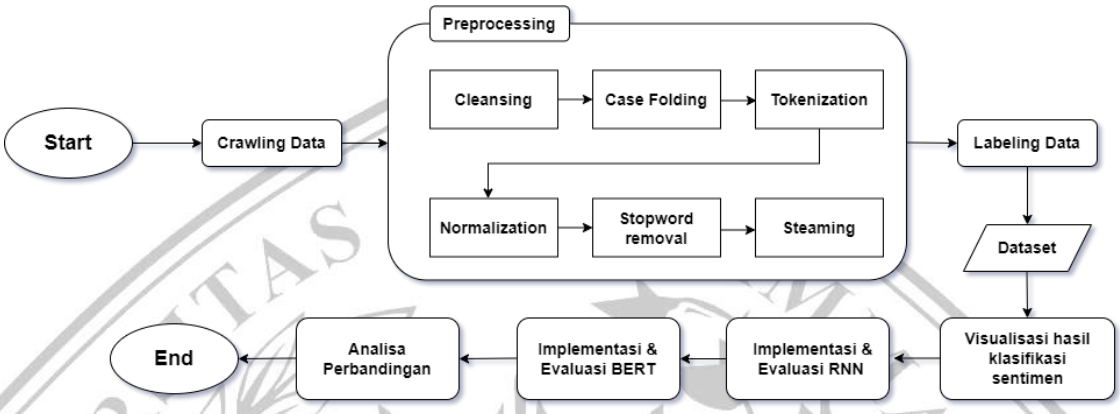
BAB III
METODELOGI PENELITIAN

3.1 Rancangan Penelitian



Gambar 1. Tahapan Penelitian

3.2 Data Crawling

Data untuk penelitian ini diperoleh dari hasil *Crawling* 12.915 respon netizen Tiktok mengenai konten bahasa isyarat dengan hastag #twinklingwatermelon, #handsignlanguage, #signlanguage dan #bahasaisyarat sejak tanggal 1 Desember hingga 10 Desember 2023 dari beberapa konten di aplikasi Tiktok. Proses *crawling* data dilakukan menggunakan Python dan *scraper library* Tiktok dengan layanan API yang dirancang khusus untuk mengambil data dari tiktok.

Tabel 1. Sempel dataset

No	Full_text	Label
1	is this malaysia	Non-influential
2	i want to learn sign butdont know any good source do you have any source for me	Influential
3	hyjiyn and they say kdrama isbad influence	Non-influential
4	i hope you will be givensmooth run	Influential
5	you are not deaf you are so special	Influential
6	really fypnow learn the sign language all come	Non-influential

7	beyond grateful to the person who founded sign language	Influential
8	qualnome onde posso assistant	Non-influential
9	shes so prettyyrit	Non-influential
10	thank you for teaching me the sign language now if the un is not afraid anymore	Influential

3.3 Preprocessing Data

Preprocessing data adalah serangkaian langkah-langkah yang dilakukan untuk membersihkan, penyortiran, konversi data mentah menjadi menjadi representasi logis untuk pemodelan dan analisis. Tujuan dari preprocessing data adalah untuk meningkatkan kualitas data, memperbaiki masalah yang mungkin akan terjadi, dan membuat data agar sesuai dengan penggunaan dalam algoritma atau model yang akan diterapkan.

Proses data preprocessing adalah langkah yang sangat penting dalam persiapan data sebelum menerapkan algoritma atau model analisis sehingga data dapat diubah menjadi representasi yang lebih bersih, konsisten, dan siap untuk digunakan dalam berbagai konteks analisis data.

3.3.1 Data Cleaning

Data cleaning merupakan langkah penting dalam preprocessing data yang bertujuan untuk membersihkan dan mempersiapkan data mentah untuk digunakan secara efektif dalam analisis dan pemodelan. Proses ini mengcangkup identifikasi dan penanganan masalah yang mungkin muncul dalam dataset seperti missing value, outliers, duplikasi dan lain-lain. Hal ini dapat membantu menghilangkan *noise* yang tidak diinginkan atau simbol yang tidak relevan dari teks.

