

**ANALISIS SENTIMEN ASPEK ULASAN PENGGUNAAN
APLIKASI TINDER MENGGUNAKAN TEKNIK DEEP
LEARNING**



SKRIPSI

Galih Isa Yudhistira

00000048134

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG
2025**

**ANALISIS SENTIMEN ASPEK ULASAN PENGGUNAAN
APLIKASI TINDER MENGGUNAKAN TEKNIK DEEP
LEARNING**



Diajukan sebagai Salah Satu Syarat untuk Memperoleh

Gelar Sarjana Komputer (S.Kom)

Galih Isa Yudhistira

00000048134

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG
2025**

HALAMAN PERNYATAAN TIDAK PLAGIAT

Dengan ini saya,

Nama : Galih Isa Yudhistira

Nomor Induk Mahasiswa : 00000048134

Program studi : Sistem Informasi

Skripsi dengan judul:

ANALISIS SENTIMEN ASPEK ULASAN PENGGUNAAN APLIKASI TINDER MENGGUNAKAN TEKNIK DEEP LEARNING

merupakan hasil karya saya sendiri bukan plagiat dari karya ilmiah yang ditulis oleh orang lain, dan semua sumber, baik yang dikutip maupun dirujuk, telah saya nyatakan dengan benar serta dicantumkan di Daftar Pustaka.

Jika di kemudian hari terbukti ditemukan kecurangan/penyimpangan, baik dalam pelaksanaan skripsi maupun dalam penulisan laporan skripsi, saya bersedia menerima konsekuensi dinyatakan **TIDAK LULUS** untuk Tugas Akhir yang telah saya tempuh.

Tangerang, 23 Mei 2025



A handwritten signature in black ink.

(Galih Isa Yudhistira)

HALAMAN PERSETUJUAN

Skripsi dengan judul

ANALISIS SENTIMEN ASPEK ULASAN PENGUNAAN APLIKASI TINDER MENGGUNAKAN TEKNIK DEEP LEARNING

Oleh

Nama : Galih Isa Yudhistira
NIM : 00000048134
Program Studi : Sistem Informasi
Fakultas : Teknik dan Informatika

Telah disetujui untuk diajukan pada

Sidang Ujian Skripsi Universitas Multimedia Nusantara

Tangerang, Tanggal Pengumpulan

Pembimbing



Dr. Erick Fernando, S.Kom., M.S.I,
1029118501

Ketua Program Studi Sistem Informasi



Ririn Ikana Desanti, S.Kom., M.Kom
313058001

HALAMAN PENGESAHAN

Skripsi dengan judul

ANALISIS SENTIMEN ASPEK ULASAN PENGGUNAAN APLIKASI TINDER MENGGUNAKAN TEKNIK DEEP LEARNING

Oleh

Nama : Galih Isa Yudhistira
NIM : 00000048134
Program Studi : Sistem Informasi
Fakultas : Teknik dan Informatika

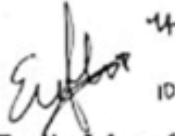
Telah diujikan pada hari Kamis, 19 Juni 2025

Pukul 15.00 s.d 17.00 dan dinyatakan

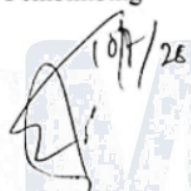
LULUS

Dengan susunan penguji sebagai berikut.

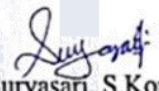
Ketua Sidang


Monika Evelin Johan, S.Kom.,
M.M.S.I .
_03270595501

Pembimbing


Dr. Erick Fernando, S.Kom.,
M.S.I,
_1029118501

Penguji


Suryasari, S.Kom., M.T.
0323088301

Ketua Program Studi Sistem Informasi


Ririn Ikana Desanti, S.Kom., M.Kom
313058001

HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan dibawah ini.

Nama : Galih Isa Yudhistira
NIM : 000000048134
Program Studi : Sistem Informasi :
Jenjang : D3/S1/S2* (pilih salah satu)
Judul Karya Ilmiah : ANALISIS SENTIMEN ASPEK ULASAN PADA
APLIKASI TINDER

Menyatakan dengan sesungguhnya bahwa saya bersedia* (pilih salah satu):

- ☒ Saya bersedia memberikan izin sepenuhnya kepada Universitas Multimedia Nusantara untuk mempublikasikan hasil karya ilmiah saya ke dalam repositori Knowledge Center sehingga dapat diakses oleh Sivitas Akademika UMN/Publik. Saya menyatakan bahwa karya ilmiah yang saya buat tidak mengandung data yang bersifat konfidensial.
- ☐ Saya tidak bersedia mempublikasikan hasil karya ilmiah ini ke dalam repositori Knowledge Center, dikarenakan: dalam proses pengajuan publikasi ke jurnal/konferensi nasional/internasional (dibuktikan dengan letter of acceptance) **.
- ☐ Lainnya, pilih salah satu:
 - ☐ Hanya dapat diakses secara internal Universitas Multimedia Nusantara
 - ☐ Embargo publikasi karya ilmiah dalam kurun waktu 3 tahun.

Tangerang, 23 Mei 2025

Yang menyatakan,



(Galih Isa Yudhistira)

KATA PENGANTAR

Segala puji dan syukur saya haturkan kepada Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan skripsi ini tepat pada waktunya. Skripsi yang berjudul "Analisis Sentimen Aspek Ulasan Penggunaan Aplikasi Tinder Menggunakan Teknik Deep Learning" ini diajukan sebagai salah satu syarat untuk memperoleh gelar Strata 1 pada Program Studi Sistem Informasi, Fakultas Teknik Informatika, Universitas Multimedia Nusantara.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan kontribusi, baik secara langsung maupun tidak langsung, sehingga tugas akhir ini dapat terselesaikan dengan baik. Untuk itu, penulis mengucapkan terimakasih:

1. Dr. Ir. Andrey Andoko, M.Sc. selaku Rektor Universitas Multimedia Nusantara.
2. Dr. Eng. Niki Prastomo, S.T., M.Sc., selaku Dekan Fakultas Teknik dan Informatika Universitas Multimedia Nusantara.
3. Ririn Ikana Desanti, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Informasi Universitas Multimedia Nusantara.
4. Dr. Erick Fernando, S.Kom., M.S.I, sebagai Pembimbing yang telah memberikan bimbingan, arahan, dan motivasi atas terselesainya tugas akhir ini.
5. Keluarga saya yang telah memberikan bantuan dukungan material dan moral, sehingga penulis dapat menyelesaikan tugas akhir ini.

Semoga karya ilmiah ini dapat memberikan kontribusi yang berarti dalam memberikan wawasan baru yang dapat berguna dalam konteks yang relevan.

Tangerang,



(Galih Isa Yudhistira)

ANALISIS SENTIMEN ASPEK ULASAN PADA APLIKASI TINDER

(Galih Isa Yudhistira)

ABSTRAK

Perkembangan pesat teknologi dan internet di Indonesia telah mengubah cara masyarakat berinteraksi, termasuk dalam penggunaan aplikasi mobile. Indonesia menempati posisi penting dalam peta global penggunaan aplikasi mobile, dengan Tinder sebagai salah satu platform kencan daring paling populer, yang pada tahun 2020 menguasai sekitar 57,6% pangsa pasar aplikasi kencan di Indonesia. Dengan lebih dari 100 juta unduhan dan jutaan ulasan pengguna di Google Play Store, Tinder menjadi sumber data penting untuk memahami pengalaman dan sentimen pengguna. Analisis sentimen otomatis terhadap ulasan ini menjadi alat krusial untuk mengidentifikasi opini positif, dan negatif secara efisien.

Penelitian ini mengkaji performa tiga model pembelajaran mendalam, yaitu Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), dan Convolutional Neural Network (CNN), dalam menganalisis sentimen ulasan pengguna Tinder di Google Play Store. Data yang digunakan terdiri dari 5000 ulasan teks terbaru dalam kurun waktu Februari 2025, yang diproses melalui tahapan Knowledge Discovery In Databases (KDD) meliputi seleksi, pra-pemrosesan, transformasi, data mining, dan evaluasi.

Model BiLSTM yang dikombinasikan dengan penyesuaian bobot kelas dan tuning hyperparameter menunjukkan performa terbaik dengan akurasi 94% serta recall kelas positif 86%, menandakan keseimbangan optimal antara sensitivitas dan presisi. Model ini melampaui BiLSTM dengan penyesuaian bobot kelas saja yang memiliki akurasi 93% dan recall positif 79%, serta BiLSTM standar dengan akurasi 92% dan recall positif 77%. Model LSTM menunjukkan performa yang bervariasi antara 92-94%, namun tuning hyperparameter pada LSTM dasar justru menurunkan akurasi menjadi 92%. Sementara itu, model CNN mempertahankan kestabilan performa pada kisaran 90-91%, meskipun dengan presisi dan recall kelas positif yang lebih rendah dibanding model LSTM dan BiLSTM. Kesimpulannya, optimasi BiLSTM melalui penyesuaian bobot kelas dan tuning hyperparameter merupakan pendekatan paling efektif untuk klasifikasi sentimen berbasis teks dalam penelitian ini.

Kata kunci: Analisis Sentimen, Google Play, Aplikasi Kencan Tinder, CNN, BiLSTM, LSTM

SENTIMENT ANALYSIS OF REVIEW ASPECTS OF TINDER APP USAGE USING DEEP LEARNING TECHNIQUES

(Galih Isa Yudhistira)

ABSTRACT (English)

The rapid development of technology and the internet in Indonesia has transformed how people interact, including their use of mobile applications. Indonesia holds a significant position in the global landscape of mobile app usage, with Tinder being one of the most popular daring dating platforms, capturing around 57.6% of the Indonesian dating app market share in 2020. With over 100 million downloads and millions of user reviews on the Google Play Store, Tinder serves as an important data source for understanding user experiences and sentiments. Automated sentiment analysis of these reviews becomes a crucial tool for efficiently identifying positive and negative opinions.

This study examines the performance of three deep learning models, namely Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Convolutional Neural Network (CNN), in analyzing the sentiment of Tinder user reviews on the Google Play Store. The data consists of 5,000 recent text reviews collected within a five-year period up to February 2025, processed through the Knowledge Discovery In Databases (KDD) stages including selection, preprocessing, transformation, data mining, and evaluation.

The BiLSTM model combined with class weight adjustment and hyperparameter tuning demonstrated the best performance with an accuracy of 94% and a positive class recall of 86%, indicating an optimal balance between sensitivity and precision. This model outperformed the BiLSTM with only class weight adjustment, which achieved 93% accuracy and 79% positive recall, as well as the standard BiLSTM with 92% accuracy and 77% positive recall. The LSTM model showed performance varying between 92-94%, but hyperparameter tuning on the basic LSTM actually decreased accuracy to 92%. Meanwhile, the CNN model maintained stable performance around 90-91%, although its precision and positive class recall were lower than those of the LSTM and BiLSTM models. In conclusion, optimizing BiLSTM through class weight adjustment and hyperparameter tuning is the most effective approach for text-based sentiment classification in this study. Keywords: Analisis Sentimen, Aplikasi Kencan, Google Play, Tinder, CNN, BiLSTM, LSTM

DAFTAR ISI

HALAMAN PERNYATAAN TIDAK PLAGIAT	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS	v
KATA PENGANTAR.....	vi
ABSTRAK	vii
ABSTRACT (English)	viii
DAFTAR ISI.....	ix
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR RUMUS	xvii
DAFTAR LAMPIRAN	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	7
1.3 Batasan Masalah	7
1.4 Tujuan dan Manfaat Penelitian.....	8
1.4.1 Tujuan Penelitian.....	8
1.4.2 Manfaat Penelitian.....	9
1.5 Sistematika Penelitian.....	9
BAB II LANDASAN TEORI	11
2.1 Penelitian Terkait	11
2.2 Teori Penelitian.....	21
2.2.1 Analisis Sentimen.....	21
2.2.2 Aplikasi Kencan	22
2.2.3 Tinder.....	23
2.2.4 Google Play Reviews.....	24
2.3 Framework dan Algoritma	25
2.3.1 Knowledge Discovery In Databases (KDD).....	25

2.3.2 Class Weights	27
2.3.4 Natural Language Processing (NLP)	28
2.3.5 Text Clasification	29
2.3.6 Adam (Adaptive Moment Estimation).....	31
2.3.7 Bidirectional Long Short Term Memory (LSTM).....	32
2.3.8 Long Short Term Memory (LSTM).....	34
2.3.9 Convolutional Neural Network (CNN)	36
2.3.10 Confusion Matrix	38
2.3.11 Uji Friedman	41
2.4 Tools dan Software Penelitian	42
2.4.1 Google Colaboratory / Google Colab	42
2.4.2 Python	43
2.4.3 Visual Studio Code (VS Code)	43
BAB III METODOLOGI PENELITIAN	44
3.1. Gambaran Umum Objek Penelitian	44
3.1.1. Objek Penelitian	44
3.2 Metode Penelitian	45
3.2.1 Pre KDD (Data Collection)	47
3.2.2 Data Selection	47
3.2.3 Data Preprocessing	48
3.2.4 Data Transformation	50
3.2.5 Data Mining	51
3.2.6 Evaluation	54
3.2.7 Implementasi Aplikasi	56
3.3 Teknik Pengumpulan Data	57
3.4 Teknik Analisis Data	57
BAB IV ANALISIS DAN HASIL PENELITIAN	61
4.1 Pre KDD (Data Collection)	61
4.2 Data Selection	64
4.3 Pre-Processing	65
4.3.1 Checking For Nulls Values	65
4.3.3 Data Cleaning	65

4.3.4	Case Folding	67
4.3.5	Tokenization	68
4.3.6	Normalization	69
4.3.7	Stopword	70
4.3.8	Labeling Sentimen.....	71
4.4	Transformation.....	75
4.4.1	Inisialisasi Tokenizer	75
4.4.2	Tokenisasi dan Padding	75
4.4.3	Encoding Label	76
4.4.4	Split Data Latih & Uji	76
4.5	Data Mining	77
4.5.1	Pemodelan BILSTM	77
4.5.2	Pemodelan BILSTM Class weights	78
4.5.3	Pemodelan BILSTM Class Weights Hyperparameter Tuning.....	80
4.5.4	Pemodelan LSTM	82
4.5.5	Pemodelan LSTM Class weights	83
4.5.6	Pemodelan LSTM Class weights Hyperparameter	85
4.5.7	Pemodelan CNN.....	87
4.5.8	Pemodelan CNN Class weights	88
4.5.9	Pemodelan CNN Class weights Hyperparameter	90
4.6	Evaluation	92
4.6.1	Hasil Evaluasi BILSTM	92
4.6.2	Hasil Evaluasi BILSTM Class weights	94
4.6.3	Hasil Evaluasi BILSTM Class weights Hyperparameter	96
4.6.4	Hasil Evaluasi LSTM.....	97
4.6.5	Hasil Evaluasi LSTM Class weights.....	99
4.6.6	Hasil Evaluasi LSTM Class weights Hyperparameter.....	101
4.6.7	Hasil Evaluasi CNN	103
4.6.8	Hasil Evaluasi CNN Class weights	105
4.6.9	Hasil Evaluasi CNN Class weights Hyperparameter	107
4.6.10	Hasil Uji Statistik Friedman	108
4.7	Implementasi Model Aplikasi.....	112

4.7.1 Fitur Pembersihan Data Teks.....	112
4.7.2 Fitur Analisa Sentimen.....	115
4.7.3 Fitur Visualisasi Data	117
4.7.4 Fitur Filterisasi Kata	120
4.7.5 Analisis Kesalahan	122
4.8 Hasil & Pembahasan	127
4.8.1 Tabel Penelitian Terdahulu	127
4.8.2 Tabel Penelitian Ini.....	128
BAB V SIMPULAN DAN SARAN	131
5.1 Simpulan.....	131
5.2 Saran.....	132
DAFTAR PUSTAKA	134
LAMPIRAN	140



DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	11
Tabel 2. 2 Confusion Matrix	38
Tabel 3. 1 Perbandingan Framework	45
Tabel 3. 2 Variabel Dataset.....	47
Tabel 3. 3 Perbandingan Bi-LSTM, LSTM dan CNN.....	58
Tabel 4. 1 Penelitian Terdahulu	127
Tabel 4. 2 Penelitian Ini	128



DAFTAR GAMBAR

Gambar 1. 1 Negara dengan Jumlah Pengguna Internet Tertinggi[3]	1
Gambar 1. 2 Aplikasi Kencan Daring yang Paling Banyak Diunduh Secara Global (2023)[4]	2
Gambar 1. 3 Tinder, Aplikasi Kencan Terpopuler di Indonesia periode tahun 2024[6]	3
Gambar 1. 4 Kasus Penipuan Aplikasi Kencan Daring	4
Gambar 2. 1 Framework KDD[32].	26
Gambar 2. 2 Struktur BI-LSTM [39]	33
Gambar 2. 3 Prosedur LSTM[40]	34
Gambar 3. 1 Alur Penelitian.....	46
Gambar 3. 2 Alur Preprocessing	48
Gambar 3. 3 Alur Transformation.....	50
Gambar 3. 4 Alur Data Mining	52
Gambar 3. 5 Alur Evaluasi.....	55
Gambar 3. 6 Raw dataset	57
Gambar 4. 1 Tahap pengumpulan data	61
Gambar 4. 2 Tahap mengambil data ulasan	62
Gambar 4. 3 Tahap hasil pengambilan data ulasan.....	62
Gambar 4. 4 Tahap menghitung jumlah data	63
Gambar 4. 5 Tahap memberi urutan data sesuai fitur.....	63
Gambar 4. 6 Tahap hasil simpan data csv.....	64
Gambar 4. 7 Membaca file csv	65
Gambar 4. 8 Checking For Nulls Values	65
Gambar 4. 9 Tahap persiapan pustaka Python.....	66
Gambar 4. 10 Tahap pembersihan teks.....	66
Gambar 4. 11 penerapan fungsi teks	67
Gambar 4. 12 Tahap perubahan huruf kecil.....	67
Gambar 4. 13 Tahap penerapan fungsi perubahan huruf kecil	68
Gambar 4. 14 Tahap tokenisasi teks untuk pemisahan kata.....	68
Gambar 4. 15 Tahap hasil tokenisasi	69
Gambar 4. 16 Tahap normalisasi dengan kamus slang.....	69
Gambar 4. 17 Hasil penerapan konversi slang.....	70
Gambar 4. 18 Tahap pengecekan frekuensi kata unik	70
Gambar 4. 19 Tahap Stopwords.....	71
Gambar 4. 20 Tabel setelah melakukan stopword	71
Gambar 4. 21 Indonesia Sentiment Lexicon	72
Gambar 4. 22 Tahap pelabelan sentimen	73
Gambar 4. 23 Hasil tahapan pelabelan.....	73
Gambar 4. 24 Wordcloud Positif dan Negatif.....	74
Gambar 4. 25 Visualisasi distribusi sentimen.....	75
Gambar 4. 26 Tahap inisialisasi tokenizer	75

Gambar 4. 27 Tahap tokenisasi dan padding	76
Gambar 4. 28 Tahap label encode.....	76
Gambar 4. 29 Tahap split data latih uji	77
Gambar 4. 30 Impor Pustaka python BILSTM.....	77
Gambar 4. 31 Pemodelan BILSTM	78
Gambar 4. 32 Perulangan Kompilasi & Pelatihan BILSTM	78
Gambar 4. 33 Impor Pustaka python BILSTM Class weights.....	79
Gambar 4. 34 menghitung class weights BILSTM.....	79
Gambar 4. 35 Definisi model BILSTM Class Weights	80
Gambar 4. 36 Perulangan Kompilasi & Pelatihan BILSTM Class weights	80
Gambar 4. 37 Impor Pustaka python BILSTM Class weights Hyperparameter...	81
Gambar 4. 38 Pemodelan Hyperparameter BILSTM	81
Gambar 4. 39 Training hyperparameter terbaik BILSTM	82
Gambar 4. 40 Impor Pustaka python LSTM.....	82
Gambar 4. 41 Pemodelan LSTM	83
Gambar 4. 42 Perulangan Kompilasi & Pelatihan LSTM	83
Gambar 4. 43 Impor Pustaka python LSTM Class weights.....	84
Gambar 4. 44 menghitung class weights LSTM.....	84
Gambar 4. 45 Definisi model LSTM Class weights	85
Gambar 4. 46 Perulangan Kompilasi & Pelatihan LSTM Class weights	85
Gambar 4. 47 Impor Pustaka python LSTM Class weights Hyperparameter.....	86
Gambar 4. 48 Pemodelan Tuning Hyperparameter LSTM.....	86
Gambar 4. 49 Training hyperparameter terbaik LSTM	87
Gambar 4. 50 Tahap persiapan Pustaka python CNN.....	87
Gambar 4. 51 Pemodelan CNN.....	88
Gambar 4. 52 Perulangan Kompilasi & Pelatihan CNN.....	88
Gambar 4. 53 Tahap Persiapan Pustaka python CNN Class weights	89
Gambar 4. 54 menghitung class weights CNN	89
Gambar 4. 55 Definisi model CNN Class weights	90
Gambar 4. 56 Perulangan Kompilasi & Pelatihan CNN Class weights.....	90
Gambar 4. 57 Impor Pustaka python CNN Class weights Hyperparameter	91
Gambar 4. 58 Pemodelan Tuning Hyperparameter CNN	91
Gambar 4. 59 Training hyperparameter terbaik CNN	92
Gambar 4. 60 Hasil proses pelatihan BILSTM.....	92
Gambar 4. 61 Plot hasil pengujian BILSTM	93
Gambar 4. 62 Hasil Laporan Klasifikasi Model BILSTM.....	94
Gambar 4. 63 Hasil proses pelatihan BILSTM dengan Class weights	94
Gambar 4. 64 Plot hasil pengujian BILSTM Class Weights	95
Gambar 4. 65 Hasil Laporan Klasifikasi Model BILSTM Class Weights.....	96
Gambar 4. 66 Akurasi BILSTM setelah tuning	96
Gambar 4. 67 Hasil Laporan Klasifikasi Model BILSTM Class Weights Hyperparameter.....	97
Gambar 4. 68 Hasil proses pelatihan LSTM.....	98

Gambar 4. 69 Plot hasil pengujian LSTM	98
Gambar 4. 70 Hasil Laporan Klasifikasi Model LSTM.....	99
Gambar 4. 71 Hasil proses pelatihan LSTM dengan Class weights	100
Gambar 4. 72 Plot hasil pengujian LSTM Class Weights	100
Gambar 4. 73 Hasil Laporan Klasifikasi Model LSTM Class Weights.....	101
Gambar 4. 74 Akurasi LSTM setelah tuning	102
Gambar 4. 75 Hasil Laporan Klasifikasi Model LSTM Class Weights Hyperparameter.....	103
Gambar 4. 76 Hasil proses pelatihan CNN	103
Gambar 4. 77 Plot hasil pengujian CNN.....	104
Gambar 4. 78 Hasil Laporan Klasifikasi Model CNN.....	105
Gambar 4. 79 Hasil proses pelatihan CNN dengan Class weights	105
Gambar 4. 80 Plot hasil pengujian CNN Class Weights.....	106
Gambar 4. 81 Hasil Laporan Klasifikasi Model CNN Class Weights.....	107
Gambar 4. 82 Akurasi CNN setelah tuning	107
Gambar 4. 83 Hasil Laporan Klasifikasi Model CNN Class Weights Hyperparameter.....	108
Gambar 4. 84 Uji Friedman Dasar	109
Gambar 4. 85 Uji Friedman ClassWeight.....	110
Gambar 4. 86 Uji Friedman ClassWeight dan Hypertuning.....	111
Gambar 4. 87 Tampilan Fitur Pembersihan Data Teks.....	113
Gambar 4. 88 Memasukan Data Mentah.....	113
Gambar 4. 89 Pemilihan Kolom Data	114
Gambar 4. 90 Kolom Dataset Setelah Preprocessing.....	114
Gambar 4. 91 Menyimpan File Dataset	115
Gambar 4. 92 Tampilan Fitur Analisa Sentimen	116
Gambar 4. 93 Pemilihan Model Algoritma.....	116
Gambar 4. 94 Load Process Model Algoritma.....	117
Gambar 4. 95 Hasil Prediksi Algoritma.....	117
Gambar 4. 96 Tampilan Fitur Visualisasi Sentimen	118
Gambar 4. 97 Pie Chart Distribusi Sentimen	118
Gambar 4. 98 Tampilan Visualisasi Kata Wordcloud	119
Gambar 4. 99 10 Kata Paling Sering Digunakan	120
Gambar 4. 100 Tampilan Fitur Filterisasi Kata	121
Gambar 4. 101 Memasukan Filter Berdasarkan Kata	121
Gambar 4. 102 Hasil Filterisasi Kata	122
Gambar 4. 103 Klasifikasi Golongan Kata	122
Gambar 4. 104 Barplot Kesalahan Hasil Prediksi BILSTM.....	123
Gambar 4. 105 Perbaikan Kesalahan Hasil Prediksi BILSTM.....	124
Gambar 4. 106 Barplot Kesalahan Hasil Prediksi LSTM.....	124
Gambar 4. 107 Perbaikan Kesalahan Hasil Prediksi LSTM.....	125
Gambar 4. 108 Gambar 4. 109 Barplot Kesalahan Hasil Prediksi CNN	126
Gambar 4. 110 Perbaikan Kesalahan Hasil Prediksi CNN	126

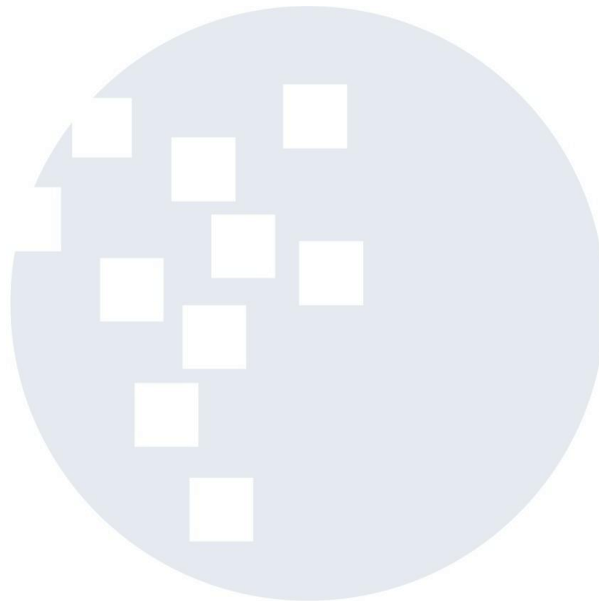
DAFTAR RUMUS

Rumus 2. 1 Rumus Adam (Adaptive Moment Estimation)	31
Rumus 2. 2 Rumus Bidirectional Long Short Term Memory (LSTM)	32
Rumus 2. 3 Rumus Long Short Term Memory (LSTM)	35
Rumus 2. 4 Prosedur Alur Analisis Sentimen CNN[43].	37
Rumus 2. 5 Rumus Accuracy	40
Rumus 2. 6 Rumus Precision	40
Rumus 2. 7 Rumus Recall	40
Rumus 2. 8 Rumus F1 - Score	41



DAFTAR LAMPIRAN

Lampiran A Turnitin Similarity Report	140
Lampiran B Form Konsultasi Bimbingan	147



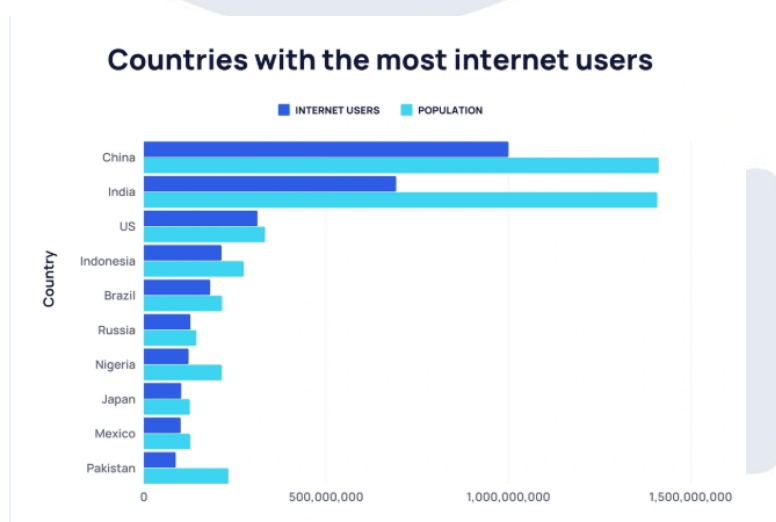
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

BAB I

PENDAHULUAN

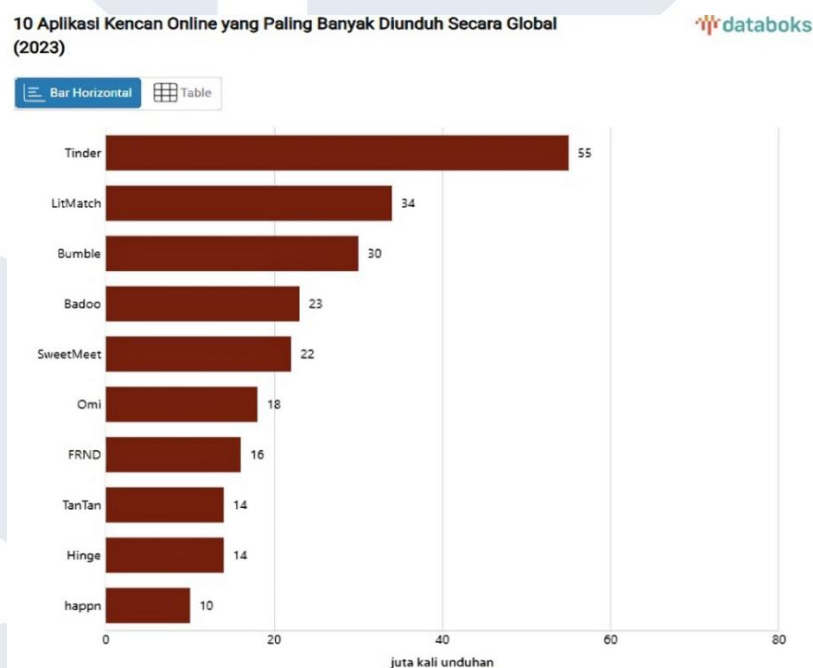
1.1 Latar Belakang

Perkembangan dunia teknologi dan internet telah membuka pintu bagi penggunaan aplikasi-aplikasi yang dapat memudahkan berbagai aktivitas manusia, seperti aplikasi belanja, transportasi, hiburan, dan berbagai keperluan lainnya. Dengan kata lain, teknologi telah menciptakan berbagai aplikasi yang memudahkan kehidupan manusia dalam berbagai aspek[1]. Di Indonesia, perkembangan teknologi dan internet sangat cepat [2]. Pertumbuhan ini didukung oleh kemajuan infrastruktur komunikasi yang luar biasa, membuat Indonesia menjadi salah satu negara dengan jumlah pengguna internet terbanyak di dunia. Berdasarkan *website* explodingtopics.com, Indonesia berada di peringkat 4 besar, bersaing dengan negara-negara yang lebih maju seperti Cina, India, dan Amerika periode 2024[3].



Gambar 1. 1 Negara dengan Jumlah Pengguna Internet Tertinggi[3]

Aplikasi kencan menjadi bukti yang terpengaruh oleh perkembangan kemajuan teknologi internet. Tinder merupakan aplikasi kencan paling banyak peminatnya. Berdasarkan *website* databoks.id periode tahun 2023 terdapat visualisasi yang menunjukkan Tinder mencapai deret pertama dari 10 aplikasi kencan yang paling banyak diunduh secara global. Aplikasi kencan Tinder yang berasal dari Amerika Serikat (AS) ini telah mencapai unduhan sampai 55 juta kali pada tahun tersebut. Tinder, sebagai bagian dari Match Group yang berbasis di Dallas, AS, memimpin pasar aplikasi kencan *daring* bersama aplikasi lain di bawah naungannya seperti Hinge, Match, OkCupid, Meetic, dan Pairs. Di posisi kedua, LitMatch mencatat 34 juta unduhan global pada 2023 dengan peningkatan signifikan di Asia Tenggara meskipun tanpa rincian angka. Sementara itu, Bumble berada di peringkat ketiga dengan 30 juta unduhan dan juga mengelola aplikasi lain seperti Badoo, Fruitz, dan Official. Badoo sendiri masuk dalam daftar aplikasi kencan terpopuler dengan total 23 juta unduhan pada 2023.



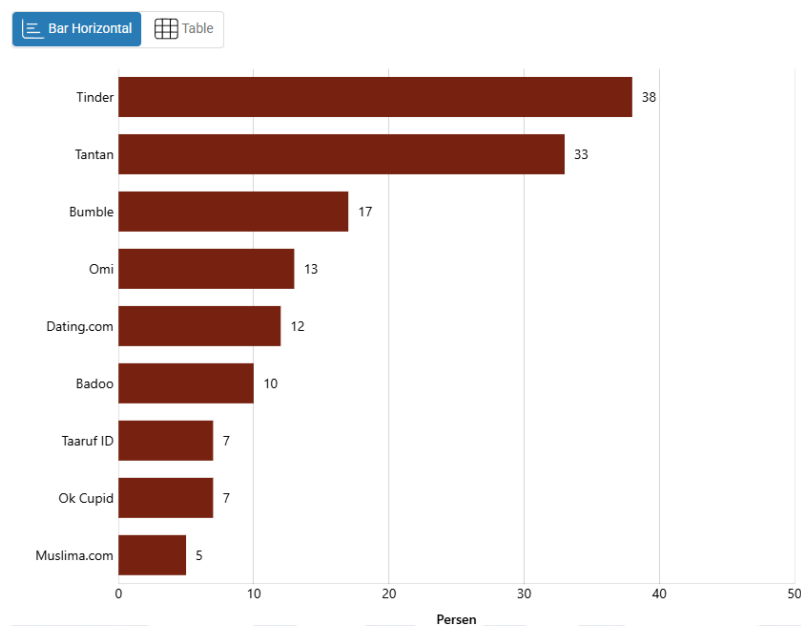
Gambar 1. 2 Aplikasi Kencan Daring yang Paling Banyak Diunduh Secara Global (2023)[4]

Perkembangan aplikasi kencan Tinder di Indonesia juga menjadi aplikasi paling laris diunduh di Indonesia. Menurut *website* databoks.id periode tahun 2024

terdapat visualisasi yang menunjukkan bahwa penggunaan aplikasi kencan daring di Indonesia cukup bervariasi. Dalam survei tersebut, sekitar 38% dari responden mengungkapkan bahwa mereka menggunakan aplikasi kencan daring Tinder, menjadikannya sebagai aplikasi yang paling banyak digunakan dibandingkan dengan aplikasi serupa lainnya[5]. Selain itu, aplikasi kencan daring Tantan, asal Tiongkok, juga cukup populer dengan penggunaan mencapai 33%, diikuti oleh Bumble dengan 17%, Omi dengan 13%, dan Dating.com dengan 12%. Terdapat juga pengguna yang memilih aplikasi seperti Badoo (10%), Taaruf (7%), Ok cupid (7%), Muslima.com (5%). Data ini memberikan gambaran yang jelas tentang preferensi pengguna aplikasi kencan daring di Indonesia pada waktu survei tersebut.

Aplikasi Kencan Online Terpopuler di Indonesia menurut Populix (Januari 2024)

databoks



Gambar 1. 3 Tinder, Aplikasi Kencan Terpopuler di Indonesia periode tahun 2024[6]

Pandangan mengenai aplikasi kencan dalam era kemajuan teknologi bervariasi[7]. Beberapa melihatnya sebagai inovasi positif yang memudahkan mencari pasangan dengan memberikan peluang lebih besar untuk bertemu orang

baru dan memperluas jaringan sosial tanpa harus berkenalan secara langsung atau konvensional. Selain hal positif, disamping itu ada juga hal negatif yang ditimbulkan dari dating apps ini.

Aplikasi kencan daring kerap disalahgunakan untuk tindak kejahatan digital seperti love scamming, modus penipuan yang memanipulasi emosi korban demi keuntungan finansial atau data pribadi[8]. Pelaku membangun ilusi hubungan romantis untuk mendekati target, lalu memanfaatkan informasi pribadi yang diperoleh untuk pemerasan atau pengurasan harta, menyebabkan kerugian materi sekaligus trauma emosional, hilangnya kepercayaan, gangguan sosial, hingga pencurian identitas[9]. Maraknya praktik ini seiring dengan pesatnya penggunaan platform digital menuntut kewaspadaan ekstra pengguna. Selain risiko kejahatan, aplikasi ini juga berpotensi memicu gangguan kesehatan mental seperti kecemasan, kecanduan, dan stres akibat ekspektasi yang tidak terpenuhi[9]. Tingginya jumlah ulasan negatif yang mencerminkan dampak psikologis dan kasus penipuan pada aplikasi kencan daring menunjukkan perlunya evaluasi mendalam terhadap pengalaman pengguna[10].

Korban Penipuan Aplikasi Kencan di Jakpus Tersebar di Vietnam-Thailand

Azhar Bagas Ramadhan - detikNews

Rabu, 29 Jan 2025 08:20 WIB



Gambar 1. 4 Kasus Penipuan Aplikasi Kencan Daring

Analisis sentimen muncul sebagai solusi efektif untuk memahami pola emosional dalam ribuan ulasan pengguna di platform seperti Google Play Store. Metode ini menjadi semakin relevan melihat popularitas aplikasi seperti Tinder

yang sejak peluncuran versi Android pada Juli 2013 telah menarik lebih dari 800.000 pengguna awal [9], dan hingga September 2023 tercatat memiliki 100 juta unduhan dengan 6,17 juta ulasan yang terdiri dari rating numerik dan komentar teks - sebuah sumber data kaya yang dapat mengungkap wawasan mendalam tentang persepsi pengguna.

Dalam melihat ulasan aplikasi di Google Play Store yang seringkali cukup banyak, waktu yang cukup lama terkadang diperlukan untuk membaca semua ulasan secara lengkap. Namun, waktu yang dimiliki untuk membaca hanya sebagian ulasan saja terkadang terbatas. Oleh karena itu, diperlukan suatu platform atau alat yang dapat secara otomatis dan mudah mengidentifikasi ulasan tersebut. Analisis ini bertujuan untuk memahami sentimen yang terkandung dalam teks ulasan yang bersifat positif atau negatif sehingga gambaran mengenai kualitas aplikasi dan pengalaman pengguna secara umum dapat diperoleh. Hal ini berguna agar sentimen atau tanggapan pengguna terhadap aplikasi tersebut dapat dipahami dengan cepat tanpa harus membaca semua ulasan secara rinci.

Analisis sentimen merupakan metode pengolahan data yang digunakan untuk mengidentifikasi dan memahami asumsi, opini, dan unsur subjektivitas dalam sebuah teks[11]. Secara keseluruhan, teknik ini memungkinkan analisis dan interpretasi berbagai elemen subjektif yang terkandung dalam teks secara komputasi[11]. Tugas utama dalam analisis sentimen adalah klasifikasi terhadap polaritas teks yang terdapat dalam dokumen, kalimat, atau pendapat[12]. Polaritas ini merujuk pada apakah teks tersebut memiliki unsur positif, negatif, atau netral[13]. Oleh karena itu, perlu dilakukan analisis terhadap opini-opini ini dalam konteks penelitian ini, sehingga ukuran untuk menilai kualitas layanan aplikasi kencana daring dapat diperoleh berdasarkan perspektif pengguna.

Perkembangan analisis sentimen dari waktu ke waktu memperluas pemahaman pembaca serta memberikan dasar untuk penelitian selanjutnya. Penelitian terdahulu mengenai analisis sentimen turut mencegah pembaca melakukan kesalahan dan pengulangan yang sama. Dalam penelitian perlunya pembelajaran dari apa yang

telah dicapai pada masa lalu dan dari apa yang berhasil maupun tidak berhasil. Salah satu metode yang banyak digunakan adalah LSTM (Long Short-Term Memory), yang telah diuji coba dalam berbagai studi. Misalnya, LSTM digunakan dalam analisis sentimen terhadap tweet berbahasa Nepal terkait COVID-19 dan menghasilkan tingkat akurasi 76,04%[13]. LSTM juga digunakan untuk mengkategorikan 1.500 ulasan aplikasi Instagram menjadi sentimen positif dan negatif, dengan akurasi 77,77%[14]. Selanjutnya, LSTM diuji pada ulasan aplikasi Traveloka dengan lima kategori sentimen, yakni sangat tidak puas, tidak puas, netral, puas, dan sangat puas, dan berhasil mencapai akurasi 80%[15]. Pada ulasan pengguna aplikasi Shopee dari Google Play Store, LSTM juga menunjukkan performa yang baik dengan akurasi keseluruhan sebesar 83% dalam klasifikasi tiga kelas sentimen: positif, netral, dan negatif[16]. Selain itu, penelitian terdahulu menunjukkan bahwa BiLSTM menonjol dengan akurasi sangat tinggi yaitu 99,14%, yang mengindikasikan kemampuannya dalam memahami konteks dua arah secara mendalam[14]. Di sisi lain, LSTM juga menunjukkan performa yang baik dengan akurasi sekitar 91%, menunjukkan keseimbangan antara performa dan efisiensi[66]. Sementara itu, CNN mencapai akurasi serupa sebesar 91%, dengan keunggulan pada kecepatan dan efisiensi pemrosesan[65].

Melihat dari penelitian terdahulu, perbandingan hasil akurasi dari algoritma Bidirectional Long Short-Term Memory (BiLSTM), Long Short-Term Memory (LSTM), dan Convolutional Neural Network (CNN) di atas, pada penelitian ini akan dilakukan uji coba pada 5000 data ulasan pengguna Aplikasi Kencan Tinder dengan menggunakan ketiga algoritma tersebut. Alasan pemilihan algoritma ini untuk menganalisis sentimen pada ulasan Tinder adalah karena BiLSTM mampu menangkap konteks teks jangka panjang dengan memproses informasi dalam dua arah, LSTM efektif dalam memodelkan dependensi urutan data, dan CNN memiliki keunggulan dalam mengenali pola lokal pada data teks. Data ulasan Tinder sangat cocok untuk analisis ini karena mencerminkan beragam pengalaman pengguna yang dapat membantu dalam memahami sentimen secara mendalam, sekaligus memberikan informasi berharga untuk pengembangan aplikasi.

1.2 Rumusan Masalah

Berdasarkan konteks yang telah dijelaskan sebelumnya, maka rumusan masalah penelitian ini adalah sebagai berikut:

1. Bagaimana performa algoritma LSTM, BiLSTM, dan CNN dalam mengklasifikasikan sentimen ulasan pengguna aplikasi Tinder di Google Play Store?
2. Sejauh mana tuning hyperparameter (jumlah unit LSTM/BiLSTM, dropout rate, dimensi embedding, kernel size, dan filter CNN) dapat meningkatkan akurasi dan performa model dalam klasifikasi sentimen?
3. Seberapa besar dampak ketidakseimbangan kelas sentimen terhadap performa model LSTM, BiLSTM, dan CNN?
4. Apakah teknik Class weights efektif dalam mengatasi ketidakseimbangan kelas pada data teks ulasan aplikasi Tinder untuk ketiga algoritma tersebut?
5. Bagaimana hasil analisis sentimen terhadap opini pengguna dapat dijadikan tolok ukur untuk menilai kualitas layanan aplikasi kencan Tinder?

1.3 Batasan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka rumusan masalah penelitian ini adalah sebagai berikut:

1. Sumber dataset yang digunakan berupa ulasan pengguna dari aplikasi kencan Tinder di Google Play Store.
2. Data yang diuji merupakan teks.
3. Data ulasan dibatasi dengan kategori paling terbaru yaitu dalam kurun waktu 12 Februari 2025 sebanyak 5000 data.
4. Penelitian ini menggunakan tiga algoritma utama: Long Short Term Memory (LSTM), Bidirectional LSTM (BiLSTM), dan Convolutional Neural Network (CNN).

