

Prediksi Harga Brent Crude Oil Menggunakan Algoritma Long Short-Term Memory (LSTM)

Galang Maulana^{1*}, Nur Aini²

Universitas Amikom Yogyakarta, Indonesia^{1,2}

Email: ^{1*}galangmaulana@students.amikom.ac.id, ²nuraini@amikom.ac.id

Kata Kunci:

LSTM, Minyak Mentah
Brent, Prediksi, RMSE, MAPE

Abstrak: Minyak bumi, khususnya *Brent Crude Oil*, memegang peran penting dalam perekonomian global sebagai sumber energi utama dan komoditas dengan volume perdagangan tertinggi. Fluktuasi harga minyak mentah sering dijadikan indikator kondisi ekonomi global yang dapat mempengaruhi banyak sektor industri. Oleh karena itu, prediksi harga minyak sangat penting untuk meminimalkan risiko dan mendukung pengambilan keputusan yang lebih tepat. Penelitian ini bertujuan untuk mengembangkan model prediksi harga *Brent Crude Oil* menggunakan algoritma *Long Short-Term Memory* (LSTM), yang dipilih karena kemampuannya dalam menangani data deret waktu dan mempertahankan informasi jangka panjang. Data historis harga penutupan *Brent Crude Oil* dari 2 Januari 2018 hingga 27 September 2024 digunakan dalam penelitian ini. Metodologi melibatkan normalisasi data dan pembagian data menjadi tiga bagian: pelatihan, validasi, dan pengujian, diikuti dengan pengembangan model LSTM. Hasil evaluasi menunjukkan bahwa model LSTM dengan konfigurasi 70unit *neuron*, 2 *hidden layer*, 200 *epoch*, dan *batch size* 64 menunjukkan performa yang sangat baik, dengan menghasilkan RMSE sebesar 1,574829, R² sebesar 0,904717, dan MAPE sebesar 1,496758. Hasil ini menunjukkan bahwa model LSTM mampu menangkap pola historis harga *Brent Crude Oil* dengan akurasi yang sangat baik.

This is an open access article under the CC BY License (<https://creativecommons.org/licenses/by/4.0>).



Copyright holders:
Galang Maulana (2025)

PENDAHULUAN

Minyak bumi, yang juga dikenal sebagai minyak mentah, merupakan cairan berwarna coklat gelap yang mudah terbakar, terbentuk dari campuran kompleks hidrokarbon dan ditemukan di lapisan atas kerak bumi (M. Nadzir, R. G. Akbar, 2021). Selain berfungsi sebagai

komoditas dengan volume perdagangan tertinggi di dunia, minyak mentah juga merupakan sumber energi yang sangat penting dalam kegiatan ekonomi (Peng et al., 2020). Perubahan harga minyak dunia sering kali dianggap sebagai indikator kondisi perekonomian global, sehingga fluktuasinya kerap menjadi topik perbincangan hangat dalam berbagai forum ekonomi dan politik di sejumlah negara (Purnomo et al., 2020).

Banyak faktor yang berkontribusi terhadap ketidakstabilan harga minyak mentah. Selain ditentukan oleh tingkat penawaran dan permintaan, harga minyak juga dipengaruhi oleh berbagai sumber volatilitas alami, seperti siklus bisnis, aktivitas spekulatif, serta pengaruh politik (Zavadska et al., 2020). Mengingat fluktuasi harga minyak yang tidak terduga, prediksi harga minyak menjadi sangat penting untuk mengurangi risiko dan mendukung pengambilan keputusan yang lebih bijak dalam menghadapi perubahan harga yang sulit di prediksi.

Penelitian ini menggunakan data historis *Brent Crude Oil* untuk memprediksi harga minyak. *Brent* adalah salah satu jenis minyak mentah yang dijadikan sebagai patokan untuk harga minyak secara global (Fatkhurokhman Fauzi et al., 2023). Dalam memprediksi harga minyak dapat digunakan data deret waktu, Deret waktu merupakan data yang dikumpulkan secara berurutan dalam interval waktu tertentu, seperti harian, bulanan, tahunan, atau sesuai jangka waktu lainnya (Cahyani et al., 2023). Beberapa metode, seperti *Autoregressive Integrated Moving Average* (ARIMA), *Recurrent Neural Network* (RNN), dan *Long Short Term Memory* (LSTM), dapat diterapkan untuk memprediksi data deret waktu.

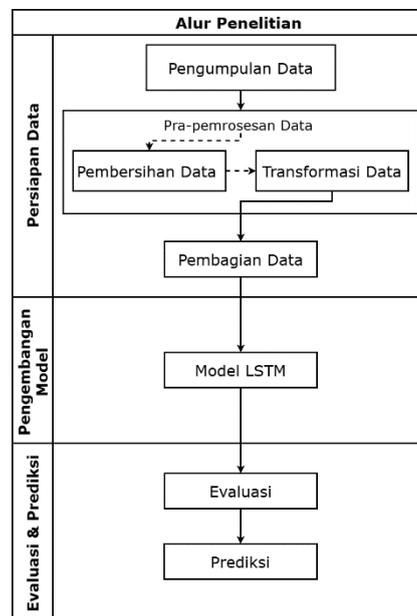
LSTM merupakan salah satu algoritma pembelajaran mendalam yang dipilih karena kemampuannya dalam mengelola pola *sequence* data serta mempertahankan informasi jangka panjang (Silalahi et al., 2024). LSTM memiliki sel memori yang menyimpan informasi untuk diteruskan ke *neuron* berikutnya, sementara unit gerbang mengatur bagian-bagian informasi yang perlu dipertahankan atau dihapus (Andiyantama et al., 2021). Berbagai penelitian terdahulu telah menunjukkan kemampuan LSTM dalam memprediksi data deret waktu. Salah satunya adalah penelitian yang dilakukan oleh Karno (Karno, 2020), yang membahas perbandingan antara metode LSTM dan ARIMA dalam memprediksi harga saham Telkom. Penelitian tersebut menunjukkan bahwa LSTM lebih unggul dibandingkan ARIMA dalam menangkap pola kompleks pada data runtun waktu. Dengan menggunakan kombinasi transformasi data, seperti *log*, *moving average*, dan *exponential weighted moving average*, hasil analisis menunjukkan bahwa model LSTM menghasilkan kesalahan prediksi rendah dengan nilai RMSE sebesar 1%, sementara ARIMA memiliki nilai RMSE sebesar 2%.

