw=pd.read_csv
9.     ('/content/drive/MyDrive/DATASET/slang_word_list.csv', sep=',', header=None)
10.    slang_word_dict = {}
11.
12.    for item in slang_words_raw.values:
13.        slang_word_dict[item[0]] = item[1]
14.
15.        for index in range(len(text_list)):
16.            if text_list[index] in slang_word_dict.keys():
17.                text_list[index] = slang_word_dict[text_list[index]]
18.
19.    return ' '.join(text_list)

```

Gambar 4.3 Kode tahapan *normalize*

4.2.3 Stemming

Tahap ini dilakukan untuk mengubah kata-kata pada data cuitan ke dalam bentuk kata dasarnya. Kode program yang digunakan seperti yang terlihat pada Gambar 4.4.

```

1. import re
2. import string
3. import unicodedata
4. import nltk
5. from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
6.
7. def stemming(str):
8.     # Melakukan proses stemming
9.     factory = StemmerFactory()
10.    stemmer = factory.create_stemmer() text = stemmer.stem(str)
11.    return text

```

Gambar 4.4 Kode tahapan *stemming*

4.2.4 Remove Stopwords

Pada bagian ini, kata-kata yang kurang bermakna dihilangkan untuk alasan efisiensi data. Daftar kata-kata yang kurang penting dibuat dalam sebuah *file* sebagai kamus dalam tahap ini. Selain menghapus kata yang kurang penting, proses ini juga melakukan pemecahan kalimat menjadi bagian-bagian penting yang dinamakan *token*. Kode program tahap ini dapat dilihat pada Gambar 4.5.


```

1. import re
2. import string
3. import unicodedata
4. import nltk
5. from nltk import word_tokenize, sent_tokenize from nltk.corpus
6. import stopwords
7.
8. def remove_stopword(str):
9.     # Melakukan proses stopwords
10.    stop_words = set(stopwords.words('/content/drive/MyDrive/DATASET/stopwordID.csv
11.    '))
12.    word_tokens = word_tokenize(str)
13.    filtered_sentence = [w for w in word_tokens if not w in stop_words]
14.    return ' '.join(filtered_sentence)

```

Gambar 4.5 Kode tahapan *remove stopwords*

Setelah semua data telah melalui proses *preprocessing* selanjutnya data cuitan sudah bisa diproses ke tahap selanjutnya. Tabel 4.1 merupakan contoh penerapan tahap *preprocessing* pada data cuitan.

Tabel 4.1 Contoh penerapan *preprocessing*

No	Sebelum	Sesudah
1	vivanewscom: Truk Hantam Kendaraan di Cianjur, Salah Satunya Rombongan Pengantin https://t.co/wjwvdPTRBa #vivanews"	truk hantam kendara cianjur salah satu rombongan pengantin
2	MUI Kalbar: Larangan Cadar-Celana Cingkrang KhawatirTimbulkan Gejala https://t.co/SOouK1vBIP	mui kalbar larang cadar celana cingkrang khawatir timbul gejala
3	@pengguna13 @pengguna00 Biasa gaya hidup sok...ibarat org udik baru melek liat metropolitan jd jumpalitan dan gayanya gak karu karuan....	biasa gaya hidup sok orang udik melek lihat metropolitan jadi jumpalitan dan gayanya tidak karu karuan

43 Klasifikasi

Klasifikasi merupakan tahapan yang dilakukan untuk memperoleh model terbaik untuk klasifikasi cuitan yang mengandung *cyberbullying* dan *non-cyberbullying*. Tahapan yang dilakukan dalam klasifikasi yaitu:

4.3.1 *Splitting Dataset*

Proses *splitting* data sebesar 6835 dibagi ke dalam data *train*, data *test*, dan data *validation*. Gambar 4.6 merupakan kode program untuk membagi antara data *train*, data *test*, dan data *validation*.

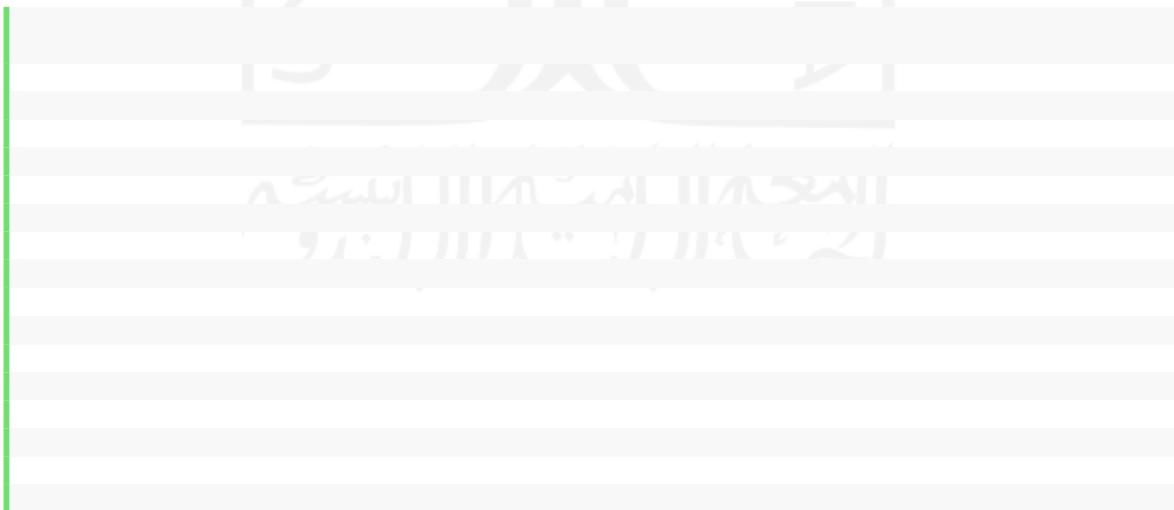


Gambar 4.6 *Splitting data train, data test, dan data validation*

4.3.2 *Tokenizing*

Tokenizing merupakan operasi untuk memisahkan teks menjadi potongan-potongan token sebelum dianalisis lebih lanjut. Proses ini bertujuan agar mendapatkan kata-kata unik yang akan digunakan sebagai kamus bahasa atau *vocabulary*.