5. Penelitian ini berfokus pada evaluasi performa model dalam mengklasifikasikan sentimen ulasan Tinder setelah tuning hyperparameter dan penerapan teknik Class weights. Hanya model yang digunakan dalam penelitian ini yang akan dianalisis, tanpa membandingkan algoritma lain di luar ketiga algoritma tersebut.
6. Ulasan Tinder yang dianalisis mencakup dua kelas sentimen: positif negatif dan netral.
7. Penelitian ini hanya membahas teknik Class weights sebagai metode untuk mengatasi ketidakseimbangan kelas dan tidak membandingkannya dengan teknik oversampling lainnya.
8. Penelitian ini mengkaji pengaruh tuning hyperparameter seperti jumlah unit LSTM/BiLSTM, dropout rate, dimensi embedding, serta parameter penting pada CNN terhadap performa model. Hyperparameter selain yang disebutkan tidak akan dibahas dalam penelitian ini.

1.4 Tujuan dan Manfaat Penelitian

1.4.1 Tujuan Penelitian

Adapun tujuan dari dilakukannya penelitian ini merupakan sebagai berikut:

1. Menganalisis dan membandingkan performa algoritma LSTM, BiLSTM, dan CNN dalam klasifikasi sentimen ulasan pengguna aplikasi Tinder.
2. Mengevaluasi dampak penerapan class weighting dan tuning hyperparameter terhadap peningkatan akurasi model.
3. Mengkaji pengaruh ketidakseimbangan kelas dan efektivitas teknik class weighting dalam meningkatkan performa klasifikasi sentimen.
4. Menginterpretasi hasil klasifikasi sentimen sebagai indikator persepsi pengguna terhadap kualitas layanan aplikasi.

1.4.2 Manfaat Penelitian

Berikut merupakan harapan – harapan manfaat dari penelitian ini yaitu:

1. Memberikan acuan dalam pemilihan algoritma deep learning yang paling efektif untuk analisis sentimen ulasan aplikasi.
2. Menyediakan wawasan teknis tentang penanganan data tidak seimbang menggunakan class weighting dan tuning hyperparameter.
3. Berkontribusi dalam pengembangan metode analisis sentimen, khususnya pada ulasan aplikasi sosial di platform digital.
4. Memberikan gambaran umum tentang persepsi dan pengalaman pengguna Tinder berdasarkan hasil klasifikasi sentimen.

1.5 Sistematika Penelitian

Penelitian ini terdiri dari 5 bagian, yaitu:

- BAB I Pendahuluan

Bab pertama, pendahuluan, memberikan gambaran umum tentang latar belakang penelitian, merumuskan pertanyaan, menetapkan batasan masalah, menjabarkan tujuan penelitian, menjelaskan manfaat penelitian, dan menyajikan sistematika penulisan. Latar belakang memberikan konteks secara keseluruhan, rumusan masalah merinci pertanyaan, batasan masalah menetapkan ruang lingkup, tujuan penelitian menunjukkan hasil yang diharapkan, manfaat penelitian menjelaskan relevansinya, dan sistematika penulisan menyajikan struktur secara ringkas dan terorganisir.

- BAB II Landasan Teori

Pada bab dua, landasan teori dirancang untuk membahas teori-teori yang relevan dengan penelitian yang dilakukan. Selain itu, pada bab ini akan dijelaskan berbagai teori, seperti Analisis Sentimen, Dating Apps, Google Play Reviews, dan LSTM, yang akan menjadi

dasar dari penelitian ini. Selain penjelasan teori, bab ini juga akan menyajikan tinjauan pustaka yang berasal dari jurnal-jurnal sebagai sumber informasi tambahan.

- **BAB III Metodologi Penelitian**

Pada bab tiga metodologi penelitian ini meliputi penjelasan mengenai objek penelitian secara keseluruhan dan penjelasan mengenai alur penelitian yang akan digunakan untuk mengumpulkan data, mengolah data dan mengevaluasi data yang diolah dengan menggunakan metode yang dilakukan dalam penelitian penulis.

- **BAB IV Analisis dan Hasil Penelitian**

Pada bab empat, analisa dan hasil penelitian dipaparkan, mengulas langkah-langkah penelitian dan penemuan-penemuan sesuai dengan kerangka kerja yang digunakan.

- **BAB V Simpulan dan Saran**

Bab kelima ini merangkum poin-poin dari bab-bab sebelumnya, memberikan gambaran umum. Penulis juga memberikan saran positif kepada Tinder untuk pengembangan aplikasi. Ini bukan hanya sebuah kesimpulan, tetapi juga sebuah platform untuk pandangan positif tentang topik ini.

BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Beberapa penelitian terdahulu telah memberikan kontribusi yang berharga dalam mendukung penelitian ini.

Tabel 2. 1 Penelitian Terkait

NO	Author	Title	Journal	Method / Tools	Application / Result
1.	Mamata Tharu, Sitaram Pokhrel, Badri Raj Lamichhane (2023)	Sentiment Analysis of Nepali COVID-19 Tweets using BERT-LSTM	Journal of Engineering and Sciences Volume: 2 Nomor Jurnal: 1 Halaman : 49–56 Tahun Terbit: 2023,	LSTM	Penelitian ini menggabungkan BERT dan LSTM untuk analisis sentimen tweet berbahasa Nepal terkait COVID-19. BERT digunakan untuk mengekstrak makna kata secara kontekstual, sedangkan LSTM menangkap urutan dalam data. Dataset yang digunakan adalah NepCov19Tweets. Model mencapai akurasi 76,04% dan F1-score 76%. Hasilnya lebih baik

NO	Author	Title	Journal	Method / Tools	Application / Result
					dibandingkan metode sebelumnya.
2.	Titis Arwindarti Esther Irawati Setiawan, Syaiful Imron (2023)	Analisis Sentimen Komentar Instagram menggunakan Long Short-Term Memory (LSTM) dan Word2Vec	Teknika Journal Volume: 13 Nomor: 1 Bulan/Tahun: Maret 2023	LSTM	Model LSTM dikombinasikan dengan Word2Vec untuk menganalisis komentar Instagram. Word2Vec membantu menangkap makna semantik komentar. LSTM dipakai untuk menangani sifat sekuensial data. Akurasi model mencapai 84%
3.	Dany Eka Saputra & Auliya Rahman Isnain (2024)	Implementasi algoritma convolutional neural network untuk analisis sentimen bacapres 2024 pada kolom komentar youtube mata najwa 1.	JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika) Volume: 9 Nomor: 3	CNN	Penelitian menggunakan CNN untuk analisis sentimen bacapres 2024 di kolom komentar YouTube. Model menunjukkan skor F1 tinggi hingga 94% di beberapa kategori data. Penggunaan SMOTE dan

NO	Author	Title	Journal	Method / Tools	Application / Result
			<p>Bulan/Tahun: September 2024</p> <p>Halaman : 1431–1441</p>		<p>tuning parameter meningkatkan performa. Skor F1 makro berkisar antara 60–65%. Hal ini menunjukkan ketidakseimbangan data masih berpengaruh.</p>
4.	Mujiono Sadikin and Abi Fauzan (2023)	Evaluation of Machine Learning Approach for Sentiment Analysis using Yelp Dataset	<p>Nama Jurnal: Advances in Artificial Intelligence and Machine Learning</p> <p>Volume: 3</p> <p>Nomor: 3</p> <p>Tahun: 2023</p> <p>Halaman : 74–87</p>	LSTM	<p>Model LSTM dan MLP dibandingkan untuk analisis ulasan restoran di Yelp. Word2Vec digunakan dalam tahap ekstraksi fitur. LSTM menunjukkan kinerja lebih unggul dibanding MLP. Akurasi LSTM mencapai 91%, sedangkan MLP hanya 76%. LSTM efektif untuk data sekuensial.</p>
5.	Ekin Fergan	Course Review Sentiment Analysis: A	IEEE Access	LSTM	<p>Penelitian membandingkan model ML dan DL pada</p>

NO	Author	Title	Journal	Method / Tools	Application / Result
	Tsegaye Misikir Tashu (2023)	Comparative Study Of Machine Learning and Deep Learning Methods	Volume: 10 Nomor Jurnal: - Tahun: 2022		ulasan Coursera. Teknik seperti TF-IDF, GloVe, dan berbagai model seperti Naive Bayes, SVM, LSTM, dan BERT digunakan. BERT memberikan hasil terbaik. Masalah ketidakseimbangan data diidentifikasi sebagai tantangan utama. Studi ini menyarankan solusi balancing untuk kinerja lebih baik.
6.	Minji Kim, Subeen Kim, Yoonha Park, Sangwoo Bahn, Sung Hee Ahn, Bhavadharani	User Sentiment Analysis Based on Securities Application Elements	Nama Jurnal: Behavioral Sciences Volume: 14 Nomor: 3 Tahun: 2024	logistic regression, support vector machine (SVM), LSTM (Long Short-Term Memory), and 1D CNN	Analisis dilakukan terhadap ulasan aplikasi sekuritas Korea. Metode ABSA digunakan untuk mengidentifikasi sentimen terhadap elemen tertentu seperti "update" dan

NO	Author	Title	Journal	Method / Tools	Application / Result
	NambiN arayanan (2024)		Halaman : 1-19		“login”. Model yang digunakan mencakup logistic regression, SVM, LSTM, dan CNN. Studi menunjukkan potensi ABSA dalam mendukung desain aplikasi berbasis pengguna. ABSA juga hemat biaya dan skalabel.
7.	Rosa Eliviani, Dwi Diana Wazaumi (2024)	Exploring Sentiment Trends: Deep Learning Analysis of Social Media Reviews on Google Play Store by Netizens	Nama Jurnal: Internati onal Journal of Advance s in Data and Informati on Systems (IJADIS) Volume: 5 Nomor: 1 Tanggal Terbit:	LSTM	Penelitian ini menganalisis 1.500 ulasan aplikasi Instagram. Model LSTM digunakan untuk klasifikasi sentimen. Akurasi model sebesar 77,77%, namun F1-score hanya 0,15. Hal ini menunjukkan kinerja masih moderat. Studi memberi wawasan tentang

NO	Author	Title	Journal	Method / Tools	Application / Result
			28 Maret 2024		kelebihan dan kelemahan aplikasi.
8.	Fitroh, Fahmi Hudaya (2023)	Systematic Literature Review: Analisis Sentimen Berbasis Deep Learning	Nama Jurnal: Nasional dan Sistem Informasi Teknisi Volume: 9 Nomor: 2 Maret 2023	LSTM	Penelitian ini merupakan tinjauan sistematis terhadap penggunaan deep learning untuk analisis sentimen. LSTM dan CNN adalah model yang paling sering digunakan. LSTM mencatat akurasi tertinggi hingga 99,59%. Rata-rata akurasi di berbagai studi adalah 89%. Studi ini menyoroti efektivitas deep learning secara umum.
9.	Musfiroh, Abu Tholib, Zainal Arifin (2024)	Sentiment Analysis of Shopee App Reviews on Google Play Store Using TF-IDF and Long Short-Term	Nama Jurnal: Journal of Electrical Engineering and Computer	LSTM	Analisis dilakukan pada 3.565 ulasan aplikasi Shopee dari Google Play. Sentimen dibagi menjadi positif, netral, dan negatif.

NO	Author	Title	Journal	Method / Tools	Application / Result
		Memory (LSTM)	(JEECOM) Volume: 6 Nomor: 2 2024		LSTM digunakan dengan TF-IDF untuk praproses data. Akurasi keseluruhan mencapai 83%. Namun, model kesulitan mengklasifikasi sentimen netral karena ketidakseimbangan data.
10.	Fitroh, Fitroh Hudaya, Fahmi (2023)	Analisis Sentimen Berbasis Deep Learning	Nama Jurnal: Nasional dan Sistem Informasi Teknosi Volume: 9 Nomor: 2 Mei 2024	Bidirectional Long Short-Term Memory (BiLSTM)	Model BiLSTM digunakan untuk menangkap konteks dua arah dalam data teks. Model diuji untuk klasifikasi sentimen. Hasil evaluasi menunjukkan akurasi 99,14% dan F1-score 99,12%. Model sangat seimbang dalam hal presisi dan recall. Ini menunjukkan kinerja yang hampir sempurna.

NO	Author	Title	Journal	Method / Tools	Application / Result
11.	Rina Refianti, Achmad Benny Mutiara, Ryan Arya Putra (2024)	A Lexicon-Based Long Short-Term Memory (LSTM) Model for Sentiment Analysis to Classify Halodoc Application Reviews on Google Playstore	Journal of Applied Data Sciences (JADS) Volume: 5 Nomor Jurnal: 1 Tahun: 2024	LSTM (Long Short-Term Memory)	Model LSTM berbasis leksikon digunakan untuk ulasan aplikasi Halodoc. Sentimen diklasifikasikan menjadi positif, negatif, dan netral. Dataset terdiri dari lebih dari 19.000 ulasan. Model mencapai akurasi 85,3% dan F1-score sebanding. Aplikasi web dengan Flask juga dikembangkan untuk klasifikasi.
12.	Jatla Srikanth, Avula Damodaram Shanmugam (2023)	A Deep LSTM-RNN Classification Method for COVID-19 Twitter Review Based on Sentiment Analysis	Scalable Computing: Practice and Experience (SCPE) Volume: 24 Nomor Jurnal: 3	Long Short-Term Memory (LSTM) with Synthetic Minority Oversampling Technique (SMOTE)	Model LSTM dengan SMOTE digunakan untuk analisis sentimen tweet COVID-19. Sentimen dibagi menjadi senang, sedih, netral, dan marah. Akurasi model mencapai 97,5% dengan presisi 98%.

NO	Author	Title	Journal	Method / Tools	Application / Result
			Tahun: 2024		Model ini mengungguli metode seperti Bi-LSTM dan CNN-LSTM. Hasil memberikan wawasan berharga tentang persepsi publik.
13.	Angela Lisanthoni, Ellexia Leonie Gunawan, Chelsea Ayu Adhigiadany, Dwi Arman Prasetya (2024)	Penerapan LSTM dalam Analisis Sentimen Berbasis Lexicon untuk Meningkatkan Sistem Pemantauan Citra PLN di Platform Digital	Prosiding Seminar Nasional Sains Data Volume: 4 Nomor Jurnal: 2 Tahun: 2024	LSTM (Long Short-Term Memory)	Metode ini diterapkan untuk mengklasifikasi sentimen dalam tweet yang terkait dengan COVID-19, mengkategorikannya ke dalam empat kelas: senang, sedih, netral, dan marah. Model deep learning yang diusulkan mencapai hasil yang mengesankan, dengan akurasi 97,5%, presisi 98%, recall 89%, dan mengungguli metode lain seperti Bi-LSTM, LSTM-RNN, dan

NO	Author	Title	Journal	Method / Tools	Application / Result
					CNN-LSTM. Evaluasi tersebut menunjukkan peningkatan kinerja yang signifikan dibandingkan dengan model yang sudah ada,
14.	Jimmy Tjen (2025)	Pengaruh Jenis Stopwords terhadap Akurasi Model Multinomial Naïve Bayes dalam Proses Sentimen Analisis	Vol. 16 No. 01 (2025): Jurnal Buana Informatika, Volume 16, Nomor 01, April 2025	Uji Friedman	Berhasil menemukan cara bagaimana melihat seberapa perbedaan performa antar model signifikan secara statistik

Tabel 2.1 merangkum sejumlah penelitian terdahulu yang menjadi dasar bagi studi ini. Berbagai riset, seperti nomor 7 dan 8, menunjukkan bahwa LSTM efektif dalam mengkategorikan ulasan pengguna ke dalam sentimen positif maupun negatif dengan akurasi tinggi. Penelitian nomor 3 dan 8 menegaskan keunggulan CNN dalam menangkap pola penting dalam teks, sehingga tetap handal untuk klasifikasi baik biner maupun multi kelas. Selanjutnya, penelitian nomor 10 dan 12 mengkaji pendekatan gabungan BiLSTM dan CNN, yang memadukan kemampuan BiLSTM membaca konteks dua arah dan keahlian CNN mengekstrak fitur, menghasilkan representasi teks lebih kaya serta akurasi yang lebih baik dibanding model tunggal. Selain itu, penelitian nomor 14 menyoroti perbedaan performa algoritma berdasarkan metrik accuracy, precision, recall, dan F1-Score dengan melakukan uji

Friedman. Temuan dari penelitian pendekatan ini memberikan fondasi kuat untuk memahami pengalaman pengguna serta menilai kualitas layanan aplikasi kencan. Penelitian-penelitian lain dalam tabel juga menambah wawasan terkait prapemrosesan data yang digunakan dalam studi ini.

Penelitian ini tidak hanya berfokus pada pemetaan sentimen positif dan negatif atau mengevaluasi performa model seperti BiLSTM, LSTM, dan CNN. Kebaruan utama terletak pada tujuan yang lebih spesifik, yaitu menggali pola kata-kata yang berkaitan dengan kasus penipuan yang sering terjadi dalam aplikasi kencan daring. Dengan demikian, studi ini berupaya mengidentifikasi elemen-elemen linguistik yang relevan dengan indikasi penipuan, yang belum banyak dibahas dalam penelitian sebelumnya. Pendekatan ini memberikan kontribusi baru bagi pemahaman dan pencegahan risiko di platform kencan daring sekaligus memperkaya analisis sentimen konvensional dengan aspek konteks keamanan pengguna.

2.2 Teori Penelitian

2.2.1 Analisis Sentimen

Analisis sentimen, yang sering disebut sebagai opinion mining, adalah sebuah disiplin ilmu yang menggabungkan teknik-teknik dari bidang ilmu komputer, linguistik, dan psikologi untuk mengekstrak, mengidentifikasi, dan mengukur sentimen subjektif yang terkandung di dalam teks[15]. Tujuan utamanya adalah untuk memahami opini, emosi, dan sikap yang terkandung dalam teks mengenai suatu topik, produk, layanan, atau individu[15]. Proses ini menggunakan Natural Language Processing (NLP) dan algoritme pembelajaran mesin untuk mengotomatiskan identifikasi polaritas sentimen (positif, negatif, atau netral) dan tingkat intensitas[16]. Secara lebih luas, analisis sentimen memungkinkan pemahaman yang lebih dalam tentang bagaimana orang merespons suatu masalah, menjadikannya alat yang berharga di berbagai bidang mulai dari pemasaran hingga kebijakan publik[16].

Dengan meningkatnya jumlah data teks di era digital, analisis sentimen menjadi semakin penting di berbagai sektor[17]. Dalam dunia bisnis, perusahaan

menggunakan analisis sentimen untuk memantau citra merek di media sosial, menganalisis umpan balik pelanggan, dan mengidentifikasi tren pasar[17]. Di dunia politik, teknik ini digunakan untuk mengukur opini publik terhadap kandidat atau kebijakan tertentu, memprediksi hasil pemilu, dan mengidentifikasi isu-isu yang memicu perdebatan[18]. Selain itu, analisis sentimen juga diterapkan di bidang pendidikan, kesehatan, keuangan, dan keamanan untuk mengidentifikasi pola yang relevan dan membantu membuat keputusan yang lebih tepat. Kemampuan analisis sentimen untuk memproses data dalam jumlah besar dengan cepat memberikan wawasan berharga yang sebelumnya sulit diperoleh[18].

Meskipun analisis sentimen menawarkan banyak potensi, ada beberapa tantangan dan keterbatasan yang harus diperhatikan[18]. Salah satu tantangan terbesar adalah kompleksitas bahasa manusia, termasuk penggunaan sarkasme, ironi, dan konteks spesifik yang dapat memengaruhi pemahaman sentimen[19]. Selain itu, perbedaan budaya dan bahasa dapat memengaruhi cara sentimen diekspresikan dan dipahami[20]. Mengembangkan model analisis sentimen yang akurat dan dapat diandalkan membutuhkan pemahaman yang mendalam tentang nuansa bahasa dan konteks budaya[19]. Oleh karena itu, penelitian terus mengembangkan teknik dan algoritme yang lebih canggih, seperti deep learning dan transfer learning, untuk mengatasi tantangan ini dan meningkatkan efektivitas analisis sentimen di berbagai area aplikasi[21].

2.2.2 Aplikasi Kencan

Perkembangan aplikasi kencan dalam kemajuan teknologi disebabkan oleh kombinasi antara inovasi teknologi dan kebutuhan manusia akan koneksi antarpribadi[7]. Dengan memanfaatkan kemajuan dalam komputasi mobile, kecerdasan buatan, dan analisis data, aplikasi kencan memberikan pengalaman kencan yang lebih efisien dan terhubung[22]. Algoritma pencocokan yang canggih, pemetaan lokasi berbasis GPS, dan integrasi media sosial memberikan kemampuan untuk menciptakan pertemuan yang lebih relevan[22]. Dampak-dampak ini tidak hanya membuat pencarian pasangan menjadi lebih mudah, tetapi juga membentuk pergeseran budaya dalam cara orang menjalin hubungan dan merespons

perkembangan sosial[23]. Oleh karena itu, aplikasi kencan tidak hanya mencerminkan kemajuan teknologi, tetapi juga berperan dalam membentuk dinamika hubungan manusia di era digital ini[24].

Aplikasi kencan melibatkan pemahaman aspek-aspek psikologis, sosial, teknologi, dan budaya. Teori psikologi membantu dalam memahami mengapa orang mencari pasangan dan bagaimana mereka berinteraksi di dunia digital[25]. Aplikasi kencan mencerminkan perkembangan teknologi komunikasi dan bagaimana orang berinteraksi secara daring[24]. Selain itu, aplikasi kencan juga memengaruhi perubahan dalam tatanan sosial dan budaya, dengan teori budaya dan sosiologi membantu dalam memahami perkembangan hubungan, norma sosial, dan bagaimana identitas dibangun dalam era digital[26]. Etika dan privasi juga menjadi perhatian penting, termasuk pertimbangan etika dan hak-hak individu dalam lingkungan digital[27].

2.2.3 Tinder

Tinder merupakan aplikasi pencarian pasangan berbasis digital yang pertama kali diluncurkan pada tahun 2012 dan mengalami pertumbuhan yang signifikan dalam waktu relatif singkat. Aplikasi ini memanfaatkan teknologi geolokasi serta sistem interaksi berbasis “swipe” untuk mempertemukan pengguna yang menunjukkan ketertarikan satu sama lain. Sejak kemunculannya, Tinder terus melakukan inovasi dengan menghadirkan berbagai fitur premium, seperti Tinder Plus dan Tinder Gold, serta memperluas jangkauan layanannya melalui program Tinder U yang secara khusus ditujukan bagi kalangan mahasiswa. Kehadiran berbagai fitur tersebut memperkuat eksistensi Tinder sebagai salah satu platform kencan daring paling populer di tingkat global, khususnya di kalangan generasi muda[28].

Tingkat popularitas Tinder semakin meningkat seiring dengan terjadinya perubahan pola interaksi sosial, terutama selama masa pandemi Covid-19 yang membatasi interaksi langsung antarindividu. Hingga saat ini, aplikasi tersebut telah diunduh lebih dari 400 juta kali dengan jumlah pengguna aktif bulanan yang mencapai jutaan. Kondisi ini menunjukkan bahwa teknologi digital memiliki peran

strategis dalam membentuk cara baru masyarakat dalam membangun relasi interpersonal. Selain menyediakan kemudahan dalam menjalin hubungan, Tinder juga menunjukkan kepedulian terhadap aspek keamanan pengguna melalui penyediaan fitur anti-penipuan, tombol darurat (“panic button”), serta pusat edukasi mengenai keamanan digital. Oleh karena itu, Tinder tidak hanya berfungsi sebagai sarana pencarian pasangan, melainkan juga sebagai cerminan dari dinamika transformasi sosial dalam era digital[28].

2.2.4 Google Play Reviews

Istilah "Google Play Reviews" mengacu pada tanggapan dan pengamatan yang dilakukan pelanggan di Google Play Store terkait aplikasi seluler yang telah mereka gunakan dan seberapa puas mereka dengan aplikasi tersebut. Evaluasi ini dapat dievaluasi untuk mendapatkan wawasan tentang bagaimana perasaan pengguna tentang aplikasi dan untuk menemukan area yang perlu diperbaiki. Dalam beberapa tahun terakhir, Ulasan Google Play telah berkembang menjadi sumber data penting untuk bisnis dan pengembang, memungkinkan mereka untuk lebih memahami persyaratan dan kecenderungan audiens target mereka masing-masing [29]. Evaluasi tersebut dapat dievaluasi menggunakan berbagai metode, salah satunya dikenal sebagai analisis sentimen, untuk mendapatkan wawasan yang bermanfaat dan meningkatkan pengalaman pengguna aplikasi secara keseluruhan. Proses mengidentifikasi dan mengekstraksi informasi subyektif dari data tekstual disebut sebagai analisis sentimen. Analisis sentimen merupakan metodologi yang mencakup penggunaan pemrosesan bahasa alami, pembelajaran mesin, dan analitik teks. Saat diterapkan pada pengaturan Ulasan Google Play, analisis sentimen dapat digunakan untuk mengkategorikan ulasan secara otomatis sebagai nada positif, negatif, atau tidak memihak berdasarkan bahasa yang digunakan dalam ulasan [30].

Word clouds, bar charts, dan heat maps merupakan beberapa alat dan teknik yang dapat digunakan setelah sentimen Ulasan Google Play dianalisis untuk memberikan sinopsis yang jelas dan ringkas tentang sentimen umum terkait aplikasi [31] . Hal ini dapat dicapai setelah temuan analisis telah diilustrasikan. Ini dapat membantu bisnis dan pengembang dalam menentukan masalah atau kekhawatiran

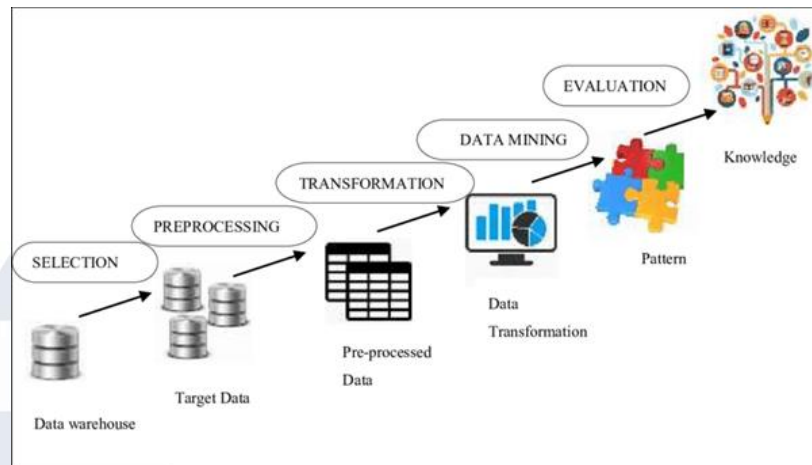
yang paling sering disuarakan oleh pengguna dan memprioritaskan pekerjaan yang mereka lakukan untuk meningkatkan pengalaman pengguna.

Secara keseluruhan, Ulasan Google Play mampu menawarkan informasi yang mencerahkan terkait perasaan pengguna terhadap aplikasi seluler. Bisnis dan pengembang perangkat lunak dapat mengevaluasi evaluasi ini menggunakan teknik dari bidang analisis sentimen untuk mendapatkan wawasan yang signifikan dan meningkatkan pengalaman pengguna. Menganalisis Ulasan Google Play telah menjadi tahap penting dalam proses peningkatan kualitas dan fungsionalitas aplikasi seluler secara keseluruhan sebagai akibat langsung dari semakin pentingnya komentar pengguna dalam proses pengembangan aplikasi seluler.

2.3 Framework dan Algoritma

2.3.1 Knowledge Discovery In Databases (KDD)

Knowledge Discovery in Database (KDD) merupakan metode yang digunakan untuk melakukan proses penambangan data secara sistematis. Fayyad et al. (1996) mengartikan KDD sebagai serangkaian langkah yang memakai teknik data mining untuk mengekstrak informasi berharga dan pola-pola dalam data yang tersembunyi. Proses ini menggunakan algoritma khusus untuk mendeteksi pola-pola tersebut dalam kumpulan data yang besar. Dunham (2003) menyatakan bahwa KDD melibatkan beberapa tahapan penting, yakni seleksi data, pra-pemrosesan, transformasi, penambangan, serta interpretasi dan evaluasi. Masing-masing tahap tersebut berkontribusi dalam menyiapkan data dan menemukan pengetahuan baru yang bermanfaat[32].



Gambar 2. 1 Framework KDD[32].

Dalam pengertian yang lebih luas, KDD adalah serangkaian tahapan yang menyeluruh untuk mengungkap pola dan makna dari data mentah. Proses ini tidak hanya sebatas penerapan teknik data mining semata, melainkan meliputi seluruh prosedur mulai dari pengambilan data yang relevan hingga evaluasi hasil. Tahapan ini mencakup proses pembersihan serta persiapan data agar cocok untuk dianalisis lebih lanjut. Selanjutnya, data diolah melalui transformasi sebelum diterapkan algoritma penambangan untuk menemukan pola tersembunyi. Akhirnya, hasil yang diperoleh dievaluasi dan diinterpretasikan sehingga dapat dimanfaatkan untuk pengambilan keputusan[32].

Seleksi data merupakan tahap awal dalam KDD yang berfokus pada pemilihan subset data yang relevan dari kumpulan data besar. Tujuannya adalah agar proses analisis nantinya dapat berjalan dengan efektif dan terarah. Pada tahap pra-pemrosesan, data yang sudah diseleksi dibersihkan dari kesalahan dan inkonsistensi serta disiapkan dalam bentuk yang dapat diproses lebih lanjut. Transformasi data dilakukan dengan tujuan mengubah data ke bentuk representasi yang sesuai untuk penerapan teknik analitik. Dengan data yang telah siap, proses penambangan dilanjutkan dengan penggunaan algoritma untuk mengekstraksi pola dan informasi penting dari data tersebut[32].

Proses penambangan data (data mining) adalah bagian inti dari KDD yang menggunakan beragam algoritma untuk menemukan pola, hubungan, dan tren yang

tersembunyi secara signifikan dalam data. Metode ini memungkinkan penggalian informasi yang sebelumnya tidak terlihat melalui teknik analisis konvensional. Setelah pola ditemukan, tahap evaluasi dilakukan untuk menilai relevansi dan keakuratan pola-pola tersebut. Interpretasi hasil berguna untuk memberikan makna yang jelas dan aplikatif dari pola yang dihasilkan. Tahapan ini memastikan bahwa pengetahuan yang diperoleh dapat digunakan secara efektif dalam pengambilan keputusan strategis[32].

Tahapan evaluasi dalam KDD berfungsi untuk menguji dan memverifikasi hasil-hasil yang diperoleh dari proses penambangan data. Proses ini melibatkan pengecekan keandalan pola dan informasi yang telah ditemukan sesuai dengan tujuan analisis. Evaluasi juga membantu dalam menyesuaikan algoritma agar menghasilkan informasi yang lebih baik. Interpretasi hasil memungkinkan para praktisi untuk memahami arti dan dampak dari pola yang terdeteksi dalam konteks bisnis atau penelitian. Akhirnya, tahap ini memastikan bahwa pengetahuan yang dihasilkan berguna dan dapat diaplikasikan secara optimal[32].

2.3.2 Class Weights

Class weights adalah cara yang sering digunakan dalam pembelajaran mesin untuk mengatasi masalah data yang tidak seimbang, di mana jumlah data pada satu kelas jauh lebih banyak dibanding kelas lain[33]. Kondisi ini sering membuat model cenderung “memperhatikan” kelas yang dominan saja, sehingga hasil prediksi untuk kelas yang lebih sedikit jumlahnya jadi kurang akurat[33]. Akibatnya, meskipun model terlihat punya performa yang bagus secara keseluruhan, sebenarnya ia kurang mampu mengenali pola penting dari kelas minoritas[33]. Masalah ini sangat umum terjadi terutama pada data nyata yang tidak selalu seimbang. Oleh karena itu, diperlukan teknik khusus agar model bisa belajar secara adil dari semua kelas yang ada.

Teknik class weights bekerja dengan memberikan bobot lebih besar pada kelas yang jumlah datanya lebih sedikit selama proses pelatihan model. Dengan begitu, kesalahan yang terjadi pada kelas minoritas dianggap lebih penting dan model terdorong untuk lebih fokus mempelajari pola dari kelas tersebut[33].

Berbeda dengan metode lain seperti oversampling atau undersampling yang mengubah jumlah data, class weights tidak mengubah data sama sekali sehingga proses pelatihan tetap efisien dan tidak menambah kompleksitas[33]. Pendekatan ini sangat praktis karena bisa langsung diterapkan pada banyak algoritma tanpa perlu manipulasi data[33]. Hal ini membuat class weights menjadi solusi yang elegan untuk masalah ketidakseimbangan data.

Dalam konteks penggunaan algoritma seperti LSTM, BiLSTM, dan CNN yang sering dipakai untuk pemrosesan data berurutan dan teks, class weights menjadi pilihan tepat untuk menangani ketidakseimbangan kelas[33]. Model-model ini biasanya memproses data yang kompleks dan besar, sehingga mengubah jumlah data dengan oversampling atau undersampling bisa memperlambat pelatihan dan menambah risiko overfitting[33]. Dengan class weights, model tetap belajar dari data asli namun dengan perhatian lebih pada kelas minoritas, sehingga performa prediksi menjadi lebih seimbang dan akurat. Selain itu, class weights mudah diintegrasikan dalam framework deep learning yang umum digunakan untuk LSTM, BiLSTM, dan CNN

2.3.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah cabang dari kecerdasan buatan (AI) yang mempelajari bagaimana komputer dapat memahami, memproses, dan menghasilkan bahasa manusia secara alami[34]. NLP menggabungkan linguistik, ilmu komputer, dan pembelajaran mesin untuk menganalisis teks atau ucapan dalam bentuk yang dapat dipahami mesin. Dalam praktiknya, NLP menggunakan berbagai teknik, seperti tokenisasi, stemming, lemmatization, dan parsing untuk memecah bahasa menjadi unit yang lebih kecil dan mudah diproses[34]. Pendekatan NLP terdiri dari metode berbasis aturan (rule-based) dan metode berbasis statistik atau pembelajaran mesin, termasuk model modern seperti transformator (contohnya BERT dan GPT)[34]. Teknologi ini mendasari berbagai aplikasi, seperti penerjemahan otomatis, analisis sentimen, pengenalan entitas bernama (NER), chatbot, dan sistem rekomendasi berbasis teks.

2.3.5 Text Classification

Text Classification merupakan teknik machine learning yang dapat mengorganisir dan mengelompokkan beragam jenis data, termasuk dokumen, rekam medis, dan teks lainnya[35]. Fungsi klasifikasi teks sebagai kerangka kerja efektif dalam pemrosesan data sangat umum, digunakan secara luas untuk tujuan seperti deteksi spam, kategorisasi berita, dan pengelompokan topik[35]. Di samping itu, dalam konteks analisis sentimen, text classification memegang peranan penting[35]. Teknik ini membantu mengkategorikan teks ke dalam kelas sentimen positif, dan negatif, dengan menggunakan algoritma machine learning untuk memahami pola linguistik dan kontekstual dalam teks[35]. Pendekatan ini memungkinkan pengenalan dan penilaian sentimen secara otomatis, memudahkan analisis besar-besaran terhadap perasaan yang terungkap dalam teks, seperti ulasan produk atau komentar di media sosial[35]

Text classification dapat dilaksanakan melalui dua pendekatan: manual dan otomatis[35]. Dalam klasifikasi manual, proses mencakup pemilihan kategori, pembuatan panduan klasifikasi, dan penentuan kategori teks oleh annotator manusia. Meskipun memberikan hasil berkualitas tinggi, metode ini memerlukan waktu yang lebih lama. Di sisi lain, klasifikasi otomatis menggunakan teknologi machine learning, NLP, dan metode lain untuk mengklasifikasikan teks secara otomatis dengan efisiensi yang lebih tinggi. Pendekatan otomatis memungkinkan penanganan volume teks besar dan peningkatan kinerja model melalui pembelajaran dari data baru. Meski demikian, klasifikasi manual tetap relevan saat interpretasi kontekstual dan pemahaman mendalam diperlukan untuk klasifikasi yang sangat akurat. Penggabungan kedua pendekatan ini menciptakan solusi yang holistik dan adaptif, memenuhi berbagai kebutuhan klasifikasi teks.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Secara umum, klasifikasi teks mencakup empat tingkat cakupan yang berbeda untuk diterapkan[36] :

1. Tingkat Dokumen

Pada tingkat dokumen, algoritme mengidentifikasi kategori yang relevan dari seluruh dokumen, memberikan gambaran umum yang luas tentang konten.

2. Tingkat Paragraf:

Tingkat paragraf membawa analisis lebih dalam, di mana algoritme memperoleh kategori yang relevan dari satu paragraf. Hal ini memberikan pemahaman yang lebih rinci tentang subbagian dokumen.

3. Tingkat Kalimat:

Pada tingkat kalimat, fokus analisis berpindah ke tingkat kalimat tunggal. Algoritme mencari kategori yang relevan untuk setiap kalimat, menghasilkan informasi yang lebih spesifik dan terfokus.

4. Tingkat Sub-Kalimat:

Tingkat sub-kalimat memberikan tingkat perincian tertinggi, di mana algoritme mendapatkan kategori yang relevan dari sub-ekspresi dalam satu kalimat. Hal ini memberikan pemahaman yang terperinci pada tingkat frasa atau struktur kalimat.

Langkah terpenting dalam klasifikasi teks mencakup pemilihan pengklasifikasi terbaik berdasarkan data yang ada dan pemahaman yang menyeluruh tentang konsep setiap metode yang digunakan. Setiap metode memiliki kelebihan dan keefektifannya masing-masing. Klasifikasi teks, sebagai metode pembelajaran mesin yang sering digunakan dalam penelitian, unggul karena kemudahannya untuk dipahami. Pendekatan ini memberikan fleksibilitas untuk menentukan kategori yang relevan pada berbagai tingkat

detail, mulai dari konteks keseluruhan dokumen hingga detail terkecil dalam sebuah kalimat[37].

2.3.6 Adam (Adaptive Moment Estimation)

Adam merupakan salah satu metode optimasi yang paling populer di dunia machine learning, khususnya deep learning[38]. Metode ini dirancang untuk membantu model belajar dengan lebih cepat ketika berhadapan dengan data yang kompleks. Fitur yang membuat Adam istimewa adalah kemampuannya untuk melacak dua jenis informasi saat latihan berlangsung: berapa banyak perubahan (gradien) yang dibutuhkan, dan seberapa sering perubahan itu terjadi. Informasi ini kemudian dikoreksi dan digunakan untuk memperbarui bobot dalam model. Dengan pendekatan ini, Adam dapat membuat keputusan yang lebih tepat dalam mengarahkan proses pembelajaran, sehingga model dapat belajar dengan lebih akurat tanpa terlalu banyak melakukan trial and error[38].

Hitung estimasi momentum pertama (mean of gradients):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

Rumus 2. 1 Rumus Adam (Adaptive Moment Estimation)

m_t : Estimasi momentum pertama pada iterasi ke-t, yaitu rata-rata bergerak dari gradien saat ini dan sebelumnya. Ini berfungsi untuk mengakumulasi gradien secara eksponensial.

β_1 : Koefisien decay untuk momentum pertama, biasanya bernilai 0.9. Menentukan seberapa besar pengaruh gradien sebelumnya terhadap estimasi saat ini.

m_{t-1} : Estimasi momentum pertama pada iterasi sebelumnya (ke-t-1).

g_t : Gradien dari fungsi loss pada iterasi ke-t. Ini adalah arah dan besarnya perubahan parameter yang diinginkan.

$1 - \beta_1$: Bobot untuk gradien saat ini, memastikan bahwa gradien terbaru tetap berkontribusi pada estimasi momentum.

Keunggulan lain dari Adam adalah kemudahan penggunaannya[38]. Metode ini tidak terlalu sensitif terhadap pengaturan awal parameter, seperti laju pembelajaran, sehingga cocok digunakan oleh siapa saja, baik pemula maupun peneliti yang sudah berpengalaman. Dalam banyak kasus, Adam mampu mempercepat proses pelatihan model dan meningkatkan kinerja, bahkan ketika bekerja dengan data dalam jumlah besar atau struktur model yang kompleks. Karena keandalannya, Adam kini menjadi salah satu algoritme bawaan yang paling banyak digunakan dalam platform pembelajaran mesin modern. Hal ini menjadikan Adam pilihan utama dalam berbagai aplikasi machine learning saat ini[38].

2.3.7 Bidirectional Long Short Term Memory (LSTM)

Bidirectional Long Short-Term Memory (BiLSTM) merupakan pengembangan dari model Long Short-Term Memory (LSTM) yang dibuat untuk meningkatkan pemahaman konteks pada data yang berurutan, terutama pada pemrosesan teks. LSTM sendiri merupakan salah satu jenis Recurrent Neural Network (RNN) yang didesain untuk mengatasi masalah kehilangan informasi pada jaringan RNN biasa, sehingga mampu mengingat informasi pada data berurutan yang panjang[39]. Namun, LSTM standar hanya memproses data dalam satu arah, dari awal hingga akhir, sehingga informasi yang muncul setelah suatu titik dalam urutan tidak dapat digunakan untuk prediksi pada titik tersebut[39].

$$ht \rightarrow = \text{LSTMforward}(xt, ht - 1 \rightarrow)$$

$$ht \leftarrow = \text{LSTMbackward}(xt, ht + 1 \leftarrow)$$

Rumus 2. 2 Rumus Bidirectional Long Short Term Memory (LSTM)

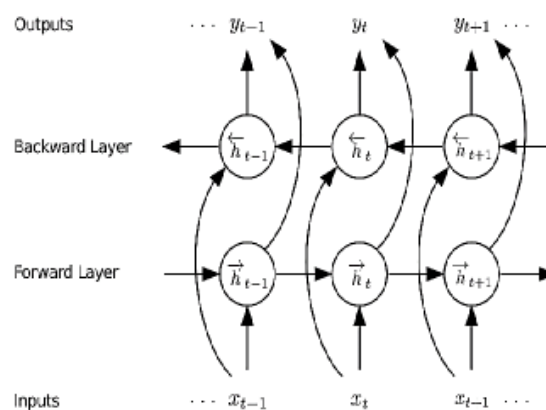
$ht \rightarrow$: Hidden state dari LSTM arah maju pada waktu t

$ht \leftarrow$: Hidden state dari LSTM arah mundur pada waktu t

ht : Output gabungan dari kedua arah, biasanya berupa concatenation

BiLSTM mengatasi keterbatasan ini dengan menggunakan dua lapisan LSTM yang berjalan secara bersamaan, satu lapisan memproses data dari awal hingga akhir (maju), dan lapisan lainnya dari akhir hingga awal (mundur). Dengan cara ini, setiap elemen dalam urutan dapat memperoleh konteks dari arah sebelumnya dan yang akan datang. Pendekatan ini membuat BiLSTM sangat efektif untuk tugas-tugas seperti analisis sentimen, klasifikasi teks, dan deteksi spam karena mampu menangkap informasi yang lebih akurat dan tepat[39].

Pada gambar 2.2 terlihat diagram arsitektur jaringan saraf rekuren bidirectional yang terdiri dari dua lapisan utama, yaitu forward layer dan backward layer. Input data diproses secara simultan dari arah maju (forward) dan mundur (backward), sehingga setiap titik waktu dapat memperoleh informasi dari konteks sebelumnya dan sesudahnya. Node-node yang diberi label 'h' menunjukkan status tersembunyi pada setiap langkah waktu, dengan panah yang menggambarkan aliran informasi antar lapisan dan waktu. Pendekatan ini memungkinkan model untuk menangkap pola yang lebih kaya dan kontekstual dalam data berurutan, sehingga meningkatkan performa pada tugas-tugas seperti pemrosesan bahasa alami[39].

