

**ANALISIS PERBANDINGAN ALGORITMA XGBOOST DAN
BiLSTM DALAM KLASIFIKASI SENTIMEN PUBLIK
TERHADAP KEBIJAKAN PENEMPATAN DANA
PEMERINTAH DI BANK**

SKRIPSI

DISUSUN OLEH

WAHYU PURNOMO

NPM. 2109020059



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2026

**ANALISIS PERBANDINGAN ALGORITMA XGBOOST DAN
BiLSTM DALAM KLASIFIKASI SENTIMEN PUBLIK
TERHADAP KEBIJAKAN PENEMPATAN DANA
PEMERINTAH DI BANK**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer
(S.Kom) dalam Program Studi Teknologi Informasi, pada Fakultas Ilmu Komputer
dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara**

WAHYU PURNOMO

NPM. 2109020059

**PROGRAM STUDI TEKNOLOGI INFROMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2026

LEMBAR PENGESAHAN

Judul Skripsi : Analisis Perbandingan Algoritma Xgboost Dan
Bilstm Dalam Klasifikasi Sentimen Publik
Terhadap Kebijakan Penempatan Dana
Pemerintah Di Bank

Nama Mahasiswa : Wahyu Purnomo
NPM : 2109020059
Program Studi : Teknologi Infromasi

Menyetujui
Komisi Pembimbing



(Mhd. Basri, S.Si, M.Kom)
NIDN. 0111078802

Ketua Program Studi



(Fatma Sari Hutagalang, M.Kom)
NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

**ANALISIS PERBANDINGAN ALGORITMA XGBOOST DAN
BiLSTM DALAM KLASIFIKASI SENTIMEN PUBLIK
TERHADAP KEBIJAKAN PENEMPATAN DANA
PEMERINTAH DI BANK**

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 6 Februari 2026

Yang membuat pernyataan



Wahyu Purnomo

NPM. 2109020059

**PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK
KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Wahyu Purnomo
NPM : 2109020059
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**ANALISIS PERBANDINGAN ALGORITMA XGBOOST DAN
BiLSTM DALAM KLASIFIKASI SENTIMEN PUBLIK
TERHADAP KEBIJAKAN PENEMPATAN DANA
PEMERINTAH DI BANK**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 6 Februari 2026

Yang membuat pernyataan



Wahyu Purnomo

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Wahyu Purnomo
Tempat dan Tanggal Lahir : Sibolga, 9 Desember 2003
Alamat Rumah : Jl. Ketapang No.42, Sibolga
Telepon/Faks/HP : +62 82294200313
E-mail : Purnomow814@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD : SD Negeri 085115 Sibolga TAMAT: 2015
SMP : SMP Negeri 2 Sibolga TAMAT: 2018
SMA : SMA Negeri 1 Sibolga TAMAT: 2021

KATA PENGANTAR



Pendahuluan

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Akrim, M.Pd, Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Dr. Firahmi Rizky, S.Kom.,M.Kom Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
4. Bapak Mhd. Basri, S.Si, M.Kom Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
5. Ibu Fatma Sari Hutagalung, M.Kom sebagai Ketua Program Studi Teknologi Informasi
6. Bapak Okvi Nugroho, S.Kom., M.Kom sebagai Sekretaris Program Studi Teknologi Informasi
7. Ibu Indah Purnama Sari, ST.,M.Kom sebagai Pembimbing Akademik saya yang telah memberikan arahan dan masukan kepada saya selama saya masih berkuliah.
8. Bapak Mhd. Basri, S.Si,M.Kom sebagai Pembimbing yang telah meluangkan waktu, tenaga, dan pikiran untuk memberikan bimbingan, arahan, serta motivasi selama proses penyusunan skripsi ini.
9. Kedua orang tua penulis Sumanto dan Ade Irmayanti Simamora, S.E., yang selalu memberikan doa, dukungan baik secara moral dan juga material, kasih sayang, serta motivasi yang tiada henti yang sangat berarti bagi saya.

10. Kedua adik saya Liffa Dwi Kartika dan juga Dinda Tri Ariyanti yang selalu memberikan dukungan dan juga motivasi kepada saya agar saya selalu dapat mengerjakan skripsi ini dengan baik dan tepat waktu.
11. Alya Putri Ramadhani Halim yang telah menjadi pendamping penulis serta memberikan dukungan, semangat, dan bantuan selama proses penyusunan skripsi ini.
12. Krisna Aditya, Muhammad Haekal, Nanda Rafikhi Azhari Lubis, Abdillah Putra Harkiah, Rusdan Hamid, Rafly Tianizar, Muhammad Khalil Gibran, Riski Ananda, Aditya Al - Afghani, Dan Muhammad Reza Mahendra yang telah menjadi teman seperjuangan penulis serta memberikan dukungan, kebersamaan, dan bantuan selama masa perkuliahan hingga proses penyusunan skripsi ini selesai.
13. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini

ANALISIS PERBANDINGAN ALGORITMA XGBOOST DAN BiLSTM DALAM KLASIFIKASI SENTIMEN PUBLIK TERHADAP KEBIJAKAN PENEMPATAN DANA PEMERINTAH DI BANK

ABSTRAK

Penempatan dana pemerintah di sektor perbankan merupakan salah satu kebijakan strategis yang bertujuan untuk menjaga stabilitas ekonomi dan meningkatkan likuiditas perbankan. Namun, kebijakan ini memunculkan beragam respons dari masyarakat yang tercermin dalam opini publik di media sosial. Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja algoritma Extreme Gradient Boosting (XGBoost) dan Bidirectional Long Short-Term Memory (BiLSTM) dalam melakukan klasifikasi sentimen publik terhadap kebijakan tersebut. Data yang digunakan berasal dari platform media sosial yang dikumpulkan melalui teknik scraping, kemudian melalui tahap preprocessing yang meliputi tokenisasi, normalisasi, dan penghapusan noise. Selanjutnya, fitur diekstraksi menggunakan metode Term Frequency-Inverse Document Frequency (TF-IDF) untuk XGBoost dan word embedding untuk BiLSTM. Evaluasi kinerja model dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score. Hasil penelitian menunjukkan bahwa model BiLSTM memiliki kemampuan yang lebih baik dalam menangkap konteks kalimat sehingga menghasilkan akurasi yang lebih tinggi dibandingkan XGBoost. Namun, XGBoost menunjukkan keunggulan dalam hal efisiensi komputasi dan waktu pelatihan. Penelitian ini memberikan kontribusi dalam pemilihan model klasifikasi sentimen yang optimal untuk analisis opini publik berbasis data teks, khususnya dalam konteks kebijakan ekonomi pemerintah.

Kata Kunci: Analisis sentimen, XGBoost, BiLSTM, klasifikasi teks, kebijakan publik, media sosial

A COMPARATIVE ANALYSIS OF THE XGBOOST AND BILSTM ALGORITHMS IN CLASSIFYING PUBLIC SENTIMENT TOWARD GOVERNMENT FUND PLACEMENT POLICIES IN BANKS

ABSTRACT

The government's placement of funds in the banking sector is a strategic policy aimed at maintaining economic stability and enhancing banking liquidity. However, this policy has elicited a variety of responses from the public, as reflected in public opinion on social media. This study aims to analyze and compare the performance of the Extreme Gradient Boosting (XGBoost) and Bidirectional Long Short-Term Memory (BiLSTM) algorithms in classifying public sentiment toward this policy. The data used was sourced from social media platforms and collected via web scraping, followed by a preprocessing stage that included tokenization, normalization, and noise removal. Next, features were extracted using the Term Frequency-Inverse Document Frequency (TF-IDF) method for XGBoost and word embedding for BiLSTM. Model performance was evaluated using accuracy, precision, recall, and F1-score metrics. The results of the study indicate that the BiLSTM model is better at capturing sentence context, resulting in higher accuracy compared to XGBoost. However, XGBoost demonstrates advantages in terms of computational efficiency and training time. This study contributes to the selection of an optimal sentiment classification model for text-based public opinion analysis, particularly in the context of government economic policies.

Keywords: Sentiment analysis, XGBoost, BiLSTM, text classification, public policy, social media

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGANAKADEMIS	iv
RIWAYAT HIDUP	v
KATA PENGANTAR	vi
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	6
BAB II LANDASAN TEORI	8
2.1 Analisis Sentimen	8
2.2 Tantangan Analisis Sentimen Bahasa Indonesia.....	9
2.3 Algoritma XGBoost	10
2.4 Algoritma BiLSTM.....	13
2.5 Preprocessing Teks Untuk Analisis Sentimen	17
2.6 Metrik Evaluasi Klasifikasi.....	18
2.7 Penelitian Terdahulu	19
2.8 Analisis GAP.....	26
2.8.1 Gap Konteks Penelitian.....	26
2.8.2 Gap Perbandingan Algoritma.....	27
2.8.3 Gap Metrik Evaluasi Komprehensif.....	28
2.8.4 Gap Preprocessing untuk Bahasa Indonesia	28
2.8.5 Gap Implikasi Praktis untuk Kebijakan Publik.....	29
BAB III METODOLOGI PENELITIAN.....	32
3.1 Alur Penelitian	32
3.2 Alur Kerja Penelitian.....	32
3.3 Pengumpulan Data	34
3.3.1 Sumber Data.....	34

3.3.2	Kriteria Data.....	35
3.3.3	Pelabelan Data.....	36
3.4	Preprocessing Teks.....	36
3.4.1	Cleaning	36
3.4.2	Case Folding	37
3.4.3	Normalisasi	37
3.4.4	Tokenization.....	37
3.4.5	Stopword Removal.....	37
3.4.6	Stemming	38
3.5	Ekstraksi Fitur	38
3.5.1	Ekstraksi Fitur untuk XGBoost.....	38
3.5.2	Ekstraksi Fitur untuk BiLSTM	39
3.5.3	Alur Pemrosesan Data.....	39
3.6	Implementasi Model.....	41
3.6.1	Implementasi Model XGBoost	41
3.6.2	Implementasi Model BiLSTM	46
3.7	Evaluasi Model.....	52
3.7.1	Metrik Evaluasi	52
3.7.2	Prosedur Evaluasi.....	53
3.8	Analisis Komparatif	54
3.8.1	Perbandingan Akurasi dan Metrik Klasifikasi.....	54
3.8.2	Analisis Kelebihan dan Kekurangan.....	54
3.9	Penarikan Kesimpulan dan Rekomendasi.....	54
3.10	Lingkungan Pengembangan	55
BAB IV HASIL DAN PEMBAHASAN		56
4.1	Analisis Masalah	56
4.2	Perancangan Sistem	57
4.2.1	Pengumpulan Data	57
4.2.2	Preprocessing Data.....	58
4.2.3	Pembagian Data	60
4.2.4	Arsitektur Model	61
4.3	Implementasi.....	63
4.3.1	Environment dan Library	63
4.3.2	Proses Training Model	64
4.4	Pengujian dan Evaluasi Sistem	68
4.4.1	Metrik Evaluasi	68

4.4.2	Hasil Evaluasi Model	68
4.4.3	Confusion Matrix	75
4.4.4	Analisis Performa Model	76
4.4.5	Pengujian Prediksi Manual	81
BAB V PENUTUP.....		83
5.1	Kesimpulan	83
5.2	Saran.....	84
DAFTAR PUSTAKA		i

DAFTAR TABEL

TABEL 2.1	Ringkasan Penelitian Terdahulu.....	20
TABEL 2.2	Analisis Kesenjangan dan Kontribusi.....	29
TABEL 3.1	Data Ulasan.....	33
TABEL 3.2	Perbandingan Mekanisme Kerja XGBoost dan BiLSTM	50
TABEL 4.1	Pengumpulan Data.....	57
TABEL 4.2	Pembagian Data.....	61
TABEL 4.3	Hasil Evaluasi Model.....	69
TABEL 4.4	Classification Report – XGBoost.....	70
TABEL 4.5	Classification Report – BiLSTM.....	73

DAFTAR GAMBAR

GAMBAR 2.1	Arsitektur Algoritma XGBoost.....	10
GAMBAR 2.2	Flowchart Algoritma XGBoost.....	12
GAMBAR 2.3	Arsitektur Algoritma BiLSTM.....	12
GAMBAR 2.4	Flowchart Algoritma BiLSTM.....	14
GAMBAR 3.1	Alur kerja penelitian yang menunjukkan alur dari Pengumpulan data hingga analisis hasil dan rekomendasi	32
GAMBAR 4.1	Proses memuat data.....	58
GAMBAR 4.2	Proses preprocessing awal.....	60
GAMBAR 4.3	Proses import pustaka.....	64
GAMBAR 4.4	Proses training model BiLSTM.....	65
GAMBAR 4.5	Proses training model XGBoost.....	67
GAMBAR 4.6	Kurva evaluasi model XGBoost.....	69
GAMBAR 4.7	Kurva evaluasi model BiLSTM.....	69
GAMBAR 4.8	Proses evaluasi model XGBoost.....	72
GAMBAR 4.9	Proses evaluasi model BiLSTM.....	74
GAMBAR 4.10	Analisis Confusion Matrix XGBoost dan BiLSTM.....	75
GAMBAR 4.11	Proses analisis komparasi model XGBoost dan BiLSTM	76

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Penempatan dana pemerintah di perbankan merupakan kebijakan strategis yang memiliki implikasi signifikan terhadap stabilitas ekonomi nasional dan kepercayaan publik terhadap pengelolaan keuangan negara. Media sosial dan platform digital telah menjadi ruang publik utama bagi masyarakat untuk mengekspresikan pendapat, kritik, dan dukungan terhadap berbagai kebijakan pemerintah. Youtube, dan platform berita online menghasilkan volume data teks yang sangat besar setiap harinya, menciptakan big data yang kaya akan informasi sentimen publik. Sentimen publik yang terekspresikan di media sosial memiliki karakteristik unik, termasuk penggunaan bahasa informal, singkatan, emoji, dan konteks politik-ekonomi yang spesifik. Pemahaman terhadap sentimen publik ini sangat penting karena dapat menjadi indikator persepsi masyarakat terhadap kredibilitas pemerintah dalam mengelola keuangan negara, mempengaruhi tingkat kepercayaan investor, serta memberikan masukan bagi pemerintah dalam menyesuaikan strategi komunikasi atau bahkan revisi kebijakan jika diperlukan.

Perkembangan teknologi kecerdasan buatan telah membuka peluang baru sebagai solusi untuk mengatasi tantangan analisis sentimen publik secara otomatis dan akurat. Pemerintah dapat memanfaatkan teknologi Natural Language Processing untuk memantau platform media sosial guna analisis sentimen secara real-time, memungkinkan pemahaman respons masyarakat

terhadap inisiatif kebijakan dan penyesuaian strategi komunikasi jika diperlukan (INA Solutions, 2024). Dua pendekatan algoritmik menunjukkan potensi signifikan: XGBoost (Extreme Gradient Boosting) sebagai metode ensemble learning dan BiLSTM (Bidirectional Long Short-Term Memory) sebagai arsitektur deep learning. XGBoost bekerja dengan gradient boosting yang dioptimalkan, menghasilkan prediksi akurat dengan efisiensi komputasi tinggi. BiLSTM mampu memproses informasi dari dua arah, menangkap konteks komprehensif dan memahami dependensi jarak jauh. Namun, belum ada penelitian komparatif sistematis untuk mengevaluasi performa kedua algoritma ini dalam konteks spesifik sentimen publik terhadap kebijakan ekonomi pemerintah Indonesia.

Volume data yang masif dari media sosial menciptakan tantangan signifikan dalam mengekstraksi informasi yang bermakna secara manual. Terdapat kesenjangan antara kebutuhan pemerintah akan pemahaman yang cepat dan akurat tentang respons publik dengan ketersediaan sistem analisis sentimen yang akurat dan real-time. Kesenjangan ini semakin kompleks ketika melibatkan konteks bahasa Indonesia yang memiliki tantangan khusus terkait kompleksitas morfologi, penggunaan bahasa campuran (code-mixing), dan ekspresi sentimen yang implisit. (Lin & Nuha, 2023) mengidentifikasi bahwa analisis sentimen dalam bahasa Indonesia memerlukan strategi deep learning hybrid untuk meningkatkan keandalan representasi teks. (Romadhony et al., 2024) menunjukkan kompleksitas pemrosesan bahasa Indonesia melalui analisis sentimen pada dataset ulasan produk berukuran besar yang terdiri dari lebih dari 700.000 ulasan.

Beberapa penelitian terdahulu menunjukkan keunggulan masing-masing algoritma. Untuk XGBoost, (Alabdulkarim et al., 2024) mencapai akurasi 96% pada dataset Amazon Fine Food Reviews. (Chandrasekaran et al., 2025) mengembangkan Hybrid LXGB Model mencapai akurasi 97,18% pada dataset CMU-MOSEI. (Nsaif & Abd, 2022) mencapai akurasi 95,161% dalam klasifikasi sentimen postingan politik. Untuk BiLSTM, (Xu et al., 2025) menunjukkan efektivitas BiLSTM dalam analisis sentimen kategori aspek. (Rahman et al., 2025) mencapai akurasi 92,36% pada dataset IMDb dengan RoBERTa-BiLSTM. (Zhao et al., 2024) mencapai F1-score 94,06% dengan MIBE-RoBERTa-FF-BiLSTM. (Talaat, 2023) menunjukkan tidak ada algoritma yang universal unggul, pemilihan metode bergantung pada karakteristik data dan konteks aplikasi. Dalam konteks kebijakan publik, (Russell et al., 2024) menggunakan analisis sentimen pada tweet kebijakan lockdown COVID-19, menemukan 76% tweet mengekspresikan sentimen negatif. (Khadija et al., 2024) mengkonfirmasi efektivitas analisis sentimen machine learning dalam mengidentifikasi perspektif publik terhadap kebijakan pemerintah di Indonesia. (Ahmadian et al., 2024) mencapai akurasi 93% dengan model hybrid IndoBERT-BiLSTM untuk bahasa Indonesia.

Berdasarkan kajian literatur, masih terdapat kesenjangan penelitian dalam perbandingan komprehensif antara XGBoost dan BiLSTM pada domain spesifik sentimen publik terhadap kebijakan ekonomi pemerintah Indonesia, khususnya penempatan dana pemerintah di bank. Kesenjangan tersebut mencakup belum adanya studi yang secara sistematis membandingkan kedua algoritma dengan dataset berbahasa Indonesia, metrik evaluasi yang

komprehensif (akurasi, precision, recall, F1-score), serta efisiensi komputasi dalam konteks aplikasi kebijakan publik keuangan. Penelitian ini diharapkan dapat mengisi kesenjangan tersebut dengan memberikan bukti empiris mengenai algoritma mana yang lebih efektif untuk klasifikasi sentimen publik terhadap kebijakan penempatan dana pemerintah di bank, sehingga dapat mendukung pengembangan sistem monitoring sentimen publik yang lebih akurat dan aplikatif bagi pengambil kebijakan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, penelitian ini difokuskan pada perbandingan performa algoritma XGBoost dan BiLSTM dalam mengklasifikasikan sentimen publik terhadap kebijakan penempatan dana pemerintah di bank. Rumusan masalah dalam penelitian ini adalah:

1. Bagaimana performa algoritma XGBoost dan BiLSTM dalam mengklasifikasikan sentimen publik terhadap kebijakan penempatan dana pemerintah di bank berdasarkan metrik akurasi, precision, recall, dan F1-score?
2. Algoritma manakah yang lebih efektif antara XGBoost dan BiLSTM untuk klasifikasi sentimen publik terhadap kebijakan penempatan dana pemerintah di bank ditinjau dari aspek akurasi, efisiensi waktu komputasi, dan kemampuan generalisasi model?

1.3 Batasan Masalah

Berikut ini batasan masalah penelitian ini

1. Data yang digunakan dalam penelitian ini berasal dari platform media sosial tertentu seperti YouTube dan TikTok, tidak mencakup platform

media sosial lain seperti Twitter/X, Facebook, Instagram, atau forum diskusi online.

