

# Analisis Sentimen Aspek Ulasan Pada Aplikasi Tinder

Galih Isa Yudhistira<sup>1</sup>, xxxx<sup>2</sup>

<sup>1</sup>Faculty of Engineering and Informatics Information Systems Study, Multimedia Nusantara University, Banten 15810, Indonesia

<sup>2</sup> Faculty of Engineering and Informatics Information Systems Study, Multimedia Nusantara University, Banten 15810, Indonesia

Corresponding author: Galih Isa Yudhistira (e-mail: [galih.yudhistira@student.umn.ac.id](mailto:galih.yudhistira@student.umn.ac.id)).

**ABSTRACT** Perkembangan pesat teknologi dan internet di Indonesia telah mengubah cara masyarakat berinteraksi, termasuk dalam penggunaan aplikasi mobile seperti Tinder, yang pada tahun 2020 menguasai sekitar 57,6% pangsa pasar aplikasi kencan di Indonesia dengan lebih dari 100 juta unduhan dan jutaan ulasan pengguna di Google Play Store. Penelitian ini mengkaji performa tiga model pembelajaran mendalam, yaitu Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), dan Convolutional Neural Network (CNN), dalam menganalisis sentimen dari 5000 ulasan teks terbaru yang diproses melalui tahapan Knowledge Discovery In Databases (KDD) meliputi seleksi, pra-pemrosesan, transformasi, data mining, dan evaluasi. Hasil pengujian menunjukkan bahwa model CNN mencapai akurasi tertinggi sebesar 98% pada pengujian dasar dengan penerapan teknik Class Weight untuk mengatasi ketidakseimbangan kelas, sementara setelah tuning hyperparameter, model LSTM meningkat performanya hingga 98,2%, dan BiLSTM memberikan hasil sangat kompetitif dengan akurasi mencapai 98,7%. Penerapan Class Weight terbukti efektif dalam meningkatkan sensitivitas model terhadap kelas minoritas, sehingga ketiga model tersebut menunjukkan potensi besar dalam analisis sentimen ulasan aplikasi Tinder di Indonesia.

**INDEX TERMS** Analisis Sentimen, Google Play, Aplikasi Kencan Tinder, Convolutional Neural Network, Bidirectional Long Short-Term Memory, Long Short-Term Memory.

## I. INTRODUCTION

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

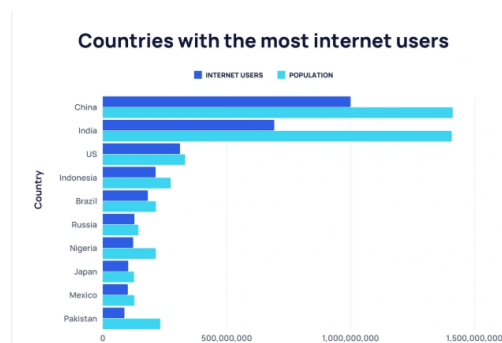


Figure 1 Negara dengan Jumlah Pengguna Internet Tertinggi[3]

Aplikasi kencan menjadi bukti yang terpengaruh oleh perkembangan kemajuan teknologi internet. Tinder merupakan aplikasi kencan paling banyak peminatnya. Berdasarkan website databoks.id periode tahun 2023 terdapat visualisasi yang menunjukkan Tinder mencapai deret pertama dari 10 aplikasi kencan yang paling banyak diunduh secara global. Aplikasi kencan Tinder yang berasal dari

Amerika Serikat (AS) ini telah mencapai unduhan sampai 55 juta kali pada tahun tersebut. Tinder, sebagai bagian dari Match Group yang berbasis di Dallas, AS, memimpin pasar aplikasi kencan online bersama aplikasi lain di bawah naungannya seperti Hinge, Match, OkCupid, Meetic, dan Pairs. Di posisi kedua, LitMatch mencatat 34 juta unduhan global pada 2023 dengan peningkatan signifikan di Asia Tenggara meskipun tanpa rincian angka. Sementara itu, Bumble berada di peringkat ketiga dengan 30 juta unduhan dan juga mengelola aplikasi lain seperti Badoo, Fruitz, dan Official. Badoo sendiri masuk dalam daftar aplikasi kencan terpopuler dengan total 23 juta unduhan pada 2023.

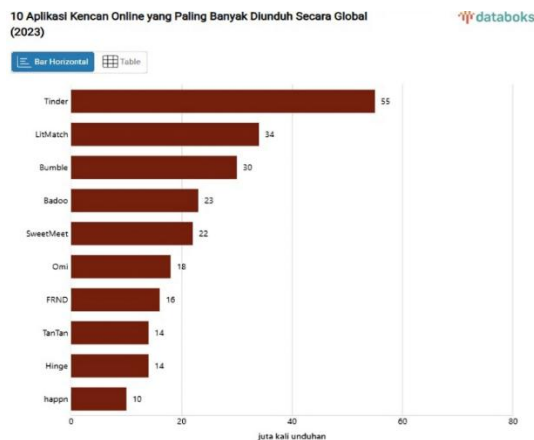


Figure 2 Aplikasi Kencan Online yang Paling Banyak Diunduh Secara Global (2023)[4]

Perkembangan aplikasi kencan Tinder di Indonesia juga menjadi aplikasi paling laris diunduh di Indonesia. Menurut website databoks.id periode tahun 2024 terdapat visualisasi yang menunjukkan bahwa penggunaan aplikasi kencan daring di Indonesia cukup bervariasi. Dalam survei tersebut, sekitar 38% dari responden mengungkapkan bahwa mereka menggunakan aplikasi kencan daring Tinder, menjadikannya sebagai aplikasi yang paling banyak digunakan dibandingkan dengan aplikasi serupa lainnya[5]. Selain itu, aplikasi kencan daring TanTan, asal Tiongkok, juga cukup populer dengan penggunaan mencapai 33%, diikuti oleh Bumble dengan 17%, Omi dengan 13%, dan Datingcom dengan 12%. Terdapat juga pengguna yang memilih aplikasi seperti Badoo (10%), Taaruf (7%), Ok cupid (7%), Muslima.com (5%). Data ini memberikan gambaran yang jelas tentang preferensi pengguna aplikasi kencan daring di Indonesia pada waktu survei tersebut.

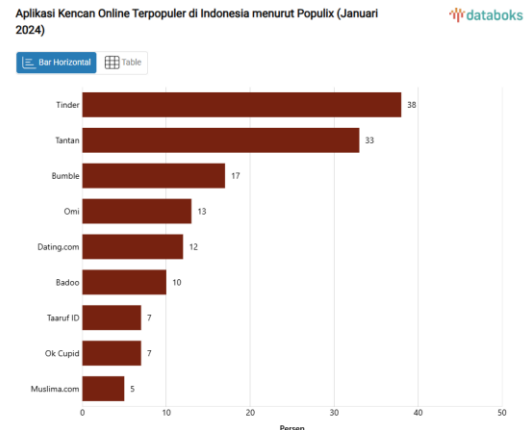


Figure 3 Tinder, Aplikasi Kencan Paling Banyak di Indonesia periode tahun 2024[6]

Pandangan mengenai aplikasi kencan dalam era kemajuan teknologi bervariasi[7]. Beberapa melihatnya sebagai inovasi positif yang memudahkan mencari pasangan dengan memberikan peluang lebih besar untuk bertemu orang baru dan memperluas jaringan sosial tanpa harus berkenalan secara langsung atau konvensional. Selain hal positif, disamping itu ada juga hal negatif yang ditimbulkan dari dating apps ini. Perlu diwaspadai karena juga dapat membawa dampak negatif. Salah satu, dampak negatif yang sering muncul merupakan gangguan kesehatan mental, di mana pengguna mungkin mengalami tekanan, kecanduan, atau perasaan cemas karena ekspektasi yang tinggi[8]. Hal ini tercermin dari banyaknya ulasan negatif yang ditemukan di Google Play Store, yang menunjukkan adanya dampak psikologis yang dialami sebagian pengguna.

Untuk mengevaluasi pengalaman pengguna, salah satu metode yang efektif adalah analisis sentimen. Dengan menganalisis ulasan pengguna di platform seperti Google Play Store, kita dapat mengidentifikasi opini dan perasaan pengguna terhadap aplikasi tersebut. Pada bulan Juli 2013, Tinder mulai tersedia untuk pengguna ponsel Android melalui unduhan dari platform Google Play Store. Permintaan untuk aplikasi ini sangat tinggi, mencapai lebih dari 800.000 orang[9]. Menurut informasi yang dapat ditemukan pada Google Play Store pada tanggal 6 September 2023, aplikasi Tinder telah berhasil mencapai lebih dari 100 juta unduhan dengan rating keseluruhan sebesar 4,4. Di platform tersebut, terdapat sekitar 6,17 juta ulasan dari pengguna, yang terdiri dari dua komponen utama: nilai rating numerik yang mencerminkan penilaian keseluruhan pengalaman pengguna, dan komentar teks yang memberikan wawasan yang lebih mendalam.

Dalam melihat ulasan aplikasi di Google Play Store yang seringkali cukup banyak, kadang-kadang memerlukan waktu yang cukup lama untuk membaca

semua ulasan secara lengkap. Namun, terkadang kita hanya memiliki sedikit waktu untuk membaca sebagian ulasan saja. Perlunya suatu platform atau alat yang dapat secara otomatis dan mudah mengidentifikasi apakah ulasan tersebut. Analisis ini bertujuan untuk memahami sentimen yang terkandung dalam teks ulasan yang bersifat positif, atau negatif sehingga dapat memberi gambaran mengenai kualitas aplikasi dan pengalaman pengguna secara umum. Hal ini berguna agar kita dapat dengan cepat memahami sentimen atau tanggapan pengguna terhadap aplikasi tersebut tanpa harus membaca semua ulasan secara rinci.

Analisis sentimen merupakan metode pengolahan data yang digunakan untuk mengidentifikasi dan memahami asumsi, opini, dan unsur subjektivitas dalam sebuah teks[10]. Secara keseluruhan Teknik ini memungkinkan kita untuk menganalisis dan menginterpretasikan berbagai elemen subjektif yang terkandung dalam teks secara komputasi[10]. Tugas utama dalam analisis sentimen merupakan melakukan klasifikasi terhadap polaritas teks yang terdapat dalam dokumen, kalimat, atau pendapat[11]. Polaritas ini merujuk pada apakah teks tersebut memiliki unsur positif, negatif atau netral[12]. Oleh karena itu, perlu dilakukan analisis terhadap opini-opini ini dalam konteks penelitian ini, sehingga dapat digunakan sebagai ukuran untuk menilai kualitas layanan aplikasi kencan online menurut perspektif pengguna.

Perkembangan sentimen analisis dari waktu ke waktu memperluas pemahaman kita serta memberikan dasar untuk penelitian selanjutnya. Penelitian terdahulu mengenai sentimen analisis turut mencegah kita melakukan kesalahan dan pengulangan yang sama. Dalam penelitian kita sendiri kita juga belajar dari apa yang telah dicapai pada masa lalu dan apa yang telah berhasil atau tidak berhasil. Metode LSTM (Long Short-Term Memory) pernah diuji coba dalam analisis sentimen terhadap tweet berbahasa Nepal yang terkait dengan COVID-19 yang menghasilkan tingkat akurasi mencapai 76,04%[13]. Metode LSTM (Long Short-Term Memory) selanjutnya diuji coba dalam analisis sentiment terhadap 1.500 ulasan aplikasi Instagram, dengan mengkategorikan sentimen pengguna sebagai positif atau negatif. Model LSTM mencapai akurasi 77,77%[14]. Metode LSTM (Long Short-Term Memory) selanjutnya diuji coba dalam analisis sentiment terhadap ulasan aplikasi Traveloka, mengklasifikasikan sentimen pengguna ke dalam beberapa kategori seperti sangat tidak puas, tidak puas, netral, puas, dan sangat puas. Model LSTM (Long Short-Term Memory) mencapai akurasi 80%, yang secara efektif membedakan tingkat kepuasan pengguna[15]. Metode LSTM (Long Short-Term Memory) selanjutnya diuji coba dalam analisis

sentiment terhadap 3.565 ulasan pengguna aplikasi Shopee dari Google Play Store, mengkategorikannya ke dalam tiga kelas sentimen: positif, dan negatif. Model ini mencapai akurasi keseluruhan 83%[16].

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

## II. Related work

Sejumlah penelitian sebelumnya telah memberikan sumbangsih berarti dalam mengeksplorasi analisis sentimen terhadap layanan di aplikasi Google Playstore, mencakup pendekatan klasifikasi yang digunakan serta strategi dalam mengatasi permasalahan seperti data yang tidak seimbang dan proses pelabelan.