Selain itu, penelitian Rizkilloh & Widiyanesti (Moch Farryz Rizkilloh & Sri Widiyanesti, 2022) juga menunjukkan keunggulan algoritma LSTM dalam memprediksi harga *cryptocurrency* yang memiliki volatilitas sangat tinggi. Dengan menggunakan langkah-langkah seperti pengumpulan data, *preprocessing*, pembagian data, desain model, pelatihan, pengujian, dan evaluasi, LSTM terbukti mampu menghasilkan prediksi dengan baik, seperti pada prediksi harga DOGE dengan RMSE sebesar 0,0544. Hasil serupa ditemukan dalam penelitian yang dilakukan oleh Hansun & Young (Hansun & Young, 2021), yang memanfaatkan LSTM untuk memprediksi indeks sektor keuangan LQ45. Dengan menggunakan model sederhana tiga lapisan, penelitian ini berhasil menghasilkan kesalahan prediksi yang relatif kecil untuk saham BBCA dan BMRI, dengan masing-masing nilai MAPE sebesar 19,1020 dan 18,6135, yang menunjukkan efektivitas LSTM dalam memodelkan data deret waktu.

Berdasarkan latar belakang yang telah dipaparkan, penulis memutuskan untuk mengembangkan model prediksi harga *Brent Crude Oil* dengan menerapkan algoritma LSTM menggunakan data historis harga penutupan (*Price*) dari 2 Januari 2018 hingga 27 September 2024. Penelitian ini juga bertujuan untuk mengevaluasi tingkat akurasi model dalam melakukan prediksi harga minyak. Diharapkan, hasil dari model ini dapat membantu mengurangi risiko yang terkait dengan fluktuasi harga minyak dan memberikan dukungan dalam pengambilan keputusan strategis yang lebih tepat, serta memberikan wawasan berharga yang dapat digunakan oleh pelaku industri minyak dalam menghadapi ketidakpastian harga di pasar global.

METODE

Berikut ini disajikan diagram alur penelitian yang menggambarkan tahapan-tahapan yang dilakukan dalam memprediksi harga *Brent Crude Oil* menggunakan algoritma LSTM.



Gambar 1. Alur Penelitian

Pengumpulan Data

Data *Brent Crude Oil* diperoleh dari situs web *Investing.com* (<https://www.investing.com/commodities/brent-oil>) dalam format CSV (*Comma Separated Values*), mencakup periode dari 2 Januari 2018 hingga 27 September 2024. Secara keseluruhan, terdapat 1.742 baris data yang telah terkumpul. Dataset terdiri dari fitur *Date*, *Price*, *Open*, *High*, *Low*, *Vol.*, dan *Change*. Namun, fitur yang akan digunakan hanya *Price* dan *Date* saja. Fitur *Price* dipilih karena penelitian akan berfokus kepada prediksi harga penutupan *Brent Crude Oil*, sedangkan fitur *Date* diperlukan untuk menjaga urutan kronologis data, sehingga model dapat melakukan prediksi berdasarkan urutan waktu yang tepat.

Pra-Pemrosesan Data (*Data Preprocessing*)

Pra-pemrosesan adalah tahap awal pengolahan data yang bertujuan untuk

mempersiapkan data mentah sebelum digunakan dalam model pembelajaran mesin atau algoritma analisis (Silalahi et al., 2024). Data yang diperoleh dari berbagai sumber sering kali tidak siap untuk dianalisis secara langsung, karena bisa mengandung nilai yang hilang, *noise*, atau format yang tidak sesuai. Oleh karena itu, pra-pemrosesan diperlukan untuk meningkatkan kualitas data agar model dapat mempelajari pola dengan lebih baik dan menghasilkan hasil yang akurat.

Dalam penelitian ini, tahap pra-pemrosesan mencakup beberapa langkah berikut:

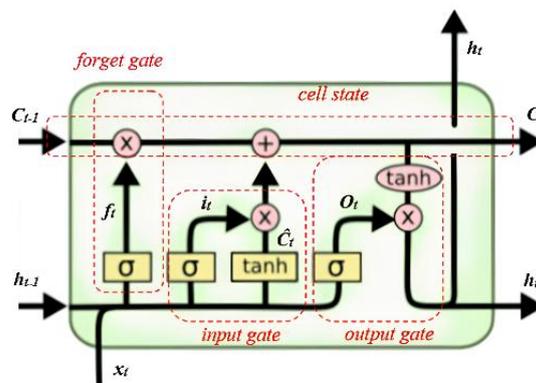
- Pembersihan Data (*Data Cleaning*): Proses ini meliputi identifikasi dan penanganan data yang hilang (*missing values*) pada data historis harga penutupan minyak *Brent Crude Oil*.
- Transformasi Data (*Data Transformation*): Data yang telah dibersihkan kemudian diubah formatnya agar sesuai dengan kebutuhan model. Pada penelitian ini, normalisasi data dilakukan menggunakan teknik *Min-Max Scaler* untuk menskalakan nilai data pada rentang $[0,1]$. Hal ini bertujuan untuk mempercepat konvergensi selama proses pelatihan model LSTM.

