

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Tabel 1 merangkum beberapa penelitian terdahulu yang berkaitan dengan topik penelitian ini.

Tabel 1 Penelitian Terkait

No	Nama dan Tahun	Judul	Subjek	Hasil
1	Abdul Rosyd, Ade Irma Purnamasari, dan Irfan Ali (2024) [5]	Penerapan Metode <i>Long Short-Term Memory</i> (LSTM) dalam Memprediksi Harga Saham PT Bank BCA	Saham BCA (Low)	LSTM efektif dalam menganalisis dan memprediksi pergerakan harga saham dengan nilai MAPE sebesar 0.71%, RMSE sebesar 40.85, dan MSE sebesar 6662.76.
2	Putra dan Daffa Khalish Hardy (2024) [6]	Analisis Perbandingan Model Prediksi Harga Saham melalui LSTM dan ARIMA	Saham AMRT dan MIDI (Close)	LSTM lebih akurat dengan RMSE terendah 15,00 dan akurasi 99.5% untuk saham Alfamidi.
3	Julian, Roby, dan Muhammad Rizky Pribadi (2021) [8]	Peramalan Harga Saham Pertambangan pada BEI Menggunakan <i>Long Short-Term Memory</i> (LSTM)	Saham ANTM, TINS, INCO (Close)	RMSE terkecil: TINS (31.76 pada <i>epoch</i> 200), RMSE terbesar: INCO (139.67 pada <i>epoch</i> 50).

No	Nama dan Tahun	Judul	Subjek	Hasil
4	Budiprasetyo, Gunawan, Mamluatul Hani'ah, dan Darin Zahira Aflah (2023) [9]	Prediksi Harga Saham Syariah Menggunakan Algoritma <i>Long Short-Term Memory</i> (LSTM)	Saham ANTM, ERAA, KLBF, SMGR, WIKA	MAPE terbaik: KLBF (1.51%), SMGR (1.83%), ERAA (2.24%). LSTM menghasilkan prediksi akurat pada berbagai pola <i>time step</i> dan <i>epoch</i> .
5	Kwanda, Kevin, Dyah Erny Herwindiati, dan Manatap Dolok Lauro (2024) [10]	Perbandingan LSTM dan <i>Bidirectional LSTM</i> pada Sistem Prediksi Harga Saham Berbasis <i>Website</i>	Saham PT BRI (Persero) Tbk, PT Adaro Energy Indonesia Tbk, PT BCA Tbk, PT Aneka Tambang Tbk, dan PT Timah Tbk, PT Bank Mandiri (Persero) Tbk,.	Model LSTM menunjukkan kinerja yang sedikit lebih baik dibandingkan dengan model BiLSTM dalam memprediksi harga saham PT Bank Rakyat Indonesia (BBRI), dengan perbedaan sekitar 0,14% pada nilai MAPE dan 0,0009 pada nilai MAE.
6	Arfan, Adhib, dan Lussiana ETP. (2020) [11]	Perbandingan Algoritma <i>Long Short-Term Memory</i> dengan SVR pada Prediksi Harga Saham di Indonesia	Saham UNVR.JK, KAEF.JK, dan GGRM.JK	LSTM berhasil memprediksi harga saham untuk periode 2017–2019 dengan akurasi yang tinggi dan tingkat <i>error</i> yang rendah, serta menunjukkan nilai <i>loss</i> yang lebih unggul dibandingkan algoritma SVR.

Dari penelitian-penelitian terdahulu yang disajikan pada Tabel 1, dapat dilihat bahwa algoritma *Long Short-Term Memory* (LSTM) secara konsisten menunjukkan performa yang unggul dalam melakukan prediksi harga saham. Penelitian-penelitian tersebut mendukung pemilihan algoritma LSTM sebagai algoritma yang akan diimplementasikan dalam pembangunan model prediksi pada penelitian ini. Diharapkan penelitian ini dapat memberikan hasil yang sebanding atau lebih baik dari penelitian-penelitian sebelumnya.

2.2 Keterkaitan Penelitian

Penelitian kali ini memiliki keterkaitan dengan penelitian-penelitian terdahulu yang juga menerapkan algoritma *Long Short-Term Memory* (LSTM) dalam melakukan prediksi harga saham. Namun penelitian ini memperluas penerapan algoritma LSTM dalam memprediksi harga saham PT Astra Internasional Tbk (ASII.JK) yang belum banyak dibahas dalam literatur sebelumnya. Dengan mengimplementasikan dua lapisan LSTM dan evaluasi menyeluruh menggunakan metrik evaluasi MAPE, RMSE, dan MAE, penelitian ini diharapkan dapat berkontribusi dalam menambah wawasan baru mengenai penerapan LSTM serta menambah perspektif baru terhadap prediksi harga saham pada sektor otomotif.

