



Received 00th January 20xx
Accepted 00th February 20xx
Published 00th March 20xx

Open Access

DOI: 10.35472/x0xx0000

Analisis Prediksi Penjualan pada Data Walmart dengan Long Short-Term Memory (LSTM)

Sarah Natalia Geraldine^a, Angelica Noviana^b,
Afifah Syaharani^c, Muhammad Rendy^d, Ericson
Chandra Sihombing^e, Rohmi Dyah Astuti^f

^a sarah.121450022@student.itera.ac.id

^b angelica.121450064@student.itera.ac.id

^c afifah.121450097@student.itera.ac.id

^d muhammad.121450045@student.itera.ac.id

^e ericson.121450026@student.itera.ac.id

Abstract: Sales prediction analysis is an important aspect in supporting business planning, especially for large retail companies like Walmart. This research uses Long Short-Term Memory (LSTM), one of the deep neural network architectures, to predict sales based on time series data. This model was chosen for its ability to capture complex patterns and long-term dependencies in the data. The dataset used includes sales information from Walmart from January 2010 to September 2012. Stages included data collection, pre-processing, model training, and performance evaluation using metrics. Model evaluation was conducted using RMSE, MAE, and MAPE, which resulted in values of 2152542.70, 1808226.93, and 3.80%, respectively. The results show that the LSTM model is able to provide sales predictions with good accuracy, so it can be a reliable tool to support strategic decision-making in the retail industry.

Keywords: LSTM, sales prediction, walmart, deep learning, time series

Abstrak: Analisis prediksi penjualan merupakan aspek penting dalam mendukung perencanaan bisnis, terutama bagi perusahaan ritel besar seperti Walmart. Penelitian ini menggunakan Long Short-Term Memory (LSTM), salah satu arsitektur jaringan saraf dalam, untuk memprediksi penjualan berdasarkan data deret waktu. Model ini dipilih karena kemampuannya dalam menangkap pola kompleks dan dependensi jangka panjang pada data. Dataset yang digunakan mencakup informasi penjualan dari Walmart dari Januari 2010 hingga September 2012. Tahapan meliputi pengumpulan data, pra-pemrosesan, pelatihan model, hingga evaluasi kinerja menggunakan metrik. Evaluasi model dilakukan dengan menggunakan RMSE, MAE, dan MAPE, yang masing-masing menghasilkan nilai 2152542.70, 1808226.93, dan 3.80%. Hasil menunjukkan bahwa model LSTM mampu memberikan prediksi penjualan dengan akurasi yang baik, sehingga dapat menjadi alat yang andal untuk mendukung pengambilan keputusan strategis di industri ritel.

Kata Kunci: LSTM, prediksi penjualan, walmart, deep learning, deret waktu.





Pendahuluan

Di era modern, kemajuan teknologi khususnya di bidang analisis data, telah berkembang pesat dan memberikan peluang besar bagi perusahaan industri untuk menghadapi tantangan yang semakin kompleks [1]. Salah satu tantangan utama yang dihadapi adalah tingkat persaingan yang semakin ketat, yang menuntut perusahaan untuk melakukan perencanaan parameter produksi secara efektif [1]. Dalam hal ini, menganalisa dan memprediksi total penjualan mingguan menjadi strategi penting untuk memahami tren pasar, mengoptimalkan operasional, dan menghadapi persaingan dengan lebih baik [2].

Peramalan (*forecasting*) merupakan kombinasi seni dan ilmu yang bertujuan untuk memprediksi kejadian di masa depan [3]. Pendekatan ini dapat melibatkan pembuatan model matematis atau prediksi intuitif yang bersifat subjektif, namun tetap didasarkan pada pertimbangan yang matang. Peramalan penjualan akan menjadi langkah suatu perusahaan untuk bersaing dan meningkatkan laba perusahaan dengan melibatkan perbedaan waktu antara sekarang dengan masa lalu [3]. Salah satu metode yang dapat digunakan untuk peramalan ini adalah *Long Short - Term Memory* (LSTM) [4].

Long Short-Term Memory (LSTM) merupakan salah satu jenis algoritma dalam *Deep Learning* yang merupakan pengembangan dari *Recurrent Neural Networks* (RNN) [4]. LSTM digunakan untuk mengekstrak informasi dari data dan dapat digunakan untuk melakukan klasifikasi baik pada citra teks maupun gambar. Selain itu, LSTM memiliki kemampuan untuk mengingat informasi yang telah disimpan dalam waktu yang lama [4]. Untuk menilai akurasi model, digunakan perhitungan dengan parameter *Root Mean Squared Error* (RMSE), di mana semakin kecil nilai RMSE yang diperoleh, semakin tinggi tingkat akurasi model tersebut [5]. Selain itu, digunakan juga *Mean Absolute Percentage Error* (MAPE), yang dihitung dari rata - rata nilai absolut dan kemudian dikalikan dengan 100% [5].