Pembagian Data (*Data Splitting*)

Data Splitting adalah langkah penting yang melibatkan pembagian dataset menjadi dua bagian utama: data pelatihan dan data pengujian. Data pelatihan berfungsi untuk melatih model dan mengidentifikasi pola dalam data, sedangkan data pengujian digunakan untuk mengevaluasi kemampuan model dalam memprediksi data yang belum pernah dijumpai sebelumnya (Silalahi et al., 2024). Pada penelitian ini, data dibagi menjadi tiga bagian utama, yaitu data latih, data validasi, dan data uji, dengan proporsi masing-masing 64% untuk data latih, 16% untuk data validasi, dan 20% untuk data uji.

Model LSTM (*Long Short-Term Memory*)

LSTM adalah salah satu arsitektur jaringan saraf berulang RNN yang pertama kali diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997 (Andiyantama et al., 2021). RNN dirancang secara khusus untuk menangani permasalahan pada data berurutan (*sequence data*). Namun, RNN sering menghadapi kendala *vanishing gradient* yang muncul ketika data sekuensial memiliki panjang yang signifikan (Selle et al., 2022). LSTM mampu mempelajari dan menangkap informasi jangka pendek maupun jangka panjang dalam data deret waktu, menjadikannya lebih efektif untuk tugas prediksi dan pemrosesan deret waktu. Pada model LSTM, *node* tersembunyi dalam RNN digantikan oleh *unit* memori yang dirancang khusus untuk mengatasi masalah *vanishing gradient*. Kunci utama dari LSTM terletak pada penggunaan keadaan sel dan gerbang yang memungkinkan model ini menangani tantangan tersebut secara efektif.



Gambar 2. Struktur Memory Sel LSTM [9]

Berikut ini adalah tahapan proses penerapan metode LSTM:

- *Forget gate* (f_t)

Proses dimulai dengan forget gate, bertugas memutuskan informasi apa dari cell state yang perlu dilupakan (Karno, 2020). Proses ini dihitung menggunakan fungsi aktivasi sigmoid (σ), menerima input berupa status tersembunyi sebelumnya (h_{t-1}) dan input saat ini (x_t). Fungsi sigmoid menghasilkan nilai antara 0 (menghapus informasi) dan 1 (menyimpan informasi).

Persamaan yang digunakan seperti berikut:

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

Penjelasan:

f_t = forget gate pada waktu t

σ = fungsi aktivasi sigmoid

W_f = bobot forget gate

h_{t-1} = nilai hidden state cell pada waktu sebelumnya

x_t = nilai input saat ini

b_f = bias forget gate

- *Input gate* (i_t)

Input gate berfungsi untuk memfilter dan menambah informasi baru ke dalam cell state. Dalam prosesnya, digunakan dua fungsi aktivasi, yaitu fungsi sigmoid bertugas memutuskan seberapa banyak informasi yang akan diperbarui, dan fungsi tanh membuat kandidat vektor nilai baru untuk dimasukkan ke memori (Akbar et al., 2023).

Persamaan yang digunakan dapat dijelaskan sebagai berikut:

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

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

Penjelasan:

i_t = input gate pada waktu t

W_i = bobot input gate

b_i = bias input gate

\tilde{C}_t = candidate cell state

tanh = fungsi aktivasi tanh

W_c = bobot candidate gate

b_c = bias candidate gate

- *Cell state* (c_t)

Cell state berfungsi untuk menyimpan informasi jangka panjang yang relevan. Pembaruan nilai dalam cell state dilakukan dengan memanfaatkan informasi dari input gate dan forget gate. Setelah informasi yang perlu dilupakan diputuskan oleh

forget gate, informasi baru yang dipilih oleh *input gate* akan digabungkan dengan memori yang ada untuk memperbarui *cell state* (Selle et al., 2022).

$$c_t = f_t * c_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Penjelasan:

c_t = *cell state* pada waktu t

i_t = nilai *input gate*

\tilde{C}_t = *candidate gate*

f_t = *forget gate*

c_{t-1} = nilai *cell state* sebelumnya

- *Output gate* (o_t)

Berfungsi untuk memilih bagian dari *memory cell* yang akan dihasilkan, dengan memanfaatkan fungsi aktivasi *sigmoid*. Nilai dari *memory cell* diproses melalui fungsi aktivasi *tanh*. Kemudian, kedua nilai dari *gate* tersebut dikalikan untuk menghasilkan keluaran akhir (h_t).

Persamaan yang digunakan dapat dijelaskan sebagai berikut:

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

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

Penjelasan:

o_t = *output gate* pada waktu t

W_o = bobot *output gate*

b_o = bias *output gate*

h_t = *hidden state* pada waktu t

c_t = *cell state*

Pengukuran performa model (*Evaluasi Model*)

Pengukuran performa adalah proses evaluasi yang digunakan untuk menilai seberapa baik suatu model atau sistem bekerja dalam melakukan prediksi (Sabar Sautomo & Hilman Ferdinandus Pardede, 2021). Pengukuran ini penting untuk memahami tingkat akurasi, ketepatan, dan efisiensi model dalam memberikan hasil prediksi yang diharapkan, dengan cara membandingkan nilai prediksi model dengan nilai aktual dari data yang tersedia. Berikut adalah beberapa evaluasi yang digunakan dalam pengukuran performa beserta persamaannya:

- MSE (*Mean Squared Error*)

MSE menghitung rata-rata dari kuadrat perbedaan antara nilai prediksi dan nilai aktual. Metrik ini memberikan penalti yang lebih besar untuk kesalahan yang besar karena perbedaan dikuadratkan (Sabar Sautomo & Hilman Ferdinandus Pardede, 2021).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (7)$$

Dimana y_i adalah nilai aktual, \hat{y}_i adalah nilai prediksi, dan n adalah jumlah data.

