BAB 2
LANDASAN TEORI

Bab ini membahas teori-teori yang digunakan sebagai dasar dalam penelitian mengenai analisis sentimen pengaruh *influencer* terhadap produk yang dipromosikan di X menggunakan metode *Long Short-Term Memory* (LSTM). Pembahasan dalam bab ini meliputi pemasaran melalui *influencer*, media sosial X, analisis sentimen, penambangan data, penambangan teks, *crawling data*, pra-pemrosesan teks, *labeling*, tokenisasi, *padding*, *word embedding*, RNN, LSTM, Bi-LSTM, pembagian *dataset*, *oversampling*, evaluasi model, dan penelitian terdahulu. Seluruh teori tersebut digunakan untuk mendukung proses analisis sentimen berbasis teks yang dilakukan pada bab metodologi dan hasil penelitian.

2.1 Pemasaran melalui influencer

Pemasaran melalui *influencer* merupakan strategi pemasaran digital yang memanfaatkan figur tertentu di media sosial untuk memperkenalkan, menjelaskan, atau merekomendasikan produk kepada audiens. Dalam strategi ini, *influencer* berperan sebagai perantara antara pelaku usaha dan calon konsumen melalui konten yang dipublikasikan. Konten digital yang disampaikan secara tepat dapat memengaruhi minat audiens terhadap suatu produk karena audiens memperoleh informasi melalui media yang dekat dengan aktivitas mereka sehari-hari [7]. Oleh karena itu, pemasaran melalui *influencer* menjadi salah satu strategi yang relevan dalam aktivitas promosi digital.

Penggunaan *influencer* dalam pemasaran tidak hanya berkaitan dengan penyebaran informasi produk, tetapi juga dengan pembentukan persepsi audiens. Ketika *influencer* menyampaikan pengalaman penggunaan, ulasan, atau rekomendasi, audiens dapat memberikan respons yang berbeda-beda. Respons tersebut dapat berupa ketertarikan, dukungan, kepercayaan, kritik, ataupun penolakan terhadap produk yang dipromosikan. Respons publik inilah yang dapat dianalisis untuk mengetahui kecenderungan penerimaan masyarakat terhadap suatu produk.

Dalam penelitian ini, pemasaran melalui *influencer* digunakan sebagai konteks utama karena opini publik terhadap produk yang dipromosikan menjadi objek analisis sentimen. Pengaruh *influencer* tidak diukur melalui data transaksi

atau angka penjualan secara langsung. Pengaruh yang dimaksud dibatasi pada kecenderungan sentimen publik terhadap produk yang dipromosikan. Dengan demikian, analisis dilakukan berdasarkan opini pengguna X yang berkaitan dengan produk, promosi, ulasan, rekomendasi, dan *endorsement*.

2.2 Media sosial X

Media sosial merupakan platform digital yang memungkinkan pengguna untuk membuat, membagikan, dan menanggapi berbagai bentuk konten. Melalui media sosial, pengguna dapat menyampaikan pengalaman, penilaian, kritik, rekomendasi, maupun opini terhadap suatu produk. Karakteristik tersebut membuat media sosial menjadi salah satu sumber data yang relevan dalam penelitian analisis opini publik. Pada penelitian ini, media sosial yang digunakan sebagai sumber data adalah X.

X merupakan salah satu media sosial berbasis teks yang banyak digunakan untuk menyampaikan pendapat secara cepat, singkat, dan terbuka. Dalam penelitian ini, penyebutan X mengacu pada platform yang sebelumnya telah dijelaskan pada Bab 1 sebagai media sosial yang dahulu dikenal sebagai Twitter. Data pada X umumnya berbentuk teks sehingga dapat digunakan dalam penelitian berbasis *text mining* dan analisis sentimen. Penelitian terkait analisis sentimen pada data media sosial menunjukkan bahwa teks pendek dapat dimanfaatkan untuk memahami kecenderungan opini publik melalui pendekatan komputasional [8].

Teks pada X memiliki karakteristik yang ringkas, spontan, dan tidak selalu menggunakan bahasa baku. Karakteristik tersebut membuat data X sesuai untuk digunakan dalam penelitian sentimen, terutama ketika data dikumpulkan berdasarkan kata kunci, bahasa, dan rentang waktu tertentu. Pada penelitian ini, X digunakan sebagai sumber data utama karena teks yang dikumpulkan dapat mencerminkan opini masyarakat terhadap produk yang dikaitkan dengan aktivitas promosi *influencer*. Data tersebut kemudian diproses untuk mengetahui kecenderungan sentimen positif dan negatif.

2.3 Analisis sentimen

Analisis sentimen merupakan proses pengolahan data teks untuk mengetahui kecenderungan opini yang terkandung dalam suatu pernyataan. Analisis ini dapat digunakan untuk mengelompokkan teks ke dalam kategori positif, negatif, atau

netral. Dalam kajian yang lebih luas, analisis sentimen juga dapat dikembangkan menjadi analisis berbasis aspek untuk mengetahui opini terhadap bagian tertentu dari suatu objek [9]. Dengan demikian, analisis sentimen dapat digunakan untuk memahami persepsi publik terhadap suatu topik atau produk.