Metode *Long Short Term Memory* (LSTM) pada penelitian ini digunakan untuk memprediksi penjualan mingguan di Walmart berdasarkan data penjualan di bulan sebelumnya dengan menggunakan *Window Time Series* selama 30 hari. Proses pemodelan dilakukan dengan membagi dataset menjadi data latih dan data uji dengan rasio 80:20, serta menerapkan teknik normalisasi *MinMax Scaler* untuk mengubah rentang data penjualan ke dalam skala 0-1.

Arsitektur model yang dibangun terdiri dari dua *layer* LSTM dengan jumlah unit 128 dan 64, disertai *dropout* sebesar 0,2 untuk mencegah *overfitting*, serta satu *layer dense* dengan aktivasi ReLU. Tujuan dilakukannya penelitian ini adalah untuk mengetahui performa metode LSTM dalam memprediksi penjualan mingguan dengan mengukur metrik evaluasi seperti *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE), dan *Mean Absolute Percentage Error* (MAPE).

Metode

Penelitian ini menggunakan dataset laporan penjualan mingguan selama periode 2010 hingga 2012. Dataset ini bersumber dari <https://www.kaggle.com/> dan mencakup berbagai faktor yang mempengaruhi penjualan, termasuk hari libur, suhu, harga bahan bakar, indeks harga konsumen (CPI), dan tingkat pengangguran.

Informasi mengenai data yang digunakan dalam penelitian ini mencakup sebagai berikut :

- Kolom **"Store"** mengidentifikasi terkait indeks toko, sementara kolom **"Date"** mencatat tanggal akhir minggu terkait, kolom **"Weekly Sales"** mencatat total penjualan mingguan untuk setiap toko.
- Kolom **"Holiday Flag"** menunjukkan minggu-minggu yang mencakup peristiwa hari libur penting.
- Kolom **"Temperature"** memberikan data suhu rata-rata mingguan.
- **"Fuel Price"** mengacu pada harga bahan bakar mingguan.
- Kolom **"CPI"** (Consumer Price Index) menunjukan harga barang dan jasa konsumen.
- Kolom **"Unemployment Rate"** mencatat tingkat pengangguran.

Penelitian ini berfokus pada analisis **"Weekly Sales"** dan **"Date"** sebagai kolom utama.

- **Weekly Sales:** Variabel target yang merepresentasikan total penjualan mingguan dari masing-masing toko. Data ini menjadi fokus utama untuk memprediksi penjualan di masa mendatang berdasarkan pola historis dan pengaruh faktor eksternal lainnya.
- **Date:** Tanggal akhir minggu yang merepresentasikan waktu pengamatan. Kolom ini berperan penting dalam membangun model deret waktu (time series) untuk analisis menggunakan



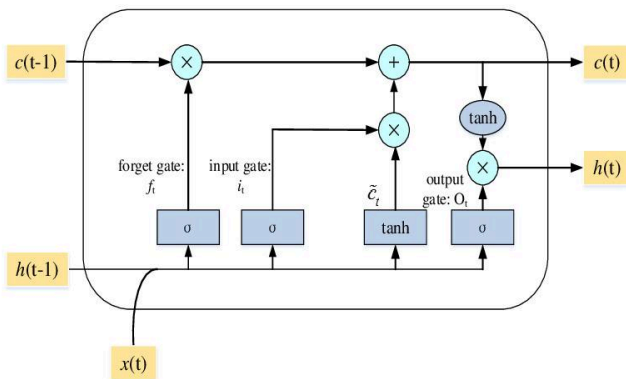
LSTM (Long Short-Term Memory), karena kemampuan LSTM untuk menangkap pola temporal dalam data historis.

Pemanfaatan kedua kolom ini memungkinkan analisis yang lebih mendalam tentang pola musiman, trend jangka panjang, dan dampak hari libur terhadap penjualan, sehingga menghasilkan prediksi yang lebih akurat.

A. LSTM (Long Short-Term Memory)

Dalam penelitian ini kami mengimplementasikan model Long Short-Term Memory (LSTM) yang dirancang untuk memproses data sekuensial dengan memperhitungkan ketergantungan temporal.

LSTM adalah jenis Recurrent Neural Network (RNN) yang memiliki mekanisme kontrol dalam bentuk tiga gerbang utama (*forget gate*, *input gate*, dan *output gate*). Mekanisme ini memungkinkan LSTM untuk mempertahankan informasi yang relevan dalam jangka waktu yang panjang serta melupakan informasi yang tidak penting.



Gambar 2. Arsitektur LSTM (Sumber: [mdpi.com/2071-1050/13/24/13609](https://doi.org/10.21961/jst.v1i1.13609))

Sel LSTM menerima dua masukan: *input* saat ini $x(t)$ dan keadaan tersembunyi sebelumnya $h(t-1)$. Kedua masukan ini diberikan ke tiga gerbang utama: *forget gate* f , *input gate* i , dan *output gate* o [6].