- **RMSE (Root Mean Squared Error)**
RMSE adalah akar dari MSE. Ini memberikan gambaran tentang seberapa besar perbedaan antara prediksi dan nilai aktual dalam skala yang sama dengan data (Hastomo et al., 2021).

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

Dimana n adalah jumlah data, y_i adalah nilai aktual, dan \hat{y}_i adalah nilai prediksi.

- **MAPE (Mean Absolute Percentage Error)**
MAPE menghitung rata-rata persentase kesalahan antara nilai prediksi dan nilai aktual. Ini berguna untuk melihat kesalahan relatif terhadap nilai asli (Budiprasetyo et al., 2023).

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Dimana y_i adalah nilai aktual, \hat{y}_i adalah nilai prediksi, dan n adalah jumlah data.

- **Koefisien Determinasi (*R-squared*)**
R-squared atau R^2 mengukur seberapa besar variabilitas dalam data aktual yang bisa dijelaskan oleh model prediksi (Chicco et al., 2021). Nilainya berkisar antara 0 hingga 1, di mana semakin dekat ke 1 menunjukkan model yang semakin baik dalam menjelaskan data.

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

Di mana \bar{y} adalah rata-rata dari nilai aktual. Nilai R^2 mendekati 1 menunjukkan model yang lebih baik dalam menjelaskan variabilitas data.

Prediksi

Setelah model terbaik diperoleh, model kemudian disimpan dalam format *.h5* menggunakan pustaka *TensorFlow*. Format *.h5* digunakan untuk menyimpan model LSTM beserta arsitektur, bobot, dan konfigurasi yang telah dilatih, sehingga model dapat dimuat kembali tanpa perlu pelatihan ulang. Selain itu, objek pendukung lainnya yang diperlukan untuk prediksi disimpan dalam format *.pkl* menggunakan pustaka *pickle*. Objek-objek yang disimpan dalam format *pkl* mencakup *scaler*, yang berguna untuk normalisasi data, di mana penyimpanan *scaler* ini penting agar data prediksi di masa depan dapat melalui proses skala yang sama dengan

data saat pelatihan, serta *DataFrame* yang berisi data historis harga *Brent* yang telah diolah atau dibersihkan untuk keperluan pelatihan. Dengan menyimpan *DataFrame* dalam format *pkl*, proses pemuatan dataset dapat dilakukan lebih cepat tanpa harus mengolah atau memproses data ulang dari awal, yang sangat efisien terutama untuk dataset berukuran besar.

Model yang telah disimpan bersama dengan *scaler* dan *DataFrame* kemudian digunakan untuk melakukan prediksi terhadap data harga *Brent* selama 10 hari ke depan, di mana data prediksi ini berada di luar dataset pelatihan maupun pengujian. Prediksi ini bertujuan untuk memberikan gambaran mengenai tren harga penutupan *Brent Crude Oil* yang belum tercatat dalam dataset, sehingga dapat memproyeksikan pergerakan harga berdasarkan pola yang telah dipelajari dari data historis selama proses pelatihan model.

HASIL DAN PEMBAHASAN

Pengumpulan Data

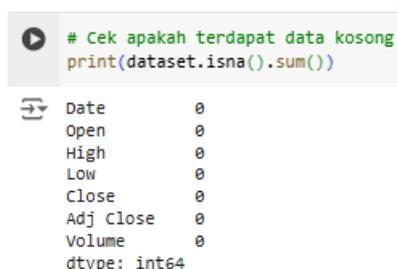
Data yang digunakan dalam penelitian ini merupakan data harga *Brent Crude Oil* diambil dari periode 2 Januari 2018 hingga 27 September 2024. Dataset terdiri dari 7 fitur, yaitu *Date*, *Price*, *Open*, *High*, *Low*, *Vol.*, dan *Change*, dengan total sebanyak 1.742 data. Namun, dalam penelitian ini, hanya fitur *Date* dan *Price* yang digunakan. Fitur *Price* dipilih karena penelitian ini berfokus kepada prediksi harga penutupan *Brent Crude Oil*, sedangkan fitur *Date* diperlukan untuk menjaga urutan kronologis data, sehingga model dapat melakukan prediksi berdasarkan urutan waktu yang tepat. Data diperoleh dari situs *Investing.com* (<https://www.investing.com/commodities/brent-oil>) dalam format CSV.

Pra-pemrosesan Data (*Data Preprocessing*)

Ada beberapa tahap yang dilakukan pada proses pra-pemrosesan data, yaitu Pembersihan Data (*Data Cleaning*) dan Transformasi Data (*Data Transformation*).

1. Pembersihan Data (*Data Cleaning*)

Setelah data diperoleh, langkah selanjutnya adalah pembersihan data dengan memeriksa apakah terdapat *missing values* pada dataset. Berikut adalah kode untuk melakukan pengecekan dan menghitung jumlah *missing values* di setiap kolom dataset:



```
# Cek apakah terdapat data kosong
print(dataset.isna().sum())
```

```
Date      0
Open      0
High      0
Low       0
Close     0
Adj Close 0
Volume    0
dtype: int64
```

Gambar 3. Pengecekan Dataset

Pada Gambar 3. tidak terdapat nilai kosong dalam dataset. Oleh karena itu, dapat dilanjutkan ke langkah berikutnya. Namun, Jika terdapat nilai kosong pada dataset *Brent Crude Oil*, nilai tersebut dapat diatasi dengan pengisian berdasarkan rata-rata dan penghapusan jika data tersebut tidak signifikan.