Pada penelitian ini, analisis sentimen digunakan untuk mengelompokkan opini publik terhadap produk yang dipromosikan oleh *influencer*. Kelas sentimen yang digunakan hanya terdiri dari dua kategori, yaitu positif dan negatif. Sentimen positif menunjukkan adanya dukungan, ketertarikan, kepuasan, atau penilaian baik terhadap produk. Sebaliknya, sentimen negatif menunjukkan adanya kritik, keluhan, ketidakpuasan, atau penilaian buruk terhadap produk maupun promosi yang dilakukan.

Analisis sentimen pada data X dapat dilakukan dengan berbagai pendekatan, mulai dari metode sederhana hingga metode berbasis *deep learning*. Penggunaan pendekatan jaringan saraf pada data teks dapat membantu model memahami pola opini dalam teks pendek yang terdapat pada media sosial [8]. Dalam penelitian ini, pendekatan yang digunakan adalah LSTM karena metode tersebut mampu memproses data berurutan. Pendekatan ini sesuai dengan karakteristik data teks yang memiliki susunan kata dan konteks.

2.4 Penambangan data

Penambangan data atau *data mining* merupakan proses untuk menemukan pola, hubungan, atau informasi penting dari kumpulan data. Proses ini digunakan ketika data yang tersedia memiliki jumlah besar dan memerlukan pengolahan lebih lanjut agar dapat menghasilkan informasi yang bermanfaat. Dalam proses *data mining*, data dapat melalui tahapan pemilihan, pembersihan, transformasi, pemodelan, evaluasi, dan interpretasi hasil [10]. Tahapan tersebut membantu data mentah diubah menjadi informasi yang dapat digunakan dalam pengambilan kesimpulan.

Dalam penelitian ini, *data mining* digunakan untuk membantu menemukan pola sentimen dari kumpulan data teks. Data yang semula berbentuk teks tidak terstruktur perlu diolah agar dapat digunakan dalam proses klasifikasi. Melalui proses tersebut, opini publik yang tersebar di X dapat diubah menjadi informasi yang lebih terarah. Informasi tersebut kemudian digunakan untuk menggambarkan kecenderungan pengaruh *influencer* terhadap produk yang dipromosikan.

2.5 Penambangan teks

Penambangan teks atau *text mining* merupakan proses pengolahan data berbentuk teks untuk memperoleh informasi yang dapat dianalisis. Data teks memiliki karakteristik yang lebih bebas dibandingkan data numerik karena tidak selalu memiliki struktur yang seragam. Data media sosial sering mengandung singkatan, kata tidak baku, simbol, tautan, *hashtag*, *mention*, dan emotikon. Oleh karena itu, data teks perlu melalui beberapa tahap pengolahan sebelum dapat digunakan dalam pemodelan.

Text mining digunakan untuk mengolah data teks tidak terstruktur agar informasi di dalamnya dapat diproses dan dianalisis secara komputasional [11]. Dalam penelitian ini, *text mining* digunakan untuk mengolah data berbahasa Indonesia sebelum masuk ke tahap klasifikasi sentimen. Tahapan yang dilakukan meliputi pembersihan teks, pemberian label, tokenisasi, representasi numerik, dan klasifikasi menggunakan model LSTM. Dengan proses tersebut, data opini publik dapat dianalisis secara lebih sistematis.

2.6 Pengumpulan data melalui crawling

Crawling data merupakan proses pengambilan data secara otomatis dari sumber digital berdasarkan kriteria tertentu. Kriteria tersebut dapat berupa kata kunci, bahasa, periode waktu, atau topik tertentu. Teknik ini sering digunakan untuk mengumpulkan data dari internet atau media sosial dalam jumlah besar. Dalam penelitian berbasis media sosial, *crawling* membantu memperoleh data yang relevan dengan topik penelitian.

Web crawling merupakan proses pengambilan halaman atau data dari sumber digital secara otomatis berdasarkan aturan tertentu [12]. Dalam penelitian ini, proses *crawling* dilakukan untuk memperoleh data berbahasa Indonesia yang berkaitan dengan *influencer*, produk, promosi, *review*, rekomendasi, dan *endorsement*. Data dikumpulkan berdasarkan rentang tahun 2020 sampai 2024. Data hasil *crawling* tersebut kemudian digabungkan dan diproses pada tahap berikutnya.

2.7 Pra-pemrosesan teks

Pra-pemrosesan teks atau *text preprocessing* merupakan tahap awal dalam pengolahan data teks yang bertujuan untuk membersihkan dan menyiapkan data

sebelum digunakan dalam proses pemodelan. Data dari X biasanya masih mengandung elemen yang tidak diperlukan, seperti URL, simbol, angka, tanda baca, *mention*, *hashtag*, serta bentuk kata yang tidak seragam. Tahap *preprocessing* diperlukan agar data menjadi lebih bersih, konsisten, dan lebih mudah diproses oleh model [13]. Dalam penelitian ini, hasil pra-pemrosesan digunakan sebagai *input* utama sebelum data diubah menjadi bentuk numerik.

Tahapan *text preprocessing* dalam penelitian ini dilakukan secara berurutan agar data teks memiliki bentuk yang lebih seragam. Proses pertama adalah *case folding*, yaitu mengubah seluruh huruf menjadi huruf kecil. Proses kedua adalah *cleaning*, yaitu membersihkan teks dari URL, simbol, angka yang tidak relevan, tanda baca tertentu, *mention*, dan *hashtag*. Proses ini dilakukan agar elemen yang tidak berkontribusi terhadap analisis sentimen dapat dikurangi.

