

**KLASIFIKASI KOMENTAR BERMUATAN PROMOSI JUDI
ONLINE PADA DATA YOUTUBE MENGGUNAKAN
ARSITEKTUR TEXT CNN**

TUGAS AKHIR

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

Oleh:

Amdhan Anggoro

121140226



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

LEMBAR PENGESAHAN

Saya menyatakan bahwa Tugas Akhir berjudul “Klasifikasi Komentar Bermuatan Promosi Judi Online pada Data YouTube Menggunakan Arsitektur Text CNN” merupakan hasil karya saya sendiri dan belum pernah diajukan, baik sebagian maupun seluruhnya, di Institut Teknologi Sumatera atau institusi pendidikan lain oleh saya maupun pihak lain.

Lampung Selatan, 14-11-2025

Penulis,

Amdhan Anggoro

NIM. 121140226

Foto 2x3

Diperiksa dan disetujui oleh,

Pembimbing

1. Martin Clinton Tosima Manullang, Ph.D.

NIP. 19930109 2019 03 1 017

.....

2. Martin Clinton Tosima Manullang, Ph.D.

NIP. 19930109 2019 03 1 017

.....

Penguji

1. Rahman Indra Kesuma, S.Kom., M.Cs.

NIP. 19910530 2019 03 1 018

.....

2. Imam Ekowicaksono, S.Si., M.Si.

NIP. 19890517 2019 03 1 013

.....

Disahkan oleh,

Koordinator Program Studi Teknik Informatika

Fakultas Teknologi Industri

Institut Teknologi Sumatera

Andika Setiawan, S.Kom., M.Cs.

NIP. 19911127 2022 03 1 007

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir dengan judul “Klasifikasi Komentar Bermuatan Promosi Judi Online pada Data YouTube Menggunakan Arsitektur Text CNN” adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.

Nama : Amdhan Anggoro

NIM : 121140226

Tanda Tangan :

Tanggal :

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama : Amdhan Anggoro

NIM : 121140226

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Jenis Karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Sumatera **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

Klasifikasi Komentar Bermuatan Promosi Judi Online pada Data YouTube

Menggunakan Arsitektur Text CNN

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Lampung Selatan

Pada tanggal : 14-11-2025

Yang menyatakan

Amdhan Anggoro

KATA PENGANTAR

Pada halaman ini mahasiswa berkesempatan untuk menyatakan terima kasih secara tertulis kepada pembimbing dan pihak lain yang telah memberi bimbingan, nasihat, saran dan kritik, kepada mereka yang telah membantu melakukan penelitian, kepada perorangan atau lembaga yang telah memberi bantuan keuangan, materi dan/atau sarana. Cara menulis kata pengantar beraneka ragam, tetapi hendaknya menggunakan kalimat yang baku. Ucapan terima kasih agar dibuat tidak berlebihan dan dibatasi pada pihak yang terkait secara ilmiah (berhubungan dengan subjek/materi penelitian).

Puji syukur kehadiran Allah SWT/Tuhan Yang Maha Esa atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga penyusunan tugas akhir ini telah terselesaikan dengan baik. Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. [Rektor ITERA] selaku Rektor Institut Teknologi Sumatera.
2. [Dekan FTI] selaku Dekan Fakultas Teknologi Industri.
3. [Koor Prodi IF] selaku Ketua Program Studi Teknik Informatika.
4. [Dosen Pembimbing] selaku Dosen Pembimbing atas ide, waktu, tenaga, perhatian, dan masukan yang telah disumbangsihkan kepada penulis.
5. [Isi nama lainnya]

Akhir kata penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi kita semua.

RINGKASAN

Klasifikasi Komentar Bermuatan Promosi Judi Online pada Data YouTube

Menggunakan Arsitektur Text CNN

Amdhan Anggoro

Fenomena meningkatnya promosi judi online pada kolom komentar YouTube menimbulkan kekhawatiran terkait penyalahgunaan platform digital, terutama karena pola bahasa yang digunakan sering dibuat terselubung dan sulit dikenali oleh sistem moderasi konvensional. Berdasarkan permasalahan tersebut, penelitian ini merumuskan kebutuhan model klasifikasi otomatis yang mampu mendeteksi komentar bermuatan promosi judi online secara akurat. Tujuan penelitian mencakup pembangunan arsitektur model yang efektif, evaluasi pengaruh hiperparameter terhadap performa model, serta penetapan konfigurasi terbaik yang mampu melakukan generalisasi pada berbagai variasi komentar.

Metodologi penelitian meliputi proses pengumpulan data komentar YouTube, pra-pemrosesan teks melalui case folding, penghilangan tanda baca, normalisasi slang, tokenisasi, dan augmentasi data untuk memperkaya keragaman sampel. Data yang telah bersih kemudian dibagi menjadi data latih dan uji sebelum digunakan dalam pengembangan arsitektur JOC-TextCNN, yaitu modifikasi TextCNN yang memanfaatkan tiga cabang konvolusi—standard, dilated, dan depthwise-separable—serta mekanisme Squeeze-and-Excitation untuk memperkuat ekstraksi fitur. Evaluasi dilakukan melalui studi ablasi yang menguji perubahan learning rate, pemilihan optimizer, dan variasi dropout guna menganalisis kontribusi masing-masing parameter terhadap performa model.

Hasil penelitian menunjukkan bahwa konfigurasi terbaik diperoleh dengan optimizer AdamW, learning rate $5e-5$, dan dropout 0.2, menghasilkan Akurasi 0.9587, Presisi 0.9594, Recall 0.9588, dan F1-Score 0.9587. Analisis data menunjukkan bahwa learning rate memiliki pengaruh signifikan terhadap stabilitas konvergensi model, sementara AdamW terbukti lebih unggul

dibandingkan Muon karena kemampuan adaptasinya terhadap arsitektur multi-branch. Model JOC-TextCNN secara keseluruhan mampu mengenali pola bahasa promosi yang kompleks dan tersembunyi, sehingga dianggap efektif untuk tugas klasifikasi komentar promosi judi online. Kesimpulan penelitian menegaskan bahwa arsitektur ini memiliki kemampuan generalisasi yang baik, sementara saran untuk penelitian selanjutnya mencakup perluasan studi ablasi pada konfigurasi filter dan SE block, serta peningkatan kualitas dataset melalui deduplikasi dan penambahan sampel yang lebih variatif.

ABSTRAK

Klasifikasi Komentar Bermuatan Promosi Judi Online pada Data YouTube

Menggunakan Arsitektur Text CNN

Amdhan Anggoro

Maraknya penyebaran promosi judi online pada kolom komentar YouTube menimbulkan risiko bagi pengguna, terutama karena konten tersebut sering disamarkan melalui variasi bahasa dan pola penulisan tertentu. Penelitian ini bertujuan membangun model klasifikasi yang mampu mendeteksi komentar bermuatan promosi judi online secara akurat menggunakan arsitektur JOC-TextCNN, sebuah modifikasi dari TextCNN yang menggabungkan multi-branch convolution (standard, dilated, dan depthwise-separable) serta mekanisme Squeeze-and-Excitation. Metodologi penelitian meliputi pengumpulan data komentar YouTube, proses pra-pemrosesan dan augmentasi teks, perancangan model, serta pengujian melalui studi ablasi untuk mengevaluasi pengaruh hiperparameter. Hasil penelitian menunjukkan bahwa konfigurasi terbaik diperoleh pada model yang menggunakan optimizer AdamW, learning rate $5e-5$, dan dropout 0.2, dengan performa Akurasi 0.9587, Presisi 0.9594, Recall 0.9588, dan F1-Score 0.9587. Temuan ini menegaskan bahwa arsitektur JOC-TextCNN efektif dalam menangkap pola bahasa promosi yang kompleks dan mampu menghasilkan generalisasi yang baik. Model yang dikembangkan dapat menjadi dasar pengembangan sistem deteksi otomatis untuk membantu mitigasi peredaran konten promosi judi online di platform digital.

Kata Kunci: *TextCNN, Judi Online, Klasifikasi Teks, YouTube, Deep Learning*

ABSTRACT

Classification of Online Gambling Promotional Comments on YouTube Data

Using Text CNN Architecture

Amdhan Anggoro

The rampant spread of online gambling promotions in the YouTube comment column poses a risk to users, especially since the content is often disguised through variations in certain languages and writing patterns. This study aims to build a classification model that is able to accurately detect comments containing online gambling promotions using the JOC-TextCNN architecture, a modification of TextCNN that combines multi-branch convolution (standard, dilated, and depthwise-separable) and Squeeze-and-Excitation mechanisms. The research methodology includes the collection of YouTube comment data, the pre-processing and augmentation of the text, the design of the model, and testing through ablation studies to evaluate the influence of hyperparameters. The results showed that the best configuration was obtained in the model using the AdamW optimizer, learning rate $5e-5$, and dropout 0.2, with an Accuracy performance of 0.9587, Precision 0.9594, Recall 0.9588, and F1-Score 0.9587. These findings confirm that the JOC-TextCNN architecture is effective in capturing complex promotional language patterns and is capable of generating good generalizations. The developed model can be the basis for the development of an automatic detection system to help mitigate the circulation of online gambling promotional content on digital platforms.

Keywords: *TextCNN, Online Gambling, Text Classification, YouTube, Deep Learning*

DAFTAR ISI

LEMBAR PENGESAHAN	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PERSETUJUAN PUBLIKASI	iv
KATA PENGANTAR	v
RINGKASAN	vi
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR RUMUS	xvi
DAFTAR KODE	xvii
BABI PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	5
1.4 Batasan Masalah	5
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA	8
2.1 Tinjauan Pustaka	8
2.2 Dasar Teori	15

2.2.1	Deteksi Konten Teks Berbasis Klasifikasi	15
2.2.2	<i>Natural Language Processing (NLP)</i>	16
2.2.3	Komentar YouTube dan Judi <i>Online</i>	16
2.2.4	Fenomena Judi <i>Online</i> di Media Sosial	18
2.2.5	<i>Dataset</i>	18
2.2.6	<i>Text Processing</i>	19
2.2.7	<i>Deep Learning</i>	21
2.2.8	Arsitektur <i>Text Convolutional Neural Network</i>	22
2.2.8.1	Input Layer	23
2.2.8.2	<i>Word Embedding Layer</i>	23
2.2.8.3	<i>Convolutional Layer</i>	23
2.2.8.4	Activation Function (ReLU)	24
2.2.8.5	<i>Max-Pooling Layer</i>	24
2.2.8.6	Concatenation Layer	25
2.2.8.7	<i>Fully Connected Layer</i>	25
2.2.8.8	<i>Output Layer (Sigmoid)</i>	26
2.2.9	<i>Loss Function</i>	26
2.2.10	<i>Hyperparameter</i>	26
2.2.11	<i>Feature Extraction</i>	27
2.2.12	<i>Confusion Matrix</i>	27
2.2.12.1	<i>Accuracy</i>	28
2.2.12.2	<i>Precision</i>	29
2.2.12.3	<i>Recall</i>	29
2.2.12.4	<i>F1-Score</i>	30
BAB III METODE PENELITIAN		31
3.1	Alur Penelitian	31
3.2	Penjabaran Langkah Penelitian	32
3.2.1	Identifikasi Masalah	32

3.2.2	Studi Literatur	32
3.2.3	Pengumpulan Data	32
3.2.4	<i>Pre-Processing</i> dan Augmentasi Data	34
3.2.5	Pengembangan Model	35
3.2.6	Pengujian	36
3.2.7	Evaluasi	36
3.3	Alat dan Bahan Tugas Akhir	36
3.3.1	Alat	36
3.3.2	Bahan	37
3.4	Metode Pengembangan Model	37
3.4.1	<i>Input Dataset</i>	38
3.4.2	<i>Data Labeling</i>	39
3.4.3	<i>Preprocessing Data</i>	40
3.4.3.1	<i>Case Folding</i>	40
3.4.3.2	<i>URL Removal</i>	40
3.4.3.3	<i>Punctuation Cleaning</i>	41
3.4.3.4	<i>Repeated Character Reduction</i>	41
3.4.3.5	<i>Slang Word Normalization</i>	42
3.4.3.6	<i>Tokenization</i>	42
3.4.3.7	<i>Stopword Removal</i>	43
3.4.3.8	<i>Stemming</i>	44
3.4.4	Split Data	44
3.4.5	Pembuatan Model	45
3.4.5.1	<i>Input Layer</i>	47
3.4.5.2	<i>Embedding Layer</i>	47
3.4.5.3	<i>Unsqueeze Operation</i>	47
3.4.5.4	<i>Permute Operation</i>	47
3.4.5.5	<i>Convolutional Layer</i>	48
3.4.5.6	<i>Triple Branch Architecture</i>	48

3.4.5.7	<i>Activation Function (ReLU)</i>	49
3.4.5.8	<i>Max-Pooling Layer</i>	49
3.4.5.9	<i>Concatenation Layer</i>	49
3.4.5.10	<i>SE Block (Squeeze-and-Excitation)</i>	49
3.4.5.11	<i>Dropout</i>	49
3.4.5.12	<i>Fully Connected Layer</i>	50
3.4.5.13	<i>Output Layer (Sigmoid)</i>	50
3.4.6	<i>Hyperparameter Tuning</i>	50
3.5	<i>Ilustrasi Perhitungan</i>	51
3.5.1	<i>Input Layer</i>	51
3.5.2	<i>Embedding Layer</i>	51
3.5.3	<i>Permute Operation</i>	52
3.5.4	<i>Convolutional Layer</i>	52
3.5.5	<i>Activation Layer (ReLU)</i>	52
3.5.6	<i>Max Pooling</i>	53
3.5.7	<i>Concatenation</i>	53
3.5.8	<i>Dropout</i>	53
3.5.9	<i>Fully Connected Layer</i>	53
3.5.10	<i>Output Layer(Sigmoid)</i>	54
3.6	<i>Rancangan Pengujian</i>	54
3.6.1	<i>Studi Ablasi</i>	54
3.6.1.1	<i>Konsep Studi Ablasi</i>	54
3.6.1.2	<i>Skenario Pengujian</i>	54
3.6.1.3	<i>Skema Pengujian</i>	56
BAB IV HASIL DAN PEMBAHASAN		57
4.1	<i>Hasil</i>	57
4.1.1	<i>Pengumpulan Data</i>	57
4.1.2	<i>Pre-Processing dan Augmentasi Data</i>	57

- 4.1.2.1 *Pre-Processing Data* 57
 - 4.1.2.2 Augmentasi Data 58
 - 4.1.3 Pengembangan Model 59
 - 4.1.3.1 Perbandingan Model 63
 - 4.1.3.2 Pemilihan Model 64
 - 4.1.4 Pengujian Model 65
 - 4.1.4.1 *Baseline Model*..... 66
 - 4.1.4.2 Perubahan *Learning Rate*..... 66
 - 4.1.4.3 Perubahan *Optimizer*..... 67
 - 4.1.4.4 Perubahan *Dropout* 67
 - 4.1.5 Evaluasi..... 68
- 4.2 Pembahasan 68
- BAB V KESIMPULAN DAN SARAN 70**
 - 5.1 Kesimpulan..... 70
 - 5.2 Saran 71
- DAFTAR PUSTAKA 72**
- LAMPIRAN 77**
 - A Dataset 77
 - B Hasil Wawancara 77
 - C Rincian Kasus Uji 77

DAFTAR TABEL

Tabel 2.1	Literasi Penelitian Terdahulu.....	10
Tabel 2.2	Sumber Dataset	19
Tabel 3.1	Pengumpulan Dataset.....	32
Tabel 3.2	Label Dataset	33
Tabel 3.3	Data Labeling.....	39
Tabel 3.4	Proses <i>Case Folding</i>	40
Tabel 3.5	Proses <i>URL Removal</i>	41
Tabel 3.6	Proses <i>Punctuation Cleaning</i>	41
Tabel 3.7	Proses <i>Repeated Character Reduction</i>	42
Tabel 3.8	Proses <i>Slang Word Normalization</i>	42
Tabel 3.9	Proses <i>Tokenization</i>	43
Tabel 3.10	Proses <i>Stopword Removal</i>	43
Tabel 3.11	Proses <i>Stemming</i>	44
Tabel 3.12	Hyperparameter Model Dasar (Ringan, Sedang, Berat) dan Model JOC	51
Tabel 4.1	Hyperparameter Model Dasar (Ringan, Sedang, Berat) dan Model JOC	64
Tabel 4.2	Hasil Perbandingan Model CNN pada Validasi	65
Tabel 4.3	Hasil Perbandingan Studi Ablasi Model JOC-TextCNN	66

DAFTAR GAMBAR

Gambar 2.1	Diagram Alur NLP	16
Gambar 2.2	Komentar Youtube yang Mengandung Judi Online	17
Gambar 2.3	Transaksi Judi Online Tahun 2017-2023[21]	18
Gambar 2.4	Jaringan Tiruan Deep Learning	21
Gambar 2.5	Diagram Arsitektur Model Kim Yoon[24]	22
Gambar 2.6	Confusion Matrix	28
Gambar 3.1	Alur Penelitian	31
Gambar 3.2	Diagram Alur Pengembangan Model	38
Gambar 3.3	Diagram Alur <i>Preprocessing Text</i>	40
Gambar 3.4	Diagram Arsitektur Model Dasar Text CNN	45
Gambar 3.5	Diagram Arsitektur Model JOC Text CNN	46
Gambar 3.6	Diagram Skema Studi Ablasi	56

DAFTAR RUMUS

Rumus 2.1	<i>Embedding Layer</i>	23
Rumus 2.2	<i>Convolutional Layer</i>	23
Rumus 2.3	<i>Activation Function</i>	24
Rumus 2.4	<i>Max-Pooling Layer</i>	25
Rumus 2.5	<i>Concatenation</i>	25
Rumus 2.6	<i>Fully Connected</i>	25
Rumus 2.7	<i>Output Layer</i>	26
Rumus 2.8	<i>Binary Cross Entropy</i>	26
Rumus 2.9	<i>Accuracy</i>	28
Rumus 2.10	<i>Precision</i>	29
Rumus 2.11	<i>Recall</i>	29
Rumus 2.12	<i>F1-Score</i>	30

DAFTAR KODE

Kode 4.1 Implementasi kode preprocessing data teks	58
Kode 4.2 Implementasi kode augmentasi data teks	59
Kode 4.3 Arsitektur Model Ringan, Sedang dan Berat	61
Kode 4.4 Model JOC-TextCNN	61

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi digital telah membawa dampak besar terhadap berbagai aspek kehidupan masyarakat modern, khususnya dalam hal interaksi sosial dan akses terhadap informasi[1]. Media sosial dan platform berbagi video seperti YouTube memungkinkan distribusi konten dalam jumlah besar secara cepat dan terbuka, mendorong terjadinya diseminasi opini publik secara masif. Di sisi lain, kemajuan ini juga memunculkan tantangan baru dalam bentuk penyalahgunaan ruang digital untuk menyebarkan konten negatif dan ilegal, salah satunya adalah promosi judi online[2]. Fenomena ini menjadi semakin mengkhawatirkan ketika para pelaku menyusupkan konten-konten bermuatan ajakan judi secara terselubung dalam ruang-ruang yang sangat dekat dengan pengguna umum, termasuk anak muda.