2. Transformasi Data (*Data Transformation*)

Fitur *Date* yang awalnya bertipe data *object* perlu dikonversi menjadi *datetime64* untuk memudahkan analisis deret waktu. Konversi ini penting guna

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1742 entries, 0 to 1741
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        1742 non-null   datetime64[ns]
1   Price       1742 non-null   float64
2   Open        1742 non-null   float64
3   High        1742 non-null   float64
4   Low         1742 non-null   float64
5   Vol.        1742 non-null   object
6   Change %    1742 non-null   object
dtypes: datetime64[ns](1), float64(4), object(2)
memory usage: 95.4+ KB
```

memastikan pengurutan, pengelompokan, dan perhitungan selisih waktu yang lebih akurat.

Gambar 4. Informasi Dataset setelah fitur *Date* diubah

Pada tahap transformasi data, dilakukan proses normalisasi data. Setelah memilih kolom *Date* sebagai indeks dan hanya menggunakan kolom *Price* untuk melakukan prediksi, data tersebut dinormalisasi menggunakan *MinMaxScaler* agar semua nilai berada dalam rentang antara 0 dan 1. Proses normalisasi ini penting untuk mengurangi ketidakseimbangan dalam data dan mempermudah algoritma dalam melakukan prediksi atau analisis berikutnya. Nilai pada fitur *Price* telah dinormalisasi dan hasilnya dapat dilihat pada Gambar 5.

```
      Normalisasi (Price)
Date
2018-01-02      0.430713
2018-01-03      0.442228
2018-01-04      0.444411
2018-01-05      0.440341
2018-01-08      0.442426
...
2024-09-23      0.500993
2024-09-24      0.513500
2024-09-25      0.497915
2024-09-26      0.479948
2024-09-27      0.484415

