



# Peramalan Deret Waktu Penjualan Bulanan Menggunakan Pendekatan Deep Learning Metode Long Short-Term Memory

Intan Permatasari\*, Sulistyio Puspitodjati

Program Magister Teknologi dan Rekayasa, Magister Manajemen Sistem Informasi, Universitas Gunadarma, Depok, Indonesia

Email: <sup>1,\*</sup>intansari92p@gmail.com, <sup>2</sup>sulistyio@staff.gunadarma.ac.id

Email Penulis Korespondensi: intansari92p@gmail.com

**Abstrak**—Perusahaan sering melakukan prediksi penjualan untuk menentukan jumlah stok barang di gudang guna mengendalikan persediaan dan memastikan ketersediaan barang sebelum proses penjualan. Penelitian ini bertujuan untuk membandingkan kinerja metode *Autoregressive Integrated Moving Average (ARIMA)* dan *Long Short-Term Memory (LSTM)* dalam memprediksi jumlah stok barang bulanan. Metode penelitian yang digunakan adalah penelitian kuantitatif dengan pendekatan eksperimen, di mana model LSTM sebagai bagian dari metode *deep learning* diterapkan untuk mempelajari pola data deret waktu secara terstruktur. Data yang digunakan merupakan data historis permintaan produk bulanan selama periode 2012–2018 dengan pendekatan *one time step prediction*. Evaluasi kinerja model dilakukan menggunakan *Root Mean Square Error (RMSE)* sebagai indikator tingkat akurasi prediksi. Hasil penelitian menunjukkan bahwa model LSTM menghasilkan nilai RMSE rata-rata sebesar 5,39 dari sepuluh kali pengujian, yang lebih kecil dibandingkan metode ARIMA pada penelitian sebelumnya dengan dataset yang sama. Hasil ini menunjukkan bahwa metode LSTM memiliki kinerja prediksi yang lebih baik dan berpotensi digunakan sebagai dasar pengambilan keputusan dalam perencanaan stok barang bulanan.

**Kata Kunci:** Pembelajaran Mendalam; Peramalan; Deret Waktu; *Long Short-Term Memory*; *Root Mean Square Error*

**Abstract**—Companies often perform sales forecasting to determine inventory levels in order to control stock and ensure product availability prior to sales activities. This study aims to compare the performance of the Autoregressive Integrated Moving Average (ARIMA) method and the Long Short-Term Memory (LSTM) method in forecasting monthly sales quantities. This research employs a quantitative experimental approach, where the LSTM model as part of a deep learning framework is used to learn structured time series patterns. The dataset consists of historical monthly demand data from 2012 to 2018 using a one time step prediction approach. Model performance is evaluated using Root Mean Square Error (RMSE) as the accuracy metric. The results show that the LSTM model achieves an average RMSE of 5.39 across ten experimental runs, which is lower than that of the ARIMA method reported in previous studies using the same dataset. These findings indicate that the LSTM method provides better forecasting performance and can serve as a reliable basis for monthly inventory planning decisions.

**Keywords:** Deep Learning; Forecasting; Time Series; Long Short-Term Memory; Root Mean Square Error

## 1. PENDAHULUAN

Dalam kegiatan sehari-hari perusahaan dagang, stok merupakan asset utama yang mempengaruhi keuntungan di dalam perusahaan. Manajemen stok yang efisien dapat mengurangi resiko bisnis dari kerugian sehingga penting untuk bisa memprediksi dan memastikan ketersediaan stok barang sebagai kunci keberhasilan suatu perusahaan dagang. Dengan adanya manajemen stok yang baik, dapat mengurangi biaya produksi yang tidak digunakan sehingga dapat meningkatkan penjualan, loyalitas pelanggan dan keuntungan (Kilimci et al., 2019). Namun, permintaan produk sering kali bersifat fluktuatif sehingga sulit diprediksi secara akurat apabila dilakukan secara manual.

Perkembangan teknologi informasi, khususnya kecerdasan buatan (*Artificial Intelligence*) telah banyak dimanfaatkan untuk membantu pengambilan keputusan di berbagai bidang. Salah satu cabang dari AI seperti *Machine Learning* dapat belajar secara otomatis dengan cara mengumpulkan data, menemukan pola dengan cara melatih algoritma, lalu membuat klasifikasi dan membuat model prediksi lalu menggunakannya untuk memproses data sehingga dapat digunakan untuk membantu *manager* di perusahaan untuk mengambil keputusan dan menyelesaikan suatu permasalahan yang rumit. Dalam konteks peramalan, *Algoritma forecasting* mempelajari supaya menghasilkan data prediksi dengan memprediksi kejadian yang belum terjadi seperti memperkirakan kejadian di masa yang akan datang dengan cara mengambil data dimasa lalu dipelajari sehingga membentuk suatu pola informasi (Rizal & Soraya, 2018).

*Neural Network* yang merupakan bagian dari *Machine Learning* khususnya pada *Deep Learning* sering digunakan diberbagai bidang seperti peramalan cuaca, *finance*, kesehatan, biologi dan data *stock market*. Algoritma dalam *forecasting* dapat dibagi 2 yaitu *linear* (AR, MA, ARIMA, ARMA) dan *non-linear models* (ARCH, GARCH, *Neural Network*). Ada beberapa tipe arsitektur yang digunakan *Deep Learning* seperti *Recurrent Neural Network (RNN)*, *Multilayer Perceptron (MLP)*, *Convolutional Neural Network (CNN)*, *Artificial Neural Networks (ANNs)*, *Gated Recurrent Unit (GRU)*, *Convolutional Neural Network-LSTM (CNN-LSTM)* dan dalam penelitian ini yang digunakan *Long Short-Term Memory (LSTM)* yang merupakan jenis *Neural Network* yang mengatasi kendala *vanishing gradient* dari model sebelumnya *Recurrent Neural Network (RNN)*.

Data permintaan yang tidak kontinyu (intermittent demand) menimbulkan tantangan dalam *forecasting* karena banyak periode dengan permintaan nol atau rendah; penggunaan metode agregasi tunggal dan ganda bersama LSTM dapat membantu meningkatkan akurasi (Azizi et al., 2022). Pendekatan LSTM mampu menangkap ketergantungan jangka panjang dalam data *time-series* sehingga dapat digunakan untuk memprediksi fluktuasi permintaan produk dan membantu mengoptimalkan stok agar tidak terjadi kelebihan atau kekurangan stok (Susilo et al., 2025).



Pemodelan untuk memprediksi stok barang sudah pernah diuji dan dianalisis dengan menggunakan model *ARIMA* dengan data yang sama (Rushadi & Puspitodjati, 2020). Dalam model *hybrid* ini, *ARIMA* menangkap tren jangka panjang di data inventori sedangkan *LSTM* menyesuaikan fluktuasi harian penjualan, sehingga stok dapat diatur lebih (Wang & Wang, 2025). Metode klasik seperti *ARIMA* dan Prophet dapat bekerja baik untuk pola musiman atau tren yang relatif stabil, namun ketika data mempunyai variasi yang kompleks atau pola *non-linear*, *LSTM* seringkali memberikan kinerja yang lebih baik (Suryawan et al., 2024).

Tujuan penelitian ini adalah untuk menerapkan metode Long Short-Term Memory (*LSTM*) dalam peramalan penjualan bulanan serta mengevaluasi kinerjanya dibandingkan metode *Autoregressive Integrated Moving Average (ARIMA)* pada data penjualan yang sama. Penelitian ini diarahkan untuk menghasilkan model peramalan yang mampu mendukung pengambilan keputusan dalam perencanaan dan pengendalian stok.

