

**ANALISIS SENTIMEN KOMENTAR CYBERBULLYING PADA
SOSIAL MEDIA TIKTOK MENGGUNAKAN ARSITEKTUR
TEXTCNN**

TUGAS AKHIR

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1) di
Program Studi Teknik Informatika, Fakultas Teknologi Industri,
Institut Teknologi Sumatera

Oleh:

Nikola Arinanda

121140202



ITERA

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2025**

MOTTO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

DAFTAR ISI

Motto	ii
DAFTAR ISI	iii
DAFTAR TABEL.....	vi
DAFTAR GAMBAR.....	vii
DAFTAR RUMUS	viii
DAFTAR KODE.....	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan.....	5
1.6.1 Bab I Pendahuluan	5
1.6.2 Bab II Tinjauan Pustaka.....	5
1.6.3 Bab III.....	5
1.6.4 Bab IV.....	5
1.6.5 Bab V.....	6
BAB II TINJAUAN PUSTAKA	7
2.1 Tinjauan Pustaka.....	7
2.2 Dasar Teori.....	12
2.2.1 <i>Cyberbullying</i>	12
2.2.2 Media Sosial	12

2.2.2.1	TikTok	12
2.2.3	<i>Natural Language Processing (NLP)</i>	13
2.2.4	Analisis Sentimen	13
2.2.5	<i>Text Pre-processing</i>	14
2.2.5.1	<i>Normalization</i>	14
2.2.5.2	IndoBERT Tokenizer	14
2.2.5.3	<i>Augmentation</i>	15
2.2.6	<i>Word Embedding</i>	15
2.2.6.1	Trainable Word Embedding	16
2.2.7	Deep Learning	16
2.2.8	TextCNN	16
2.2.8.1	<i>Hyperparameter Tuning CNN</i>	19
2.2.9	<i>Confusion Matrix</i>	21
BAB III	METODE PENELITIAN	26
3.1	Alur Penelitian	26
3.2	Penjabaran Langkah Penelitian	26
3.2.1	Identifikasi Masalah	27
3.2.2	Tinjauan Pustaka	28
3.2.3	Pengumpulan Dataset	28
3.2.4	Prapemrosesan Data	29
3.2.5	<i>Word Embedding</i>	29
3.2.6	<i>Modelling (TextCNN)</i>	30
3.2.7	Evaluasi Model	30
3.3	Alat dan Bahan Tugas Akhir	31
3.3.1	Alat	31
3.3.2	Bahan	31
3.4	Metode Pengembangan	32
3.5	Penjabaran Langkah Penelitian	32

3.5.1	<i>Data Collecting</i>	32
3.6	Validasi Dataset	33
3.7	Prapemrosesan Data	33
3.8	Pembagian Data	38
3.9	Arsitektur Model TextCNN	38
3.10	Ilustrai Perhitungan Metode	42
3.10.1	<i>Input Layer</i>	42
3.10.2	<i>Embedding Layer</i>	42
3.10.3	<i>Permute Operation</i>	42
3.10.4	<i>Convolutional Layer</i>	43
3.10.5	<i>Activation Layer (ReLU)</i>	43
3.10.6	<i>Max-Pooling Layer</i>	43
3.10.7	<i>Concatenation Layer</i>	43
3.10.8	<i>Dropout Layer</i>	44
3.10.9	<i>Fully Connected Layer</i>	44
3.10.10	<i>Output Layer</i>	44
3.11	Rancangan Pengujian	44
3.11.1	Skema Pengujian	44
DAFTAR PUSTAKA		46

DAFTAR TABEL

Tabel 2.1	Literasi Penelitian Terdahulu.....	10
Tabel 2.2	Literasi Penelitian Terdahulu.....	11
Tabel 2.3	Literasi Penelitian Terdahulu.....	20
Tabel 2.4	Confusion Matrix untuk Klasifikasi Biner.....	22
Tabel 3.1	Sample Isi Dataset.....	28
Tabel 3.2	Sample Dataset Awal.....	33
Tabel 3.3	Tahapan <i>Case Folding</i>	34
Tabel 3.4	Tahapan <i>Text Cleaning</i>	34
Tabel 3.5	Tahapan <i>Augmentation</i>	35
Tabel 3.6	Tahapan <i>Tokenization</i>	36
Tabel 3.7	Tahapan <i>Normalization</i>	37
Tabel 3.8	Tahapan <i>Stopword Removal</i>	37

DAFTAR GAMBAR

Gambar 2.1	Arsitektur model TextCNN.....	17
Gambar 3.1	Alur Penelitian.....	26
Gambar 3.2	Metode Pengembangan.....	32
Gambar 3.3	Pembagian Data.....	38

DAFTAR RUMUS

Rumus 2.1	<i>Convolution Operation</i>	17
Rumus 2.2	<i>Softmax Function</i>	18
Rumus 2.3	<i>Accuracy Formula</i>	23
Rumus 2.4	<i>Precision Formula</i>	24
Rumus 2.5	<i>Recall Formula</i>	24
Rumus 2.6	<i>F1-Score Formula</i>	25

DAFTAR KODE

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat telah membawa perubahan signifikan dalam cara manusia beraktivitas, khususnya dalam berkomunikasi. Berbagai teknologi baru bermunculan, seperti internet dan media sosial, yang memberikan kemudahan dalam berinteraksi. Namun, di balik kemudahan tersebut, muncul pula tantangan serius, salah satunya terkait dampaknya terhadap kesehatan mental [1]. Salah satu gangguan kesehatan mental yang muncul seiring kemajuan teknologi komunikasi, khususnya media sosial, adalah *cyberbullying*.

Cyberbullying adalah tindakan agresif yang disengaja dan berulang kali dilakukan menggunakan perangkat elektronik terhadap korban yang sulit membela diri [2]. Berdasarkan studi yang dilakukan oleh S. Hinduja *et al.*, *cyberbullying* memiliki beberapa kriteria, yaitu disengaja, berulang, dan berbahaya [3]. Dampak dari *cyberbullying* meliputi rasa malu, takut, dan terintimidasi yang dapat mengganggu kesehatan mental korban serta berpotensi menimbulkan depresi berlebih. Perilaku *cyberbullying* diamati sering terjadi di berbagai platform media sosial.

Perkembangan internet dalam bidang teknologi informasi telah menghadirkan berbagai alternatif media komunikasi, salah satunya media sosial. Salah satu platform sosial media yang paling populer adalah TikTok, yang dikembangkan oleh ByteDance dan menggunakan format video pendek sebagai media utama, sehingga memberikan pengalaman baru dibandingkan dengan media sosial lainnya. Berdasarkan laporan We Are Social tahun 2025, TikTok menempati peringkat ke-4 sebagai platform media sosial yang paling banyak digunakan di Indonesia [4]. Popularitasnya membuat TikTok digunakan

oleh jutaan orang dari berbagai usia, latar belakang, dan budaya [5]. Namun, seperti media sosial pada umumnya, karakteristik penggunaan bahasa di TikTok cenderung singkat, tidak baku, dan informal. Hal ini menimbulkan tantangan tersendiri dalam mendeteksi adanya komentar *cyberbullying*, karena proses identifikasi secara manual memerlukan waktu dan usaha yang besar.

Mengatasi masalah tersebut, B.A. Prameswari *et al* (2023) melalui penelitiannya menawarkan solusi berupa model analisis sentimen yang mampu mengklasifikasikan komentar *cyberbullying* dan *non-cyberbullying* [6]. Analisis sentimen memungkinkan proses identifikasi sentimen dalam teks, yang dapat dikategorikan menjadi sentimen positif dan negatif. Pendekatan modern dalam analisis sentimen banyak mengandalkan teknik *deep learning*. Metode ini, yang merupakan bagian dari *machine learning* dengan jaringan saraf tiruan berlapis-lapis, memiliki kemampuan yang baik dalam membaca pola kompleks dari data yang tidak terstruktur, termasuk data teks, tanpa perlu fitur yang dirancang secara manual [7]. Dengan kemampuan tersebut, model dapat menganalisis dan memahami teks secara lebih akurat secara otomatis.

Seiring dengan meningkatnya penggunaan *deep learning* dalam analisis sentimen, peran *Natural Language Processing* (NLP) juga menjadi krusial. NLP berfokus pada interaksi antara komputer dan bahasa manusia, memungkinkan mesin untuk memproses, menganalisis, memahami, dan menghasilkan bahasa alami. Hal ini sangat penting untuk mengurai makna di balik komentar-komentar pengguna dan mengidentifikasi sentimen yang terkandung di dalamnya [6]. Selain itu, kemajuan dalam arsitektur model *deep learning* telah meningkatkan kemampuan NLP dalam menangkap konteks dan nuansa bahasa yang kompleks. Implementasi NLP pada analisis sentimen tidak hanya membantu dalam klasifikasi polaritas teks, tetapi juga dapat digunakan untuk mendeteksi pola bahasa yang berpotensi mengarah pada perilaku tertentu seperti *cyberbullying*.

Penelitian sebelumnya juga telah mencoba mengatasi masalah analisis

sentimen *cyberbullying* dengan menggunakan arsitektur canggih seperti *Bidirectional Encoder Representations from Transformers* (BERT) melalui proses *fine-tuning* pada dataset spesifik. Model-model berbasis transformer seperti BERT memang menunjukkan performa tinggi dalam banyak tugas *Natural Language Processing* (NLP) [6]. Namun, hasil penelitian sebelumnya menunjukkan kecenderungan terjadinya *overfitting*, di mana model menjadi terlalu spesifik terhadap data pelatihan sehingga performanya menurun pada data baru yang belum pernah dilihat sebelumnya, terutama pada dataset dengan ukuran terbatas atau distribusi data yang tidak seimbang [8].

