

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era digital yang berkembang pesat, ulasan film dari pengguna melalui media sosial, situs daring seperti *IMDB*, maupun forum diskusi telah menjadi salah satu sumber informasi penting bagi calon penonton dan pelaku industri perfilman. Ulasan-ulasan ini mencerminkan pendapat, kritik, serta kepuasan atau ketidakpuasan terhadap film yang telah ditonton. Namun demikian, meskipun jumlah data ulasan terus meningkat setiap harinya, pengolahan informasi tersebut secara manual memiliki sejumlah kelemahan. Selain memakan waktu dan tenaga, pendekatan manual juga rentan terhadap subjektivitas serta inkonsistensi dalam interpretasi makna. Hal ini menjadi hambatan utama dalam memperoleh gambaran sentimen publik yang akurat dan representatif terhadap suatu film. Oleh karena itu, diperlukan metode otomatis yang mampu mengklasifikasikan dan menganalisis opini publik secara cepat, konsisten, dan efisien.

Salah satu pendekatan yang digunakan untuk menangani permasalahan ini adalah analisis sentimen (*sentiment analysis*), yaitu proses untuk mengidentifikasi dan mengklasifikasikan opini dalam teks ke dalam kategori seperti positif, negatif atau positif. Meski demikian, metode tradisional seperti *Naive Bayes*, *SVM*, dan *Decision Tree* masih memiliki keterbatasan, terutama dalam memahami konteks dan urutan kata dalam sebuah kalimat. Untuk menjawab tantangan tersebut, model *deep learning* seperti *Long Short-Term Memory (LSTM)* hadir sebagai solusi yang

lebih adaptif. *LSTM*, yang merupakan jenis *Recurrent Neural Network (RNN)*, dirancang khusus untuk menangani data sekuensial dan mempertahankan informasi penting dalam rentang waktu yang panjang. Karakteristik ini menjadikan *LSTM* sangat cocok untuk tugas analisis sentimen terhadap ulasan film, yang umumnya mengandung konteks emosional dan dependensi antar kata dalam struktur kalimat. Dukungan dari berbagai studi sebelumnya juga menunjukkan bahwa *LSTM* mampu menghasilkan akurasi yang tinggi dibandingkan pendekatan lain, menjadikannya pilihan yang layak dan relevan dalam penelitian ini.

Mengatasi kelemahan metode konvensional dalam pemrosesan teks kini semakin banyak dilakukan melalui pendekatan berbasis deep learning. Salah satu arsitektur yang relevan dalam konteks ini adalah *Long Short-Term Memory (LSTM)*, sebuah jenis *Recurrent Neural Network (RNN)* yang dirancang untuk mengatasi masalah vanishing gradient serta mampu mempertahankan informasi dari urutan data yang panjang. Arsitektur *LSTM* dilengkapi dengan elemen-elemen gerbang yang mengatur proses penyimpanan dan pembuangan informasi, menjadikannya sangat efektif dalam menangkap ketergantungan jangka panjang dalam data sekuensial seperti teks ulasan film.

Berbagai penelitian terkini telah menunjukkan keunggulan *LSTM* dalam menganalisis sentimen ulasan film. Misalnya, penelitian Witanto *et al*, menunjukkan bahwa *LSTM* dengan *optimizer* RMSprop menghasilkan akurasi sebesar 80,07%, lebih baik dibandingkan Adam[1]. Sementara itu, penelitian Putri *et al*, menerapkan *LSTM* dalam analisis sentimen berbasis aspek dan mencapai akurasi tinggi hingga 94,80% dalam aspek akting film[2]. Pendekatan yang lebih

kompleks juga dilakukan oleh Nkhata *et al*, yang menggabungkan *BERT* dengan *Bi-LSTM* dan berhasil mencapai akurasi 97,67% dalam klasifikasi biner[3].

Selain itu, kombinasi teknik *embedding* seperti *Word2Vec* dengan *LSTM* juga terbukti efektif dalam studi oleh Linghui Li *et al*, yang diterapkan pada dataset Douban TOP250[4]. Dalam konteks perbandingan, penelitian oleh Hussain dan Naseer menemukan bahwa *Bi-LSTM* memiliki performa yang lebih baik dalam hal keseimbangan presisi dan recall dibandingkan *Logistic Regression* dan *LSTM* biasa[5].

Secara teknis, keunggulan utama *LSTM* terletak pada strukturnya yang memungkinkan penyimpanan informasi dalam jangka waktu yang panjang, sehingga makna kata yang bergantung pada konteks sebelumnya tetap bisa dipertahankan. Ini berbeda dengan pendekatan tradisional yang hanya mengandalkan frekuensi kata dan tidak mempertimbangkan urutan atau makna kontekstual.

Berkaca pada berbagai kelebihan tersebut, penelitian ini merancang dan mengimplementasikan sistem klasifikasi sentimen otomatis terhadap ulasan film menggunakan model *Long Short-Term Memory (LSTM)* berbasis *deep learning*. Implementasi yang diusulkan diharapkan mampu memberikan solusi yang efisien dalam menangani data opini publik berskala besar, baik dalam Bahasa Indonesia maupun Inggris. Di samping itu, penelitian ini turut berkontribusi dalam pengembangan teknologi pemrosesan bahasa alami (*Natural Language Processing*) yang lebih adaptif dan kontekstual terhadap dinamika data modern.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah bagaimana merancang dan mengimplementasikan model *Long Short Term Memory (LSTM)* untuk mendeteksi sentimen dalam ulasan film secara otomatis dengan akurasi yang tinggi?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan mengimplementasikan model *LSTM* untuk mendeteksi sentimen ulasan film secara otomatis. Penelitian ini diharapkan memberikan manfaat akademis sebagai referensi dalam pengembangan analisis sentimen berbasis *deep learning*, serta manfaat praktis dalam membantu pelaku industri perfilman dan masyarakat memahami opini publik terhadap film secara cepat dan efisien.

1.4 Batasan Penelitian

Dalam penelitian ini, peneliti memberikan beberapa batasan penelitian yaitu:

1. Jumlah data eksperimen yang digunakan yaitu berupa ulasan film dalam bahasa Indonesia, filmnya sebanyak 10 dengan format *.xlsx/csv* yang diambil dari platform *Imdb*.
2. Analisis sentimen diklasifikasikan ke dalam tiga kategori: positif, negatif, dan netral. Sentimen positif menunjukkan penilaian baik, misalnya "Film ini sangat menghibur dan aktingnya luar biasa!". Sentimen negatif menunjukkan penilaian buruk, seperti "Ceritanya membosankan dan tidak

masuk akal.". Sedangkan netral bersifat objektif tanpa emosi, contohnya

"Film ini berdurasi dua jam dan berlatar di Jakarta."

