

BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

2.1.1. Machine Learning

Machine learning merupakan salah satu cabang dari kecerdasan buatan yang berfokus pada pengembangan algoritma dan model yang mampu belajar dari data. Melalui machine learning, sistem dapat mengenali pola, melakukan klasifikasi, dan membuat prediksi tanpa harus diprogram secara eksplisit. Proses ini memungkinkan komputer untuk meningkatkan kinerjanya seiring dengan bertambahnya pengalaman dari data yang diolah.

Beragam teknik dan algoritma diterapkan dalam machine learning, bergantung pada jenis permasalahan atau tujuan analisis. Sebagai contoh, untuk melakukan prediksi penjualan di masa mendatang, dapat digunakan metode seperti regresi atau jaringan saraf tiruan (*neural network*), termasuk model *Long Short-Term Memory* (LSTM) dan ARIMA. Untuk keperluan pengelompokan data, algoritma seperti *K-Means* digunakan, sementara untuk klasifikasi, teknik seperti *Decision Tree*, *Support Vector Machine* (SVM), atau *Naive Bayes* banyak diterapkan. Adapun dalam pencarian pola hubungan atau asosiasi antaritem, algoritma seperti Apriori atau *FP-Growth* menjadi pilihan yang umum digunakan.

2.1.2. Recurrent Neural network (RNN)

Recurrent Neural network adalah jenis jaringan saraf tiruan yang dirancang untuk memproses data berbentuk urutan atau sekuens, seperti teks, suara, atau data waktu (Deleviar et al., 2025). RNN memiliki kemampuan untuk menyimpan

informasi dari *Input* sebelumnya, memungkinkan jaringan ini untuk memprediksi hasil berdasarkan data yang berurutan.

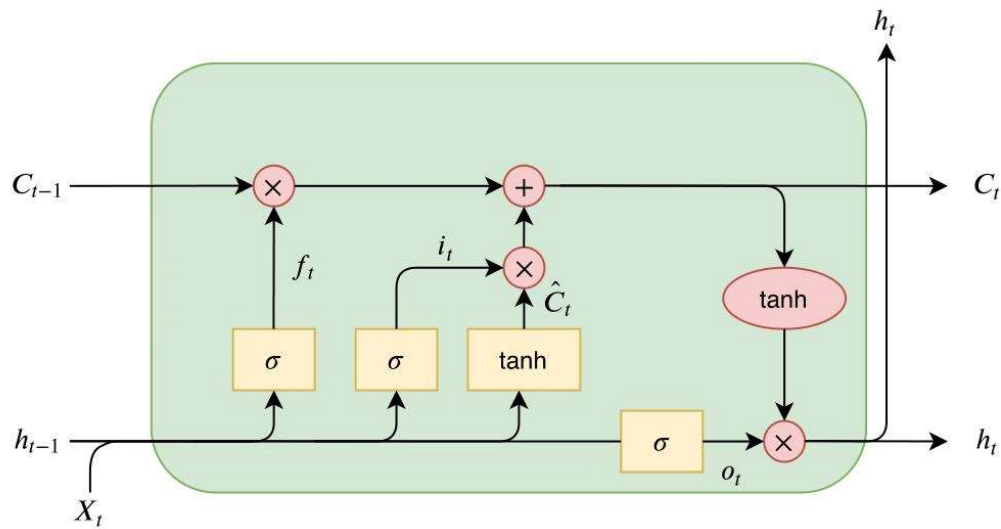
Namun, RNN tradisional menghadapi tantangan besar dalam mempertahankan informasi untuk jangka panjang. Masalah seperti *vanishing gradient* atau *exploding gradient* sering terjadi, di mana informasi yang lebih jauh dalam urutan menjadi sangat lemah atau berlebihan ketika diperbarui. Untuk mengatasi hal ini, dikembangkan *Long Short-Term Memory*, yang merupakan varian dari RNN yang dirancang untuk mengatasi masalah tersebut dan memungkinkan jaringan untuk menyimpan informasi dalam jangka waktu lebih lama. LSTM menggunakan komponen yang disebut "memori sel" yang berfungsi untuk menyimpan informasi sepanjang urutan.