Meskipun berbagai penelitian telah membahas penerapan metode *ARIMA* dan *Long Short-Term Memory (LSTM)* dalam peramalan penjualan dan manajemen stok, masih terdapat celah penelitian yang perlu dikaji lebih lanjut. Beberapa penelitian menggunakan *dataset* yang berbeda sehingga perbandingan langsung antara *ARIMA* dan *LSTM* pada *dataset* yang sama masih terbatas. Selain itu, sebagian studi lebih berfokus pada pengembangan model *hybrid* tanpa mengevaluasi secara khusus kinerja *LSTM* secara mandiri pada data penjualan bulanan menggunakan pendekatan *one time step prediction*. Oleh karena itu, diperlukan penelitian yang secara eksplisit membandingkan kinerja metode *LSTM* dan *ARIMA* pada *dataset* yang sama dengan skenario eksperimen yang terkontrol dan metrik evaluasi yang konsisten.

Kontribusi utama dari penelitian ini adalah menyajikan evaluasi empiris kinerja model Long Short-Term Memory (*LSTM*) dalam peramalan penjualan bulanan menggunakan pendekatan *one time step prediction*. Penelitian ini membandingkan secara langsung kinerja *LSTM* dengan metode *Autoregressive Integrated Moving Average (ARIMA)* pada *dataset* yang sama berdasarkan nilai *Root Mean Square Error (RMSE)*. Hasil penelitian ini diharapkan dapat memberikan kontribusi ilmiah dalam pemilihan metode peramalan yang lebih akurat serta menjadi dasar pengambilan keputusan dalam perencanaan dan pengendalian stok.

## 2. METODOLOGI PENELITIAN

### 2.1 Kerangka Dasar Penelitian

Metode penelitian yang digunakan pada penelitian ini yaitu metode penelitian kuantitatif dengan pendekatan eksperimen. Penelitian eksperimen merupakan salah satu bentuk metode penelitian yang menggunakan pendekatan kuantitatif (Asrin, 2022). Metode penelitian eksperimen termasuk dalam pendekatan kuantitatif yang digunakan untuk menganalisis pengaruh perlakuan atau variabel bebas terhadap variabel terikat. Pendekatan ini dilakukan melalui proses percobaan yang terkontrol guna mengamati hubungan sebab-akibat antara variabel yang diteliti (Sugiyono, 2019).

Pada penelitian ini, model *Long Short-Term Memory (LSTM)* digunakan untuk melakukan prediksi data deret waktu permintaan produk, sedangkan *Root Mean Square Error (RMSE)* digunakan sebagai metrik evaluasi kinerja model. *RMSE* berfungsi untuk mengukur tingkat kesalahan prediksi model dengan membandingkan nilai hasil prediksi terhadap nilai aktual, di mana nilai *RMSE* yang lebih kecil menunjukkan tingkat akurasi model yang lebih baik (Díaz González, 2019).

*Dataset* yang digunakan berupa data deret waktu permintaan produk yang diperoleh dari suatu institusi nasional. Data tersebut mencakup periode tujuh tahun terakhir, yaitu dari tahun 2012 hingga 2018, dan dikonversi ke dalam format *comma separated values (CSV)* sebelum diproses menggunakan bahasa pemrograman *Python* dengan dukungan pustaka *Pandas* (Mwiti, 2018). Data yang digunakan tidak tersedia secara publik, namun akses data dapat diberikan dengan persetujuan institusional. *Dataset* terdiri dari 83 baris data historis dan 1 baris data tambahan untuk kebutuhan peramalan penjualan pada bulan Desember 2018. Setiap data memuat atribut waktu (bulan dan tahun) serta nilai penjualan yang selanjutnya dipisahkan ke dalam data pelatihan (*training*) dan data pengujian (*testing*) sesuai dengan proporsi yang telah ditentukan. Data deret waktu tersebut kemudian digunakan sebagai dasar pembelajaran model untuk menghasilkan prediksi penjualan satu periode ke depan. Data penjualan periode 2012–2018 yang digunakan dalam penelitian ini ditunjukkan pada Tabel 1.

**Tabel 1.** *Dataset* Penjualan Suatu Produk di Perusahaan Tahun 2012 – 2018

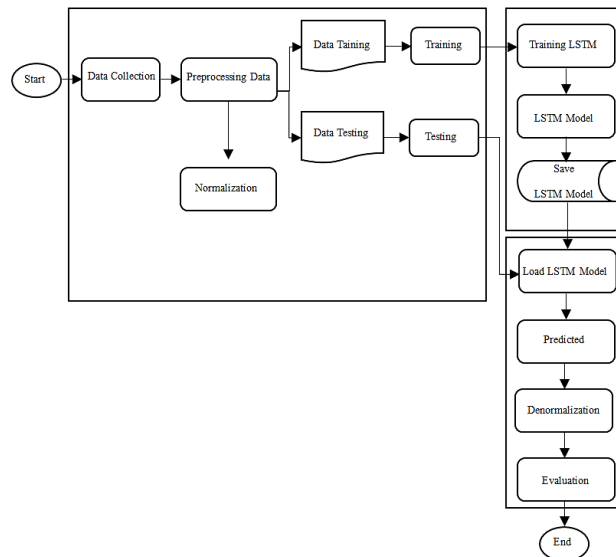
Month	Sales	Month	Sales	Month	Sales	Month	Sales	Month	Sales	Month	Sales	Month	Sales
12-01	19	13-01	20	14-01	17	15-01	9	16-01	0	17-01	21	18-01	19
12-02	19	13-02	23	14-02	19	15-02	28	16-02	8	17-02	17	18-02	15
12-03	0	13-03	22	14-03	15	15-03	21	16-03	24	17-03	19	18-03	19
12-04	18	13-04	18	14-04	25	15-04	24	16-04	22	17-04	21	18-04	15
12-05	0	13-05	20	14-05	13	15-05	20	16-05	22	17-05	15	18-05	21
12-06	20	13-06	24	14-06	31	15-06	21	16-06	12	17-06	11	18-06	5
12-07	15	13-07	19	14-07	11	15-07	9	16-07	14	17-07	20	18-07	15
12-08	17	13-08	15	14-08	11	15-08	15	16-08	21	17-08	16	18-08	17
12-09	18	13-09	20	14-09	16	15-09	14	16-09	15	17-09	10	18-09	11
12-10	14	13-10	21	14-10	6	15-10	32	16-10	21	17-10	18	18-10	14

Month	Sales	Month	Sales	Month	Sales	Month	Sales	Month	Sales	Month	Sales	Month	Sales
12-11	17	13-11	27	14-11	10	15-11	20	16-11	18	17-11	14	18-11	12
12-12	17	13-12	6	14-12	17	15-12	20	16-12	23	17-12	15	18-12	19

Tabel 1 menyajikan keseluruhan data penjualan bulanan yang disusun secara kronologis dalam satu rangkaian deret waktu. Penyajian data secara terintegrasi ini memungkinkan pengamatan pola fluktuasi penjualan dari waktu ke waktu, termasuk kecenderungan tren dan variasi musiman yang menjadi karakteristik utama data deret waktu. Data pada tabel ini selanjutnya digunakan sebagai dasar dalam proses pemodelan dan evaluasi kinerja metode peramalan yang diterapkan.

### 2.2 Tahapan Penelitian

Adapun tahapan penelitian berturut – turut dalam penelitian ini yaitu penyiapan data, pembuatan model, dan terakhir evaluasi model sebagai pengujian. Tahapan awal persiapan data yang terdiri dari pengumpulan data, lalu *preprocessing*, kemudian representasi data, dan juga pemilahan data (*dataset splitting*). Proses tahapan diilustrasikan pada Gambar 1 berikut. *Preprocessing* data terdiri dari normalisasi *dataset* dan data *cleansing*. Setelah *preprocessing* kemudian *dataset* dibagi menjadi *dataset training* dan *dataset testing*. Tahap selanjutnya untuk proses *forecasting* dibagi dua yaitu proses *training* dan selanjutnya proses *prediction*. Proses *training* dilakukan secara terpisah dengan proses prediksi supaya waktu proses dapat lebih singkat. Selanjutnya eksperimen dilakukan dengan *training LSTM* dan menghasilkan model LSTM yang selanjutnya memuat file model dan digunakan pada proses prediksi. Tahap akhir eksperimen kemudian didenormalisasi dari data hasil *testing* untuk kemudian mendapatkan nilai prediksi dan evaluasi menghasilkan nilai dari kinerja model. Tahapan penelitian ini dapat dilihat pada Gambar 1 berikut.