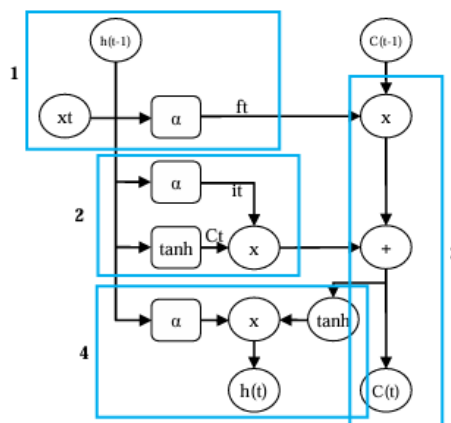


Gambar 2. 2 Struktur BI-LSTM [39]

2.3.8 Long Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) adalah jenis Recurrent Neural Network (RNN) yang dikembangkan untuk mengatasi masalah yang sering dihadapi oleh RNN tradisional, terutama dalam memproses data urutan yang panjang. RNN standar cenderung mengalami masalah seperti vanishing gradient dan exploding gradient, yang menyebabkan mereka kesulitan untuk mengingat informasi dalam urutan yang sangat panjang. LSTM mengatasi masalah ini dengan struktur yang lebih kompleks, memungkinkan model untuk mengingat informasi dalam jangka waktu yang lebih lama, bahkan lebih dari seribu langkah waktu, tergantung pada kompleksitas jaringan dan data yang diproses[40].

Pada gambar 2.3 ini merupakan struktur dari jaringan Long Short-Term Memory (LSTM), yang digunakan untuk memproses data sekuensial[41]. LSTM memiliki tiga komponen utama: input, hidden state, dan cell state. Pada langkah pertama, input x_t dan hidden state sebelumnya $h(t-1)$ diproses melalui gate yang disebut forget gate f_t yang memutuskan informasi mana yang akan dilupakan dari cell state sebelumnya $C(t-1)$. Output dari forget gate kemudian digunakan untuk memperbarui cell state dengan informasi yang relevan[41].



Gambar 2. 3 Prosedur LSTM[40]

Pada langkah kedua, informasi baru yang relevan diproses melalui input gate i_t yang mengontrol informasi mana yang akan ditambahkan ke dalam cell state.

Selain itu, ada juga cell state candidate \tilde{C}_t , yang mewakili informasi baru yang bisa ditambahkan ke dalam memori model. Kedua komponen ini bekerja sama untuk memodifikasi dan memperbarui cell state yang akan digunakan dalam langkah berikutnya[41].

Terakhir, pada langkah ketiga dan keempat, cell state yang baru diperbarui digunakan untuk menghasilkan output[41]. Output gate o_t memutuskan bagian mana dari cell state yang akan digunakan untuk menghasilkan hidden state h_t , yang akhirnya menjadi output model[41]. Hidden state ini membawa informasi yang telah diproses, dan output ini selanjutnya dapat digunakan untuk prediksi atau diteruskan ke langkah waktu berikutnya[41]. Dengan demikian, LSTM mampu memproses urutan data dalam waktu panjang dengan memilih untuk mengingat atau melupakan informasi yang ada[41].

LSTM dilengkapi dengan unit memori yang disebut cell state. Cell state ini bertanggung jawab untuk menyimpan dan membawa informasi sepanjang urutan waktu. Struktur LSTM memiliki tiga jenis gerbang utama yang mengatur bagaimana informasi akan diproses, disimpan, dan dikeluarkan dari jaringan. Gerbang-gerbang ini membantu LSTM untuk memutuskan apa yang harus diingat, apa yang harus dilupakan, dan informasi mana yang perlu disalurkan ke langkah waktu berikutnya[42].

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

Rumus 2. 3 Rumus Long Short Term Memory (LSTM)

Analisis sentimen adalah salah satu aplikasi yang paling umum dari LSTM, karena tugas ini sering kali melibatkan pemahaman konteks panjang dalam sebuah urutan teks. LSTM sangat cocok untuk ini karena dapat menangkap dependensi jangka panjang antara kata-kata dalam kalimat. Model LSTM menerima teks yang telah diproses, diubah menjadi urutan kata atau token, dan biasanya diubah menjadi representasi numerik (seperti vektor kata menggunakan Word2Vec, GloVe, atau embeddings lainnya).

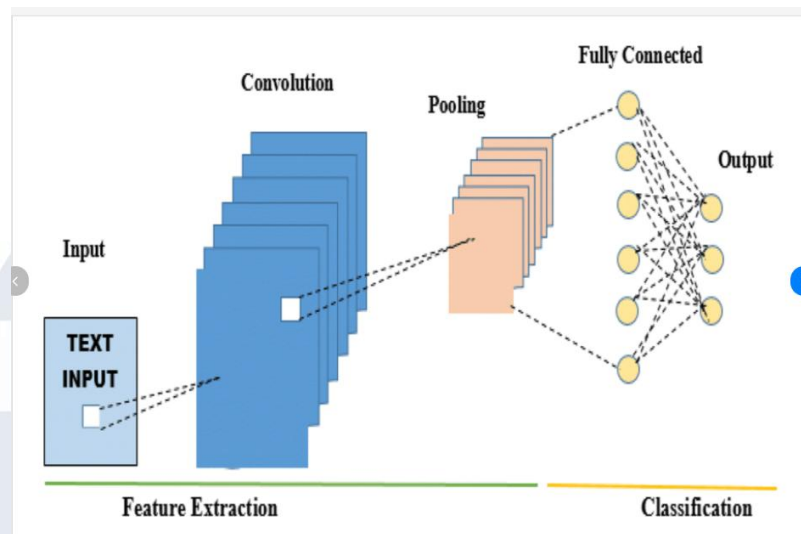
Melalui gerbang-gerbangnya, LSTM menyaring informasi yang relevan untuk menentukan sentimen, di mana kata-kata seperti "kecewa" dan "harapan" akan memiliki dampak signifikan dalam menentukan apakah sentimen secara keseluruhan positif atau negatif, meskipun berada dalam urutan kalimat yang panjang. Setelah memproses urutan kata, LSTM akan mengeluarkan prediksi sentimen (positif, dan negatif,) berdasarkan informasi yang telah diproses, dengan output sering kali berupa kelas sentimen atau skor probabilitas dari sentimen yang diprediksi

2.3.9 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu arsitektur deep learning yang dirancang untuk mengekstrak fitur secara otomatis dari data input yang memiliki dimensi spasial, seperti gambar atau teks[54]. Awalnya dikembangkan untuk pemrosesan gambar digital, CNN telah terbukti sangat efektif dalam memproses data berbasis teks, terutama dalam tugas-tugas seperti klasifikasi dokumen, deteksi emosi, dan analisis sentimen[54]. Salah satu komponen utama CNN adalah lapisan konvolusi, yang menggunakan operasi matematis berupa konvolusi untuk mengekstrak fitur penting dari data input. Secara matematis, output dari lapisan konvolusi pada posisi tertentu dihitung dengan menjumlahkan hasil perkalian antara nilai-nilai input dalam sebuah jendela kecil (kernel atau filter) dengan bobot kernel tersebut, kemudian ditambah dengan bias.

Convolutional Neural Network (CNN) berfungsi dengan mengidentifikasi fitur-fitur lokal dari data masukan, yang dalam konteks teks berupa frasa atau rangkaian kata bermakna. Metode ini sangat efektif dalam menangkap pola-pola penting dalam kalimat yang berperan dalam menentukan sentimen positif, negatif, maupun netral.

Pada gambar 2.5 di bawah ini menggambarkan alur pemrosesan data teks menggunakan arsitektur Convolutional Neural Network (CNN) dalam tugas analisis sentimen[43]. CNN dalam konteks ini digunakan untuk mengekstraksi fitur penting dari teks masukan dan melakukan klasifikasi berdasarkan emosi atau opini yang terkandung di dalamnya[43].



Rumus 2. 4 Prosedur Alur Analisis Sentimen CNN[43].

Secara umum, proses kerja CNN dalam analisis sentimen terbagi menjadi dua tahap utama, yaitu ekstraksi fitur (feature extraction) dan klasifikasi (classification)[43].

1. Input

Proses dimulai dari teks yang telah diubah menjadi representasi vektor—biasanya dalam bentuk embedding seperti Word2Vec, GloVe, atau embedding dari transformer. Representasi ini mengubah setiap kata menjadi angka agar bisa diproses oleh jaringan saraf[43].

2. Convolutional Layer

Di tahap ini, beberapa filter konvolusional diterapkan ke urutan kata untuk mendeteksi pola lokal seperti frasa penting yang mengandung makna sentimen, misalnya "sangat baik", "mengecewakan", atau "tidak disarankan". Setiap filter bertindak seperti jendela kecil yang mengamati kata-kata secara bertahap[43].

3. Pooling Layer

Setelah fitur lokal terdeteksi, hasilnya diproses lebih lanjut melalui pooling, yang berfungsi untuk menyaring informasi paling menonjol dari setiap fitur[43]. Biasanya digunakan max pooling, yang memilih nilai

tertinggi untuk menjaga fitur paling relevan sambil mengurangi dimensi data.

4. Fully Connected Layer

Fitur-fitur penting yang sudah disaring kemudian dikirim ke lapisan saraf penuh (fully connected layer). Di sinilah terjadi penggabungan informasi untuk membuat prediksi sentimen secara menyeluruh[43].

5. Output (Klasifikasi Sentimen)

Akhirnya, CNN menghasilkan output berupa label sentimen, yang bisa berupa dua kelas (positif/negatif) atau tiga kelas (positif, negatif, netral), tergantung pada kebutuhan sistem[43].

2.3.10 Confusion Matrix

Confusion Matrix merupakan alat evaluasi yang sangat penting dalam menilai hasil analisis sentimen, karena untuk mengukur tingkat akurasi prediksi yang dihasilkan oleh algoritma klasifikasi. Alat ini bekerja dengan membandingkan hasil prediksi algoritma dengan data nyata yang ada. Sebagaimana terlihat pada Tabel 2.1, Confusion Matrix menyajikan visualisasi dan ringkasan kinerja algoritma dalam bentuk tabel yang memuat data aktual serta hasil klasifikasi. Dengan menggunakan Confusion Matrix dapat mengevaluasi seberapa efektif algoritma dalam mengklasifikasikan data secara tepat dan akurat [11].

Tabel 2. 2 Confusion Matrix

Confusion Matrix	Actual Data		
	Positive		Negative
Prediction Result	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Melalui tabel Confusion Matrix yang disajikan di atas dapat mengevaluasi sebagai berikut:

- a. **True Positive (TP)** menunjukkan jumlah data yang sebenarnya positif dan berhasil diprediksi sebagai positif oleh model.
- b. **False Positive (FP)** menggambarkan jumlah data yang sebenarnya negatif namun salah diklasifikasikan sebagai positif oleh model.
- c. **False Negative (FN)** mengacu pada jumlah data yang sebenarnya positif tetapi keliru diprediksi sebagai negatif oleh model.
- d. **True Negative (TN)** mencerminkan jumlah data yang sebenarnya negatif dan berhasil diprediksi sebagai negatif oleh model.

Menggunakan keempat data tersebut dapat mengestimasi berbagai nilai yang berguna untuk mengukur kinerja metode yang telah dikembangkan[44]. Dengan memanfaatkan keempat data tersebut, dapat menghitung sejumlah metrik yang bermanfaat untuk mengevaluasi kinerja metode yang telah disusun.

Setelah Confusion Matrix terbentuk, kinerja algoritma klasifikasi dapat dievaluasi melalui sejumlah parameter, termasuk akurasi, presisi, recall, dan skor F1.

1. Accuracy

Nilai Accuracy mencerminkan tingkat keakuratan model yang telah dibentuk dan diukur sebagai perbandingan antara data yang berhasil diklasifikasikan dengan benar ($TP+TN$) terhadap seluruh jumlah data

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

$(TP+TN+FP+FN)$ [45]. Berikut merupakan cara perhitungan akurasi dapat diperjelas dengan berikut:

$$\text{Accuracy} = \frac{TP + TN}{\text{Jumlah Data}}$$

Rumus 2. 5 Rumus Accuracy

2. Precision

Precision merupakan ukuran yang mengevaluasi sejauh mana model mampu mengenali kelas yang diminta dengan benar dalam perbandingan dengan semua hasil prediksi dari kelas tersebut. Dalam konteks ini, precision mencerminkan kemampuan model untuk memprediksi data aktual positif dari semua hasil prediksi yang dinyatakan sebagai kelas positif[45]. Rumus perhitungan precision merupakan sebagai berikut:

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

Rumus 2. 6 Rumus Precision

3. Recall

Recall, yang juga dikenal sebagai "sensitivitas" atau "true positive rate," mengukur sejauh mana model dapat mengenali kelas yang diminta dari seluruh data aktual yang termasuk dalam kelas tersebut. Dalam konteks ini, recall mencerminkan kemampuan model dalam memprediksi data aktual positif dari total keseluruhan data aktual positif dalam dataset[45]. Perhitungan recall dapat dinyatakan dalam rumus berikut:

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

Rumus 2. 7 Rumus Recall

4. F1 score

F1-score, atau F-Measure, merupakan suatu metrik yang memberikan gambaran tentang keseimbangan antara precision dan recall, khususnya ketika terjadi ketidakseimbangan dalam kelas data[45]. Rumus untuk menghitung F1-score merupakan sebagai berikut:

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Rumus 2. 8 Rumus F1 - Score

2.3.11 Uji Friedman

Uji Friedman adalah metode statistik non-parametrik yang digunakan untuk membandingkan tiga atau lebih algoritma atau model pada data yang berulang atau berpasangan. Uji ini sangat berguna ketika data tidak memenuhi asumsi normalitas yang diperlukan dalam ANOVA berulang. Fungsi utamanya adalah menguji hipotesis nol yang menyatakan tidak ada perbedaan signifikan antara median performa dari beberapa kelompok atau perlakuan yang diuji.[46]

Prinsip kerja uji Friedman melibatkan pemberian peringkat terhadap hasil pengujian tiap model secara terpisah untuk setiap sampel data, lalu membandingkan peringkat tersebut untuk melihat konsistensi perbedaan performa. Statistik uji yang dihitung kemudian digunakan untuk memperoleh nilai p-value. Jika p-value kurang dari tingkat signifikansi (seperti 0,05), maka hipotesis nol ditolak, dan diartikan bahwa ada perbedaan performa yang signifikan secara statistik antar model[46].

Uji Friedman biasanya dilakukan dengan mempertimbangkan berbagai parameter evaluasi kinerja model, seperti akurasi, presisi, recall, dan F1-score. Parameter-parameter ini memberikan gambaran menyeluruh tentang performa model dalam berbagai aspek, mulai dari kemampuan klasifikasi yang benar hingga keseimbangan antara presisi dan recall. Dengan menggunakan uji Friedman terhadap metrik-metrik ini, peneliti dapat secara statistik membuktikan apakah ada perbedaan yang signifikan antar model atau algoritma yang diuji. Pendekatan ini sangat membantu dalam memilih model terbaik secara objektif berdasarkan bukti

data yang konsisten dan tidak bergantung pada asumsi distribusi data. Oleh karena itu, uji Friedman menjadi alat penting dalam evaluasi dan validasi performa berbagai model machine learning dalam penelitian[46].

2.4 Tools dan Software Penelitian

2.4.1 Google Colaboratory / Google Colab

Google Colaboratory, atau yang sering disebut sebagai Google Colab, merupakan sebuah platform komputasi berbasis cloud yang menyerupai Jupyter Notebook[47]. Google Colab memberikan kemampuan kepada pengguna untuk menjalankan kode Python serta menjalankan proyek-proyek terkait data science dan pemrograman tanpa harus menghadapi kerumitan dalam konfigurasi atau pengaturan lingkungan pengembangan pada komputer lokal mereka. Dengan Google Colab, pengguna memiliki akses ke lingkungan pengembangan Python yang sangat kuat dan lengkap melalui peramban web yang umumnya digunakan, seperti Google Chrome, Mozilla Firefox, atau Opera. Ini berarti tidak perlu lagi menghabiskan waktu berharga untuk menginstal dan mengonfigurasi perangkat lunak di komputer lokal. Salah satu keunggulan utama dari Google Colab merupakan ketersediaan *library* dan paket Python yang sangat beragam tanpa perlu melakukan instalasi manual[48]. Semua *library* yang sering digunakan dalam ilmu data dan pengembangan Python sudah terinstal secara default[48]. Ini merupakan hal yang sangat berguna bagi pemula yang ingin memulai perjalanan mereka dalam pemrograman Python, sekaligus memberikan kemudahan bagi para profesional yang ingin menjalankan proyek tanpa hambatan teknis[48]. Google Colab memudahkan proses data scraping dengan menyediakan lingkungan pengembangan Python berbasis cloud[49]. Platform ini memberikan akses cepat dan mudah ke berbagai perpustakaan dan alat yang sering digunakan dalam proses web scraping[49]. Dengan sumber daya cloud yang kuat, Google Colab mempermudah pengumpulan, analisis, dan penyimpanan data dari berbagai sumber secara efisien. Oleh karena itu, Google Colab merupakan pilihan yang sangat bijak untuk para pengembang, peneliti, atau siapa pun yang ingin meningkatkan keterampilan pemrograman Python tanpa harus repot dengan konfigurasi perangkat

lunak di komputer lokal mereka, sekaligus dengan akses yang mudah ke sumber daya komputasi yang dibutuhkan.

2.4.2 Python

Python, yang merupakan produk ciptaan Guido van Rossum, merupakan sebuah bahasa pemrograman tingkat tinggi yang menawarkan sejumlah keunggulan penting. Salah satunya merupakan fleksibilitas yang diberikan dalam pemrograman berorientasi objek (OOP), yang memungkinkan pengembang untuk membangun solusi perangkat lunak yang kuat. Python juga dikenal dengan kemampuannya beroperasi secara lintas platform, yang artinya kode yang ditulis dalam Python dapat berjalan di berbagai sistem operasi. Untuk memperluas fungsionalitasnya, Python memiliki lebih dari 100.000 paket yang tersedia dalam perpustakaan Python. Keberagaman paket ini memberikan pengembang alat yang kuat untuk mengembangkan model, aplikasi, dan perangkat lunak dengan beragam fungsi dan fitur. Kombinasi fleksibilitas, lintas platform, dan keberagaman paket membuat Python menjadi pilihan yang sangat kuat dan serbaguna untuk berbagai keperluan pengembangan perangkat lunak[50].

2.4.3 Visual Studio Code (VS Code)

Visual Studio Code (VS Code) adalah sebuah teks editor sumber terbuka (open-source) yang dikembangkan oleh Microsoft untuk membantu pengembang dalam menulis, mengedit, dan menjalankan kode secara efisien. Aplikasi ini ringan, cepat, dan mendukung berbagai bahasa pemrograman seperti Python, JavaScript, C++, dan banyak lagi. Dengan fitur-fitur unggulan seperti syntax highlighting, autocompletion berbasis IntelliSense, integrasi Git bawaan, debugger, serta terminal terintegrasi, VS Code menjadi pilihan populer di kalangan pengembang. Selain itu, aplikasi ini dapat diperluas dengan ribuan ekstensi melalui marketplace-nya untuk menyesuaikan kebutuhan pengembangan proyek, baik itu membangun aplikasi web, API, maupun proyek berbasis data. Dukungan lintas platform (Windows, macOS, dan Linux) menjadikan VS Code fleksibel untuk digunakan oleh pengembang di berbagai lingkungan kerja.

BAB III

METODOLOGI PENELITIAN

3.1. Gambaran Umum Objek Penelitian

3.1.1. Objek Penelitian

Tinder, sebagai aplikasi kencan *daring* terpopuler di Indonesia, ternyata menarik perhatian banyak pengguna, seperti yang terlihat pada data dari databoks.id pada tahun 2020. Dalam survei tersebut, sekitar 57,6% responden mengonfirmasi menggunakan Tinder, menjadikannya aplikasi kencan *daring* yang paling dominan. Tinder diluncurkan pada Juli 2013 untuk pengguna Android, dengan permintaan yang tinggi yang tercermin dalam lebih dari 800.000 unduhan. Pada September 2023, Tinder telah mencapai 100 juta unduhan dengan rating 4,4 di Google Play Store. Analisis sentimen dari ulasan pengguna menjadi relevan mengingat banyaknya ulasan. Metode ini memungkinkan identifikasi polaritas teks, positif atau negatif, sehingga membantu penelitian untuk mengevaluasi kualitas layanan aplikasi kencan *daring* secara efisien dan mendalam dari sudut pandang pengguna.

Penelitian ini memiliki tujuan untuk mendalami tentang sentimen yang ada dalam ulasan pengguna terhadap aplikasi kencan Tinder di Google Play Store. Data yang digunakan dalam penelitian ini berasal dari pengambilan ulasan di Google Play Store dan merupakan dataset yang diperbaharui hingga tahun 2023. Setiap ulasan akan dianalisis untuk memahami bagaimana pengguna memandang, menilai aplikasi serta menjadi pertimbangan masyarakat dalam menggunakan aplikasi Tinder. Selain itu, penelitian ini akan mencoba berbagai teknik klasifikasi untuk menentukan pendekatan yang paling akurat dalam menganalisis sentimen ulasan pengguna terhadap aplikasi Tinder.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2 Metode Penelitian

Penelitian yang ada dalam penulisan ini menggunakan model Knowledge Discovery In Databases (KDD), yang juga dikenal berperan dalam proses KDD. KDD, dengan pendekatan sistematisnya untuk menangani kompleksitas data, memfasilitasi fase prapemrosesan yang menyeluruh. Proses ini mencakup 5 tahap Data Selection, PreProcessing, Transformation, Data mining, Evaluation[51].

Tabel 3. 1 Perbandingan Framework

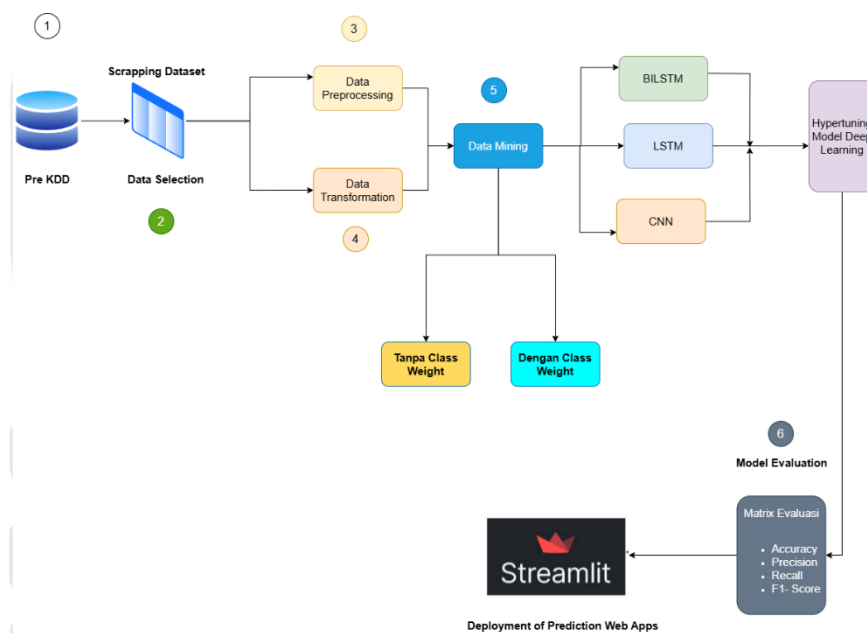
Faktor Pembeda	CRISP-DM	KDD
Tahap Penyelesaian	Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation	Pre KDD, Data Selection, PreProcessing, Transformation, Data mining, Evaluation
Tujuan	Memberikan sebuah solusi terhadap masalah pada sekelompok bisnis tertentu	Memberikan informasi tersembunyi pada pola dan trend yang terdapat pada dataset.
Keterlibatan Dalam Kesesuaian Riset	Dalam analisis sentimen, CRISP-DM melibatkan pemangku kepentingan sejak awal. Tahap Business Understanding digunakan untuk merumuskan tujuan seperti klasifikasi opini pelanggan. Tim produk dan pemasaran berperan dalam setiap tahap proses. Hasil analisis diuji kesesuaiannya dengan kebutuhan bisnis. Pendekatan iteratif memudahkan implementasi dan penyesuaian model.	Analisis sentimen dengan pendekatan KDD cenderung bersifat teknis dan eksploratif. Peneliti biasanya bekerja sendiri dalam mengekstraksi pola dari data. Keterlibatan pemangku kepentingan minim di tahap awal proses. Fokus utamanya adalah menemukan pengetahuan tanpa mempertimbangkan kebutuhan bisnis secara langsung. Pengguna bisnis umumnya baru dilibatkan saat hasil akhir perlu diinterpretasikan.

Faktor Pembeda	CRISP-DM	KDD
Keberlanjutan	Memiliki fase Deployment tersendiri untuk memastikan model terus diperbarui.	KDD terlalu klasik kurang menekankan aspek pemeliharaan setelah penemuan pengetahuan.

Sumber : [[52]]

Berdasarkan Tabel 3.2, metode KDD memiliki tahapan yang efektif dalam melakukan analisis data untuk mengungkap informasi tersembunyi di dalamnya. Oleh karena itu, metode KDD dianggap paling sesuai untuk diaplikasikan dalam analisis data pada penelitian ini. Dengan demikian, metode KDD dipilih sebagai pendekatan utama dalam penelitian yang dilakukan.

Secara keseluruhan, implementasi KDD diharapkan dapat menghasilkan pengetahuan yang mendalam mengenai persepsi pengguna terhadap aplikasi kencan dan mendukung pengembang dalam meningkatkan kualitas layanan berdasarkan ulasan yang sah. Berikut adalah gambaran mengenai langkah-langkah dalam alur penelitian KDD pada gambar 3.1[51].



Gambar 3. 1 Alur Penelitian

3.2.1 Pre KDD (Data Collection)

Tahap awal dari penelitian ini melibatkan pengumpulan data dari ulasan aplikasi Tinder di Google Play Store. Data tersebut diambil melalui proses pengambilan data dari platform Google Play Store dengan melakukan crawling data menggunakan google playstore web scrapper untuk kebutuhan penelitian.

3.2.2 Data Selection

Hasil pengumpulan data kemudian akan disaring berdasarkan klasifikasi sentimen yang akan dibagi menjadi dua kelas, yaitu sentimen positif dan sentimen negatif[53]. Berikut merupakan tabel detail isi dari dataset yang akan digunakan untuk penelitian :

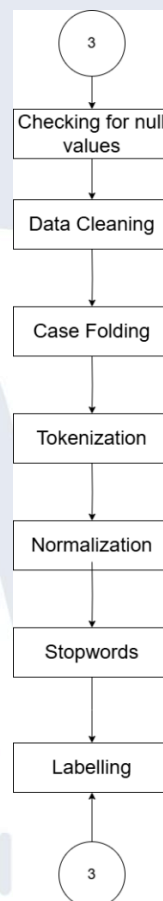
Tabel 3. 2 Variabel Dataset

Variable	Jenis Variable	Keterangan
User name	Categorical variable	Variabel ini mewakili nama pengguna yang memberikan ulasan pada aplikasi Tinder di Google Play Store.
Score	Numerical variable	Variabel ini mewakili peringkat atau score yang diberikan pada aplikasi Tinder di Google Play Store.
At	Numerical variable	Variabel ini mewakili keterangan tanggal, bulan, tahun dan, jam user atau pengguna memberi ulasan mengenai aplikasi Tinder di Google Play Store
Content	Categorical variable	Variabel ini mewakili isi komentar atau ulasan

		pengguna mengenai aplikasi Tinder di Google Play Store
--	--	---

3.2.3 Data Preprocessing

Sebelum proses data mining dimulai, dilakukan preprocessing sangat penting sebelum data mining karena memastikan data bersih, konsisten, dan siap untuk dianalisis. Data mentah sering mengandung kesalahan, duplikasi, atau nilai yang hilang yang dapat menurunkan akurasi model. Preprocessing juga membantu menyamakan format data dan mengurangi kompleksitas melalui seleksi fitur. Dengan data yang sudah diproses, hasil data mining menjadi lebih akurat dan bermakna. Tanpa preprocessing, proses analisis bisa menghasilkan informasi yang salah atau menyesatkan sesuai pada alur yang ada pada gambar 3.3 .[53].



Gambar 3. 2 Alur Preprocessing

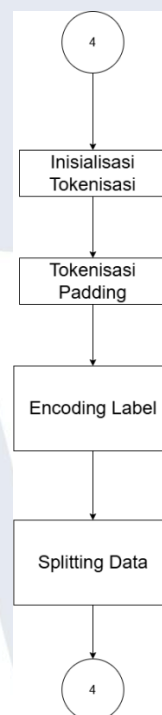
1. Checking For Nulls Values, merupakan proses pembersihan data agar model machine learning tidak terganggu oleh data yang hilang. Dengan mengetahui kolom yang memiliki nilai null, pengguna dapat mengambil tindakan seperti imputasi atau penghapusan data.
2. Data Cleaning merupakan Proses pembersihan teks ulasan dimulai dengan mengubah seluruh huruf menjadi huruf kecil untuk menjaga konsistensi analisis. Selanjutnya, berbagai elemen tidak relevan seperti tagar, mention, angka, emoji, tanda baca, serta karakter newline dan carriage return dihapus atau diganti agar teks menjadi satu baris dan lebih bersih. Dengan pembersihan ini, data teks siap untuk analisis lanjutan seperti tokenisasi atau klasifikasi sentimen[54].
3. Case Folding, adalah sebuah proses mengubah seluruh huruf dalam teks menjadi huruf kecil penting untuk menjaga konsistensi dan menghindari perbedaan penulisan kata yang sama. Langkah ini mencegah variasi akibat kapitalisasi yang dapat mengganggu analisis. Dengan demikian, proses analisis teks menjadi lebih akurat[54].
4. Tokenization, adalah proses tokenisasi membagi teks menjadi kata-kata individu menggunakan metode `word_tokenize` dari NLTK. Hal ini memudahkan analisis dengan memisahkan teks menjadi unit yang lebih kecil dan mudah dikelola. Hasil tokenisasi disimpan dalam kolom baru untuk digunakan dalam analisis sentimen atau pemrosesan lanjutan.
5. Normalization, adalah proses memuat kamus slang dari file CSV 'kamusalay.csv' dan mengonversinya menjadi dictionary Python digunakan untuk mengganti kata slang dalam kalimat menjadi kata normal agar analisis teks lebih akurat dan formal. Selanjutnya, dilakukan pengecekan statistik frekuensi kata pada kolom 'slang' di DataFrame untuk mengetahui jumlah kata unik yang akan digunakan dalam tahap penentuan stopwords.

6. Stopword removal, adalah proses menyaring kata-kata yang tidak relevan dalam proses preprocessing teks agar analisis menjadi lebih akurat dan fokus pada kata-kata bermakna[55].

7. Labeling Sentimen, adalah melakukan proses daftar kata positif, dan negatif. dalam bahasa Indonesia digunakan sebagai referensi dalam analisis sentimen berbasis leksikon tanpa perlu pelatihan model.

3.2.4 Data Transformation

Proses transformasi dilakukan untuk merepresentasikan data sesuai dengan kebutuhan dalam proses data mining yang akan dilaksanakan. Data yang sudah dipilih dan dibersihkan akan diubah agar sesuai dengan persiapan untuk proses analisis sentimen[53].



Gambar 3. 3 Alur Transformation

1. Inisialisasi Tokenisasi, merupakan proses membuat objek tokenisasi dengan menyesuaikan jumlah data sebanyak 5000. Proses ini penting agar teks dapat diubah menjadi urutan angka yang bisa dipahami oleh model machine learning.

2. Tokenisasi Padding merupakan proses lanjutan dari sebelumnya untuk menyesuaikan setiap kata dalam teks diganti dengan indeks numerik sesuai kamus yang dibuat oleh tokenizer. Selanjutnya, kode melakukan padding pada urutan angka tersebut agar semua urutan memiliki panjang yang sama, yaitu 200 token. Padding ini penting agar data input seragam dan bisa diproses oleh model deep learning.

3. Encoding Label, adalah sebuah proses untuk mengubah label sentimen dari bentuk teks menjadi bentuk numerik. Proses ini dilakukan dengan mengonversi setiap label teks menjadi angka yang sesuai. Setelah itu, label numerik tersebut diubah lagi menjadi format one-hot encoding menggunakan fungsi `to_categorical()`, sehingga setiap label direpresentasikan dalam bentuk vektor biner yang cocok untuk klasifikasi multi-kelas.

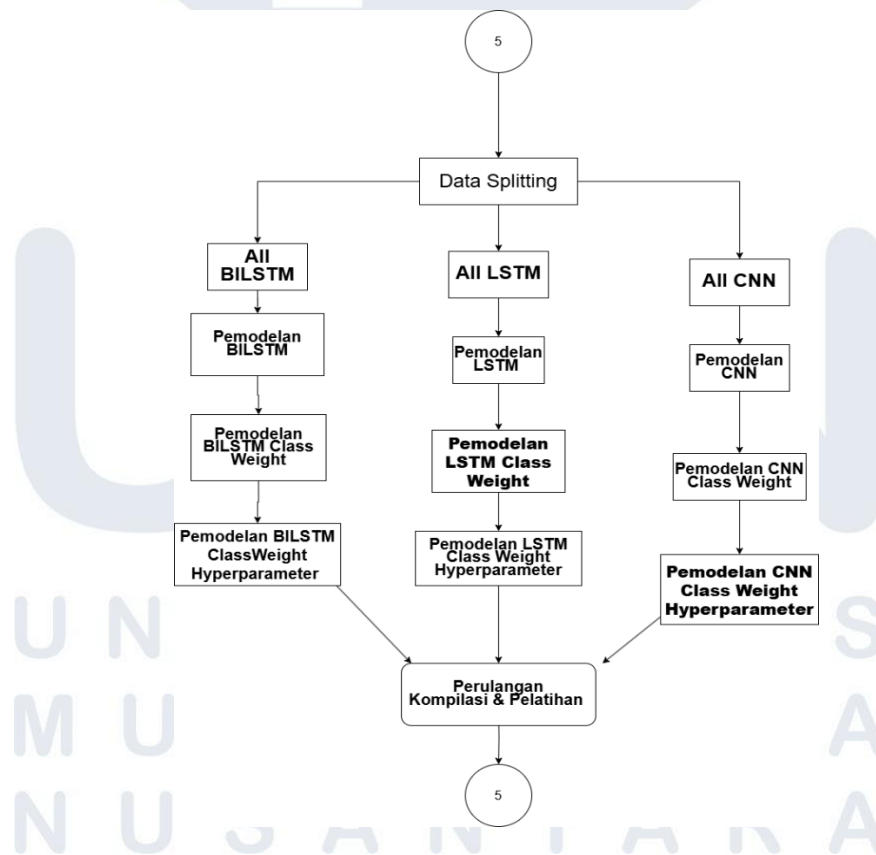
4. Splitting Data, adalah proses melakukan pembagian data menjadi set pelatihan dan pengujian. Pada tahap ini mengimpor fungsi dari modul yang relevan, lalu menerapkannya dengan parameter fitur, label, ukuran data uji, dan `random_state` untuk memastikan hasil yang konsisten. Proses ini merupakan langkah penting dalam persiapan data untuk melatih dan menguji model machine learning secara efektif.

3.2.5 Data Mining

Pada tahap ini, penerapan teknik data mining difokuskan pada pembangunan model analisis sentimen yang telah melalui proses transformasi data. Langkah-langkah yang dilakukan meliputi pemilihan dataset yang relevan dan persiapan data untuk memastikan kualitas yang optimal. Dalam penelitian ini, metode yang digunakan untuk menganalisis sentimen adalah model deep learning, yaitu Bidirectional Long Short-Term Memory (BiLSTM), Long Short-Term Memory (LSTM), dan Convolutional Neural Network (CNN). Penelitian ini menggunakan class weights untuk mengatasi ketidakseimbangan kelas dalam data, sehingga model dapat lebih fokus pada kelas yang kurang terwakili selama proses pelatihan. Model-model tersebut dilatih untuk mengenali pola dan tren dari ulasan

pengguna secara efektif. Pengujian dilakukan menggunakan data yang belum pernah dipakai sebelumnya guna memastikan keandalan dan akurasi model dalam mengklasifikasikan ulasan ke dalam kategori sentimen positif dan negatif berdasarkan pola yang ditemukan. Selain itu, penelitian ini juga membandingkan kinerja ketiga metode tersebut untuk menentukan pendekatan yang paling optimal dalam analisis sentimen[53]. Penelitian ini juga akan melakukan perbandingan kinerja antara kedua metode tersebut[53].

Berikut merupakan alur untuk data mining pada gambar 3.4 yang menggambarkan alur untuk klasifikasi sentimen menggunakan tiga arsitektur—BiLSTM, LSTM, dan CNN—yang dimulai dari proses data splitting, dilanjutkan pemodelan dasar tiap arsitektur, penyesuaian class weights, hingga tahap hyperparameter tuning; seluruh model kemudian masuk ke siklus kompilasi-pelatihan terulang untuk memperoleh performa optimal.



Gambar 3. 4 Alur Data Mining

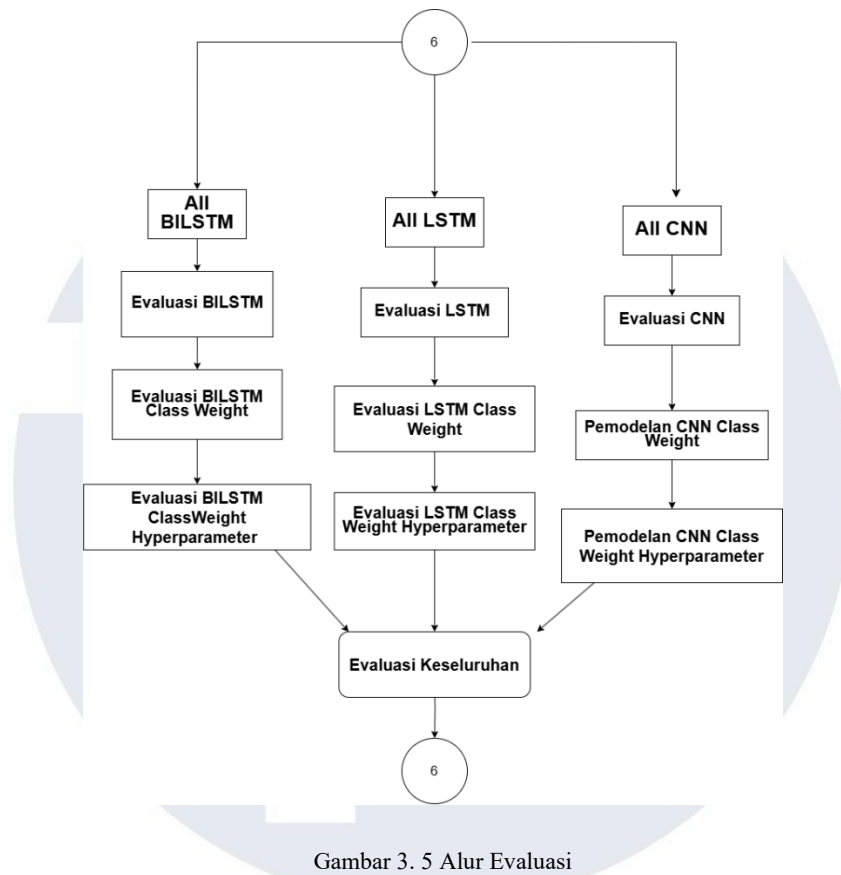
1. Membagi data (train-test splitting) merupakan langkah pertama membagi data ke dalam dua bagian, yaitu data latih (train) dan data uji (test), agar proses pelatihan dan pengujian model bisa berjalan dengan adil dan objektif.
2. Membangun model dengan algoritma BiLSTM setelah data dibagi, model pertama yang digunakan adalah BiLSTM. Pada tahap awal ini, model dibangun tanpa penyesuaian apapun, untuk melihat performa dasarnya.
3. Menyesuaikan BiLSTM dengan class weight Model BiLSTM kemudian ditingkatkan dengan menambahkan class weight, yaitu penyesuaian bobot antar kelas, agar model bisa lebih adil dalam mengenali semua jenis sentimen, termasuk yang jumlah datanya lebih sedikit.
4. Meningkatkan BiLSTM dengan tuning parameter Setelah menggunakan class weight, model BiLSTM dioptimalkan lagi dengan menyetel parameter-parameter penting (hyperparameter) agar hasil prediksi makin akurat dan stabil.
5. Membangun model dengan algoritma LSTM Proses yang sama dilakukan dengan model LSTM. Model ini dibangun dari awal untuk melihat performanya dalam menganalisis sentimen.
6. Menyesuaikan LSTM dengan class weight LSTM kemudian diberi penyesuaian bobot antar kelas, agar bisa lebih sensitif terhadap data yang tidak seimbang.
7. Meningkatkan LSTM dengan tuning parameter Model LSTM juga dioptimalkan melalui pengaturan parameter untuk mendapatkan hasil klasifikasi yang lebih maksimal.
8. Membangun model dengan algoritma CNN Selanjutnya digunakan model CNN. Model ini dibangun dan diuji untuk mengetahui kemampuannya dalam mengenali pola-pola sentimen dalam data teks.

9. Menyesuaikan CNN dengan class weight Agar hasilnya lebih seimbang, CNN juga dilengkapi dengan class weight, seperti halnya model sebelumnya.
10. Meningkatkan CNN dengan tuning parameter Terakhir, model CNN disempurnakan dengan menyetel parameter-parameter tertentu untuk mencapai performa terbaik.
11. Melakukan pelatihan dan evaluasi berulang Semua model yang telah dikembangkan kemudian dilatih dan dievaluasi secara berulang, agar hasil akhirnya benar-benar optimal dan bisa diandalkan dalam menganalisis sentimen pengguna.

3.2.6 Evaluation

Tahap evaluasi model yang dilakukan setelah model analisis sentimen berhasil terbentuk dalam proses data mining. Tujuan tahap evaluasi ini adalah untuk mengukur performa model. Evaluasi model didasarkan pada penggunaan metode Bidirectional Long Short-Term Memory (BiLSTM), Long Short-Term Memory (LSTM), dan Convolutional Neural Network (CNN), yang menghasilkan nilai confusion matrix untuk menilai akurasi, F1-score, recall, dan presisi[53].





Gambar 3. 5 Alur Evaluasi

1. Evaluasi Model BiLSTM melakukan evaluasi awal terhadap model BiLSTM yang telah dibangun sebelumnya, tanpa penerapan class weight maupun tuning hyperparameter, untuk mengetahui performa dasarnya dalam klasifikasi sentimen.
2. Evaluasi Model BiLSTM dengan Class Weight menguji performa model BiLSTM setelah diberikan penyesuaian bobot antar kelas (class weight) agar mampu menangani ketidakseimbangan data antar kelas sentimen.
3. Evaluasi Model BiLSTM dengan Class Weight dan Hyperparameter Tuning melanjutkan evaluasi BiLSTM dengan tambahan optimasi hyperparameter untuk mendapatkan hasil yang lebih maksimal dan stabil dalam memprediksi sentimen.
4. Evaluasi Model LSTM melakukan evaluasi terhadap model LSTM dasar tanpa penyesuaian class weight atau tuning hyperparameter untuk mendapatkan baseline performa.