[1742 rows x 1 columns]
```

Gambar 5. normalisasi data

Pembagian Data (*Data Splitting*)

Proses *Data Splitting* adalah teknik membagi dataset menjadi beberapa subset yang digunakan untuk melatih dan menguji model prediktif. Pada tahap awal, dataset dibagi menjadi dua bagian, yaitu 80% sebagai data pelatihan (*Train*) dan 20% sebagai data pengujian (*Test*). Dari total dataset sebanyak 1.742 data, 1393 data dialokasikan untuk pelatihan, sedangkan 349 data sisanya digunakan untuk pengujian. Selanjutnya, 1.393 data yang dialokasikan untuk pelatihan, sebanyak 20% atau sekitar 278 data digunakan sebagai data validasi, sedangkan sisanya, yaitu 1.115 data, digunakan untuk proses pelatihan. Pembagian data validasi ini dilakukan secara otomatis selama proses pelatihan menggunakan parameter *validation_split=0.2* pada fungsi *model.fit()*. Tujuan dari data validasi adalah untuk memantau

kinerja model selama pelatihan dan mengurangi risiko *overfitting*, yaitu kondisi di mana model terlalu menyesuaikan diri dengan data pelatihan dan kehilangan kemampuan untuk generalisasi pada data baru. Data validasi memungkinkan evaluasi kinerja model secara berkala, sementara data pengujian hanya digunakan pada tahap akhir untuk menilai performa model secara menyeluruh.

Tabel 1. Informasi Pembagian Data

Pembagian Data	Jumlah Data	Rasio (%)
Data Latih	1.115	64%
Data Validasi	278	16%
Data Uji/Test	349	20%
Jumlah	1742	100%

Model Long Short-Term Memory (LSTM)

Tahapan utama dalam pengembangan model prediksi harga *Brent Crude Oil* menggunakan LSTM mencakup desain arsitektur jaringan, pemilihan parameter pelatihan, pembagian data, serta optimasi model dengan *loss function* dan *optimizer* yang tepat. Dalam implementasinya, model ini dibangun dengan menggunakan *library TensorFlow*, yang menyediakan alat untuk membangun dan melatih model jaringan saraf, serta *Matplotlib* untuk visualisasi performa model pada data.

- Struktur Jaringan LSTM:

Arsitektur jaringan LSTM dalam penelitian ini dirancang dengan beberapa lapisan untuk menangkap pola kompleks dalam data sekuensial. Pada *hidden layer* pertama LSTM, parameter *return_sequences=True* digunakan agar *output* tetap berupa sekuens yang diteruskan ke lapisan berikutnya, sehingga dapat memproses pola dalam urutan data. *hidden layer* terakhir LSTM menggunakan *return_sequences=False* untuk menghasilkan *output* tunggal sebagai nilai prediksi akhir. Model menerima *input* dengan bentuk (60, 1), di mana 60 menunjukkan jumlah *timestep*, yaitu panjang urutan waktu yang digunakan untuk melihat pola, dan 1 menunjukkan jumlah fitur dalam setiap *timestep*.

Untuk mengurangi risiko *overfitting*, setiap lapisan LSTM diikuti oleh lapisan *Dropout* dengan tingkat *dropout* 0,2, yang secara acak menonaktifkan 20% *neuron* dalam pelatihan untuk meningkatkan generalisasi model. Pada lapisan *output*, digunakan lapisan *Dense* dengan satu *neuron* untuk menghasilkan nilai prediksi akhir sebagai keluaran model.

- Pengujian parameter:

Pengujian model ini dilakukan dengan berbagai konfigurasi arsitektur jaringan LSTM:

- Unit *neuron* (50, 60, 70, dan 80)
- Layer LSTM (2, 3, dan 4)
- *Batch size* (32 dan 64)
- *Epoch* (50, 100, 150, dan 200)

- Parameter tetap

Berikut adalah parameter-parameter yang tidak akan diubah selama pengujian:

- *Timestep* (60)
- *Dropout layer* (0.2) untuk setiap *layer* LSTM

- *Dense layer (1)* sebagai *output*
- Penggunaan Loss Function dan Optimizer
Model ini memanfaatkan fungsi *loss Mean Squared Error* (MSE) untuk mengukur rata-rata kuadrat kesalahan antara nilai aktual dan prediksi selama pelatihan. Fungsi ini digunakan karena sesuai untuk masalah regresi, dengan memberikan penalti lebih besar pada kesalahan yang signifikan. Selain itu, *optimizer Adam* digunakan untuk mempercepat proses konvergensi menuju nilai optimal dari bobot model.
- Pembagian Data Pelatihan dan Validasi
Sebanyak 80% dataset digunakan untuk pelatihan, dengan 20% dari data tersebut dialokasikan untuk validasi menggunakan parameter *validation_split=0.2* pada fungsi *model.fit()*. Fungsi *model.fit()* dari *TensorFlow* digunakan untuk melatih model dengan menyesuaikan bobot berdasarkan data pelatihan. *TensorFlow* adalah *library open-source* yang sering digunakan dalam pengembangan model *deep learning*. Proses pelatihan dilakukan dengan jumlah *epoch* dan ukuran *batch* yang telah ditentukan, serta pengacakan data dimatikan dengan menggunakan *shuffle=False*. Tujuan dari validasi ini adalah untuk mengevaluasi kemampuan generalisasi model serta meminimalkan risiko *overfitting*. Setelah pelatihan, performa model dipantau melalui kurva pembelajaran (*learning curve*), yang menggambarkan perubahan nilai loss pada data pelatihan dan data validasi di setiap *epoch*.

Pengujian Model LSTM

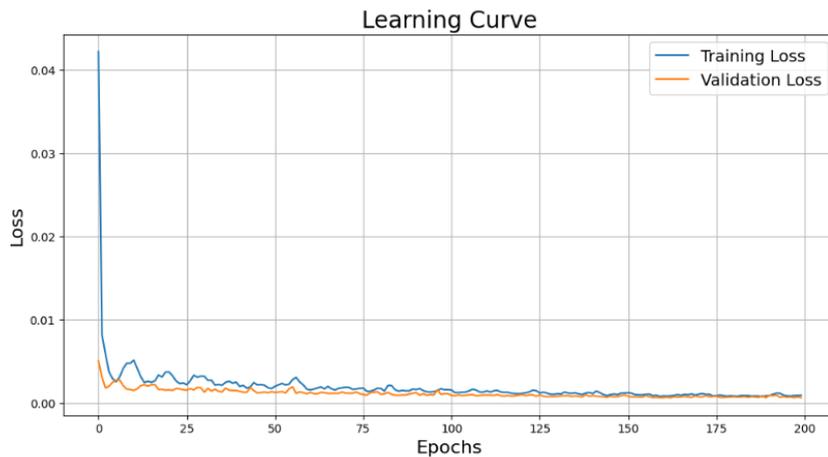
Pengujian model dilakukan untuk mengevaluasi performa model LSTM dalam memprediksi harga Brent Crude Oil. Proses evaluasi mencakup analisis terhadap kinerja serta efektivitas model berdasarkan berbagai metrik evaluasi, yaitu RMSE, R², dan MAPE. Pengujian ini dilakukan dengan menguji berbagai kombinasi parameter model, meliputi jumlah *hidden layer* LSTM (2, 3, dan 4), unit *neuron* (50, 60, 70, dan 80), *epoch* (50, 100, 150, dan 200), serta ukuran *batch* (32 dan 64). Secara keseluruhan, kombinasi ini menghasilkan 96 model berbeda yang diuji.