Forget gate f menentukan informasi dari keadaan sel sebelumnya $c(t-1)$ yang harus dipertahankan. Proses ini menggunakan fungsi aktivasi sigmoid pada gabungan $x(t)$ dan $h(t-1)$, menghasilkan vektor dengan nilai antara 0 hingga 1 untuk setiap elemen dalam keadaan sel.

Nilai mendekati 1 berarti elemen tersebut harus disimpan, sedangkan nilai mendekati 0 berarti elemen tersebut harus dilupakan [6].

Rumus matematisnya:

$$f = \sigma(W_f \cdot [h(t-1), x(t)] + b_f) \quad (1)$$

di mana, W_f adalah bobot yang dipelajari untuk forget gate, dan b_f adalah bias [6].

Input gate i menentukan informasi baru dari $x(t)$ dan $h(t-1)$ yang akan ditambahkan ke keadaan sel. Fungsi sigmoid digunakan untuk menghasilkan nilai antara 0 dan 1, menunjukkan seberapa besar setiap elemen harus diperbarui [7].

Rumus matematis:

$$i = \sigma(W_i \cdot [h(t-1), x(t)] + b_i) \quad (2)$$

di mana, W_i adalah bobot input gate, dan b_i adalah bias [7].

Selain itu, input gate menggunakan fungsi aktivitas tanh untuk menghasilkan kandidat nilai baru yang dapat ditambahkan ke keadaan sel:

$$\bar{c} = \tanh(W_c \cdot [h(t-1), x(t)] + b_c) \quad (3)$$

Keadaan sel baru $c(t)$ dihitung dengan mengalikan $c(t-1)$ dengan forget gate f , lalu menambahkan hasil perkalian input gate i dan kandidat keadaan sel \bar{c} [8].

$$c(t) = f \cdot c(t-1) + i \cdot \bar{c} \quad (4)$$

Output gate o menentukan informasi dari $c(t)$ yang digunakan untuk menghitung keadaan tersembunyi baru $h(t)$ [8]. Fungsi sigmoid digunakan pada $x(t)$ dan $h(t-1)$, menghasilkan nilai antara 0 dan 1 untuk mengatur informasi yang diungkapkan [9].

Rumus matematis:

$$o = \sigma(W_o \cdot [h(t-1), x(t)] + b_o) \quad (5)$$

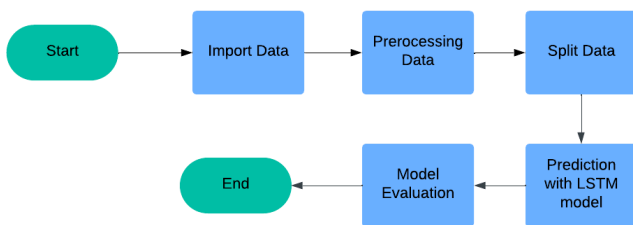
keadaan tersembunyi baru $h(t)$ dihitung dengan mengalikan o dengan fungsi aktivasi tanh dari $c(t)$:

$$h(t) = o \cdot \tanh(c(t)) \quad (6)$$

Arsitektur ini memungkinkan LSTM untuk secara selektif mengingat dan melupakan informasi, sehingga mampu memodelkan ketergantungan jangka panjang dalam data sekuensial [10].

B. Research Design

Desain penelitian ini berfokus pada pemanfaatan data penjualan historis untuk menghasilkan model prediksi yang dapat digunakan oleh Walmart untuk merencanakan strategi bisnis masa depan. Dataset ini mencakup catatan penjualan mingguan dari 45 toko Walmart. Penelitian ini dirancang untuk memanfaatkan kemampuan Long Short-Term Memory (LSTM), sebuah algoritma *deep learning* yang secara khusus dirancang untuk menangkap pola dalam data temporal dan menangani hubungan jangka panjang antara variabel-variabel tersebut. Desain penelitian dalam metode ini dijelaskan secara terstruktur sebagai berikut:



Gambar 3. Research Design Flowchart

Penelitian ini menggunakan berbagai library untuk mendukung analisis dan prediksi penjualan. Dalam tahap pra-pemrosesan, MinMax Scaler dari Sklearn digunakan untuk menormalisasi data agar berada dalam rentang 0-1, yang penting untuk meningkatkan performa model LSTM. Evaluasi model dilakukan menggunakan metrik *Mean Squared Error* (MSE) dan *Mean Absolute Error* (MAE) untuk mengukur akurasi prediksi. Arsitektur model menggunakan Sequential dari library TensorFlow, di mana lapisan LSTM berfungsi menangkap hubungan temporal data, *dense* sebagai lapisan keluaran untuk prediksi, dan *dropout* digunakan untuk mencegah *overfitting*.

1. Data Preprocessing

a. Data Preprocessing

Langkah pertama dalam proses *data preprocessing* adalah memastikan bahwa data mentah yang dimuat dapat digunakan secara efektif oleh model pembelajaran mesin. Dataset yang digunakan dalam penelitian ini mencakup data penjualan mingguan dari 45 toko Walmart. Berikut adalah beberapa langkah utama yang dilakukan dalam tahap ini:

- Membaca Dataset: Dataset dimuat menggunakan pustaka pandas. File ini berisi data mentah dengan beberapa kolom seperti tanggal, penjualan mingguan (Weekly Sales), serta variabel lain seperti

bahan bakar (Fuel Price) dan indeks harga konsumen (CPI).