Salah satu studi menerapkan algoritma *Bidirectional Long Short Term Memory* (BiLSTM) untuk memproses data secara dua arah untuk menangkap konteks yang lebih kaya dalam analisis sentimen. BiLSTM menggabungkan informasi dari masa lalu dan masa depan dalam urutan data, sehingga meningkatkan pemahaman terhadap ketergantungan urutan kata. Model ini dilatih dan diuji menggunakan dataset yang relevan untuk tugas klasifikasi sentimen, dengan fokus pada peningkatan performa. Hasil evaluasi menunjukkan bahwa model mencapai akurasi sebesar 99,14%, recall 99,1%, presisi 99,1%, dan F1-score 99,12%. Angka-angka ini mengindikasikan bahwa model memiliki keseimbangan yang sangat baik antara presisi dan recall, menghasilkan performa yang hampir sempurna.

Dengan demikian, BiLSTM sangat efektif dalam mengenali pola sentimen dan cocok digunakan dalam berbagai aplikasi analisis opini dan media sosial[17].

Penelitian menggunakan metode lain Long Short-Term Memory (LSTM), sebuah metode deep learning, yang dikombinasikan dengan Word2Vec, sebuah teknik penyematan kata, untuk analisis sentimen komentar Instagram. LSTM digunakan untuk menangkap ketergantungan berurutan dalam data, sementara Word2Vec mengubah kata-kata menjadi representasi vektor yang mampu menangkap makna semantik komentar dengan lebih baik. Model ini dilatih dan diuji menggunakan dataset komentar Instagram yang relevan untuk tugas klasifikasi sentimen. Hasil evaluasi menunjukkan bahwa model mencapai akurasi sebesar 91%, presisi 92,70%, recall 89%, dan F-measure 90,81%. Angka-angka ini mengindikasikan bahwa kombinasi LSTM dan Word2Vec sangat efektif dalam mengenali pola sentimen dengan akurasi dan presisi yang tinggi. Dengan demikian, pendekatan ini sangat cocok digunakan untuk analisis sentimen pada komentar media sosial seperti Instagram[18]

Arsitektur model algoritma Convolutional Neural Network (CNN) untuk analisis sentimen pada kolom komentar YouTube terkait bacapres 2024 di program Mata Najwa. CNN digunakan untuk mengekstrak fitur penting dari data teks secara otomatis melalui lapisan konvolusi yang mampu menangkap pola lokal dalam komentar. Model ini dilatih dan diuji menggunakan dataset komentar yang relevan untuk mengklasifikasikan sentimen dengan akurasi tinggi. Hasil evaluasi menunjukkan peningkatan performa yang signifikan dengan skor F1 tertinggi mencapai 94% untuk kategori data air dan taman hiburan, 93% untuk data alami, dan 94% secara keseluruhan. Angka-angka ini mengindikasikan bahwa CNN sangat efektif dalam mengenali pola sentimen yang kompleks dalam data teks komentar YouTube. Dengan demikian, CNN sangat cocok digunakan dalam aplikasi analisis opini publik di media sosial dan platform digital lainnya[19].

Pada analisis sentimen pada bagian pra pemrosesan merupakan salah satu tahapan yang sangat penting. Pada suatu penelitian menggunakan beberapa teknik pra-pemrosesan teks yang penting untuk meningkatkan kualitas analisis sentimen. Pertama, dilakukan tokenisasi untuk mengekstraksi unit-unit kata dari teks mentah. Selanjutnya, penghapusan kata henti diterapkan dengan berbagai metode seperti pendekatan klasik, hukum Zipf, informasi bersama, dan pengambilan sampel acak berbasis istilah untuk menghilangkan kata-kata yang kurang bermakna. Proses stemming juga digunakan dengan metode pemangkasan, statistik, dan metode campuran, contohnya Lovins Stemmer dan Porter Stemmer, untuk mengubah kata ke bentuk dasarnya. Selain itu, algoritma TF/IDF diterapkan untuk memberikan bobot pada istilah berdasarkan frekuensi dan pentingnya dalam dokumen. Kombinasi teknik ini bertujuan untuk mengurangi dimensi ruang istilah, sehingga meningkatkan akurasi penambangan teks serta daya ingat dan efisiensi dalam tugas pemrosesan data. Dengan demikian, pendekatan ini sangat efektif dalam mempersiapkan data teks untuk analisis sentimen yang lebih akurat dan efisien[21].

Dalam penelitian analisis sentimen lainnya, berbagai teknik pra-pemrosesan telah diterapkan, di antaranya adalah tokenisasi yang bertujuan untuk memecah teks mentah menjadi satuan kata. Penghapusan kata-kata umum atau kata henti dilakukan melalui berbagai pendekatan seperti metode konvensional, prinsip hukum Zipf, analisis informasi bersama (mutual information), serta teknik pengambilan sampel acak berbasis istilah, guna menyaring kata-kata yang tidak memberikan kontribusi signifikan terhadap makna teks. Proses stemming juga diterapkan untuk mengembalikan kata ke bentuk dasarnya, menggunakan pendekatan pemangkasan, metode statistik, atau gabungan keduanya, seperti yang digunakan dalam Lovins Stemmer

dan Porter Stemmer. Selain itu, algoritma TF-IDF dimanfaatkan untuk menentukan bobot setiap kata berdasarkan frekuensinya dalam dokumen serta relevansinya. Seluruh rangkaian metode ini bertujuan untuk mereduksi kompleksitas data dan ruang istilah, sehingga dapat meningkatkan akurasi analisis teks serta efisiensi dalam pemrosesan, dan pada akhirnya menghasilkan data yang lebih siap untuk analisis sentimen secara menyeluruh[22].

Penggunaan metode deep learning yang baik jika menggunakan optimizer yang baik seperti Adam (Adaptive Moment Estimation) yang sangat penting untuk mempercepat dan menstabilkan proses pelatihan model. Peran utamanya adalah mengatur dan memperbarui bobot model secara efisien berdasarkan gradien dari fungsi kerugian (loss function), dengan tujuan mempercepat konvergensi menuju solusi optimal. Dalam penelitian ini, penggunaan optimizer Adam membuahkan hasil yang sangat baik pada metode deep learning, khususnya dalam pelatihan model Vision Transformer untuk klasifikasi penyakit paru-paru berbasis citra X-ray. Adam menunjukkan performa unggul dengan menghasilkan akurasi validasi tertinggi sebesar 93,9%, serta nilai presisi, recall, dan F1-score yang konsisten lebih baik dibandingkan optimizer lainnya seperti AdamW, SGD, dan LAMB. Selain itu, Adam juga mencatat waktu pelatihan tercepat per epoch, yaitu rata-rata 652,71 detik, yang mencerminkan efisiensi komputasi tinggi. Kombinasi antara kecepatan dan akurasi ini menegaskan bahwa Adam merupakan pilihan yang sangat efektif dalam penerapan deep learning untuk klasifikasi citra medis[23].

### III. METHODOLOGY

Tahapan implementasi dalam penelitian ini menggunakan model *Knowledge Discovery In Databases* (KDD), yang juga dikenal berperan dalam proses KDD. KDD, dengan pendekatan sistematisnya untuk menangani kompleksitas data, memfasilitasi fase prapemrosesan yang menyeluruh. Proses ini mencakup 5 tahap *Data Selection*, *PreProcessing*, *Transformation*, *Data mining*, *Evaluation*[24].

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

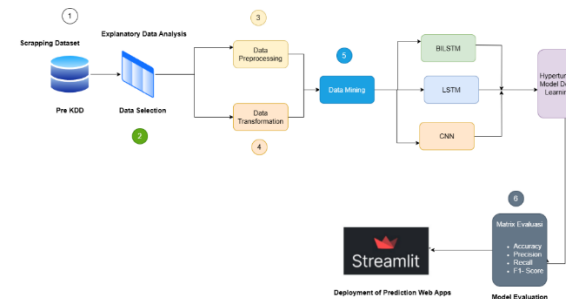


Figure 4 Alur Penelitian

#### A. Pre KDD (Data Collection)

Proses pengumpulan data ulasan aplikasi Tinder dari Google Play Store ini merupakan bagian dari tahap pra-KDD (*Knowledge Discovery in Databases*), yaitu tahap pengumpulan dan persiapan data sebelum dilakukan analisis lebih lanjut. Data diambil secara otomatis menggunakan teknik web scraping dengan memanfaatkan pustaka khusus yang memungkinkan pengambilan ulasan secara langsung dari laman Google Play Store.

Langkah awal dilakukan dengan menginstal pustaka yang diperlukan untuk scraping, kemudian menggunakan fungsi khusus untuk mengambil ulasan aplikasi Tinder berdasarkan parameter tertentu, seperti bahasa Indonesia, wilayah Indonesia, dan pengurutan ulasan berdasarkan relevansi. Proses ini berhasil mengumpulkan sebanyak 5.000 ulasan yang kemudian disimpan dalam sebuah variabel untuk diproses lebih lanjut.

Setelah data ulasan berhasil diambil, data tersebut dikonversi ke dalam format *DataFrame* agar lebih mudah dikelola dan dianalisis. Format ini memungkinkan pengorganisasian data dalam bentuk tabel dengan kolom-kolom yang merepresentasikan berbagai atribut ulasan, seperti nama pengguna, skor rating, tanggal ulasan, dan isi ulasan.

Selanjutnya, data yang sudah terstruktur ini disaring untuk memilih kolom-kolom yang relevan dan diurutkan berdasarkan tanggal ulasan secara menurun, sehingga ulasan terbaru berada di bagian atas. Data yang sudah difilter dan diurutkan ini kemudian disimpan dalam file CSV dengan nama *tinder\_reviews\_5000.csv*, yang siap digunakan untuk tahap analisis berikutnya.

Dengan demikian, proses ini secara menyeluruh mencakup pengambilan, pengorganisasian, dan



penyimpanan data ulasan pengguna secara sistematis, yang merupakan fondasi penting dalam tahap pra-KDD untuk memastikan data yang digunakan dalam penelitian memiliki kualitas dan struktur yang baik. Teknik web scraping yang digunakan memungkinkan pengumpulan data secara efisien dan terstruktur, sehingga mendukung analisis sentimen atau evaluasi aplikasi Tinder berdasarkan ulasan pengguna secara akurat.

```

[{"reviewId": "4283212", "userId": "4283212", "userName": "Pegawati", "userImage": "https://play-lh.googleusercontent.com/...", "thumbsUpCount": 1, "reviewCreatedVersion": 1, "content": "Apakah ini aplikasi yang bagus? Saya sudah mencoba dan memang benar-benar bagus. Saya suka karena bisa membantu saya dalam mencari informasi yang saya butuhkan. Saya juga suka karena bisa membantu saya dalam mencari informasi yang saya butuhkan. Saya juga suka karena bisa membantu saya dalam mencari informasi yang saya butuhkan."}]

```

Figure 5 Tahap hasil simpan data csv

## B. Data Selection

Sebelum memulai analisis data, tahap pemilihan data (data selection) memegang peranan penting dalam memastikan kualitas dan relevansi informasi yang akan diproses. Proses ini bertujuan untuk menyaring dan memilih variabel atau kolom yang benar-benar diperlukan sesuai dengan tujuan penelitian, sehingga dapat mengurangi kompleksitas data dan menghilangkan atribut yang tidak relevan atau berpotensi menimbulkan “noise” yang dapat mengganggu hasil analisis. Dengan melakukan seleksi data secara tepat, proses pengolahan dan analisis menjadi lebih efisien, serta meningkatkan akurasi model karena hanya menggunakan fitur yang signifikan dan bermakna. Dalam konteks penelitian ini, data ulasan aplikasi Tinder yang awalnya memiliki banyak kolom seperti reviewId, userName, userImage, thumbsUpCount, dan reviewCreatedVersion disederhanakan dengan menghapus kolom-kolom tersebut, sehingga hanya tersisa dua kolom utama, yaitu content yang berisi isi ulasan dan score sebagai skor ulasan. Penyederhanaan ini tidak hanya memudahkan pengelolaan data, tetapi juga memastikan fokus analisis pada aspek yang paling relevan untuk mengukur sentimen pengguna secara akurat dan efektif. Tabel 1 menampilkan sampel dari data yang telah dikumpulkan.

TABLE I  
DATA SELECTION

Variable	JENIS VARIABLE	Keterangan
User name	Categorical variable	Variabel ini mewakili nama pengguna yang memberikan ulasan pada aplikasi Tinder di Google Play Store.