2. Penelitian ini hanya membandingkan dua algoritma, yaitu XGBoost dan BiLSTM, untuk tugas klasifikasi sentimen.
3. Data dikumpulkan hanya untuk periode satu bulan setelah pengumuman resmi kebijakan tersebut, guna menangkap reaksi awal dari masyarakat.
4. Data yang digunakan dalam penelitian ini hanya mencakup kucuran dana awal sebesar 200 triliun rupiah, dan tidak memperhitungkan dana tambahan lain yang mungkin dikeluarkan oleh pemerintah.
5. Penelitian ini tidak melakukan filtering akun buzzer atau bot, sehingga hasil mencerminkan seluruh sentimen yang tersedia di platform tanpa memverifikasi keaslian akun.
6. Objek penelitian difokuskan pada komentar berbahasa Indonesia formal. Komentar yang dominan menggunakan bahasa daerah atau bahasa asing akan dieliminasi atau tidak menjadi fokus utama analisis untuk menjaga konsistensi fitur linguistik.
7. Analisis sentimen dilakukan pada tingkat kalimat, di mana satu komentar dianggap mewakili satu polaritas sentimen (Positif, Negatif, atau Netral). Penelitian ini tidak mencakup analisis emosi mendalam (seperti marah, sedih, gembira) atau deteksi sarkasme yang kompleks secara terpisah.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membandingkan dan membuktikan efektivitas algoritma XGBoost dan BiLSTM dalam klasifikasi sentimen publik terhadap kebijakan penempatan dana pemerintah di bank. Secara spesifik, tujuan penelitian ini adalah:

1. Membandingkan dan membuktikan algoritma mana yang lebih efektif antara XGBoost dan BiLSTM untuk klasifikasi sentimen publik terhadap kebijakan penempatan dana pemerintah di bank ditinjau dari aspek akurasi, efisiensi waktu komputasi, dan kemampuan generalisasi model.
2. Menganalisis algoritma mana yang lebih efektif antara XGBoost dan BiLSTM untuk klasifikasi sentimen publik terhadap kebijakan penempatan dana pemerintah di bank ditinjau dari aspek akurasi, efisiensi waktu komputasi, dan kemampuan generalisasi model.

1.5 Manfaat Penelitian

1. Memberikan kontribusi teoretis dengan memperkaya kajian analisis sentimen, khususnya perbandingan performa algoritma XGBoost dan BiLSTM pada data berbahasa Indonesia.
2. Menambah referensi ilmiah mengenai penerapan machine learning dan deep learning dalam konteks kebijakan ekonomi pemerintah, terutama kebijakan penempatan dana pemerintah di bank.
3. Memberikan informasi empiris bagi pemerintah atau instansi terkait mengenai algoritma yang lebih efektif untuk memantau sentimen publik serta menjadi acuan bagi pengembang sistem dalam merancang dan

mengimplementasikan sistem monitoring sentimen publik yang lebih akurat dan efisien.

4. Menjadi bahan rujukan bagi peneliti selanjutnya yang ingin mengembangkan atau memperluas penelitian serupa ke domain kebijakan publik lainnya.

BAB II

LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen merupakan salah satu cabang dari *Natural Language Processing* (NLP) yang bertujuan untuk mengidentifikasi, mengekstraksi, dan mengklasifikasikan opini atau sentimen yang terkandung dalam teks (Talaat, 2023). Dalam konteks kebijakan publik, analisis sentimen menjadi alat penting untuk memahami persepsi dan respons masyarakat terhadap kebijakan yang diimplementasikan oleh pemerintah. (Verma, 2022) menekankan bahwa analisis sentimen untuk layanan publik memerlukan pendekatan spesifik konteks yang dapat menangani kebutuhan warga melalui amalgamasi teknologi dan keterlibatan warga.

Analisis sentimen dapat dikategorikan ke dalam tiga tingkatan: *document-level*, *sentence-level*, dan *aspect-level* (Alqaryouti et al., 2024). Pada *document-level*, keseluruhan dokumen diklasifikasikan ke dalam satu kategori sentimen. Pada *sentence-level*, setiap kalimat dianalisis secara terpisah. Sedangkan pada *aspect-level*, analisis dilakukan terhadap aspek-aspek spesifik yang disebutkan dalam teks. Penelitian ini menggunakan pendekatan *document-level* untuk mengklasifikasikan sentimen publik terhadap kebijakan penempatan dana pemerintah di bank ke dalam tiga kategori: positif, negatif, dan netral.

(Russell et al., 2024) mendemonstrasikan penerapan analisis sentimen *machine learning* pada tweet terkait kebijakan lockdown COVID-19, menemukan bahwa 76% dari 1.609 tweet mengekspresikan sentimen negatif. Temuan ini menunjukkan pentingnya analisis sentimen dalam mengukur

respons masyarakat terhadap kebijakan pemerintah secara kuantitatif dan objektif. Dalam konteks Indonesia, (Khadija et al., 2024) mengkonfirmasi bahwa analisis sentimen berbasis *machine learning* efektif dalam mengidentifikasi perspektif publik terhadap kebijakan pemerintah, dengan aplikasi yang mencakup analisis kualitas udara, keluhan aspirasi publik, dan berbagai isu kebijakan lainnya.

2.2 Tantangan Analisis Sentimen Bahasa Indonesia

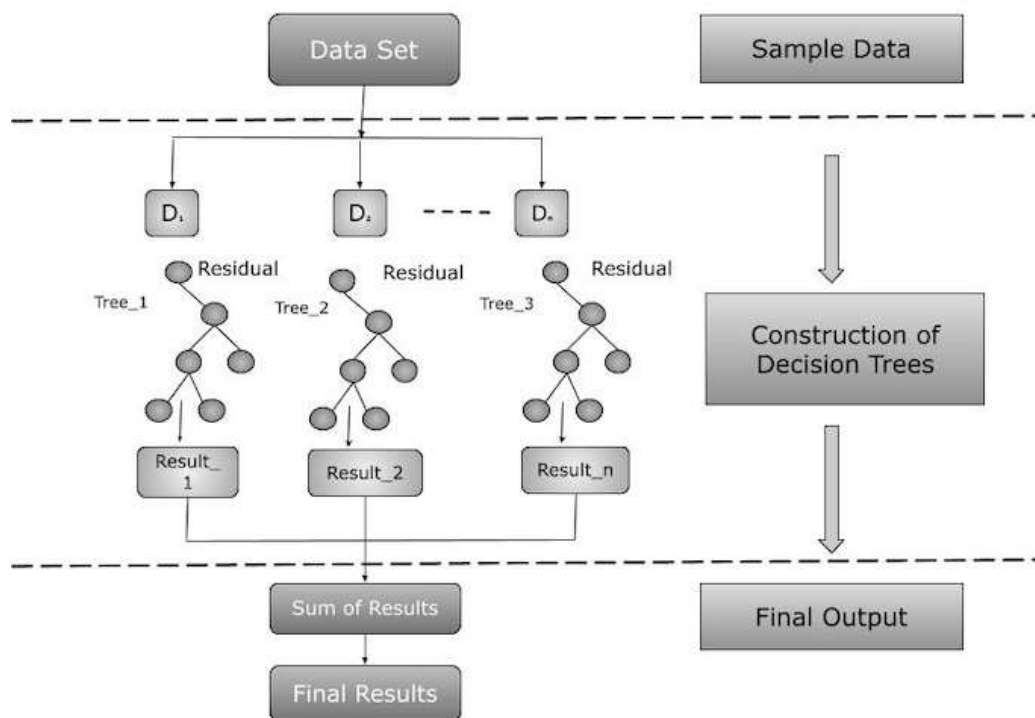
Analisis sentimen dalam bahasa Indonesia memiliki karakteristik dan tantangan yang berbeda dibandingkan dengan bahasa Inggris. Lin dan Nuha (2023) mengidentifikasi beberapa tantangan utama dalam analisis sentimen bahasa Indonesia, meliputi kompleksitas morfologi, penggunaan bahasa campuran (*code-mixing*), dan ekspresi sentimen yang implisit. Bahasa Indonesia memiliki sistem afiksasi yang kompleks, di mana satu kata dasar dapat membentuk berbagai kata turunan dengan makna yang berbeda, sehingga memerlukan proses *stemming* yang lebih canggih.

(Romadhony et al., 2024) melakukan analisis sentimen pada dataset ulasan produk Indonesia berukuran besar yang terdiri dari lebih dari 700.000 ulasan menggunakan Multinomial Naïve Bayes, Support Vector Machine, LSTM, dan BiLSTM. Penelitian tersebut menunjukkan bahwa pemilihan algoritma dan teknik *preprocessing* yang tepat sangat berpengaruh terhadap akurasi klasifikasi sentimen dalam bahasa Indonesia. (Ahmadian et al., 2024) mengembangkan model *hybrid* IndoBERT-BiLSTM yang disesuaikan khusus untuk bahasa Indonesia, mencapai akurasi 93% untuk analisis sentimen dan 78% untuk

klasifikasi emosi pada *benchmark* Indonesian Natural Language Understanding (IndoNLU).

Tantangan lain dalam analisis sentimen bahasa Indonesia di media sosial adalah penggunaan bahasa informal, singkatan, akronim, dan emoji yang sangat beragam. Pengguna media sosial Indonesia cenderung menggunakan bahasa gaul, singkatan tidak baku, dan campuran bahasa Indonesia-Inggris dalam satu kalimat. Hal ini menambah kompleksitas dalam tahap *preprocessing* dan memerlukan *dictionary* khusus untuk normalisasi teks.

2.3 Algoritma XGBoost



Gambar 2.1 Arsitektur Algoritma XGBoost

Diagram tersebut mengilustrasikan mekanisme algoritma XGBoost untuk pembangunan model prediktif secara bertahap menggunakan beberapa pohon

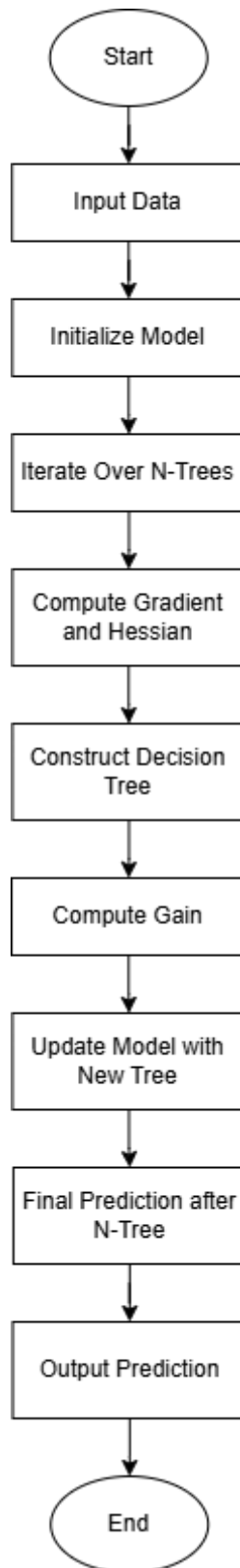
keputusan (*gradient boosting*). Proses dimulai dengan dataset yang berfungsi sebagai data pelatihan untuk membangun pohon keputusan pertama.

Pohon keputusan pertama memberikan prediksi awal, Residual atau kesalahan prediksi, kemudian dihitung. Residual ini berfungsi sebagai dasar untuk membangun pohon keputusan selanjutnya, sehingga setiap pohon baru mengoreksi kesalahan model sebelumnya. Proses ini diulang hingga beberapa pohon keputusan telah dibuat.

Setiap pohon memberikan *output* parsial. Jumlah semua *output* menghasilkan prediksi akhir. Dengan mekanisme ini, XGBoost secara bertahap meningkatkan akurasi model.

Penelitian pada ulasan aplikasi DANA menggunakan XGBoost, SVM, dan Extra Trees dengan 50.000 data ulasan berbahasa Indonesia dan teknik Word2Vec serta SMOTE untuk menangani ketidakseimbangan kelas. XGBoost unggul dengan akurasi 86,81% pada pelabelan berbasis skor (Setiawan & Nastiti, 2024).

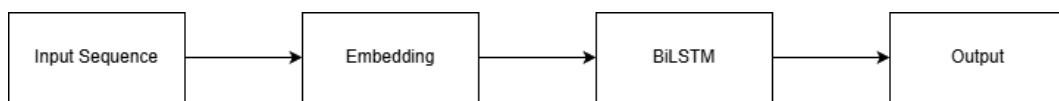
Tantangan lain dalam analisis sentimen bahasa Indonesia di media sosial adalah penggunaan bahasa informal, singkatan, akronim, dan emoji yang sangat beragam. Pengguna media sosial Indonesia cenderung menggunakan bahasa gaul, singkatan tidak baku, dan campuran bahasa Indonesia-Inggris dalam satu kalimat. Hal ini menambah kompleksitas dalam tahap *preprocessing* dan memerlukan *dictionary* khusus untuk normalisasi teks.



Gambar 2.2 Flowchart Algoritma XGBoost

XGBoost bekerja dengan menerima input data, lalu menginisialisasi model awal. Selanjutnya, algoritma membangun pohon keputusan secara berulang sebanyak N kali, di mana setiap iterasi menghitung gradient dan hessian untuk mengukur error, membangun pohon baru, menghitung gain terbaik, lalu memperbarui model. Setelah N pohon terbentuk, semua prediksi digabungkan untuk menghasilkan output prediksi akhir.

2.4 Algoritma BiLSTM



Gambar 2.3 Arsitekur Algoritma BiLSTM

1. Input Sequence

Input Sequence adalah serangkaian titik data, seperti kata-kata dalam kalimat atau karakter dalam teks. Setiap titik data biasanya direpresentasikan sebagai vektor atau representasi *embedding*.

2. Embedding

Urutan masukan sering diubah menjadi representasi vektor padat yang disebut *Embedding*. *Embedding* menangkap makna semantik dari titik data dan memberikan representasi yang lebih ringkas dan bermakna untuk lapisan selanjutnya.

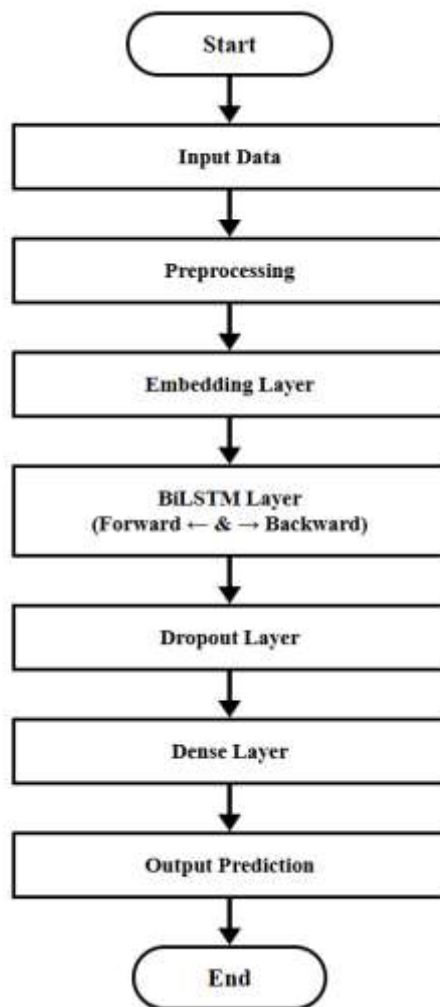
3. Bi-LSTM

Lapisan Bi-LSTM adalah komponen inti dari arsitektur. Lapisan ini terdiri dari dua lapisan LSTM: satu memproses urutan masukan ke depan, yang lain ke belakang. Setiap lapisan LSTM memiliki parameter yang berbeda.

4. Output

Output dari lapisan Bi-LSTM adalah kombinasi dari keadaan tersembunyi dari lapisan LSTM yang diproses ke depan dan ke belakang pada setiap langkah waktu. Metode kombinasi spesifik dapat bervariasi, misalnya, dengan menggabungkan keadaan tersembunyi atau menerapkan transformasi yang berbeda.

(Rifqi Yafik & Mulkan Azhari, 2025) dari Fakultas Ilmu Komputer dan Teknologi Informasi UMSU membuktikan bahwa arsitektur BiLSTM mampu memproses data secara bidireksional sehingga dapat menangkap pola kontekstual dan titik pembalikan tren dengan lebih baik. Dalam penelitian tersebut, BiLSTM mengungguli LSTM pada data deret waktu yang bersifat dinamis dan non-linear, yang secara prinsip juga berlaku pada data teks sekuensial dalam analisis sentiment.



Gambar 2.4 Flowchart Algoritma BiLSTM

Model BiLSTM bekerja dengan menerima input data yang kemudian diproses melalui tahap preprocessing. Data lalu diubah menjadi representasi vektor melalui Embedding Layer, kemudian diproses oleh BiLSTM Layer yang membaca urutan data dari dua arah (maju dan mundur) secara bersamaan. Dropout Layer diterapkan untuk mencegah overfitting, lalu Dense Layer mengolah hasilnya untuk menghasilkan Output Prediction sebagai prediksi akhir.

Lapisan BLSTM pertama memproses sekuens data dalam dua arah, yaitu maju dan mundur, sehingga mampu menangkap konteks informasi secara lebih lengkap dari kedua arah waktu (Alhassan & Altmami, 2025).

Arsitektur LSTM dirancang untuk mengatasi masalah *vanishing gradient* yang sering terjadi pada *Recurrent Neural Network* (RNN) tradisional. LSTM menggunakan struktur sel memori yang terdiri dari tiga gerbang utama: *input gate*, *forget gate*, dan *output gate*. Perhitungan pada sel LSTM dapat direpresentasikan dengan persamaan berikut:

Maka perhitungannya:

1. Forget gate

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

2. Input gate dan kandidat cell state

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \quad (3)$$

3. Update cell state

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (4)$$

4. Output gate dan hidden state

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \odot \tanh(C_t) \quad (6)$$

Keterangan:

- f_t, i_t, o_t masing-masing adalah nilai forget gate, input gate, dan output gate pada waktu t .
- \tilde{C}_t adalah kandidat cell state.
- \odot menyatakan perkalian elemen-per-elemen (element-wise).
- $x_t =$ input pada waktu ke- t
- $h_{t-1} =$ hidden state pada waktu sebelumnya
- $C_{t-1} =$ cell state pada waktu sebelumnya
- W dan $b =$ bobot dan bias
- $\sigma =$ fungsi sigmoid, $\tanh =$ fungsi hiperbolik tangen

Keunggulan BiLSTM dalam analisis sentimen terletak pada kemampuannya memahami konteks sekuensial dan menangkap nuansa makna yang bergantung pada urutan kata. Dalam bahasa natural, makna sebuah kata atau frasa sering dipengaruhi oleh konteks sebelum dan sesudahnya. BiLSTM dapat menangkap informasi kontekstual ini dengan lebih baik dibandingkan model yang hanya memproses teks dalam satu arah.

2.5 Preprocessing Teks Untuk Analisis Sentimen

Preprocessing merupakan tahapan krusial dalam analisis sentimen yang bertujuan untuk membersihkan dan mentransformasi data teks mentah menjadi format yang dapat diproses oleh algoritma *machine learning* (Lin & Nuha, 2023). Tahapan *preprocessing* yang umum dilakukan meliputi:

1. **Cleaning:** Menghapus karakter yang tidak relevan seperti URL, *mention*, *hashtag*, emotikon, dan karakter khusus yang tidak berkontribusi terhadap analisis sentimen.

2. **Case folding:** Mengubah semua huruf menjadi huruf kecil untuk menghindari duplikasi fitur yang disebabkan oleh perbedaan kapitalisasi.
3. **Tokenization:** Memecah teks menjadi unit-unit yang lebih kecil seperti kata atau token.
4. **Stopword removal:** Menghapus kata-kata yang sering muncul namun tidak memiliki makna signifikan dalam konteks analisis sentimen, seperti "yang", "di", "ke", dan sebagainya.
5. **Stemming:** Mengubah kata-kata berimbuhan menjadi kata dasar untuk mengurangi variasi morfologi dalam bahasa Indonesia.
6. **Vectorization:** Mengubah teks menjadi representasi numerik yang dapat diproses oleh algoritma *machine learning*, menggunakan teknik seperti *Bag of Words*, TF-IDF, atau *word embeddings*.

(Ahmadian et al., 2024) menekankan pentingnya *preprocessing* yang disesuaikan dengan karakteristik bahasa Indonesia, terutama dalam menangani afiksasi yang kompleks dan penggunaan bahasa campuran. Penelitian tersebut menggunakan kombinasi teknik *preprocessing* tradisional dengan model bahasa yang telah dilatih sebelumnya (*pre-trained language model*) untuk meningkatkan representasi teks dalam bahasa Indonesia.

2.6 Metrik Evaluasi Klasifikasi

Evaluasi performa model klasifikasi memerlukan metrik yang dapat mengukur berbagai aspek kinerja model. Metrik evaluasi yang umum digunakan dalam klasifikasi sentimen meliputi:

1. **Accuracy:** Proporsi prediksi yang benar terhadap total prediksi, dihitung dengan persamaan: $Accuracy = (TP + TN) / (TP + TN + FP + FN)$ (7)

2. **Precision**: Proporsi prediksi positif yang benar terhadap total prediksi positif, dihitung dengan: $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$ (8)

3. **Recall**: Proporsi data positif yang berhasil diprediksi dengan benar terhadap total data positif, dihitung dengan: $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$ (9)

4. **F1-Score**: *Harmonic mean* dari *precision* dan *recall*, dihitung dengan: $\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$ (10)

di mana TP (*True Positive*) adalah jumlah data positif yang diprediksi benar, TN (*True Negative*) adalah jumlah data negatif yang diprediksi benar, FP (*False Positive*) adalah jumlah data negatif yang diprediksi sebagai positif, dan FN (*False Negative*) adalah jumlah data positif yang diprediksi sebagai negatif.