3.3.2 Case Folding

Case folding merupakan suatu proses mengonversi semua karakter dalam sebuah teks ke bentuk yang seragam, biasanya dengan mengubah seluruh teks menjadi huruf kecil atau huruf besar. Tujuan dari case folding ialah untuk menghilangkan variasi dalam penulisan karakter, sehingga memudahkan perbandingan dan analisis teks tanpa memperhatikan perbedaan huruf besar atau kecil.

3.3.3 Tokenization

Tokenisasi merupakan proses membagi sebuah teks atau kalimat menjadi unit-unit yang lebih kecil atau disebut dengan token. Token dapat berupa kata, frasa, atau katrakter tergantung pada tingkat granularitas yang diinginkan. Tokenisasi merupakan langkah penting dalam pemrosesan bahasa alami NLP (*Natural language Processing*) dan membantu dalam pemahaman struktur teks dan pengolahan informasi lebih lanjut. Tokenisasi ini diproses dengan menggunakan library dari NLTK yaitu nltk.tokenize.

3.3.4 Normalization

Normalisasi dalam konteks pemrosesan data merujuk pada serangkaian langkah untuk mengubah atau menyusun data sehingga memiliki format atau skala yang beragam. Tujuan dari normalisasi adalah mengubah kata-kata yang disingkat, tidak standar, atau salah eja menjadi kata standar. Normalisasi umumnya digunakan dalam berbagai bidang seperti statistika, *machine learning* dan *data mining*. Tujuan normalisasi adalah untuk mengubah data dari dataset menjadi bahasa standar dan formal.

3.3.5 Stopword removal

Stopword removal merupakan penghapus stopword, termasuk kata-kata umum yang tidak membawa makna atau fungsi signifikan dalam konteks analisis. Contoh stopword termasuk artikel, konjungsi dan reposisi. Selain itu, kamus stopword juga ditambahkan berdasarkan kata-kata yang tidak diperlukan dalam dataset.

3.3.6 Stemming

Steming merupakan proses mengurangi kata ke bentuk dasar atau akar mereka. Ini bertujuan untuk menghilangkan variasi infleksi dan membawa kata-kata terkait ke bentuk dasar umum. Ini membantu mengurangi redudansi dan mengkonsoldasikan kata-kata serupa.

3.4 Data Labelling

Dari hasil crawling data respon masyarakat terhadap konten bahasa isyarat pada media sosial tiktok yang telah di dapat, tahap selanjutnya adalah membuat label menggunakan *Lexicon*. Sering kali sentimen sebuah komentar dari media sosial sulit untuk ditentukan secara benar ketika komentar tersebut mengandung unsur sarkasme[15]. Sarkasme merupakan bentuk khusus dari ironi yang terjadi ketika seseorang

mengungkapkan sesuatu yang sebenarnya bertentangan dengan makna harfiahnya sehingga hal ini menjadi masalah yang sulit dalam proses analisis sentimen bahkan bagi manusia. *Lexicon* VADER dapat mengatasi komentar yang mengandung sarkasme dalam kalimatnya tanpa harus melalui pelabelan manual dari manusia karena memanfaatkan daftar kata-kata yang telah diakurasi dengan cermat[16]. Metode ini memungkinkan penandaan otomatis berdasarkan analisis sentimen terhadap kata-kata dalam teks, untuk mempermudah klasifikasi tanpa memerlukan pelabelan manual. *Lexicon* VADER telah dikembangkan dengan menggunakan lexicon sentimen yang dianggap sebagai standar emas untuk analisis sentimen dengan fitur dan karakteristik unik di media sosial yang seringkali menghasilkan teks yang pendek, informal dan kaya akan ekspresi sentimen[17]. Dengan menerapkan metode lexicon, teks dapat diidentifikasi dalam kategori positif, negatif atau netral. Namun, mengingat sifat data yang tidak terstruktur di media sosial, diperlukan fleksibilitas metodologis dan adaptasi [18]. Penggunaan analisis teks otomatis dapat memberikan gambaran komprehensif tentang dataset yang luas dan menghindari seleksi data kualitatif [19].

