KLASIFIKASI PELANGGARAN UNDANG-UNDANG ITE MENGGUNAKAN LSTM DAN BILSTM

Disusun Oleh:

N a m a	: Akmal Perdana Hesaputra
NIM	: 19523196

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2023

HALAMAN PENGESAHAN DOSEN PEMBIMBING

KLASIFIKASI PELANGGARAN UNDANG-UNDANG ITE MENGGUNAKAN LSTM DAN BILSTM

TUGAS AKHIR

Disusun Oleh:

N a m a : Akmal Perdana H.

NIM : 19523196

Yogyakarta, 5 Juli 2023

Pembimbing,

(Dhomas Hatta Fudholi, S.T, M.Eng, Ph.D)

HALAMAN PENGESAHAN DOSEN PENGUJI

KLASIFIKASI PELANGGARAN UNDANG-UNDANG ITEMENGGUNAKAN LSTM DAN BILSTM

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana

di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 5 Juli 2023

Tim Penguji

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

Anggota 1

Rahadian Kurniawan, S.Kom., M.Kom. _____

Anggota 2

Sri Mulyati, S.Kom., M.Kom. _____

Mengetahui,

Ketua Program Studi Informatika – Program SarjanaFakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T, M.Eng, Ph.D)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Akmal Perdana H.

NIM : 19523196

Tugas akhir dengan judul:

KLASIFIKASI PELANGGARAN UNDANG-UNDANG ITE 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, 5 Juli 2023

(Akmal Perdana Hesaputra)

HALAMAN PERSEMBAHAN

السَّلَامُ عَلَيْكُمْ وَرَحْمَةُ اللهِ وَبَرَكَاتُهُ

Alhamdulillahirobbil'alamin, Sesungguhnya segala puji hanya milik Allah ﷻ, kami memuji-Nya, memohon pertolongan dan ampunan kepada-Nya, kami berlindung kepada Allah dari kejahatan diri-diri kami dan kejelekan amal perbuatan kami, barangsiapa yang Allah ﷻ beri petunjuk, maka tidak ada yang dapat menyesatkannya, dan barangsiapa yang Allah ﷻ sesatkan, maka tidak ada yang dapat memberinya petunjuk. Shalawat beriring salam semoga senantiasa tercurah kepada manusia paling mulia Rasullullah Muhammad ﷺ dan semoga kelak kita termasuk diantara umatnya yang akan berbahagia mendapatkan syafa'atnya di akhirat. *Aamiin ya robbal'alamin*.

Pada kesempatan ini penulis senantiasa bersyukur kepada Allah ﷻ karena hanya dengan pertolongan-Nya penulis dapat menyelesaikan tugas akhir dan studi ini. Kemudian penulis ingin mempersembahkan skripsi ini kepada:

Kedua orang tua ku tersayang Zulheriaty (mama) dan Nur Saptono (bapak) yang senantiasa mendorong, mensupport serta mendukung, sehingga penulis dapat menyelesaikan studi ini.

Dosen pembimbing tugas akhir bapak Dhomas Hatta Fudholi, S.T, M.Eng, Ph.D yang telah sabar membimbing dalam menyelesaikan tugas akhir ini.

Semua keluarga Riau, Magelang, dan juga Jogja serta adik kandung penulis yang telah mensupport serta memberi motivasi dari awal hingga akhir studi.

Semua teman-teman yang tidak dapat penulis sebutkan satu per satu yang telah menemani dan membantu dalam menyelesaikan studi ini.

HALAMAN MOTO

“Maka, sesungguhnya beserta kesulitan ada kemudahan, sesungguhnya beserta kesulitan ada kemudahan” (QS. Al-Insyirah[94]: 5-6)

“Mohonlah pertolongan (kepada Allah) dengan sabar dan salat. Sesungguhnya (salat) itu benar-benar berat, kecuali bagi orang-orang yang khusyuk” (QS. Al-Baqarah[2]: 45)

“... Boleh jadi kamu membenci sesuatu, padahal ia amat baik bagimu, dan boleh jadi kamu menyukai sesuatu, padahal amat buruk bagimu; Allah mengetahui, sedang kamu tidak mengetahui” (QS. Al-Baqarah[2]: 216)

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya**…”**

(QS.Al-Baqarah[2]: 286)

KATA PENGANTAR

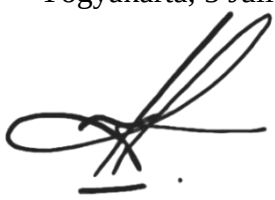
Alhamdulillahirobbil'alamin, Sesungguhnya segala puji hanya milik Allah ﷻ, kami memuji-Nya, memohon pertolongan dan ampunan kepada-Nya, kami berlindung kepada Allah dari kejahatan diri-diri kami dan kejelekan amal perbuatan kami, barangsiapa yang Allah ﷻ beri petunjuk, maka tidak ada yang dapat menyesatkannya, dan barangsiapa yang Allah ﷻ sesatkan, maka tidak ada yang dapat memberinya petunjuk. Shalawat beriring salam semoga senantiasa tercurah kepada manusia paling mulia Rasullullah Muhammad ﷺ dan semoga kelak kita termasuk diantara umatnya yang akan berbahagia mendapatkan syafa'atnya di akhirat. *Aamiin ya robbal'alamin*.

Banyak pihak yang terlibat dalam penyusunan skripsi ini, oleh karena itu penulis ingin menyampaikan terima kasih kepada:

1. Kedua orang tua penulis yang selalu mendukung serta mendo'akan anak-anaknya.
2. Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
3. Dhomas Hatta Fudholi, S.T, M.Eng, Ph.D selaku Ketua Program Studi Informatika Program Sarjana sekaligus Dosen Pembimbing.
4. Seluruh dosen Jurusan Informatika, yang telah membimbing serta memberi ilmu yang bermanfaat.
5. Semua keluarga, yang telah memberikan semangat dan motivasi.
6. Seluruh teman-teman yang telah menemani serta memberikan semangat positif.
7. Seluruh orang yang telah memberi dukungan yang tidak bisa penulis sebutkan satu per satu.

Dalam penulisan tugas akhir ini, sangat jauh dari kata sempurna, namun penulis tetap berharap dengan adanya laporan skripsi ini dapat memberi bermanfaat kepada semua orang dan dapat mengembangkan hal-hal lainnya menjadi lebih besar serta memberi manfaat bagi agama, nusa, dan bangsa.

Yogyakarta, 5 Juli 2023



(Akmal Perdana Hesaputra)

SARI

Kejahatan dan perbuatan yang dilarang akibat penggunaan teknologi informasi telah menjadi pantauan dan masalah serius di beberapa negara. Penerapan undang-undang yang mengatur penggunaan teknologi informasi dilakukan untuk memaksa warga negaranya menahan diri dari perilaku tersebut. Indonesia memberlakukan sanksi yang berbeda untuk setiap orang yang melakukan kejahatan atau perbuatan yang dilarang dalam penggunaan teknologi informasi. Indonesia memberlakukan sanksi berbeda untuk setiap kejahatan dalam penggunaan teknologi informasi yang diatur dalam Undang-undang No. 19 Tahun 2016 tentang perubahan atas Undang-undang No. 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (ITE). Dalam menentukan sanksi dibutuhkan seorang ahli, dan penentuan pasal dibutuhkan waktu yang lama, untuk itu dibutuhkan pendekatan untuk melakukan otomatisasi klasifikasi sanksi pelanggaran berdasarkan UU ITE tersebut.

Tujuan penelitian ini adalah untuk membangun model yang menggunakan dua algoritma deep learning yaitu LSTM dan BiLSTM untuk mengklasifikasi kejahatan dalam UU ITE khususnya pada media sosial Twitter. Dalam pengujiannya, penelitian ini membagi setiap kejahatan dan perbuatan yang dilarang dalam UU ITE kedalam 5 kelas yaitu, pornografi, berita bohong (*hoaks*), *cyberbullying*, ujaran kebencian, dan netral. Berdasarkan hasil percobaan yang dilakukan, model BiLSTM dengan dropout *batch size* 64 *epoch* 25 merupakan model terbaik dengan performa *accuracy* sebesar 0.9875 dan *F1-Score* sebesar 0.9685 berdasarkan data latih serta *accuracy* sebesar 0.972 dan *F1-Score* sebesar 0.900 berdasarkan data uji.

Kata kunci: pelanggaran, klasifikasi, LSTM, BiLSTM, Twitter

GLOSARIUM

Batch size	jumlah sampel data yang diberikan pada jaringan.
Epoch	Jumlah tahapan dalam proses pelatihan data.
Pre-processing	proses membersihkan data dari noise yang tidak diperlukan.
Loss Function	fungsi yang mengukur kesalahan antara output prediksi dan target.
Categorical Crossentropy	fungsi yang umum digunakan dalam tugas klasifikasi multikelas yang mengukur perbedaan antara distribusi probabilitas prediksi model dengan distribusi probabilitas yang sebenarnya dari data
Softmax	fungsi yang umum digunakan dalam lapisan output dari model jaringan saraf untuk melakukan klasifikasi multikelas
Adam	algoritma optimasi yang umum digunakan dalam pelatihan model jaringan saraf dalam pembelajaran mesin yang menggabungkan konsep dari algoritma RMSprop dan algoritma momentum
Scrapping	metode pengambilan data yang dilakukan dengan pemrograman.
Good fitting	performa akurasi training, testing, dan validasi yang seimbang.
Overfitting	performa akurasi training, testing, dan validasi yang tidak seimbang karena terlalu kompleks.
Underfitting	performa akurasi training, testing, dan validasi yang tidak seimbang karena terlalu sederhana.

DAFTAR ISI

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 15
8.1 Latar Belakang Masalah 15
8.2 Rumusan Masalah 17
8.3 Batasan Masalah 17
8.4 Tujuan Penelitian 17
8.5 Manfaat Penelitian 17
8.6 Sistematika Penulisan 18
BAB II LANDASAN TEORI 19
1.1 Penelitian Terdahulu 19
1.2 Klasifikasi Teks 23
1.3 *Natural Language Processing* (NLP) 24
1.4 Deep Learning 24
1.5 Neural Network 26
1.6 Long Short-Term Memory (LSTM) 27
1.7 Bidirectional Long Short-Term Memory (BiLSTM) 28
1.8 *Performance Evaluation Measure* (PEM) 29
BAB III METODOLOGI PENELITIAN 32
2.1 Pengumpulan Data 32
2.2 *Pre-processing* 33
2.2.1 *Cleaning* Data 34
2.2.2 *Casefolding* 35

2.2.3 Normalisasi *slang-word* 36
2.2.4 Menghapus *Stopword* 36
2.3 Pelabelan Data 37
2.4 Membangun Model Klasifikasi 38
2.5 Evaluasi 41
BAB IV HASIL DAN PEMBAHASAN 43
3.1 Dataset 43
3.2 *Pre-processing* 43
3.2.1 *Cleaning* Data 44
3.2.2 *Casefolding* 45
3.2.3 Normalisasi *Slang-word* 45
3.2.4 Menghapus *Stopword* 46
3.2.5 Hasil *pre-processing* 47
3.3 Pelabelan Data 48
3.4 Membangun Model Klasifikasi 49
3.4.1 *Splitting* Dataset 49
3.4.2 *Tokenizing* 49
3.4.3 *Create* Model 50
3.4.4 *Training* Model 53
3.5 Evaluasi 54
3.6 Pengujian *Interface* 73
BAB V PENUTUP 76
4.1 Kesimpulan 76
4.2 Saran 77
DAFTAR PUSTAKA 78
LAMPIRAN 81

DAFTAR TABEL

Tabel 2. 1 Daftar penelitian terdahulu 21
Tabel 3.1 Sumber data sekunder penelitian 33
Tabel 3.2 Contoh proses *cleaning* data 34
Tabel 3. 3 Contoh proses *casefolding* 35
Tabel 3.4 Contoh normalisasi *slang-word* 36
Tabel 3. 5 Contoh penghapusan *stopword* 36
Tabel 3.6 Contoh pelabelan data 38
Tabel 3.7 Skenario hyperparameter model LSTM 40
Tabel 3.8 Skenario hyperparameter model BiLSTM 41
Tabel 4. 1 Hasil *pre-processing* 47
Tabel 4.2 Perbandingan *loss function* LSTM 62
Tabel 4.3 Perbandingan *loss function* BiLSTM 62
Tabel 4.4 Metrik evaluasi LSTM berdasarkan hasil *training* data latih 63
Tabel 4.5 Metrik evaluasi BiLSTM berdasarkan hasil *training* data latih 63
Tabel 4.6 Metrik evaluasi LSTM berdasarkan data uji 71
Tabel 4.7 Metrik evaluasi BiLSTM berdasarkan data uji 71
Tabel 4.8 Hasil prediks salah model BiLSTM dengan dropout *batch size* 64 *epoch* 25 72

DAFTAR GAMBAR

Gambar 2.1 *Supervised learning*.....25
Gambar 2.2 *Semi supervised learning* 25
Gambar 2. 3 *Unsupervised learning* 26
Gambar 2.4 Arsitektur LSTM.....28
Gambar 2.5 Arsitektur BiLSTM.....29
Gambar 2. 6 *Confusion matrix*.....30
Gambar 3. 1 Tahapan penelitian 32
Gambar 3. 2 Tahapan *pre-processing*.....34
Gambar 3.3 Tahapan membangun model klasifikasi.....38
Gambar 3.4 Perbedaan penggunaan dropout 39
Gambar 3.5 Alur kerja model 40
Gambar 3.6 *Multiclass confusion matrix*.....42
Gambar 4. 1 Contoh data yang terkumpul.....43
Gambar 4. 2 Kode program *cleaning* data.....45
Gambar 4. 3 Kode program *casefolding*.....45
Gambar 4. 4 Kode program normalisasi *slang word* 46
Gambar 4.5 Kode program Menghapus s*topword*.....47
Gambar 4. 6 Sebaran data.....48
Gambar 4. 7 Kode program *splitting* data.....49
Gambar 4. 8 Kode program *tokenizing*.....50
Gambar 4. 9 Kode program LSTM tanpa dropout.....51
Gambar 4.10 Model arsitektur LSTM tanpa dropout 51
Gambar 4.11 Kode program BiLSTM tanpa dropout.....51
Gambar 4.12 Model arsitektur BiLSTM tanpa dropout.....52
Gambar 4.13 Kode program LSTM dengan dropout.....52
Gambar 4.14 Model arsitektur LSTM dengan dropout 52
Gambar 4. 15 Kode program BiLSTM dengan dropout.....53
Gambar 4. 16 Model arsitektur BiLSTM dengan dropout.....53
Gambar 4.17 Kode program *training*.....54
Gambar 4.18 Grafik *training* Model LSTM tanpa dropout 55
Gambar 4.19 Grafik *training* Model BiLSTM tanpa dropout 55
Gambar 4.20 Grafik *training* Model LSTM dengan dropout *batch size* 32 *epoch* 20.....56

Gambar 4.21 Grafik *training* Model BiLSTM dengan dropout *batch size* 32 *epoch* 20.....57
Gambar 4.22 Grafik *training* Model LSTM dengan dropout *batch size* 32 *epoch* 25.....57
Gambar 4.23 Grafik *training* Model BiLSTM dengan dropout *batch size* 32 *epoch* 25.....58
Gambar 4.24 Grafik *training* Model LSTM dengan dropout *batch size* 64 *epoch* 20.....59
Gambar 4.25 Grafik *training* Model BiLSTM dengan dropout *batch size* 64 *epoch* 20.....59
Gambar 4.26 Grafik *training* Model LSTM dengan dropout *batch size* 64 *epoch* 25.....60
Gambar 4.27 Grafik *training* Model BiLSTM dengan dropout *batch size* 64 *epoch* 25.....60
Gambar 4.28 Komparasi Grafik *Training* Tiap Model.....61
Gambar 4.29 *Confusion matrix* LSTM tanpa dropout.....64
Gambar 4.30 *Confusion matrix* BiLSTM tanpa dropout.....65
Gambar 4.31 *Confusion matrix* LSTM dengan dropout *batch size* 32 *epoch* 20.....66
Gambar 4.32 *Confusion matrix* BiLSTM dengan dropout *batch size* 32 *epoch* 20.....66
Gambar 4.33 *Confusion matrix* LSTM dengan dropout *batch size* 32 *epoch* 25.....67
Gambar 4.34 *Confusion matrix* BiLSTM dengan dropout *batch size* 32 *epoch* 25.....67
Gambar 4.35 *Confusion matrix* LSTM dengan dropout *batch size* 64 *epoch* 20.....68
Gambar 4.36 *Confusion matrix* BiLSTM dengan dropout *batch size* 64 *epoch* 20.....69
Gambar 4.37 *Confusion matrix* LSTM dengan dropout *batch size* 64 *epoch* 25.....69
Gambar 4.38 *Confusion matrix* BiLSTM dengan dropout *batch size* 64 *epoch* 25.....70
Gambar 4.39 Prediksi Label Netral.....74
Gambar 4.40 Prediksi Label Pornografi.....74
Gambar 4.41 Prediksi Label Hoaks.....74
Gambar 4.42 Prediksi Label *Cyberbullying*.....74
Gambar 4.43 Prediksi Label *Hatespeech*.....75

BAB I
PENDAHULUAN

8.1 Latar Belakang Masalah

Perkembangan Teknologi Informasi yang sangat pesat memiliki berbagai manfaat yang didapat bagi masyarakat. Masyarakat dapat dengan mudah mengakses berbagai informasi seperti berinteraksi melalui sosial media, menginformasikan berita, dan sebagainya. Namun, selain memiliki manfaat, perkembangan teknologi informasi juga memiliki dampak negatif dengan munculnya berbagai kejahatan ataupun perbuatan yang dilarang dalam penggunaan teknologi informasi. Berbagai kasus kejahatan yang terjadi akibat penggunaan teknologi informasi adalah pornografi, berita bohong (hoaks), ujaran kebencian, *cyberbulling*, dan sebagainya. Salah satu tempat penyebaran kejahatan dan perbuatan dilarang akibat penggunaan teknologi informasi adalah sosial media Twitter.

Twitter merupakan salah satu sosial media dengan pengguna terbanyak di Indonesia. Dilansir dari datareportal.com, Digital 2022: Indonesia yang dirilis We are Social (Hootsuite), saat ini pengguna sosial media di Indonesia mencapai 191 juta jiwa dan 58,3% dari data tersebut adalah pengguna Twitter (Kemp, 2022). Berdasarkan data dari infopublik.id, Data Statistik Penangangan Konten Internet Negatif mencatat total pemblokiran konten negatif yang telah dilakukan Kementerian Kominfo mencapai 437.741 konten (Sudoyo, 2023). Dari total pemblokiran konten negatif yang telah diblokir, Twitter menjadi media sosial yang tercatat paling banyak dengan 124.837 konten. Pemblokiran yang telah dilakukan merupakan Langkah awal dalam memberantas kejahatan dan perbuatan yang dilarang akibat penggunaan teknologi informasi di Indonesia.

Kejahatan dan perbuatan yang dilarang akibat penggunaan teknologi informasi telah menjadi pantauan dan masalah serius di beberapa negara. Penerapan undang-undang yang mengatur penggunaan teknologi informasi dilakukan untuk memaksa warga negaranya menahan diri dari perilaku tersebut. Indonesia memberlakukan sanksi yang berbeda untuk setiap orang yang melakukan kejahatan atau perbuatan yang dilarang dalam penggunaan teknologi informasi. Dalam pemberlakuan sanksi kejahatan dan perbuatan yang dilarang tersebut, pemerintah Indonesia mengaturnya dalam Undang-undang No. 19 Tahun 2016 tentang perubahan atas Undang-undang No. 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (ITE). Dalam menentukan sanksi dibutuhkan seorang ahli, dan penentuan pasal

dibutuhkan waktu yang lama, untuk itu dibutuhkan pendekatan untuk melakukan otomatisasi klasifikasi sanksi pelanggaran berdasarkan UU ITE tersebut.

Terdapat berbagai penelitian terkait klasifikasi, salah satunya ialah penelitian yang berjudul Analysis Text of Hate Speech Detection using Recurrent Neural Network (Saksesi et al., 2018). Penelitian ini melakukan deteksi pada ujaran kebencian berbahasa indonesia, dimana data yang dipakai adalah data tweet yang diambil dari Twitter menggunakan Twitter API sebanyak 1235 record. Penelitian ini menggunakan Word2vec sebagai word embedding, lalu data akan dilanjutkan dan diolah menggunakan metode Long Short Term Memory (LSTM) dan menghasilkan akurasi yang tinggi yaitu sebesar 91%.

Selanjutnya penelitian oleh (Fadli & Hidayatullah, 2021) yang berjudul Identifikasi *Cyberbullying* pada Media Sosial Twitter Menggunakan Metode LSTM dan BiLSTM. Penelitian ini membandingkan algoritma LSTM dan BiLSTM yang memiliki performa yang relatif sama. Akurasi dari masing masing algoritma dituliskan sebagai berikut Long Short Term Memory 81,60 % dan Bidirectional Long Short Term Memory 81,78 %. Lalu, untuk nilai dari F1-Score dari masing-masing algoritma sebagai berikut Long Short Term Memory 77,88 % dan Bidirectional Long Short Term Memory 77,89 %.

Selain itu, Penelitian oleh (Hesaputra et al., 2022) yang berjudul Identifikasi Konten Dewasa pada Cuitan Twitter Menggunakan Metode BiLSTM Sebagai Upaya Mengatasi Penyebaran Pornografi Untuk Indonesia Maju. Penelitian ini bertujuan untuk mengembangkan model menggunakan algoritma Bidirectional Long Short Term Memory (BiLSTM) untuk membedakan antara konten dewasa dan non-dewasa pada Twitter. Penelitian ini juga menggunakan Twitter API dan library Tweepy dalam pengumpulan datanya. Dalam pengujiannya, model BiLSTM Double layer dengan dropout merupakan model terbaik Accuracy 98.34% dan F1-Score sebesar 98.32%.

Berdasarkan latar belakang diatas, penelitian ini bertujuan untuk membangun model untuk mengklasifikasi kejahatan dan perbuatan yang dilarang dalam UU ITE khususnya pada media sosial Twitter. Penelitian ini dilakukan karena masih sangat jarang penelitian yang menghasilkan model untuk mendeteksi dan mengklasifikasi kejahatan dan perbuatan yang dilarang dalam UU ITE. Dalam penelitian ini, penulis akan menggunakan algoritma LSTM (Long Short-Term Memory) dan BiLSTM (Bidirectional Long Short-Term Memory) untuk melakukan klasifikasi. Dengan dikembangkannya penelitian ini, diharapkan dapat membantu menegakkan hukum khususnya UU ITE serta mengurangi kasus kejahatan dan penyalahgunaan teknologi informasi di media sosial Twitter.

8.2 Rumusan Masalah

Klasifikasi kejahatan dan perbuatan yang dilarang menjadi salah satu cara untuk menentukan pelanggaran dan sanksi terhadap UU ITE. Proses klasifikasi tersebut dapat menggunakan berbagai metode ataupun algoritma yang telah ada, diantaranya LSTM dan BiLSTM. Berdasarkan latar belakang, dapat diambil rumusan masalah bagaimana membangun serta mengevaluasi model klasifikasi pelanggaran Undang-Undang ITE menggunakan algoritma LSTM dan BiLSTM.