YouTube menjadi salah satu platform yang sering dimanfaatkan untuk penyebaran konten perjudian karena tingginya jumlah pengguna aktif serta terbukanya kolom komentar bagi siapa saja[3]. Komentar-komentar yang bersifat promosi judi biasanya dikemas secara menarik, seperti dalam bentuk testimoni atau link tersembunyi, sehingga tampak alami dan tidak langsung terdeteksi sebagai pelanggaran[4]. Keberadaan komentar semacam ini jelas mengancam ekosistem digital, apalagi jika tidak segera ditindak secara sistematis. Menurut Undang-Undang Informasi dan Transaksi Elektronik (UU ITE) Pasal 27 ayat (2), penyebaran konten bermuatan perjudian termasuk tindakan pidana[5]. Namun sayangnya, kompleksitas algoritma platform serta tingginya volume data membuat pengawasan manual menjadi tidak efektif, sehingga memerlukan pendekatan yang lebih adaptif dan cerdas.

Pemerintah Indonesia mencatat bahwa jumlah korban yang terlibat dalam

aktivitas judi online telah mencapai angka 2,37 juta orang. Dari total tersebut, sekitar 80.000 di antaranya masih berusia di bawah 10 tahun, yang menunjukkan bahwa fenomena ini telah menyentuh berbagai lapisan usia, termasuk anak-anak. Menteri Komunikasi dan Informatika, Budi Arie Setiadi, menyatakan bahwa jumlah pemain judi online saat ini telah mencapai 2,7 juta orang, mayoritasnya adalah remaja berusia 17 hingga 20 tahun. Data dari PPATK bahkan lebih mencengangkan, yakni terdapat sekitar 3,5 juta pemain judi online di Indonesia, 80 persen di antaranya berasal dari kalangan ekonomi menengah ke bawah. Pada tahun 2023, nilai transaksi judi online mencapai Rp 327 triliun, naik tajam 213 persen dari tahun sebelumnya. Lonjakan ini sangat signifikan, mengingat pada tahun 2018 transaksi serupa hanya mencapai Rp 3,97 triliun. Selama lima tahun, peningkatannya mencapai lebih dari 8.000 persen[6]. Observasi yang dilakukan di Universitas Negeri Padang juga menemukan adanya mahasiswa yang secara terang-terangan bermain judi online di lingkungan kampus, mempertegas bahwa kelompok ini sangat rentan menjadi sasaran promosi judi digital[7].

Mengingat tingginya penyebaran promosi judi online di media sosial, khususnya melalui kolom komentar YouTube, diperlukan upaya deteksi otomatis untuk mengidentifikasi konten bermuatan ilegal tersebut. Komentar-komentar ini sering bersifat terselubung, menggunakan frasa seperti “slot gacor”, “WD lancar”, atau “situs terpercaya”, yang membuatnya sulit dikenali secara manual [8]. Tantangan lainnya adalah sifat komentar yang tidak terstruktur, informal, dan sangat bervariasi. Oleh karena itu, dibutuhkan pendekatan komputasional berbasis deep learning yang mampu memproses data dalam jumlah besar secara cepat, efisien, dan akurat. Arsitektur Text CNN dinilai sesuai karena kemampuannya dalam mengenali pola-pola lokal pada urutan kata dan mengekstraksi fitur tekstual penting dari data komentar[9]. Untuk mengimplementasikan pendekatan ini secara efektif, diperlukan dataset yang relevan dan representatif terhadap fenomena yang diteliti.

Dataset yang digunakan dalam penelitian ini berasal dari dua sumber

utama. Pertama, dataset yang diunduh dari platform Kaggle sebanyak ± 6.000 komentar YouTube yang dikumpulkan dari siaran langsung “Wayang Jogja Night Carnival #9” pada channel @TribunJogjaOfficial, dengan pelabelan manual oleh pembuat dataset. Kedua, peneliti menambahkan ± 2.000 data komentar tambahan yang dikumpulkan secara mandiri dari video YouTube dengan topik serupa dan diberi label secara manual berdasarkan indikator promosi judi online. Sehingga total data yang digunakan berjumlah sekitar ± 8.000 komentar, terdiri dari dua kelas yaitu: 0 untuk non-promosi dan 1 untuk promosi judi online. Dataset ini bersifat seimbang dan telah melalui proses pembersihan awal (cleaning) untuk memastikan kualitas data yang dianalisis.

Faktor-faktor tekstual seperti pola frasa promosi, istilah slang, dan struktur kalimat informal menjadi indikator penting dalam membedakan komentar bermuatan promosi judi online dari komentar biasa. Arsitektur Convolutional Neural Network (CNN) dinilai tepat untuk tugas ini karena kemampuannya mengenali pola lokal dalam urutan kata melalui operasi konvolusi[10]. CNN mampu mengekstraksi fitur penting dari representasi teks seperti hasil word embedding yang relevan untuk klasifikasi konten terselubung secara otomatis. Penelitian sebelumnya menunjukkan bahwa CNN sering kali menghasilkan akurasi lebih tinggi dibanding metode tradisional seperti Naïve Bayes atau SVM[11]. Oleh karena itu, penelitian ini memilih pendekatan CNN untuk menangani kompleksitas komentar YouTube yang padat, informal, dan sering kali mengandung istilah tidak baku.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk membangun dan menganalisis efektivitas arsitektur TextCNN dalam mendeteksi komentar bermuatan promosi judi online pada platform YouTube. Keunikan penelitian ini terletak pada fokus analisis faktor tekstual khas seperti istilah “slot gacor”, “maxwin”, “WD lancar”, serta ragam bahasa informal yang umum digunakan dalam promosi terselubung. Hingga saat ini, penelitian yang secara spesifik membahas deteksi komentar promosi judi online berbahasa Indonesia

masih sangat terbatas, sementara sebagian besar studi terdahulu berfokus pada spam umum, hate speech, atau analisis sentimen formal. Karakteristik bahasa pada domain ini menimbulkan tantangan teknis tersendiri, karena pelaku sering memodifikasi kata dan frasa untuk menghindari deteksi otomatis.

Sebagai solusi, penelitian ini mengembangkan arsitektur TextCNN klasik (Kim, 2014) menjadi JOC-TextCNN. Pengembangan dilakukan dengan menambahkan cabang konvolusi ganda yang mencakup konvolusi standar, dilated convolution, dan depthwise separable convolution dengan variasi ukuran kernel yang lebih lebar (3–7). Selain itu, disisipkan mekanisme Squeeze-and-Excitation (SE) untuk menimbang kanal secara adaptif sehingga fitur penting dapat lebih ditekankan. Pada tahap representasi masukan, digunakan tokenizer subword IndoBERT dengan embedding yang mendukung padding_idx, dipadukan dengan pipeline preprocessing dan augmentasi berbasis domain untuk menangani variasi bahasa pada komentar YouTube. Dengan rancangan ini, model tidak hanya lebih efisien dari sisi jumlah parameter dan kebutuhan memori, tetapi juga mampu menangkap konteks linguistik yang lebih luas. Proses pelatihan dilakukan dari awal (from scratch) menggunakan BCEWithLogitsLoss dalam skema 5-fold cross-validation, sedangkan performa dievaluasi menggunakan metrik akurasi, precision, recall, F1-score (makro dan mikro), serta confusion matrix untuk menilai kekuatan dan kelemahan prediksi dibandingkan penelitian terdahulu.

1.2 Rumusan Masalah

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

1. Seberapa efektif arsitektur Text CNN dalam mendeteksi dan mengklasifikasikan komentar YouTube yang mengandung promosi judi online?
2. Hyperparameter apa saja yang paling berkontribusi terhadap kinerja model CNN dalam mendeteksi komentar promosi judi online?

3. Modifikasi arsitektur apa saja pada TextCNN yang dapat meningkatkan performa model, khususnya dalam hal F1-Score, presisi, dan recall, dibandingkan dengan arsitektur dasar?

1.3 Tujuan Penelitian

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

1. Menganalisis efektivitas arsitektur Text CNN dalam mengklasifikasikan komentar YouTube yang mengandung promosi judi online.
2. Mengidentifikasi dan mengevaluasi hyperparameter yang memengaruhi kinerja model CNN dalam mendeteksi konten promosi secara terselubung.
3. Mengevaluasi kontribusi berbagai modifikasi arsitektur TextCNN terhadap performa model.

1.4 Batasan Masalah

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

1. Bahasa pemrograman yang digunakan adalah bahasa pemrograman Python.
2. Dalam melakukan penelitian analisis sentimen hanya menggunakan data komentar judi online dari tahun 2024 pada platform YouTube
3. Data yang dianalisis adalah komentar yang diambil dari YouTube berbahasa indonesia.
4. Pengklasifikasian pada analisis sentiment kometar judi online hanya berdasarkan dua kategori yaitu promosi judi dan non-promosi.

1.5 Manfaat Penelitian

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

1. Sebagai kontribusi ilmiah yang dapat menjadi referensi penelitian

selanjutnya dalam bidang deteksi konten berbasis teks, khususnya yang memanfaatkan arsitektur Text CNN untuk mengidentifikasi komentar bermuatan promosi ilegal pada media sosial seperti YouTube.

2. Memberikan informasi dan solusi awal dalam mendeteksi komentar bermuatan promosi judi online, sehingga dapat membantu pihak berwenang, pengelola platform, dan peneliti keamanan digital dalam membangun sistem deteksi dini terhadap penyebaran konten negatif atau ilegal di ruang digital.

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.

Bab I

Bab ini berisikan penjelasan latar belakang dari topik penelitian yang berlangsung, rumusan masalah dari masalah yang dihadapi pada penjelasan di latar belakang, tujuan dari penelitian, batasan dari penelitian, manfaat dari hasil penelitian, dan sistematika penulisan tugas akhir.

Bab II

Bab ini membahas mengenai tinjauan pustaka dari penelitan terdahulu dan dasar teori yang berkaitan dengan penelitian ini.

Bab III

Bab ini berisikan penjelasan alur kerja sistem, alat dan data yang digunakan, metode yang digunakan, dan rancangan pengujian.

Bab IV

Bab ini membahas hasil implementasi dan pengujian dari penelitian yang dilakukan, serta analisis dan evaluasi yang dapat dipetik dari hasil.

Bab V

Bab ini membahas kesimpulan dari hasil penelitian dan juga saran untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian ini menggunakan tinjauan pustaka dari beberapa Pustaka yang telah dilakukan. Tinjauan Pustaka ini dapat dilihat pada Tabel 2.1 Tinjauan Pustaka.

Penelitian yang dilakukan oleh Mathoril Hudha, Endang Supriyati, dan Tri Listyorini pada tahun 2022 dengan judul “Analisis Sentimen Pengguna YouTube terhadap Tayangan #MataNajwaMenantiTerawan dengan Metode Naïve Bayes Classifier”[12]. Permasalahan dalam penelitian ini adalah banyaknya komentar pro dan kontra pada tayangan Mata Najwa yang sulit dianalisis secara manual karena jumlah data besar, tidak terstruktur, serta beragam opini publik. Hasil penelitian menunjukkan bahwa metode Naïve Bayes Classifier mampu mengklasifikasikan komentar dengan akurasi tertinggi sebesar 90,36%, sehingga cukup efektif dalam menentukan kecenderungan sentimen positif maupun negatif pengguna YouTube terhadap tayangan tersebut.

Penelitian yang dilakukan oleh Dany Eka Saputra dan Auliya Rahman Isnain pada tahun 2024 dengan judul “Implementasi Algoritma Convolutional Neural Network untuk Analisis Sentimen Bacapres 2024 pada Kolom Komentar YouTube Mata Najwa”[13]. Permasalahan dalam penelitian ini adalah banyaknya komentar masyarakat terkait calon presiden 2024 pada kanal YouTube Mata Najwa yang jumlahnya masif (45.736 komentar), tidak terstruktur, dan sulit dipetakan secara manual. Hasil penelitian menunjukkan bahwa algoritma Convolutional Neural Network (CNN) berhasil melakukan klasifikasi sentimen dengan akurasi keseluruhan mencapai 91%, membuktikan CNN efektif untuk analisis opini publik di media sosial terkait isu politik.

Penelitian yang dilakukan oleh George Alexander Suwito, Imam

Cholissodin, dan Putra Pandu Adikara pada tahun 2022 dengan judul “Analisis Sentimen Citayam Fashion Week pada Komentar YouTube dengan Metode CNN”[14]. Permasalahan dalam penelitian ini adalah komentar YouTube yang bersifat informal, tidak terstruktur, dan sulit dianalisis secara manual, sehingga diperlukan metode otomatis. Hasil penelitian menunjukkan CNN memberikan hasil yang sangat baik dengan akurasi tertinggi mencapai 97%, nilai precision sebesar 97%, nilai Recall sebesar 98%, dan nilai F-measure sebesar 97%.

Penelitian yang dilakukan oleh Fawwaz Zaini Ahmad, Muhammad Fauzi Satria Arifandy, Muhammad Rasyad Caesarardhi, dan Nur Aini Rakhmawati pada tahun 2021 dengan judul “Bagaimana Masyarakat Menyikapi Pembelajaran Tatap Muka: Analisis Komentar Masyarakat pada Media Sosial YouTube Menggunakan Algoritma Deep Learning Sekuensial dan LDA”[15]. Permasalahan dalam penelitian ini adalah adanya ketidakseimbangan data sentimen (imbalanced dataset) yang diatasi dengan upsampling, serta model LSTM yang meskipun mampu mencapai akurasi hingga 84% masih menunjukkan tingkat false classification yang cukup tinggi. Hasil penelitian menunjukkan model dengan optimisasi ADAM memperoleh akurasi terbaik sebesar 78%, namun kualitas klasifikasi masih dipengaruhi distribusi data yang tidak seimbang.

Penelitian yang dilakukan oleh Ibnu Afdhal, Rahmad Kurniawan, Iwan Iskandar, Roni Salambue, Elvia Budianita, dan Fadhilah Syafria pada tahun 2022 dengan judul “Penerapan Algoritma Random Forest untuk Analisis Sentimen Komentar di YouTube tentang Islamofobia”[16]. Permasalahan dalam penelitian ini adalah sulitnya menganalisis komentar islamofobia di YouTube secara manual karena jumlahnya besar dan tidak terstruktur, serta masih terdapat kesalahan klasifikasi terutama pada komentar negatif. Hasil penelitian menunjukkan algoritma Random Forest mampu memberikan akurasi tertinggi sebesar 79%, dengan distribusi 631 komentar positif dan 369 komentar negatif.

Tabel 2.1 Literasi Penelitian Terdahulu

No.	Judul	Masalah	Metode	Hasil
1.	ANALISIS SENTIMEN PENGGUNA YOUTUBE TERHADAP TAYANGAN #MATA NAJWA MENANTI TERAWAN DENGAN METODE NAÏVE BAYES CLASSIFIER	Penelitian ini terkendala dominasi sentimen netral, menunjukkan keterbatasan Naïve Bayes dan tidak dibandingkan dengan metode klasifikasi lainnya.	NAÏVE BAYES CLASSIFIER	Algoritma NAÏVE BAYES menghasilkan akurasi sebesar 90,36%. Sentimen netral lebih mendominasi dengan jumlah 1.232 data netral, 90 data negatif dan 78 data positif.

No.	Judul	Masalah	Metode	Hasil
2.	IMPLEMENTASI ALGORITMA CONVOLUTIONAL NEURAL NETWORK UNTUK ANALISIS SENTIMEN BACAPRES 2024 PADA KOLOM KOMENTAR YOUTUBE MATA NAJWA	Tantangan dalam menangani fluktuasi cepat opini publik serta diksi bahasa yang beragam dan kompleks. Analisis sentimen menghadapi kesulitan dalam menangkap dinamika emosional yang berubah-ubah secara real-time dan konteks bahasa Indonesia yang penuh makna ganda	CONVOLUTIONAL NEURAL NETWORK	Penelitian ini mencapai akurasi keseluruhan 91% yang mengesankan. Metode CNN menggunakan fase arsitektur dan fine tuning dengan optimizer Adam secara efektif mengkategorikan sentimen ke dalam kelas positif negatif dan netral.