2.1.3. LSTM (*Long Short-Term Memory*)

Long Short-Term Memory adalah jenis jaringan saraf yang merupakan bagian dari *Recurrent Neural Networks*. LSTM dirancang untuk mengatasi masalah *vanishing gradient* yang sering muncul pada RNN tradisional ketika menangani urutan data panjang (Sidiq & Nurzaman, 2025). LSTM memiliki kemampuan untuk menyimpan informasi dalam jangka waktu lama dengan memanfaatkan tiga komponen utama: *Input gate*, *Forget gate*, dan *Output gate*. Hal ini memungkinkan LSTM untuk mempertahankan informasi penting sepanjang urutan data dan mengabaikan informasi yang tidak relevan.

Pada unit LSTM, terdapat tiga gerbang yang mengontrol aliran informasi dalam *Cell state*, yaitu *Input gate*, *Forget gate*, dan *Output gate* sebagaimana ditunjukkan pada Gambar 2.2. *Input gate* (I_t) berfungsi untuk menentukan

informasi yang akan dimasukkan ke dalam *Cell state*. *Forget gate* (F_t) bertanggung jawab untuk menentukan informasi mana yang akan dipertahankan atau dibuang dari *Cell state*. *Output gate* (O_t) berperan dalam memilih informasi yang akan diambil dari *Cell state* untuk dijadikan *Output*. Berikut adalah persamaan yang digunakan dalam metode ini.



Gambar 2. 1 Arsitektur LSTM
(Sumber: Data penelitian,2025)

1. *Forget gate*

Gate ini menentukan bagian informasi yang akan dilupakan dari memori sebelumnya, untuk penggunaan rumus *forget gate* dijelaskan pada rumus 2.1.

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

Keterangan:

f_t = *Forget gate*

σ = fungsi sigmoid

W_f = nilai bobot *Forget gate*

h_{t-1} = nilai *Output* sebelum *state*

Cell

x_t = nilai *Input*

b_f = nilai bias *Forget gate*

2. *Input gate*

Input gate menentukan informasi mana yang akan disimpan dalam memori. *Output* nya adalah nilai antara 0 dan 1 yang mengindikasikan sejauh mana informasi baru diterima dan dimasukkan ke dalam status memori, untuk penggunaan rumus *input gate* dijelaskan pada rumus 2.2.

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

Keterangan:

w_i = nilai bobot *Input gate*

b_i = nilai bias *Input gate*

3. *Candidate Memory Cell*

Cell kandidat menghasilkan nilai calon memori baru yang akan ditambahkan ke status memori, untuk penggunaan rumus *candidate memory cell* dijelaskan pada rumus 2.3.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \textbf{Rumus 2.3 Candidate Memory Cell}$$

Keterangan:

$\tanh$ = membatasi hasil *Output* antara -1 dan 1

b_c = nilai bias *Cell gate*

w_c = nilai bobot *Cell gate*

4. *Update Cell*

Status memori diperbarui dengan menggabungkan hasil dari *Forget gate* dan *Input gate*, untuk penggunaan rumus *update cell* dijelaskan pada rumus 2.4.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad \textbf{Rumus 2.4 Update Cell}$$

Keterangan:

C_t = nilai baru *Cell state*

C_{t-1} = nilai *Cell state* sebelumnya

5. *Output Gate*

Output gate digunakan untuk memperbarui status memori selanjutnya, untuk penggunaan rumus *output gate* dijelaskan pada rumus 2.5.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \textbf{Rumus 2.5 Output gate}$$

Keterangan:

W_o = nilai bobot *Output gate*

b_o = nilai bias *Output gate*

6. *Output*

Output LSTM dihitung dengan mengalikan *Output gate* dengan fungsi aktivasi dari status memori yang diperbarui. Ini menghasilkan nilai yang digunakan untuk prediksi atau keluaran lainnya seperti yang dijelaskan pada rumus 2.6.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \textbf{Rumus 2.6 Output}$$

Keterangan:

h_t = nilai *Output LSTM*

C_t = nilai *Cell state* terbaru

2.1.4. *Preprocessing Data*

Preprocessing data merupakan langkah krusial dalam analisis maupun pemodelan data. Tahap ini melibatkan berbagai proses untuk mengolah data mentah agar siap digunakan dalam model statistik, pembelajaran mesin, atau bentuk analisis lainnya (Manullang et al., 2023). *Preprocessing* data mencakup sejumlah tahapan penting, seperti data cleaning, normalisasi data, sequence data, dan data splitting untuk membedakan antara data pelatihan dan data pengujian. Langkah-langkah ini bertujuan untuk menjamin bahwa data yang digunakan memiliki kualitas yang memadai dan siap untuk digunakan dalam proses pelatihan model LSTM.