Fungsi *fit_on_texts* bertujuan untuk membuat indeks pada suatu kata atau karakter berdasarkan frekuensi kemunculannya, semakin sering muncul maka indeksnya akan semakin kecil. Fungsi *text_to_sequences* bertujuan mengurutkan setiap kata berdasarkan urutan indeksnya. Fungsi *pad_sequences* bertujuan untuk menyamakan ukuran dimensi pada seluruh teks sesuai dengan nilai *maxlen*, *maxlen* merupakan panjang maksimum semua *sequences*. Gambar 4.7 merupakan kode program untuk *tokenizing* pada proses klasifikasi.



Gambar 4.7 *Tokenizing*

4.3.3 Create Model

Gambar 4.8 dan Gambar 4.9 menunjukkan kode program model LSTM dan BiLSTM. Tipe model *sequential* menunjukkan bahwa model dibuat berdasarkan *layer-by-layer*, fungsi *add* digunakan untuk menambahkan *layer* pada model yang sedang dibangun. Model dengan tipe ini memungkinkan untuk membangun sebuah model dengan tumpukan *layer* secara berurutan. Hal ini menunjukkan bahwa data masuk dari satu *layer* ke *layer* lainnya sesuai dengan urutan *layer*.

Kasus untuk klasifikasi dua kelas kategori model yang dibangun menggunakan fungsi aktivasi *sigmoid* dan fungsi *loss categorical_crossentropy*. Langkah berikutnya setelah konfigurasi *layer* selesai yaitu *compile*, *compile* bertujuan agar konfigurasi *layer* yang telah dibuat dapat digunakan untuk proses *training*. Pada proses *compile* parameter yang digunakan yaitu *loss*, *optimizer* dan *metrics*. Fungsi *loss* bertujuan untuk menghitung kuantitas yang harus diminimalkan oleh model selama pelatihan. Fungsi *optimizer* bertujuan untuk mengatur respon model untuk estimasi error setiap kali bobot model diperbarui. Fungsi *metric* bertujuan untuk menilai kinerja dari model. Untuk kasus *text classification* fungsi *optimizer* yang digunakan adalah *adam*. Fungsi *metrics* menggunakan *accuracy*, maka kinerja model dinilai berdasarkan akurasi.

```

1. # Membuat model LSTM
2. def get_LSTM():
3.     model_LSTM = Sequential()
4.     model_LSTM.add(Embedding(input_dim = vocab, output_dim = 32,
5.     input_length = maxlen, embeddings_initializer = initializer))
6.     model_LSTM.add(LSTM(32, recurrent_initializer = initializer,
7.     kernel_initializer = initializer))
8.     model_LSTM.add(Dense(3, activation='sigmoid', kernel_initializer =
9.     initializer))
10.    model_LSTM.compile(loss='CategoricalCrossentropy', optimizer='adam',
11.    metrics=['accuracy'])
12.    print(model_LSTM.summary())
13.    return model_LSTM
14. # Membuat model LSTM dengan dropout
15. def get_LSTMWITHDROPOUT():
16.    model_LSTM = Sequential()
17.    model_LSTM.add(Embedding(input_dim = vocab, output_dim = 32,
18.    input_length = maxlen, embeddings_initializer = initializer))
19.    model_LSTM.add(Dropout(0.5))
20.    model_LSTM.add(LSTM(32, recurrent_initializer = initializer,
21.    kernel_initializer = initializer))
22.    model_LSTM.add(Dense(3, activation='sigmoid', kernel_initializer =
23.    initializer))
24.    model_LSTM.compile(loss='CategoricalCrossentropy', optimizer='adam',
25.    metrics=['accuracy'])
26.    print(model_LSTM.summary())
27.    return model_LSTM

```

Gambar 4.8 Model LSTM

```

1. # Membuat model BiLSTM
2. def get_BiLSTM():
3.     model_BiLSTM = Sequential()
4.     model_BiLSTM.add(Embedding(input_dim = vocab, output_dim = 32,
5.     input_length = maxlen, embeddings_initializer = initializer))
6.     model_BiLSTM.add(Bidirectional(LSTM(32, recurrent_initializer =
7.     initializer, kernel_initializer = initializer)))
8.     model_BiLSTM.add(Dense(3, activation='sigmoid', kernel_initializer =
9.     initializer))
10.    model_BiLSTM.compile(loss='CategoricalCrossentropy', optimizer='adam',
11.    metrics=['accuracy'])
12.    print(model_BiLSTM.summary())
13.    return model_BiLSTM
14. # Membuat model BiLSTM dengan dropout
15. def get_BiLSTMWITHDROPOUT():
16.    model_BiLSTM = Sequential()
17.    model_BiLSTM.add(Embedding(input_dim = vocab, output_dim = 32,
18.    input_length = maxlen, embeddings_initializer = initializer))
19.    model_BiLSTM.add(Dropout(0.5))
20.    model_BiLSTM.add(Bidirectional(LSTM(32, recurrent_initializer =
21.    initializer, kernel_initializer = initializer)))
22.    model_BiLSTM.add(Dense(3, activation='sigmoid', kernel_initializer =
23.    initializer))
24.    model_BiLSTM.compile(loss='CategoricalCrossentropy', optimizer='adam',
25.    metrics=['accuracy'])
26.    print(model_BiLSTM.summary())
27.    return model_BiLSTM

```

Gambar 4.9 Model BiLSTM

4.3.4 Training Model

Training atau menguji model LSTM dan BiLSTM dilakukan dengan menggunakan fungsi *fit*. Parameter yang digunakan adalah *x* untuk *input* data, *y* untuk target data, *batch_size* untuk merupakan jumlah sampel yang akan dilatih setiap *epoch*, *epoch* merupakan jumlah untuk melatih model, *validation_data* berfungsi untuk digunakan sebagai data validasi, *verbose* berfungsi untuk menampilkan *progress bar* dari setiap *epoch*. Gambar 4.10 dan Gambar 4.11 menunjukkan kode program untuk *training* model LSTM dan BiLSTM.