3. Model yang digunakan adalah *Long Short Term Memory (LSTM)* tanpa dibandingkan dengan model lain.
4. Khusus film Indonesia di bioskop Indonesia movie.

1.5 Manfaat Penelitian

Penelitian ini bermanfaat untuk mengembangkan sistem deteksi sentimen otomatis pada ulasan film menggunakan model *LSTM*, sehingga dapat membantu analisis opini publik secara lebih cepat dan akurat.

1.6 Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari pokok-pokok permasalahan yang akan dibahas pada masing-masing yang diuraikan menjadi beberapa bagian sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, batasan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini berisi teori – teori yang digunakan pada penelitian ini. Teori – teori yang berhubungan dengan Analisis Sentimen Dalam Ulasan Film Menggunakan Model *Long Short Term Memory (LSTM)* Berbasis *Deep Learning*.

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi tahapan-tahapan penelitian, mulai dari identifikasi masalah, pengumpulan data, analisa model, perancangan sistem, implementasi sistem, dan pengujian sistem.

BAB 4 ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi Analisa dan perancangan Analisis Sentimen Dalam Ulasan Film Menggunakan Model *Long Short Term Memory (LSTM)* Berbasis *Deep Learning*.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dan pengujian pada aplikasi yang akan dibangun.

BAB 6 PENUTUP

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan serta saran untuk penelitian atau pengembangan sistem di masa mendatang.

BAB II

LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen, juga dikenal sebagai *opinion mining*, adalah bidang *Natural Language Processing (NLP)* yang bertujuan mengidentifikasi opini atau emosi (positif atau negatif) dalam sebuah teks. Teknik ini diterapkan pada data seperti ulasan produk atau media sosial untuk memberikan wawasan penting bagi organisasi, mulai dari evaluasi kepuasan pelanggan, menjaga citra merek, hingga pengambilan keputusan strategis berdasarkan tren pasar [6].

Proses analisis sentimen diawali dengan pra-pemrosesan teks yang meliputi tokenisasi, pembuangan *stop words*, dan *stemming* atau *lemmatization* untuk membersihkan data. Setelah itu, fitur diekstraksi untuk dianalisis menggunakan berbagai pendekatan, seperti metode berbasis leksikon atau *machine learning*. Perkembangan terkini di bidang *deep learning* telah meningkatkan akurasi analisis dengan memungkinkan model memahami konteks emosional secara lebih mendalam [7].

Penerapan analisis sentimen sangat luas dan terus berkembang di berbagai sektor. Dalam bisnis, teknologi ini digunakan untuk mengevaluasi kampanye pemasaran dan pendapat pelanggan, sementara di bidang politik dapat mengukur pandangan publik terhadap kebijakan. Seiring ledakan data teks digital, analisis sentimen menjadi alat vital untuk mengubah data mentah menjadi informasi yang mendukung pengambilan keputusan[8].

2.2 Ulasan Film sebagai Objek Analisis

Ulasan film dari berbagai platform seperti IMDb, Rotten Tomatoes, dan media sosial merupakan sumber data yang kaya untuk analisis sentimen. Data ini tidak hanya mencerminkan opini pribadi penonton, tetapi juga memberikan gambaran mendalam tentang berbagai aspek film seperti alur cerita dan akting. Analisis terhadap ulasan ini memungkinkan para peneliti dan pelaku industri untuk memahami pandangan audiens, mengenali tren, hingga memprediksi potensi komersial sebuah film [9].

Teks ulasan film seringkali mengandung tantangan linguistik seperti sarkasme, metafora, dan bahasa informal, yang memerlukan pemahaman konteks mendalam untuk analisis yang akurat. Baik opini yang disampaikan secara eksplisit maupun implisit menuntut metode analisis yang canggih. Untuk mengatasi kompleksitas ini, pendekatan *deep learning* terbukti efektif karena kemampuannya dalam mengenali pola-pola nuansa bahasa dari kumpulan data ulasan yang besar untuk mengklasifikasikan sentimen secara tepat [10].

Pemanfaatan analisis sentimen pada ulasan film sangat berguna bagi banyak pihak. Rumah produksi dapat menggunakan hasilnya sebagai masukan untuk strategi pemasaran dan pengembangan film di masa depan. Bagi para peneliti, analisis ini membantu memahami pembentukan opini publik, sementara dalam bidang NLP, dataset ulasan film populer digunakan untuk mengembangkan serta menguji algoritma analisis sentimen baru [11].

2.3 *Natural Language Processing (NLP)*

Natural Language Processing (NLP) adalah cabang dari kecerdasan buatan

(AI) yang memungkinkan komputer untuk memahami, menafsirkan, dan berinteraksi dengan bahasa manusia. Cakupan *NLP* sangat luas, meliputi tugas-tugas seperti penerjemahan otomatis, analisis sentimen, dan pengenalan suara. Kemampuan ini menjadikan *NLP* sebagai teknologi inti dalam berbagai aplikasi modern, termasuk asisten virtual, mesin pencari, dan sistem rekomendasi [12].

Perkembangan *NLP* telah berevolusi dari pendekatan berbasis aturan dan statistik ke model *machine learning* yang lebih canggih. Beberapa tahun terakhir, kemajuan pesat didorong oleh teknik *deep learning* dengan arsitektur seperti *Recurrent Neural Network (RNN)* dan Transformer. Model-model ini secara signifikan meningkatkan kemampuan komputer dalam menangkap konteks dan nuansa bahasa, sehingga menghasilkan interaksi manusia-komputer yang lebih alami [13].

Penerapan *NLP* melibatkan proses sistematis yang diawali dengan pra-pemrosesan data, seperti tokenisasi, normalisasi, dan penghapusan *stop words*. Setelah data dibersihkan, model dilatih pada kumpulan teks besar untuk mengenali pola linguistik. Dengan terus meningkatnya kapasitas komputasi dan ketersediaan data, *NLP* kini semakin mampu mengatasi tantangan kompleks seperti ambiguitas makna, ironi, dan sarkasme dalam teks [14].

2.4 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) adalah varian arsitektur *Recurrent Neural Network (RNN)* yang dirancang khusus untuk mengatasi masalah *vanishing gradient* pada *RNN* biasa. Keunggulan utama *LSTM* adalah kemampuannya mempelajari dependensi jangka panjang dalam data sekuensial seperti teks atau

deret waktu. Hal ini dimungkinkan oleh mekanisme gerbang (*gates*) internal yang secara cerdas mengatur aliran informasi, memutuskan data mana yang perlu disimpan, dilupakan, atau dikeluarkan, sehingga informasi relevan dapat dipertahankan lebih lama [15].