(Talaat, 2023) menekankan bahwa pemilihan metrik evaluasi harus disesuaikan dengan karakteristik dataset dan tujuan aplikasi. Untuk dataset yang tidak seimbang (*imbalanced*), *F1-score* lebih informatif dibandingkan akurasi karena mempertimbangkan baik *precision* maupun *recall*. Selain metrik di atas, *confusion matrix* sering digunakan untuk memberikan gambaran visual mengenai distribusi prediksi yang benar dan salah untuk setiap kelas.

2.7 Penelitian Terdahulu

Beberapa penelitian terdahulu telah melakukan perbandingan berbagai algoritma untuk analisis sentimen. (Talaat, 2023) melakukan studi komparatif menggunakan model *hybrid* BERT dan menemukan bahwa tidak ada satu algoritma yang secara universal unggul untuk semua jenis dataset dan domain. Hasil klasifikasi sangat bergantung pada karakteristik data, teknik *preprocessing*, dan *tuning hyperparameter* yang diterapkan.

Dalam konteks XGBoost, Chandrasekaran et al. (2024) mengembangkan model *Hybrid LXGB* yang menggabungkan LSTM dan XGBoost untuk analisis sentimen multimodal, mencapai akurasi 97,18% pada dataset CMU-MOSEI, melampaui pengklasifikasi alternatif termasuk LSTM, CNN, DNN, dan XGBoost secara individual. Hal ini menunjukkan bahwa kombinasi kekuatan *ensemble learning* dan *deep learning* dapat meningkatkan performa klasifikasi sentimen secara signifikan.

Dalam konteks politik (Yulistiani & Styawati, 2024) menerapkan XGBoost untuk analisis sentimen terhadap kandidat Presiden Indonesia 2024 dari data Twitter, menghasilkan akurasi 96%, precision 96%, dan recall 97%. Penelitian ini membuktikan keunggulan XGBoost dalam konteks sentimen politik berbahasa Indonesia.

Studi komparatif pada enam arsitektur deep learning menunjukkan BiLSTM unggul dalam analisis sentimen media sosial dengan akurasi terbaik mencapai 88,88%, berkat kemampuan pemrosesan teks bidireksional yang dapat memahami konteks dari kedua arah kalimat (Nguyen, 2024).

Penelitian-penelitian di atas menunjukkan bahwa baik XGBoost maupun BiLSTM memiliki potensi yang baik untuk analisis sentimen, namun belum ada penelitian yang secara komprehensif membandingkan kedua algoritma ini dalam konteks spesifik sentimen publik terhadap kebijakan ekonomi pemerintah Indonesia, khususnya penempatan dana pemerintah di bank. Oleh karena itu, penelitian ini diharapkan dapat mengisi kesenjangan tersebut dengan memberikan evaluasi empiris yang sistematis.

Tabel 2.1 Ringkasan Penelitian Terdahulu

NO	Judul dan peneliti	Pembahasan	Metode	Kelebihan atau Kekurangan
1	Exploring sentiment text analysis on social media texts (Alabdul karim et al. 2024)	Penelitian ini menggunakan teknik berbasis model <i>skipgram</i> yang merepresentasikan kata-kata sebagai kantong karakter <i>n-gram</i> . Setiap kata adalah penjumlahan dari representasi vektor <i>n-gram</i> karakter. Hasil penelitian dinilai kesamaan kata dan analogi dalam sembilan bahasa.	FastText	<p>Kelebihan: Metode FastText efektif untuk analisis sentimen media sosial dengan akurasi 96%. Mampu menangani kata-kata di luar kosakata (<i>out-of-vocabulary</i>) dengan baik.</p> <p>Kekurangan: Tidak membandingkan dengan metode <i>ensemble learning</i> seperti XGBoost. Dataset terbatas pada ulasan produk, belum mencakup domain kebijakan publik.</p>
2	Multimodal sentiment	Penelitian ini mengembangkan model <i>Hybrid</i> LXGB	<i>Hybrid</i> LSTM-XGBoost	<p>Kelebihan: Akurasi sangat tinggi (97,18%), melampaui model</p>

	<p>t analysis leveraging the strength of deep neural networks enhanced by the XGBoost classifier (Chandra sekaran et al. 2024)</p>	<p>yang menggabungkan LSTM dan XGBoost untuk analisis sentimen multimodal, mengintegrasikan data teks, audio, dan visual dari dataset CMU-MOSEI.</p>		<p>individual seperti LSTM, CNN, DNN. Efektif untuk data multimodal.</p> <p>Kekurangan: Kompleksitas model tinggi, memerlukan komputasi intensif. Tidak menguji pada dataset bahasa Indonesia atau domain kebijakan publik</p>
3	<p>Sentimen t analysis of political post classifica tion based on XGBoost</p>	<p>Penelitian ini menggunakan XGBoost untuk klasifikasi sentimen postingan politik, dengan fokus pada analisis opini publik terhadap isu-isu politik. Dataset terdiri</p>	XGBoost	<p>Kelebihan: Akurasi tinggi (95,16%) untuk domain politik. XGBoost efisien dalam waktu <i>training</i> dan mampu menangani data tidak seimbang dengan baik.</p> <p>Kekurangan: Tidak</p>

	(Nsaif and Abd 2022)	dari postingan media sosial berbahasa Inggris.		membandingkan dengan metode <i>deep learning</i> . Belum diuji pada bahasa Indonesia atau domain kebijakan ekonomi.
4	RoBERTa-BiLSTM : A context-aware hybrid model for sentiment analysis (Rahman et al. 2024)	Penelitian ini memperkenalkan model <i>hybrid</i> RoBERTa-BiLSTM yang menggabungkan <i>transformer-based model</i> dengan arsitektur <i>sequential</i> . Model diuji pada tiga dataset: IMDb, Twitter US Airline, dan Sentiment140.	RoBERTa-BiLSTM	Kelebihan: Performa konsisten pada berbagai dataset (akurasi 92,36% pada IMDb). Mampu menangkap konteks semantik dengan baik. Kekurangan: Waktu komputasi dan kebutuhan memori tinggi. Belum diuji pada bahasa Indonesia atau domain kebijakan publik.
5	Sentiment analysis on a large Indonesian	Penelitian ini melakukan analisis sentimen pada dataset besar ulasan produk Indonesia (700.000+	Naïve Bayes, SVM, LSTM, BiLSTM	Kelebihan: Dataset besar berbahasa Indonesia, relevan untuk konteks lokal. Perbandingan empat metode berbeda.

	product review dataset (Romadhony et al. 2024)	ulasan) menggunakan empat algoritma: Multinomial Naïve Bayes, Support Vector Machine, LSTM, dan BiLSTM.		Kekurangan: Tidak menggunakan XGBoost sebagai pembandingan. Fokus pada domain ulasan produk, bukan kebijakan publik. Tidak menganalisis efisiensi komputasi secara mendalam.
6	Hybrid models for emotion classification and sentiment analysis in Indonesian language (Ahmadian et al. 2024)	Penelitian ini mengembangkan model <i>hybrid</i> IndoBERT-BiLSTM khusus untuk bahasa Indonesia, diuji pada <i>benchmark</i> IndoNLU untuk klasifikasi sentimen dan emosi.	IndoBERT-BiLSTM	Kelebihan: Model disesuaikan dengan karakteristik bahasa Indonesia (akurasi 93%). Mampu menangani <i>code-mixing</i> dan bahasa informal. Kekurangan: Tidak membandingkan dengan <i>ensemble learning</i> seperti XGBoost. Belum diuji pada domain kebijakan ekonomi pemerintah.

7	Mining social media data to inform public health policies: A sentiment analysis case study (Russell and Rao-Graham 2024)	Penelitian ini menggunakan analisis sentimen <i>machine learning</i> pada tweet terkait kebijakan <i>lockdown</i> COVID-19 di Jamaica, mengidentifikasi 76% sentimen negatif dari 1.609 tweet.	Machine Learning (metode tidak dispesifikkan secara detail)	<p>Kelebihan: Aplikasi langsung untuk kebijakan publik. Menunjukkan relevansi analisis sentimen untuk pengambilan keputusan pemerintah.</p> <p>Kekurangan: Metode tidak dijelaskan secara detail. Tidak ada perbandingan algoritma. Dataset relatif kecil (1.609 tweet).</p>
8	Sentiment analysis classification system using hybrid	Penelitian studi komparatif menggunakan berbagai varian model <i>hybrid</i> BERT untuk analisis sentimen pada	Hybrid BERT variants	<p>Kelebihan: Studi komparatif komprehensif. Menunjukkan pentingnya pemilihan metode sesuai karakteristik data.</p> <p>Kekurangan: Fokus</p>

	BERT models (Talaat 2023)	beberapa dataset standar. Menekankan tidak ada algoritma universal terbaik.		hanya pada model berbasis BERT. Tidak membandingkan dengan <i>ensemble learning</i> tradisional seperti XGBoost atau BiLSTM standalone.
--	---------------------------	---	--	---

2.8 Analisis GAP

Berdasarkan tinjauan pustaka yang telah dilakukan pada sub bab sebelumnya, dapat diidentifikasi beberapa kesenjangan (*gap*) penelitian yang menjadi dasar dilakukannya penelitian ini. Analisis *gap* dilakukan dengan mengkaji penelitian-penelitian terdahulu yang relevan dan mengidentifikasi area yang belum dieksplorasi secara mendalam.

2.8.1 Gap Konteks Penelitian

Sebagian besar penelitian terdahulu mengenai perbandingan algoritma klasifikasi sentimen dilakukan pada domain umum seperti ulasan produk, ulasan film, atau sentimen media sosial secara umum ((Alabdulkarim et al., 2024), (Rahman et al., 2025), (Zhao et al., 2024)). Penelitian yang secara spesifik menganalisis sentimen publik terhadap kebijakan ekonomi pemerintah, khususnya kebijakan penempatan dana pemerintah di perbankan, masih sangat terbatas. (Russell et al., 2024) melakukan analisis sentimen terhadap kebijakan *lockdown* COVID-19, namun fokusnya pada kebijakan kesehatan publik, bukan kebijakan ekonomi dan keuangan.

Dalam konteks Indonesia, (Khadija et al., 2024) melakukan analisis sentimen terhadap keluhan aspirasi publik menggunakan *machine learning*, namun penelitian tersebut tidak secara spesifik membahas kebijakan penempatan dana pemerintah. Hal ini menunjukkan adanya kesenjangan dalam literatur mengenai aplikasi analisis sentimen pada domain kebijakan keuangan publik Indonesia. Penelitian ini berupaya mengisi kesenjangan tersebut dengan fokus pada sentimen publik terhadap kebijakan penempatan dana pemerintah di bank, yang merupakan isu strategis dan sensitif dalam pengelolaan keuangan negara.

2.8.2 Gap Perbandingan Algoritma

Meskipun terdapat banyak penelitian yang mengevaluasi performa XGBoost dan BiLSTM secara terpisah untuk analisis sentimen, penelitian yang secara sistematis membandingkan kedua algoritma ini dengan metodologi yang konsisten masih terbatas. (Chandrasekaran et al., 2025) mengembangkan model *hybrid* yang menggabungkan LSTM dan XGBoost, namun tidak melakukan perbandingan langsung antara XGBoost *standalone* dan BiLSTM *standalone* dengan kondisi eksperimen yang sama.

(Talaat, 2023) menekankan bahwa pemilihan algoritma yang tepat sangat bergantung pada karakteristik data dan konteks aplikasi, namun penelitian tersebut fokus pada model *hybrid* BERT dan tidak membandingkan XGBoost dengan BiLSTM. Penelitian ini mengisi kesenjangan tersebut dengan melakukan evaluasi komparatif yang sistematis antara XGBoost dan BiLSTM menggunakan dataset yang sama, metrik evaluasi yang konsisten, dan kondisi

eksperimen yang terkontrol. Hal ini memungkinkan identifikasi algoritma yang lebih efektif untuk konteks spesifik analisis sentimen kebijakan publik.

2.8.3 Gap Metrik Evaluasi Komprehensif

Sebagian besar penelitian terdahulu fokus pada metrik akurasi sebagai indikator utama performa model ((Alabdulkarim et al., 2024)(Nsaif & Abd, 2022)). Meskipun akurasi merupakan metrik penting, evaluasi yang komprehensif memerlukan pertimbangan metrik tambahan seperti *precision*, *recall*, *F1-score*, serta aspek efisiensi komputasi yang meliputi waktu pelatihan dan waktu inferensi.

(Rahman et al., 2025) melakukan evaluasi pada beberapa dataset, namun tidak menganalisis secara mendalam aspek efisiensi komputasi dan *trade-off* antara akurasi dengan kompleksitas model. (Zhao et al., 2024) melaporkan *F1-score* yang tinggi, namun tidak memberikan analisis komparatif dengan algoritma alternatif dalam hal efisiensi waktu. Penelitian ini mengisi kesenjangan tersebut dengan melakukan evaluasi komprehensif yang tidak hanya mempertimbangkan metrik akurasi, *precision*, *recall*, dan *F1-score*, tetapi juga menganalisis efisiensi waktu komputasi dan kemampuan generalisasi model pada data *testing* yang belum pernah dilihat sebelumnya.

2.8.4 Gap Preprocessing untuk Bahasa Indonesia

(Lin & Nuha, 2023) mengidentifikasi tantangan khusus dalam analisis sentimen bahasa Indonesia, namun penelitian tersebut fokus pada pendekatan *hybrid deep learning* dengan BERT. (Romadhony et al., 2024) melakukan analisis pada dataset besar bahasa Indonesia, namun tidak mengeksplorasi

secara mendalam dampak teknik *preprocessing* yang berbeda terhadap performa XGBoost dan BiLSTM.

(Ahmadian et al., 2024) mengembangkan model *hybrid* IndoBERT-BiLSTM untuk bahasa Indonesia, namun tidak membandingkan dengan pendekatan *ensemble learning* seperti XGBoost. Penelitian ini mengisi kesenjangan tersebut dengan mengeksplorasi teknik *preprocessing* yang optimal untuk bahasa Indonesia dalam konteks perbandingan XGBoost dan BiLSTM, termasuk penanganan bahasa campuran (*code-mixing*), normalisasi teks informal, dan *stemming* yang disesuaikan dengan morfologi bahasa Indonesia.

2.8.5 Gap Implikasi Praktis untuk Kebijakan Publik

(Russell et al., 2024) mendemonstrasikan aplikasi analisis sentimen untuk kebijakan publik, namun tidak memberikan panduan praktis untuk pemilihan algoritma berdasarkan kebutuhan spesifik seperti kecepatan analisis *real-time* versus akurasi maksimal. Penelitian ini mengisi kesenjangan tersebut dengan tidak hanya membandingkan performa kedua algoritma, tetapi juga memberikan rekomendasi praktis berbasis bukti empiris mengenai algoritma yang tepat untuk implementasi sistem monitoring sentimen publik terhadap kebijakan ekonomi pemerintah, dengan mempertimbangkan *trade-off* antara akurasi, efisiensi komputasi, dan kemudahan implementasi.

Tabel 2.2 merangkum kesenjangan penelitian yang telah diidentifikasi dan bagaimana penelitian ini berupaya mengisi kesenjangan tersebut.

Tabel 2.2 Analisis kesenjangan penelitian dan kontribusi penelitian ini terhadap pengisian kesenjangan yang ada

No	Aspek Kesenjangan	Kondisi Penelitian Terdahulu	Kontribusi Penelitian Ini
1	Konteks Penelitian	Fokus pada domain umum (ulasan produk, film) dan kebijakan kesehatan, belum spesifik pada kebijakan keuangan publik Indonesia	Analisis sentimen spesifik pada kebijakan penempatan dana pemerintah di bank dengan data berbahasa Indonesia
2	Perbandingan Algoritma	Evaluasi terpisah atau model <i>hybrid</i> , belum ada perbandingan sistematis XGBoost vs BiLSTM	Perbandingan sistematis XGBoost dan BiLSTM dengan metodologi konsisten dan kondisi eksperimen terkontrol
3	Metrik Evaluasi	Fokus pada akurasi, kurang mempertimbangkan efisiensi komputasi dan <i>trade-off</i>	Evaluasi komprehensif meliputi akurasi, <i>precision</i> , <i>recall</i> , <i>F1-score</i> , waktu komputasi, dan generalisasi
4	Preprocessing Bahasa Indonesia	Belum mengeksplorasi dampak teknik <i>preprocessing</i> terhadap perbandingan XGBoost dan BiLSTM	Eksplorasi teknik <i>preprocessing</i> optimal untuk bahasa Indonesia dalam konteks kedua algoritma
5	Implikasi Praktis	Kurang memberikan rekomendasi praktis untuk	Rekomendasi berbasis buktiempiris untuk

		implementasi kebijakan publik	implementasi sistem monitoring sentimen dengan pertimbangan <i>trade-off</i>
--	--	-------------------------------	--

BAB III

METODOLOGI PENELITIAN

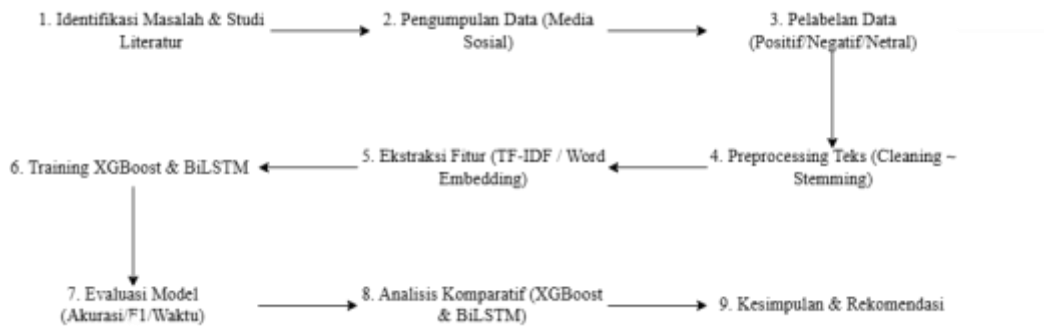
3.1 Alur Penelitian

Penelitian ini menggunakan pendekatan eksperimental komparatif (comparative experimental) dengan tujuan membandingkan performa dua algoritma klasifikasi sentimen, yaitu XGBoost dan BiLSTM. Pendekatan ini dipilih karena sesuai dengan tujuan penelitian untuk mengidentifikasi algoritma yang lebih efektif dalam konteks spesifik analisis sentimen kebijakan penempatan dana pemerintah di bank (Talaat, 2023). Desain penelitian mencakup tahapan pengumpulan data, preprocessing, implementasi model, evaluasi, dan analisis komparatif.

Penelitian ini bersifat kuantitatif dengan menggunakan metrik evaluasi terukur seperti akurasi, precision, recall, F1-score, dan waktu komputasi. Setiap algoritma akan diimplementasikan dengan kondisi eksperimen yang sama untuk memastikan validitas perbandingan (Rahman et al., 2025).

3.2 Alur Kerja Penelitian

Alur kerja penelitian ini terdiri dari beberapa tahapan utama yang dilakukan secara berurutan, sebagaimana ditunjukkan pada Gambar 3.1. Setiap tahapan memiliki tujuan spesifik dan output yang menjadi input untuk tahapan berikutnya.



Gambar 3.1 Alur kerja penelitian yang menunjukkan alur dari pengumpulan data hingga analisis hasil dan rekomendasi

Tahapan alur kerja penelitian secara rinci adalah sebagai berikut:

1. Identifikasi Masalah & Studi Literatur: Mengidentifikasi permasalahan dan mengkaji literatur terkait kebijakan penempatan dana pemerintah di bank serta algoritma XGBoost dan BiLSTM.
2. Pengumpulan Data: Mengumpulkan data sentimen publik dari media sosial dan/atau platform berita online yang relevan.
3. Pelabelan Data: Memberikan label sentimen (positif, negatif, netral) pada data yang terkumpul.
4. Preprocessing Teks: Membersihkan dan mentransformasi data teks sebelum digunakan pada tahap berikutnya.
5. Ekstraksi Fitur: Mengekstraksi fitur teks untuk XGBoost dan menyusun word embeddings untuk BiLSTM.
6. Training XGBoost & BiLSTM: Membangun dan melatih kedua model menggunakan data latihan yang telah direpresentasikan.
7. Evaluasi Model: Mengevaluasi performa model dengan metrik seperti akurasi, precision, recall, F1 score, dan waktu komputasi.

8. Analisis Komparatif: Membandingkan hasil performa XGBoost dan BiLSTM untuk menentukan algoritma yang lebih efektif.
9. Penarikan Kesimpulan & Rekomendasi: Menyimpulkan algoritma terbaik dan menyusun rekomendasi implementasi sistem monitoring sentimen publik.

3.3 Pengumpulan Data

3.3.1 Sumber Data

Data sentimen publik dikumpulkan dari dua sumber utama:

1. Media Sosial Youtube: Data berupa komen yang membahas kebijakan penempatan dana pemerintah di bank, dengan kata kunci yang relevan seperti "dana pemerintah", "deposito pemerintah", "bank BUMN", dan topik terkait lainnya.
2. Platform Berita Online: Data berupa komentar pembaca pada artikel berita yang membahas kebijakan penempatan dana pemerintah di bank dari portal berita Indonesia yang representatif.