5. Evaluasi Model LSTM dengan Class Weight memberikan penyesuaian bobot pada model LSTM dan mengevaluasinya guna melihat peningkatan performa ketika menghadapi distribusi kelas yang tidak seimbang.
6. Evaluasi Model LSTM dengan Class Weight dan Hyperparameter Tuning mengoptimalkan performa LSTM melalui kombinasi class weight dan penyetelan hyperparameter, lalu dilakukan evaluasi terhadap hasil klasifikasinya.
7. Evaluasi Model CNN model CNN dievaluasi pertama kali dalam bentuk dasar, tanpa penyesuaian khusus, untuk mengetahui performa awalnya dalam klasifikasi sentimen.
8. Evaluasi Model CNN dengan Class Weight selanjutnya, CNN diberi bobot antar kelas (class weight) dan dievaluasi kembali untuk melihat kemampuannya menangani ketimpangan kelas sentimen.
9. Evaluasi Model CNN dengan Class Weight dan Hyperparameter Tuning model CNN ditingkatkan melalui penyetelan hyperparameter dan penggunaan class weight, lalu dievaluasi guna mengukur peningkatan performanya.
10. Evaluasi Keseluruhan merupakan langkah terakhir melakukan evaluasi menyeluruh terhadap semua model (BiLSTM, LSTM, CNN) berdasarkan uji statistik Friedman terhadap nilai validasi loss untuk menentukan seberapa signifikan nilai model dalam klasifikasi sentimen.

3.2.7 Implementasi Aplikasi

Dalam tahap pengembangan sistem analisis sentimen ini, model machine learning yang telah dikembangkan diterapkan ke dalam platform aplikasi web yang dibangun dengan memanfaatkan Streamlit sebagai framework berbasis Python. Aplikasi ini dirancang untuk secara otomatis mendeteksi sentimen pengguna terhadap layanan Tinder dengan mengandalkan algoritma terbaik yang telah dipilih melalui evaluasi kinerja sebelumnya. Proses analisis dilakukan secara real-time sehingga pengguna dapat memperoleh hasil yang cepat dan akurat. Fitur-fitur

seperti pembersihan data, pemilihan model, analisis sentimen, hingga visualisasi hasil disediakan secara lengkap untuk membantu pengguna memahami data secara komprehensif. Dengan demikian, aplikasi ini menjadi alat yang efektif dalam mendukung pengambilan keputusan berbasis analisis sentimen dari data teks.

3.3 Teknik Pengumpulan Data

Penelitian ini memanfaatkan metode pengumpulan data menggunakan google-play-scraper, sebuah platform web scraper yang fokus pada penghasilan ulasan aplikasi di Google Play Store[53]. Rentang waktu pengumpulan data mencakup lima tahun terakhir, dimulai dari 22 Februari 2020. Dalam melakukan crawling data, penelitian ini menggunakan *library* dari Node.js yang memungkinkan pengembang atau peneliti mengekstrak data aplikasi Android dari Google Play Store. Fungsi utama *library* ini adalah mengambil informasi detail aplikasi, ulasan pengguna, peringkat, dan data lain secara otomatis. *Library* tersebut menjadi elemen kunci dalam mengumpulkan ulasan dari Google Play Store dengan skala besar. Kemampuannya untuk disesuaikan dengan kategori data terkini, data yang lebih lama, dan kemampuan untuk menyesuaikan data berdasarkan negara yang ingin dikumpulkan. Pengambilan data ulasan aplikasi kencan Tinder ini sebanyak 5.000 data.

	A	B	C	D
	Username	User ID	Comment	Contact
1	Doni Nugraha	1	29/08/2023 15:33 Buruk	
2	Andre Andre	5	29/08/2023 14:53 bagus	
3	Rifaldi Is	5	29/08/2023 14:57 bagus	
4	Rizwan Muzaffar	5	29/08/2023 10:20 aplikasi ini jodoh yg pas	
5	reza asan	5	29/08/2023 07:08 bagus	
6	nathanach christian	1	29/08/2023 00:37 aplikasi BUKAN	
7	Nico Zai	5	28/08/2023 22:07 bagus	
8	Ne Uiy	1	28/08/2023 20:51 Oh ada ngapain? aku aku di cikal dasar tinder glak benaran yodak	
9	Ridens35 jave	1	28/08/2023 12:39 Sepertinya banyak yg kecewa sama aplikasi ini, cuma tujuan nyedot duit doang	
10	gha syahrta	5	28/08/2023 07:03 bagus	
11	Hermanito	5	28/08/2023 05:41 bagus dan selalu lancar trm kasih smoga lebih baik lagi dalam segala hal	
12	Luiz Indra Kusuma	1	27/08/2023 20:10 aplikasi buat bantu cari cewe yang mah single	
13	Its me Story	1	27/08/2023 19:57 Aneh, aku ga dibelun, foto asli, verifikasi susah. Mau ga klo di belun akunya mending di hapus. Nunggu pratinjau beberapa	
14	Musa Andika	5	27/08/2023 17:51 top	
15	Andy Yanto	4	27/08/2023 17:48 apk sangat bagus	
16	Mamat Monchell 96	1	27/08/2023 16:44 Aplikasi bodok verifikasi di bikin susah emang ga robot tolak pakai verifikasi kalau ini manusia bodoh	
17	Soni Kattiah	1	27/08/2023 15:46 kenapa aku saya di block? Padahal saya baru download dan daftar akun beberapa minggu yg lalu. Hari ini saya buka lagi tiba t	
18	Aisun Midjid	1	27/08/2023 14:59 Akun ku blok di block ya padahal cuma masuk 1 hari	
19	Dikan Irsan	1	27/08/2023 14:33 Apl apaan sih, orang baru bikin akun baru, belum ngapa ngapain udah diblang melanggar, dan tidak bisa membuat akun baru lag	
20	Muhammadhuan Boski	1	27/08/2023 13:14 1st pokonya mah	
21	Indenipussy	1	27/08/2023 13:03 sekarang ini juga di block gak kayak dulu dulu bebas, mana susah buat ngverifikasi lagi :)	
22	Panandi Andika	3	27/08/2023 11:07 premiumnya mahal banget	
23	aj an	1	27/08/2023 10:19 Sudah top-up dana untuk langganan tinder 1 bulan 122.000 tapi pas mau pembayaran ada keterangan saldo anda tidak cukup, t	
24	Yulian	5	27/08/2023 08:23 baik	
25	Yusuf Indra Pratama	5	27/08/2023 07:30 Good	
26	Virginia Mawitry Putri	1	27/08/2023 06:53 Udah pernah dapat cewe disini, kebanyakan ngabisan sesuatu yg ga beres (gombalan basi) rata-rata pemain cewe dan suka mi	
27	Roy Mita	5	27/08/2023 04:20 aplikasi ini lumayan bagus semoga banyak yang suka dengan aplikasi ini	
28	Utara Almad	1	27/08/2023 00:09 aku ini bener dulu ya?	

Gambar 3. 6 Raw dataset

3.4 Teknik Analisis Data

Pada analisis ini, tujuan utamanya adalah menilai sentimen para pengguna aplikasi kencan Tinder berdasarkan seluruh ulasan yang tercatat di kolom komentar Google Play Store. Di ranah deep learning sendiri, tiga arsitektur yang kerap diadu

untuk tugas serupa ialah Bidirectional LSTM (Bi-LSTM), LSTM satu arah, dan Convolutional Neural Network (CNN) untuk teks. Tabel 3.2 berikut merangkum perbandingan ketiganya—mulai dari cara membaca kalimat hingga kebutuhan komputasi—agar memudahkan dalam memilih model paling tepat untuk memetakan nada positif, negatif, maupun netral dalam curhatan pengguna Tinder.

Meskipun model berbasis transformer seperti BERT telah terbukti unggul dalam berbagai tugas NLP, dalam studi ini memilih untuk fokus pada model yang lebih ringan dan dapat dilatih dari nol menggunakan sumber daya komputasi terbatas. Dengan menggunakan LSTM, BiLSTM, dan CNN sebagai baseline, kami juga dapat mengevaluasi secara lebih transparan kemampuan pemrosesan konteks teks tanpa bergantung pada representasi yang telah dipra-latih.

Tabel 3. 3 Perbandingan Bi-LSTM, LSTM dan CNN

Aspek Penilaian	Bi-LSTM	LSTM	CNN
Mekanisme Pemrosesan Teks	Menggunakan pendekatan baca maju dan mundur (bidirectional) sehingga mampu menangkap konteks penuh tanpa kehilangan informasi kontekstual.	Memproses teks secara unidirectional yang memungkinkan pemahaman pada konteks teks secara linier, namun berpotensi melewatkan informasi penting di akhir teks.	Mengandalkan pemindaian fragmen-fragmen penting (n-gram) dalam teks, mirip metode pembacaan cepat yang kurang memperhatikan urutan kata secara utuh.

Aspek Penilaian	Bi-LSTM	LSTM	CNN
Sensitivitas terhadap Negasi dan Sarkasme	Memiliki keunggulan dalam mengenali kalimat negasi dan sarkasme secara akurat berkat konteks dua arah yang komprehensif.	Memiliki sensitivitas yang memadai terhadap kata-kata negasi, meskipun kurang mendalam dalam menangkap sarkasme dibanding model Bi-LSTM.	Rentan terhadap kesalahan interpretasi apabila makna sarkasme sangat bergantung pada urutan kata dan konteks semantik.
Efisiensi Komputasi	Memerlukan sumber daya komputasi yang besar, termasuk data latih yang banyak dan perangkat keras GPU berkapasitas tinggi, sehingga kompleksitasnya tinggi.	Memiliki tingkat efisiensi menengah dengan kebutuhan komputasi yang moderat, cukup optimal digunakan pada perangkat hardware standar.	Paling efisien secara komputasi dan memori, dapat dilatih dengan cepat pada mesin standar dan cocok untuk aplikasi berbasis sistem waktu nyata.
Kecepatan Pelatihan dan Inferensi	Proses pelatihan dan prediksi relatif lambat karena harus melakukan komputasi dua arah secara simultan, yakni maju dan mundur.	Kecepatan pelatihan dan inferensi berada pada level sedang, lebih cepat dibanding Bi-LSTM namun lebih lambat dibanding CNN.	Memiliki kecepatan pelatihan dan inferensi tertinggi, sangat ideal untuk aplikasi dengan kebutuhan respons waktu nyata yang cepat.
Kompleksitas Model (Ukuran File)	Memiliki ukuran model paling besar karena kompleksitas arsitektur yang	Ukuran model berada pada kategori sedang, relatif seimbang antara	Memiliki ukuran file model terkecil, memberikan efisiensi penyimpanan

Aspek Penilaian	Bi-LSTM	LSTM	CNN
	mendukung pemrosesan dua arah dan lapisan berlapis.	kompleksitas dan performa.	yang optimal serta memudahkan implementasi pada lingkungan dengan keterbatasan sumber daya.

Sumber : [43]

Bi-LSTM, LSTM, dan CNN masing-masing memiliki keunggulan dan keterbatasan dalam analisis sentimen ulasan Tinder: Bi-LSTM unggul dalam akurasi dan pemahaman konteks dua arah, cocok untuk analisis mendalam dengan sumber daya komputasi tinggi; LSTM menawarkan keseimbangan antara akurasi dan efisiensi dengan pembacaan satu arah, ideal untuk produksi dengan sumber daya sedang; sedangkan CNN paling cepat dan ringan, cocok untuk analitik real-time pada server berdaya terbatas, meski kurang sensitif terhadap konteks panjang seperti sarkasme atau negasi kompleks[43].



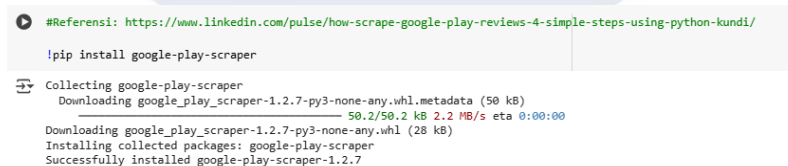
BAB IV

ANALISIS DAN HASIL PENELITIAN

4.1 Pre KDD (Data Collection)

Pada tahap awal, data ulasan diambil dengan teknik web scraping dari laman Google Play Store, dengan rentang waktu pengambilan data dari 22 Februari 2025. Scraping menghasilkan dataset sebanyak 5.000 ulasan yang berupa file `tinder_reviews_5000.csv`. Data ini berisi ulasan pengguna terhadap aplikasi Tinder yang akan dianalisis lebih lanjut.

Pada gambar 4.1, perintah pertama yang dijalankan adalah untuk menginstal pustaka `google-play-scraper`. Perintah `!pip install google-play-scraper` ini digunakan untuk mengunduh dan menginstal pustaka yang diperlukan untuk mengambil data ulasan aplikasi dari Google Play Store. Pada bagian output, terlihat bahwa pustaka berhasil diunduh dan dipasang dengan sukses, yang ditandai dengan pesan "Successfully installed google-play-scraper-1.2.7."



```
#Referensi: https://www.linkedin.com/pulse/how-scrape-google-play-reviews-4-simple-steps-using-python-kundi/

!pip install google-play-scraper

Collecting google-play-scraper
  Downloading google_play_scraper-1.2.7-py3-none-any.whl.metadata (50 kB)
    50.2/50.2 kB 2.2 MB/s eta 0:00:00
  Downloading google_play_scraper-1.2.7-py3-none-any.whl (28 kB)
Installing collected packages: google-play-scraper
Successfully installed google-play-scraper-1.2.7
```

Gambar 4. 1 Tahap pengumpulan data

Pada gambar 4.2, dapat dilihat kode yang digunakan untuk mengambil data ulasan dari aplikasi Tinder di Google Play Store. Fungsi `reviews` digunakan untuk mengambil ulasan berdasarkan parameter yang ditentukan. Di sini, aplikasi yang dituju adalah Tinder dengan ID `com.tinder`, dan parameter lainnya mengatur bahasa ('id' untuk Bahasa Indonesia) dan negara ('id' untuk Indonesia). Selain itu, diatur pula untuk mengurutkan ulasan berdasarkan relevansi (`Sort.MOST_RELEVANT`) dan mengambil sebanyak 5000 ulasan (`count=5000`). Hasil dari pengambilan ulasan ini disimpan dalam variabel `result`.

```

from google_play_scraper import Sort, reviews

# Memperbarui untuk mengambil ulasan dari aplikasi tinder di Google Play Store
result, continuation_token = reviews(
    'com.tinder', # ID aplikasi tanpa parameter 'hl=id' di akhir
    lang='id', # Mengatur bahasa ke Bahasa Indonesia
    country='id', # Mengatur negara ke Indonesia
    sort=Sort.MOST_RELEVANT, # Urutkan berdasarkan relevansi
    count=5000, # Mengambil 1000 ulasan
    filter_score_with=None # Mengambil semua ulasan tanpa filter berdasarkan skor
)

# Mengonversi hasil ke dalam DataFrame untuk memudahkan manipulasi data
import pandas as pd
import numpy as np

```

Gambar 4. 2 Tahap mengambil data ulasan

Pada gambar 4.3, data ulasan yang diambil (result) dikonversi menjadi format DataFrame menggunakan kode `df_tinder = pd.DataFrame(np.array(result), columns=['review'])`. DataFrame ini memungkinkan peneliti untuk lebih mudah mengelola dan menganalisis data ulasan. Data yang awalnya berbentuk array diubah menjadi format tabular yang lebih terstruktur. Kemudian, kolom ulasan yang diambil dimasukkan dengan nama review. Setelah itu, untuk memastikan data yang diambil sesuai, perlunya menggunakan `df_tinder.head()` untuk menampilkan 5 baris pertama dari DataFrame yang berisi ulasan tersebut.

```

df_tinder = pd.DataFrame(np.array(result), columns=['review'])
df_tinder = df_tinder.join(pd.DataFrame(df_tinder.pop('review').tolist()))

# Menampilkan 5 baris pertama dari DataFrame
df_tinder.head()

```

Gambar 4. 3 Tahap hasil pengambilan data ulasan

Pada gambar 4.4, terlihat langkah untuk menghitung jumlah data yang telah diambil. Fungsi `len(df_busu.index)` digunakan untuk menghitung jumlah baris atau ulasan yang ada di dalam DataFrame `df_busu`, yang dihasilkan sebanyak 5000 ulasan. Selanjutnya, bagian kode ini memilih kolom-kolom yang relevan seperti `userName`, `score`, `at` (tanggal), dan `content` (isi ulasan) untuk dianalisis lebih lanjut. Data ini kemudian diurutkan berdasarkan kolom `at` (tanggal ulasan) dengan menggunakan fungsi `sort_values(by='at', ascending=False)`, yang mengurutkan ulasan berdasarkan tanggal secara menurun, sehingga ulasan terbaru muncul di atas.

```
[ ] len(df_busu.index) #count the number of data we got
5000

df_busu[['userName', 'score', 'at', 'content']].head() #preview userName, rating, date-time, and reviews only
```

	userName	score	at	content
0	Pengguna Google	1	2025-02-16 02:37:40	Bang saya sudah beli langganan gold dan platin...
1	Pengguna Google	3	2025-02-16 05:00:59	Bagus si, banyak orang yg pakai ini, tapi smua...
2	Pengguna Google	1	2025-02-02 03:38:00	Aplikasi tinder parah bener, masa uda top up s...
3	Pengguna Google	5	2024-12-28 06:14:44	kelebihannya kita dicarikan pasangan yang baik...
4	Pengguna Google	1	2025-01-10 03:55:33	Saya sudah melakukan pembayaran untuk berlang...

Gambar 4. 4 Tahap menghitung jumlah data

Pada gambar 4.5, setelah data diurutkan berdasarkan tanggal, hasilnya disimpan dalam DataFrame baru yang disebut my_df. Kolom yang ditampilkan adalah userName, score, at, dan content untuk memudahkan analisis lebih lanjut. Data yang sudah diurutkan ini kemudian disimpan dalam file CSV menggunakan perintah `df_tinder.to_csv("tinder_reviews_5000.csv", index=False)`. File CSV yang disimpan ini berisi ulasan yang sudah difilter dan diurutkan sesuai dengan urutan yang diinginkan, siap digunakan untuk analisis atau penyimpanan lebih lanjut.

```
[ ] #Run This Code to Sort the Data By Date
new_df = df_busu[['userName', 'score', 'at', 'content']]
sorted_df = new_df.sort_values(by='at', ascending=False) #sort by Newest, change to True if you want to sort by oldest.
sorted_df.head()
```

	userName	score	at	content
691	Pengguna Google	1	2025-02-18 00:30:29	Min ko saya belum bisa juga masuk ke akun saya...
269	Pengguna Google	2	2025-02-17 13:47:03	Tinder nga bisa pake akses internet wifi yaaa
8	Pengguna Google	1	2025-02-17 13:06:26	Tidak dapat berkorim pesan dan melihat siapa y...
41	Pengguna Google	4	2025-02-16 22:00:20	Mau nge chat aja harus bayar mahal, tolong jan...
274	Pengguna Google	4	2025-02-16 14:37:48	baru coba semoga bagus

```
my_df = sorted_df[['userName', 'score', 'at', 'content']] #get userName, rating, date-time, and reviews only

[ ] my_df.head()
```

	userName	score	at	content
691	Pengguna Google	1	2025-02-18 00:30:29	Min ko saya belum bisa juga masuk ke akun saya...
269	Pengguna Google	2	2025-02-17 13:47:03	Tinder nga bisa pake akses internet wifi yaaa
8	Pengguna Google	1	2025-02-17 13:06:26	Tidak dapat berkorim pesan dan melihat siapa y...
41	Pengguna Google	4	2025-02-16 22:00:20	Mau nge chat aja harus bayar mahal, tolong jan...
274	Pengguna Google	4	2025-02-16 14:37:48	baru coba semoga bagus

```
[ ] # Menyimpan hasilnya dalam file CSV
df_tinder.to_csv("tinder_reviews_5000.csv", index=False) # Menyimpan file dengan nama yang sesuai dengan bulan dan tahun saat ini
```

Gambar 4. 5 Tahap memberi urut data sesuai fitur

Pada gambar 4.6, merupakan hasil dari file CSV yang disimpan berdasarkan filter sesuai dengan urutan yang dibutuhkan untuk membantu penelitian. data yang diperoleh melalui proses web scraping dari aplikasi kencan Tinder yang ada di Google Play Store. Data ini berisi berbagai elemen penting yang diambil dari ulasan pengguna, seperti nama pengguna, rating ulasan, isi ulasan, serta link yang mengarah ke halaman aplikasi di Google Play. Setiap baris data merepresentasikan

satu ulasan pengguna, dan setiap ulasan mencakup informasi terkait dengan kepuasan atau feedback pengguna terhadap aplikasi Tinder. Proses scraping ini memungkinkan pengumpulan data secara otomatis dan terstruktur, yang selanjutnya dapat digunakan untuk menganalisis sentimen pengguna atau mengevaluasi area perbaikan dalam aplikasi Tinder berdasarkan ulasan yang tersedia.



Gambar 4. 6 Tahap hasil simpan data csv

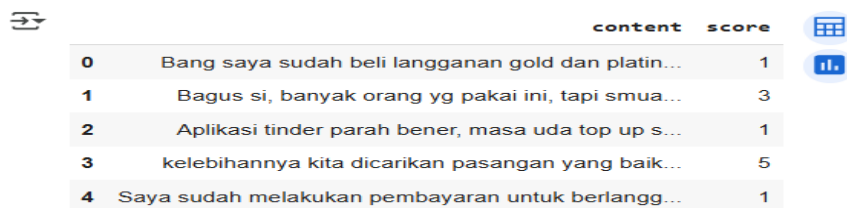
4.2 Data Selection

Pada langkah data selection ini penulis melakukan pemilihan kolom dataset sesuai dengan kebutuhan analisis.

Pada gambar 4.7, merupakan suatu proses kode Python menggunakan pandas untuk membaca file CSV berisi data ulasan Tinder, lalu menghapus beberapa kolom yang tidak diperlukan, seperti `reviewId`, `userName`, `userImage`, `thumbsUpCount`, dan `reviewCreatedVersion`. Hasilnya, yang tersisa hanya dua kolom utama, yaitu `content` (isi ulasan) dan `score` (skor ulasan). Proses ini bertujuan untuk membersihkan dan menyederhanakan dataset agar siap dianalisis lebih lanjut.

```
df = pd.read_csv('tinder_reviews_5000.csv')
df.head()
```

```
[ ] df = df.drop(['reviewId', 'userName', 'userImage', 'thumbsUpCount', 'reviewCreatedVersion', 'atReplyContent', 'repliedAtAppVersion'], axis=1)
df.head()
```



	content	score
0	Bang saya sudah beli langganan gold dan platin...	1
1	Bagus si, banyak orang yg pakai ini, tapi smua...	3
2	Aplikasi tinder parah bener, masa uda top up s...	1
3	kelebihannya kita dicarikan pasangan yang baik...	5
4	Saya sudah melakukan pembayaran untuk berlangg...	1

Gambar 4. 7 Membaca file csv

4.3 Pre-Processing

4.3.1 Checking For Nulls Values

Pada gambar 4.8 terlihat potongan kode Python yang digunakan untuk memeriksa keberadaan nilai null dalam sebuah DataFrame. Kode tersebut menggunakan metode `isnull()` yang mengembalikan DataFrame boolean, kemudian fungsi `any()` untuk mengecek apakah ada nilai null di setiap kolom. Langkah ini penting dalam proses pembersihan data agar model machine learning tidak terganggu oleh data yang hilang. Dengan mengetahui kolom yang memiliki nilai null, pengguna dapat mengambil tindakan seperti imputasi atau penghapusan data.

```
[ ] # Checking for null values
    df.isnull().values.any()
```

Gambar 4. 8 Checking For Nulls Values

4.3.3 Data Cleaning

Pada gambar 4.9 ditampilkan cuplikan kode yang berisi sejumlah pustaka Python yang digunakan dalam proses analisis data ulasan aplikasi Tinder. Pustaka `pandas` dan `numpy` digunakan untuk manipulasi dan pengolahan data dalam bentuk tabel dan array. Untuk membersihkan teks dari karakter yang tidak diinginkan seperti simbol, angka, atau emoji, digunakan pustaka `string` dan `re` (regular expression). Proses tokenisasi dan pemrosesan bahasa alami dilakukan dengan bantuan pustaka `nlTK`, khususnya fungsi `word_tokenize` yang memecah teks menjadi kata-kata. Visualisasi data dilakukan menggunakan `matplotlib`, `seaborn`, dan `WordCloud` untuk menampilkan distribusi sentimen dan kata-kata yang dominan secara visual.

```

# Manipulasi data
import pandas as pd
import numpy as np

# Pembersihan teks
import string
import re

# Tokenisasi dan NLP
import nltk
from nltk.tokenize import word_tokenize
nltk.download('punkt') # untuk word_tokenize

# Visualisasi
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud

```

Gambar 4. 9 Tahap persiapan pustala Python

Pada gambar 4.10 ditampilkan fungsi `cleaningText` yang digunakan untuk membersihkan teks ulasan dari berbagai elemen yang tidak relevan. Proses dimulai dengan mengubah seluruh huruf menjadi huruf kecil agar konsistensi analisis tetap terjaga. Selanjutnya, fungsi ini menghapus tagar, mention, angka, karakter superscript, emoji, tanda baca, serta mengganti karakter pemisah seperti garis miring dengan spasi. Fungsi ini juga menghilangkan karakter newline dan carriage return untuk memastikan teks berada dalam satu baris serta menghapus spasi berlebih yang dapat mengganggu proses tokenisasi. Dengan pembersihan ini, data teks menjadi lebih bersih dan siap untuk analisis lebih lanjut seperti tokenisasi atau klasifikasi sentimen.

```

def cleaningText(text):
    # 1. Lowercasing
    text = text.lower()

    # 2. Menghapus tagar (#) dan mention (@)
    text = re.sub(r'[@#][A-Za-z0-9_]+', '', text)

    # 3. Menghapus angka
    text = re.sub(r'\d+', '', text)

    # 4. Menghapus karakter superscript
    text = re.sub(r'[2314567890]', '', text)

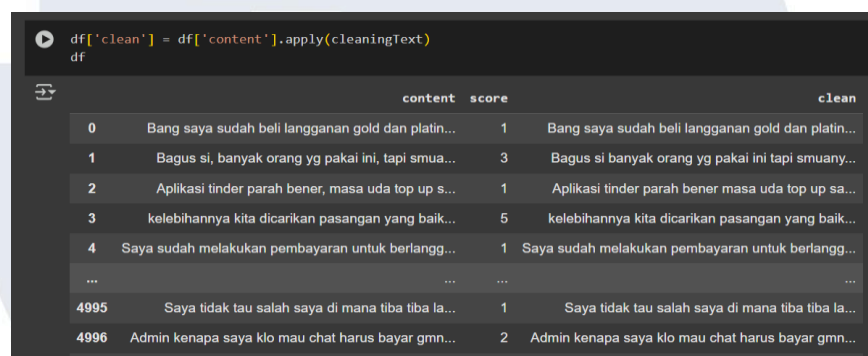
    # 5. Mengganti '/' atau '\' dengan spasi
    text = text.replace('/', ' ').replace('\', ' ')

    # 6. Menghapus emoji
    emoji_pattern = re.compile(
        "[
            u'\U0001F600-\U0001F64F' # emotikon wajah
            u'\U0001F300-\U0001F5FF' # simbol & pictographs
            u'\U0001F680-\U0001F6FF' # transport & map symbols
            u'\U0001F1E0-\U0001F1FF' # bendera negara
            u'\U00002500-\U00002BEF' # karakter Cina, simbol
            u'\U00002702-\U000027B0' # simbol tambahan
            u'\U000024C2-\U0001F251"

```

Gambar 4. 10 Tahap pembersihan teks

Pada gambar 4.11 ditampilkan proses penerapan fungsi `cleaningText` pada kolom `content` untuk menghasilkan kolom baru bernama `clean`. Setiap baris teks dalam kolom `content` dibersihkan dari karakter-karakter yang tidak relevan menggunakan metode preprocessing yang telah didefinisikan sebelumnya. Hasilnya, kolom `clean` berisi teks ulasan yang telah diproses dan siap digunakan untuk tahap analisis selanjutnya seperti tokenisasi atau normalisasi.



	content	score	clean
0	Bang saya sudah beli langganan gold dan platin...	1	Bang saya sudah beli langganan gold dan platin...
1	Bagus si, banyak orang yg pakai ini, tapi smua...	3	Bagus si banyak orang yg pakai ini tapi smuan...
2	Aplikasi tinder parah bener, masa uda top up s...	1	Aplikasi tinder parah bener masa uda top up sa...
3	kelebihannya kita dicarikan pasangan yang baik...	5	kelebihannya kita dicarikan pasangan yang baik...
4	Saya sudah melakukan pembayaran untuk berlangg...	1	Saya sudah melakukan pembayaran untuk berlangg...
...
4995	Saya tidak tau salah saya di mana tiba tiba la...	1	Saya tidak tau salah saya di mana tiba tiba la...
4996	Admin kenapa saya klo mau chat harus bayar gmn...	2	Admin kenapa saya klo mau chat harus bayar gmn...

Gambar 4. 11 penerapan fungsi teks

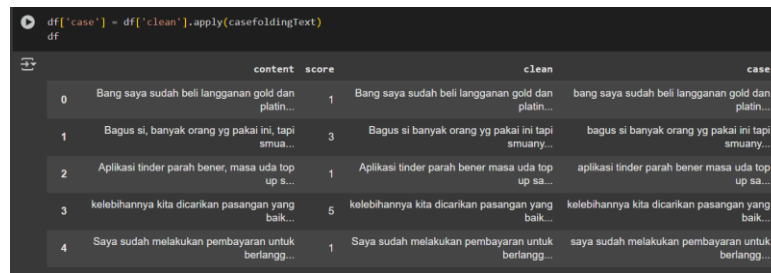
4.3.4 Case Folding

Pada gambar 4.12 ditampilkan fungsi casefoldingText yang digunakan untuk mengubah seluruh huruf dalam teks menjadi huruf kecil. Proses ini penting dilakukan agar teks memiliki konsistensi dan tidak terjadi perbedaan antara kata yang sama namun ditulis dengan huruf kapital. Langkah ini juga membantu meningkatkan akurasi dalam proses analisis teks selanjutnya.

```
[ ] def casefoldingText(text):
    text = text.lower()
    return text
```

Gambar 4. 12 Tahap perubahan huruf kecil

Pada gambar 4.13 ditunjukkan proses penerapan fungsi `casefoldingText` terhadap kolom `clean` dan hasilnya disimpan dalam kolom baru bernama `case`. Langkah ini memastikan bahwa seluruh kata dalam teks telah dikonversi ke huruf kecil secara konsisten. Hasil dari proses ini akan digunakan dalam tahapan pemrosesan teks berikutnya seperti tokenisasi dan analisis sentimen.



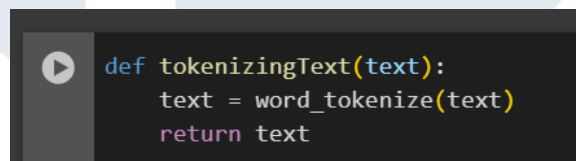
```
df['case'] = df['clean'].apply(casefoldingText)
```

	content	score	clean	case
0	Bang saya sudah beli langganan gold dan platin...	1	Bang saya sudah beli langganan gold dan platin...	bang saya sudah beli langganan gold dan platin...
1	Bagus si, banyak orang yg pakai ini, tapi smuanya...	3	Bagus si banyak orang yg pakai ini tapi smuanya...	bagus si banyak orang yg pakai ini tapi smuanya...
2	Aplikasi tinder parah bener, masa uda top up s...	1	Aplikasi tinder parah bener masa uda top up sa...	aplikasi tinder parah bener masa uda top up sa...
3	kelebihannya kita dicarikan pasangan yang baik...	5	kelebihannya kita dicarikan pasangan yang baik...	kelebihannya kita dicarikan pasangan yang baik...
4	Saya sudah melakukan pembayaran untuk berlang...	1	Saya sudah melakukan pembayaran untuk berlang...	saya sudah melakukan pembayaran untuk berlang...

Gambar 4. 13 Tahap penerapan fungsi perubahan huruf kecil

4.3.5 Tokenization

Pada gambar 4.14, fungsi `tokenizingText` diterapkan untuk membagi teks menjadi token atau kata-kata individu menggunakan metode `word_tokenize` dari NLTK. Proses ini memungkinkan teks untuk dianalisis lebih lanjut dengan memisahkan kata-kata dalam bentuk yang lebih mudah dikelola. Hasil tokenisasi disimpan dalam kolom baru, yang akan digunakan dalam analisis sentimen atau langkah pemrosesan lainnya.



```
def tokenizingText(text):
    text = word_tokenize(text)
    return text
```

Gambar 4. 14 Tahap tokenisasi teks untuk pemisahan kata

Pada gambar 4.15 terlihat hasil tokenizing dari sebuah teks berbahasa Indonesia yang membahas tentang aplikasi Tinder, langganan Gold, dan proses pembayaran. Tokenizing ini memecah kalimat menjadi kata-kata atau token yang lebih kecil untuk memudahkan analisis teks lebih lanjut. Dari token-token tersebut, terlihat adanya pembahasan mengenai pengalaman pengguna, kelebihan aplikasi, serta proses transaksi yang dilakukan. Informasi ini biasanya digunakan dalam pemrosesan bahasa alami (NLP) untuk memahami konteks dan makna dari teks asli.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

token
[bang, saya, sudah, beli, langganan, gold, dan...
[bagus, si, banyak, orang, yg, pakai, ini, tap...
[aplikasi, tinder, parah, bener, masa, uda, to...
[kelebihannya, kita, dicarikan, pasangan, yang...
saya, sudah, melakukan, pembayaran, untuk, be...

Gambar 4. 15 Tahap hasil tokenisasi

4.3.6 Normalization

4.3.6.1 Penggunaan Kamus Slangword

Pada gambar 4.16 terlihat kode Python yang memuat kamus slang dari file CSV bernama 'kamusalay.csv' dan mengubahnya menjadi dictionary Python dengan pasangan kata slang dan kata normal. Fungsi `cvt_slang` kemudian digunakan untuk mengonversi sebuah kalimat dengan mengganti setiap kata slang yang ada dalam kamus menjadi kata normalnya. Proses ini penting dalam preprocessing teks agar kata-kata tidak baku atau slang dapat distandarisasi sehingga analisis teks menjadi lebih akurat. Dengan cara ini, data teks yang awalnya berisi bahasa sehari-hari yang informal bisa diubah menjadi bentuk yang lebih formal dan mudah diproses oleh model.

```
[ ] slang_dict = pd.read_csv('kamusalay.csv', encoding='latin1', names=['alay', 'normal'])
    slang_dict = dict(zip(slang_dict.alay.tolist(), slang_dict.normal.tolist()))

[ ] def cvt_slang(sentence):
    return ' '.join([slang_dict[x] if x in slang_dict else x for x in sentence])
```

Gambar 4. 16 Tahap normalisasi dengan kamus slang

Pada gambar 4.17 terlihat proses penerapan fungsi konversi slang pada kolom 'token' di DataFrame untuk menghasilkan kolom baru bernama 'slang'. Fungsi ini menggantikan kata-kata slang dalam token dengan padanan kata normal sesuai kamus yang sudah dibuat sebelumnya. Dengan cara ini, teks yang awalnya mengandung bahasa tidak baku menjadi lebih standar dan siap untuk analisis selanjutnya.

`df['slang'] = df['token'].apply(cvt_slang)`

	content	score	clean	case	token	slang
0	Bang saya sudah beli langganan gold dan platin...	1	Bang saya sudah beli langganan gold dan platin...	bang saya sudah beli langganan gold dan platin...	[bang, saya, sudah, beli, langganan, gold, dan...	bang saya sudah beli langganan gold dan platin...
1	Bagus si, banyak orang yg pakai ini, tapi smua...	3	Bagus si banyak orang yg pakai ini tapi smuany...	bagus si banyak orang yg pakai ini tapi smuany...	[bagus, si, banyak, orang, yg, pakai, ini, tapi...	bagus sih banyak orang yang pakai ini tapi sem...
2	Aplikasi tinder parah bener, masa uda top up s...	1	Aplikasi tinder parah bener masa uda top up sa...	aplikasi tinder parah bener masa uda top up sa...	[aplikasi, tinder, parah, bener, masa, uda, to...	aplikasi tinder parah bener masa sudah top up ...
4998	Aplikasi ini sangat jelek gak bisa masuk terus	1	Aplikasi ini sangat jelek gak bisa masuk terus	aplikasi ini sangat jelek gak bisa masuk terus	[aplikasi, ini, sangat, jelek, gak, bisa, masu...	aplikasi ini sangat jelek tidak bisa masuk terus
4999	Min, maaf ni ane koreksi ko akhir akhir ini ti...	2	Min maaf ni ane koreksi ko akhir akhir ini tin...	min maaf ni ane koreksi ko akhir akhir ini tin...	[min, maaf, ni, ane, koreksi, ko, akhir, akhir...	min maaf nih saya koreksi kok akhir akhir ini ...

Gambar 4. 17 Hasil penerapan konversi slang

4.3.6.2 Statistik Frekuensi Kata (Low Frequency Words)

Pada gambar 4.18 terlihat proses pengecekan statistik frekuensi kata pada kolom 'slang' di DataFrame untuk mengetahui banyaknya kata – kata unik untuk proses penentuan tahap berikutnya yaitu stopwords.

```
from collections import Counter

# Tokenisasi dan hitung frekuensi kata unik
all_words = " ".join(df["slang"]).split()
word_freq = Counter(all_words)

# Tampilkan 50 kata unik yang paling sering muncul
word_freq.most_common(50)
```

```
[('tidak', 4814),
 ('di', 2703),
 ('bisa', 2683),
 ('yang', 2404),
 ('saya', 2316),
 ('sudah', 1984),
 ('aplikasi', 1878),
 ('tinder', 1571),
 ('ini', 1562),
 ('dan', 1409),
 ('akun', 1235),
 ('saja', 1228),
 ('nya', 1221),
 ('ada', 1202),
```

Gambar 4. 18 Tahap pengecekan frekuensi kata unik

4.3.7 Stopword

Pada gambar 4.19 terlihat definisi daftar stopwords dalam bahasa Indonesia yang terdiri dari dua list, yaitu custom_stopwords dan additional_stopwords. custom_stopwords berisi kata-kata umum dan informal yang sering muncul dalam data dan dianggap kurang bermakna untuk analisis, seperti "tidak", "saya", "bang", dan "hade". Sedangkan additional_stopwords adalah kata-kata tambahan yang juga sering muncul dan dipilih berdasarkan frekuensi tinggi, seperti "di", "yang", "ini", dan "dan".

```

# Stopwords baru, berdasarkan kata frekuensi tinggi yang cocok
custom_stopwords = [
    'tidak', 'saya', 'aku', 'bang', 'hade', 'hadee', 'haduh', 'gan', 'di', 'masa', 'masak', 'yang', 'ini', 'dan', 'saja', 'ada', 'lagi',
    'baru', 'padahal', 'kalau', 'tapi', 'juga', 'harus', 'terus',
    'apa', 'sana', 'untuk', 'banyak', 'apakah', 'malah', 'ke', 'dengan',
    'nya', 'ya', 'sih', 'kok', 'banget' # Tambahkan dari kata informal yang masih relevan
]

# Tambahkan stopwords dari kata frekuensi tinggi yang dipilih
additional_stopwords = [
    'tidak', 'di', 'yang', 'ini', 'dan', 'saja', 'ada', 'lagi',
    'baru', 'padahal', 'kalau', 'tapi', 'juga', 'harus', 'terus',
    'apa', 'sana', 'untuk', 'banyak', 'apakah', 'malah', 'ke', 'dengan'
]

# Gabungkan semua stopwords
final_stopwords = list(set(custom_stopwords + additional_stopwords))

```

Gambar 4. 19 Tahap Stopwords

Kedua daftar ini kemudian digabung menjadi satu list `final_stopwords` yang akan digunakan untuk menghapus kata-kata tersebut dari teks agar analisis menjadi lebih fokus pada kata-kata penting. Penggunaan stopwords ini sangat membantu dalam meningkatkan kualitas hasil analisis teks.

Pada gambar 4.20, terlihat tabel yang menunjukkan contoh kalimat sebelum dan sesudah proses pembersihan teks. Kolom "Slang (Before)" berisi kalimat asli yang mengandung slang dan kata tidak baku, sedangkan kolom "Stopwords (After)" berisi kalimat yang sudah dibersihkan dari kata-kata tidak penting dan slang. Proses ini membantu memfokuskan analisis pada kata-kata yang lebih bermakna dan relevan. Dengan pembersihan ini, model NLP dapat lebih efektif dalam memahami dan mengklasifikasikan sentimen dari teks. Ini menunjukkan pentingnya tahap preprocessing dalam pipeline analisis teks.

Slang (Before)	Stopwords (After)
bang saya sudah beli langganan gold dan platin...	sudah beli langganan gold platinum pesan satup...
bagus si banyak orang yg pakai ini tapi semuanya...	bagus orang pakai semuanya serba berbayar suda...
aplikasi tinder parah benar masa sudah top up saldo...	aplikasi tinder parah benar suda top up saldo...
kelebihannya kita dicarikan pasangan yang baik...	kelebihannya kita dicarikan pasangan yang baik...
saya sudah melakukan pembayaran untuk berlangg...	sudah melakukan pembayaran berlangganan akun g...

Gambar 4. 20 Tabel setelah melakukan stopword

4.3.8 Labeling Sentimen

Pada gambar 4.21, terlihat bahwa penelitian ini menggunakan metode pelabelan kamus lexicon Indonesia yang menggunakan dua file utama yaitu "positive.tsv" dan "negative.tsv" yang merupakan bagian dari InSet (Indonesia Sentiment Lexicon). Pelabelan ini memanfaatkan daftar kata yang sudah diklasifikasikan berdasarkan sentimen positif dan negatif untuk menentukan label

sentimen pada data teks. Dengan menggunakan kamus lexicon ini, proses analisis sentimen menjadi lebih terstruktur dan dapat mengidentifikasi nuansa emosi dalam teks secara lebih akurat. Penggunaan kamus ini penting dalam penerapan sistem pengolahan bahasa alami khususnya untuk bahasa Indonesia.