Struktur inti sel *LSTM* terdiri dari *cell state* sebagai jalur utama informasi dan tiga gerbang pengendali utama: *forget gate*, *input gate*, dan *output gate*. *Forget gate* menentukan informasi mana dari state sebelumnya yang harus dibuang, *input gate* mengatur informasi baru yang akan disimpan, dan *output gate* memutuskan informasi apa yang akan dijadikan keluaran (*output*). Ketiga gerbang ini menggunakan fungsi aktivasi sigmoid untuk mengelola aliran data secara selektif [16]. *Forget gate* berfungsi untuk menentukan bagian informasi dari waktu sebelumnya yang akan dilupakan, dan dirumuskan sebagai:

$$\text{Gerbang Lupa (Forget Gate)} \quad f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

$$\text{Gerbang Input (Input Gate)} \quad i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\text{Gerbang Output (Output Gate)} \quad o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4)$$

Keterangan:

$f_t, \quad i_t, \quad o_t$ = output dari masing-masing gerbang

σ = fungsi aktivasi sigmoid

$W_f, \quad W_i, \quad W_o$ = bobot untuk masing-masing gerbang

h_{t-1} = *hidden state* dari waktu sebelumnya

x_t = *input* saat waktu t

$b_f, \quad b_i, \quad b_o$ = bias dari masing-masing gerbang

Berkat mekanisme gerbangnya, *LSTM* menjadi arsitektur fundamental untuk

berbagai tugas *Natural Language Processing (NLP)* seperti analisis sentimen, pemodelan bahasa, dan penerjemahan mesin. Meskipun arsitektur Transformer kini mendominasi banyak tugas *NLP*, *LSTM* tetap menjadi solusi yang andal dan efisien, terutama untuk kasus-kasus dengan sumber daya komputasi yang lebih terbatas [17].

2.5 Deep Learning

Deep Learning adalah sub-bidang *Machine Learning* yang menggunakan Jaringan Saraf Tiruan (JST) berlapis-lapis untuk meniru cara kerja otak manusia. Keunggulan utamanya adalah kemampuan untuk mempelajari pola dan fitur secara otomatis langsung dari data mentah yang kompleks dan bervolume besar, seperti gambar, suara, dan teks. Pendekatan ini telah merevolusi berbagai bidang, mulai dari pengenalan objek hingga pemrosesan bahasa alami [18].

Arsitektur umum dalam *deep learning* mencakup *Convolutional Neural Networks (CNN)* untuk data visual, *Recurrent Neural Networks (RNN)* beserta variannya seperti *LSTM* untuk data sekuensial, dan Transformer untuk tugas *NLP*. Proses pembelajarannya melibatkan *forward propagation* untuk menghasilkan prediksi dan *backpropagation* untuk mengoreksi kesalahan dengan menyesuaikan bobot jaringan. Dalam proses tersebut, komponen penting seperti fungsi aktivasi, salah satunya adalah fungsi sigmoid, berperan untuk mengolah sinyal keluaran dari setiap neuron [19].

Salah satu komponen penting dalam proses forward propagation adalah fungsi aktivasi [20]. Salah satu fungsi yang umum digunakan dalam *deep learning* adalah fungsi sigmoid, yang dirumuskan sebagai berikut:

$$\text{Fungsi Aktivasi Sigmoid } \sigma(x) = \frac{1}{1+e^{-x}} \quad (1)$$

Keterangan:

$\sigma(x)$ = Fungsi sigmoid, yaitu hasil dari transformasi *input* x

x = *Input* numerik, bisa berupa nilai skalar, vektor, atau matriks

e = Bilangan Euler (≈ 2.71828), basis dari logaritma natural

e^{-x} = Eksponensial negatif dari *input* x

$1 + e^{-x}$ = Penjumlahan 1 dengan hasil eksponensial negatif dari x

$\frac{1}{1+e^{-x}}$ = Bentuk akhir dari fungsi sigmoid, nilainya berada diantara 0 dan 1

Penerapan *deep learning* telah menghasilkan kemajuan besar di berbagai bidang, seperti deteksi objek pada visi komputer dan penerjemahan mesin pada NLP. Namun, teknologi ini juga dihadapkan pada tantangan, seperti kebutuhan akan data pelatihan dalam jumlah masif, beban komputasi yang tinggi, serta sifatnya yang sering dianggap sebagai "kotak hitam" (*black box*) karena kompleksitas dalam interpretasi modelnya." [21].

2.6 Flowchart

Flowchart adalah gambaran grafis yang merepresentasikan langkah-langkah dan urutan proses dalam suatu program atau sistem. Dengan menggunakan simbol-simbol standar seperti *terminator*, *decision*, proses, maupun *input/output*, *flowchart* membantu seorang analis maupun programmer dalam memecah permasalahan ke dalam bagian yang lebih sederhana sehingga lebih mudah dipahami dan dievaluasi. Bagi mahasiswa atau pemula dalam bidang pemrograman, *flowchart* menjadi alat bantu penting sebelum memasuki implementasi kode karena memberikan

visualisasi alur logika yang jelas[22].

Selain digunakan sebagai alat analisis, *flowchart* juga menjadi sarana pembelajaran algoritma dan logika pemrograman. Melalui perangkat lunak pendukung seperti *Flowgorithm*, mahasiswa dapat mempraktikkan penyusunan algoritma dengan lebih mudah karena *flowchart* dapat dijalankan dan divisualisasikan hasilnya. Pendekatan ini tidak hanya memudahkan dalam memahami konsep dasar algoritma, tetapi juga meningkatkan kemampuan problem solving secara terstruktur [23].

Perkembangan penelitian terbaru menunjukkan bahwa *flowchart* tidak hanya digunakan untuk perancangan, tetapi juga dapat dijadikan dasar dalam membangun perangkat lunak yang mampu mengonversi diagram menjadi kode program secara otomatis. Dengan cara ini, *flowchart* bukan hanya media pembelajaran, tetapi juga sarana untuk meningkatkan efisiensi dalam proses pengembangan aplikasi. Konsep ini sudah mulai diterapkan dalam penelitian multidisiplin yang menyoroti hubungan *flowchart*, *pseudocode*, dan implementasi algoritma dalam pendidikan tinggi [24].

2.7 Website

Website merupakan kumpulan halaman yang dapat diakses melalui internet dan berisi informasi, layanan, maupun aplikasi interaktif. Dalam konteks pembelajaran, website dapat dimanfaatkan sebagai media yang mendukung proses belajar mengajar karena mampu menyajikan materi secara lebih menarik, interaktif, serta mudah diakses kapan saja dan di mana saja. Pemanfaatan website juga dapat meningkatkan kemandirian siswa dalam menggali informasi serta memperluas sumber belajar yang tidak terbatas hanya pada buku cetak [25].