- Konversi Format Tanggal: Kolom Date, yang awalnya dalam format teks, dikonversi ke format *datetime*. Konversi ini penting agar data tanggal dapat digunakan langsung untuk analisis berbasis waktu.
- Menjadikan Kolom Tanggal Sebagai Indeks: Setelah format tanggal dikonversi, kolom Date ditetapkan sebagai indeks dari DataFrame. Langkah ini memastikan bahwa data terorganisasi dalam urutan kronologis, yang merupakan prasyarat untuk bekerja dengan data deret waktu (*time series*).
- Seleksi Variabel Target: Kolom Weekly Sales dipilih sebagai target utama dalam proses peramalan. Jika dataset berisi data dari beberapa toko, penjualan mingguan dapat diakumulasi dengan menjumlahkan nilai pada kolom weekly sales berdasarkan tanggal tertentu.

Proses ini bertujuan untuk memastikan bahwa data bersih, terstruktur, dan siap untuk analisis lebih lanjut.

b. Normalization

Proses normalisasi dilakukan untuk memastikan bahwa semua nilai numerik berada pada skala yang seragam. Dalam alur penelitian ini, normalisasi dilakukan menggunakan fungsi *MinMaxScaler* dari pustaka *sklearnScaler* ini mengubah nilai pada kolom Weekly Sales menjadi rentang antara 0 hingga 1 sesuai dengan formula berikut :

$$X^I = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (7)$$

Proses ini sangat penting untuk mencegah dominasi fitur berskala besar terhadap fitur lainnya selama pelatihan model, terutama untuk algoritma berbasis *gradient descent* seperti LSTM. Normalisasi memastikan bahwa setiap variabel memiliki pengaruh yang seimbang dalam proses optimasi model.

2. Sliding Window Process

Proses *sliding window* adalah teknik yang digunakan untuk mengubah data deret waktu menjadi kumpulan pasangan input-output yang dapat digunakan dalam pelatihan model *machine learning*, terutama untuk model yang memanfaatkan data berurutan seperti LSTM. Berikut langkah-langkahnya:

a. Menentukan Periode Window_size

Parameter *window_size* diatur ke nilai tertentu. Nilai ini menentukan jumlah langkah waktu sebelumnya yang dimasukkan sebagai fitur input. Pada penelitian ini:

- Parameter *window_size* menentukan seberapa banyak data historis yang digunakan untuk memprediksi nilai berikutnya.
- Nilai *window_size* dipilih berdasarkan eksperimen atau keragaman data yang dianalisis.
- Meningkatkan nilai *window_size* memungkinkan model mempelajari lebih banyak pola historis, tetapi juga meningkatkan kompleksitas komputasi.

b. Membuat Urutan Input

Fungsi `create_dataset` digunakan untuk mengonversi data yang sudah dinormalisasi menjadi urutan input-output. Proses ini melibatkan:

- Iterasi pada data: Dataset diproses dengan membuka jendela data secara berurutan selama periode *window_size*
- Pembentukan pasangan input-target:
(Data dari periode *window_size* sebelumnya digunakan sebagai input *x*.)
(Data dari langkah waktu berikutnya digunakan sebagai target output *y*.)

Proses ini memungkinkan model untuk memanfaatkan data historis untuk membuat prediksi langkah berikutnya.

c. Pembersihan dan Penyesuaian Data

Data input yang dihasilkan dari langkah sebelumnya perlu dikonversi ke format yang sesuai untuk model LSTM, yaitu dalam bentuk [samples, time steps, features]. Pada penelitian ini:

- **Samples:** Jumlah pasangan input/output yang dihasilkan dari data
- **Time Steps:** Panjang *lookback*, yaitu jumlah langkah waktu historis
- **Features:** Jumlah fitur dalam data. dalam kasus ini, hanya ada satu fitur, yaitu Weekly Sales.

Proses sliding window ini sangat esensial dalam pengolahan data deret waktu karena mengkonversi data mentah menjadi format yang dapat dimanfaatkan secara optimal oleh model LSTM.

3. Splitting Data

Langkah pertama dalam pembangunan model LSTM adalah membagi dataset menjadi dua bagian utama: *training set*

dan *testing set*. Pembagian data ini dilakukan dengan rasio 80:20, yang berarti 80% dari data digunakan untuk melatih model, sementara 20% sisanya digunakan untuk evaluasi model. Tujuan dari pembagian ini adalah untuk memastikan bahwa model dilatih menggunakan sebagian besar data, sedangkan sisa data digunakan untuk menguji kemampuan model dalam memprediksi data yang belum pernah dilihat sebelumnya (testing data).