Gambar 4. 21 Indonesia Sentiment Lexicon

Pada gambar 4.22 terlihat kode yang memastikan kolom 'stopwords' bertipe string dan mengganti nilai kosong (NaN) dengan string kosong agar data siap diproses. Selain itu, terdapat definisi daftar kata positif, negatif, dan netral dalam bahasa Indonesia yang akan digunakan untuk analisis sentimen. Daftar kata ini berfungsi sebagai referensi untuk menentukan apakah sebuah teks mengandung sentimen positif, dan negatif. berdasarkan kemunculan kata-kata yang ada pada kamus tersebut. Pendekatan ini termasuk dalam metode leksikon (lexicon-based), karena menggunakan kamus kata yang sudah didefinisikan sebelumnya tanpa perlu pelatihan model.

```

# Fungsi pelabelan sentimen
def label_sentiment(text):
    words = word_tokenize(str(text).lower())
    positive_count = sum(1 for word in words if word in positive_words)
    negative_count = sum(1 for word in words if word in negative_words)

    if positive_count > negative_count:
        return 'positif'
    elif negative_count > positive_count:
        return 'negatif'

```

Gambar 4. 22 Tahap pelabelan sentimen

Pada gambar 4.23 terlihat tabel yang menampilkan contoh kalimat dalam tiga tahap pemrosesan teks, yaitu kolom "slang" yang berisi teks asli dengan bahasa sehari-hari, kolom "stopwords" yang berisi teks setelah penghapusan kata-kata tidak penting, dan kolom "sentimen" yang menunjukkan hasil analisis sentimen berupa label positif, atau negatif. Tabel ini memperlihatkan bagaimana teks yang awalnya informal dan panjang menjadi lebih ringkas dan fokus setelah penghapusan stopwords, sehingga memudahkan proses analisis sentimen. Hasil sentimen yang diberikan juga menunjukkan variasi perasaan pengguna terhadap aplikasi Tinder berdasarkan isi review mereka.

slang	stopwords	sentimen
bang saya sudah beli langganan gold dan platin...	sudah beli langganan gold platinum pesan satup...	positif
bagus sih banyak orang yang pakai ini tapi sem...	bagus orang pakai semuanya serba berbayar suda...	negatif
aplikasi tinder parah benar masa sudah top up ...	aplikasi tinder parah benar sudah top up saldo...	negatif

Gambar 4. 23 Hasil tahapan pelabelan

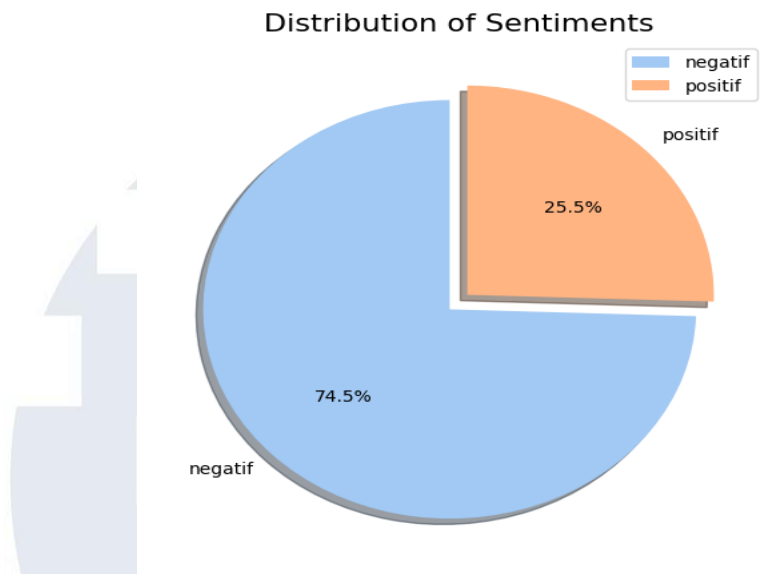
Pada gambar 4.24, disajikan representasi visual berupa word cloud yang merefleksikan frekuensi kemunculan kata-kata dalam korpus teks dengan sentimen positif dan negatif. Kata-kata dominan pada word cloud positif, seperti "suka,"

si yang signifikan dalam konteks afektif positif. Sebaliknya, pada konteks negatif, kata-kata seperti "tidak," "salah," dan "kalah" muncul. Hal ini menunjukkan bahwa word cloud dapat menangkap nuansa emosional. Selain itu, word cloud juga dapat mengidentifikasi kata-kata yang paling sering muncul dalam suatu teks, yang dapat membantu dalam memahami tema atau topik yang dibahas. Dalam konteks ini, word cloud berfungsi sebagai alat eksplorasi data yang efektif untuk menganalisis pola distribusi leksikal berdasarkan polaritas sentimen. Dengan demikian, word cloud dapat menjadi alat yang berguna untuk analisis teks Bahasa Indonesia. Oleh karena itu, word cloud ini merupakan alat yang penting untuk analisis kualitatif yang lebih mendalam terkait karakteristik linguistik yang ada dalam teks yang dianalisis.



Gambar 4. 24 Wordcloud Positif dan Negatif

menominasi dengan persentase sebesar 74,5%, sedangkan sentimen negatif mencapai 25,5%. Visualisasi ini memberikan gambaran kuantitatif mengenai sebaran sentimen dalam dataset yang sedang diteliti. Informasi ini penting untuk memahami karakteristik utama dari data, khususnya dalam konteks analisis sentimen berbahasa Indonesia.



Gambar 4. 25 Visualisasi distribusi sentimen

4.4 Transformation

4.4.1 Inisialisasi Tokenizer

Pada gambar 4.26 terlihat kode Python yang membuat objek Tokenizer dengan parameter `num_words=5000` yang berarti tokenizer akan membatasi hanya pada 5000 kata paling sering muncul dalam data. Selain itu, ada parameter `oov_token='<OOV>'` yang berfungsi sebagai token pengganti untuk kata-kata yang tidak dikenal atau tidak ada dalam kamus tokenizer. Setelah itu, tokenizer dilatih dengan memproses teks dari kolom 'stopwords' menggunakan metode `fit_on_texts()`, sehingga tokenizer dapat membangun kamus kata berdasarkan data tersebut. Proses ini penting agar teks dapat diubah menjadi urutan angka yang bisa dipahami oleh model machine learning.

```
tokenizer = Tokenizer(num_words=5000, oov_token='<OOV>')
tokenizer.fit_on_texts(df['stopwords'])
```

Gambar 4. 26 Tahap inisialisasi tokenizer

4.4.2 Tokenisasi dan Padding

Pada gambar 4.27 terlihat kode Python yang mengubah teks pada kolom 'stopwords' menjadi urutan angka menggunakan fungsi `tokenizer.texts_to_sequences()`. Setiap kata dalam teks diganti dengan indeks numerik sesuai kamus yang dibuat oleh tokenizer. Selanjutnya, kode melakukan

padding pada urutan angka tersebut dengan `pad_sequences()` agar semua urutan memiliki panjang yang sama, yaitu 200 token. Padding ini penting agar data input seragam dan bisa diproses oleh model deep learning.

```
x = tokenizer.texts_to_sequences(df['stopwords'])
x = pad_sequences(x, maxlen=200)
```

Gambar 4. 27 Tahap tokenisasi dan padding

4.4.3 Encoding Label

Pada gambar 4.28 terlihat kode Python yang melakukan inisialisasi objek `LabelEncoder` untuk mengubah label sentimen dari bentuk teks menjadi bentuk numerik. Proses ini dilakukan dengan fungsi `fit_transform()` yang mengonversi setiap label teks menjadi angka yang sesuai. Setelah itu, label numerik tersebut diubah lagi menjadi format one-hot encoding menggunakan fungsi `to_categorical()`, sehingga setiap label direpresentasikan dalam bentuk vektor biner yang cocok untuk klasifikasi multi-kelas. Langkah ini penting agar model deep learning dapat memahami dan memproses label dengan benar selama pelatihan.

```
# Inisialisasi LabelEncoder
label_encoder = LabelEncoder()

# Mengonversi label teks ke numerik
df['label_encoded'] = label_encoder.fit_transform(df['sentimen'])

# Mengonversi label numerik ke one-hot encoding
y = to_categorical(df['label_encoded'])
```

Gambar 4. 28 Tahap label encode

4.4.4 Split Data Latih & Uji

Pada gambar 4.29 terlihat sebuah potongan kode Python yang melakukan pembagian data menjadi set pelatihan dan pengujian menggunakan fungsi `train_test_split`. Kode ini mengimpor fungsi dari modul yang relevan, lalu menerapkannya dengan parameter fitur, label, ukuran data uji, dan `random_state` untuk memastikan hasil yang konsisten. Proses ini merupakan langkah penting dalam persiapan data untuk melatih dan menguji model machine learning secara efektif.

```
[ ] from sklearn.model_selection import train_test_split

# Misalnya, X adalah fitur Anda yang telah diproses sebelumnya
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Gambar 4. 29 Tahap split data latih uji

4.5 Data Mining

4.5.1 Pemodelan BILSTM

Pada gambar 4.30, terlihat kode Python yang memuat berbagai pustaka penting untuk proyek machine learning dan pemrosesan data. Kode ini mengimpor pustaka seperti pandas dan numpy untuk manipulasi data, matplotlib dan seaborn untuk visualisasi, serta sklearn untuk pemodelan dan evaluasi. Selain itu, ada juga impor dari TensorFlow dan Keras yang digunakan untuk membangun dan melatih model deep learning, khususnya untuk pemrosesan teks dengan modul seperti Tokenizer dan pad_sequences. Secara keseluruhan, kode ini menyiapkan fondasi yang lengkap untuk analisis data dan pengembangan model pembelajaran mesin berbasis teks.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Embedding, LSTM, Dense, SpatialDropout1D,
                                     Bidirectional, Dropout, Conv1D, MaxPooling1D, GlobalMaxPooling1D)
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
```

Gambar 4. 30 Impor Pustaka python BILSTM

Pada gambar 4.31, terlihat definisi model BI-LSTM menggunakan Keras dengan arsitektur Sequential. Model ini dimulai dengan layer embedding yang mengubah input menjadi representasi vektor berdimensi 128, diikuti oleh layer bidirectional LSTM dengan 64 unit yang memungkinkan model memproses informasi dari dua arah. Selanjutnya, ada layer dropout dengan tingkat 0.5 untuk mencegah overfitting, kemudian dua layer dense, satu dengan aktivasi ReLU untuk menangkap pola non-linear, dan yang terakhir dengan aktivasi softmax untuk

klasifikasi ke dalam tiga kelas. Model ini cocok digunakan untuk tugas klasifikasi teks yang memerlukan pemahaman konteks dari kedua arah kalimat.

```
bi_lstm_model = Sequential([
    Embedding(input_dim=5000, output_dim=128, input_length=200),
    Bidirectional(LSTM(64)),
    Dropout(0.5),
    Dense(64, activation='relu'),
    Dense(2, activation='softmax')
])
```

Gambar 4. 31 Pemodelan BILSTM

Pada gambar 4.32, terlihat sebuah kode Python yang melakukan pelatihan model untuk BI-LSTM menggunakan loop. Setiap model dikompilasi dengan fungsi loss `categorical_crossentropy`, optimizer Adam, dan metrik akurasi untuk evaluasi. Proses pelatihan dilakukan selama 5 epoch dengan batch size 64, serta menggunakan data validasi untuk memantau performa model. Hasil pelatihan dari masing-masing model disimpan dalam dictionary `history_dict` untuk analisis lebih lanjut.

```
models = {'BILSTM': bi_lstm_model}

history_dict = {}
for name, model in models.items():
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    print(f"\nTraining {name} model...")
    history = model.fit(
        X_train, y_train,
        epochs=5,
        batch_size=64,
        validation_data=(X_test, y_test),
        verbose=1
    )
    history_dict[name] = history
```

Gambar 4. 32 Perulangan Kompilasi & Pelatihan BILSTM

4.5.2 Pemodelan BILSTM Class weights

Pada gambar 4.33 terlihat potongan kode Python yang memuat berbagai pustaka penting untuk analisis data dan pembelajaran mesin. Pustaka seperti `pandas` dan `numpy` digunakan untuk manipulasi data dan perhitungan numerik, sedangkan `matplotlib` dan `seaborn` berfungsi untuk visualisasi data secara grafis. Selain itu, terdapat juga modul dari `scikit-learn`, `tensorflow`, dan `keras` yang mendukung pembangunan dan pelatihan model machine learning serta deep learning. Dengan mengimpor modul-modul ini, kode tersebut menyiapkan lingkungan yang lengkap

untuk melakukan eksplorasi data, pemodelan, dan evaluasi hasil secara menyeluruh.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.utils.class_weight import compute_class_weight

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Embedding, LSTM, Dense, SpatialDropout1D,
                                     Bidirectional, Dropout, Conv1D, MaxPooling1D, GlobalMaxPooling1D)
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
```

Gambar 4. 33 Impor Pustaka python BILSTM Class weights

Pada gambar 4.34 terlihat potongan kode Python yang menggunakan fungsi `compute_class_weight` dari pustaka scikit-learn untuk menghitung bobot kelas pada dataset yang tidak seimbang. Kode ini menampilkan hasil bobot kelas dalam bentuk angka desimal yang sesuai dengan label kelas 0, dan 1. Komentar dalam bahasa Indonesia menjelaskan tujuan dari perhitungan bobot ini, yaitu untuk mengatasi ketidakseimbangan data agar model dapat belajar secara lebih efektif. Dengan menggunakan bobot kelas ini, proses pelatihan model menjadi lebih adil terhadap kelas minoritas sehingga meningkatkan akurasi prediksi.

```
# Hitung class weights untuk menangani imbalance
class_weights = compute_class_weight(class_weight='balanced', classes=np.unique(y_train_labels), y=y_train_labels)
class_weight_dict = dict(enumerate(class_weights))
print("Class weights:", class_weight_dict)

Class weights: {0: np.float64(0.6337135614702155), 1: np.float64(2.3696682464454977)}
```

Gambar 4. 34 menghitung class weights BILSTM

Pada gambar 4.35 terlihat definisi model bi-LSTM menggunakan Keras/TensorFlow yang terdiri dari beberapa lapisan berurutan. Model dimulai dengan lapisan Embedding yang berfungsi mengubah data teks menjadi representasi vektor berdimensi tetap. Selanjutnya, terdapat lapisan Bidirectional LSTM yang mampu menangkap konteks dari kedua arah urutan data, diikuti oleh lapisan Dropout untuk mencegah overfitting.

```

bi_lstm_model = Sequential([
    Embedding(input_dim=5000, output_dim=128, input_length=200),
    Bidirectional(LSTM(64)),
    Dropout(0.5),
    Dense(64, activation='relu'),
    Dense(2, activation='softmax')
])

```

Gambar 4. 35 Definisi model BILSTM Class Weights

Pada gambar 4.36 terlihat potongan kode Python yang melakukan pelatihan model BILSTM secara berulang dengan menambahkan `compute_class_weight`. Setiap model dikompilasi dengan fungsi loss categorical crossentropy dan optimizer Adam untuk mengoptimalkan proses pembelajaran. Proses pelatihan dilakukan dengan memasukkan data pelatihan dan validasi, serta menampilkan progres pelatihan secara real-time.

```

models = {'BiLSTM': bi_lstm_model}

history_dict = {}
for name, model in models.items():
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    print(f"\nTraining {name} model...")
    history = model.fit(
        X_train, y_train,
        epochs=5,
        batch_size=64,
        validation_data=(X_test, y_test),
        class_weight=class_weight_dict, # Menambahkan class_weight di sini
        verbose=1
    )
    history_dict[name] = history

```

Gambar 4. 36 Perulangan Kompilasi & Pelatihan BILSTM Class weights

4.5.3 Pemodelan BILSTM Class Weights Hyperparameter Tuning

Pada gambar 4.37 terlihat serangkaian perintah import pada bahasa pemrograman Python yang memuat berbagai pustaka penting seperti pandas, numpy, matplotlib, seaborn, scikit-learn, tensorflow, dan keras-tuner. Pustaka-pustaka ini digunakan untuk keperluan pengolahan data, visualisasi, pembangunan model machine learning, serta optimasi hyperparameter. Selain itu, terdapat modul-modul khusus yang mendukung preprocessing data, pembuatan model neural network, dan evaluasi performa model secara menyeluruh.

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.utils.class_weight import compute_class_weight

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Embedding, LSTM, Dense, SpatialDropout1D,
                                     Bidirectional, Dropout, Conv1D, MaxPooling1D, GlobalMaxPooling1D)
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical

!pip install keras-tuner
import keras_tuner as kt

```

Gambar 4. 37 Impor Pustaka python BiLSTM Class weights Hyperparameter

Pada gambar 4.38 terlihat definisi model BiLSTM (Bidirectional Long Short-Term Memory) menggunakan bahasa pemrograman Python dengan framework Keras. Model ini terdiri dari lapisan Embedding untuk mengubah kata menjadi representasi vektor, diikuti oleh lapisan Bidirectional LSTM yang mampu menangkap konteks dari dua arah dalam data berurutan. Selanjutnya terdapat lapisan Dropout yang berfungsi untuk mencegah overfitting dengan mengacak sebagian neuron selama pelatihan, dan diakhiri dengan lapisan Dense yang menggunakan aktivasi softmax untuk klasifikasi. Fungsi build_bilstm_model ini dirancang untuk membangun dan mengembalikan model BiLSTM yang siap dilatih pada data teks.

```

A. BiLSTM : Membuat model

def build_bilstm_model(hp):
    model = Sequential()
    model.add(Embedding(input_dim=5000, output_dim=hp.Int('embedding_dim', 64, 256, step=32), input_length=200))
    model.add(Bidirectional(LSTM(units=hp.Int('lstm_units', 32, 256, step=32), return_sequences=False)))
    model.add(Dropout(hp.Float('dropout', 0.2, 0.5, step=0.1)))
    model.add(Dense(units=hp.Int('dense_units', 32, 128, step=32), activation='relu'))
    model.add(Dense(2, activation='softmax'))

    model.compile(
        optimizer=tf.keras.optimizers.Adam(hp.Float('learning_rate', 1e-4, 1e-2, sampling='LOG')),
        loss='categorical_crossentropy',
        metrics=['accuracy'])

    return model

```

Gambar 4. 38 Pemodelan Hyperparameter BiLSTM

Pada gambar 4.39 terlihat penggunaan Keras Tuner dengan metode RandomSearch untuk melakukan pencarian hyperparameter terbaik pada model BiLSTM. Proses tuning ini menggunakan metrik validasi akurasi (val_accuracy) sebagai acuan untuk memilih model terbaik dari beberapa percobaan (max_trials). Hasil pencarian disimpan dalam direktori bernama 'bilstm_tuner' agar dapat diakses

kembali atau dianalisis lebih lanjut. Selanjutnya, tuner ini dijalankan dengan data pelatihan dan validasi untuk mengoptimalkan performa model secara otomatis.

```
4. Mencari model terbaik

tuner = kt.RandomSearch(
    build_bilstm_model,
    objective='val_accuracy',
    max_trials=3,
    executions_per_trial=1,
    directory='bilstm_tuner',
    project_name='sentiment_bilstm'
)

tuner.search(X_train, y_train, epochs=5, validation_data=(X_test, y_test))
```

Gambar 4. 39 Training hyperparameter terbaik BILSTM

4.5.4 Pemodelan LSTM

Pada gambar 4.40, terlihat kode Python yang memuat berbagai pustaka penting untuk proyek machine learning dan pemrosesan data. Kode ini mengimpor pustaka seperti pandas dan numpy untuk manipulasi data, matplotlib dan seaborn untuk visualisasi, serta sklearn untuk pemodelan dan evaluasi. Selain itu, ada juga impor dari TensorFlow dan Keras yang digunakan untuk membangun dan melatih model deep learning, khususnya untuk pemrosesan teks dengan modul seperti Tokenizer dan pad_sequences. Secara keseluruhan, kode ini menyiapkan fondasi yang lengkap untuk analisis data dan pengembangan model pembelajaran mesin berbasis teks.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Embedding, LSTM, Dense, SpatialDropout1D,
                                     Bidirectional, Dropout, Conv1D, MaxPooling1D, GlobalMaxPooling1D)
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
```

Gambar 4. 40 Impor Pustaka python LSTM

Pada gambar 4.41, ditampilkan sebuah model LSTM yang dibangun menggunakan arsitektur Sequential. Model ini diawali dengan layer embedding yang mengubah input dengan dimensi 5000 menjadi representasi vektor berdimensi 128, dengan panjang input 200. Selanjutnya, terdapat layer LSTM dengan 128 unit

yang dilengkapi dropout dan recurrent dropout sebesar 0.2 untuk mencegah overfitting. Terakhir, model ini menggunakan layer dense dengan 3 unit dan fungsi aktivasi softmax, yang biasanya digunakan untuk klasifikasi multi-kelas.

```
lstm_model = Sequential([
    Embedding(input_dim=5000, output_dim=128, input_length=200),
    LSTM(128, dropout=0.2, recurrent_dropout=0.2),
    Dense(2, activation='softmax')
])
```

Gambar 4. 41 Pemodelan LSTM

Pada gambar 4.42, terlihat sebuah kode Python yang melakukan pelatihan model untuk LSTM menggunakan loop. Setiap model dikompilasi dengan fungsi loss categorical_crossentropy, optimizer Adam, dan metrik akurasi untuk evaluasi. Proses pelatihan dilakukan selama 5 epoch dengan batch size 64, serta menggunakan data validasi untuk memantau performa model. Hasil pelatihan dari masing-masing model disimpan dalam dictionary history_dict untuk analisis lebih lanjut.

```
models = {'LSTM': lstm_model}

history_dict = {}
for name, model in models.items():
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    print(f"\nTraining {name} model...")
    history = model.fit(
        X_train, y_train,
        epochs=5,
        batch_size=64,
        validation_data=(X_test, y_test),
        verbose=1
    )
    history_dict[name] = history
```

Gambar 4. 42 Perulangan Kompilasi & Pelatihan LSTM

4.5.5 Pemodelan LSTM Class weights

Pada gambar 4.43 terlihat potongan kode Python yang memuat berbagai pustaka penting untuk analisis data dan pembelajaran mesin. Pustaka seperti pandas dan numpy digunakan untuk manipulasi data dan perhitungan numerik, sedangkan matplotlib dan seaborn berfungsi untuk visualisasi data secara grafis. Selain itu, terdapat juga modul dari scikit-learn, tensorflow, dan keras yang mendukung pembangunan dan pelatihan model machine learning serta deep learning. Dengan mengimpor modul-modul ini, kode tersebut menyiapkan lingkungan yang lengkap

untuk melakukan eksplorasi data, pemodelan, dan evaluasi hasil secara menyeluruh.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.utils.class_weight import compute_class_weight

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Embedding, LSTM, Dense, SpatialDropout1D,
                                     Bidirectional, Dropout, Conv1D, MaxPooling1D, GlobalMaxPooling1D)
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
```

Gambar 4. 43 Impor Pustaka python LSTM Class weights

Pada gambar 4.44 terlihat potongan kode Python yang menggunakan fungsi `compute_class_weight` dari pustaka scikit-learn untuk menghitung bobot kelas pada dataset yang tidak seimbang. Kode ini menampilkan hasil bobot kelas dalam bentuk angka desimal yang sesuai dengan label kelas 0, dan 1. Komentar dalam bahasa Indonesia menjelaskan tujuan dari perhitungan bobot ini, yaitu untuk mengatasi ketidakseimbangan data agar model dapat belajar secara lebih efektif. Dengan menggunakan bobot kelas ini, proses pelatihan model menjadi lebih adil terhadap kelas minoritas sehingga meningkatkan akurasi prediksi.

```
# Hitung class weights untuk menangani imbalance
class_weights = compute_class_weight(class_weight='balanced', classes=np.unique(y_train_labels), y=y_train_labels)
class_weight_dict = dict(enumerate(class_weights))
print("Class weights:", class_weight_dict)

Class weights: {0: np.float64(0.6337135614702155), 1: np.float64(2.3696682464454977)}
```

Gambar 4. 44 menghitung class weights LSTM

Pada gambar 4.45 terlihat definisi model LSTM menggunakan Keras/TensorFlow yang terdiri dari beberapa lapisan utama. Model dimulai dengan lapisan Embedding yang mengubah input teks menjadi representasi vektor berdimensi 128 dengan panjang input 200. Selanjutnya, terdapat lapisan LSTM dengan dropout yang berfungsi menangkap pola urutan data sekaligus mencegah overfitting selama pelatihan.

```

lstm_model = Sequential([
    Embedding(input_dim=5000, output_dim=128, input_length=200),
    LSTM(128, dropout=0.2, recurrent_dropout=0.2),
    Dense(2, activation='softmax')
])

```

Gambar 4. 45 Definisi model LSTM Class weights

Pada gambar 4.46 terlihat potongan kode Python yang menunjukkan proses pelatihan model LSTM untuk analisis sentimen. Kode ini mencakup definisi model, kompilasi dengan fungsi loss, optimizer, dan metrik evaluasi yang sesuai untuk klasifikasi multi-kelas. Selanjutnya, model dilatih menggunakan data pelatihan dengan jumlah epoch dan batch size yang telah ditentukan, serta validasi menggunakan data pengujian. Komentar dalam kode menggunakan bahasa Indonesia dan Inggris membantu memperjelas setiap langkah dalam proses pelatihan.

```

models = {'LSTM': lstm_model}

history_dict = {}
for name, model in models.items():
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    print(f"\nTraining {name} model...")
    history = model.fit(
        X_train, y_train,
        epochs=5,
        batch_size=64,
        validation_data=(X_test, y_test),
        class_weight=class_weight_dict, # Menambahkan class_weight di sini
        verbose=1
    )
    history_dict[name] = history

```

Gambar 4. 46 Perulangan Kompilasi & Pelatihan LSTM Class weights

4.5.6 Pemodelan LSTM Class weights Hyperparameter

Pada gambar 4.47 terlihat serangkaian perintah import pada bahasa pemrograman Python yang memuat berbagai pustaka penting seperti pandas, numpy, matplotlib, seaborn, scikit-learn, tensorflow, dan keras-tuner. Pustaka-pustaka ini digunakan untuk keperluan pengolahan data, visualisasi, pembangunan model machine learning, serta optimasi hyperparameter. Selain itu, terdapat modul-modul khusus yang mendukung preprocessing data, pembuatan model neural network, dan evaluasi performa model secara menyeluruh.

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.utils.class_weight import compute_class_weight

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Embedding, LSTM, Dense, SpatialDropout1D,
                                     Bidirectional, Dropout, Conv1D, MaxPooling1D, GlobalMaxPooling1D)
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical

!pip install keras-tuner
import keras_tuner as kt

```

Gambar 4. 47 Impor Pustaka python LSTM Class weights Hyperparameter

Pada gambar 4.48 terlihat definisi fungsi `build_lstm_model` yang digunakan untuk membangun model Long Short-Term Memory (LSTM) menggunakan framework Keras. Model ini terdiri dari lapisan Embedding untuk representasi kata, diikuti oleh lapisan LSTM dengan konfigurasi unit dan dropout untuk mengurangi overfitting. Selanjutnya terdapat lapisan Dense sebagai output dengan aktivasi softmax untuk klasifikasi multi-kelas. Fungsi ini juga mengatur optimizer, loss function, dan metrik evaluasi yang digunakan selama proses pelatihan model.

```

def build_lstm_model(hp):
    model = Sequential()
    model.add(Embedding(input_dim=5000, output_dim=hp.int('embedding_dim', 64, 256, step=32), input_length=200))
    model.add(LSTM(
        units=hp.int('lstm_units', 32, 256, step=32),
        dropout=hp.Float('dropout', 0.2, 0.5, step=0.1),
        recurrent_dropout=hp.Float('recurrent_dropout', 0.2, 0.5, step=0.1)
    ))
    model.add(Dense(units=hp.int('dense_units', 32, 128, step=32), activation='relu'))
    model.add(Dense(3, activation='softmax'))

    model.compile(
        optimizer=tf.keras.optimizers.Adam(hp.Float('learning_rate', 1e-4, 1e-2, sampling='log')),
        loss='categorical_crossentropy',
        metrics=['accuracy']
    )
    return model

```

Gambar 4. 48 Pemodelan Tuning Hyperparameter LSTM

Pada gambar 4.49 menunjukkan kode Python yang menggunakan Keras Tuner dengan metode RandomSearch untuk mencari model LSTM terbaik. Proses ini melibatkan pengaturan tuner dengan parameter seperti tujuan optimasi (objective), jumlah percobaan maksimal (max trials), dan direktori penyimpanan hasil. Selanjutnya, tuner melakukan pencarian dengan melatih model menggunakan data pelatihan dan validasi untuk mengevaluasi performa setiap kombinasi hyperparameter. Pendekatan ini bertujuan menemukan konfigurasi model yang optimal secara otomatis sehingga meningkatkan akurasi prediksi.

```

4. Mencari model terbaik

tuner_lstm = kt.RandomSearch(
    build_lstm_model,
    objective='val_accuracy',
    max_trials=3,
    executions_per_trial=1,
    directory='lstm_tuner',
    project_name='sentiment_lstm'
)

tuner_lstm.search(x_train, y_train, epochs=5, validation_data=(x_test, y_test))

```

Gambar 4. 49 Training hyperparameter terbaik LSTM

4.5.7 Pemodelan CNN

Pada gambar 4.50, terlihat kode Python yang memuat berbagai pustaka penting untuk proyek machine learning dan pemrosesan data. Kode ini mengimpor pustaka seperti pandas dan numpy untuk manipulasi data, matplotlib dan seaborn untuk visualisasi, serta sklearn untuk pemodelan dan evaluasi. Selain itu, ada juga impor dari TensorFlow dan Keras yang digunakan untuk membangun dan melatih model deep learning, khususnya untuk pemrosesan teks dengan modul seperti Tokenizer dan pad_sequences. Secara keseluruhan, kode ini menyiapkan fondasi yang lengkap untuk analisis data dan pengembangan model pembelajaran mesin berbasis teks.

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Embedding, LSTM, Dense, SpatialDropout1D,
                                     Bidirectional, Dropout, Conv1D, MaxPooling1D, GlobalMaxPooling1D)
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical

```

Gambar 4. 50 Tahap persiapan Pustaka python CNN

Pada gambar 4.51, ditampilkan sebuah model Convolutional Neural Network (CNN) yang dibangun menggunakan arsitektur Sequential. Model ini diawali dengan layer embedding yang mengubah input berdimensi 5000 menjadi representasi vektor berdimensi 128 dengan panjang input 200. Selanjutnya, terdapat layer Conv1D dengan 128 filter dan ukuran kernel 5 yang menggunakan fungsi aktivasi ReLU, diikuti oleh MaxPooling1D dengan ukuran pool 2 untuk mereduksi dimensi fitur. Model ini kemudian menggunakan GlobalMaxPooling1D sebelum

dihubungkan ke layer dense dengan 3 unit dan fungsi aktivasi softmax untuk klasifikasi multi-kelas.

```
cnn_model = Sequential([
    Embedding(input_dim=5000, output_dim=128, input_length=200),
    Conv1D(128, 5, activation='relu'),
    MaxPooling1D(pool_size=2),
    GlobalMaxPooling1D(),
    Dense(2, activation='softmax')
])
```

Gambar 4. 51 Pemodelan CNN

Pada gambar 4.52, terlihat sebuah kode Python yang melakukan pelatihan model untuk CNN menggunakan loop. Setiap model dikompilasi dengan fungsi loss `categorical_crossentropy`, optimizer Adam, dan metrik akurasi untuk evaluasi. Proses pelatihan dilakukan selama 5 epoch dengan batch size 64, serta menggunakan data validasi untuk memantau performa model. Hasil pelatihan dari masing-masing model disimpan dalam dictionary `history_dict` untuk analisis lebih lanjut.

```
models = {'CNN': cnn_model}
history_dict = {}
for name, model in models.items():
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    print(f'\nTraining {name} model...')
    history = model.fit(
        X_train, y_train,
        epochs=5,
        batch_size=64,
        validation_data=(X_test, y_test),
        verbose=1
    )
    history_dict[name] = history
```

Gambar 4. 52 Perulangan Kompilasi & Pelatihan CNN

4.5.8 Pemodelan CNN Class weights

Pada gambar 4.53 terlihat potongan kode Python yang memuat berbagai pustaka penting untuk analisis data dan pembelajaran mesin. Pustaka seperti `pandas` dan `numpy` digunakan untuk manipulasi data dan perhitungan numerik, sedangkan `matplotlib` dan `seaborn` berfungsi untuk visualisasi data secara grafis. Selain itu, terdapat juga modul dari `scikit-learn`, `tensorflow`, dan `keras` yang mendukung pembangunan dan pelatihan model machine learning serta deep learning. Dengan mengimpor modul-modul ini, kode tersebut menyiapkan lingkungan yang lengkap untuk melakukan eksplorasi data, pemodelan, dan evaluasi hasil secara menyeluruh.


```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.utils.class_weight import compute_class_weight

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Embedding, LSTM, Dense, SpatialDropout1D,
                                     Bidirectional, Dropout, Conv1D, MaxPooling1D, GlobalMaxPooling1D)
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical

```

Gambar 4. 53 Tahap Persiapan Pustaka python CNN Class weights

Pada gambar 4.54 terlihat potongan kode Python yang menggunakan fungsi `compute_class_weight` dari pustaka scikit-learn untuk menghitung bobot kelas pada dataset yang tidak seimbang. Kode ini menampilkan hasil bobot kelas dalam bentuk angka desimal yang sesuai dengan label kelas 0, dan 1W. Komentar dalam bahasa Indonesia menjelaskan tujuan dari perhitungan bobot ini, yaitu untuk mengatasi ketidakseimbangan data agar model dapat belajar secara lebih efektif. Dengan menggunakan bobot kelas ini, proses pelatihan model menjadi lebih adil terhadap kelas minoritas sehingga meningkatkan akurasi prediksi.

```

# Hitung class weights untuk menangani imbalance
class_weights = compute_class_weight(class_weight='balanced', classes=np.unique(y_train_labels), y=y_train_labels)
class_weight_dict = dict(enumerate(class_weights))
print("Class weights:", class_weight_dict)

Class weights: {0: np.float64(0.6337135614702155), 1: np.float64(2.3696682464454977)}

```

Gambar 4. 54 menghitung class weights CNN

Pada gambar 4.55, terlihat bisa melihat bagaimana sebuah model convolutional neural network (CNN) dibuat menggunakan Keras dengan susunan lapisan yang berurutan. Model ini diawali dengan lapisan Embedding yang mengubah teks menjadi representasi vektor dengan dimensi tetap agar bisa diproses lebih lanjut. Setelah itu, lapisan Conv1D berperan mengekstrak pola-pola lokal dari data urutan, kemudian lapisan MaxPooling1D dan GlobalMaxPooling1D membantu menyaring dan merangkum fitur-fitur terpenting dengan mengurangi dimensi data.

```

cnn_model = Sequential([
    Embedding(input_dim=5000, output_dim=128, input_length=200),
    Conv1D(128, 5, activation='relu'),
    MaxPooling1D(pool_size=2),
    GlobalMaxPooling1D(),
    Dense(2, activation='softmax')
])

```

Gambar 4. 55 Definisi model CNN Class weights

Pada gambar 4.56 tersebut menampilkan potongan kode Python yang digunakan untuk melatih model CNN (Convolutional Neural Network) dalam sebuah proyek machine learning. Kode ini menginisialisasi model, kemudian mengompilasinya dengan fungsi loss categorical_crossentropy dan optimizer Adam, serta metrik akurasi untuk evaluasi. Proses pelatihan dilakukan selama 5 epoch dengan batch size 64, menggunakan data pelatihan dan data validasi yang telah disediakan. Selain itu, kode ini juga menerapkan class weights untuk mengatasi ketidakseimbangan kelas dalam data, sehingga model dapat belajar lebih adil terhadap setiap kelas. Hasil pelatihan disimpan dalam sebuah dictionary untuk setiap model yang dilatih, memudahkan analisis performa setelah proses selesai.

```

models = {'CNN': cnn_model}

history_dict = {}
for name, model in models.items():
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    print(f'\nTraining (name) model...')
    history = model.fit(
        X_train, y_train,
        epochs=5,
        batch_size=64,
        validation_data=(X_test, y_test),
        class_weight=class_weight_dict, # Menambahkan class_weight di sini
        verbose=1
    )
    history_dict[name] = history

```

Gambar 4. 56 Perulangan Kompilasi & Pelatihan CNN Class weights

4.5.9 Pemodelan CNN Class weights Hyperparameter

Pada gambar 4.57 terlihat serangkaian perintah import pada bahasa pemrograman Python yang memuat berbagai pustaka penting seperti pandas, numpy, matplotlib, seaborn, scikit-learn, tensorflow, dan keras-tuner. Pustaka-pustaka ini digunakan untuk keperluan pengolahan data, visualisasi, pembangunan model machine learning, serta optimasi hyperparameter. Selain itu, terdapat modul-modul khusus yang mendukung preprocessing data, pembuatan model neural network, dan evaluasi performa model secara menyeluruh.

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.utils.class_weight import compute_class_weight

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Embedding, LSTM, Dense, SpatialDropout1D,
                                     Bidirectional, Dropout, Conv1D, MaxPooling1D, GlobalMaxPooling1D)
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical

!pip install keras-tuner

import keras_tuner as kt

```

Gambar 4. 57 Impor Pustaka python CNN Class weights Hyperparameter

Pada gambar 4.58 terlihat kode Python yang mendefinisikan model Convolutional Neural Network (CNN) menggunakan TensorFlow/Keras. Model ini terdiri dari beberapa layer, termasuk Embedding untuk representasi kata, Conv1D untuk ekstraksi fitur lokal, MaxPooling1D untuk pengurangan dimensi, serta Dense dan Dropout untuk klasifikasi dan pencegahan overfitting. Hyperparameter tuning dilakukan menggunakan objek 'hp' yang memungkinkan penyesuaian parameter seperti jumlah filter dan ukuran kernel secara otomatis. Model ini dikompilasi dengan optimizer Adam dan fungsi loss categorical crossentropy, yang sesuai untuk tugas klasifikasi multi-kelas.

```

def build_lstm_model(hp):
    model = Sequential()
    model.add(Embedding(input_dim=5000, output_dim=hp.Int('embedding_dim', 64, 256, step=32), input_length=200))
    model.add(Bidirectional(LSTM(units=hp.Int('lstm_units', 32, 256, step=32), return_sequences=False)))
    model.add(Dropout(hp.Float('dropout', 0.2, 0.5, step=0.1)))
    model.add(Dense(units=hp.Int('dense_units', 32, 128, step=32), activation='relu'))
    model.add(Dense(2, activation='softmax'))

    model.compile(
        optimizer=tf.keras.optimizers.Adam(hp.Float('learning_rate', 1e-4, 1e-2, sampling='LOG')),
        loss='categorical_crossentropy',
        metrics=['accuracy'])

    return model

```

Gambar 4. 58 Pemodelan Tuning Hyperparameter CNN

Pada gambar 4.59 terlihat kode Python yang mendefinisikan dan menjalankan proses Keras Tuner dengan metode RandomSearch untuk model Convolutional Neural Network (CNN). Kode ini menginisialisasi tuner dengan parameter seperti tujuan optimasi ('val_accuracy'), jumlah percobaan maksimal (max trials), dan nama proyek untuk pengelolaan hasil tuning. Selanjutnya, tuner melakukan pencarian kombinasi hyperparameter terbaik dengan melatih model menggunakan data pelatihan dan validasi.

```

tuner_cnn = kt.RandomSearch(
    build_cnn_model,
    objective='val_accuracy',
    max_trials=3,
    executions_per_trial=1,
    directory='cnn_tuner',
    project_name='sentiment_cnn'
)

tuner_cnn.search(X_train, y_train, epochs=5, validation_data=(X_test, y_test))

```

Gambar 4. 59 Training hyperparameter terbaik CNN

4.6 Evaluation

4.6.1 Hasil Evaluasi BiLSTM

Pada gambar 4.60 ditampilkan log pelatihan model BiLSTM selama 5 epoch yang memperlihatkan peningkatan performa secara signifikan dari awal sampai akhir pelatihan. Akurasi pada data pelatihan meningkat secara konsisten dari sekitar 74,71% pada epoch pertama menjadi hampir sempurna yaitu 99,93% pada epoch kelima. Penurunan nilai loss pelatihan juga sangat drastis, dari 0.5866 pada epoch pertama menjadi sangat rendah yaitu 0.0052 pada epoch terakhir, menunjukkan model semakin baik dalam memprediksi data pelatihan. Akurasi validasi juga menunjukkan tren positif dan stabil, mulai dari 82,30% pada epoch pertama dan mencapai 93,89% pada epoch kelima, dengan nilai loss validasi juga menurun secara umum dari 0.4164 menjadi 0.3030. Meskipun terdapat sedikit fluktuasi pada nilai loss validasi, tidak ada indikasi overfitting yang serius karena akurasi validasi tetap tinggi dan tidak menurun drastis. Hal ini menunjukkan bahwa model berhasil belajar dengan baik dan generalisasi ke data validasi cukup baik selama proses pelatihan.

```

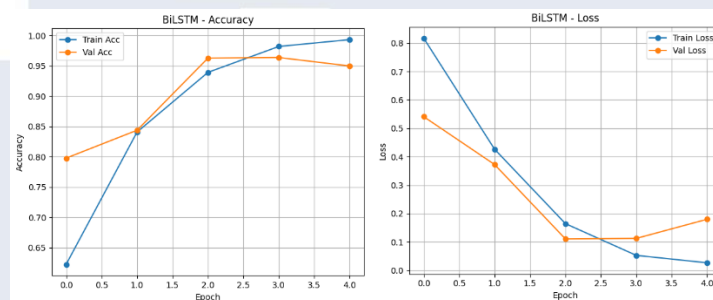
Training BiLSTM model...
Epoch 1/5
52/52 ----- 35s 598ms/step - accuracy: 0.7471 - loss: 0.5866 - val_accuracy: 0.8230 - val_loss: 0.4164
Epoch 2/5
52/52 ----- 27s 514ms/step - accuracy: 0.8797 - loss: 0.2923 - val_accuracy: 0.9084 - val_loss: 0.2289
Epoch 3/5
52/52 ----- 26s 499ms/step - accuracy: 0.9828 - loss: 0.0683 - val_accuracy: 0.9170 - val_loss: 0.3324
Epoch 4/5
52/52 ----- 28s 541ms/step - accuracy: 0.9965 - loss: 0.0159 - val_accuracy: 0.9353 - val_loss: 0.3011
Epoch 5/5
52/52 ----- 41s 526ms/step - accuracy: 0.9993 - loss: 0.0052 - val_accuracy: 0.9389 - val_loss: 0.3030

```

Gambar 4. 60 Hasil proses pelatihan BiLSTM

Pada gambar 4.61 menunjukkan performa model BiLSTM selama 5 epoch dengan dua metrik utama, akurasi dan loss, pada data pelatihan dan validasi. Grafik akurasi yang terus meningkat menandakan bahwa model berhasil meningkatkan kemampuan mengenali pola data secara konsisten, sementara penurunan loss mengindikasikan kesalahan prediksi semakin berkurang. Tidak adanya perbedaan

signifikan antara akurasi dan loss pada data pelatihan dan validasi menunjukkan model ini tidak mengalami overfitting selama proses training. Penurunan loss validasi yang stabil setelah epoch ke-3 juga menggambarkan bahwa model mulai mencapai titik konvergensi yang baik pada dataset ini. Dengan demikian, grafik ini secara keseluruhan mencerminkan BiLSTM berhasil belajar secara efektif dan menghasilkan performa yang dapat diandalkan pada tahap awal pelatihan.



Gambar 4. 61 Plot hasil pengujian BiLSTM

Pada gambar 4.62 ditampilkan laporan klasifikasi dari model BiLSTM yang digunakan sebelum dilakukan penanganan masalah imbalance pada data. Model ini menunjukkan kinerja yang sangat baik pada kelas negatif dengan nilai precision dan recall yang cukup tinggi, yaitu 0.93 dan 0.96, menghasilkan f1-score sebesar 0.95 dari total 799 sampel. Namun, pada kelas positif performa model terlihat lebih rendah dengan precision sebesar 0.82 dan recall sebesar 0.72, sehingga nilai f1-score juga menurun menjadi 0.77 dari 201 sampel. Nilai akurasi keseluruhan model mencapai 0.91, dengan rata-rata makro f1-score sebesar 0.86, serta weighted average f1-score sebesar 0.91, yang menunjukkan performa model cukup baik secara umum meskipun terdapat perbedaan kemampuan pada masing-masing kelas. Hal ini mengindikasikan kemungkinan model masih kurang optimal dalam mengenali kelas positif yang merupakan kelas minoritas, sehingga penanganan imbalance data tetap diperlukan untuk meningkatkan kualitas prediksi pada kelas tersebut.

BiLSTM Classification Report:				
26/26		2s	69ms/step	
	precision	recall	f1-score	support
negatif	0.95	0.97	0.96	610
positif	0.90	0.86	0.88	209
accuracy			0.94	819
macro avg	0.93	0.91	0.92	819
weighted avg	0.94	0.94	0.94	819

Gambar 4. 62 Hasil Laporan Klasifikasi Model BiLSTM

4.6.2 Hasil Evaluasi BiLSTM Class weights

Pada gambar 4.63 ditampilkan log pelatihan model BiLSTM selama 5 epoch yang memperlihatkan peningkatan performa yang konsisten sepanjang pelatihan. Akurasi pada data pelatihan meningkat secara signifikan dari sekitar 49,19% pada epoch pertama menjadi hampir sempurna yaitu 99,59% pada epoch kelima. Nilai loss pelatihan menurun drastis dari 0.6926 pada epoch pertama menjadi sangat rendah yaitu 0.0121 pada epoch terakhir, yang menunjukkan kemampuan model dalam mengurangi kesalahan prediksi pada data pelatihan. Akurasi validasi juga menunjukkan tren yang positif dan stabil, mulai dari 75,70% pada epoch pertama serta mencapai 92,92% pada epoch kelima. Nilai loss validasi secara umum juga menurun dari 0.5264 menjadi sekitar 0.3357, meskipun terdapat beberapa fluktuasi kecil yang wajar pada proses training. Selain itu, informasi waktu per langkah pelatihan yang tercatat menunjukkan efisiensi proses training dengan durasi yang relatif singkat pada setiap epoch. Semua metrik ini menandakan bahwa model BiLSTM semakin baik dalam mempelajari pola dari data pelatihan dan mampu melakukan generalisasi yang baik pada data validasi.