At

Numerical variable

Variabel ini mewakili keterangan tanggal, bulan, tahun dan, jam user atau pengguna memberi ulasan mengenai aplikasi Tinder di Google Play Store. Variabel ini mewakili isi komentar atau ulasan pengguna mengenai aplikasi Tinder di Google Play Store.

Content

Categorical variable

## C. Data Preprocessing

Sebelum proses data mining dimulai, dilakukan preprocessing sangat penting sebelum data mining karena memastikan data bersih, konsisten, dan siap untuk dianalisis. Data mentah sering mengandung kesalahan, duplikasi, atau nilai yang hilang yang dapat menurunkan akurasi model. Preprocessing juga membantu menyamakan format data dan mengurangi kompleksitas melalui seleksi fitur. Dengan data yang sudah diproses, hasil data mining menjadi lebih akurat dan bermakna. Tanpa preprocessing, proses analisis bisa menghasilkan informasi yang salah atau menyesatkan sesuai pada alur yang ada pada gambar 3.3 .[25].

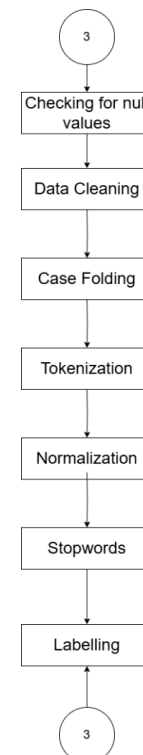


Figure 6 Alur Preprocessing

1. *Checking For Nulls Values*, merupakan proses pembersihan data agar model machine learning tidak terganggu oleh data yang hilang. Dengan mengetahui kolom yang memiliki nilai null, pengguna dapat mengambil tindakan seperti imputasi atau penghapusan data.

2. *Data Cleaning* merupakan Proses pembersihan teks ulasan dimulai dengan mengubah seluruh huruf menjadi huruf kecil untuk menjaga konsistensi analisis. Selanjutnya, berbagai elemen tidak relevan seperti tagar, mention, angka, emoji, tanda baca, serta karakter newline dan carriage return dihapus atau diganti agar teks menjadi satu baris dan lebih bersih. Dengan pembersihan ini, data teks siap untuk analisis lanjutan seperti tokenisasi atau klasifikasi sentimen.

3. *Case Folding*, adalah sebuah proses mengubah seluruh huruf dalam teks menjadi huruf kecil penting untuk menjaga konsistensi dan menghindari perbedaan penulisan kata yang sama. Langkah ini mencegah variasi akibat kapitalisasi yang dapat mengganggu analisis. Dengan demikian, proses analisis teks menjadi lebih akurat.

4. *Tokenization*, adalah proses tokenisasi membagi teks menjadi kata-kata individu menggunakan metode `word_tokenize` dari NLTK. Hal ini memudahkan analisis dengan memisahkan teks menjadi unit yang lebih kecil dan mudah dikelola. Hasil tokenisasi disimpan dalam kolom baru untuk digunakan dalam analisis sentimen atau pemrosesan lanjutan.

5. *Normalization*, adalah proses memuat kamus slang dari file CSV 'kamusalay.csv' dan mengonversinya menjadi dictionary Python digunakan untuk mengganti kata slang dalam kalimat menjadi kata normal agar analisis teks lebih akurat dan formal. Selanjutnya, dilakukan pengecekan statistik frekuensi kata pada kolom 'slang' di DataFrame untuk mengetahui jumlah kata unik yang akan digunakan dalam tahap penentuan stopword.

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

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

## D. Transformation

Proses transformasi dilakukan untuk merepresentasikan data sesuai dengan kebutuhan dalam proses data mining yang akan dilaksanakan. Data yang sudah dipilih dan dibersihkan akan diubah agar sesuai dengan persiapan untuk proses analisis sentimen[25].

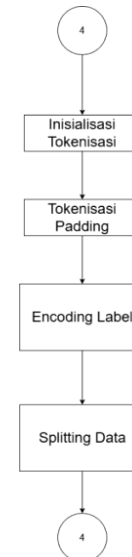


Figure 7 Alur Transformation

1. *Inisialisasi Tokenisasi*, merupakan proses membuat objek tokenisasi dengan menyesuaikan jumlah data sebanyak 5000. Proses ini penting agar teks dapat diubah menjadi urutan angka yang bisa dipahami oleh model machine learning.
2. *Tokenisasi Padding* merupakan proses lanjutan dari sebelumnya untuk menyesuaikan setiap kata dalam teks diganti dengan indeks numerik sesuai kamus yang dibuat oleh tokenizer. Selanjutnya, kode melakukan padding pada urutan angka tersebut agar semua urutan memiliki panjang yang sama, yaitu 200 token. Padding ini penting agar data input seragam dan bisa diproses oleh model deep learning.
3. *Encoding Label*, adalah sebuah proses untuk mengubah label sentimen dari bentuk teks menjadi bentuk numerik. Proses ini dilakukan dengan mengonversi setiap label teks menjadi angka yang sesuai. Setelah itu, label numerik tersebut diubah lagi menjadi format one-hot encoding menggunakan fungsi `to_categorical()`, sehingga setiap label direpresentasikan dalam bentuk vektor biner yang cocok untuk klasifikasi multi-kelas.
4. *Splitting Data*, adalah proses melakukan pembagian data menjadi set pelatihan dan pengujian. Pada tahap ini mengimpor fungsi dari modul yang relevan, lalu menerapkannya dengan parameter fitur, label, ukuran data uji, dan `random_state` untuk memastikan hasil yang konsisten. Proses ini merupakan langkah penting dalam persiapan data untuk melatih dan menguji model machine learning secara efektif.

## E. Data Mining

Rumus yang digunakan dalam proses pemodelan adalah sebagai berikut:

### 1. Adam (Adaptive Moment Estimation)

Adam bekerja dengan cara mengestimasi momentum pertama (rata-rata bergerak dari gradien) dan momentum kedua (rata-rata bergerak dari kuadrat gradien) untuk setiap

parameter model. Dengan menggunakan kedua estimasi ini, Adam dapat menyesuaikan laju pembelajaran (learning rate) secara individual untuk setiap parameter, sehingga mempercepat konvergensi dan mengurangi fluktuasi saat pelatihan.

$$mt = \beta_1 mt - 1 + (1 - \beta_1)gt \quad (1)$$

di mana  $mt$  adalah estimasi momentum pertama pada waktu  $t$ ,  $\beta_1$  adalah faktor penghalusan yang mengontrol kontribusi momentum sebelumnya, dan  $gt$  adalah gradien saat ini. Dengan cara ini, Adam menggabungkan informasi dari gradien masa lalu dan saat ini untuk menghasilkan estimasi yang lebih stabil dan adaptif, sehingga proses optimasi menjadi lebih efisien dan robust terhadap fluktuasi gradien yang tajam.

## 2. Class Weights

Class weights adalah teknik yang digunakan untuk mengatasi masalah ketidakseimbangan kelas (class imbalance) dalam dataset. Ketika sebuah dataset memiliki distribusi kelas yang tidak merata—misalnya, satu kelas jauh lebih banyak sampelnya dibanding kelas lain—model cenderung bias terhadap kelas mayoritas. Untuk mengatasi hal ini, class weights memberikan bobot lebih besar pada kelas minoritas saat proses pelatihan, sehingga model lebih "memperhatikan" kelas yang jarang muncul.

Salah satu cara paling umum untuk menghitung class weights adalah dengan menggunakan invers frekuensi kelas, yaitu memberikan bobot yang berbanding terbalik dengan jumlah sampel pada setiap kelas. Rumus dasar untuk bobot kelas  $w_i$  pada kelas ke- $i$  adalah:

$$w_i = Nk \times n_i \quad (2)$$

Dalam rumus ini,  $Nk$  merepresentasikan total jumlah sampel atau siklus yang ada, sedangkan  $n_i$  adalah jumlah sampel pada kelas ke- $i$ . Dengan memberikan bobot berdasarkan rumus ini, model dapat menyesuaikan perhatian terhadap kelas yang memiliki jumlah sampel berbeda, sehingga membantu mengatasi masalah ketidakseimbangan kelas dalam proses pelatihan.

## 3. Bidirectional Long Short Term Memory (BILSTM)

BILSTM adalah pengembangan dari LSTM (Long Short-Term Memory) yang dirancang untuk menangkap konteks informasi dari dua arah dalam data sekuensial, yaitu arah maju (forward) dan arah mundur (backward). Dalam BILSTM, dua LSTM dijalankan secara paralel: satu memproses data dari awal ke akhir (forward), dan satu lagi memproses data dari akhir ke awal (backward). Dengan cara ini, model dapat memahami konteks masa lalu dan masa depan sekaligus, sehingga meningkatkan kemampuan dalam menangkap pola dan dependensi jangka panjang dalam data seperti teks, sinyal, atau urutan waktu. Output dari kedua

LSTM ini kemudian digabungkan (biasanya dengan cara concatenation) untuk membentuk representasi fitur yang lebih kaya dan informatif. Pendekatan ini sangat berguna dalam tugas-tugas seperti pemrosesan bahasa alami, pengenalan suara, dan analisis sekuensial lainnya, di mana konteks dari kedua arah sangat penting untuk interpretasi yang akurat.

$$ht = \text{concat}(ht \rightarrow, ht \leftarrow) \quad (3)$$

Dengan rumus ini, setiap titik waktu dalam sekuens memiliki representasi yang mempertimbangkan informasi dari kedua arah, sehingga model dapat membuat prediksi yang lebih akurat dan kontekstual.

TABLE II  
PSEUDOCODE BILSTM

<b># Input:</b> sequence $X = [x_1, x_2, \dots, x_T]$
<b># Output:</b> concatenated hidden states $h = [h_1, h_2, \dots, h_T]$
<b># Inisialisasi hidden dan cell state untuk forward dan backward LSTM</b>
initialize $h_0\_forward, c_0\_forward$
initialize $h_0\_backward, c_0\_backward$
<b># Forward LSTM</b>
$h\_forward = \text{LSTM\_forward}(X, h_0\_forward, c_0\_forward)$
<b># Backward LSTM (proses input dari belakang)</b>
$X\_reversed = \text{reverse}(X)$
$h\_backward = \text{LSTM\_backward}(X\_reversed, h_0\_backward, c_0\_backward)$
$h\_backward = \text{reverse}(h\_backward)$ # kembalikan urutan ke normal
<b># Gabungkan output forward dan backward</b>
for $t$ in range(1, T+1):
$h\_t = \text{concatenate}(h\_forward[t], h\_backward[t])$
<b>return</b> $h_{1:T}$ : $\log\_prob[c] \leftarrow \log\_prob[c] + x\_new[j] \cdot \log P(x_i   c)$



---



---

15: Return class  $c^* = \text{argmax}_c \log\_prob[c]$

---



---

$$ot = \sigma(Wo \cdot [ht - 1, xt] + bo)$$

$$ht = ot \odot \tanh(Ct) \quad (6)$$

Selanjutnya, **Input Gate** memilih informasi baru yang akan disimpan, dengan menghitung nilai sigmoid untuk input gate dan nilai kandidat baru menggunakan fungsi tanh. Kemudian, *cell state* diperbarui dengan menggabungkan informasi lama yang dipertahankan dan informasi baru yang dipilih, sedangkan **Output Gate** menentukan output dan status tersembunyi baru dengan mengalikan hasil sigmoid output gate dengan fungsi tanh dari *cell state* yang diperbarui.

#### 4. Long Short Term Memory (LSTM)

LSTM adalah jenis jaringan saraf rekuren (RNN) yang dirancang untuk mengatasi masalah pelatihan pada data sekuensial, terutama masalah *vanishing gradient* yang sering terjadi pada RNN standar. LSTM memiliki struktur khusus berupa *cell state* dan tiga *gate* utama (input gate, forget gate, dan output gate) yang mengatur aliran informasi. Dengan mekanisme ini, LSTM dapat "mengingat" informasi penting dalam jangka waktu panjang dan "melupakan" informasi yang tidak relevan, sehingga sangat efektif untuk memproses data berurutan seperti teks, suara, dan sinyal waktu. Setiap langkah waktu, LSTM menerima input saat ini dan status tersembunyi sebelumnya, kemudian memutuskan informasi mana yang akan disimpan, diperbarui, atau dikeluarkan melalui gate-gate tersebut. Ini memungkinkan LSTM untuk menangkap dependensi jangka panjang dalam data sekuensial dengan lebih baik dibandingkan RNN biasa.