VADER (*Valence Aware Dictionary and sEntiment Reasoner*) merupakan sebuah algoritma analisis sentimen yang memanfaatkan pembobotan nilai kata-kata untuk menentukan nilai polaritas teks, baik itu positif, negatif atau netral. Algoritma ini dikembangkan dengan memperhitungkan cakupan yang luas dalam pengenalan dan penanganan berbagai jenis bahasa dan gaya ekpresi sentimen yang umumnya ditemukan di media sosial melalui pengoperasian dengan memanfaatkan kamus kata-kata yang telah diberi skor sentimen sebelumnya. Dengan memperhitungkan bobot nilai untuk setiap kata dalam suatu opini, maka didapatkan hasil untuk menentukan polaritasnya sebagai dasar untuk mengidentifikasi sifat sentimen. Jika nilai polaritas >1 maka opini dianggap positif, sebaliknya jika nilai polaritasnya <0 opini dianggap negatif, sedangkan jika nilai polaritas = 0, maka opini dianggap netral. Dalam menentukan nilai polaritas suatu teks, digunakan rumus [20][Pamungkas & Putri, 2016 dikutip dalam Hamka & Sari, 2022].

$$Sentiments_{score} = \sum_{i=1}^{n} Sentiments_{score} + W_{positive} + W_{negative}$$

Analisis disusun berdasarkan respon pengguna terhadap konten bahasa isyarat, dimana label *Influential* diidentifikasi menggunakan sentimen positif dan negatif

sedangkan label *Non-influential* diidentifikasi menggunakan sentimen netral dari pengguna Tiktok pada konten tersebut [21]. Sentimen tersebut didapatkan berdasarkan skor sentimen pada nilai polaritas yang dihasilkan melalui penjumlahan bobot positif, negatif dan netral setiap opini yang merujuk pada kamus VADER. Penentuan polaritas sentimen pada suatu ulasan dilakukan dengan memanfaatkan persamaan [22][Vu & Le, 2017].

$$Sentiments_{score} \begin{cases} positif, & jika\ Sentiments_{score} > 0 \\ netral, & jika\ Sentiments_{score} = 0 \\ negatif, & jika\ Sentiments_{score} < 0 \end{cases}$$

Dalam penelitian ini, *Valence Aware Dictionary and sEntiment Reasoner* (VADER) dipilih sebagai alat analisis sentimen. VADER, sebuah leksikon sentimen sumber terbuka, mampu memahami elemen seperti emotikon dan bahasa gaul yang sering digunakan di media sosial. *Lexicon Based* merupakan sebuah pendekatan yang mencangkup frasa, ekpresi, atau konten dalam bentuk teks yang biasa ditemukan di ruang obrolan, dialog, pesan, ulasan dan lainnya[9]. Sehingga secara keseluruhan, pendekatan ini memberikan wawasan mendalam tentang perilaku pengguna berpengaruh dan aspek bahasa yang mempengaaruhi interaksi mereka di platform Tiktok.

3.5 Visualisasi Klasifikasi

Visualisasi klasifikasi sentimen melibatkan representasi grafis dari hasil analisis sentimen pada teks. Ini bisa dilakukan dengan berbagai cara, salah satunya adalah menggunakan visualisasi seperti grafik batang, pie chart, atau word cloud untuk menunjukkan distribusi sentimen *Influential* dan *Non-influential* dalam dataset atau hasil klasifikasi.

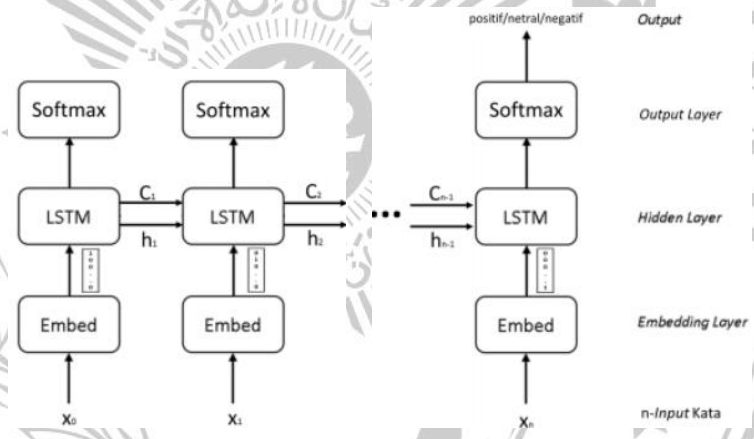
Dalam analisis sentimen, dataset teks dievaluasi menggunakan teknik-teknik pemrosesan bahasa alami (NLP) untuk menentukan polaritas sentimen dari setiap unit teks, seperti kalimat atau dokumen. Visualisasi kemudian digunakan untuk mewakili hasil analisis ini dalam bentuk yang lebih mudah dipahami.