No.	Judul	Masalah	Metode	Hasil
3.	Analisis Sentimen Citayam Fashion Week pada Komentar YouTube dengan Metode Convolutional Neural Network	Komentar yang tidak terstruktur dan informal menyulitkan ekstraksi fitur yang konsisten. Meski CNN akurat, performanya bergantung pada konfigurasi parameter sehingga beresiko overfitting jika diterapkan pada konteks berbeda	Convolutional Neural Network	Sistem ini memiliki nilai evaluasi matrik terbaik yaitu nilai akurasi sebesar 97%, nilai presisi sebesar 97%, nilai Recall sebesar 98%, dan nilai F-measure sebesar 97%.

No.	Judul	Masalah	Metode	Hasil
4.	Bagaimana Masyarakat Menyikapi Pembelajaran Tatap Muka: Analisis Komentar Masyarakat pada Media Sosial Youtube Menggunakan Algoritma Deep Learning Sekuensial dan LDA	Ketidak seimbangan data sentimen diatasi dengan upsampling. Namun, penggunaan LSTM masih menghasilkan klasifikasi yang salah seperti false neutral dan false positive yang mempengaruhi kualitas hasil	Deep Learning Sekuensial-LSTM	Algoritma yang digunakan adalah pemodelan data LSTM dengan optimizer Adam. Setelah pemodelan setiap model dievaluasi dan berhasil memperoleh akurasi tertinggi yaitu sebesar 78%.

No.	Judul	Masalah	Metode	Hasil
5.	Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia	Tantangan muncul dalam mengklasifikasi komentar negatif implisit. Random forest mengalami false classification yang signifikan dan tidak dibandingkan dengan algoritma lain yang mungkin lebih sensitif secara semantik	Random Forest	Algoritma random forest menghasilkan akurasi sebesar 79%. Algoritma random forest dianggap baik dalam melakukan klasifikasi sentimen dengan cepat.

Berdasarkan lima studi terdahulu yang dianalisis, dapat disimpulkan bahwa metode Convolutional Neural Network (CNN) menunjukkan kinerja yang lebih unggul dalam klasifikasi sentimen komentar media sosial, khususnya YouTube, dibandingkan algoritma lain seperti Naïve Bayes, LSTM, dan Random Forest. Beberapa studi membuktikan bahwa CNN mampu mencapai akurasi tinggi, seperti 91% dan bahkan 97%, serta nilai precision dan recall yang stabil. Sementara itu, metode lain cenderung memiliki keterbatasan. Misalnya, Naïve Bayes tidak mampu membedakan nuansa sentimen dalam topik kontroversial, LSTM menunjukkan kesalahan klasifikasi signifikan (false positive/false neutral),

dan Random Forest kurang sensitif terhadap komentar negatif yang bersifat halus. Ini menunjukkan bahwa CNN lebih adaptif terhadap data tidak terstruktur dan informal yang khas pada komentar YouTube.

Dengan mempertimbangkan konteks penelitian ini yang berfokus pada deteksi komentar judi online di YouTube, sebuah topik yang sensitif, informal, dan penuh variasi semantik, maka penggunaan arsitektur text CNN sangat tepat. Arsitektur text CNN memiliki keunggulan dalam mengekstraksi fitur kontekstual dari teks dan menangani data dalam jumlah besar tanpa perlu fitur manual seperti pada pendekatan klasik. Selain itu, arsitektur text CNN telah terbukti efektif di berbagai studi terdahulu yang menggunakan komentar YouTube sebagai objek penelitian. Oleh karena itu, arsitektur text CNN sangat relevan untuk digunakan dalam tugas akhir ini untuk membangun sistem klasifikasi sentimen yang akurat.

2.2 Dasar Teori

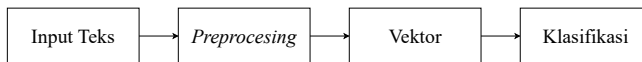
Berikut ini merupakan dasar teori yang digunakan pada penelitian ini:

2.2.1 Deteksi Konten Teks Berbasis Klasifikasi

Deteksi konten berbasis teks merupakan salah satu cabang penting dalam klasifikasi teks pada bidang Natural Language Processing (NLP), yang berfokus pada proses identifikasi keberadaan informasi atau unsur spesifik dalam suatu teks. Dalam penelitian ini, tujuan utamanya adalah mengenali komentar-komentar YouTube yang mengandung unsur promosi judi online[17]. Pendekatan ini berbeda dengan analisis sentimen yang bertujuan mengukur opini atau emosi pengguna, karena deteksi konten lebih menitikberatkan pada pemahaman intensi (intent) dari sebuah pernyataan. Proses deteksi dilakukan menggunakan pendekatan klasifikasi biner berbasis arsitektur deep learning, seperti Convolutional Neural Network (CNN), untuk mengidentifikasi konten secara otomatis dan efisien[18].

2.2.2 *Natural Language Processing (NLP)*

Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan yang berfokus pada bagaimana komputer dapat memahami, memproses, dan menghasilkan bahasa manusia secara efisien. Dalam konteks penelitian ini, NLP menjadi komponen penting dalam tahap awal pengolahan data teks komentar YouTube[19]. Komentar yang bersifat tidak terstruktur dan sering kali mengandung variasi bahasa informal perlu diubah menjadi bentuk yang dapat dimengerti oleh sistem klasifikasi otomatis. Proses dalam NLP ini mencakup beberapa tahap utama, seperti case folding (mengubah semua huruf menjadi huruf kecil), tokenisasi (memisahkan kalimat menjadi kata-kata), stopwords removal (menghapus kata umum yang tidak bermakna sentimen), normalisasi dan stemming.

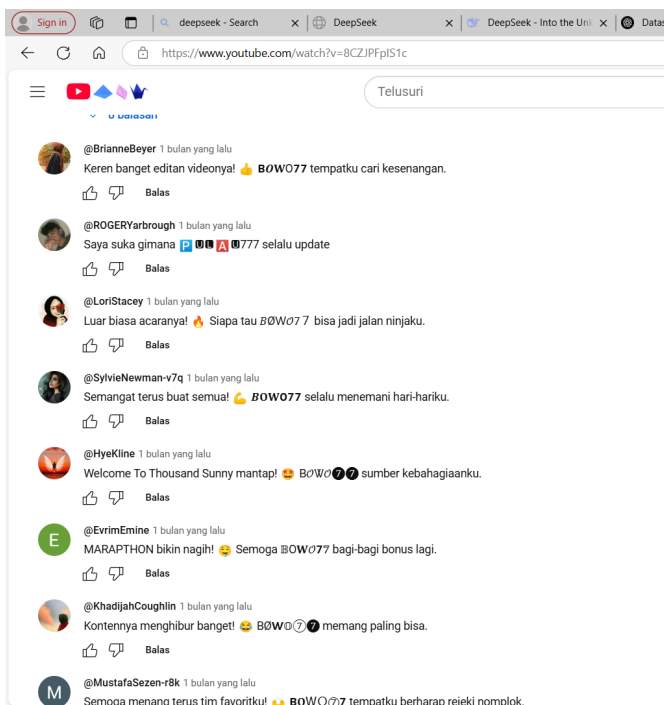


Gambar 2.1 Diagram Alur NLP

Tahapan tersebut bertujuan untuk menyederhanakan struktur teks sebelum dilakukan transformasi numerik melalui word embedding. NLP menjadi fondasi bagi proses deep learning selanjutnya, karena membantu menyaring informasi yang relevan dari komentar-komentar yang mengandung unsur promosi judi online.

2.2.3 *Komentar YouTube dan Judi Online*

Platform YouTube menyediakan ruang komentar sebagai sarana interaksi sosial antarpengguna dan antara pengguna dengan pembuat konten. Namun, ruang ini juga sering disalahgunakan untuk aktivitas ilegal, termasuk penyebaran promosi judi online[20].



Gambar 2.2 Komentar Youtube yang Mengandung Judi Online

Oknum pelaku memanfaatkan celah ini dengan cara menyisipkan pesan terselubung yang dikemas dalam bentuk testimoni, ajakan halus, atau link tersembunyi yang mengarah ke situs judi. Karakteristik komentar-komentar ini sering kali sulit dikenali secara manual karena menggunakan bahasa yang ambigu dan cenderung persuasif. Oleh karena itu, penting dilakukan klasifikasi otomatis untuk mengidentifikasi komentar yang berpotensi mengandung promosi judi. Pendekatan berbasis CNN dan NLP memungkinkan sistem untuk mendeteksi pola-pola kata yang mencurigakan dan menganalisis sentimen dari komentar secara komprehensif. Identifikasi ini sangat penting untuk mencegah penyebaran konten yang bertentangan dengan peraturan perundang-undangan dan menjaga ekosistem digital yang sehat.

2.2.4 Fenomena Judi *Online* di Media Sosial

Tahun	Nilai Transaksi	Jumlah Transaksi	Presentasi Kenaikan
2017	Rp2.009.676.571.607	250.726	-
2018	Rp3.975.512.890.359	666.104	98%
2019	Rp6.183.134.907.079	1.845.832	56%
2020	Rp15.768525.166.418	5.634.499	155%
2021	Rp57.910.725.296.081	43.597.112	267%
2022	Rp104.417.674.955.287	104.791.427	80%
2023	Rp327.000.000.000.000	168.000.000	213%

Gambar 2.3 Transaksi Judi Online Tahun 2017-2023[21]

Fenomena judi online di Indonesia telah berkembang menjadi masalah sosial yang kompleks, terutama karena didorong oleh penetrasi media sosial yang sangat tinggi di kalangan anak muda. Pelaku judi memanfaatkan media sosial sebagai sarana penyebaran promosi karena sifatnya yang terbuka, interaktif, dan menjangkau audiens secara luas tanpa batas geografis. Data dari berbagai lembaga, seperti Kominfo dan PPATK, menunjukkan bahwa mayoritas pemain judi online berasal dari kelompok usia produktif, yakni antara 13 hingga 34 tahun, dan sebagian besar dari mereka termasuk kategori ekonomi menengah ke bawah. Hal ini menandakan adanya kerentanan sosial yang tinggi terhadap praktik judi online. Media sosial seperti YouTube menjadi salah satu saluran utama penyebaran promosi ini karena fitur kolom komentarnya yang longgar dalam moderasi. Oleh karena itu, diperlukan sistem cerdas berbasis teknologi seperti CNN untuk mendeteksi dan menganalisis komentar-komentar bermuatan negatif. Deteksi dini terhadap komentar semacam ini sangat penting untuk mendukung upaya pemerintah dan masyarakat dalam mengurangi dampak negatif judi online terhadap generasi muda.

2.2.5 *Dataset*

Dataset yang digunakan dalam penelitian ini terdiri dari dua sumber utama:

Tabel 2.2 Sumber Dataset

Sumber Dataset	Deskripsi
Kaggle (Wayang Jogja)	Komentar dari video live streaming @TribunJogja Official
Koleksi Pribadi	Komentar dikumpulkan langsung dari YouTube dan dilabeli sesuai kriteria

Dalam konteks penelitian ini, data dikumpulkan dari kolom komentar YouTube yang sering menjadi media penyebaran promosi judi online secara terselubung. Komentar-komentar tersebut biasanya menggunakan frasa tidak formal dan kode seperti “slot gacor”, “WD lancar”, atau “situs terpercaya”. Oleh karena itu, data komentar YouTube menjadi sumber yang tepat untuk eksperimen deteksi otomatis konten bermuatan judi. Komentar dalam penelitian ini diperlakukan sebagai satuan teks pendek (short text) yang bersifat informal, tidak terstruktur, dan mengandung banyak elemen khas pengguna media sosial. Dataset semacam ini sangat relevan digunakan dalam studi pemrosesan bahasa alami berbasis deep learning.

2.2.6 Text Processing

Text processing adalah proses awal dalam pengolahan data teks yang bertujuan menyiapkan komentar mentah agar dapat dianalisis secara efektif oleh model klasifikasi[22]. Proses ini disesuaikan dengan karakteristik komentar YouTube yang cenderung informal, tidak terstruktur, dan sering mengandung spam. Tahapan yang diterapkan meliputi pembersihan teks, normalisasi slang, penghapusan stopword, dan stemming untuk menyederhanakan input sebelum diolah lebih lanjut oleh model. Berikut tahapan preprocessing yang diterapkan:

1. Case Folding

Case folding merupakan tahap mengubah seluruh huruf dalam teks menjadi huruf kecil. Proses ini dilakukan untuk menyeragamkan kata, sehingga model tidak membedakan antara huruf kapital dan non-kapital.

2. URL Removal

URL removal adalah proses menghapus tautan dari teks. URL dianggap tidak memiliki nilai semantik dalam analisis sentimen dan lebih sering menjadi noise.

3. Punctuation Cleaning

Punctuation cleaning bertujuan menghapus tanda baca yang tidak relevan dengan analisis. Dengan mengurangi tanda baca, teks menjadi lebih bersih dan mudah diproses.

4. Repeated Character Reduction

Repeated character reduction adalah normalisasi kata yang memiliki pengulangan huruf berlebihan. Hal ini penting agar variasi tidak menimbulkan kata baru yang tidak bermakna.

5. Slang Word Normalization

Slang word normalization dilakukan dengan mengganti kata tidak baku atau singkatan menjadi bentuk standar agar maknanya jelas.

6. Tokenization

Tokenization adalah proses memecah kalimat menjadi unit-unit kecil berupa kata atau frasa, sehingga lebih mudah dianalisis oleh model.

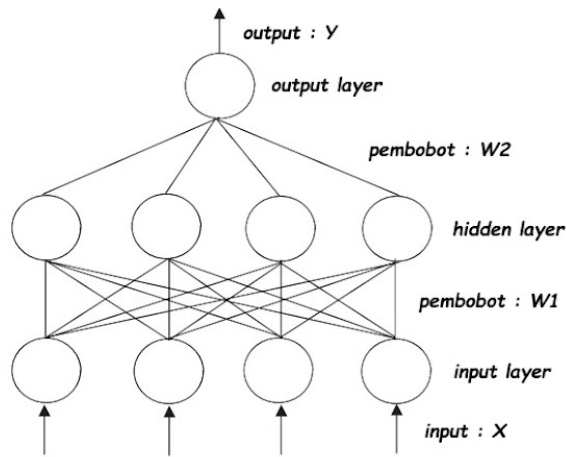
7. Stopword Removal

Stopword removal menghapus kata-kata umum yang sering muncul tetapi tidak memberi kontribusi besar dalam analisis, seperti kata sambung atau kata depan.

8. Stemming

Stemming adalah tahap mengubah kata berimbuhan ke bentuk dasarnya untuk menyatukan variasi kata.

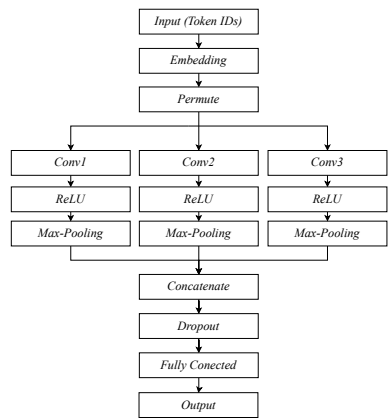
2.2.7 Deep Learning



Gambar 2.4 Jaringan Tiruan Deep Learning

Deep learning merupakan salah satu pendekatan canggih dalam bidang machine learning yang memanfaatkan jaringan saraf tiruan dengan banyak lapisan (deep neural networks) untuk mengekstraksi dan mempelajari representasi data secara hierarkis[23]. Teknologi ini memungkinkan komputer untuk mengenali pola kompleks dari data mentah tanpa memerlukan fitur buatan secara eksplisit. Dalam konteks klasifikasi teks, seperti pada komentar YouTube yang mengandung unsur judi online, deep learning dapat mengidentifikasi struktur sintaksis dan semantik dalam kalimat secara otomatis. Keunggulan utamanya adalah kemampuannya untuk belajar dari data besar secara efisien, serta menyesuaikan bobot-bobot internal berdasarkan masukan. Hal ini menjadikan deep learning sangat cocok untuk mengolah data teks informal yang penuh variasi bahasa.

2.2.8 *Arsitektur Text Convolutional Neural Network*



Gambar 2.5 Diagram Arsitektur Model Kim Yoon[24]

Convolutional Network (CNN) merupakan salah satu arsitektur deep learning yang awalnya banyak digunakan pada bidang pengolahan citra karena kemampuannya dalam mengenali pola spasial. Seiring perkembangan penelitian, CNN tidak hanya terbatas pada pengolahan gambar, melainkan juga mulai diadaptasi pada pemrosesan bahasa alami (Natural Language Processing/NLP). Hal ini didasari oleh kesamaan karakteristik antara data citra dan teks, yaitu sama-sama memiliki struktur sekuensial yang dapat diuraikan menjadi pola-pola lokal. Keunggulan utama CNN terletak pada kemampuannya dalam mengekstraksi fitur penting secara otomatis dari data masif tanpa harus mendefinisikan aturan manual terlebih dahulu.

Dalam konteks teks, CNN mampu menangkap representasi n-gram atau urutan kata-kata tertentu yang sering muncul bersamaan, yang kemudian dijadikan sebagai ciri penting dalam proses klasifikasi komentar. Kemampuan ini membuat CNN relevan digunakan dalam penelitian yang berfokus pada deteksi konten tertentu, termasuk komentar bermuatan promosi judi online di

platform YouTube. Untuk mendukung proses tersebut, CNN biasanya terdiri atas sejumlah komponen inti yang bekerja secara berurutan, mulai dari representasi kata hingga tahap klasifikasi akhir. Secara umum, arsitektur CNN untuk teks dapat diuraikan melalui beberapa komponen utama berikut.

2.2.8.1 Input Layer

Komentar YouTube yang telah melewati tahap preprocessing seperti case folding, tokenisasi, stopword removal dan stemming dimasukkan sebagai input dalam bentuk urutan kata.