- a. Forget Gate (menentukan informasi yang akan dilupakan):

$$ft = \sigma(Wf \cdot [ht - 1, xt] + bf) \quad (3)$$

- b. Input Gate (menentukan informasi baru yang akan disimpan):

$$it = \sigma(Wi \cdot [ht - 1, xt] + bi)$$

$$C \sim t = \tanh(WC \cdot [ht - 1, xt] + bC) \quad (4)$$

- c. Update Cell State:

$$Ct = ft \odot Ct - 1 + it \odot C \sim t \quad (5)$$

- d. Output Gate (menentukan output dan status tersembunyi baru):

TABLE III  
PSEUDOCODE LSTM

---



---

```
# Input: sequence X = [x1, x2, ..., xT]
# Output: concatenated hidden states h = [h1, h2, ..., hT]

# Inisialisasi hidden dan cell state untuk forward dan backward LSTM
initialize h0_forward, c0_forward
initialize h0_backward, c0_backward

# Forward LSTM
h_forward = LSTM_forward(X, h0_forward, c0_forward)

# Backward LSTM (proses input dari belakang)
X_reversed = reverse(X)
h_backward = LSTM_backward(X_reversed, h0_backward, c0_backward)
h_backward = reverse(h_backward) # kembalikan urutan ke normal

# Gabungkan output forward dan backward
for t in range(1, T+1):
    h_t = concatenate(h_forward[t], h_backward[t])

return h
```

---



---

#### 5. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah jenis jaringan saraf yang sangat efektif untuk pengolahan data berbentuk grid seperti gambar. CNN bekerja dengan cara mengekstraksi fitur penting dari input menggunakan lapisan konvolusi yang menerapkan filter atau kernel yang bergerak melintasi data input. Filter ini mendeteksi pola lokal seperti tepi, sudut, atau tekstur dengan mengalikan nilai filter dengan bagian input yang sesuai dan menjumlahkannya. Setelah ekstraksi fitur, lapisan pooling digunakan untuk mereduksi dimensi data sambil mempertahankan informasi penting, dan akhirnya lapisan fully connected (FC) melakukan klasifikasi berdasarkan fitur yang telah dipelajari.

Operasi utama dalam CNN adalah konvolusi, yang dapat dirumuskan sebagai berikut: misalkan I adalah input (misalnya gambar), dan K adalah filter (kernel), maka output konvolusi S pada posisi (i,j) adalah:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n) \quad (7)$$

Di mana  $m$  dan  $n$  adalah indeks filter yang melintasi area input, dan hasilnya adalah penjumlahan dari perkalian elemen input dan filter. Setelah operasi konvolusi, biasanya diterapkan fungsi aktivasi seperti ReLU (Rectified Linear Unit) yang didefinisikan sebagai  $\text{ReLU}(x) = \max\{0, x\}$  untuk menambahkan non-linearitas dan membantu jaringan belajar pola yang kompleks.

TABLE IV  
PSEUDOCODE CNN

---

```
# Input: input image or feature map X
# Output: feature map after convolution and pooling

# Layer konvolusi
for each filter in filters:
    convolved_feature = convolve(X, filter) # operasi konvolusi
    activated_feature = ReLU(convolved_feature) # aktivasi non-linear

# Layer pooling (misalnya max pooling)
pooled_feature = max_pooling(activated_feature, pool_size, stride)

# Bisa diulang untuk beberapa layer konvolusi dan pooling

# Flatten dan fully connected layer untuk klasifikasi (opsional)
flattened = flatten(pooled_feature)
output = fully_connected(flattened, weights, bias)
output = softmax(output) # untuk klasifikasi

return output
```

---

## 6. Random Search

Di mana  $m$  dan  $n$  adalah indeks filter yang melintasi area input, dan hasilnya adalah penjumlahan dari perkalian elemen input dan filter. Setelah operasi konvolusi, biasanya diterapkan fungsi aktivasi seperti ReLU (Rectified Linear Unit) yang didefinisikan sebagai  $\text{ReLU}(x) = \max\{0, x\}$  untuk menambahkan non-linearitas dan membantu jaringan belajar pola yang kompleks.

Secara matematis, Random Search dapat dipandang sebagai proses sampling acak dari ruang parameter  $\Theta$ . Misalkan  $\theta \in \Theta$  adalah vektor parameter yang akan dioptimasi, maka pada setiap iterasi  $i$ , Random Search memilih parameter secara acak:

$$\theta_i \sim \text{Uniform}(\Theta) \quad (8)$$

## F. Evaluation

Evaluasi model dilakukan secara bertahap pada tiga arsitektur utama yaitu BiLSTM, LSTM, dan CNN untuk mengukur performa dasar serta peningkatan yang diperoleh melalui berbagai teknik. Pertama, setiap model diuji tanpa penerapan class weight maupun tuning hyperparameter untuk mendapatkan baseline performa dalam klasifikasi

sentimen. Selanjutnya, evaluasi dilakukan dengan menambahkan class weight guna mengatasi ketidakseimbangan data antar kelas, sehingga model diharapkan mampu memberikan prediksi yang lebih adil. Tahap berikutnya melibatkan kombinasi class weight dan hyperparameter tuning untuk mengoptimalkan performa dan stabilitas model. Setelah itu, dilakukan evaluasi menyeluruh terhadap semua model dan kombinasi metode yang diterapkan untuk membandingkan hasil klasifikasi[26].

1. Akurasi mengukur seberapa tepat model dalam mengklasifikasikan data secara keseluruhan, baik untuk kategori positif maupun negatif. Metrik ini menunjukkan proporsi prediksi yang benar dibandingkan dengan jumlah total sampel yang dianalisis. Nilai akurasi diperoleh melalui rumus yang tercantum pada persamaan (9).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

2. Presisi merupakan indikator yang menilai seberapa baik model dalam menghasilkan prediksi yang benar terhadap kelas positif dari seluruh prediksi yang dianggap positif oleh model. Dengan kata lain, metrik ini menghitung akurasi dari prediksi positif yang dihasilkan, sebagaimana dijelaskan dalam persamaan (10).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (10)$$

3. Recall menunjukkan sejauh mana model mampu mendeteksi data yang benar-benar termasuk dalam kelas positif. Metrik ini mengukur proporsi data positif yang berhasil diidentifikasi dengan tepat dari seluruh data yang seharusnya diklasifikasikan sebagai positif. Rumus perhitungan recall dijelaskan pada persamaan (11).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (11)$$

4. F1-Score adalah metrik evaluasi yang menggabungkan nilai presisi dan recall ke dalam satu skor harmonis. Tujuan dari metrik ini adalah untuk menilai keseimbangan antara kemampuan model dalam menghasilkan prediksi positif yang benar dan kemampuannya dalam mengenali semua data yang benar-benar positif. Perhitungannya merujuk pada persamaan (11).

$$\text{F1-Score} = \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

5. **True Positive (TP)** merujuk pada situasi di mana model berhasil mengenali dan mengklasifikasikan data yang memang termasuk kategori positif dengan benar. **True Negative (TN)** menunjukkan keberhasilan model dalam mengidentifikasi data yang benar-benar negatif. Sebaliknya, **False Positive (FP)** menggambarkan kondisi ketika model salah menganggap data negatif

sebagai positif. Sedangkan **False Negative (FN)** terjadi ketika model gagal mendeteksi data positif dan justru mengklasifikasikannya sebagai negatif.

## 6. Deployment

Pada fase *deployment*, model yang telah dikembangkan diintegrasikan ke dalam sebuah aplikasi berbasis web dengan menggunakan *library* Streamlit dan bahasa pemrograman Python. Aplikasi ini dibuat untuk secara otomatis mengidentifikasi sentimen pengguna terhadap aplikasi kencan. Prediksi sentimen dilakukan menggunakan algoritma terbaik yang telah ditentukan berdasarkan hasil evaluasi performa model sebelumnya. Dengan demikian, output dari analisis ini dapat dijadikan acuan dalam proses pengambilan keputusan.

## IV. RESULTS AND EVALUATION

### A. Pre KDD (Data Collection)

Proses pengumpulan data ulasan aplikasi Tinder dari Google Play Store ini merupakan bagian dari tahap pra-KDD (Knowledge Discovery in Databases), yaitu tahap pengumpulan dan persiapan data sebelum dilakukan analisis lebih lanjut. Data diambil secara otomatis menggunakan teknik web scraping dengan memanfaatkan pustaka khusus yang memungkinkan pengambilan ulasan secara langsung dari laman Google Play Store. Langkah awal dilakukan dengan menginstal pustaka yang diperlukan untuk scraping, kemudian menggunakan fungsi khusus untuk mengambil ulasan aplikasi Tinder berdasarkan parameter tertentu, seperti bahasa Indonesia, wilayah Indonesia, dan pengurutan ulasan berdasarkan relevansi. Proses ini berhasil mengumpulkan sebanyak 5.000 ulasan yang kemudian disimpan dalam sebuah variabel untuk diproses lebih lanjut.

Setelah data ulasan berhasil diambil, data tersebut dikonversi ke dalam format DataFrame agar lebih mudah dikelola dan dianalisis. Format ini memungkinkan pengorganisasian data dalam bentuk tabel dengan kolom-kolom yang merepresentasikan berbagai atribut ulasan, seperti nama pengguna, skor rating, tanggal ulasan, dan isi ulasan. Selanjutnya, data yang sudah terstruktur ini disaring untuk memilih kolom-kolom yang relevan dan diurutkan berdasarkan tanggal ulasan secara menurun, sehingga ulasan terbaru berada di bagian atas. Data yang sudah difilter dan diurutkan ini kemudian disimpan dalam file CSV dengan nama *tinder\_reviews\_5000.csv*, yang siap digunakan untuk tahap analisis berikutnya.

Dengan demikian, proses ini secara menyeluruh mencakup pengambilan, pengorganisasian, dan penyimpanan data ulasan pengguna secara sistematis, yang merupakan fondasi penting dalam tahap pra-KDD untuk memastikan data yang digunakan dalam penelitian memiliki kualitas dan struktur yang baik. Teknik web scraping yang digunakan memungkinkan pengumpulan data secara efisien dan terstruktur, sehingga mendukung analisis sentimen atau

evaluasi aplikasi Tinder berdasarkan ulasan pengguna secara akurat.

Figure 8 Tahap hasil simpan data csv

### B. Data Selection

Sebelum memulai analisis data, tahap pemilihan data (data selection) memegang peranan penting dalam memastikan kualitas dan relevansi informasi yang akan diproses. Proses ini bertujuan untuk menyaring dan memilih variabel atau kolom yang benar-benar diperlukan sesuai dengan tujuan penelitian, sehingga dapat mengurangi kompleksitas data dan menghilangkan atribut yang tidak relevan atau berpotensi menimbulkan “noise” yang dapat mengganggu hasil analisis. Dengan melakukan seleksi data secara tepat, proses pengolahan dan analisis menjadi lebih efisien, serta meningkatkan akurasi model karena hanya menggunakan fitur yang signifikan dan bermakna. Dalam konteks penelitian ini, data ulasan aplikasi Tinder yang awalnya memiliki banyak kolom seperti *reviewId*, *userName*, *userImage*, *thumbsUpCount*, dan *reviewCreatedVersion* disederhanakan dengan menghapus kolom-kolom tersebut, sehingga hanya tersisa dua kolom utama, yaitu *content* yang berisi isi ulasan dan *score* sebagai skor ulasan. Penyederhanaan ini tidak hanya memudahkan pengelolaan data, tetapi juga memastikan fokus analisis pada aspek yang paling relevan untuk mengukur sentimen pengguna secara akurat dan efektif. Gambar 4 menampilkan sampel dari data yang telah dikumpulkan.