```

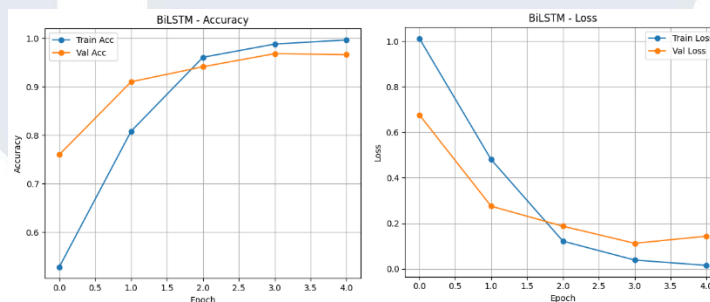
Training BiLSTM model...
Epoch 1/5 35s 516ms/step - accuracy: 0.4919 - loss: 0.6926 - val_accuracy: 0.7570 - val_loss: 0.5264
Epoch 2/5 36s 448ms/step - accuracy: 0.7993 - loss: 0.4874 - val_accuracy: 0.8742 - val_loss: 0.2863
Epoch 3/5 42s 471ms/step - accuracy: 0.9445 - loss: 0.1485 - val_accuracy: 0.8852 - val_loss: 0.3027
Epoch 4/5 41s 462ms/step - accuracy: 0.9841 - loss: 0.0411 - val_accuracy: 0.9267 - val_loss: 0.2983
Epoch 5/5 25s 483ms/step - accuracy: 0.9959 - loss: 0.0121 - val_accuracy: 0.9292 - val_loss: 0.3357

```

Gambar 4. 63 Hasil proses pelatihan BiLSTM dengan Class weights

Pada gambar 4.64 ditampilkan dua grafik yang menggambarkan performa model BiLSTM selama proses pelatihan (training) dan validasi (validation) dalam

5 epoch. Grafik akurasi menunjukkan bahwa model mengalami peningkatan performa secara konsisten untuk data pelatihan maupun validasi, dimana nilai akurasi validasi mulai sekitar 0.78 pada epoch pertama dan naik hingga mendekati 0.98 di epoch keempat. Hal ini menandakan model mampu mengenali pola data dengan baik dan melakukan generalisasi yang efektif pada data validasi. Sementara itu, grafik loss memperlihatkan penurunan yang tajam pada awal pelatihan, baik pada data training maupun validasi, menunjukkan bahwa kesalahan prediksi model semakin berkurang secara signifikan. Meski pada epoch kelima terdapat sedikit kenaikan nilai loss validasi, hal ini masih dalam batas yang wajar dan tidak mengindikasikan overfitting yang parah, sehingga bisa disimpulkan bahwa model berhasil belajar dengan baik tanpa kehilangan kemampuan generalisasi.



Gambar 4. 64 Plot hasil pengujian BILSTM Class Weights

Pada gambar 4.65 ditampilkan laporan klasifikasi hasil pemodelan BiLSTM setelah penerapan penanganan imbalance dengan metode Class Weight. Model menunjukkan performa yang sangat baik pada kelas negatif dengan precision sebesar 0.93 dan recall yang sangat tinggi yaitu 0.98, menghasilkan f1-score yang unggul yaitu 0.95 dari total 610 sampel. Di sisi kelas positif, model berhasil mempertahankan precision yang tinggi sebesar 0.94 dengan recall sebesar 0.78, sehingga f1-score tercatat 0.85 dari 209 sampel. Akurasi keseluruhan model mencapai 0.93, dengan rata-rata makro f1-score sebesar 0.90 dan weighted average f1-score sebesar 0.93. Hasil ini mengindikasikan bahwa penanganan imbalance dengan pemberian bobot kelas efektif dalam menjaga keseimbangan dan meningkatkan kemampuan model dalam mengenali kedua kelas, terutama kelas minoritas, dengan performa yang konsisten dan seimbang.

BiLSTM Classification Report:				
26/26	2s 80ms/step			
	precision	recall	f1-score	support
negatif	0.93	0.98	0.95	610
positif	0.94	0.78	0.85	209
accuracy			0.93	819
macro avg	0.93	0.88	0.90	819
weighted avg	0.93	0.93	0.93	819

Gambar 4. 65 Hasil Laporan Klasifikasi Model BiLSTM Class Weights

4.6.3. Hasil Evaluasi BiLSTM Class weights Hyperparameter

Pada gambar 4.66 terlihat proses evaluasi model terbaik yang telah diperoleh melalui tuning menggunakan Keras. Model diuji menggunakan data uji untuk mengukur performanya dengan metrik akurasi dan loss. Hasil evaluasi menunjukkan akurasi sebesar 0.9243 dengan nilai loss sebesar 0.3056, menandakan model memiliki kemampuan prediksi yang baik meskipun tidak setinggi yang diharapkan pada beberapa eksperimen lainnya. Langkah evaluasi ini penting untuk memastikan bahwa model yang dihasilkan benar-benar optimal dan siap digunakan untuk prediksi pada data baru, meskipun ada indikasi bahwa performa model dapat ditingkatkan dengan tuning lebih lanjut atau penyesuaian parameter.

```

5. Dapatkan dan evaluasi model terbaik

▶ best_model = tuner.get_best_models(num_models=1)[0]
  best_hyperparameters = tuner.get_best_hyperparameters(num_trials=1)[0]

  loss, accuracy = best_model.evaluate(X_test, y_test)
  print(f"Akurasi setelah tuning: {accuracy:.4f}")

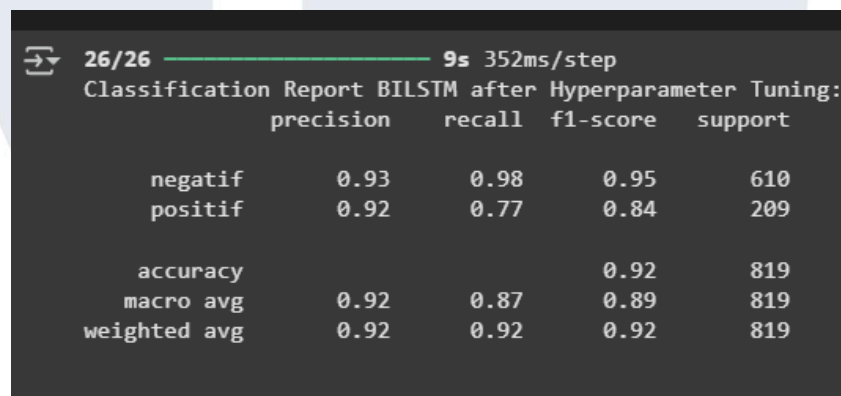
📄 /usr/local/lib/python3.11/dist-packages/keras/src/saving/saving_lib.py:757: UserWa
  saveable.load_own_variables(weights_store.get(inner_path))
26/26 8s 272ms/step - accuracy: 0.9062 - loss: 0.3056
Akurasi setelah tuning: 0.9243

```

Gambar 4. 66 Akurasi BiLSTM setelah tuning

Pada gambar 4.67 ditampilkan laporan klasifikasi hasil pemodelan BiLSTM setelah penerapan penanganan imbalance menggunakan Class Weight dan pengoptimalan hyperparameter. Model menunjukkan performa yang sangat baik pada kelas negatif dengan precision sebesar 0.93, recall yang sangat tinggi yaitu

0.98, dan menghasilkan f1-score unggul sebesar 0.95 dari total 610 sampel. Sedangkan pada kelas positif, model memiliki precision yang cukup tinggi yaitu 0.92 dengan recall sebesar 0.77, sehingga f1-score tercatat sebesar 0.84 dari 209 sampel. Akurasi keseluruhan model mencapai 0.92, dengan rata-rata makro f1-score sebesar 0.89 dan weighted average f1-score sebesar 0.92. Hasil ini menunjukkan adanya peningkatan performa dan stabilitas setelah kedua tahap tersebut, yang menandakan keberhasilan tuning hyperparameter serta pemberian bobot kelas dalam meningkatkan keseimbangan dan efektivitas model dalam mengenali kedua kelas, terutama dalam pengenalan kelas minoritas.



```

26/26 ————— 9s 352ms/step
Classification Report BiLSTM after Hyperparameter Tuning:
      precision    recall  f1-score   support

negatif      0.93      0.98      0.95        610
positif      0.92      0.77      0.84        209

accuracy          0.92          0.92          0.92          819
macro avg      0.92      0.87      0.89          819
weighted avg   0.92      0.92      0.92          819

```

Gambar 4. 67 Hasil Laporan Klasifikasi Model BiLSTM Class Weights Hyperparameter

4.6.4 Hasil Evaluasi LSTM

Pada gambar 4.68 terlihat proses pelatihan model LSTM selama 5 epoch yang menunjukkan peningkatan akurasi dan penurunan nilai loss secara bertahap. Akurasi pada data pelatihan meningkat signifikan dari 73,15% pada epoch pertama menjadi hampir sempurna yaitu 99,51% pada epoch kelima. Nilai loss juga menurun drastis dari 0,5952 menjadi sangat rendah yaitu 0,0222 pada epoch terakhir. Pada data validasi, akurasi mengalami peningkatan yang stabil, mencapai 93,41% di epoch kelima, sedangkan nilai loss validasi menurun dari awal yaitu 0,4013 menjadi 0,2184. Hasil ini menandakan bahwa model LSTM mampu belajar dengan baik, menjaga kestabilan performa pada data validasi, dan menunjukkan kemampuan generalisasi yang cukup baik pada data yang belum pernah dilihat sebelumnya.

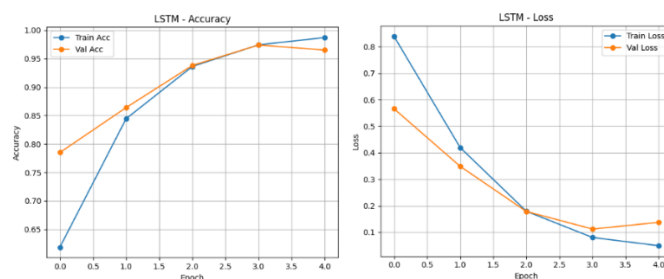
```

Training LSTM model...
Epoch 1/5
52/52 ----- 43s 730ms/step - accuracy: 0.7315 - loss: 0.5952 - val_accuracy: 0.8846 - val_loss: 0.4013
Epoch 2/5
52/52 ----- 39s 697ms/step - accuracy: 0.8756 - loss: 0.3165 - val_accuracy: 0.8571 - val_loss: 0.3636
Epoch 3/5
52/52 ----- 41s 702ms/step - accuracy: 0.9435 - loss: 0.1562 - val_accuracy: 0.9194 - val_loss: 0.2382
Epoch 4/5
52/52 ----- 42s 721ms/step - accuracy: 0.9883 - loss: 0.0484 - val_accuracy: 0.9328 - val_loss: 0.2200
Epoch 5/5
52/52 ----- 41s 720ms/step - accuracy: 0.9951 - loss: 0.0222 - val_accuracy: 0.9341 - val_loss: 0.2184

```

Gambar 4. 68 Hasil proses pelatihan LSTM

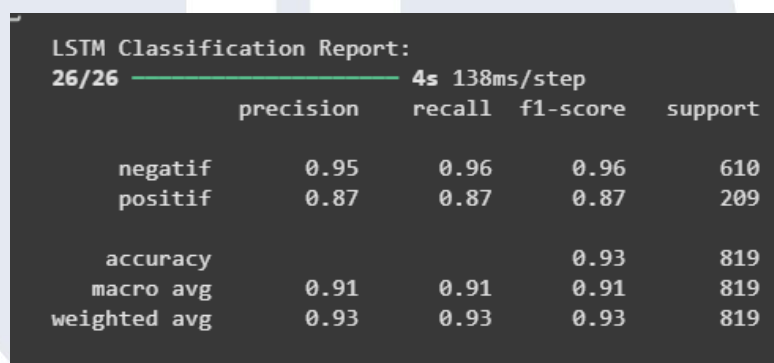
Pada gambar 4.69, dua grafik garis memperlihatkan performa model LSTM selama lima epoch pelatihan. Grafik akurasi penunjukkan adanya peningkatan yang konsisten pada kedua data pelatihan dan validasi, yang berarti model secara bertahap dapat mengenali pola data lebih baik seiring waktu. Hal ini menunjukkan bahwa proses training berjalan efektif dan model tidak mengalami kesulitan belajar pada epoch awal. Grafik loss yang menurun secara signifikan pada data training dan juga validasi mengindikasikan bahwa kesalahan prediksi berkurang, memperkuat bahwa model semakin fit terhadap data. Selain itu, tidak adanya lonjakan drastis pada loss validasi menandakan model tidak mengalami overfitting signifikan, sehingga dapat digeneralisasi dengan baik pada data baru. Keseluruhan hasil ini menggambarkan bahwa model LSTM berhasil mencapai keseimbangan antara kemampuan belajar dan generalisasi dalam rentang epoch yang dipelajari.



Gambar 4. 69 Plot hasil pengujian LSTM

Pada gambar 4.70 ditampilkan laporan klasifikasi hasil pemodelan LSTM yang dilakukan sebelum penerapan penanganan imbalance pada data. Model ini menunjukkan performa yang sangat baik pada kedua kelas dengan nilai precision dan recall yang tinggi. Untuk kelas negatif, precision tercatat sebesar 0.95 dan recall mencapai 0.96, menghasilkan f1-score sangat baik yaitu 0.96 dari total 610 sampel, menandakan model mampu mengenali data negatif dengan sangat andal dan konsisten. Pada kelas positif, precision mencapai 0.87 dan recall juga 0.87,

menghasilkan f1-score sebesar 0.87 dari 209 sampel, yang memperlihatkan kemampuan model dalam mendeteksi kelas positif secara seimbang dan stabil. Akurasi keseluruhan model mencapai 0.93, dengan rata-rata makro f1-score sebesar 0.91 serta weighted average f1-score sebesar 0.93. Hasil ini menunjukkan performa yang sangat memadai dan keseimbangan model yang baik pada kedua kelas, serta indikasi model sudah cukup sensitif dalam mengenali kelas minoritas meskipun belum menerapkan teknik penanganan imbalance.



```

LSTM Classification Report:
26/26 ————— 4s 138ms/step

```

	precision	recall	f1-score	support
negatif	0.95	0.96	0.96	610
positif	0.87	0.87	0.87	209
accuracy			0.93	819
macro avg	0.91	0.91	0.91	819
weighted avg	0.93	0.93	0.93	819

Gambar 4. 70 Hasil Laporan Klasifikasi Model LSTM

4.6.5 Hasil Evaluasi LSTM Class weights

Pada gambar 4.71 ditampilkan proses pelatihan model LSTM selama 5 epoch yang mencakup metrik akurasi, loss, serta akurasi dan loss pada data validasi. Terlihat bahwa nilai akurasi pada data pelatihan meningkat secara signifikan dari 55,61% pada epoch pertama menjadi hampir sempurna yaitu 99,37% pada epoch kelima. Sebaliknya, nilai loss pelatihan menurun drastis dari 0,6847 menjadi sangat kecil yaitu 0,0264, menandakan model berhasil meminimalkan kesalahan prediksi secara efektif. Pada data validasi, akurasi validasi juga menunjukkan peningkatan yang stabil dari 81,20% di awal hingga mencapai 94,02% pada epoch terakhir. Sementara itu, nilai loss validasi menurun dari 0,4464 menjadi 0,2243, mengindikasikan peningkatan kesesuaian model terhadap data yang belum pernah dilihat. Waktu per langkah pelatihan rata-rata sekitar 600 ms memberikan gambaran efisiensi pelatihan yang cukup baik. Hasil ini menunjukkan bahwa model LSTM mampu belajar secara efektif, meningkatkan performa secara konsisten selama pelatihan, serta mempertahankan kestabilan dan kemampuan generalisasi pada data validasi.

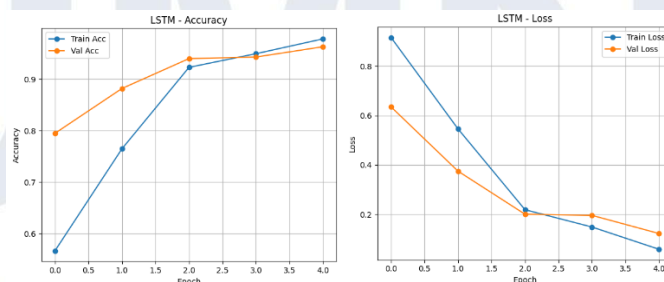
```

Training LSTM model...
Epoch 1/5
52/52 ----- 38s 630ms/step - accuracy: 0.5561 - loss: 0.6847 - val_accuracy: 0.8120 - val_loss: 0.4464
Epoch 2/5
52/52 ----- 40s 605ms/step - accuracy: 0.8796 - loss: 0.3291 - val_accuracy: 0.8852 - val_loss: 0.2697
Epoch 3/5
52/52 ----- 42s 626ms/step - accuracy: 0.9554 - loss: 0.1167 - val_accuracy: 0.9316 - val_loss: 0.2102
Epoch 4/5
52/52 ----- 41s 615ms/step - accuracy: 0.9902 - loss: 0.0442 - val_accuracy: 0.9292 - val_loss: 0.2083
Epoch 5/5
52/52 ----- 32s 613ms/step - accuracy: 0.9937 - loss: 0.0264 - val_accuracy: 0.9402 - val_loss: 0.2243

```

Gambar 4. 71 Hasil proses pelatihan LSTM dengan Class weights

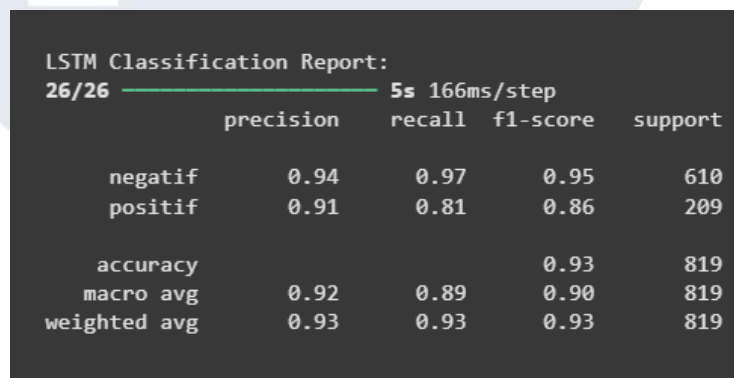
Pada gambar 4.72, terdapat dua grafik garis yang memperlihatkan performa model LSTM selama proses pelatihan dan validasi dalam beberapa epoch. Grafik pertama di sebelah kiri menunjukkan tren kenaikan akurasi baik pada data pelatihan maupun validasi, yang mengindikasikan bahwa model secara konsisten semakin mampu mengenali pola dan fitur dalam data seiring berjalannya pelatihan. Hal ini menjadi indikasi positif bahwa model belajar dengan efektif dan tidak mengalami kesulitan mempelajari data. Pada grafik kedua di sebelah kanan, penurunan nilai loss pada kedua data pelatihan dan validasi memperkuat temuan bahwa model semakin optimal dalam meminimalkan kesalahan prediksi. Selain itu, perhatikan bahwa garis loss validasi tidak menunjukkan lonjakan naik yang signifikan, menandakan model ini relatif stabil dan tidak mengalami overfitting yang serius selama rentang epoch ini. Dengan demikian, performa model di gambar ini mencerminkan proses pelatihan yang baik dengan peningkatan generalisasi yang seimbang antara data pelatihan dan validasi, namun tentu evaluasi lanjutan tetap diperlukan untuk memastikan konsistensi hasil.



Gambar 4. 72 Plot hasil pengujian LSTM Class Weights

Pada gambar 4.73 ditampilkan laporan klasifikasi hasil pemodelan LSTM setelah penerapan penanganan imbalance pada data pelatihan menggunakan metode Class Weights. Model menunjukkan performa yang sangat baik pada kelas negatif dengan precision sebesar 0.94, recall mencapai 0.97, dan f1-score sangat tinggi

yaitu 0.95 dari total 610 sampel, yang menandakan kemampuan model dalam mengenali data negatif dengan sangat andal dan konsisten. Sedangkan pada kelas positif, model memperoleh precision 0.91, recall 0.81, dan f1-score 0.86 dari 209 sampel, yang menunjukkan terjadi peningkatan sensitivitas (recall) kelas positif dibanding beberapa model sebelumnya, menandakan model berhasil mengenali lebih banyak data positif meskipun ada sedikit penurunan recall dibandingkan kelas negatif. Akurasi keseluruhan model tercatat 0.93, dengan rata-rata makro f1-score sebesar 0.90 serta weighted average f1-score sebesar 0.93, yang menggambarkan keseimbangan performa yang lebih baik antara kelas mayoritas dan minoritas. Hasil ini memperlihatkan bahwa penerapan Class Weights efektif dalam membantu model meningkatkan ketelitian terutama pada kelas minoritas tanpa mengurangi performa keseluruhan.



LSTM Classification Report:				
	precision	recall	f1-score	support
negatif	0.94	0.97	0.95	610
positif	0.91	0.81	0.86	209
accuracy			0.93	819
macro avg	0.92	0.89	0.90	819
weighted avg	0.93	0.93	0.93	819

Gambar 4. 73 Hasil Laporan Klasifikasi Model LSTM Class Weights

4.6.6 Hasil Evaluasi LSTM Class weights Hyperparameter

Pada gambar 4.74 ditampilkan potongan kode Python yang digunakan untuk mendapatkan model LSTM terbaik setelah proses tuning hyperparameter dengan memanggil `tuner_lstm.get_best_models(num_models=1)[0]`. Setelah model terbaik diperoleh, model tersebut dievaluasi pada data uji (`X_test` dan `y_test`) dengan metode `evaluate()`, yang menghasilkan nilai loss dan akurasi. Hasil akurasi model pada data validasi dicetak dengan format yang menunjukkan akurasi sampai 4 angka di belakang koma, yaitu sebesar 0.9206 atau 92,06%. Waktu evaluasi model tercatat relatif singkat, yaitu sekitar 3 detik dengan kecepatan 66 ms per langkah. Gambar ini menggambarkan tahap akhir dari proses pelatihan dan tuning model, di

mana performa model diuji secara objektif pada data yang belum pernah dilihat untuk memastikan kualitas prediksi model sebelum digunakan lebih lanjut.

```
5. Dapatkan dan evaluasi model terbaik

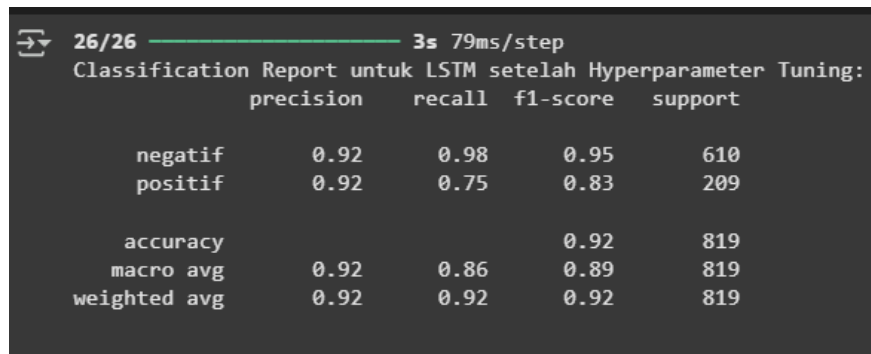
# Mendapatkan model terbaik
best_lstm_model = tuner_lstm.get_best_models(num_models=1)[0]

# Evaluasi model
loss, accuracy = best_lstm_model.evaluate(X_test, y_test)
print(f"Akurasi LSTM setelah tuning: {accuracy:.4f}")

/usr/local/lib/python3.11/dist-packages/keras/src/saving/saving_lib.py:757: UserWarning:
saveable.load_own_variables(weights_store.get(inner_path))
26/26 ————— 3s 66ms/step - accuracy: 0.9043 - loss: 0.3483
Akurasi LSTM setelah tuning: 0.9206
```

Gambar 4. 74 Akurasi LSTM setelah tuning

Pada gambar 4.75 ditampilkan laporan klasifikasi model LSTM setelah melalui dua tahap penting, yaitu penerapan penanganan ketidakseimbangan kelas (imbalance) dengan metode Class Weights serta penyempurnaan performa melalui proses tuning hyperparameter. Model menunjukkan performa yang kuat dengan precision sebesar 0.92 pada kedua kelas negatif dan positif, yang berarti model memiliki ketepatan tinggi dalam memprediksi kedua kelas tersebut. Recall untuk kelas negatif sangat tinggi, yakni 0.98, yang menandakan model sangat efektif mengenali data negatif, sementara recall kelas positif tercatat 0.75, menunjukkan kemampuan yang baik dalam mendeteksi data positif meskipun masih ada ruang untuk perbaikan. Nilai f1-score mencapai 0.95 untuk kelas negatif dan 0.83 untuk kelas positif, mencerminkan keseimbangan antara precision dan recall yang cukup baik terutama pada kelas minoritas. Akurasi keseluruhan model adalah 0.92, dengan macro average f1-score sebesar 0.89 dan weighted average f1-score 0.92, yang mengindikasikan kestabilan dan konsistensi performa model secara keseluruhan. Hasil ini memperlihatkan keberhasilan strategi penggabungan penanganan imbalance dengan tuning hyperparameter dalam meningkatkan kinerja model, khususnya dalam hal memahami dan mengklasifikasikan kelas mayoritas dan minoritas dengan lebih seimbang dan optimal.



```

26/26 ————— 3s 79ms/step
Classification Report untuk LSTM setelah Hyperparameter Tuning:

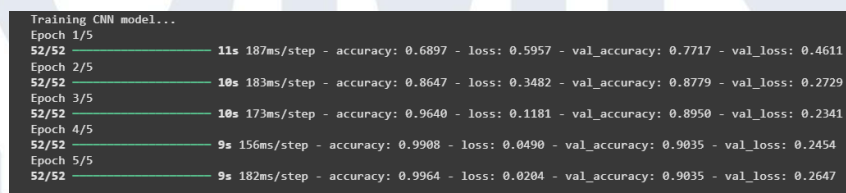
```

	precision	recall	f1-score	support
negatif	0.92	0.98	0.95	610
positif	0.92	0.75	0.83	209
accuracy			0.92	819
macro avg	0.92	0.86	0.89	819
weighted avg	0.92	0.92	0.92	819

Gambar 4. 75 Hasil Laporan Klasifikasi Model LSTM Class Weights Hyperparameter

4.6.7 Hasil Evaluasi CNN

Pada gambar 4.76 terlihat proses pelatihan model CNN yang mencatatkan peningkatan signifikan pada metrik akurasi dan penurunan loss di setiap epoch, baik pada data pelatihan maupun validasi. Pada epoch pertama, akurasi model masih relatif rendah, yakni sekitar 0.6897 dengan loss sebesar 0.5957, menunjukkan bahwa model masih dalam tahap awal pembelajaran. Namun, seiring bertambahnya epoch, akurasi model meningkat secara konsisten menjadi 0.9964 pada epoch ke-5, dengan penurunan loss hingga mencapai 0.0204 yang sangat rendah, menandakan model mulai sangat matang dalam mengenali pola data pelatihan. Validasi akurasi juga memperlihatkan tren positif, dengan nilai yang stabil di kisaran 0.8950 sampai 0.9035 sepanjang epoch ke-3 hingga ke-5, meskipun nilai validasi loss sedikit berfluktuasi, model tetap menunjukkan tidak adanya indikasi overfitting yang berarti. Secara keseluruhan, proses pelatihan ini menggambarkan bahwa model CNN berhasil belajar dan meningkatkan performanya secara signifikan dan konsisten selama 5 epoch.



```

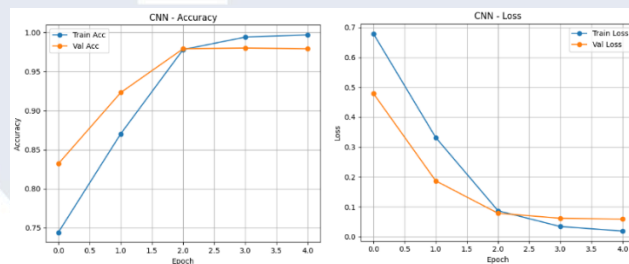
Training CNN model1...
Epoch 1/5
52/52 ————— 11s 187ms/step - accuracy: 0.6897 - loss: 0.5957 - val_accuracy: 0.7717 - val_loss: 0.4611
Epoch 2/5
52/52 ————— 10s 183ms/step - accuracy: 0.8647 - loss: 0.3482 - val_accuracy: 0.8779 - val_loss: 0.2729
Epoch 3/5
52/52 ————— 10s 173ms/step - accuracy: 0.9640 - loss: 0.1181 - val_accuracy: 0.8950 - val_loss: 0.2341
Epoch 4/5
52/52 ————— 9s 156ms/step - accuracy: 0.9908 - loss: 0.0490 - val_accuracy: 0.9035 - val_loss: 0.2454
Epoch 5/5
52/52 ————— 9s 182ms/step - accuracy: 0.9964 - loss: 0.0204 - val_accuracy: 0.9035 - val_loss: 0.2647

```

Gambar 4. 76 Hasil proses pelatihan CNN

Pada gambar 4.77 ditampilkan dua grafik garis yang memperlihatkan performa model Convolutional Neural Network (CNN) selama proses pelatihan dalam beberapa epoch, dengan fokus pada metrik akurasi dan loss baik untuk data

pelatihan (train) maupun data validasi (val). Grafik akurasi menunjukkan tren kenaikan yang konsisten pada kedua dataset, di mana akurasi pelatihan mulai dari sekitar 0.75 pada epoch awal dan terus meningkat hingga mendekati 1.00 pada epoch ke-4, sementara akurasi validasi juga mengikuti pola serupa dengan nilai yang sedikit lebih tinggi pada awalnya dan akhirnya hampir menyamai akurasi pelatihan, menandakan kemampuan model yang bagus dalam generalisasi data. Pada sisi lain, grafik loss memperlihatkan penurunan drastis sejak epoch pertama, dari sekitar 0.7 hingga turun mendekati 0.0 pada epoch ke-4 untuk kedua data pelatihan dan validasi, hal ini menunjukkan bahwa model semakin efisien dalam mengurangi kesalahan prediksi seiring waktu pelatihan.



Gambar 4. 77 Plot hasil pengujian CNN

Pada gambar 4.78 terlihat laporan klasifikasi dari model CNN yang menunjukkan performa yang baik pada data uji sebanyak 819 sampel. Model memiliki nilai precision untuk kelas negatif sebesar 0.93 dan kelas positif sebesar 0.83, yang berarti model cukup akurat dalam memprediksi kelas negatif dan agak lebih rendah untuk kelas positif. Recall untuk kelas negatif tercatat 0.94, sedangkan untuk kelas positif sebesar 0.78, menunjukkan bahwa model lebih mampu mengenali data negatif dibandingkan data positif. F1-score untuk kelas negatif adalah 0.94 dan untuk kelas positif adalah 0.81, yang mencerminkan keseimbangan antara precision dan recall pada masing-masing kelas. Akurasi keseluruhan model mencapai 0.90, dengan nilai macro average dan weighted average pada precision, recall, dan f1-score berada di kisaran 0.86 - 0.90, mengindikasikan performa model yang cukup stabil namun ada ketidakseimbangan dalam prediksi kelas, terutama pada kelas positif.

CNN Classification Report:				
26/26		1s 23ms/step		
	precision	recall	f1-score	support
negatif	0.93	0.94	0.94	610
positif	0.83	0.78	0.81	209
accuracy			0.90	819
macro avg	0.88	0.86	0.87	819
weighted avg	0.90	0.90	0.90	819

Gambar 4. 78 Hasil Laporan Klasifikasi Model CNN

4.6.8 Hasil Evaluasi CNN Class weights

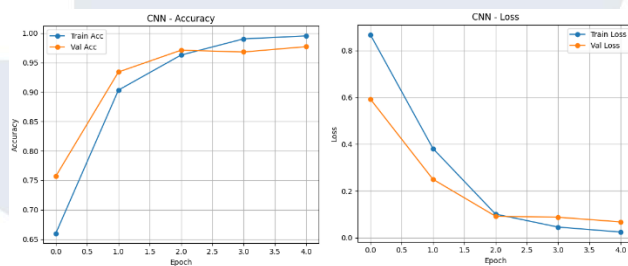
Pada gambar 4.79 terlihat proses pelatihan model CNN selama 5 epoch yang menunjukkan peningkatan yang signifikan pada metrik akurasi dan penurunan nilai loss baik pada data pelatihan maupun validasi. Pada epoch pertama, akurasi model masih berada pada angka 0.5945 dengan loss sebesar 0.6709, yang menandakan model berada pada tahap awal pembelajaran dengan kemampuan pengenalan pola yang masih terbatas. Seiring bertambahnya epoch, akurasi meningkat secara konsisten mencapai 0.9981 di epoch kelima, dengan loss menurun drastis menjadi 0.0203, yang menunjukkan model mulai sangat matang dalam mempelajari data pelatihan. Akurasi validasi juga menunjukkan peningkatan dari 0.7827 pada epoch pertama menjadi 0.9048 pada epoch terakhir, meskipun nilai validasi loss sedikit berfluktuasi, menunjukkan bahwa model cukup baik dalam menggeneralisasi ke data yang belum dilihat sebelumnya. Secara keseluruhan, grafik pelatihan ini mencerminkan proses pelatihan yang efektif dengan performa yang meningkat pesat dari epoch ke epoch dan tanpa tanda-tanda overfitting yang signifikan.

Training CNN model...				
Epoch 1/5				
52/52	10s 174ms/step	- accuracy: 0.5945	- loss: 0.6709	- val_accuracy: 0.7827 - val_loss: 0.4801
Epoch 2/5				
52/52	10s 168ms/step	- accuracy: 0.8726	- loss: 0.3442	- val_accuracy: 0.8803 - val_loss: 0.2744
Epoch 3/5				
52/52	10s 169ms/step	- accuracy: 0.9613	- loss: 0.1199	- val_accuracy: 0.8926 - val_loss: 0.2521
Epoch 4/5				
52/52	9s 143ms/step	- accuracy: 0.9855	- loss: 0.0491	- val_accuracy: 0.8889 - val_loss: 0.2518
Epoch 5/5				
52/52	10s 146ms/step	- accuracy: 0.9981	- loss: 0.0203	- val_accuracy: 0.9048 - val_loss: 0.2571

Gambar 4. 79 Hasil proses pelatihan CNN dengan Class weights

Pada gambar 4.80 ditampilkan dua grafik garis yang berdampingan yang memperlihatkan performa model Convolutional Neural Network (CNN) selama

proses pelatihan dalam beberapa epoch, dengan fokus pada metrik akurasi dan loss baik untuk data pelatihan (train) maupun data validasi (val). Grafik sebelah kiri menunjukkan peningkatan akurasi yang konsisten pada kedua dataset seiring bertambahnya epoch, di mana akurasi pelatihan mulai dari sekitar 0.65 dan naik secara gradual hingga mencapai hampir 1.00 pada epoch ke-4, sedangkan akurasi validasi juga meningkat dengan pola serupa dan sedikit lebih tinggi dari akurasi pelatihan pada epoch awal, yang menunjukkan kemampuan model untuk melakukan generalisasi dengan baik. Grafik sebelah kanan memperlihatkan penurunan nilai loss yang signifikan pada kedua data pelatihan dan validasi, mulai dari nilai sekitar 0.85 hingga mendekati 0.0 pada epoch ke-4, menandakan model semakin efektif dalam meminimalkan kesalahan prediksi.



Gambar 4. 80 Plot hasil pengujian CNN Class Weights

Pada gambar 4.81 terlihat laporan klasifikasi dari model CNN setelah diterapkan penanganan imbalance menggunakan class weight pada data uji sebanyak 819 sampel. Model menunjukkan performa yang cukup baik dengan nilai precision untuk kelas negatif sebesar 0.94 dan kelas positif sebesar 0.82. Recall untuk kelas negatif tercatat sebesar 0.94, sedangkan untuk kelas positif sebesar 0.83, menandakan model mampu mengenali sebagian besar sampel dari masing-masing kelas, meskipun untuk kelas positif nilainya sedikit lebih rendah. F1-score untuk kelas negatif adalah 0.94 dan untuk kelas positif adalah 0.83, yang mencerminkan keseimbangan antara precision dan recall pada masing-masing kelas. Akurasi keseluruhan model mencapai 0.91, dengan nilai macro average dan weighted average untuk precision, recall, dan f1-score berada di kisaran 0.88 hingga 0.91, mengindikasikan performa model yang cukup konsisten namun terdapat sedikit gap performa terutama pada kelas positif. Hal ini menunjukkan bahwa

meskipun penanganan imbalance telah diterapkan, terdapat tantangan dalam mengoptimalkan klasifikasi untuk kelas minoritas (positif).

CNN Classification Report:

26/26			1s 26ms/step	
	precision	recall	f1-score	support
negatif	0.94	0.94	0.94	610
positif	0.82	0.83	0.83	209
accuracy			0.91	819
macro avg	0.88	0.89	0.88	819
weighted avg	0.91	0.91	0.91	819

Gambar 4. 81 Hasil Laporan Klasifikasi Model CNN Class Weights

4.6.9 Hasil Evaluasi CNN Class weights Hyperparameter

Pada gambar 4.82 terlihat kode Python yang digunakan untuk mendapatkan model CNN terbaik dari proses tuning dengan mengambil model terbaik dari tuner (`best_cnn_model`) berdasarkan nilai evaluasi. Selanjutnya, model tersebut dievaluasi menggunakan data uji (`X_test`, `y_test`) untuk mengukur performa secara objektif. Hasil evaluasi berupa nilai loss dan akurasi dicetak pada konsol, di mana akurasi model setelah tuning tercatat sebesar 0.9084. Nilai ini menunjukkan bahwa model CNN yang diperoleh berhasil melakukan klasifikasi dengan tingkat keberhasilan sekitar 90.84% pada data yang belum pernah dilihat sebelumnya, yang menandakan tuning hyperparameter mampu meningkatkan kemampuan generalisasi model.

```

5. Dapatkan dan evaluasi model terbaik

# Mendapatkan model terbaik
best_cnn_model = tuner_cnn.get_best_models(num_models=1)[0]

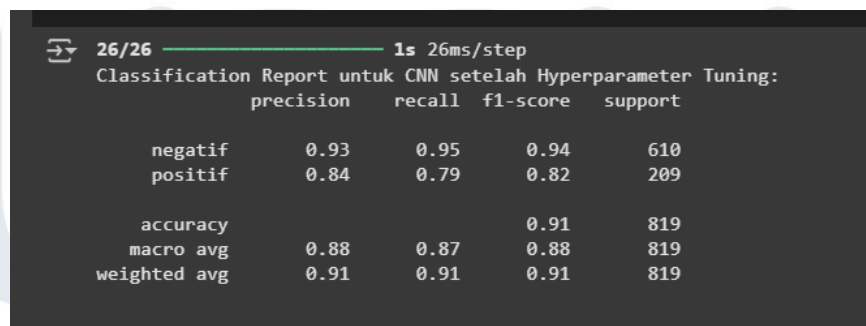
# Evaluasi model
loss, accuracy = best_cnn_model.evaluate(X_test, y_test)
print(f"Akurasi CNN setelah tuning: {accuracy:.4f}")

/usr/local/lib/python3.11/dist-packages/keras/src/saving/saving_lib.py:757:
saveable.load_own_variables(weights_store.get(inner_path))
26/26 1s 23ms/step - accuracy: 0.9075 - loss: 0.3721
Akurasi CNN setelah tuning: 0.9084

```

Gambar 4. 82 Akurasi CNN setelah tuning

Pada gambar 4.83 terlihat laporan klasifikasi dari model Convolutional Neural Network (CNN) yang telah melalui proses penanganan imbalance dengan teknik class weight serta dilakukan tuning hyperparameter untuk mengoptimalkan performa. Data uji yang digunakan sebanyak 819 sampel. Model menunjukkan performa yang cukup baik dengan nilai precision untuk kelas negatif sebesar 0.93 dan kelas positif sebesar 0.84. Recall untuk kelas negatif tercatat sebesar 0.95, sedangkan kelas positif sebesar 0.79, yang menunjukkan model mampu mengenali sebagian besar sampel dari kelas negatif dengan baik, namun performa pengenalan kelas positif sedikit lebih rendah. F1-score untuk kelas negatif sebesar 0.94 dan untuk kelas positif sebesar 0.82, mencerminkan keseimbangan antara precision dan recall yang baik pada kelas negatif dan cukup pada kelas positif. Akurasi keseluruhan model mencapai 0.91, dengan nilai macro average dan weighted average untuk precision, recall, dan f1-score berada di kisaran 0.87 hingga 0.91, menunjukkan konsistensi performa model di antara kelas, meskipun masih terdapat tantangan pada kelas minoritas (positif). Hasil ini mengindikasikan bahwa tuning hyperparameter dan penanganan imbalance berhasil meningkatkan performa model, walaupun pengklasifikasian kelas positif masih perlu perhatian lebih lanjut.