Melihat tantangan tersebut, arsitektur TextCNN muncul sebagai alternatif yang menjanjikan. Penelitian terdahulu terkait TextCNN telah membuktikan keunggulannya dalam mengekstraksi fitur penting dari rangkaian kata dan telah banyak digunakan dalam berbagai tugas klasifikasi teks, termasuk analisis sentimen serta deteksi spam atau misinformasi. Keunggulan utamanya adalah kemampuannya dalam mempelajari representasi fitur secara otomatis melalui *convolution layer*, yang efektif menangkap pola lokal dan global dalam teks [9]. Temuan ini menegaskan potensi penggunaan arsitektur TextCNN dalam melakukan analisis sentimen, khususnya untuk klasifikasi komentar *cyberbullying* di platform media sosial TikTok.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka permasalahan penelitian dirumuskan sebagai berikut:

1. Bagaimana mengklasifikasikan komentar yang termasuk dan tidak termasuk *cyberbullying* pada platform TikTok dengan arsitektur TextCNN?
2. Bagaimana menguji performa model TextCNN dalam mengklasifikasikan komentar yang termasuk dan tidak termasuk *cyberbullying*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan di atas, maka tujuan dari penelitian ini adalah:

1. Mengembangkan model klasifikasi untuk mendeteksi komentar yang termasuk dan tidak termasuk *cyberbullying* pada platform TikTok menggunakan arsitektur TextCNN.
2. Melakukan pengujian dan evaluasi performa model TextCNN dalam mengklasifikasikan komentar *cyberbullying* dan komentar non-*cyberbullying* menggunakan matriks evaluasi berdasarkan *confusion matrix*.

1.4 Batasan Masalah

Adapun batasan masalah dari penelitian ini agar sesuai dengan yang diharapkan adalah sebagai berikut:

1. Penelitian ini hanya menggunakan data komentar yang berasal dari platform TikTok.
2. Dataset komentar TikTok yang digunakan dalam penelitian ini diambil dari dataset yang telah dikumpulkan dan dipublikasikan oleh B.A. Prameswari *et al* (2023) [6].
3. Data yang dianalisis berjumlah 1.510 komentar dan seluruhnya menggunakan bahasa Indonesia.

1.5 Manfaat Penelitian

Adapun manfaat yang diperoleh dari hasil penelitian ini adalah sebagai berikut:

1. Memberikan kontribusi dalam pengembangan arsitektur TextCNN yang dapat diimplementasikan untuk mendeteksi komentar *cyberbullying* di media sosial, khususnya pada platform TikTok.
2. Menjadi acuan atau referensi dalam penerapan arsitektur TextCNN untuk

klasifikasi teks pendek, terutama pada komentar media sosial yang memiliki karakteristik bahasa informal dan ringkas.

1.6 Sistematika Penulisan

Sistematika penulisan berisi pembahasan apa yang akan ditulis disetiap Bab. Sistematika pada umumnya berupa paragraf yang setiap paragraf mencerminkan bahasan setiap Bab.

1.6.1 Bab I Pendahuluan

Bab ini membahas latar belakang yang melandasi penelitian, perumusan masalah yang ingin diselesaikan, serta tujuan yang ingin dicapai. Selain itu, dijelaskan juga batasan masalah, manfaat dari penelitian, dan sistematika penulisan sebagai panduan struktur laporan.

1.6.2 Bab II Tinjauan Pustaka

Bab ini menguraikan teori-teori yang menjadi dasar penelitian, seperti konsep *cyberbullying*, analisis sentimen, dan text processing. Juga dijelaskan arsitektur model TextCNN yang digunakan dalam penelitian ini.

1.6.3 Bab III

Bab ini menjelaskan metode yang digunakan dalam penelitian, mulai dari teknik pengumpulan data hingga tahapan prapemrosesan data. Selain itu, dibahas pula perancangan model TextCNN serta metode evaluasi performa model.

1.6.4 Bab IV

Bab ini menyajikan hasil pelatihan dan pengujian model yang telah dibangun. Dilengkapi dengan analisis performa dan interpretasi hasil klasifikasi sentimen untuk menilai keberhasilan model.

1.6.5 Bab V

Bab terakhir ini berisi kesimpulan dari hasil penelitian yang telah dilakukan. Penulis juga memberikan saran sebagai masukan untuk pengembangan penelitian di masa mendatang.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian ini mengacu serta menggunakan beberapa jurnal penelitian dan berbagai literatur yang diperoleh dari berbagai sumber sebagai dasar proses riset dan penulisan tugas akhir. Tinjauan pustaka akan mencakup berbagai informasi dan teori yang relevan dengan penelitian serta penulisan tugas akhir ini. Tinjauan Pustaka ini dapat dilihat pada Tabel 2.1 Tinjauan Pustaka.

Penelitian yang dilakukan oleh B.A. Prameswari *et al.* (2023) yaitu membangun model prediksi untuk mendeteksi komentar *cyberbullying* dari media sosial TikTok. Penulis secara independen mengumpulkan dan memberi label pada 1.510 komentar TikTok menjadi kategori *cyberbullying* (CB) dan *non-cyberbullying* (Non_CB). Pendekatan *deep learning* digunakan dalam penelitian ini, khususnya arsitektur *Bidirectional Encoder Representations from Transformers* (BERT), untuk proses analisis sentimen pada dataset. Setelah *fine-tuning*, model prediksi mencapai akurasi validasi 0,63 pada epoch kesembilan. Studi ini menekankan pentingnya membangun model prediksi untuk mendeteksi komentar *cyberbullying* dan berkontribusi pada pemahaman serta pencegahan perilaku ini di platform media sosial.

Penelitian yang dilakukan oleh D. Nugraha dan P. Astuti (2023) bertujuan untuk mengukur tingkat *cyberbullying* di Indonesia dengan menganalisis komentar di media sosial Instagram menggunakan algoritma *Support Vector Machine* (SVM). Penelitian ini mengumpulkan 400 data komentar *cyberbullying* dari Instagram, yang dibagi rata menjadi 200 data positif dan 200 data negatif. Data diproses melalui tahapan preprocessing seperti *cleansing*, *transform case*, *tokenize*, *stem*, *filter stopword*, dan *filter tokens by length* menggunakan alat RapidMiner. Model klasifikasi yang dihasilkan

menggunakan 400 dataset untuk pelatihan menunjukkan akurasi sebesar 84,25%, presisi 80,22%, recall 92,50%, dan nilai AUC sebesar 0,928.

Penelitian yang dilakukan oleh A.Z. Abdullah *et al.* (2025) bertujuan menganalisis sentimen publik terhadap Tweet Pemilu Presiden Indonesia 2024. Penulis mengumpulkan 62.955 entri dari Twitter, 126.673 dari IndoNews, dan dataset gabungan berjumlah 189.628 entri, yang kemudian diberi label positif, netral, dan negatif. Pendekatan model hibrida *Convolutional Neural Network* (CNN)-*Long Short-Term Memory* (LSTM) dengan perluasan fitur *Word2Vec* dan optimasi *Genetic Algorithm* (GA) digunakan untuk analisis sentimen. Model hibrida CNN-LSTM yang dioptimalkan dengan GA mencapai akurasi tertinggi 84,78% untuk data berita, menunjukkan peningkatan 3,59%. Studi ini mengilustrasikan penerapan inovatif model hibrida CNN-LSTM untuk analisis sentimen dalam konteks pemilihan nasional, meningkatkan akurasi dan efisiensi dalam memahami opini publik dan dinamika politik.

Penelitian yang dilakukan oleh S.A. Ardiyansa *et al.* (2024) bertujuan mengklasifikasi sentimen tweet berbahasa Indonesia pada platform Twitter. Penulis mengumpulkan 6.137 tweet dengan 5 jenis label berbeda (joy, sadness, fear, love, dan anger). Proses pra-pemrosesan data mencakup pembersihan, pengubahan ekspresi/emoji menjadi kata, *stemming*, dan *embedding* kata. Penelitian ini membandingkan beberapa arsitektur, termasuk BERT, LSTM, CNN, serta arsitektur gabungan Transformer-LSTM, LSTM-CNN, dan Transformer-CNN. Model *Transformer-CNN* menunjukkan performa paling unggul dengan akurasi 85,71% pada data uji dan 99,90% pada data latih. Akurasi ini lebih baik dibandingkan arsitektur lainnya. Meskipun waktu pelatihannya 30,17 menit lebih lama dari beberapa model lain, *Transformer-CNN* juga sudah konvergen pada iterasi ke-5. Studi ini menyimpulkan bahwa *Transformer-CNN* adalah pilihan terbaik untuk klasifikasi sentimen yang membutuhkan akurasi tinggi.

Penelitian yang dilakukan oleh S. Chen (2025) mengeksplorasi evolusi

teknik analisis sentimen dari model tradisional hingga pendekatan *deep learning* yang lebih canggih. Penulis menggunakan dataset IMDb yang berisi 50.000 ulasan film untuk klasifikasi sentimen biner (positif atau negatif), dengan 20% data pelatihan digunakan untuk validasi. Metodologi melibatkan perbandingan kinerja model seperti CNN, RNN, LSTM, LSTM-CNN, dan BERT menggunakan metrik *F-Score*, *Precision*, *Accuracy*, dan *Recall*. Model CNN dan BERT mencapai akurasi tertinggi 0,90, dengan CNN unggul dalam Recall (0,95) dan BERT menunjukkan kinerja seimbang. Model RNN memiliki akurasi terendah 0,68. Studi ini menyimpulkan bahwa CNN dan BERT adalah model *deep learning* yang berkinerja lebih baik untuk analisis sentimen dan menyoroti potensi serta tantangan analisis sentimen di masa depan, termasuk penanganan multi-bahasa dan isu etika.