2.2.8.2 Word Embedding Layer

Sebelum teks dapat diproses CNN, setiap kata terlebih dahulu dipetakan ke dalam vektor numerik dengan dimensi tetap. Representasi ini disebut *word embedding*. Word embedding memungkinkan model memahami kemiripan semantik antar kata. Misalnya, kata “taruhan” dan “judi” akan berada lebih dekat dibanding “judi” dan “sekolah” pada ruang vektor.

Secara matematis, kalimat dengan panjang n kata dapat direpresentasikan sebagai:

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}, \quad \mathbf{x}_i \in \mathbb{R}^d \quad (\text{Rumus 2.1})$$

dengan d adalah dimensi embedding. Matriks embedding ini akan menjadi input bagi lapisan konvolusi.

2.2.8.3 Convolutional Layer

Lapisan konvolusi berfungsi sebagai ekstraktor fitur utama. CNN menggunakan beberapa filter (kernel) dengan panjang berbeda untuk menangkap pola n-gram. Misalnya, kernel berukuran 3 menangkap pola trigram, kernel 4 menangkap pola 4 kata, dan seterusnya.

Operasi konvolusi pada teks dapat diformulasikan sebagai:

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

(Rumus 2.2)

dengan:

- \mathbf{w} = bobot filter berukuran $k \times d$,
- $\mathbf{x}_{i:i+k-1}$ = embedding kata ke- i sampai ke- $(i + k - 1)$,
- b = bias,
- f = fungsi aktivasi (ReLU).

Hasil konvolusi menghasilkan *feature map* yang menggambarkan pola penting dari urutan kata.

2.2.8.4 Activation Function (ReLU)

Fungsi aktivasi merupakan komponen penting dalam jaringan saraf karena berfungsi untuk memperkenalkan non-linearitas ke dalam model. Tanpa fungsi aktivasi, jaringan hanya akan melakukan transformasi linear yang membatasi kemampuannya dalam mempelajari representasi kompleks dari data. Salah satu fungsi aktivasi yang paling populer digunakan adalah Rectified Linear Unit (ReLU).

ReLU didefinisikan secara matematis sebagai:

$$f(x) = \max(0, x)$$

(Rumus 2.3)

Fungsi ini akan mengembalikan nilai input apabila bernilai positif, dan nol apabila input bernilai negatif. ReLU dipilih karena sederhana secara komputasi, mampu mengurangi permasalahan vanishing gradient, serta mempercepat proses pelatihan jaringan. Dalam konteks pemrosesan teks, ReLU membantu CNN untuk menyoroti fitur-fitur penting yang dihasilkan oleh lapisan konvolusi sehingga dapat diproses lebih efektif pada tahap selanjutnya.

2.2.8.5 Max-Pooling Layer

Untuk mereduksi dimensi *feature map* sekaligus mempertahankan informasi paling dominan, digunakan teknik *max-pooling*. Max-pooling memilih

nilai maksimum dari setiap feature map:

$$\hat{c} = \max\{c_1, c_2, \dots, c_{n-k+1}\} \quad (\text{Rumus 2.4})$$

Dengan cara ini, CNN hanya mempertahankan fitur paling kuat, sehingga tidak terjebak pada *noise* atau variasi yang kurang relevan.

2.2.8.6 Concatenation Layer

Setelah proses ekstraksi fitur melalui konvolusi, aktivasi, dan pooling, setiap filter menghasilkan representasi vektor yang berbeda sesuai dengan pola n-gram yang ditangkap. Agar informasi dari berbagai filter tersebut dapat digabungkan secara menyeluruh, digunakan concatenation layer. Lapisan ini bertugas menggabungkan seluruh hasil max-pooling dari setiap filter menjadi satu vektor fitur besar yang merepresentasikan komentar secara komprehensif.

Secara matematis, jika terdapat m buah filter dengan hasil pooling $\hat{c}_1, \hat{c}_2, \dots, \hat{c}_m$, maka operasi concatenation dapat dituliskan sebagai:

$$\mathbf{z} = [\hat{c}_1; \hat{c}_2; \dots; \hat{c}_m] \quad (\text{Rumus 2.5})$$

Hasil vektor \mathbf{z} ini mengandung fitur dominan dari seluruh filter yang digunakan, sehingga mampu menangkap berbagai variasi pola teks pada komentar YouTube. Vektor inilah yang kemudian menjadi masukan bagi lapisan fully connected untuk menghasilkan representasi lebih abstrak sebelum tahap klasifikasi akhir.

2.2.8.7 Fully Connected Layer

Hasil pooling dari berbagai filter kemudian digabungkan menjadi satu vektor:

$$\mathbf{h} = f(\mathbf{W}\mathbf{z} + \mathbf{b}) \quad (\text{Rumus 2.6})$$

dengan m adalah jumlah filter. Vektor \mathbf{z} mengandung informasi penting dari seluruh komentar. Lapisan *fully connected* kemudian memproses \mathbf{z} untuk

menghasilkan representasi yang lebih abstrak dan siap digunakan pada tahap klasifikasi.

2.2.8.8 Output Layer (Sigmoid)

Lapisan output bertugas mengklasifikasikan komentar ke dalam dua kelas. Pada penelitian ini digunakan fungsi aktivasi *sigmoid* karena masalah yang dihadapi adalah klasifikasi biner:

$$\hat{y} = \sigma(\mathbf{W}\mathbf{z} + b) = \frac{1}{1 + e^{-(\mathbf{W}\mathbf{z} + b)}} \quad (\text{Rumus 2.7})$$

dengan \hat{y} bernilai antara 0 hingga 1. Nilai mendekati 0 diinterpretasikan sebagai komentar non-promosi, sedangkan nilai mendekati 1 diinterpretasikan sebagai komentar promosi judi online.

2.2.9 Loss Function

Penelitian ini menggunakan Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss), yang sesuai untuk klasifikasi biner (judol vs non-judol). Fungsi ini menggabungkan sigmoid dan binary cross entropy dalam satu formulasi yang lebih stabil secara numerik dibandingkan digunakan terpisah. Secara matematis, model menghasilkan logit z yang diubah menjadi probabilitas kelas positif p melalui fungsi sigmoid:

$$\mathcal{L}(y, p) = -[y \cdot \log(p) + (1 - y) \cdot \log(1 - p)] \quad (\text{Rumus 2.8})$$

dengan $p = \sigma(z) = \frac{1}{1 + e^{-z}}$ sebagai probabilitas kelas positif hasil fungsi sigmoid, dan $y \in \{0, 1\}$ sebagai label sebenarnya. Nilai loss dirata-ratakan pada tiap batch untuk memperoleh nilai total kerugian.

2.2.10 Hyparameter

Dalam pelatihan model deep learning, pemilihan hyperparameter dan fungsi pelatihan yang tepat sangat memengaruhi performa akhir model[25]. Hyperparameter adalah parameter yang ditentukan sebelum proses pelatihan

dimulai dan tidak diubah selama training. Sementara itu, fungsi-fungsi pelatihan seperti loss function dan activation function merupakan bagian penting dari arsitektur dan proses pembelajaran model

2.2.11 *Feature Extraction*

Feature extraction atau ekstraksi fitur merupakan tahap penting dalam pemrosesan data teks sebelum dilakukan klasifikasi. Dalam konteks penelitian ini, fitur yang dimaksud adalah representasi numerik dari kata-kata dalam komentar YouTube yang mengandung unsur judi online[26]. Proses ini dilakukan menggunakan teknik word embedding pada embedding layer, di mana setiap kata dalam komentar dikonversi menjadi vektor berdimensi tetap yang menggambarkan makna semantik kata tersebut. Vektor-vektor ini selanjutnya menjadi input utama bagi arsitektur CNN dalam mengenali pola-pola yang relevan. Dengan fitur yang telah diekstraksi secara tepat, model dapat membedakan komentar bermuatan negatif seperti promosi judi, dari komentar lain yang bersifat positif.

2.2.12 *Confusion Matrix*

Confusion matrix merupakan metode evaluasi yang banyak digunakan untuk mengukur performa model klasifikasi, termasuk dalam analisis sentimen terhadap komentar judi online di YouTube[27]. Matriks ini menyajikan hasil prediksi model dalam bentuk tabel dua dimensi yang mencakup empat komponen utama:

- True Positive (TP) — komentar positif yang berhasil diklasifikasikan sebagai positif.
- True Negative (TN) — komentar negatif yang diklasifikasikan negatif.
- False Positive (FP) — komentar negatif yang salah diprediksi sebagai positif.
- False Negative (FN) — komentar positif yang salah diklasifikasikan

negatif.

Dari keempat komponen ini dapat dihitung metrik evaluasi seperti akurasi, presisi, recall, dan F1-score, yang berguna dalam menilai keandalan model dalam mengenali komentar terkait promosi judi secara tepat.

		True Values	
		Positive	Negative
Predicted Values	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Gambar 2.6 Confusion Matrix

2.2.12.1 Accuracy

Accuracy merupakan metrik evaluasi yang paling umum digunakan dalam klasifikasi. Metrik ini mengukur proporsi prediksi yang benar terhadap seluruh data uji yang digunakan[28]. Dengan kata lain, accuracy menunjukkan seberapa banyak model mampu mengklasifikasikan data dengan tepat dibandingkan dengan keseluruhan jumlah data. Semakin tinggi nilai accuracy, semakin baik performa model secara umum. Namun, pada data yang tidak seimbang, accuracy seringkali kurang menggambarkan kinerja model yang sebenarnya karena kelas mayoritas cenderung lebih dominan. Perhitungan *Accuracy* dapat dilihat pada Rumus 2.9 sebagai berikut.

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

(Rumus 2.9)

2.2.12.2 *Precision*

Precision mengukur sejauh mana prediksi positif yang dihasilkan model benar-benar sesuai dengan kelas positif. Metrik ini fokus pada kualitas prediksi positif, yaitu dari semua data yang diprediksi sebagai kelas positif, berapa banyak yang benar-benar positif[29]. Precision sangat penting ketika kesalahan prediksi positif (false positive) memiliki konsekuensi besar. Misalnya, dalam deteksi komentar promosi judi online, precision yang tinggi berarti sebagian besar komentar yang ditandai sebagai promosi benar-benar komentar promosi, sehingga mengurangi potensi kesalahan dalam penandaan. Perhitungan *Precision* dapat dilihat pada Rumus 2.10 sebagai berikut.

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

(Rumus 2.10)

2.2.12.3 *Recall*

Recall atau sensitivitas mengukur kemampuan model dalam menemukan seluruh data positif yang ada. Metrik ini melihat dari sisi kelas positif, yaitu dari semua data yang benar-benar positif, berapa banyak yang berhasil ditemukan oleh model[30]. Recall menjadi penting ketika tujuan utama adalah meminimalkan kesalahan pengabaian data positif (false negative). Dalam penelitian ini, recall yang tinggi berarti model mampu mendeteksi sebanyak mungkin komentar yang bermuatan promosi judi online. Perhitungan *Recall* dapat dilihat pada Rumus 2.11 sebagai berikut.

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

(Rumus 2.11)

2.2.12.4 *F1-Score*

F1-Score merupakan ukuran harmonis yang menggabungkan precision dan recall ke dalam satu nilai tunggal. Metrik ini dihitung sebagai rata-rata harmonis antara precision dan recall, sehingga memberikan gambaran yang lebih seimbang ketika terdapat trade-off antara keduanya[31]. F1-Score sangat berguna pada kasus data yang tidak seimbang, karena mampu menilai kinerja model secara adil dengan mempertimbangkan baik ketepatan dalam memprediksi kelas positif maupun kelengkapan dalam menemukan seluruh data positif. Nilai F1-Score yang tinggi menunjukkan bahwa model memiliki keseimbangan yang baik antara precision dan recall. Perhitungan *F1-Score* dapat dilihat pada Rumus 2.12 sebagai berikut.

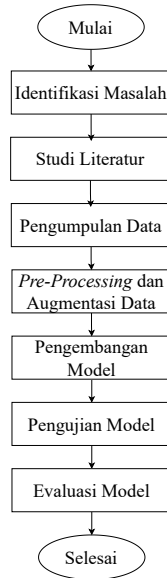
$$F1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (\text{Rumus 2.12})$$

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Pada penelitian ini, alur dirancang untuk memastikan setiap tahapan pemrosesan dilakukan secara sistematis dan efisien. Alur penelitian ini mencerminkan langkah-langkah utama yang dilakukan pada 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

Penelitian ini berangkat dari fenomena meningkatnya komentar bermuatan promosi judi online pada platform YouTube. Komentar-komentar tersebut tidak hanya menurunkan kualitas ruang digital, tetapi juga berpotensi mendorong penyebaran perilaku negatif di masyarakat. Oleh karena itu, dibutuhkan sebuah metode yang mampu mendeteksi secara otomatis komentar yang mengandung unsur promosi judi online.

3.2.2 Studi Literatur

Untuk memperkuat landasan penelitian, dilakukan studi literatur terhadap penelitian-penelitian terdahulu terkait analisis sentimen, pemrosesan bahasa alami (Natural Language Processing), serta penerapan metode deep learning dalam klasifikasi teks. Salah satu arsitektur yang banyak digunakan adalah Convolutional Neural Network (CNN), yang terbukti mampu mengekstraksi pola penting dalam data sekuensial, termasuk teks. Studi ini menjadi pijakan dalam menentukan metode dan strategi pengembangan model.

3.2.3 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini terdiri dari dua sumber utama, yaitu:

Tabel 3.1 Pengumpulan Dataset

Sumber	Jumlah Data	Labeling	Deskripsi
--------	-------------	----------	-----------

Sumber	Jumlah Data	Labeling	Deskripsi
Kaggle (Wayang Jogja)	±6.000	Manual oleh uploader	Komentar dari video live streaming @TribunJogja Official
Koleksi Pribadi	±2.000	Manual oleh peneliti	Komentar dikumpulkan langsung dari YouTube dan dilabeli sesuai kriteria

Total dataset berjumlah sekitar 8.000 komentar, terdiri dari dua kelas:

- Label 0: Komentar non-promosi judi online
- Label 1: Komentar promosi judi online

Dataset ini telah melalui proses pembersihan awal (cleaning) yang mencakup penghapusan karakter khusus, simbol, emoji, dan tautan yang tidak relevan. Dataset bersifat seimbang, artinya distribusi antara kedua label relatif setara, sehingga sesuai untuk digunakan dalam pelatihan model klasifikasi.

Tabel 3.2 Label Dataset

Komentar	Label
BAGI BAGI MAXWIN TOGEL 62!!	1
keren jogjakuuu	0
Makswin lagi hari ini, mntp sih situsnya	1
happy aniversary Yogya	0
Baru daftar, maen sebentar, lgsng jepe	1

Komentar	Label
bangga jadi warga Yogya yang kaya budaya...	0

Akuisisi data dalam penelitian ini dilakukan melalui dua sumber. Pertama, dataset publik diperoleh dari Kaggle dengan judul Deteksi Judi Online, yang sudah dilengkapi dengan label dan struktur tabel terstandarisasi. Kedua, peneliti mengumpulkan komentar secara manual dari YouTube menggunakan Comment Scraper pada periode Januari–Maret 2025 dari kanal Reza Arap. Komentar hasil scraping kemudian dilabeli secara manual oleh dua annotator native bahasa Indonesia, dengan mengacu pada indikator seperti kemunculan istilah khas promosi judi online (misalnya “togel”, “maxwin”, “jepe”), penyebutan kata-kata ajakan seperti “situs” atau “daftar”, serta pola kalimat persuasif yang bersifat promosi.

Kombinasi antara dataset publik dan hasil pelabelan manual ini menghasilkan total sekitar 8.000 komentar dengan variasi bahasa dan gaya promosi yang beragam, sehingga menjadi sumber data yang representatif untuk melatih dan mengevaluasi model deteksi komentar judi online.

3.2.4 Pre-Processing dan Augmentasi Data

Tahapan text preprocessing dilakukan untuk memastikan data komentar memiliki format yang bersih, seragam, dan siap diproses oleh model. Mengingat karakteristik komentar YouTube yang cenderung noisy, banyak mengandung slang, singkatan, serta karakter non-standar, proses ini menjadi tahap penting dalam meningkatkan kualitas representasi teks. Langkah-langkah yang diterapkan meliputi case folding, punctuation cleaning, tokenization, stopword removal, dan stemming, yang diadaptasi dari praktik umum dalam penelitian klasifikasi teks media sosial.

Selain tahapan dasar tersebut, penelitian ini juga menambahkan dua proses normalisasi lanjutan yang dirancang khusus untuk menangani karakteristik unik

komentar promosi judi online. Pertama, dilakukan repeated character reduction untuk menormalkan kata dengan huruf berulang secara berlebihan, yang sering muncul dalam konteks ekspresif seperti “gaacoorrr” menjadi “gacor”. Kedua, diterapkan slang word normalization menggunakan kamus khusus yang berisi padanan kata baku dari istilah informal seperti “wd”, “slot”, atau “maxwin”. Kedua langkah tambahan ini terbukti efektif dalam meningkatkan performa model pada teks tidak terstruktur dan informal, sebagaimana ditunjukkan oleh berbagai studi terdahulu. Setelah data bersih diperoleh, dilakukan proses augmentasi data dengan teknik substitusi sinonim dan random insertion untuk menyeimbangkan distribusi kelas, sehingga model dapat belajar lebih baik dan mengurangi bias akibat ketidakseimbangan data.