**Gambar 1.** Alur Eksperimen pada *Dataset*

Berdasarkan Gambar 1, alur penelitian disajikan dalam bentuk diagram yang menunjukkan keterkaitan antar proses secara berurutan dari tahap awal hingga akhir. Visualisasi ini menegaskan hubungan logis antara proses penyiapan data, pemodelan, dan evaluasi, serta memperlihatkan bagaimana keluaran dari setiap tahap menjadi masukan bagi tahap berikutnya dalam keseluruhan proses *forecasting*.

Pada penelitian ini juga diamati perubahan nilai RMSE dari hasil pengujian model untuk menilai kestabilan kinerja model LSTM. Proses eksperimen dilakukan dengan menjalankan model pada skenario pengujian yang telah ditentukan, sehingga diperoleh nilai kesalahan prediksi yang dapat digunakan sebagai dasar evaluasi kinerja model peramalan.

### 2.3 Preprocessing Data

Tahapan *preprocessing* merupakan langkah krusial dalam menyiapkan data agar sesuai dengan format masukan model LSTM. Teknik yang umum digunakan antara lain normalisasi dan pembentukan data sekuensial dengan *sliding window* (Wijayanto et al., 2024). Proses *preprocessing* data bertujuan untuk memastikan data yang digunakan dalam pelatihan model telah bersih, relevan dan berada dalam skala yang sesuai, karena kualitas data sangat mempengaruhi kinerja model pada *Machine Learning*.

Sebelum proses pemodelan dilakukan, data penjualan diproses melalui tahap *preprocessing* yang meliputi normalisasi data dan pembentukan fitur lag. Normalisasi bertujuan untuk menyamakan skala data sehingga proses pelatihan model dapat berjalan secara optimal, sedangkan pembentukan fitur lag digunakan untuk merepresentasikan pola temporal pada data deret waktu agar dapat dipelajari oleh model.

Normalisasi data dilakukan menggunakan teknik min-max scaling, yaitu dengan mengubah nilai data aktual ke dalam rentang tertentu, umumnya antara 0 hingga 1. Proses normalisasi ini bertujuan untuk mempercepat proses pembelajaran model dan mencegah dominasi nilai tertentu dalam proses pelatihan. Persamaan normalisasi data ditunjukkan pada Persamaan (1).

$$x^r = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

Pada persamaan normalisasi,  $x'$  merepresentasikan nilai hasil normalisasi, sedangkan  $x$  merupakan nilai data aktual yang akan dinormalisasi. Adapun  $x_{min}$  menunjukkan nilai minimum dan  $x_{max}$  merepresentasikan nilai maksimum dari data aktual yang digunakan sebagai acuan dalam proses normalisasi.

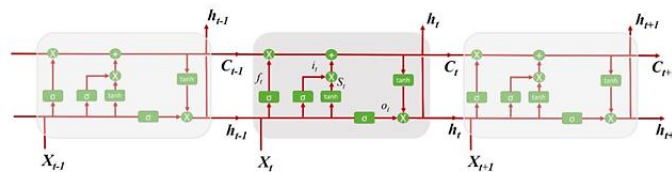
Setelah proses prediksi dilakukan, hasil keluaran model masih berada dalam bentuk data ter-normalisasi. Oleh karena itu, dilakukan proses denormalisasi untuk mengembalikan nilai hasil prediksi ke skala data aktual agar mudah diinterpretasikan. Proses denormalisasi dilakukan menggunakan Persamaan (2).

$$d = d'(max - min) + min \tag{2}$$

Pada persamaan denormalisasi, adapun  $d$  merepresentasikan nilai hasil denormalisasi,  $d'$  merupakan nilai data normalisasi,  $max$  menunjukkan nilai maksimum dari data aktual dan  $min$  merepresentasikan nilai minimum dari data aktual yang digunakan sebagai acuan untuk mengembalikan data hasil prediksi ke skala aslinya.

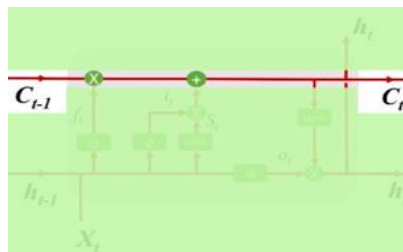
### 2.4 Long Short-Term Memory (LSTM)

Struktur dasar sel Long Short-Term Memory (LSTM) yang digunakan dalam penelitian ini ditunjukkan pada Gambar 2 berikut.



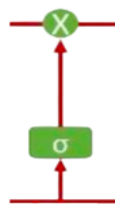
**Gambar 2. Cell LSTM**

Pada arsitektur LSTM (Gambar 2) tersebut, *hidden state* dan *cell state* berperan penting dalam menyimpan informasi jangka panjang. *Hidden state* membawa informasi dari langkah sebelumnya, sementara *cell state* menyimpan informasi jangka panjang yang dapat digunakan untuk prediksi di masa depan (Alghifari et al., 2022). Jaringan LSTM terdiri dari beberapa blok memori yang disebut sel. Terdapat 2 jenis *states* yang ditransfer yaitu *cell state* dan *hidden state*. Blok memori difungsikan untuk mengingat dan memanipulasi sesuatu. Selanjutnya struktur *cell state* sebagai jalur utama aliran informasi pada arsitektur LSTM ditunjukkan pada Gambar 3.



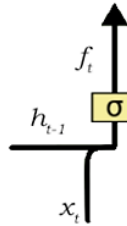
**Gambar 3. Cell State Rajagukguk et al. (2020)**

*Cell state* atau garis *horizontal* pada Gambar 3 yang melintasi  $(C_{t-1}, C_t)$  berfungsi sebagai jalur utama aliran informasi yang membawa informasi dari sel sebelumnya ke sel selanjutnya. Mekanisme gate pada arsitektur LSTM yang berperan dalam mengatur aliran informasi tersebut ditunjukkan pada Gambar 4.



**Gambar 4. Gate LSTM Rajagukguk et al. (2020)**

*Gates* pada Gambar 4 berfungsi sebagai mekanisme pengatur aliran informasi yang bekerja melalui operasi perkalian *pointwise* dan lapisan *sigmoid*. Salah satu komponen utama dari mekanisme tersebut, yaitu *forget gate*, yang ditunjukkan pada Gambar 5 berikut.

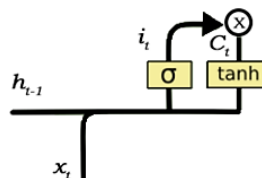


**Gambar 5.** Forget Gate Ryan (2020)

Pada Gambar 5, *cell state* LSTM memiliki kemampuan untuk menghapus maupun mempertahankan informasi melalui proses penyaringan guna mengoptimalkan kinerja jaringan. Proses ini dilakukan oleh lapisan sigmoid yang mentransmisikan informasi dalam bentuk nilai antara 0 dan 1, di mana nilai 0 menunjukkan informasi tidak diteruskan, sedangkan nilai 1 menunjukkan informasi dipertahankan. Mekanisme kerja *forget gate* tersebut dirumuskan pada persamaan (3) berikut.

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \quad (3)$$

Pertama, *forget gate* ( $f_t$ ) bekerja menggunakan lapisan *sigmoid* untuk menentukan informasi dari *cell state* sebelumnya yang perlu dipertahankan atau dihapus. Nilai  $h_{t-1}$  sebagai *hidden layer* dari sel sebelumnya dan  $x_t$  sebagai *input* baru dikombinasikan melalui operasi perkalian matriks bobot dan penambahan bias, kemudian diproses oleh fungsi *sigmoid* untuk menghasilkan nilai 0 hingga 1 sebagai pengendali aliran informasi (apakah informasi tersebut dibuang atau disimpan). Mekanisme *input gate* sebagai tahap berikutnya dalam arsitektur LSTM ditunjukkan pada Gambar 6.