Tabel 2.1 Literasi Penelitian Terdahulu

No.	Peneliti (tahun)	Judul Penelitian [sitasi]	Permasalahan	Ekstraksi Fitur	Metode Klasifikasi	Hasil Penelitian
1.	B.A. Prameswari et al (2023) [6]	<i>Building Prediction Model for Detecting Cyberbullying using TikTok Comments</i>	Deteksi cyberbullying pada komentar TikTok dengan arsitektur BERT	IndoBERT	BERT	Akurasi validasi model mencapai 0.63 pada epoch kesembilan
2.	D. Nugraha dan P. Astuti (2023) [10]	<i>Analisis Sentimen Cyberbullying Pada Media Sosial Instagram Menggunakan Metode Support Vector Machine</i>	Menganalisis sentimen <i>bullying online</i> di kolom komentar Instagram	TF-IDF	<i>Support Vector Machine</i>	Akurasi sebesar 84,25%, presisi 80,22%, recall 92,50%, dan nilai AUC sebesar 0,928
3.	A.Z. Abdullah et al. (2025) [11]	<i>Sentiment Analysis Accuracy for 2024 Indonesian Election Tweets Using CNN-LSTM With Genetic Algorithm Optimization</i>	Menganalisis sentimen terhadap Pemilu Indonesia 2024 untuk wawasan opini publik yang lebih dalam	<i>Word2Vec, TF-IDF</i>	<i>Hybrid CNN-LSTM, Genetic Algorithm (optimasi)</i>	Akurasi tertinggi 84,78% untuk data berita (peningkatan 3,59%)

Tabel 2.2 Literasi Penelitian Terdahulu

No.	Peneliti (tahun)	Judul Penelitian [sitasi]	Permasalahan	Ekstraksi Fitur	Metode Klasifikasi	Hasil Penelitian
4.	S.A. Ardiyansa <i>et al.</i> (2024) [12]	<i>Klasifikasi Sentimen Tweet dengan Arsitektur Hybrid Transformers-CNN pada Platform Twitter</i>	Mengklasifikasi sentimen tweet berbahasa Indonesia dengan akurasi tinggi, mengatasi tantangan volume data besar dan kompleksitas bahasa informal	IndoBERT	<i>Hybrid Transformer-CNN</i>	Akurasi 85,71% pada data uji dan 99,90% pada data latih. Konvergen pada iterasi ke-5.
5.	S. Chen (2025) [13]	<i>Sentiment Analysis Techniques for Deep Learning Classification and Comparison</i>	Membandingkan performa CNN, RNN, LSTM, LSTM-CNN, dan BERT pada tugas analisis sentimen biner menggunakan dataset IMDB.	Word2Vec	CNN, RNN, LSTM, LSTM-CNN dan BERT	Model CNN secara umum memperoleh <i>confussion matrix</i> yang lebih baik dari model lain dengan akurasi 90%, presisi 87%, recall 95%, specificity 84%, dan F1-score 91%

2.2 Dasar Teori

Penelitian ini didasarkan pada beberapa dasar teori yang berperan sebagai acuan dalam proses analisis dan pengembangan. Berikut adalah beberapa dasar teori yang digunakan sebagai acuan dalam pelaksanaan penelitian ini.

2.2.1 *Cyberbullying*

Cyberbullying merupakan bentuk perundungan yang terjadi di ranah digital, dengan pelaku biasanya menyembunyikan identitas di balik akun anonim. Tindakan ini dapat berupa hinaan, ancaman, pelecehan verbal, atau penyebaran informasi yang merugikan korban melalui berbagai media sosial dan platform digital [14]. Berbeda dengan *bullying* yang terjadi secara fisik, *cyberbullying* bersifat persisten, karena jejak digital yang ditinggalkan dapat bertahan dalam waktu lama dan menyebar lebih cepat.

2.2.2 Media Sosial

Media sosial adalah bentuk aplikasi berbasis internet yang memungkinkan penggunanya untuk menciptakan, berbagi, dan bertukar informasi secara interaktif dalam ruang digital [15]. Media sosial merupakan produk dari perkembangan Web 2.0 yang menekankan pada konten yang dihasilkan pengguna (user generated content) dan keterhubungan antarpengguna. Karakteristik utama media sosial meliputi interaktivitas, keterhubungan, serta visibilitas konten, sehingga ia tidak hanya berfungsi sebagai sarana komunikasi, tetapi juga sebagai ruang publik baru yang membentuk pola interaksi, identitas, dan komunitas dalam masyarakat digital.

2.2.2.1 TikTok

Menurut We Are Social, pada tahun 2025 TikTok adalah salah satu platform media sosial yang paling banyak digunakan di Indonesia [4]. Platform ini memungkinkan penggunanya untuk mengekspresikan kreativitas dalam pembuatan video dengan berbagai konten. Banyak pengguna, utamanya

generasi muda, menilai TikTok sebagai media hiburan untuk melihat video-video kreatif yang menghibur. Konten yang ditampilkan di TikTok beragam, dan penggunaanya dapat mengunggah atau melihat video yang terkadang sesuai umur maupun yang tidak sesuai umur. Sebagai media sosial, TikTok juga melibatkan elemen seperti teks, gambar, dan video [5].

2.2.3 *Natural Language Processing (NLP)*

Natural Language Processing (NLP) adalah cabang dari kecerdasan buatan yang mempelajari bagaimana komputer dapat memahami, memproses, dan menghasilkan bahasa manusia, baik dalam bentuk teks maupun ucapan. NLP menggabungkan teknik dari linguistik, ilmu komputer, dan pembelajaran mesin untuk mengubah data bahasa alami menjadi informasi yang dapat diolah oleh sistem. Ruang lingkupnya mencakup berbagai tugas, seperti penerjemahan otomatis, analisis sentimen, pengenalan suara, ringkasan teks, dan penjawaban pertanyaan. Proses NLP umumnya melibatkan tahapan seperti tokenisasi, penghapusan kata umum (stopword removal), stemming atau lemmatisasi, serta analisis sintaksis dan semantik untuk memahami struktur dan makna bahasa secara lebih mendalam [16].

2.2.4 Analisis Sentimen

Analisis sentimen atau sentiment analysis adalah suatu teknik dalam *Natural Language Processing (NLP)* yang bertujuan untuk mengetahui emosi, sikap, atau opini yang terkandung dalam suatu teks. Teknik ini berperan penting dalam memahami respons pengguna terhadap suatu topik, produk, atau layanan. Analisis sentimen ini dapat menganalisis komentar, ulasan, atau teks yang ditulis oleh pengguna, sistem dapat mengkategorikan apakah teks tersebut bersifat positif, negatif, atau netral [17]. Hasil klasifikasi ini dapat dimanfaatkan untuk pengambilan keputusan, evaluasi layanan, hingga perancangan strategi yang lebih tepat sasaran.

2.2.5 Text Pre-processing

Text pre-processing mencakup berbagai teknik untuk meningkatkan kualitas data agar lebih sesuai dalam proses analisis dan pelatihan model.

2.2.5.1 Normalization

Normalization adalah proses menyederhanakan dan menyeragamkan teks agar sesuai dengan kaidah bahasa standar. Tahap ini bertujuan untuk mengurangi variasi penulisan yang tidak relevan sehingga analisis data menjadi lebih konsisten dan akurat. Proses normalisasi mencakup berbagai langkah, seperti mengubah seluruh huruf kapital menjadi huruf kecil (case folding), menghapus tanda baca atau karakter khusus yang tidak diperlukan, menghilangkan spasi berlebih, serta mengonversi kata tidak baku atau singkatan menjadi bentuk baku [18]. Dengan melakukan normalisasi, data teks menjadi lebih terstruktur dan siap diproses pada tahap analisis berikutnya.

2.2.5.2 IndoBERT Tokenizer

IndoBERT Tokenizer adalah komponen pra-pemrosesan teks yang dikembangkan khusus untuk bahasa Indonesia sebagai bagian dari model IndoBERT. Tokenizer ini berfungsi untuk memecah teks bahasa Indonesia menjadi token-token yang dapat dipahami oleh model, dengan mempertimbangkan karakteristik linguistik khusus bahasa Indonesia. Proses tokenisasi melibatkan pemisahan teks menjadi subkata (subword) menggunakan algoritma WordPiece, yang memungkinkan penanganan kata-kata yang jarang muncul dengan lebih efektif. IndoBERT Tokenizer juga menangani berbagai fitur bahasa Indonesia seperti imbuhan, partikel, dan struktur morfologi yang kompleks, serta mampu mengkonversi teks ke dalam representasi numerik dengan menambahkan token khusus seperti [CLS] di awal dan [SEP] di akhir kalimat untuk keperluan klasifikasi dan pemisahan segmen teks.

2.2.5.3 *Augmentation*

Augmentation adalah proses meningkatkan variasi data pelatihan dengan cara menghasilkan data baru yang berasal dari data yang sudah ada, tanpa mengubah makna dasarnya. Teknik ini banyak digunakan dalam bidang *machine learning* untuk memperkaya set data dan meningkatkan kemampuan generalisasi model. Beberapa metode augmentasi yang umum diterapkan antara lain penggantian kata dengan sinonim (*synonym replacement*), pertukaran posisi kata (*word swapping*), penghapusan kata tertentu (*random deletion*), atau penerjemahan bolak-balik (*back translation*) untuk menghasilkan variasi kalimat dari teks asli [19]. Dengan menambahkan keragaman data, augmentasi dapat mengurangi risiko *overfitting* dan membantu model belajar pola yang lebih umum, terutama pada kasus di mana set data tidak seimbang atau jumlah data terbatas.

2.2.6 *Word Embedding*

Word embedding adalah proses merubah kata-kata menjadi representasi numerik agar dapat dipahami dan diolah oleh komputer. Word embedding merupakan teknik representasi kata dalam bentuk vektor berdimensi tetap, di mana setiap vektor menyimpan informasi tentang makna semantik serta hubungan antar kata dalam suatu ruang vektor [20]. Dengan pendekatan ini, kata-kata yang memiliki makna serupa akan direpresentasikan oleh vektor yang posisinya saling berdekatan, sehingga memungkinkan model untuk mengenali kesamaan konteks dan hubungan linguistik secara lebih efektif. Teknik word embedding dapat dilatih dari awal menggunakan dataset tertentu atau memanfaatkan model pra-latih seperti Word2Vec, GloVe, dan FastText yang telah dibangun dari korpus besar untuk meningkatkan kinerja model.

2.2.6.1 Trainable Word Embedding

Trainable word embedding adalah representasi kata dalam bentuk vektor berdimensi rendah yang dibangun secara langsung melalui proses pelatihan model. Berbeda dengan pretrained embedding seperti Word2Vec atau GloVe yang sudah dilatih sebelumnya di korpus besar, *trainable embedding* biasanya diinisialisasi secara acak dan nilainya diperbarui melalui mekanisme *backpropagation* selama model belajar dari data. Dengan cara ini, *embedding* yang dihasilkan bersifat adaptif dan spesifik terhadap tugas (task-specific), misalnya klasifikasi sentimen atau deteksi topik, sehingga mampu menangkap pola distribusi kata yang relevan dengan konteks dataset yang digunakan.