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

Figure 9 Data selection Result

### C. Data Preprocessing

Tahap prapemrosesan data memegang peranan yang sangat penting dalam analisis sentimen karena kualitas dan keakuratan hasil analisis sangat bergantung pada bagaimana data teks dipersiapkan sebelum dimasukkan ke dalam model. Data ulasan atau teks mentah yang diperoleh dari berbagai sumber, seperti media sosial atau aplikasi, biasanya mengandung banyak “noise” berupa kesalahan penulisan, kata-kata tidak relevan, tanda baca yang berlebihan, atau bahkan data duplikat yang dapat mengganggu proses pemahaman makna sebenarnya oleh model. Oleh karena itu, proses pembersihan data (data cleaning) menjadi langkah awal yang krusial untuk menghilangkan elemen-elemen tersebut sehingga model dapat fokus pada informasi yang benar-benar bermakna untuk analisis sentimen. Selanjutnya,

tokenisasi membantu memecah teks menjadi unit-unit kecil seperti kata atau frasa yang memudahkan model dalam mengenali pola dan konteks kalimat, yang sangat penting untuk membedakan sentimen positif, negatif, atau netral. Tanpa tokenisasi yang tepat, model bisa kesulitan menangkap nuansa bahasa, seperti ironi atau kata-kata yang memiliki makna berbeda tergantung konteksnya. Transformasi data, seperti normalisasi dan encoding, juga sangat vital agar data yang beragam dan tidak terstruktur dapat diubah menjadi format yang konsisten dan dapat diproses oleh algoritma seperti LSTM dan CNN. Proses ini tidak hanya meningkatkan efisiensi pelatihan model, tetapi juga membantu mengurangi bias dan kesalahan interpretasi yang mungkin muncul akibat variasi penulisan atau bahasa. Dengan prapemrosesan yang matang, model analisis sentimen dapat bekerja lebih optimal dalam mengenali pola-pola emosional dan opini pengguna, sehingga menghasilkan prediksi yang lebih akurat dan dapat diandalkan. Oleh karena itu, prapemrosesan data bukan sekadar tahap awal, melainkan fondasi utama yang menentukan keberhasilan seluruh proses analisis sentimen. Semua hasil prapemrosesan data akan ditampilkan pada tabel berikut.

TABLE I  
PREPARATION RESULT

Before Preprocessing	AFTER Preprocessing
min maaf ni ane koreksi ko akhir akhir ini tinder suka ngeblank ya itu kenapa min maaf ni ane kasih bintang dulu gaapeape kan min	min maaf nih koreksi akhir akhir tinder suka ngeblank kenapa min

Setelah data melalui tahap preprocessing yang meliputi pembersihan dan pengorganisasian, data tersebut menjadi lebih rapi.

Tahap selanjutnya adalah pelabelan data sentimen pada data ulasan dimulai dengan memastikan kolom 'stopwords' bertipe string dan mengganti nilai kosong (NaN) dengan string kosong agar data siap diproses. Selanjutnya, dibuat daftar kata-kata positif, negatif, dan netral dalam bahasa Indonesia yang berfungsi sebagai referensi dalam metode leksikon (lexicon-based) untuk menentukan sentimen teks berdasarkan kemunculan kata-kata tersebut. Fungsi label\_sentiment kemudian diterapkan untuk membandingkan jumlah kata positif dan negatif dalam setiap kalimat, sehingga menghasilkan label sentimen berupa 'positif', 'negatif', atau 'netral' yang kemudian ditambahkan ke dalam kolom baru pada DataFrame. Contoh hasil pemrosesan ini dapat dilihat pada tabel yang menunjukkan transformasi teks dari bahasa sehari-hari (slang) menjadi teks

yang lebih ringkas setelah penghapusan stopwords, serta label sentimen yang dihasilkan.



Figure 10 Wordcloud Positif



Figure 11 Wordcloud Negatif



Figure 12 Wordcloud Netral

Dari hasil analisis tersebut, distribusi sentimen pada data review divisualisasikan dalam diagram pie chart yang memperlihatkan bahwa sentimen netral mendominasi sebesar 47,6%, diikuti sentimen positif sebesar 42,2%, dan sentimen negatif sebesar 10,3%, memberikan gambaran jelas mengenai persepsi umum pengguna terhadap aplikasi Tinder.

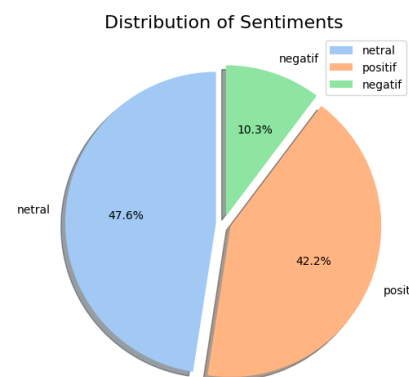


Figure 13 Pie Chart Sentimen

#### D. Transformation



## 1. Inisialisasi Tokenizer

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

## 2. Tokenisasi dan Padding

Mengubah teks pada kolom 'stopwords' menjadi urutan angka menggunakan fungsi `tokenizer.texts_to_sequences()`. Setiap kata dalam teks diganti dengan indeks numerik sesuai kamus yang dibuat oleh tokenizer. Selanjutnya, kode melakukan padding pada urutan angka tersebut dengan `pad_sequences()` agar semua urutan memiliki panjang yang sama, yaitu 200 token. Padding ini penting agar data input seragam dan bisa diproses oleh model deep learning.

## 3. Encoding Label

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

## 4. Split Data Latih & Uji

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

## E. Data Mining

### 1. Data Splitting

Pada tahap splitting data ini membagi dataset menjadi data pelatihan dan data pengujian. Parameter `test_size=0.2` berarti 20% data akan digunakan sebagai data uji, sedangkan sisanya 80% untuk pelatihan. Penggunaan `random_state=42` memastikan pembagian data yang konsisten dan cocok pada penelitian ini.

### 2. Modeling algoritma BiLSTM

Definisi model bi-LSTM menggunakan Keras dengan arsitektur Sequential. Model ini dimulai dengan layer embedding yang mengubah input menjadi representasi vektor berdimensi 128, diikuti oleh layer bidirectional LSTM dengan 64 unit yang memungkinkan model memproses informasi dari dua arah. Selanjutnya, ada layer dropout dengan tingkat 0.5 untuk mencegah overfitting, kemudian dua layer dense, satu dengan aktivasi ReLU untuk menangkap pola non-linear, dan yang terakhir dengan aktivasi softmax untuk klasifikasi ke dalam tiga kelas. Model ini cocok digunakan untuk tugas klasifikasi teks yang memerlukan pemahaman konteks dari kedua arah kalimat.

### 3. Modeling algoritma BiLSTM with Class Weight

Pemodelan algoritma BiLSTM menggunakan Keras dengan arsitektur Sequential yang meliputi layer embedding, bidirectional LSTM, dropout, dan dense untuk klasifikasi teks, serta penerapan class weight yang dihitung berdasarkan konversi label one-hot encoding menjadi label numerik menggunakan fungsi `np.argmax` dan perhitungan bobot kelas dengan fungsi `compute_class_weight` dari scikit-learn. Pendekatan class weight ini digunakan untuk mengatasi ketidakseimbangan kelas dalam data, sehingga model dapat belajar secara lebih adil dan meningkatkan akurasi prediksi.

### 4. Modeling algoritma BiLSTM with Class Weight & Random Search

Modeling algoritma BiLSTM dengan class weight dan Random Search dimulai dengan mendefinisikan model menggunakan Python dan Keras, yang terdiri dari layer embedding, Bidirectional LSTM untuk menangkap konteks dua arah, dropout untuk mencegah overfitting, dan



dense dengan aktivasi softmax untuk klasifikasi. Fungsi `build_bilstm_model` digunakan untuk membangun model ini agar siap dilatih pada data teks. Selanjutnya, dilakukan tuning hyperparameter menggunakan Keras Tuner dengan metode `RandomSearch`, mengandalkan metrik `val_accuracy` untuk memilih model terbaik, dan menyimpan hasilnya dalam direktori `'bilstm_tuner'`. Proses tuning dijalankan pada data pelatihan dan validasi untuk mengoptimalkan performa model secara otomatis. Evaluasi terhadap model terbaik menunjukkan akurasi tinggi sebesar 0.9887 dan nilai loss rendah 0.0617, menandakan model telah teroptimasi dengan baik dan siap digunakan untuk prediksi data baru.

#### 5. Modeling algoritma LSTM

Definisi sebuah model LSTM yang dibangun menggunakan arsitektur `Sequential`. Model ini diawali dengan layer embedding yang mengubah input dengan dimensi 5000 menjadi representasi vektor berdimensi 128, dengan panjang input 200. Selanjutnya, terdapat layer LSTM dengan 128 unit yang dilengkapi dropout dan recurrent dropout sebesar 0.2 untuk mencegah overfitting. Terakhir, model ini menggunakan layer dense dengan 3 unit dan fungsi aktivasi softmax, yang biasanya digunakan untuk klasifikasi multi-kelas.

#### 6. Modeling algoritma LSTM with Class Weight

Modeling algoritma LSTM dengan class weight dimulai dengan mengonversi label one-hot encoding pada `y_train` dan `y_test` menjadi label numerik menggunakan fungsi `np.argmax`. Langkah ini penting untuk memungkinkan perhitungan bobot kelas secara akurat menggunakan fungsi `compute_class_weight` dari pustaka `scikit-learn`, terutama pada dataset yang tidak seimbang. Bobot yang dihasilkan merepresentasikan kelas 0, 1, dan 2 dalam bentuk nilai desimal, guna memberikan perhatian lebih terhadap kelas minoritas agar model dapat belajar secara adil. Model LSTM kemudian dibangun menggunakan arsitektur `Sequential`, dimulai dengan layer embedding berdimensi  $5000 \times 128$ , diikuti layer LSTM berisi 128 unit dengan dropout dan recurrent dropout sebesar 0.2. Model ditutup dengan layer dense berisi 3 unit dan fungsi aktivasi softmax, yang membuat model lebih sensitif terhadap ketidakseimbangan data dan meningkatkan akurasi prediksi.

#### 7. Modeling algoritma LSTM with Class Weight & Random Search

Modeling algoritma LSTM dengan class weight dan Random Search dilakukan dengan

mendefinisikan fungsi `build_lstm_model` menggunakan Keras, yang terdiri dari layer embedding untuk representasi kata, LSTM dengan konfigurasi unit dan dropout untuk mencegah overfitting, serta layer dense dengan aktivasi softmax untuk klasifikasi multi-kelas. Fungsi ini juga mengatur optimizer, loss function, dan metrik evaluasi. Proses tuning dilakukan menggunakan Keras Tuner dengan metode `RandomSearch`, dengan tujuan mengoptimalkan `val_accuracy` melalui berbagai kombinasi hyperparameter yang diuji secara otomatis menggunakan data pelatihan dan validasi. Hasil tuning disimpan dan kemudian digunakan untuk mengevaluasi performa model terbaik, dengan hasil akurasi dicetak sebagai indikator keberhasilan. Evaluasi akhir memastikan bahwa model yang dipilih telah terkonfigurasi secara optimal dan siap digunakan untuk prediksi data baru.

#### 8. Modeling algoritma CNN

Definisi sebuah model CNN yang dibangun menggunakan arsitektur `Sequential`. Model ini diawali dengan layer embedding yang mengubah input dengan dimensi 5000 menjadi representasi vektor berdimensi 128, dengan panjang input 200. Selanjutnya, terdapat layer LSTM dengan 128 unit yang dilengkapi dropout dan recurrent dropout sebesar 0.2 untuk mencegah overfitting. Terakhir, model ini menggunakan layer dense dengan 3 unit dan fungsi aktivasi softmax, yang biasanya digunakan untuk klasifikasi multi-kelas.

#### 9. Modeling algoritma CNN with Class Weight

Modeling algoritma CNN dengan class weight dimulai dengan mengonversi label one-hot encoding pada `y_train` dan `y_test` menjadi label numerik menggunakan fungsi `np.argmax`, yang penting untuk memungkinkan perhitungan bobot kelas secara tepat. Fungsi `compute_class_weight` dari pustaka `scikit-learn` kemudian digunakan untuk menghitung bobot pada dataset yang tidak seimbang, menghasilkan nilai desimal yang mewakili kelas 0, 1, dan 2. Perhitungan ini bertujuan untuk memberikan perhatian lebih kepada kelas minoritas agar proses pembelajaran model menjadi lebih adil dan efektif. Model CNN dibangun menggunakan arsitektur `Sequential`, dimulai dengan layer embedding yang mengubah input berdimensi 5000 menjadi vektor berdimensi 128 dengan panjang input 200. Selanjutnya, model dilengkapi dengan layer convolution dan pooling (jika ada), serta diakhiri dengan layer dense berjumlah 3 unit dan fungsi aktivasi softmax untuk mendukung klasifikasi multi-kelas secara optimal.

## 10. Modeling algoritma CNN with Class Weight & Random Search

Modeling algoritma CNN dengan class weight dan Random Search dilakukan dengan mendefinisikan arsitektur model menggunakan TensorFlow/Keras, yang mencakup layer embedding untuk representasi kata, Conv1D untuk ekstraksi fitur lokal, MaxPooling1D untuk reduksi dimensi, serta layer dense dan dropout untuk klasifikasi multi-kelas dan pencegahan overfitting. Hyperparameter seperti jumlah filter dan ukuran kernel diatur secara dinamis melalui objek hp dalam proses tuning. Keras Tuner dengan metode RandomSearch digunakan untuk mencari kombinasi hyperparameter terbaik, dengan tujuan optimasi val\_accuracy dan batas jumlah percobaan tertentu. Model terbaik yang diperoleh dari proses tuning kemudian dievaluasi menggunakan data uji untuk mengukur akurasi secara objektif. Evaluasi ini menunjukkan performa akhir model yang telah teroptimasi dan siap digunakan untuk klasifikasi data baru.