```

1. # Melakukan pengujian model LSTM
2. history1 = LSTM.fit(X_train, Y_train, batch_size=batch_size, epochs=num_epochs,
3.     verbose=1, validation_data = (X_val, Y_val))
4. # Melakukan pengujian model LSTM dengan dropout
5. history2 = LSTMWITHDROPOUT.fit(X_train, Y_train, batch_size=batch_size,
6.     epochs=num_epochs, verbose=1, validation_data=(X_val, Y_val))

```

Gambar 4.10 Training LSTM

Gambar 4.11 Training BiLSTM

44 Evaluasi

Tahap evaluasi bertujuan untuk mengukur kinerja dari model LSTM dan BiLSTM yang dibuat. Data *test* yang digunakan untuk menguji kinerja model sebanyak 1367 data atau 20% dari keseluruhan total data. Hasil evaluasi adalah sebagai berikut:

4.4.1 Skenario Model

Semua model akan menggunakan *embedding layer* dengan ukuran 32 dimensi. Semua layer dengan 32 *units* dan sebuah *dense layer* di mana fungsi aktivasinya menggunakan *Sigmoid* dan fungsi optimasinya menggunakan *Adam*. Semua model akan dijalankan menggunakan 20 *epoch* dan 32 *batch size*. Hasil dari model yang ditunjukkan pada Tabel 3.6 dan Tabel 3.7 akan dibandingkan satu sama lain untuk melihat model manakah yang terbaik dalam klasifikasi dengan melihat beberapa parameter yaitu *accuracy*, *precision*, *recall*, dan *f1-score*. Berdasarkan hasil cacah dataset menunjukkan ketidakseimbangan yaitu 43% banding 57% sehingga perlukan beberapa parameter di atas untuk melihat model manakah yang memiliki performa terbaik.

4.4.2 Analisis Skenario

Langkah setelah membuat membuat skenario dan menguji model LSTM dan BiLSTM selanjutnya didapatkan hasil berupa *confusion matrix* di mana *confusion matrix* model LSTM dapat dilihat pada Tabel 4.2 dan Tabel 4.3. Sedangkan *confusion matrix* model BiLSTM dapat dilihat pada Tabel 4.4 dan Tabel 4.5.

Tabel 4.2 *Confusion matrix* LSTM

LSTM			
<i>Actual Values</i>	<i>Predicted Value</i>		
		0	1
	0	743	54
	1	46	524

Tabel 4.3 *Confusion matrix* LSTM+Dropout

LSTM + Dropout			
<i>Actual Values</i>	<i>Predicted Value</i>		
		0	1
	0	743	54
	1	46	524

Tabel 4.4 *Confusion matrix* BiLSTM

BiLSTM			
<i>Actual Values</i>	<i>Predicted Value</i>		
		0	1
	0	772	25
	1	56	514

Tabel 4.5 *Confusion matrix* BiLSTM+Dropout

BiLSTM + Dropout			
<i>Actual Values</i>	<i>Predicted Value</i>		
		0	1
	0	771	26
	1	50	520

Setelah mendapat *confusion matrix* maka nilai berupa *accuracy*, *precision*, *recall*, dan *f1-score* dapat diketahui. Tabel 4.6 merupakan hasil evaluasi model LSTM dan Tabel 4.7 merupakan hasil evaluasi model BiLSTM.

Tabel 4.6 *Accuracy, Precision, Recall, dan F1-Score* LSTM

Metode	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
LSTM	93.77	91.59	92.45	92.02
LSTM + Dropout	94.87	94.23	92.45	93.33

Tabel 4.7 *Accuracy, Precision, Recall, dan F1-Score* BiLSTM

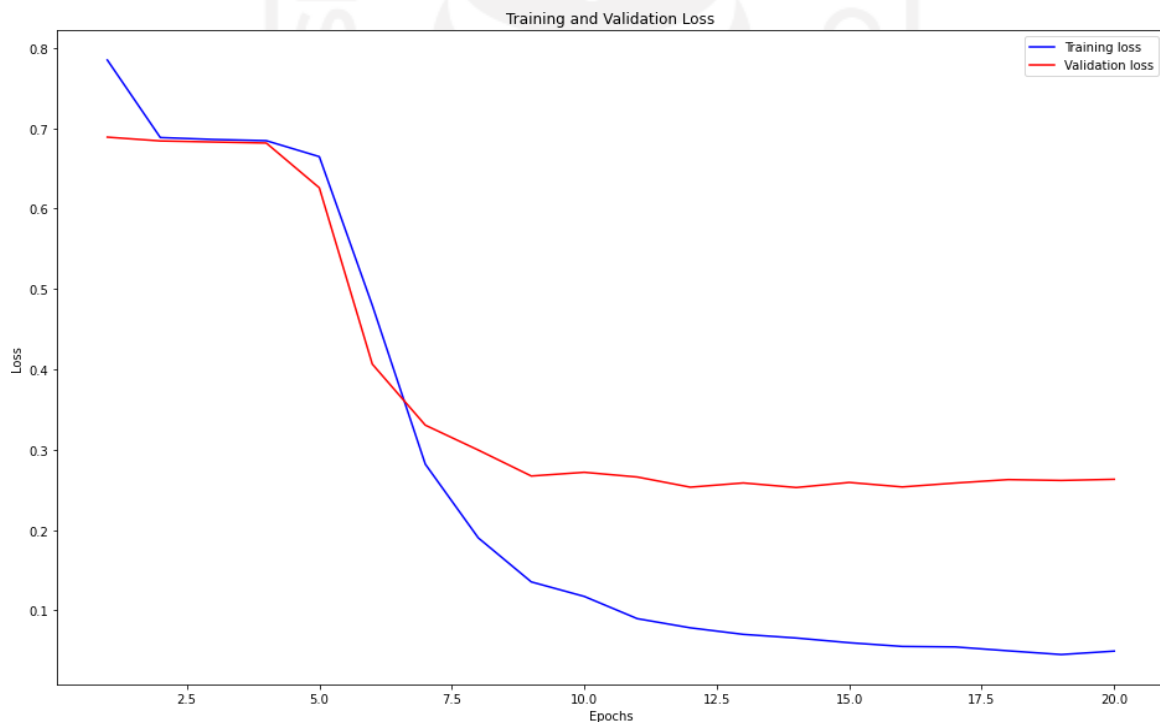
Metode	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
BiLSTM	94.51	95.05	90.57	92.75
BiLSTM + Dropout	95.24	94.29	93.40	93.84

Analisis berdasarkan Tabel 4.6 menunjukkan bahwa model LSTM dan model LSTM dengan *dropout* terbukti memiliki hasil yang tidak terpaut jauh di mana hasil *accuracy* yang didapat model LSTM sebesar 93.77 dan model LSTM dengan *dropout* sebesar 94.87. Lalu, untuk hasil *precision* terpaut lumayan jauh yaitu model LSTM sebesar 91.59 dan model LSTM dengan *dropout* sebesar 94.23. Selanjutnya, untuk hasil *recall* model LSTM dan model LSTM dengan *dropout* memiliki hasil yang sama. Terakhir, untuk hasil *f1-score* model LSTM mendapat hasil sebesar 92.02 dan model LSTM dengan *dropout* mendapat hasil sebesar 93.33.