8.3 Batasan Masalah

Untuk mengetahui gambaran yang lebih jelas dari penelitian ini, maka diberikan batasan masalah sebagai berikut:

a. Data yang digunakan bersifat statis menggunakan data dari Twitter.
b. Data *tweet* yang digunakan hanya yang berbahasa Indonesia.
c. Dataset yang digunakan berasal dari beberapa penelitian terkait sebelumnya.
d. Pengklasifikasian berdasarkan 4 pelanggaran yaitu pornografi, berita bohong (hoaks), *cyberbullying*, dan ujaran kebencian (*hatespeech*).
e. Hasil akhir berupa prototipe yang dapat mendeteksi masukan data berupa kalimat dan memberikan luaran berupa klasifikasi pelanggaran dari kalimat tersebut.

8.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

a. Melakukan klasifikasi untuk mendeteksi pada konten Twitter cuitan yang mengandung pelanggaran Undang-undang ITE 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.

8.5 Manfaat Penelitian

Manfaat yang bisa diperoleh dari penelitian ini adalah membantu penegakan hukum untuk memberikan pedoman dalam mengambil keputusan terhadap sanksi pidana yang dijatuhkan sehingga perbuatan yang dilarang dalam undang-undang ITE dapat ditangani melalui pendeteksian konten pada media sosial Twitter.

8.6 Sistematika Penulisan

Sistematika tugas akhir ini dibagi dalam lima bagian, yaitu:

a. BAB I PENDAHULUAN

Bab ini berisi uraian tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penelitian.

b. BAB II LANDASAN TEORI

Bab ini berisi uraian tentang penelitian sebelumnya serta dasar teori dalam melakukan klasifikasi teks.

c. BAB III METODE PENELITIAN

Bab ini berisi uraian tentang langkah-langkah yang dikerjakan selama penelitian.

d. BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi uraian tentang hasil dari pengklasifikasian dan deteksi sanksi pidana berdasarkan kejahatan dan perbuatan yang dilarang dalam UU ITE pada Twitter, serta pembahasan dari hasil tersebut..

e. BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan akhir dan juga saran untuk perbaikan dan pengembangan penelitian selanjutnya

BAB II
LANDASAN TEORI

1.1 Penelitian Terdahulu

Bagian ini membahas beberapa penelitian sebelumnya tentang klasifikasi teks. Terdapat berbagai penelitian terkait klasifikasi, diantaranya ialah penelitian oleh (Alita et al., 2020) yang berjudul Implementasi Algoritma Multiclass SVM Pada Opini Publik Berbahasa Indonesia di Twitter. Penelitian ini menggunakan 2000 dataset yang didapat dari opini twitter berbahasa Indonesia tentang layanan BPJS dan telekomunikasi seluler. Penelitian ini juga memanfaatkan algoritma Multiclass Support Vector Machine untuk mengklasifikasikan opini public berbahasa Indonesia yang didapat dari Twitter serta membandingkan dengan algoritma Naïve Bayes Classifier. Berdasarkan percobaannya, hasil penelitian mereka menyatakan bahwa algoritma Multiclass Support Vector Machine lebih unggul dibanding algoritma Naïve Bayes Classifier dengan nilai 89.00% untuk F1-Score.

Selain itu, penelitian oleh (Manoppo & Fudholi, 2021) yang berjudul Deteksi *Cyberbullying* Berdasarkan Unsur Perbuatan Pidana yang Dilanggar dengan Naïve Bayes dan Support Vector Machine. Penelitian ini mengklasifikasikan lima unsur perbuatan pidana terkait karakteristik *cyberbullying* yaitu penghinaan, menuduh dan bersifat pencemaran, rasa kebencian terkait SARA, ancaman kekerasan, dan ancaman membuka rahasia dengan menggunakan 5000 tweet sebagai dataset. Penelitian ini juga menggunakan ekstraksi fitur dengan metode N-grams dengan pembobotan TF-IDF. Hasil penelitian tersebut dapat dengan tepat memprediksi potensi *Cyberbullying* setelah dilakukan resampling dengan over-sampling menggunakan SMOTE melalui unsur pelanggaran tindak pidana dengan *performance measurement* rata-rata diatas 90%.

Disisi lain penelitian oleh (Hidayatullah et al., 2019) dengan judul Adult Content Classification on Indonesian. Penelitian ini melakukan klasifikasi serta membandingkan algoritma Long Short Term Memory (LSTM) dengan berbagai algoritma pada data Twitter untuk klasifikasi konten dewasa. Dari penelitian tersebut menunjukkan bahwa model LSTM double layer dengan dropout merupakan model yang memiliki akurasi 98,38%. Dalam penelitian tersebut menyampaikan bahwa algoritma LSTM menghasilkan performa yang lebih unggul dari algortima *machine learning* tradisional, seperti *Support Vector Classification, Logistic Regression,* dan *Multinomial Naïve Bayes*.

Kemudian penelitian oleh (Hidayatullah et al., 2020) yang berjudul Sentiment Analysis on Twitter using Neural Network: Indonesian Presidential Election 2019 Dataset. Penelitian ini bertujuan untuk mengklasifikasikan sentimen pada tweet pemilihan presiden Indonesia 2019 data dengan menggunakan *neural network*. Penelitian ini melatih kumpulan data menggunakan beberapa varian algoritma *deep neural network*, termasuk Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), CNN-LSTM, Gated Recurrent Unit (GRU)-LSTM dan BiLSTM. Selain itu, sebagai perbandingan dengan model pembelajaran mendalam kami, kami juga melatih dataset kami menggunakan algoritme pembelajaran mesin tradisional lainnya, yaitu Support Vector Machine (SVM), Logistic Regression (LR) dan Multinomial Naïve Bayes (MNB). Hasil dari percobaan ini menunjukkan bahwa BiLSTM mencapai kinerja terbaik dengan akurasi sebesar 84,60%. Penelitian lainnya oleh (Pipin & Kurniawan, 2022) yang berjudul Analisis Sentimen Kebijakan MBKM Berdasarkan Opini Masyarakat di Twitter Menggunakan LSTM. Penelitian ini menganalisis sentiment *multiclass* pada *tweet* Bahasa Indonesia ke dalam enam kelas emosi pada 658 *tweet* menggunakan algoritma Long Short-Term Memory. Hasil dari penelitian tersebut mendapatkan nilai 80.42% untuk akurasi.

Selanjutnya penelitian yang berjudul A Hybrid CNN-BiLSTM Model For Drug Named Entity Recognition (Fudholi. et al., 2022). Penelitian ini bertujuan untuk mengekstrak entitas nama obat. Dalam penelitiannya, penulis menggunakan model arsitektur CNN-BiLSTM hybrid yang secara otomatis mendeteksi fitur tingkat kata dan karakter. Berdasarkan percobaan, model tersebut menghasilkan *Recall* 0.903, *Precision* 0.892, dan *F1-Score* 0.892. Penelitian lainnya oleh (Hesaputra et al., 2022) yang berjudul Identifikasi Konten Dewasa pada Cuitan Twitter Menggunakan Metode BiLSTM Sebagai Upaya Mengatasi Penyebaran Pornografi Untuk Indonesia Maju. Penelitian ini melakukan klasifikasi konten dewasa pada Twitter menggunakan algoritma Bidirectional Long Short-Term Memory (BiLSTM). Pada pengujiannya penelitian ini menggunakan metode Dropout untuk mendapatkan hasil yang *good fitting*. Berdasarkan pengujiannya, didapat hasil 98.34% untuk Accuracy dan 98.32% untuk F1-Score.

Di lain sisi penelitian oleh (Widianto & Fudholi, 2021) yang berjudul Klasifikasi Emosi Pada Teks Dengan Menggunakan Metode Deep Learning. Penelitian ini bertujuan melakukan klasifikasi pada teks untuk mengetahui jenis emosi menggunakan BERT (Bidirectional Encoder Represantions from Tranformers). Berdasarkan pengujian yang telah dilakukan, metode BERT menghasilkan akurasi sebesar 89.2% dengan sembilan kelas emosi sedangkan

metode indoBERT menghasilkan akurasi 76%. Penelitian yang dilakukan oleh (Tandijaya et al., 2021) yang berjudul Klasifikasi dalam Pembuatan Portal Berita Online dengan Menggunakan Metode BERT. Penelitian ini bertujuan untuk mengklasifikasikan kategori dari sebuah berita. Metode yang digunakan dalam penelitian ini adalah BERT dan Pengujian dengan pretrained model indobenchmark/indobert-base-p1 mendapatkan hasil yang baik dimana akurasi dapat mencapai 87.548%.

Penelitian oleh (Kusnadi et al., 2021) yang berjudul Analisis Sentimen Terhadap Game Genshin Impact Menggunakan BERT. Penelitian ini berfokus pada analisis sentimen dengan tujuan mengetahui apakah ulasan terpercaya yang dikumpulkan dari Google Play Store memiliki sentimen netral, baik atau sentimen buruk sehingga dapat membantu pengembangan permainan kedepannya. Penelitian ini menggunakan metode BERT dan memiliki nilai macro averaged f1-score 0.75 atau 75%. Selain itu, penelitian oleh (Ilham & Maharani, 2022) yang berjudul Analyze Detection Depression In Social Media Twitter Using Bidirectional Encoder Representations from Transformers. Penelitian ini berfokus untuk mendeteksi orang yang mengalami depresi dengan menggunakan metode BERT (Bidirentional Encoder Representations from Transformers). Berdasarkan percobaan, penelitian tersebut menghasil akurasi terbaik sebesar 0,7176. Tabel 2.1 merupakan beberapa penelitian terdahulu terkait dengan klasifikasi. Tabel tersebut berisi tentang judul, deskripsi dataset yang digunakan, metode yang digunakan, jumlah label atau kelas pada penelitian serta hasil dari penelitian.

Tabel 2. 1 Daftar penelitian terdahulu

No	Judul	Dataset	Label	Metode	Hasil
1.	Implementasi Algoritma Multiclass SVM Pada Opini Publik Berbahasa Indonesia di Twitter	2000 opini public berbahasa Indonesia mengenai jaringan telekomunikasi seluler dan layanan BPJS	Multi class	*Support Vector Machine*	*F1-Score 89.00%*
2.	Deteksi *Cyberbullying* Berdasarkan Unsur Perbuatan Pidana yang Dilanggar dengan Naïve	5000 tweet Bahasa Indonesia	Multi class	*Naïve Bayes* dan *Support*	*performance measurement* rata-rata diatas 90%.

	Bayes dan Support Vector Machine			*Vector Machine*	
3.	Adult Content Classification on Indonesian	51.954 twit bahasa indonesia	Binary class	LSTM	*Accuracy* 98,38%
4.	Sentiment Analysis on Twitter using Neural Network: Indonesian Presidential Election 2019 Dataset	115,931 twit bahasa indonesia	Binary class	BiLSTM	*Accuracy* 84.60%
5.	Analisis Sentimen Kebijakan MBKM Berdasarkan Opini Masyarakat di Twitter Menggunakan LSTM	*tweet* Bahasa Indonesia ke dalam enam kelas emosi pada 658 tweet	Multi class	LSTM	*Binary* 80.42%
6.	A Hybrid CNN-BiLSTM Model For Drug Named Entity Recognition	18.075 dataset bahasa indonesia	Multi class	CNN-BiLSTM	*F1-score* 89.2%, *precision* 88.1%, dan *recall* 90.3%
7.	Identifikasi Konten Dewasa pada Cuitan Twitter Menggunakan Metode BiLSTM Sebagai Upaya Mengatasi Penyebaran Pornografi Untuk Indonesia Maju.	52.338 tweet Bahasa Indonesia	Binary class	BiLSTM	*F1-Score* 98.32% dan *Accuracy* 98.34%
8.	Klasifikasi Emosi Pada Teks Dengan Menggunakan Metode Deep Learning	2.700 dataset bahasa indonesia	Multi class	BERT dan IndoBERT	*Accuracy* BERT 89.2% dan

					IndoBERT 76%
9.	Klasifikasi dalam Pembuatan Portal Berita Online dengan Menggunakan Metode BERT	Bahasa indonesia 6.309 dataset	Multi class	BERT	*Accuracy* 87.548%
10.	Analisis Sentimen Terhadap Game Genshin Impact Menggunakan BERT	Ulasan dari Google Play Store	Multi class	BERT	*Precision* 86%, R*ecall* 78%, dan *F1-Score* 82%
11.	Analyze Detection Depression In Social Media Twitter Using Bidirectional Encoder Representations from Transformers	3.867 dataset bahasa indonesia	Single class	BERT	*Accuracy* 71.76%

Berdasarkan penelitian-penelitian yang telah disebutkan, kami berfokus untuk membangun model klasifikasi multi kelas menggunakan algoritma LSTM dan BiLSTM untuk klasifikasi kejahatan dan perbuatan yang dilarang dalam UU ITE pada Twitter. Kemudian dari model yang didapat akan dibuat *interface* sederhana untuk melihat apakah model dapat berjalan dengan baik. Diharapkan dengan adanya penelitian ini dapat membantu menegakkan hukum khususnya UU ITE serta membantu mengurangi kasus kejahatan dan penyalahgunaan teknologi informasi.

1.2 Klasifikasi Teks

Klasifikasi teks adalah proses pengelompokan suatu data ataupun dokumen ke dalam suatu kategori yang telah dibuat dan didefinisikan ke dalam kategori kelas (Torregrosa et al., 2022). Tujuan dari klasifikasi teks adah untuk menganalisa, memproses, dan mengekstrak suatu informasi yang terkandung dalam teks. Dalam klasifikasi emosi pada teks, suatu teks akan diambil sebuah informasi yang terkandung di dalamnya untuk mengetahui emosi yang terkandung dalam teks opini tersebut. Teks-teks opini cenderung berbentuk data yang tidak

terstruktur, maka perlu dilakukan tahap *preprocessing* agar menjadi lebih terstruktur serta informasi dapat diserap dengan mudah. *Preprocessing* teks bertujuan untuk memecah setiap dokumen menjadi kata-kata individu dengan mewakili masing-masing sebagai vektor fitur. Untuk tujuan pengindeksan dokumen, proses pemilihan fitur dan fase preprocessing teks utama harus digunakan untuk memilih kata kunci. Tahap preprocessing teks, di sisi lain, membagi dokumen teks input menjadi fitur yang dikenal sebagai (*tokenization, words, terms* atau *attributes*) (Kadhim, 2018).

1.3 *Natural Language Processing* (NLP)

Natural language processing atau pengolahan bahasa alami merupakan salah satu cabang ilmu yang meneliti dan mengembangkan cara kerja komputer untuk memahami dan memproses bahasa alami sebagai kalimat maupun ucapan (Vig & Belinkov, 2019). Pada *natural language processing* (NLP) digunakan untuk mengevaluasi bagaimana bahasa dalam teks sesuai dengan aturan tata bahasa. Banyak organisasi dewasa ini memiliki begitu banyak data suara dan teks dari berbagai saluran komunikasi seperti email, pesan teks, umpan berita media sosial, video, audio, dan banyak lagi. Organisasi tersebut menggunakan perangkat lunak NLP untuk memproses data ini secara otomatis, menganalisis maksud atau sentimen dalam pesan, dan merespons komunikasi manusia dalam waktu nyata.

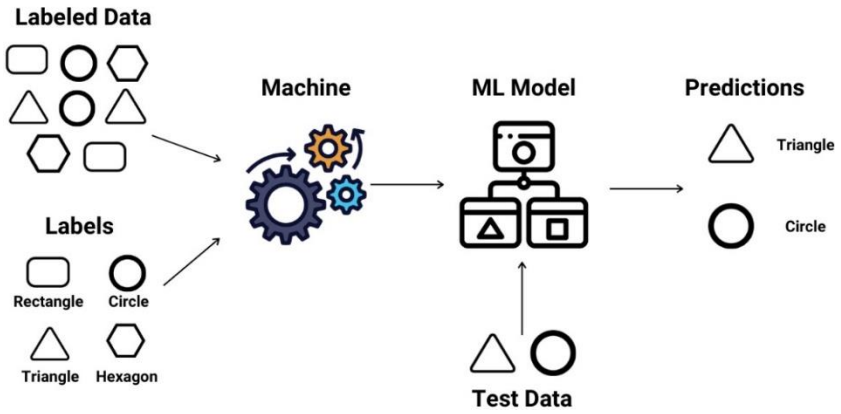
1.4 Deep Learning

Deep learning adalah cabang ilmu dari *Machine Learing* yang menjadi bagian dari kecerdasan buatan. Teknik *Deep Learning* adalah bagian dari *neural network* yang terkenal ketika struktur multilayer, banyak dipakai karena dapat menangani banyak masalah sekaligus dan memberikan solusi yang unik (Amigo, 2021). *Deep Learning* dapat mempelajari suatu data agar dapat mempresentasikan data dengan baik, sehingga dapat melakukan prediksi dengan baik. *Deep learning* dibagi menjadi tiga metode diantaranya.

a. Supervised Learning

Supervised learning merupakan salah satu pendekatan dalam pembelajaran mesin dimana sebuah model diajarkan untuk mempelajari hubungan antara input data dan output yang diminta. Pada Gambar 2.1 merepresentasikan metode *supervised learning.* Data latih yang digunakan pada metode ini telah diberi label terlebih dahulu, kemudian dilakukan pelatihan terhadap data latih menggunakan algoritma yang telah ditentukan. Setelah pelatihan dilakukan maka didapat sebuah model yang dapat memprediksi sesuai dengan label yang dilatih pada

data latih.

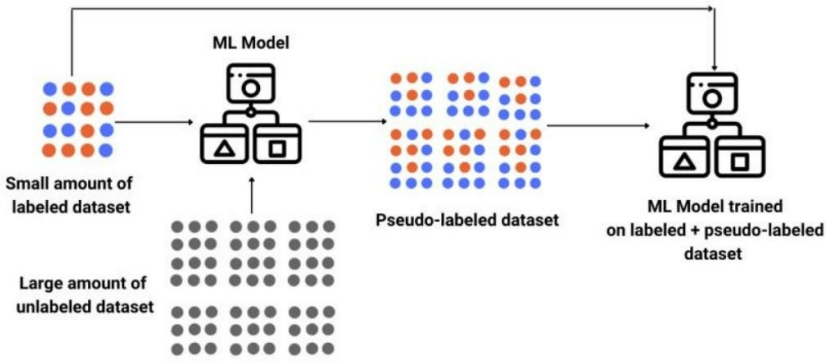


Gambar 2.1 *Supervised learning*

Sumber: enjoyalgorithms.com

b. Semi Supervised Learning

Semi supervised learning merupakan salah satu pendekatan dalam pembelajaran mesin dimana kita memiliki Sebagian data yang diberi label dan sebagiannya lagi tidak diberi label. Pada Gambar 2.2 merepresentasikan metode *semi supervised learning*. Sebagian kecil data latih yang diberi label dilatih menggunakan algoritma yang sudah ditentukan sehingga mendapat sebuah model yang akan melabeli seluruh atau Sebagian besar data lainnya yang dinamakan *pseudo label*. Setelah seluruh data latih diberi label, data latih tersebut Kembali dilatih untuk mendapatkan sebuah model yang akan memprediksi sesuai dengan label yang dilatih pada data latih.

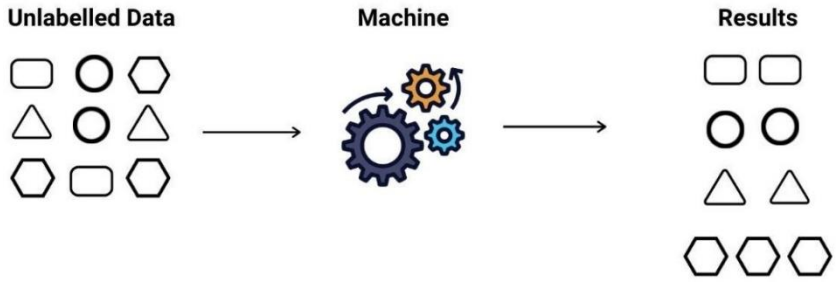


Gambar 2.2 *Semi supervised learning*

Sumber: enjoyalgorithms.com

c. Unsupervised Learning

Unsupervised learning merupakan salah satu pendekatan dalam pembelajaran mesin di mana data latih tidak memiliki label atau output yang diketahui sebelumnya. Gambar 2.3 merepresentasikan metode *unsupervised learning*. Pada tersebut dapat dilihat bahwa data latih yang digunakan tidak diberi label, namun langsung dilakukan pelatihan terhadap data latih menggunakan algoritma yang sudah ditentukan. Output dari metode ini berupa pengelompokan data yang dilakukan oleh algoritma.



Gambar 2. 3 *Unsupervised learning*

Sumber: enjoyalgorithms.com

1.5 Neural Network

Neural network adalah cabang penelitian yang meniru bagaimana otak manusia memproses informasi dan menghasilkan hasil. Dengan meniru prinsip pembelajaran yang terinspirasi dari bagaimana sistem saraf manusia dan organisme biologis lainnya berfungsi. *Neural network*, juga dikenal sebagai jaringan saraf tiruan, adalah salah satu teknik pembelajaran mesin yang paling terkenal. Neuron adalah sel khusus yang menyusun sistem saraf. Akson dan dendrit berfungsi sebagai mekanisme penghubung antara neuron-neuron. Hubungan antara akson dan dendrit disebut sinapsis (Chatterjee et al., 2019).

Arsitektur jaringan yang menghubungkan tingkat yang berbeda, disebut sebagai arsitekturjaringan. Lapisan antara lapisan input dan output dikenal sebagai *hidden layer*, sedangkan lapisan di atas lapisan output dikenal sebagai *hidden units*. Unit-unit ini tidak dapat langsung dilihat sebagai *input* atau *output* dari luar, oleh karena itu disebut dengan istilah "*hidden*". Lapisan tersembunyi yang terdiri dari unit tersembunyi, yang masing-masing merupakan unit saraf, merupakan inti dari jaringan saraf. Ini menerapkan non-linearitas ke jumlah tertimbang dari input di lapisan ini. Setiap unit lapisan berkomunikasi satu sama lain dengan menggunakan semua unit di lapisan bawah sebagai *input* dan *output*, serta koneksi

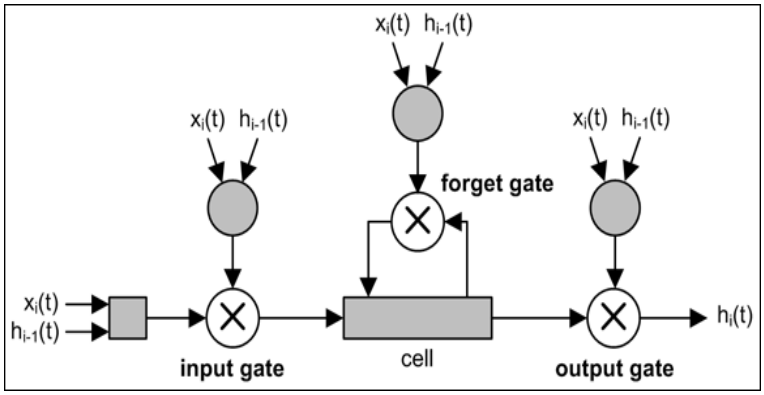
antara setiap pasangan unit dalam dua lapisan yang berdekatan. Unit *input* ditambahkan bersama oleh setiap unit tersembunyi (Jurafsky & Martin, 2019).

Secara umum terdapat dua jenis arsitektur *neural network*, yaitu feed-forward network danrecurrent/recursive network.

1. *Feed-forward Networ*k atau *Multi-Layer Perceptrons* (MLP) adalah jaringan di mana komponen tidak digilir dan *output* dikirim kembali ke tingkat bawah. Hal ini memungkinkan jaringan untuk beroperasi dengan *input* ukuran tetap atau *input* panjangvariabel dari serangkaian bagian yang dapat diabaikan. Jaringan belajar untuk mengintegrasikan komponen *input* saat mereka dimasukkan ke dalamnya. Data hanya mengalir dari *input* ke *output* dalam satu arah. Jaringan semacam ini digunakan untuk pengenalan pola karena lebih sederhana. Pengenalan gambar sering menggunakan jeniskhusus jaringan umpan maju yang disebut *Convolutional Neural Network (CNN* atau *ConvNet)*
2. Dalam situasi di mana input berurutan. *Recurrent Neural Network (RNN)* sering digunakan ketika jaringan sedang memproses teks atau suara,. Urutan objek dimasukkan ke dalam RNN, dan menghasilkan vektor dengan ukuran tertentu yang berisi urutan tersebut. RNN memungkinkan *loop* dan memungkinkan data mengalir dikedua arah melalui jaringan. Jaringan ini lebih canggih dan kuat dari CNN.