1. *Data cleaning* merupakan bagian penting dalam proses *Preprocessing* data yang bertujuan untuk memastikan bahwa dataset yang digunakan memiliki kualitas yang baik sebelum dianalisis lebih lanjut. Salah satu tahapan penting dalam proses ini adalah menghilangkan atribut dan kolom yang tidak di perlukan.
2. Normalisasi data bertujuan untuk menyamakan skala berbagai variabel sehingga nilai-nilainya berada dalam rentang yang seragam. Teknik yang digunakan adalah *Min-Max Normalization* dengan rumus yang ditunjukkan pada Rumus 2.7 berikut ini:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Rumus 2.7 *min-max Normalization*

Keterangan:

X' = nilai hasil normalisasi

X = nilai x

X_{min} = nilai minimum

X_{max} = nilai maksimum

3. *Sequence* data adalah data yang disusun berdasarkan urutan tertentu (biasanya waktu), dan sangat cocok digunakan dalam prediksi deret waktu seperti penjualan harian. LSTM membutuhkan *sequence data* agar bisa belajar dari pola-pola yang terbentuk dari masa lalu untuk memprediksi masa depan. jika jumlah *sequence* terlalu kecil maka pola data yang dipelajari algoritma LSTM akan semakin kecil juga.
4. Data splitting adalah proses membagi dataset menjadi dua bagian, yaitu data training dan data testing. Data training digunakan untuk melatih model agar mengenali pola, sedangkan data testing digunakan untuk menguji kemampuan model dalam memprediksi data yang belum pernah dilihat sebelumnya.

2.1.5. Prediksi

Prediksi merupakan proses memperkirakan kejadian atau nilai yang akan terjadi di masa depan berdasarkan data atau informasi yang tersedia saat ini atau sebelumnya. Proses ini biasanya melibatkan penggunaan model statistik, algoritma, atau metode lainnya untuk menganalisis pola dalam data historis, yang kemudian digunakan untuk meramalkan hasil atau kejadian di masa depan (Wasik et al., 2024).

Dalam analisis data, prediksi banyak diterapkan di berbagai bidang seperti peramalan cuaca, analisis pasar saham, prediksi penjualan, dan lainnya. Berbagai teknik digunakan untuk melakukan prediksi, mulai dari metode statistik yang sederhana hingga algoritma pembelajaran mesin (*machine learning*) dan jaringan saraf tiruan (seperti LSTM atau RNN) yang lebih canggih dan mampu menangani data urutan (*time series*) dengan lebih efektif. Sebagai contoh, prediksi sederhana bisa berupa perkiraan suhu udara besok berdasarkan data suhu sebelumnya, atau memperkirakan jumlah produk yang akan terjual bulan depan berdasarkan data penjualan yang tercatat pada bulan-bulan sebelumnya.

2.1.6. Optimizer

Optimizer merupakan algoritma yang digunakan untuk melatih jaringan saraf tiruan dengan cara memperbarui parameter seperti bobot dan bias, sehingga jaringan dapat mempelajari data pelatihan dan menghasilkan *Output* yang lebih akurat (Sari & Pratama, 2024). Ada berbagai jenis *Optimizer* yang dapat digunakan untuk melatih jaringan saraf tiruan, di antaranya adalah:

1. Adam

Adaptive Moment Estimation adalah *Optimizer* yang populer dalam model pembelajaran mesin seperti LSTM, terutama karena kemampuannya untuk mengadaptasi learning rate berdasarkan rata-rata gradien dan kuadrat gradien. Adam menghitung dua momen-moment pertama (rata-rata gradien) dan moment kedua (rata-rata kuadrat gradien) untuk mengatur learning rate dengan lebih baik, menghindari pembaruan yang terlalu besar atau kecil. Keuntungan utama Adam dalam prediksi penjualan adalah mempercepat konvergensi, menyesuaikan

learning rate secara otomatis, dan memberikan stabilitas, yang sangat berguna ketika data penjualan memiliki fluktuasi musiman atau perubahan tren yang tajam.