**Gambar 6.** Input Gate Ryan (2020)

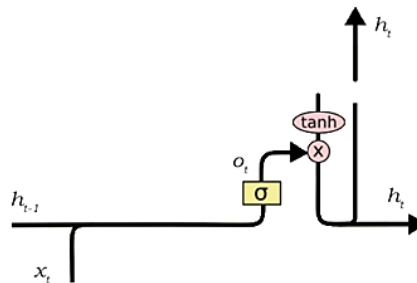
Kedua, pada Gambar 6 memutuskan informasi baru yang akan disimpan di dalam sel. Ini memiliki dua bagian. Ada 2 lapisan di sini: lapisan sigmoid berfungsi memutuskan nilai yang akan diperbarui ( $i_t$ ), yang kedua lapisan *tanh* untuk membuat nilai kandidat baru ( $\tilde{C}_t$ ).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (5)$$

$$C_t = (f_t * C_{t-1} + i_t * \tilde{C}_t) \quad (6)$$

Ketiga, melakukan filter kembali dengan menggabungkan hasil keluaran dari *tanh* (berupa nilai -1 sampai 1) dengan keluaran *sigmoid* dengan perkalian dan menghasilkan hasil informasi yang berguna ke dalam sel melalui operasi penjumlahan menghasilkan *new state* ( $C_t$ ).



**Gambar 7.** Output Gate Ryan (2020)

Tidak semua hasil informasi sepanjang sel dapat cocok digunakan pada waktu tertentu. Gambar 7 fungsi gerbang keluaran kembali dapat dipecah menjadi tiga langkah:

1. Membuat vektor kembali dengan menskalakan nilai -1 sampai 1 dari fungsi *tanh*.
2. Membuat filter kembali melalui *sigmoid* dengan menggabungkan informasi  $h_{t-1}$  dan  $x_t$ , di sini dapat mengatur nilai *output* yang diperlukan.
3. Mengalikan hasil dari langkah 2 dengan hasil di langkah 1, sebagai output ke hidden state baru.

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

$$h_t = O_t * \tanh(C_t) \quad (8)$$

Mekanisme gating (*Input, Output, forget*) memungkinkan model LSTM untuk mengontrol aliran informasi dan mengatasi masalah yang muncul dari *Vanishing Gradient* dan *Exploding Gradient* yang dialami oleh RNN sehingga tidak dapat mengingat memori jangka panjang, dan LSTM menjadi solusinya. *Long Short-Term Memory* (LSTM) merupakan jenis jaringan saraf tiruan yang dikembangkan untuk mengatasi masalah pada *Recurrent Neural Network* (RNN) dalam memproses data sekuensial jangka panjang. Model ini memiliki mekanisme gerbang (*gate*) yang berfungsi dalam mempertahankan atau melupakan informasi pada waktu tertentu, sehingga lebih stabil dalam melakukan prediksi deret waktu (*time-series*) (Khumaidi et al., 2020).

#### 2.4.1 Evaluasi

Jaringan memori jangka pendek (*Long Short-Term Memory / LSTM*) merupakan jenis jaringan saraf berulang (*Recurrent Neural Network / RNN*) yang memiliki kemampuan untuk mempelajari dan mengingat informasi dalam jangka waktu yang panjang tanpa bergantung pada jendela observasi tertentu sebagai masukan. Karakteristik ini menjadikan LSTM sesuai untuk digunakan dalam pemodelan dan peramalan data deret waktu yang bersifat dinamis.

Model *Long Short-Term Memory (LSTM)* pertama kali diperkenalkan oleh Hochreiter dan Schmidhuber (1997) sebagai pengembangan dari *Recurrent Neural Network (RNN)* untuk mengatasi permasalahan *vanishing gradient* pada pemodelan data sekuensial. Mekanisme *forget gate*, *input gate*, dan *output gate* memungkinkan LSTM mempertahankan informasi penting dalam jangka panjang secara lebih stabil (Gers et al., 2000). Konsep ini menjadi dasar pengembangan berbagai arsitektur *deep learning* modern dalam pemodelan deret waktu (Goodfellow et al., 2016).

Untuk mengevaluasi kinerja model prediksi stok atau penjualan, digunakan beberapa metrik kesalahan, yaitu *Mean Absolute Error (MAE)*, *Root Mean Square Error (RMSE)*, dan *Mean Absolute Percentage Error (MAPE)*. Penggunaan metrik tersebut bertujuan untuk mengukur tingkat perbedaan antara nilai prediksi dan nilai aktual secara kuantitatif. Dalam penelitian ini, RMSE digunakan sebagai metrik utama karena mampu memberikan penalti yang lebih besar terhadap kesalahan prediksi yang bernilai besar, sehingga sesuai untuk evaluasi peramalan penjualan. RMSE merupakan metode standar yang digunakan untuk mengukur kesalahan model dalam memprediksi data kuantitatif. Secara matematis, perhitungan RMSE ditunjukkan pada Persamaan (9).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (9)$$

Dimana  $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$  adalah nilai prediksi,  $y_1, y_2, \dots, y_n$  adalah nilai yang diamati dan  $n$  adalah nomor observasi. Perhitungan nilai RMSE dilakukan menggunakan program Python. Semakin kecil nilai RMSE, maka semakin baik kinerja model dalam melakukan prediksi penjualan. Dengan demikian, model dengan nilai RMSE terendah dipilih sebagai model prediksi yang paling optimal.

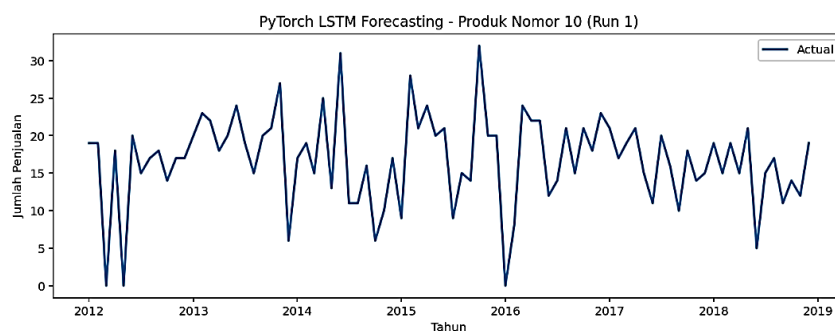
#### 2.5 Tools (Python, Keras, Pytorch)

Penelitian ini menggunakan bahasa pemrograman Python karena kemudahannya dalam pengolahan data dan pengembangan model pembelajaran mesin serta sifatnya yang open source. Beberapa pustaka yang digunakan antara lain NumPy untuk operasi matriks, Pandas untuk pengolahan data, Matplotlib untuk visualisasi, serta Scikit-learn untuk *preprocessing* data. Pengembangan dan pelatihan model *Long Short-Term Memory (LSTM)* pada penelitian ini dilakukan menggunakan Keras sebagai application programming interface (API) yang berjalan di atas backend TensorFlow.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Pembahasan

Untuk memahami karakteristik data yang digunakan dalam penelitian ini, dilakukan visualisasi data penjualan produk dalam bentuk grafik garis. Visualisasi tersebut disajikan pada Gambar 8.



Gambar 8. Line Plot Dataset Penjualan Produk Tahun 2012-2018



Gambar 8 memperlihatkan plot data *time series* penjualan produk selama periode 2012–2018 yang digunakan sebagai dasar pengembangan model *Long Short-Term Memory (LSTM)*. Secara umum, data menunjukkan pola fluktuatif dengan variasi yang cukup tajam pada beberapa periode tertentu. Tren jangka panjang tidak menunjukkan kecenderungan peningkatan atau penurunan yang konsisten, sehingga data dapat dikategorikan sebagai data non-stasioner. Fluktuasi ekstrem terlihat terutama pada awal periode pengamatan serta pada tahun 2015, di mana terjadi peningkatan penjualan lebih dari 30%. Peningkatan tersebut dapat dipengaruhi oleh beberapa faktor seperti permintaan pasar, faktor musiman, maupun pengaruh eksternal lainnya, namun faktor-faktor tersebut tidak dianalisis lebih lanjut dalam penelitian ini.