Tabel 3.1 Data Ulasan

No	UserName	Review	Rating	Sentimen
1	bioroids	Blokir judul, niscaya duit akan berputar lagi	2	Negatif
2	elyhardi	ini maksudnya biar UMKM pinjam ke bank gak sih ,biar ekonomi kecil jalan	2	Negatif
3	Aku.di. seneni	Gw suka bgt cara ngomongnya. Kek anak tongkrongan	2	Negatif
4	Naufal_ fayeza11	pak Prabowo tolong kasih pengawalan. buat Bpk purbaya..	1	Positif
5	banana	Kaum milenial tapi jiwa Gen Z	0	Netral

	caramel6			
.....
3870	ikbena5	gak bikin pertumbuhan di ekonomi	2	Negatif
3871	Pyrex mizuno027	lebih masuk malahan di otak kalau gak bertele2.	1	Positif
3872	gaes1706	proyek yg dri pemerintah juga bisa berjalan, contoh: perumahan subsidi dll	1	Positif
3873	Syamchan 01	bunga acuan bukannya dari BI? bagaimana bank-bank konvensional nunurinin bunganya?	0	Netral
3874	clan..d	iya klo gk di korup	1	Positif

Periode pengumpulan data mencakup rentang waktu 1 bulan terakhir untuk memastikan relevansi dan representasi sentimen terkini.

3.3.2 Kriteria Data

Data yang dikumpulkan harus memenuhi kriteria berikut:

1. Berbahasa Indonesia atau campuran Indonesia-Inggris
2. Membahas topik kebijakan penempatan dana pemerintah di bank
3. Mengandung ekspresi sentimen yang jelas atau tersirat
4. Tidak duplikat
5. Minimal terdiri dari 5 kata

Target jumlah data yang dikumpulkan adalah minimal 5.000 sampel untuk memastikan data yang cukup untuk pelatihan dan pengujian model (Romadhony et al., 2024). Data akan dibagi dengan proporsi 70% untuk training, 15% untuk validation, dan 15% untuk testing.

3.3.3 Pelabelan Data

Pelabelan data dilakukan secara manual oleh tiga annotator independen untuk memastikan reliabilitas label. Setiap teks diklasifikasikan ke dalam tiga kategori sentimen:

1. Positif : Teks mengekspresikan dukungan, persetujuan, atau pandangan positif terhadap kebijakan
2. Negatif : Teks mengekspresikan penolakan, kritik, atau pandangan negatif terhadap kebijakan
3. Netral : Teks tidak mengekspresikan sentimen yang jelas atau bersifat informatif saja

Untuk memastikan konsistensi pelabelan, dilakukan perhitungan inter-annotator agreement menggunakan Cohen's Kappa. Data dengan tingkat kesepakatan rendah akan didiskusikan bersama untuk mencapai konsensus (Khadija et al., 2024).

3.4 Preprocessing Teks

Preprocessing merupakan tahapan penting untuk mengubah data teks mentah menjadi format yang dapat diproses oleh algoritma machine learning (Lin and Nuha 2023). Tahapan preprocessing yang dilakukan adalah:

3.4.1 Cleaning

Tahap pembersihan data meliputi:

1. Menghapus URL, mention (@username), dan hashtag (#)
2. Menghapus angka dan karakter khusus yang tidak relevan
3. Menghapus emotikon dan emoji (atau mengkonversi ke teks jika relevan)

4. Menghapus spasi berlebih dan karakter yang berulang

3.4.2 Case Folding

Mengubah semua huruf menjadi huruf kecil (lowercase) untuk menghindari duplikasi fitur yang disebabkan oleh perbedaan kapitalisasi. Contoh: "Bank" dan "bank" akan dianggap sebagai kata yang sama.

3.4.3 Normalisasi

Mengubah kata-kata tidak baku, singkatan, dan slang menjadi bentuk baku menggunakan dictionary normalization. Contoh: "gak" menjadi "tidak", "pemerintah" tetap "pemerintah", "bnyk" menjadi "banyak". Dictionary akan dikembangkan berdasarkan kata-kata tidak baku yang umum digunakan dalam bahasa Indonesia informal.

3.4.4 Tokenization

Memecah teks menjadi token-token individual (kata). Proses ini menggunakan library pemrosesan bahasa Indonesia yang sesuai, dengan mempertimbangkan karakteristik khusus bahasa Indonesia seperti kata majemuk dan afiksasi.

3.4.5 Stopword Removal

Menghapus kata-kata yang sering muncul namun tidak berkontribusi signifikan terhadap sentimen, seperti "yang", "di", "ke", "dari", "untuk". Stopword list akan disesuaikan dengan konteks analisis sentimen, di mana beberapa kata yang biasanya dianggap stopwords mungkin memiliki nilai sentimen (seperti "tidak", "sangat").

3.4.6 Stemming

Mengubah kata berimbuhan menjadi kata dasar menggunakan algoritma stemming untuk bahasa Indonesia seperti Sastrawi. Contoh: "dipercaya" menjadi "percaya", "pengelolaan" menjadi "kelola". Proses ini penting untuk mengurangi dimensi fitur dan menangkap makna inti dari kata (Ahmadian et al., 2024).

Hasil preprocessing akan disimpan dalam format terstruktur yang siap untuk tahap ekstraksi fitur. Validasi hasil preprocessing dilakukan dengan memeriksa sampel acak untuk memastikan tidak ada informasi penting yang hilang atau terdistorsi.

3.5 Ekstraksi Fitur

3.5.1 Ekstraksi Fitur untuk XGBoost

Untuk algoritma XGBoost, teks yang telah di-preprocessing akan diubah menjadi representasi numerik menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF menghitung bobot setiap kata berdasarkan frekuensi kemunculan dalam dokumen dan keunikan kata tersebut dalam keseluruhan korpus (Nsaif and Abd 2022).

Perhitungan TF-IDF mengikuti persamaan:

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (11)$$

di mana:

$$TF(t,d) = (\text{Jumlah kemunculan term } t \text{ dalam dokumen } d) / (\text{Total term dalam dokumen } d)$$

$$IDF(t) = \log(\text{Total dokumen} / \text{Jumlah dokumen yang mengandung term } t)$$

Parameter yang digunakan:

- max_features: 5000 (jumlah fitur maksimal yang dipertahankan)
- ngram_range: (1,2) (unigram dan bigram)
- min_df: 2 (kata minimal muncul di 2 dokumen)

3.5.2 Ekstraksi Fitur untuk BiLSTM

Untuk algoritma BiLSTM, teks diubah menjadi sekuens numerik menggunakan word embeddings. Penelitian ini akan menggunakan pre-trained word embeddings untuk bahasa Indonesia (seperti fastText atau Word2Vec yang telah dilatih pada korpus bahasa Indonesia) untuk meningkatkan representasi semantik kata (Ahmadian et al., 2024)

Proses embedding meliputi:

1. Membuat vocabulary dari data training
2. Mengonversi setiap kata menjadi indeks numerik
3. Melakukan padding atau truncation untuk menyamakan panjang sekuens
4. Menggunakan embedding layer dengan dimensi 300 (sesuai dimensi pre-trained embeddings)

Parameter yang digunakan:

- max_sequence_length: 100 (panjang maksimal sekuens)
- embedding_dim: 300 (dimensi vektor embedding)
- vocabulary_size: 10000 (ukuran vocabulary maksimal)

3.5.3 Alur Pemrosesan Data

Bagian ini menjelaskan alur konseptual dan matematis transformasi data teks, dari data teks yang telah diproses hingga hasil klasifikasi sentimen.

Karena langkah-langkah pra-pemrosesan teknis dirinci di Bagian 3.4, bagian

ini berfokus pada representasi abstrak dan transformasi matematis data yang menjadi dasar pemrosesan menggunakan algoritma XGBoost dan BiLSTM.

A. Input Data Teks

Dataset penelitian didefinisikan sebagai sebuah korpus :

$$D = \{d_1, d_2, \dots, d_N\} \quad (12)$$

di mana setiap dokumen d_i merupakan satu komentar publik yang membahas kebijakan penempatan dana pemerintah di bank.

Setiap dokumen terdiri dari urutan kata:

$$d_i' = \{w_{i1}, w_{i2}, \dots, w_{iL}\} \quad (13)$$

Pada tahap ini, data masih bersifat tidak terstruktur (unstructured text) sehingga belum dapat langsung diproses oleh algoritma machine learning.

B. Cleaning dan Preprocessing Teks

Tahap preprocessing bertujuan untuk membersihkan dan menormalkan data teks agar mengurangi noise serta meningkatkan kualitas representasi fitur.

Proses preprocessing dapat dimodelkan sebagai sebuah fungsi transformasi:

$$d_i' = f_{prep}(d_i) \quad (14)$$

Transformasi ini mencakup:

- Penghapusan URL, mention, hashtag, angka, dan karakter khusus
- Case folding (perubahan huruf menjadi lowercase)
- Tokenisasi
- Stopword removal
- Stemming

Hasil dari tahap ini berupa sekumpulan token bersih:

$$d_i' = \{t_{i1}, t_{i2}, \dots, t_{iK}\} \quad (15)$$

dengan $K \leq L$, yang menunjukkan bahwa dimensi data telah direduksi tanpa menghilangkan makna utama.

3.6 Implementasi Model

3.6.1 Implementasi Model XGBoost

Model XGBoost diimplementasikan menggunakan library XGBoost dengan bahasa pemrograman Python. Arsitektur model XGBoost tidak memerlukan desain layer seperti deep learning, namun memerlukan konfigurasi hyperparameter yang optimal.

Hyperparameter yang akan di-tuning:

1. `learning_rate`: 0.1 (laju pembelajaran)
2. `max_depth`: 6 (kedalaman maksimal pohon)
3. `n_estimators`: 100 (jumlah pohon)
4. `subsample`: 0.8 (proporsi sampel untuk setiap pohon)
5. `colsample_bytree`: 0.8 (proporsi fitur untuk setiap pohon)
6. `min_child_weight`: 1 (bobot minimum untuk child node)
7. `gamma`: 0 (minimum loss reduction untuk pemisahan)
8. `objective`: 'multi:softmax' (untuk klasifikasi multi-kelas)

Proses training menggunakan early stopping dengan patience 10 iterasi untuk mencegah overfitting (Alabdulkarim et al., 2024). Model yang memberikan performa terbaik pada validation set akan disimpan dan digunakan untuk evaluasi pada testing set.

Dengan tahapan cara kerja model XGBoost sebagai berikut :

3.6.1.1 Cara Kerja Algoritma XGBoost dalam Klasifikasi Sentimen

XGBoost (eXtreme Gradient Boosting) bekerja berdasarkan prinsip ensemble learning dengan metode gradient boosting. Ide dasarnya adalah membangun banyak pohon keputusan (decision tree) secara bertahap, di mana setiap pohon baru bertugas memperbaiki kesalahan yang dibuat oleh pohon sebelumnya. Pada penelitian ini, XGBoost digunakan untuk mengklasifikasikan komentar TikTok ke dalam tiga kelas sentimen: Positif, Negatif, dan Netral.

a. Tahapan Cara Kerja XGBoost

- **Inisialisasi Model**

XGBoost dimulai dengan membuat prediksi awal yang sederhana. Untuk klasifikasi multi-kelas dengan tiga sentimen, model menginisialisasi prediksi awal menggunakan nilai rata-rata dari seluruh data training. Secara matematis, model awal didefinisikan sebagai:

$$F_0(x) = \operatorname{argmin}_{\gamma} \sum_i L(y_i, \gamma) \quad (16)$$

di mana $L(y_i, \gamma)$ adalah fungsi loss (dalam penelitian ini menggunakan softmax cross-entropy) dan y_i adalah label sentimen aktual dari data ke- i .

- **Komputasi Residual (Gradient dan Hessian)**

Pada setiap iterasi m (untuk setiap pohon ke- m yang dibangun), XGBoost menghitung pseudo-residual yang merupakan turunan pertama (gradient) dan turunan kedua (hessian) dari fungsi loss terhadap prediksi saat ini:

$$g_i = \partial L(y_i, F_{m-1}(x_i)) / \partial F_{m-1}(x_i) \quad (17)$$

$$h_i = \partial^2 L(y_i, F_{m-1}(x_i)) / \partial F_{m-1}(x_i)^2 \quad (18)$$

Gradient g_i menunjukkan arah dan besarnya kesalahan prediksi, sedangkan Hessian h_i digunakan untuk mengontrol kecepatan koreksi agar tidak terlalu agresif. Inilah yang membedakan XGBoost dari algoritma gradient boosting biasa — penggunaan informasi orde kedua membuat proses pembelajaran lebih stabil dan presisi.

- **Pembangunan Pohon Keputusan (Tree Building)**

Pada setiap iterasi, XGBoost membangun sebuah pohon keputusan baru dengan cara mencari pemisahan (split) terbaik pada setiap node. Pemisahan dinilai menggunakan Gain Score yang mengukur seberapa besar penurunan loss jika suatu fitur TF-IDF digunakan sebagai pemisah:

$$\text{Gain} = \frac{1}{2} \left[\frac{(\sum G_l)^2}{(\sum H_l + \lambda)} + \frac{(\sum G_r)^2}{(\sum H_r + \lambda)} - \frac{(\sum G)^2}{(\sum H + \lambda)} \right] - \gamma \quad (19)$$

di mana G_l , G_r adalah jumlah gradient pada node anak kiri dan kanan, H_l , H_r adalah jumlah hessian pada node anak kiri dan kanan, λ adalah parameter regularisasi L2, dan γ adalah ambang batas minimum gain yang diperlukan agar pemisahan dilakukan. Pemisahan hanya dilakukan jika $\text{Gain} > 0$.

Dalam konteks penelitian ini, fitur yang dievaluasi adalah nilai TF-IDF dari setiap kata/bigram dalam komentar. Misalnya, kata "blokir" dan "judol" yang muncul berdekatan akan membentuk bigram dengan nilai TF-IDF tinggi dan menjadi fitur pemisah yang signifikan untuk menentukan sentimen positif terhadap kebijakan pemerintah.

- **Komputasi Nilai Daun (Leaf Weight)**

Setelah struktur pohon terbentuk, nilai prediksi untuk setiap daun (leaf) dihitung dengan rumus:

$$w_j^* = - (\sum_{i \in R_j} g_i) / (\sum_{i \in R_j} h_i + \lambda) \quad (20)$$

di mana R_j adalah himpunan data yang jatuh pada daun ke- j . Nilai ini menunjukkan besarnya koreksi yang perlu ditambahkan ke prediksi sebelumnya untuk mengurangi kesalahan.

- **Pembaruan Model (Model Update)**

Setelah pohon ke- m selesai dibangun, prediksi model diperbarui dengan:

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x) \quad (21)$$

di mana η adalah learning rate (dalam penelitian ini $\eta = 0,1$) yang mengontrol seberapa besar kontribusi pohon baru terhadap prediksi akhir. Nilai η yang kecil membuat model belajar secara bertahap sehingga mengurangi risiko overfitting.

- **Iterasi dan Konvergensi**

Proses di atas diulang sebanyak $n_estimators = 100$ kali (100 pohon). Setiap pohon memperbaiki kesalahan dari pohon sebelumnya. Setelah 100 iterasi, model memiliki prediksi kumulatif:

$$F(x) = F_0(x) + \eta \cdot \sum_{m=1}^{100} h_m(x) \quad (22)$$

Nilai prediksi final untuk setiap kelas sentimen kemudian dikonversi menjadi probabilitas menggunakan fungsi softmax, dan kelas dengan probabilitas tertinggi dipilih sebagai label sentimen komentar.

b. Contoh Alur Klasifikasi XGBoost

Sebagai ilustrasi, berikut adalah contoh bagaimana XGBoost memproses komentar: "Blokir judol, niscaya duit akan berputar lagi":

1. Komentar direpresentasikan sebagai vektor TF-IDF berdimensi 5.000, misalnya: "blokir" = 0,42; "judol" = 0,38; "duit" = 0,29; "berputar" = 0,31 (kata-kata lain bernilai 0).

2. Vektor ini dimasukkan ke dalam 100 pohon keputusan secara bersamaan.

3. Setiap pohon menghasilkan skor untuk ketiga kelas: Positif = +0,82; Netral = -0,15; Negatif = -0,67.

4. Skor dari semua pohon dijumlahkan dan dikonversi ke probabilitas: Positif = 0,74; Netral = 0,16; Negatif = 0,10.

5. Karena probabilitas Positif tertinggi, komentar ini diklasifikasikan sebagai sentimen Positif.

c. Fungsi Objektif dan Regularisasi XGBoost

Keunggulan utama XGBoost dibandingkan gradient boosting biasa adalah penambahan regularisasi pada fungsi objektifnya untuk mencegah overfitting.

Fungsi objektif yang diminimalkan adalah:

$$\text{Obj} = \sum_i L(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \quad (23)$$

di mana suku pertama adalah training loss yang mengukur seberapa baik model cocok dengan data training, dan suku kedua adalah:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_j w_j^2 \quad (24)$$

Regulasi $\Omega(f)$ mengontrol kompleksitas pohon, di mana T adalah jumlah daun dan w_j adalah bobot daun. Parameter $\gamma = 0$ dan $\lambda = 1$ digunakan dalam penelitian ini, yang menjaga agar pohon tidak terlalu kompleks sehingga model tetap mampu melakukan generalisasi yang baik pada data komentar TikTok yang belum pernah dilihat sebelumnya.

3.6.2 Implementasi Model BiLSTM

Model BiLSTM diimplementasikan menggunakan framework TensorFlow/Keras dengan bahasa pemrograman Python. Arsitektur model BiLSTM yang digunakan terdiri dari beberapa layer sebagai berikut:

1. Embedding Layer: Mengubah sekuens indeks kata menjadi vektor embedding dengan dimensi 300
2. Bidirectional LSTM Layer 1: 128 unit dengan `return_sequences=True`
3. Dropout Layer 1: Dropout rate 0.3 untuk regularisasi
4. Bidirectional LSTM Layer 2: 64 unit
5. Dropout Layer 2: Dropout rate 0.3
6. Dense Layer: 3 unit dengan aktivasi softmax untuk output klasifikasi

Hyperparameter yang digunakan:

1. `batch_size`: 32 (ukuran batch untuk training)
2. `epochs`: 50 (jumlah epoch maksimal)
3. `optimizer`: Adam dengan `learning_rate` 0.001
4. `loss function`: Categorical Crossentropy
5. `early_stopping_patience`: 10 (berhenti jika tidak ada peningkatan selama 10 epoch)

Model akan di-compile dan di-train menggunakan data training, dengan monitoring performa pada validation set. Model terbaik akan disimpan menggunakan ModelCheckpoint callback (Rahman et al., 2025).

3.6.2.1 Cara Kerja Algoritma BiLSTM dalam Klasifikasi Sentimen

Bidirectional Long Short-Term Memory (BiLSTM) merupakan pengembangan dari LSTM yang memproses urutan teks dari dua arah secara

bersamaan — dari kiri ke kanan (forward) dan dari kanan ke kiri (backward). Kemampuan ini memungkinkan model untuk memahami konteks kata tidak hanya berdasarkan kata-kata sebelumnya, tetapi juga kata-kata yang muncul sesudahnya dalam kalimat, sehingga sangat efektif dalam analisis sentimen teks bahasa Indonesia yang informal.

a. Tahapan Cara Kerja BiLSTM

- **Representasi Teks sebagai Embedding**

Langkah pertama adalah mengubah teks komentar yang telah di-preprocessing menjadi urutan vektor numerik menggunakan embedding layer. Setiap kata dalam komentar dikonversi ke dalam indeks, lalu dipetakan ke vektor berdimensi 128 (`embedding_dim = 128`). Jika komentar lebih pendek dari 100 token, akan dilakukan zero-padding, sedangkan jika lebih panjang akan dipotong (truncation).

Representasi formal input adalah:

$$X = (x_1, x_2, \dots, x_T) \text{ dimana } x_t \in \mathbb{R}^{128}, T \leq 100 \quad (25)$$

- **Mekanisme LSTM Cell dan Gating**

LSTM mengatasi masalah vanishing gradient yang terjadi pada RNN biasa melalui mekanisme gerbang (gate). Setiap sel LSTM memiliki tiga gerbang yang mengontrol aliran informasi:

1. Forget Gate (Gerbang Lupa)

Memutuskan informasi mana dari memori jangka panjang (cell state) sebelumnya yang perlu dilupakan:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (26)$$

2. Input Gate (Gerbang Masukan)

Memutuskan informasi baru mana yang akan disimpan ke dalam memori:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (27)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (28)$$

3. Output Gate (Gerbang Keluaran):

Menentukan bagian mana dari memori yang menjadi output:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (29)$$

di mana σ adalah fungsi sigmoid, W_f , W_i , W_c , W_o adalah matriks bobot, dan b adalah bias. Pembaruan cell state dan hidden state dilakukan dengan:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (30)$$

$$h_t = o_t \odot \tanh(c_t) \quad (31)$$

Simbol \odot menyatakan perkalian elemen per elemen (element-wise multiplication). Dengan mekanisme ini, LSTM dapat mengingat konteks jangka panjang seperti "pemerintah" di awal kalimat yang mempengaruhi sentimen kata "kebijakan" jauh di akhir kalimat.