2.3 PT Astra Internasional Tbk (ASII.JK)

PT Astra Internasional adalah salah satu perusahaan terbesar di industri otomotif di Indonesia. Perusahaan multinasional ini didirikan pada tahun 1957, kegiatannya berada di berbagai kota di Indonesia dengan total lebih dari 200 anak perusahaan, mitra dan usaha patungan, serta lebih dari 200.000 karyawan [12]. Kinerja saham PT Astra Internasional Tbk (ASII.JK) pada kuartal III tahun 2024 cukup baik dengan kenaikan pendapatan sebesar 2,2% menjadi Rp246,3 triliun dan kenaikan laba bersih sebesar 0,63% menjadi Rp25,85 triliun apabila dibandingkan dengan periode yang serupa pada tahun sebelumnya [13]. Pergerakan saham ASII cukup stabil di rentang Rp4.900 – Rp4.940 dengan volume harian 17,9 juta per lembar [14]. Namun harga tersebut menunjukkan penurunan sebesar 8,44% dalam setahun kemarin karena adanya fluktuasi pasar dan tantangan sektor otomotif [15]. Saham ASII yang mencerminkan industri otomotif Indonesia dan merupakan perusahaan

yang berpengaruh di pasar modal Indonesia, membuat saham ini penting untuk dijadikan studi kasus.

2.4 Prediksi Harga Saham

Prediksi harga saham adalah proses yang melibatkan analisis dan penentuan nilai harga suatu saham. Prosesnya sering dilakukan dengan memprediksi harga numerik atau memprediksi arah harga. Dalam prediksi harga numerik, untuk memprediksi harga sebenarnya dari suatu saham digunakan model pembelajaran seperti regresi, sedangkan dalam prediksi arah harga, digunakan model pembelajaran seperti *classifier* dalam memprediksi arah naik turunnya pergerakan harga saham [16].

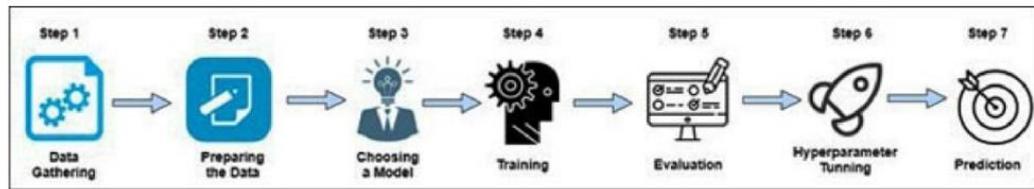
Saham sendiri merupakan instrumen keuangan yang menunjukkan hak milik individu maupun organisasi pada sebuah entitas bisnis. Hal ini berarti setiap investor saham memiliki bagian dari suatu perusahaan tersebut. Berdasarkan sejumlah karya ilmiah, harga saham merepresentasikan nilai sebuah perusahaan dan mencatatkan kekayaan yang dimiliki oleh perusahaan penerbitnya [5]. Saham memiliki kaitan yang searah dan berkesinambungan dengan tingkat *return* yang diharapkan dari suatu investasi, sehingga risiko yang dihadapi oleh seorang investor cukup besar [17].

Harga saham sangat fluktuatif sepanjang waktu, hal itu dipengaruhi oleh level penawaran dan permintaan pada saham itu sendiri. Selain itu, beberapa faktor lain yang juga ikut mempengaruhi perubahan harga saham menurut Ginting & Munthe pada tahun 2017 adalah laba per lembar saham, tingkat suku bunga, jumlah kas dan dividen yang dibagikan, kebijakan distribusi dividen, jumlah laba yang di peroleh, serta tingkat risiko dan imbal hasilnya [18].