3.2.5 Pengembangan Model

Pengembangan model dalam penelitian ini dilakukan dengan mengadaptasi arsitektur dasar TextCNN (Kim, 2014) menjadi varian yang lebih kompleks dan kontekstual, yaitu JOC-TextCNN. Pengembangan dilakukan melalui penambahan multi-branch convolution yang terdiri atas standard convolution, dilated convolution, dan depthwise separable convolution untuk menangkap fitur teks pada berbagai skala konteks linguistik. Selain itu, disisipkan Squeeze-and-Excitation (SE) Block guna memberikan penimbangan kanal secara adaptif, sehingga fitur penting memperoleh bobot yang lebih tinggi selama proses pembelajaran.

Pada tahap representasi masukan, digunakan tokenizer IndoBERT yang diintegrasikan dengan embedding layer khusus agar model dapat memanfaatkan tokenisasi subword yang sesuai dengan karakteristik bahasa Indonesia. Proses ini dipadukan dengan pipeline preprocessing dan augmentasi berbasis domain untuk meningkatkan robustnes model terhadap variasi bahasa informal dan ejaan tidak baku yang sering muncul pada komentar YouTube.

Selanjutnya, dilakukan penyesuaian hyperparameter utama, meliputi

ukuran dan jumlah filter konvolusi, tingkat dropout, serta jenis optimizer, guna memperoleh konfigurasi pelatihan yang optimal. Kombinasi modifikasi tersebut membuat JOC-TextCNN mampu menyeimbangkan kompleksitas arsitektur dengan efisiensi komputasi, sekaligus meningkatkan kemampuan model dalam mengenali pola-pola linguistik khas pada komentar promosi judi online.

3.2.6 Pengujian

Tahap pengujian dilakukan untuk menilai kemampuan model dalam mendeteksi komentar bermuatan promosi judi online. Metode k-fold cross validation digunakan agar hasil evaluasi lebih akurat dan tidak bergantung pada satu set data tertentu. Selain itu, dilakukan studi ablation untuk menguji pengaruh masing-masing komponen arsitektur terhadap performa model.

3.2.7 Evaluasi

Penelitian ini dievaluasi menggunakan sejumlah metrik, yaitu accuracy, precision, recall, dan F1-score, serta didukung oleh confusion matrix untuk melihat distribusi hasil klasifikasi tiap kelas. Hasil evaluasi tersebut dianalisis guna menilai performa model dalam mendeteksi komentar bermuatan promosi judi online pada YouTube, sekaligus dibandingkan dengan capaian penelitian terdahulu sebagai bahan pembahasan.

3.3 Alat dan Bahan Tugas Akhir

Dalam menjalani penelitian, beberapa alat dan bahan digunakan untuk memastikan penelitian berjalan dengan baik.

3.3.1 Alat

Dalam membuat pengukuran frekuensi denyut nadi non-kontak dalam penelitian, berikut adalah alat-alat yang digunakan:

1. Laptop dengan spesifikasi, Processor 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz 2.69 GHz, RAM 8,00 GB.

2. *Visual Studio Code* sebagai *tools* untuk *text editor*.
3. `pyth`
4. `charset-normalizer` versi 3.4.1
5. `matplotlib` versi 3.10.1
6. `nltk` versi 3.9.1
7. `numpy` versi 2.2.4
8. `openpyxl` versi 3.1.5
9. `pandas` versi 2.2.3
10. `tokenizers` versi 0.21.1
11. `torch` versi 2.6.0

3.3.2 Bahan

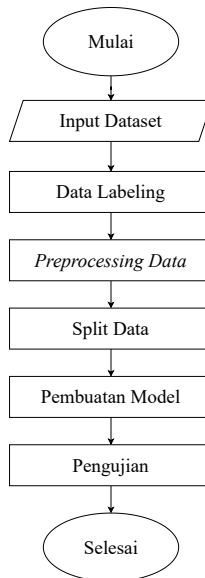
Dataset diperoleh dari platform Kaggle dengan judul “youtubechatjogjacleam” (tautan: <https://www.kaggle.com/datasets/yaemico/deteksi-judi-online>) dan koleksi pribadi yang diambil dari livestreaming youtube. Total komentar dalam dataset mencapai 8.350 data, dengan distribusi kelas yang seimbang, yaitu sekitar 4.163 komentar non-promosi (label 0) dan 4.187 komentar promosi judi online (label 1).

3.4 Metode Pengembangan Model

Metode pengembangan model pada penelitian ini mengikuti tahapan yang sistematis agar proses klasifikasi komentar dapat berjalan dengan baik. Setiap tahap dijabarkan mulai dari penginputan dataset, data labeling, *preprocessing data*, split data, pembuatan model, hingga tahap pengujian. Alur tersebut divisualisasikan dalam diagram pengembangan model, sehingga memudahkan pembaca untuk memahami keterkaitan antar langkah secara keseluruhan.

Diagram alur pengembangan model ini berfungsi sebagai gambaran umum proses yang dilakukan, sekaligus menjadi panduan dalam implementasi teknis

penelitian. Dengan adanya diagram, alur penelitian dapat dipahami secara lebih terstruktur, mulai dari data mentah hingga menghasilkan model yang siap digunakan untuk mendeteksi komentar promosi judi online di YouTube. Daiagram alur pengembangan model yang akan digunakan pada penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Alur Pengembangan Model

3.4.1 *Input Dataset*

Tahap pertama dimulai dengan memanfaatkan dataset yang diperoleh dari platform Kaggle dengan judul “youtubechatjogjacleam”. Dataset ini terdiri atas 8.350 komentar pengguna YouTube yang terkumpul dari berbagai unggahan. Data tersebut mencakup komentar dengan beragam variasi bahasa, gaya penulisan, serta penggunaan simbol. Seluruh komentar ini menjadi sumber

utama yang akan diolah dan dianalisis dalam penelitian untuk mendeteksi keberadaan promosi judi online.

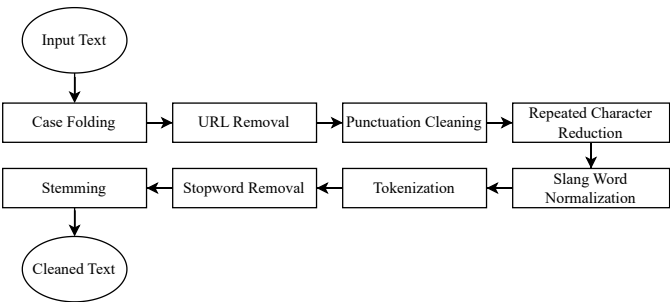
3.4.2 Data Labeling

Agar model dapat melakukan klasifikasi, setiap komentar diberi label sesuai kategorinya. Pada dataset ini terdapat dua kelas utama, yaitu komentar non-promosi (label 0) dan komentar promosi judi online (label 1). Proses pelabelan sangat penting karena menentukan target output dari model. Dengan adanya label yang jelas, model dapat belajar pola bahasa yang membedakan kedua jenis komentar tersebut, sehingga menghasilkan sistem deteksi yang lebih akurat. Proses data labeling yang akan digunakan pada penelitian ini dapat dilihat pada tabel 3.1.

Tabel 3.3 Data Labeling

Komentar	Label	Kategori
BAGI BAGI MAXWIN TOGEL 62!!	1	promosi judol
HUT kota Yogyakarta tercinta semoga selalu sukses	0	non-promosi judol
BWTOGEL lagi gacorr	1	promosi judol

3.4.3 Preprocessing Data



Gambar 3.3 Diagram Alur *Preprocessing Text*

3.4.3.1 Case Folding

Case folding adalah proses mengubah seluruh huruf dalam teks menjadi huruf kecil. Tujuannya untuk menyeragamkan kata sehingga ”Judi” dan ”judi” dianggap sama, serta mengurangi keragaman kata yang tidak perlu.

Tabel 3.4 Proses *Case Folding*

Sebelum	Sesudah
BAGI BAGI MAXWIN TOGEL 62!!	bagi bagi maxwin togel 62!!
HUT kota Yogyakarta tercinta semoga selalu sukses	hut kota yogyakarta tercinta semoga selalu sukses
BWTOGEL lagi gacorr	bwtogel lagi gacorr

3.4.3.2 URL Removal

URL removal adalah penghapusan tautan dari teks. Hal ini dilakukan karena URL tidak memiliki makna penting dalam analisis sentimen dan hanya menambah noise pada data.

Tabel 3.5 Proses *URL Removal*

Sebelum	Sesudah
bagi bagi maxwin togel 62!!	bagi bagi maxwin togel 62!!
hut kota yogyakarta tercinta semoga selalu sukses	hut kota yogyakarta tercinta semoga selalu sukses
bwtogel lagi gacorr	bwtogel lagi gacorr

3.4.3.3 *Punctuation Cleaning*

Punctuation cleaning adalah penghapusan tanda baca dari teks. Tanda baca umumnya tidak berkontribusi pada klasifikasi sentimen, sehingga pembersihan ini membantu mengurangi noise dan mempermudah tahap pemrosesan selanjutnya.

Tabel 3.6 Proses *Punctuation Cleaning*

Sebelum	Sesudah
bagi bagi maxwin togel 62!!	bagi bagi maxwin togel 62
hut kota yogyakarta tercinta semoga selalu sukses	hut kota yogyakarta tercinta semoga selalu sukses
bwtogel lagi gacorr	bwtogel lagi gacorr

3.4.3.4 *Repeated Character Reduction*

Repeated character reduction adalah normalisasi kata dengan pengulangan huruf, seperti "judiiii" menjadi "judi". Proses ini menjaga konsistensi kata sehingga lebih mudah dipahami oleh model.

Tabel 3.7 Proses *Repeated Character Reduction*

Sebelum	Sesudah
bagi bagi maxwin togel 62	bagi bagi maxwin togel 62
hut kota yogyakarta tercinta semoga selalu sukses	hut kota yogyakarta tercinta semoga selalu sukses
bwtogel lagi gacorr	bwtogel lagi gacorr

3.4.3.5 *Slang Word Normalization*

Slang word normalization adalah proses mengganti kata tidak baku atau bahasa gaul menjadi bentuk standar, misalnya “gk” menjadi “tidak”. Normalisasi ini membuat data lebih konsisten dan mudah dipahami oleh model.

Tabel 3.8 Proses *Slang Word Normalization*

Sebelum	Sesudah
bagi bagi maxwin togel 62	bagi bagi maxwin togel 62
hut kota yogyakarta tercinta semoga selalu sukses	hut kota yogyakarta tercinta semoga selalu sukses
bwtogel lagi gacorr	bwtogel lagi gacorr

3.4.3.6 *Tokenization*

Tokenization adalah pemisahan teks menjadi unit-unit kecil yang disebut token, biasanya berupa kata atau frasa. Tahap ini penting untuk memudahkan analisis dan pemrosesan lebih lanjut.

Tabel 3.9 Proses *Tokenization*

Sebelum	Sesudah
bagi bagi maxwin togel 62	["bagi", "bagi", "maxwin", "togel", "62"]
hut kota yogyakarta tercinta semoga selalu sukses	["hut", "kota", "yogyakarta", "tercinta", "semoga", "selalu", "sukses"]
bwtogel lagi gacor	["bwtogel", "lagi", "gacor"]

3.4.3.7 *Stopword Removal*

Stopword removal adalah proses menghapus kata-kata umum seperti “dan”, “di”, atau “yang” yang tidak memiliki makna signifikan dalam analisis. Hal ini membantu model fokus pada kata yang lebih informatif.

Tabel 3.10 Proses *Stopword Removal*

Sebelum	Sesudah
["bagi", "bagi", "maxwin", "togel", "62"]	["bagi", "bagi", "maxwin", "togel", "62"]
["hut", "kota", "yogyakarta", "tercinta", "semoga", "selalu", "sukses"]	["hut", "kota", "yogyakarta", "tercinta", "sukses"]
["bwtogel", "lagi", "gacor"]	["bwtogel", "gacor"]

3.4.3.8 Stemming

Stemming adalah proses mengubah kata berimbuhan ke bentuk dasarnya, misalnya “bermain” menjadi “main”. Tujuannya untuk mengurangi variasi kata sehingga memperkaya representasi fitur teks.

Tabel 3.11 Proses Stemming

Sebelum	Sesudah
["bagi", "bagi", "maxwin", "togel", "62"]	"bagi", "bagi", "maxwin", "togel", "62"
["hut", "kota", "yogyakarta", "tercinta", "sukses"]	["hut", "kota", "yogyakarta", "cinta", "sukses"]
["bwrtogel", "gacor"]	["bwrtogel", "gacor"]

3.4.4 Split Data

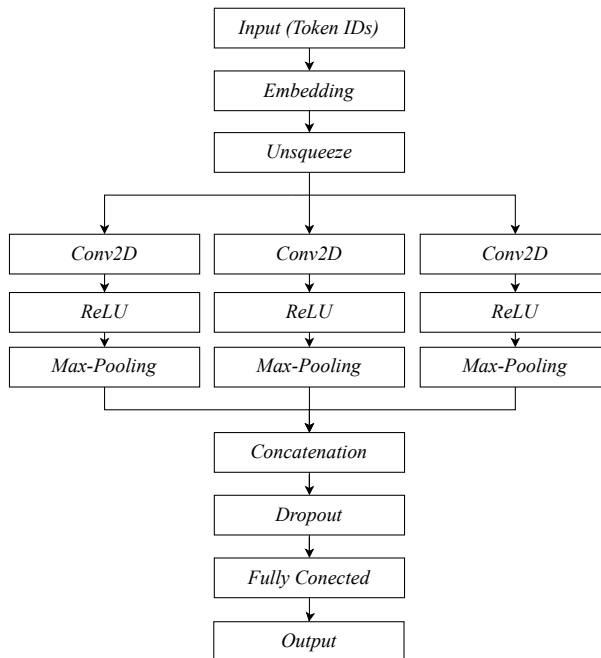
Dalam penelitian ini digunakan metode 5-fold cross validation untuk membagi dataset. Seluruh data dibagi menjadi lima bagian yang berukuran sama. Pada setiap iterasi pelatihan, empat bagian (sekitar 80%) digunakan sebagai data latih, sedangkan satu bagian sisanya (sekitar 20%) digunakan sebagai data validasi. Proses ini dilakukan secara bergantian, sehingga setiap bagian data akan memperoleh giliran sebagai data validasi. Dengan demikian, rasio latih:validasi pada setiap fold tetap 80:20, namun pembagian dilakukan berulang sebanyak lima kali agar evaluasi model menjadi lebih rata dan menyeluruh.

Pelatihan dilakukan dari awal (training from scratch) tanpa memanfaatkan bobot pra-latih (pre-trained weights), dengan parameter model diinisialisasi secara acak. Selama proses pelatihan, model dioptimasi menggunakan algoritma Adam dengan Binary CrossEntropy sebagai fungsi loss untuk menangani klasifikasi biner. Setiap epoch pelatihan diulang hingga model mencapai

konvergensi, dan evaluasi pada setiap fold membantu mengurangi risiko overfitting dan memberikan estimasi performa yang lebih akurat pada data baru.

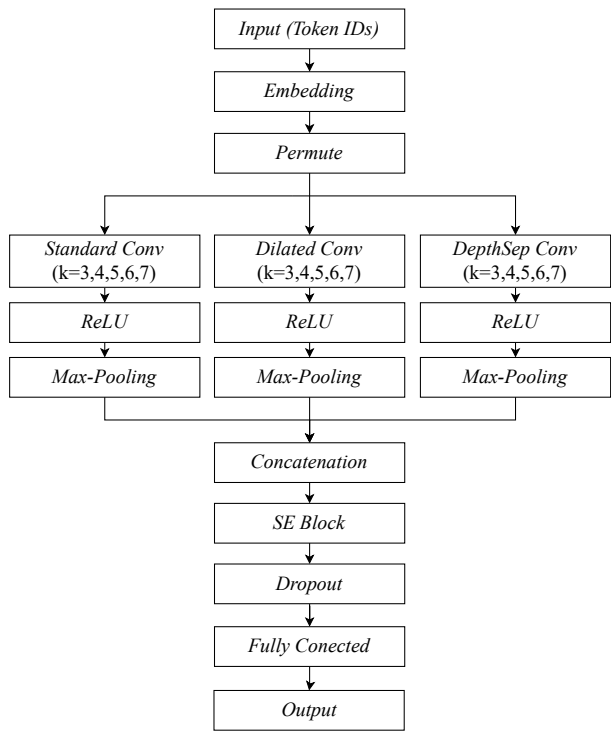
3.4.5 Pembuatan Model

Tahap pemodelan dilakukan dengan membangun arsitektur Convolutional Neural Network (CNN) menggunakan bahasa pemrograman Python dan pustaka PyTorch. Model yang dikembangkan terdiri atas beberapa lapisan sesuai dengan rancangan. Pada penelitian ini digunakan dua rancangan arsitektur, yaitu arsitektur model dasar dan arsitektur model JOC (Judi Online Comment). Detail ilustrasi masing-masing arsitektur dapat dilihat pada Gambar 3.4 dan Gambar 3.5.



Gambar 3.4 Diagram Arsitektur Model Dasar Text CNN

Gambar 3.4 menunjukkan rancangan arsitektur model dasar TextCNN yang terdiri atas embedding layer, unsqueeze, convolutional layer, ReLU, max-pooling layer, concatenation, dropout, dan fully connected layer.



Gambar 3.5 Diagram Arsitektur Model JOC Text CNN

Gambar 3.5 memperlihatkan rancangan arsitektur model JOC yang merupakan pengembangan dari arsitektur TextCNN dasar dengan tingkat kompleksitas yang lebih tinggi. Jika pada model dasar hanya digunakan tiga jalur convolution sederhana dengan ukuran kernel berbeda, maka pada model JOC ditambahkan berbagai cabang convolution seperti standard convolution, dilated convolution, dan depthwise separable convolution dengan variasi kernel

yang lebih beragam. Selain itu, model JOC juga dilengkapi dengan blok Squeeze-Excite (SE) untuk memperkuat representasi fitur serta penerapan dropout tambahan guna mencegah overfitting. Modifikasi ini menjadikan model JOC lebih kaya dalam mengekstraksi pola linguistik yang kompleks dan lebih efektif dalam mendeteksi komentar bermuatan promosi judi online dibandingkan arsitektur dasar.