Selain sebagai media informasi, *website* dapat dirancang untuk memberikan fitur-fitur khusus, seperti kuis interaktif, forum diskusi, hingga sistem evaluasi hasil belajar. Hal ini membuat *website* tidak hanya menjadi sumber referensi, tetapi juga wadah interaksi antara guru dan siswa. Dengan demikian, keberadaan *website* mampu meningkatkan efektivitas pembelajaran dan mendukung tercapainya tujuan pendidikan yang lebih baik [26].

Lebih jauh lagi, *website* juga digunakan untuk mengembangkan sistem pembelajaran berbasis digital yang sesuai dengan kebutuhan siswa. Pengembangan media pembelajaran interaktif berbasis web memungkinkan guru untuk mengintegrasikan multimedia, seperti gambar, video, dan animasi, sehingga materi menjadi lebih mudah dipahami. Oleh karena itu, *website* berperan penting dalam transformasi digital pendidikan di era industri 4.0 [27].

2.8 Python

Python adalah bahasa pemrograman tingkat tinggi yang populer karena kesederhanaan sintaks dan fleksibilitasnya. Bahasa ini mendukung banyak paradigma pemrograman seperti prosedural, fungsional, hingga berorientasi objek, sehingga sangat cocok digunakan dalam dunia pendidikan maupun penelitian. Dalam bidang akademik, *Python* membantu mahasiswa dan peneliti untuk menyelesaikan berbagai persoalan pemrograman dengan lebih cepat dan efisien [28].

Selain itu, *Python* juga memiliki pustaka (*library*) yang sangat lengkap untuk berbagai kebutuhan, mulai dari analisis data, visualisasi, hingga pengembangan kecerdasan buatan. Dengan adanya pustaka seperti *NumPy*, *Pandas*, dan *Matplotlib*,

pengguna dapat melakukan manipulasi dan eksplorasi data secara lebih mudah. Hal ini menjadikan *Python* sebagai salah satu bahasa yang banyak digunakan dalam penelitian berbasis data maupun pembelajaran berbasis komputer [29].

Dalam pengembangan ilmu pengetahuan, *Python* berperan penting pada penelitian di bidang matematika, sains, dan teknologi informasi. Implementasi *Python* dalam berbagai studi kasus seperti perhitungan matematis, simulasi, maupun analisis big data, menunjukkan bahwa *Python* mampu mendukung perkembangan riset modern. Dengan dukungan komunitas yang besar dan dokumentasi yang lengkap, *Python* semakin relevan digunakan di era digital saat ini [30].

2.9 Visual Studio Code

Visual Studio Code (VS Code) merupakan salah satu code editor yang dikembangkan oleh *Microsoft* dengan dukungan lintas platform, sehingga dapat dijalankan pada sistem operasi *Windows*, *Linux*, maupun *macOS*. *VS Code* populer di kalangan developer karena tampilannya yang sederhana, ringan, serta dilengkapi berbagai ekstensi yang mendukung pemrograman berbagai bahasa, termasuk *Python*, *JavaScript*, *Java*, dan *C++*. Hal ini menjadikan *VS Code* sebagai salah satu pilihan utama dalam pengembangan perangkat lunak modern [31].

Selain itu, *Visual Studio Code* menyediakan fitur bawaan seperti *debugging*, *syntax highlighting*, *intelliSense*, dan *integrated terminal* yang sangat membantu programmer dalam menulis kode. Dukungan ekstensi yang beragam memungkinkan pengguna menyesuaikan editor sesuai kebutuhan spesifik proyek. Dengan berbagai kemudahan tersebut, *VS Code* tidak hanya digunakan oleh

programmer profesional, tetapi juga mahasiswa yang baru belajar pemrograman untuk meningkatkan keterampilan logika dan pemahaman kode [32].

Dalam dunia pendidikan, *VS Code* sering dimanfaatkan sebagai media pembelajaran dalam mata kuliah pemrograman maupun proyek penelitian. Penggunaannya yang fleksibel membuat mahasiswa dapat berlatih mengembangkan aplikasi berbasis *web*, *mobile*, maupun desktop secara lebih efisien. Dengan komunitas global yang besar, dokumentasi yang lengkap, serta dukungan teknologi terkini, *VS Code* semakin berperan penting sebagai sarana pengembangan perangkat lunak di era digital [33].

2.10 *Blackbox Testing*

Blackbox Testing merupakan metode pengujian perangkat lunak yang berfokus pada fungsionalitas sistem tanpa memperhatikan struktur internal atau kode program. Teknik ini digunakan untuk memastikan bahwa *input* tertentu menghasilkan *output* sesuai dengan yang diharapkan, sehingga dapat mendeteksi kesalahan fungsi aplikasi. Dengan demikian, penguji dapat mengevaluasi kualitas perangkat lunak dari sudut pandang pengguna akhir tanpa harus memahami detail teknis pemrograman [34].

Metode *Blackbox Testing* biasanya diterapkan pada tahap akhir pengembangan perangkat lunak untuk memverifikasi bahwa seluruh fitur sudah berjalan sesuai spesifikasi. Jenis pengujian yang umum digunakan dalam metode ini antara lain *equivalence partitioning*, *boundary value analysis*, dan *decision table testing*. Penggunaan metode ini membantu pengembang menemukan *bug* yang berkaitan dengan logika aplikasi maupun kesalahan antarmuka yang mungkin

terlewat pada tahap pengujian berbasis kode [35].

Selain itu, *Blackbox Testing* banyak dimanfaatkan dalam penelitian akademik maupun implementasi aplikasi berbasis *web*, *mobile*, maupun *desktop*. Metode ini mempermudah mahasiswa dan peneliti dalam menguji fungsionalitas aplikasi secara praktis, tanpa memerlukan pemahaman mendalam terhadap struktur kode. Dengan fleksibilitas dan efektivitasnya, *Blackbox Testing* menjadi pendekatan penting dalam menjamin kualitas perangkat lunak yang dikembangkan [36].

2.11 Hypertext Markup Language (HTML)

Hypertext Markup Language (HTML) merupakan bahasa markup standar yang digunakan untuk menyusun dan menampilkan halaman web. *HTML* berfungsi untuk menentukan struktur dasar halaman seperti heading, paragraf, gambar, tautan, hingga tabel. Seiring perkembangan teknologi, *HTML* telah berevolusi hingga *HTML5* yang lebih interaktif dengan dukungan multimedia seperti audio, video, dan canvas untuk grafis. Dengan keunggulan ini, *HTML* tetap menjadi fondasi utama dalam pengembangan aplikasi berbasis *web* modern [37].