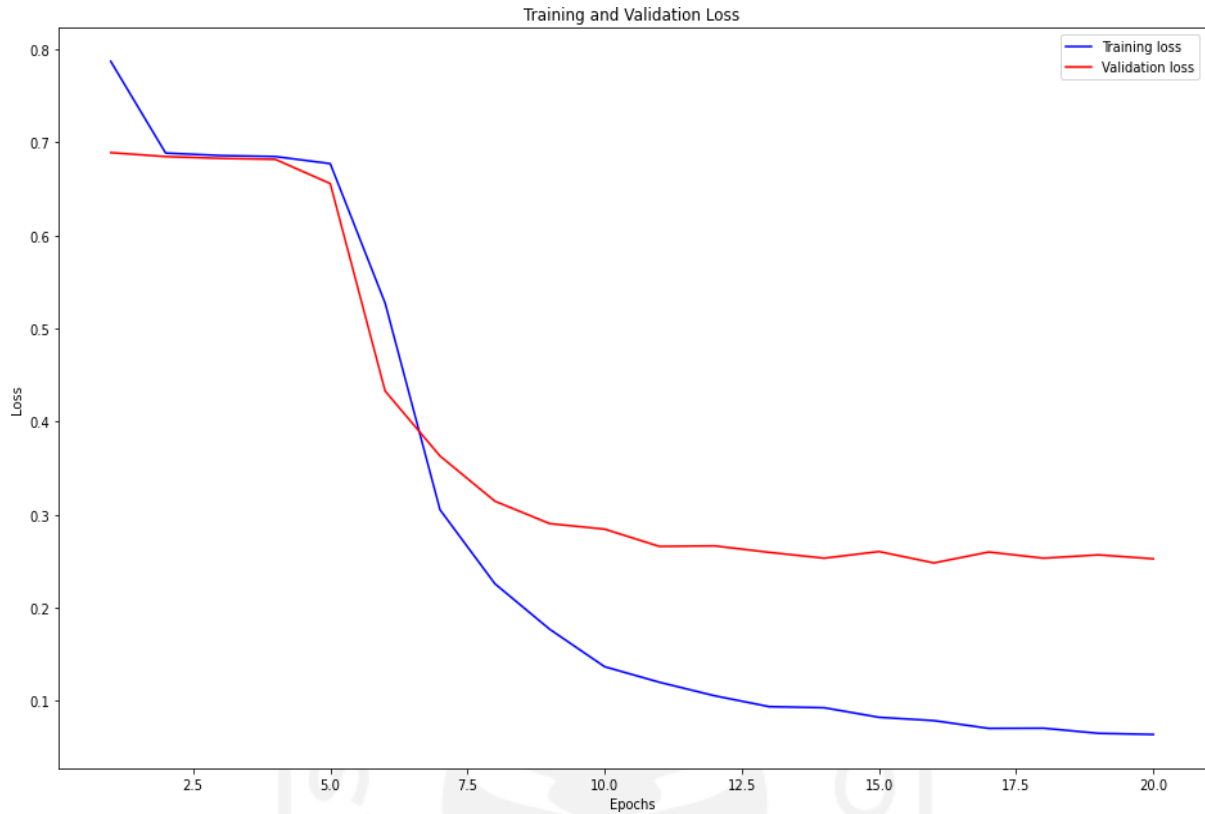
Analisis selanjutnya berdasarkan Tabel 4.7 menunjukkan bahwa model BiLSTM dan model BiLSTM terbukti memiliki hasil yang tidak terpaut juga di mana hasil *accuracy* yang didapat model BiLSTM sebesar 94.51 dan model BiLSTM dengan *dropout* sebesar 95.24. Lalu, untuk hasil *precision* model BiLSTM memiliki hasil yang lebih baik daripada model BiLSTM dengan *dropout* di mana hasil dari model BiLSTM sebesar 95.05 dan model BiLSTM dengan *dropout* sebesar 94.29. Selanjutnya, untuk hasil *recall* model BiLSTM dan model BiLSTM dengan *dropout* memiliki hasil yang terpaut lumayan jauh di mana hasil yang didapat model BiLSTM sebesar 90.57 dan model BiLSTM dengan *dropout* sebesar 93.40. Terakhir, untuk hasil *f1-score* model LSTM mendapat hasil sebesar 92.75 dan model LSTM dengan *dropout* mendapat hasil sebesar 93.84.

Berdasarkan analisis Tabel 4.6 dan Tabel 4.7 didapatkan bahwa dari kedua model yang digunakan pada penelitian ini terbukti bahwa model BiLSTM dengan *dropout* mendapat hasil yang terbaik dari skenario yang dibuat dengan hasil *accuracy*, *precision*, *recall*, dan *f1-score* sebesar 95.24; 94.29; 93.40; 93.84. Dengan demikian, setelah didapat model dengan hasil terbaik dari skenario yang dibuat, maka dapat dilakukan langkah deteksi kalimat apakah termasuk *cyberbullying* atau bukan *cyberbullying* pada langkah deteksi konten.

Di samping itu, berdasarkan nilai-nilai tersebut maka dapat diketahui grafik model tersebut. Beberapa model diterapkan *dropout* agar dapat mengurangi *overfitting* ketika pelatihan data. Berdasarkan Tabel 4.6, model LSTM yang menggunakan *dropout* merupakan model yang memiliki akurasi tertinggi dibandingkan LSTM. Model LSTM memiliki *loss* 0.315, sedangkan model LSTM dengan *dropout* 0.5 memiliki *loss* 0.285. Gambar 4.12 dan Gambar 4.13 merupakan grafik dari hasil evaluasi dari model LSTM dan LSTM yang menggunakan *dropout*.