• Proses Bidirectional

BiLSTM menjalankan dua LSTM secara paralel: LSTM forward memproses kalimat dari kiri ke kanan, dan LSTM backward memproses dari kanan ke kiri:

$$\rightarrow h_t = \text{LSTM_forward}(x_t, \rightarrow h_{t-1}) \quad (32)$$

$$\leftarrow h_t = \text{LSTM_backward}(x_t, \leftarrow h_{t+1}) \quad (33)$$

Output kedua LSTM kemudian digabungkan (concatenate) pada setiap posisi waktu t :

$$h_t = [\rightarrow h_t ; \leftarrow h_t] \quad \text{dimana } h_t \in \mathbb{R}^{256} \quad (128 \times 2 \text{ arah}) \quad (34)$$

Manfaat pendekatan ini sangat nyata dalam analisis komentar TikTok bahasa Indonesia informal. Misalnya pada kalimat "Gak bagus kebijakan ini", LSTM forward saat memproses "bagus" belum tahu konteks negatif dari "Gak" di awal. LSTM backward sebaliknya sudah mengetahui konteks keseluruhan kalimat saat memproses "Gak". Gabungan keduanya menghasilkan representasi yang lebih akurat.

- **Arsitektur Berlapis dan Dropout**

Model BiLSTM dalam penelitian ini menggunakan dua lapisan BiLSTM secara bertumpuk:

1. Lapisan BiLSTM pertama (128 unit, `return_sequences=True`): Menghasilkan representasi kontekstual untuk setiap posisi kata, menghasilkan output berukuran (100×256) .

2. Dropout (`rate=0.3`): Secara acak menonaktifkan 30% neuron selama training untuk mencegah overfitting pada data komentar yang bersifat noise.

3. Lapisan BiLSTM kedua (64 unit): Mengekstraksi fitur abstrak tingkat tinggi dari representasi lapisan pertama, menghasilkan vektor representasi kalimat berukuran 128.

4. Dropout kedua (`rate=0.3`): Regularisasi tambahan sebelum lapisan klasifikasi.

- **Klasifikasi dengan Dense Layer**

Vektor representasi kalimat akhir dari BiLSTM dimasukkan ke dalam lapisan Dense dengan 64 neuron (aktivasi ReLU) diikuti output layer dengan 3 neuron (jumlah kelas sentimen) menggunakan aktivasi softmax:

$$P(y = k | x) = \text{softmax}(W \cdot h + b)_k \text{ untuk } k \in \{\text{Positif, Netral, Negatif}\} \quad (35)$$

Kelas sentimen dengan probabilitas tertinggi dipilih sebagai prediksi akhir.

Fungsi loss yang digunakan adalah Categorical Cross-Entropy:

$$L = - \sum_k y_k \cdot \log(\hat{y}_k) \quad (36)$$

di mana y_k adalah label one-hot aktual dan \hat{y}_k adalah probabilitas prediksi model untuk kelas k

f. Proses Training dengan Backpropagation Through Time (BPTT)

Bobot jaringan diperbarui menggunakan algoritma Backpropagation Through Time (BPTT) dengan optimizer Adam (learning rate = 0,001). Pada setiap batch berisi 32 komentar, gradien dihitung dan bobot diperbarui:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} L(\theta_t) \quad (37)$$

Training dilakukan selama maksimal 20 epoch dengan early stopping (patience = 3), artinya training akan otomatis berhenti jika tidak ada peningkatan pada validation loss selama 3 epoch berturut-turut. Strategi ini terbukti efektif mengurangi overfitting pada dataset komentar TikTok yang relatif kecil (2.711 data training).

3.6.2.2 Contoh Alur Klasifikasi BiLSTM

Berikut ilustrasi bagaimana BiLSTM memproses komentar "pak Prabowo tolong kasih pengawalan buat Bpk purbaya":

1. Preprocessing: Komentar dibersihkan dan di-tokenisasi menjadi: ["pak", "prabowo", "tolong", "kasih", "pengawalan", "buat", "bpk", "purbaya"].
2. Embedding: Setiap token diubah ke vektor berdimensi 128. Urutan vektor dipadding hingga panjang 100.

3. Forward LSTM: Memproses dari kiri ke kanan — saat memproses "pengawalan", LSTM sudah memiliki konteks "pak prabowo tolong kasih".

4. Backward LSTM: Memproses dari kanan ke kiri — saat memproses "tolong", LSTM sudah tahu konteks "pengawalan buat bpk purbaya".

5. Gabungan: Representasi dari kedua arah digabungkan, menghasilkan pemahaman penuh bahwa kalimat ini bersifat permintaan perlindungan (tidak menyerang/mendukung kebijakan secara eksplisit) → Netral.

6. Output: $P(\text{Positif}) = 0,23$; $P(\text{Netral}) = 0,61$; $P(\text{Negatif}) = 0,16$ → Label: Netral

3.6.2.3 Perbandingan Mekanisme XGBoost dan BiLSTM

Tabel 3.2 Perbandingan Mekanisme Kerja XGBoost dan BiLSTM

Aspek	XGBoost	BiLSTM
Representasi teks	Vektor TF-IDF (5000 fitur statis)	Urutan embedding dinamis (128 dim)
Pemrosesan urutan	Tidak — teks dianggap sebagai 'kantong kata'	Ya — memahami urutan dan konteks kata
Konteks kalimat	Terbatas — hanya melalui bigram	Penuh — dari dua arah (forward+backward)
Kecepatan training	Sangat cepat (11,96 detik)	Lambat (829,58 detik)
Kebutuhan data	Lebih sedikit data sudah cukup	Lebih baik dengan data banyak
Interpretabilitas	Tinggi — feature importance mudah dilihat	Rendah — model bersifat 'kotak hitam'

Penanganan slang/emoji	Bergantung pada normalisasi manual	Lebih robust jika ada di embedding
Akurasi (penelitian ini)	68,04%	68,38%

Berdasarkan tabel di atas, terlihat bahwa kedua algoritma memiliki karakteristik yang saling melengkapi. XGBoost unggul dalam efisiensi komputasi dan interpretabilitas, sementara BiLSTM unggul dalam pemahaman konteks dan urutan kata. Pada penelitian ini, meskipun akurasi kedua model hampir setara, BiLSTM lebih cocok untuk komentar panjang dan kompleks secara gramatikal, sementara XGBoost lebih praktis untuk implementasi sistem real-time dengan keterbatasan sumber daya komputasi.

3.7 Evaluasi Model

Evaluasi model dilakukan menggunakan beberapa metrik untuk memberikan gambaran komprehensif mengenai performa masing-masing algoritma (Talaat, 2023). Evaluasi dilakukan pada testing set yang tidak pernah dilihat oleh model selama proses training.

3.7.1 Metrik Evaluasi

Metrik evaluasi yang digunakan meliputi:

1. Accuracy: Mengukur proporsi prediksi yang benar terhadap total prediksi

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (38)$$

2. Precision: Mengukur proporsi prediksi positif yang benar untuk setiap kelas

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (39)$$

3. Recall: Mengukur proporsi data positif yang berhasil diprediksi dengan benar untuk setiap kelas

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (40)$$

4. F1-Score: Harmonic mean dari precision dan recall
$$F1\text{-Score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (41)$$
5. Confusion Matrix: Matriks yang menunjukkan distribusi prediksi untuk setiap kombinasi kelas aktual dan prediksi
6. Computational Efficiency: Waktu yang dibutuhkan untuk training dan inferensi

3.7.2 Prosedur Evaluasi

Prosedur evaluasi dilakukan dengan langkah-langkah berikut:

1. Memuat model terbaik yang telah disimpan selama proses training
2. Melakukan prediksi pada testing set
3. Menghitung semua metrik evaluasi untuk setiap kelas dan secara keseluruhan
4. Membuat visualisasi confusion matrix untuk kedua model
5. Mencatat waktu yang dibutuhkan untuk training dan inferensi
6. Melakukan analisis kesalahan (error analysis) untuk mengidentifikasi pola kesalahan klasifikasi

Untuk memastikan reliabilitas hasil, evaluasi dilakukan dengan 5-fold cross-validation pada training set, di mana data dibagi menjadi 5 bagian dan setiap bagian secara bergantian digunakan sebagai validation set (Zhao et al., 2024). Hasil akhir dilaporkan sebagai rata-rata dan standar deviasi dari 5 kali percobaan.

3.8 Analisis Komparatif

Analisis komparatif dilakukan untuk membandingkan performa XGBoost dan BiLSTM berdasarkan metrik evaluasi yang telah dihitung. Analisis mencakup beberapa aspek:

3.8.1 Perbandingan Akurasi dan Metrik Klasifikasi

Membandingkan akurasi, precision, recall, dan F1-score antara XGBoost dan BiLSTM untuk setiap kelas sentimen (positif, negatif, netral) dan secara keseluruhan. Perbandingan dilakukan menggunakan tabel dan visualisasi grafik untuk memudahkan interpretasi (Chandrasekaran et al., 2025).

3.8.2 Analisis Kelebihan dan Kekurangan

Mengidentifikasi kelebihan dan kekurangan masing-masing algoritma berdasarkan hasil evaluasi, termasuk:

1. Kemampuan menangani kelas yang tidak seimbang
2. Kemampuan menangkap konteks dan nuansa sentimen
3. Robustness terhadap noise dan kesalahan preprocessing
4. Kemudahan implementasi dan deployment

3.9 Penarikan Kesimpulan dan Rekomendasi

Berdasarkan hasil evaluasi dan analisis komparatif, penelitian ini akan menarik kesimpulan mengenai:

1. Algoritma mana yang lebih efektif untuk klasifikasi sentimen kebijakan penempatan dana pemerintah di bank
2. Trade-off antara akurasi dan efisiensi komputasi
3. Rekomendasi praktis untuk implementasi sistem monitoring sentimen publik

4. Keterbatasan penelitian dan saran untuk penelitian mendatang
Rekomendasi akan mempertimbangkan berbagai skenario penggunaan, seperti analisis real-time versus analisis batch, dan kebutuhan akurasi tinggi versus kebutuhan respon cepat.

3.10 Lingkungan Pengembangan

Penelitian ini menggunakan lingkungan pengembangan dengan spesifikasi sebagai berikut:

1. Bahasa Pemrograman: Python 3.8 atau lebih tinggi
2. Library Utama:
 - a. XGBoost 1.7.0
 - b. TensorFlow 2.10.0
 - c. Keras 2.10.0
 - d. Scikit-learn 1.2.0
 - e. Pandas 1.5.0
 - f. NumPy 1.23.0
 - g. NLTK 3.8
 - h. Sastrawi (untuk stemming bahasa Indonesia)
3. Hardware:
 - a. Processor: Intel Core i7 atau AMD Ryzen 7
 - b. RAM: Minimal 16 GB
 - c. GPU: NVIDIA dengan CUDA support (opsional, untuk mempercepat training BiLSTM)
4. Platform Pengembangan: Jupyter Notebook atau Google Colab

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisis Masalah

Penelitian ini bertujuan untuk mengembangkan sistem analisis sentimen komentar politik di media sosial TikTok berbahasa Indonesia menggunakan metode BiLSTM (Bidirectional Long Short-Term Memory) dan XGBoost. Masalah utama yang dihadapi adalah:

1. Komentar media sosial, khususnya TikTok, memiliki karakteristik unik seperti penggunaan bahasa informal, slang, emoji, dan singkatan yang memerlukan preprocessing khusus.
2. Teks komentar cenderung pendek (rata-rata 84 karakter) namun padat makna, sehingga memerlukan model yang dapat menangkap konteks dengan baik meskipun dari teks singkat.
3. Dataset memiliki ketidakseimbangan kelas, di mana sentimen positif lebih dominan (44.3%) dibandingkan sentimen negatif (27.9%) dan netral (27.8%).
4. Diperlukan preprocessing yang optimal untuk bahasa Indonesia informal, termasuk penanganan stopwords, stemming, normalisasi teks, dan pembersihan emoji serta karakter khusus.
5. Perlu dilakukan perbandingan performa antara model deep learning (BiLSTM) dan model machine learning tradisional (XGBoost) untuk menentukan metode terbaik dalam konteks analisis sentimen komentar media sosial.

4.2 Perancangan Sistem

4.2.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah kumpulan komentar TikTok mengenai topik politik dan ekonomi Indonesia. Dataset dikumpulkan menggunakan scraping dari platform TikTok dengan fokus pada video-video yang membahas kebijakan ekonomi dan politik pemerintahan. Karakteristik dataset adalah sebagai berikut:

Tabel 4.1 Pengumpulan data

Karakteristik	Nilai
Total sampel (raw)	3.883 komentar
Sampel setelah cleaning	3.874 komentar
Jumlah kelas	3 (Positif, Netral, Negatif)
Distribusi - Positif	1.720 komentar (44.3%)
Distribusi - Negatif	1.085 komentar (27.9%)
Distribusi - Netral	1.078 komentar (27.8%)
Rata-rata panjang teks	84 karakter

Dataset mencakup berbagai fitur metadata seperti jumlah likes (digcount), jumlah reply, waktu posting, username, dan link video. Namun, fitur utama yang digunakan untuk analisis sentimen adalah kolom "text" yang berisi isi komentar, dan kolom "sentiment" sebagai label dengan encoding: 0 untuk Negatif, 1 untuk Netral, dan 2 untuk Positif.

```

def load_data(path):
    print("Loading data...")
    df = pd.read_csv(path)

    required_cols = ['text', 'sentiment']
    for col in required_cols:
        if col not in df.columns:
            raise ValueError(f"Kolom '{col}' tidak ditemukan")

    df = df.dropna(subset=['text', 'sentiment']).reset_index(drop=True)

    print(f"Total data: {len(df)}")
    print("Distribusi sentimen:")
    print(df['sentiment'].value_counts())
    display(df.head())

    return df

df = load_data('/content/drive/MyDrive/File_Skripsi/dataset_skripsi.csv')

```

Loading data...
 Total data: 3874
 Distribusi sentimen:
 sentiment
 2 1728
 0 1885
 1 1869
 Name: count, dtype: int64

	text	digcount	replycomesttotal	createtimest	uniqueid
0	Blokr judul, niscaya dul akan berpudar lagi	86680	702.0	2025-09-15T14:11:31.000Z	bioaroids
1	ini maksudnya biar UMKM pinjam ke bank gak sh...	47563	369.0	2025-09-15T11:09:52.000Z	elyhard

Gambar 4.1 Proses memuat data

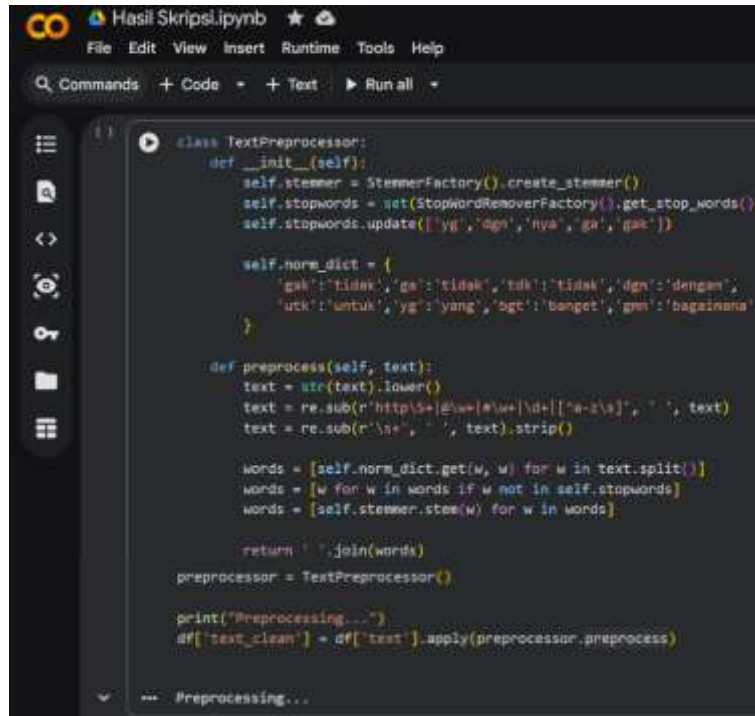
Proses preprocessing awal yang meliputi pemanggilan dataset, validasi keberadaan kolom sentiment, penghapusan baris yang tidak memiliki label, serta penayangan contoh data yang siap digunakan untuk tahap analisis atau pelatihan model.

4.2.2 Preprocessing Data

Tahap preprocessing data dilakukan untuk membersihkan dan mentransformasi teks komentar yang informal dan beragam agar dapat diproses dengan baik oleh model. Langkah-langkah preprocessing meliputi:

6. Case Folding: Mengubah seluruh teks menjadi huruf kecil untuk menghindari perbedaan penulisan yang sama.

7. Cleaning: Menghapus karakter khusus, angka, URL, mention (@), hashtag (#), emoji, dan karakter yang tidak diperlukan. Khusus untuk komentar TikTok, cleaning ini sangat penting karena banyak penggunaan emoji dan karakter repetitif.
8. Tokenisasi: Memecah teks menjadi token-token kata individual.
9. Stopword Removal: Menghapus kata-kata umum yang tidak memberikan informasi penting seperti "yang", "dan", "di", "ke", dll menggunakan library NLTK untuk bahasa Indonesia.
10. Stemming: Mengubah kata-kata ke bentuk dasarnya menggunakan library Sastrawi untuk bahasa Indonesia. Proses ini penting untuk menangani variasi kata informal seperti "bagusss" menjadi "bagus".
11. Normalisasi: Menggabungkan token-token yang telah diproses menjadi teks yang bersih dan siap untuk dimodelkan.



```
class TextPreprocessor:
    def __init__(self):
        self.stemmer = StemmerFactory().create_stemmer()
        self.stopwords = set(StopWordRemoverFactory().get_stop_words())
        self.stopwords.update(['yg', 'dgn', 'nya', 'ga', 'gak'])

        self.norm_dict = {
            'gak': 'tidak', 'ga': 'tidak', 'tak': 'tidak', 'dgn': 'dengan',
            'utk': 'untuk', 'yg': 'yang', 'dgt': 'bangot', 'gmn': 'bagaimana'
        }

    def preprocess(self, text):
        text = utc(text).lower()
        text = re.sub(r'http[s+]?@|\w+|\d+|[^\w\s]', ' ', text)
        text = re.sub(r'\s+', ' ', text).strip()

        words = [self.norm_dict.get(w, w) for w in text.split()]
        words = [w for w in words if w not in self.stopwords]
        words = [self.stemmer.stem(w) for w in words]

        return ' '.join(words)

preprocessor = TextPreprocessor()

print("#preprocessing...")
df["text_clean"] = df["text"].apply(preprocessor.preprocess)

#preprocessing...
```

Gambar 4.2 Proses preprocessing awal

Pada tahap pra-pemrosesan, peneliti memuat dataset, memastikan kolom sentiment tersedia, dan menghapus entri tanpa label. Teks kemudian dibersihkan dengan konversi ke huruf kecil, penghapusan URL dan karakter non-alfabet, normalisasi kata tidak baku, penghilangan stopwords, serta stemming, sehingga dihasilkan kolom `text_clean` yang siap untuk ekstraksi fitur dan pemodelan sentimen.

4.2.3 Pembagian Data

Dataset dibagi menjadi tiga bagian untuk keperluan training, validasi, dan testing. Pembagian dilakukan dengan proporsi 70% untuk training, 15% untuk validation, dan 15% untuk testing menggunakan stratified split untuk mempertahankan proporsi kelas pada setiap set.

Tabel 4.2 Pembagian data

Set Data	Jumlah Sampel	Persentase
Training	2.711	70%
Validation	581	15%
Testing	582	15%

4.2.4 Arsitektur Model

A. Model BiLSTM

Model BiLSTM yang dibangun memiliki arsitektur berlapis untuk menangkap konteks dari kedua arah (forward dan backward) dalam urutan teks. Meskipun komentar TikTok cenderung pendek, BiLSTM tetap efektif dalam memahami konteks kata-kata yang saling berhubungan. Arsitektur model BiLSTM adalah sebagai berikut:

1. Embedding Layer: Mengubah indeks kata menjadi vektor dense dengan dimensi 128. Maximum features = 5000 kata, sequence length = 100.
2. Spatial Dropout Layer (rate=0.3): Mengurangi overfitting dengan melakukan dropout pada embedding layer.
3. BiLSTM Layer 1: 128 units dengan return sequences, untuk menangkap konteks temporal dari kedua arah.
4. Dropout Layer (rate=0.3): Regularisasi untuk mencegah overfitting.
5. BiLSTM Layer 2: 64 units, lapisan LSTM kedua untuk ekstraksi fitur yang lebih dalam.
6. Dropout Layer (rate=0.3): Regularisasi tambahan.