Pada tahun 2016 terjadi penurunan penjualan yang cukup signifikan dibandingkan tahun sebelumnya. Setelah periode tersebut, pola penjualan cenderung menunjukkan fluktuasi yang relatif lebih stabil, meskipun tetap terdapat variasi antarbulan. Karakteristik data ini menunjukkan bahwa pola penjualan tidak hanya dipengaruhi oleh tren jangka panjang, tetapi juga oleh dinamika jangka pendek yang bersifat volatil. Oleh karena itu, dibutuhkan model prediksi yang mampu mempelajari ketergantungan temporal jangka panjang sekaligus cukup fleksibel dalam menghadapi perubahan pola data.

Model LSTM dipilih karena memiliki mekanisme gating yang memungkinkan model untuk mengingat, melupakan, dan memperbarui informasi secara selektif. Kemampuan ini menjadikan LSTM lebih unggul dibandingkan model RNN konvensional dalam menangani permasalahan *vanishing gradient* serta dalam memodelkan data deret waktu yang memiliki dependensi jangka panjang. Dengan karakteristik data penjualan yang fluktuatif dan tidak stasioner, LSTM dinilai sesuai untuk digunakan dalam peramalan stok bulanan.

### 3.2 Skenario Eksperimen

Untuk mengevaluasi performa model LSTM secara objektif, penelitian ini menyusun skenario eksperimen yang dirancang secara sistematis. Pembagian data dilakukan secara kronologis dengan proporsi 80% data *training* dan 20% data *testing* tanpa penggunaan *random seed*. Pendekatan ini dipilih untuk menjaga urutan temporal data dan mencegah kebocoran informasi masa depan ke dalam data latih.

Parameter yang diuji dalam skenario eksperimen meliputi dua komponen utama, yaitu komposisi *dataset* (P1) dan nilai range interval normalisasi (P2). Rincian skenario eksperimen ditampilkan pada Tabel 2.

**Tabel 2.** Skenario Eksperimen

	1	2
P1	70/30	80/20
P2	[-1,1]	[1,1]

Berdasarkan Tabel 2, parameter P1 mencakup dua variasi pembagian data, yaitu 70% *training* dan 30% *testing* serta 80% *training* dan 20% *testing*. Parameter P2 merepresentasikan rentang nilai hasil normalisasi data. Hasil pengujian menunjukkan bahwa kombinasi parameter P1.1 dan P2.1 menyebabkan proses *training* gagal karena ketidakstabilan nilai input setelah normalisasi. Oleh karena itu, eksperimen selanjutnya difokuskan pada kombinasi parameter P1.2 dan P2.2 yang menghasilkan proses pelatihan yang stabil.

Setelah konfigurasi parameter ditentukan, dilakukan pengulangan eksperimen untuk menguji konsistensi hasil prediksi. Rincian jumlah pengulangan eksperimen disajikan pada Tabel 3.

**Tabel 3.** Jumlah Eksperimen

Eksperimen	Parameter
1	P1.2, P2.2
2	P1.2, P2.2
3	P1.2, P2.2
4	P1.2, P2.2
5	P1.2, P2.2
6	P1.2, P2.2
7	P1.2, P2.2
8	P1.2, P2.2
9	P1.2, P2.2
10	P1.2, P2.2

Berdasarkan Tabel 3, seluruh eksperimen menggunakan kombinasi parameter yang sama, yaitu P1.2 dan P2.2, dengan jumlah pengulangan sebanyak 10 kali. Pengulangan ini dilakukan untuk meminimalkan pengaruh inisialisasi bobot awal model LSTM yang bersifat acak, sehingga hasil evaluasi yang diperoleh lebih representatif.

### 3.3 Preprocessing dan Normalisasi Data

Sebelum data digunakan dalam proses pelatihan model LSTM, dilakukan tahap *preprocessing* berupa normalisasi data. Metode normalisasi yang digunakan adalah *MinMaxScaler* dengan rentang nilai 0 hingga 1. Fitur (*X*) dan target (*y*)



dinormalisasi secara terpisah menggunakan dua scaler yang berbeda untuk memastikan proses *inverse transform* dapat dilakukan secara tepat.

Sebelum proses normalisasi, scaler terlebih dahulu di-fit pada data *training* untuk menghindari terjadinya data leakage. Selanjutnya, data *testing* dan data *forecasting* hanya melalui proses transform menggunakan scaler yang telah ditentukan. Contoh hasil normalisasi data penjualan untuk produk nomor 10 selama periode 2012–2018 ditampilkan pada Tabel 4.

**Tabel 4.** Hasil Normalisasi Tipe Produk Nomor 10 Tahun 2012-2018

No	Tipe Produk	Normalisasi Value	Bulan	Tahun
1	10	0.59375	Jan	2012
2	10	0.59375	Feb	
3	10	0	Mar	
4	10	0.5625	Apr	
5	10	0	Mei	
6	10	0.625	Jun	
7	10	0.46875	Jul	
8	10	0.53125	Agust	
9	10	0.5625	Sept	
10	10	0.4375	Okt	
11	10	0.53125	Nov	
12	10	0.53125	Des	
13	10	0.625	Jan	2013
14	10	0.71875	Feb	
15	10	0.6875	Mar	
16	10	0.5625	Apr	
17	10	0.625	Mei	
18	10	0.75	Jun	
19	10	0.59375	Jul	
20	10	0.46875	Agust	
21	10	0.625	Sept	
22	10	0.65625	Okt	
23	10	0.84375	Nov	
24	10	0.1875	Des	
25	10	0.53125	Jan	2014
26	10	0.59375	Feb	
27	10	0.46875	Mar	
28	10	0.78125	Apr	
29	10	0.40625	Mei	
30	10	0.96875	Jun	
31	10	0.34375	Jul	
32	10	0.34375	Agust	
33	10	0.5	Sept	
34	10	0.1875	Okt	
35	10	0.3125	Nov	
36	10	0.53125	Des	
37	10	0.28125	Jan	2015
38	10	0.875	Feb	
39	10	0.65625	Mar	
40	10	0.75	Apr	
41	10	0.625	Mei	
42	10	0.65625	Jun	
43	10	0.28125	Jul	
44	10	0.46875	Agust	
45	10	0.4375	Sept	
46	10	1	Okt	
47	10	0.625	Nov	
48	10	0.625	Des	

No	Tipe Produk	Normalisasi Value	Bulan	Tahun
49	10	0	Jan	2016
50	10	0.25	Feb	
51	10	0.75	Mar	
52	10	0.6875	Apr	
53	10	0.6875	Mei	
54	10	0.375	Jun	
55	10	0.4375	Jul	
56	10	0.65625	Agust	
57	10	0.46875	Sept	
58	10	0.65625	Okt	
59	10	0.5625	Nov	
60	10	0.71875	Des	
61	10	0.65625	Jan	2017
62	10	0.53125	Feb	
63	10	0.59375	Mar	
64	10	0.65625	Apr	
65	10	0.46875	Mei	
66	10	0.34375	Jun	
67	10	0.625	Jul	
68	10	0.5	Agust	
69	10	0.3125	Sept	
70	10	0.5625	Okt	
71	10	0.4375	Nov	
72	10	0.46875	Des	
73	10	0.59375	Jan	2018
74	10	0.46875	Feb	
75	10	0.59375	Mar	
76	10	0.46875	Apr	
77	10	0.65625	Mei	
78	10	0.15625	Jun	
79	10	0.46875	Jul	
80	10	0.53125	Agust	
81	10	0.34375	Sept	
82	10	0.4375	Okt	
83	10	0.375	Nov	
84	10	0.59375	Des	

Berdasarkan Tabel 4, terlihat bahwa nilai penjualan asli telah berhasil ditransformasikan ke dalam skala yang seragam tanpa menghilangkan variasi pola data. Proses normalisasi ini penting untuk memastikan kestabilan proses pembelajaran model LSTM, mengingat model *deep learning* sensitif terhadap perbedaan skala nilai input.