4. Membangun Model

Setelah data dibagi, model Long Short-Term Memory (LSTM) dibuat untuk menangkap pola temporal dari data deret waktu. Model LSTM yang digunakan dalam penelitian ini terdiri dari beberapa lapisan yang saling terhubung untuk mengekstrak fitur dan membuat prediksi yang akurat.

a. Lapisan LSTM pertama

Lapisan pertama adalah lapisan LSTM dengan 256 unit. Lapisan ini bertujuan untuk membentuk urutan data input dengan panjang yang sama dengan parameter *window_size*. parameter `return_sequences=True` memastikan bahwa lapisan ini mengembalikan sekumpulan status tersembunyi yang akan diulang pada setiap langkah urutan input. Pengaturan ini memungkinkan model untuk menangkap hubungan temporal yang lebih dalam dengan mengembalikan urutan.

b. Lapisan Dropout (0.3)

Setelah lapisan LSTM pertama, lapisan dropout diterapkan dengan tingkat 30% untuk menghindari overfitting. Dropout adalah teknik regularisasi yang secara acak menghapus beberapa neuron dari jaringan selama pelatihan, membuat model menjadi lebih tahan terhadap overfitting dan lebih kuat dalam mempelajari pola umum data.

c. Lapisan LSTM Kedua

Lapisan LSTM Kedua mengandung 128 unit dengan pengaturan `return_sequences=True`. memastikan bahwa lapisan ini mengembalikan sekumpulan status tersembunyi yang akan diulang pada setiap langkah urutan input.

d. Lapisan Dropout Kedua (0,3)

Lapisan dropout kedua diterapkan dengan tingkat 30% untuk mencegah overfitting lebih lanjut setelah lapisan LSTM kedua. Langkah regularisasi ini sangat penting untuk menjaga kemampuan model agar tetap dapat menggeneralisasi dengan baik pada data baru yang belum pernah dilihat sebelumnya.

e. Lapisan LSTM Ketiga

Lapisan LSTM Kedua mengandung 64 unit dengan pengaturan `return_sequences=True` memastikan bahwa lapisan ini mengembalikan sekumpulan status tersembunyi yang akan diulang pada setiap langkah urutan input.

f. Lapisan Dropout Ketiga (0,3)

Lapisan dropout kedua diterapkan dengan tingkat 30% untuk mencegah overfitting lebih lanjut setelah lapisan LSTM kedua. Langkah regularisasi ini sangat penting untuk menjaga kemampuan model agar tetap dapat menggeneralisasi dengan baik pada data baru yang belum pernah dilihat sebelumnya.

g. Lapisan LSTM Keempat

Lapisan LSTM Kedua mengandung 32 unit dengan pengaturan `return_sequences=False`, pengaturan ini menyebabkan lapisan kedua menghasilkan satu vektor status tersembunyi, yang menunjukkan bahwa data temporal dari urutan input telah cukup diproses oleh lapisan LSTM pertama. Vektor ini kemudian diteruskan ke lapisan berikutnya untuk pemrosesan lebih lanjut.

h. Lapisan Dense (Fully Connected Layer)

Pada Penelitian ini model dirancang dengan penambahan dua lapisan dense untuk memperbaiki performa prediksi. lapisan Dense pertama terdiri dari 64 neuron dengan fungsi aktivasi ReLU dan regularisasi L2 sebesar 0.01, diikuti oleh lapisan Dropout dengan tingkat dropout sebesar 30% untuk mengurangi overfitting. selanjutnya, lapisan Dense kedua menggunakan 32 neuron dengan fungsi aktivasi ReLU dan regularisasi L2 sebesar 0.01. model ini diakhiri dengan satu lapisan output yang terdiri dari satu neuron untuk menghasilkan nilai prediksi.

5. Model Compilation and Training Process

Setelah mendefinisikan bentuk model, model dioptimalkan menggunakan algoritma optimasi Adam dan fungsi mean absolute error (MAE). Proses pelatihan berlangsung selama 50 epoch, dengan ukuran *batch* adalah 64. Sebuah epoch mengacu pada jumlah iterasi penuh melalui seluruh dataset pelatihan, sementara ukuran *batch* menunjukkan jumlah sampel yang digunakan sebelum memperbarui bobot model. Selama proses pelatihan, data pelatihan (X_{train} , y_{train}) dan data validasi (X_{test} , y_{test}) diserahkan ke model untuk menilai kinerjanya sepanjang proses pelatihan.

6. Model Evaluation Process

Proses evaluasi model ini bertujuan untuk menilai kinerja model yang telah dilatih menggunakan matrik evaluasi

utama seperti *Mean Absolute Percentage Error* (MAPE), *Mean Absolute Error* (MAE), dan *Root Mean Squared Error* (RMSE). Pada proses ini data uji digunakan untuk mengukur kemampuan model dalam memprediksi nilai yang belum pernah diketahui sebelumnya. Data uji (X_{test}) menghasilkan prediksi (y_{pred}) yang kemudian dibandingkan dengan nilai aktual (y_{true}).