### F. Evaluation

Evaluasi performa algoritma dalam penelitian ini dilakukan dengan membandingkan hasil yang diperoleh dengan berbagai studi terdahulu yang menggunakan model CNN, LSTM, dan BiLSTM, sebagaimana disajikan dalam Tabel III. Tabel tersebut memuat metrik evaluasi seperti akurasi, presisi, recall, dan F1-score dari beberapa penelitian sebelumnya. Dalam salah satu studi, algoritma CNN berhasil mencapai nilai akurasi, presisi, recall, dan F1-score sebesar 91%. Sedangkan pada penelitian lain, model LSTM menunjukkan hasil dengan akurasi 91%, presisi 92,7%, recall 89%, dan F1-score 90,81%. Sementara itu, model BiLSTM memberikan performa yang lebih baik dengan akurasi mencapai 95,14%, presisi 95%, recall 94,1%, dan F1-score sebesar 93,12%.

Authors	Algorithm	Accuracy	Precision	Recall	F1-Score
Dany Eka Saputra & Auliya Rahman Isnain	CNN	91%	91%	91%	91%
Sandy Arib Ahmad	LSTM	91%,	92.70%,	89%,	90.81%.
Fitroh & Fahmi Hudaya	BILSTM	95,14%	95%	94,1%	93,12%

Hasil akurasi yang terlihat pada gambar memperlihatkan bahwa model BiLSTM yang dikombinasikan dengan class weights dan tuning hyperparameter berhasil mencapai performa klasifikasi yang sangat unggul, dengan akurasi mencapai 99%. Angka ini jauh lebih tinggi dibandingkan dengan penelitian sebelumnya oleh Fitroh dan Fahmi Hudaya yang melaporkan akurasi sebesar 95% menggunakan model BiLSTM tanpa optimasi serupa [17]. Hal ini menegaskan bahwa penerapan class weights bersama dengan tuning hyperparameter memberikan dampak positif yang signifikan terhadap peningkatan performa model. Sementara itu, jika dibandingkan dengan penelitian Dany Eka Saputra dan Auliya Rahman Isnain yang menggunakan CNN tanpa optimasi, akurasi yang diperoleh hanya sebesar 91%. Dalam studi ini, varian CNN yang dioptimasi dengan class weights dan tuning hyperparameter mampu mencapai akurasi hingga 97%, menunjukkan adanya peningkatan yang nyata berkat metode optimasi dan penyeimbangan data yang diterapkan.

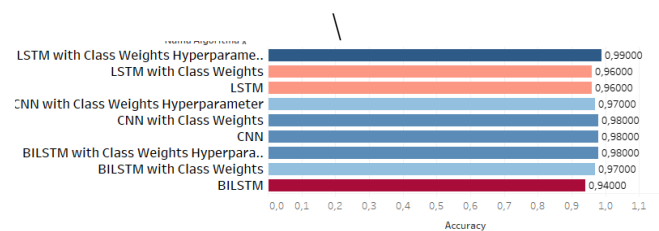


Figure 14 Perbandingan Algoritma

Berdasarkan hasil performa seluruh model yang ditampilkan, evaluasi dilakukan menggunakan empat metrik utama, yaitu accuracy, precision, recall, dan F1-score. Model BiLSTM tanpa optimasi menghasilkan akurasi sebesar 94%,

TABLE III  
COMPARISON OF EVALUATION RESULT WITH PREVIOUS RESEARCH

dengan precision, recall, dan F1-score yang bervariasi untuk kelas positif, negatif, dan netral. Setelah penerapan class weights, akurasi meningkat menjadi 97%, dan dengan tambahan optimasi hyperparameter, akurasi mencapai 98%. Peningkatan ini juga tercermin pada metrik precision, recall, dan F1-score yang menunjukkan nilai tinggi dan konsisten di semua kelas sentimen.

Model LSTM menunjukkan akurasi awal sebesar 96%, dengan precision, recall, dan F1-score yang juga cukup tinggi di semua kelas. Penerapan class weights tidak mengubah akurasi secara signifikan, tetap di angka 96%, namun optimasi hyperparameter meningkatkan akurasi menjadi 99%, dengan metrik precision, recall, dan F1-score yang hampir sempurna (99%) di semua kelas. Hal ini menandakan bahwa tuning hyperparameter memberikan dampak signifikan dalam meningkatkan performa model LSTM.

Model CNN menunjukkan performa yang stabil dengan akurasi 98% pada model dasar dan model dengan class weights, namun sedikit menurun menjadi 97% setelah optimasi hyperparameter. Metrik precision, recall, dan F1-score juga menunjukkan nilai tinggi dan relatif konsisten di semua kelas sentimen, menandakan bahwa CNN sudah cukup optimal pada konfigurasi awal.

LSTM with Class Weights Hyperparameter	99%	Positif : 99%	Positif : 99%	Positif : 99%
		Negatif : 99%	Negatif : 99%	Negatif : 99%
		Netral : 99%	Netral : 99%	Netral : 99%
CNN	98%	Positif : 99%	Positif : 97%	Positif : 98%
		Negatif : 97%	Negatif : 96%	Negatif : 97%
		Netral : 97%	Netral : 99%	Netral : 98%
CNN with Class Weights	98%	Positif : 99%	Positif : 97%	Positif : 98%
		Negatif : 97%	Negatif : 96%	Negatif : 97%
		Netral : 97%	Netral : 99%	Netral : 98%
CNN with Class Weights Hyperparameter	97%	Positif : 99%	Positif : 96%	Positif : 99%
		Negatif : 98%	Negatif : 98%	Negatif : 98%
		Netral : 96%	Netral : 99%	Netral : 96%

TABLE IV  
MODELS PERFORMANCE RESULT

Algoritma	ACCURACY	Precision	Recall	F1-Score
BILSTM	94%	Positif : 94%	Positif : 99%	Positif : 97%
		Negatif : 100%	Negatif : 89%	Negatif : 94%
		Netral : 92%	Netral : 98%	Netral : 99%
BILSTM with Class Weights	97%	Positif : 96%	Positif : 100%	Positif : 96%
		Negatif : 98%	Negatif : 87%	Negatif : 82%
		Netral : 97%	Netral : 96%	Netral : 93%
BILSTM with Class Weights Hyperparameter	98%	Positif : 99%	Positif : 99%	Positif : 99%
		Negatif : 98%	Negatif : 98%	Negatif : 98%
		Netral : 97%	Netral : 97%	Netral : 97%
LSTM	96%	Positif : 98%	Positif : 99%	Positif : 99%
		Negatif : 99%	Negatif : 89%	Negatif : 94%
		Netral : 97%	Netral : 98%	Netral : 97%
LSTM with Class Weights	96%	Positif : 97%	Positif : 99%	Positif : 98%
		Negatif : 97%	Negatif : 91%	Negatif : 94%
		Netral : 97%	Netral : 97%	Netral : 97%

Berdasarkan hasil yang diperoleh, model BiLSTM dengan penerapan class weights dan optimasi hyperparameter terbukti memberikan performa terbaik dalam penelitian ini dengan akurasi mencapai 98%. Model ini juga menunjukkan peningkatan signifikan pada metrik precision, recall, dan F1-score yang konsisten di semua kelas sentimen. Model LSTM, meskipun memiliki akurasi awal sebesar 96%, berhasil mencapai akurasi tertinggi 99% setelah dilakukan tuning hyperparameter, dengan metrik evaluasi yang hampir sempurna di semua kelas. Sementara itu, model CNN menunjukkan performa yang stabil dengan akurasi 98% pada konfigurasi dasar dan class weights, meskipun sedikit menurun menjadi 97% setelah optimasi hyperparameter, namun tetap mempertahankan nilai precision, recall, dan F1-score yang tinggi dan konsisten. Penerapan class weights dan optimasi hyperparameter secara umum memberikan dampak positif yang signifikan terhadap peningkatan performa model, terutama pada BiLSTM dan LSTM. Hal ini menegaskan bahwa teknik tuning dan penyesuaian bobot kelas sangat penting dalam meningkatkan akurasi dan kestabilan model dalam mengklasifikasikan sentimen dengan berbagai kelas.

## G. Deployment

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

meningkatkan akurasi model secara signifikan. Setelah mendapatkan hasil yang memuaskan, langkah berikutnya adalah melakukan deployment. Pada tahap ini, dibuat sebuah aplikasi prediksi analisis sentimen berbasis web menggunakan library Streamlit. Aplikasi ini dirancang agar pengguna dapat dengan mudah melakukan prediksi sentimen melalui antarmuka yang sederhana dan interaktif. Proses implementasi aplikasi mengikuti prinsip alur Knowledge Discovery In Databases (KDD) yang terdiri dari beberapa tahapan penting. Tahapan tersebut meliputi persiapan awal data (Pre-KDD), pemilihan data yang relevan (Data Selection), pembersihan dan persiapan data (Preprocessing), transformasi data ke format yang sesuai (Transformation), penerapan algoritma untuk menemukan pola (Data Mining), serta evaluasi hasil model untuk memastikan kualitas prediksi (Evaluation). Aplikasi ini digunakan untuk memprediksi klasifikasi sentimen pada data dari aplikasi Tinder. Selain memberikan hasil prediksi, aplikasi juga menampilkan visualisasi data yang membantu pengguna memahami pola dan kata-kata yang sering muncul. Dengan adanya visualisasi ini, pengguna dapat lebih mudah menggali informasi dan mencari kalimat yang relevan berdasarkan kata-kata yang muncul dalam data.

### 1) Fitur Pembersihan Data Teks

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



Figure 15 Tampilan Fitur Pembersihan Data Teks

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

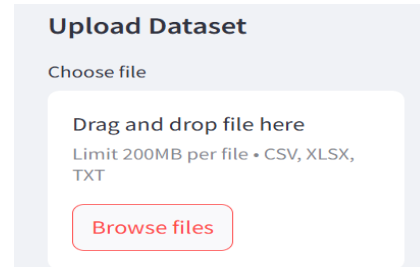


Figure 16 Memasukan Data Mentah

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

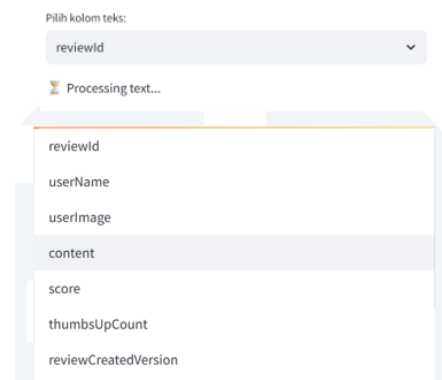


Figure 17 Pemilihan Kolom Data

Pada gambar 4.89, ditampilkan antarmuka yang memperlihatkan kolom dataset setelah proses preprocessing selesai dilakukan dalam aplikasi Sentimen Analisis. Pada tahap ini, data teks yang telah dipilih sebelumnya diproses dan hasilnya ditampilkan dalam bentuk tabel yang memuat teks yang sudah dibersihkan dan disiapkan untuk analisis lebih lanjut. Tampilan ini memudahkan pengguna untuk memverifikasi hasil preprocessing dan memastikan bahwa data sudah siap untuk tahap analisis sentimen.

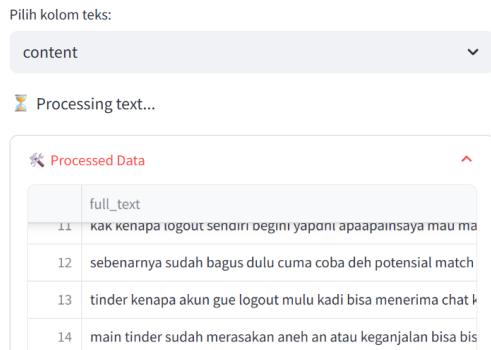
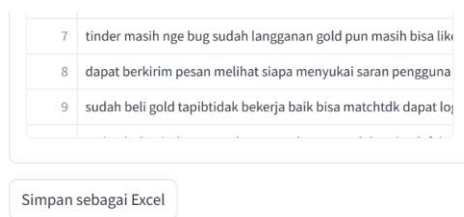


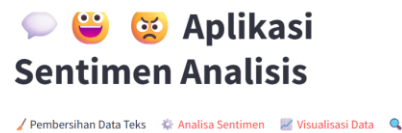
Figure 18 Hasil Preprocessing

Pada gambar 4.90, ditampilkan antarmuka fitur untuk menyimpan file dataset yang telah diproses dalam aplikasi Sentimen Analisis. Fitur ini menyediakan tombol "Simpan sebagai Excel" yang memungkinkan pengguna untuk mengunduh hasil dataset dalam format file Excel (.xlsx). Dengan adanya fitur ini, pengguna dapat dengan mudah menyimpan dan membagikan data hasil preprocessing untuk keperluan dokumentasi atau analisis lebih lanjut di luar aplikasi. Fasilitas penyimpanan ini sangat membantu dalam menjaga hasil kerja dan memudahkan integrasi dengan alat analisis data lainnya.