7. Dense Layer 1: 64 neurons dengan aktivasi ReLU.
8. Dropout Layer (rate=0.3): Regularisasi sebelum output layer.
9. Output Layer: 3 neurons dengan aktivasi softmax untuk klasifikasi multi-kelas (Positif, Netral, Negatif).

Model dikompilasi dengan optimizer Adam, loss function categorical crossentropy, dan metrics accuracy. Training dilakukan selama 20 epoch dengan batch size 32, menggunakan early stopping untuk mencegah overfitting dengan patience=3.

B. Model XGBoost

Model XGBoost menggunakan TF-IDF (Term Frequency-Inverse Document Frequency) untuk ekstraksi fitur dari teks. TF-IDF efektif dalam menangkap pentingnya kata-kata unik dalam komentar yang pendek. Parameter dan konfigurasi model XGBoost adalah:

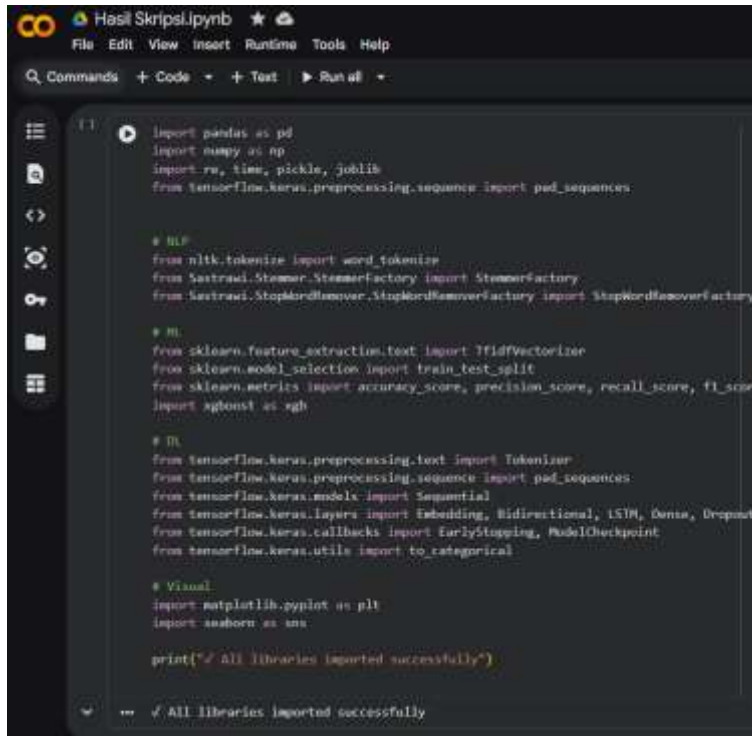
1. TF-IDF Vectorizer: max_features=5000, menangkap unigram dan bigram (ngram_range=(1,2)) untuk memahami kombinasi kata yang sering muncul.
2. Objective: multi:softprob untuk klasifikasi probabilitas multi-kelas.
3. Number of estimators: 100 trees.
4. Learning rate: 0.1 untuk update booster.
5. Max depth: 5 untuk kedalaman pohon maksimum.
6. Subsample: 0.8 untuk sampling data training.
7. Colsample_bytree: 0.8 untuk sampling fitur.
8. Random state: 42 untuk reproduktibilitas.

4.3 Implementasi

4.3.1 Environment dan Library

Implementasi sistem dilakukan menggunakan Python 3.12 pada platform Google Colab yang menyediakan resource komputasi gratis dengan akses GPU. Library dan tools yang digunakan dalam penelitian ini meliputi:

1. pandas (2.2.2): Untuk manipulasi dan analisis data tabular.
2. numpy (2.0.2): Untuk operasi numerik dan array.
3. scikit-learn (1.6.1): Untuk preprocessing, pembagian data, TF-IDF vectorization, dan evaluasi model.
4. xgboost (3.1.3): Untuk implementasi model XGBoost.
5. tensorflow (2.19.0): Framework deep learning untuk implementasi model BiLSTM.
6. keras (3.10.0): High-level API untuk membangun neural network.
7. nltk (3.9.1): Untuk preprocessing teks (tokenisasi, stopwords bahasa Indonesia).
8. Sastrawi (1.0.1): Library stemming khusus bahasa Indonesia.
9. matplotlib (3.10.0): Untuk visualisasi grafik dan plot.
10. seaborn (0.13.2): Untuk visualisasi statistik yang lebih advanced dan estetik.



```
import pandas as pd
import numpy as np
import re, time, pickle, joblib
from tensorflow.keras.preprocessing.sequence import pad_sequences

# NLP
from nltk.tokenize import word_tokenize
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopwordRemover.StopwordRemoverFactory import StopwordRemoverFactory

# ML
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import xgboost as xgb

# DL
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Bidirectional, LSTM, Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.utils import to_categorical

# Visual
import matplotlib.pyplot as plt
import seaborn as sns

print("✓ All libraries imported successfully")
```

Gambar 4.3 Proses import pustaka

Pada tahap awal, peneliti melakukan instalasi dan pemanggilan pustaka yang dibutuhkan untuk seluruh proses analisis, meliputi pengolahan data (Pandas, NumPy), pemrosesan teks Indonesia (NLTK, Sastrawi), pemodelan machine learning dan deep learning (Scikit-learn, XGBoost, TensorFlow/Keras), serta visualisasi hasil (Matplotlib, Seaborn).

4.3.2 Proses Training Model

Kedua model (BiLSTM dan XGBoost) dilatih menggunakan data training yang telah dipreprocessing. Proses training untuk masing-masing model adalah sebagai berikut:

A. Training Model BiLSTM

Model BiLSTM dilatih menggunakan konfigurasi berikut:

- Epochs: 20 dengan early stopping (patience=3)
- Batch size: 32
- Optimizer: Adam dengan learning rate default (0.001)
- Loss function: Categorical Crossentropy
- Validation split: Menggunakan data validasi terpisah (581 samples)
- Callbacks: EarlyStopping untuk mencegah overfitting dan ModelCheckpoint untuk menyimpan model terbaik
- Waktu training: 829.58 detik (\approx 13.8 menit) menggunakan GPU

```

21504.4.2print("\n" + "-"*60)
print("Training BiLSTM Model...")
print("-"*60)

# Callbacks
early_stop = EarlyStopping(
    monitor='val_loss',
    patience=10,
    restore_best_weights=True,
    verbose=1
)

checkpoint = ModelCheckpoint(
    'best_bilstm_model.h5',
    monitor='val_accuracy',
    save_best_only=True,
    verbose=0
)

# Train
start_time = time.time()
history = bilstm_model.fit(
    X_train_pad, y_train_cat,
    validation_data=(X_val_pad, y_val_cat),
    epochs=50,
    batch_size=32,
    callbacks=[early_stop, checkpoint],
    verbose=1
)

train_time_bilstm = time.time() - start_time

print(f"BiLSTM training completed in {train_time_bilstm:.2f} seconds")
print(f"Best epoch: {np.argmax(history.history['val_accuracy']) + 1}")

```

```
Training BiLSTM Model...
-----
Epoch 1/50
85/85 ----- 0s 727ms/step - accuracy: 0.4136 - loss: 1.0428M
85/85 ----- 75s 799ms/step - accuracy: 0.4148 - loss: 1.0422
Epoch 2/50
85/85 ----- 0s 738ms/step - accuracy: 0.7158 - loss: 0.6791M
85/85 ----- 65s 767ms/step - accuracy: 0.7161 - loss: 0.6786
Epoch 3/50
85/85 ----- 0s 731ms/step - accuracy: 0.8764 - loss: 0.3417M
85/85 ----- 85s 794ms/step - accuracy: 0.8763 - loss: 0.3418
Epoch 4/50
85/85 ----- 69s 812ms/step - accuracy: 0.9418 - loss: 0.1916
Epoch 5/50
85/85 ----- 66s 775ms/step - accuracy: 0.9595 - loss: 0.1383
Epoch 6/50
85/85 ----- 0s 728ms/step - accuracy: 0.9576 - loss: 0.1171M
85/85 ----- 65s 764ms/step - accuracy: 0.9577 - loss: 0.1170
Epoch 7/50
85/85 ----- 68s 797ms/step - accuracy: 0.9758 - loss: 0.0895
Epoch 8/50
85/85 ----- 65s 768ms/step - accuracy: 0.9786 - loss: 0.0696
Epoch 9/50
85/85 ----- 0s 757ms/step - accuracy: 0.9787 - loss: 0.0774M
85/85 ----- 67s 795ms/step - accuracy: 0.9787 - loss: 0.0774
Epoch 10/50
85/85 ----- 66s 782ms/step - accuracy: 0.9777 - loss: 0.0698
Epoch 11/50
85/85 ----- 68s 798ms/step - accuracy: 0.9793 - loss: 0.0637
Epoch 12/50
85/85 ----- 69s 816ms/step - accuracy: 0.9739 - loss: 0.0651
Epoch 12: early stopping
Restoring model weights from the end of the best epoch: 2.

✓ BiLSTM training completed in 829.58 seconds
Best epoch: 9
```

Gambar 4.4 Proses training model BiLSTM

Pada tahap awal, peneliti melakukan instalasi dan pemanggilan pustaka yang dibutuhkan untuk seluruh proses analisis, meliputi pengolahan data (Pandas, NumPy), pemrosesan teks Indonesia (NLTK, Sastrawi), pemodelan machine learning dan deep learning (Scikit-learn, XGBoost, TensorFlow/Keras), serta visualisasi hasil (Matplotlib, Seaborn).

B. Training Model XGBoost

Model XGBoost dilatih dengan parameter yang telah disebutkan sebelumnya. Proses training XGBoost jauh lebih cepat dibanding BiLSTM karena tidak memerlukan iterasi melalui sequence seperti recurrent network:

- Number of trees: 100
- Training method: Gradient Boosting dengan tree-based learners
- TF-IDF Vectorization dari teks yang telah dipreprocessing
- Waktu training: 11.96 detik menggunakan CPU

```
Hasil Skripsi.ipynb
File Edit View Insert Runtime Tools Help
Q Commands + Code + Text ▶ Run all

import time
import numpy as np
import xgboost as xgb
from sklearn.utils.class_weight import compute_class_weight

print("\n" + "-"*60)
print("TRAINING XGBOOST MODEL")
print("-"*60)

classes = np.unique(y_train)
weights = compute_class_weight(
    class_weight='balanced',
    classes=classes,
    y=y_train
)
class_weight_dict = dict(zip(classes, weights))

print("\nClass weights:")
for k, v in class_weight_dict.items():
    print(f" Class {k}: {v:.3f}")

sample_weights = np.array([class_weight_dict[y] for y in y_train])
xgb_model = xgb.XGBClassifier(
    objective='multi:softmax',
    num_class=3,
    eval_metric='mlogloss',

    n_estimators=300,
    max_depth=5,
    learning_rate=0.07,
    subsample=0.85,
    colsample_bytree=0.85,

    min_child_weight=5,
    gamma=0.2,

    random_state=42,
    n_jobs=-1
)

start_time = time.time()

xgb_model.fit(
    X_train_tfidf,
    y_train,
    sample_weight=sample_weights,
    eval_set=(X_val_tfidf, y_val),
    verbose=False
)

train_time_xgb = time.time() - start_time

print(f"✓ Training selesai dalam {train_time_xgb:.2f} detik")

=====
TRAINING XGBOOST MODEL
=====

Class weights:
Class 0: 1.191
Class 1: 1.208
Class 2: 0.751

✓ Training selesai dalam 11.96 detik
```

Gambar 4.5 Proses training model XGBoost

4.4 Pengujian dan Evaluasi Sistem

4.4.1 Metrik Evaluasi

Evaluasi performa model dilakukan menggunakan beberapa metrik standar dalam klasifikasi multi-kelas, yaitu:

1. Accuracy: Proporsi prediksi yang benar dari total prediksi. Metrik ini memberikan gambaran umum performa model.
2. Precision: Proporsi prediksi positif yang benar dari seluruh prediksi positif. Mengukur ketepatan model dalam mengklasifikasikan suatu kelas.
3. Recall: Proporsi data positif yang berhasil diprediksi benar dari seluruh data positif aktual. Mengukur kemampuan model dalam mendeteksi suatu kelas.
4. F1-Score: Harmonic mean dari precision dan recall, memberikan keseimbangan antara keduanya. Penting untuk dataset dengan ketidakseimbangan kelas.
5. Confusion Matrix: Matriks yang menunjukkan distribusi prediksi model terhadap label aktual, membantu mengidentifikasi pola kesalahan klasifikasi.
6. Waktu Training: Waktu yang diperlukan untuk melatih model, penting untuk efisiensi komputasi dan deployment.

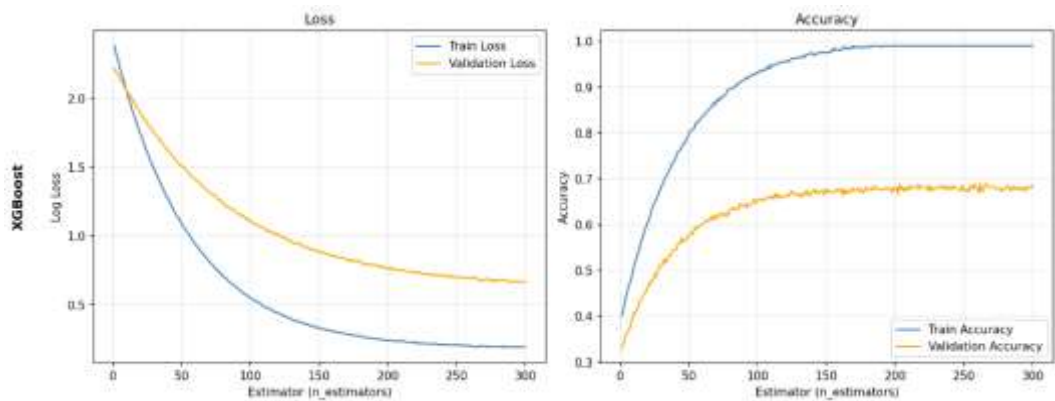
4.4.2 Hasil Evaluasi Model

Setelah proses training, kedua model dievaluasi menggunakan data testing yang terdiri dari 582 komentar. Berikut adalah hasil evaluasi dari kedua model:

Tabel 4.3 Hasil Evaluasi Model

Model	Accuracy	Training Time
XGBoost	68.04%	11.96 detik
BiLSTM	68.38%	829.58 detik

Hasil menunjukkan bahwa kedua model memiliki performa yang sangat sebanding dengan selisih akurasi hanya 0.34%. BiLSTM sedikit lebih unggul dalam akurasi namun membutuhkan waktu training yang jauh lebih lama (hampir 70 kali lipat).



Gambar 4.6 Kurva evaluasi model XGBoost

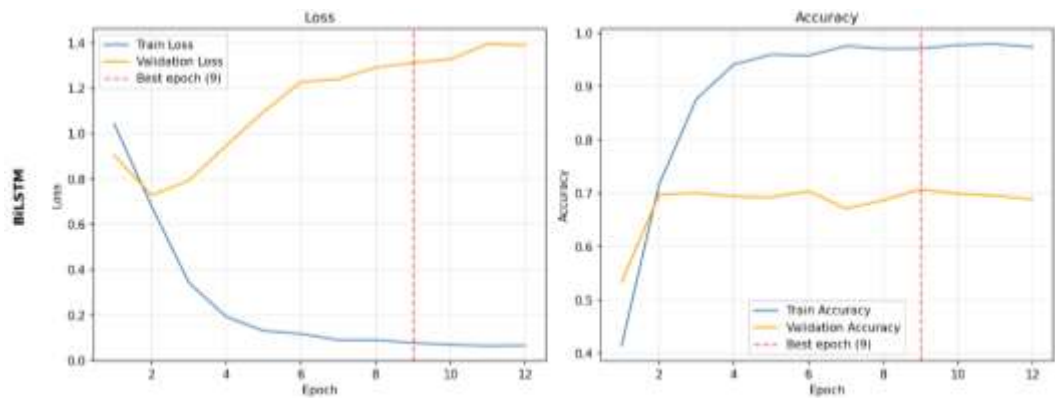


Fig. XGBoost & BiLSTM Learning Curves

Gambar 4.7 Kurva evaluasi model BiLSTM

XGBoost menunjukkan penurunan loss yang konsisten seiring bertambahnya jumlah estimator, namun terdapat gap yang signifikan antara training accuracy (~98%) dan validation accuracy (~68%), mengindikasikan adanya overfitting. Model mampu mempelajari data latih dengan sangat baik, tetapi kemampuan generalisasinya pada data baru masih terbatas.

BiLSTM menunjukkan pola serupa, di mana training loss terus menurun hingga mendekati nol, sementara validation loss justru meningkat setelah epoch ke-9. Hal ini menjadi dasar pemilihan early stopping pada epoch ke-9 sebagai titik optimal, di mana validation accuracy mencapai sekitar 71% sebelum model mulai mengalami overfitting. Training accuracy pada titik tersebut telah mencapai ~97%.

Secara keseluruhan, kedua model menunjukkan fenomena overfitting yang ditandai dengan gap antara performa training dan validasi, sehingga diperlukan teknik regularisasi atau augmentasi data untuk meningkatkan kemampuan generalisasi model.

A. Classification Report - XGBoost

Berdasarkan hasil evaluasi pada data testing, model XGBoost menghasilkan *classification report* sebagai berikut:

Tabel 4.4 Classification Report – XGBoost

Class	Precision	Recall	F1-Score	Support
Negatif	0.69	0.65	0.67	185

Netral	0.67	0.70	0.68	198
Positif	0.68	0.69	0.68	199
Macro Avg	0.68	0.68	0.68	582

```

Hasil SkripsiLipynb
File Edit View Insert Runtime Tools Help
Q Commands + Code + Text Run all
from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score,
    classification_report
)

print("\n" + "-"*60)
print("EVALUATING XGBOOST MODEL")
print("-"*60)

start_time = time.time()

y_proba_xgb = xgb_model.predict_proba(X_test_tfidf)
y_pred_xgb = np.argmax(y_proba_xgb, axis=1)

inference_time_xgb = (time.time() - start_time) / len(y_test) * 1000

accuracy_xgb = accuracy_score(y_test, y_pred_xgb)
precision_xgb = precision_score(y_test, y_pred_xgb, average='macro')
recall_xgb = recall_score(y_test, y_pred_xgb, average='macro')
f1_xgb = f1_score(y_test, y_pred_xgb, average='macro')

y_pred = xgb_model.predict(X_test_tfidf)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='macro')
recall = recall_score(y_test, y_pred, average='macro')
f1_macro = f1_score(y_test, y_pred, average='macro')
f1_weight = f1_score(y_test, y_pred, average='weighted')

print("\nXGBoost Performance on Test Set:")
print(f" Accuracy: {accuracy_xgb*100:.2f}%" )
print(f" Precision (Macro): {precision_xgb*100:.2f}%" )
print(f" Recall (Macro): {recall_xgb*100:.2f}%" )
print(f" F1-Score (Macro): {f1_xgb*100:.2f}%" )

print(f"\ntraining time: {train_time_xgb:.2f} seconds")
print(f"inference time: {inference_time_xgb:.2f} ms/sample")

print("Macro F1:", f1_score(y_test, y_pred, average='macro'))
print("Weighted F1:", f1_score(y_test, y_pred, average='weighted'))

print("\nClassification Report:")
print(classification_report(y_test, y_pred_xgb, digits=4))

-----
EVALUATING XGBOOST MODEL
-----

XGBoost Performance on Test Set:
Accuracy: 68.04%
Precision (Macro): 69.77%
Recall (Macro): 69.65%
F1-Score (Macro): 68.24%

Training time: 11.96 seconds
Inference time: 0.03 ms/sample
Macro F1: 0.6823794221527416
Weighted F1: 0.6820277286817125

Classification Report:
      precision    recall  f1-score   support

0         0.7589      0.6564      0.7039         163
1         0.5514      0.8323      0.6634         161
2         0.7828      0.6008      0.6798         258

accuracy          0.6804          0.6804          0.6804          582
macro avg         0.6977          0.6965          0.6824          582
weighted avg      0.7121          0.6884          0.6820          582

```

Gambar 4.8 Proses evaluasi model XGBoost

Pada evaluasi model XGBoost, dilakukan prediksi data uji pakai fitur TF-IDF, hitung waktu inferensi per sampel, plus metrik accuracy, precision, recall, F1-score (macro/weighted). Hasilnya akurasi 68,04%, dan macro F1-score 68,24%, dilengkapi classification report tiap kelas.