Selain sebagai dasar pembuatan halaman, *HTML* juga memiliki peran penting dalam pengembangan antarmuka pengguna (UI) karena menentukan bagaimana informasi ditampilkan kepada user. Dukungan elemen semantik dalam *HTML5* membantu mesin pencari (SEO) serta meningkatkan aksesibilitas bagi pengguna dengan keterbatasan tertentu. Oleh karena itu, *HTML* tidak hanya penting bagi programmer, tetapi juga bagi desainer web yang ingin menyajikan konten secara lebih terstruktur dan ramah pengguna [38].

Dalam dunia pendidikan maupun penelitian, *HTML* banyak dipelajari sebagai

materi dasar pemrograman *web*. Mahasiswa biasanya mengenal *HTML* sebelum mempelajari *CSS* dan *JavaScript* karena sifatnya yang fundamental. Implementasi *HTML* dalam berbagai proyek penelitian *web* juga memperlihatkan peran pentingnya dalam membangun aplikasi berbasis *browser*, baik untuk sistem informasi, *e-learning*, maupun layanan publik. Hal ini menjadikan *HTML* tetap relevan dan terus dipelajari hingga saat ini [39].

2.12 *Cascading Style Sheet (CSS)*

Cascading Style Sheets (CSS) merupakan bahasa *stylesheet* yang digunakan untuk mengendalikan tampilan visual dokumen *HTML*. Dengan *CSS*, pengembang dapat mengatur elemen halaman seperti warna, ukuran huruf, margin, padding, serta tata letak sehingga tampilan menjadi lebih konsisten dan menarik. Perkembangan terbaru menghadirkan fitur seperti *Flexbox*, *Grid Layout*, dan *media query* yang memungkinkan desain web bersifat responsif di berbagai perangkat. *CSS* dengan demikian menjadi fondasi penting dalam pengembangan web modern karena mampu meningkatkan estetika sekaligus fleksibilitas antarmuka [40].

Selain aspek estetika, *CSS* memiliki peran signifikan dalam mendukung pengalaman pengguna (*user experience*) dan aksesibilitas. Penataan visual yang konsisten membantu navigasi dan pemahaman informasi, sementara efek interaktif seperti *hover* atau *transition* meningkatkan keterlibatan pengguna. Optimalisasi *CSS* melalui pemangkasan kode yang tidak terpakai, penggunaan *critical CSS*, serta teknik minifikasi juga berpengaruh pada kecepatan muat halaman dan kinerja situs. Hal ini menjadikan *CSS* tidak hanya alat presentasi, melainkan juga komponen yang berkontribusi pada performa dan kualitas teknis sebuah *website* [41].

Dalam dunia pendidikan dan penelitian, *CSS* diperkenalkan setelah *HTML* sebagai dasar untuk memahami pengembangan web. Mahasiswa maupun peserta pelatihan biasanya mempelajari konsep box model, spesifisitas, dan metodologi penulisan kode seperti BEM sebelum mendalami framework modern seperti *Bootstrap*, *Tailwind*, atau *Materialize*. Implementasi *CSS* dalam proyek pendidikan, pelatihan masyarakat, maupun penelitian sistem informasi menunjukkan peran pentingnya dalam menghasilkan antarmuka yang interaktif, responsif, dan mudah dipelihara. Hal ini memperlihatkan bahwa *CSS* tetap relevan sebagai keterampilan yang harus dikuasai di era transformasi digital [42].

2.13 *Java Script*

Java Script merupakan bahasa pemrograman yang digunakan untuk membuat halaman *web* lebih interaktif dan dinamis. Bahasa ini berfungsi untuk memanipulasi elemen *HTML* serta mengatur gaya melalui *CSS* secara langsung, sehingga pengalaman pengguna menjadi lebih responsif. Seiring perkembangan teknologi, *JavaScript* mendukung paradigma *asynchronous programming* yang memungkinkan pertukaran data dengan *server* tanpa perlu memuat ulang halaman. Hal ini menjadikan *Java Script* sebagai salah satu komponen utama dalam pengembangan aplikasi berbasis *web modern* [43].

Selain digunakan pada sisi klien, *Java Script* kini juga berkembang melalui *runtime environment Node.js*, yang memungkinkan bahasa ini dijalankan di sisi *server*. Perkembangan ini memperluas fungsinya sehingga dapat digunakan untuk pengembangan *full-stack*. Berbagai *framework* populer seperti *React*, *Angular*, dan *Vue* mendukung efisiensi pembangunan antarmuka, sedangkan *Node.js*

memungkinkan integrasi API dan basis data. Hal ini menjadikan *Java Script* bahasa yang serbaguna dalam ekosistem pengembangan *web* [44].

Dalam konteks pendidikan dan penelitian, *Java Script* diperkenalkan setelah *HTML* dan *CSS* sebagai materi lanjutan dalam pemrograman *web*. Mahasiswa dan peserta pelatihan biasanya mempelajari konsep *DOM (Document Object Model)*, *event handling*, serta manipulasi data sebelum beralih ke framework yang lebih kompleks. *Java Script* juga banyak digunakan dalam penelitian terkait aplikasi e-learning, sistem informasi, dan layanan publik berbasis *web* interaktif. Dengan demikian, *Java Script* tetap relevan sebagai keterampilan dasar bagi pengembang di era digital [45].

2.14 Word Embedding

Word Embedding adalah teknik representasi kata sebagai vektor numerik di dalam ruang berdimensi tinggi. Tujuannya adalah mengubah kata dari format diskrit (seperti *one-hot encoding*) menjadi representasi padat yang mampu menangkap hubungan semantik dan sintaksis. Dalam model ini, kata-kata dengan makna serupa akan memiliki posisi yang berdekatan dalam ruang vektor, menjadikannya fondasi penting untuk berbagai tugas *Natural Language Processing (NLP)* seperti analisis sentimen dan klasifikasi teks [46].

Beberapa pendekatan populer untuk membangun *Word Embedding* antara lain *Word2Vec*, *GloVe*, dan *FastText*. *Word2Vec* memiliki dua arsitektur utama: CBOW yang memprediksi kata dari konteksnya, dan Skip-gram yang memprediksi konteks dari sebuah kata. Sementara itu, *GloVe* menggabungkan statistik global dan lokal untuk representasi yang lebih kaya, dan *FastText* unggul dalam menangani kata-

kata di luar kosakata (*out-of-vocabulary*) dengan memanfaatkan informasi level sub-kata (*character n-grams*) [47].