### 3.4 Arsitektur Model dan Proses *Training* LSTM

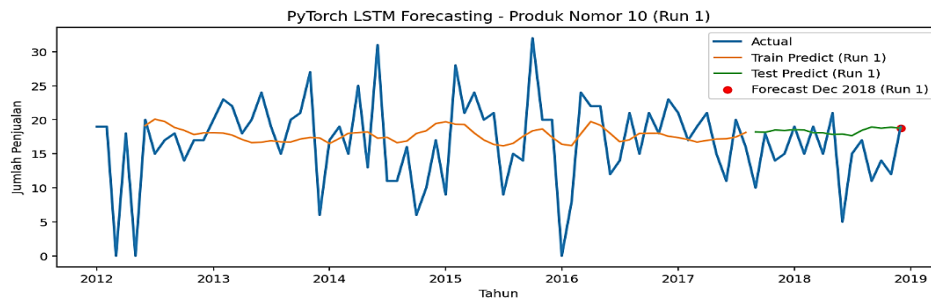
Arsitektur model LSTM yang digunakan dalam penelitian ini dirancang dengan konfigurasi sederhana. Model terdiri dari satu lapisan LSTM dengan jumlah hidden unit sebanyak 50. Optimasi model dilakukan menggunakan *optimizer* ADAM. Penggunaan *learning rate* diterapkan sebagai upaya untuk mencegah terjadinya overfitting selama proses pelatihan. Pada tahap pengaturan hyperparameter, dilakukan eksperimen dengan varian *learning rate* sebesar 0.01, ukuran batch sebesar 1, dan jumlah *epoch* sebanyak 250. Data dibentuk dalam format supervised learning dengan pendekatan *sliding window*, di mana lima bulan data historis digunakan sebagai input untuk memprediksi penjualan pada bulan ke-6. Sebelum diproses oleh model LSTM, data terlebih dahulu diskalakan dalam bentuk dua dimensi (2D), kemudian dikembalikan ke dalam bentuk tiga dimensi (3D) agar sesuai dengan kebutuhan input LSTM. Contoh data stok yang digunakan berasal dari penjualan produk tipe 10 dengan urutan data [19, 19, 0, 18, 0, 20, 15, ...]. Dengan *window size* sebesar 5, pembentukan data supervised dilakukan dengan pasangan input dan target seperti  $X = [19, 19, 0, 18, 0]$  dengan  $y = 20$  serta  $X = [19, 0, 18, 0, 20]$  dengan  $y = 15$ .

Pengaturan arsitektur dan hyperparameter dibuat tanpa percobaan pengaturan lanjutan karena pola *dataset* deret waktu penjualan sudah cukup menunjukkan hasil yang akurat dengan konfigurasi sederhana. Dalam penelitian oleh Zahara et al. (2019), penggunaan berbagai algoritma optimasi diuji, dan ADAM menunjukkan performa yang stabil dan akurat. Hasil serupa ditunjukkan oleh Enrico et al. (2022) yang menemukan bahwa model dengan *optimizer* ADAM

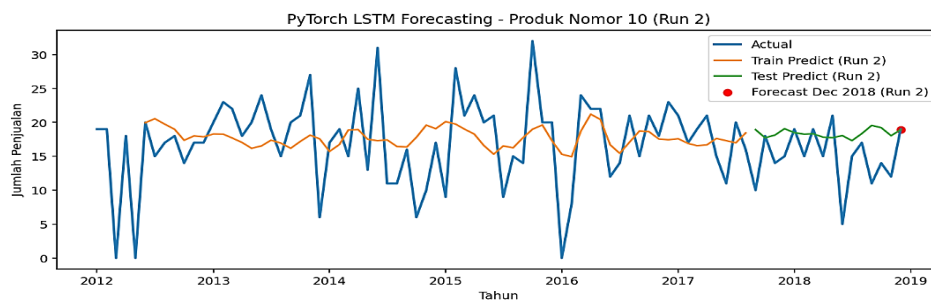
menghasilkan akurasi tertinggi (98,45%) dibandingkan optimizer lainnya. Penelitian oleh Khumaidi et al. (2024) menunjukkan bahwa model LSTM yang dilatih menggunakan ADAM dan fungsi loss MSE menghasilkan RMSE yang rendah baik pada data latih maupun data uji. Proses pelatihan dalam penelitian ini menggunakan *optimizer ADAM* dengan *learning rate* 0.001 karena stabilitas dan kemampuannya menyesuaikan parameter secara adaptif (Budiarso et al., 2024). Pada penelitian Optimasi LSTM Mengurangi Overfitting, penulis menggunakan ADAM dalam optimasi model LSTM untuk klasifikasi teks dengan tujuan mengurangi overfitting dan meningkatkan generalisasi pada data uji (Putrama Alkhairi et al., 2024).

### 3.5 Prediksi & Evaluasi Model LSTM

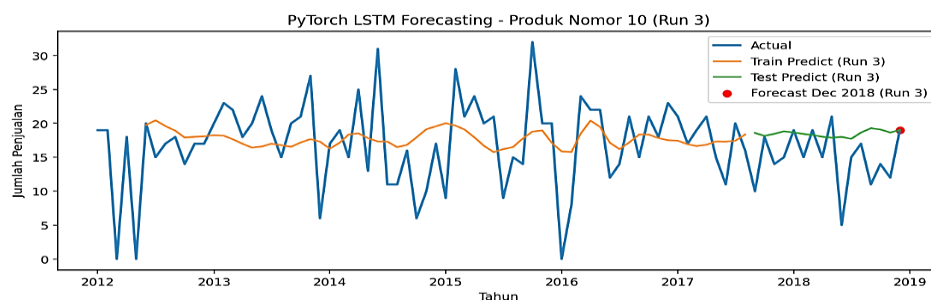
Untuk melihat hasil prediksi model LSTM pada setiap pengulangan eksperimen, ditampilkan grafik perbandingan antara data aktual dan data hasil prediksi. Visualisasi hasil eksperimen disajikan pada Gambar 9 hingga Gambar 17.



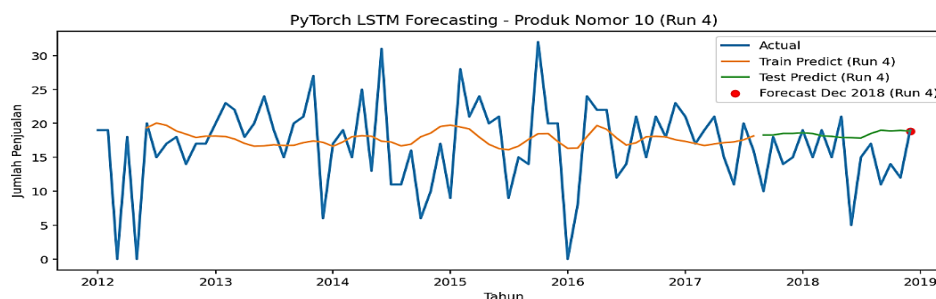
**Gambar 9.** Plot Eksperimen Ke-1 Produk Nomor 10 Tahun 2012-2018



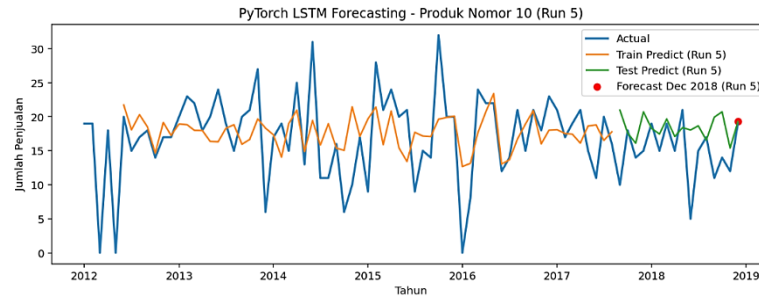
**Gambar 10.** Plot Eksperimen Ke-2 Produk Nomor 10 Tahun 2012-2018