Tujuan utama dari evaluasi ini adalah meningkatkan kemampuan model untuk dapat melakukan generalisasi terhadap data baru, sehingga performa model tetap konsisten pada data yang tidak digunakan selama pelatihan.

a. Mean Absolute Percentage Error (MAPE)

MAPE mengukur tingkat kesalahan relatif antara nilai aktual dan prediksi. Rumus MAPE adalah:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \quad (8)$$

MAPE dinyatakan dalam persentase, memberikan wawasan tentang seberapa besar rata-rata kesalahan relatif. MAPE rendah menunjukkan akurasi prediksi yang tinggi. Namun, metrik ini sensitif terhadap nilai kecil pada y_i , sehingga perlu kehati-hatian jika dataset mengandung nilai nol atau mendekati nol.

b. Mean Absolute Error (MAE)

MAE mengukur rata-rata kesalahan absolut antara nilai aktual dan prediksi. Rumus MAE adalah:

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

MAE mengindikasikan sejauh mana prediksi rata-rata meleset dalam satuan asli data. Metrik ini memberikan nilai kesalahan absolut tanpa mempertimbangkan arah kesalahan, sehingga cocok untuk menilai akurasi keseluruhan model.

c. Root Mean Squared Error (RMSE)

RMSE mengukur rata-rata kesalahan kuadrat dengan memperbesar pengaruh kesalahan besar. Rumus RMSE adalah:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (10)$$

RMSE memberikan gambaran tentang besarnya kesalahan rata-rata dengan penalti yang lebih tinggi untuk kesalahan besar. Nilai RMSE lebih besar dari MAE jika terdapat

kesalahan besar dalam prediksi, menjadikannya metrik yang lebih sensitif.

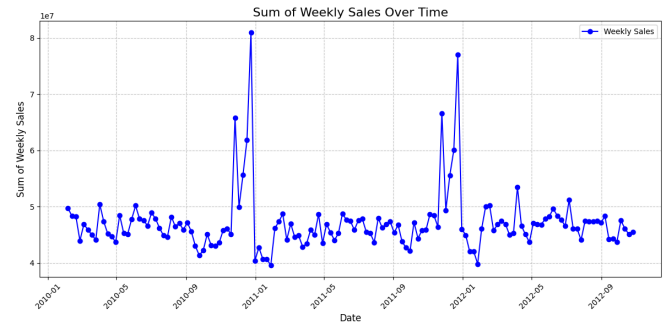
7. Future Sales Prediction

Proses prediksi penjualan masa depan dalam penelitian ini dirancang untuk mengevaluasi kemampuan model LSTM dalam membuat prediksi berdasarkan data historis menggunakan teknik sliding window. Pada tahap pertama, proses prediksi dilakukan secara interaktif. Model memprediksi langkah waktu berikutnya dengan menganalisis data penjualan sebelumnya dari periode tertentu (*window_size*). Setelah setiap prediksi, hasil prediksi ditambahkan ke data masukan, sementara data historis yang lebih lama dihapus. Hasil akhirnya adalah rangkaian prediksi yang dinamis dan berkelanjutan.

Metode *predict_next_n_week_sales()* digunakan untuk menghasilkan proyeksi penjualan jangka panjang selama *n* minggu berikutnya. Fungsi ini memperbarui data masukan secara mingguan dengan hasil prediksi sebelumnya dan mengembalikan nilai prediksi ke skala aslinya menggunakan transformasi balik (*inverse transform*). Dalam penelitian ini, fungsi ini diaplikasikan untuk memprediksi penjualan selama 8 minggu ke depan, memberikan gambaran menyeluruh tentang tren penjualan yang diharapkan.

Hasil dan Pembahasan

Dataset yang digunakan pada penelitian ini merupakan data penjualan dari toko Walmart. Berdasarkan data penjualan mingguan di toko Walmart dari rentang waktu Januari 2010 hingga September 2012, dapat dilihat pada **Gambar 4** terdapat fluktuasi pola musiman yang konsisten dengan penjualan mingguan berkisar antara 4-5 juta. Selain itu, terdapat pula peningkatan yang signifikan dalam penjualan di sekitar awal tahun 2011 dan 2012 yang memungkinkan indikasi adanya promosi liburan atau kejadian khusus pada periode tersebut. Promosi yang ditawarkan perusahaan menarik minat konsumen untuk berbelanja kebutuhan sehari-hari selama hari libur.