2.5 Pembelajaran Mesin

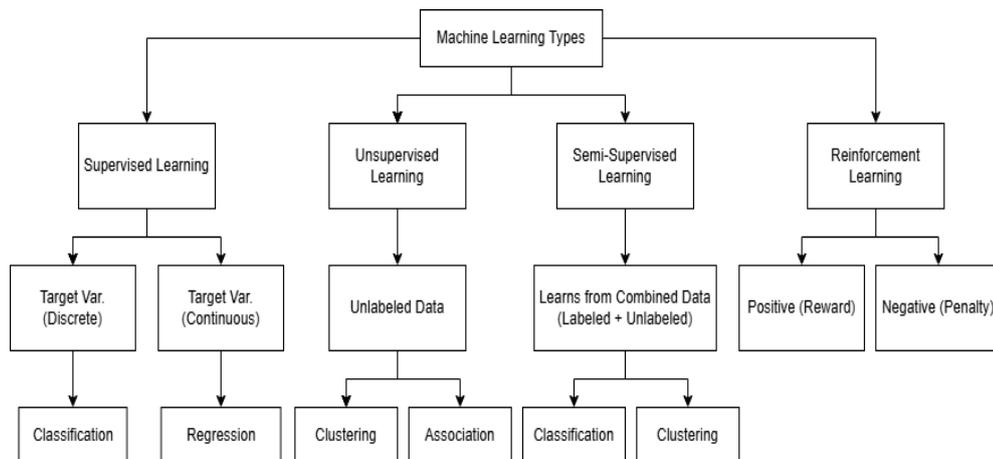
Pembelajaran Mesin/*Machine Learning* merupakan cabang *Artificial Intelligent*. Teknik pembelajaran mesin mencakup beberapa bidang interdisipliner seperti matematika, statistik, jaringan saraf buatan, data mining, dan kecerdasan buatan. Pembelajaran mesin dapat dianggap sebagai suatu metode pemrograman komputer yang memanfaatkan data historis untuk membuat model, dan memungkinkan proses pembelajaran dengan tujuan mencapai performa terbaik dalam menggali

informasi dari suatu kumpulan data [3]. Terdapat 7 langkah pada proses pembelajaran mesin yang disajikan pada Gambar 1.



Gambar 1 *Steps of Machine Learning*
Sumber : [19]

Proses pembelajaran mesin didasarkan pada berbagai algoritma dalam mengatasi persoalan data. Ilmuwan data menyatakan bahwa tak ada satu pun jenis algoritma yang paling cocok untuk semua jenis masalah. Jadi, algoritma yang digunakan pada pembelajaran mesin bergantung pada jenis masalah yang ingin dipecahkan [20]. Algoritma pembelajaran mesin dibagi menjadi empat kategori, yaitu *Supervised*, *Unsupervised*, *Semi-supervised*, dan *Reinforcement* [21].

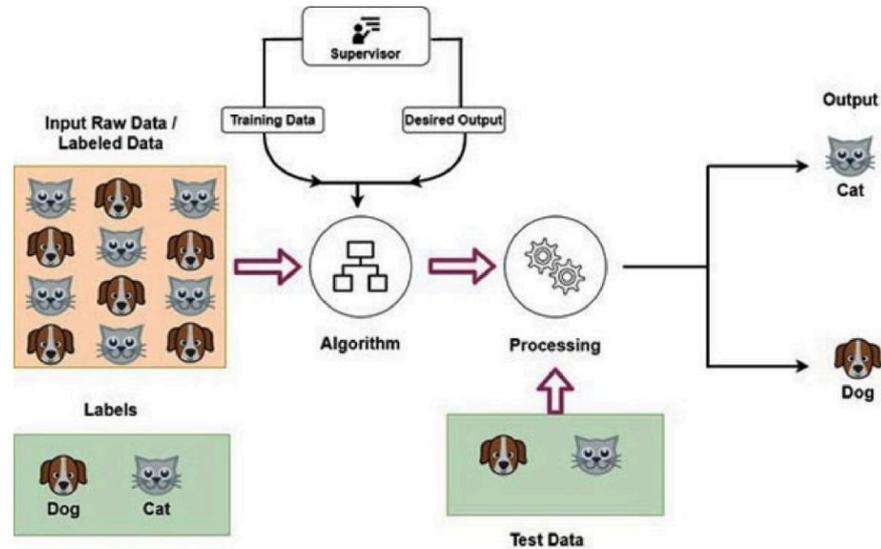


Gambar 2 Diagram Klasifikasi Teknik *Machine Learning*

a. *Supervised Learning*

Supervised Learning adalah metode pembelajaran mesin yang memetakan *input* ke *output* menggunakan data pelatihan berlabel. Tugas utamanya adalah melakukan klasifikasi dan regresi, dimana klasifikasi dilakukan untuk memisahkan data dan regresi untuk menyaring data [21]. Selama proses pelatihan, algoritma mencari pola dalam data *input* dan menghubungkannya dengan *output* yang diinginkan. Setelah dilatih,

algoritma dapat menerima data baru yang belum pernah dikenali sebelumnya dan memprediksi label atau *output* untuk data tersebut berdasarkan apa yang telah dipelajari [19].