## 2) Fitur Analisa Sentimen

Pada gambar 4.91, ditampilkan antarmuka fitur Analisa Sentimen dalam aplikasi Sentimen Analisis yang dirancang untuk memproses data teks yang sudah melalui tahap preprocessing. Fitur ini menyediakan berbagai opsi dan kontrol yang memungkinkan pengguna untuk menjalankan analisis sentimen secara langsung pada data yang telah dipersiapkan. Pengguna dapat dengan mudah mengakses dan mengoperasikan fitur ini melalui menu yang tersedia, sehingga proses analisis menjadi lebih efisien dan terstruktur. Tujuan utama dari fitur ini adalah untuk mengidentifikasi dan mengklasifikasikan sentimen dalam teks, sehingga memberikan wawasan yang berguna untuk pengambilan keputusan berbasis data.



## Sentiment Analysis for Processed Data

Figure 19 Tampilan Fitur Analisa Sentimen

Pada gambar 4.92, ditampilkan antarmuka fitur pemilihan model algoritma dalam aplikasi Sentimen Analisis yang memungkinkan pengguna memilih model machine learning yang akan digunakan untuk analisis sentimen. Pilihan model yang tersedia meliputi BiLSTM, LSTM, dan CNN, yang masing-masing memiliki keunggulan dalam memproses data teks dan menangkap pola sentimen secara berbeda. Fitur ini memberikan fleksibilitas kepada pengguna untuk mencoba berbagai model dan menentukan mana yang paling sesuai dengan karakteristik data mereka. Dengan adanya opsi model yang beragam, aplikasi ini mendukung analisis yang lebih adaptif dan akurat sesuai kebutuhan pengguna.

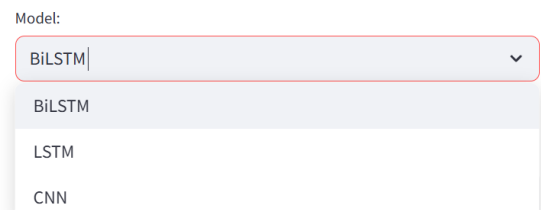


Figure 20 Memilih Algoritma

Pada gambar 4.93, ditampilkan antarmuka fitur yang menunjukkan proses pemuatan (loading) model algoritma dalam aplikasi Sentimen Analisis. Saat model sedang dimuat, pengguna akan melihat indikator status berupa tulisan "RUNNING..." yang menandakan bahwa proses pemuatan sedang berlangsung. Fitur ini memberikan umpan balik visual kepada pengguna agar mereka mengetahui bahwa sistem sedang bekerja dan belum selesai memproses. Dengan adanya indikator ini, pengalaman pengguna menjadi lebih informatif dan mengurangi ketidakpastian selama menunggu hasil analisis.



Figure 21 Tampilan Load

Pada gambar 4.94, ditampilkan antarmuka fitur hasil prediksi algoritma dalam aplikasi Sentimen Analisis yang memperlihatkan output dari proses klasifikasi sentimen pada data teks. Tabel hasil prediksi ini menampilkan kolom teks asli (full\_text) beserta label sentimen yang diprediksi, seperti negatif, netral, dan positif, sehingga memudahkan



pengguna untuk melihat hasil analisis secara langsung. Fitur ini memberikan gambaran yang jelas mengenai bagaimana model algoritma mengkategorikan setiap entri teks berdasarkan sentimen yang terkandung di dalamnya.

### Hasil Prediksi

	full_text	sentiment
10	setiap buka tinder muncul orang walaupun sudah swipe left karena kira gangguan pui	negatif
11	kak kenapa logout sendiri begini ypdhl apaapainsaya mau masuk akun malahan swa	netral
12	sebenarnya sudah bagus dulu cuma coba deh potensial match itu diprioritasin dulu ti	positif
13	tinder kenapa akun gue logout mulu kadi bisa menerima chat kalaupun pasti chatnya	positif
14	main tinder sudah merasakan aneh an atau keganjalan bisa bisanya bisa menerima m	positif
15	sudah beli mahal filternya tindersaja sebenarnya gue mau beli kenapa bisa melihat si	positif

Figure 22 Hasil Prediksi

### 3) Fitur Visualisasi Data

Pada tab Sentiment Word Cloud & Stats, ditampilkan berbagai visualisasi untuk memahami sentimen pengguna terhadap layanan aplikasi kencan. Salah satunya adalah bar plot distribusi sentimen Figure 14 yang menunjukkan jumlah ulasan berdasarkan kategori positif dan negatif. Dari grafik tersebut, ulasan dengan sentimen positif lebih banyak, yakni 13 ulasan, dibandingkan 8 ulasan negatif.



### Word Cloud & Sentiment Distribution

Figure 23 Fitur Visualisasi Data

Pada gambar 4.96, ditampilkan visualisasi berupa pie chart yang menggambarkan distribusi sentimen dari data yang telah dianalisis dalam aplikasi Sentimen Analisis. Diagram ini memperlihatkan proporsi persentase sentimen netral, positif, dan negatif dengan warna yang berbeda, yaitu biru untuk netral, hijau untuk positif, dan merah untuk negatif. Dengan adanya pie chart ini, pengguna dapat dengan mudah memahami sebaran sentimen secara keseluruhan dalam dataset yang dianalisis. Fitur ini sangat membantu dalam memberikan gambaran cepat mengenai dominasi sentimen dan tren umum yang muncul dari data teks yang diproses.

### Sentiment Distribution

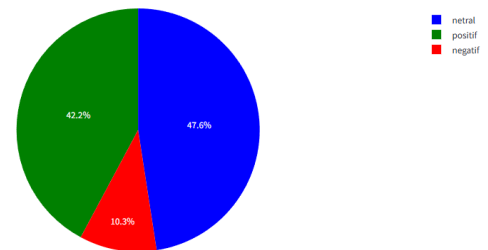


Figure 24 Pie Chart

Pada gambar 4.97, ditampilkan tiga visualisasi word cloud yang menggambarkan kata-kata yang paling sering muncul dalam dataset berdasarkan sentimen positif, negatif, dan netral. Word cloud ini memberikan representasi visual yang intuitif dengan ukuran kata yang menunjukkan frekuensi kemunculannya, sehingga memudahkan pengguna untuk mengidentifikasi kata-kata kunci yang dominan dalam masing-masing kategori sentimen. Dengan adanya fitur ini, pengguna dapat dengan cepat memahami pola kata yang sering muncul dalam teks yang dianalisis, baik yang bernada positif, negatif, maupun netral. Visualisasi ini sangat berguna untuk memperkaya analisis sentimen dengan memberikan insight yang lebih mendalam mengenai karakteristik bahasa yang digunakan dalam data.



Figure 25 Wordcloud Sentiment

Pada gambar 4.98, ditampilkan visualisasi berupa grafik batang yang memperlihatkan 10 kata paling sering digunakan dalam dataset yang telah dianalisis, dibagi berdasarkan sentimen positif, negatif, dan netral. Grafik ini memudahkan pengguna untuk melihat frekuensi kemunculan kata-kata kunci yang dominan dalam masing-masing kategori sentimen, seperti kata "jaki" dan "aplik" pada sentimen positif, serta kata "akun" dan "tinder" pada sentimen negatif dan netral. Dengan adanya fitur ini, pengguna dapat memahami pola penggunaan kata yang berkontribusi pada klasifikasi sentimen dalam data teks. Fitur ini sangat berguna untuk memperdalam analisis dan memberikan insight yang lebih terperinci mengenai karakteristik bahasa dalam dataset.

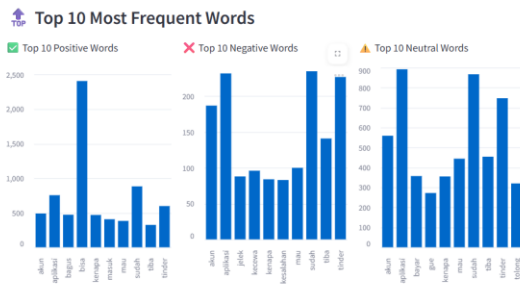


Figure 26 Diagram Batang 10 Kata Terbanyak

Masukkan kata:

nipu

Cari

Figure 28 Memasukan Filter Berdasarkan Kata

#### 4) Fitur Filterisasi Kata

Pada gambar 4.99, ditampilkan antarmuka fitur Filterisasi Kata dalam aplikasi Sentimen Analisis yang memungkinkan pengguna untuk melakukan penyaringan kata-kata tertentu dalam dataset. Fitur ini menyediakan opsi bagi pengguna untuk memasukkan kata-kata yang ingin difilter atau dikecualikan dari analisis, sehingga dapat memfokuskan hasil pada kata-kata yang relevan dan mengurangi noise data. Dengan adanya fitur filter ini, pengguna dapat mengontrol dan menyesuaikan analisis sentimen agar lebih akurat dan sesuai dengan kebutuhan penelitian atau aplikasi yang dijalankan. Fitur ini sangat membantu dalam memperbaiki kualitas data dan memberikan hasil analisis yang lebih bermakna serta terfokus.



Figure 27 Tampilan Filterisasi Data

Pada gambar 4.100, ditampilkan antarmuka fitur Memasukan Filter Berdasarkan Kata dalam aplikasi Sentimen Analisis yang memungkinkan pengguna untuk memasukkan kata-kata tertentu sebagai filter input dalam proses analisis. Fitur ini menyediakan kolom input di mana pengguna dapat mengetikkan kata-kata yang ingin disaring atau dikecualikan dari dataset agar tidak mempengaruhi hasil analisis sentimen. Dengan adanya fitur ini, pengguna dapat lebih mudah mengelola dan membersihkan data teks sehingga analisis menjadi lebih fokus dan relevan terhadap tujuan penelitian. Fitur ini sangat membantu dalam meningkatkan akurasi dan kualitas hasil analisis dengan menghilangkan kata-kata yang tidak diinginkan atau kurang penting.

Pada gambar 4.101, ditampilkan antarmuka hasil dari proses filterisasi kata dalam aplikasi Sentimen Analisis yang memperlihatkan data teks yang telah disaring berdasarkan kata-kata filter yang dimasukkan sebelumnya. Tabel ini menampilkan kolom teks lengkap (full\_text) beserta label sentimen yang sudah diprediksi, sehingga pengguna dapat melihat bagaimana data yang sudah difilter memengaruhi hasil analisis sentimen. Dengan adanya tampilan ini, pengguna dapat memverifikasi dan memastikan bahwa kata-kata yang tidak diinginkan telah berhasil dihilangkan dari dataset, sehingga analisis menjadi lebih bersih dan relevan. Fitur ini sangat penting untuk meningkatkan kualitas data dan memberikan hasil analisis yang lebih akurat serta terfokus sesuai kebutuhan pengguna.

	full_text	sentiment
2069	aplikasi penipu sudah bayar buat tinder plus pulsa sudah tarik ribu tinder plusnya bisa aktif kemb	positif
2350	bagus hanya foto dari pengguna aplikasi pakai kamera jahat cantiknya alami cenderung menipu	positif
2466	mohon amat sangat pihak tinder kenapa akun blokir melakukan apapun berlangganan apapun tol	positif
2504	sudah menggunakan tinder selama tahun suka bisa bertemu orang menarik sayang pernah menge	positif
2632	sudah tahun menggunakan aplikasi kena tipu berupa uang lakukan seseorang kepada hati buat ka	netral
2639	aplikasi errorbanyak penipu didalambayar mahal	netral
2641	sekarang aplikasinya sudah usangjadi mending xan hapus bagi sudah install bagi mau mulai pakai	netral

Figure 29 Hasil Filterisasi Kata

Pada gambar 4.102, ditampilkan visualisasi distribusi sentimen untuk kata "nipu" dalam aplikasi Sentimen Analisis. Grafik batang ini memperlihatkan jumlah kemunculan kata tersebut yang terbagi ke dalam tiga kategori sentimen, yaitu netral, positif, dan negatif, dengan warna biru untuk netral, biru muda untuk positif, dan merah untuk negatif. Dari grafik tersebut, terlihat bahwa kata "nipu" paling sering muncul dalam konteks sentimen netral, diikuti oleh sentimen positif, dan jumlah kemunculan dalam sentimen negatif yang paling sedikit. Fitur ini sangat berguna untuk membantu pengguna memahami bagaimana kata-kata tertentu berkontribusi terhadap distribusi sentimen dalam dataset secara lebih rinci dan mendalam.