Tahapan berikutnya adalah normalisasi kata, tokenisasi, *stopword removal*, dan *stemming*. Normalisasi kata digunakan untuk mengubah kata tidak baku atau singkatan menjadi bentuk yang lebih standar. Tokenisasi digunakan untuk memecah kalimat menjadi unit kata atau *token* agar dapat diproses secara terpisah. Setelah itu, *stopword removal* digunakan untuk menghapus kata umum yang tidak memiliki kontribusi besar terhadap makna sentimen, sedangkan *stemming* digunakan untuk mengubah kata berimbuhan menjadi bentuk dasar.

Setelah seluruh tahapan *preprocessing* dilakukan, data teks menjadi lebih terstruktur dan siap diproses pada tahap berikutnya. Hasil akhir dari tahap ini digunakan dalam bentuk kolom `stemmed_text`. Kolom tersebut kemudian digunakan sebagai masukan untuk proses tokenisasi lanjutan, *padding*, dan *word embedding*. Dengan demikian, *preprocessing* berperan penting dalam membentuk kualitas data sebelum pelatihan model.

2.8 Pelabelan data

Pelabelan data atau *labeling* merupakan proses pemberian kategori atau kelas pada data. Dalam analisis sentimen, label digunakan untuk menunjukkan kecenderungan opini yang terkandung dalam teks. Label tersebut menjadi acuan bagi model untuk mempelajari hubungan antara teks dan kategori sentimen. Dengan adanya label, model dapat dilatih untuk membedakan pola teks yang mengarah pada sentimen positif dan negatif.

Pada penelitian ini, label sentimen terdiri dari dua kelas, yaitu positif dan negatif. Label positif diberikan pada data yang mengandung dukungan,

ketertarikan, kepuasan, atau penilaian baik terhadap produk maupun promosi. Label negatif diberikan pada data yang berisi kritik, keluhan, ketidakpuasan, atau penilaian buruk. Label teks disimpan pada kolom `sentiment`, sedangkan label numerik disimpan pada kolom `sentiment_binary`.

2.9 Tokenisasi dan padding

Tokenisasi merupakan proses mengubah teks menjadi urutan *token* atau indeks numerik berdasarkan kosakata yang dibentuk dari data latih. Setiap kata yang muncul dalam data diberi indeks tertentu sehingga teks dapat direpresentasikan sebagai urutan angka. Proses ini diperlukan karena model LSTM tidak dapat memproses teks mentah secara langsung. Dengan tokenisasi, setiap kalimat dapat diubah menjadi bentuk numerik yang dapat digunakan sebagai masukan model.

Padding merupakan proses menyamakan panjang urutan *token* pada setiap data teks. Dalam penelitian ini, setiap teks memiliki panjang yang berbeda sehingga perlu disamakan agar dapat diproses dalam bentuk matriks. Apabila panjang teks lebih pendek dari nilai maksimum, nilai nol ditambahkan hingga panjangnya sama. Apabila panjang teks melebihi nilai maksimum, teks dipotong sesuai batas panjang yang telah ditentukan.

2.10 Representasi kata

Word embedding merupakan teknik untuk mengubah kata menjadi representasi numerik dalam bentuk vektor. Teknik ini diperlukan karena model *deep learning* tidak dapat memproses teks mentah secara langsung. Dengan *word embedding*, kata-kata dalam teks dapat direpresentasikan sebagai angka yang memiliki nilai vektor tertentu. Representasi ini membuat model dapat mempelajari hubungan antar kata berdasarkan pola kemunculannya.

Cara kerja *word embedding* dimulai dari hasil tokenisasi yang mengubah kata menjadi indeks numerik. Setiap indeks kemudian dipetakan ke dalam matriks *embedding* yang berisi vektor berdimensi tertentu. Wang dkk. menjelaskan bahwa *word embedding* dapat digunakan untuk merepresentasikan hubungan antar kata dalam ruang vektor [14]. Dalam penelitian ini, *word embedding* dilakukan melalui lapisan `Embedding` dengan dimensi 128.

Rumus pemetaan indeks kata ke vektor *embedding* digunakan untuk

menjelaskan proses perubahan indeks numerik menjadi vektor. Pada rumus tersebut, $\mathbf{e}_i$ merupakan vektor *embedding* untuk kata ke-i, $\mathbf{W}_e$ merupakan matriks *embedding*, dan x_i merupakan indeks kata hasil tokenisasi. Pemetaan ini membuat setiap indeks kata memiliki representasi numerik yang dapat digunakan sebagai masukan model. Rumus pemetaan indeks kata ke vektor *embedding* ditunjukkan pada Rumus 2.1.

$$\mathbf{e}_i = \mathbf{W}_e[x_i] \tag{2.1}$$

Dalam model LSTM, hasil *word embedding* menjadi masukan awal sebelum data diproses oleh lapisan LSTM. Lapisan *embedding* membantu model memahami pola kata berdasarkan bobot vektor yang dipelajari selama pelatihan. Penggunaan representasi kata bersama model LSTM telah digunakan dalam penelitian analisis sentimen berbasis teks [15]. Oleh karena itu, *word embedding* menjadi bagian penting dalam arsitektur model yang digunakan dalam penelitian ini.