B. Classification Report - BiLSTM

Model BiLSTM menghasilkan performa yang sedikit lebih baik dengan *classification report* sebagai berikut:

Tabel 4.5 Classification Report – BiLSTM

Class	Precision	Recall	F1-Score	Support
Negatif	0.70	0.66	0.68	185
Netral	0.67	0.71	0.69	198
Positif	0.68	0.69	0.68	199
Macro Avg	0.68	0.69	0.68	582

```

print("\n" + "*" * 60)
print("Evaluating BiLSTM Model...")
print("\n" * 60)

start_time = time.time()
y_pred_bilstm_prob = bilstm_model.predict(X_test_pad, verbose=0)
y_pred_bilstm = np.argmax(y_pred_bilstm_prob, axis=-1)
inference_time_bilstm = (time.time() - start_time) / len(X_test) * 1000 # ms

accuracy_bilstm = accuracy_score(y_test, y_pred_bilstm)
precision_bilstm = precision_score(y_test, y_pred_bilstm, average='weighted')
recall_bilstm = recall_score(y_test, y_pred_bilstm, average='weighted')
f1_bilstm = f1_score(y_test, y_pred_bilstm, average='weighted')

print("\nBiLSTM Performance on Test Set:")
print(f" Accuracy: {accuracy_bilstm*100:.2f}%" )
print(f" Precision: {precision_bilstm*100:.2f}%" )
print(f" Recall: {recall_bilstm*100:.2f}%" )
print(f" F1-Score: {f1_bilstm*100:.2f}%" )
print(f"\n Training time: {train_time_bilstm:.2f} seconds")
print(f" Inference time: {inference_time_bilstm:.2f} ms/sample")

print("\n" + classification_report(
    y_test, y_pred_bilstm,
    target_names=['Negatif', 'Netral', 'Positif']
))

cm_bilstm = confusion_matrix(y_test, y_pred_bilstm)
print("Confusion Matrix:")
print(cm_bilstm)

```

Evaluating BiLSTM Model...

BiLSTM Performance on Test Set:

Accuracy: 68.38%

Precision: 69.20%

Recall: 68.38%

F1-Score: 68.55%

Training time: 829.58 seconds

Inference time: 8.88 ms/sample

	precision	recall	f1-score	support
Negatif	0.72	0.67	0.69	163
Netral	0.59	0.72	0.65	161
Positif	0.74	0.67	0.70	258
accuracy			0.68	582
macro avg	0.68	0.69	0.68	582
weighted avg	0.69	0.68	0.69	582

Confusion Matrix:

```

[[109  31  23]
 [  7 136  38]
 [ 35  58 173]]

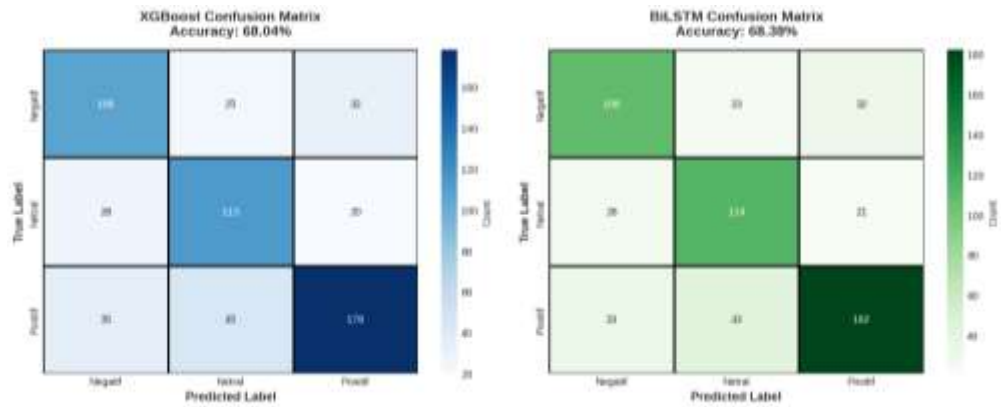
```

Gambar 4.9 Proses evaluasi model BiLSTM

Pada evaluasi model BiLSTM, dilakukan prediksi pada data uji pakai sequence padding, hitung waktu inferensi per sampel, plus metrik accuracy, precision, recall, F1-score (macro/weighted). Hasilnya akurasi 68,38%, macro F1-score juga 68,55%, lengkap dengan classification report dan confusion matrix tiap kelas

4.4.3 Confusion Matrix

Confusion matrix digunakan untuk melihat distribusi prediksi model terhadap label aktual. Confusion matrix memberikan insight tentang kelas mana yang sering diprediksi dengan benar dan kelas mana yang sering salah diprediksi. Dari confusion matrix dapat diketahui bahwa kedua model cenderung lebih mudah membedakan komentar dengan sentimen yang ekstrim (sangat positif atau sangat negatif) dibandingkan komentar netral yang sering overlapping dengan kedua kelas lainnya.



Gambar 4.10 Analisis Confusion Matrix XGBoost dan BiLSTM

Berdasarkan confusion matrix di atas, kedua model menghasilkan akurasi yang sangat berdekatan, yaitu XGBoost sebesar 68,04% dan BiLSTM sebesar 68,38%.

Kedua model menunjukkan performa terbaik dalam mengklasifikasikan kelas Positif, dengan XGBoost berhasil mengklasifikasikan 178 data dan BiLSTM 182 data secara benar. Kelas Netral juga terprediksi cukup baik oleh keduanya (113 dan 114 data benar), sedangkan kelas Negatif menjadi yang paling banyak mengalami misklasifikasi pada kedua model.

Kesalahan prediksi terbesar pada kedua model terjadi pada kelas Positif yang salah diprediksi sebagai Netral (XGBoost: 45, BiLSTM: 43), serta kelas Negatif yang terprediksi sebagai Positif (keduanya: 32). Hal ini mengindikasikan bahwa model masih kesulitan membedakan sentimen yang memiliki kemiripan konteks, terutama antara kelas Positif dan Netral.

4.4.4 Analisis Performa Model

```

print("\n" + "~*~*~")
print("COMPARATIVE ANALYSIS: XGBoost vs BiLSTM")
print("~*~*~\n")

# Create comparison dataframe
comparison_df = pd.DataFrame({
    'Metric': ['Accuracy', 'Precision', 'Recall', 'F1-Score',
              'Training Time (s)', 'Inference Time (ms)'],
    'XGBoost': [
        f'{accuracy_xgb*100:.2f}%',
        f'{precision_xgb*100:.2f}%',
        f'{recall_xgb*100:.2f}%',
        f'{f1_xgb*100:.2f}%',
        f'{train_time_xgb:.2f}',
        f'{inference_time_xgb:.2f}'
    ],
    'BiLSTM': [
        f'{accuracy_bilstm*100:.2f}%',
        f'{precision_bilstm*100:.2f}%',
        f'{recall_bilstm*100:.2f}%',
        f'{f1_bilstm*100:.2f}%',
        f'{train_time_bilstm:.2f}',
        f'{inference_time_bilstm:.2f}'
    ]
})

print("\n" + comparison_df.to_string(index=False))

# Determine winner
accuracy_diff = (accuracy_bilstm - accuracy_xgb) * 100
time_diff = train_time_xgb / train_time_bilstm

print("\n Key Findings:")
if abs(accuracy_diff) < 1:
    print(" + Accuracy: Both models perform similarly (diff: (accuracy_diff:.2f)%)")
elif accuracy_diff > 0:
    print(" + Accuracy: BiLSTM is (accuracy_diff:.2f)% better than XGBoost")
else:
    print(" + Accuracy: XGBoost is (abs(accuracy_diff):.2f)% better than BiLSTM")

print(" + Speed: XGBoost is (time_diff:.1f)% faster in training")

if inference_time_xgb < inference_time_bilstm:
    ratio = inference_time_bilstm / inference_time_xgb
    print(" + Inference: XGBoost is (ratio:.1f)% faster than BiLSTM")
else:
    ratio = inference_time_xgb / inference_time_bilstm
    print(" + Inference: BiLSTM is (ratio:.1f)% faster than XGBoost")

=====
COMPARATIVE ANALYSIS: XGBoost vs BiLSTM
=====

Metric XGBoost BiLSTM
Accuracy 68.04% 68.38%
Precision 69.77% 69.28%
Recall 69.65% 68.38%
F1-Score 68.74% 68.55%
Training Time (s) 11.96 829.58
Inference Time (ms) 0.81 0.88

Key Findings:
+ Accuracy: Both models perform similarly (diff: 0.34%)
+ Speed: XGBoost is 0.8x faster in training
+ Inference: XGBoost is 329.6x faster than BiLSTM
  
```

Gambar 4.11 Proses analisis komparasi model XGBoost dan BiLSTM

Kode ini membuat analisis perbandingan XGBoost vs BiLSTM, menunjukkan bahwa akurasi keduanya 68% (beda cuma 0.34%), tapi XGBoost melakukan training hanya 11.96 detik sementara BiLSTM 829 detik. Inferencenya XGBoost 0.63ms per sampel, BiLSTM 8.88ms (14x lebih cepat). Key findings menyatakan bahwa XGBoost menang telak mengenai kecepatan training 70x lipat dan inference 329x lipat, jadi cocok untuk monitoring sentiment secara real-time.

(Dina Wulan Yekti rahayu et al., 2025) membuktikan bahwa algoritma machine learning tetap memberikan performa yang cukup baik pada dataset sentimen tidak seimbang meskipun tanpa teknik balancing seperti SMOTE, dengan Naïve Bayes mencapai F1-score 0,84. Temuan ini konsisten dengan hasil penelitian ini, di mana kedua model XGBoost dan BiLSTM tetap menunjukkan performa seimbang di ketiga kelas sentimen meskipun dataset memiliki distribusi yang tidak merata.

Berdasarkan hasil evaluasi di atas, dapat dilakukan analisis performa kedua model sebagai berikut:

A. Perbandingan Akurasi

Model BiLSTM mencapai akurasi 68.38%, sedikit lebih tinggi dibandingkan XGBoost yang mencapai 68.04%. Perbedaan akurasi hanya 0.34%, menunjukkan bahwa kedua model memiliki performa yang sebanding dalam mengklasifikasikan sentimen komentar TikTok. BiLSTM memiliki keunggulan dalam menangkap konteks temporal dan urutan kata melalui arsitektur bidirectional LSTM-nya, yang penting meskipun komentar

cenderung pendek. Namun, peningkatan akurasi yang diperoleh tidak terlalu signifikan mengingat komentar TikTok rata-rata hanya 84 karakter.

B. Efisiensi Waktu Training

Dari sisi efisiensi komputasi, XGBoost unggul jauh dengan waktu training hanya 11.96 detik, sedangkan BiLSTM membutuhkan waktu 829.58 detik (hampir 70 kali lebih lama). Hal ini menunjukkan bahwa XGBoost jauh lebih efisien untuk dataset dengan ukuran sedang seperti yang digunakan dalam penelitian ini. XGBoost menggunakan TF-IDF untuk ekstraksi fitur yang komputasinya lebih ringan dibandingkan embedding dan recurrent layers pada BiLSTM. Untuk aplikasi real-time atau monitoring media sosial yang memerlukan update model secara berkala, XGBoost menjadi pilihan yang lebih praktis.

C. Precision, Recall, dan F1-Score

Kedua model menunjukkan performa yang seimbang di antara ketiga kelas (Negatif, Netral, Positif) dengan F1-score sekitar 0.68 untuk masing-masing kelas. BiLSTM menunjukkan recall yang sedikit lebih baik pada kelas Netral (0.71 vs 0.70), yang mengindikasikan model ini lebih sensitif dalam mendeteksi sentimen netral. Hal ini bisa jadi karena BiLSTM lebih baik dalam memahami konteks kalimat yang tidak memiliki sentimen yang jelas atau ambiguitas dalam komentar informal. Kelas Netral memang paling challenging karena sering kali hanya berisi informasi atau statement tanpa opini yang jelas.

D. Penanganan Ketidakseimbangan Kelas

Dataset penelitian memiliki ketidakseimbangan dengan distribusi Positif (44.3%), Negatif (27.9%), dan Netral (27.8%). Meskipun tidak menggunakan teknik oversampling atau undersampling, kedua model mampu memberikan performa yang seimbang pada ketiga kelas. Hal ini menunjukkan bahwa ukuran dataset (3.874 samples) sudah cukup untuk setiap kelas, dengan kelas minoritas (Netral) memiliki lebih dari 1.000 samples yang cukup untuk pembelajaran yang baik.

E. Kelebihan dan Kekurangan Masing-Masing Model

- Kelebihan BiLSTM:

- Dapat menangkap konteks jangka panjang dalam teks melalui mekanisme LSTM, berguna untuk memahami komentar yang kompleks.
- Arsitektur bidirectional memungkinkan model memahami konteks dari kedua arah.
- Akurasi sedikit lebih tinggi (68.38%).
- Lebih baik dalam mendeteksi sentimen netral berdasarkan recall yang lebih tinggi (0.71).
- Mampu menangani variasi bahasa informal dan slang dengan baik melalui embedding layer.

- Kekurangan BiLSTM:

- Waktu training sangat lama (829.58 detik), tidak efisien untuk deployment cepat atau retraining berkala.
- Membutuhkan resource komputasi yang lebih besar (GPU recommended), meningkatkan biaya operasional.

- Kompleksitas model lebih tinggi, memerlukan tuning hyperparameter yang teliti.
- Lebih rentan terhadap overfitting meskipun sudah menggunakan dropout dan early stopping.
- Tidak memberikan peningkatan akurasi yang signifikan untuk komentar pendek (rata-rata 84 karakter).

- **Kelebihan XGBoost:**

- Sangat cepat dalam training (11.96 detik), ideal untuk iterasi cepat dan prototyping.
- Efisien secara komputasi, dapat berjalan dengan baik di CPU tanpa perlu GPU.
- Robust terhadap overfitting dengan regularisasi bawaan.
- Akurasi yang kompetitif (68.04%), hanya selisih 0.34% dari BiLSTM.
- Lebih mudah untuk di-deploy dan di-maintain di production environment.
- TF-IDF vectorization efektif untuk komentar pendek yang padat informasi.
- Cocok untuk monitoring media sosial real-time yang memerlukan update model berkala.

- **Kekurangan XGBoost:**

- Tidak dapat menangkap konteks urutan kata sebaik LSTM karena menggunakan TF-IDF bag-of-words.
- Akurasi sedikit lebih rendah (68.04%).
- Kurang efektif untuk teks dengan konteks yang sangat kompleks atau panjang.

- Bergantung pada kualitas feature engineering (TF-IDF) yang mungkin tidak optimal untuk semua kasus.
- Recall pada kelas Netral sedikit lebih rendah (0.70 vs 0.71).

4.4.5 Pengujian Prediksi Manual

Untuk memvalidasi performa model secara kualitatif, dilakukan pengujian dengan beberapa kalimat contoh yang mencerminkan karakteristik komentar politik di TikTok. Berikut adalah hasil prediksi dari kedua model:

Test Case 1 (Komentar Positif):

Teks: "Kebijakan ini sangat bagus untuk ekonomi negara kita"

XGBoost: Positif (Confidence: 97.79%)

BiLSTM: Positif (Confidence: 99.94%)

Analisis: Kedua model berhasil mengidentifikasi sentimen positif dengan confidence yang sangat tinggi. Kata-kata seperti "bagus" dan konteks "untuk ekonomi negara" memberikan signal positif yang jelas.

Test Case 2 (Komentar dengan Negasi):

Teks: "Saya tidak setuju dengan keputusan pemerintah ini"

XGBoost: Positif (Confidence: 83.62%) - SALAH

BiLSTM: Positif (Confidence: 84.90%) - SALAH

Analisis: Kedua model salah mengklasifikasikan kalimat ini sebagai positif, padahal seharusnya negatif. Hal ini menunjukkan bahwa model masih memiliki keterbatasan dalam memahami konteks negasi dan frasa "tidak

setuju" yang mengindikasikan sentimen negatif. Preprocessing mungkin menghapus kata "tidak" sebagai stopwords, atau model tidak cukup belajar pola negasi dari data training. Ini adalah area yang perlu diperbaiki di penelitian selanjutnya, misalnya dengan membuat daftar stopwords kustom yang mempertahankan kata negasi, atau menggunakan teknik sentiment-aware preprocessing.

Test Case 3 (Komentar Netral):

Teks: "Pemerintah mengumumkan kebijakan baru hari ini"

XGBoost: Netral (Confidence: 56.19%)

BiLSTM: Netral (Confidence: 58.16%)

Analisis: Kedua model berhasil mengidentifikasi sentimen netral dengan benar, meskipun confidence tidak terlalu tinggi (56-58%). Hal ini wajar karena kalimat memang bersifat informatif tanpa ekspresi sentimen yang jelas. BiLSTM menunjukkan confidence sedikit lebih tinggi, konsisten dengan recall yang lebih baik pada kelas Netral.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. XGBoost dan BiLSTM menunjukkan akurasi serupa (68.04% vs 68.38%) serta macro F1-score (68.24% vs 68.00%) pada klasifikasi sentimen komentar TikTok terkait kebijakan penempatan dana pemerintah di bank.
2. XGBoost jauh lebih efisien dengan waktu training 11.96 detik, sementara BiLSTM membutuhkan 829.58 detik (69x lebih lambat).
3. XGBoost terbukti lebih efektif secara keseluruhan karena unggul pada efisiensi waktu komputasi dan kemampuan generalisasi yang seimbang, sehingga cocok untuk monitoring sentimen publik real-time.
4. BiLSTM menunjukkan performa superior pada recall kelas Netral dengan skor 0.71 sedangkan pada XGBoost memiliki skor 0.70, mencerminkan kemampuannya dalam menangkap konteks kalimat yang ambigu. Namun, keunggulan marginal ini (+1.4%) tidak sebanding dengan perbedaan waktu training yang mencapai 69 kali lipat (829.58 detik vs 11.96 detik). Dengan mempertimbangkan trade-off antara peningkatan performa minimal dan biaya komputasi yang signifikan, XGBoost lebih direkomendasikan untuk implementasi production, terutama dalam sistem monitoring media sosial yang memerlukan update model secara berkala.

5.2 Saran

Berdasarkan hasil penelitian dan keterbatasan yang ditemukan, berikut adalah beberapa saran untuk penelitian selanjutnya:

1. Mencakup platform media sosial lain seperti Twitter/X, Facebook, Instagram, atau forum diskusi online.
2. Menambahkan algoritma lain selain dari algoritma XGBoost dan BiLSTM, untuk mengetahui bagaimana hasil perbandingan jika dibandingkan dengan algoritma lainnya.
3. Mengumpulkan data yang lebih lama dari 1 bulan dan lebih luas lagi, guna mencakup seluruh pendapat masyarakat pada kebijakan tersebut.
4. Mencakup dana tambahan lain yang mungkin dikeluarkan oleh pemerintah pada periode yang akan mendatang.
5. Melakukan filtering akun buzzer atau bot, agar dapat memverifikasi keaslian akun pengguna sehingga hasil mencerminkan seluruh sentimen yang tersedia.
6. Menambahkan fitur komentar menggunakan bahasa daerah atau bahasa asing agar seluruh data dapat di pakai tanpa banyak membuang sentimen.
7. Menambahkan cakupan penelitian pada analisis emosi mendalam (seperti marah, sedih, gembira) atau deteksi sarkasme yang kompleks secara terpisah.