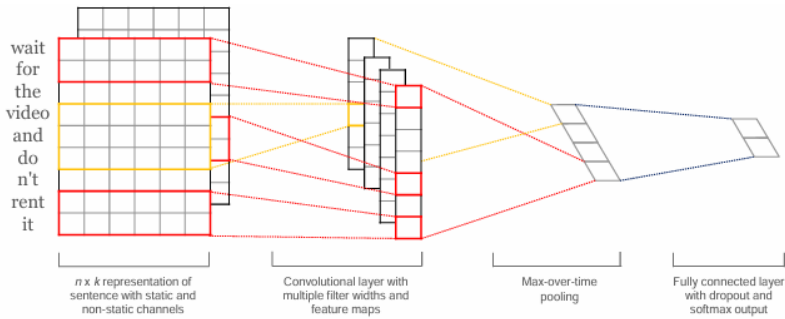
2.2.7 Deep Learning

Deep learning adalah pendekatan dalam pembelajaran mesin yang memanfaatkan jaringan saraf tiruan berlapis-lapis untuk mempelajari representasi data secara otomatis. Teknik ini mampu mengenali pola yang kompleks dan abstrak pada berbagai jenis data, termasuk teks, tanpa memerlukan perancangan fitur secara manual [21]. Dalam pemrosesan bahasa alami, deep learning telah melahirkan berbagai arsitektur model yang efektif, seperti Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Transformer, serta varian berbasis Convolutional Neural Network (CNN) seperti TextCNN. Setiap arsitektur memiliki keunggulan masing-masing, mulai dari kemampuan memahami urutan kata, mempertahankan konteks jangka panjang, hingga memproses teks secara paralel untuk efisiensi yang lebih tinggi.

2.2.8 TextCNN

TextCNN adalah varian dari arsitektur *Convolutional Neural Network* (CNN) yang merupakan bagian dari algoritma *deep learning* [22]. CNN adalah algoritma yang digunakan untuk memahami fitur atau karakteristik visual dalam gambar, sehingga dapat membedakan setiap gambar berdasarkan fitur yang

sudah dikenali [7]. CNN tidak hanya dapat digunakan pada gambar namun juga dapat digunakan pada teks dengan beberapa penyesuaian seperti perubahan dari konvolusi 2 dimensi menjadi konvolusi 1 dimensi yang merepresentasikan setiap kalimat [22].



Gambar 2.1 Arsitektur model TextCNN

1. Convolutional Layer

CNN merupakan jenis deep learning yang memanfaatkan convolutional layer sebagai penyusun neural network yang dibangun. Convolutional layer merupakan lapisan pertama dari tahap dalam arsitektur CNN [38]. Convolutional layer menggunakan pendekatan sliding window dan weight sharing untuk menyederhanakan proses perhitungan sehingga mempercepat proses pelatihan.

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b)$$

Rumus 2.1 Convolution Operation

Keterangan:

c_i = Hasil konvolusi pada posisi ke- i

$f(\cdot)$ = Fungsi aktivasi non-linear (misalnya ReLU)

\mathbf{w} = Vektor bobot filter (*kernel*)

$\mathbf{x}_{i:i+h-1}$ = Potongan vektor input dari posisi i hingga $i + h - 1$

b = Bias yang ditambahkan setelah operasi konvolusi

Pada lapisan ini digunakan fungsi aktivasi non-linear, seperti Rectified Linear Unit (ReLU), untuk menambahkan sifat non-linear sehingga jaringan dapat mempelajari pola yang lebih kompleks. Fungsi ini bekerja dengan memberikan keluaran nol apabila nilai masukan $\mathbf{x} \leq 0$, dan mengembalikan nilai masukan itu sendiri apabila $\mathbf{x} \geq 0$.

2. *Max Pooling Layer*

Max Pooling layer adalah komponen penting kedua dalam CNN setelah convolution layer. Lapisan ini berfungsi melakukan down-sampling secara non-linear untuk mereduksi ukuran data. Tujuannya adalah mengekstrak informasi yang paling relevan dari feature map dengan mengabaikan nilai-nilai yang kurang signifikan, sehingga beban komputasi dapat berkurang. Pada tugas akhir ini digunakan metode pooling, khususnya max pooling yang banyak diaplikasikan dalam pemrosesan bahasa alami (NLP). Max pooling bekerja dengan memilih nilai maksimum dari setiap area pada feature map.

3. *Fully Connector Softmax layer*

Setelah lapisan *fully connected* memproses fitur, fungsi softmax diterapkan pada output untuk mengubah skor menjadi distribusi probabilitas. Proses ini menghitung setiap skor secara eksponensial lalu menormalkannya dengan jumlah seluruh skor. Hasil akhirnya adalah probabilitas untuk tiap kelas dengan total selalu bernilai satu. Fungsi ini menghasilkan total probabilitas setiap kelas sejumlah 1 sehingga efektif untuk klasifikasi multi-kelas.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}, \quad i = 1, 2, \dots, K$$

Rumus 2.2 *Softmax Function*

Keterangan:

\mathbf{z} = Vektor skor input (logit) sebelum normalisasi

z_i = Skor input untuk kelas ke- i

K = Jumlah kelas

$\sigma(\mathbf{z})_i$ = Probabilitas kelas ke- i setelah normalisasi

e = Bilangan eksponensial alami (≈ 2.71828)

Ini berarti bahwa setelah lapisan fully connected memproses fitur-fitur yang dihasilkan dari lapisan sebelumnya, sebuah fungsi softmax diterapkan pada outputnya. Fungsi softmax ini bertugas mengubah nilai-nilai numerik (skor) menjadi distribusi probabilitas dari label atau kelas yang ada. Proses ini dilakukan dengan mengekspresikan setiap skor dalam bentuk nilai eksponensial, lalu menormalkannya terhadap jumlah seluruh nilai eksponensial, sehingga diperoleh probabilitas yang proporsional. Output akhirnya adalah probabilitas untuk setiap kategori, yang menunjukkan seberapa besar kemungkinan input termasuk dalam kategori tertentu. Dengan demikian, jumlah total probabilitas untuk semua kelas akan selalu bernilai satu, menjadikan softmax sangat berguna pada tugas klasifikasi multi-kelas.

2.2.8.1 *Hyperparameter Tuning CNN*

Pemilihan hyperparameter yang tepat merupakan langkah krusial dalam membangun model deep learning yang efektif. Hyperparameter adalah parameter yang ditentukan sebelum proses pelatihan dimulai dan memiliki pengaruh signifikan terhadap performa model. Nilai hyperparameter yang digunakan masih mengacu pada pengaturan default yang diambil dari jurnal terdahulu [9]. Penelitian ini tetap melakukan proses hyperparameter tuning untuk mencari kombinasi nilai terbaik guna meningkatkan performa model. Pemilihan kombinasi dilakukan secara bertahap dengan mempertimbangkan

hasil evaluasi dari masing-masing percobaan. Pada Tabel 3.1 adalah Hyperparameter yang akan digunakan.

Tabel 2.3 Literasi Penelitian Terdahulu

<i>Kernel Size</i>	<i>Learning Rate</i>	<i>Batch Size</i>	<i>Epoch</i>
[3, 4]	0.0001	16	[50]

Berikut adalah penjelasan mengenai masing-masing hyperparameter yang digunakan:

1. *Kernel Size*

Kernel size adalah ukuran dimensi *kernel* atau jendela konvolusi yang digunakan untuk memproses data input. Ukuran ini menunjukkan jumlah elemen *input* yang dipertimbangkan secara bersamaan pada setiap operasi konvolusi, di mana *kernel* akan bergerak melintasi *input* sesuai dengan langkah pergeseran (*stride*) yang ditentukan. Ukuran *kernel* memengaruhi jumlah nilai *input* yang digunakan untuk menghasilkan satu nilai keluaran pada *feature map*. Pada arsitektur TextCNN ini akan menggunakan dua kernel size yaitu 3 dan 4 untuk menangkap *feature map* yang berbeda dari setiap teks. Disini digunakan nilai awal *kernel size* 3 dan 4 karena mampu menangkap detail halus dari komentar TikTok yang kebanyakan berupa kalimat singkat.

2. *Learning Rate*

Learning rate adalah salah satu *hyperparameter* pada proses pelatihan model yang mengatur besar langkah pembaruan bobot pada setiap iterasi. Nilai *learning rate* menentukan seberapa cepat atau lambat model menyesuaikan bobotnya selama proses pembelajaran. Pada penelitian ini menggunakan nilai *learning rate* awal 0.0001 untuk dapat memperoleh optimum global yang sulit dicapai jika *learning rate* terlalu besar dan

lama dicapai jika menggunakan *learning rate* yang terlalu kecil.

3. *Batch Size*

Batch size adalah jumlah sampel data yang diproses dalam satu iterasi selama pelatihan model. *Batch size* kecil dapat mempercepat proses pelatihan, namun berpotensi menurunkan akurasi model. Sebaliknya, *batch size* besar umumnya dapat meningkatkan akurasi, tetapi memerlukan waktu pelatihan yang lebih lama. Pada penelitian ini nilai batch size awal yang digunakan adalah 16, ini dilakukan agar proses training tidak terlalu lama dan nilai akurasi yang diperoleh juga cukup baik.

4. *Epoch*

Epoch adalah satu siklus penuh pelatihan model terhadap seluruh dataset. Semakin besar jumlah *epoch*, semakin banyak peluang model untuk mempelajari informasi dari data. Namun, jumlah epoch yang terlalu tinggi dapat menyebabkan *overfitting*, yaitu kondisi saat model terlalu menghafal data latih sehingga kesulitan mengenali pola pada data baru. Sebaliknya, jumlah epoch yang terlalu sedikit dapat mengakibatkan *underfitting*, di mana model belum cukup mempelajari pola yang terdapat pada data pelatihan. Di penelitian ini menggunakan nilai awal *epoch* 50 untuk memberi ruang agar terhindar dari optimum lokal yang dapat terjadi jika nilai epoch terlalu kecil.

2.2.9 *Confusion Matrix*

Confusion matrix merupakan salah satu metode evaluasi yang sering digunakan dalam analisis sentimen. Metode ini berbentuk tabel yang berfungsi menilai kinerja model dengan membandingkan hasil prediksi terhadap nilai aktual dari dataset. Pada klasifikasi biner, confusion matrix terdiri dari empat komponen utama, yaitu true positive (TP), true negative (TN), false positive

(FP), dan false negative (FN). Berdasarkan confusion matrix, berbagai metrik evaluasi seperti akurasi, presisi, recall, dan F1-score dapat dihitung untuk mengukur performa model secara lebih mendetail.