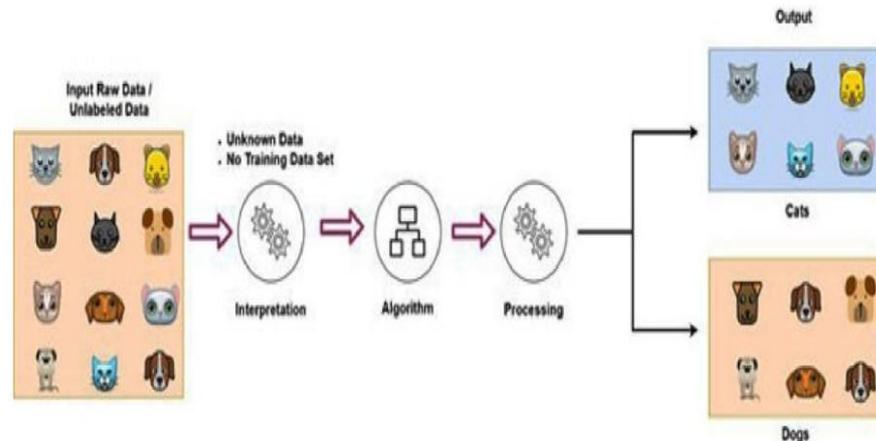


Gambar 3 *Supervised Learning*
Sumber : [19]

Gambar 3 menunjukkan kumpulan gambar kucing dan anjing yang sudah diberi label. Gambar-gambar ini beserta labelnya akan diberikan kepada model agar model dapat belajar untuk membedakan keduanya. Jika model mempelajarinya dengan baik, maka model dapat mengidentifikasi gambar baru yang tanpa label [19].

b. *Unsupervised Learning*

Unsupervised Learning adalah metode pembelajaran mesin yang digunakan untuk pengelompokan, estimasi kepadatan, pembelajaran fitur, pengurangan dimensi, aturan asosiasi, dan deteksi anomali. Metode ini menganalisis data tidak berlabel tanpa intervensi manusia [21]. Berbeda dengan *Supervised Learning*, pembelajaran ini hanya tersedia data *input* tanpa *output* yang sesuai. Dalam metode ini, pengguna tidak perlu mengarahkan atau mengawasi model, karena tidak ada jawaban benar atau panduan dari pengawas. Algoritma akan secara otomatis mempelajari data, menemukan pola, dan mengelompokkan data berdasarkan kesamaan [19].



Gambar 4 *Unsupervised Learning*
Sumber : [19]

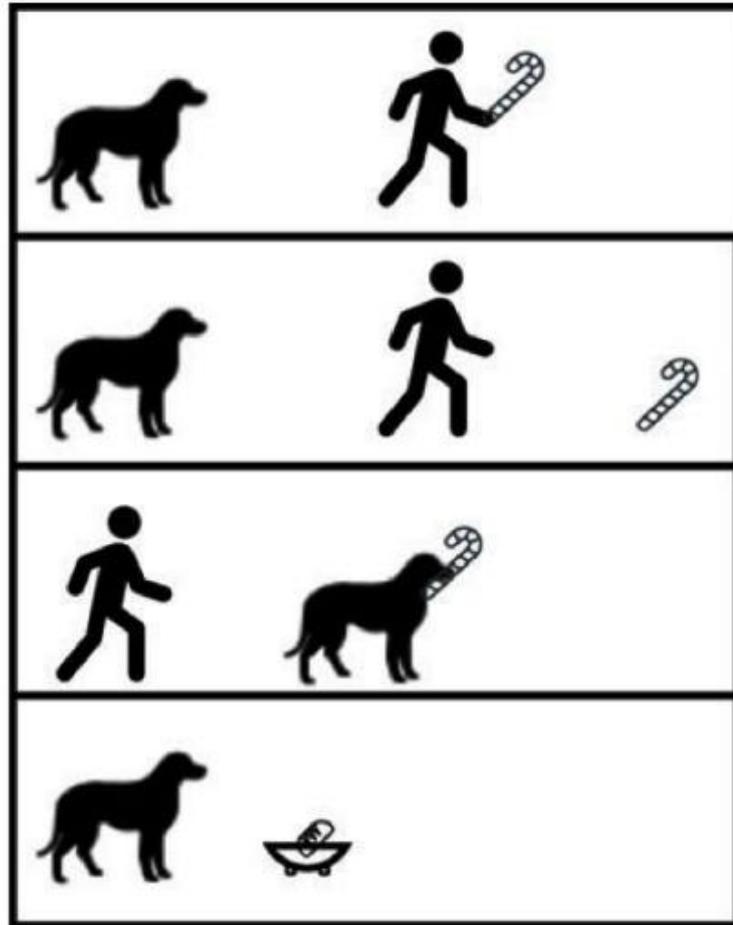
Gambar 4 menunjukkan kumpulan gambar berbagai jenis kucing dan anjing. Algoritma *Unsupervised Learning* akan mencoba memahami fitur-fitur pada gambar tersebut tanpa pelatihan sebelumnya. Tugasnya adalah mengelompokkan gambar-gambar tersebut ke dalam grup berdasarkan kesamaan yang ditemukan [19].