3.6 Implementasi dan Evaluasi RNN

Implementasi RNN (Recurrent Neural Network) melibatkan langkah-langkah yang penting dalam membangun model untuk menganalisis data sekuensial. RNN memiliki kemampuan untuk memodelkan ketergantungan antara kata-kata atau token dalam urutan khususnya pada respon pengguna Tiktok, dimana interaksi pengguna seringkali melibatkan

komentar atau respon berturut-turut yang membentuk narasi atau konversasi. Pembuatan model RNN melibatkan desain struktur yang sesuai, seperti penggunaan layer-layer LSTM atau GRU yang memungkinkan jaringan untuk memahami hubungan dan pola dalam data yang berurutan.

Salah satu metode *deep learning* yang diterapkan dalam penelitian ini adalah RNN dengan arsitektur *Long Short-Term Memory* (LSTM)[23]. Penggunaan model RNN dengan arsitektur LSTM untuk analisis sentimen merupakan pendekatan yang kuat dalam memahami dan mengklasifikasi sentimen dari teks dengan mempertimbangkan konteks temporal dari urutan tersebut. LSTM menggunakan unit memori sel untuk mempertahankan informasi jangka panjang dan mengatasi masalah vanishing gradient yang umum terjadi dalam RNN. Proses pelatihan kemudian dilakukan dengan menggunakan data pelatihan, dimana model menyesuaikan bobot-bobotnya agar dapat memahami pola dalam data sekuensial. [24] Berikut ilustrasi model RNN dengan arsitektur LSTM dapat dilihat pada Gambar 2.



Gambar 2. Model RNN arsitektur LSTM

Model RNN dengan arsitektur LSTM merupakan jaringan saraf yang kuat untuk memproses data urutan seperti teks dengan memperhitungkan konteks temporal. Pada Gambar 2 langkah pertama dalam membuat model adalah menentukan jumlah lapisan dan jenis lapisan yang akan digunakan. Dalam hal ini, digunakan lapisan embedding sebagai input pertama untuk mengonversi kata-kata menjasi representasi vektor. Selanjutnya, kita menambahkan lapisan LSTM yang memiliki unit memori sel untuk mempertahankan informasi jangka panjang dan mengatasi masalah vanishing gradient. Lapisan LSTM ini memungkinkan model untuk memahami hubungan dan konteks antara kata-kata dalam

teks. Setelah lapisan LSTM, selanjutnya menambahkan lapisan dense untuk menghasilkan ouput yang diinginkan seperti klasifikasi teks. Setelah menentukan arsitektur model, langkah berikutnya adalah mengompilasi model dengan menentukan fungsi loss, optimizer, dan metrik evaluasi yang sesuai. Sehingga setelahnya model dilatih dengan menggunakan data pelatihan dan dievaluasi kinerja model menggunakan data pengujian. Koneksi-koneksi antara setiap langkah waktu dalam layer tersembunyi dan arah koneksi dari layer tersembunyi ke layer output ini membentuk struktur rekursif yang memungkinkan RNN untuk memproses data sekuensial dan mempertahankan informasi sepanjang urutan waktu.

Penggunaan metrik evaluasi seperti akurasi, presisi, recall atau F1-score memberikan pemahaman yang mendalam tentang sejauh mana model mampu mengklasifikasikan atau memprediksi data yang belum pernah dilihat sebelumnya. Analisis kinerja melibatkan pemeriksaan hasil prediksi dan memahami kemampuan serta batasan model dalam menangani data sekuensial. RNN disebut memiliki memori yang dapat diingat karena hasilnya bergantung pada komputasi sebelumnya. Ini adalah perbedaan utama antara RNN dan jaringan saraf biasa, di mana RNN mampu menyimpan informasi dari komputasi sebelumnya dan menggunakannya pada elemen berikutnya dalam urutan masukan[6]. Kemampuan ini menjadikan RNN cocok untuk memodelkan urutan data. RNN memiliki kapasitas untuk memahami urutan vektor input yang panjang dan mengenali ketergantungan jarak yang lebih luas. Kemampuan RNN untuk mempertimbangkan konteks pada setiap langkah waktu saat menganalisis urutan kata dalam teks juga merupakan salah satu keunggulannya.