1.6 Long Short-Term Memory (LSTM)

LSTM merupakan adaptasi dari *Recurrent Neural Network* (RNN) yang memiliki *memory cell* tambahan untuk menyimpan informasi dalam jangka waktu yang lebih panjang. Salah satu keunggulan LSTM adalah kemampuannya dalam menangani masalah vanishing gradient yang sering terjadi pada RNN Ketika memproses data sekuensial yang panjang. Hal ini dicapai dengan penggunaan satu set *gate* yang berfungsi untuk mengendalikan aliran informasi yang masuk ke dalam memori LSTM (Manaswi, 2018). Dengan adanya penambahan *memory cell* dan penggunaan *gate*, LSTM dapat mempertahankan informasi jangka panjang dan mencegah hilangnya *gradient* selama proses pembelajaran pada data sekuensial yang panjang. Gambar 2.4 menunjukkan arsitektur dari LSTM.



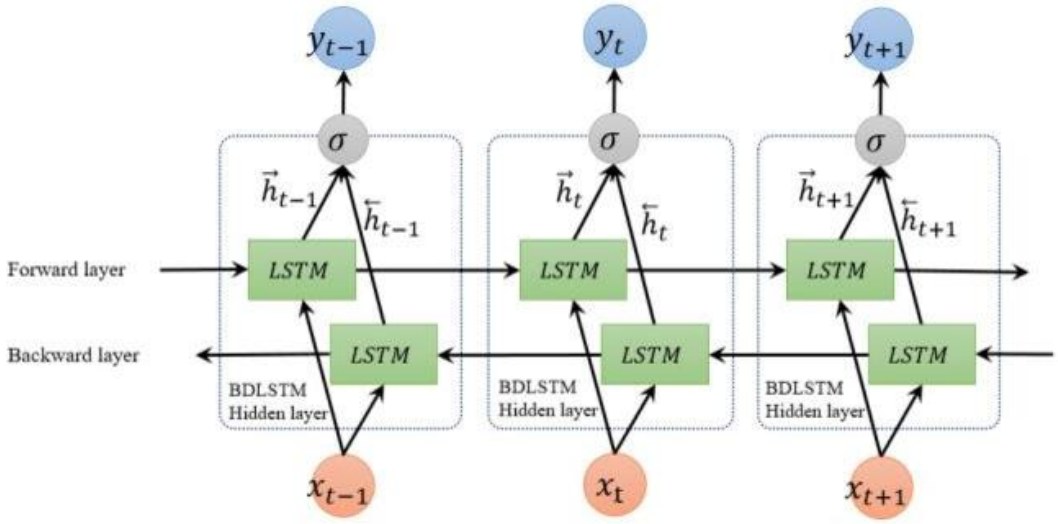
Gambar 2.4 Arsitektur LSTM

Sumber: indoml.com

Pada Gambar 2.4 diatas terdapat beberapa bagian dari arsitektur LSTM. *Cell state* adalah komponen dalam LSTM yang berfungsi untuk menyimpan informasi dari satu langkah waktu ke langkah waktu berikutnya. *Gate units* memiliki peran penting dalam memproses informasi yang masuk dan keluar dari *cell state*. *Gate units* terdiri dari tiga jenis, yaitu *input gate*, *forget gate*, dan *output gate*. *Input gate* berfungsi untuk menentukan nilai input mana yang akan diteruskan dan diperbarui dalam *cell state*. *Forget gate* bertugas untuk mengidentfikasi dan memutuskan informasi mana yang harus dihapus dari *cell state*. *Output gate* mengatur proses *output* yang dihasilkan dari *cell state* pada langkah waktu tertentu. Dengan adanya *input gate*, *forget gate*, dan *output gate*, LSTM dapat mengatur aliran informasi dalam *cell state* secara efisien serta memungkinkan penyimpanan informasi yang relevan dan menghasilkan *output* yang tepat.

1.7 Bidirectional Long Short-Term Memory (BiLSTM)

BiLSTM merupakan pengembangan dari model LSTM dimana terdapat dua lapisan yang bekerja secara berkebalikan arah. Model ini sangat efektif dalam mengenali pola dalam kalimat karena setiap kata dalam dokumen diproses secara sekuensial. Pemahaman konten *tweet* dapat tercapai melalui pembelajaran urutan kata-kata. Lapisan bawah bergerak maju (*forward*), memahami dan memproses dari kata pertama ke kata terakhir, sementara lapisan atas bergerak mundur (*backward*), memahami dan memproses dari kata terakhir ke kata pertama. Dengan adanya lapisan dua arah yang saling berlawanan ini, model dapat memperoleh pemahaman yang lebih dalam melihat perspektif dari kata-kata sebelumnya dan kata-kata berikutnya. Hal ini memungkinkan model untuk memahami konteks dalam *tweet* dengan lebih baik. Gambar 2.5 menggambarkan arsitektur dari BiLSTM tersebut.



Gambar 2.5 Arsitektur BiLSTM

Sumber: gabormelli.com

Pada Gambar 2.5 terlihat bahwa setiap hidden unit pada lapisan bawah dan atas digabungkan untuk membentuk nilai fitur kata dengan dimensi yang lebih panjang dibandingkan dengan penggunaan LSTM biasa. Karena meliki fitur yang lebih panjang, maka informasi yang akan diproses pada tahap selanjutnya, yaitu jaringan saraf feed forward, dapat diklasifikasikan dengan lebih akurat. Keluaran dari *forward layer* $\vec{\boldsymbol{h}}$ dihitung secara iteratif menggunakan input dalam bentuk positif berurutan dari waktu T – n ke $T - 1$, sedangkan keluarn dari *backward layer* $\overleftarrow{\boldsymbol{h}}$ dihitung menggunakan masukan terbalik dari waktu T – n ke $T - 1$. BiLSTM memiliki keunggulan dalam pelabelan sekuensial karena dapat mengakses informasi sebelum dan sesudah suatu titik dalam sekuensial. Pada dasarnya, hidden state pada LSTM hanya mengambil informasi dari masa lalu dan tidak memiliki akses langsung ke informasi yang akan datang. Untuk mengatasi masalah tersebut dapat menggunkan BiLSTM (Ma & Hovy, 2016). BiLSTM terdiri dari dua LSTM, yaitu LSTM maju (*forward*) dan LSTM mundur (*backward*), yang bekerja Bersama untuk menangkap informasi dari kedua arah, baik sebelum maupun sesudah titik yang sedang diproses. Dengan demikian, BiLSTM memungkinkan pengambilan informasi kontekstual yang lebih lengkap dalam proses pelabelan sekuensial.

1.8 *Performance Evaluation Measure* (PEM)

Performance Evaluation Measure (PEM) merupakan tahapan yang digunakan untuk mengukur performa dan kinerja model. PEM biasanya direpresentasikan dengan sebuah

confusion matrix. *Confussion matrix* adalah tabel yang merangkum kinerja model. Penggunaan *confussion matrix* dapat mengidentifikasi kelemahan dalam operasi algoritma. Ini digunakan untuk mengevaluasi efektivitas model klasifikasi. Dengan menggunakan *confussion matrix* peneliti dapat mengetahui *error* pada operasi algoritma yang dijalankan. Gambar 2.6 menunjukan konsep dasar dalam *confussion matrix*.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FP
	Negative	FN	TN

Gambar 2. 6 *Confusion matrix*

Keterangan:

a. TP adalah *true positive* yaitu sebuah data dengan kondisi aktual yang mampu memprediksi dengan benar.
b. TN adalah *true negative* adalah data negatif yang diprediksi dengan benar.
c. FP adalah *false positive* yaitu sebuah data prediksi yang tidak sesuai.
d. FN adalah *false negative* yaitu sebuah data negatif yang diprediksi tidak sesuai.

Ada banyak metrik evaluasi untuk menghitung nilai PEM selain *confusion matrix* diantaranya yaitu *accuracy*, *precision*, *recall*, dan *F1-Score*. *Accuracy* merupakan perbandingan antara prediksi benar yang dijawab sistem dengan keseluruhan data yang mengukur tingkat kedekatan antara nilai prediksi dengan nilai actual. Rumus *accuracy* ditunjukkan pada persamaan (2.1). Dari persamaan tersebut dapat dilihat bahwa nilai *true positive* dan *true negative* dijumlahkan kemudian dibagi dengan jumlah nilai dari seluruh hasil *confusion matrix* kemudian dikalikan seratus persen.

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

Precision merupakan nilai yang mengukur tingkat ketepatan antara prediksi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Rumus *precision* ditunjukkan pada

persamaan (2.2). Dari persamaan tersebut dapat dilihat bahwa nilai *true positive* dibagi dengan jumlah nilai *true positive* dan *false positive* kemudian dikalikan seratus persen.

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

Recall merupakan ketepatan antara prediksi yang sama dengan prediksi yang pernah dipanggil sebelumnya untuk mengukur tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Rumus *recall* ditunjukkan pada persamaan (2.3). Dari persamaan tersebut dapat dilihat bahwa nilai *true positive* dibagi dengan jumlah nilai *true positive* dan *false negative* kemudian dikalikan seratus persen.

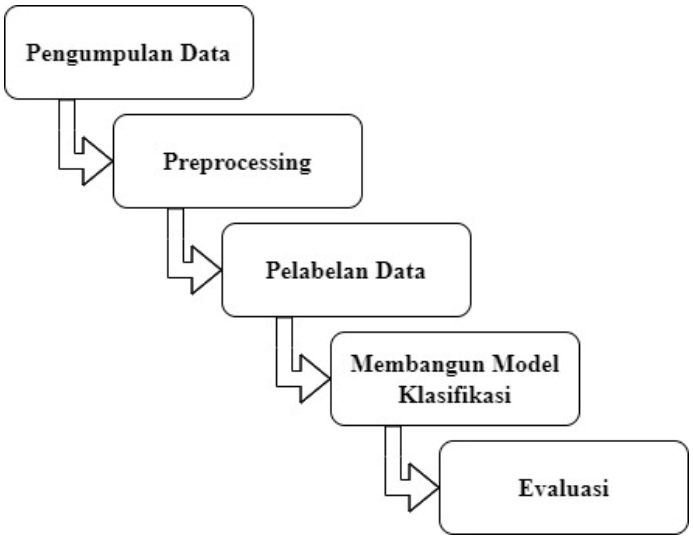
$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2.3)$$

F1-Score merupakan perbandingan rata-rata *precision* dan *recall* yang dibobotkan dan berfungsi untuk menghitung jumlah distribusi data yang tidak seimbang. Rumus *F1-Score* ditunjukkan pada persamaan (2.4). Dari persamaan tersebut dapat dilihat bahwa *precision* dikalikan *recall* dikalikan dua kemudian dibagi jumlah *precision* ditambah *recall* kemudian dikalikan seratus persen.

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

BAB III
METODOLOGI PENELITIAN

Bagian ini menjelaskan metodologi yang digunakan pada penelitian. Alur metode dalam penelitian ini dapat dilihat pada gambar 3.1. Penelitian ini memuat langkah-langkah yang diantaranya pengumpulan data, *preprocessing,* pelabelan data, pemodelan, pengujian, dan evaluasi.



Gambar 3. 1 Tahapan penelitian

2.1 Pengumpulan Data

Dalam penelitian ini melakukan klasifikasi pelanggaran Undang-Undang ITE yang dibagi kedalam lima kelas, yaitu netral, pornografi, hoaks, *cyberbullying*, dan *hatespeech*. Adapun pemilihan lima kelas tersebut berdasarkan kejahatan pada media sosial terbanyak berdasarkan pasal-pasal pelanggaran Undang-Undang ITE yang diberitakan oleh sindonews.com (Kiswondari, 2023). Untuk melakukan proses klasifikasi, maka diperlukan dataset untuk melatih model yang akan dibangun. Dataset yang akan digunakan berupa opini masyarakat pada media sosial Twitter berupa teks. Pada awalnya dalam proses pengumpulan data, dilakukan dengan menggunakan teknik *scrapping*. Untuk data yang diperoleh menggunakan teknik *scraping* menggunakan Twitter API dengan memanfaatkan *library tweepy* untuk mengakses data *tweet*. Proses pengumpulan data dilakukan berdasarkan akun dan kata kunci dari Twitter.

Proses tersebut dilakukan pada tanggal 18 Maret 2022. Dari proses tersebut, data yang telah berhasil terkumpul berjumlah 17.384 data dan dijadikan kedalam bentuk file *excel*.

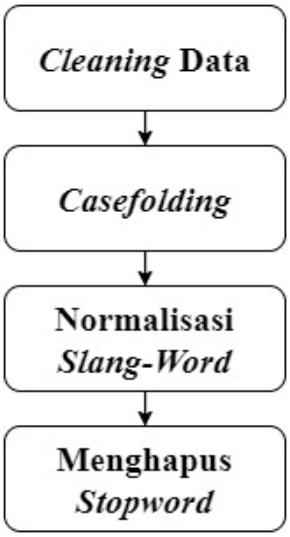
Namun, setelah memperoleh dataset dari *scrapping*, ternyata dataset tersebut masih belum memenuhi kriteria dari lima kelas klasifikasi yang ditetapkan. Oleh karena itu, dilakukan pengumpulan data dengan menggunakan data sekunder yang diperoleh dari berbagai dari penelitian sebelumnya. Proses pengumpulan data sekunder ini dilakukan dengan melihat penelitian yang menggunakan data terkait dengan pasal-pasal pelanggaran Undang-Undang ITE. Selain itu, data sekunder tersebut juga harus diperoleh dari media sosial Twitter. Tabel 3.1 menampilkan daftar sumber data sekunder dalam penelitian ini.

Tabel 3.1 Sumber data sekunder penelitian

No.	Sumber	Jenis Data	Jumlah
1.	Scrapping dari Twitter	*Hatespeech* SARA	17.384
2.	(Hidayatullah et al., 2019)	Pornografi	51.954
3.	(Atmajaya et al. 2022)	Hoaks dan *hatespeech*	6.947
4.	(Panjaitan, 2021)	*Cyberbullying*	104
5.	(Ibrohim & Budi, 2019)	*Cyberbullying*	13.169

2.2 *Pre-processing*

Pada tahap ini data yang telah terkumpulkan akan diproses untuk mengambil informasi yang terkandung didalamnya karena masih banyak data yang kotor dan adanya berbagai simbol yang tidak diperlukan. Proses ini diperlukan untuk mengurangi data noise dan tidak konsisten. Pada proses *pre-processing* yang dilakukan antara lain adalah menghapus URL, menghapus karakter khusus Twitter (seperti: #hashtag, RT, cc, @username/mention), menghapus digit, menghapus non-ASCII, menghapus *punctuation* (tanda baca), menghapus karakter tertawa, menghapus *multi-space*, menghapus karakter berulang, *casefolding*, normalisasi kata tidak baku (*slangword)*, dan menghapus *stopword*. Dalam penelitian ini, kami melakukan *pre-processing* data untuk dataset Twitter dengan mengadopsi langkah-langkah *pre-processing* yang dilakukan oleh (Hesaputra et al., 2022). Gambar 3.2 menampilkan tahapan *pre-processing* yang dilakukan pada penelitian ini.



Gambar 3. 2 Tahapan *pre-processing*

2.2.1 *Cleaning* Data

Tidak semua *tweet* yang diperoleh merupakan teks yang terstruktur dan sesuai dengan ketentuan penulisan yang benar. *Cleaning* data bertujuan untuk membersihkan dataset yang telah dikumpulkan dengan menghapus URL, menghapus karakter khusus Twitter (seperti: #hashtag, RT, cc, @username/mention dan lain-lain), menghapus digit, menghapus non-ASCII, menghapus *punctuation* (tanda baca), menghapus karakter tertawa, menghapus *multi-space*, menghapus karakter berulang. Proses ini dijalankan menggunakan bahasa pemrograman Python pada Jupyter Notebook yang hasilnya disimpan dalam file berformat xlsx. Tabel 3.2 menampilkan contoh penerapan *cleaning* data pada dataset. Tabel tersebut berisi data yang belum dilakukan *pre-processing* pada kolom sebelum. Sedangkan pada kolom sesudah berisi data yang sudah dilakukan *pre-processing*.

Tabel 3.2 Contoh proses *cleaning* data

Sebelum	Sesudah
Wow, geraknya bisa naik turun 😍#bigolive #dessy #mulus #seksi #cantik https://t.co/hWlp7OBiSj	Wow geraknya bisa naik turun
Lagi jam kosong dikelas dua abg ini live gogo sambil remas toketnya #bokepabg #bokepsma #cewekbispak #abgmesum #sangekberat https://t.co/BPOfZkiC6h	Lagi jam kosong dikelas dua abg ini live gogo sambil remas toketnya

Enak susunya ma ma ma 🤣 #abgseksi #pahamulus #cantik #montok #semok https://t.co/KnpfktUjha	Enak susunya ma ma ma
@FeliciaNichola halo semua nya... salken follow @BiroSeks update cerita dewasa tiap harinya..muach.. 😘 😘	halo semua nya salken follow update cerita d ewasa tiap harinya muach
Pinter dan Persis Presiden *JOKO WIDODO.* Pakdhe @jokowi idola banget hingga ke luar negeri. Top !!. kami masyarakat Imdonesia bangga https://t.co/oPlvZe1U8u	Pinter dan Persis Presiden JOKO WIDODO Pakdhe idola banget hingga ke luar negeri T op kami masyarakat Imdonesia bangga
@pinatih88 @ChusnulCh__ @jokowi @mohmahfudmd @PDI_Perjuangan Disitu OTAK?? DARIPADA NDESO BISANYA HANYA NGUTANG, JUAL, BOHONG??... NYUSAHIN RAKYAT?? https://t.co/4ysflxpMkm	Disitu OTAK DARIPADA NDESO BISANYA HANYA NGUTANG JUAL BOHONG NYUSAHIN RAKYAT

2.2.2 *Casefolding*

Casefolding digunakan untuk melakukan generalisasi penggunaan huruf kapital. *Casefolding* akan mengubah semua huruf menjadi huruf kecil (*lowercase*) agar kalimat opini pada dataset memiliki bentuk atau jenis yang sama dan dapat memudahkan peneliti menentukan label untuk setiap data pada dataset karena data terlihat lebih rapi dan beraturan. Tabel 3.3 menampilkan contoh penerapan *casefolding* pada dataset.

Tabel 3. 3 Contoh proses *casefolding*

Sebelum	**Sesudah**
keturunan Minang jd Bapak Bangsa Negara Luar n nhoax Tuanku Abdul Rahman Tuanku Muhammad RM Malaysia n n Sultan Hasah valid	keturunan minang jd bapak bangsa negara luar n nhoax tuanku abdul rahman tuanku muhammad rm malaysia n n sultan hasah valid
Disitu OTAK DARIPADA NDESO BISANYA HANYA NGUTANG JUAL BOHONG NYUSAHIN RAKYAT	disitu otak daripada ndeso bisanya hanya ngutang jual bohong nyusahin rakyat
Pinter dan Persis Presiden JOKO WIDODO Pakdhe idola banget hingga ke luar negeri Top kami masyarakat Imdonesia bangga	pinter dan persis presiden joko widodo pakdhe idola banget hingga ke luar negeri top kami masyarakat imdonesia bangga
Pendeta Yason Yikwa memperoleh penghargaan dari Mensos Ia berjasa	pendeta yason yikwa memperoleh penghargaan dari mensos ia berjasa

melindungi warga non Papua saat kerusuhan terjadi di Wamen valid	melindungi warga non papua saat kerusuhan terjadi di wamen valid
Nasib profesi dokter era ini n nDibayar BPJS seharga parkir mobil nDibakar separatis saat kerusuhan Wamena nDipukuli aparat saat valid	nasib profesi dokter era ini n ndibayar bpjs seharga parkir mobil ndibakar separatis saat kerusuhan wamena ndipukuli aparat saat valid

2.2.3 Normalisasi *slang-word*

Slang-word adalah tahapan untuk mengubah kata-kata gaul menjadi kata baku yang sesuai dengan bahasa indonesia pada kalimat opini. Penggunaan normalisasi *slangword* berfungsi untuk membuat teks opini lebih jelas saat proses deteksi teks. Tabel 3.4 menampilkan contoh penerapan normalisasi *slang*. Penerapan normalisasi *slang* pada dataset yaitu merubah kata-kata yang kurang baku menjadi bentuk yang lebih baku.

Tabel 3.4 Contoh normalisasi *slang-word*

Sebelum	Sesudah
Lu tu idiot ato emg dungu dr kecil ya kaga paham" klo org ngomong	kamu itu idiot atau memang dungu dari kecil iya tidak paham-paham kalau orang bicara
heboh bet bantah sndiri klo ga bego ya bkn cebong nmanya	heboh banget bantah sendiri kalau tidak bego iya bukan cebong namanya
temen yg di ig kadang so ngartis	temen yang di instagram kadang sok ngartis

2.2.4 Menghapus *Stopword*

Setelah melalui proses *cleaning*, *casefolding*, dan normalisasi *slangword*, kemudian teks akan diproses dengan menghapus *stopword*. *Stopword* berisi daftar kumpulan kata yang dinilai kurang penting. Tabel 3.5 menampilkan contoh penghapusan *stopword* pada dataset. Tahap ini bertujuan untuk menghapus kata-kata yang dinilai tidak memiliki makna. Tahap ini dilakukan dengan memecah kata pada kalimat menjadi unit-unit yang dinamakan token untuk dilakukan pembobotan pada tahap selanjutnya.

Tabel 3. 5 Contoh penghapusan *stopword*

Sebelum	Sesudah
dia tidak sadar bahwa lebih kampungan dari si mc yang dia sendiri mengatakan bego	dia tidak sadar bahwa kampungan mc yang dia sendiri mengatakan bego

eman balas chat makan teman tidur saja kurang dibalas jangan mengatakan sombong kurang aktif kamu kalau tidak paham	eman balas chat makan teman tidur saja dibalas jangan katakan sombong aktif kamu tidak paham
alhamdulillah terima kasih iya rob orang sombong tidak berhak menjadi wakil rakyat	terima kasih orang sombong tidak berhak menjadi wakil rakyat

2.3 Pelabelan Data

Dataset yang sudah melalui proses *pre-processing* dan data sudah memenuhi kriteria, selanjutnya peneliti melakukan *labeling.* Adapun untuk pelabelan data dilakukan secara manual dilakukan dengan mengidentifikasi pola dari kalimat berdasarkan pasal-pasal pelanggaran terhadap Undang-Undang ITE. Adapun kalimat yang mengandung pornografi diberi label 1 sesuai dengan pasal 27 ayat 1 tentang pornografi. Label 2 berdasarkan pasal 28 Ayat 1 tentang hoaks atau berita bohong yang merujuk kepada sistem pelabelan yang telah dilakukan pada penelitian (Atmajaya et al. 2022). Label 3 berdasarkan pasal 45B tentang *cyberbulling* atau perundungan sosial. Label 4 untuk pasal 28 ayat 2 tentang ujaran kebencian SARA serta label 0 untuk kalimat yang bukan termasuk sanksi pidana atau netral.