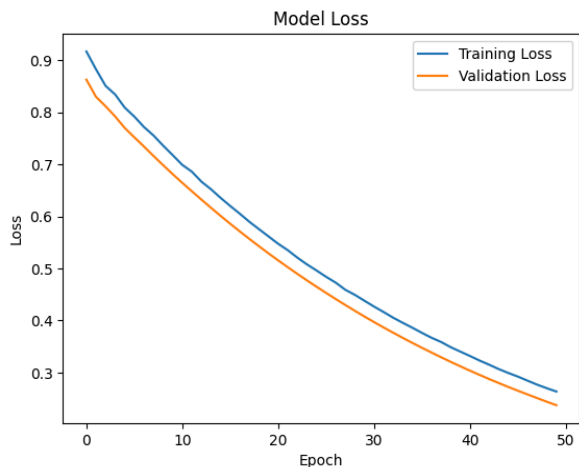
Gambar 4. Agregasi Total Penjualan per Minggu

Pemodelan LSTM yang dibangun pada penelitian ini memiliki empat *layer*, meliputi 4 LSTM *layer* dan 3 *dense layer*. Masing-masing *layer* LSTM diikuti dengan *dropout layer* sebesar 30%, di mana *layer* LSTM pertama sampai *layer* keempat memiliki masing-masing 256, 128, 64, dan 32 *neuron*. Selanjutnya, *dense layer* diisi dengan 64 *neuron*, 32 *neuron*, dan 1 *neuron* pada *output layer*.

Tabel 1. Konfigurasi Model

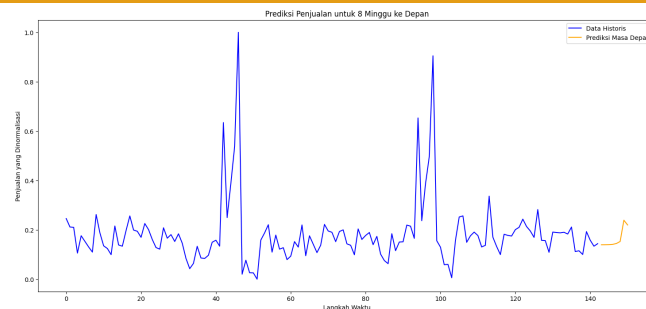
No	Jenis	Nilai/Uraian
1	Layer LSTM 1	256
2	Dropout 1	0,3
3	Layer LSTM 2	128
4	Dropout 2	0,3
5	Layer LSTM 3	64
6	Dropout 3	0,3
7	Layer LSTM 4	32
8	Dropout 4	0,3
9	Dense Layer 1	64
10	Dense Layer 2	32
11	Activation	ReLU
12	Dense Layer 3	1
13	Loss Function	MSE
14	Optimizer	Adam
15	Epochs	50
16	Batch Size	2

Pembagian dataset untuk pemodelan pada penelitian ini dibagi dengan perbandingan 80:20, di mana sebanyak 80% dari total data akan digunakan sebagai data *training* dan 20% sisanya akan digunakan sebagai data *testing*. Dengan menggunakan *epoch* sebesar 50 dan *batch size* sebanyak 2, evaluasi model pada penelitian ini dapat dikatakan cukup baik dalam memprediksi dikarenakan nilai *loss*-nya rendah berdasarkan visualisasi yang ditunjukkan pada **Gambar 5**. Grafik tersebut artinya model tidak mengalami *overfitting* saat proses *training* sambil menjaga kemampuan generalisasi pada data validasi.



Gambar 5. Performa Loss Model

Dapat dilihat hasil dari prediksi penjualan yang ditampilkan pada **Gambar 6**, bahwa data historis (garis biru) menunjukkan pola penjualan yang berfluktuasi dengan beberapa puncak signifikan, termasuk lonjakan tajam di sekitar langkah waktu ke-40 dan ke-100, yang kemungkinan disebabkan oleh peristiwa khusus atau faktor musiman yang memicu kenaikan penjualan drastis; secara keseluruhan, tren data historis cukup tidak stabil dengan fluktuasi yang sering terjadi. Di sisi lain, prediksi untuk 8 minggu kedepan (garis oranye) terlihat lebih halus dibandingkan data historis, dengan sedikit kenaikan pada minggu-minggu terakhir.



Gambar 6. Hasil Prediksi Penjualan Minggu ke Delapan

Hasil evaluasi kinerja model pada penelitian ini menggunakan tiga metrik evaluasi, yaitu RMSE, MAE, dan MAPE. Berikut adalah kinerja model dalam memprediksi penjualan untuk delapan minggu berikutnya.

Tabel 2. Hasil Evaluasi Model

Matrik Evaluasi	Nilai/Uraian
RMSE	2152542.70
MAE	1808226.93
MAPE	3.80%

Berdasarkan nilai RMSE sekitar **2,152,542.70** menunjukkan bahwa model mengalami kesalahan yang signifikan antara data prediksi dan data aktual. Hasil perhitungan MAE mengindikasikan bahwa rata-rata prediksi model meleset sekitar **1,808,226.93** dari nilai aktual. Perbandingan yang signifikan antara perhitungan RMSE dengan MAE kemungkinan terjadi pada saat penjualan mengalami peningkatan ekstrem yang terjadi secara mendadak. Namun, kesalahan yang cukup besar dalam perhitungan RMSE dan MAE dapat diterima ketika data penelitiannya berskala besar. Nilai MAPE memberikan persentase kesalahan **3.80%**, dimana semakin kecil nilai MAPE, maka model tersebut semakin baik dalam memprediksi model. Artinya, matrik evaluasi ini menunjukkan bahwa model yang dibangun sudah baik dalam memberikan prediksi penjualan.