Penerapan *Word Embedding* secara signifikan meningkatkan kemampuan algoritma *machine learning* dalam memproses bahasa manusia. Representasi vektor ini memungkinkan model untuk memahami hubungan semantik antar kata, seperti hubungan antara "raja" dan "ratu", yang sangat berguna dalam tugas NLP seperti penerjemahan mesin dan analisis sentimen. Meskipun model modern seperti Transformer telah menggunakan *embedding* kontekstual yang lebih dinamis, pemahaman dasar *Word Embedding* tetap menjadi pilar fundamental dalam studi NLP [48].

2.15 Confusion Matrix

Confusion matrix digunakan sebagai teknik evaluasi pada penelitian ini. *Confusion matrix* merupakan metode evaluasi yang terdiri dari hasil prediksi oleh sistem dengan hasil yang aktual atau sebenarnya. Tabel 1 merupakan tabel *confusion matrix*.

Tabel 2.1 Confusion Matrix

		Prediksi	
		Negatif	Positif
Aktual	Negatif	<i>True Negative (TN)</i>	<i>False Positive (FP)</i>
	Positif	<i>False Negative (FN)</i>	<i>True Positive (TP)</i>

Keterangan:

- True Negative (TN)* merupakan jumlah kalimat negatif yang dideteksi sebagai kalimat negatif.

- b. *False Negative (FN)* merupakan jumlah kalimat positif yang dideteksi sebagai kalimat negatif.
- c. *False Positive (FP)* merupakan jumlah kalimat negatif yang dideteksi sebagai kalimat positif.
- d. *True Positive (TP)* merupakan jumlah kalimat positif yang dideteksi sebagai kalimat positif.

Berdasarkan Tabel 1, terdapat beberapa metrik yang digunakan untuk menilai kinerja suatu algoritma, di antaranya adalah *accuracy*, *precision*, *recall*, dan *F1 Score*. Dalam *confusion matrix*, *True Positive (TP)* menunjukkan jumlah ulasan positif yang berhasil diklasifikasikan dengan benar sebagai positif. *False Positive (FP)* mengacu pada jumlah ulasan negatif yang keliru diklasifikasikan sebagai positif. Sementara itu, *True Negative (TN)* merepresentasikan jumlah ulasan negatif yang diprediksi dengan benar, dan *False Negative (FN)* adalah jumlah ulasan positif yang salah diklasifikasikan sebagai negatif[49].

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

Recall merupakan perbandingan antara jumlah kalimat positif yang benar dideteksi sebagai kalimat positif dengan jumlah data aktual yang merupakan kalimat positif.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

Precision adalah perbandingan antara jumlah kalimat positif yang benar dideteksi sebagai kalimat positif dengan jumlah kalimat positif yang dideteksi.

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

F1 Score adalah metric pengukuran yang menggabungkan antara *precision* dan *recall*. *F1 Score* menghitung kinerja keseluruhan klasifikasi.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

Accuracy atau akurasi sendiri merupakan perbandingan antara jumlah prediksi yang tepat dengan total keseluruhan data, dan dapat dihitung menggunakan rumus tertentu.

2.16 Penelitian Terkait

Berikut adalah penelitian sebelumnya yang dijadikan acuan dan referensi dalam menyelesaikan masalah ini.

Tabel 2.2 Tabel Penelitian Sebelumnya

No	Nama	Tahun	Judul	Hasil
1	Rani Saputri, Anna Baita[50]	2025	Analisis Sentimen Review Film Avatar 2 pada Platform IMDb Menggunakan LSTM dan GRU	Hasil penelitian menunjukkan bahwa model <i>Long Short Term Memory (LSTM)</i> yang dikombinasikan dengan <i>Gated Recurrent Unit (GRU)</i> mampu menganalisis sentimen ulasan film <i>Avatar: The Way of Water</i> dengan baik, ditunjukkan oleh <i>F1-Score</i> tertinggi sebesar 0,72 pada aspek karakter dalam skenario pembagian data 90:10, serta <i>F1-Score</i> terendah sebesar 0,51 pada aspek editing dalam skenario 80:20; hal ini menegaskan bahwa teknik pembagian data turut memengaruhi performa model dalam mengenali pola

				sentimen secara optimal.
2.	Nur Faizal Basri, Ema Utami[51]	2025	Penerapan Model <i>Word2Vec</i> dan LSTM dalam Analisis Sentimen Ulasan Pengguna <i>Mobile legends</i>	Penelitian ini membuktikan bahwa kombinasi <i>Word2Vec</i> dan LSTM efektif untuk analisis sentimen ulasan pengguna <i>Mobile Legends</i> dengan akurasi 87,88% dan keseimbangan <i>precision</i> serta <i>recall</i> yang baik, mengungguli metode <i>Naive Bayes</i> dan SVM; meskipun begitu, model masih memiliki keterbatasan seperti ketidakseimbangan data dan kesulitan menangani bahasa informal, sehingga disarankan penggunaan model NLP yang lebih canggih seperti BERT dan penambahan kategori sentimen netral untuk hasil yang lebih akurat dan representatif, serta memberikan wawasan bagi pengembang game dalam memahami opini pengguna dan meningkatkan kepuasan pemain.
3.	Fa'iz Yahya, Dian Eka Ratnawati, ,Bayu Rahayudi[52]	2025	Analisis Sentimen Ulasan Pengguna Dari <i>Google Maps</i> Menggunakan Metode <i>Long Short-Term Memory</i> (Studi Kasus: Rumah Sakit Gatoel)	Berdasarkan hasil penelitian, proses analisis dilakukan melalui tahapan <i>scrapping</i> , pelabelan, preprocessing, perancangan arsitektur LSTM, pelatihan, dan evaluasi menggunakan <i>confusion matrix</i> . Model diuji dengan berbagai parameter, dan kinerja terbaik diperoleh tanpa <i>SMOTE</i> , dengan pembagian data 70:30 dan 10 <i>epoch</i> , menghasilkan akurasi 96%,