Dalam proses pelabelan data terdapat berbagai kendala yang dihadapi, diantaranya sulitnya untuk melakukan identifikasi kalimat yang mengandung konten hoaks. Dalam pelabelan hoaks tersebut diperlukan pengecekan terhadap keakuratan data apakah termasuk kedalam hoaks atau tidak. Namun, proses pelabelan hoaks tersebut masih memungkinkan untuk dilakukan dengan menggunakan sistem pelabelan hoaks yang dilakukan oleh (Atmajaya et al. 2022) dalam penelitian yang berjudul ITE Law Enforcement Support through Detection Tools of Fake News, Hate Speech, and Insults in Digital Media. Sehingga berdasarkan hal tersebut label tentang berita bohong atau hoaks tetap dimasukkan kedalam penelitian ini.

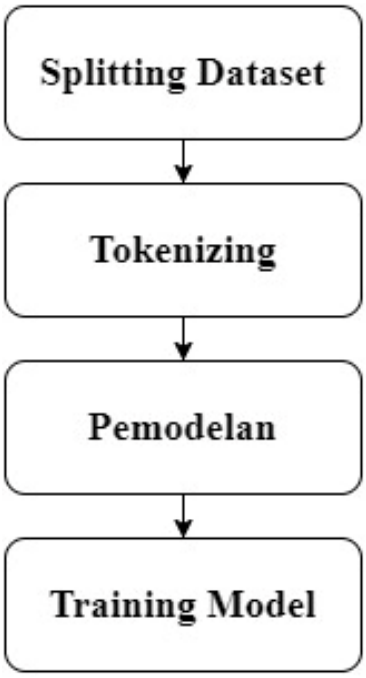
Setelah dataset dilabeli, kemudian dataset tersebut dicek untuk validasi yang dilakukan oleh seorang mahasiswa hukum. Apabila ditemukan label yang tidak sesuai maka akan diganti dengan label yang sesuai. Dalam penggunaan dataset untuk membangun model klasifikasi tidak menggunakan semua data yang diperoleh, hal ini disebabkan karena keterbatasan dalam proses pelabelan serta bertujuan agar data yang digunakan menjadi seimbang sehingga memiliki performa model yang seimbang. Berikut adalah contoh kalimat yang sudah dilabeli pada dataset yang dapat dilihat pada tabel 3.6 berikut.

Tabel 3.6 Contoh pelabelan data

Kalimat	Label
saat konferensi pers kemenangan ini prabowo ditemani beberapa orang tak ada sandiaga uno	0
abg cantik mesum sama pacar dipinggir jalan	1
gedung putih dibanjiri masa profesional ahok dan menuntut fpi dibubarkan	2
woi kecoa kamu bisa baca dari atas jangan terlalu dungu iya dasar kecoa	3
si cina babi ahok bloon banget sok pintar banget lagi asal bukan ahok	4

2.4 Membangun Model Klasifikasi

Setelah dataset dikumpulkan, dilanjutkan dengan pre-processing, kemudian dilakukan pelabelan pada dataset, langkah selanjutnya adalah membangun model klasifikasi. Pada proses ini dilakukan langkah-langkah yang terdiri dari *splitting* dataset, *tokenizing*, pemodelan, dan *training* model. Gambar 3.3 menunjukkan tahapan dari membangun model klasifikasi.

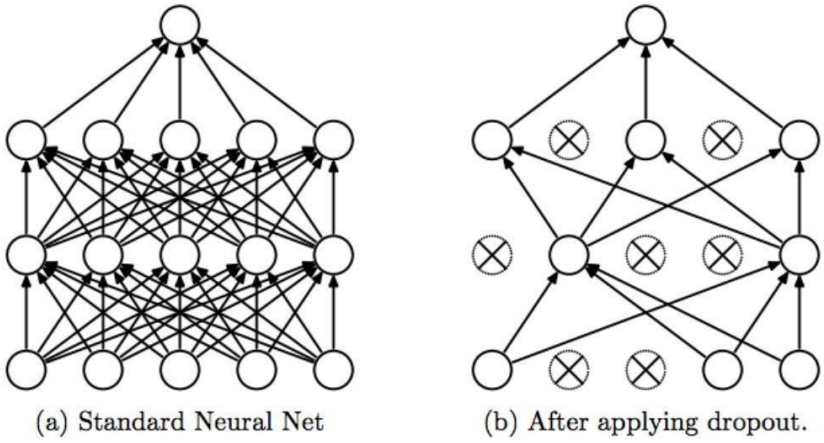


Gambar 3.3 Tahapan membangun model klasifikasi

Langkah pertama yang dilakukan adalah *splitting* atau membagi dataset menjadi data latih (*train*), data uji (*test*), dan data validasi(*validation*). Data latih digunakan untuk melatih

model agar dapat memahami data. Data uji digunakan untuk menguji model serta mendapat metrik evaluasi seperti *Accuracy, Presicion, Recall,* dan *F1-Score* yang akan dievaluasi pada tahap berikutnya. Data validasi digunakan untuk melakukan validasi hasil prediksi model saat melakukan pengujian.

Langkah kedua adalah *tokenizing* yang bertujuan untuk memecah teks menjadi token agar mendapatkan kata-kata unik yang akan digunakan sebagai kamus bahasa atau kosakata. Selanjutnya, langkah ketiga adalah membangun (*create*) model LSTM dan BiLSTM dan dilanjutkan dengan melakukan pengujian model. Dalam proses pengujian model sering kali terjadi *overfitting* yang berakibat pada hasil kinerja model. *Overfitting* pada model terjadi karena model yang dibangun terlalu kompleks dan fokus pada *training* dataset tertentu, sehingga tidak dapat melakukan prediksi dengan akurat pada dataset lain yang serupa, oleh karena itu, penelitian ini menerapkan metode dropout guna meminimalisir terjadinya *overfitting*. Metode dropout ini dapat mengatasi *overfitting* dengan cara menghapus Sebagian kontribusi neuron saat aktivasi. Gambar 3.4 menggambarkan perbedaan antara menggunakan dropout dengan tidak menggunakan dropout terhadap model.

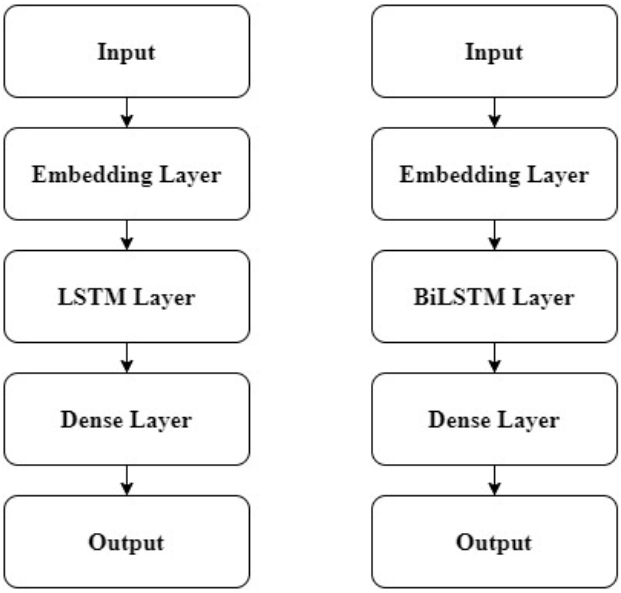


Gambar 3.4 Perbedaan penggunaan dropout

Sumber: towardsdatascience.com

Algoritma LSTM dan BiLSTM digunakan dalam proses ini. Ada tiga lapisan yang digunakan dalam membangun model tersebut yaitu lapisan *embedding,* lapisan LSTM atau lapisan BiLSTM, dan lapisan *dense*. Lapisan e*mbedding* digunakan untuk mengubah data latih menjadi vektor numerik yang merepresentasikan kedekatan makna tiap kata atau *word embedding*. Lapisan LSTM digunakan untuk menjalankan fungsi LSTM, sedangkan lapisan BiLSTM terdiri dari dua lapisan LSTM yaitu lapisan LSTM maju (*forward*) dan lapisan LSTM mundur (*backward*) sehingga kombinasi tersebut dapat menangkap informasi dari kedua arah.

Sementara lapisan *dense* digunakan sebagai lapisan *output*. Gambar 3.5 menampilkan alur kerja pada model.



Gambar 3.5 Alur kerja model

Pada tahap ini dibuat beberapa skenario untuk memperoleh model dengan performa terbaik yang digunakan untuk melakukan pendeteksian konten pelanggaran Undang-Undang ITE. Tabel 3.7 dan Tabel 3.8 menampilkan daftar skenario hyperparameter model pada penelitian ini.

Tabel 3.7 Skenario hyperparameter model LSTM

Activation	Optimizer	LSTM Unit	Batch Size	Epoch	Dropout
Softmax	Adam	64, 64	32	20	-
Softmax	Adam	64, 64	32	20	0.9
Softmax	Adam	64, 64	32	25	0.9
Softmax	Adam	64, 64	64	20	0.9
Softmax	Adam	64, 64	64	25	0.9

Tabel 3.8 Skenario hyperparameter model BiLSTM

Activation	Optimizer	LSTM Unit	Batch Size	Epoch	Dropout
Softmax	Adam	64, 64	32	20	-
Softmax	Adam	64, 64	32	20	0.9
Softmax	Adam	64, 64	32	25	0.9
Softmax	Adam	64, 64	64	20	0.9
Softmax	Adam	64, 64	64	25	0.9

Pemilihan hyperparameter yang digunakan pada penelitian ini merujuk kepada hyperparameter yang digunakan pada penelitian (Hesaputra et al., 2022) namun dengan beberapa modifikasi. Pada penelitian tersebut menggunakan LSTM unit 128, 128 sedangkan pada penelitian ini menggunakan LSTM unit yang lebih sedikit dengan tujuan mengurangi kompleksitas *layer* atau lapisan pada model dikarenakan jumlah data yang digunakan lebih sedikit. Kemudian untuk jumlah *batch size* yang digunakan tidak memiliki perbedaan. Kemudian untuk jumlah *epoch* yang digunakan pada penelitian ini lebih banyak dimana pada penelitan tersebut hanya menggunakan 5 *epoch* sedangkan penelitian ini menggunakan 20 dan 25 *epoch* dengan tujuan dengan jumlah LSTM unit serta data yang lebih sedikit mendapatkan hasil yang lebih baik juga memiliki performa yang seimbang (*good fitting*). Kemudian untuk skenario yang menggunakan dropout memiliki perbedaan dimana pada penelitian tersebut menggunakan dropout yaitu 0.9 dan 0.5 untuk masing-masing skenario. Sementara pada penelitian ini menggunakan dropout 0.9 untuk masing-masing skenario dengan maksud mengurangi kompleksitas model sehingga memiliki hasil yang lebih baik dan seimbang (*good fitting*).

2.5 Evaluasi

Setelah melatih model, selanjutnya dilakukan evaluasi terhadap model. Evaluasi digunakan untuk mengukur performa dan kinerja dari model yang telah dilatih. Tahap evaluasi ini dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* berguna untuk merangkum kinerja model dalam mengevaluasi efektivitas model klasifikasi sehingga peneliti dapat mengetahui *error* pada operasi algoritma yang dijalankan (Fudholi, 2022). Gambar 3.6 menampilkan *multiclass confusion matrix* yang digunakan pada penelitian ini.

		Predicted Class			
		C_1	C_2	....	C_N
Actual Class	C_1	TN	FP	TN	TN
	C_2	FN	TP	FN	FN
	....	TN	FP	TN	TN
	C_N	TN	FP	TN	TN

Gambar 3.6 *Multiclass confusion matrix*

Berdasarkan *confusion matrix* maka dapat diperoleh nilai performa dari metrik evaluasi yang mengacu kepada *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Nilai metrik evaluasi tersebut dihitung dan digunakan sebagai nilai ukur evaluasi terhadap model. Pada Gambar 3.6 diatas terdiri dari lebih dari dua kelas yang merepresentasikan hasil kelas aktual dan kelas prediksi.

BAB IV
HASIL DAN PEMBAHASAN

3.1 Dataset

Dalam penelitian ini, data yang digunakan merupakan kombinasi dari data primer dan data sekunder. Data primer diambil menggunakan teknik *scrapping* menggunakan Twitter API dengan memanfaatkan *library tweepy* untuk mengakses data *tweet*. Proses pengumpulan data dilakukan berdasarkan akun dan kata kunci dari Twitter. Proses tersebut dilakukan pada tanggal 18 Maret 2022. Dari proses tersebut, data yang telah berhasil terkumpul berjumlah 17.384 data dan dijadikan kedalam bentuk file *excel*. Kemudian untuk data sekunder diperoleh dari penelitian yang telah dilakukan sebelumnya. Gambar 4.1 menampilkan Contoh data yang berhasil terkumpul.

12	Abg sange montok main colmek saat rumah sepi#bokepindo #abgsange #colmek #bokeplokal #bokepsma https://t.co/YBSLqv4Jqa
13	@ichaayureal halo semua nya... salkenfollow @BiroSeks update cerita dewasa tiap harinya..muach.. 😘😘
14	"RT @HDouboegiez: Ngabalin: Pemerintahan Jokowi Dibully Buzzer\n\nYang pelihara buzzer siapa, yg si bully siapa, ngakak so hard\n#DewanPerampok\u2valid26"
15	"RT @Novirhmwtihoax: Plis twit do your majic ini di kampung aku ini kebakaran hutan dan sangat dekat sama perumahan sampe dekarang api masih be\u2valid26"
16	"@dikychandra_ Ini ya siDewan yg gila Hormat itu.. mirip \ud83d\ude49 gabon. Berisik, bau busuk. #DewanPengkhianatRakyat #dewanperampokrakyat"
17	"RT @YantiFiqri: Angkat tagar barisan rakyat sejati #DewanPerampokRakyat \n#DewanPerampokRakyat\n#DewanPerampokRakyat https:\/\/t.co\/NsjRZBQ5QO"
18	"RT @Mia_Andy__: @Aiek_EsThreeM Yang bikin ini pinter and kreatif\n\n#DewanPerampokRakyat https:\/\/t.co\/l54QJvKUKH"
19	@setkabgoid @PutraWadapi @jokowi Apa yg disampaikan jokowi dia langgar sendiri..kualitas kepemimpinan jokowi lemah..
20	"RT @kompascom: Kondisi kabut asap ekstrem saat ini sedang melanda Kota Palembang, Sumsel,. Asap berasal dari kebakaran hutan dan lahan di S\u2valid26"
21	Lengan tangan abg ini putih mulus banget guys 😍#lengantangan #abghot #perawan #seksi #semok #binal #igo #indo https://t.co/CLWhg8086j
22	Wow, geraknya bisa naik turun 😍#bigolive #dessy #mulus #seksi #cantik https://t.co/hWlp7OBiSj
23	Jilbab cantik main sama kursi #jilbab #manis #unyu #kursi https://t.co/0KTXjVqtOa
24	Bagian tubuh yang paling kalian suka dari aku itu apa?#cerewet #cantikcantik #manis #igo #cuek https://t.co/rTBebBzqQo
25	Abg lugu tapi mau buka bukaan..#abgngentot #Abghot #buka #open #seksi https://t.co/GsILQd6wnr
26	Oh ademnya lihat cewek kayak gini...#jilbab #cantikcantik #manis #manja #imut https://t.co/JO8T9dsJNr
27	Enak susunya ma ma ma 🤭#abgseksi #pahamulus #cantik #montok #semok https://t.co/KnpfktUjha
28	"RT @hipohan: Nasib profesi dokter era ini\n\nDibayar BPJS seharga parkir mobil\nDibakar separatis saat kerusuhan Wamena\nDipukuli aparat saat d\u2valid26"
29	Pembantuku ternyata nafsuin dan bisa memuaskanku#bokephot #SANGE_AAAAAAAAAH #videongentot #bokepviral https://t.co/JGal6EWxOo
30	@jokowi Kapan buat masyarakat umumnya pak presiden..... kami juga butuh aktifitas diluar rumah buat menyambung hidup pak.....

Gambar 4. 1 Contoh data yang terkumpul

3.2 *Pre-processing*

Setelah dataset terkumpul, kemudian dilanjutkan dengan proses *pre*-processing. Proses ini diperlukan untuk mengurangi *noise* pada data dan data yang tidak konsisten. Tahapan ini memanfaatkan bahasa pemrograman Python. Berikut merupakan penjelasan dari setiap langkah proses *pre-processing* yang dilakukan.

3.2.1 *Cleaning* Data

Setelah dataset terkumpul, selanjutnya akan dilakukan *cleaning* data. *Cleaning* data berfungsi untuk membersihkan data *tweet* dari berbagai *noise* seperti simbol, tanda baca, dan lainnya. Adapun untuk implementasi kode program yang digunakan pada proses cleaning data ini ditunjukkan pada Gambar 4.2.

```
import re
import string
import unicodedata
import nltk
import pandas as pd

def remove_url(str):
    str = re.sub(r'(?i)\b((?:https?://|www\d{0,3}[.]|[a-z0-9.\-]+[.][a-
z]{2,4}/)(?:[^\s()<>]+|\(([^\s()<>]+|(\([^\s()<>]+\)))*\))+(?:\(([^\s()<>]+|(\([^
\s()<>]+\)))*\)|[^\s`!()\[\]{};:\'".,<>?«»“”‘’]))', '', str)
    return str

def remove_digit(str):
    str = re.sub(r'[^a-z ]*([.0-9])*\d', ' ', str)
    return str

def remove_non_ascii(str):
    str = unicodedata.normalize('NFKD', str).encode('ascii',
'ignore').decode('utf-8', 'ignore')
    return str

def remove_twitter_char(str):
    # mention
    str = re.sub(r'(?:@[\w_]+)', ' ', str)
    # hashtag
    str = re.sub(r"(?:\#+[\w_]+[\w\'_\-]*[\w_]+)", " ", str)
    # RT/cc
    str = re.sub('RT', ' ', str)
    return str

def remove_punctuation(str):
    str = re.sub(r'[^\s\w]', ' ', str)
    return str

def remove_multi_space(str):
    str = re.sub('[\s]+', ' ', str)
```

```
    return str

def remove_repeated_character(str):
    str = re.sub(r'(.)\1{2,}', r'\1', str)
    return str

def remove_laugh(str):
    str =
re.sub(r"\b(?:(h|a|e)*(?:(ha|he|hue))+h?|(?:l+o+)+l+)|(?:(w|k)*(?:wk)+(w?|k?))\b"
, ' ', str)
    return str
```

Gambar 4. 2 Kode program *cleaning* data

3.2.2 *Casefolding*

Setelah dilakukan proses *cleaning* data, selanjutnya dilakukan *casefolding*. Proses ini bertujuan untuk memudahkan peneliti menentukan label pada setiap data pada dataset sehingga terlihat lebih teratur dan rapi. Adapun untuk implementasi kode program yang digunakan pada proses *casefolding* ini ditunjukkan pada Gambar 4.3.

```
import re
import string
import unicodedata
import nltk
import pandas as pd

def casefolding(str):
    str = str.lower()
    return ' '.join(str.split())
```

Gambar 4. 3 Kode program *casefolding*

3.2.3 Normalisasi *Slang-word*

Dalam sebuah kalimat terdapat kata-kata yang kurang bermakna dan bahasa yang tidak baku, maka dari itu kata tersebut harus dihilangkan dan diubah kedalam bentuk bahasa yang lebih baku. Tahapan ini memerlukan kamus daftar bahasa tidak baku dengan basis bahasa indonesia. Untuk menerapkannya peneliti perlu menyesuaikan leksikonnya ke dalam bahasa indonesia. Adapun untuk implementasi kode program yang digunakan pada proses normalisasi *slang-word* ini ditunjukkan pada Gambar 4.4.

```
import re
import string
import unicodedata
import nltk
import pandas as pd

def normalize_slang_word(str):
    text_list = str.split(' ')
    slang_words_raw = pd.read_csv('data/add/slang_word_list.csv', sep=',',
header=None)
    slang_word_dict = {}

    for item in slang_words_raw.values:
        slang_word_dict[item[0]] = item[1]

        for index in range(len(text_list)):
            if text_list[index] in slang_word_dict.keys():
                text_list[index] = slang_word_dict[text_list[index]]

    return ' '.join(text_list)
```

Gambar 4. 4 Kode program normalisasi *slang word*

3.2.4 Menghapus *Stopword*

Stopword berfungsi untuk efisiensi data dengan cara menghilangkan kata-kata yang kurang bermakna. Pada tahap ini dibuat daftar kata yang dinilai kurang penting kedalam sebuah file yang digunakan sebagai kamus. Adapun untuk implementasi kode program yang digunakan pada proses menghapus *stopword* ini ditunjukkan pada Gambar 4.5.

```
import re
import string
import unicodedata
import nltk
import pandas as pd
from nltk import word_tokenize, sent_tokenize
from nltk.corpus import stopwords

def remove_stopword(str):
    stop_words = pd.read_csv('data/add/slang_word_list.csv', sep=',',
header=None)
    word_tokens = word_tokenize(str)
    filtered_sentence = [w for w in word_tokens if not w in stop_words]
```

```
    return ' '.join(filtered_sentence)
```

Gambar 4.5 Kode program Menghapus s*topword*

3.2.5 Hasil *pre-processing*

Setelah melakukan *pre-processing* pada dataset, selanjutnya data sudah dapat diproses ke tahap berikutnya. Tabel 4.1 menampilkan hasil dari tahap *pre-processing* pada dataset. Data yang sudah dipreprocessing terlihat lebih rapi dan tertata, sehingga memudahkan untuk dilakukan pelabelan pada proses berikutnya.

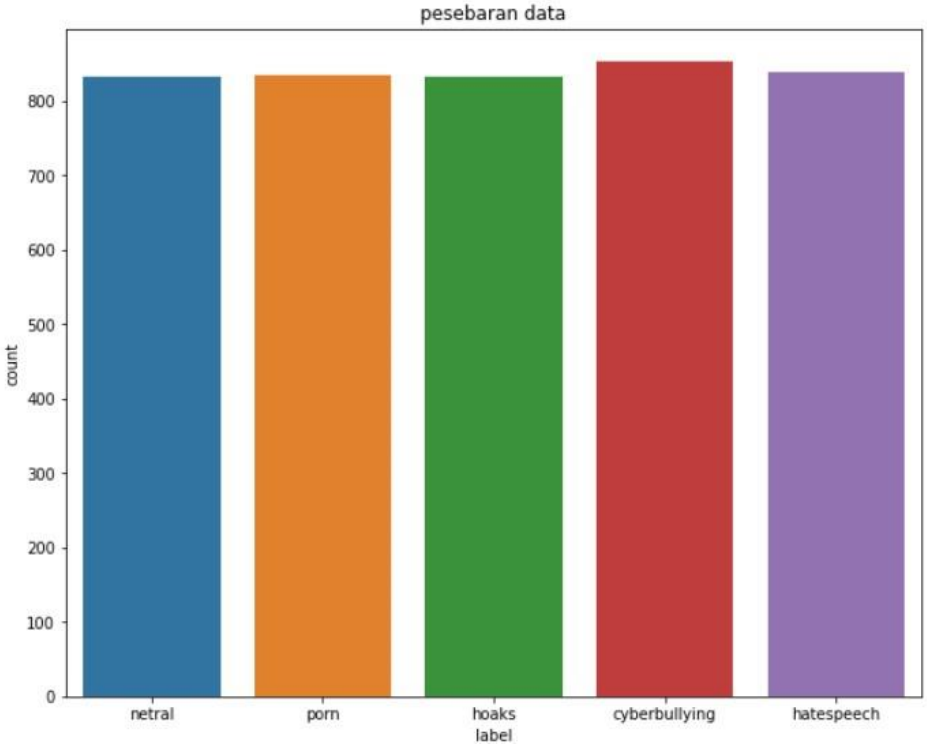
Tabel 4. 1 Hasil *pre-processing*

Sebelum	**Sesudah**
"RT @MarawaBungHaris: 3 keturunan Minang jd Bapak Bangsa Negara Luar\n\nhoax Tuanku Abdul Rahman Tuanku Muhammad (RM Malaysia)\n\n2 Sultan Hasah B\u2valid26"	keturunan minang jadi bapak bangsa negara luar dan nhoax tuanku abdul rahman tuanku muhammad rm malaysia dan dan sultan hasah
@detikcom @detikhot Cerita Seks Dewasa Terbaru : Kenikmatan tubuh anak gadis majikanku #ceritasex #ceritadewasa #ceritalendir #ceritaseks #cerpendewasa #cerpenseks #cewek #TimnasDay #konsersabyan #RezimPanikHarga2Naik #KPKWajibIndependen #90LDENERA #toyotaiampossible https://t.co/WkZnaXvK2a https://t.co/1QiJVrKhWP	cerita seks dewasa terbaru kenikmatan tubuh anak gadis majikanku
Lengan tangan abg ini putih mulus banget guys 😍	lengan tangan abg ini putih mulus banget teman-teman

#lengantangan #abghot #perawan #seksi #semok #binal #igo #indo https://t.co/CLWhg8086j	
Abg lugu tapi mau buka bukaan.. #abgngentot #Abghot #buka #open #seksi https://t.co/GsILQd6wnr	abg lugu tapi mau buka bukaan

3.3 Pelabelan Data

Setelah dataset dikumpulkan, dilanjutkan dengan dilakukan preprocessing data, Langkah selanjutnya adalah pelabelan data. Dari hasil yang didapat setelah proses preprocessing, kemudian dilakukan pelabelan manual berdasarkan kriteria yang telah ditentukan dan divalidasi oleh seorang mahasiswa hukum. Setelah dilabeli dan divalidasi, maka didapat dataset sebanyak 4193 data yang terdiri dari 833 data netral, 834 data pornografi, 833 data *hoaks*, 854 data *cyberbullying*, dan 839 data *hate speech*. Gambar 4.6 menunjukkan sebaran data yang digunakan untuk klasifikasi.