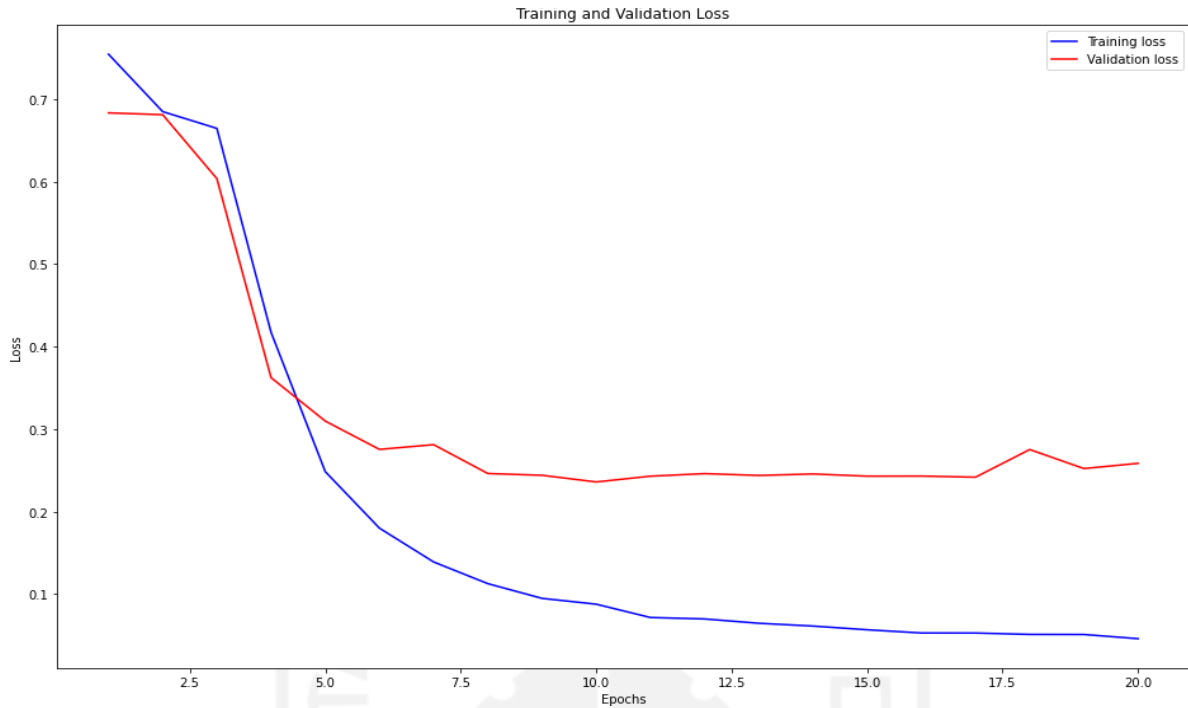


Gambar 4.12 Grafik LSTM

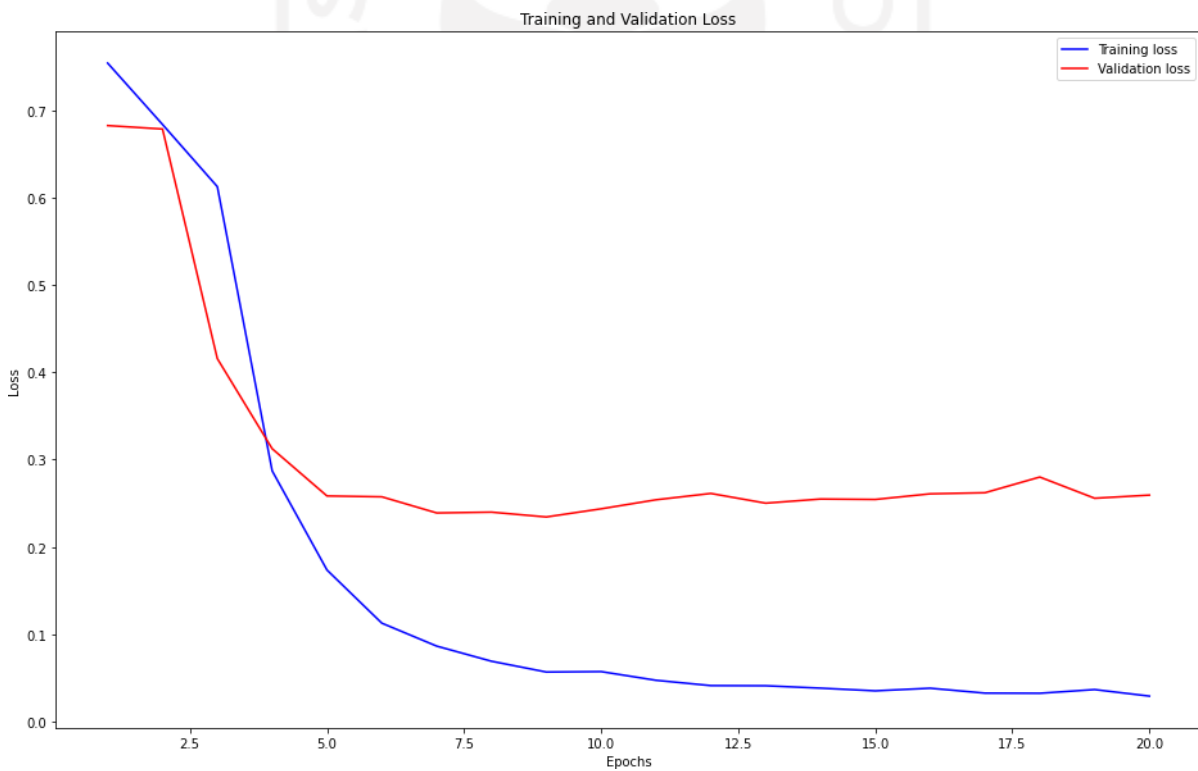


Gambar 4.13 Grafik LSTM+*dropout*

Model BiLSTM yang menggunakan *dropout* merupakan model yang memiliki akurasi tertinggi dibandingkan BiLSTM. Model BiLSTM memiliki *loss* 0.295, sedangkan model BiLSTM dengan *dropout* 0.5 memiliki *loss* 0.27. Gambar 4.14 dan Gambar 4.15 merupakan grafik dari hasil evaluasi dari model BiLSTM dan model BiLSTM yang menggunakan *dropout*.

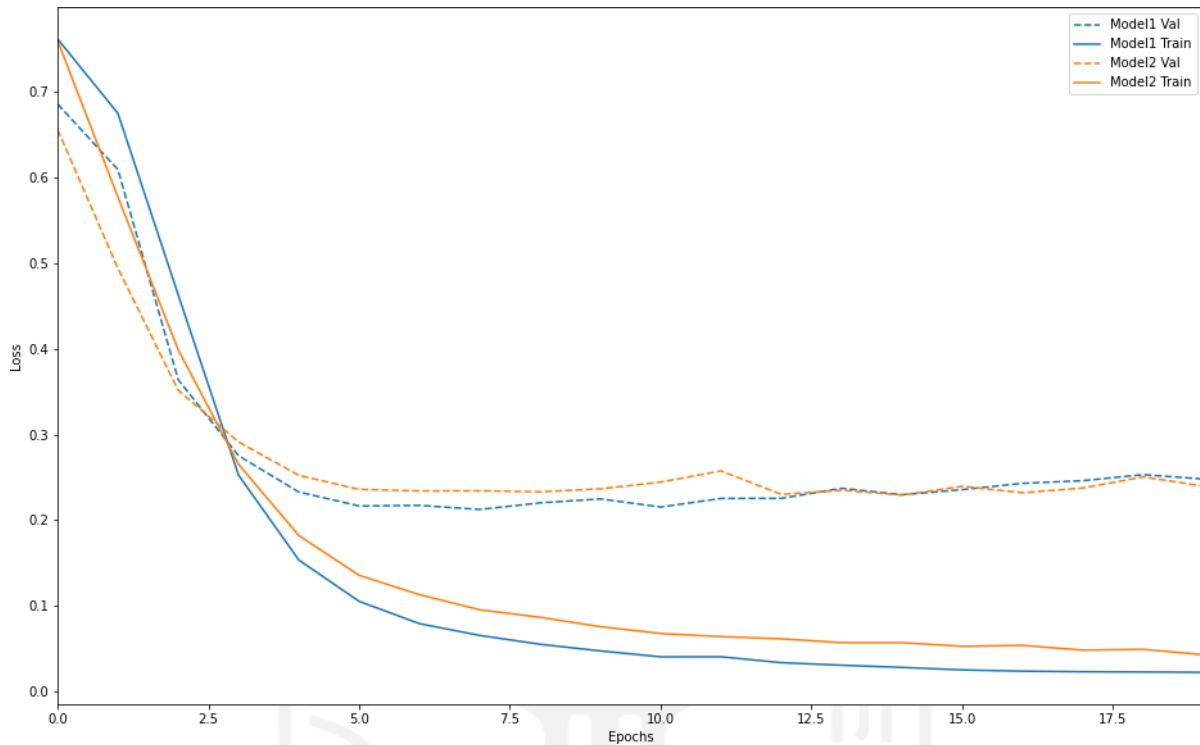


Gambar 4.14 Grafik BiLSTM

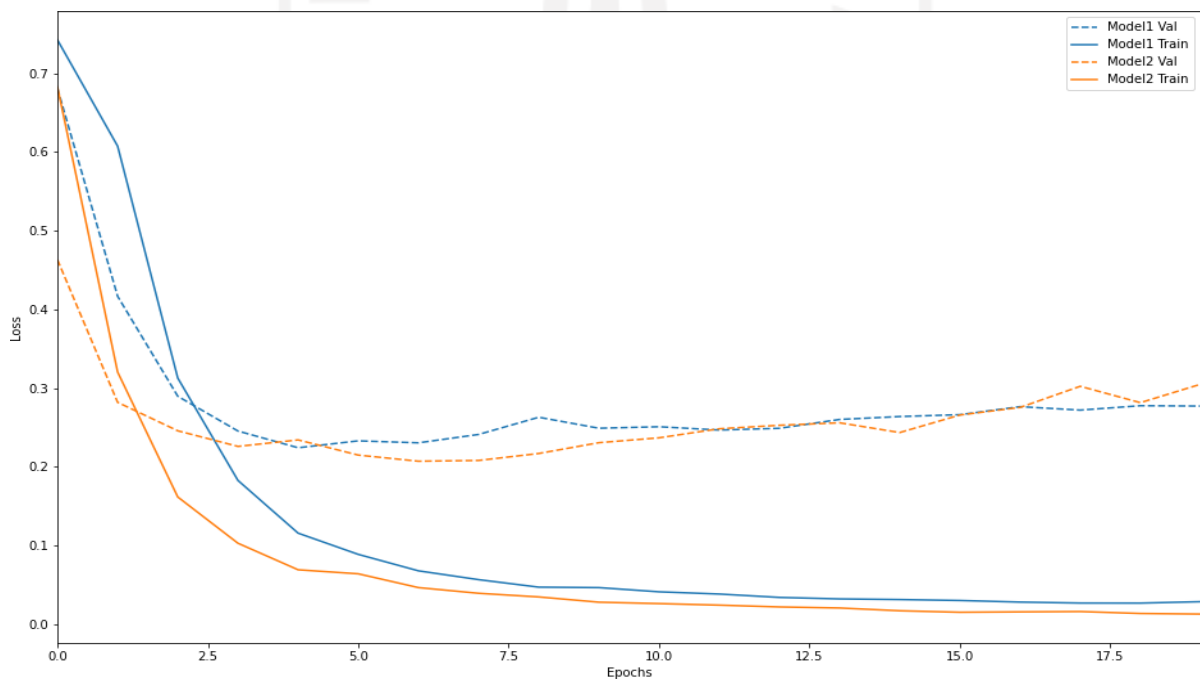


Gambar 4.15 Grafik BiLSTM+dropout

Berdasarkan Gambar 4.16 dan Gambar 4.17 dapat diketahui bahwa model yang menggunakan *dropout* memiliki nilai *loss* yang cukup rendah dan cukup mengatasi *overfitting*. Gambar 4.16 merupakan grafik gabungan dari model LSTM dan Gambar 4.17 merupakan grafik gabungan dari model BiLSTM.



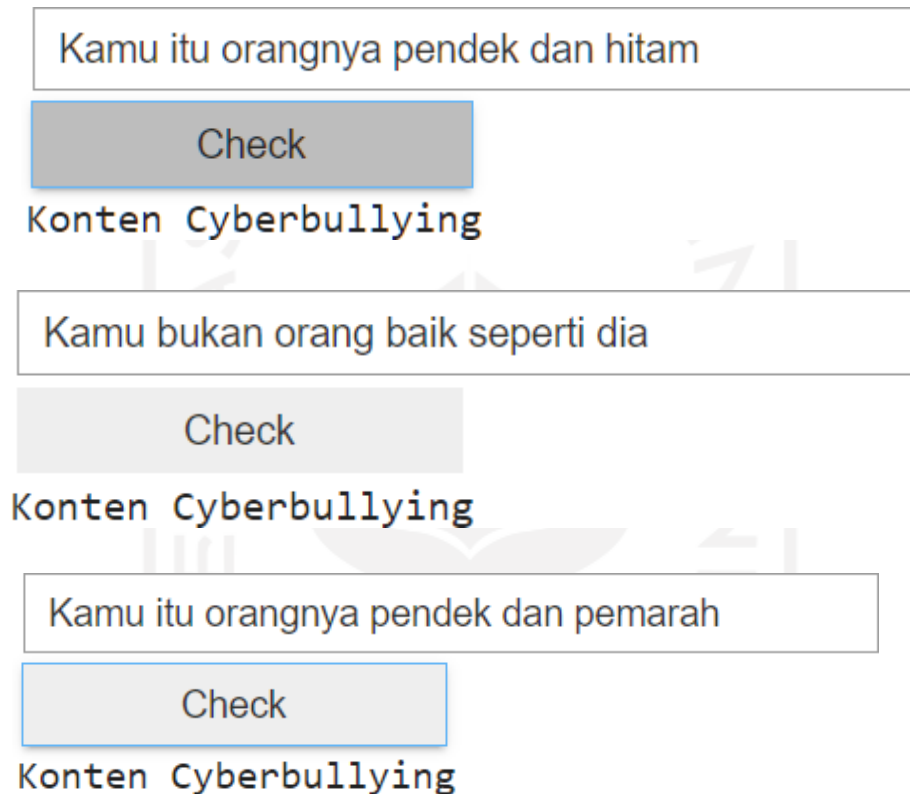
Gambar 4.16 Grafik gabungan LSTM



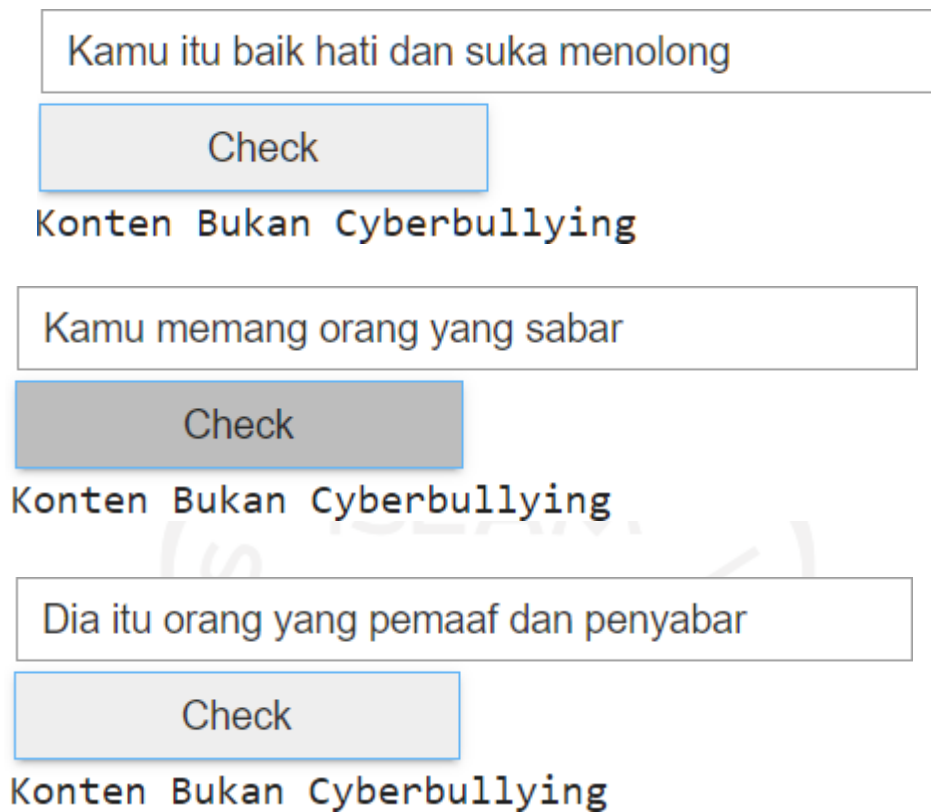
Gambar 4.17 Grafik gabungan BiLSTM

45 Deteksi Konten

BiLSTM merupakan model dengan akurasi terbaik dibandingkan dengan LSTM maka langkah selanjutnya adalah menguji kalimat apakah termasuk konten *cyberbullying* atau bukan *cybebullying*. Gambar 4.18 dan Gambar 4.19 merupakan hasil klasifikasi kalimat menggunakan model BiLSTM.



Gambar 4.18 Contoh hasil klasifikasi kalimat *cyberbullying*



Gambar 4.19 Contoh hasil klasifikasi kalimat *non-cyberbullying*

Berdasarkan hasil deteksi kalimat di atas semua kalimat yang diuji sudah dapat diprediksi secara tepat. Akan tetapi, penelitian ini masih ditemukan kekurangan yaitu beberapa kalimat yang dimaknai tidak semestinya. Hal tersebut terjadi karena dalam satu kalimat tersebut terdapat kata yang mengandung unsur *cyberbullying* yang jumlahnya lebih banyak daripada kata yang mengandung unsur bukan *cyberbullying*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian identifikasi cuitan pada media sosial Twitter menggunakan LSTM dan BiLSTM, dapat disimpulkan bahwa:

- a. Model LSTM dan BiLSTM terbukti dapat melakukan klasifikasi untuk mendeteksi cuitan yang mengandung *cyberbullying*. Dalam melakukan klasifikasi untuk mendeteksi cuitan yang mengandung *cyberbullying* langkah yang perlu dilakukan yaitu pengumpulan data, *preprocessing*, klasifikasi, evaluasi, dan deteksi konten.
- b. Model BiLSTM memiliki performa yang lebih baik dalam melakukan klasifikasi untuk mendeteksi cuitan yang mengandung *cyberbullying* daripada model LSTM. Hasil yang didapat model BiLSTM dengan nilai *accuracy* sebesar 94.51; 95.24 dan nilai *F1-Score* sebesar 92.75; 93.84 tidak memiliki perbedaan yang terlalu jauh terhadap hasil yang didapat oleh model LSTM dengan nilai *accuracy* sebesar 93.77; 94.87 dan nilai *F1-Score* sebesar 92.02; 93.33.
- c. Penelitian ini menunjukkan hasil yang cukup baik, dibuktikan dengan ketepatan prediksi makna kalimat yang sebagian besar menunjukkan ketepatan dalam memaknai kalimat tersebut.
- d. Kelebihan dari penelitian ini adalah performa model dan ketepatan prediksi dalam melakukan deteksi kalimat yang cukup baik.
- e. Kekurangan dari penelitian ini adalah masih ditemukan beberapa kalimat yang bermakna *non-cyberbullying* dideteksi sebagai *cyberbullying*, dan juga sebaliknya. Hal tersebut terjadi karena dalam satu kalimat tersebut terdapat kata yang mengandung unsur *cyberbullying* yang jumlahnya lebih banyak daripada kata yang mengandung unsur bukan *cyberbullying*.

52 Saran

Berdasarkan penelitian yang dilakukan, penulis berharap penelitian ini bisa dikembangkan oleh peneliti-peneliti yang akan datang dengan beberapa saran dari penulis seperti:

- a. Menambah jumlah dataset dengan kata kunci yang lebih beragam.
- b. Melakukan penyeimbangan terhadap jumlah baris data untuk tiap-tiap kelas.
- c. Membuat daftar *stopword* dan *slang word* yang lebih kompleks.
- d. Mencoba menggunakan metode lain pada model klasifikasi.
- e. Hasil dari klasifikasi kelas *cyberbullying* bisa diekstrak menjadi kelas yang lebih spesifik lagi seperti *cyberbullying* berupa *bully*, *racism*, *sexism*, dan *attack*.
- f. Menambah algoritma untuk dibandingkan agar mendapat algoritma mana yang lebih baik.



DAFTAR PUSTAKA

- Abdulloh, N., & Hidayatullah, A. F. (2019). Deteksi Cyberbullying pada Cuitan Media Sosial Twitter. *Automata, Vol 1*(1), 1–5.
- Agrawal, S., & Awekar, A. (2018). Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms. *European Conference on Information Retrieval*, 141–153.
- Akbar, M. A., & Utari, P. (2014). CYBERBULLYING PADA MEDIA SOSIAL. *Jurnal Komunikasi Massa*, 3(2), 1–20.
- Balakrishnan, V., Khan, S., Fernandez, T., & Arabnia, H. R. (2019). Cyberbullying detection on twitter using Big Five and Dark Triad features. *Personality and Individual Differences*, 141, 252–257. <https://doi.org/10.1016/j.paid.2019.01.024>
- Clinton, B. (2019). *Pengguna Aktif Harian Twitter Indonesia Diklaim Terbanyak*. Kompas.Com. <https://tekno.kompas.com/read/2019/10/30/16062477/pengguna-aktif-harian-twitter-indonesia-diklaim-terbanyak>
- Dai, J., & Liu, X. (2014). Approach for text classification based on the similarity measurement between normal cloud models. *The Scientific World Journal*. <https://doi.org/10.1155/2014/784392>
- Hidayatullah, A. F., Yusuf, A. A., Juwairi, K. P., & Nayoan, R. A. (2019). Identifikasi Konten Kasar pada Tweet Bahasa Indonesia. *Jurnal Linguistik Komputasional (JLK)*, 2(1), 1–5. <https://doi.org/10.26418/jlk.v2i1.15>
- Hinduja, S., & Patchin, J. W. (2010). Bullying, cyberbullying, and suicide. *Archives of Suicide Research*, 14(3), 206–221. <https://doi.org/10.1080/13811118.2010.494133>
- Hosseinmardi, H., Mattson, S. A., Rafiq, R. I., Han, R., Lv, Q., & Mishra, S. (2015). Detection of Cyberbullying Incidents on the Instagram Social Network Homa. *ArXiv Preprint ArXiv:1503.03909*. https://doi.org/10.1007/978-3-319-27433-1_4
- Kominfo. (2013). *Kominfo : Pengguna Internet di Indonesia 63 Juta Orang*. Kominfo.Go.Id. https://kominfo.go.id/index.php/content/detail/3415/Kominfo+%3A+Pengguna+Internet+di+Indonesia+63+Juta+Orang/0/berita_satker
- Laksana, B. A. (2017). *Mensos: 84% Anak Usia 12-17 Tahun Mengalami Bullying*. News.Detik.Com. <https://news.detik.com/berita/d-3568407/mensos-84-anak-usia-12-17-tahun-mengalami-bullying>
- Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-

- CRF. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, 1. <https://doi.org/10.18653/v1/p16-1101>
- Mujilahwati, S. (2016). Pre-Processing Text Mining Pada Data Twitter. *Seminar Nasional Teknologi Informasi Dan Komunikasi*, 49–56.
- Rahayu, F. S. (2012). CYBERBULLYING SEBAGAI DAMPAK NEGATIF PENGGUNAAN TEKNOLOGI INFORMASI. *Journal of Information Systems*, 8(43), 22–31.
- Redaksi WE Online. (2020). *Kacau, Konten Porno Paling Banyak Berasal dari Twitter! Kemenkominfo Ambil Langkah Apa Nih?* Wartaekonomi.Co.Id. <https://www.wartaekonomi.co.id/read271389/kacau-konten-porno-paling-banyak-berasal-dari-twitter-kemenkominfo-ambil-langkah-apa-nih>
- Ron, K., & Foster, P. (1998). Special issue on applications of machine learning and the knowledge discovery process. *Journal of Machine Learning*, 30, 271–274.
- Soyusiawaty, D., & Haspiyan, R. (2015). APLIKASI KAMUS BAHASA INDONESIA – BAHASA SASAK BERBASIS WAP. *Seminar Nasional Informatika (SEMNASIF)*, 1(5), 40–47.

LAMPIRAN