c. *Semi-Supervised Learning*

Semi-Supervised Learning merupakan metode gabungan antara *supervised* dan *unsupervised* yang dapat beroperasi pada data berlabel dan tidak berlabel. Metode ini bertujuan untuk meningkatkan akurasi prediksi dibandingkan jika hanya menggunakan data berlabel [21].

d. *Reinforcement Learning*

Reinforcement Learning merupakan teknik pembelajaran mesin yang memungkinkan agen *software* dan mesin secara otomatis mengevaluasi perilaku optimal dalam konteks tertentu untuk meningkatkan efisiensinya [21]. Mesin atau agen belajar sendiri melalui pengalaman tanpa memerlukan data berlabel. Dalam *Reinforcement Learning*, agen menggunakan proses coba-coba, jika agen mencoba langkah yang berhasil, maka agen akan mendapatkan hadiah, sebaliknya untuk setiap kesalahan, agen akan mendapatkan penalti [19].



Gambar 5 *Reinforcement Learning*
Sumber : [19]

Contohnya seperti pada Gambar 5, *Reinforcement Learning* diibaratkan seperti mengajari seekor anjing. Karena anjing tidak memahami bahasa manusia, maka kita menciptakan situasi dan melihat bagaimana ia merespons. Jika responnya benar, kita memberinya hadiah, seperti roti. Ketika menghadapi situasi serupa, anjing akan mengulangi tindakan yang sama untuk mendapatkan hadiah. Dari pengalaman ini, anjing belajar apa yang harus dilakukan dan menghindari respon yang menghasilkan pengalaman negatif [19]. Pembelajaran ini efektif untuk melatih model AI yang meningkatkan otomatisasi dan efisiensi dalam sistem kompleks seperti robotika, kendaraan otonom, manufaktur, dan logistik rantai pasokan [21].

2.6 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) adalah pengembangan dari arsitektur *Recurrent Neural Network* (RNN) yang pertama kali dikenalkan oleh Hochreiter & Schmidhuber (1997). LSTM merupakan spesial unit RNN yang dirancang untuk mengatasi masalah keterkaitan jangka panjang, di mana LSTM memiliki sistem *internal* yang dikenal sebagai *cell states* dan *gates* yang berfungsi mengatur memori untuk setiap input yang diterima. Terdapat empat jenis *gates* yaitu *forget gate*, *input gate*, *cell gate*, dan *output gate* [11]. Proses LSTM diantaranya adalah sebagai berikut : [6]

1. Inisialisasi

Inisialisasi dalam konteks LSTM merupakan tahap awal di mana sistem menyiapkan nilai-nilai awal sebelum memproses data *input* untuk dua komponen penting, yaitu *Cell State* (C_{t-1}) dan *Hidden State* (h_{t-1}). Kedua komponen tersebut diinisialisasi dengan nilai tertentu agar sistem dapat mulai memproses urutan data yang diberikan.

2. Time Step

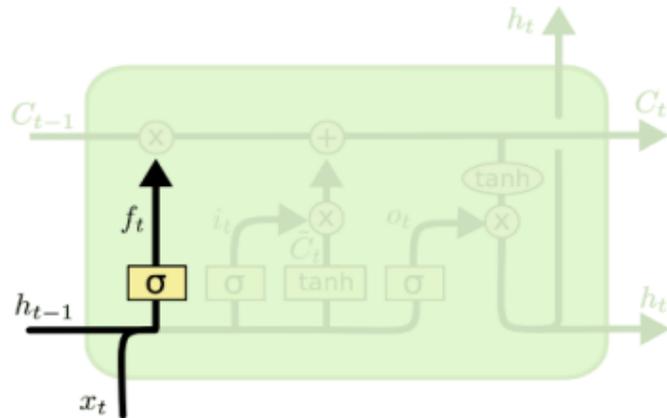
Time Step adalah proses di mana model memproses urutan data dalam langkah waktu satu per satu. Setiap *Time Step* menerima *input* baru dan informasi dari *Hidden State* sebelumnya, untuk menghasilkan prediksi atau pembaruan informasi pada langkah waktu berikutnya.

3. Input dan Hidden State Sebelumnya

Ini adalah tahap di mana model menerima dua jenis informasi pada setiap *Time Step*, yaitu *input* saat ini (x_t) dan *Hidden State* sebelumnya (h_{t-1}). *Input* saat ini merupakan data baru yang masuk pada *Time Step* saat ini, sedangkan *Hidden State* sebelumnya adalah informasi yang di bawa dari langkah waktu sebelumnya.

4. Forget Gate

Forget Gate adalah proses untuk menentukan informasi apa saja yang perlu dihilangkan dari *Cell State* sebelumnya (C_{t-1}). Penentuan dilakukan dengan melakukan perkalian elemen-elemen antara keluaran dari *Forget Gate* dan *Cell State* sebelumnya. Rumus *Forget Gate* dapat dilihat pada Persamaan 1.