Gambar 4. 6 Sebaran data

3.4 Membangun Model Klasifikasi

Setelah dataset dilabeli, tahapan selanjutnya adalah membangun model klasifikasi. Tahapan ini dilakukan untuk mendapatkan model terbaik dalam melakukan klasifikasi *tweet* yang mengandung pelanggaran Undang-Undang ITE. Tahapan-tahapan yang dilakukan dalam klasifikasi sebagai berikut:

3.4.1 *Splitting* Dataset

S*plitting* data adalah proses membagi dataset kedalam beberapa bagian yaitu, data latih (*train*), data uji (*testing*), dan data validasi. Gambar 4.7 merupakan kode program untuk membagi antara data train, data test, dan data validation. Dari total 4193 data yang telah dilabeli, dataset tersebut dibagi menjadi sebesar 6835 dibagi ke dalam 80% untuk data latih, 10% untuk data validasi dan 10% untuk data uji.

```
# Splitting Data
text = data['tweet'].values
label = data['label'].values
data_train, data_test, label_train, label_test =
train_test_split(text, label, test_size=0.2, random_state=42)
data_test, data_val, label_test, label_val =
train_test_split(data_test, label_test, test_size= 0.5,
random_state=42)
```

Gambar 4. 7 Kode program *splitting* data

3.4.2 *Tokenizing*

Tokenizing adalah proses memecah teks menjadi potongan unit-unit token sebelum dilakukan analisis lebih lanjut. Tujuan proses ini adalah untuk mendapatkan kata-kata unik yang akan digunakan sebagai kamus bahasa atau kosakata. Gambar 4.8 menampilkan kode program untuk *tokenizing* pada proses klasifikasi. Selain itu, juga terlihat jumlah *batch size* dan *epoch* yang digunakan masing-masing sebesar 32 dan 15.

```
# Tokenizer
tokenizer = Tokenizer()
tokenizer.fit_on_texts(text)

vocab = max([len(tokenizer.word_index)]) + 1 # kamus kata
maxlen = max([len(i.split()) for i in text]) # panjang input sequence
batch_size = 32 # penentuan jumlah sample yang ditraining pada tiap e
poch
num_epochs = 15 # banyak iterasi pada saat training model
```

```
initializer = initializers.RandomUniform(minval=-
0.05, maxval=0.05, seed=5)
```

Gambar 4. 8 Kode program *tokenizing*

3.4.3 *Create* Model

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 dapat digunakan 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*. Kemudian, setiap model dibangun menggunakan lapisan *embedding* dengan ukuran 128 dimensi. Setiap model juga menggunakan sebuah lapisan *dense* yang menggunakan fungsi aktivasi *softmax* serta menggunakan fungsi *loss categorical_crossentropy*.

Langkah selanjutnya adalah konfigurasi *layer compile* yang bertujuan agar konfigurasi *layer* dapat digunakan untuk proses *training*. Pada proses *compile* parameter yang digunakan yaitu *loss*, *optimizer* dan *metrics* yang terdiri dari *accuracy*, *precision*, dan *recall*. Fungsi *loss* digunakan untuk menghitung nilai yang harus diminimalkan selama pelatihan. Fungsi *optimizer* digunakan untuk mengatur respon model dalam estimasi *error* saat bobot model diperbarui. Sedangkan fungsi *metrics* digunakan untuk mengukur performa model. Dalam melakukan klasifikasi teks ini fungsi *optimizer* yang digunakan adalah *Adam*.

Gambar 4.9 dan Gambar 4.10 masing-masing menunjukkan kode program dan model algoritma LSTM tanpa dropout. Sedangkan kode program dan model algoritma BiLSTM tanpa menggunakan dropout ditunjukkan pada Gambar 4.11 dan Gambar 4.12. Selanjutnya untuk skenario menggunakan dropout, pada penelitian ini menggunakan dropout sebesar 0.9. Gambar 4.13 dan Gambar 4.14 masing-masing menunjukkan kode program dan model algoritma LSTM yang menggunakan dropout. Sedangkan untuk kode program dan model algoritma BiLSTM yang menggunakan dropout ditunjukkan pada Gambar 4.15 dan Gambar 4.16 berikut.

```
def get_model1(X, Y):
    model = Sequential()
    model.add(Embedding(input_dim = vocab, output_dim = 128, input_le
ngth = maxlen, embeddings_initializer = initializer))
    model.add(LSTM(64, recurrent_initializer = initializer, kernel_in
itializer = initializer, return_sequences=True))
    model.add(LSTM(64))
```

```
    model.add(Dense(5, activation='softmax', kernel_initializer = ini
tializer))
    model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=METRICS)
    print(model.summary())

    return model
```

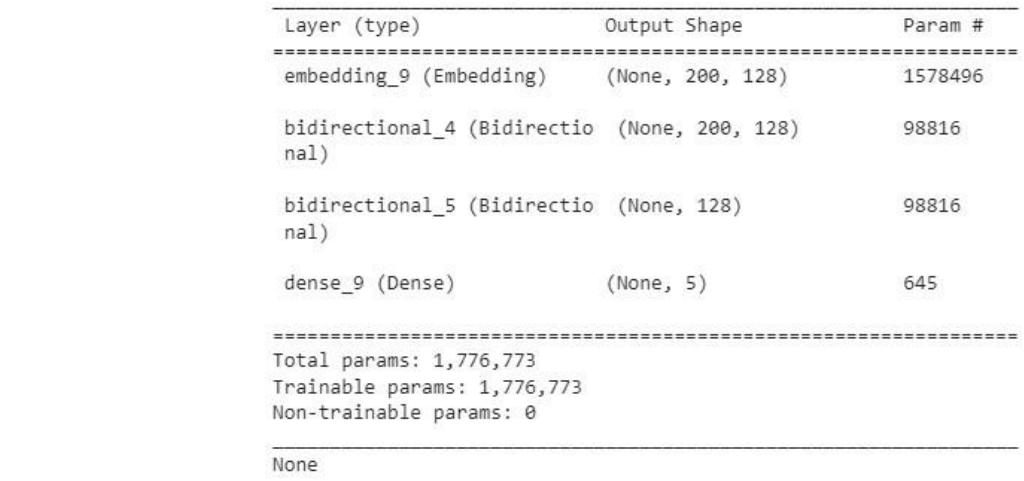
Gambar 4. 9 Kode program LSTM tanpa dropout

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_8 (Embedding)     (None, 200, 128)          1578496

 lstm_16 (LSTM)              (None, 200, 64)           49408

 lstm_17 (LSTM)              (None, 64)                33024

 dense_8 (Dense)             (None, 5)                 325

=================================================================
Total params: 1,661,253
Trainable params: 1,661,253
Non-trainable params: 0
_________________________________________________________________
None
```

Gambar 4.10 Model arsitektur LSTM tanpa dropout

```
def get_model2(X, Y):
    model = Sequential()
    model.add(Embedding(input_dim = vocab, output_dim = 128, input_le
ngth = maxlen, embeddings_initializer = initializer))
    model.add(Bidirectional(LSTM(64, recurrent_initializer = initiali
zer, kernel_initializer = initializer, return_sequences=True)))
    model.add(Bidirectional(LSTM(64)))
    model.add(Dense(5, activation='softmax', kernel_initializer = ini
tializer))
    model.compile(loss='binary_crossentropy', optimizer='adam', metri
cs=METRICS)
    print(model.summary())

    return model
```

Gambar 4.11 Kode program BiLSTM tanpa dropout



Layer (type)	Output Shape	Param #
embedding_9 (Embedding)	(None, 200, 128)	1578496
bidirectional_4 (Bidirectional)	(None, 200, 128)	98816
bidirectional_5 (Bidirectional)	(None, 128)	98816
dense_9 (Dense)	(None, 5)	645

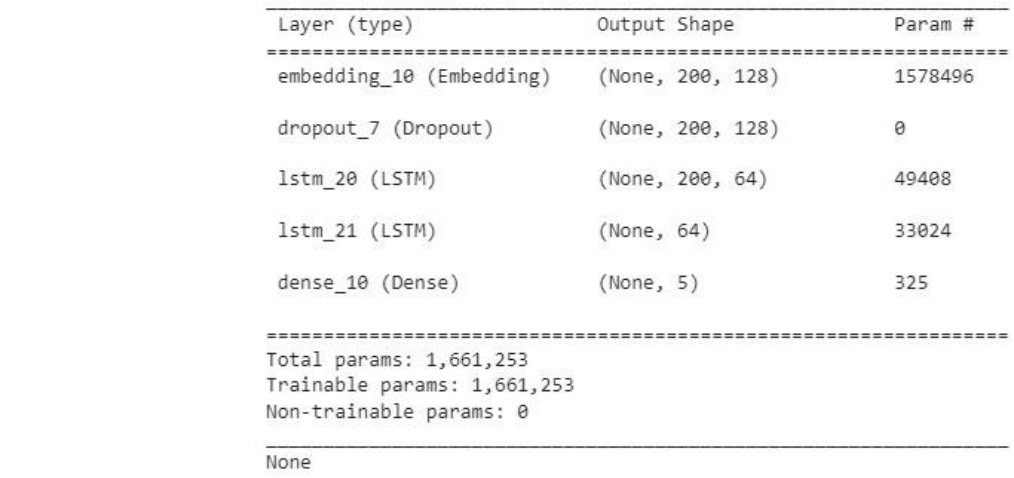
Total params: 1,776,773
Trainable params: 1,776,773
Non-trainable params: 0

None

Gambar 4.12 Model arsitektur BiLSTM tanpa dropout

```
def get_model3(X, Y):
    model = Sequential()
    model.add(Embedding(input_dim = vocab, output_dim = 128, input_le
ngth = maxlen, embeddings_initializer = initializer))
    model.add(Dropout(0.9))
    model.add(LSTM(64, recurrent_initializer = initializer, kernel_in
itializer = initializer, return_sequences=True))
    model.add(LSTM(64))
    model.add(Dense(5, activation='softmax', kernel_initializer = ini
tializer))
    model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=METRICS)
    print(model.summary())

    return model
```

Gambar 4.13 Kode program LSTM dengan dropout



Layer (type)	Output Shape	Param #
embedding_10 (Embedding)	(None, 200, 128)	1578496
dropout_7 (Dropout)	(None, 200, 128)	0
lstm_20 (LSTM)	(None, 200, 64)	49408
lstm_21 (LSTM)	(None, 64)	33024
dense_10 (Dense)	(None, 5)	325

Total params: 1,661,253
Trainable params: 1,661,253
Non-trainable params: 0

None

Gambar 4.14 Model arsitektur LSTM dengan dropout

```
def get_model4(X, Y):
    model = Sequential()
    model.add(Embedding(input_dim = vocab, output_dim = 128, input_le
ngth = maxlen, embeddings_initializer = initializer))
    model.add(Dropout(0.9))
    model.add(Bidirectional(LSTM(64, recurrent_initializer = initiali
zer, kernel_initializer = initializer, return_sequences=True)))
    model.add(Bidirectional(LSTM(64)))
    model.add(Dense(5, activation='softmax', kernel_initializer = ini
tializer))
    model.compile(loss='binary_crossentropy', optimizer='adam', metri
cs=METRICS)
    print(model.summary())

    return model
```

Gambar 4. 15 Kode program BiLSTM dengan dropout

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_11 (Embedding)     (None, 200, 128)          1578496

dropout_8 (Dropout)          (None, 200, 128)          0

bidirectional_6 (Bidirectio  (None, 200, 128)          98816
nal)

bidirectional_7 (Bidirectio  (None, 128)               98816
nal)

dense_11 (Dense)             (None, 5)                 645

=================================================================
Total params: 1,776,773
Trainable params: 1,776,773
Non-trainable params: 0
_________________________________________________________________
None
```

Gambar 4. 16 Model arsitektur BiLSTM dengan dropout

3.4.4 *Training* Model

Proses *training* pada model dilakukan menggunakan fungsi *fit*. Parameter yang digunakan adalah x untuk input data, y untuk target data, *epoch* yang merupakan jumlah tahapan dalam proses pelatihan data untuk melatih model, *batch_size* yang merupakan jumlah sampel yang akan dilatih setiap *epoch*, validation_data berfungsi untuk digunakan sebagai data validasi, *verbose* berfungsi untuk menampilkan progress bar dari setiap *epoch*. Gambar 4.17 menampilkan kode program untuk menjalankan *training* model. Pengujian tiap skenario dijalankan menggunakan *batch size* dan *epoch* berdasarkan skenario yang telah ditentukan

sebelumnya. Pada kode program tersebut, terdapat kode *batch size* dan *epoch* berwarna hijau yang dapat disesuaikan dengan skenario yang ditetapkan.

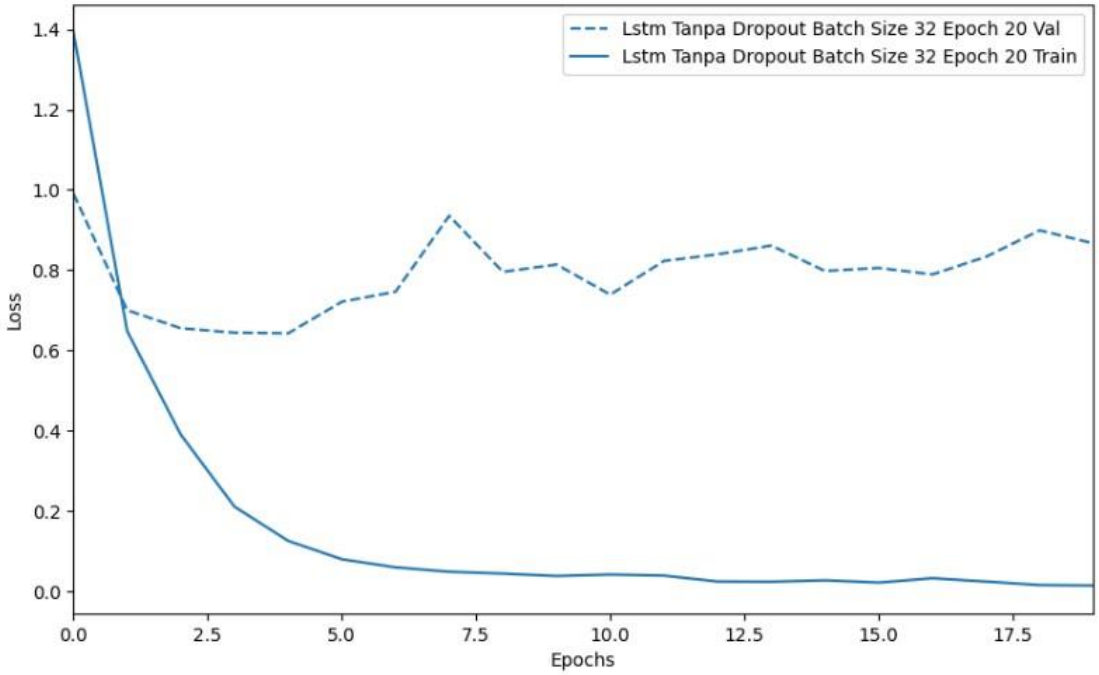
```
# training model
history = model.fit(X_train, Y_train, batch_size=batchsize,
epochs=epoch, verbose=1, validation_data=(X_val, Y_val))
```

Gambar 4.17 Kode program *training*

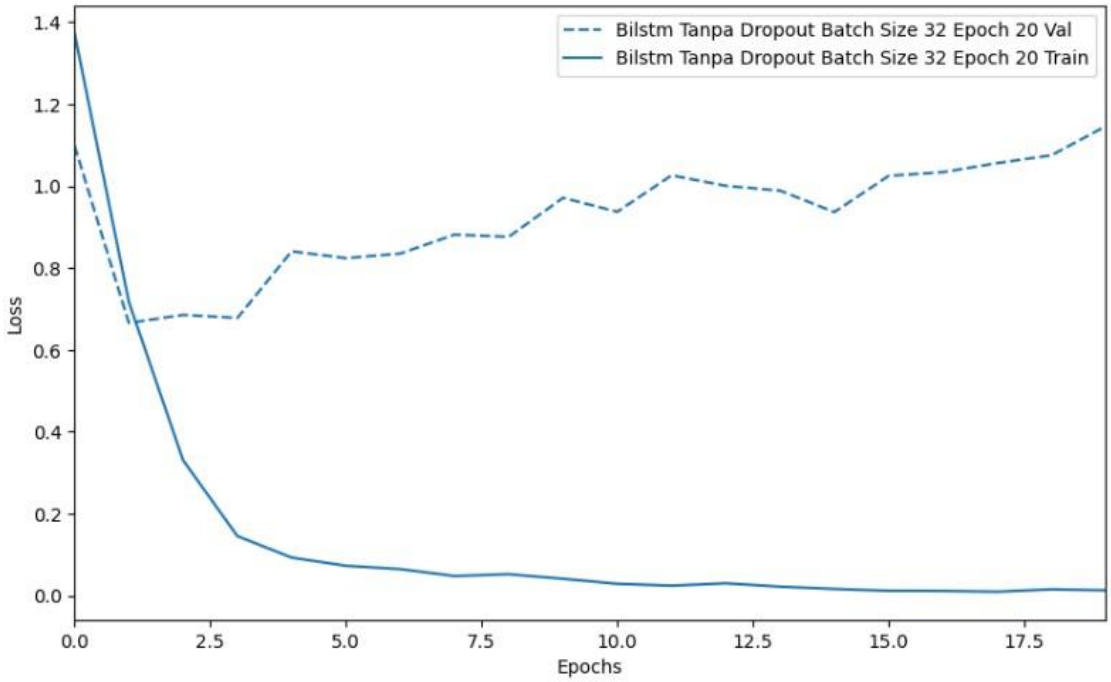
3.5 Evaluasi

Setelah model dilatih proses selanjutnya adalah evaluasi. Evaluasi merupakan tahap yang bertujuan untuk mengukur performa serta kinerja yang dihasilkan dari model. Pada tahap ini dilakukan dengan menguji model yang telah dilatih kedalam data uji dan validasi. Data uji dan data validasi yang digunakan untuk menguji kinerja model masing-masing 10% dari keseluruhan total data.

Dari proses *train* data, dapat dilihat performa tiap model menggunakan grafik *loss function* tiap model. Grafik *loss function* berguna untuk mengukur performa model yang dihasilkan dalam melakukan prediksi dalam melakukan kesalahan. Berdasarkan grafik *loss function* ini, dapat dilihat apakah model memiliki performa yang *good fitting*, *overfitting* atau *underfitting*. Pada awal percobaan dilakukan dengan menjalankan skenario tanpa dropout pada algoritma LSTM dan BiLSTM. Gambar 4.18 dan Gambar 4.19 menampilkan grafik *training* model tanpa dropout pada algoritma LSTM dan BiLSTM.



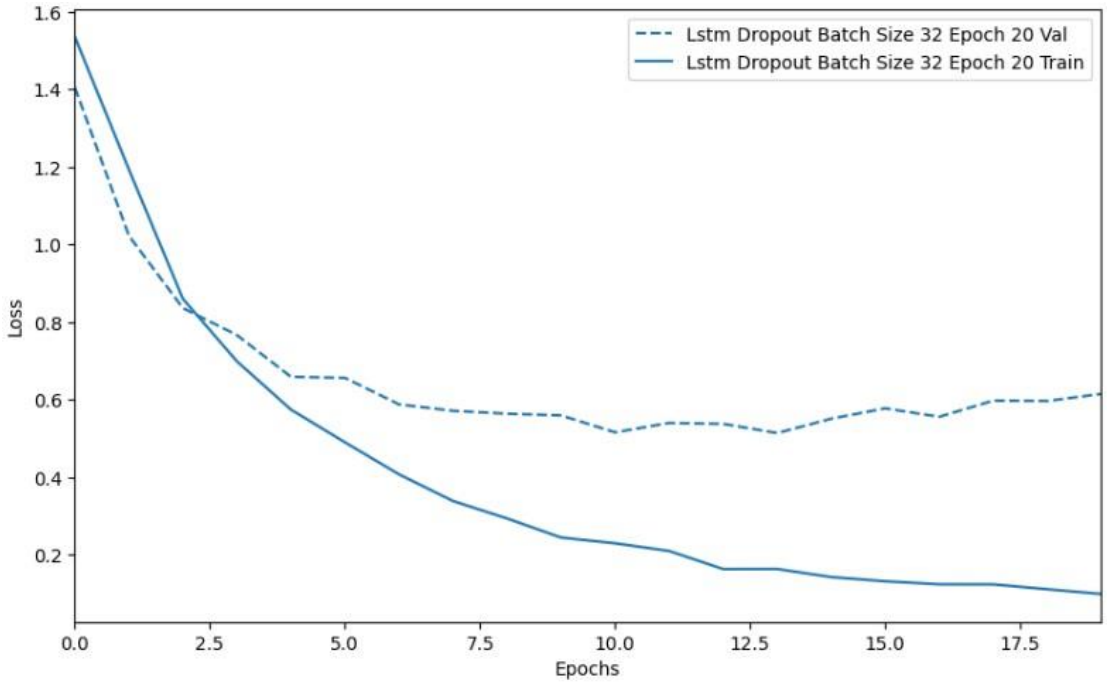
Gambar 4.18 Grafik *training* Model LSTM tanpa dropout



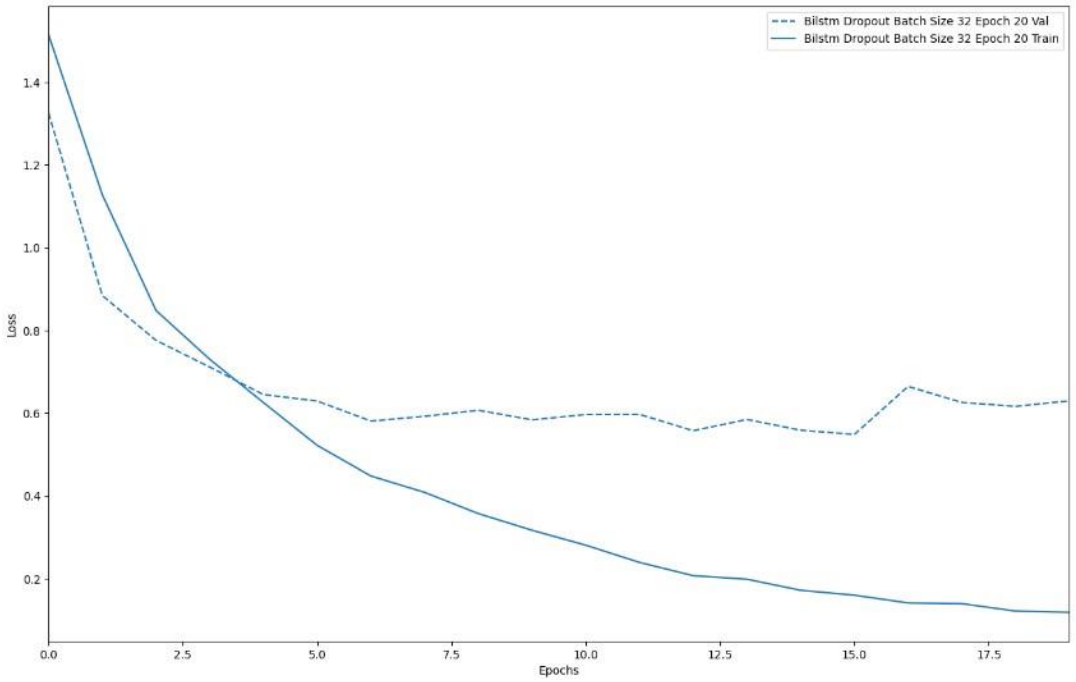
Gambar 4.19 Grafik *training* Model BiLSTM tanpa dropout

Pada Gambar 4.18 dan Gambar 4.19 diatas terlihat bahwa performa *loss function* yang *overfitting*. Gambar 4.18 menunjukkan performa *loss function* pada data *validation* yang terlihat menurun hingga *epoch* ke-4, namun pada *epoch* ke-5 performa *loss function* terus meningkat dan tidak stabil sehingga mengakibatkan performa pada model tersebut *overfitting*.