```

26/26 1s 26ms/step
Classification Report untuk CNN setelah Hyperparameter Tuning:
      precision    recall  f1-score   support

negatif      0.93      0.95      0.94       610
positif      0.84      0.79      0.82       209

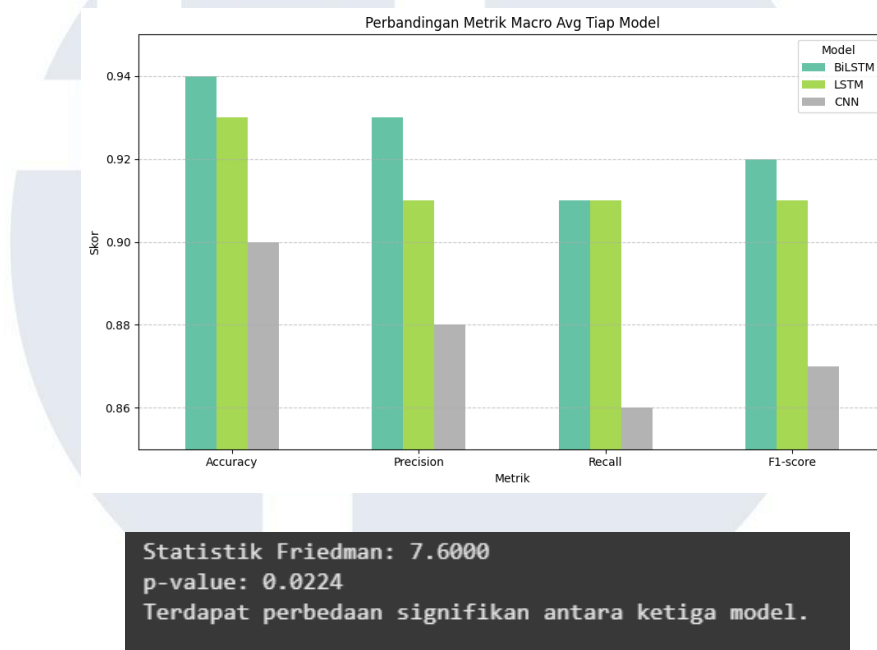
accuracy      0.91      0.91      0.91       819
macro avg      0.88      0.87      0.88       819
weighted avg      0.91      0.91      0.91       819
  
```

Gambar 4. 83 Hasil Laporan Klasifikasi Model CNN Class Weights Hyperparameter

4.6.10 Hasil Uji Statistik Friedman

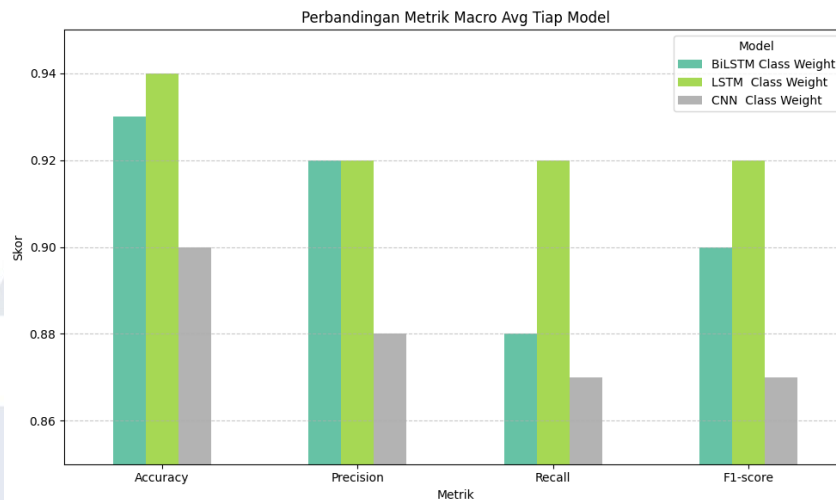
Pada gambar, diperlihatkan bahwa model BiLSTM memiliki performa terbaik dibandingkan LSTM dan CNN dalam metrik rata-rata makro, dengan akurasi tertinggi mencapai 94%. Model LSTM mengikuti di posisi kedua dengan akurasi sekitar 93%, sementara CNN memiliki nilai terendah di kisaran 90%. Selain

itu, BiLSTM juga unggul pada metrik precision, recall, dan F1-score yang menunjukkan keseimbangan yang lebih baik antara ketepatan dan sensitivitas model. Hasil uji statistik Friedman dengan nilai p sebesar 0,0224 menegaskan adanya perbedaan signifikan dalam performa ketiga model tersebut.



Gambar 4. 84 Uji Friedman Dasar

Pada gambar, diperlihatkan bahwa model LSTM dengan pengaturan class weights mencapai akurasi tertinggi yaitu 94%, mengungguli BiLSTM dan CNN dengan konfigurasi serupa. Model LSTM ini juga menunjukkan nilai precision, recall, dan F1-score yang tinggi di angka 92% ke atas, menunjukkan performa yang seimbang antara ketepatan dan sensitivitas. BiLSTM dengan class weights hyperparameter mencatatkan akurasi 94% dan performa metrik yang kompetitif, terutama pada precision dan recall. CNN dengan class weights hyperparameter menempati posisi paling bawah dengan akurasi 90% dan metrik lainnya yang sedikit lebih rendah dibanding LSTM dan BiLSTM. Hasil uji statistik Friedman dengan p-value 0,0224 menegaskan bahwa terdapat perbedaan signifikan di antara ketiga model tersebut.



```

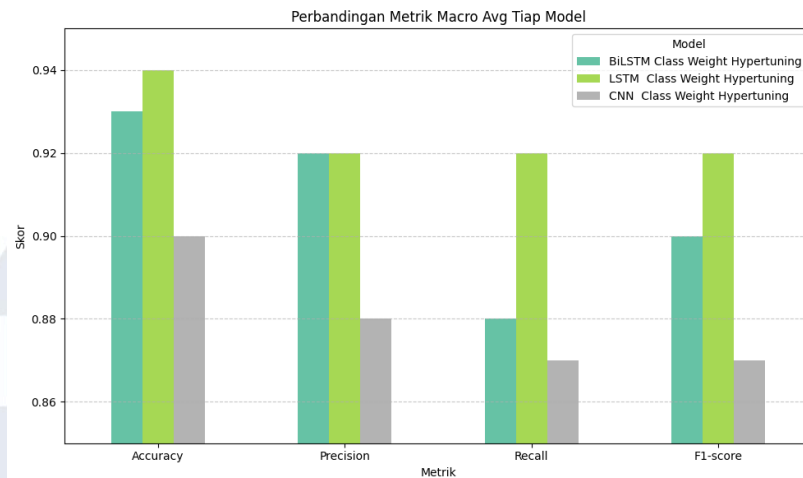
=== Hasil Uji Friedman ===
Statistik Friedman: 7.6000
p-value: 0.0224
Terdapat perbedaan signifikan antara ketiga model.

```

Gambar 4. 85 Uji Friedman ClassWeight

Pada gambar, terlihat bahwa setelah tuning class weights dan hyperparameter, model LSTM mencapai akurasi tertinggi sebesar 94%, mengungguli BiLSTM dan CNN. Model LSTM ini juga mempertahankan nilai precision, recall, dan F1-score yang konsisten tinggi di angka 92%, menunjukkan kinerja yang kuat dan stabil. BiLSTM dengan hyperparameter tuning menyusul dengan performa yang kompetitif, terutama pada metrik precision dan F1-score di angka 90% ke atas. Sementara itu, CNN masih berada di posisi paling bawah dengan akurasi dan metrik lainnya yang mendekati 90%. Hasil uji statistik Friedman setelah tuning dengan p-value 0,0231 menegaskan bahwa terdapat perbedaan signifikan dalam performa ketiga model tersebut.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