Dalam evaluasi model klasifikasi biner, perhitungan metrik seperti precision, recall, dan F1-score dilakukan secara langsung berdasarkan nilai TP, TN, FP, dan FN tanpa perlu menggunakan metode rata-rata seperti pada kasus multi-kelas. Pendekatan ini memberikan hasil evaluasi yang lebih sederhana dan jelas, karena hanya terdapat dua kelas yang dianalisis. Evaluasi pada klasifikasi biner dapat lebih fokus dalam menilai kemampuan model membedakan antara kelas positif dan negatif.

Tabel 2.4 Confusion Matrix untuk Klasifikasi Biner

Aktual	Prediksi	
	Positif	Negatif
Positif	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
Negatif	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

Berdasarkan Tabel 2.2, berikut adalah penjelasannya:

1. *True Positive (TP)*: Banyaknya data yang memang termasuk kategori positif dan berhasil diidentifikasi dengan benar sebagai positif oleh model.
2. *True Negative (TN)*: Banyaknya data yang sebenarnya tergolong negatif dan berhasil dikenali dengan tepat sebagai negatif oleh model.
3. *False Positive (FP)*: Banyaknya data yang sebenarnya negatif namun salah diklasifikasikan oleh model sebagai positif.
4. *False Negative (FN)*: Banyaknya data yang sebenarnya positif namun keliru diprediksi oleh model sebagai negatif.

Pada penelitian ini digunakan parameter tersebut untuk mengukur

accuracy, *precision*, *recall* dan *f1-score* dengan rumus sebagai berikut.

1. *Accuracy*

Accuracy merupakan ukuran yang menunjukkan sejauh mana model mampu mengklasifikasikan data dengan tepat. Metode ini sering digunakan untuk memberikan gambaran umum tentang kinerja model secara keseluruhan. Nilai *accuracy* dihitung dengan membandingkan jumlah prediksi yang benar dengan total jumlah data uji yang tersedia. Semakin besar persentase *accuracy*, semakin baik pula kemampuan model dalam mengenali dan memprediksi pola dari data yang dianalisis. Klasifikasi biner:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Rumus 2.3 *Accuracy Formula*

Keterangan:

TP = *True Positive*

FP = *False Positive*

TN = *True Negative*

2. *Precision*

Precision adalah ukuran yang menunjukkan tingkat ketepatan model dalam mengklasifikasikan data positif. Metode ini menilai seberapa banyak prediksi positif yang benar dibandingkan dengan seluruh data yang diprediksi sebagai positif oleh model. Nilai *precision* yang tinggi mengindikasikan bahwa sebagian besar prediksi positif memang benar positif. Dengan demikian, *precision* membantu menilai seberapa andal model dalam menghindari kesalahan klasifikasi positif palsu.

Precision Biner:

$$Precision = \frac{TP}{TP + FP}$$

Rumus 2.4 *Precision Formula*

Keterangan:

$TP = True\ Positive$

$FP = False\ Positive$

3. *Recall*

Recall merupakan ukuran yang menunjukkan kemampuan model dalam mengenali data yang benar pada suatu kelas tertentu. Pengukuran ini menilai sejauh mana model dapat menemukan seluruh data relevan yang termasuk dalam kelas tersebut. Nilainya diperoleh dengan membandingkan jumlah data yang berhasil diklasifikasikan dengan benar dengan total data aktual pada kelas tersebut. Semakin tinggi nilai recall, semakin baik model dalam menangkap seluruh data positif yang seharusnya terdeteksi.

Recall Biner:

$$Recall = \frac{TP}{TP + FN}$$

Rumus 2.5 *Recall Formula*

Keterangan:

$TP = True\ Positive$

$FN = False\ Negative$

4. *F1-score*

F1-score adalah ukuran yang menggabungkan precision dan recall

menjadi satu nilai tunggal. Metrik ini menghitung rata-rata tertimbang dari kedua nilai tersebut agar keseimbangan keduanya dapat dinilai. Tujuan *F1-score* adalah untuk memberikan gambaran performa model secara keseluruhan, khususnya ketika ketepatan dan kemampuan deteksi perlu diseimbangkan. Dengan demikian, *F1-score* menjadi indikator penting dalam evaluasi model klasifikasi.

F1-Score Biner:

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Rumus 2.6 *F1-Score Formula*

Keterangan:

Precision = Tingkat ketepatan prediksi positif yang dihasilkan model

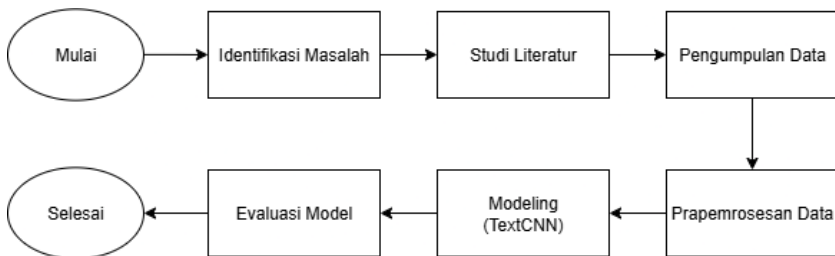
Recall = Tingkat keberhasilan model dalam menemukan seluruh data positif

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Penelitian ini disusun melalui serangkaian langkah yang terstruktur untuk mencapai tujuan yang telah ditetapkan. Fokus utama penelitian adalah memberikan solusi yang dapat diterapkan terhadap permasalahan yang ada, yaitu dengan merancang dan mengimplementasikan sistem klasifikasi komentar *cyberbullying* pada platform TikTok. Pendekatan yang digunakan mengombinasikan metode *Text Convolutional Neural Network* (TextCNN), di mana TextCNN dimanfaatkan untuk mengekstraksi fitur-fitur penting dari teks komentar, sekaligus mempelajari pola sekuensial dalam teks guna meningkatkan akurasi klasifikasi. Set data pada penelitian ini akan melalui beberapa tahap pemrosesan, meliputi prapemrosesan teks, ekstraksi fitur, serta pelatihan model. Alur lengkap dari proses penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

3.2 Penjabaran Langkah Penelitian

Untuk memperjelas setiap langkah-langkah yang telah didefinisikan pada Gambar 3.1, berikut ini akan dijelaskan secara rinci tahapan-tahapan yang dilakukan dalam penelitian ini.

3.2.1 Identifikasi Masalah

Tahap awal dalam penelitian ini dimulai dengan melakukan identifikasi terhadap permasalahan yang menjadi fokus utama. Penelitian ini bertujuan untuk mengklasifikasikan komentar *cyberbullying* berdasarkan data yang diperoleh dari media sosial *TikTok* dengan menggunakan metode *TextCNN*. Permasalahan utama yang diangkat adalah kesulitan dalam mengenali komentar *cyberbullying* pada kolom komentar *TikTok*, mengingat gaya bahasa yang digunakan di media sosial umumnya bersifat informal, singkat, dan sering kali ambigu.

Seiring meningkatnya penggunaan *TikTok* sebagai sarana berekspresi, risiko munculnya tindakan *cyberbullying* pun semakin besar, sehingga diperlukan sebuah model yang mampu mempelajari pola bahasa untuk melakukan klasifikasi komentar *cyberbullying*. Penelitian ini bertujuan untuk mengembangkan model klasifikasi yang dapat memetakan konten teks ke dalam dua kategori, yaitu *cyberbullying* dan *non-cyberbullying*.

Data yang digunakan pada penelitian ini diperoleh dari penelitian sebelumnya yang dilakukan oleh B.A. Prameswari *et al.*, yang menyediakan kumpulan komentar *TikTok* dengan dua label, yakni *cyberbullying* dan *non-cyberbullying* [6]. Pengembangan model dilakukan menggunakan metode *TextCNN*, yang berperan dalam mengekstraksi fitur penting dari teks dan mempelajari hubungan sekuensial antar kata dalam kalimat untuk kemudian mengklasifikasikannya ke dalam dua label tersebut. Model ini akan dilatih menggunakan data yang telah melalui proses prapemrosesan, sehingga diharapkan dapat memberikan prediksi yang lebih akurat dalam mengklasifikasikan komentar *cyberbullying* berdasarkan pola bahasa yang digunakan oleh pengguna *TikTok*.

3.2.2 Tinjauan Pustaka

Pada tahap ini, dilakukan tinjauan terhadap berbagai penelitian terdahulu yang membahas klasifikasi komentar *cyberbullying* berbasis media sosial, serta penelitian lain yang menggunakan metode serupa, khususnya dengan penerapan model *TextCNN*. Tinjauan ini melibatkan beragam sumber referensi, termasuk *paper*, jurnal, dan laporan penelitian yang relevan. Hasil dari studi literatur tersebut menjadi landasan penting bagi peneliti dalam memahami konsep, merancang metodologi, serta memperkuat kerangka teori yang mendasari penelitian ini.

3.2.3 Pengumpulan Dataset

Pada penelitian ini, proses pengumpulan data tidak dilakukan secara langsung, melainkan menggunakan *dataset* sekunder yang diperoleh dari penelitian sebelumnya oleh Prameswari *et al.* [6]. *Dataset* tersebut terdiri atas 1.510 baris data dengan dua kolom utama, yaitu kolom *sentiment* yang memuat dua label enumerasi: -1 untuk kategori *cyberbullying* dan 1 untuk kategori *non-cyberbullying*, serta kolom *comment* yang berisi teks komentar. Contoh data pada Tabel 3.1 diambil dari berbagai baris data untuk representasi data yang lebih baik. Data ini selanjutnya akan diolah dan dimanfaatkan sebagai dasar dalam proses pelatihan model *CNN-LSTM* untuk melakukan klasifikasi emosi pada unggahan di media sosial.

Tabel 3.1 Sample Isi Dataset

No Baris	Sentiment	Comment
1	-1	Makannya segentong buset
2	-1	Mirip ursula di little mermaid
474	1	Pede dulu, glow up belakangan
538	1	Dihh keren banget

No Baris	Sentiment	Comment
550	1	Pentingnya peradaban juga sis

3.2.4 Prapemrosesan Data

Data dari *dataset* yang diperoleh melalui sumber sekunder diproses terlebih dahulu melalui tahap prapemrosesan sebelum dilakukan analisis. Tahap prapemrosesan ini mencakup serangkaian teknik yang bertujuan untuk meningkatkan kualitas data sehingga lebih optimal digunakan dalam proses analisis dan pelatihan model. Beberapa langkah yang dilakukan meliputi *case folding* untuk mengonversi seluruh teks menjadi huruf kecil, *text cleaning* untuk menghapus karakter atau simbol yang tidak diperlukan, serta *normalization* untuk menormalkan penggunaan kata tidak baku atau *slang* dan menyeragamkan format kata. Selain itu, dilakukan pula *stopword removal* guna menghilangkan kata-kata yang tidak memberikan makna signifikan pada analisis. Setelah tahap prapemrosesan selesai, data kemudian dibagi menjadi dua bagian, yaitu 80% untuk pelatihan (*training*) dan 20% untuk pengujian (*testing*).

3.2.5 Word Embedding

Data yang telah melewati tahap prapemrosesan selanjutnya diproses menggunakan teknik *word embedding* untuk mengubah teks menjadi representasi numerik yang dapat dipahami oleh model pembelajaran mesin. Pada penelitian ini, *word embedding* diimplementasikan menggunakan kelas `torch.nn.Embedding` dari pustaka PyTorch. Lapisan Embedding ini bekerja dengan memetakan setiap kata dalam *dataset* ke dalam representasi vektor berdimensi tetap berdasarkan indeks kata.

Proses dimulai dengan membangun *vocabulary*, yaitu daftar kata unik yang diperoleh dari *dataset*. Setiap kata dalam *vocabulary* diberikan indeks numerik, kemudian indeks tersebut dimasukkan ke dalam lapisan Embedding

untuk mendapatkan representasi vektor. Bobot awal pada lapisan *Embedding* diinisialisasi secara acak dan akan diperbarui selama proses pelatihan model, sehingga representasi vektor yang dihasilkan dapat menyesuaikan dengan pola data dan meningkatkan kinerja model dalam klasifikasi.

3.2.6 *Modelling (TextCNN)*

Pada tahap ini, dilakukan proses klasifikasi terhadap data yang telah melewati tahap ekstraksi fitur agar model dapat mengenali pola-pola penting yang berkaitan dengan klasifikasi emosi. Proses klasifikasi dilakukan menggunakan arsitektur *TextCNN*, yang dirancang khusus untuk menangani data teks.

TextCNN bekerja melalui beberapa tahapan utama. Pertama, dilakukan proses *convolution* menggunakan beberapa filter dengan ukuran kernel yang berbeda untuk mengekstraksi berbagai fitur penting dari data teks. Setiap filter bertujuan untuk menangkap pola kata dan frasa yang memiliki peran signifikan dalam penentuan kelas. Selanjutnya, hasil konvolusi melewati proses *max-pooling* yang berfungsi untuk mereduksi dimensi fitur tanpa kehilangan informasi esensial, sehingga membuat model lebih efisien dan fokus pada fitur paling relevan.

Setelah tahap *pooling*, hasilnya digabungkan dan diteruskan ke lapisan *fully connected* untuk menghasilkan prediksi akhir. Dataset dibagi menjadi dua bagian, yaitu *training data* yang digunakan untuk melatih model, serta *testing data* yang digunakan untuk mengevaluasi kinerja model dalam mengklasifikasikan data baru.

3.2.7 *Evaluasi Model*

Setelah model berhasil dibangun, tahap berikutnya adalah melakukan proses evaluasi untuk menilai kinerjanya dalam mengklasifikasikan emosi pada data berbasis media sosial. Evaluasi dilakukan dengan menghitung beberapa

metrik utama, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*.

Accuracy digunakan untuk mengukur sejauh mana model mampu mengklasifikasikan data secara benar secara keseluruhan. *Precision* menilai tingkat ketepatan model dalam memprediksi data yang termasuk dalam kategori tertentu, sedangkan *recall* mengukur sejauh mana model dapat menangkap seluruh data yang seharusnya termasuk dalam kategori tersebut. Sementara itu, *F1-score* merupakan nilai harmonisasi antara *precision* dan *recall*, sehingga memberikan gambaran yang lebih seimbang mengenai performa model.

3.3 Alat dan Bahan Tugas Akhir

Adapun alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

3.3.1 Alat

Berikut adalah alat yang digunakan dalam penelitian ini:

1. Pada penelitian ini penulis menggunakan laptop dengan spesifikasi Sistem Operasi Windows 11 Home, AMD Ryzen 5 5600H, Memori 24GB DDR4, SSD 500GB.
2. *Visual Studio Code* v1.101.2, *Miniconda* v24.5.0, *Python* v3.12.4

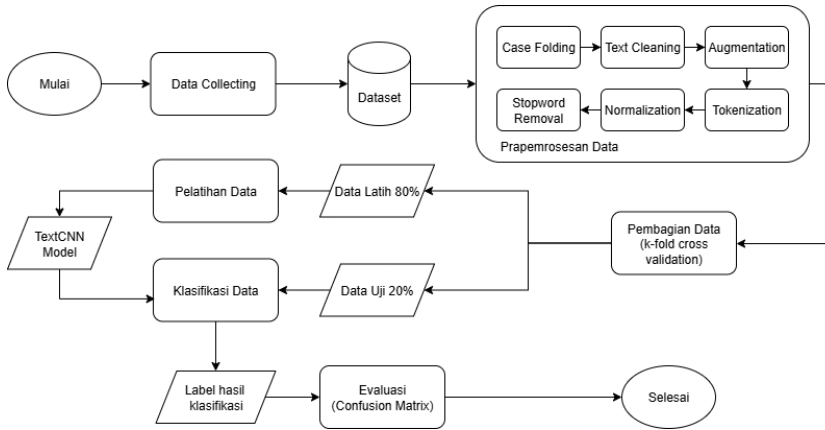
3.3.2 Bahan

Berikut adalah bahan yang digunakan dalam penelitian ini:

1. *Dataset* yang digunakan dalam penelitian ini merupakan *dataset* sekunder yang diambil dari penelitian yang dilakukan oleh B.A. Prameswari *et al.*, dengan judul *Building Prediction Model for Detecting Cyberbullying using TikTok Comments* [6].
2. Jurnal penelitian dari studi sebelumnya digunakan sebagai acuan untuk memberikan landasan teori, serta menyusun konsep dan gagasan yang mendukung pelaksanaan penelitian ini.

3.4 Metode Pengembangan

Seluruh proses yang akan dilakukan dalam klasifikasi emosi menggunakan TextCNN dapat dilihat pada Gambar 3.2.



Gambar 3.2 Metode Pengembangan

3.5 Penjabaran Langkah Penelitian

Langkah penelitian ini diuraikan berdasarkan gambar ... diatas.

3.5.1 Data Collecting

Dataset yang digunakan pada penelitian ini terdiri dari 1.510 baris komentar TikTok yang diklasifikasikan ke dalam dua kategori sentimen, yaitu *cyberbullying* dan *non-cyberbullying*. Dataset tersebut disajikan dalam format berkas CSV. Setiap baris data berisi komentar *TikTok* beserta label sentimen dalam bentuk nilai enumerasi, di mana nilai -1 menunjukkan *cyberbullying* dan nilai 1 menunjukkan *non-cyberbullying*. Gambaran umum mengenai dataset dapat dilihat pada Tabel 3.2 yang menampilkan contoh data awal.

Tabel 3.2 Sample Dataset Awal

Sentiment	Comment
-1	Makannya segentong buset
-1	Mirip ursula di little mermaid
...	...
1	Pentingnya peradaban juga sis
1	Pede dulu, glow up belakangan

3.6 Validasi Dataset

Validasi dataset merupakan proses pemeriksaan dan evaluasi untuk memastikan kualitas serta kelayakan data yang digunakan dalam penelitian. Tahap ini bertujuan untuk menjamin bahwa dataset bebas dari kesalahan yang berpotensi memengaruhi hasil analisis maupun kinerja model.

3.7 Prapemrosesan Data

Prapemrosesan data merupakan serangkaian tahapan yang dirancang untuk mempersiapkan serta meningkatkan kualitas data teks agar siap digunakan pada proses analisis lebih lanjut. Tahapan ini umumnya mencakup konversi seluruh huruf menjadi huruf kecil (*case folding*), penghapusan karakter atau simbol yang tidak diperlukan (*text cleaning*), penambahan variasi data melalui proses augmentasi (*data augmentation*), pemecahan teks menjadi satuan-satuan kata atau token (*tokenization*), penyeragaman ejaan kata tidak baku atau slang (*normalization*), serta penghapusan kata-kata yang kurang relevan terhadap analisis (*stopword removal*).

1. Case Folding

Case folding merupakan proses penyamaan format huruf pada teks dengan cara mengonversi seluruh karakter berhuruf besar (*uppercase*) menjadi huruf kecil (*lowercase*).

Tabel 3.3 Tahapan *Case Folding*

Sentiment	Comment	Hasil
-1	Cowo paling alay lebay se antero selebgram WKWKW NGAKAK	cowo paling alay lebay se antero selebgram wkwk ngakak
-1	KU KIRA MUKA TERNYATA AMPELA	ku kira muka ternyata ampela

2. *Text Cleaning*

Text Cleaning merupakan proses pembersihan teks dengan cara menghilangkan kata, karakter, atau elemen yang tidak diperlukan. Pada tahap ini, komponen yang dianggap tidak relevan atau dapat mengganggu analisis, seperti simbol, tanda baca, emoji, dan *hashtag*, akan dihapus dari teks.

Tabel 3.4 Tahapan *Text Cleaning*

Sentiment	Comment	Hasil
-1	udah sok tau salah pula @Ibnu Wardani ????	udah sok tau salah pula ibnu wardani
1	Old money tapi masih pedro sama coach becanda abangnya nih??	old money tapi masih pedro sama coach becanda abangnya nih

3. *Augmentation*

Augmentation merupakan teknik yang digunakan untuk memperbanyak jumlah data dengan cara memodifikasi data yang sudah ada tanpa mengubah makna utamanya. Dalam konteks pemrosesan teks, metode ini dapat dilakukan melalui beberapa cara, seperti sinonimisasi (mengganti kata dengan sinonimnya), penghapusan kata tertentu, penambahan kata, atau

penyusunan ulang kata dalam kalimat. Tujuan dari *augmentation* adalah meningkatkan variasi data sehingga model dapat belajar lebih baik dan menghasilkan performa klasifikasi yang lebih akurat.

Tabel 3.5 Tahapan *Augmentation*

Sentiment	Comment	Hasil	Kata/Frasa yang Diubah	
			Kata Asli	Kata Hasil
-1	ngeliat mukanya aja udah gedeg	ngeliat mukanya aja udah kesal	gedeg	kesal
1	gue gapaham dr dulu fashion style dia kyk gmna wkwkwk	gue gapaham dr duul fashion style dia kyk gmna wkwkwk	dulu	duul
-1	sebelum mangap di sikat dulu kale	sebelm mangap di sikat dulu kale	sebelum	sebelm
1	ya begitulah kalo kita bekerja sama orang arab atau keturunan timur tengah	ya begitluah kalo kita bekerja sama orang arab atau keturunan timur tengah	begitulah	begitluah

Sentiment	Comment	Hasil	Kata/Frasa yang Diubah	
			Kata Asli	Kata Hasil
-1	rama gak usah ganggu aku sama karya ya tolong	rama gausah ganggu aku sama karya ya tolong	gak usah	gausah

4. *Tokenization*

Tokenizing merupakan tahap pemecahan teks menjadi unit-unit terkecil, seperti kata, frasa, atau kalimat. Tahapan ini berperan penting dalam membantu sistem komputer memahami struktur dan makna dari teks yang sedang dianalisis.

Tabel 3.6 Tahapan *Tokenization*

Sentiment	Comment	Hasil
-1	ngeliat mukanya aja udah kesal	'ngeliat', 'mukanya', 'aja', 'udah', 'kesal'
1	aku penggemar mi burung dara enaknya nyambung terus	'aku', 'penggemar', 'mi', 'burung', 'dara', 'enaknya', 'nyambung', 'terus'

5. *Normalization*

Normalization adalah proses penyesuaian dan perbaikan ejaan pada kata-kata. Kata-kata yang tidak baku, termasuk singkatan, bahasa gaul, dan istilah slang, akan diubah menjadi bentuk yang sesuai dengan kaidah penulisan Bahasa Indonesia yang benar, merujuk pada pedoman yang tercantum dalam Kamus Besar Bahasa Indonesia (KBBI).

Tabel 3.7 Tahapan *Normalization*

Sentiment	Comment	Hasil
-1	gue gapaham dr dulu fashion style dia kyk gmna wkwkwk	saya tahu dari dulu fashion style dia seperti apa wkwkwk
1	Kalen Kalen Pada Jahad Ma Aku	kalian kalian pada jahat kepada saya

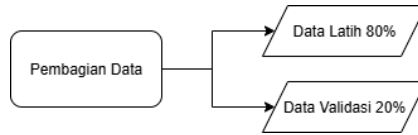
6. *Stopword Removal*

Stopword removal merupakan proses penghapusan kata-kata yang sering muncul dalam jumlah banyak namun dianggap tidak memiliki makna penting. Kata-kata yang umum tetapi kurang relevan, seperti konjungsi, kata kepunyaan, dan kata ganti orang, akan dihapus dari teks.

Tabel 3.8 Tahapan *Stopword Removal*

Sentiment	Comment	Hasil	Kata yang Dihapus
1	buatlah cita cita kamu setinggi bintang di langit tetapi jangan lupa untuk membuat anak tangga perencanaannya	buatlah cita cita setinggi bintang langit jangan lupa membuat anak tangga perencanaannya	'kamu', 'di', 'tetapi', 'untuk'
-1	minimal cuci muka lah goblok bikin sakit mata liatnya	minimal cuci muka goblok bikin sakit mata liatnya	'lah'

3.8 Pembagian Data



Gambar 3.3 Pembagian Data

Pada Gambar 3.3 menunjukkan pembagian dataset dilakukan dengan rasio 80% untuk data latih dan 20% untuk data validasi. Data latih digunakan untuk melatih model, sementara data validasi berfungsi untuk menguji kinerja model yang telah dilatih. Pembagian data menggunakan k-fold cross-validation dapat diperoleh evaluasi yang lebih akurat mengenai seberapa baik model dapat melakukan generalisasi terhadap data yang belum pernah dilihat sebelumnya.

3.9 Arsitektur Model TextCNN

TextCNN merupakan model klasifikasi teks berbasis algoritma *Convolutional Neural Network* (CNN) yang dirancang khusus untuk memproses data teks. Pada arsitektur ini, representasi teks terlebih dahulu dikonversi ke dalam bentuk vektor melalui *embedding layer*, kemudian diproses menggunakan beberapa filter konvolusi dengan ukuran berbeda untuk mengekstraksi fitur penting dari teks. Hasil ekstraksi fitur ini kemudian diringkas menggunakan *max-pooling layer* sebelum diteruskan ke *fully connected layer* dan diakhiri dengan lapisan *output* untuk menghasilkan prediksi. Berikut penjelasan rinci setiap lapisan:

1. *Embedding Layer*

Lapisan *embedding* berfungsi untuk mengonversi representasi kata berbentuk indeks menjadi vektor berdimensi kontinu yang merepresentasikan makna semantik kata. Misalnya, kata “sedih” dan “depresi” akan dipetakan menjadi vektor-vektor yang posisinya berdekatan dalam ruang vektor. Pada penelitian ini, digunakan modul

`torch.nn.Embedding` dari *PyTorch* untuk membangun representasi kata. Pendekatan ini melakukan pelatihan *embedding* secara end-to-end bersamaan dengan pelatihan model utama sehingga representasi kata yang dihasilkan lebih kontekstual dan menyesuaikan dengan karakteristik data yang digunakan.

2. *Convolutional Layer*

Tahapan awal pada proses feedforward adalah lapisan konvolusi satu dimensi (Conv1D). Pada tahap ini, data masukan yang telah direpresentasikan dalam bentuk vektor kata dari lapisan *embedding* akan diproses menggunakan operasi konvolusi satu dimensi untuk mengekstraksi fitur-fitur penting dari urutan kata. Setiap *filter* pada Conv1D akan menghasilkan *feature map* yang menyoroti pola-pola tertentu dalam teks, seperti kombinasi kata atau frasa yang memiliki makna semantik serupa. Setelah itu, fungsi aktivasi ReLU (Rectified Linear Unit) diterapkan untuk menambahkan sifat non-linearitas serta memastikan hanya fitur-fitur yang paling relevan yang dipertahankan. Pendekatan ini membuat model mampu menangkap ciri-ciri lokal dalam teks tanpa terlalu bergantung pada posisi absolut kata dalam kalimat.

3. *Activation Layer (ReLU)*

Setelah data teks diproses melalui lapisan konvolusi satu dimensi (Conv1D), hasil ekstraksi fitur langsung diteruskan ke fungsi aktivasi ReLU (Rectified Linear Unit). Pada tahap ini, nilai keluaran dari konvolusi yang bernilai negatif akan diubah menjadi nol, sedangkan nilai positif dibiarkan apa adanya. Pendekatan ini membuat model mampu mempelajari pola-pola non-linear dan mempertahankan fitur-fitur yang paling relevan. Pada implementasi model, proses aktivasi ReLU diterapkan bersamaan dengan Max-Pooling. Operasi ini memilih nilai fitur maksimum pada setiap *feature map*, sehingga hanya informasi paling dominan yang dipertahankan dari setiap filter konvolusi. Dengan cara

ini, model dapat mengekstraksi representasi paling informatif dari teks sebelum hasilnya digabungkan (concatenate) dan diteruskan ke lapisan berikutnya.

4. *Max-Pooling Layer*

Setelah melalui tahap konvolusi dan fungsi aktivasi ReLU, proses feedforward dilanjutkan dengan lapisan max-pooling. Pada tahap ini, digunakan metode Max-Pooling yang berfungsi untuk memilih nilai maksimum dari setiap *feature map* yang dihasilkan oleh lapisan konvolusi. Operasi ini dilakukan menggunakan perintah `.max(dim=2)`, di mana nilai maksimum diambil pada dimensi urutan kata (sequence length). Pendekatan ini memastikan bahwa hanya fitur paling dominan dari setiap filter konvolusi yang dipertahankan, sementara fitur-fitur dengan nilai lebih rendah diabaikan. Dengan cara ini, model menjadi lebih fokus pada informasi yang paling penting dan tidak bergantung pada posisi kata dalam teks. Selain itu, penggunaan Max-Pooling juga mengurangi dimensi data secara signifikan, sehingga proses komputasi pada lapisan berikutnya, seperti Concatenate Layer dan Fully Connected Layer, menjadi lebih efisien.

5. *Concatenate Layer*

Setelah melalui tahap Max-Pooling, setiap filter konvolusi menghasilkan vektor representasi yang menggambarkan fitur paling dominan dari masing-masing ukuran kernel. Karena model ini menggunakan beberapa ukuran kernel konvolusi, diperlukan tahap concatenate untuk menggabungkan seluruh vektor hasil ekstraksi fitur menjadi satu representasi tunggal. Proses penggabungan ini memungkinkan model untuk mengintegrasikan informasi dari berbagai pola n-gram yang ditangkap oleh setiap filter Conv1D. Misalnya, satu filter dapat mengekstraksi pola dua kata (bigram), sedangkan filter lainnya dapat menangkap pola tiga kata (trigram) atau lebih. Dengan melakukan

concatenate, model dapat memanfaatkan seluruh fitur penting yang diperoleh dari berbagai tingkat konteks, sehingga representasi teks menjadi lebih kaya dan informatif. Vektor gabungan ini kemudian diteruskan ke lapisan fully connected (dense layer) untuk tahap klasifikasi akhir. Dengan cara ini, Concatenate Layer berperan penting dalam mengintegrasikan hasil ekstraksi fitur dari berbagai jalur konvolusi menjadi satu kesatuan informasi sebelum diproses lebih lanjut.

6. *Dropout Layer*

Setelah melalui tahap concatenate, model dilanjutkan dengan lapisan dropout. Lapisan ini berfungsi untuk mengurangi overfitting dengan cara secara acak menonaktifkan sejumlah neuron selama proses pelatihan. Dengan demikian, model diharapkan dapat belajar representasi yang lebih robust dan generalizable.

7. *Fully Connected Layer*

Setelah melalui tahap *concatenate* dan *dropout*, data kemudian diteruskan ke lapisan *fully connected (dense layer)*. Pada tahap ini, seluruh fitur yang telah diekstraksi dan digabungkan sebelumnya akan diproses untuk menghasilkan prediksi akhir. Lapisan *fully connected* berfungsi untuk mengintegrasikan informasi dari seluruh fitur yang ada dan menentukan kelas output berdasarkan pola-pola yang telah dipelajari selama pelatihan.

8. *Output Layer*

Lapisan output merupakan tahap akhir dalam arsitektur model TextCNN. Pada tahap ini, hasil dari lapisan fully connected akan diproses untuk menghasilkan prediksi akhir mengenai kelas sentimen dari teks yang dianalisis. Dalam konteks klasifikasi biner, seperti pada penelitian ini yang membedakan antara kategori *cyberbullying* dan *non-cyberbullying*, lapisan output biasanya menggunakan fungsi aktivasi sigmoid. Fungsi ini mengubah nilai keluaran menjadi probabilitas antara 0 dan 1, di mana nilai mendekati 1 menunjukkan prediksi positif (misalnya, *non-*

cyberbullying), sedangkan nilai mendekati 0 menunjukkan prediksi negatif (misalnya, *cyberbullying*). Dengan demikian, lapisan output berperan penting dalam menentukan hasil akhir klasifikasi berdasarkan fitur-fitur yang telah diekstraksi dan diproses oleh lapisan-lapisan sebelumnya.

3.10 Ilustrai Perhitungan Metode

Pada tahap ini, dilakukan ilustrasi perhitungan metode *TextCNN* yang digunakan dalam penelitian ini. Ilustrasi ini mencakup langkah-langkah utama dalam proses klasifikasi teks, mulai dari representasi kata hingga prediksi akhir.

3.10.1 *Input Layer*

Komentar ditokenisasi menggunakan *tokenizer* IndoBERT sehingga menghasilkan deretan token ID. *Contoh*:

$$\text{"Makannya segentong buset"} \mapsto [245, 678, 934]$$

3.10.2 *Embedding Layer*

Proses *word embedding* memproyeksikan setiap token ID kedalam ruang vektor berdimensi 300. Berikut perhitungannya:

$$245 \mapsto \mathbf{x}_1 \in \mathbb{R}^{300}, \quad 678 \mapsto \mathbf{x}_2 \in \mathbb{R}^{300}, \quad \dots$$

Hasil embedding seluruh token ID akan membentuk matriks:

$$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix}, \quad X \in \mathbb{R}^{3 \times 300}$$

3.10.3 *Permute Operation*

Agar sesuai dengan format Conv1D PyTorch, matriks embedding diubah dari bentuk (*seq_len*, *embed_dim*) menjadi (*embed_dim*, *seq_len*):

$$X' \in \mathbb{R}^{300 \times 3}$$

3.10.4 Convolutional Layer

Filter konvolusi dengan kernel size 2 dan 4 diterapkan pada X' . Misalnya untuk kernel size = 2:

$$s_i = W \cdot x_{i:i+1} + b$$

dengan $W \in \mathbb{R}^{2 \times 300}$.

3.10.5 Activation Layer (ReLU)

Untuk mengeliminasi nilai negatif dan mempertahankan nilai positif, hasil konvolusi s_i dilewatkan ke fungsi aktivasi ReLU:

$$c_i = f(s_i) = \max(0, s_i)$$

Contoh hasil *feature map* setelah ReLU:

$$C_1 = [0.2, 0.9, 0.5]$$

$$C_2 = [0.4, 0.7, 0.2]$$

3.10.6 Max-Pooling Layer

Ambil nilai maksimum dari setiap *feature map* untuk merangkum informasi paling penting:

$$\text{MaxPool}(C_1) = \max(0.2, 0.9, 0.5) = 0.9$$

$$\text{MaxPool}(C_2) = \max(0.4, 0.7, 0.2) = 0.7$$

Sehingga output pooling adalah:

$$\begin{bmatrix} 0.9 \\ 0.7 \end{bmatrix}$$

3.10.7 Concatenation Layer

Gabungkan hasil pooling dari semua filter menjadi satu vektor fitur gabungan:

$$\text{Fitur Gabungan} = \begin{bmatrix} 0.9 \\ 0.7 \end{bmatrix} = [0.9, 0.7]$$

Jika ada lebih banyak filter, tambahkan hasil pooling mereka ke vektor ini.

3.10.8 *Dropout Layer*

Dengan dropout rate 0.1 (10% neuron dimatikan acak saat training).

$$\text{Fitur Gabungan Setelah Dropout} = [0.9, 0.7]$$

3.10.9 *Fully Connected Layer*

Vektor dropout z diproyeksikan dengan bobot W dan bias b :

$$\text{Output} = W \cdot z + b$$

Misalnya hasil perhitungan adalah:

$$\text{Output} = \begin{bmatrix} 1.086 \\ 0.888 \end{bmatrix}$$

3.10.10 *Output Layer*

Aktivasi sigmoid diterapkan untuk mendapatkan probabilitas kelas:

$$\sigma(1.086) \approx 0.747 \rightarrow \text{kelas 1 (cyberbullying)}$$

$$\sigma(0.888) \approx 0.708 \rightarrow \text{kelas 0 (non-cyberbullying)}$$

Kelas prediksi adalah kelas dengan probabilitas tertinggi, yaitu kelas 1 (*cyberbullying*) dengan probabilitas 0.747.

3.11 *Rancangan Pengujian*

Rancangan pengujian dilakukan secara sistematis agar hasil evaluasi bersifat andal, terukur, dan dapat direplikasi.

3.11.1 *Skema Pengujian*

Pada tahap ini, dilakukan proses pengujian untuk menilai kinerja model yang telah dibangun. Pengujian dilakukan dengan menggunakan 5 fold data

yang telah dipisahkan sebelumnya sebagai data latih dan data validasi. Hasil prediksi dari model akan dibandingkan dengan label asli untuk menghitung metrik evaluasi seperti *accuracy* yang dihitung berdasarkan rumus Rumus 2.3, *precision* yang dihitung berdasarkan rumus Rumus 2.4, *recall* yang dihitung berdasarkan rumus Rumus 2.5, dan *F1-score* yang dihitung berdasarkan rumus Rumus 2.6. Metrik-metrik ini memberikan gambaran mengenai seberapa baik model dalam mengklasifikasikan data baru dan membantu mengidentifikasi area yang perlu ditingkatkan.

DAFTAR PUSTAKA

- [1] Queene Br Sembiring. “Dampak Teknologi Digital Terhadap Kesehatan Mental Generasi Muda”. *Circle Archive* 1.4 (2024).
- [2] Yessi Mareta Andari Putri et al. “Cyberbullying di media sosial tiktok terhadap remaja sekolah menengah pertama”. *Jurnal common* 7.1 (2023), pp. 33–44.
- [3] Sameer Hinduja and Justin W Patchin. “The Role of Hope in Bullying and Cyberbullying Prevention”. *Frontiers in Sociology* 10 (), p. 1576372.
- [4] Simon Kemp. *Digital 2025: Indonesia*. Diakses: 11 Agustus 2025. 2025.
- [5] Puput Silva Rosiana et al. “Analisis aplikasi tiktok berdasarkan prinsip dan paradigma interaksi manusia dan komputer menggunakan evaluasi heuristic”. *Jurnal Informatika dan Teknik Elektro Terapan* 11.3 (2023).
- [6] Bunga Aura Prameswari et al. “Building Prediction Model for Detecting Cyberbullying using TikTok Comments”. *2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*. 2023, pp. 1–7.
- [7] Shiyang Liao et al. “CNN for situations understanding based on sentiment analysis of twitter data”. *Procedia computer science* 111 (2017), pp. 376–381.
- [8] Xue Ying. “An overview of overfitting and its solutions”. *Journal of physics: Conference series*. Vol. 1168. IOP Publishing. 2019, p. 022022.
- [9] Yoon Kim. “Convolutional neural networks for sentence classification”. *arXiv preprint arXiv:1408.5882* (2014).

- [10] Damar Nugraha and Puji Astuti. "Analisis sentimen cyberbullying pada sosial media Instagram menggunakan metode Support Vector Machine". *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS: Journal of Information System* 8.2 (2023), pp. 153–164.
- [11] Athallah Zacky Abdullah and Erwin Budi Setiawan. "Sentiment Analysis Accuracy for 2024 Indonesian Election Tweets Using CNN-LSTM With Genetic Algorithm Optimization". *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi* 9.1 (2025), pp. 1–14.
- [12] Safrizal Ardana Ardiyansa et al. "Klasifikasi Sentimen Tweet dengan Arsitektur Hybrid Transformers-CNN pada Platform Twitter". *The Indonesian Journal of Computer Science* 14.3 (2025).
- [13] Shuaiyu Chen. "Sentiment Analysis Techniques for Deep Learning Classification and Comparison". *CONF-CIAP 2025: Proceedings of the 4th International Conference on Computing Innovation and Applied Physics*. Ed. by Omer Burak Istanbul Marwan Omar Anil Fernando. 2025.
- [14] Ahya Ghina Qolbya, Aleissya Sahira Siswandi, and Raissa Dwifandra Putri. "Empati dan Cyberbullying pada Remaja Pengguna Media Sosial: Sebuah Kajian Literatur". *Flourishing Journal* 3.9 (2023), pp. 352–359.
- [15] Sapta Sari. "Literasi media pada generasi milenial di era digital". *Professional: Jurnal komunikasi dan administrasi publik* 6.2 (2019), pp. 30–42.
- [16] Yuan-Fu Liao and Yih-Ru Wang. "Some experiences on applying deep learning to speech signal and natural language processing". *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*. IEEE. 2018, pp. 83–94.

- [17] Agung Pambudi and Suprpto Suprpto. “Effect of sentence length in sentiment analysis using support vector machine and convolutional neural network method”. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)* 15.1 (2021), pp. 21–30.
- [18] Ruishuang Wang et al. “Convolutional recurrent neural networks for text classification”. *2019 international joint conference on neural networks (IJCNN)*. IEEE. 2019, pp. 1–6.
- [19] Jason Wei and Kai Zou. “EDA: Easy data augmentation techniques for boosting performance on text classification tasks”. *arXiv preprint arXiv:1901.11196* (2019).
- [20] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. *Advances in neural information processing systems* 26 (2013).
- [21] Ian Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.
- [22] Ronan Collobert et al. “Natural language processing (almost) from scratch.” *Journal of machine learning research* 12.7 (2011).