Tabel 2. Model LSTM Terbaik

<i>Unit Neuron</i>	<i>Hidden layer</i>	<i>Epoch</i>	<i>Batch_Size</i>	<i>RMSE</i>	<i>R²</i>	<i>MAPE</i>
50	2	200	64	1,716663	0,886782	1,621201
60	2	200	64	1,619675	0,899213	1,531272
70	2	200	64	1,574829	0,904717	1,496758
80	2	200	64	1,741366	0,883500	1,645282

Tabel 2 menampilkan model terbaik untuk setiap variasi unit *neuron*, yang dipilih berdasarkan RMSE terendah, R² tertinggi, dan MAPE terendah. Hasil pengujian menunjukkan bahwa seluruh model terbaik menggunakan konfigurasi 2 *hidden layer*, 200 *epoch*, dan *batch size* 64.

Model dengan 70 unit *neuron* memiliki performa terbaik, dengan nilai RMSE 1,574829, R² 0,904717, dan MAPE 1,496758, yang menunjukkan kemampuan model yang baik dalam memprediksi harga *Brent Crude Oil*. Visualisasi hasil prediksi untuk konfigurasi terbaik pada unit *neuron* 70 akan disajikan pada bagian berikut.



Gambar 6 Kurva pembelajaran model LSTM layer 2 dengan 70 unit

Gambar 6 menampilkan kurva pembelajaran (*learning curve*) dari model LSTM dengan konfigurasi 70 unit *neuron*, 2 *hidden layer*, batch size sebesar 64, dan 200 epoch. Pada kurva, terlihat perkembangan *training loss* dan *validation loss* selama proses pelatihan model.

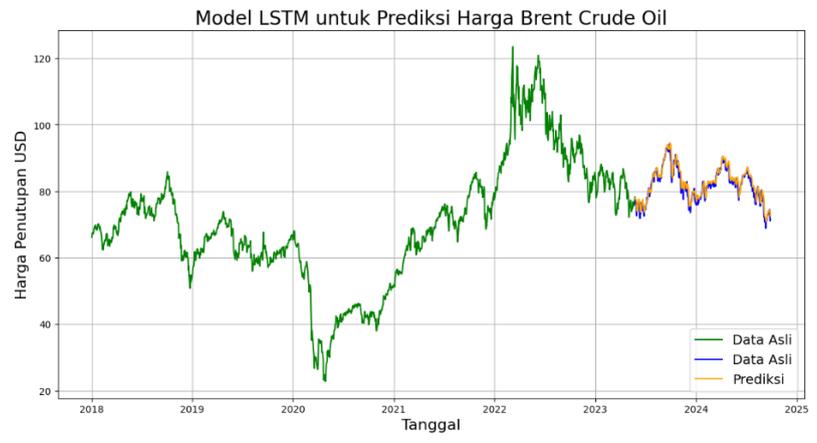
Pada beberapa *epoch awal*, model mencatat nilai *training loss* yang cukup tinggi, yaitu sekitar 0,0887. Namun, seiring bertambahnya jumlah *epoch*, nilai *training loss* menunjukkan penurunan yang signifikan. Pada akhirnya, model mencapai stabilitas dengan nilai *training loss* mendekati 0,00079 pada *epoch* ke-200. Sementara itu, nilai *validation loss* pada awal pelatihan relatif rendah, yakni 0,0050, dan terus menurun secara bertahap hingga mencapai 0,00063 pada *epoch* ke-200. Kedekatan antara nilai *training loss* dan *validation loss* mengindikasikan bahwa model telah mencapai konvergensi tanpa menunjukkan gejala *overfitting* maupun *underfitting*.



Gambar 7. harga aktual vs prediksi model LSTM *layer 2* dengan unit 50

Gambar 7 menunjukkan bahwa hasil peramalan memiliki kualitas yang baik, ditunjukkan oleh kemampuan pola prediksi yang secara konsisten mengikuti pola data aktual. Hal ini mengindikasikan bahwa model prediktif berhasil menangkap fluktuasi harga penutupan dengan baik dalam periode yang diamati. Berdasarkan hasil pengukuran evaluasi model, peramalan ini

memiliki RMSE sebesar 1,574829, R^2 sebesar 0,904717, dan MAPE sebesar 1,496758. Nilai-nilai ini menunjukkan tingkat kesalahan prediksi yang rendah.



Gambar 8. visualisasi model LSTM unit 70

Gambar 8 menyajikan gambaran menyeluruh mengenai hasil prediksi model terhadap harga penutupan minyak mentah *Brent*.

Prediksi

Setelah melalui proses evaluasi, model terbaik diperoleh dengan konfigurasi 70 unit *neuron*, 2 *hidden layer*, 200 *epoch*, dan *batch size* 64. Model ini disimpan dalam format *.h5* menggunakan pustaka *TensorFlow*, yang mencakup arsitektur, bobot, dan konfigurasi model sehingga dapat dimuat kembali tanpa perlu pelatihan ulang. Selain itu, objek pendukung seperti *scaler* dan *DataFrame* berisi data historis yang telah diolah disimpan dalam format *.pkl* menggunakan *pustaka pickle*. Penyimpanan *scaler* memastikan data prediksi di masa mendatang dinormalisasi dengan cara yang konsisten seperti pada saat pelatihan. *DataFrame* disimpan untuk mempercepat pemuatan dataset tanpa memerlukan pengolahan ulang. Model beserta objek pendukung ini kemudian digunakan untuk memprediksi harga *Brent Crude Oil* selama 10 hari ke depan, yang berada di luar data pelatihan dan pengujian, untuk memberikan gambaran proyeksi tren harga berdasarkan pola data historis. Berikut hasil implementasi ataupun pengujian.

Tabel 3. Prediksi dan Harga Asli

Tanggal	Prediksi	Harga Asli
28-09-2024	71,79	-
29-09-2024	71,74	-
30-09-2024	71,95	71,70
01-10-2024	72,29	73,21
02-10-2024	72,70	73,54
03-10-2024	73,14	77,11
04-10-2024	73,59	77,43
05-10-2024	74,00	-

06-10-2024	74,39	-
07-10-2024	74,73	80,35

Berdasarkan data prediksi harga Brent Crude Oil yang disajikan pada tabel 4.5, terdapat beberapa tanggal di mana data harga asli tidak tercatat, yaitu 28-09-2024, 29-09-2024, 05-10-2024, dan 06-10-2024. Hal ini disebabkan oleh liburunya pasar Brent Crude Oil pada akhir pekan, sehingga tidak ada aktivitas perdagangan atau pencatatan harga.

Pada tanggal 30 September 2024, model menghasilkan prediksi sebesar \$71,95, yang mendekati harga aktual sebesar \$71,70, dengan selisih hanya \$0,25. Pada 1 Oktober 2024, prediksi sebesar \$72,29 juga mendekati harga aktual \$73,21, dengan selisih \$0,92. Demikian pula, pada 2 Oktober 2024, prediksi sebesar \$72,70 tidak jauh berbeda dari harga aktual \$73,54, dengan selisih \$0,84. Namun, terdapat beberapa tanggal di mana perbedaan antara prediksi dan harga aktual signifikan.

Pada 3 Oktober 2024, prediksi sebesar \$73,14 lebih rendah dibandingkan harga aktual \$77,11, dengan selisih \$3,97. Kondisi serupa terjadi pada 4 Oktober 2024, di mana prediksi sebesar \$73,59 lebih rendah dari harga aktual \$77,43, dengan selisih \$3,84. Pada 7 Oktober 2024, perbedaan menjadi lebih mencolok, dengan prediksi sebesar \$74,73 yang cukup jauh dari harga aktual \$80,35, menghasilkan selisih sebesar \$5,62.