Tabel 3. Hasil Prediksi Penjualan Minggu ke Delapan

Minggu	Prediksi Penjualan
Minggu ke-1	0.1394003331661224
Minggu ke-2	0.1395572423934936
Minggu ke-3	0.1399566382169723
Minggu ke-4	0.1409661620855331
Minggu ke-5	0.1444596052169799
Minggu ke-6	0.1526626199483871
Minggu ke-7	0.2388277351856231
Minggu ke-8	0.2195452302694320

Setiap minggunya, angka penjualan yang diprediksi akan mengalami peningkatan dan penurunan. Pada minggu pertama hingga minggu ketujuh, prediksi penjualan mengalami peningkatan mulai dari 0,139 hingga 1.238. Namun, pada minggu kedelapan, terdapat penurunan prediksi penjualan menjadi 0.219.

Kesimpulan

Penelitian ini berhasil menunjukkan bahwa model Long Short-Term Memory (LSTM) dapat digunakan untuk memprediksi penjualan mingguan di Walmart dengan hasil yang cukup baik. Model ini dirancang dengan 4 lapisan LSTM yang masing-masing memiliki 256, 128, 64, dan 32 neuron, dilengkapi dengan dropout sebesar 30% untuk mengurangi risiko overfitting. Evaluasi model menunjukkan tingkat kesalahan yang kecil, yaitu RMSE sebesar 2152542.70, MAE sebesar 1808226.93, dan MAPE sebesar 3,80%. Prediksi model menunjukkan pola peningkatan dan penurunan penjualan selama delapan minggu ke depan. Hal ini membuktikan bahwa model LSTM mampu mengenali pola musiman dan tren dalam data penjualan, sekaligus mempertimbangkan pengaruh faktor eksternal seperti suhu udara dan harga bahan bakar. Secara keseluruhan, model ini terbukti efektif dalam menganalisis data penjualan historis dan memberikan hasil yang baik dalam penelitian ini.

Benturan Kepentingan

Tidak ada konflik kepentingan yang terkait dengan penelitian ini, baik yang bersifat finansial, pribadi, maupun profesional, yang dapat mempengaruhi objektivitas dan hasil dari penelitian yang dilaporkan.

Ucapan Terimakasih

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Ibu Rohmi Dyah Astuti, selaku dosen pembimbing, atas segala bimbingan, dukungan, dan arahan yang diberikan selama pengerjaan tugas ini. Bimbingan yang diberikan sangat berharga dan berperan penting dalam mendukung penulis menyelesaikan penelitian ini dengan baik. Penulis juga mengucapkan terima kasih kepada teman-teman kelompok yang telah bekerja sama dengan sangat baik, saling membantu, dan memberikan kontribusi positif sehingga penelitian ini dapat terselesaikan tepat waktu.

Referensi

- [1] M. T. Tombeng dan Z. Ardian, "Prediksi Penjualan Supermarket Menggunakan Pendekatan Deep Learning," *Cogito Smart Journal*, vol. 7, no. 1, pp. 160–170, Juni 2021.
- [2] L. Wiranda and M. Sadikin, "Penerapan Long Short Term Memory pada Data Time Series untuk Memprediksi Penjualan Produk PT. Metiska Farma," *Jurnal Nasional Pendidikan Teknik Informatika*, vol. 8, no. 3, pp. 184–196, Dec. 2019.
- [3] G. Tamami dan M. Arifin, "Penggunaan LSTM dalam Membangun Prediksi Penjualan untuk Aplikasi Laptop Lens," *Jurnal FASILKOM*, vol. 14, no. 2, pp. 301–308, Agustus 2024.
- [4] A. M. Kusuma, R. A. Harianto, dan E. Pramana, "Prediksi Stok Produk Sari Roti untuk Penjualan Online Melalui Whatsapp Menggunakan Metode LightGBM dan LSTM," *JOUTICA*, vol. 8, no. 2, pp. 45–50, 2023.
- [5] F. C. Yulianto and N. Latifah, "Peramalan Penjualan Laptop Menggunakan Metode Long Short Term Memory (LSTM)," *Jurnal Fasilkom*, vol. 14, no. 2, pp. 428–436, Aug. 2024.
- [6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [7] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2014. [Online]. Available: <https://doi.org/10.48550/arXiv.1308.0850>.
- [8] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. ICANN'99 Int. Conf. on Artificial Neural Networks*, Edinburgh, Scotland, vol. 2, pp. 850–855, IEE, London, 1999.
- [9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, 2014. [Online]. Available: <https://arxiv.org/abs/1412.3555>.
- [10] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning," *arXiv preprint arXiv:1506.00019*, 2015. [Online]. Available: <https://arxiv.org/abs/1506.00019>.