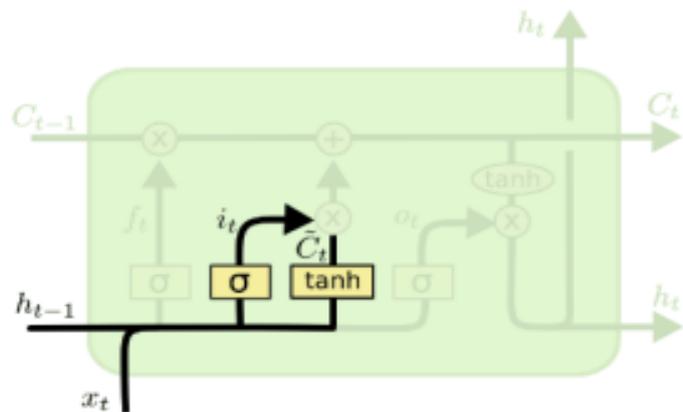
Gambar 6 *Forget Gate Layer*
Sumber : [22]

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

Proses ini memungkinkan model melupakan informasi yang tidak lagi relevan, sehingga hanya akan ada informasi penting yang digunakan dalam prediksi selanjutnya.

5. *Input Gate*

Input Gate adalah proses untuk menetapkan informasi baru apa saja yang akan dimasukkan ke dalam *Cell State*.



Gambar 7 *Input Gate Layer*
Sumber : [22]

Prosesnya melibatkan dua langkah, yaitu :

1. Penentuan nilai calon baru (\tilde{C}_t) yaitu dengan cara mengolah *input* menggunakan fungsi aktivasi tanh. Rumusnya dapat diamati pada Persamaan 2.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

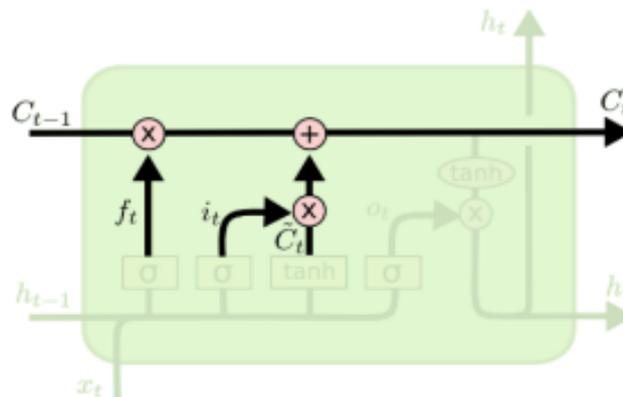
2. Melakukan perkalian antara nilai C_{t-1} dengan keluaran dari *Input Gate*. Rumus dapat dilihat pada Persamaan 3.

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

Proses ini memungkinkan model untuk menyesuaikan informasi relevan yang perlu disimpan dan dipelajari.

6. Pembaruan *Cell State*

Pembaruan *Cell State* adalah proses di mana *Cell State* yang baru, dihitung dengan menjumlahkan hasil dari *Forget Gate* dan *Input Gate*. Rumus pembaruan *Gate* dapat dilihat pada Persamaan 4.



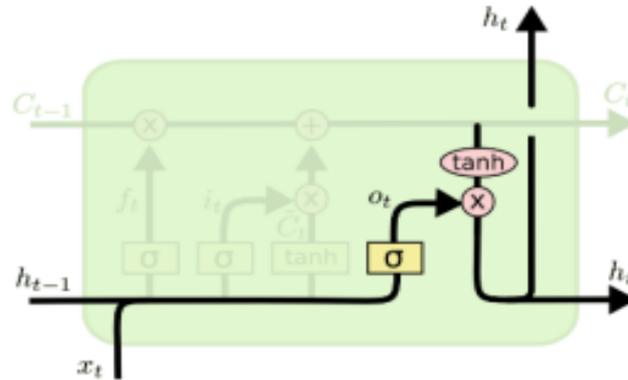
Gambar 8 Pembaruan *Gate Layer*
Sumber : [22]

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (4)$$

Pembaruan ini memungkinkan model mempertahankan data yang relevan dan menghapus data yang tidak relevan, sehingga model dapat mempelajari pola dalam data sekuensial.

7. *Output Gate*

Output Gate adalah proses menetapkan nilai *Hidden State* yang diperoleh pada *Time Step* saat ini.