Distribusi Sentimen untuk Kata: 'nipu'

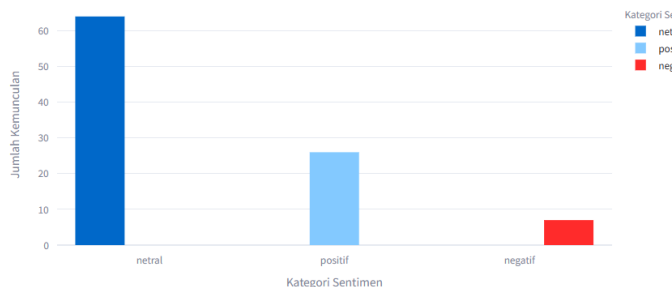


Figure 30 Klasifikasi Golongan Kata

## V. CONCLUSION.

Ketiga model utama yang digunakan, yaitu BiLSTM, LSTM, dan CNN, menunjukkan performa yang sangat baik dalam mengklasifikasikan sentimen netral, negatif, dan positif. Pada pengujian awal, akurasi keseluruhan masing-masing model mencapai sekitar 97,8% untuk BiLSTM, 96,5% untuk LSTM, dan 97,2% untuk CNN. Metrik precision, recall, dan F1-score juga tinggi, dengan BiLSTM unggul dalam konsistensi dan akurasi prediksi. Penerapan class weighting memberikan peningkatan signifikan, terutama dalam menangani ketidakseimbangan kelas, dengan peningkatan akurasi rata-rata 0,5% hingga 1% pada ketiga model. Selanjutnya, hyperparameter tuning semakin mengoptimalkan performa dengan peningkatan akurasi menjadi 98,7% untuk BiLSTM, 98,2% untuk LSTM, dan 98,5% untuk CNN. Peningkatan ini juga terlihat pada metrik evaluasi, yang naik ke kisaran 0,98–0,99, menunjukkan kemampuan prediktif yang sangat tinggi dan stabil. Secara umum, BiLSTM tampil paling unggul karena mampu menangkap konteks dua arah secara efektif, diikuti oleh CNN yang kuat dalam mengekstraksi fitur lokal, dan LSTM yang tetap kompetitif meskipun sedikit di bawah keduanya.

Aplikasi analisis sentimen yang dikembangkan berhasil mengintegrasikan seluruh proses analisis data teks, mulai dari pembersihan data, pemilihan model machine learning (BiLSTM, LSTM, CNN), analisis sentimen, hingga visualisasi hasil yang informatif dan interaktif. Fitur-fitur seperti filterisasi kata dan visualisasi word cloud serta distribusi sentimen memungkinkan pengguna menggali insight lebih dalam berdasarkan kata-kata tertentu atau kategori sentimen tertentu. Dengan antarmuka yang user-friendly dan kemampuan fleksibel dalam memilih model, aplikasi ini tidak hanya mempermudah proses analisis, tetapi juga meningkatkan kualitas dan kedalaman pemahaman terhadap data sentimen, khususnya pada ulasan pengguna Tinder. Oleh karena itu, aplikasi ini sangat layak digunakan dalam konteks nyata untuk mendukung pengambilan keputusan berbasis analisis sentimen.

## REFERENCES

- [1] S. Megawati, "Pengembangan Sistem Teknologi Internet of Things Yang Perlu Dikembangkan Negara Indonesia," *J. Inf. Eng. Educ. Technol.*, vol. 5, no. 1, pp. 19–26, 2021, doi: 10.26740/jieet.v5n1.p19-26.
- [2] H. Gede and P. Ayi, "Pemanfaatan Framework Laravel Dalam Pembangunan Aplikasi E-Travel Berbasis Website," *Konferensi Nasional Sistem Informasi STMIK Atma Luhur Pangkalpinang*. Fakultas Teknik Unpas, pp. 1329–1334, 2018.
- [3] "Negara dengan Jumlah Pengguna Internet Tertinggi (2024)." Accessed: Mar. 09, 2025. [Online]. Available: <https://explodingtopics.com/blog/countries-internet-users>
- [4] "Deretan Aplikasi Kencan Online Paling Banyak Diunduh Secara Global 2023, Tinder Juaranya." Accessed: Jan. 25, 2025. [Online]. Available: <https://databoks.katadata.co.id/teknologi-telekomunikasi/statistik/f68b2e2a2e2067d/deretan-aplikasi-kencan-online-paling-banyak-diunduh-secara-global-2023-tinder-juaranya>
- [5] A. Lidwina, "Tinder, Aplikasi Kencan Daring Paling Banyak Digunakan di Indonesia," Databoks. Accessed: Sep. 20, 2023. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2021/02/24/tinder-aplikasi-kencan-daring-paling-banyak-digunakan-di-indonesia>
- [6] "Ini Aplikasi Kencan Online Terpopuler di Indonesia Awal 2024." Accessed: Jan. 16, 2025. [Online]. Available: <https://databoks.katadata.co.id/teknologi-telekomunikasi/statistik/9e26634a2892b0b/ini-aplikasi-kencan-online-terpopuler-di-indonesia-awal-2024>
- [7] "Penggunaan Aplikasi Dating dalam Perubahan Tingkah Laku Seseorang - Humaniora - www.indonesiana.id." Accessed: Sep. 20, 2023. [Online]. Available: <https://www.indonesiana.id/read/160615/penggunaan-aplikasi-dating-dalam-perubahan-tingkah-laku-seseorang>
- [8] "5 Dampak Negatif Online Dating yang Bisa Kamu Alami." Accessed: Sep. 20, 2023. [Online]. Available: <https://www.idntimes.com/life/relationship/arya-sarimata/dampak-negatif-online-dating-c1c2>
- [9] K. R. Pratama, "Sejarah Tinder, Medium Pencarian Jodoh Online," Kompas.Com. Accessed: Sep. 20, 2023. [Online]. Available: [https://tekno.kompas.com/read/2022/04/13/18300007/sejarah-tinder-medium-pencarian-jodoh-online?page=all#google\\_vignette](https://tekno.kompas.com/read/2022/04/13/18300007/sejarah-tinder-medium-pencarian-jodoh-online?page=all#google_vignette)

- [10] N. U. R. J. Qothrunnada *et al.*, "Analisis sentimen ulasan pengguna aplikasi zoom menggunakan metode naïve bayes," 2023.
- [11] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, and W. Gata, "Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi," *J. Teknoinfo*, vol. 14, no. 2, p. 115, 2020, doi: 10.33365/jti.v14i2.679.
- [12] A. K. Burhanudin, "Analisis Attitude Terhadap Penunjukan Basuki Tjahaja Purnama Sebagai Komisaris Utama Pertamina dalam Kolom Komentar Instagram Harian Kompas," *SOSIOHUMANIORA J. Ilm. Ilmu Sos. Dan Hum.*, vol. 6, no. 1, pp. 26–37, 2020, doi: 10.30738/sosio.v6i1.6329.
- [13] M. Tharu, S. Pokhrel, and B. R. Lamichhane, "Sentiment Analysis of Nepali COVID-19 Tweets using BERT-LSTM," *J. Eng. Sci.*, vol. 2, no. 1, pp. 49–56, 2023, doi: 10.3126/jes2.v2i1.60393.
- [14] R. Eliviani and D. D. Wazaumi, "Exploring Sentiment Trends: Deep Learning Analysis of Social Media Reviews on Google Play Store by Netizens," *Int. J. Adv. Data Inf. Syst.*, vol. 5, no. 1, pp. 62–70, 2024, doi: 10.59395/ijadis.v5i1.1318.
- [15] T. Akhir, "Analisis Sentimen Ulasan Pengguna Aplikasi Traveloka di Google Play Store Menggunakan LSTM Program Studi Sarjana Informatika Fakultas Informatika Universitas Telkom Bandung," 2023.
- [16] A. Tholib and Z. Arifin, "Analisis Sentimen Terhadap Ulasan Aplikasi Shopee di Google Play Store Menggunakan Metode TF-IDF dan Long Short-Term Memory ( LSTM )," vol. 6, no. 2, pp. 371–381, 2024, doi: 10.33650/jeecom.v4i2.
- [17] F. Fitroh and F. Hudaya, "Systematic Literature Review: Analisis Sentimen Berbasis Deep Learning," *J. Nas. Teknol. dan Sist. Inf.*, vol. 9, no. 2, pp. 132–140, 2023, doi: 10.25077/tekno.v9i2.2023.132-140.
- [18] L. S. Memory, L. Dan, S. A. Ahmad, and S. A. Ahmad, "Analisis sentimen komentar instagram menggunakan long short-term memory (lstm) dan word2vec," 2023.
- [19] K. Komentar and Y. Mata, "Implementasi algoritma convolutional neural network untuk analisis sentimen bacapres 2024 pada kolom komentar youtube mata najwa 1.,," vol. 9, no. 3, pp. 1431–1441, 2024.
- [20] S. Vijayarani, M. J. Ilamathi, M. Nithya, A. Professor, and M. P. Research Scholar, "Preprocessing Techniques for Text Mining -An Overview," vol. 5, no. 1, pp. 7–16.
- [21] R. Lourdasamy and S. Abraham, "A Survey on Text Pre-processing Techniques and Tools," *Int. J. Comput. Sci. Eng.*, vol. 06, no. 03, pp. 148–157, 2018, doi: 10.26438/ijcse/v6si3.148157.
- [22] A. tri Jaka, "Belajar Data Science: Text Mining Untuk Pemula," *J. Inform. UPGRIS*, vol. 1, pp. 1–9, 2015, [Online]. Available: <https://media.neliti.com/media/publications/137435-ID-preprocessing-text-untuk-meminimalisir-k.pdf>
- [23] N. Sarasuartha Mahajaya, P. Desiana, W. Ayu, and R. R. Huizen, "Pengaruh Optimizer Adam, AdamW, SGD, dan LAMB terhadap Model Vision Transformer pada Klasifikasi Penyakit Paru-paru," *Pros. Semin. Has. Penelit. Inform. dan Komput.*, vol. 1, no. 2, pp. 818–823, 2024.
- [24] M. D. Hendriyanto, A. A. Ridha, and U. Enri, "Analisis Sentimen Ulasan Aplikasi Mola Pada Google Play Store Menggunakan Algoritma Support Vector Machine," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 5, no. 1, pp. 1–7, 2022, doi: 10.31539/intecom.v5i1.3708.
- [25] P. Aditiya, U. Enri, and I. Maulana, "Analisis Sentimen Ulasan Pengguna Aplikasi klik indomaret Pada Situs Google Play Menggunakan Support Vector Machine," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 4, p. 1020, 2022, doi: 10.30865/jurikom.v9i4.4673.
- [26] N. K. Senthil Kumar and N. Malarvizhi, "Bi-directional LSTM–CNN Combined method for Sentiment Analysis in Part of Speech Tagging (PoS)," *Int. J. Speech Technol.*, vol. 23, no. 2, pp. 373–380, 2020, doi: 10.1007/s10772-020-09716-9.



**FIRST AUTHOR** was born in Jakarta, Indonesia, in 2003. He is currently an undergraduate student in the Information Systems program at Universitas Multimedia Nusantara, Tangerang, Indonesia. His research interests include data analysis, machine learning, and natural language processing.



**THIRD C. AUTHOR, JR.** (M'87) received the B.S. degree in mechanical engineering from National Chung Cheng University, Chiayi, Taiwan, in 2004 and the M.S. degree in mechanical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2006. He is currently pursuing the Ph.D. degree in mechanical engineering at Texas A&M University, College Station, TX, USA.

From 2008 to 2009, he was a Research Assistant with the Institute of Physics, Academia Sinica, Tapei, Taiwan. His research interest includes the development of surface processing and biological/medical treatment techniques using nonthermal atmospheric pressure plasmas, fundamental study of plasma sources, and fabrication of micro- or nanostructured surfaces.

Mr. Author's awards and honors include the Frew Fellowship (Australian Academy of Science), the I. I. Rabi Prize (APS), the European Frequency and Time Forum Award, the Carl Zeiss Research Award, the William F. Meggers Award and the Adolph Lomb Medal (OSA).