2.1.7. Evaluasi Model

Evaluasi model adalah langkah krusial untuk menilai kinerja LSTM dalam memprediksi penjualan (Tamami & Arifin, 2024). Tanpa evaluasi yang akurat, sulit untuk menentukan sejauh mana model dapat diandalkan dan seberapa baik kemampuannya dalam menghasilkan prediksi yang tepat. Berikut ini adalah beberapa metode evaluasi yang sering digunakan untuk menilai performa model LSTM dalam prediksi penjualan.

2.1.7.1. Root Mean Squared Error (RMSE)

Root Mean Squared Error adalah metrik yang mengukur akar kuadrat dari rata-rata kuadrat perbedaan antara nilai prediksi dan nilai aktual. RMSE lebih sensitif terhadap kesalahan besar karena kuadrat perbedaan antara nilai aktual dan prediksi diberi bobot lebih besar. Secara matematis, RMSE dapat dinyatakan dalam Persamaan pada rumus 2.8.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Rumus 2. 8 RMSE

Keterangan:

y_i = Nilai aktual (observasi ke-i)

$\hat{y}_i$ = Nilai prediksi (prediksi ke-i)

n = Jumlah data (observasi)

2.1.7.2. Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error adalah metrik yang mengukur rata-rata kesalahan dalam persentase antara nilai prediksi dan nilai aktual. MAPE berguna untuk menilai akurasi model dalam bentuk relatif atau persentase, yang memudahkan perbandingan antar dataset dengan skala yang berbeda, seperti yang dinyatakan pada persamaan rumus 2.9.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad \textbf{Rumus 2. 9 MAPE}$$

Keterangan:

y_i = Nilai aktual

$\hat{y}_i$ = Nilai prediksi

n = Jumlah data

2.1.8. Google Colab

Google Colab adalah platform berbasis cloud yang dirancang untuk mendukung kegiatan pengembangan dan eksperimen dalam bidang pembelajaran mesin serta kecerdasan buatan. Google colab digunakan untuk membangun, melatih, dan menguji model LSTM untuk prediksi penjualan. Dalam kaitannya dengan pemodelan jaringan saraf, khususnya yang bersifat sekuensial, Colab memberikan dukungan optimal terhadap berbagai pustaka, termasuk *TensorFlow*.

Sebagai pustaka open-source yang luas digunakan dalam pengembangan model deep learning, *TensorFlow* dapat dijalankan secara efisien di Google Colab, termasuk untuk membangun dan melatih model *Long Short-Term Memory*. Dengan tersedianya akses ke GPU maupun TPU secara gratis, pengguna dapat melatih

model LSTM dengan waktu komputasi yang lebih singkat dan efisien langsung dari platform Colab.

Paduan antara Google Colab, *TensorFlow*, dan arsitektur LSTM memberikan kemudahan bagi peneliti maupun pengembang dalam mengerjakan proyek berbasis data sekuensial. Ketersediaan sumber daya komputasi yang fleksibel dan lingkungan pemrograman yang kolaboratif memungkinkan proses eksperimen dan iterasi model berlangsung lebih cepat. Dengan demikian, platform ini menjadi alat yang strategis dalam kegiatan riset maupun pengembangan solusi berbasis kecerdasan buatan secara praktis dan ekonomis.

2.2. Penelitian Terdahulu

Berikut studi yang mengacu pada hasil-hasil penelitian terdahulu yang relevan:

1. Penelitian berjudul "Prediksi Belanja Pemerintah Indonesia Menggunakan *Long Short-Term Memory* (LSTM)" menggunakan algoritma LSTM untuk memprediksi pengeluaran pemerintah berdasarkan data time-series harian daHJJGJJri Kementerian Keuangan RI. Dengan pendekatan deep learning, peneliti membangun model LSTM multi-layer (1–4 hidden layers) dan melakukan seleksi atribut menggunakan *correlation coefficient* untuk meningkatkan akurasi. Hasil terbaik diperoleh dari model dengan tiga hidden layers dan atribut terpilih, dengan MSE 0.2325, RMSE 0.4820, MAE 0.3292, dan MAPE 0.4214 jauh lebih akurat dibandingkan ARIMA. Proses dilakukan menggunakan *Python* (Keras) melalui tahap pembersihan data, normalisasi, encoding, dan evaluasi. (Sabar Sautomo & Hilman Ferdinandus Pardede, 2021).