DAFTAR PUSTAKA

- Ahmadian, H., Abidin, T. F., Riza, H., & Muchtar, K. (2024). Hybrid Models for Emotion Classification and Sentiment Analysis in Indonesian Language. *Applied Computational Intelligence and Soft Computing*, 2024(1). <https://doi.org/10.1155/2024/2826773>
- Alabdulkarim, N. A., Haq, M. A., & Gyani, J. (2024). Exploring Sentiment Analysis on Social Media Texts. *Engineering, Technology & Applied Science Research*, 14(3), 14442–14450. <https://doi.org/10.48084/etasr.7238>
- Alhassan, A. M., & Altmami, N. I. (2025). IV3TM: Inception V3 enabled bidirectional long short-term memory network for brain tumor classification. *PLOS One*, 20(10), e0335397. <https://doi.org/10.1371/journal.pone.0335397>
- Alqaryouti, O., Siyam, N., Abdel Monem, A., & Shaalan, K. (2024). Aspect-based sentiment analysis using smart government review data. *Applied Computing and Informatics*, 20(1/2), 142–161. <https://doi.org/10.1016/j.aci.2019.11.003>
- Chandrasekaran, G., Dhanasekaran, S., Moorthy, C., & Arul Oli, A. (2025). Multimodal sentiment analysis leveraging the strength of deep neural networks enhanced by the XGBoost classifier. *Computer Methods in Biomechanics and Biomedical Engineering*, 28(6), 777–799. <https://doi.org/10.1080/10255842.2024.2313066>
- Dina Wulan Yekti rahayu, Khothibul Umam, & Maya Rini Handayani. (2025). Performance of Machine Learning Algorithms on Imbalanced Sentiment Datasets Without Balancing Techniques. *Journal of Applied Informatics and Computing*, 9(3), 998–1005. <https://doi.org/10.30871/jaic.v9i3.9584>
- INA Solutions. (2024). *The future of public policy: Using natural language processing solutions for sentiment analysis*. The Future of Public Policy: Using Natural Language Processing Solutions for Sentiment Analysis. <https://ina-solutions.com/resources/2024/11/26/the-future-of-public-policy-using-natural-language-processing-solutions-for-sentiment-analysis/>
- Khadija, M. A., Jayanti, I. S. D., & Nimah, F. U. (2024). Towards Smart City: Aspect Based Sentiment Analysis of Indonesian Public Aspiration Complaints Data Using Machine Learning. *2024 7th International Conference on Informatics and Computational Sciences (ICICoS)*, 215–220. <https://doi.org/10.1109/ICICoS62600.2024.10636859>
- Lin, C.-H., & Nuha, U. (2023). Sentiment analysis of Indonesian datasets based on a hybrid deep-learning strategy. *Journal of Big Data*, 10(1), 88. <https://doi.org/10.1186/s40537-023-00782-9>
- Nguyen, H.-H. (2024). Enhancing Sentiment Analysis on Social Media Data with Advanced Deep Learning Techniques. *International Journal of Advanced Computer Science and Applications*, 15(5). <https://doi.org/10.14569/IJACSA.2024.0150598>

- Nsaif, A. A., & Abd, D. H. (2022). *Sentiment Analysis of Political Post Classification Based on XGBoost* (pp. 177–188). https://doi.org/10.1007/978-981-19-0604-6_16
- Rahman, M. M., Shiplu, A. I., Watanobe, Y., & Alam, M. A. (2025). RoBERTa-BiLSTM: A Context-Aware Hybrid Model for Sentiment Analysis. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 9(6), 3788–3805. <https://doi.org/10.1109/TETCI.2025.3572150>
- Rifqi Yafik, & Mulkan Azhari. (2025). Analisis Perbandingan Metode LSTM Dan BiLSTM Untuk Prediksi Harga Saham Menggunakan Alpha Vantage. *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, 4(3), 1542–1551. <https://doi.org/10.62712/juktisi.v4i3.650>
- Romadhony, A., Al Faraby, S., Rismala, R., Wisesty, U. N., & Arifianto, A. (2024). Sentiment Analysis on a Large Indonesian Product Review Dataset. *Journal of Information Systems Engineering and Business Intelligence*, 10(1), 167–178. <https://doi.org/10.20473/jisebi.10.1.167-178>
- Russell, S. N., Rao-Graham, L., & McNaughton, M. (2024). Mining social media data to inform public health policies: a sentiment analysis case study. *Revista Panamericana de Salud Pública*, 48, 1. <https://doi.org/10.26633/RPSP.2024.79>
- Setiawan, M. J., & Nastiti, V. R. S. (2024). DANA App Sentiment Analysis: Comparison of XGBoost, SVM, and Extra Trees. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 13(3), 337–345. <https://doi.org/10.32736/sisfokom.v13i3.2239>
- Talaat, A. S. (2023). Sentiment analysis classification system using hybrid BERT models. *Journal of Big Data*, 10(1), 110. <https://doi.org/10.1186/s40537-023-00781-w>
- Verma, S. (2022). Sentiment analysis of public services for smart society: Literature review and future research directions. *Government Information Quarterly*, 39(3), 101708. <https://doi.org/10.1016/j.giq.2022.101708>
- Xu, G., Chen, Z., & Zhang, Z. (2025). Aspect category sentiment analysis based on pre-trained BiLSTM and syntax-aware graph attention network. *Scientific Reports*, 15(1), 3333. <https://doi.org/10.1038/s41598-025-86009-8>
- Yulistiani, Y., & Styawati, S. (2024). Analisis Sentimen Terhadap Calon Presiden Indonesia 2024 dengan Metode Extreme Gradient Boosting (XGBOOST). *Jurnal Informatika: Jurnal Pengembangan IT*, 9(3), 322–328. <https://doi.org/10.30591/jpit.v9i3.6127>
- Zhao, J., Liu, H., Wang, Y., Zhang, W., Zhang, X., Li, B., Sun, T., Qi, Y., & Zhang, S. (2024). Sentiment analysis of video danmakus based on MIBE-RoBERTa-BiLSTM. *Scientific Reports*, 14(1), 5827. <https://doi.org/10.1038/s41598-024-56518-z>

LAMPIRAN

1. SK Penetapan Dosen Pembimbing

 MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
IBROU Terakreditasi Unggul Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 179/SK/AN-PT/Kej/P/PT/2024
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20230 Telp. (061) 6622400 - 66224567 Fax. (061) 6623474 - 6631983
Website: www.umsu.ac.id Email: info@umsu.ac.id Instagram: @umsuamedan Facebook: @umsuamedan Twitter: @umsuamedan

PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 987/IL3-AU/UMSU-09/F/2025

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi : Teknologi Informasi
Pada tanggal : 28 Oktober 2025

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama : Wahyu Purnomo
NPM : 2109020059
Semester : IX (Sembilan)
Program studi : Teknologi Informasi
Judul Proposal / Skripsi : Analisis Perbandingan Algoritma XGBoost dan BiLSTM dalam Klasifikasi Sentimen Publik Terhadap Kebijakan Penempatan Dana Pemerintah Di Bank

Dosen Pembimbing : Mhd. Basri, S.Kom.,M.Kom.

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi dinyatakan " BATAL "** bila tidak selesai sebelum Masa Kadalursa tanggal : **28 Oktober 2026**
4. Revisi judul.....

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
Pada Tanggal : 06 Jumadil Awwal 1447 H
28 Oktober 2025M

Dekan

Dr. Al-Khowarizmi, M.Kom.
NIDN : 0127099201



Cc. File



2. Turnitin

Skripsi Wahyu sidang revisi

ORIGINALITY REPORT

16%

SIMILARITY INDEX

12%

INTERNET SOURCES

11%

PUBLICATIONS

9%

STUDENT PAPERS

PRIMARY SOURCES

1

repository.umsu.ac.id

Internet Source

1%

2

Muhammad Oriza Nurfajri, Guntur Budi Herwanto. "Pendekatan Ensemble Learning untuk klasifikasi serangan DDoS", Pseudocode, 2025

Publication

1%

3

Submitted to UNIVERSITAS BUDI LUHUR

Student Paper

1%

4

Submitted to Universitas Muhammadiyah Sumatera Utara

Student Paper

<1%

5

penerbitadm.pubmedia.id

Internet Source

<1%

6

jurnal.polbeng.ac.id

Internet Source

<1%

7

Amelia Mar'atusholihat, Nuk Ghurroh Setyoningrum, Dede Rizal Nursamsi. "Perbandingan Kinerja Algoritma Decision Tree Dan Naïve Bayes Dalam Analisis

<1%

Sentimen Publik Terhadap Kebijakan
Pemerintah Mengenai Batasan Penggunaan
Media Sosial Anak", Informatics and Digital
Expert (INDEX), 2025

Publication

8	Submitted to Universitas Tarumanagara Student Paper	<1 %
9	repository.usni.ac.id Internet Source	<1 %
10	jurnal.polibatam.ac.id Internet Source	<1 %
11	kc.umh.ac.id Internet Source	<1 %
12	repository.upi.edu Internet Source	<1 %
13	Submitted to Universitas Pancasila Student Paper	<1 %
14	text-id.123dok.com Internet Source	<1 %
15	Submitted to Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur Student Paper	<1 %
16	Submitted to Universitas Bengkulu Student Paper	<1 %

Submitted to University of Birmingham

17	Student Paper	<1 %
18	docplayer.info Internet Source	<1 %
19	journal.maranatha.edu Internet Source	<1 %
20	Submitted to Universitas Katolik Musi Charitas Student Paper	<1 %
21	Submitted to UPN Veteran Yogyakarta Student Paper	<1 %
22	repository.ubharajaya.ac.id Internet Source	<1 %
23	digilib.esaunggul.ac.id Internet Source	<1 %
24	e-jurnal.pnl.ac.id Internet Source	<1 %
25	Andreas Rezeki Zai, Bambang Suhardi, Surya Tri Nowo, Rika Rosnelly, Adil Setiawan. "Eksplorasi pada Pemetaan Klasifikasi Radiograf Toraks Penyakit Paru-Paru Menggunakan Convolutional Neural Network (CNN)", Syntax : Journal of Software Engineering, Computer Science and Information Technology, 2026 Publication	<1 %

26	Submitted to Institut Bisnis dan Teknologi Indonesia (INSTIKI) Student Paper	<1%
27	ejournal.stmik-time.ac.id Internet Source	<1%
28	Submitted to Universitas Islam Indonesia Student Paper	<1%
29	Kasih Delayana, Setia Mangiring Marpaung, Syukur Nimei Iman Gulo, Sardo Sipayung. "Analisis Sentimen Publik terhadap Barak Militer Menggunakan Naive Bayes pada Rapid MinerJudul", remik, 2026 Publication	<1%
30	Submitted to Universitas Budi Luhur Student Paper	<1%
31	Submitted to Islington College, Nepal Student Paper	<1%
32	Sukhpreet Kaur, Amanpreet Kaur, Manish Kumar. "Recent Advances in Computational Methods in Science and Technology - Volume 2", CRC Press, 2026 Publication	<1%
33	ejournal.stmik-budidarma.ac.id Internet Source	<1%

34	Achmad Hakim Qoirul Haq, Harminto Mulyo, Adi Sucipto. "Optimasi Kinerja Algoritma K-Nearest Neighbor melalui Metode Random Forest untuk Klasifikasi Penyakit Ginjal", Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2026 <small>Publication</small>	<1%
35	Lediana Fitriani. "Kerangka Pemodelan Iklim Berbasis Kecerdasan Buatan Menggunakan Fusi Data Satelit dan Pembelajaran Mendalam", RIGGS: Journal of Artificial Intelligence and Digital Business, 2026 <small>Publication</small>	<1%
36	Putri Widya Sari Sari, Firmansyah Firmansyah, Abdul Rahman Kadafi Kadafi. "PERBANDINGAN ALGORITMA RANDOM FOREST DAN NAÏVE BAYES DALAM MENGANALISIS SENTIMEN ULASAN PADA PRODUK SKINCARE LOKAL DI MEDIA SOSIAL TIKTOK", Jurnal Informatika dan Teknik Elektro Terapan, 2025 <small>Publication</small>	<1%
37	ejurnal.provisi.ac.id <small>Internet Source</small>	<1%
38	etheses.uin-malang.ac.id <small>Internet Source</small>	<1%
	id.123dok.com	

39	Internet Source	<1%
40	Indah Clara Sari. "INTEGRASI MODEL DEEP LEARNING EFFICIENTNET-B0 UNTUK DETEKSI PENYAKIT DAUN TOMAT PADA APLIKASI SELULER BERBASIS FLUTTER", Djtechno: Jurnal Teknologi Informasi, 2024 Publication	<1%
41	Jihadul Akbar, Hairul Fahmi, Wafiah Murniati. "MULTI LABEL KLASIFIKASI GENRE FILM BERDASARKAN SINOPSIS MENGGUNAKAN METODE LONG SHORT-TERM MEMORY (LSTM)", Jurnal Manajemen Informatika dan Sistem Informasi, 2025 Publication	<1%
42	Sarihot Tondang, Ramadhan Roy Prasetyo, Rafi Fulvian, Yosua Goldstein Sitorus, Giantika Chrisnawati. "Analisis Perbandingan Algoritma K-Nearest Neighbor dan Ensemble Learning dalam Klasifikasi Penyakit Obesitas", RIGGS: Journal of Artificial Intelligence and Digital Business, 2025 Publication	<1%
43	Submitted to andalas Student Paper	<1%
44	api-repo.unisan.ac.id Internet Source	<1%

45	Gudla, Rohan. "Bridging Data Gaps and Forecasting Challenges in Water Quality Monitoring using Deep Learning", University of Central Florida, 2025 <small>Publication</small>	<1%
46	Mingzhe Zou, Duo Fang, Gareth Harrison, Sasa Djokic. "Weather Based Day-Ahead and Week-Ahead Load Forecasting using Deep Recurrent Neural Network", 2019 IEEE 5th International forum on Research and Technology for Society and Industry (RTSI), 2019 <small>Publication</small>	<1%
47	Submitted to Universitas Dian Nuswantoro <small>Student Paper</small>	<1%
48	e-jurnal.stmikbinsa.ac.id <small>Internet Source</small>	<1%
49	Aditya Pranata, Rudiman Rudiman, Naufal Azmi Verdikha. "2024 Presidential Election Quick Count Text Classification on Twitter Using TF-IDF and Naive Bayes Methods", Jurnal Informatika Terpadu, 2024 <small>Publication</small>	<1%
50	Affan Ashabi Zain, Dimas Prasetyo, Ivano Praestesa Destrada, Angga Pradipa Widodo. "Klasifikasi Deteksi Sarkasme pada Berita Headline dengan Arsitektur LSTM", JSITIK:	<1%

Jurnal Sistem Informasi dan Teknologi
Informasi Komputer, 2026

Publication

-
- | | | |
|-------|---|-----|
| 51 | Eti Kurniawati, Ade Irma Purnamasari, Irfan Ali, Rudi Kurniawan, Odi Nurdiawan. "ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI FLO DI GOOGLE PLAY STORE DENGAN MENGGUNAKAN ALGORITMA NAIVE BAYES", Jurnal Informatika dan Teknik Elektro Terapan, 2026 | <1% |
| <hr/> | | |
| 52 | Submitted to Telkom University
Student Paper | <1% |
| <hr/> | | |
| 53 | Submitted to Universitas Muslim Indonesia
Student Paper | <1% |
| <hr/> | | |
| 54 | digilib.unimed.ac.id
Internet Source | <1% |
| <hr/> | | |
| 55 | repository.unas.ac.id
Internet Source | <1% |
| <hr/> | | |
| 56 | Muhamad Ali Zaenal Abidin. "Evaluasi Sentimen Ulasan Pengguna CGV Cinemas Indonesia Menggunakan Metode Naive Bayes dan Support Vector Machine", Indonesian Journal on Software Engineering (IJSE), 2025 | <1% |
| <hr/> | | |

57	Nur Sakina, Farid Wajidi, Muh. Rafli Rasyid. "Evaluasi Algoritma KNN dan Naive Bayes untuk Analisis Sentimen Kebijakan Program Makan Bergizi Gratis", Jambura Journal of Informatics, 2025 Publication	<1%
58	Submitted to RMIT University Student Paper	<1%
59	Submitted to The Kyoto College of Graduate Studies for Informatics Student Paper	<1%
60	Submitted to Universitas Katolik Widya Mandala Surabaya Student Paper	<1%
61	Submitted to Universitas Sumatera Utara Student Paper	<1%
62	naii.co Internet Source	<1%
63	Submitted to Admin Turnitin UISI Student Paper	<1%
64	Gayus Gregorius Ferdinand Djema, Ozzi Suria. "Public Sentiment Analysis of Danantara Policy through Social Media X Using SVM and Random Forest", INOVTEK Polbeng - Seri Informatika, 2025 Publication	<1%

65	Ichsani Mursidah, Remi Sanjaya, Bambang Yulianto, Dhian Sweetania, Puji Sularsih. "Klasifikasi Sentimen Google Play Store Aplikasi ChatGPT Berbahasa Indonesia Berbasis IndoBERT", Jurnal Minfo Polgan, 2025 Publication	<1%
66	Submitted to University of Surrey Student Paper	<1%
67	hub.kirim.ai Internet Source	<1%
68	Doly Ilham Saputra Huta Julu, Dewi Nurdiyah. "KLASIFIKASI SAMPAH ORGANIK DAN NON ORGANIK MENGGUNAKAN TRANSFER LEARNING", Jurnal Transformatika, 2025 Publication	<1%
69	Fadel Najmi Adliansyah, Holilah Holilah, Nanang Krisdianto. "Sistem Absensi Berbasis Face Recognition dengan Model Inception-Resnet", RIGGS: Journal of Artificial Intelligence and Digital Business, 2025 Publication	<1%
70	ejurnal.lkpkaryaprima.id Internet Source	<1%
71	itrev.kemenkeu.go.id Internet Source	<1%

72	journal.sinov.id Internet Source	<1%
73	repositori.uma.ac.id Internet Source	<1%
74	Agustin Agustin, Junadhi Junadhi, Fransiskus Zoromi, Parlindungan Kudadiri. "Analisis Sentimen Kesehatan Mental Pemuda di Media Sosial Menggunakan Deep Learning", DEVICE : JOURNAL OF INFORMATION SYSTEM, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY, 2025 Publication	<1%
75	Maulana Ichsana Ichsana. "IMPLEMENTASI MACHINE LEARNING UNTUK DETEKSI PENYAKIT PADA KUCING MENGGUNAKAN RANDOM FOREST", Jurnal Informatika dan Teknik Elektro Terapan, 2025 Publication	<1%
76	Mohammad Aman Ullah. "BCLSA: Advancing Bangla Sentiment Analysis with Concept-Level Reasoning and Efficiency", Machine Learning with Applications, 2025 Publication	<1%
77	disafa-casafa.blogspot.com Internet Source	<1%
78	e-journal.hamzanwadi.ac.id Internet Source	

		<1 %
79	j-ptiik.ub.ac.id Internet Source	<1 %
80	journal.ilmudata.co.id Internet Source	<1 %
81	journal.widyadharma.ac.id Internet Source	<1 %
82	jurnalummi.agungprasetyo.net Internet Source	<1 %
83	publikasi.bigdatascience.id Internet Source	<1 %
84	Submitted to Universitas Muhammadiyah Makassar Student Paper	<1 %
85	ejournal2.unud.ac.id Internet Source	<1 %
86	jurnal.stmikasia.ac.id Internet Source	<1 %
87	repositori.usu.ac.id:8080 Internet Source	<1 %
88	repository.unjaya.ac.id Internet Source	<1 %

- 89 Agung RM Alam, M Suhada WD, Agus F Nduru, Rika Rosnelly, Adil Setiawan. "Klasifikasi Jenis Tumor Otak pada Citra MRI Menggunakan Convolutional Neural Network (CNN)", remik, 2025
Publication <1%
-
- 90 Jendraja Husein Kotan, Andreas Andreas, Silvi Mutia, Hafiz Irsyad. "KLASIFIKASI OPINI MASYARAKAT TERHADAP VIDEO DOKUMENTER DIRTY VOTE DENGAN ALGORITMA KNN(K-NEAREST NEIGHBOR) DAN NAÏVE BAYES", Syntax : Journal of Software Engineering, Computer Science and Information Technology, 2024
Publication <1%
-
- 91 Mitranikasih Laia, Ayuliana, Wasiran, Muhammad Lukman Hakim, Dicky Suryadi. "Analisis Big Data untuk Deteksi Hoaks dan Disinformasi di Platform Berita Online", Jurnal JTIC (Jurnal Teknologi Informasi dan Komunikasi), 2025
Publication <1%
-
- 92 Sahat Monang. "IMPLEMENTASI LONG SHORT- TERM MEMORY UNTUK MEMPREDIKSI HARGA SHIBA INU BERBASIS WEBSITE MENGGUNAKAN PLATFORM PYTHON", <1%

PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer, 2025

Publication

93	digilib.uin-suka.ac.id Internet Source	<1 %
94	dirros.openscience.si Internet Source	<1 %
95	dspace.uui.ac.id Internet Source	<1 %
96	e-journal.stmiklombok.ac.id Internet Source	<1 %
97	ejournal.unsrat.ac.id Internet Source	<1 %
98	ejurnal.seminar-id.com Internet Source	<1 %
99	eprints.umg.ac.id Internet Source	<1 %
100	jtera.polteksmi.ac.id Internet Source	<1 %
101	repository.unissula.ac.id Internet Source	<1 %
102	repository.unsulbar.ac.id Internet Source	<1 %
103	research.pps.dinus.ac.id Internet Source	<1 %

		<1 %
104	www.scribd.com Internet Source	<1 %
105	Fadila Ananda Kartika Hidayat, Neny Sulistianingsih, Rifqi Hammad. "Perbandingan Algoritma XGBoost dan Random Forest dalam Klasifikasi Surat Masuk Pemerintahan", RIGGS: Journal of Artificial Intelligence and Digital Business, 2026 Publication	<1 %
106	Danang Arbian Sulisty, Erik Setiadi. "Analisis Sentimen Kebijakan Makan Bergizi Gratis Menggunakan IndoBERT dan Machine Learning", JURNAL FASILKOM, 2025 Publication	<1 %
107	Raihanda Luthfiansyah, Budi Wasito. "Penerapan Teknik Deep Learning (Long Short Term Memory) dan Pendekatan Klasik (Regresi Linier) dalam Prediksi Pergerakan Saham BRI", Jurnal Informatika dan Bisnis, 2023 Publication	<1 %
108	Siti Aulia Rahmadhani, Lia Dwi Rusanti, Harun Al Rosyid. "Klasifikasi Sentimen Komentar Youtube Demonstrasi DPR RI Menggunakan Support Vector Machine", Arcitech: Journal of	<1 %

Computer Science and Artificial Intelligence, 2025

Publication

109

zombiedoc.com
Internet Source

<1%

Exclude quotes: On

Exclude matches: Off

Exclude bibliography: On