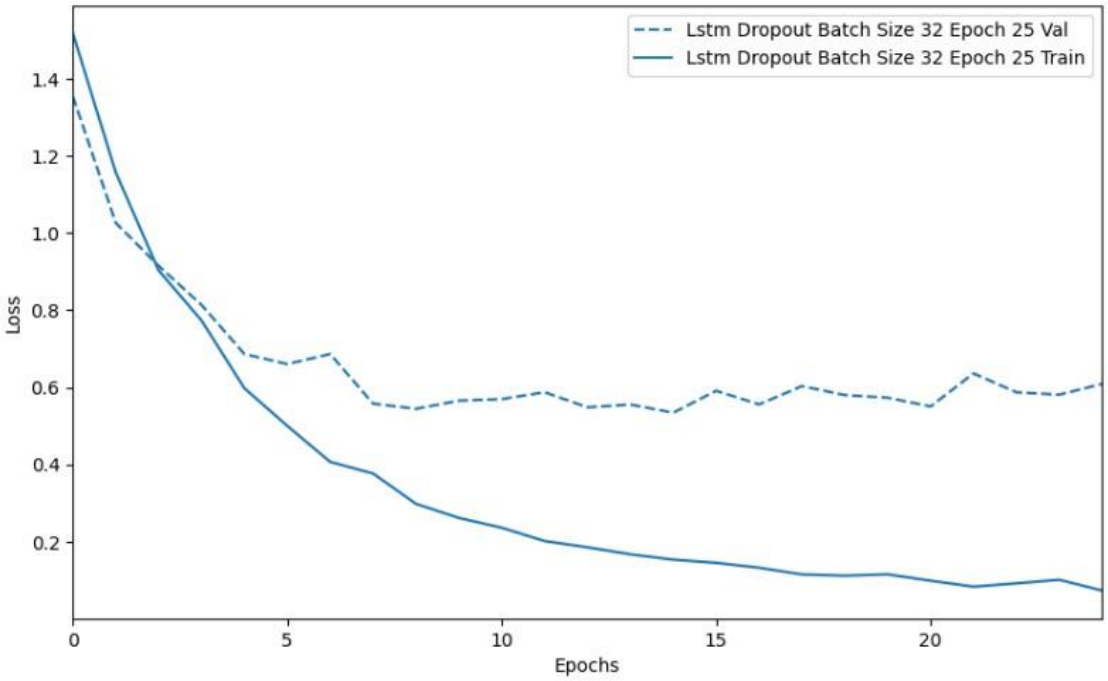
Gambar 4.19 menunjukkan performa *loss function* pada data *validation* yang terlihat menurun hingga *epoch* ke-1, namun pada *epoch* ke-2 performa *loss function* terus meningkat dan tidak stabil sehingga mengakibatkan performa pada model tersebut *overfitting*. Oleh karena itu, untuk mengurangi *overfitting* pada model maka diterapkan salah satu metode untuk mengurangi *overfitting* yaitu dropout. Pada Gambar 4.20, Gambar 4.21, Gambar 4.23, dan Gambar 4.24 menampilkan grafik *training* model algoritma LSTM dan BiLSTM menggunakan dropout dengan *batch size* 32.



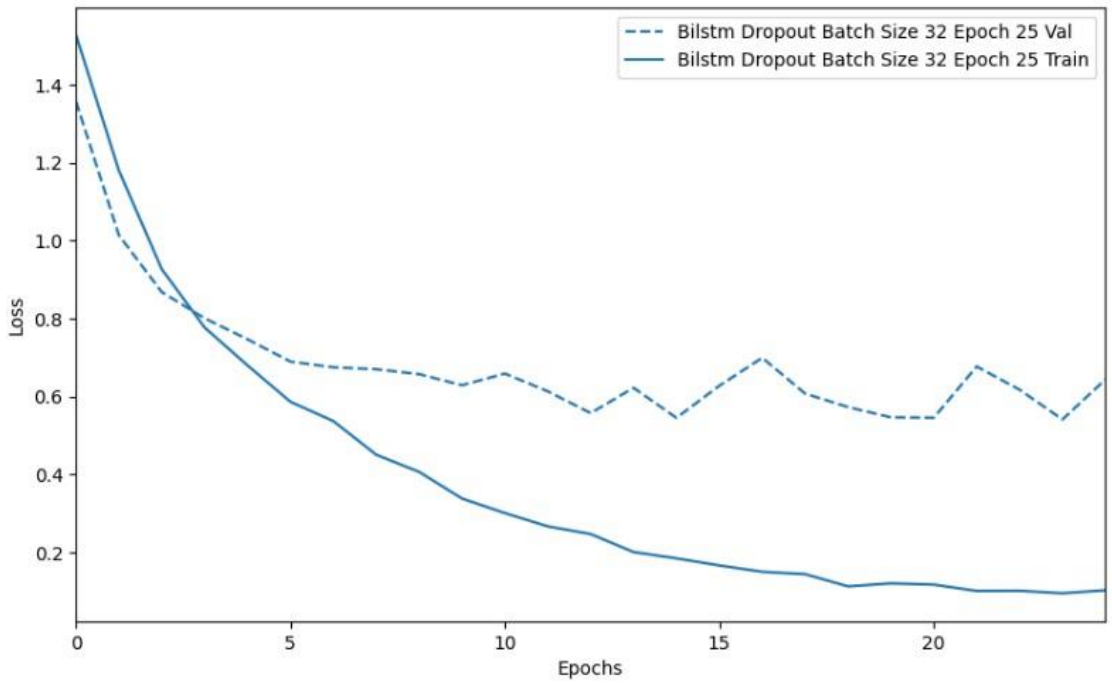
Gambar 4.20 Grafik *training* Model LSTM dengan dropout *batch size* 32 *epoch* 20



Gambar 4.21 Grafik *training* Model BiLSTM dengan dropout *batch size* 32 *epoch* 20



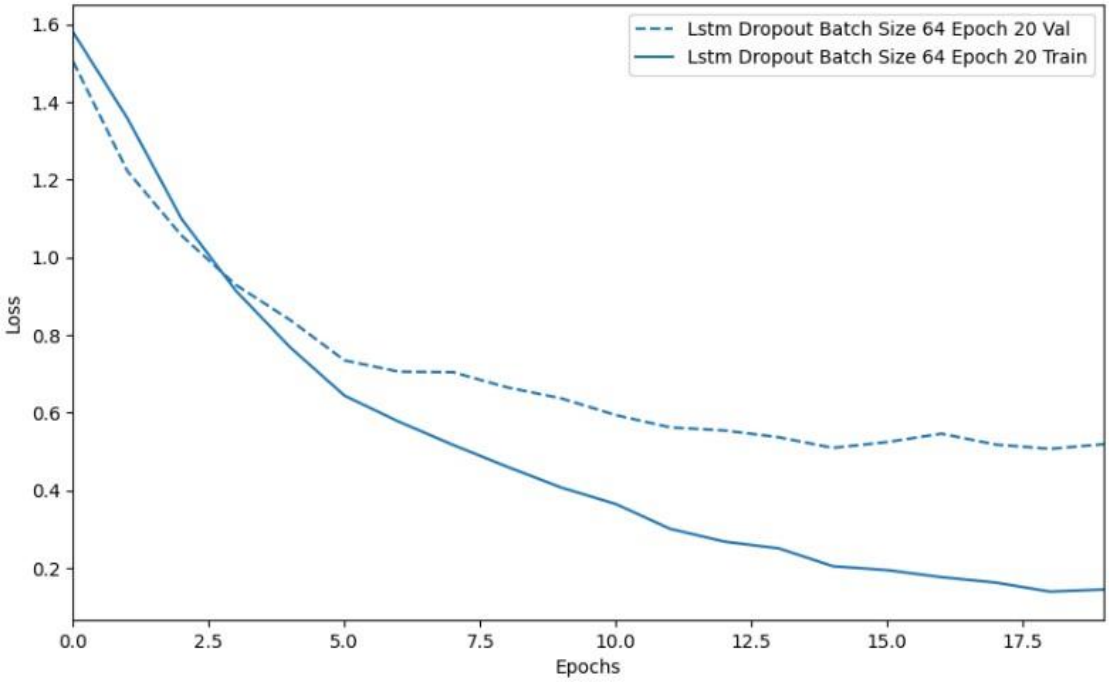
Gambar 4.22 Grafik *training* Model LSTM dengan dropout *batch size* 32 *epoch* 25



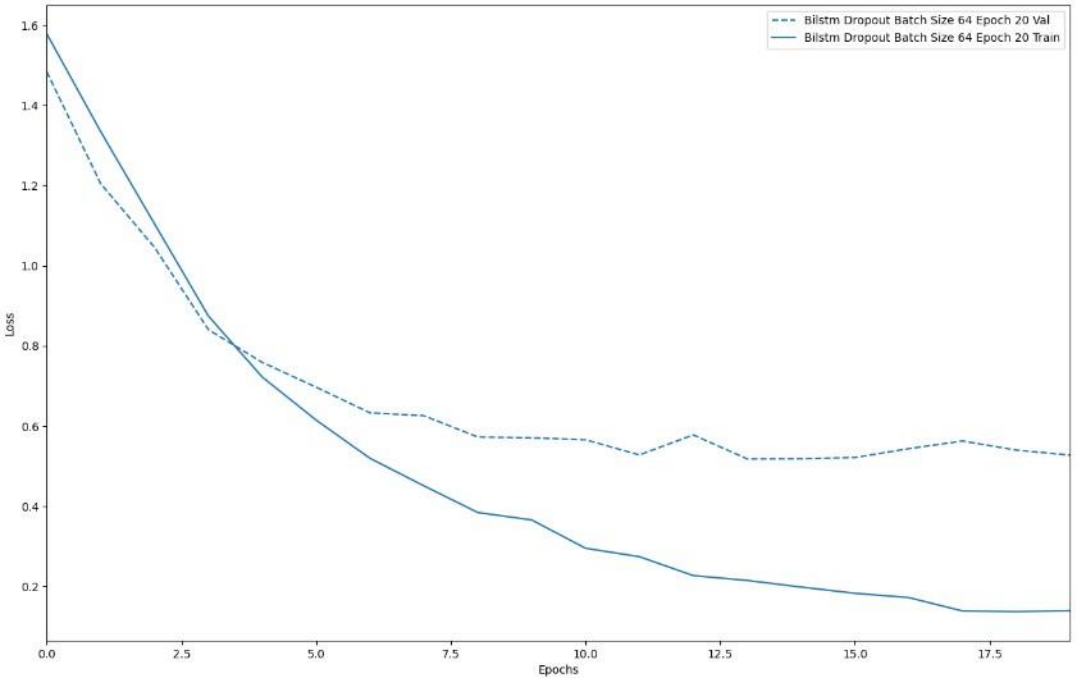
Gambar 4.23 Grafik *training* Model BiLSTM dengan dropout *batch size* 32 *epoch* 25

Gambar 4.20 menunjukkan performa *loss function* model LSTM menggunakan dropout dengan *batch size* 32 *epoch* 20 pada data *validation* yang cukup baik, namun dari *epoch* ke-13 performa *loss function* cenderung meningkat hingga *epoch* ke-20. Gambar 4.21 menunjukkan performa *loss function* model BiLSTM menggunakan dropout dengan *batch size* 32 *epoch* 20 pada data *validation* yang tidak stabil, ditandai dengan mengalami penurunan yang cukup baik hingga *epoch* ke-15, namun pada *epoch* ke-16 performa *loss function* mengalami sedikit peningkatan dan kembali mengalami sedikit penurunan hingga *epoch* ke-20. Selanjutnya, untuk melihat kelanjutan dari *training* model tersebut, maka *training* model tersebut dilanjutkan dengan *epoch* yang lebih tinggi.

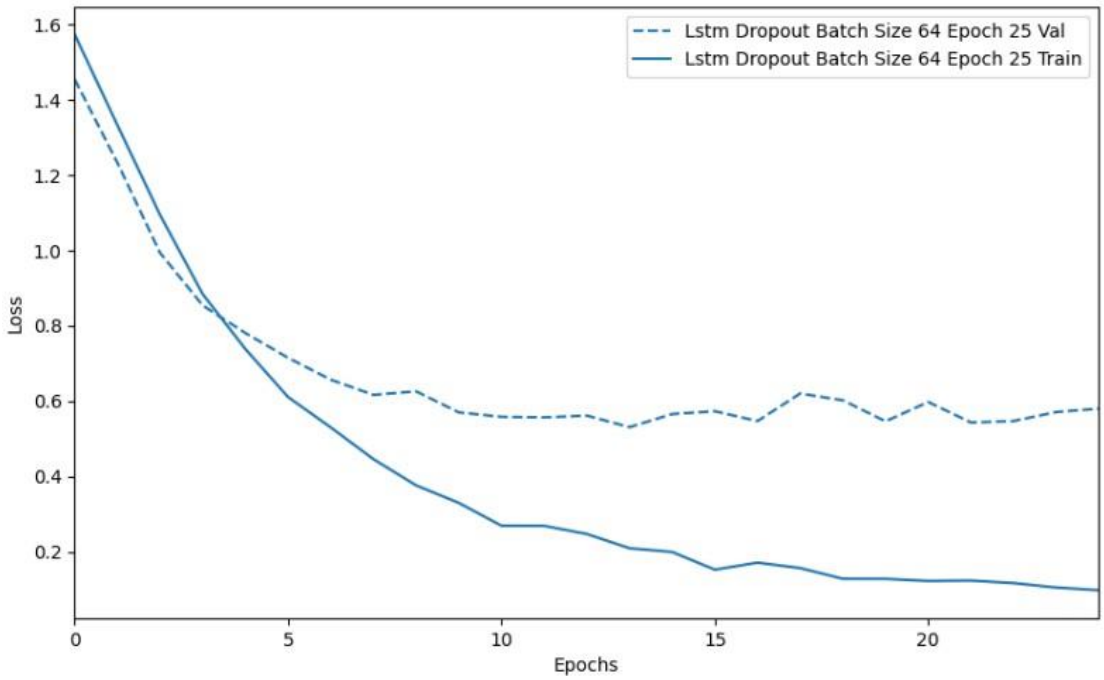
Pada Gambar 4.22 dan Gambar 4.23 menampilkan performa *loss function* model LSTM dan BiLSTM menggunakan dropout dengan *batch size* 32 *epoch* 25 pada data *validation* yang cukup baik namun tidak stabil yang ditandai dengan grafik *validation* yang naik turun. Oleh karena itu, dari keempat grafik tersebut menghasilkan model yang cukup baik namun cenderung tidak stabil sehingga untuk membuat model menjadi lebih stabil maka diperlukan *batch size* yang lebih tinggi. Gambar 4.24, Gambar 4.25, Gambar 4.26, dan Gambar 4.27 menampilkan grafik *Training* algoritma LSTM dan BiLSTM yang menggunakan dropout dengan *batch size* 64.



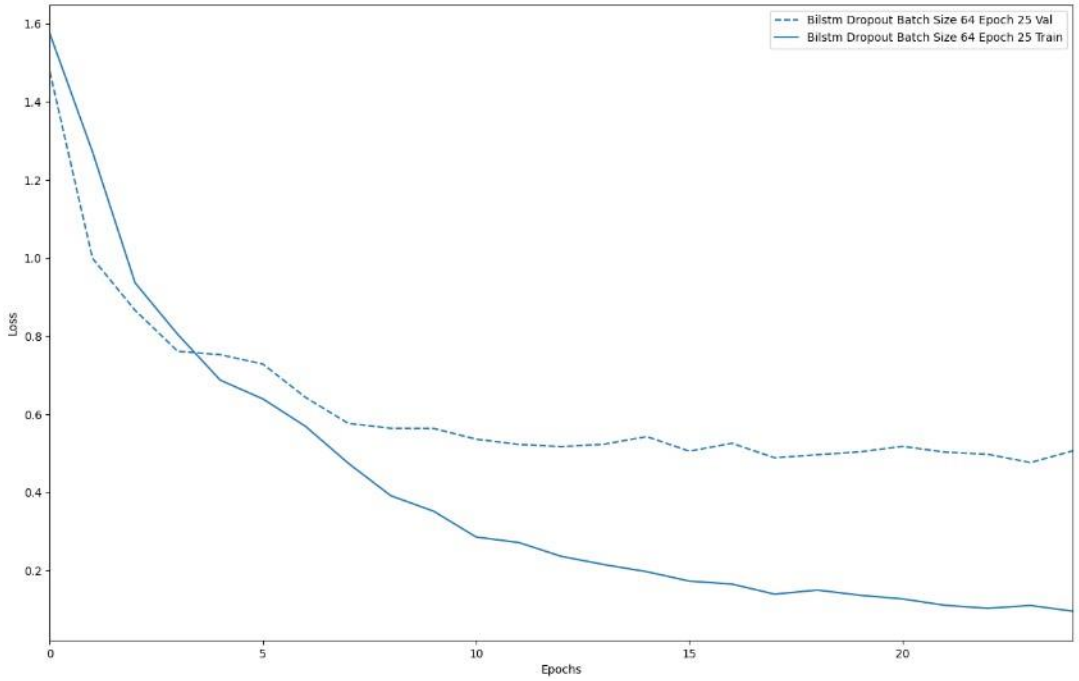
Gambar 4.24 Grafik *training* Model LSTM dengan dropout *batch size* 64 *epoch* 20



Gambar 4.25 Grafik *training* Model BiLSTM dengan dropout *batch size* 64 *epoch* 20



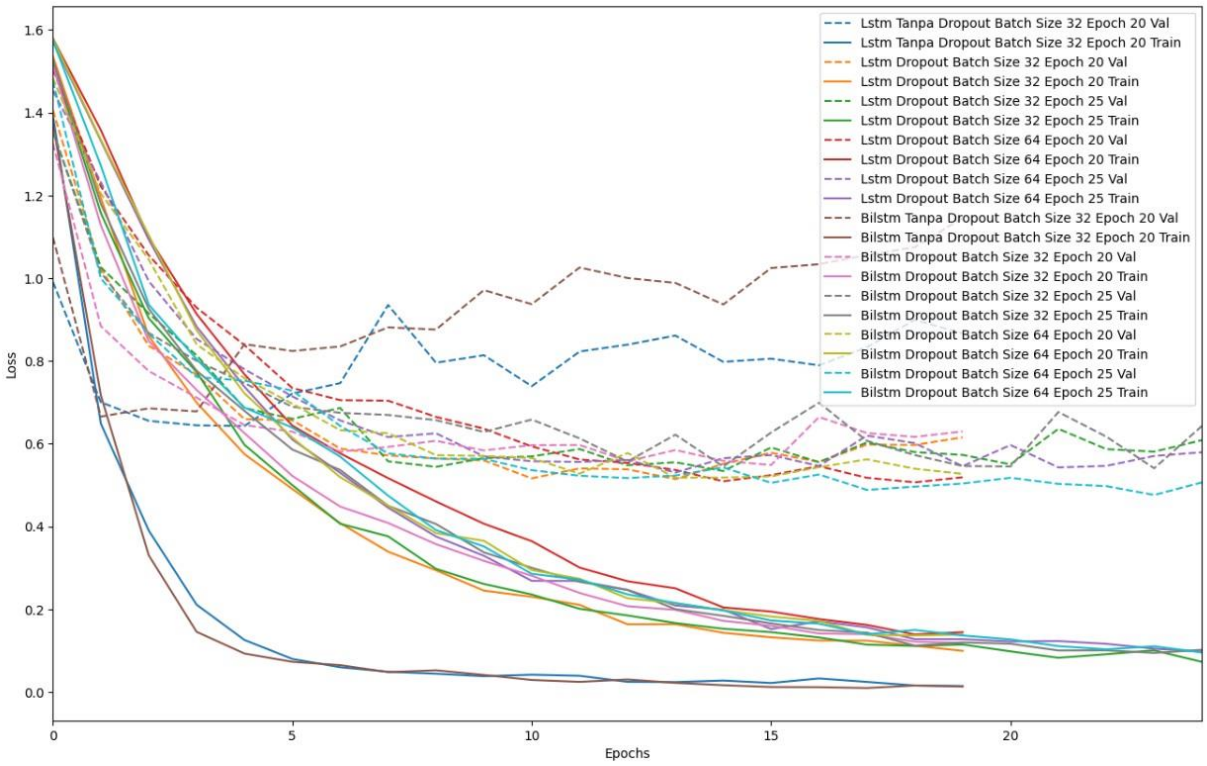
Gambar 4.26 Grafik *training* Model LSTM dengan dropout *batch size* 64 *epoch* 25



Gambar 4.27 Grafik *training* Model BiLSTM dengan dropout *batch size* 64 *epoch* 25

Pada Gambar 4.24 menunjukkan performa *loss function* model LSTM menggunakan dropout dengan *batch size* 64 *epoch* 20 pada data *validation* yang baik (*good fitting*), ditandai dengan penurunan yang signifikan hingga *epoch* ke-5 dan dilanjutkan dengan penurunan yang

teratur hingga *epoch* ke-20 dan terlihat model tersebut stabil. Gambar 4.25 menunjukkan performa *loss function* model BiLSTM menggunakan dropout dengan *batch size* 64 *epoch* 20 pada data *validation* yang baik (*good fitting*), ditandai dengan penurunan yang signifikan hingga *epoch* ke-3 dan dilanjutkan dengan penurunan yang teratur hingga *epoch* ke-20 dan terlihat grafik model tersebut stabil. Gambar 4.26 menunjukkan performa *loss function* model LSTM menggunakan dropout dengan *batch size* 64 *epoch* 25 pada data *validation* yang tidak terlalu baik, ditandai dengan adanya naik turun pada pertengahan proses *training*. Gambar 4.27 menunjukkan performa *loss function* model BiLSTM menggunakan dropout dengan *batch size* 64 *epoch* 25 pada data *validation* yang baik (*good fitting*), ditandai dengan penurunan yang signifikan hingga *epoch* ke-3 dan dilanjutkan dengan penurunan yang teratur hingga *epoch* ke-25 dan terlihat grafik model tersebut stabil. Dari pengujian yang telah dilakukan, performa *loss function* tiap model dibandingkan untuk melihat model yang memiliki performa terbaik dalam melakukan klasifikasi. Untuk memudahkan dalam proses membandingkan, penulis melakukan komparasi masing-masing model dalam sebuah grafik *loss function*. Gambar 4.28 menampilkan perbandingan grafik setiap model.



Gambar 4.28 Komparasi Grafik *Training* Tiap Model

Pada Gambar 4.28 dapat dilihat bahwa model yang menggunakan dropout memiliki performa yang lebih seimbang (*good fitting*) dibanding dengan model tanpa dropout yang memiliki performa *overfitting*. Hal tersebut akan berpengaruh terhadap kinerja model dalam melakukan klasifikasi yang mana jika model tersebut *overfitting* maka model akan berlebihan dalam mengenal pola pada teks atau kalimat sehingga model akan kesulitan dalam memprediksi. Selain itu, penggunaan *batch size* dan *epoch* juga berpengaruh terhadap performa model. Dengan *epoch* yang lebih tinggi maka dapat melihat kelanjutan dari *training* data sehingga apabila performa model sudah baik maka *training* akan dicukupkan pada *epoch* tersebut. Kemudian dengan *batch size* yang lebih tinggi performa model dapat menjadi lebih stabil. Untuk mempermudah melihat angka perbandingan *loss function* tiap model maka dibuat tabel perbandingan *loss function* yang direpresentasikan pada Tabel 4.2 dan Tabel 4.3.

Tabel 4.2 Perbandingan *loss function* LSTM

Activation	**Optimizer**	**LSTM Unit**	**Batch Size**	**Epoch**	**Dropout**	**Training Loss**	**Validation Loss**
Softmax	Adam	64, 64	32	20	-	0.0146	0.8668
Softmax	Adam	64, 64	32	20	0.9	0.0997	0.6154
Softmax	Adam	64, 64	32	25	0.9	0.0729	0.6088
Softmax	Adam	64, 64	64	20	0.9	0.1447	0.5189
Softmax	Adam	64, 64	64	25	0.9	0.0977	0.5794

Tabel 4.3 Perbandingan *loss function* BiLSTM

Activation	**Optimizer**	**LSTM Unit**	**Batch Size**	**Epoch**	**Dropout**	**Training Loss**	**Validation Loss**
Softmax	Adam	64, 64	32	20	-	0.0132	1.1461
Softmax	Adam	64, 64	32	20	0.9	0.1189	0.6298
Softmax	Adam	64, 64	32	25	0.9	0.1021	0.6432
Softmax	Adam	64, 64	64	20	0.9	0.1387	0.5272
Softmax	**Adam**	**64, 64**	**64**	**25**	**0.9**	**0.0957**	**0.5066**

Dari Tabel 4.2 dan Tabel 4.3 diatas model terbaik adalah model yang memiliki performa *loss function* antara data *training* dan data *validation* yang *good fitting* dan stabil berdasarkan grafik *training* serta memiliki angka *training loss* yang rendah dan selisih antara *training loss*

dan *validation loss* yang kecil. Berdasarkan kedua tabel perbandingan *loss function* diatas model terbaik adalah model BiLSTM dengan dropout yang di *training* dengan *epoch* 25 dan *batch size* 64. Setelah melakukan *training* data dari masing-masing model, maka didapat nilai performa *loss function* dari masing-masing model tersebut. Selanjutnya dari model yang telah di *training* tersebut dapat dihitung performa model dalam melakukan prediksi dengan menggunakan metrik evaluasi. Adapun untuk metrik evaluasi yang dihasilkan yaitu *accuracy*, *precision*, *recall*, dan *F1-Score*. Hasil perhitungan dari performa model tersebut akan dibandingkan untuk mengetahui model terbaik berdasarkan data latih. Tabel 4.4 dan Tabel 4.5 masing-masing menyajikan metrik evaluasi dari algoritma LSTM dan BiLSTM berdasarkan hasil *training* data latih.

Tabel 4.4 Metrik evaluasi LSTM berdasarkan hasil *training* data latih

LSTM Unit	Batch Size	Epoch	Dropout	Accuracy	Precision	Recall	F1-Score
64, 64	32	20	-	0.9982	0.9955	0.9955	0.9955
64, 64	32	20	0.9	0.9871	0.9698	0.9657	0.9677
64, 64	32	25	0.9	0.9905	0.9776	0.9750	0.9762
64, 64	64	20	0.9	0.9804	0.9557	0.9457	0.9506
64, 64	64	25	0.9	0.9866	0.9683	0.9648	0.9665

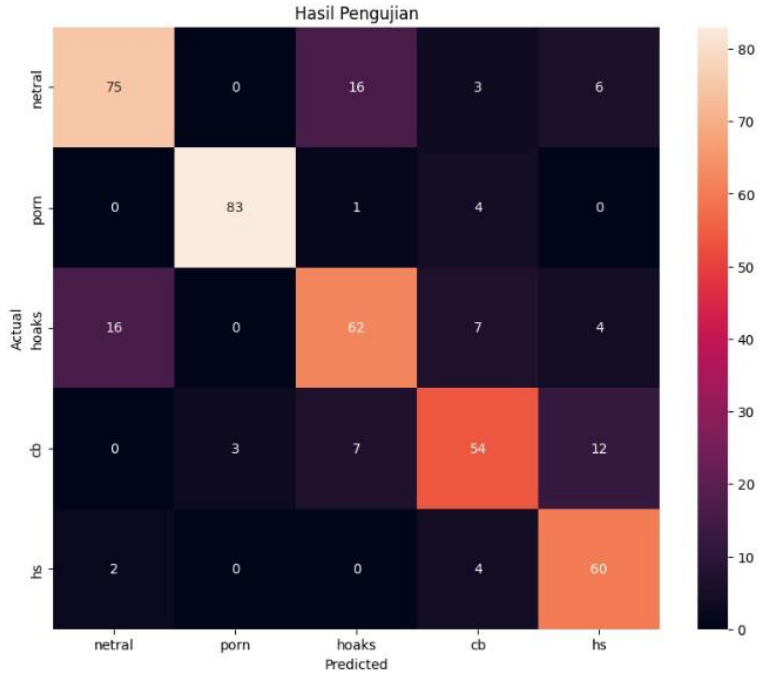
Tabel 4.5 Metrik evaluasi BiLSTM berdasarkan hasil *training* data latih

LSTM Unit	Batch Size	Epoch	Dropout	Accuracy	Precision	Recall	F1-Score
64, 64	32	20	-	0.9975	0.9940	0.9934	0.9936
64, 64	32	20	0.9	0.9837	0.9611	0.9571	0.9590
64, 64	32	25	0.9	0.9859	0.9670	0.9624	0.9646
64, 64	64	20	0.9	0.9807	0.9544	0.9487	0.9515
64, 64	**64**	**25**	**0.9**	**0.9875**	**0.9712**	**0.9660**	**0.9685**

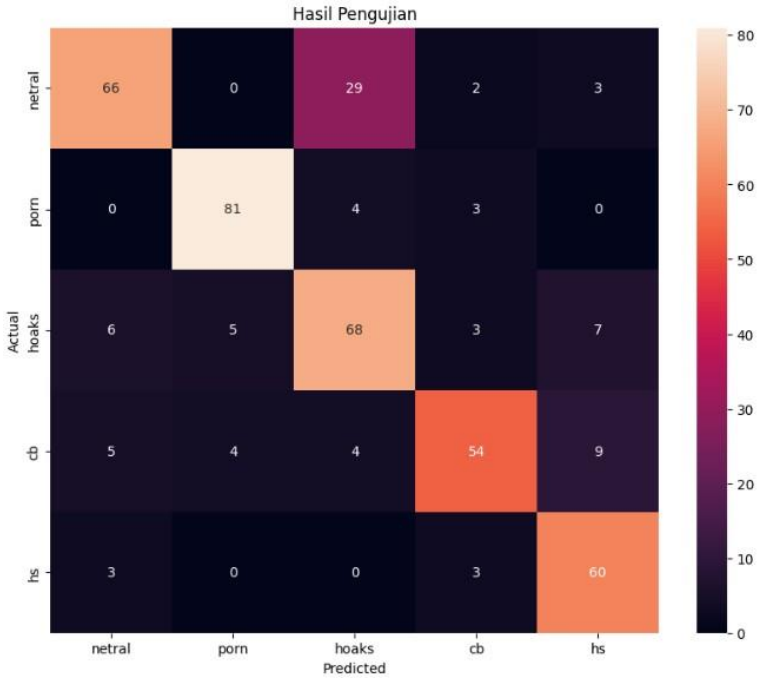
Berdasarkan Tabel 4.4 dan Tabel 4.5 metrik evaluasi berdasarkan hasil *training* dapat terlihat performa. Dari hasil perhitungan serta grafik *loss function* yang didapat, model terbaik adalah model yang memiliki nilai *F1-Score* yang tinggi dan juga memiliki performa *loss function* yang seimbang atau *good* fitting. Dari model yang dihasilkan terlihat bahwa model terbaik adalah model BiLSTM dengan dropout yang di *training* dengan *epoch* 25 dan *batch size* 64. Model tersebut memiliki performa *F1-Score* yang tinggi serta memiliki performa *loss*

function yang seimbang (*good fitting*). Hal ini sangat mempengaruhi performa model sehingga apabila model tersebut digunakan maka model tidak akan kesulitan untuk mengenali pola dari teks ataupun kalimat yang di input.

Setelah melakukan komparasi grafik *loss function* serta mendapatkan nilai metrik evaluasi dari data latih, selanjutnya nilai performa setiap model akan diukur menggunakan *confusion matrix* berdasarkan data uji. Dalam hal ini *confusion matrix* merangkum kinerja model dalam mengevaluasi efektivitas model untuk melakukan klasifikasi sehingga peneliti dapat mengetahui *error* pada operasi algoritma yang dijalankan. Pada *confusion matrix* ini terdapat lima kelas yang digunakan pada penelitian ini. Pada sumbu X merupakan kelas hasil prediksi model. Sedangkan sumbu Y merupakan kelas aktual dari data. Gambar 4.29 dan Gambar 4.30 menampilkan *confusion matrix* dari model algortima LSTM dan BiLSTM tanpa dropout.

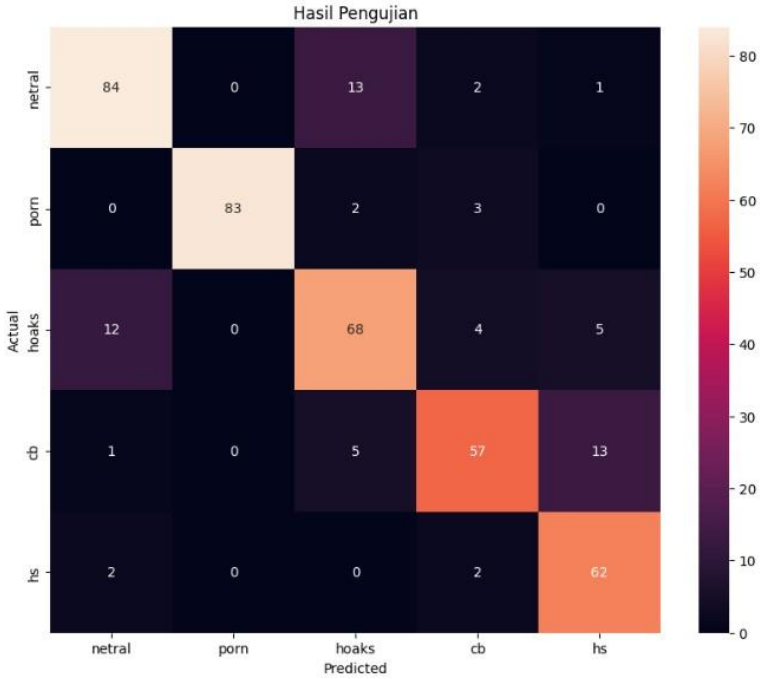


Gambar 4.29 *Confusion matrix* LSTM tanpa dropout

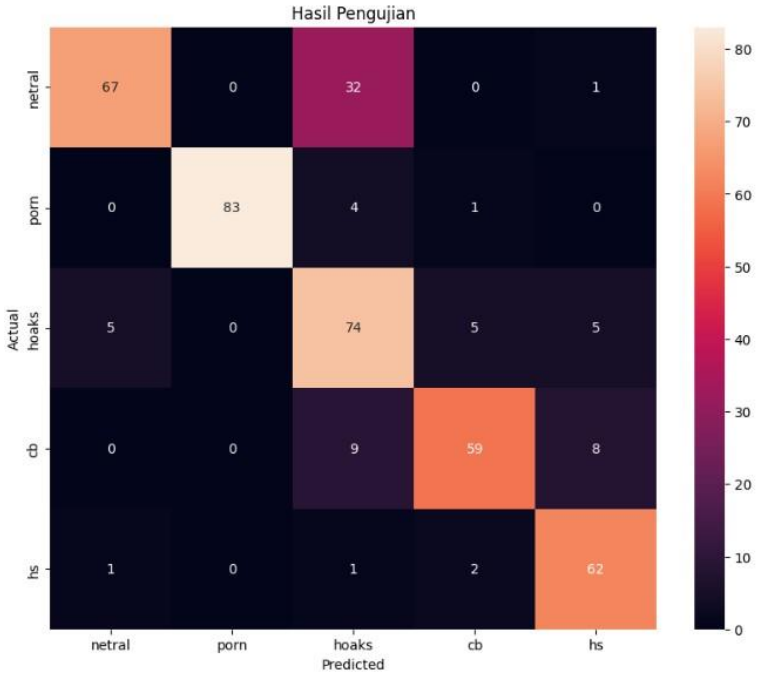


Gambar 4.30 *Confusion matrix* BiLSTM tanpa dropout

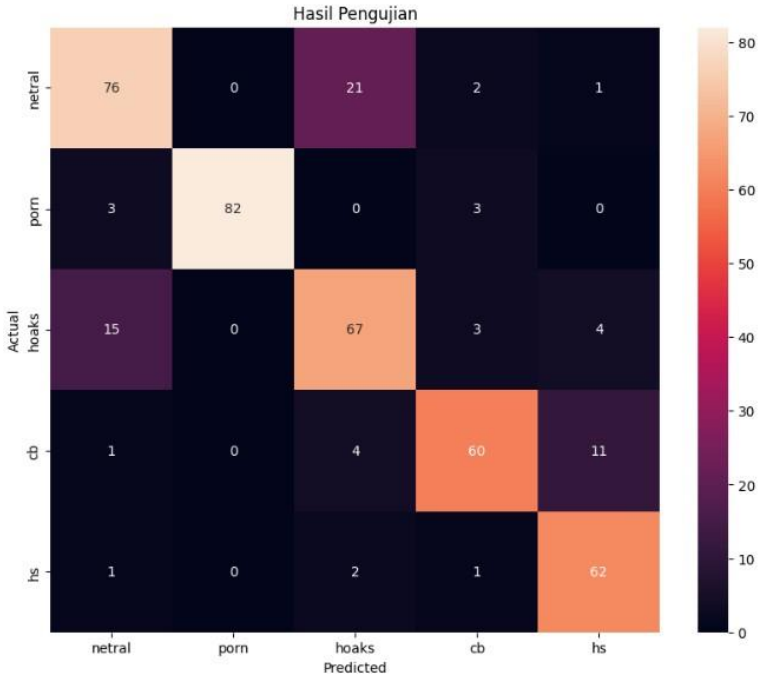
Pada Gambar 4.29 menunjukkan *confusion matrix* dari model LSTM tanpa dropout berdasarkan data uji. Berdasarkan hasil pengujian, terdapat beberapa kesalahan dalam prediksi pada label. Kesalahan prediksi model LSTM tanpa dropout tersebut terdapat pada label aktual netral yang diprediksi hoaks dan hoaks yang diprediksi netral serta label aktual *cyberbullying* yang diprediksi *hatespeech*. Gambar 4.30 menunjukkan *confusion matrix* dari model BiLSTM tanpa dropout berdasarkan data uji. Berdasarkan *confusion matrix* tersebut kesalahan prediksi model LSTM tanpa dropout tersebut meliputi kekurangan pada prediksi label aktual netral yang diprediksi hoaks. Gambar 4.31, Gambar 4.32, Gambar 4.33, dan Gambar 4.34 menampilkan algoritma LSTM dan BiLSTM menggunakan dropout dengan *batch size* 32.



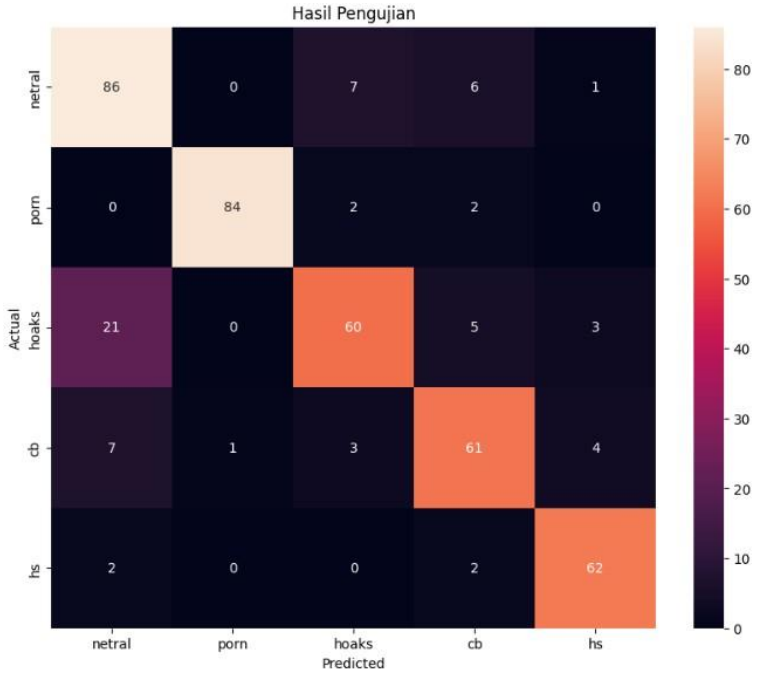
Gambar 4.31 *Confusion matrix* LSTM dengan dropout *batch size* 32 *epoch* 20



Gambar 4.32 *Confusion matrix* BiLSTM dengan dropout *batch size* 32 *epoch* 20



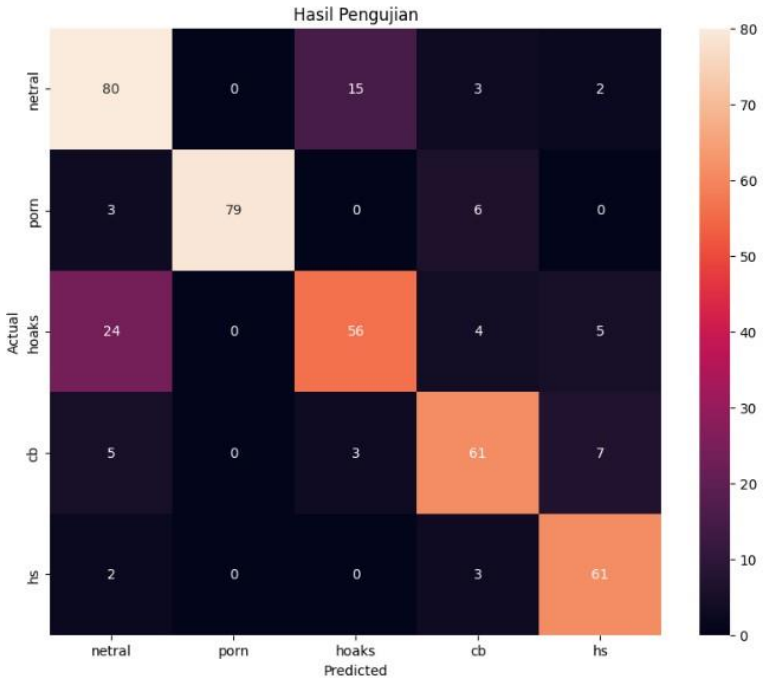
Gambar 4.33 *Confusion matrix* LSTM dengan dropout *batch size* 32 *epoch* 25



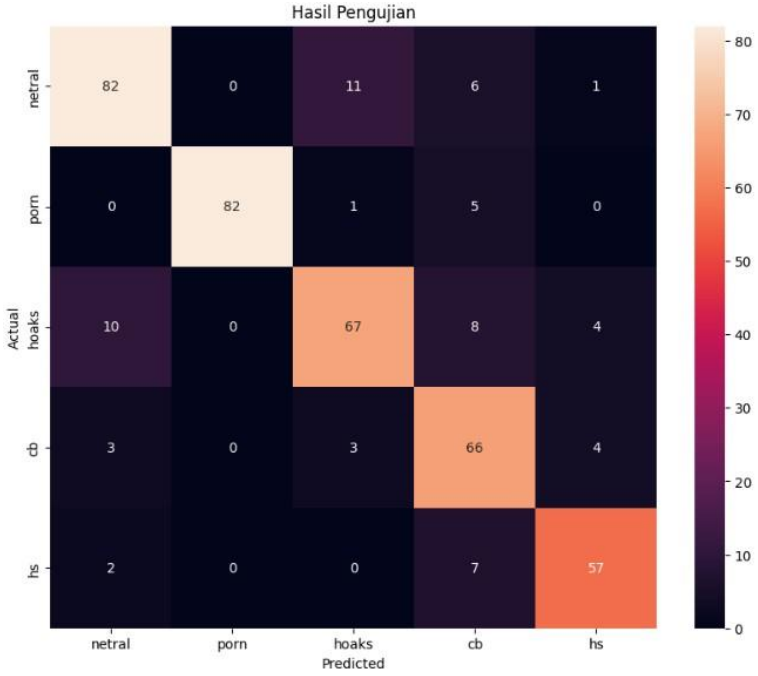
Gambar 4.34 *Confusion matrix* BiLSTM dengan dropout *batch size* 32 *epoch* 25

Gambar 4.31 dan Gambar 4.33 menunjukkan *confusion matrix* dari model LSTM menggunakan dropout dengan *batch size* 32 dan *epoch* 20 serta LSTM menggunakan dropout dengan *batch size* 32 dan *epoch* 25. Pada kedua model tersebut terdapat beberapa kelemahan dalam prediksi pada label yang meliputi kesalahan label aktual netral yang diprediksi hoaks

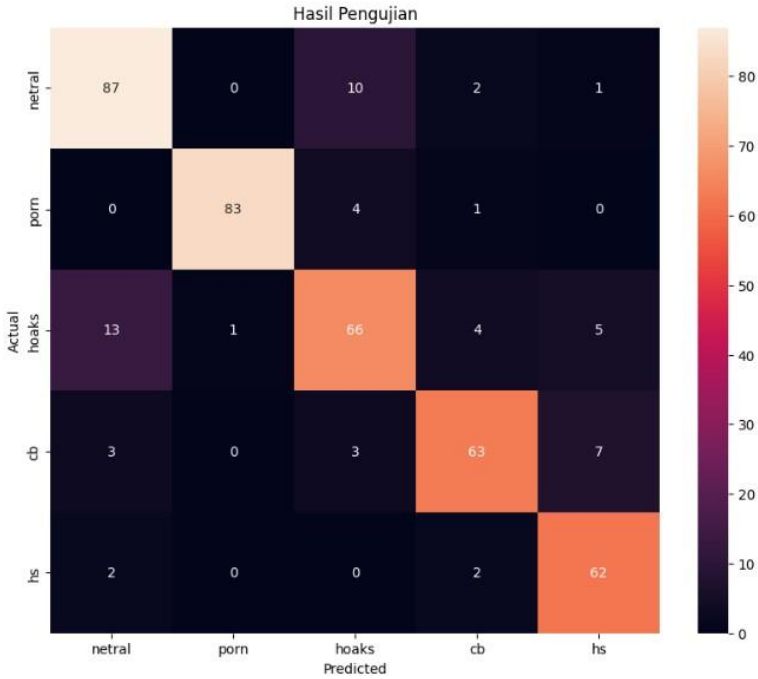
dan hoaks yang diprediksi netral serta label aktual *cyberbullying* yang diprediksi *hatespeech*. Gambar 4.32 menunjukkan *confusion matrix* dari model BiLSTM menggunakan dropout dengan *batch size* 32 dan *epoch* 20. Model tersebut memiliki kelemahan dalam prediksi pada label aktual netral yang ditandai dengan cukup banyak kesalahan prediksi label aktual netral yang diprediksi hoaks. Kemudian Gambar 4.34 menampilkan *confusion matrix* dari model BiLSTM menggunakan dropout dengan *batch size* 32 dan *epoch* 25. Model tersebut memiliki kelemahan dalam prediksi pada label aktual hoaks yang ditandai dengan cukup banyak kesalahan prediksi label aktual hoaks yang diprediksi netral. Gambar 4.35, Gambar 4.36, Gambar 4.37, dan Gambar 4.38 menampilkan algoritma LSTM dan BiLSTM menggunakan dropout dengan *batch size* 64.