2.11 Klasifikasi

Klasifikasi merupakan proses pengelompokan data ke dalam kelas tertentu berdasarkan pola yang telah dipelajari dari data latih. Dalam penelitian ini, klasifikasi digunakan untuk menentukan apakah suatu data teks termasuk ke dalam sentimen positif atau sentimen negatif. Model mempelajari hubungan antara teks hasil *preprocessing* dan label sentimen. Setelah proses pelatihan selesai, model digunakan untuk memprediksi kelas sentimen pada data uji.

Klasifikasi sentimen dapat digunakan untuk mengelompokkan opini masyarakat terhadap suatu topik tertentu. Pada penelitian berbasis LSTM, proses klasifikasi dilakukan dengan mengubah teks menjadi representasi numerik, mempelajari pola urutan kata, dan menghasilkan kelas prediksi berdasarkan probabilitas keluaran model [16]. Dalam penelitian ini, klasifikasi dilakukan menggunakan metode LSTM dengan masukan berupa teks hasil *preprocessing*. Sebelum masuk ke model, teks terlebih dahulu melalui tokenisasi, *padding*, dan *word embedding*.

2.12 Kecerdasan buatan

Kecerdasan buatan atau *Artificial Intelligence* merupakan bidang ilmu yang berfokus pada pengembangan sistem yang dapat melakukan tugas tertentu secara cerdas. Tugas tersebut dapat berupa pengenalan pola, pengambilan keputusan, pemrosesan bahasa alami, prediksi, dan klasifikasi. Dalam konteks penelitian ini, kecerdasan buatan menjadi dasar konseptual dari proses klasifikasi sentimen. Model yang dibangun diarahkan untuk mengenali pola teks dan menghasilkan prediksi sentimen.

Thurzo dkk. menjelaskan bahwa kecerdasan buatan telah diterapkan dalam berbagai bidang karena kemampuannya dalam membantu proses analisis dan pengambilan keputusan berbasis data [17]. Dalam penelitian ini, kecerdasan buatan digunakan dalam bentuk model pembelajaran yang mampu memproses data teks. Model tersebut mempelajari hubungan antara opini pengguna dan label sentimen. Hasil pembelajaran digunakan untuk mengklasifikasikan opini ke dalam kelas positif atau negatif.

2.13 Machine learning

Machine learning merupakan cabang dari kecerdasan buatan yang memungkinkan sistem belajar dari data. Melalui proses pelatihan, model dapat mengenali pola tertentu dan menggunakan pola tersebut untuk melakukan prediksi atau klasifikasi terhadap data baru. Proses pembelajaran dilakukan berdasarkan contoh data yang telah memiliki label. Dengan demikian, model dapat mempelajari hubungan antara fitur masukan dan target keluaran.

Machine learning memungkinkan sistem mempelajari pola dari data dan menggunakan pola tersebut untuk melakukan prediksi atau klasifikasi terhadap data baru [18]. Dalam penelitian ini, konsep *machine learning* digunakan sebagai dasar dalam proses klasifikasi sentimen. Model mempelajari hubungan antara teks dan label sentimen. Setelah pelatihan dilakukan, model menghasilkan prediksi terhadap data uji yang belum digunakan dalam proses pelatihan.

2.14 Deep learning

Deep learning merupakan pengembangan dari *machine learning* yang menggunakan jaringan saraf tiruan dengan beberapa lapisan. Pendekatan ini mampu mempelajari pola yang lebih kompleks, terutama pada data tidak terstruktur

seperti teks, gambar, dan suara. Pada data teks, *deep learning* dapat digunakan untuk mempelajari pola urutan kata dan konteks kalimat. Hal ini membuat *deep learning* relevan digunakan dalam penelitian analisis sentimen.

Deep learning memiliki kemampuan untuk mempelajari representasi data secara lebih mendalam dibandingkan metode *machine learning* konvensional [19]. Penelitian lain juga menunjukkan bahwa pendekatan *deep learning* dapat diterapkan dalam klasifikasi data dan pengenalan pola pada berbagai bidang [20]. Dalam penelitian ini, *deep learning* digunakan karena data yang dianalisis berbentuk teks dan memiliki urutan kata. Metode yang digunakan adalah LSTM karena metode tersebut mampu mempelajari pola sekuensial dalam data teks.

2.15 Recurrent Neural Network

Recurrent Neural Network (RNN) merupakan jenis jaringan saraf tiruan yang dirancang untuk memproses data berurutan. Data berurutan merupakan data yang memiliki keterkaitan antar elemen berdasarkan susunan tertentu, seperti teks, suara, atau deret waktu. Pada data teks, urutan kata memiliki peran penting karena perubahan susunan kata dapat memengaruhi makna kalimat. Oleh karena itu, RNN dapat digunakan untuk mempelajari pola urutan dalam data teks.

Cara kerja RNN dilakukan dengan memproses data secara berurutan dari satu waktu ke waktu berikutnya. Pada setiap langkah waktu, RNN menerima masukan saat ini dan menggabungkannya dengan informasi dari keadaan tersembunyi sebelumnya. Rumus keadaan tersembunyi RNN digunakan untuk menjelaskan hubungan antara masukan saat ini, keadaan tersembunyi sebelumnya, bobot, bias, dan fungsi aktivasi. Rumus keadaan tersembunyi RNN ditunjukkan pada Rumus 2.2.

$$\mathbf{h}_t = \phi(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \tag{2.2}$$

Pada Rumus 2.2, $\mathbf{h}_t$ merupakan keadaan tersembunyi pada waktu ke-t, $\mathbf{x}_t$ merupakan masukan pada waktu ke-t, dan $\mathbf{h}_{t-1}$ merupakan keadaan tersembunyi sebelumnya. Simbol $\mathbf{W}$ merupakan bobot, $\mathbf{b}$ merupakan bias, dan ϕ merupakan fungsi aktivasi. Dengan mekanisme tersebut, RNN dapat membawa informasi dari langkah sebelumnya ke langkah berikutnya. Hal ini membuat RNN dapat digunakan untuk memproses data yang memiliki urutan.