				<i>precision</i> 97%, <i>recall</i> 98%, dan <i>f-measure</i> 96%. Analisis juga mengungkap permasalahan utama yang berkaitan dengan keterbatasan pendanaan pada beberapa sektor.
4.	Dimas Surya Prasetyo, Karina Auliasari, Yosep Agus Pranoto[53]	2025	Analisis Sentimen Pada Komentar Media Sosial Terkait Isu Joki Dengan Menggunakan Metode LSTM	Penelitian ini menggunakan metode <i>Long Short-Term Memory (LSTM)</i> untuk mengklasifikasikan sentimen positif dan negatif pada 4.055 komentar media sosial, dengan hasil akurasi 74,3%, <i>precision</i> 76,4%, dan <i>recall</i> 81,7%, meskipun masih terdapat kesalahan dalam mendeteksi sentimen negatif. Hasil ini menunjukkan efektivitas LSTM dalam menangani data teks informal dan kompleks, serta relevan bagi institusi pendidikan dan pembuat kebijakan dalam mengevaluasi sistem akademik. Untuk meningkatkan performa, disarankan penerapan tuning hyperparameter, perluasan dataset, atau penggunaan model yang lebih kompleks seperti transformer atau Bi-LSTM, serta eksplorasi pada analisis multi kategori atau konteks non-pendidikan.
5.	Juniardi, Tukino, Bayu Priyatna, Shofa Shofia Hilabi[54]	2025	Klasifikasi Sentimen Analisis Ulasan Aplikasi <i>Alfagift</i> Menggunakan Algoritma <i>Long Short Term Memory</i>	Hasil penelitian menunjukkan bahwa metode LSTM dengan optimasi Adadelta mampu mengklasifikasikan ulasan pengguna aplikasi <i>Alfagift</i> ke dalam kategori transaksi, sistem, promosi, pelayanan, dan keamanan dengan baik. Pengujian menggunakan tiga

				skenario berbeda menunjukkan bahwa <i>learning rate</i>, <i>batch size</i>, dan <i>epoch</i> memengaruhi performa model, di mana skenario kedua menghasilkan performa terbaik dengan akurasi 95%, <i>precision</i> 94%, <i>recall</i> 93%, dan <i>F1-Score</i> 94%, mencerminkan keseimbangan yang baik antara ketepatan dan sensitivitas prediksi.
6.	Diny Wahyuni, Naufal Fadhillah dan Winda Widya Ariestya[55]	2024	Metode <i>Long Short-Term Memory</i> dan <i>Lexicon Based</i> Untuk Analisis Sentimen Ulasan Aplikasi TikTok	Penelitian ini menunjukkan bahwa kombinasi metode <i>Lexicon-Based</i> dan <i>Long Short-Term Memory (LSTM)</i> berhasil mengklasifikasikan sentimen ulasan aplikasi TikTok dengan akurasi 90,05%, <i>precision</i> 92,14%, <i>recall</i> 97,35%, dan <i>F1-Score</i> 98,66%, dengan distribusi ulasan negatif 59,5%, positif 30%, dan netral 10,5%, yang mengindikasikan kecenderungan sentimen negatif pada saat pengambilan data; pengembangan lebih lanjut disarankan melalui implementasi visualisasi hasil klasifikasi secara <i>real-time</i> berbasis <i>website</i>.
7.	Mega Vebika Shyahrin, Yuliant Sibaroni Diyas Puspendari[56]	2023	Penerapan Metode <i>Long Short-Term Memory</i> dan <i>Word2Vec</i> dalam Analisis Sentimen Ulasan pada Aplikasi Ferizy	Penelitian ini melakukan analisis sentimen terhadap 5.000 ulasan aplikasi Ferizy di <i>Google Playstore</i> menggunakan metode klasifikasi LSTM dan dua variasi ekstraksi fitur <i>Word2Vec</i>, yaitu skip-gram dan CBOW. Hasil pengujian menunjukkan bahwa variasi skip-gram menghasilkan

				akurasi lebih tinggi sebesar 88,20% dibanding CBOW yang hanya mencapai 74,20%. Temuan ini mendukung penggunaan <i>skip-gram</i> untuk analisis sentimen, dan untuk penelitian selanjutnya disarankan menggunakan lebih banyak data serta menambahkan proses normalisasi pada tahap pra-pemrosesan agar hasil lebih optimal.
8.	Iin Kurniasari, Achmad Arif Alfin, Eko Widodo[57]	2023	Implementasi <i>Long Short-Term Memory</i> (LSTM) dan <i>Word Embedding</i> Model pada Analisis Sentimen Layanan Uang Elektronik Ovo dan Link Aja	Studi ini memanfaatkan teknologi LSTM dan <i>word embedding</i> untuk menganalisis sentimen pelanggan terhadap layanan uang elektronik OVO dan LinkAja, yang hasilnya menunjukkan dominasi ulasan negatif pada kedua aplikasi, dengan OVO memperoleh 91% ulasan negatif dari 1.720 data dan LinkAja sebesar 82% dari 1.580 data, terutama pada kategori transaksi dan akun. Akurasi model untuk OVO mencapai 83% dengan hasil validasi dan testing sebesar 79%, sementara LinkAja memiliki akurasi 84%, validasi 79%, dan testing 76%, dengan nilai <i>precision</i> , <i>recall</i> , dan <i>F1-Score</i> berkisar antara 67%–84%. Temuan ini dapat menjadi masukan berharga bagi penyedia layanan untuk memahami keluhan pelanggan dan meningkatkan kualitas layanan.

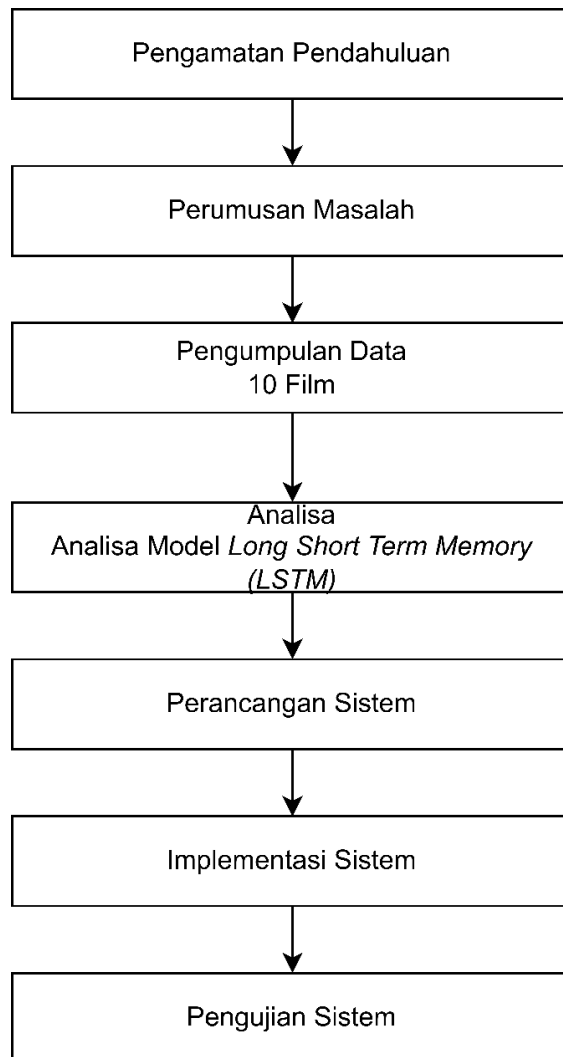
9.	M. Aasya Aldin Islamy, Indriati, Putra Pandu Adikara[58]	2022	Analisis Sentimen IMDB <i>Movie Reviews</i> menggunakan <i>Metode Long Short Term Memory</i> dan <i>FastText</i>	Penelitian ini menyimpulkan bahwa kombinasi metode LSTM dan <i>FastText</i> menghasilkan performa terbaik dalam analisis sentimen dengan konfigurasi 2 layer LSTM, hidden size 64, optimizer Adam, learning rate 0,001, dan 30 epoch, menghasilkan akurasi 85,9%, precision 84,7%, <i>recall</i> 87,5%, dan <i>F1-score</i> 86,3%. Metode ini terbukti lebih unggul dibanding tanpa <i>FastText</i> , dengan selisih <i>F1-score</i> sebesar 0,053, karena kemampuan <i>FastText</i> merepresentasikan kata-kata yang belum pernah muncul saat pelatihan.
10.	Sharfina Febbi Handayani, Riszki Wijayatun Pratiwi, Dairoh, Dwi Intan Af'idah[58]	2022	Analisis Sentimen pada Data Ulasan <i>Twitter</i> dengan <i>Long- Short Term Memory</i>	Berdasarkan hasil pengujian, penggunaan metode LSTM untuk klasifikasi <i>teks</i> ulasan <i>Twitter</i> berbahasa Indonesia dengan parameter <i>tuning Word2Vec</i> (CBOW), 150 <i>neuron</i> , 30 <i>epoch</i> , dan fungsi aktivasi <i>softmax</i> menghasilkan akurasi sebesar 57,35%, yang selaras dengan hasil penelitian sebelumnya pada topik dan dataset berbeda yang juga menggunakan LSTM.