2. Penelitian berjudul “Peramalan Penjualan Sediaan Farmasi Menggunakan Long Short-Term Memory: Studi Kasus pada Apotik Suganda” bertujuan mengatasi ketidaktepatan pengadaan stok obat yang masih dilakukan secara manual, sering menyebabkan kekurangan atau kelebihan persediaan. Berbagai metode digunakan, termasuk Least Square, Exponential Smoothing, ARIMA, dan LSTM. LSTM dipilih karena mampu menangkap pola non-linier dalam data *time series*, dengan konfigurasi 4 langkah *Input* dan 400 epoch. Data diambil dari Apotek Suganda (Juni–September 2019) dan diproses dengan *Python*. Hasilnya, LSTM paling akurat dengan MAPE rata-rata 4,6151%, jauh lebih baik dibanding ARIMA (14,62%) (Anshory et al., 2020).
3. Penelitian berjudul ”Prediksi Harga Cabai menggunakan Metode Long-Short Term Memory” memanfaatkan LSTM untuk memprediksi harga cabai rawit di Kota Malang, dengan data harian dari Januari 2021 hingga Juli 2022. Model LSTM yang dikembangkan berhasil mencapai Mean Square Error (MSE) terkecil sebesar 0,0155, menunjukkan kemampuan LSTM dalam menangani data *time series* dengan fluktuasi tinggi (David et al., 2023).
4. Penelitian berjudul “prediksi harga beras menggunakan metode recurrent *neural network* dan long short-term memory” secara ilmiah menjawab permasalahan fluktuasi harga beras di Indonesia dengan menerapkan metode RNN dan LSTM berdasarkan teori, metode, dan penelitian terdahulu. Menggunakan data BPS tahun 2013–2023 dan pendekatan KDD,

penelitian ini menunjukkan bahwa model RNN menghasilkan RMSE terkecil 0.73 (beras medium) dan LSTM menghasilkan MAE terkecil 2.31 (beras luar kualitas), membuktikan akurasi prediksi yang baik untuk mendukung pengambilan keputusan (Santoso et al., 2024).

5. Penelitian “Peramalan Penjualan Laptop Menggunakan Metode Long Short Term Memory (LSTM) ”mengatasi masalah fluktuasi permintaan laptop yang menyebabkan kesulitan restock serta risiko overstock dan stockout, dengan membangun model prediksi berbasis *Long Short-Term Memory* (LSTM) menggunakan data deret waktu dan didukung hasil penelitian sebelumnya. Evaluasi model menunjukkan performa yang akurat, terutama pada kategori mid-end dengan RMSE sebesar 22,39 dan R^2 sebesar 0,9890, sementara kategori low-end dan high-end masing-masing memiliki RMSE sebesar 91,22 dan 96,82. Hasil ini membuktikan bahwa LSTM efektif dalam memprediksi penjualan mingguan dan mendukung pengambilan keputusan dalam manajemen stok (Yulianto & Latifah, 2024).
6. Penelitian “Implementasi Model Long Short Term Memory (LSTM) dalam Prediksi Harga Saham” membahas penerapan model *Long Short-Term Memory* (LSTM) untuk prediksi harga saham, dengan fokus pada harga saham Bank Central Asia (BBCA). Masalah yang dihadapi dalam penelitian ini adalah tantangan untuk memprediksi fluktuasi harga saham, yang dapat membantu investor dalam pengambilan keputusan. Metodologi yang digunakan mencakup beberapa langkah, yaitu pra-pemrosesan data (seperti normalisasi Min-Max dan penanganan data yang hilang), pemodelan LSTM

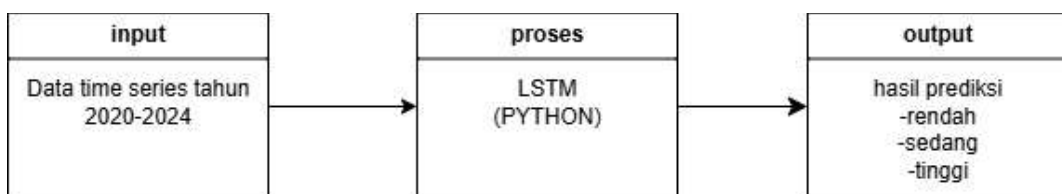
dengan variasi timesteps (30, 60, dan 90 hari), serta evaluasi kinerja model melalui metrik seperti RMSE, MAPE, dan R-Squared. Algoritma yang digunakan adalah LSTM, yang merupakan bentuk *Recurrent Neural network* (RNN) yang efektif dalam menangani data deret waktu dengan memanfaatkan sel memori untuk menyimpan informasi historis. Model ini memberikan hasil terbaik dengan timesteps 60 hari, dengan nilai RMSE sebesar 114,17, MAPE 0,96%, dan nilai R-Squared 0,98, yang mengindikasikan prediksi yang sangat akurat (ABBAS, 2022).