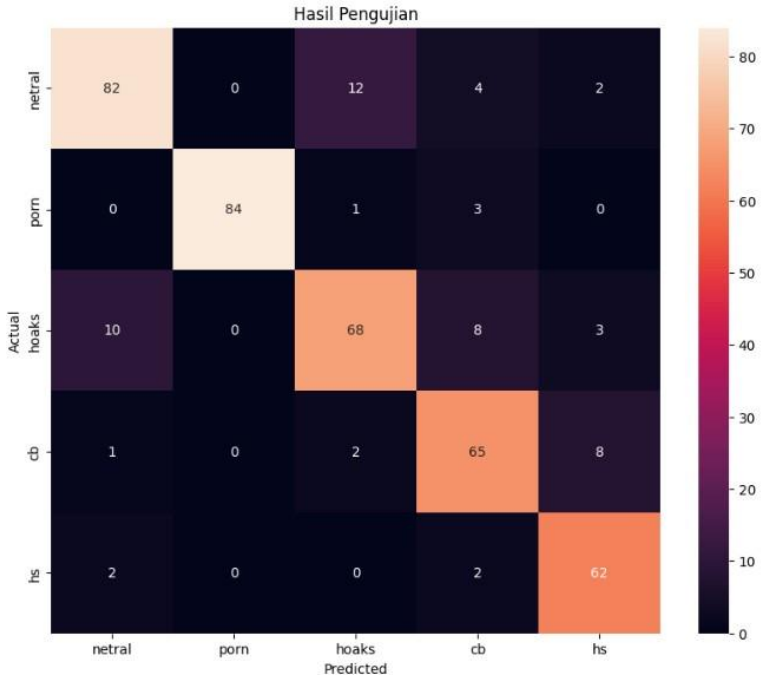
Gambar 4.35 *Confusion matrix* LSTM dengan dropout *batch size* 64 *epoch* 20



Gambar 4.36 *Confusion matrix* BiLSTM dengan dropout *batch size* 64 *epoch* 20



Gambar 4.37 *Confusion matrix* LSTM dengan dropout *batch size* 64 *epoch* 25



Gambar 4.38 *Confusion matrix* BiLSTM dengan dropout *batch size* 64 *epoch* 25

Gambar 4.35 menunjukkan *confusion matrix* dari model LSTM menggunakan dropout dengan *batch size* 64 dan *epoch* 20. Pada model tersebut terdapat kelemahan dalam prediksi pada label yang meliputi kesalahan label aktual netral yang diprediksi hoaks dan label hoaks yang diprediksi netral. Gambar 3.36 menunjukkan *confusion matrix* dari model BiLSTM menggunakan dropout dengan *batch size* 64 dan *epoch* 20. Model tersebut memiliki performa yang baik dalam prediksi ditandai dengan minimnya kesalahan dalam prediksi label. Gambar 4.37 dan Gambar 3.38 menunjukkan *confusion matrix* dari model LSTM menggunakan dropout dengan *batch size* 64 dan *epoch* 25 dan BiLSTM menggunakan dropout dengan *batch size* 64 dan *epoch* 25. Kedua model tersebut memiliki performa yang baik dalam prediksi ditandai dengan minimnya kesalahan dalam prediksi label.

Setelah mendapatkan *confusion matrix* dari masing-masing model, selanjutnya akan dihitung nilai performa dari masing-masing model tersebut. Adapun untuk metrik evaluasi yang dihasilkan dari *confusion matrix* yaitu *accuracy*, *precision*,*recall*, dan *F1-Score*. Hasil perhitungan dari performa model tersebut akan dibandingkan untuk mengetahui model terbaik. Tabel 4.6 dan Tabel 4.7 masing-masing menyajikan metrik evaluasi dari algoritma LSTM dan BiLSTM berdasarkan hasil *training* data latih.

Tabel 4.6 Metrik evaluasi LSTM berdasarkan data uji

LSTM Unit	Batch Size	Epoch	Dropout	Accuracy	Precision	Recall	F1-Score
64, 64	32	20	-	0.970	0.805	0.835	0.820
64, 64	32	20	0.9	0.968	0.847	0.881	0.864
64, 64	64	25	0.9	0.956	0.832	0.885	0.858
64, 64	64	20	0.9	0.957	0.782	0.844	0.812
64, 64	32	25	0.9	0.973	0.876	0.874	0.875

Tabel 4.7 Metrik evaluasi BiLSTM berdasarkan data uji

LSTM Unit	Batch Size	Epoch	Dropout	Accuracy	Precision	Recall	F1-Score
64, 64	32	20	-	0.959	0.848	0.831	0.839
64, 64	32	20	0.9	0.962	0.917	0.858	0.886
64, 64	32	25	0.9	0.964	0.842	0.885	0.863
64, 64	64	20	0.9	0.964	0.846	0.881	0.863
64, 64	**64**	**25**	**0.9**	**0.972**	**0.872**	**0.930**	**0.900**

Dari hasil perhitungan metrik evaluasi berdasarkan data uji pada Tabel 4.6 dan Tabel 4.7 yang didapat, model terbaik adalah model yang memiliki nilai *F1-Score* yang paling tinggi. Dari model yang dihasilkan terlihat bahwa model terbaik adalah model BiLSTM menggunakan dropout dengan *batch size* 64 dan *epoch* 25 yang memiliki performa *F1-Score* yang paling tinggi dibanding model lainnya. Dari perhitungan tersebut menunjukkan bahwa hasil tersebut sejalan dengan hasil yang didapat dari proses *training* data latih yang menunjukkan bahwa model BiLSTM menggunakan dropout dengan *batch size* 64 dan *epoch* 25 sebagai model terbaik. Setelah melihat metrik evaluasi berdasarkan model, selanjutnya akan disajikan contoh data prediksi yang salah dari model terbaik. Tabel 4.8 menampilkan contoh hasil prediksi salah dari model BiLSTM menggunakan dropout dengan *batch size* 64 dan *epoch* 25.

Tabel 4.8 Hasil prediks salah model BiLSTM dengan dropout *batch size* 64 *epoch* 25

No.	Teks	Label Aktual	Label Prediksi
1.	ini masalah yang sangat serius dan perlu penanganan segera oleh karena itu saya sangat mendukung kebijakan gubernur dki anies baswedan untuk penggunaan bus bertenaga listrik di dki jakarta untuk mengurangi polusi udara dan kebisingan	netral	hoaks
2.	komisioner komnas ham choirul anam mengatakan menkopolhukam berusaha menghalang halangi penyelesaian pelanggaran ham berat yang terjadi di papua	netral	hoaks
3.	pemain sepak bola asal ingris david beckham ternyata berkunjung ke pedalaman wamena papua secara diam diam pemain ternama dunia ini ternyata memberikan bantuan kemanusian setelah mendegar tragedi kematian anak di duga pada tahun silam lalu pada kedatangan sang bintang sepak bola itu akhirnya dikagetkan dengan kaos baju yang dikenakan david beckham yang bertulisan free west papua	hoaks	netral
4.	barack obama mewakili negara amerika serikat telah menyatakan kemenangan prabowo sandi	hoaks	netral
5.	memang situ mengerti uu kriminal anak sebelumnya kan ada anak yang juga menghina presiden dibawah umur kok di penjara tahun sedangkan anak cina yang ngancam mau bunuh presiden tidak di apain lucu kan anda yang super tolol waras tidak otak kamu	*cyberbullying*	hoaks
6.	tolol itu banyak yang bela penista agama walau sudah tersangka	*cyberbullying*	*hatespeech*

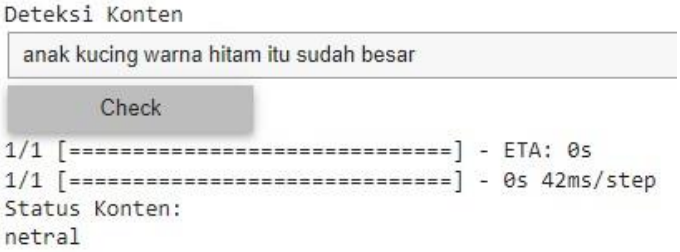
7.	jokowi komunis	*hatespeech*	*cyberbullying*

Berdasarkan Tabel 4.8 yang menampilkan beberapa contoh dari prediksi salah model BiLSTM dengan dropout terdapat beberapa kelemahan dari model tersebut. Pada beberapa kalimat yang memiliki label netral, model tersebut memprediksi bahwa kalimat memiliki label hoaks begitu juga sebaliknya. Hal tersebut disebabkan karena adanya kemiripan kata pada kalimat yang membuat model sulit untuk mendeteksi label dari kalimat tersebut. Selain itu, pada kalimat yang berlabel *cyberbullying* terdapat beberapa kalimat yang diprediksi salah oleh model. Hal tersebut dikarenakan pada kalimat meski terdapat kata yang menjurus kepada *cyberbullying*, namun karena pada dataset kalimat yang berlabel *cyberbullying* memiliki kalimat yang tidak terlalu panjang, sehingga model memprediksi bahwa kalimat tersebut bukan termasuk *cyberbullying*.

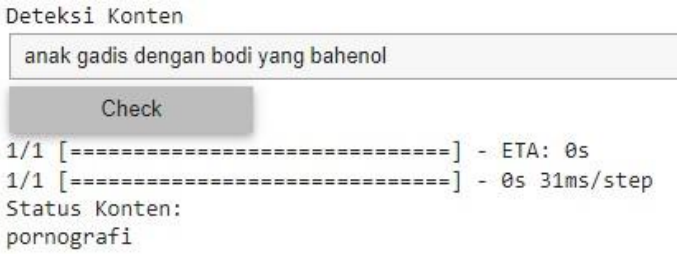
3.6 Pengujian *Interface*

Setelah melakukan pengujian serta melakukan evaluasi terhadap masing-masing model, maka diperoleh model BiLSTM dengan dropout yang merupakan model dengan performa terbaik dibandingkan model yang lainnya. Oleh karena itu, pengujian menggunakan *interface* akan menggunakan model BiLSTM dengan dropout tersebut. Pada *interface* tersebut terdapat kolom input yang berfungsi sebagai tempat untuk memasukkan input berupa teks atau kalimat. Kemudian terdapat button ‘check’ yang berfungsi sebagai tombol perintah untuk mengecek input tersebut.

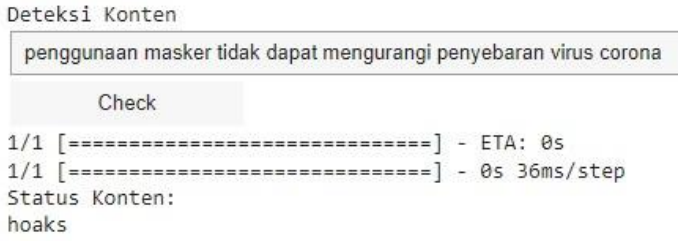
Gambar 4.39 menampilkan prediksi label netral yang merupakan kalimat yang tidak mengandung pelanggaran terhadap Undan-Undang ITE. Gambar 4.40 menampilkan prediksi label pornografi yang merupakan kalimat yang mengandung pelanggaran terhadap pasal 27 ayat 1 tentang pornografi Undang-Undang ITE. Gambar 4.41 menampilkan prediksi label hoaks yang merupakan kalimat yang mengandung pelanggaran terhadap pasal 28 ayat 1 tentang berita bohong dan menyesatkan Undang-Undang ITE. Sedangkan Gambar 4.42 menampilkan prediksi label *cyberbullying* yang merupakan kalimat yang mengandung pelanggaran terhadap pasal 45B tentang *cyberbullying* Undang-Undang ITE. Gambar 4.43 menampilkan prediksi label *hatespeech* yang merupakan kalimat yang mengandung pelanggaran terhadap pasal 28 ayat 2 tentang ujaran kebencian SARA Undang-Undang ITE.



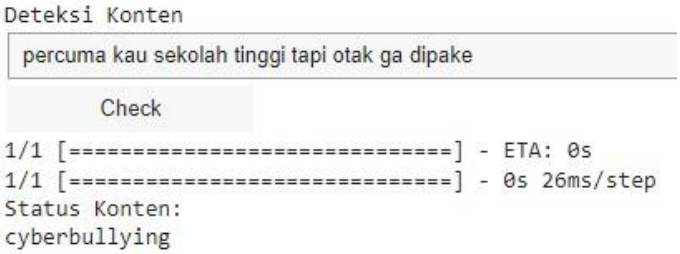
Gambar 4.39 Prediksi Label Netral



Gambar 4.40 Prediksi Label Pornografi



Gambar 4.41 Prediksi Label Hoaks



Gambar 4.42 Prediksi Label *Cyberbullying*

Gambar 4.43 Prediksi Label *Hatespeech*

BAB V
PENUTUP

4.1 Kesimpulan

Dalam penelitian ini, kami mempresentasikan Klasifikasi Pelanggaran Undang-Undang ITE pada Twitter Menggunakan LSTM dan BiLSTM. Dalam percobaan yang dilakukan, kami membandingkan dua algoritma yaitu LSTM dan BiLSTM yang mana setiap algoritma memiliki dua skenario yaitu tanpa dropout dan dengan dropout. Berdasarkan hasil pengujian yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut:

a. Algoritma LSTM dan BiLSTM terbukti dapat melakukan klasifikasi pelanggaran Undang-Undang ITE berdasarkan 5 kelas (netral, pornografi, hoaks, *cyberbullying*, dan *hatespeech*)
b. Model dengan dropout memiliki performa yang lebih baik (*good fitting*) dalam *loss function* dibandingkan dengan model tanpa dropout yang memiliki performa (*overfitting*).
c. Berdasarkan pengujian setiap algoritma LSTM dan BiLSTM yang menggunakan dropout, algoritma BiLSTM memiliki performa yang lebih unggul dibanding LSTM.
d. Model BiLSTM dengan dropout *batch size* 64 *epoch* 25 adalah model terbaik dalam penelitian ini yang memiliki performa *accuracy* sebesar 0.9875 dan *F1-Score* sebesar 0.9685 berdasarkan data latih serta *accuracy* sebesar 0.972 dan *F1-Score* sebesar 0.900 berdasarkan data uji.
e. Berdasarkan hasil evaluasi dari data uji diperoleh bahwa hasil tersebut sejalan dengan hasil yang didapat dari proses *training* data latih yang menunjukkan bahwa model BiLSTM menggunakan dropout dengan *batch size* 64 dan *epoch* 25 sebagai model terbaik.
f. Berdasarkan hasil pengujian, penelitian ini menampilkan hasil yang cukup baik dengan ketepatan memprediksi kalimat yang dibuktikan dengan percobaan menggunakan *interface* sederhana.

4.2 Saran

Berdasarkan penelitian yang telah dilakukan, diharapkan dengan adanya penelitian ini dapat mengembangkan berbagai penelitian lainnya oleh peneliti-peneliti yang akan datang dengan beberapa saran dari penulis seperti:

a. Melabeli kelas pada dataset oleh praktisi hukum.
b. Membuat daftar *stopword* dan slang word yang lebih kompleks.
c. Menguji dengan metode ataupun algoritma lain.
d. Hasil dari klasifikasi kelas dapat diekstrak menjadi kelas yang lebih spesifik lagi seperti *hatespeech* berupa individu, komunitas, agama, suku, adat dan sebagainya.
e. Menambah algoritma terbaru serta dapat membandingkannya agar memperoleh algoritma yang lebih baik.

DAFTAR PUSTAKA

Alita, D., Fernando, Y., & Sulistiani, H. (2020). Implementasi Algoritma Multiclass SVM pada Opini Publik Berbahasa Indonesia di Twitter. *Jurnal Tekno Kompak*, 14.2: 86-91.

Amigo, J. M. (2021). Data mining, machine learning, deep learning, chemometrics: Definitions, common points and trends (Spoiler Alert: VALIDATE your models!). *Brazilian Journal of Analytical Chemistry*, 8.32: 45-61.

Atmajaya, P. A., Amorokhman, F. I., Prasetya, M. D., Ihsan, A. F., & Junaedi, D. (2022). ITE Law Enforcement Support through Detection Tools of Fake News, Hate Speech, and Insults in Digital Media. *2022 5th International Conference on Information and Communications Technology (ICOIACT)*, pp. 452-456.

Chatterjee, A., Narahari, K. N., Joshi, M., & Agrawal, P. (2019). SemEval-2019 Task 3: EmoContext Contextual Emotion Detection in Text. *Proceedings of the 13th international workshop on semantic evaluation*, p. 39-48.

Fadli, H. F., & Hidayatullah, A. F. (2021). Identifikasi cyberbullying pada media sosial twitter menggunakan metode lstm dan bilstm. *Automata 2.1*.

Fudholi, D. H. (2022). Klasifikasi Emosi Pada Teks Menggunakan Metode Deep Learning.

Fudholi, D. H., Nayoan, R. A., Hidayatullah, A. F., & Arianto, D. B. (2022). A Hybrid CNN-BiLSTM Model For Drug Named Entity Recognition. *Journal of Engineering Science and Technology*, 17(1), 0730-0744.

Hesaputra, A. P., Saputra, R. D., & Wibowo, Y. H. (2022). Identifikasi Konten Dewasa pada Cuitan Twitter Menggunakan Metode BiLSTM Sebagai Upaya Mengatasi Penyebaran Pornografi Untuk Indonesia Maju. *Khazanah: Jurnal Mahasiswa*, 14.02.

Hidayatullah, A. F., Cahyaningtyas, S., & Hakim, A. M. (2019). Sentiment Analysis on Twitter using Neural Network: Indonesian Presidential Election 2019 Dataset. *IOP conference series: materials science and engineering*, Vol. 1077, No. 1, p. 012001.

Hidayatullah, A. F., Hakim, A. M., & Sembada, A. A. (2019). Adult Content Classification on Indonesian. *2019 International Conference on Advanced Computer Science and information Systems (ICACSIS)*, (pp. 235-240) IEEE.

Ibrohim, M. O., & Budi, I. (2019). Multi-label hate speech and abusive language detection in Indonesian Twitter. *Proceedings of the third workshop on abusive language online*, p. 46-57.

Ilham, F., & Maharani, W. (2022). Analyze Detection Depression In Social Media Twitter Using Bidirectional Encoder Representations from Transformers. *Journal of Information System Research (JOSH)*, 3(4), 476-482.

Jurafsky, D., & Martin, J. H. (2019, August 29). *Speech and Language Processing.* Retrieved from web.stanford.edu: https://web.stanford.edu/~jurafsky/slp3/old_oct19/17.pdf

Kadhim, A. I. (2018). An evaluation of preprocessing techniques for text classification. *International Journal of Computer Science and Information Security (IJCSIS)*, 16(6), 22-32.

Kemp, S. (2022, Februari 15). *Digital 2022: Indonesia*. Retrieved from Datareportal.com: https://datareportal.com/reports/digital-2022-indonesia

Kiswondari. (2023, Februari 4). *Juta Konten Negatif di Medsos Ditemukan: Pornografi, Hoaks, dan Ujaran Kebencian*. Retrieved from sindonews.com3: https://nasional.sindonews.com/read/1014341/15/3-juta-konten-negatif-di-medsos-ditemukan-pornografi-hoaks-dan-ujaran-kebencian-1675512131

Kusnadi, R., Yusuf, Y., Andriantony, A., Yaputra, R. A., & Caintan, M. (2021). ANALISIS SENTIMEN TERHADAP GAME GENSHIN IMPACT MENGGUNAKAN BERT. *Rabit: Jurnal Teknologi dan Sistem Informasi Univrab*, 6(2), 122-129.

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*.

Manaswi, N. K. (2018). *RNN and LSTM.* Apress, Berkeley, CA.

Manoppo, T. N., & Fudholi, D. H. (2021). Deteksi Cyberbullying Berdasarkan Unsur Perbuatan Pidana yang Dilanggar dengan Naïve Bayes dan Support Vector Machine. *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, 5.1: 10-19.

Panjaitan, A. T. (2021). Deteksi Perundungan Siber Pada Tweet Berbahasa Indonesia Menggunakan Support Vector Machine Dengan Lexicon Based Features. *Doctoral dissertation, Universitas Brawijaya*.

Pipin, S. J., & Kurniawan, H. (2022). Analisis Sentimen Kebijakan MBKM Berdasarkan Opini Masyarakat di Twitter Menggunakan LSTM. *Jurnal SIFO Mikroskil*, vol, 23, 197-208.

Saksesi, A. S., Nasrun, M., & Setianingsih, C. (2018). Analysis Text of Hate Speech Detection Using Recurrent Neural Network. *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, (pp. 242-248).

Sudoyo, W. (2023, Januari 12). *Selama 2022, Kominfo Blokir 238.226 Konten Negatif*. Retrieved from Infopublik.id: https://www.infopublik.id/kategori/nasional-sosial-budaya/701902/selama-2022-kominfo-blokir-238-226-konten-negatif#

Tandijaya, J. H., Liliana, L., & Sugiarto, I. (2021). Klasifikasi dalam Pembuatan Portal Berita Online dengan Menggunakan Metode BERT. *Jurnal Infra*, 9(2), 320-325.

Torregrosa, J., Bello-Orgaz, G., Martinez-Camara, E., Ser, J. D., & Camacho, D. (2022). A survey on extremism analysis using natural language processing: definitions, literature review, trends and challenges. *Journal of Ambient Intelligence and Humanized Computing*, 1-37.

Vig, J., & Belinkov, Y. (2019). Analyzing the Structure of Attention in a Transformer Language Model. *arXiv preprint arXiv:1906.04284*.

Widianto, A. B., & Fudholi, D. H. (2021). Klasifikasi Emosi pada Teks dengan Menggunakan Metode Deep Learning. *Syntax Literate; Jurnal Ilmiah Indonesia*, 6(1), 546-553.

LAMPIRAN