Gambar 9 *Output Gate Layer*

Sumber : [22]

Prosesnya terdiri dari dua langkah, yaitu :

1. Penentuan nilai *Hidden State* baru (h_t) yaitu dengan cara mengolah *Cell State* yang sudah diperbarui menggunakan fungsi aktivasi tanh.
2. Melakukan perkalian antara nilai hasil dengan keluaran dari *Output Gate* dengan rumus pada Persamaan 5.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

Hidden State yang dihasilkan oleh *Output Gate* ini berfungsi sebagai *output* model untuk *Time Step* tersebut, dan akan digunakan sebagai *input* untuk *Time Step* berikutnya.

8. *Output* dan *Hidden State*

Output dan *Hidden State* adalah proses di mana model LSTM menghasilkan dua komponen penting setelah memproses *input* pada *Time Step* tertentu, yaitu hasil keluaran LSTM pada *Time Step* saat ini (y_t) dan *Hidden State* yang telah diperbaharui (h_t). Rumus pembaruan *Hidden Cell* dapat dilihat pada Persamaan 6.

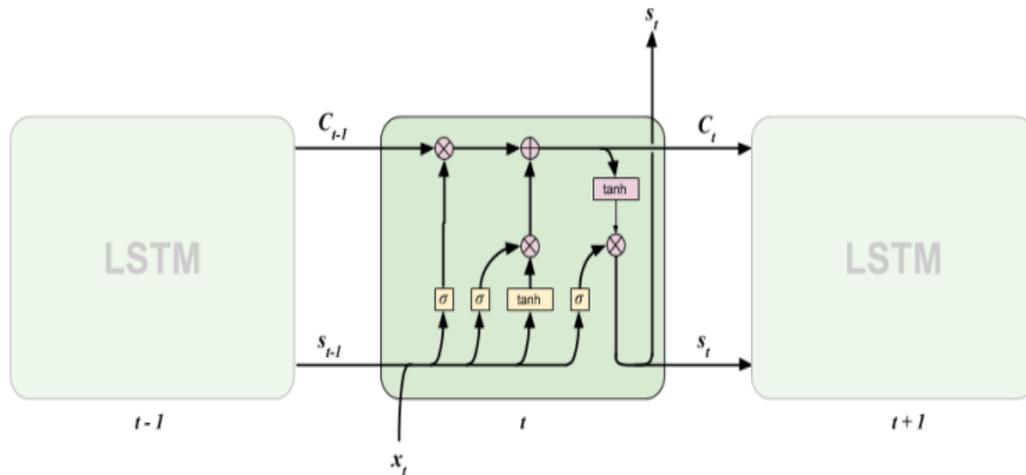
$$h_t = o_t * \tanh(C_t) \quad (6)$$

Proses ini berfungsi untuk menghasilkan hasil pada *Time Step* tersebut dan mempertahankan informasi penting untuk proses prediksi selanjutnya.

9. Iterasi

Iterasi adalah proses pengulangan langkah-langkah pemrosesan pada setiap *Time Step* hingga seluruh urutan data selesai diproses.

Gambar 10 merupakan arsitektur LSTM yang memberikan representasi visual tentang cara kerja *output gate*, *input gate*, dan *forget gate* yang memungkinkan sel mempertahankan informasi penting sambil mengatur arus informasi [23].



Gambar 10 Arsitektur Model LSTM
Sumber :[8]

2.7 Python

Python tumbuh dari bahasa yang disebut ABC, yang dikembangkan di Belanda pada tahun 1980-an sebagai alternatif untuk BASIC. Bahasa ini dirancang untuk membantu para ilmuwan memanfaatkan mikrokomputer, yang merupakan teknologi baru pada saat itu. Python adalah bahasa pemrograman yang bekerja dengan cara ditafsirkan, berbeda dengan bahasa pemrograman seperti C atau FORTRAN yang memerlukan proses kompilasi. Dalam python, kode hanya bisa dijalankan jika ada interpreter Python yang aktif. Interpreter ini berfungsi sebagai penghubung antara kode program dan sistem tempat program tersebut dijalankan. Tugasnya adalah menerjemahkan instruksi dari kode Python agar bisa dipahami dan dijalankan oleh sistem, sekaligus dapat menangani hal-hal teknis terkait *platform* tersebut [24].

2.8 *Min-Max Scaler Normalization*

Normalisasi merupakan teknik yang digunakan untuk menjamin bahwa semua data memiliki ukuran yang sama. Teknik normalisasi *Min-Max Scaler* menjadi salah satu normalisasi terbaik untuk meningkatkan kinerja pengklasifikasian [25].