7. Penelitian "Rice Price Prediction with *Long Short-Term Memory* (LSTM) *Neural network*" bertujuan memprediksi harga beras menggunakan model *Long Short-Term Memory* (LSTM). Masalah yang dihadapi adalah fluktuasi harga beras yang dipengaruhi oleh variabilitas cuaca, hasil panen, dan luas lahan. Teknik yang digunakan adalah LSTM, jaringan saraf tiruan yang cocok untuk menganalisis data deret waktu. Model dilatih menggunakan data harian dari 2015 hingga 2023 yang mencakup kondisi cuaca, hasil panen, dan sejarah harga beras. Proses pra-pemrosesan data meliputi penanganan data yang hilang, normalisasi dengan skala min-max, dan interpolasi data. LSTM bekerja dengan menggunakan gerbang (*forget*, *Input*, dan *Output*) untuk mengelola ketergantungan jangka panjang dan pendek pada data. Hasilnya, model LSTM menghasilkan RMSE sebesar 0,054, menunjukkan akurasi prediksi yang tinggi. Model ini dapat memprediksi fluktuasi harga beras dengan baik, memberikan wawasan

berharga untuk pengambilan keputusan oleh petani, pedagang, dan pembuat kebijakan (Darussalam & Arief, 2018)

2.3. Kerangka Pemikiran

Kerangka pemikiran merupakan rancangan berpikir yang sistematis untuk tahapan penyelesaian masalah dalam suatu penelitian. Berikut ini kerangka pemikiran yang dibuat penulis:



Gambar 2. 2 Kerangka pemikiran
(Sumber: Data penelitian, 2025)

Flowchart pada gambar 2.3 ini menjelaskan alur tentang prediksi penjualan menggunakan metode LSTM :

1. *Input* (Data *Time series* Tahun 2020–2024)

Langkah awal dalam sistem ini adalah mengumpulkan data penjualan yang tersusun secara kronologis dari tahun 2020 hingga 2024. Data *time series* tersebut menjadi dasar penting dalam proses prediksi karena mampu merepresentasikan kecenderungan, pola musiman, serta variasi penjualan dari waktu ke waktu. Ketepatan dan kelengkapan data pada tahap ini sangat mempengaruhi kualitas prediksi yang akan dihasilkan oleh sistem.

2. Proses (LSTM dengan *Python*)

Selanjutnya, data yang telah dikumpulkan dianalisis menggunakan algoritma *Long Short-Term Memory* yang dikembangkan dengan bantuan bahasa pemrograman

Python. Dalam penerapannya menggunakan *Python*, proses dimulai dengan tahap *Preprocessing* data, meliputi normalisasi, penghapusan data kosong, serta pembentukan jendela waktu (*time window*). Selanjutnya, model LSTM dibangun menggunakan pustaka seperti *TensorFlow* atau Keras, kemudian model dilatih menggunakan data tahun 2020–2024 untuk mempelajari pola yang ada. Setelah itu, model divalidasi guna mengukur tingkat akurasi prediksi, sebelum akhirnya digunakan untuk memperkirakan penjualan di waktu mendatang. Keunggulan utama dari LSTM adalah kemampuannya dalam menghindari permasalahan *vanishing gradient* serta kecakapannya dalam mendeteksi pola jangka pendek maupun panjang dalam data yang berurutan.

3. *Output* (hasil prediksi)

Output pada hasil prediksi dari model LSTM diklasifikasikan ke dalam tiga kategori, yaitu rendah, sedang, dan tinggi. Kategori ini menunjukkan tingkat hasil prediksi berdasarkan pola data historis, di mana "rendah" mencerminkan nilai yang kecil, "sedang" mewakili nilai menengah, dan "tinggi" menunjukkan nilai besar atau peningkatan signifikan, yang dapat digunakan untuk mendukung pengambilan keputusan strategis.