BAB III

METODOLOGI PENELITIAN

Penelitian ini dilaksanakan dengan mengikuti tahapan demi tahapan yang saling berkaitan. Setiap tahapan dijelaskan dalam metode penelitian yang disusun secara sistematis, terstruktur, dan ditampilkan dalam bentuk skema yang jelas.

Berikut tahapan-tahapan penelitian dapat dilihat pada Gambar 3.1 dibawah:



Gambar 3.1 Tahapan Penelitian

Penjelasan dari tahapan-tahapan penelitian pada Gambar 3.1 dapat dilihat pada penjelasan dibawah ini:

3.1 Pengamatan Pendahuluan

Tahapan awal dalam penelitian ini adalah pengamatan pendahuluan yang bertujuan untuk memahami permasalahan yang ada dalam klasifikasi sentimen pada ulasan film. Dalam pengamatan ini dilakukan studi literatur terhadap berbagai penelitian sebelumnya yang menggunakan pendekatan *deep learning*, khususnya *Long Short-Term Memory (LSTM)*, dalam menganalisis data teks. Pengamatan ini menunjukkan bahwa metode LSTM menunjukkan hasil yang lebih baik dalam menangani data sekuensial seperti teks ulasan dibandingkan metode tradisional.

3.2 Perumusan Masalah

Berdasarkan pengamatan pendahuluan, dirumuskan permasalahan utama dalam penelitian ini, yaitu bagaimana membangun sistem deteksi sentimen ulasan film menggunakan model *Long Short-Term Memory (LSTM)* berbasis *deep learning* agar dapat menghasilkan klasifikasi sentimen yang akurat dan efisien.

3.3 Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa ulasan film berbahasa Indonesia yang dikumpulkan dari platform *Imdb*. Data ulasan ini mencakup beragam genre film untuk menjamin keragaman dan representasi emosi yang lebih luas. Ulasan tersebut kemudian diklasifikasikan ke dalam tiga kategori sentimen: positif, negatif, dan netral sebagai label utama dalam proses pelatihan dan pengujian model. Proses pelatihan menggunakan data latih sebesar 80%, sedangkan sisanya

20% digunakan sebagai data uji.

3.4 Analisa

Tahapan selanjutnya adalah melakukan analisa metode sistem dari penelitian proposal skripsi ini. Adapun tahapan analisa dalam penelitian skripsi ini adalah sebagai berikut:

3.4.1 Analisa *Model Long Short-Term Memory (LSTM)*

Analisis ini melibatkan pemahaman karakteristik dan arsitektur dari model *LSTM*. Model *LSTM* digunakan karena kemampuannya dalam mengingat informasi dalam jangka panjang dan pendek secara bersamaan dalam satu urutan data. Model ini sangat efektif dalam memahami konteks kata dalam ulasan film yang panjang.

Langkah-langkah dalam analisis mencakup penyesuaian parameter seperti jumlah unit *LSTM*, fungsi aktivasi, *optimizer*, dan *epoch*. Selain itu, dipertimbangkan juga proses tokenisasi dan *padding* untuk menyiapkan data teks agar sesuai untuk diproses oleh model.

3.5 Perancangan Sistem

Perancangan sistem dilakukan dengan membangun arsitektur sistem klasifikasi sentimen berbasis model *LSTM*. Proses perancangan mencakup tahapan-tahapan seperti pengumpulan dan pelabelan data ulasan film, pra-pemrosesan data (meliputi tokenisasi, penghapusan stopword, stemming, dan padding), pembangunan arsitektur jaringan *LSTM*, serta pengaturan parameter pelatihan seperti jumlah epoch, batch size, dan learning rate. Tahapan ini merupakan dasar sebelum model diimplementasikan dan diuji lebih lanjut. Sistem ini dirancang sebagai aplikasi yang

dapat diakses secara terbuka oleh publik, sehingga pengguna tidak memerlukan proses autentikasi seperti login untuk dapat melakukan analisis sentimen.

3.6 Implementasi Sistem

Beberapa komponen pendukung yang berperan penting dalam implementasi sistem diantaranya adalah perangkat keras (*hardware*) dan perangkat lunak (*software*). Adapun spesifikasi dari perangkat keras dan perangkat lunak yang digunakan sebagai berikut:

1. Perangkat keras (*hardware*), antara lain:

Processor : 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz (4 CPUs), ~3.0GHz

Memory : 8192MB RAM

System type : Windows 11 Home Single Language 64-bit (10.0, Build 26100)

Harddisk : 500 GB

2. Perangkat lunak (*software*), antara lain:

Sistem operasi : *Windows 11*

Tools : *Visual Studio Code, Web Browser Google Chrome*

3.7 Pengujian Sistem

Pengujian dilakukan untuk memastikan model dalam kondisi beragam. Data uji dipisahkan dari dataset dengan rasio 80% *training* dan 20% *validation*. Pengujian ini mengevaluasi akurasi model terhadap data yang belum pernah dilatih.