RNN memiliki keterbatasan dalam mempelajari ketergantungan jangka

panjang. Ketika urutan data terlalu panjang, informasi dari langkah awal dapat melemah selama proses pelatihan. Masalah tersebut dikenal sebagai *vanishing gradient*, yaitu kondisi ketika nilai gradien menjadi sangat kecil sehingga pembaruan bobot pada bagian awal jaringan menjadi tidak efektif [5]. Keterbatasan ini menjadi salah satu alasan penggunaan LSTM sebagai pengembangan dari RNN.

2.16 Long Short-Term Memory

Long Short-Term Memory (LSTM) merupakan pengembangan dari RNN yang dirancang untuk menangani permasalahan ketergantungan jangka panjang pada data berurutan. LSTM memiliki mekanisme *gate* yang berfungsi untuk mengatur informasi yang disimpan, dilupakan, dan dikeluarkan oleh model. Mekanisme tersebut memungkinkan LSTM mempertahankan informasi penting dalam urutan data yang lebih panjang [4]. Oleh karena itu, LSTM banyak digunakan dalam pemrosesan bahasa alami dan analisis sentimen.

LSTM memiliki beberapa komponen utama, yaitu *forget gate*, *input gate*, kandidat memori, *cell state*, *output gate*, dan *hidden state*. Setiap komponen memiliki fungsi berbeda dalam mengatur aliran informasi. Fungsi aktivasi *sigmoid* digunakan untuk menghasilkan nilai antara 0 dan 1, sedangkan fungsi aktivasi tanh digunakan untuk menghasilkan nilai antara -1 dan 1. Rumus fungsi *sigmoid* dan tanh ditunjukkan pada Rumus 2.3 dan Rumus 2.4.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.3}$$

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.4}$$

Forget gate berfungsi menentukan informasi lama yang perlu dipertahankan atau dilupakan dari *cell state* sebelumnya. Nilai yang dihasilkan oleh *forget gate* berada pada rentang 0 sampai 1, sehingga informasi dapat dikurangi atau dipertahankan sesuai kebutuhan model. Rumus *forget gate* digunakan untuk menjelaskan proses seleksi informasi lama dalam LSTM. Rumus *forget gate* ditunjukkan pada Rumus 2.5.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{2.5}$$

Pada Rumus 2.5, $\mathbf{f}_t$ merupakan nilai *forget gate*, $\mathbf{h}_{t-1}$ merupakan *hidden state* sebelumnya, dan $\mathbf{x}_t$ merupakan masukan saat ini. Simbol $\mathbf{W}_f$ merupakan bobot, sedangkan $\mathbf{b}_f$ merupakan bias. Nilai yang dihasilkan oleh *forget gate* digunakan untuk mengontrol informasi lama yang tetap dibawa ke tahap berikutnya. Dengan mekanisme tersebut, LSTM dapat mengurangi informasi yang dianggap tidak relevan.

Input gate berfungsi menentukan informasi baru yang akan ditambahkan ke dalam *cell state*. Kandidat nilai memori baru dihitung menggunakan fungsi tanh sebelum digabungkan dengan informasi yang dipilih oleh *input gate*. Rumus *input gate* dan kandidat memori digunakan untuk menjelaskan proses pembaruan informasi baru pada LSTM. Rumus *input gate* dan kandidat memori ditunjukkan pada Rumus 2.6 dan Rumus 2.7.

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{2.6}$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \tag{2.7}$$

Cell state merupakan jalur memori utama dalam LSTM. Nilai *cell state* diperbarui dengan menggabungkan informasi lama yang dipertahankan oleh *forget gate* dan informasi baru yang ditambahkan melalui *input gate*. Rumus pembaruan *cell state* digunakan untuk menjelaskan proses penggabungan informasi lama dan informasi baru dalam memori LSTM. Rumus pembaruan *cell state* ditunjukkan pada Rumus 2.8.

$$\mathbf{C}_t = \mathbf{f}_t * \mathbf{C}_{t-1} + \mathbf{i}_t * \tilde{\mathbf{C}}_t \tag{2.8}$$

Pada Rumus 2.8, $\mathbf{C}_t$ merupakan *cell state* pada waktu ke-t, sedangkan $\mathbf{C}_{t-1}$ merupakan *cell state* sebelumnya. Simbol $\mathbf{f}_t$ digunakan untuk mengatur informasi lama yang dipertahankan, sedangkan $\mathbf{i}_t$ digunakan untuk mengatur informasi baru yang dimasukkan. Komponen ini membantu LSTM mempertahankan informasi penting dalam urutan teks yang lebih panjang. Oleh karena itu, *cell state* menjadi bagian utama dalam mekanisme memori LSTM.