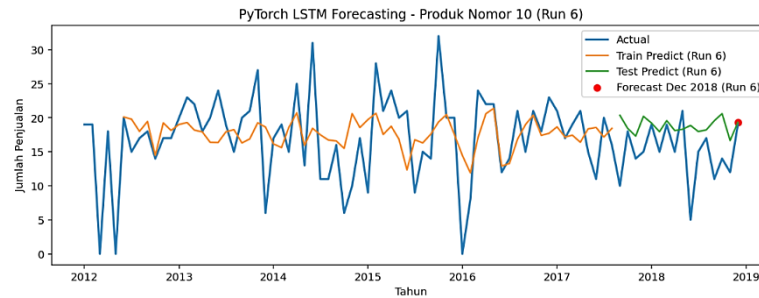
**Gambar 11.** Plot Eksperimen Ke-3 Produk Nomor 10 Tahun 2012-2018



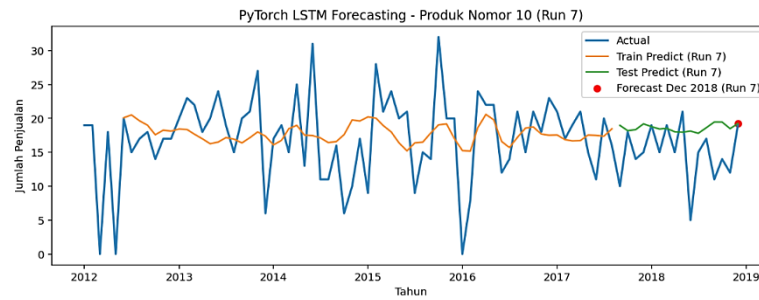
**Gambar 12.** Plot Eksperimen Ke-4 Produk Nomor 10 Tahun 2012-2018



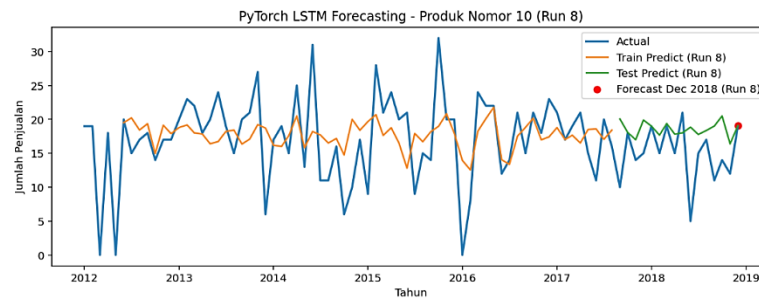
**Gambar 13.** Plot Eksperimen Ke-5 Produk Nomor 10 Tahun 2012-2018



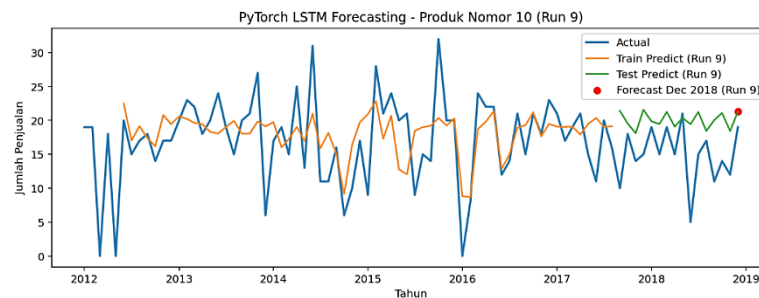
**Gambar 14.** Plot Eksperimen Ke-6 Produk Nomor 10 Tahun 2012-2018



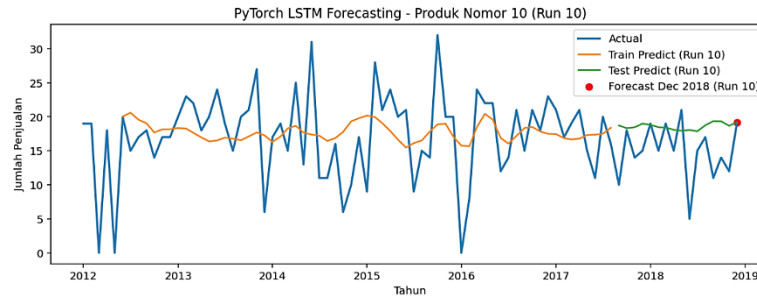
**Gambar 15.** Plot Eksperimen Ke-7 Produk Nomor 10 Tahun 2012-2018



**Gambar 16.** Plot Eksperimen Ke-8 Produk Nomor 10 Tahun 2012-2018



**Gambar 17.** Plot Hasil Eksperimen Ke-9 Produk Kesepuluh Tahun 2012-2018



**Gambar 18.** Plot Eksperimen Ke-10 Produk Nomor 10 Tahun 2012-2018

Gambar 18 Menunjukkan hasil *training* dan prediksi serta evaluasi model LSTM dalam bentuk grafik. Sumbu X secara horizontal menunjukkan keterangan waktu penjualan per tahun dimulai dari penjualan tahun 2012 sampai akhir 2018. Sedangkan sumbu Y secara vertical menunjukkan jumlah penjualan suatu produk di dalam perusahaan. Secara detail garis biru yaitu *actual* menunjukkan data hasil penjualan selama 7 tahun dari tahun 2012 sampai dengan akhir 2018. Garis biru (*actual*) menunjukkan fluktuasi secara tajam dari pertahun terdapat kenaikan dan penurunan penjualan secara ekstrem, dan terlihat data *actual* tersebut cukup *volatile*. Garis Oranye menunjukkan prediksi data *training* (*Train Predict*) yang merupakan hasil dari prediksi LSTM selama proses *training set*. Hasilnya ditampilkan cukup mendekati pola tren data aktual, tapi tidak secara presisi mengikuti fluktuasi yang ekstrem sebelumnya, dan cukup wajar karena fungsi dari pelatihan model utamanya dipakai untuk mengenali secara umum pola *dataset*. Garis Hijau menunjukkan hasil prediksi (*test predict*) yaitu hasil prediksi LSTM setelah pengujian (*test set*). *Test predict* terlihat mengikuti pola dari data *actual* tapi tidak 100% akurat.

Berdasarkan Gambar 9 sampai Gambar 17, terlihat bahwa hasil prediksi model LSTM secara konsisten mengikuti pola umum data aktual, meskipun tidak sepenuhnya menangkap fluktuasi ekstrem. Pola ini menunjukkan bahwa model mampu mempelajari tren utama data penjualan, namun cenderung melakukan *smoothing* terhadap perubahan yang sangat tajam.

Hasil akhir *training*, *testing*, dan *forecasting* model LSTM ditampilkan pada Gambar 18. Gambar 18 menunjukkan perbandingan antara data aktual, hasil prediksi pada data latih, hasil prediksi pada data uji, serta hasil *forecast* untuk bulan Desember 2018. Prediksi akhir menunjukkan nilai penjualan berada pada kisaran 18–19 unit, yang mendekati nilai aktual sebesar 19 unit. Evaluasi kuantitatif performa model dilakukan menggunakan metrik *Root Mean Square Error (RMSE)*. Hasil RMSE dari sepuluh kali pengulangan eksperimen ditampilkan pada Tabel 5 berikut.

**Tabel 5.** Hasil Rata-Rata RMSE Sepuluh Kali Pengulangan

No Produk	Run	Train_RMSE	Test_RMSE	Forecast Dec 2018	Actual Dec 2018
10	1	5.51	5.3	18.52	19
10	2	5.12	6.38	21.41	19
10	3	5.69	5.31	18.84	19
10	4	5.62	5.43	19.13	19
10	5	5.68	5.27	18.85	19
10	6	4.17	4.85	19.86	19
10	7	5.78	5.17	18.61	19
10	8	5.49	5.25	18.57	19
10	9	5.37	5.5	19.30	19
10	10	5.71	5.32	19.00	19
10	<i>Average</i>	5.414	5.39	19.21	19

Berdasarkan Tabel 5, nilai RMSE data uji relatif stabil dengan rata-rata sebesar 5,39. Nilai ini menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami *overfitting* yang signifikan.