3.4.5.1 *Input Layer*

Pada tahap awal, data komentar yang telah melalui proses tokenization akan diubah menjadi deretan indeks (token ID). Setiap kata direpresentasikan sebagai angka unik sesuai dengan kosakata tokenizer IndoBERT. Contoh: "judi online bahaya" → [1234, 5678, 9012].

3.4.5.2 *Embedding Layer*

Setiap token ID kemudian dipetakan menjadi vektor dengan dimensi tetap, dalam penelitian ini sebesar 300 dimensi (lebih besar dibandingkan model ringan yang hanya menggunakan 100 dimensi). Embedding ini memberikan representasi semantik yang lebih kaya, sehingga kata dengan makna mirip akan memiliki vektor yang berdekatan di ruang vektor.

3.4.5.3 *Unsqueeze Operation*

Operasi unsqueeze digunakan untuk menambahkan dimensi channel pada tensor embedding, mengubah format dari (B, L, E) menjadi (B, 1, L, E). Operasi ini diperlukan agar data input kompatibel dengan convolution 2D yang membutuhkan format 4 dimensi, dimana dimensi yang ditambahkan berfungsi sebagai channel untuk pemrosesan convolution.

3.4.5.4 *Permute Operation*

Sebelum masuk ke lapisan konvolusi, tensor hasil embedding perlu diformat ulang. Proses permute mengubah urutan dimensi dari [batch_size,

seq_len, embed_dim] menjadi [batch_size, embed_dim, seq_len]. Transformasi ini diperlukan karena PyTorch Conv1D mengharuskan dimensi channel berada di posisi kedua.

3.4.5.5 Convolutional Layer

Lapisan konvolusi berfungsi mengekstraksi pola lokal (local patterns) dari urutan kata. Model menggunakan beberapa filter kernel dengan ukuran berbeda untuk menangkap pola n-gram (misalnya 3, 4, 5 kata).

- Input channel: 300 (dari embedding).
- Output channel: 100 per kernel (dua kali lipat model ringan).

Hal ini memungkinkan model mengenali fitur yang lebih kompleks dan bervariasi.

3.4.5.6 Triple Branch Architecture

- Standard Branch menggunakan convolution 1D konvensional dengan 5 kernel sizes [3,4,5,6,7] untuk mengekstraksi fitur n-gram dengan panjang berbeda. Branch ini berfungsi sebagai ekstraksi fitur dasar yang telah terbukti efektif pada TextCNN tradisional.
- Branch menggunakan dilated convolution dengan dilation rate 2 untuk memperluas receptive field tanpa menambah parameter. Kernel size 3 dengan dilation 2 memiliki receptive field efektif sebesar 5, memungkinkan model menangkap long-range dependencies dan memahami konteks kalimat yang lebih luas.
- Separable Branch menggunakan factorized convolution yang memisahkan operasi spatial dan channel-wise. Terdiri dari depthwise convolution (setiap channel diproses terpisah) dan pointwise convolution (mixing informasi antar channel), menghasilkan parameter reduction yang signifikan sambil mempertahankan kemampuan ekstraksi fitur.

3.4.5.7 *Activation Function (ReLU)*

Hasil konvolusi kemudian melewati fungsi aktivasi ReLU. Fungsi ini menghilangkan nilai negatif sekaligus memperkenalkan sifat non-linear, sehingga model mampu belajar representasi fitur yang lebih abstrak.

3.4.5.8 *Max-Pooling Layer*

Dari setiap feature map, pooling mengambil nilai maksimum. Proses ini memastikan hanya fitur yang paling dominan yang dipertahankan, sekaligus mengurangi dimensi data. Output dari tahap ini adalah 100 fitur per kernel, yang merepresentasikan pola paling kuat dari teks.

3.4.5.9 *Concatenation Layer*

Hasil pooling dari berbagai filter kernel digabungkan menjadi sebuah vektor tunggal. Dengan tiga ukuran kernel, diperoleh vektor fitur berdimensi 300 (lebih besar daripada model ringan yang hanya 100). Vektor ini membawa informasi pola kata pada berbagai skala n-gram.

3.4.5.10 *SE Block (Squeeze-and-Excitation)*

SE Block adalah mekanisme attention yang mengkalibrasikan ulang fitur berdasarkan kepentingan setiap channel. Blok ini bekerja dengan cara melakukan kompresi informasi spasial (squeeze), kemudian menghasilkan bobot attention per-channel (excitation), dan akhirnya mengalikan fitur input dengan bobot tersebut. Hal ini memungkinkan model untuk memberikan fokus lebih pada fitur yang informatif dan menekan fitur yang kurang relevan.

3.4.5.11 *Dropout*

Untuk mencegah overfitting, digunakan dropout dengan rasio 20%. Pada tahap ini, beberapa neuron dipilih secara acak untuk dimatikan hanya saat proses pelatihan, sehingga model tidak bergantung pada fitur tertentu saja.

3.4.5.12 *Fully Connected Layer*

Vektor hasil concatenation diproses melalui lapisan linear (fully connected) untuk menghubungkan fitur ke ruang label. Input berupa 300 fitur gabungan, sedangkan output berupa 2 kelas, yaitu:

- Kelas 0: komentar non-promosi judi online
- Kelas 1: komentar promosi judi online

3.4.5.13 *Output Layer (Sigmoid)*

Lapisan akhir menggunakan fungsi aktivasi sigmoid. Nilai keluaran berada pada rentang 0–1. Jika nilai mendekati 1 maka komentar diklasifikasikan sebagai promosi judi online, sedangkan jika mendekati 0 maka diklasifikasikan sebagai non-promosi.

3.4.6 **Hyperparameter Tuning**

Setelah rancangan arsitektur TextCNN selesai dibuat, tahap berikutnya adalah melakukan hyperparameter tuning untuk menentukan konfigurasi pelatihan yang optimal. Pada penelitian ini, parameter umum seperti learning rate, batch size, dropout, dan optimizer disamakan pada semua model agar pengujian lebih terkontrol serta hasil antar model dapat dibandingkan secara adil. Perbedaan utama antar varian terletak pada ukuran embedding, jumlah filter konvolusi, variasi ukuran kernel, total fitur yang dihasilkan, serta ukuran layer linear. Selain itu, model JOC yang merupakan pengembangan paling kompleks dilengkapi dengan arsitektur triple branch convolutional 1D dan tambahan Squeeze-Excite (SE) block, sehingga berbeda signifikan dibanding varian ringan, sedang, maupun berat. Rincian *hyperparameter* dari ketiga model dapat dilihat pada Tabel 4.1.

Tabel 3.12 Hyperparameter Model Dasar (Ringan, Sedang, Berat) dan Model JOC

Hyperparameter	Ringan	Sedang	Berat	JOC
Embedding	100	300	512	512
Jumlah Filter	100	128	256	256
Kernel Sizes	[3,4,5]	[3,4,5]	[3,4,5,7]	[3,4,5,6,7]
Total Features	300	384	1024	3840
Ukuran Layer Linear	300→1	384→1	1024→1	3840→1
Architecture	Conv2D	Conv2D	Conv2D	Triple Branch Conv1D
SE Block	-	-	-	Ada
Dropout	0.2	0.2	0.2	0.2
Batch Size	32	32	32	32
Learning Rate	1e-4	1e-4	1e-4	1e-4
Optimizer	Adam	Adam	Adam	Adam

3.5 Ilustrasi Perhitungan

Subbab ini secara khusus menjabarkan proses alur input teks yang telah diproses (misalnya komentar “main judi online gratis sekarang”) hingga akhirnya diprediksi sebagai promosi judi online. Berikut adalah penjelasan contoh perhitungan sederhana dari alur model TextCNN yang dimulai dari input hingga menghasilkan klasifikasi:

3.5.1 *Input Layer*

Komentar ditokenisasi menggunakan tokenizer IndoBERT sehingga menghasilkan deretan token ID.

Contoh:

”main judi online gratis” → [245, 678, 321, 987]

di mana setiap angka mewakili kata tertentu dalam kosakata tokenizer.

3.5.2 *Embedding Layer*

Setiap token ID diproyeksikan ke dalam ruang vektor berdimensi 300. Misalnya:

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

Hasil embedding membentuk matriks:

$$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix}, \quad X \in \mathbb{R}^{4 \times 300}$$

3.5.3 *Permute Operation*

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

$$X' \in \mathbb{R}^{300 \times 4}$$

3.5.4 *Convolutional Layer*

Filter konvolusi dengan kernel size 2, 3, dan 4 diterapkan pada X' . Misalnya untuk kernel size = 2:

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

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

3.5.5 *Activation Layer (ReLU)*

Hasil konvolusi s_i dilewatkan ke fungsi aktivasi ReLU:

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

sehingga nilai negatif dieliminasi dan hanya nilai positif yang dipertahankan.

Contoh hasil feature maps setelah ReLU:

$$\text{Kernel 2} \rightarrow [0.2, 0.9, 0.5]$$

$$\text{Kernel 3} \rightarrow [0.8, 0.3, 0.6]$$

$$\text{Kernel 4} \rightarrow [0.4, 0.7, 0.2]$$

3.5.6 Max Pooling

Ambil nilai maksimum dari masing-masing *feature map*:

$$\text{Kernel 2} \rightarrow \max([0.2, 0.9, 0.5]) = 0.9$$

$$\text{Kernel 3} \rightarrow \max([0.8, 0.3, 0.6]) = 0.8$$

$$\text{Kernel 4} \rightarrow \max([0.4, 0.7, 0.2]) = 0.7$$

Sehingga output pooling:

$$[0.9, 0.8, 0.7]$$

3.5.7 Concatenation

Gabungkan hasil pooling dari semua kernel menjadi satu vektor fitur:

$$z = [0.9, 0.8, 0.7]$$

3.5.8 Dropout

Dengan dropout rate 0.3 (30% neuron dimatikan acak saat training).

Contoh:

$$\text{Sebelum: } [0.9, 0.8, 0.7] \rightarrow \text{Sesudah: } [0.9, 0, 0.7]$$

3.5.9 Fully Connected Layer

Vektor dropout z diproyeksikan dengan bobot W dan bias b :

$$o = Wz + b$$

Misalnya hasil perhitungan:

$$[0.2, 1.5]$$

3.5.10 *Output Layer(Sigmoid)*

Aktivasi sigmoid diterapkan pada setiap nilai:

$$\sigma(0.2) \approx 0.55 \rightarrow \text{kelas 0 (non-promosi)}$$

$$\sigma(1.5) \approx 0.82 \rightarrow \text{kelas 1 (promosi judi online)}$$

Karena nilai kelas 1 lebih tinggi, maka komentar diprediksi sebagai **"Promosi Judi Online"**.

3.6 Rancangan Pengujian

Rancangan pengujian disusun untuk menilai kinerja model *Text CNN* dalam mengklasifikasikan komentar YouTube ke dua kelas, yaitu promosi judi online (label 1) dan non-promosi (label 0). Pengujian dilakukan secara sistematis agar hasil evaluasi bersifat andal, terukur, dan dapat direplikasi.

3.6.1 Studi Ablasi

3.6.1.1 Konsep Studi Ablasi

Rancangan pengujian dalam penelitian ini dirancang untuk mengevaluasi performa berbagai arsitektur model TextCNN dalam mengklasifikasikan komentar YouTube ke dalam dua kategori, yaitu komentar promosi judi online (label 1) dan non-promosi (label 0). Proses pengujian dilakukan secara sistematis dengan melibatkan perbandingan antara model ringan, sedang, berat, serta satu model baru (Model X) yang belum pernah digunakan pada penelitian sebelumnya. Seluruh model dilatih menggunakan skema 5-fold cross validation agar hasil evaluasi lebih andal, terukur, dan dapat direplikasi. Setelah diperoleh model dengan kinerja terbaik, dilakukan pengujian lanjutan berupa studi ablasi tanpa menggunakan skema k-fold validation untuk menilai kontribusi tiap komponen arsitektur terhadap keseluruhan performa model.

3.6.1.2 Skenario Pengujian

penelitian ini merancang empat skenario studi ablasi sebagai berikut:

1. Skenario 1 – Baseline (Model Lengkap)

Pada skenario pertama, digunakan arsitektur TextCNN secara penuh sesuai rancangan penelitian ini. Model terdiri dari embedding layer berbasis nn.Embedding PyTorch, lapisan konvolusi dengan beberapa ukuran kernel, aktivasi ReLU, max-pooling, concatenation, fully connected layer, dropout, serta lapisan output dengan aktivasi sigmoid. Optimizer yang digunakan adalah Adam dengan learning rate standar sesuai rancangan awal. Skenario ini menjadi baseline untuk mengukur performa model dalam kondisi optimal dengan seluruh komponen utama aktif. Hasil dari skenario ini berfungsi sebagai acuan pembandingan terhadap skenario-skenario berikutnya.

2. Skenario 2 – Perubahan *Learning Rate*

Skenario kedua dilakukan dengan memodifikasi nilai learning rate pada proses pelatihan model. Tujuannya adalah untuk mengevaluasi pengaruh kecepatan pembelajaran terhadap konvergensi dan kinerja akhir model. Nilai learning rate yang lebih tinggi atau lebih rendah dibanding baseline dapat memperlihatkan bagaimana sensitivitas model terhadap parameter ini, apakah dapat mempercepat pelatihan, atau justru menimbulkan ketidakstabilan dalam proses optimisasi.

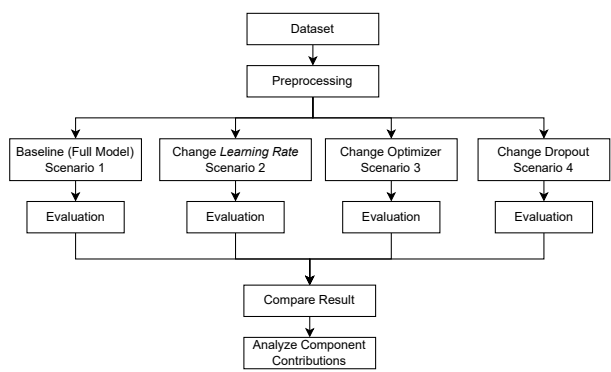
3. Skenario 3 – Perubahan *Optimizer* (Muon)

Pada skenario ketiga, optimizer Adam diganti dengan optimizer Muon. Penggantian ini bertujuan untuk menilai sejauh mana jenis algoritma optimisasi mempengaruhi kualitas pembelajaran model. Dengan menggunakan Muon, diharapkan diperoleh perbandingan performa yang jelas, apakah terjadi peningkatan, penurunan, atau stabilitas hasil dibandingkan dengan baseline yang menggunakan Adam.

4. Skenario 4 – Perubahan *Dropout*

Skenario keempat memfokuskan pengujian pada variasi nilai dropout. Dropout digunakan untuk mengurangi risiko overfitting dengan cara menonaktifkan sebagian neuron selama pelatihan. Dengan mengubah besar persentase dropout, dapat diamati pengaruhnya terhadap keseimbangan antara akurasi pada data latih dan validasi, serta sejauh mana dropout berperan dalam menjaga kemampuan generalisasi model.

3.6.1.3 Skema Pengujian



Gambar 3.6 Diagram Skema Studi Ablasi

Pengujian dilakukan menggunakan 1 fold pada setiap skenario dengan konfigurasi hyperparameter yang sama, sehingga perbedaan hasil kinerja semata-mata dipengaruhi oleh keberadaan atau penghapusan komponen. Proses pengujian dilakukan secara bertahap, dimulai dari pelatihan model hingga evaluasi performa berdasarkan metrik akurasi, presisi, recall, dan F1-score.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Bab ini menyajikan hasil penelitian yang telah dilakukan berdasarkan diagram alur penelitian pada Bab III. Pada penelitian ini, tahapan yang dilakukan meliputi pengumpulan data, *pre-processing* dan augmentasi data, pengembangan model, pengujian model dan evaluasi.

4.1.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini berasal dari platform Kaggle dengan judul ‘youtubechatjogjclean’ (tautan: <https://www.kaggle.com/datasets/yaemico/deteksi-judi-online>) serta ditambah dengan koleksi pribadi yang diperoleh dari komentar livestreaming YouTube. Jumlah total data yang digunakan mencapai 8.350 komentar dengan distribusi yang seimbang, terdiri atas 4.163 komentar non-promosi (label 1) dan 4.187 komentar promosi judi online (label 0).

4.1.2 Pre-Processing dan Augmentasi Data

4.1.2.1 Pre-Processing Data

Tahapan preprocessing dilakukan untuk memastikan data komentar memiliki format seragam, bersih, dan siap digunakan oleh model. Dataset awal berasal dari dataset_judol.csv dengan kolom [id, userName, comment, kategori], di mana hanya entri valid yang digunakan. Label terdiri dari dua kelas: judol (0) dan non-judol (1).

Proses preprocessing meliputi: case folding, pembersihan URL, mention, hashtag, simbol non-alfanumerik, digit, serta spasi ganda. Karakter berulang lebih dari tiga dikurangi, dan kata-kata slang dinormalisasi menggunakan

kamus (slang-dictionary.json). Setelah itu teks ditokenisasi, stopwords dihapus, serta stemming diterapkan secara opsional dengan pustaka Sastrawi. Hasil akhir berupa teks bersih yang kemudian diubah menjadi representasi subword menggunakan tokenizer IndoBERT (indobenchmark/indobert-base-p1) dengan panjang tetap 128 token.

Kode 4.1 Implementasi kode preprocessing data teks

```

1 import re, json, nltk
2 from nltk.tokenize import word_tokenize
3 from nltk.corpus import stopwords
4 from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
5 from transformers import AutoTokenizer
6
7 # Resource
8 INDO_STOP = set(stopwords.words('indonesian'))
9 STEMMER = StemmerFactory().create_stemmer()
10 tokenizer =
11     AutoTokenizer.from_pretrained("indobenchmark/indobert-base-p1")
12
13 # Regex dasar
14 URL_RE = re.compile(r'https?://\S+|www\.\S+')
15 MENTION = re.compile(r'[@#]\w+')
16 NON_ALNUM = re.compile(r'[^0-9A-Za-z - \s]')
17 MULTISPACE = re.compile(r'\s+')
18 DIGIT_RE = re.compile(r'\d+')
19 REPEAT_RE = re.compile(r'(\d){2,}')
20
21 def preprocess(text: str) -> str:
22     text = text.lower()
23     text = URL_RE.sub(' ', text)
24     text = MENTION.sub(' ', text)
25     text = NON_ALNUM.sub(' ', text)
26     text = DIGIT_RE.sub(' ', text)
27     text = REPEAT_RE.sub(r'\1', text)
28     toks = word_tokenize(text)
29     toks = [t for t in toks if t not in INDO_STOP]
30     return ' '.join(toks)
31
32 def to_model_inputs(text, max_len=128):
33     enc = tokenizer(text, padding="max_length",
34                     truncation=True, max_length=max_len,
35                     return_tensors="pt")
36     return enc["input_ids"].squeeze(0),
37            enc["attention_mask"].squeeze(0)

```

4.1.2.2 Augmentasi Data

Tahapan augmentasi digunakan untuk memperkaya variasi komentar dan meningkatkan robustness model terhadap keragaman bahasa pada komentar YouTube. Proses ini dilakukan setelah preprocessing dan sebelum tokenisasi IndoBERT, dengan kontrol probabilitas melalui parameter `augment_prob`.

Sumber augmentasi berasal dari berkas `augmentasi.json` yang memuat

pasangan frasa alternatif (replace_phrases) dan sinonim kata (synonyms). Strategi yang diterapkan meliputi penggantian frasa/kata dengan kandidat lain, penghapusan stopwords, serta teknik EDA ringan seperti random typo, random swap, dan random deletion.

Augmentasi hanya diterapkan pada data latih, sedangkan data validasi dan uji tetap menggunakan teks asli agar evaluasi objektif. Dengan pendekatan ini, dataset pelatihan menjadi lebih bervariasi tanpa mengubah makna komentar, sehingga model lebih adaptif terhadap perbedaan gaya bahasa, ejaan, maupun variasi penulisan.

Kode 4.2 Implementasi kode augmentasi data teks

```

1 import json, random, re
2
3 def load_augmentasi(path="augmentasi.json"):
4     with open(path, "r", encoding="utf-8") as f:
5         data = json.load(f)
6         return data.get("replace_phrases", {}), data.get("synonyms", {})
7
8 def replace_phrases(text, phrase_map):
9     for fr in sorted(phrase_map, key=len, reverse=True):
10         if fr in text:
11             cand = random.choice(phrase_map[fr])
12             text = re.sub(rf"\b{fr}\b", cand, text)
13     return text
14
15 def replace_synonyms(tokens, syn_map, p=0.3):
16     return [random.choice(syn_map[w]) if w in syn_map and
17             random.random() < p else w for w in tokens]
18
19 def augment_text(text, phrase_map, syn_map, prob=0.5):
20     if random.random() > prob: return text
21     text = replace_phrases(text, phrase_map)
22     toks = replace_synonyms(text.split(), syn_map)
23     #Random Swap
24     if len(toks) > 1 and random.random() < 0.1: random.shuffle(toks)
25     #Random Delete
26     if len(toks) > 1 and random.random() < 0.1:
27         toks.pop(random.randrange(len(toks)))
28     return " ".join(toks)

```

4.1.3 Pengembangan Model

Tujuan subbagian ini adalah merancang arsitektur CNN untuk klasifikasi komentar YouTube bertopik judi *online* (biner). Seluruh model menerima urutan token hasil tokenisasi IndoBERT dan menghasilkan satu logit (BCEWithLogitsLoss).

Bagian ini menjelaskan proses pembuatan model teks CNN untuk

klasifikasi komentar YouTube (promosi-judi *online* — non-promosi judi *online*). Semua model menerima token hasil tokenizer IndoBERT dan mengeluarkan satu logit (dipakai oleh BCEWithLogitsLoss).

Komentar YouTube yang sudah diproses (dibersihkan dari URL, mention, hashtag, angka, dan slang dinormalisasi) menjadi input bagi model. Teks tersebut ditokenisasi menggunakan tokenizer IndoBERT (indobenchmark/indobert-base-p1), yang mengubah kata/teks menjadi potongan kecil (subword tokens). Hasil tokenisasi adalah input_ids (indeks token) dan attention_mask, namun untuk CNN hanya input_ids yang digunakan. Semua urutan dipotong atau dipadatkan ke panjang tetap 128 token, dengan padding menggunakan token khusus. Ukuran kosakata (vocab_size) diambil dari tokenizer, lalu dipakai untuk membangun lapisan embedding pada model. Target (label) berupa kelas biner: 0 = judul dan 1 = non-judul.

Semua model yang akan dibuat menggunakan pola dasar TextCNN, yaitu :

- Embedding Layer : mengubah token menjadi vektor angka berdimensi tertentu (mis. 100, 300, atau 512).
- Convolutional Layers (Conv1D) : beberapa filter dengan ukuran kernel berbeda (mis. 3, 4, 5, 7) digunakan untuk menangkap pola kata/frasal.
- ReLU Activation : fungsi aktivasi untuk menambah non-linearitas.
- Global Max Pooling : memilih fitur terkuat dari setiap filter, sehingga model hanya menyimpan informasi paling penting.
- Concatenation : menggabungkan semua hasil pooling dari berbagai kernel.
- Dropout : regularisasi agar tidak terjadi overfitting.
- Linear Layer (Output) : mengeluarkan satu nilai (logit) yang akan dipakai untuk klasifikasi biner.

Ada beberapa varian model (ringan, sedang, berat), yang berbeda di ukuran

embedding, jumlah filter, dan jenis kernel yang bisa dilihat pada tabel 3.12 Hyparameter Model Dasar pada bab 3. Model paling kompleks adalah JOC-TextCNN, yang menambahkan cabang konvolusi khusus (standard, dilated, depthwise separable) ditambah modul SE (Squeeze-and-Excitation) untuk menangkap pola lokal maupun global.

Kode 4.3 Arsitektur Model Ringan, Sedang dan Berat

```

1 import torch
2 import torch.nn as nn
3 import torch.nn.functional as F
4
5 class TextCNN(nn.Module):
6     def __init__(self, vocab_size, embed_dim=100, num_classes=1,
7         filter_sizes=[3,4,5], num_filters=100, dropout=0.2):
8         super(TextCNN, self).__init__()
9         self.embedding = nn.Embedding(vocab_size, embed_dim)
10        self.convs = nn.ModuleList([
11            nn.Conv2d(1, num_filters, (fs, embed_dim)) for fs in
12            filter_sizes
13        ])
14        self.dropout = nn.Dropout(dropout)
15        self.fc = nn.Linear(len(filter_sizes) * num_filters,
16            num_classes)
17
18    def forward(self, x):
19        x = self.embedding(x)
20        x = x.unsqueeze(1)
21
22        conv_outs = []
23        for conv in self.convs:
24            c = F.relu(conv(x)).squeeze(3)
25            p = F.max_pool1d(c, c.size(2)).squeeze(2)
26            conv_outs.append(p)
27
28        out = torch.cat(conv_outs, 1)
29        out = self.dropout(out)
30        logits = self.fc(out)
31        return logits

```

Kode 4.4 Model JOC-TextCNN

```

1 import torch
2 import torch.nn as nn
3 import torch.nn.functional as F
4
5 class SEBlock(nn.Module):
6     def __init__(self, channels: int, reduction: int = 4):
7         super().__init__()
8         hidden = max(8, channels // reduction)
9         self.fc1 = nn.Linear(channels, hidden)
10        self.fc2 = nn.Linear(hidden, channels)
11
12    def forward(self, x): # (B, C)
13        s = self.fc2(F.relu(self.fc1(x)))
14        s = torch.sigmoid(s)
15        return x * s
16
17 class DepthwiseSeparableConv1d(nn.Module):
18     def __init__(self, in_ch, out_ch, kernel_size, dilation=1):
19         super().__init__()
20         padding = (kernel_size - 1) // 2 * dilation

```

```

21     self.depthwise = nn.Conv1d(in_ch, in_ch,
22                                kernel_size=kernel_size,
23                                groups=in_ch, dilation=dilation,
24                                padding=padding, bias=False)
25     self.pointwise = nn.Conv1d(in_ch, out_ch, kernel_size=1,
26                                bias=True)
27
28     def forward(self, x):
29         x = self.depthwise(x)
30         x = self.pointwise(x)
31         return x
32
33 #
34 class JOTextCNN(nn.Module):
35     def __init__(self, vocab_size, embed_dim=512, num_classes=1,
36                 kernel_sizes=(3,4,5,6,7), num_filters_per_kernel=256,
37                 dropout=0.2, pad_idx=0):
38         super().__init__()
39         self.embedding = nn.Embedding(vocab_size, embed_dim,
40                                     padding_idx=pad_idx)
41
42         # 3 branch convs
43         self.branch_std = nn.ModuleList([
44             nn.Conv1d(embed_dim, num_filters_per_kernel, k,
45                       padding=(k-1)//2) for k in kernel_sizes
46         ])
47         self.branch_dil = nn.ModuleList([
48             nn.Conv1d(embed_dim, num_filters_per_kernel, k,
49                       dilation=2, padding=((k-1)//2)*2) for k in kernel_sizes
50         ])
51         self.branch_sep = nn.ModuleList([
52             DepthwiseSeparableConv1d(embed_dim,
53                                     num_filters_per_kernel, k) for k in kernel_sizes
54         ])
55
56         self.feature_dim = 3 * len(kernel_sizes) *
57         num_filters_per_kernel
58         self.se = SEBlock(self.feature_dim, reduction=4)
59         self.dropout = nn.Dropout(dropout)
60         self.fc = nn.Linear(self.feature_dim, num_classes)
61         self.act = nn.ReLU()
62
63     def conv_branch_forward(self, x, branch):
64         pooled = []
65         for conv in branch:
66             h = self.act(conv(x))
67             p = F.max_pool1d(h, kernel_size=h.size(-1)).squeeze(-1)
68             # (B, F)
69             pooled.append(p)
70         return pooled
71
72     def forward(self, input_ids):
73         emb = self.embedding(input_ids)
74         x = emb.permute(0, 2, 1)
75
76         pooled_std = self.conv_branch_forward(x, self.branch_std)
77         pooled_dil = self.conv_branch_forward(x, self.branch_dil)
78         pooled_sep = self.conv_branch_forward(x, self.branch_sep)
79
80         z = torch.cat(pooled_std + pooled_dil + pooled_sep, dim=1)
81         # (B, C_total)
82         z = self.se(z)
83         z = self.dropout(z)
84         logits = self.fc(z)
85         # (B, 1)
86
87         return logits

```

Model dilatih menggunakan fungsi BCEWithLogitsLoss, cocok untuk

klasifikasi biner. Optimisasi biasanya menggunakan Adam, dengan opsi tambahan seperti Muon untuk eksperimen. Learning rate bisa diatur dengan scheduler (misalnya ReduceLROnPlateau) yang menurunkan laju belajar jika model tidak membaik. Early stopping juga dipakai agar pelatihan berhenti lebih awal ketika performa validasi sudah stagnan. Hiperparameter yang diuji antara lain:

- Learning rate: 1e-4, 5e-5, 1e-5
- Batch size: 16 & 32
- Dropout: 0.2 & 0.5
- Jumlah epoch: 30 (atau berhenti lebih cepat dengan early stopping)

Pelatihan menggunakan data train dan validasi dari pembagian stratified folds. Augmentasi hanya dilakukan pada data train. Model terbaik dipilih berdasarkan skor F1 tertinggi di validasi.

Evaluasi model dilakukan dengan validasi silang (cross validation), menggunakan pembagian fold yang sama untuk semua model. Metrik utama adalah F1-Score, karena data bisa tidak seimbang (misalnya komentar judul lebih sedikit). Metrik tambahan: akurasi, presisi, dan recall. Proses pengujian:

- Model dilatih pada data train, lalu diuji pada data validasi di setiap fold.
- Skor terbaik tiap fold dicatat.
- Rata-rata dan standar deviasi skor dihitung untuk menilai konsistensi model.

Selain itu, hasil dicatat di WandB agar mudah dibandingkan antar model.

4.1.3.1 Perbandingan Model

Sebagai lanjutan dari perbandingan model, tahap berikutnya adalah melakukan pengujian secara sistematis untuk memastikan hasil yang diperoleh benar-benar mencerminkan kemampuan model. Evaluasi ini dilakukan

menggunakan metode validasi silang dengan skema fold yang sama pada setiap model, sehingga perbedaan kinerja dapat dinilai secara adil dan konsisten berdasarkan metrik yang telah ditentukan.

Tabel 4.1 Hyperparameter Model Dasar (Ringan, Sedang, Berat) dan Model JOC

Hyperparameter	Ringan	Sedang	Berat	JOC
Embedding	100	300	512	512
Jumlah Filter	100	128	256	256
Kernel Sizes	[3,4,5]	[3,4,5]	[3,4,5,7]	[3,4,5,6,7]
Total Features	300	384	1024	3840
Ukuran Layer Linear	300→1	384→1	1024→1	3840→1
Architecture	Conv2D	Conv2D	Conv2D	Triple Branch Conv1D
SE Block	-	-	-	Ada
Dropout	0.2	0.2	0.2	0.2
Batch Size	32	32	32	32
Learning Rate	1e-4	1e-4	1e-4	1e-4
Optimizer	Adam	Adam	Adam	Adam

4.1.3.2 Pemilihan Model

Pemilihan model dilakukan secara pasca-eksperimen menggunakan skrip `analiseresult.py` yang mengekstraksi seluruh hasil pelatihan dari `Weights Biases (WB)`. Setiap hasil (run) yang berstatus `finished` diambil metrik utamanya, meliputi `Accuracy`, `Precision`, `Recall`, dan `F1-Score`, beserta konfigurasi pelatihannya seperti `learning rate`, `batch size`, `dropout`, dan `optimizer`.

Proses seleksi dilakukan secara berurutan dengan kriteria utama berdasarkan `F1-Score` validasi tertinggi, karena metrik ini paling mencerminkan keseimbangan antara `precision` dan `recall` pada tugas klasifikasi biner. Jika terdapat model dengan nilai `F1-Score` yang sama, maka digunakan akurasi validasi (`Accuracy`) sebagai penentu tambahan, diikuti oleh efisiensi parameter (`F1 per juta parameter`) sebagai `tie-breaker` untuk menilai keseimbangan antara performa dan kompleksitas model.

Berdasarkan hasil rekapitulasi dari skrip `analiseresult.py`, diperoleh peringkat performa model sebagai berikut:

Tabel 4.2 Hasil Perbandingan Model CNN pada Validasi

Model	F1-Score	Accuracy	Jumlah Parameter (M)	Efisiensi (F1/M)
JOC-TextCNN	0.9618	0.9613	9.0	0.107
Berat	0.9576	0.9573	6.0	0.160
X	0.9559	0.9557	4.5	0.212
Sedang	0.9536	0.9527	3.5	0.272
Ringan	0.9333	0.9313	1.5	0.622

Hasil perbandingan menunjukkan bahwa model JOC-TextCNN menempati peringkat tertinggi dengan nilai F1-Score sebesar 0.9618 dan Accuracy sebesar 0.9613. Meskipun memiliki jumlah parameter relatif besar (± 9 juta), model ini menunjukkan performa yang konsisten dan kemampuan generalisasi yang baik dalam mendeteksi komentar promosi judi online. Berdasarkan hasil tersebut, JOC-TextCNN dipilih sebagai model terbaik untuk tahap evaluasi akhir karena mampu mempertahankan keseimbangan antara precision dan recall secara optimal.

4.1.4 Pengujian Model

Setelah proses pemilihan model terbaik dilakukan, tahap berikutnya adalah pengujian. Pengujian dilakukan untuk menganalisis kontribusi dan dampak dari setiap komponen serta hiperparameter terhadap performa model JOC-TextCNN. Pengujian ini bertujuan untuk menemukan konfigurasi optimal dengan cara membandingkan beberapa variasi model terhadap sebuah model dasar (baseline). Proses evaluasi dimulai dengan menetapkan sebuah model baseline, kemudian secara bertahap dilakukan modifikasi pada satu hiperparameter pada setiap eksperimen, yaitu learning rate, optimizer, dan dropout.

Performa dari setiap model eksperimen dievaluasi menggunakan metrik Akurasi, Presisi, Recall, dan F1-Score. Metrik utama yang menjadi acuan perbandingan adalah F1-Score, karena mampu memberikan gambaran yang seimbang antara presisi dan recall. Hasil perbandingan dari seluruh skenario pengujian disajikan pada Tabel 4.3.

Tabel 4.3 Hasil Perbandingan Studi Ablasi Model JOC-TextCNN

Eksperimen	Accuracy	Precision	Recall	F1-Score
JOC-lr-5e-5	0.958683	0.959426	0.958757	0.958670
JOC-baseline	0.957485	0.957572	0.957512	0.957484
JOC-no-dropout	0.957485	0.957605	0.957517	0.957484
JOC-lr-2e-5	0.952096	0.952555	0.952155	0.952087
JOC-muon	0.864671	0.887560	0.865108	0.862741

Berikut adalah analisis mendalam untuk setiap skenario pengujian:

4.1.4.1 Baseline Model

Tahap awal pengujian dilakukan dengan menetapkan model baseline sebagai acuan dasar dalam studi ablasi. Model ini dikonfigurasi menggunakan kombinasi parameter standar, yaitu optimizer AdamW, learning rate sebesar $1e-4$, dan dropout sebesar 0.2.

Berdasarkan hasil yang ditampilkan pada Tabel 4.X, model baseline menunjukkan performa yang stabil dan cukup baik dengan Akurasi 0.9575, Presisi 0.9576, Recall 0.9575, dan F1-Score 0.9575. Nilai-nilai ini digunakan sebagai tolok ukur untuk menilai pengaruh perubahan hiperparameter pada eksperimen-eksperimen berikutnya.

4.1.4.2 Perubahan Learning Rate

Pada tahap ini, dilakukan pengujian terhadap pengaruh perubahan nilai learning rate (LR) terhadap performa model, sementara parameter lainnya seperti optimizer AdamW dan dropout 0.2 dipertahankan sama seperti pada baseline. Dua variasi nilai LR diuji, yaitu $5e-5$ dan $2e-5$.

Hasil eksperimen menunjukkan bahwa penurunan LR menjadi $5e-5$ (JOC-lr-5e-5) memberikan peningkatan performa dengan F1-Score 0.9587, lebih tinggi dibandingkan baseline (0.9575). Namun, penurunan LR lebih jauh ke $2e-5$ (JOC-lr-2e-5) justru menurunkan performa model menjadi F1-Score 0.9521. Temuan ini menunjukkan bahwa learning rate sebesar $5e-5$ merupakan

nilai yang paling optimal untuk model JOC-TextCNN pada dataset ini, sedangkan nilai yang terlalu kecil cenderung menyebabkan proses pembelajaran menjadi lambat dan tidak stabil.

4.1.4.3 Perubahan *Optimizer*

Eksperimen berikutnya berfokus pada pengujian pengaruh pergantian optimizer terhadap performa model. Pada skenario ini, optimizer AdamW yang digunakan pada baseline diganti dengan optimizer Muon, sementara learning rate ($1e-4$) dan dropout (0.2) tetap dipertahankan.

Hasil pengujian (JOC-muon) menunjukkan adanya penurunan performa yang cukup signifikan, di mana model hanya mencapai F1-Score 0.8627, jauh di bawah performa baseline (0.9575). Hal ini mengindikasikan bahwa optimizer AdamW lebih sesuai untuk melatih model JOC-TextCNN, terutama dalam menjaga stabilitas gradien dan kecepatan konvergensi selama proses pelatihan.

4.1.4.4 Perubahan *Dropout*

Pengujian terakhir dalam studi ablasi ini dilakukan untuk menganalisis pengaruh lapisan dropout terhadap kinerja model. Pada skenario ini (JOC-no-dropout), lapisan dropout dihapus sepenuhnya (dropout = 0), sementara parameter lain tetap mengikuti konfigurasi baseline.

Hasil eksperimen menunjukkan bahwa model tanpa dropout menghasilkan F1-Score 0.9575, yang hampir identik dengan model baseline yang menggunakan dropout sebesar 0.2. Hal ini menunjukkan bahwa pada dataset dan arsitektur JOC-TextCNN, efek regularisasi dari dropout tidak terlalu berpengaruh signifikan terhadap performa. Kondisi ini dapat terjadi karena model tidak mengalami overfitting yang berarti, sehingga kontribusi dropout dalam mencegah overfitting menjadi relatif kecil.

4.1.5 Evaluasi

Evaluasi model dilakukan untuk menilai efektivitas keseluruhan arsitektur JOC-TextCNN setelah dilakukan serangkaian eksperimen ablasi terhadap beberapa parameter utama, yaitu learning rate, optimizer, dan dropout. Setiap konfigurasi diuji menggunakan skema satu kali proses training (single run). Hasil evaluasi ditinjau berdasarkan metrik Accuracy, Precision, Recall, dan F1-Score, yang masing-masing memberikan gambaran berbeda terkait kemampuan model.

- Accuracy mengukur proporsi prediksi yang benar dari seluruh data.
- Precision menggambarkan ketepatan model dalam mengidentifikasi komentar bermuatan promosi judi online.
- Recall menunjukkan kemampuan model dalam menemukan semua komentar promosi yang sebenarnya ada.
- F1-Score digunakan sebagai metrik utama karena menyeimbangkan nilai precision dan recall.

Dari hasil pengujian, **model JOC-TextCNN dengan konfigurasi learning rate 5e-5, optimizer AdamW, dan dropout 0.2** menunjukkan performa terbaik dengan F1-Score sebesar 0.9587. Nilai tersebut sedikit lebih tinggi dibandingkan model baseline dan konfigurasi lainnya, menunjukkan bahwa penyesuaian kecil pada hyperparameter dapat meningkatkan kemampuan model dalam mendeteksi komentar promosi judi online secara lebih akurat dan stabil. Dengan demikian, konfigurasi ini ditetapkan sebagai model akhir yang digunakan dalam tahap analisis hasil dan interpretasi performa model pada bagian selanjutnya.

4.2 Pembahasan

Berdasarkan hasil pengujian, dapat disimpulkan bahwa learning rate merupakan faktor paling berpengaruh terhadap peningkatan performa model. Penurunan learning rate dari $1e-4$ menjadi $5e-5$ terbukti memberikan peningkatan F1-Score, sedangkan penurunan lebih lanjut ke $2e-5$ justru menurunkan performa

karena proses konvergensi yang menjadi terlalu lambat.

Dari sisi optimizer, penggunaan AdamW memberikan hasil yang jauh lebih stabil dan akurat dibandingkan Muon, yang mengalami penurunan performa signifikan hingga F1-Score 0.8627. Hal ini menunjukkan bahwa AdamW lebih sesuai untuk arsitektur konvolusional multi-branch seperti JOC-TextCNN yang memiliki banyak parameter dan interaksi antar fitur.

Sementara itu, pada eksperimen dropout, hasil menunjukkan bahwa penghapusan lapisan dropout (nilai 0) tidak berdampak besar terhadap kinerja model. F1-Score tetap berada di sekitar 0.9575, hampir identik dengan baseline. Temuan ini mengindikasikan bahwa model tidak mengalami overfitting yang signifikan, kemungkinan karena jumlah data yang cukup besar dan efek regularisasi alami dari struktur multi-cabang serta mekanisme Squeeze-and-Excitation (SE) yang digunakan.

Secara keseluruhan, model terbaik adalah JOC-TextCNN dengan konfigurasi $LR=5e-5$, $dropout=0.2$, dan optimizer AdamW, karena memberikan keseimbangan optimal antara akurasi, stabilitas, dan efisiensi parameter. Model ini menunjukkan kemampuan generalisasi yang kuat dalam mengenali pola bahasa promosi judi online yang kompleks, termasuk variasi ejaan dan penggunaan slang khas komentar YouTube.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka kesimpulan yang diperoleh adalah sebagai berikut:

1. Arsitektur modifikasi JOC-TextCNN terbukti sangat efektif dalam mendeteksi dan mengklasifikasikan komentar YouTube yang mengandung promosi judi online. Efektivitas ini dibuktikan dengan pencapaian performa model terbaik yang mampu menghasilkan nilai Akurasi 0.9587, Presisi 0.9594, Recall 0.9588, dan F1-Score 0.9587. Nilai F1-Score yang tinggi menunjukkan bahwa modifikasi arsitektur berupa kombinasi multi-branch convolution (Standard, Dilated, Depthwise-Separable) dan mekanisme Squeeze-and-Excitation (SE) berhasil meningkatkan kemampuan model dalam mengenali pola bahasa promosi yang kompleks dan terselubung.
2. Dari studi ablasi yang dilakukan, learning rate dan optimizer teridentifikasi sebagai hiperparameter yang paling berkontribusi signifikan terhadap kinerja model. Penyesuaian learning rate dari $1e-4$ menjadi $5e-5$ memberikan peningkatan performa, menunjukkan adanya nilai optimal yang sensitif terhadap kecepatan konvergensi model. Selain itu, pemilihan optimizer AdamW terbukti jauh lebih superior dibandingkan Muon, yang menegaskan bahwa mekanisme adaptive learning rate pada AdamW sangat cocok untuk arsitektur JOC-TextCNN yang kompleks. Di sisi lain, lapisan dropout tidak menunjukkan pengaruh yang signifikan, mengindikasikan bahwa model tidak mengalami overfitting yang berarti.
3. Konfigurasi model yang paling optimal adalah JOC-TextCNN yang

menggunakan optimizer AdamW, learning rate $5e-5$, dan dropout 0.2. Kombinasi ini secara konsisten menghasilkan keseimbangan terbaik antara presisi dan recall, yang tercermin dari perolehan F1-Score tertinggi di antara semua skenario pengujian. Model dengan konfigurasi ini ditetapkan sebagai model final karena kemampuannya yang unggul dalam melakukan generalisasi dan klasifikasi pada dataset komentar judi online.

5.2 Saran

Adapun saran yang dapat diberikan untuk penelitian lebih lanjut berdasarkan penelitian ini adalah sebagai berikut:

1. Memperluas studi ablasi dengan mengevaluasi variasi ukuran filter konvolusi, jumlah filter per cabang, serta rasio reduksi pada SE Block untuk menemukan konfigurasi arsitektur yang berpotensi meningkatkan performa model.
2. Perlu dilakukan pembersihan dan penyaringan data secara lebih mendalam melalui deduplikasi serta penambahan sampel yang lebih beragam agar model tidak overfitting dan mampu melakukan klasifikasi dengan generalisasi yang lebih baik.

DAFTAR PUSTAKA

- [1] Paulina Silitonga. “PENGARUH POSITIF DAN NEGATIF MEDIA SOSIAL TERHADAP PERKEMBANGAN SOSIAL, PSIKOLOGIS, DAN PERILAKU REMAJA YANG TIDAK TERBIASA DENGAN TEKNOLOGI SOSIAL MEDIA DI INDONESIA.” *Jurnal Pendidikan Sosial Dan Humaniora* 2.4 (2023), pp. 13077–13089.
- [2] Dika Sahputra et al. “Dampak Judi Online Terhadap Kalangan Remaja (Studi Kasus Tebing Tinggi)”. *Islamic Counseling: Jurnal Bimbingan Konseling Islam* 6.2 (2022), pp. 139–156.
- [3] Zahran Kamal Muharam et al. “PENGUNAAN INFLUENCER DALAM PROMOSI JUDI ONLINE DAN SENTIMEN PUBLIK”. *Indonesian Journal of Digital Public Relations (IJDPR)* 3.1 (2024), pp. 54–63.
- [4] Wiwin Octavia Isroissholikhah. “Efektivitas content creator dalam strategi promosi di era digital”. *SIBATIK JOURNAL: Jurnal Ilmiah Bidang Sosial, Ekonomi, Budaya, Teknologi, Dan Pendidikan* 2.1 (2022), pp. 121–128.
- [5] Christian Alam Tegar Charisma and Hesti Septianita. “Pertanggungjawaban Tentang Streamer Game Yang Mempromosikan Situs Slot Judi Online Saat Live Streaming Dalam Perspektif Hukum Pidana”. *UNES Law Review* 6.4 (2024), pp. 11670–11678.
- [6] Gabriel AP Saragih. “Stop Judi Online: Bekerja Dengan Jujur dan Tekun Berdasarkan Amsal 13: 11”. *Vox Divina* (2024), pp. 17–33.
- [7] Muhammad Iqbal Abdullah and Indah Sukmawati. “Analisis Dampak Judi Online Pada Mahasiswa Universitas Negeri Padang”. *Al-Irsyad: Jurnal Pendidikan Dan Konseling* 14.2 (2024), pp. 159–173.

- [8] Andrea Maulana and Ade Yuliana. “ANALISIS SENTIMEN OPINI PUBLIK TERKAIT JUDI ONLINE PADA PENGGUNA APLIKASI X MENGGUNAKAN ALGORITMA NAÏVE BAYES DAN SUPPORT VECTOR MECHINE”. *Jurnal Informatika dan Teknik Elektro Terapan* 12.3S1 (2024).
- [9] Ailsa Agripina. “UPAYA DISKOMINFO JATIM DALAM MENCEGAH JUDI ONLINE MELALUI DIGITAL PUBLIC RELATIONS PADA AKUN INTAGRAM@ KOMINFOJATIM”. PhD thesis. Universitas Bhayangkara Surabaya, 2025.
- [10] Ghufran Afham Asnawi et al. “Perbandingan Model IndoBERT dan Model Hybrid pada Analisis Sentimen Opini Masyarakat Terhadap Judi Online”. PhD thesis. UIN Ar-Raniry Banda Aceh, 2025.
- [11] Ginni Yema Sitio, Sri Agustina Rumapea, and Posma Lumbanraja. “Analisis Sentimen Pemindahan Ibu Kota Negara Di Media Sosial Twitter Menggunakan metode Convolutional Neural Network (CNN)”. *METHOTIKA: Jurnal Ilmiah Teknik Informatika* 3.2 (2023), pp. 97–104.
- [12] Mathoril Hudha, Endang Supriyati, and Tri Listyorini. “Analisis Sentimen Pengguna Youtube Terhadap Tayangan# Matanajwamenantiterawan Dengan Metode Naïve Bayes Classifier”. *JIKO (Jurnal Inform. dan Komputer)* 5.1 (2022), pp. 1–6.
- [13] Dany Eka Saputra and Auliya Rahman Isnain. “IMPLEMENTASI ALGORITMA CONVOLUTIONAL NEURAL NETWORK UNTUK ANALISIS SENTIMEN BACAPRES 2024 PADA KOLOM KOMENTAR YOUTUBE MATA NAJWA”. *JIPi (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)* 9.3 (2024), pp. 1431–1441.
- [14] George Alexander Suwito, Imam Cholissodin, and Putra Pandu Adikara. “Analisis Sentimen Citayam Fashion Week pada Komentar YouTube

- dengan Metode Convolutional Neural Network”. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* 6.12 (2022), pp. 5948–5956.
- [15] Fawwaz Zaini Ahmad et al. “Bagaimana Masyarakat Menyikapi Pembelajaran Tatap Muka: Analisis Komentar Masyarakat pada Media Sosial Youtube Menggunakan Algoritma Deep Learning Sekuensial dan LDA”. *Jurnal Linguistik Komputasional (JLK)* 4.2 (2021), p. 40.
- [16] Ibnu Afdhal et al. “Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia”. *Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia* 5.1 (2022), pp. 122–130.
- [17] Muhammad Azhar Ali. “LKP: Klasifikasi Sentimen Tweet untuk Mendeteksi Konten pada Platform Twitter Menggunakan Natural Language Processing (NLP)”. PhD thesis. Universitas Dinamika, 2024.
- [18] Nanda Mujahidah Andini et al. “Implementasi Convolutional Neural Network (CNN) Untuk Mendeteksi Ujaran Kebencian Dan Emosi Di Twitter”. *SMATIKA JURNAL: STIKI Informatika Jurnal* 14.02 (2024), pp. 314–325.
- [19] 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).
- [20] Nur Izatul Amin, Noval Sufriyanto Talani, and Muhammad Akram Mursalim. “Analisis Semiotika Pada Strategi Persuasif Iklan Judi Online di Youtube Channel PlayOJO”. *Tech Talk Journal* 1.2 (2025), pp. 1–11.
- [21] Salwa Tabina et al. “Prediksi Transaksi Judi Online di Indonesia Tahun 2045: Analisis Matematis dan Dampaknya Terhadap Ekonomi Nasional”. *Cartesius: Jurnal Pendidikan Matematika* (2024), pp. 219–227.

- [22] Rubén Romero et al. “MobyDeep: A lightweight CNN architecture to configure models for text classification”. *Knowledge-Based Systems* 257 (2022), p. 109914.
- [23] Simon Prananta Barus et al. “Perbandingan Algoritma Machine Learning dan Deep Learning Dalam Analisis Sentimen Komentar Video YouTube Berjudul Hidup Tanpa Sosial Media”. *JIKO (Jurnal Informatika dan Komputer)* 9.2 (2025), pp. 370–380.
- [24] Yoon Kim. “Convolutional neural networks for sentence classification”. *arXiv preprint arXiv:1408.5882* (2014).
- [25] Jyothis Joseph, S Vineetha, and NV Sobhana. “A survey on deep learning based sentiment analysis”. *Materials Today: Proceedings* 58 (2022), pp. 456–460.
- [26] Gagandeep Kaur and Amit Sharma. “A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis”. *Journal of big data* 10.1 (2023), p. 5.
- [27] Lal Khan et al. “Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media”. *Applied Sciences* 12.5 (2022), p. 2694.
- [28] Sayyida Tabinda Kokab, Sohail Asghar, and Shehneela Naz. “Transformer-based deep learning models for the sentiment analysis of social media data”. *Array* 14 (2022), p. 100157.
- [29] Tarun Jain et al. “Sentiment analysis on COVID-19 vaccine tweets using machine learning and deep learning algorithms”. *International Journal of Advanced Computer Science and Applications* 14.5 (2023).
- [30] Retno Kusumaningrum et al. “Deep learning-based application for multilevel sentiment analysis of Indonesian hotel reviews”. *Heliyon* 9.6 (2023).
- [31] Mohammad Eid Alzahrani et al. “Developing an Intelligent System with Deep Learning Algorithms for Sentiment Analysis of E-Commerce

Product Reviews”. *Computational Intelligence and Neuroscience* 2022.1 (2022), p. 3840071.

LAMPIRAN

A Dataset

B Hasil Wawancara

C Rincian Kasus Uji