Output gate berfungsi menentukan bagian informasi dari *cell state* yang akan dikeluarkan sebagai *hidden state*. Setelah *output gate* dihitung, nilai *hidden state* diperoleh dari hasil kombinasi *output gate* dan *cell state* yang telah melalui fungsi tanh. Rumus *output gate* dan *hidden state* digunakan untuk menjelaskan keluaran akhir pada setiap langkah waktu LSTM. Rumus *output gate* dan *hidden state* ditunjukkan pada Rumus 2.9 dan Rumus 2.10.

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{2.9}$$

$$\mathbf{h}_t = \mathbf{o}_t * \tanh(\mathbf{C}_t) \tag{2.10}$$

Dalam penelitian analisis sentimen media sosial, LSTM telah digunakan pada berbagai objek penelitian. Cahyadi dkk. menggunakan RNN dengan LSTM untuk analisis sentimen data Instagram [16]. Chakraborty dkk. menerapkan model LSTM untuk analisis sentimen pada data *tweet* COVID-19 [15]. Berdasarkan penelitian tersebut, LSTM dinilai sesuai untuk klasifikasi sentimen pada data teks.

2.17 Bidirectional Long Short-Term Memory

Bidirectional Long Short-Term Memory atau Bi-LSTM merupakan pengembangan dari LSTM yang memproses data dari dua arah, yaitu dari depan ke belakang dan dari belakang ke depan. Dengan pendekatan ini, model dapat memahami konteks kata berdasarkan informasi sebelum dan sesudah kata tersebut. Pendekatan dua arah dapat membantu model memperoleh representasi konteks yang lebih lengkap pada data teks [21]. Oleh karena itu, Bi-LSTM sering digunakan dalam tugas pemrosesan bahasa alami.

Rumus representasi Bi-LSTM digunakan untuk menjelaskan proses penggabungan *hidden state* dari arah maju dan arah mundur. Pada rumus tersebut, $\overrightarrow{\mathbf{h}}_t$ merupakan *hidden state* dari arah maju, $\overleftarrow{\mathbf{h}}_t$ merupakan *hidden state* dari arah mundur, dan $\mathbf{h}_t$ merupakan gabungan keduanya. Gabungan informasi dari dua arah tersebut kemudian digunakan sebagai representasi urutan teks. Rumus representasi Bi-LSTM ditunjukkan pada Rumus 2.11.

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t] \tag{2.11}$$

Dalam penelitian analisis sentimen, Alghifari dkk. menggunakan Bi-LSTM untuk menganalisis sentimen terhadap layanan Grab Indonesia [22]. Pada penelitian ini, Bi-LSTM digunakan sebagai salah satu model pembanding terhadap LSTM 1 *layer* dan LSTM 2 *layer*. Perbandingan dilakukan untuk mengetahui arsitektur yang menghasilkan performa klasifikasi terbaik. Hasil perbandingan tersebut kemudian digunakan sebagai dasar pemilihan model terbaik.

2.18 Pembagian dataset

Pembagian *dataset* merupakan proses memisahkan data menjadi data latih, data validasi, dan data uji. Data latih digunakan untuk membentuk pola pembelajaran model, data validasi digunakan untuk memantau performa selama pelatihan dan membantu menentukan konfigurasi tertentu, sedangkan data uji digunakan untuk mengukur kemampuan model terhadap data yang belum pernah dilihat. Pembagian data perlu dilakukan agar performa model tidak hanya dinilai dari data yang digunakan saat pelatihan. Dengan demikian, hasil evaluasi dapat menggambarkan kemampuan generalisasi model secara lebih objektif.

Dalam penelitian ini, pembagian *dataset* dilakukan dengan mempertahankan proporsi kelas menggunakan pendekatan *stratified split*. Pendekatan ini digunakan agar distribusi kelas positif dan negatif pada data latih, validasi, dan uji tetap seimbang secara proporsional. Proses pembagian data seperti ini umum digunakan dalam eksperimen pembelajaran mesin untuk mengurangi risiko bias distribusi kelas pada setiap bagian data [23, 24]. Penjelasan teknis mengenai jumlah data latih, validasi, dan uji dijelaskan lebih lanjut pada Bab 3 dan Bab 4.

2.19 Ketidakseimbangan data dan oversampling

Ketidakseimbangan data terjadi ketika jumlah data pada satu kelas lebih banyak dibandingkan kelas lainnya. Dalam klasifikasi sentimen, kondisi ini dapat membuat model cenderung mempelajari kelas mayoritas lebih dominan dibandingkan kelas minoritas. Hal tersebut dapat menyebabkan nilai akurasi terlihat baik, tetapi performa terhadap kelas minoritas menjadi kurang optimal [25]. Oleh karena itu, ketidakseimbangan data perlu diperhatikan dalam proses pelatihan model.

Oversampling merupakan salah satu pendekatan yang digunakan untuk menangani ketidakseimbangan data. Pendekatan ini dilakukan dengan menambah jumlah data pada kelas minoritas agar jumlahnya lebih seimbang dengan kelas mayoritas. Dalam penelitian ini, *oversampling* dilakukan hanya pada data latih agar data validasi dan data uji tetap merepresentasikan distribusi asli. Dengan demikian, proses evaluasi model dapat dilakukan secara lebih objektif terhadap data yang tidak digunakan dalam pelatihan.