Jika dibandingkan dengan penelitian sebelumnya yang menggunakan metode ARIMA dengan data yang sama, nilai RMSE yang dihasilkan sebesar 56483, jauh lebih besar dibandingkan RMSE model LSTM dalam penelitian ini. Perbedaan yang sangat signifikan ini menunjukkan keunggulan LSTM dalam memodelkan data deret waktu yang bersifat nonlinier dan fluktuatif. Selain itu, Susilo et al. (2025) melaporkan tingkat kesalahan sebesar 3,60% menggunakan metrik MAPE. Meskipun metrik yang digunakan berbeda, hasil tersebut menunjukkan bahwa performa LSTM dalam penelitian ini berada dalam rentang yang kompetitif.

Hasil evaluasi menunjukkan bahwa nilai RMSE rata-rata model LSTM sebesar 5,39 jauh lebih kecil dibandingkan nilai RMSE metode ARIMA yang dilaporkan pada penelitian sebelumnya dengan *dataset* yang sama. Perbedaan ini menegaskan bahwa LSTM lebih efektif dalam memodelkan pola nonlinier dan fluktuatif pada data penjualan bulanan. Temuan ini sejalan dengan penelitian terdahulu yang menunjukkan keunggulan LSTM dibandingkan metode statistik klasik dalam peramalan deret waktu dengan dinamika kompleks.



#### 4. KESIMPULAN

Berdasarkan hasil pengujian model terhadap data permintaan pada penelitian ini, dapat disimpulkan bahwa model *Long Short-Term Memory (LSTM)* mampu memberikan kinerja peramalan yang jauh lebih baik dibandingkan pendekatan konvensional yang digunakan pada penelitian sebelumnya. Model LSTM menghasilkan nilai *Root Mean Square Error (RMSE)* sebesar 5,39, yang secara signifikan lebih kecil dibandingkan nilai RMSE sebesar 56.483 yang diperoleh menggunakan metode ARIMA pada *dataset* yang sama. Perbedaan nilai kesalahan ini menunjukkan bahwa LSTM lebih efektif dalam menangkap pola nonlinier dan ketergantungan jangka panjang pada data deret waktu permintaan. Namun demikian, penelitian ini masih menggunakan pendekatan *one time step prediction*, sehingga prediksi yang dihasilkan terbatas pada satu langkah waktu ke depan. Kondisi ini menyebabkan kemampuan model dalam menangkap dinamika perubahan permintaan jangka menengah dan panjang belum dieksplorasi secara optimal. Penelitian ini juga memiliki beberapa keterbatasan yang perlu diperhatikan. Pertama, data yang digunakan hanya berasal dari satu lokasi, sehingga variasi spasial dan kemungkinan keterkaitan antar lokasi belum dipertimbangkan secara menyeluruh. Selain itu, keterbatasan data historis dapat memengaruhi kemampuan model dalam mempelajari pola permintaan yang kompleks. Dari sisi metode, penggunaan LSTM univariat belum mempertimbangkan variabel eksternal lain yang berpotensi memengaruhi permintaan. Oleh karena itu, penelitian selanjutnya disarankan untuk mengembangkan pendekatan *multistep prediction* guna menghasilkan ramalan yang lebih akurat untuk beberapa periode ke depan. Selain itu, penerapan *multivariate LSTM* dengan melibatkan data dari beberapa lokasi atau variabel pendukung lainnya diharapkan mampu meningkatkan performa model serta memberikan gambaran yang lebih komprehensif terhadap pola permintaan. Arah penelitian lanjutan ini diharapkan dapat memperkuat kontribusi model pembelajaran mendalam dalam peramalan deret waktu berbasis data permintaan.

#### REFERENCES

- Alghifari, D. R., Edi, M., & Firmansyah, L. (2022). *Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia Bidirectional LSTM Implementation for Sentiment Analysis Against Grab Indonesia Services*. 12, 89–99.
- Asrin, A. (2022). Metode Penelitian Eksperimen. *Jurnal Maqasiduna : Ilmu Humaniora, Pendidikan & Ilmu Sosial*, 1.
- Azizi, F., Wibowo, W. C., Komputer, F. I., & Indonesia, U. (2022). *JURNAL RESTI*. 5(158), 855–859.
- Díaz González, F. (2019). *Federated Learning for Time Series Forecasting Using LSTM Networks: Exploiting Similarities Through Clustering*. 2019:308, 70.
- Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10), 2451–2471.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Khumaidi, A., Raafi, R., & Solihin, I. P. (2020). *Pengujian Algoritma Long Short Term Memory untuk Prediksi Kualitas Udara dan Suhu Kota Bandung*. 15(1), 13–18.
- Kilimci, Z. H., Akyuz, A. O., Uysal, M., Akyokus, S., Uysal, M. O., Bulbul, B. A., & Ekemis, M. A. (2019). *An Improved Demand Forecasting Model Using Deep Learning Approach and Proposed Decision Integration Strategy for Supply Chain*. 2019. <https://doi.org/10.1155/2019/9067367>
- Mwiti, D. (2018). *Using a Keras Long Short-Term Memory (LSTM) Model to Predict Stock Prices*. <https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstm-model-predict-stock-prices.html>
- Rahman Wahid, A., & Witanti, A. (2025). Implementasi Algoritma Lstm Untuk Prediksi Kebutuhan Bahan Baku Restoran Di Bale Raos Kraton Yogyakarta Implementation of Lstm Algorithm for Restaurant Raw Material Demand Prediction At Bale Raos Kraton Yogyakarta. *Journal of Information Technology and Computer Science (INTECOMS)*, 8(4).
- Rajaguguk, R. A., Ramadhan, R. A. A., & Lee, H. (2020). *A Review on Deep Learning Models for Forecasting Time Series Data of Solar Irradiance and*.
- Rizal, A. A., & Soraya, S. (2018). *Multi Time Steps Prediction Dengan Recurrent Neural*. 18(1), 115–124.
- Rushadi, B., & Puspitodjati, S. (2020). Forecasting Monthly Nautical Chart Time Series Of Java Area Using Arima Method (Case study: Primkopal Pushidrosal Sales Units). *Jurnal Ilmiah Bidang Teknologi, ANGKASA2*, 12(1).
- Ryan, T. J. J. (2020). *LSTMs Explained: A Complete, Technically Accurate, Conceptual Guide with Keras*. <https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate-conceptual-guide-with-keras-2a650327e8f2>
- Sugiyono, S. (2019). *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Alfabeta Bandung.
- Suryawan, I. G. T., Putra, I. K. N., Meliana, P. M., & Sudipa, I. G. I. (2024). Performance Comparison of ARIMA, LSTM, and Prophet Methods in Sales Forecasting. *Sinkron*, 8(4), 2410–2421. <https://doi.org/10.33395/sinkron.v8i4.14057>
- Susilo, D., Ahmad Chusyairi, & Saputra, M. I. (2025). Prediksi Penjualan Untuk Optimasi Stock Produk Menggunakan Algoritma Long Short Term Memory. *Sistematis : Jurnal Ilmiah Sistem Informasi*, 1(2), 119–134. <https://doi.org/10.69533/56gyat30>



- Wang, C., & Wang, J. (2025). Research on E-Commerce Inventory Sales Forecasting Model Based on ARIMA and LSTM Algorithm. *Mathematics*, 13(11), 1–10. <https://doi.org/10.3390/math13111838>
- Wijayanto, A., Sugiharto, A., Santoso, R., Diponegoro, U., & Korespondensi, P. (2024). *Identifikasi Dini Curah Hujan Berpotensi Banjir Menggunakan Algoritma Long Short-Term Memory ( LSTM ) Dan Isolation Forest Early Identification Of Rainfall With Flood Potential Using Long Short-Term Memory ( LSTM ) And Isolation Forest Algorithms Case STU*. 11(3). <https://doi.org/10.25126/jtiik.938718>