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, model LSTM dengan konfigurasi 70 unit neuron, 2 hidden layer, 200 epoch, dan batch size sebesar 64 menunjukkan performa yang sangat baik. Model ini menghasilkan RMSE sebesar 1,574829, yang mengindikasikan rata-rata deviasi prediksi dari nilai aktual sebesar \$1,57. Nilai R2 sebesar 0,904717 menunjukkan bahwa model mampu menjelaskan sekitar 90,47% dari variabilitas data harga Brent Crude Oil. Selain itu, nilai MAPE sebesar 1,496758 mencerminkan rata-rata kesalahan prediksi yang relatif rendah, yaitu sekitar 1,50%. Hasil ini menegaskan bahwa model LSTM efektif dalam menangkap pola historis data dengan baik.

Model ini mampu memberikan prediksi yang mendekati harga aktual pada beberapa tanggal tertentu. Pada tanggal 30 September 2024, model menghasilkan prediksi sebesar \$71,95, yang mendekati harga aktual sebesar \$71,70. Hal serupa juga terjadi pada 1 Oktober 2024, di mana prediksi sebesar \$72,29 cukup mendekati harga aktual sebesar \$73,21. Pada 2 Oktober 2024, model memprediksi harga sebesar \$72,70, yang juga relatif dekat dengan harga aktual sebesar \$73,54. Namun, terdapat beberapa tanggal di mana perbedaan antara prediksi dan harga aktual cukup signifikan.

Hal ini menunjukkan bahwa, meskipun model LSTM efektif dalam mengenali pola data historis dan menghasilkan prediksi yang mendekati harga aktual pada beberapa tanggal, model ini tetap memiliki keterbatasan dalam menangkap perubahan mendadak di pasar yang disebabkan oleh faktor eksternal yang tidak tercermin dalam dataset historis yang digunakan.

DAFTAR PUSTAKA

- Akbar, R., Santoso, R., & Warsito, B. (2023). Prediksi Tingkat Temperatur Kota Semarang Menggunakan Metode Long Short-Term Memory (Lstm). *Jurnal Gaussian*, 11(4), 572–579. <https://doi.org/10.14710/j.gauss.11.4.572-579>
- Andiyantama, M. Q., Zahira, I., & Irawan, A. (2021). Prediksi Energi Listrik Kincir Angin Berdasarkan Data Kecepatan Angin Menggunakan LSTM. *JITCE (Journal of Information*

- Technology and Computer Engineering*), 5(01), 1–7. <https://doi.org/10.25077/jitce.5.01.1-7.2021>
- Budiprasetyo, G., Hani'ah, M., & Aflah, D. Z. (2023). Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM). *Jurnal Nasional Teknologi Dan Sistem Informasi*, 8(3), 164–172. <https://doi.org/10.25077/teknosi.v8i3.2022.164-172>
- Cahyani, J., Mujahidin, S., & Fiqar, T. P. (2023). Implementasi Metode Long Short Term Memory (LSTM) untuk Memprediksi Harga Bahan Pokok Nasional. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, 11(2), 346. <https://doi.org/10.26418/justin.v11i2.57395>
- Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7, 1–24. <https://doi.org/10.7717/PEERJ-CS.623>
- Fatkurokhman Fauzi, Dewi Ratnasari Wijaya, & Tiani Wahyu Utami. (2023). Brent Crude Oil Price Forecasting using the Cascade Forward Neural Network. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 7(4), 964–969. <https://doi.org/10.29207/resti.v7i4.5052>
- Hansun, S., & Young, J. C. (2021). Predicting LQ45 financial sector indices using RNN-LSTM. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-021-00495-x>
- Hastomo, W., Karno, A. S. B., Kalbuana, N., Nisfiani, E., & ETP, L. (2021). Optimasi Deep Learning untuk Prediksi Saham di ... (*Jurnal Edukasi Dan ...*, 7(2), 133–140.
- Karno, A. S. B. (2020). Analisis Data Time Series Menggunakan LSTM (Long Short Term Memory) Dan ARIMA (Autoregressive Integrated Moving Average) Dalam Bahasa Python. *Ultima InfoSys : Jurnal Ilmu Sistem Informasi*, XI(1), 1–7.
- M. Nadzir, R. G. Akbar, S. B. (2021). Problematika Pengadaan Tanah Bagi Pembangunan Untuk Kepentingan Umum Pada Sektor Migas. *Jurnal de Jure*, 13(41), 92–111.
- Moch Farryz Rizkillah, & Sri Widiyanesti. (2022). Prediksi Harga Cryptocurrency Menggunakan Algoritma Long Short Term Memory (LSTM). *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 6(1), 25–31. <https://doi.org/10.29207/resti.v6i1.3630>
- Peng, J., Li, Z., & Drakeford, B. M. (2020). Dynamic characteristics of crude oil price fluctuation-from the perspective of crude oil price influence mechanism. *Energies*, 13(17). <https://doi.org/10.3390/en13174465>
- Purnomo, S. D., Istiqomah, I., & Badriah, L. S. (2020). Pengaruh Harga Minyak Dunia Terhadap Pertumbuhan Ekonomi, Inflasi, Dan Pengangguran Di Indonesia. *Jurnal PROFIT Kajian Pendidikan Ekonomi Dan Ilmu Ekonomi*, 7(1), 13–24. <https://doi.org/10.36706/jp.v7i1.11025>
- Sabar Sautomo, & Hilman Ferdinandus Pardede. (2021). Prediksi Belanja Pemerintah Indonesia Menggunakan Long Short-Term Memory (LSTM). *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(1), 99–106. <https://doi.org/10.29207/resti.v5i1.2815>
- Selle, N., Yudistira, N., & Dewi, C. (2022). Perbandingan Prediksi Penggunaan Listrik dengan Menggunakan Metode Long Short Term Memory (LSTM) dan Recurrent Neural Network (RNN). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 9(1), 155–162. <https://doi.org/10.25126/jtiik.2022915585>
- Silalahi, R. N., Informatika, T., Komputer, F. I., Dian, U., & Semarang, N. (2024). *Komputika : Jurnal Sistem Komputer Perbandingan Kinerja Metode Linear Regression , LSTM dan GRU untuk Prediksi Harga Penutupan Saham Coca-Cola Performance Comparison Of Linear Regression , LSTM & GRU Methods For Coca-Cola Stock Closing Price Prediction*. 13. <https://doi.org/10.34010/komputika.v13i2.12265>
- Zavadzka, M., Morales, L., & Coughlan, J. (2020). Brent crude oil prices volatility during major crises. *Finance Research Letters*, 32. <https://doi.org/10.1016/j.frl.2018.12.026>