2.20 Google Colab

Google Colab merupakan platform berbasis *cloud* yang dapat digunakan untuk menulis dan menjalankan kode Python melalui peramban web. Platform ini sering digunakan dalam penelitian berbasis data karena mendukung berbagai pustaka Python dan dapat digunakan tanpa instalasi perangkat lunak secara lokal. Dalam penelitian ini, Google Colab digunakan sebagai lingkungan kerja untuk menjalankan proses pengolahan data. Proses tersebut meliputi *preprocessing*, tokenisasi, pembagian data, *oversampling*, pelatihan model, dan evaluasi model.

2.21 TensorFlow

TensorFlow merupakan pustaka *open-source* yang digunakan untuk membangun, melatih, dan mengevaluasi model *machine learning* maupun *deep learning*. TensorFlow menyediakan berbagai modul yang mendukung pembuatan model jaringan saraf, termasuk model berbasis LSTM. TensorFlow merupakan pustaka yang dapat digunakan untuk membangun dan menjalankan model *machine learning* dalam skala besar, termasuk model berbasis jaringan saraf [26]. Dalam penelitian ini, TensorFlow digunakan untuk membangun arsitektur LSTM 1 *layer*, LSTM 2 *layer*, dan Bi-LSTM.

2.22 Evaluasi model

Evaluasi model merupakan tahap yang digunakan untuk mengukur kemampuan model dalam melakukan klasifikasi terhadap data uji. Pada penelitian ini, evaluasi dilakukan menggunakan *confusion matrix*, *accuracy*, *precision*, *recall*, F1-*score*, dan AUC. Penggunaan beberapa metrik dilakukan karena *dataset* memiliki jumlah kelas positif dan negatif yang tidak seimbang. Metrik evaluasi

tersebut digunakan untuk melihat performa model dari beberapa aspek yang berbeda.

2.22.1 Confusion matrix

Confusion matrix merupakan tabel yang menunjukkan jumlah prediksi benar dan salah dari model klasifikasi. Pada klasifikasi biner, *confusion matrix* terdiri dari *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Komponen *confusion matrix* perlu dijelaskan karena menjadi dasar perhitungan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan F1-*score*. Komponen *confusion matrix* ditunjukkan pada Tabel 2.1.

Tabel 2.1. Komponen *confusion matrix*

Aktual / prediksi	Positif	Negatif
Positif	TP	FN
Negatif	FP	TN

2.22.2 Accuracy

Accuracy digunakan untuk mengukur perbandingan antara jumlah prediksi benar dengan seluruh data yang diuji. Rumus *accuracy* digunakan untuk menunjukkan perhitungan jumlah prediksi benar terhadap seluruh data. Pada rumus tersebut, TP merupakan *true positive*, TN merupakan *true negative*, FP merupakan *false positive*, dan FN merupakan *false negative*. Rumus *accuracy* ditunjukkan pada Rumus 2.12.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.12}$$

Nilai *accuracy* yang tinggi menunjukkan bahwa sebagian besar prediksi model sesuai dengan label sebenarnya. Namun, pada data yang tidak seimbang, *accuracy* tidak selalu cukup untuk menunjukkan performa model secara menyeluruh. Oleh karena itu, penelitian ini juga menggunakan *precision*, *recall*, F1-*score*, dan AUC. Penggunaan beberapa metrik tersebut membantu memberikan gambaran evaluasi model yang lebih lengkap.

2.22.3 Precision

Precision digunakan untuk mengukur ketepatan model dalam memprediksi kelas positif. Rumus *precision* digunakan untuk menunjukkan perbandingan antara prediksi positif yang benar dengan seluruh prediksi positif. Nilai *precision* yang tinggi menunjukkan bahwa prediksi positif yang diberikan oleh model cenderung benar. Rumus *precision* ditunjukkan pada Rumus 2.13.

$$Precision = \frac{TP}{TP + FP} \tag{2.13}$$

Metrik *precision* penting digunakan ketika kesalahan dalam memprediksi kelas positif perlu diminimalkan. Dalam konteks penelitian ini, *precision* membantu menunjukkan seberapa tepat model ketika memprediksi suatu opini sebagai sentimen positif. Nilai *precision* yang rendah dapat menunjukkan bahwa model masih banyak menghasilkan prediksi positif yang sebenarnya bukan positif. Oleh karena itu, *precision* digunakan bersama metrik lain agar evaluasi model lebih seimbang.

2.22.4 Recall

Recall digunakan untuk mengukur kemampuan model dalam menemukan seluruh data yang benar-benar termasuk kelas positif. Rumus *recall* digunakan untuk menunjukkan perbandingan antara prediksi positif yang benar dengan seluruh data positif sebenarnya. Nilai *recall* yang tinggi menunjukkan bahwa model mampu mengenali sebagian besar data positif dengan baik. Rumus *recall* ditunjukkan pada Rumus 2.14.

$$Recall = \frac{TP}{TP + FN} \tag{2.14}$$

Metrik *recall* penting digunakan ketika data positif yang terlewat perlu diminimalkan. Dalam penelitian ini, *recall* membantu menunjukkan kemampuan model dalam menemukan opini yang benar-benar termasuk sentimen positif. Nilai *recall* yang rendah dapat menunjukkan bahwa model masih banyak melewatkan data positif. Oleh karena itu, *recall* digunakan bersama *precision* dan F1-*score*.