3.7 Implementasi dan Evaluasi BERT

Implementasi model BERT (*Bidirectional Encoder Representations from Transformers*) dalam analisis sentimen respon pengguna media sosial Tiktok melibatkan sejumlah langkah kunci untuk memahami dan mengartikan nuansa yang kompleks dalam teks. Proses fine-tuning memungkinkan model untuk menyesuaikan diri dengan bahasa dan gaya unik yang umumnya digunakan oleh pengguna Tiktok. Setelah model BERT diimplementasikan dan disesuaikan, langkah evaluasi kinerja dilakukan menggunakan data uji yang tidak pernah dilihat oleh model sebelumnya. Evaluasi BERT melibatkan beberapa metrik tergantung pada tugas yang dikerjakan. Misalnya, untuk tugas klasifikasi teks,

metrik seperti akurasi, presisi, recall, dan F1-score sering digunakan. Sementara untuk tugas seperti pemeriksaan kesesuaian teks, metrik seperti keakuratan prediksi atau F1-score untuk kelas *Influential* dan *Non-influential* mungkin lebih relevan. Evaluasi BERT juga melibatkan pengujian kinerja model pada data uji yang berbeda-beda untuk memastikan generalisasi yang baik.

Proses analisis sentimen dengan BERT dimulai dengan pemberian teks ke model. Selama *pre-training*, model dilatih dengan berbagai tugas *pre-trained* pada data yang tidak berlabel. Dilanjutkan dengan proses *fine-tuning*, yaitu proses inisialisasi BERT dengan parameter *pre-trained*. Parameter sebelumnya akan dilatih ulang (*fine-tuned*) menggunakan data yang diberi label dari tugas turunan (*downstream task*)[25].

3.8 Analisa Perbandingan

Membandingkan implementasi model RNN dan BERT dalam analisis sentimen respon pengguna media sosial Tiktok menjadi suatu langkah yang bermakna karena keduanya menawarkan keunggulan dan pendekatan yang berbeda dalam memahami konteks dan sentimen dalam data teks. Dalam kasus ini, RNN, seperti LSTM (Long Short-Term Memory) dikenal karena kemampuannya dalam menangani data sekuensial dan memodelkan ketergantungan temporal. Model ini dapat efektif menangkap perubahan sentimen sepanjang waktu dalam respon pengguna Tiktok yang berkembang seiring dengan konten yang mereka saksikan.

Di sisi lain, BERT menjadi pilihan yang menarik karena kemampuannya dalam memahami konteks dan hubungan antara kata secara bidireksional. Keunggulan ini menjadikan model BERT pilihan yang kuat untuk mengatasi nuansa bahasa dan makna kata yang kompleks dalam teks Tiktok yang sering kali penuh dengan ekspresi kreatif dan bahasa gaul. Algoritma analisis sentimen menggunakan RNN biasanya memerlukan proses pelatihan berulang pada setiap kata dalam teks, sementara BERT telah dilatih pada dataset yang besar secara tidak terawasi. Meskipun BERT memerlukan sumber daya komputasi yang lebih besar untuk penggunaan praktisnya, keunggulan dalam memahami konteks lebih luas dan hubungan antarkata sering kali menghasilkan hasil yang lebih akurat dalam analisis sentimen pada teks yang kompleks seperti komentar atau deskripsi dalam konten bahasa isyarat di TikTok.

Perbandingan ini dapat membantu menentukan kecocokan model dengan karakteristik data respon pengguna TikTok yang unik. Jika respon pengguna cenderung bersifat temporal dan kontekstual, RNN mungkin dapat memberikan performa yang lebih baik. Sebaliknya, jika diperlukan pemahaman yang lebih mendalam terhadap makna kata dalam konteks, BERT dapat menjadi pilihan yang lebih efektif.

3.9 Analisis Lingkungan Kerja

Berikut adalah rincian spesifikasi dari perangkat lunak dan perangkat keras yang akan digunakan dalam penelitian untuk analisis sentiment:

Tabel 2. Analisa lingkungan kerja

Software	Hardware
Python 3	Gen 9 Intel® Core™ i5-9300HF (12CPUs)
Google Colaboratory	Windows 11 pro 64-bit
	16 GB