Normalisasi *Min-Max Scaler* sangat berguna dalam mengatasi masalah pada data berdimensi tinggi. *Min-Max Scaler* sendiri adalah metode normalisasi yang mengubah semua nilai menjadi rentang 0 hingga 1. Proses normalisasinya dapat dilihat pada Persamaan 7 dan 8 [26].

$$X_{std} = \frac{(X - X.min)}{(X.max - X.min)} \quad (7)$$

Keterangan :

X_{std} = Hasil normalisasi terhadap data x

x = Data x

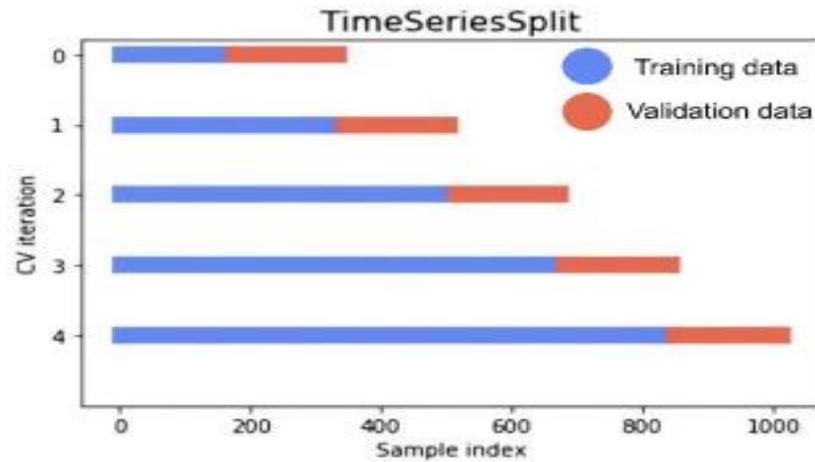
$x.min$ = Data minimum yang terdapat pada x

$x.max$ = Data maximum yang terdapat pada x

Dengan menerapkan teknik normalisasi *Min-Max Scaler*, data yang memiliki rentang nilai sangat berbeda dapat diselaraskan menjadi skala yang seragam.

2.9 *TimeSeriesSplit Cross Validation*

TimeSeriesSplit Cross Validation merupakan metode validasi yang banyak digunakan dalam analisis data *time series* untuk mempertahankan keteraturan temporal data. Metode ini membagi data *time series* menjadi beberapa lipatan (*fold*) dengan tetap menjaga urutan waktu [27]. Pada setiap lipatan, ukuran data pelatihan bertambah seiring bertambahnya waktu. Namun, kondisi ini dapat menyebabkan *hyperparameter* yang optimal pada lipatan kecil menjadi kurang sesuai untuk seluruh dataset. Untuk mengurangi dampaknya, dapat diberikan bobot berbentuk linier, kuadratik, atau pola lain pada setiap lipatan berdasarkan ukuran data latihnya, agar lipatan dengan data latih lebih besar memiliki pengaruh lebih besar dalam pemilihan model [28].



Gambar 11 *TimeSeriesSplit Cross Validation*
Sumber : [28]

2.10 MAPE, RMSE, dan MAE

Mean Absolute Percentage Error (MAPE), *Root Mean Squared Error* (RMSE), dan *Mean Absolute Error* (MAE) merupakan metrik evaluasi yang paling sering digunakan [7], yang dimanfaatkan untuk mengetahui hasil kerja model. MAPE mengukur persentase rata-rata kesalahan prediksi, di mana semakin kecil nilainya maka semakin baik kinerja modelnya. Formula MAPE dapat dilihat pada Persamaan 9 [5].

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_{\text{test}} - \text{predictions}}{y_{\text{test}}} \right| \times 100 \quad (8)$$

Untuk kriteria penilaiannya menurut Chang et al. (2007), terbagi menjadi 4 kategori yang disajikan pada Tabel 2 [29].

Tabel 2 Kriteria Penilaian MAPE

Nilai MAPE	Akurasi Prediksi
< 10%	Sangat baik
10% - 20%	Baik
20% - 50%	Cukup
>50%	Buruk

RMSE mengevaluasi sejauh mana model mampu memprediksi nilai yang sebenarnya, semakin rendah nilainya maka semakin baik kualitas prediksinya. Formula RMSE dapat dilihat pada Persamaan 10 [5].

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\text{ytest} - \text{predictions})^2} \quad (9)$$

Menurut Normah et.al. tahun 2022, semakin rendah nilai RMSE maka hasil prediksi dianggap semakin baik [30].

MAE mengambil nilai absolut pada setiap nilainya untuk menghilangkan nilai-nilai negatif. Untuk mengevaluasi model prediksi, MAE memberikan gambaran rata-rata error dari keseluruhan data secara intuitif [31]. Formula MAE dapat dilihat pada Persamaan 11.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\text{ytest} - \text{predictions}| \quad (10)$$

Menurut Suryanto (2019), kemampuan model dalam melakukan prediksi juga dianggap semakin baik jika nilai MAE semakin kecil [30].