2.22.5 F1-score

F1-*score* merupakan metrik yang menggabungkan *precision* dan *recall* ke dalam satu nilai. Rumus F1-*score* digunakan untuk menunjukkan keseimbangan antara ketepatan prediksi positif dan kemampuan model menemukan data positif. Metrik ini penting digunakan karena dapat memberikan gambaran performa yang lebih seimbang pada data yang tidak seimbang [27]. Rumus F1-*score* ditunjukkan pada Rumus 2.15.

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.15}$$

Nilai F1-*score* yang tinggi menunjukkan bahwa model memiliki keseimbangan yang baik antara *precision* dan *recall*. Pada penelitian ini, F1-*score* digunakan sebagai salah satu dasar utama dalam membandingkan performa model. Hal tersebut dilakukan karena *dataset* memiliki perbedaan jumlah data antara kelas negatif dan positif. Dengan demikian, F1-*score* membantu memilih model yang tidak hanya baik pada satu sisi metrik saja.

2.22.6 Area under curve

AUC atau *area under curve* merupakan metrik yang digunakan untuk mengukur kemampuan model dalam membedakan kelas positif dan negatif berdasarkan nilai probabilitas prediksi. AUC berkaitan dengan kurva ROC yang dibentuk dari nilai *true positive rate* (TPR) dan *false positive rate* (FPR). Nilai AUC yang semakin mendekati 1 menunjukkan bahwa model memiliki kemampuan yang semakin baik dalam membedakan kelas positif dan negatif. Rumus TPR dan FPR ditunjukkan pada Rumus 2.16 dan Rumus 2.17.

$$TPR = \frac{TP}{TP + FN} \tag{2.16}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.17}$$

Dalam penelitian ini, AUC digunakan untuk melihat kemampuan model dalam membedakan kelas sentimen berdasarkan skor probabilitas. Metrik ini

melengkapi *accuracy*, *precision*, *recall*, dan F1-*score*. Dengan adanya AUC, performa model dapat dilihat dari sudut kemampuan pemisahan kelas. Oleh karena itu, AUC digunakan sebagai salah satu metrik pembanding antar model.

2.23 Ambang klasifikasi

Ambang klasifikasi atau *threshold* merupakan nilai batas yang digunakan untuk mengubah probabilitas keluaran model menjadi kelas prediksi. Pada klasifikasi biner, model menghasilkan nilai probabilitas antara 0 dan 1. Nilai tersebut kemudian dibandingkan dengan *threshold* tertentu untuk menentukan apakah data masuk ke kelas positif atau negatif. Rumus keputusan klasifikasi berdasarkan *threshold* ditunjukkan pada Rumus 2.18.

$$\hat{y} = \begin{cases} 1, & \text{jika } p \geq \tau \\ 0, & \text{jika } p < \tau \end{cases} \tag{2.18}$$

Pada Rumus 2.18, $\hat{y}$ merupakan kelas prediksi, p merupakan probabilitas keluaran model, dan τ merupakan nilai *threshold*. Penentuan *threshold* dapat memengaruhi nilai *precision*, *recall*, dan F1-*score*. Pada data yang tidak seimbang, pemilihan *threshold* dapat dilakukan dengan mempertimbangkan keseimbangan antara *precision* dan *recall* [25, 27]. Dalam penelitian ini, penjelasan teknis mengenai pemilihan *threshold* 0,22 dijelaskan lebih lanjut pada Bab 3 dan hasilnya ditunjukkan pada Bab 4.

2.24 Penelitian terdahulu

Penelitian terdahulu digunakan sebagai acuan untuk memahami metode, objek, dan hasil penelitian yang berkaitan dengan analisis sentimen. Beberapa penelitian sebelumnya menunjukkan bahwa metode berbasis LSTM dan variasinya dapat digunakan untuk mengolah data teks dan mengklasifikasikan sentimen. Kajian penelitian terdahulu juga membantu menunjukkan perbedaan penelitian ini dengan penelitian sebelumnya. Perbedaan tersebut dapat dilihat dari objek penelitian, sumber data, rentang data, dan arsitektur model yang digunakan.

Cahyadi dkk. menggunakan RNN dengan LSTM untuk analisis sentimen pada data Instagram [16]. Alghifari dkk. menggunakan Bi-LSTM untuk analisis sentimen terhadap layanan Grab Indonesia dan menunjukkan bahwa Bi-LSTM

dapat menjadi alternatif dalam pemrosesan teks berurutan [22]. Chakraborty dkk. menerapkan model LSTM untuk analisis sentimen pada data *tweet* COVID-19 [15]. Hamza dkk. menunjukkan bahwa pendekatan jaringan saraf dan model *deep learning* dapat digunakan untuk meningkatkan prediksi sentimen pada data teks media sosial [8].

Berdasarkan penelitian terdahulu tersebut, penelitian ini menggunakan metode LSTM untuk menganalisis sentimen terhadap produk yang dipromosikan oleh *influencer* di X. Perbedaan penelitian ini terletak pada objek penelitian, yaitu opini publik terhadap produk promosi *influencer*, rentang data tahun 2020 sampai 2024, serta perbandingan tiga arsitektur model. Tiga arsitektur tersebut adalah LSTM 1 *layer*, LSTM 2 *layer*, dan Bi-LSTM. Hasil perbandingan model digunakan untuk menentukan model terbaik berdasarkan metrik evaluasi.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA