

BAB II

KAJIAN LITERATUR

1.1 Artificial Intelligence

Kecerdasan Buatan memberikan manfaat yang dapat diamati, tetapi pengumpulan, penggunaan, dan penyalahgunaan data yang digunakan untuk melatih dan mengoperasikan AI, serta algoritma itu sendiri, dapat membawa bahaya bagi masyarakat yang sebelumnya mungkin tidak mereka ketahui. Contohnya, perusahaan bisa saja mengawasi kinerja dan tindakan pegawai di lingkungan kerja secara rahasia dan tak terduga [12].

AI bekerja memakai cara memproses data lewat aturan belajar mesin (*machine learning*) yang membuat sistem bisa terus meningkatkan performanya. Salah satu bagian dari *machine learning* yang sering dipakai ialah *deep learning* [13]. Penggunaan AI di kantor bukan cuma sekadar ikut-ikutan, tapi juga jadi hal penting untuk menaikkan efisiensi dan inovasi. Di samping itu, penerapan AI bisa menaikkan hasil, dan mutu kerja para karyawan.

1.1.1 Machine Learning

Machine learning adalah teknologi yang memungkinkan komputer untuk belajar secara otomatis dari data tanpa perlu diprogram ulang terus-menerus [14]. Sistem ini bekerja dengan cara mencari pola-pola tertentu dari kumpulan informasi yang sudah ada sebelumnya untuk membangun sebuah model analitis [15]. Lewat proses tersebut, komputer bisa memberikan keputusan atau prediksi berdasarkan pengalaman yang didapatkan dari data latih [16]. Penggunaan metode ini sangat membantu dalam mengolah data yang jumlahnya sangat banyak dan memiliki struktur yang rumit [15]. Penerapan teknik ini sudah mencakup banyak hal di bidang informatika, seperti pengenalan gambar hingga pemrosesan bahasa alami [14].

Dalam dunia digital, metode ini sering digunakan untuk mengenali ciri khas tertentu dari sebuah pesan agar bisa dikelompokkan dengan benar [17]. Prosesnya dimulai dengan memberikan banyak contoh informasi agar sistem bisa memahami perbedaan antar kategori yang masuk [15]. Kemampuannya dalam memproses data secara mandiri membuat pekerjaan manusia dalam memilah informasi menjadi jauh lebih ringan [17]. Hal ini membuat pengolahan informasi bisa berjalan dengan waktu yang singkat namun tetap memberikan hasil yang sesuai dengan target [14]. Selain itu, sistem ini akan terus

berkembang menjadi lebih baik seiring dengan semakin banyaknya data baru yang dipelajari dari waktu ke waktu [15].

Pada analisis teks, cara ini berperan penting untuk mengenali kecenderungan atau kategori dari sebuah kalimat secara otomatis [17]. Salah satu teknik yang sering digunakan adalah pohon keputusan yang membagi data berdasarkan aturan-aturan logika tertentu [17]. Proses pengenalan ini dilakukan dengan membandingkan informasi baru dengan pola yang sudah tersimpan di dalam memori sistem sebelumnya [14]. Penggunaan cara tersebut memastikan bahwa setiap data yang masuk bisa dipetakan dengan rapi ke dalam kelompok yang diinginkan seperti sentimen positif atau negatif [17]. Hasil dari proses ini sangat berguna untuk melihat gambaran umum mengenai tren atau topik yang sedang ramai dibicarakan di internet [15].

1.1.2 Deep Learning

Deep learning adalah salah satu cabang dari *Artificial Intelligence* dan teknologi pembelajaran mesin yang mampu mengolah data dengan cara yang kompleks untuk memberikan prediksi atau keputusan secara otomatis [18]. Teknologi ini merupakan dasar dari berbagai inovasi modern karena kemampuannya dalam mengenali pola dan memproses data dalam volume besar untuk pengambilan keputusan yang terinformasi. Seiring perkembangannya, sistem ini tidak hanya berperan sebagai alat teknis, tetapi juga dihubungkan melalui paradigma pembelajaran terstruktur [19]. Dengan penerapan teknologi ini, *deep learning* mempunyai kemungkinan yang sangat besar untuk merevolusi berbagai banyak ilmu dengan kemampuan otomatisasi tugas-tugas kompleks, seperti pembuatan kode, pengenalan pola, dan pengembangan sistem cerdas yang lebih responsif [20].

1.1.3 Natural Language Processing

Natural Language processing (NLP) adalah cara bagi mesin untuk mempelajari bahasa manusia agar bisa berkomunikasi dengan komputer, yang memadukan ilmu komputer, kecerdasan buatan terkait dengan linguistik [21]. Inti dari NLP adalah merancang dan mengembangkan program agar manusia bisa berinteraksi dengan mesin dan mampu mengolah informasi melalui bahasa alami [22].

Pada penerapannya ada tiga tahapan yang dipakai untuk menganalisa bahasa, yaitu *stemming*, *tokenization*, *parsing* [23].

1. *Stemming* adalah suatu proses untuk menghilangkan imbuhan pada kata-kata yang memiliki variasi morfologis menjadi bentuk kata dasar. Contohnya, kata “menandai” menjadi “tanda” dengan menghilangkan imbuhan “men-i” [24].
2. *Tokenization* adalah suatu proses membagi teks menjadi potongan kata-kata. Contohnya, kata “Mall itu sangat luas” akan dipisah menjadi beberapa token, yaitu “Mall”, “itu”, “sangat”, “luas” [25].

Parsing adalah suatu proses menganalisis urutan simbol, dalam bahasa manusia atau bahasa komputer berdasarkan tata bahasa yang diterapkan [26]. Berikut ini adalah contoh kalimat diparsing menggunakan metode *Constituency parsing*. *Constituency parsing* merupakan sebuah metode yang dipakai untuk menganalisis struktur kalimat yang menggambarkan strukturnya sesuai dengan tata bahasa [27].

1.1.4 Analisis Sentimen

Analisis sentimen adalah cara untuk mengenali dan mengambil isi perasaan atau pendapat dari sebuah tulisan menggunakan bantuan komputer [28]. Di masa sekarang, media sosial jadi gudang data yang sangat besar karena di sana orang-orang bebas memberikan komentar atau ulasan apa pun [29]. Tujuan utama dari cara ini adalah untuk melihat apakah sebuah komentar itu isinya hanya informasi biasa atau sebenarnya sedang menunjukkan perasaan suka dan tidak suka dari seseorang [30]. Dengan melihat arah perasaan para pengguna ini, kita bisa lebih mudah memahami bagaimana sebuah topik atau aturan baru diterima oleh orang banyak [28]. Karena banyaknya komentar yang masuk setiap harinya, kita butuh sistem otomatis yang pintar supaya bisa mengerti maksud setiap pesan dengan cepat [31].

Cara kerja analisis ini biasanya membagi komentar ke dalam beberapa kelompok, seperti kelompok positif, negatif, atau netral [29]. Dengan teknologi *deep learning*, komputer bisa membaca ribuan komentar sekaligus untuk mencari tahu mana yang isinya mendukung dan mana yang isinya berupa kritikan [30]. Salah satu tantangan terbesarnya adalah memahami bahasa gaul, singkatan, atau sindiran yang sering sekali dipakai orang-orang saat berkomentar di internet [28]. Cara ini sangat membantu karena kita bisa langsung tahu bagaimana suasana hati masyarakat terhadap suatu masalah dalam waktu yang singkat [31]. Hasil dari pengelompokan komentar ini nantinya akan sangat membantu dalam mengevaluasi bagaimana sebenarnya tanggapan orang-orang terhadap hal-hal yang sedang ramai dibicarakan [32].

1.1.5 Filtering

Data Filtering atau penyaringan data adalah langkah pertama dalam persiapan teks untuk memastikan mutu, keteraturan, dan kebersihan data sebelum memasuki fase pelatihan model *Deep Learning*. Dalam pengelolaan dataset dari media sosial, penyaringan data dimulai dengan menghapus baris data yang kosong (data kosong) setelah melewati tahap pembersihan sebelumnya. Langkah selanjutnya adalah menghilangkan data yang berulang atau duplikat agar tidak terjadi bias distribusi kelas saat model belajar mengenali pola teks. [33]

Di samping pembersihan baris dalam tabel dataset, penyaringan data juga dilakukan dengan menetapkan ambang batas panjang kalimat untuk memilih baris teks data. Penghapusan kalimat yang terlalu pendek dikerjakan karena teks tersebut kurang mengandung informasi sintaksis dan fitur lokal yang diperlukan agar dapat diproses secara optimal oleh arsitektur model *sekuensial* seperti LSTM. [34] Tahap penting terakhir dalam penyaringan data teks yang beragam adalah deteksi atau identifikasi bahasa secara otomatis. [34] Proses penyaringan yang berbasis teks ini diterapkan dengan menggunakan pustaka *LangDetect* untuk mengidentifikasi karakter teks secara otomatis. Penyaringan melalui *LangDetect* bertujuan untuk memastikan bahwa hanya baris data teks yang benar-benar berbahasa Indonesia yang akan diteruskan ke dalam model, sehingga representasi vektor kata tetap konsisten. [33]

1.1.6 Lexicon-based (InSet)

Lexicon-based merupakan salah satu metode ilmiah dalam menganalisis perasaan yang beroperasi dengan memanfaatkan kamus kata yang berisi daftar kosakata beserta nilai bobot tertentu sebagai acuan dasarnya. [35] Metode ini menetapkan label perasaan dari sebuah teks lewat keberadaan serta keselarasan kata-kata di dalamnya dengan entitas yang telah ditentukan dalam kamus tersebut. Walaupun memberikan kesederhanaan dan keefisienan dalam komputasi, metode ini kadang-kadang mengalami kesulitan dalam menangkap nuansa bahasa, sarkasme, serta ambiguitas yang bersifat kontekstual. [36]

Dalam analisis bahasa Indonesia, salah satu kamus sentimen yang khusus digunakan adalah *InSet* (*Indonesian Sentiment Lexicon*), yaitu kamus yang disusun dengan mengumpulkan kata-kata dari platform media sosial Twitter secara langsung. Karakteristik data dari media sosial yang selalu berubah membuat penggunaan kamus *InSet* ini mempermudah proses ekstraksi teks, karena sistem tidak perlu lagi menerjemahkan teks tersebut ke dalam bahasa Inggris. [35]

Dalam penerapannya, setiap kata yang unik dalam kumpulan data akan dicocokkan dan dibandingkan langsung dengan daftar kata *InSet* yang memiliki rentang bobot nilai dari -5 hingga 5. Setelah bobot untuk masing-masing kata yang terdaftar ditemukan, nilai-nilai ini akan dijumlahkan untuk setiap kalimat guna memperoleh nilai akhir dari skor polaritas. Total akumulasi bobot ini yang menentukan pengelompokan kalimat ke dalam kategori sentimen yang ditargetkan.[35] Seiring dengan perkembangannya, metodologi berbasis kamus kata ini juga sering digabungkan atau dibandingkan dengan teknik *deep learning* untuk menilai akurasi, presisi, dan nilai *recall* dalam klasifikasi sentimen.[36]

1.2 Artificial Neural Network

Artificial Neural Network (ANN) merupakan Sebuah sistem pemodelan data *nonlinier* yang membuat model atau pola dari hubungan yang kompleks antara input dan output. *Neural network* memiliki kemampuan belajar yang sangat baik.[37] Kegunaan dari *Artificial Neural Networks* (ANN) sudah terbukti dalam berbagai bidang seperti data *mining* dan biometrik. Sejak awal desainnya, hingga berbagai inovasi dan pengembangan terbaru seperti *Convolutional Neural Networks* (CNN) dan *Recurrent Neural Networks* (RNN) [38].

Sistem *Artificial Neural Network* (ANN) menggunakan node atau neuron buatan yang saling terhubung dalam struktur berlapis yang menyerupai otak manusia, sehingga mampu menangkap hubungan *nonlinier* dalam data secara lebih efisien dibandingkan metode lainnya untuk menyelesaikan berbagai permasalahan kompleks dalam kecerdasan buatan. Di antara berbagai metode AI, *Artificial Neural Network* (ANN) menunjukkan kemampuannya untuk belajar dan beradaptasi, yang terinspirasi dari struktur dan cara kerja otak manusia[39].

1.2.1 Recurrent Neural Network

Recurrent Neural Network (RNN) merupakan arsitektur *Artificial Neural Network* yang dibuat khusus untuk menangani data yang bersifat berurutan. Pada koneksi RNN membentuk titik grafik yang mengarah sesuai urutan waktu, yang menunjukkan perilaku yang *dinamis* seiring waktu [40].

Proses komputasi pada unit RNN dasar untuk setiap langkah waktu (t) didefinisikan dengan dua persamaan utama, yakni *hidden state* dan *output gate*. *Hidden state* berfungsi sebagai memori internal jaringan untuk menyimpan informasi dari langkah waktu sebelumnya melalui persamaan (1), dan *output gate* untuk menentukan hasil keluaran (*output*) pada langkah waktu tersebut melalui persamaan (2) berikut [41]:

$$h_t = f(Ux_t + Wh_{t-1}) \dots\dots\dots(1)$$

$$z_t = \sigma(Vh_t) \dots\dots\dots(2)$$

Berdasarkan standar arsitektur jaringan saraf rekuren, definisi notasi pada persamaan di atas adalah sebagai berikut [41]:

h_t : *Hidden state* atau status tersembunyi pada waktu t .

x_t Vektor *input* (seperti representasi kata) pada waktu t .

h_{t-1} : Status tersembunyi yang membawa informasi dari urutan waktu sebelumnya.

z_t : Hasil keluaran (*output*) pada langkah waktu t .

U : Matriks bobot (*weight*) yang menghubungkan *input* ke *hidden layer*.

W : Matriks bobot yang menghubungkan antar status tersembunyi secara rekursif.

V : Matriks bobot yang menghubungkan *hidden layer* ke *output layer*.

f / σ : Fungsi aktivasi non-linier yang digunakan untuk mengatur aliran sinyal di dalam jaringan.

Hasil yang diinginkan dari rangkaian operasi ini adalah kemampuan model untuk mempertahankan konteks informasi masa lalu (h_{t-1}) saat memproses *input* data saat ini (x_t) guna menghasilkan prediksi yang akurat [41].



Gambar 1.1 Tinjauan Proses Pelatihan pada RNN [41]

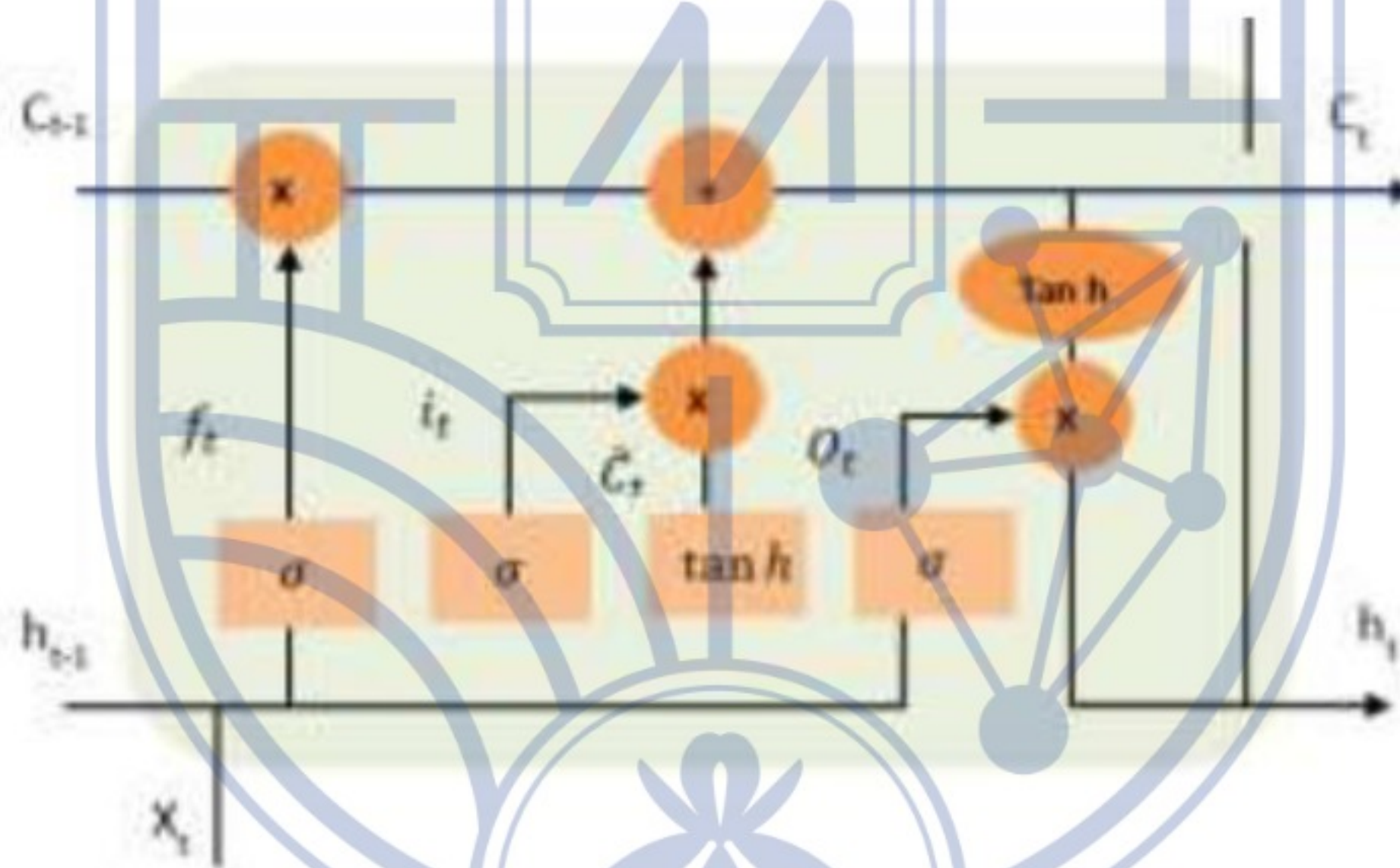
Karena kemampuannya dalam menangani data yang berurutan, RNN banyak digunakan dan sangat efisien dalam pemrosesan NLP. Jaringan ini memiliki pemahaman bawaan terhadap struktur bahasa, sehingga mampu menangkap pola *semantik* serta hubungan *konteksual* dalam teks [42].

RNN berfungsi sebagai sistem yang mampu mempertahankan informasi dalam jangka panjang dan mempertahankan informasi penting dari masa lalu yang krusial[43]. Selain itu, model ini memberikan kemampuan untuk berkembang dengan baik

karena bisa disesuaikan dengan berbagai ukuran data tanpa menyebabkan lonjakan kompleksitas komputasi dalam perhitungan. Hal ini didukung oleh tingkat ketepatan dalam proses pengambilan sampel [44]. Walaupun begitu, kinerja RNN sangat tergantung pada pemilihan metode optimasi dan fungsi aktivasi yang sesuai untuk mengatasi pola data yang tidak linier dan menjamin akurasi prediksi yang terbaik [41].

1.2.2 LSTM

Long Short-Term Memory (LSTM) merupakan salah satu jenis metode *deep learning* yang diciptakan untuk mengatasi masalah yang ada pada Recurrent Neural Network (RNN). LSTM dibentuk untuk menghindari masalah vanishing gradient dengan memanfaatkan unit memori yang digunakan untuk menyimpan informasi lebih panjang. Pada arsitekturnya, LSTM memiliki beberapa kategori yaitu input gate, forget gate, dan output gate yang mengatur aliran informasi di dalam jaringan [45]. Struktur internal dari unit sel LSTM tersebut dapat dilihat pada Gambar 2.2 [46].



Gambar 1.2 Arsitektur LSTM [46]

Proses komputasi pada unit LSTM dasar untuk setiap langkah waktu (t) didefinisikan dengan rangkaian persamaan utama, yakni input gate, forget gate, cell state candidate, output gate, update cell state, dan hidden state output. Input gate berfungsi untuk menentukan informasi baru yang akan masuk ke memori melalui persamaan (3), forget gate memutuskan informasi lama yang harus dibuang melalui persamaan (4), cell state candidate untuk menghitung kandidat nilai fitur baru melalui persamaan (5), output gate untuk menyaring hasil sel memori yang akan dikeluarkan melalui persamaan (6), update cell state sebagai proses pembaruan memori utama melalui persamaan (7), serta hidden state output untuk menghasilkan representasi akhir unit melalui persamaan (8) berikut [46]:

$$it = \sigma(Wi \times [xt + ht - 1]) + bi \dots\dots\dots(3)$$

$$ft = \sigma(Wf \times [xt + ht - 1]) + bf) \dots\dots\dots(4)$$

$$Ct = \tanh(WC \times [xt + ht - 1] + bC) \dots\dots\dots(5)$$

$$ot = \sigma(Wo \times [xt + ht - 1] + bo) \dots\dots\dots(6)$$

$$Ct = ft \times Ct - 1 + it \times Ct \dots\dots\dots(7)$$

$$ht = ot \times \tanh(Ct) \dots\dots\dots(8)$$

Definisi notasi pada rangkaian persamaan LSTM di atas adalah sebagai berikut [46]:

ft, it, ot : Nilai aktivasi untuk masing-masing gerbang (*forget*, *input*, dan *output*).

$ht - 1$: Merupakan status tersembunyi (*hidden state*) dari urutan sebelumnya yang membawa konteks jangka pendek.

Ct : *Cell state* yang berfungsi sebagai memori jangka panjang jaringan.

ht : *Hidden state* atau status tersembunyi pada langkah waktu t .

xt : Vektor *input* pada langkah waktu t .

Wi, Wf, WC, Wo : Matriks bobot (*weights*) yang berfungsi mengatur kekuatan sinyal input pada gerbang *input*, *forget*, *candidate*, dan *output*.

bi, bf, bC, bo : Nilai bias yang berfungsi sebagai parameter tambahan untuk mengoptimalkan fleksibilitas komputasi pada masing-masing gerbang.

$ht - 1$: Operasi penggabungan (*concatenation*) antara *hidden state* sebelumnya dan *input* saat ini.

σ : Fungsi aktivasi *Sigmoid* yang memetakan nilai ke dalam rentang 0 hingga 1 untuk mengatur aliran data.

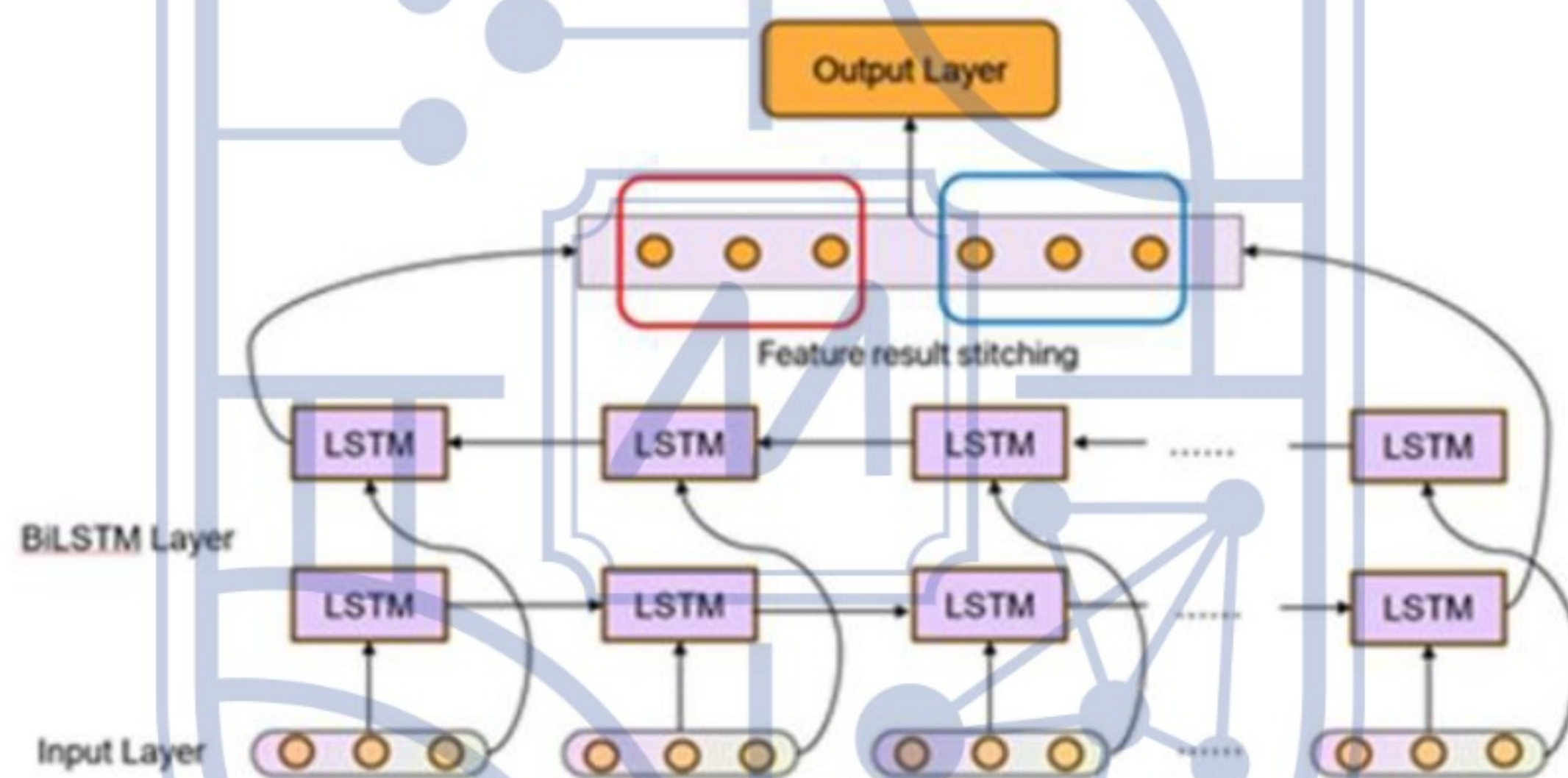
$\tanh$: Fungsi aktivasi *Hyperbolic Tangent* yang menstabilkan nilai jaringan ke dalam rentang -1 hingga 1.

Arsitektur LSTM juga banyak dipilih karena berfungsi baik dalam komputer *edge* (*edge computing*) seperti pada perangkat *Raspberry pi*, dimana efisiensi energi dan kecepatan pemrosesan sangatlah penting [47]. Selain itu, LSTM juga memiliki kemampuan yang sangat baik dalam bidang memahami pola untuk jangka panjang pada data sekuensial. Hal ini menjadikan LSTM sangat cocok dalam mengelola data yang bersifat dinamis, non-linear, dan memiliki fluktuasi tinggi, seperti pada kasus prediksi harga saham[48].

1.2.3 Bi-LSTM

BiLSTM (Bidirectional Long Short-Term Memory) adalah jenis khusus dari Recurrent Neural Network (RNN) yang menggabungkan prinsip BRNN dan LSTM. Pada

dasarnya, model ini terdiri dari dua jaringan LSTM yang berdiri sendiri. Satu jaringan bertugas memproses urutan data ke arah depan, sementara jaringan yang lainnya memproses ke arah sebaliknya. Cara kerja ini memungkinkan BiLSTM untuk memproses data bertahap dengan mempertimbangkan informasi dari masa lalu dan masa depan secara bersamaan [49]. Jaringan BiLSTM bekerja dengan melatih data *input* sebanyak dua kali secara paralel. Model ini terdiri dari dua jaringan LSTM terpisah, yaitu *forward* LSTM dan *backward* LSTM. Jaringan *forward* memproses urutan data dari awal ke akhir. Sebaliknya, jaringan *backward* memproses urutan data yang sama tetapi dari arah belakang ke depan. Setelah kedua proses itu selesai, hasil pembacaan dari arah *forward* dan *backward* digabungkan untuk menghasilkan satu *output* [50].



Gambar 2.1 Struktur Bi-LSTM [49]

Proses komputasi pada unit Bi-LSTM dasar untuk setiap langkah waktu (t) didefinisikan dengan rangkaian persamaan utama, yakni *bidirectional input gate*, *bidirectional forget gate*, *update cell state*, *bidirectional output gate*, *hidden state combination*, serta *classification & softmax*. *Bidirectional input gate* berfungsi menentukan informasi baru dari kedua arah yang akan diperbarui melalui persamaan (9), *bidirectional forget gate* memutuskan informasi lama yang tidak relevan untuk dibuang melalui persamaan (10), *update cell state* sebagai proses inti pembaruan memori jangka panjang melalui persamaan (11), *bidirectional output gate* untuk menyaring informasi yang akan dikeluarkan sebagai fitur melalui persamaan (12), *hidden state combination* untuk menghasilkan representasi fitur lengkap hasil penggabungan konteks melalui persamaan (13), serta *classification & softmax* untuk memetakan fitur ke dalam label sentimen dan menghasilkan probabilitas distribusi kelas melalui persamaan (14) dan (15) berikut [49]:

$$i_t = \sigma(W_{ix}^f x_t + W_{ih}^f h_{t-1}^f + W_{ic}^f c_{t-1}^f + b_i^f) \odot \sigma(W_{ix}^b x_t + W_{ih}^b h_{t+1}^b + W_{ic}^b c_{t+1}^b + b_i^b) \dots\dots\dots(9)$$

$$f_t = \sigma(W_{fx}^f x_t + W_{fh}^f h_{t-1}^f + W_{fc}^f c_{t-1}^f + b_f^f) \odot \sigma(W_{fx}^b x_t + W_{fh}^b h_{t+1}^b + W_{fc}^b c_{t+1}^b + b_f^b) \dots\dots\dots(10)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_{cx}^f x_t + W_{ch}^f h_{t-1}^f + b_c^f) + i_t \odot \tanh(W_{cx}^b x_t + W_{ch}^b h_{t+1}^b + b_c^b) \dots\dots\dots(11)$$

$$o_t = \sigma(W_{ox}^f x_t + W_{oh}^f h_{t-1}^f + W_{oc}^f c_t^f + b_o^f) \odot \sigma(W_{ox}^b x_t + W_{oh}^b h_{t+1}^b + W_{oc}^b c_t^b + b_o^b) \dots\dots\dots(12)$$

$$h_t = o_t \odot \tanh(C_t) \dots\dots\dots(13)$$

$$y_t = \text{Softmax}(W_{hy} h_t + b_y) \dots\dots\dots(14)$$

$$\text{Softmax}(\mathbf{O})_j = \frac{e^{o_j}}{\sum_{k=1}^M e^{o_k}} \dots\dots\dots(15)$$

Definisi notasi dari persamaan (9) sampai (15) dijabarkan secara rinci sebagai berikut [49]:

x_t : Vektor *input* pada langkah waktu t yang diproses secara simultan oleh kedua jalur .

h_{t-1}^f : Status tersembunyi (*hidden state*) dari jalur maju (*forward*) pada langkah waktu sebelumnya.

h_{t+1}^b : Status tersembunyi dari jalur mundur (*backward*) pada langkah waktu masa depan (urutan setelahnya).

c_{t+1}^b : Memori jangka panjang (*cell state*) yang membawa informasi dari urutan sebelumnya.

W (Weight): Matriks bobot yang mengatur kekuatan sinyal antar lapisan, di mana penulisan indeksnya menunjukkan tujuan gerbang (huruf i untuk *Input Gate*, f untuk *Forget Gate*, o untuk *Output Gate*, atau c untuk *Cell State*), asal informasi data (huruf x untuk *Input*, h untuk *Hidden State*, atau c untuk *Cell State* sebelumnya), serta arah jalur pada *superscript* (f untuk *Forward* atau b untuk *Backward*).

b (Bias): Nilai bias sebagai parameter tambahan optimasi, di mana penulisan indeksnya menunjukkan tujuan gerbang (huruf i untuk *Input Gate*, f untuk *Forget Gate*, o untuk *Output Gate*, atau c untuk *Cell State*), asal informasi data (huruf x untuk *Input*, h untuk *Hidden State*, atau c untuk *Cell State* sebelumnya), serta arah jalur pada *superscript* (f untuk *Forward* atau b untuk *Backward*).

σ : Fungsi aktivasi *Sigmoid* yang bertugas mengatur aliran informasi dalam rentang 0 hingga 1.

$\tanh$: Fungsi aktivasi *Hyperbolic Tangent* yang menstabilkan nilai ke dalam rentang -1 hingga 1.

y_t : Vektor *output* hasil klasifikasi pada langkah waktu t .

Softmax : Fungsi aktivasi untuk menormalisasi *output* menjadi distribusi probabilitas kelas.

BiLSTM cocok untuk menangkap asosiasi data jangka pendek maupun jangka panjang. Jaringan ini berguna untuk memproses data yang disusun secara berurutan [51]. BiLSTM efektif untuk memproses data yang berurutan karena kemampuannya dalam mengakses representasi konteks dari masa lalu maupun masa depan. Jaringan ini bekerja dengan memanfaatkan kombinasi lapisan tersembunyi yang bergerak maju dan mundur. Saat memproses sebuah kalimat, lapisan maju akan membaca kata dari awal hingga akhir, sementara lapisan mundur membacanya dari akhir kembali ke awal. Melalui proses merangkum informasi dari kedua arah pergerakan kata ini, BiLSTM menghasilkan status tersembunyi yang memuat data kontekstual dari keseluruhan kalimat secara utuh [52].

BiLSTM memiliki kelebihan dalam mempelajari fitur deret waktu (time series) secara dua arah, yang memungkinkannya untuk mempertimbangkan informasi dari masa lalu dan masa depan secara penuh melalui kombinasi lapisan LSTM yang bergerak maju dan mundur. Kemampuan untuk memfasilitasi masukan urutan informasi dari kedua arah ini membuat model menjadi lebih tangguh (*robust*) dan mampu secara maksimal mengeksplorasi fitur sampel data yang memiliki ketergantungan jangka panjang, sehingga pada akhirnya dapat meningkatkan akurasi prediksi dengan lebih baik dibandingkan LSTM tradisional [53].

1.3 Word Embedding

Word embedding adalah teknik untuk mengubah kata-kata menjadi angka koordinat atau vektor agar bisa diproses oleh komputer [54]. Lewat cara ini, setiap kata tidak lagi dianggap sebagai teks biasa, melainkan sebagai titik-titik data yang memiliki posisi tertentu [15]. Teknik ini memungkinkan sistem untuk menangkap makna dari sebuah kata berdasarkan hubungan dengan kata-kata lain di sekitarnya [14]. Hal tersebut sangat membantu karena komputer jadi bisa memahami konteks sebuah kalimat dengan lebih mendalam [54]. Hasilnya, pengolahan bahasa alami menjadi lebih mudah dilakukan karena data teks sudah berubah menjadi format matematis.

Dalam prosesnya, kata-kata yang punya makna mirip biasanya akan diletakkan pada posisi yang berdekatan dalam ruang data [14]. Hal ini sangat berguna untuk mengenali sinonim atau hubungan antar kata tanpa harus diajarkan secara manual satu per satu [54]. Model ini mempelajari pola bahasa dari kumpulan dokumen yang sangat besar agar bisa mengenali karakteristik setiap kata [15]. Namun, terkadang pola yang dipelajari juga bisa membawa pandangan yang kurang objektif jika data awalnya tidak disaring dengan baik. Oleh karena itu, pemilihan data yang bersih sangat berpengaruh pada kualitas vektor kata yang dihasilkan [16].

Pada sistem yang lebih canggih, teknik ini menjadi dasar penting sebelum data teks dimasukkan ke dalam model pembelajaran yang lebih berat [14]. Dengan menggunakan perwakilan kata yang tepat, proses pengelompokan kalimat menjadi kategori tertentu bisa berjalan dengan lebih baik [54]. Sistem tidak lagi bingung saat menghadapi kata-kata baru karena sudah bisa menebak maknanya dari pola yang mirip [15]. Penggunaan cara ini juga mempermudah sistem dalam menangani bahasa sehari-hari yang sering muncul di internet [54]. Pada akhirnya, teknik ini menjadi fondasi utama agar mesin bisa berkomunikasi dan memahami tulisan manusia dengan cara yang lebih pintar [14].

1.3.1 BERT

BERT (*Bidirectional Encoder Representations from Transformers*) adalah model bahasa yang dibuat berdasarkan arsitektur *transformer*. Model ini secara khusus menggunakan arsitektur *transformer* yang hanya terdiri dari bagian *encoder* dan juga menggunakan beberapa lapisan *encoder* dengan mekanisme *self-attention* untuk memproses urutan teks yang masuk. Berbeda dengan model-model sebelumnya, BERT biasa dilakukan secara dua arah, sehingga ia dapat menangkap dan memahami makna konteks kata dari kedua arah sekaligus[55].

Dalam *training*, BERT tercatat sebagai model dua arah yang pertama di bidang NLP yang menerapkan cara memperbaiki token yang disembunyikan (*corrupted tokens*) sebagai tugas sebelum *training*[56].

1.3.2 DistilBERT

DistilBERT adalah model kecerdasan buatan yang dibuat agar bisa mengerti maksud bahasa manusia dengan cara yang lebih praktis daripada model sebelumnya [57]. Model ini dikembangkan melalui teknik pelatihan khusus di mana sistem yang skalanya lebih kecil belajar dari sistem yang jauh lebih besar [10]. Melalui proses tersebut, struktur model

menjadi jauh lebih ringan tanpa kehilangan kemampuan intinya dalam memahami sebuah teks [9]. Penggunaan model ini membantu komputer bekerja secara maksimal tanpa harus membebani memori penyimpanan yang besar [8]. Penerapan model ini sangat memudahkan pengolahan ribuan data teks karena pengerjaannya tidak membuat perangkat menjadi lambat [58].

Dalam proses pengolahan teks, DistilBERT menggunakan mekanisme *embedding* untuk mentransformasi token menjadi representasi vektor yang kaya akan informasi. Secara matematis, hasil *embedding* dari sebuah token (u_t) merupakan hasil penggabungan dari tiga komponen utama sesuai dengan persamaan (16) berikut [59]:

$$u_t = u_x \oplus u_y \oplus u_s \dots \dots \dots (16)$$

Definisi notasi pada persamaan (16) adalah sebagai berikut [59]:

u_t : Hasil *embedding* akhir dari token t .

u_x : *Token embedding* yang mewakili makna semantik dari sub-kata.

u_y : *Position embedding* yang merepresentasikan urutan posisi token.

u_s : *Segment embedding* yang membedakan urutan antar kalimat.

Keunggulan utama dari model ini terletak pada prosesnya yang sangat singkat saat membaca data di perangkat dengan spesifikasi standar [8]. Karena ukurannya yang ringkas, sistem ini sering digunakan untuk membangun alat pemantau opini yang membutuhkan hasil dalam waktu cepat [60]. Meski bekerja dengan kecepatan tinggi, hasil pengelompokan kalimat yang didapat tetap memiliki tingkat ketepatan yang baik [9]. Selain itu, model ini juga membantu mempersingkat waktu pengerjaan dalam sebuah penelitian maupun pembuatan aplikasi baru [58]. Hal ini membuat pengolahan data dalam jumlah banyak bisa selesai lebih cepat namun hasilnya tetap berkualitas tinggi [10].

Pada penggunaan di media sosial, model ini sangat membantu dalam menangkap inti dari komentar-komentar pendek yang bahasanya tidak beraturan [57]. Sistem tersebut mempelajari hubungan antar kata secara mendalam sehingga kategori komentar yang berisi dukungan atau keluhan dapat dipisahkan secara jelas [60]. Pemisahan antara rasa suka atau tidak suka dilakukan dengan cara mengenali pola kalimat yang ada dalam data teks tersebut [9]. Kemampuan tersebut sangat penting untuk memastikan bahwa data yang dikumpulkan mencerminkan pendapat asli dari masyarakat [10]. Proses ini membuat setiap pesan yang masuk dapat langsung dikelompokkan ke dalam kategori sentimen yang paling sesuai [58].

Model ini terbukti sangat berguna untuk memetakan bagaimana suasana hati publik terhadap masalah besar yang sedang ramai dibicarakan [8]. Ribuan komentar dapat

dikelompokkan menjadi bagian positif atau negatif dengan cara yang jauh lebih mudah [60]. Sistem ini memiliki pemahaman yang baik terhadap bahasa sehari-hari yang sering muncul saat orang berinteraksi di internet [57]. Kemampuannya dalam mengolah teks dengan hasil yang tepat namun tetap hemat tenaga menjadikannya alat yang sangat praktis bagi pengguna [9]. Pada akhirnya, sistem ini membuktikan bahwa model yang lebih kecil tetap bisa memberikan hasil yang sangat baik dalam dunia digital [58].

1.3.3 Backpropagation

Backpropagation merupakan sebuah metode pelatihan pada jaringan saraf tiruan yang berfungsi untuk memperkecil nilai kesalahan prediksi dengan cara menyesuaikan nilai bobot di dalam jaringan secara berkala.[61] Di dalam sistem klasifikasi sentimen, proses *Backpropagation* ini bekerja sebagai tahap evaluasi setelah model selesai menebak label teks. Algoritma ini akan mengirimkan laporan tingkat kesalahan dari lapisan *output* kembali ke lapisan jaringan di belakangnya, sehingga model bisa belajar dan menjadi lebih adaptif pada proses pengujian berikutnya.[62]

Secara matematis, proses penyesuaian tersebut diawali dengan menghitung nilai kerugian total atau *Loss Function* menggunakan metode *Categorical Cross-Entropy Loss*. [63] Nilai kerugian dari keseluruhan sampel merupakan hasil akumulasi selisih antara probabilitas prediksi model dengan label sentimen target asli sesuai dengan persamaan (17) berikut [64]:

$$L = - \sum_{i=1}^M y_i \cdot \log(\hat{y}_i) \dots \dots \dots (17)$$

Definisi notasi pada persamaan (17) adalah sebagai berikut [64]:

L = Nilai kerugian (*Loss value*) total yang dihasilkan.

M = jumlah kelas kategori.

y_i = Nilai probabilitas aktual atau label target pada kelas ke- i

$\hat{y}_i$ = Nilai probabilitas hasil prediksi model untuk kelas ke- i

Secara matematis, setelah nilai kesalahan global diketahui, nilai selisih kesalahan atau *Error* dari sebuah sampel teks merupakan hasil pengurangan langsung antara hasil probabilitas tebakan model dengan nilai aktual targetnya. Komponen *Error* ini diformulasikan secara ringkas melalui persamaan (18) berikut:

$$Error = \hat{y} - y \dots \dots \dots (18)$$

Definisi notasi pada persamaan (18) adalah sebagai berikut:

Error = Nilai selisih kesalahan dari hasil tebakan model.

$\hat{y}$ = Nilai probabilitas hasil tebakan yang dikeluarkan oleh model.

y = Nilai label target asli dari data teks.

Secara matematis, nilai selisih kesalahan tersebut kemudian diolah kembali untuk mencari arah perbaikan bobot jaringan yang dikenal dengan istilah *Gradien*. [65] Pada arsitektur model yang memproses urutan kalimat, nilai gradien ini didapatkan dari hasil perkalian antara nilai *Error* dengan kondisi tersembunyi atau *hidden state* (h_t) yang menyimpan informasi urutan kata. [63] Rumus pembentukan komponen gradien ini dituliskan melalui persamaan (19) berikut:

$$\text{Gradien} = \text{Error} \times h_t \dots\dots\dots(19)$$

Definisi notasi pada persamaan (19) adalah sebagai berikut :

Gradien = Nilai arah acuan untuk memperbaiki bobot jaringan.

Error = Nilai selisih kesalahan dari hasil tebakan model.

h_t = Nilai komponen *hidden state* yang menyimpan informasi urutan kata.

Secara matematis, nilai bobot baru yang dihasilkan pada tahap akhir merupakan hasil reduksi dari bobot lama oleh perkalian antara parameter laju *learning rate* dengan nilai komponen gradien. [62] Proses pembaruan nilai bobot ini dioperasionalkan melalui persamaan (20) berikut [61]:

$$W_{\text{baru}} = W_{\text{lama}} - (\eta \times \text{Gradien}) \dots\dots\dots(20)$$

Definisi notasi pada persamaan (20) adalah sebagai berikut [61]:

W_{baru} = Nilai bobot baru hasil akhir perbaikan model.

W_{lama} = Nilai bobot lama sebelum diperbaiki.

η = Nilai laju pembelajaran (*learning rate*) untuk mengatur kecepatan perubahan bobot.

Gradien = Nilai arah acuan untuk memperbaiki bobot jaringan.

Seluruh rangkaian hitungan umpan balik ini akan terus berjalan secara berulang pada setiap siklus pelatihan data sampai model berhasil meminimalkan nilai kesalahan hingga mendekati angka nol dan menghasilkan tingkat akurasi klasifikasi sentimen yang paling optimal. [64]

1.4 Confusion Matrix

Confusion matrix merupakan metode evaluasi yang digunakan untuk mengukur kinerja suatu model klasifikasi dengan cara membandingkan label aktual pada data uji dengan hasil prediksi yang dikeluarkan oleh sistem [28]. Secara fundamental, metode ini pertama-tama dikembangkan untuk klasifikasi biner yang membandingkan dua kategori, di

mana komponen utamanya terdiri dari *True Positive* (TP) sebagai data positif yang diprediksi benar, *True Negative* (TN) sebagai data negatif yang diprediksi benar, *False Positive* (FP) sebagai data negatif yang salah diprediksi sebagai positif, dan *False Negative* (FN) sebagai data positif yang salah diprediksi sebagai negatif [62]. Struktur hubungan antar komponen biner tersebut disajikan dalam Tabel 2.1 sebagai berikut [28]:

Tabel 1.1 Struktur Confusion Matrix

	Prediksi Positif	Prediksi Negatif
Aktual Positif	True Positif (TP)	False Negatif (FN)
Aktual Negatif	False Positif (FP)	True Negatif (TN)

Nilai-nilai yang terdapat pada tabel tersebut kemudian digunakan untuk menghitung metrik performa klasifikasi. Akurasi berfungsi untuk mengukur tingkat kebenaran prediksi model secara keseluruhan melalui persamaan (21), presisi digunakan untuk mengukur ketepatan antara data yang diminta dengan hasil prediksi sistem melalui persamaan (22), *recall* bertujuan untuk mengukur keberhasilan model dalam menemukan kembali informasi melalui persamaan (23), serta *F1-score* yang berfungsi sebagai rata-rata harmonis untuk menyeimbangkan nilai presisi dan *recall* melalui persamaan (24) berikut [66]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(21)$$

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(22)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(23)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision+ Recall} \dots\dots\dots(24)$$

Berdasarkan rangkaian formula di atas, definisi notasi yang digunakan adalah sebagai berikut [66]:

Accuracy: Tingkat akurasi atau kebenaran prediksi secara keseluruhan.

Precision: Nilai presisi atau ketepatan hasil prediksi positif.

Recall: Nilai perolehan kembali atau sensitivitas model terhadap data aktual.

F1 - Score: Nilai rata-rata harmonis antara precision dan recall.

Dalam penelitian ini, klasifikasi dilakukan untuk tiga kategori sentimen (Positif, Negatif, dan Netral), sehingga metode evaluasi yang digunakan bersifat *multi-class*. Untuk mengidentifikasi performa setiap label secara terpisah, diterapkan pendekatan *One-vs-Rest* (OvR) dengan mengevaluasi satu kelas spesifik sebagai target utama dan kelas lainnya digabungkan sebagai kelompok di luar target. Pada pendekatan ini, nilai TP merupakan

jumlah data aktual kelas target yang diprediksi sebagai kelas target, sedangkan TN adalah keseluruhan data aktual di luar target yang diprediksi bukan sebagai kelas target, yang mencakup hasil prediksi pada kelas lain beserta klasifikasi silang antar kelas luarnya. Selanjutnya, nilai FP mendefinisikan data aktual di luar target yang diprediksi sebagai kelas target, sementara FN adalah data aktual kelas target yang diprediksi sebagai kelompok luar. Berdasarkan parameter tersebut, distribusi data dari masing-masing kelas dievaluasi menggunakan persamaan biner untuk menghasilkan nilai *precision*, *recall*, dan *f1-score* secara spesifik pada tiap label. Setelah performa setiap kelas diketahui, evaluasi tahap akhir dilakukan untuk melihat kinerja model secara keseluruhan melalui perhitungan akurasi *multi-class* melalui persamaan (25), *Macro Average* melalui persamaan (26), dan *Weighted Average* melalui persamaan (27). *Macro Average* menghitung rata-rata metrik setiap kelas secara mandiri tanpa memandang jumlah data, sedangkan *Weighted Average* menghitung rata-rata dengan memberikan bobot berdasarkan rasio jumlah data aktual (*support*) untuk menyesuaikan dengan distribusi data yang ada. Rumus untuk evaluasi tersebut adalah sebagai berikut [62]:

$$\text{Multi-classAccuracy} = \frac{\sum_{i=1}^N TP_i}{\text{TotalData}} \dots\dots\dots(25)$$

$$\text{MacroAverage} = \frac{1}{N} \sum_{i=1}^N \text{Metrik}_i \dots\dots\dots(26)$$

$$\text{WeightedAverage} = \sum_{i=1}^N (W_i \times \text{Metrik}_i) \dots\dots\dots(27)$$

Berdasarkan rangkaian formula di atas, definisi notasi yang digunakan adalah sebagai berikut [62]:

TP_i : Jumlah prediksi benar pada setiap kelas.

$TotalData$: Keseluruhan jumlah sampel dokumen atau teks yang diuji.

N : Jumlah total kategori kelas klasifikasi (sebanyak 3 kelas).

$Metrik_i$: Nilai performa evaluasi (Presisi, Recall, atau F1-Score) pada kelas.

W_i : Bobot proporsi kelas atau jumlah data kelas yang dibagi Total Data.

1.5 IKN

Pembangunan Ibu Kota Nusantara (IKN) di Kalimantan Timur merupakan langkah strategis pemerintah yang membawa perubahan besar dalam struktur pembangunan nasional [32]. Sebagai proyek mega nasional, transisi pusat pemerintahan ini memicu diskusi yang sangat luas di berbagai kalangan masyarakat Indonesia [67]. Informasi mengenai perkembangan proyek ini dapat diakses secara terbuka, sehingga memunculkan berbagai reaksi yang disampaikan melalui kanal digital [29]. Beragam pandangan yang muncul

mencakup harapan akan pemerataan ekonomi hingga masukan mengenai jalannya kebijakan di lapangan [31]. Dinamika percakapan ini mencerminkan bagaimana masyarakat secara aktif memantau setiap tahapan transformasi besar yang sedang dilakukan oleh pemerintah [67].

Komentar yang diberikan masyarakat terhadap IKN sering kali berisi ungkapan setuju maupun tidak setuju terhadap proses pemindahan ibu kota tersebut [67]. Berbagai tanggapan warga di media sosial juga mencakup keresahan serta harapan yang disampaikan secara langsung melalui pesan singkat [32]. Pengolahan data teks menunjukkan adanya beragam emosi, di mana ungkapan rasa suka cita dan dukungan tetap muncul di antara kritik yang disampaikan oleh warga [31]. Pemanfaatan teknologi komputer dalam memetakan suara publik ini terbukti membantu dalam memberikan gambaran cepat mengenai kecenderungan sikap masyarakat secara luas [29]. Dengan memahami pola komentar ini, cara penyampaian informasi mengenai perkembangan IKN dapat disesuaikan agar lebih mudah diterima oleh seluruh lapisan rakyat [32].

1.6 Media Sosial

Platform media sosial X, yang sebelumnya dikenal dengan nama Twitter, merupakan salah satu media sosial yang memiliki pengaruh besar dan basis pengguna yang sangat luas di seluruh dunia [29]. Platform ini pertama kali diperkenalkan pada tahun 2006 dan berfungsi sebagai sarana utama bagi masyarakat untuk berbagi informasi dan menyampaikan opini secara ringkas [68]. Namun, platform ini kini telah mengalami perubahan besar secara resmi dengan berganti nama dan identitas dari Twitter menjadi aplikasi X [69].

Alasan utama pemanfaatan platform X adalah kemampuannya dalam menyediakan data yang mencerminkan sikap dan emosi masyarakat secara langsung dan cepat [29]. Karakteristik platform ini memungkinkan adanya hubungan interaksi yang dinamis, sehingga informasi dapat tersebar luas dan diakses dengan mudah dalam waktu singkat [68]. Melalui jutaan data teks yang dihasilkan setiap hari, tren yang sedang berkembang dapat dipantau berdasarkan emosi dan sikap manusia dari berbagai latar belakang budaya maupun bahasa [31]. Selain itu, platform ini dianggap sebagai forum terbuka yang sangat efektif untuk mengumpulkan pendapat asli publik guna kebutuhan pengolahan data di era digital [29]. Dengan dukungan teknologi komputer, pesan-pesan singkat di platform X ini akhirnya dapat dikelompokkan untuk memetakan dinamika bahasa manusia secara efektif dan akurat [31].

1.7 Metode *Waterfall*

Metode *Waterfall* merupakan suatu model yang diterapkan dalam pengembangan perangkat lunak yang memiliki struktur kerja yang teratur dan berurutan, di mana setiap tahap harus diselesaikan sepenuhnya sebelum bergerak ke fase selanjutnya. Pendekatan yang terencana ini terbukti sangat efektif dalam menyusun alur kerja secara bertahap untuk mengurangi kemungkinan munculnya kesalahan dalam sistem [70].

Dalam studi ini, metode *Waterfall* diadaptasi ke dalam tahapan pelaksanaan yang berurutan, dimulai dari studi literatur, pengumpulan data, hingga tahap analisis dan perancangan. Tahap analisis dilakukan untuk mengidentifikasi kebutuhan sistem secara rinci, termasuk kebutuhan fungsional dan proses kerja dalam mengolah data teks opini. Hasil analisis tersebut kemudian diterjemahkan pada tahap perancangan untuk mendesain antarmuka pengguna serta merancang struktur alur kerja sistem berbasis website secara menyeluruh [71].

Setelah rancangan sistem selesai, tahapan berlanjut pada fase implementasi yang berfokus pada penulisan kode bahasa pemrograman Python untuk melatih arsitektur kecerdasan buatan berdasarkan desain yang telah dibuat [71]. Rangkaian pengembangan perangkat lunak ini pada akhirnya ditutup dengan tahap pengujian sistem secara komprehensif [70]. Proses pengujian ini sangat krusial untuk memverifikasi fungsionalitas aplikasi secara keseluruhan, serta memvalidasi tingkat akurasi dan performa dari model klasifikasi yang telah dibangun agar siap untuk digunakan [70].