```

=== Hasil Uji Friedman (Setelah Tuning) ===
Statistik Friedman: 7.5385
p-value: 0.0231
Terdapat perbedaan signifikan antara ketiga model.

```

Gambar 4. 86 Uji Friedman ClassWeight dan Hypertuning

Secara keseluruhan model BiLSTM yang dikombinasikan dengan penyesuaian bobot kelas dan tuning hyperparameter menunjukkan performa terbaik dengan akurasi makro tertinggi mencapai 94%, mengungguli LSTM dan CNN yang masing-masing memiliki akurasi sekitar 93% dan 90%. Model LSTM dengan pengaturan class weights bahkan mencapai akurasi tertinggi 94% serta performa metrik precision, recall, dan F1-score di atas 92%, menunjukkan keseimbangan optimal antara ketepatan dan sensitivitas. Setelah tuning class weights dan hyperparameter, LSTM mempertahankan performa stabil dengan akurasi 94%, sementara BiLSTM juga menampilkan hasil kompetitif terutama pada precision dan F1-score di atas 90%. CNN konsisten berada di posisi paling bawah dengan akurasi dan metrik lainnya mendekati 90%. Hasil uji statistik Friedman dengan p-value sekitar 0,022-0,023 menegaskan adanya perbedaan signifikan dalam performa ketiga model tersebut.

4.7 Implementasi Model Aplikasi

Dari hasil evaluasi yang dilakukan, terlihat bahwa performa algoritma mengalami peningkatan secara bertahap. Dimulai dari penggunaan algoritma dasar, kemudian ditingkatkan dengan teknik untuk mengatasi ketidakseimbangan data, dan akhirnya dengan penerapan hyperparameter tuning. Semua langkah ini berhasil meningkatkan akurasi model secara signifikan.

Setelah mendapatkan hasil yang memuaskan, langkah berikutnya adalah melakukan deployment. Pada tahap ini, dibuat sebuah aplikasi prediksi analisis sentimen berbasis web menggunakan *library* Streamlit. Aplikasi ini dirancang agar pengguna dapat dengan mudah melakukan prediksi sentimen melalui antarmuka yang sederhana dan interaktif.

Proses implementasi aplikasi mengikuti prinsip alur Knowledge Discovery In Databases (KDD) yang terdiri dari beberapa tahapan penting. Tahapan tersebut meliputi persiapan awal data (Pre-KDD), pemilihan data yang relevan (Data Selection), pembersihan dan persiapan data (Preprocessing), transformasi data ke format yang sesuai (Transformation), penerapan algoritma untuk menemukan pola (Data Mining), serta evaluasi hasil model untuk memastikan kualitas prediksi (Evaluation).

Aplikasi ini digunakan untuk memprediksi klasifikasi sentimen pada data dari aplikasi Tinder. Selain memberikan hasil prediksi, aplikasi juga menampilkan visualisasi data yang membantu pengguna memahami pola dan kata-kata yang sering muncul. Dengan adanya visualisasi ini, pengguna dapat lebih mudah menggali informasi dan mencari kalimat yang relevan berdasarkan kata-kata yang muncul dalam data.

4.7.1 Fitur Pembersihan Data Teks

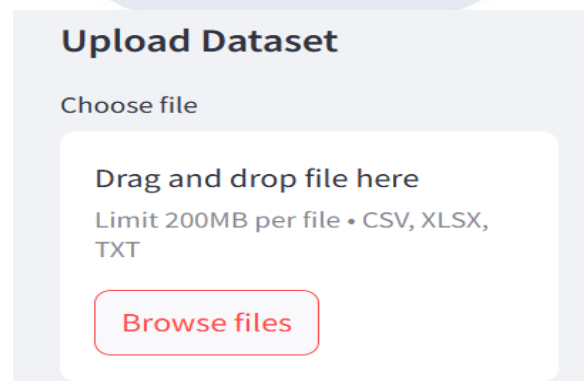
Pada gambar 4.87, ditampilkan antarmuka fitur pembersihan data teks dalam aplikasi Sentimen Analisis berbasis web. Fitur ini menyediakan berbagai opsi untuk membersihkan dan mempersiapkan data teks sebelum dilakukan analisis, seperti menghapus karakter yang tidak diperlukan, menghilangkan kata-kata umum (stopwords), dan mengubah teks menjadi huruf kecil agar konsisten.

Aplikasi Sentimen Analisis

 **Pembersihan Data Teks**  Analisa Sentimen  Visualisasi Data 

Gambar 4. 87 Tampilan Fitur Pembersihan Data Teks

Pada gambar 4.88, ditampilkan antarmuka fitur untuk memasukkan data mentah ke dalam aplikasi Sentimen Analisis. Fitur ini memungkinkan pengguna untuk mengunggah dataset dengan cara drag and drop atau memilih file secara manual melalui tombol "Browse files". Sistem mendukung berbagai format file seperti CSV, XLSX, dan TXT dengan batas ukuran maksimal 200MB per file. Dengan adanya fitur ini, pengguna dapat dengan mudah memuat data mentah yang akan diproses lebih lanjut dalam analisis sentimen.



Gambar 4. 88 Memasukan Data Mentah

Pada gambar 4.89, ditampilkan antarmuka fitur pemilihan kolom data teks dalam aplikasi Sentimen Analisis. Pengguna dapat memilih kolom yang berisi data teks yang akan dianalisis melalui menu dropdown yang tersedia. Setelah kolom dipilih, sistem akan memproses teks dari kolom tersebut untuk tahap analisis selanjutnya. Fitur ini memudahkan pengguna dalam menentukan sumber data yang tepat dari dataset yang telah diunggah agar analisis sentimen dapat berjalan dengan akurat.

Pilih kolom teks:

reviewId

Processing text...

reviewId

userName

userImage

content

score

thumbsUpCount

reviewCreatedVersion

Gambar 4. 89 Pemilihan Kolom Data

Pada gambar 4.90, ditampilkan antarmuka yang memperlihatkan kolom dataset setelah proses preprocessing selesai dilakukan dalam aplikasi Sentimen Analisis. Pada tahap ini, data teks yang telah dipilih sebelumnya diproses dan hasilnya ditampilkan dalam bentuk tabel yang memuat teks yang sudah dibersihkan dan disiapkan untuk analisis lebih lanjut. Tampilan ini memudahkan pengguna untuk memverifikasi hasil preprocessing dan memastikan bahwa data sudah siap untuk tahap analisis sentimen.

Pilih kolom teks:

content

Processing text...

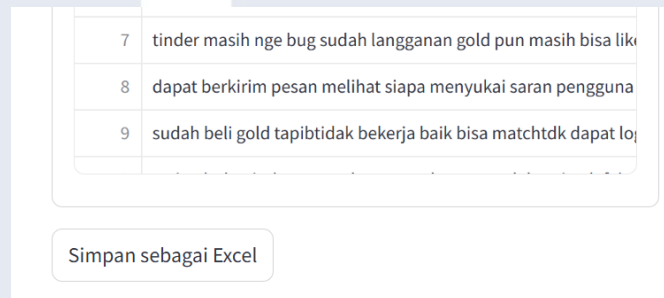
Processed Data

	full_text
11	kak kenapa logout sendiri begini yapani apaapainsaya mau ma
12	sebenarnya sudah bagus dulu cuma coba deh potensial match
13	tinder kenapa akun gue logout mulu kadi bisa menerima chat k
14	main tinder sudah merasakan aneh an atau keganjalan bisa bis

Gambar 4. 90 Kolom Dataset Setelah Preprocessing

Pada gambar 4.91, ditampilkan antarmuka fitur untuk menyimpan file dataset yang telah diproses dalam aplikasi Sentimen Analisis. Fitur ini menyediakan tombol "Simpan sebagai Excel" yang memungkinkan pengguna untuk mengunduh

hasil dataset dalam format file Excel (.xlsx). Dengan adanya fitur ini, pengguna dapat dengan mudah menyimpan dan membagikan data hasil preprocessing untuk keperluan dokumentasi atau analisis lebih lanjut di luar aplikasi. Fasilitas penyimpanan ini sangat membantu dalam menjaga hasil kerja dan memudahkan integrasi dengan alat analisis data lainnya.



Gambar 4. 91 Menyimpan File Dataset

4.7.2 Fitur Analisa Sentimen

Pada gambar 4.92, ditampilkan antarmuka fitur Analisa Sentimen dalam aplikasi Sentimen Analisis yang dirancang untuk memproses data teks yang sudah melalui tahap preprocessing. Fitur ini menyediakan berbagai opsi dan kontrol yang memungkinkan pengguna untuk menjalankan analisis sentimen secara langsung pada data yang telah dipersiapkan. Pengguna dapat dengan mudah mengakses dan mengoperasikan fitur ini melalui menu yang tersedia, sehingga proses analisis menjadi lebih efisien dan terstruktur. Tujuan utama dari fitur ini adalah untuk mengidentifikasi dan mengklasifikasikan sentimen dalam teks, sehingga memberikan wawasan yang berguna untuk pengambilan keputusan berbasis data.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

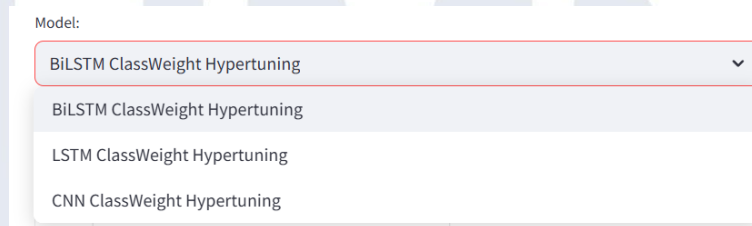
Aplikasi Sentimen Analisis

 Pembersihan Data Teks  Analisa Sentimen  Visualisasi Data 

Sentiment Analysis for Processed Data

Gambar 4. 92 Tampilan Fitur Analisa Sentimen

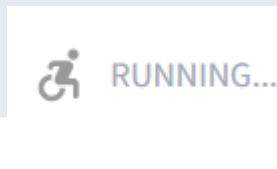
Pada gambar 4.93, ditampilkan antarmuka fitur pemilihan model algoritma dalam aplikasi Sentimen Analisis yang memungkinkan pengguna memilih model machine learning yang akan digunakan untuk analisis sentimen. Pilihan model yang tersedia meliputi BiLSTM, LSTM, dan CNN, yang masing-masing memiliki keunggulan dalam memproses data teks dan menangkap pola sentimen secara berbeda. Fitur ini memberikan fleksibilitas kepada pengguna untuk mencoba berbagai model dan menentukan mana yang paling sesuai dengan karakteristik data mereka. Dengan adanya opsi model yang beragam, aplikasi ini mendukung analisis yang lebih adaptif dan akurat sesuai kebutuhan pengguna.



Gambar 4. 93 Pemilihan Model Algoritma

Pada gambar 4.94, ditampilkan antarmuka fitur yang menunjukkan proses pemuatan (loading) model algoritma dalam aplikasi Sentimen Analisis. Saat model sedang dimuat, pengguna akan melihat indikator status berupa tulisan "RUNNING..." yang menandakan bahwa proses pemuatan sedang berlangsung. Fitur ini memberikan umpan balik visual kepada pengguna agar mereka mengetahui bahwa sistem sedang bekerja dan belum selesai memproses. Dengan adanya

indikator ini, pengalaman pengguna menjadi lebih informatif dan mengurangi ketidakpastian selama menunggu hasil analisis.



Gambar 4. 94 Load Process Model Algoritma

Pada gambar 4.95, ditampilkan antarmuka fitur hasil prediksi algoritma dalam aplikasi Sentimen Analisis yang memperlihatkan output dari proses klasifikasi sentimen pada data teks. Tabel hasil prediksi ini menampilkan kolom teks asli (full_text) beserta label sentimen yang diprediksi, seperti negatif, netral, dan positif, sehingga memudahkan pengguna untuk melihat hasil analisis secara langsung. Fitur ini memberikan gambaran yang jelas mengenai bagaimana model algoritma mengkategorikan setiap entri teks berdasarkan sentimen yang terkandung di dalamnya.

Hasil Prediksi

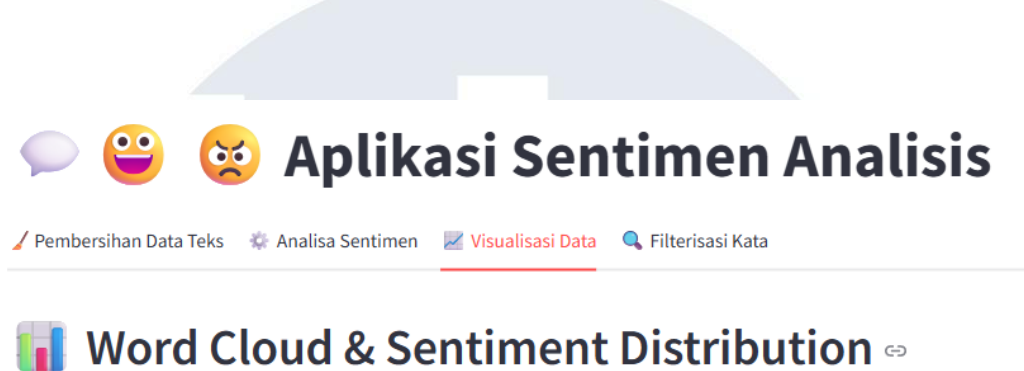
	full_text	sentiment
190	lama buka tiba tiba blokir rekomendasi bot buat cha	negatif
191	dulu memakai tinder masalah sejak beberapa hari su	negatif
192	tingkatkan gratisnya hehe	positif
193	apaan akhir kenapa ngelag sedikit aplikasi tiba kelua	negatif
194	please kenapa bisa buka awalnya kira jaringan lah te	negatif

Gambar 4. 95 Hasil Prediksi Algoritma

4.7.3 Fitur Visualisasi Data

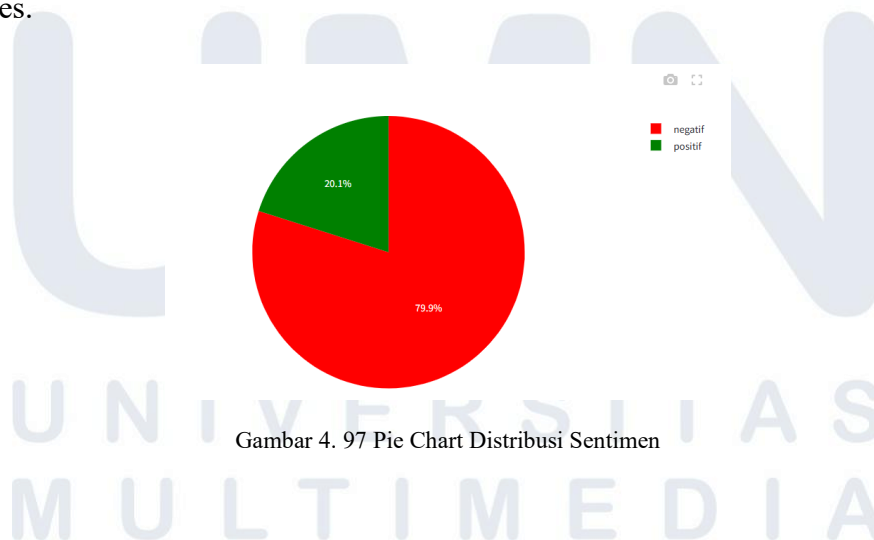
Pada gambar 4.96, ditampilkan antarmuka fitur Visualisasi Data dalam aplikasi Sentimen Analisis yang menampilkan bagian Word Cloud & Sentiment Distribution. Fitur ini memberikan representasi visual dari kata-kata yang paling sering muncul dalam dataset serta distribusi sentimen yang telah dianalisis, sehingga memudahkan pengguna untuk memahami pola dan tren sentimen secara keseluruhan. Dengan adanya visualisasi ini, pengguna dapat dengan cepat mengidentifikasi kata kunci penting dan proporsi sentimen positif, negatif, maupun

netral dalam data. Fitur ini sangat berguna untuk memperkaya analisis dan memberikan insight yang lebih intuitif terhadap hasil sentimen analisis.



Gambar 4. 96 Tampilan Fitur Visualisasi Sentimen

Pada gambar 4.97, ditampilkan visualisasi berupa pie chart yang menggambarkan distribusi sentimen dari data yang telah dianalisis dalam aplikasi Sentimen Analisis. Diagram ini memperlihatkan proporsi persentase sentimen netral, positif, dan negatif dengan warna yang berbeda, yaitu biru untuk netral, hijau untuk positif, dan merah untuk negatif. Dengan adanya pie chart ini, pengguna dapat dengan mudah memahami sebaran sentimen secara keseluruhan dalam dataset yang dianalisis. Fitur ini sangat membantu dalam memberikan gambaran cepat mengenai dominasi sentimen dan tren umum yang muncul dari data teks yang diproses.



Gambar 4. 97 Pie Chart Distribusi Sentimen

Pada gambar 4.98, ditampilkan tiga visualisasi word cloud yang menggambarkan kata-kata yang paling sering muncul dalam dataset berdasarkan sentimen positif, dan negatif. Word cloud ini memberikan representasi visual yang

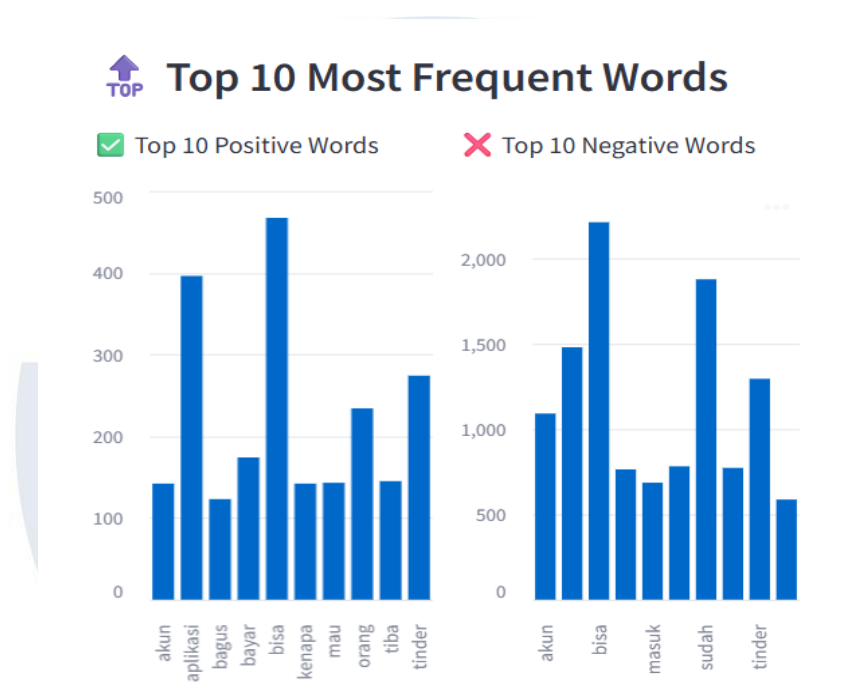
intuitif dengan ukuran kata yang menunjukkan frekuensi kemunculannya, sehingga memudahkan pengguna untuk mengidentifikasi kata-kata kunci yang dominan dalam masing-masing kategori sentimen. Dengan adanya fitur ini, pengguna dapat dengan cepat memahami pola kata yang sering muncul dalam teks yang dianalisis, baik yang bernada positif, negatif, maupun netral. Visualisasi ini sangat berguna untuk memperkaya analisis sentimen dengan memberikan insight yang lebih mendalam mengenai karakteristik bahasa yang digunakan dalam data.



Gambar 4. 98 Tampilan Visualisasi Kata Wordcloud

Pada gambar 4.99, ditampilkan visualisasi berupa grafik batang yang memperlihatkan 10 kata paling sering digunakan dalam dataset yang telah dianalisis, dibagi berdasarkan sentimen positif, negatif, dan netral. Grafik ini memudahkan pengguna untuk melihat frekuensi kemunculan kata-kata kunci yang dominan dalam masing-masing kategori sentimen, seperti kata "jaki" dan "aplik" pada sentimen positif, serta kata "akun" dan "tinder" pada sentimen negatif dan netral. Dengan adanya fitur ini, pengguna dapat memahami pola penggunaan kata yang berkontribusi pada klasifikasi sentimen dalam data teks. Fitur ini sangat berguna untuk memperdalam analisis dan memberikan insight yang lebih terperinci mengenai karakteristik bahasa dalam dataset.

UNIVERSITAS
MULTIMEDIA
NUSANTARA







Gambar 4. 99 10 Kata Paling Sering Digunakan

4.7.4 Fitur Filterisasi Kata

Pada gambar 4.100, ditampilkan antarmuka fitur Filterisasi Kata dalam aplikasi Sentimen Analisis yang memungkinkan pengguna untuk melakukan penyaringan kata-kata tertentu dalam dataset. Fitur ini menyediakan opsi bagi pengguna untuk memasukkan kata-kata yang ingin difilter atau dikecualikan dari analisis, sehingga dapat memfokuskan hasil pada kata-kata yang relevan dan mengurangi noise data. Dengan adanya fitur filter ini, pengguna dapat mengontrol dan menyesuaikan analisis sentimen agar lebih akurat dan sesuai dengan kebutuhan penelitian atau aplikasi yang dijalankan. Fitur ini sangat membantu dalam memperbaiki kualitas data dan memberikan hasil analisis yang lebih bermakna serta terfokus.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Aplikasi Sentimen Analisis

 Pembersihan Data Teks  Analisa Sentimen  Visualisasi Data  **Filterisasi Kata**

Filter Words

Gambar 4. 100 Tampilan Fitur Filterisasi Kata

Pada gambar 4.101, ditampilkan antarmuka fitur Memasukan Filter Berdasarkan Kata dalam aplikasi Sentimen Analisis yang memungkinkan pengguna untuk memasukkan kata-kata tertentu sebagai filter dalam proses analisis. Fitur ini menyediakan kolom input di mana pengguna dapat mengetikkan kata-kata yang ingin disaring atau dikecualikan dari dataset agar tidak mempengaruhi hasil analisis sentimen. Dengan adanya fitur ini, pengguna dapat lebih mudah mengelola dan membersihkan data teks sehingga analisis menjadi lebih fokus dan relevan terhadap tujuan penelitian. Fitur ini sangat membantu dalam meningkatkan akurasi dan kualitas hasil analisis dengan menghilangkan kata-kata yang tidak diinginkan atau kurang penting.

Masukkan kata:

nipu

Cari

Gambar 4. 101 Memasukan Filter Berdasarkan Kata

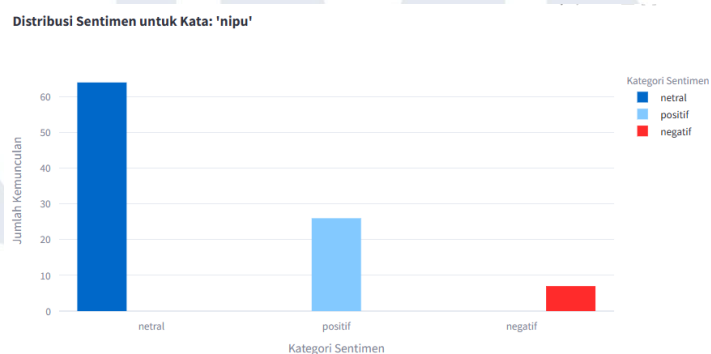
Pada gambar 4.102, ditampilkan antarmuka hasil dari proses filterisasi kata dalam aplikasi Sentimen Analisis yang memperlihatkan data teks yang telah disaring berdasarkan kata-kata filter yang dimasukkan sebelumnya. Tabel ini menampilkan kolom teks lengkap (full_text) beserta label sentimen yang sudah diprediksi, sehingga pengguna dapat melihat bagaimana data yang sudah difilter memengaruhi hasil analisis sentimen. Dengan adanya tampilan ini, pengguna dapat memverifikasi dan memastikan bahwa kata-kata yang tidak diinginkan telah berhasil dihilangkan dari dataset, sehingga analisis menjadi lebih bersih dan

relevan. Fitur ini sangat penting untuk meningkatkan kualitas data dan memberikan hasil analisis yang lebih akurat serta terfokus sesuai kebutuhan pengguna.

	full_text	sentiment
753	akhir sering dapat sms dari tinder katanya tetapi sudah pakai tinder unduh ulang mau	negatif
764	aplikasi penipuan bisa login setelah hubungin facebook nomor itu adalah mengambil	negatif
768	pemerasan uang cuma mengklik top up belum disetujui langsung terjadi autopay pen	positif
790	lucu tinder main blok sudah akun lama menipu pakai foto palsu swipe sewajarnya uni	negatif
833	maaf kasih rating rendah soalnya pengalaman buruk gara tinder niatnya mau cari jode	negatif
844	oknum atau penipuan ujungnya meminta transfer hati-hati	negatif
878	hati-hati buat main tinder jadi korban penipuan uang alasan pinjam uang kembalikan	negatif
912	kenapa setelah daftar behavior bakal data ehongerny memang aplikasi penipuan hari da	negatif

Gambar 4. 102 Hasil Filterisasi Kata

Pada gambar 4.103, ditampilkan visualisasi distribusi sentimen untuk kata "nipu" dalam aplikasi Sentimen Analisis. Grafik batang ini memperlihatkan jumlah kemunculan kata tersebut yang terbagi ke dalam tiga kategori sentimen, yaitu netral, positif, dan negatif, dengan warna biru untuk netral, biru muda untuk positif, dan merah untuk negatif. Dari grafik tersebut, terlihat bahwa kata "nipu" paling sering muncul dalam konteks sentimen netral, diikuti oleh sentimen positif, dan jumlah kemunculan dalam sentimen negatif yang paling sedikit. Fitur ini sangat berguna untuk membantu pengguna memahami bagaimana kata-kata tertentu berkontribusi terhadap distribusi sentimen dalam dataset secara lebih rinci dan mendalam.



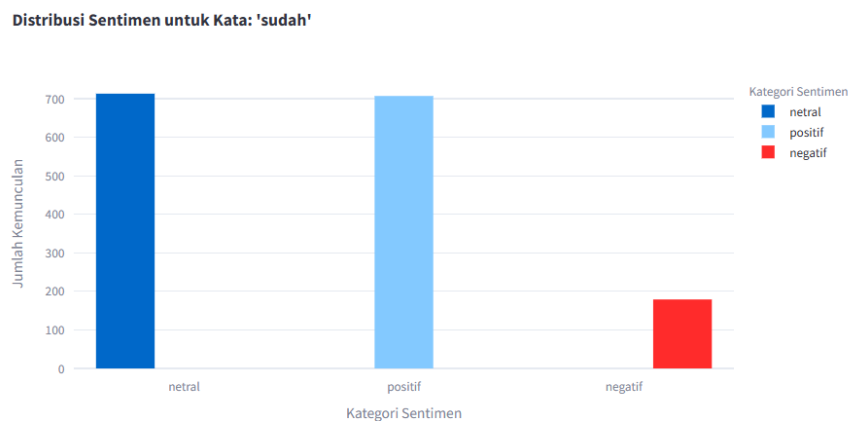
Gambar 4. 103 Klasifikasi Golongan Kata

4.7.5 Analisis Kesalahan

4.7.5.1 Kesalahan Prediksi BiLSTM

Pada gambar 4.104 ini menunjukkan hasil prediksi yang salah dari algoritma Bidirectional Long Short-Term Memory (BiLSTM). Algoritma BiLSTM

seharusnya mampu menangkap konteks dari dua arah, namun pada gambar ini terlihat bahwa prediksi sentimen untuk kata "sudah" tidak sesuai dengan ekspektasi. Munculnya kesalahan klasifikasi ini mengindikasikan bahwa model gagal membedakan antara konteks positif, negatif, dan netral secara tepat. Visualisasi ini memperlihatkan adanya tumpang tindih kategori sentimen yang seharusnya bisa diantisipasi oleh BiLSTM. Oleh karena itu, gambar ini menggambarkan bagaimana model BiLSTM terkadang masih menghasilkan prediksi yang kurang akurat dalam analisis sentimen.



Gambar 4. 104 Barplot Kesalahan Hasil Prediksi BILSTM

Pada gambar ini merupakan hasil dari perbaikan masalah yaitu dengan mencari pustaka Lexicon Indonesia yang lebih memiliki arti luas supaya mampu mendefinisikan arti sentimen yang lebih tepat. Selain memilih kamus Lexicon Indonesia pada analisis kesalahan ini adanya perubahan yaitu menghapus atau menghilangkan kata netral karena dianggap mempunyai makna bias yang membuat penelitian ini tidak pasti. Hasil dari perbaikan kesalahan ini membuat makna sentiment menjadi lebih pasti

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Filter Words

Masukkan kata:

sudah

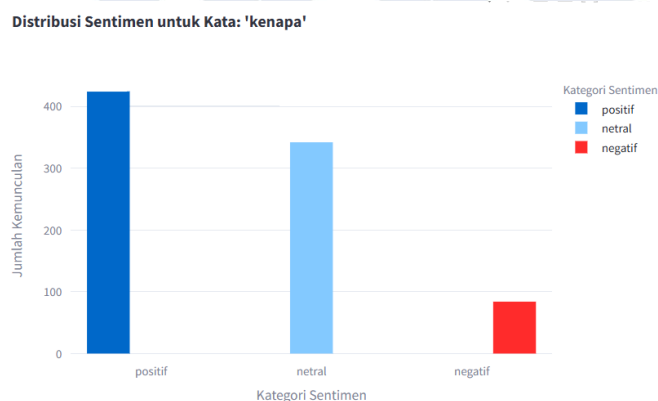
Cari

	full_text	sentiment
0	sudah beli langganan gold platinum pesan satupun masukjempol sampai keriting colok kanan colok kiri man	negatif
1	bagus orang pakai semuanya serba berbayar sudah ikut bulananpun masih perintilan bayar sekarang verifika	negatif
2	aplikasi tinder parah benar sudah top up saldo sudah potong masih enggak bisa terbuka mau lihat siapa mer	negatif
4	sudah melakukan pembayaran berlangganan akun gold bukti pembayarannya masih tinder terbaca tetap dia	negatif
5	aplikasi crash rusak bikin akun langsung bekukan akunku melanggar apapun sok kalian mau instal aplikasi m	negatif
7	tinder masih nge bug sudah langganan gold pun masih bisa like bisa match selalu ketemu orang serasa stuck	negatif
9	sudah beli gold tapiitidak bekerja baik bisa matchtdk dapat login tiba tiba logout aneh apps retas	negatif
10	setiap buka tinder muncul orang walaupun sudah swipe left karena kira gangguan putuskan menghapus aku	negatif
12	sebenarnya sudah bagus dulu cuma coba deh potensial match itu diprioritasin dulu tiap hari notifikasi orang	negatif

Gambar 4. 105 Perbaikan Kesalahan Hasil Prediksi BILSTM

4.7.5.2 Kesalahan Prediksi LSTM

Pada gambar 4.106 ini memperlihatkan hasil prediksi yang tidak akurat dari algoritma Long Short-Term Memory (LSTM). Prediksi sentimen untuk kata "kenapa" menunjukkan distribusi yang tidak konsisten dengan konteks sebenarnya, di mana kategori positif dan netral tampak tumpang tindih. Hal ini mengindikasikan bahwa model LSTM mengalami kesulitan dalam membedakan sentimen dengan tepat untuk kata tersebut. Kesalahan ini menegaskan keterbatasan LSTM dalam menangkap nuansa konteks secara penuh pada beberapa kasus. Oleh karena itu, gambar ini menggambarkan tantangan yang masih dihadapi dalam penggunaan LSTM untuk analisis sentimen secara presisi.



Gambar 4. 106 Barplot Kesalahan Hasil Prediksi LSTM

Pada gambar 4.107 ini merupakan hasil dari perbaikan masalah yaitu dengan mencari pustaka Lexicon Indonesia yang lebih memiliki arti luas supaya

mampu mendefinisikan arti sentimen yang lebih tepat. Selain memilih kamus Lexicon Indonesia pada analisis kesalahan ini adanya perubahan yaitu menghapus atau menghilangkan kata netral karena dianggap mempunyai makna bias yang membuat penelitian ini tidak pasti. Hasil dari perbaikan kesalahan ini membuat makna sentiment menjadi lebih pasti

Masukkan kata:

kenapa

Cari

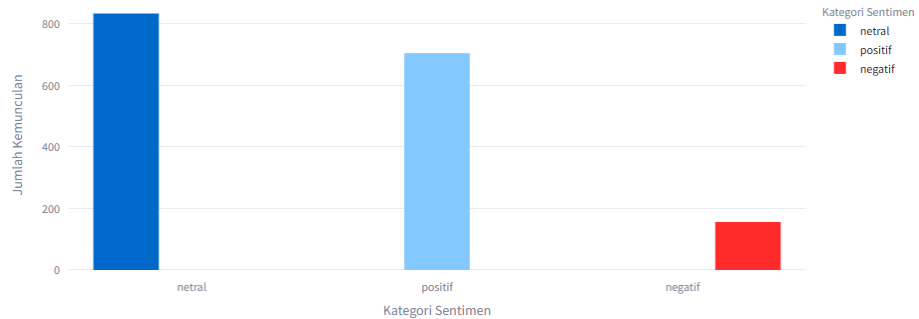
	full_text	sentiment
6	sejauh masih make tinder buat mencari teman menurutku biasa biasa tinder fitur spesial cuma bisa swipe do	negatif
11	kak kenapa logout sendiri begini yadhl apaapainsaya mau masuk akun malahan swafoto sedang ditinjauin	negatif
12	sebenarnya sudah bagus dulu cuma coba deh potensial match itu diprioritasin dulu tiap hari notifikasi orang	negatif
13	tinder kenapa akun gue logout mulu kadi bisa menerima chat walaupun pasti chatnya kehapus dikira unmatc	negatif
15	sudah beli mahal filturnya tindersaja sebenarnya gue mau beli kenapa bisa melihat siapa menyukai beli dulu	negatif
21	kenapa langganan tinder gold bisa menerima pesan dari teman luar negara	negatif
22	parah asli sudah berlangganan akun seperti dibekukan awal daftar lancar lancar kenapa mau berlangganan t	negatif
23	aduh bagaimana kira juta itukan kenapa muncul spesifikasi bidadari semua waduh sesuai namanya tinder bil	negatif
26	bikin akun karena sebelumnya rusak bisa kirim pesan eh rusak kenapa tinderr verifikasi lo semalam	negatif
37	kenapa tinder bisa kirim pesan match orang sebelumnya bisa semenjak proses swafoto akun seperti bekukar	negatif

Gambar 4. 107 Perbaikan Kesalahan Hasil Prediksi LSTM

4.7.5.3 Kesalahan Prediksi CNN

Pada gambar 4.108 ini memperlihatkan hasil prediksi yang kurang tepat dari algoritma CNN pada klasifikasi sentimen kata "aplikasi". Distribusi sentimen yang terlihat antara kategori netral dan positif tampak sangat berdekatan, sehingga menunjukkan ambiguitas dalam prediksi. Kesalahan ini mengindikasikan bahwa CNN mengalami kesulitan dalam membedakan konteks kata tersebut secara jelas. Model CNN yang seharusnya mampu menangkap pola lokal dari data teks ini tampaknya kurang efektif dalam mengklasifikasikan sentimen secara akurat pada kasus ini. Oleh karena itu, gambar ini menggambarkan tantangan yang dihadapi CNN dalam menghasilkan prediksi sentimen yang presisi untuk kata tersebut.

Distribusi Sentimen untuk Kata: 'aplikasi'



Gambar 4. 108 Gambar 4. 109 Barplot Kesalahan Hasil Prediksi CNN

Pada gambar 4.109 ini merupakan hasil dari perbaikan masalah yaitu dengan mencari pustaka Lexicon Indonesia yang lebih memiliki arti luas supaya mampu mendefinisikan arti sentimen yang lebih tepat. Selain memilih kamus Lexicon Indonesia pada analisis kesalahan ini adanya perubahan yaitu menghapus atau menghilangkan kata netral karena dianggap mempunyai makna bias yang membuat penelitian ini tidak pasti. Hasil dari perbaikan kesalahan ini membuat makna sentiment menjadi lebih pasti

Masukkan kata:

aplikasi

Cari

	full_text	sentiment
2	aplikasi tinder parah benar sudah top up saldo sudah potong masih enggak bisa terbuka mau lihat siapa mer	negatif
3	kelebihannya kita dicarikan pasangan baik terhindar dari penipuan seperti serbuan begitu mudahnya para w	negatif
4	sudah melakukan pembayaran berlangganan akun gold bukti pembayarannya masih tinder terbaca tetap dia	negatif
5	aplikasi crash rusak bikin akun langsung bekukan akunku melanggar apapun sok kalian mau instal aplikasi m	negatif
14	main tinder sudah merasakan aneh an atau keganjalan bisa bisanya bisa menerima mengirim pesan haha en	negatif
20	aplikasi jelas giliran login bisa notifikasi masuk pakai nomor handphone bisa email apalagi	negatif
24	aplikasi tinder sekarang buruk anjir kualitasnya makin jelk tiba keluar sendiri akunnya sudhln pakai aplikasi	negatif
28	adit tinder kebancuan nggak atau kayak aplikasi kebancuan	negatif

Gambar 4. 110 Perbaikan Kesalahan Hasil Prediksi CNN

4.8 Hasil & Pembahasan

4.8.1 Tabel Penelitian Terdahulu

Tabel 4. 1 Penelitian Terdahulu

Penelitian Terdahulu					
Peneliti	Algoritma	Akurasi	Precision	Recall	F1-score
Dany Eka Saputra & Auliya Rahman Isnain [56]	CNN	91%	91%	91%	91%
Titis Arwindarti Esther Irawati Setiawan, & Syaiful Imron [57]	LSTM	84%,	92.70%,	89%,	90.81%.
Fitroh & Fahmi Hudaya[58]	BiLSTM	99,14%	99,1%	99,1%	99,12%

Penelitian terdahulu menunjukkan bahwa berdasarkan hasil penelitian terdahulu, BiLSTM menonjol dengan akurasi sangat tinggi yaitu 99,14%, yang mengindikasikan kemampuannya dalam memahami konteks dua arah secara mendalam, seperti yang dilaporkan oleh Fitroh & Fahmi Hudaya[58]. Di sisi lain, LSTM yang digunakan oleh Titis Arwindarti Esther Irawati Setiawan, Syaiful Imron memberikan hasil yang cukup baik dengan akurasi sekitar 84%, menunjukkan keseimbangan yang baik antara performa dan efisiensi[57]. Sementara itu, CNN yang dipakai oleh Dany Eka Saputra & Auliya Rahman Isnain juga mencapai akurasi 91% dengan keunggulan pada kecepatan dan efisiensi pemrosesan[56]. Berdasarkan perbandingan ini, penelitian saat ini memilih ketiga algoritma tersebut untuk mengakomodasi berbagai kebutuhan analisis sentimen: BiLSTM untuk hasil yang paling akurat dan pemahaman konteks yang lebih dalam, LSTM sebagai pilihan yang seimbang antara performa dan sumber daya, serta CNN

untuk aplikasi yang membutuhkan kecepatan dan efisiensi tinggi. Pendekatan ini memungkinkan penelitian untuk mendapatkan gambaran sentimen yang komprehensif dan sesuai dengan berbagai kondisi penggunaan.

4.8.2 Tabel Penelitian Ini

Tabel 4. 2 Penelitian Ini

Penelitian Ini				
Algoritma	Akurasi	Precision	Recall	F1-score
BILSTM	92%	Positif : 92% Negatif : 93%	Positif : 77% Negatif : 98%	Positif : 84% Negatif : 95%
BILSTM with Class Weights	93%	Positif : 92% Negatif : 93%	Positif : 79% Negatif : 98%	Positif : 85% Negatif : 95%
BILSTM with Class Weights Hyperparameter	94%	Positif : 90% Negatif : 95%	Positif : 86% Negatif : 97%	Positif : 88% Negatif : 96%
LSTM	93%	Positif : 87% Negatif : 95%	Positif : 87% Negatif : 96%	Positif : 87% Negatif : 96%
LSTM with Class Weights	93%	Positif : 88% Negatif : 96%	Positif : 89% Negatif : 96%	Positif : 88% Negatif : 96%

Penelitian Ini				
Algoritma	Akurasi	Precision	Recall	F1-score
LSTM with Class Weights Hyperparameter	92%	Positif : 92% Negatif : 92%	Positif : 75% Negatif : 98%	Positif : 83% Negatif : 95%
CNN	90%	Positif : 83% Negatif : 93%	Positif : 78% Negatif : 94%	Positif : 81% Negatif : 94%
CNN with Class Weights	90%	Positif : 83% Negatif : 93%	Positif : 78% Negatif : 95%	Positif : 81% Negatif : 94%
CNN with Class Weights Hyperparameter	91%	Positif : 84% Negatif : 93%	Positif : 79% Negatif : 95%	Positif : 82% Negatif : 94%

Model BILSTM yang dikombinasikan dengan penyesuaian bobot kelas dan tuning hyperparameter menunjukkan kinerja terbaik dengan akurasi mencapai 94%. Recall kelas positif meningkat signifikan menjadi 86%, meskipun recall kelas negatif sedikit menurun menjadi 97%. Penyesuaian ini juga memperbaiki presisi dan skor F1 untuk kedua kelas, menandakan keseimbangan optimal antara sensitivitas dan ketepatan pengklasifikasian. Model BILSTM dengan penyesuaian bobot kelas saja memberikan peningkatan akurasi menjadi 93% dan kenaikan recall kelas positif menjadi 79%, dengan presisi yang stabil pada 92% dan recall kelas negatif tetap tinggi di 98%. Model BILSTM standar memiliki akurasi 92%, presisi untuk kelas positif dan negatif masing-masing 92% dan 93%, namun recall positif

masih terbatas pada 77% sedangkan recall negatif sangat tinggi pada 98%, menandakan model lebih andal mengidentifikasi ulasan negatif dibanding positif. Selanjutnya, model LSTM dengan atau tanpa penyesuaian bobot kelas menunjukkan akurasi di kisaran 92-94%, meskipun tuning hyperparameter pada LSTM dasar menurunkan akurasi menjadi 92%. Sedangkan model CNN mempertahankan performa stabil pada rentang 90-91%, tetapi dengan presisi dan recall kelas positif yang lebih rendah dibandingkan model LSTM dan BILSTM. Secara keseluruhan, kombinasi BILSTM dengan penyesuaian bobot kelas dan tuning hyperparameter memberikan keseimbangan terbaik antara akurasi tinggi dan metrik presisi-recall yang seimbang untuk aplikasi pengenalan sentimen.



BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Model BILSTM yang dikombinasikan dengan penyesuaian bobot kelas dan tuning hyperparameter menunjukkan performa terbaik dengan akurasi 94% dan recall kelas positif 86%, menandakan keseimbangan optimal antara sensitivitas dan presisi. Model ini melampaui BILSTM dengan penyesuaian bobot kelas saja yang mencapai 93% akurasi dan recall positif 79%, serta BILSTM standar dengan akurasi 92% dan recall positif 77%. Model LSTM memiliki performa bervariasi antara 92–94%, namun tuning hyperparameter justru menurunkan akurasi menjadi 92%. Sementara itu, model CNN mempertahankan kestabilan performa pada kisaran 90–91%, meskipun dengan presisi dan recall kelas positif yang lebih rendah dibanding model LSTM dan BILSTM.

Secara keseluruhan, model BILSTM yang dikombinasikan dengan penyesuaian bobot kelas dan tuning hyperparameter menunjukkan performa terbaik dengan akurasi makro tertinggi mencapai 94%, mengungguli LSTM dan CNN yang masing-masing memiliki akurasi sekitar 93% dan 90%. Model LSTM dengan pengaturan class weights bahkan mencapai akurasi tertinggi 94%, serta performa metrik precision, recall, dan F1-score di atas 92%, menunjukkan keseimbangan optimal antara ketepatan dan sensitivitas. Setelah tuning class weights dan hyperparameter, LSTM mempertahankan performa stabil dengan akurasi 94%, sementara BiLSTM juga menampilkan hasil kompetitif terutama pada precision dan F1-score di atas 90%. CNN konsisten berada di posisi paling bawah dengan akurasi dan metrik lainnya mendekati 90%. Hasil uji statistik Friedman dengan p-value sekitar 0,022–0,023 menegaskan adanya perbedaan signifikan dalam performa ketiga model tersebut.

Kontribusi utama dari penelitian ini terletak pada pengembangan sebuah aplikasi analisis sentimen yang terintegrasi secara menyeluruh, mulai dari tahap pra-pemrosesan teks, pemilihan model deep learning, proses klasifikasi sentimen,

hingga visualisasi hasil yang interaktif dan informatif. Aplikasi ini dilengkapi dengan fitur seperti word cloud dan visualisasi distribusi sentimen yang memungkinkan pengguna menggali informasi lebih mendalam dari data, baik berdasarkan kata kunci maupun kategori sentimen tertentu.

Meskipun demikian, penelitian ini memiliki beberapa keterbatasan. Dataset yang digunakan hanya mencakup ulasan dari satu platform aplikasi (Tinder), sehingga penerapan model pada domain lain masih perlu diteliti lebih lanjut. Selain itu, meskipun model deep learning memberikan performa tinggi, proses pelatihannya membutuhkan sumber daya komputasi yang besar dan waktu yang relatif lama, sehingga perlu mempertimbangkan efisiensi dalam penerapan praktisnya.

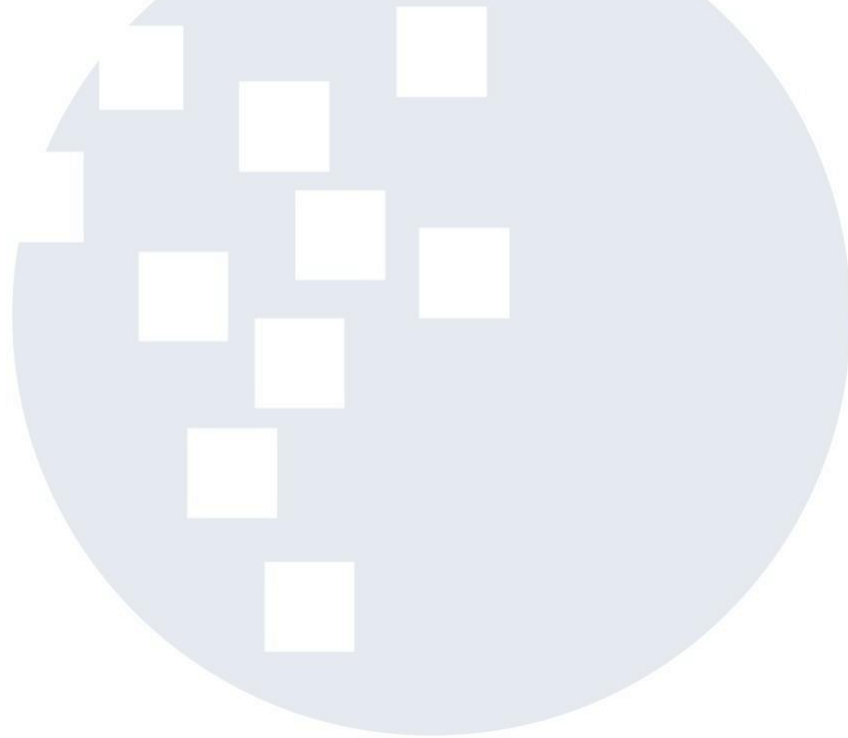
Sebagai tindak lanjut, penelitian ini dapat dikembangkan lebih lanjut dengan menerapkan pendekatan transfer learning atau memanfaatkan model bahasa terlatih sebelumnya (seperti BERT) untuk mengevaluasi kemungkinan peningkatan performa. Selain itu, integrasi dengan data secara real-time dan penerapan analisis aspek (aspect-based sentiment analysis) juga menjadi arah pengembangan yang potensial di masa mendatang.

5.2 Saran

Berdasarkan hasil penelitian ini, beberapa saran yang dapat dijadikan acuan untuk pengembangan studi selanjutnya antara lain:

1. Disarankan menggunakan data dari berbagai platform seperti Twitter, Instagram, atau marketplace digital untuk menguji kemampuan generalisasi model dengan ragam gaya bahasa dan konteks sosial yang beragam.
2. Perlu dilakukan evaluasi model deep learning seperti GRU, Transformer, serta model pretrained seperti BERT dan RoBERTa. Bandingkan juga dengan model klasik seperti Random Forest, XGBoost, dan LightGBM untuk melihat keunggulan tiap metode.
3. DANN dapat membantu mengatasi perbedaan distribusi antara data latih dan uji dari sumber berbeda, sehingga meningkatkan generalisasi model dalam skenario domain adaptasi.

4. Analisis aspek memungkinkan pengenalan opini terhadap fitur spesifik dalam ulasan panjang dan kompleks, sehingga menghasilkan informasi sentimen yang lebih detail dan bernilai.



UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA

DAFTAR PUSTAKA

- [1] S. Megawati, "Pengembangan Sistem Teknologi Internet of Things Yang Perlu Dikembangkan Negara Indonesia," *J. Inf. Eng. Educ. Technol.*, vol. 5, no. 1, pp. 19–26, 2021, doi: 10.26740/jieet.v5n1.p19-26.
- [2] H. Gede and P. Ayi, "Pemanfaatan Framework Laravel Dalam Pembangunan Aplikasi E-Travel Berbasis Website," *Konferensi Nasional Sistem Informasi STMIK Atma Luhur Pangkalpinang. Fakultas Teknik Unpas*, pp. 1329–1334, 2018.
- [3] "Negara dengan Jumlah Pengguna Internet Tertinggi (2024)." Accessed: Mar. 09, 2025. [Daring]. Available: <https://explodingtopics.com/blog/countries-internet-users>
- [4] "Deretan Aplikasi Kencan Daring Paling Banyak Diunduh Secara Global 2023, Tinder Juaranya." Accessed: Jan. 25, 2025. [Daring]. Available: <https://databoks.katadata.co.id/teknologi-telekomunikasi/statistik/f68b2e2a2e2067d/deretan-aplikasi-kencan-daring-paling-banyak-diunduh-secara-global-2023-tinder-juaranya>
- [5] A. Lidwina, "Tinder, Aplikasi Kencan Daring Paling Banyak Digunakan di Indonesia," *Databoks*. Accessed: Sep. 20, 2023. [Daring]. Available: <https://databoks.katadata.co.id/datapublish/2021/02/24/tinder-aplikasi-kencan-daring-paling-banyak-digunakan-di-indonesia>
- [6] "Ini Aplikasi Kencan Daring Terpopuler di Indonesia Awal 2024." Accessed: Jan. 16, 2025. [Daring]. Available: <https://databoks.katadata.co.id/teknologi-telekomunikasi/statistik/9e26634a2892b0b/ini-aplikasi-kencan-daring-terpopuler-di-indonesia-awal-2024>
- [7] "Penggunaan Aplikasi Dating dalam Perubahan Tingkah Laku Seseorang - Humaniora - www.indonesiana.id." Accessed: Sep. 20, 2023. [Daring]. Available: <https://www.indonesiana.id/read/160615/penggunaan-aplikasi-dating-dalam-perubahan-tingkah-laku-seseorang>
- [8] "Korban Penipuan Aplikasi Kencan di Jakpus Tersebar di Vietnam-Thailand." Accessed: Jun. 05, 2025. [Daring]. Available: <https://news.detik.com/berita/d-7753771/korban-penipuan-aplikasi-kencan-di-jakpus-tersebar-di-vietnam-thailand>
- [9] "Aplikasi Dating Dapat Berpengaruh pada Kesehatan Mental." Accessed: Jun. 05, 2025. [Daring]. Available: <https://www.halodoc.com/artikel/aplikasi-dating-dapat-berpengaruh-pada-kesehatan-mental?srsId=AfmBOorrY3bRSIbtSG-hTZH5LJDzvGTgnyz55SdXPUfez3iPFDjrBS94>
- [10] "Waspada Penipuan Berkedok Aplikasi Kencan Daring, Kenali Ciri-cirinya! - Regional Liputan6.com." Accessed: Jun. 03, 2025. [Daring].

Available: <https://www.liputan6.com/regional/read/5557192/waspada-penipuan-berkedok-aplikasi-kencan-daring-kenali-ciri-cirinya>

- [11] N. U. R. J. Qothrunnada et al., “Analisis sentimen ulasan pengguna aplikasi zoom menggunakan metode naïve bayes,” 2023.
- [12] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, and W. Gata, “Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi,” *J. Teknoinfo*, vol. 14, no. 2, p. 115, 2020, doi: 10.33365/jti.v14i2.679.
- [13] A. K. Burhanudin, “Analisis Attitude Terhadap Penunjukan Basuki Tjahaja Purnama Sebagai Komisaris Utama Pertamina dalam Kolom Komentar Instagram Harian Kompas,” *SOSIOHUMANIORA J. Ilm. Ilmu Sos. Dan Hum.*, vol. 6, no. 1, pp. 26–37, 2020, doi: 10.30738/sosio.v6i1.6329.
- [14] A. Tholib and Z. Arifin, “Analisis Sentimen Terhadap Ulasan Aplikasi Shopee di Google Play Store Menggunakan Metode TF-IDF dan Long Short-Term Memory (LSTM),” vol. 6, no. 2, pp. 371–381, 2024, doi: 10.33650/jecom.v4i2.
- [15] C. G. Indrayanto, D. E. Ratnawati, and B. Rahayudi, “Analisis Sentimen Data Ulasan Pengguna Aplikasi MyPertamina di Indonesia pada Google Play Store menggunakan Metode Random Forest,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 3, pp. 1131–1139, 2023, [Daring]. Available: <http://j-ptiik.ub.ac.id>
- [16] H. M. Alghamdi, S. H. A. Hamza, A. M. Mashraqi, and S. Abdel-Khalek, “Seeker Optimization with Deep Learning Enabled Sentiment Analysis on Social Media,” *Comput. Mater. Contin.*, vol. 73, no. 3, pp. 5985–5999, 2022, doi: 10.32604/cmc.2022.031732.
- [17] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” *Artif. Intell. Rev.*, vol. 55, no. 7, pp. 5731–5780, 2022, doi: 10.1007/s10462-022-10144-1.
- [18] F. R. Irawan, A. Jazuli, and T. Khotimah, “Analisis Sentimen Terhadap Pengguna Gojek Menggunakan Metode K-Nearset Neighbors Sentiment Analysis of Gojek Users Using K-Nearest Neighbor,” *JIKO (Jurnal Inform. dan Komputer)*, vol. 5, no. 1, pp. 62–68, 2022, doi: 10.33387/jiko.
- [19] M. Ngafifi, “Kemajuan Teknologi Dan Pola Hidup Manusia Dalam Perspektif Sosial Budaya,” *J. Pembang. Pendidik. Fondasi dan Apl.*, vol. 2, no. 1, pp. 33–47, 2014, doi: 10.21831/jppfa.v2i1.2616.
- [20] S. Jayachandran and A. Dumala, “Recurrent neural network based sentiment analysis of social media data during corona pandemic under national lockdown,” *J. Intell. Fuzzy Syst.*, vol. 44, no. 2, pp. 2131–2146, 2023, doi: 10.3233/JIFS-221883.
- [21] K. Lan, D. tong Wang, S. Fong, L. sheng Liu, K. K. L. Wong, and N. Dey,

- “A Survey of Data Mining and Deep Learning in Bioinformatics,” *J. Med. Syst.*, vol. 42, no. 8, pp. 1–20, 2018, doi: 10.1007/s10916-018-1003-9.
- [22] C. F. Prima, I. N. Suarsana, and N. M. Wiasti, “Tinder Sebagai Platform Pencarian Jodoh di Zaman Digital,” *Humanis*, vol. 25, no. 3, p. 379, 2021, doi: 10.24843/jh.2021.v25.i03.p15.
- [23] “5 Dampak Negatif Daring Dating yang Bisa Kamu Alami.” Accessed: Sep. 20, 2023. [Daring]. Available: <https://www.idntimes.com/life/relationship/arya-sarimata/dampak-negatif-daring-dating-clc2>
- [24] J. J. D. La Cruz, N. M. Punyanunt-Carter, and J. S. Wrench, “Dating App Communication: Personal Characteristics, Motives and Behavioural Intent,” *Media Watch*, vol. 14, no. 2, pp. 131–154, 2023, doi: 10.1177/09760911231160240.
- [25] M. F. Thomas, A. Binder, and J. Matthes, “The psychological influence of dating app matches: The more matches the merrier?,” *New Media Soc.*, 2023, doi: 10.1177/14614448231161598.
- [26] M. Stoicescu and M. G. Flaherty, “Tinder and Time Work through the Lens of Gender: Temporal Agency, Technology, and Intimacy,” *Soc. Psychol. Q.*, p. 01902725231205860, Oct. 2023, doi: 10.1177/01902725231205860.
- [27] G. Bonilla-Zorita, M. D. Griffiths, and D. J. Kuss, “Dating App Use and Wellbeing: An Application-Based Pilot Study Employing Ecological Momentary Assessment and Objective Measures of Use,” *Int. J. Environ. Res. Public Health*, vol. 20, no. 9, 2023, doi: 10.3390/ijerph20095631.
- [28] K. R. Pratama, “Sejarah Tinder, Medium Pencarian Jodoh Daring,” *Kompas.Com*. Accessed: Sep. 20, 2023. [Daring]. Available: https://tekno.kompas.com/read/2022/04/13/18300007/sejarah-tinder-medium-pencarian-jodoh-daring?page=all#google_vignette
- [29] D. P. M. Kumar, J. A. R, J. V. J, D. B. M, and J. Manani, “Sentimental Analysis on Quick Commerce Platform Based on Customer Reviews in Google Play Store,” *Int. J. Res. Sci. Eng.*, vol. 2, no. 26, pp. 29–42, 2022, doi: 10.55529/ijrise.26.29.42.
- [30] A. Ligthart, C. Catal, and B. Tekinerdogan, “Systematic reviews in sentiment analysis: a tertiary study,” *Artif. Intell. Rev.*, vol. 54, no. 7, pp. 4997–5053, 2021, doi: 10.1007/s10462-021-09973-3.
- [31] Y. Chen and S. Skiena, “Building sentiment lexicons for all major languages,” in *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference, 2014*, pp. 383–389. doi: 10.3115/v1/p14-2063.
- [32] “Proses Data Mining KDD – School of Information Systems.” Accessed: Jun. 29, 2025. [Daring]. Available:

<https://sis.binus.ac.id/2021/09/30/proses-data-mining-kdd/>

- [33] H. Akbar and W. K. Sanjaya, "Kajian Performa Metode Class Weight Random Forest pada Klasifikasi Imbalance Data Kelas Curah Hujan," *J. Sains, Nalar, dan Apl. Teknol. Inf.*, vol. 3, no. 1, 2023, doi: 10.20885/snati.v3i1.30.
- [34] "Apa itu NLP (Natural Language Processing atau Pemrosesan Bahasa Alami)? | IBM." Accessed: Jan. 29, 2025. [Daring]. Available: <https://www.ibm.com/id-id/topics/natural-language-processing>
- [35] Y. I. Alzoubi, A. E. Topcu, and A. E. Erkaya, "Machine Learning-Based Text Classification Comparison: Turkish Language Context," *Appl. Sci.*, vol. 13, no. 16, p. 9428, 2023.
- [36] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Inf.*, vol. 10, no. 4, p. 150, 2019, doi: 10.3390/info10040150.
- [37] W. K. Sari, D. P. Rini, and R. F. Malik, "Text Classification Using Long Short-Term Memory," in 2019 International Conference on Electrical Engineering and Computer Science (ICECOS), 2019, pp. 150–155. doi: 10.1109/ICECOS47637.2019.8984558.
- [38] A. V.- Klasifikasi and E. Wajah, "Evaluasi optimizer adam dan rmsprop pada arsitektur vgg-19 klasifikasi ekspresi wajah manusia," vol. 10, no. 2, pp. 1414–1426, 2025.
- [39] R. Siringoringo, J. Jamaluddin, R. Perangin-angin, E. J. G. Harianja, G. Lumbantoruan, and E. N. Purba, "Model Bidirectional Lstm Untuk Pemrosesan Sekuensial Data Teks Spam," *METHOMIKA J. Manaj. Inform. dan Komputerisasi Akunt.*, vol. 7, no. 2, pp. 265–271, 2023, doi: 10.46880/jmika.vol7no2.pp265-271.
- [40] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *Eur. Phys. J. Spec. Top.*, vol. 228, no. 10, pp. 2313–2324, Oct. 2019, doi: 10.1140/EPJST/E2019-900046-X.
- [41] T. Akhir, "Analisis Sentimen Ulasan Pengguna Aplikasi Traveloka di Google Play Store Menggunakan LSTM Program Studi Sarjana Informatika Fakultas Informatika Universitas Telkom Bandung," 2023.
- [42] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *Eur. Phys. J. Spec. Top.*, vol. 228, no. 10, pp. 2313–2324, Oct. 2019, doi: 10.1140/EPJST/E2019-900046-X.
- [43] N. K. Senthil Kumar and N. Malarvizhi, "Bi-directional LSTM–CNN Combined method for Sentiment Analysis in Part of Speech Tagging (PoS)," *Int. J. Speech Technol.*, vol. 23, no. 2, pp. 373–380, 2020, doi: 10.1007/s10772-020-09716-9.

- [44] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 5, no. 2, pp. 697–711, 2021.
- [45] "Machine Learning and the Internet of Medical Things in Healthcare - Google Books." Accessed: Oct. 24, 2023. [Daring]. Available: https://books.google.co.id/books?hl=en&lr=&id=wHQJEAAAQBAJ&oi=fnd&pg=PP1&dq=machine+learning+and+the+internet+of+medical+things+in&ots=3RoczGzCh_&sig=pUp3Kjfvpgbm0eSeIjg2oHAUzjg&redir_esc=y#v=onepage&q=machine+learning+and+the+internet+of+medical+things+in&f=false
- [46] J. Tjen, "Pengaruh Jenis Stopwords terhadap Akurasi Model Multinomial Naïve Bayes dalam Proses Sentimen Analisis," pp. 13–22.
- [47] K. K. Dewi, I. Kaniawulan, and C. D. Lestari, "ANALISIS SENTIMEN PENGGUNA APLIKASI JAMSOSTEK MOBILE (JMO) PADA APPSTORE MENGGUNAKAN METODE NAIVE BAYES," *Simtek J. Sist. Inf. dan Tek. Komput.*, vol. 8, no. 2, pp. 333–338, 2023.
- [48] T. S. Gunawan et al., "Development of video-based emotion recognition using deep learning with Google Colab," *Telkomnika (Telecommunication Comput. Electron. Control.)*, vol. 18, no. 5, pp. 2463–2471, 2020, doi: 10.12928/TELKOMNIKA.v18i5.16717.
- [49] B. A. Maulana and N. K. S. Noriska, "The Use of Machine Learning in the Analysis of Pawnshop Digital Service on Customer Satisfaction," *Indones. J. Contemp. Multidiscip. Res.*, vol. 2, no. 4, pp. 681–702, 2023.
- [50] P. Virtanen et al., "SciPy 1.0: fundamental algorithms for scientific computing in Python (vol 17, pg 261, 2020)," *Nat. Methods*, vol. 17, no. 3, p. 352, 2020.
- [51] M. D. Hendriyanto, A. A. Ridha, and U. Enri, "Analisis Sentimen Ulasan Aplikasi Mola Pada Google Play Store Menggunakan Algoritma Support Vector Machine," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 5, no. 1, pp. 1–7, 2022, doi: 10.31539/intecom.v5i1.3708.
- [52] "Metode Data Mining – School of Information Systems." Accessed: May 19, 2025. [Daring]. Available: <https://sis.binus.ac.id/2023/06/08/metode-data-mining/>
- [53] P. Aditiya, U. Enri, and I. Maulana, "Analisis Sentimen Ulasan Pengguna Aplikasi klik indomaret Pada Situs Google Play Menggunakan Support Vector Machine," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 4, p. 1020, 2022, doi: 10.30865/jurikom.v9i4.4673.
- [54] A. A. Permana, W. A. Noviyanto, and D. A. Kristiyanti, "Sentimen Analisis Opini Masyarakat Terhadap UMKM Pada Media Sosial Twitter

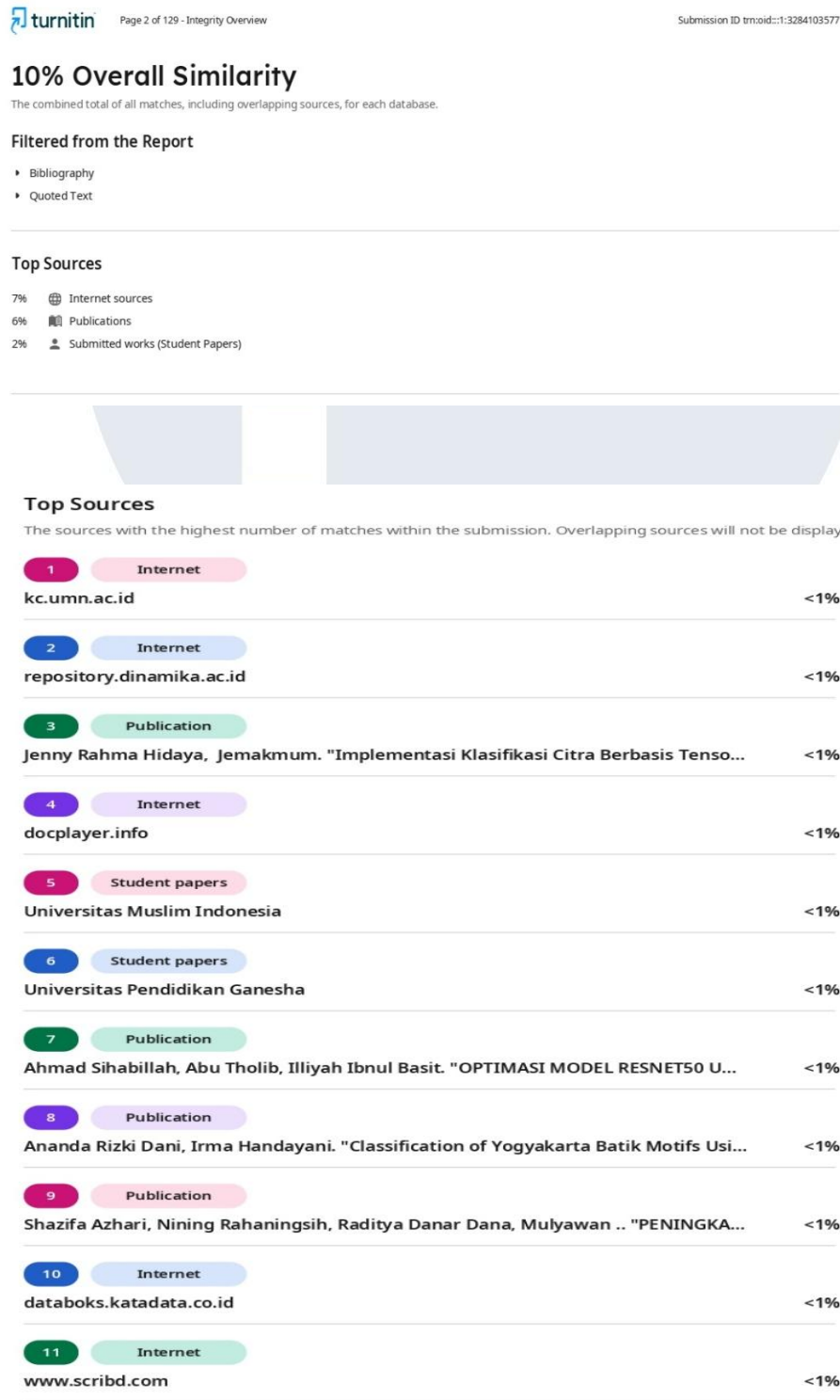
Dengan Metode Naïve Bayes Classifier,” J. Minfo Polgan, vol. 12, no. 1, pp. 163–170, 2023, doi: 10.33395/jmp.v12i1.12337.

- [55] M. F. Fahrezi and A. A. Permana, “Sentimen Analisis Opini Masyarakat Pada Sosial Media Twitter Terhadap Organisasi Aksi Cepat Tanggap Menggunakan Naïve Bayes Classifier,” JT J. Tek., vol. 11, no. 02, pp. 113–121, 2022, [Daring]. Available: <http://jurnal.umt.ac.id/index.php/jt/index>
- [56] K. Komentar and Y. Mata, “Implementasi algoritma convolutional neural network untuk analisis sentimen bacapres 2024 pada kolom komentar youtube mata najwa 1.,” vol. 9, no. 3, pp. 1431–1441, 2024.
- [57] L. S. Memory, L. Dan, S. A. Ahmad, and S. A. Ahmad, “Analisis sentimen komentar instagram menggunakan long short-term memory (lstm) dan word2vec,” 2023.
- [58] F. Fitroh and F. Hudaya, “Systematic Literature Review: Analisis Sentimen Berbasis Deep Learning,” J. Nas. Teknol. dan Sist. Inf., vol. 9, no. 2, pp. 132–140, 2023, doi: 10.25077/teknosi.v9i2.2023.132-140.



LAMPIRAN

Lampiran A Turnitin Similarity Report



12	Internet	discovery.researcher.life	<1%
13	Internet	jurnal.polibatam.ac.id	<1%
14	Internet	dspace.uui.ac.id	<1%
15	Internet	repository.upy.ac.id	<1%
16	Publication	Romario Onsu, Daniel Febrian Sengkey, Feisy Diane Kambey. "Implementasi Bi-LS...	<1%
17	Publication	Hendra Sutrisno, Nurul Anisa Sri Winarsih. "Klasifikasi Kategori Produk untuk Ma...	<1%
18	Internet	digilibadmin.unismuh.ac.id	<1%
19	Internet	ejurnal.umri.ac.id	<1%
20	Publication	Giesta Rahguna Putri, Muhammad Akbar Maulana, Samsul Bahri. "Perbandingan ...	<1%
21	Publication	Wanda Putra Ramadhan, Didi Juardi. "ANALISIS SENTIMEN ULASAN APLIKASI BTN...	<1%
22	Publication	Saiful Nur Budiman, Sri Lesanti, Erwan. "Analisis Sentimen Berdasarkan Hasil Re...	<1%
23	Internet	text-id.123dok.com	<1%
24	Publication	Randi Afif Afif, Aji Supriyanto, Rr. Fitri Damaryanti, Wahyu Prasetya Adi. "Analisis ...	<1%
25	Internet	ejurnal.ung.ac.id	<1%



Page 4 of 129 - Integrity Overview

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

26	Internet	jurnal.stkipggritulungagung.ac.id	<1%
27	Internet	eprints.umm.ac.id	<1%
28	Publication	Muhammad Sidiq Pramono, Aditya Permana Wibowo. "PENERAPAN CONVOLUTIO...	<1%
29	Publication	Rizky Saputra Cendikiawan, Ali Ibrahim, Mira Afrina, Rizka Dhini Kurnia. "Sentime...	<1%
30	Internet	www.nepjol.info	<1%
31	Publication	Dion Pratama Putra, Giri Wahyu Wiriasto, Paniran Paniran. "Detection of Lumpy ...	<1%
32	Internet	123dok.com	<1%
33	Publication	Gagas Ezhar Rahmayadi, Deny Haryadi, R. Bhima Danniswara, Valentino Kenny S...	<1%
34	Internet	cyberhub.id	<1%
35	Publication	Galih Hermawan. "Klasifikasi Pengemudi Terganggu Berdasarkan Citra Menggun...	<1%
36	Publication	Patria Gita Laksamana, Willy Prihartono, Fathurrohman .. "IMPLEMENTASI NAÏVE ...	<1%
37	Internet	repository.teknokrat.ac.id	<1%
38	Publication	Fanny Rahmasari, Nining Rahaningsih, Raditya Danar Dana, Cep Lukman Rohmat...	<1%
39	Internet	ejurnal.politeknikpratama.ac.id	<1%

40	Publication	Alfi Syahri, Angraini Angraini, Fitriani Muttakin. "Sentimen Analisis Pada Ulasan A...	<1%
41	Publication	Erzal Syahreza Aswir, Wasilah Wasilah. "Implementasi YOLOv3 Menggunakan Fit...	<1%
42	Student papers	UIN Walisongo	<1%
43	Student papers	Universitas Budi Luhur	<1%
44	Internet	eksplora.stikom-bali.ac.id	<1%
45	Internet	www.rctipus.com	<1%
46	Publication	Luthfi Krisna Bayu, Tutut Wuriyanto. "Analisis sentimen mobil listrik mengguna...	<1%
47	Publication	Rizal Dwi Rizki Apriliansyah, Rini Astuti, Willy Prihartono, Ryan Hamonangan. "PE...	<1%
48	Internet	smart.stmikplk.ac.id	<1%
49	Publication	Ratu Nurmalika, Makmun Makmun, Bambang Yulianto, Ichsan Mursidah, Dhian ...	<1%
50	Student papers	Universitas Andalas	<1%
51	Publication	Fitri Khoirunnisa, Salman Topiq. "ANALISIS SENTIMEN TERHADAP KEPERCAYAAN M...	<1%
52	Student papers	University of Wollongong	<1%
53	Publication	Wildan Amru Hidayat, Vinna Rahmayanti Setyaning Nastiti. "PERBANDINGAN KIN...	<1%



Page 6 of 129 - Integrity Overview



54	Publication	Wulandari Juslan, Alva Hendi Muhammad. "Evaluasi Kinerja Metode Peningkatan ..."	<1%
55	Internet	repo.itera.ac.id	<1%
56	Internet	aimos.ugm.ac.id	<1%
57	Internet	conference.upgris.ac.id	<1%
58	Publication	Rivan Kurnia, Teguh Nurhadi Suharsono. "Sentimen Analisis Review Pada Aplikasi..."	<1%
59	Student papers	Universitas Brawijaya	<1%
60	Internet	journal.nspn.org.ng	<1%
61	Internet	sinta.unud.ac.id	<1%
62	Student papers	Universitas Maritim Raja Ali Haji	<1%
63	Internet	johannessimatupang.wordpress.com	<1%
64	Internet	siat.ung.ac.id	<1%
65	Publication	Bayu Samodera, Kartini Kartini, Muhammad Muharrom Al Haromainy. "IMPLEME..."	<1%
66	Publication	Farin Junita Fauzan, M Afdal, Rice Novita, Mustakim. "PENERAPAN MACHINE LE..."	<1%
67	Internet	adoc.tips	<1%



Page 7 of 129 - Integrity Overview

UNIVERSITAS
MULTIMEDIA
NUSANTARA

68	Internet	catholicislander.com	<1%
69	Internet	pt.scribd.com	<1%
70	Internet	toffeeev.com	<1%
71	Internet	www.coursehero.com	<1%
72	Publication	Fajar Astuti Hermawati, Nenden Siti Fatonah, Hermawan Ali Mangambali. "Aplika...	<1%
73	Internet	es.scribd.com	<1%
74	Internet	industri.uin-suka.ac.id	<1%
75	Internet	jurnal.untan.ac.id	<1%
76	Internet	repository.ub.ac.id	<1%
77	Internet	repository.unhas.ac.id	<1%
78	Internet	research-report.umm.ac.id	<1%
79	Internet	www.berotak.com	<1%
80	Publication	Alvina Lia, Abdul Rahim, Taghfirul Azhima Yoga Siswa. "ANALISIS SENTIMEN APLI...	<1%
81	Publication	Tim Muhaemin Abdul Azis, Tri Wahyudi. "Analisa Sentimen Rencana Pemindahan ...	<1%



Page 8 of 129 - Integrity Overview



82	Publication	Khusnul Khotimah, Martanto Martanto, Arif Rinaldi Dikananda, Ahmad Rifa'i. "AN...	<1%
83	Publication	Muhammad Fauzan Novriandy, Basuki Rahmat, Achmad Junaidi. "KLASIFIKASI CI...	<1%
84	Internet	artikelpendidikan.id	<1%
85	Internet	docobook.com	<1%
86	Internet	ejurnal.seminar-id.com	<1%
87	Internet	ejurnal.undana.ac.id	<1%
88	Internet	eprints.binadarma.ac.id	<1%
89	Internet	garuda.kemdikbud.go.id	<1%
90	Internet	journal.unpar.ac.id	<1%
91	Internet	jurnal.unmer.ac.id	<1%
92	Internet	ojs.unud.ac.id	<1%
93	Internet	pintu.co.id	<1%
94	Internet	sekolahmalam.wordpress.com	<1%
95	Internet	seminar.iaii.or.id	<1%

96	Internet	slamet10018075.blogspot.com	<1%
97	Internet	utamanews.com	<1%
98	Internet	www.internationaljournalofspecialeducation.com	<1%
99	Internet	www.repository.trisakti.ac.id	<1%
100	Publication	Ryo Aldyanto, Baswara Yua Kristama. "Dampak Proses Keterbukaan Diri Penggu...	<1%
101	Publication	Zumhur Alamin, Sutriawan, Siti Mutmainah, Muhammad Hayun. "Optimasi Ekst...	<1%

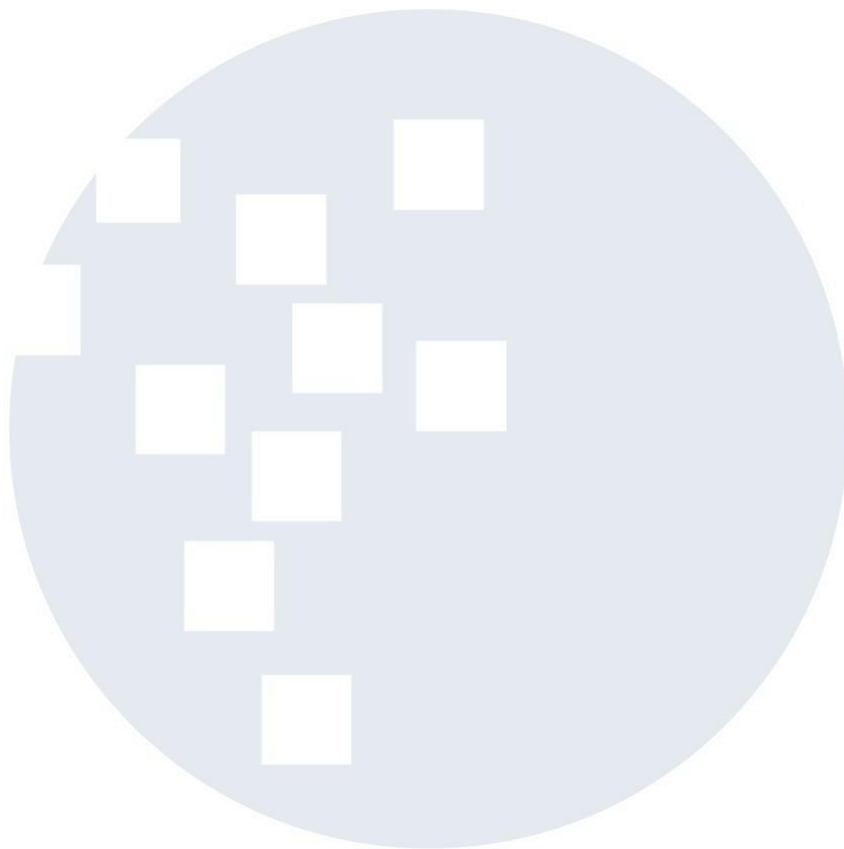
Lampiran B Form Konsultasi Bimbingan

Form Bimbingan Skripsi Program Studi Information Systems Semester Genap 2024/2025



Nama : GALIH ISA YUDHISTIRA
NIM : 00000048134
Angkatan : 2020
Dosen Pembimbing : Dr. Erick Fernando (Pembimbing)

No	Tanggal	Jam	Keterangan	Tanggal Approval
1	23 Februari 2025	11:00	Bimbingan Pertama : Penentuan Algoritma	25 Mei 2025 11:9
2	25 Februari 2025	11:00	Bimbingan Kedua : Memperbaiki Bab 1	25 Mei 2025 11:9
3	03 Maret 2025	10:59	Bimbingan Ketiga : Proses Penyelesaian Bab 1	25 Mei 2025 11:9
4	10 Maret 2025	10:59	Bimbingan Keempat : Memulai Bab 2	25 Mei 2025 11:9
5	17 Maret 2025	10:59	Bimbingan Kelima : Perbaikan Bab 2	25 Mei 2025 11:9
6	24 Maret 2025	10:59	Bimbingan Keenam : Memulai Bab 3	25 Mei 2025 11:9
7	31 Maret 2025	10:59	Bimbingan Ketujuh : Perbaikan Bab 3	25 Mei 2025 11:9
8	01 April 2025	10:59	Bimbingan Kedelapan : Penyelesaian Bab 3	25 Mei 2025 11:9
9	07 April 2025	10:59	Bimbingan Kesembilan : Memulai Bab 4	25 Mei 2025 11:9
10	14 April 2025	10:59	Bimbingan Kesepuluh : Proses Bab 4	25 Mei 2025 11:9
11	21 April 2025	10:59	Bimbingan Kesebelas : Penyelesaian Bab 4	25 Mei 2025 11:9
12	28 April 2025	10:59	Bimbingan Keduabelas : Penyelesaian Bab 4	25 Mei 2025 11:9
13	05 Mei 2025	10:59	Bimbingan Ketigabelas : Penyelesaian Bab 4	25 Mei 2025 11:9
14	12 Mei 2025	10:59	Bimbingan Keempatbelas : Memulai Bab 5	25 Mei 2025 11:7
15	20 April 2025	10:59	Bimbingan Kelimabelas : Perbaikan dan Penyelesaian Seluruh Bab	25 Mei 2025 11:9



FORMULIR KONSULTASI SKRIPSI – FAKULTAS TEKNIK & INFORMATIKA

Dosen Pembimbing :

Jurusan :

Semester :

Nama :

NIM :

Tanggal Konsultasi	Agenda/Pokok Bahasan	Saran Perbaikan	Paraf Dosen Pembimbing

Catatan : Form ini wajib dibawa pada saat konsultasi & dilampirkan didalam skripsi (**Minimal 8 kali Konsultasi**)

Tangerang,20....

Dosen Pembimbing



FORMULIR KONSULTASI SKRIPSI – FAKULTAS TEKNIK & INFORMATIKA



Dosen Pembimbing :
Jurusan :
Semester :
Nama :
NIM :

Tanggal Konsultasi	Agenda/Pokok Bahasan	Saran Perbaikan	Paraf Dosen Pembimbing

Catatan : Form ini wajib dibawa pada saat konsultasi & dilampirkan di dalam skripsi (**Minimal 8 kali Konsultasi**)

Tangerang,20....

Dosen Pembimbing



UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA



UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA