

Visualisasi Prediksi Harga Saham Menggunakan Model Long Short-Term Memory (LSTM)

Windi Irmayani¹, Wanty Eka Jayanti²

Info Artikel

Diterima Januari 15, 2025
Revisi Februari 17, 2025
Terbit Maret 31, 2025

Keywords:

Prediksi
Harga Saham
Long Short-Term Memory

ABSTRACT

The stock market is a dynamic investment instrument affected by various factors, making it difficult to analyze. Traditional methods such as technical and fundamental analysis are often less accurate in capturing non-linear patterns and complex historical data. Deep learning algorithms, especially Long Short-Term Memory (LSTM), are beginning to be used to improve the accuracy of stock price predictions due to their ability to handle time series data and retain long-term information. This study focuses on the stock price prediction of Apple Inc., a leading technology company that reflects global industry trends. LSTM is used to overcome the limitations of traditional methods and provide a more accurate prediction model, especially in capturing stock price fluctuations due to new product launches, financial reports, and regulatory changes. This study aims to contribute to academics and market players in making more accurate prediction-based investment decisions, while expanding insights into the application of deep learning in the stock market. The result, it appears that the overall data trend is moving upwards, although there are some local (minor) declines.

Identitas Penulis:

Windi Irmayani¹, Wanty Eka Jayanti²
Universitas Bina Sarana Informatika, Program Studi Informatika Kampus Kota Pontianak¹, Program Studi Sistem Informasi Kampus Kota Pontianak²
Jl. Abdul Rahman Saleh No.18 A Pontianak^{1,2}
Email: windi.wnr@bsi.ac.id¹, wanty.wej@bsi.ac.id²

1. PENDAHULUAN

Pasar saham merupakan salah satu instrumen investasi yang sangat diminati oleh investor di seluruh dunia. Harga saham yang dinamis dan dipengaruhi oleh berbagai faktor seperti kondisi ekonomi, berita global, dan sentimen pasar menjadikannya subjek yang kompleks untuk dianalisis. Dalam upaya memprediksi harga saham, metode konvensional seperti analisis teknikal dan fundamental sering kali digunakan. Namun, keterbatasan dalam menangkap pola-pola kompleks nonlinear dan memiliki dependensi temporal jangka panjang. Keterbatasan ini telah mendorong peneliti untuk mengeksplorasi model yang lebih canggih, termasuk pendekatan berbasis deep learning[1].

Seiring berkembangnya teknologi, algoritma *deep learning* mulai diadopsi untuk meningkatkan akurasi prediksi harga saham. Salah satu pendekatan yang menunjukkan hasil menjanjikan adalah penggunaan *Long Short-Term Memory* (LSTM), sebuah jenis jaringan saraf tiruan yang dirancang khusus untuk menangani data deret waktu (*time series*) dan memiliki kemampuan untuk mempertahankan informasi dalam jangka panjang. Selain digunakan pada data jangka panjang, metode LSTM memiliki memori untuk menyimpan informasi yang akan digunakan kembali pada proses perhitungan untuk *gate* selanjutnya[2]. LSTM dapat mengatasi masalah yang dihadapi oleh metode tradisional dengan memanfaatkan ketergantungan temporal yang lebih baik dalam data historis.

Apple Inc., sebagai salah satu perusahaan teknologi terkemuka di dunia, memiliki saham yang sangat diperdagangkan dan mencerminkan tren industri teknologi global. Fluktuasi harga saham harus menjadi perhatian utama investor karena sangat berpengaruh terhadap return yang akan diterima dan melakukan manajemen risiko [3]. Prediksi harga saham memainkan peran penting dalam pengambilan Keputusan investasi, memungkinkan investor untuk memaksimalkan keuntungan dan meminimalkan risiko[4].

Penelitian terkait penggunaan LSTM dalam memprediksi harga saham telah menunjukkan hasil yang menggembirakan, dengan peningkatan akurasi yang signifikan dibandingkan metode prediksi tradisional. LSTM (Long Short Term Memory) merupakan salah satu jenis Recurrent Neural Network (RNN) yang merupakan modifikasi dari RNN [5]. Oleh karena itu, penelitian ini bertujuan untuk memanfaatkan LSTM dalam memvisualisasikan prediksi harga saham Apple Inc., dengan harapan dapat memberikan model yang lebih akurat dan dapat diandalkan untuk membantu pengambilan keputusan investasi.

Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan model prediksi harga saham yang lebih andal, tidak hanya untuk akademisi tetapi juga untuk investor dan pelaku pasar yang ingin mengambil keputusan investasi berdasarkan prediksi yang lebih tepat. Dengan memanfaatkan kekuatan LSTM untuk menangkap pola data historis yang kompleks, studi ini berupaya memberikan wawasan baru tentang implementasi algoritma *deep learning* dalam konteks pasar saham.

2. METODE

Metode merupakan cara atau prosedur yang sistematis dan terstruktur yang digunakan untuk mencapai tujuan atau menyelesaikan suatu masalah. Berikut ini metode yang dilakukan dalam penelitian ini:

Desain Penelitian

Penelitian ini termasuk dalam kategori penelitian kuantitatif dengan pendekatan eksperimental, di mana model prediksi harga saham dibangun dengan data historis. Studi kasus penelitian ini menggunakan saham Apple Inc. sebagai objek studi, dengan fokus pada prediksi harga saham berdasarkan data historis dalam bentuk deret waktu.

Pengumpulan Data

Sumber data, data historis harga saham Apple Inc. diperoleh dari penyedia data finansial Yahoo Finance. Data yang digunakan mencakup harga penutupan (*closing price*), harga tertinggi (*high*), harga terendah (*low*), dan volume perdagangan (*trading volume*) pada periode tertentu. Periode data, menentukan periode data yang akan digunakan, misalnya data harga saham harian dalam periode 5 tahun terakhir, untuk memastikan relevansi dan ketepatan dalam prediksi. Preprocessing Data, dengan melakukan proses pembersihan dan transformasi data, seperti menghapus nilai yang hilang (*missing values*), normalisasi atau standarisasi data (misalnya, menggunakan *Min-Max scaling* atau *Z-score normalization*) untuk meningkatkan kinerja model LSTM.

Proses Pelatihan (Training)

Pembagian Data data dibuat menjadi dua set, yaitu data pelatihan (*training data*) dan pengujian (*testing data*). Biasanya, 80% data digunakan untuk pelatihan. Untuk evaluasi yang lebih baik, teknik *cross-validation* juga bisa digunakan. Pelatihan Model, menjelaskan bagaimana model LSTM akan dilatih menggunakan data pelatihan, termasuk optimisasi dengan algoritma seperti Adam (*Adaptive Moment Estimation*) dan penggunaan *loss function* seperti *Mean Squared Error* (MSE).

Pemodelan dengan LSTM

LSTM bekerja untuk memodelkan data deret waktu. LSTM adalah varian dari jaringan saraf tiruan yang dirancang khusus untuk mengatasi masalah ketergantungan jangka panjang dalam data temporal. LSTM memiliki tiga komponen utama: *forget gate*, *input gate*, dan *output gate*, yang memungkinkan model untuk "mengingat" informasi penting dan "melupakan" informasi yang tidak relevan. Arsitektur Model, menentukan arsitektur LSTM yang akan digunakan, seperti jumlah layer LSTM, jumlah neuron pada setiap layer, dan jenis fungsi aktivasi. *Hyperparameter Tuning*, penjelasan tentang proses pencarian *hyperparameter* terbaik, seperti ukuran *batch*, jumlah *epoch*, dan *learning rate*, yang dapat dilakukan dengan teknik seperti *grid search* atau *random search* untuk menemukan kombinasi terbaik.

Analisis Hasil

Interpretasi hasil menyajikan dan menganalisis hasil dari model LSTM.

3. HASIL

Desain Penelitian

Desain penelitian adalah rencana atau struktur yang digunakan untuk melakukan penelitian. Desain ini menggambarkan bagaimana penelitian akan dilaksanakan, termasuk jenis data yang akan dikumpulkan, cara pengumpulan data, serta teknik analisis yang akan digunakan. Desain penelitian bertujuan untuk memastikan bahwa penelitian dilakukan secara sistematis, terstruktur, dan valid, sehingga hasil yang diperoleh dapat dipercaya dan relevan.

Kali ini penelitian menggunakan studi kasus yaitu penelitian mendalam mengenai pergerakan harga saham perhari dan memperkirakan pergerakan harga saham selanjutnya. Hal ini akan menjadi alternatif

referensi bagi para akademik, investor dan pihak-pihak yang berkepentingan untuk mengambil keputusan. Dan data saham Apple Inc. (AAPL) diambil dari website Yahoo Finance www.finance.yahoo.com.



Gambar 1 laman Informasi Saham Apple Inc.

Pengumpulan Data

Secara keseluruhan, pengumpulan data yang baik dan tepat sangat penting untuk kesuksesan penelitian. Tanpa data yang relevan dan berkualitas, bahkan model terbaik pun tidak akan dapat memberikan hasil yang akurat.

```
# Ambil data saham dari Yahoo Finance
```

```
ticker = "AAPL"
```

```
data = yf.download(ticker, period="1y", interval="1d")
```

```
[*****100%*****] 1 of 1 completed
```

Gambar 2 Skrip Python Pengambilan Data Saham Dari Yahoo Finance

Skrip Python yang digunakan untuk mengambil data saham dari Yahoo Finance menggunakan library yfinance. Berikut adalah penjelasan tentang koding tersebut:

1. Komentar dalam Bahasa Indonesia: # Ambil data saham dari Yahoo Finance Komentar ini menjelaskan bahwa skrip ini digunakan untuk mengambil data saham dari Yahoo Finance.
2. Mendefinisikan Ticker: ticker = "AAPL" Di sini, variabel ticker diset ke "AAPL", yang merupakan kode saham Apple Inc. Ticker ini digunakan untuk mengidentifikasi saham yang ingin diambil datanya.
3. Mengunduh Data Saham: data = yf.download(ticker, period="1y", interval="1d") Baris ini menggunakan fungsi download dari library yfinance untuk mengambil data saham berdasarkan ticker yang telah didefinisikan ("AAPL").

Berikut adalah parameter yang digunakan:

ticker: Kode saham yang ingin diambil datanya (dalam hal ini "AAPL").

period="1y": Data yang diambil adalah selama satu tahun terakhir.

interval="1d": Interval data adalah harian, artinya data akan diambil setiap hari.

Fungsi ini akan mengunduh data saham Apple (AAPL) untuk periode 1 tahun terakhir dengan interval harian.

4. Indikasi Proses Berhasil: Di bagian bawah, ada tanda progres [***** 100% *****] yang menunjukkan bahwa proses pengunduhan data telah selesai 100%.

Hasil dari kode ini adalah data saham Apple (AAPL) dalam bentuk dataframe yang berisi harga-harga saham harian dalam 1 tahun terakhir. Data ini biasanya mencakup kolom seperti 'Open', 'High', 'Low', 'Close', 'Adj Close', dan 'Volume'.

```
# Normalisasi data
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data['Close'].values.reshape(-1, 1))
```

Gambar 3 Skrip Python Normalisasi Data

Skrip bertujuan untuk melakukan normalisasi data menggunakan teknik MinMaxScaler dari pustaka `scikit-learn`. Berikut adalah penjelasan langkah demi langkah tentang apa yang dilakukan oleh kode ini:

1. Komentar: # Normalisasi data. Komentar ini menjelaskan bahwa kode berikut berfungsi untuk menormalkan data.
2. Membuat objek MinMaxScaler: `scaler = MinMaxScaler(feature_range=(0, 1))`. Di sini, objek MinMaxScaler dibuat dengan parameter `feature_range=(0, 1)`. Fungsi dari MinMaxScaler adalah untuk merubah data ke dalam rentang (skala) tertentu, dalam hal ini antara 0 dan 1. Ini adalah proses normalisasi, yang berguna agar nilai data menjadi lebih konsisten dan dapat diterima dengan baik oleh banyak algoritma pembelajaran mesin.
3. Melakukan transformasi pada data: `scaled_data = scaler.fit_transform(data['Close'].values.reshape(-1, 1))`. Baris ini melakukan dua hal utama:
 - `data['Close']`: Menyaring kolom "Close" dari DataFrame data. Kolom ini biasanya berisi harga penutupan saham untuk setiap hari.
 - `values`: Mengambil nilai dari kolom "Close" dalam bentuk array NumPy.
 - `reshape(-1, 1)`: Merubah bentuk array menjadi vektor kolom (dimensi 2D), yang diperlukan oleh MinMaxScaler.
 - `scaler.fit_transform()`: Proses pertama, `fit`, menghitung nilai minimum dan maksimum dari data. Proses kedua, `transform`, mengubah data tersebut ke dalam rentang yang telah ditentukan (0 hingga 1). Hasil dari transformasi ini adalah data yang sudah dinormalisasi dan disimpan dalam variabel `scaled_data`.

Secara keseluruhan, kode ini digunakan untuk menormalkan data harga penutupan saham (dalam kolom "Close") sehingga nilainya berada dalam rentang 0 hingga 1. Normalisasi ini penting untuk memperbaiki kinerja model Machine Learning, terutama ketika data memiliki skala yang sangat bervariasi.

```
def create_dataset(dataset, look_back=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-look_back-1):
        a = dataset[i:(i+look_back), 0]
        dataX.append(a)
        dataY.append(dataset[i + look_back, 0])
    return np.array(dataX), np.array(dataY)
```

Gambar 4 Skrip Python Create Dataset

Fungsi `create_dataset` digunakan untuk mengubah data *time series* menjadi format yang dapat digunakan oleh model pembelajaran mesin. Dalam fungsi ini: `dataX` berisi sekumpulan data fitur (yaitu nilai-nilai sebelumnya) yang digunakan untuk memprediksi nilai masa depan. `dataY` berisi nilai target (nilai masa depan yang ingin diprediksi). Dengan parameter `look_back`, Anda bisa mengontrol seberapa banyak data sebelumnya yang digunakan untuk memprediksi data yang akan datang.

Fungsi ini sangat berguna untuk *time series forecasting* di mana prediksi bergantung pada urutan data sebelumnya, seperti prediksi harga saham, suhu, atau permintaan produk.

Proses Pelatihan (Training)

Proses pelatihan dalam *machine learning* adalah tahap di mana model belajar dari data yang diberikan untuk memprediksi atau mengambil keputusan. Dalam konteks *time series forecasting* menggunakan model LSTM (Long Short-Term Memory).

```
# Buat dataset untuk training
look_back = 20 # Reduced look_back value
trainX, trainY = create_dataset(train_data, look_back)
testX, testY = create_dataset(test_data, look_back)
```

Gambar 5 Skrip Python Persiapan Dataset Untuk Training

Fungsi kode ini digunakan untuk membagi data menjadi dua set yang berbeda: Dataset pelatihan (trainX, trainY) yang digunakan untuk melatih model. Dataset pengujian (testX, testY) yang digunakan untuk mengevaluasi performa model.

Dengan menggunakan look_back=20, model akan mempelajari pola dari 20 titik data sebelumnya untuk membuat prediksi tentang titik data berikutnya. Ini sangat penting dalam time series forecasting, di mana model perlu mempelajari urutan data masa lalu untuk memprediksi masa depan.

Pemodelan dengan LSTM

Permodelan (*Modeling*) dalam konteks *machine learning* adalah proses membangun, melatih, dan mengevaluasi model untuk memecahkan masalah tertentu menggunakan data. Model adalah representasi matematis yang digunakan untuk membuat prediksi atau keputusan berdasarkan data yang dimasukkan. Dalam permodelan *machine learning*, ada beberapa langkah utama yang perlu dilakukan untuk mencapai model yang efektif dan akurat.

```
# Check if testX is empty before reshaping
if testX.size > 0:
    # Reshape input menjadi format yang sesuai untuk LSTM
    trainX = np.reshape(trainX, (trainX.shape[0], trainX.shape[1], 1))
    testX = np.reshape(testX, (testX.shape[0], testX.shape[1], 1))

    # Buat model LSTM
    model = Sequential()
    model.add(LSTM(units=50, return_sequences=True, input_shape=(look_back, 1)))
    model.add(LSTM(units=50))
    model.add(Dense(units=1))
    model.compile(optimizer='adam', loss='mean_squared_error')
    model.fit(trainX, trainY, epochs=100, batch_size=32)
else:
    print("testX is still empty. Consider further reducing look_back or increasing test data size.")
```

Gambar 6 Skrip Python Permodelan LSTM

Reshaping Data: Kode ini memastikan data memiliki format yang sesuai untuk digunakan dalam model LSTM. Membangun Model LSTM: Membuat model LSTM dengan satu layer LSTM dan satu layer Dense untuk memprediksi nilai output. Pelatihan Model: Model ini dilatih menggunakan data pelatihan untuk mempelajari hubungan antara data historis (time series) dan nilai masa depan. Cek Data: Memastikan bahwa data pengujian tidak kosong sebelum model digunakan untuk evaluasi. Ini adalah proses umum dalam membangun model LSTM untuk prediksi time series, di mana data input terdiri dari urutan waktu dan model mencoba untuk memprediksi nilai berikutnya dalam urutan tersebut.

Analisis Hasil

Analisis hasil dalam konteks *machine learning* merujuk pada tahap di mana kita mengevaluasi dan menilai kinerja model setelah pelatihan dilakukan. Proses ini sangat penting untuk memahami seberapa baik model telah belajar dari data yang diberikan, serta untuk mengidentifikasi area yang memerlukan perbaikan. Analisis hasil membantu kita untuk memastikan bahwa model dapat memberikan hasil yang relevan dan akurat untuk aplikasi yang dimaksud.

```

# Prepare data for plotting
if testX.size > 0: # Added conditional for plotting
    trainPredictPlot = np.empty_like(scaled_data)
    trainPredictPlot[:, :] = np.nan
    trainPredictPlot[look_back:len(trainPredict)+look_back, :] = trainPredict

    testPredictPlot = np.empty_like(scaled_data)
    testPredictPlot[:, :] = np.nan
    testPredictPlot[len(trainPredict)+(look_back*2)+1:len(scaled_data)-1, :] = testPredict

# Visualisasi data
plt.figure(figsize=(16,8))
plt.plot(scaler.inverse_transform(scaled_data), label='Data Asli')
plt.plot(trainPredictPlot, label='Prediksi Training')
plt.plot(testPredictPlot, label='Prediksi Testing')
plt.legend()
plt.show()

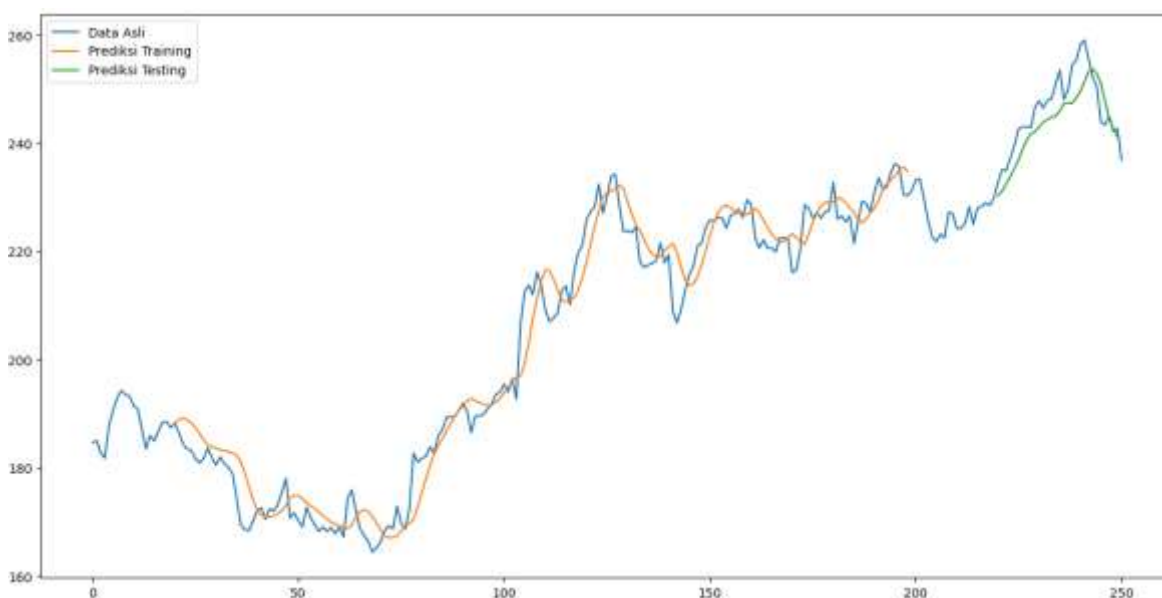
```

Gambar 7 Skrip Python Visualisasi Data

Kode ini digunakan untuk menyiapkan dan menampilkan visualisasi data yang membandingkan data asli dengan hasil prediksi dari model LSTM untuk data pelatihan dan data pengujian. Langkah-langkahnya meliputi:

1. Membuat array kosong untuk prediksi dan mengisinya dengan hasil prediksi model.
2. Menggunakan `scaler.inverse_transform` untuk mengubah data yang dinormalisasi kembali ke bentuk semula.
3. Menggambar grafik yang menunjukkan perbandingan antara data asli dan hasil prediksi pelatihan dan pengujian.

Visualisasi ini sangat berguna untuk mengevaluasi seberapa baik model melakukan prediksi terhadap data yang sebenarnya, terutama dalam kasus time series.



Gambar 8 Grafik Prediksi Harga Saham Apple Inc. (AAPL)

Grafik perbandingan antara data asli dan hasil prediksi untuk data training dan testing. Berikut adalah penjelasan tentang grafik tersebut:

1. Data Asli (dalam warna biru):

Garis biru mewakili data asli yang digunakan dalam model. Data ini adalah data yang belum diproses atau tidak berubah. Biasanya, ini adalah data yang sebenarnya yang ingin kita prediksi, misalnya harga saham atau nilai suatu parameter.

2. Prediksi Training (dalam warna oranye):

Garis oranye mewakili hasil prediksi model pada data training. Data pelatihan adalah data yang digunakan untuk melatih model. Prediksi ini dihasilkan berdasarkan pola yang dipelajari oleh model LSTM menggunakan data pelatihan. Perhatikan bahwa prediksi pada data pelatihan relatif dekat dengan data asli. Ini menunjukkan bahwa model cukup baik dalam mempelajari pola dari data pelatihan.

3. Prediksi Testing (dalam warna hijau):

Garis hijau mewakili hasil prediksi model pada data testing. Data pengujian digunakan untuk menguji kemampuan model dalam memprediksi data yang belum pernah dilihat sebelumnya. Meskipun prediksi hijau juga cukup baik, terlihat bahwa hasil prediksi untuk data pengujian ini sedikit lebih berfluktuasi dan tidak setepat prediksi pada data pelatihan. Hal ini bisa menunjukkan adanya sedikit overfitting atau kesulitan model dalam menangkap pola pada data pengujian.

4. Perbandingan antara Data Asli dan Prediksi:

Grafik ini memberikan gambaran yang jelas tentang seberapa baik model LSTM dalam memprediksi nilai berdasarkan data pelatihan dan data pengujian. Pada data pelatihan, model cukup akurat dalam memprediksi nilai yang sangat mirip dengan data asli. Namun, pada data pengujian, meskipun masih mendekati data asli, ada perbedaan kecil yang terlihat pada akhir grafik (bagian hijau).

5. Interpretasi:

Kinerja Model: Model berhasil menangkap pola dari data pelatihan dengan cukup baik, tetapi mungkin perlu perbaikan pada tahap pengujian untuk mengurangi kesalahan prediksi pada data yang tidak terlihat sebelumnya.

Overfitting atau Underfitting: Jika prediksi pelatihan sangat baik tetapi prediksi pengujian buruk, bisa menunjukkan overfitting, di mana model terlalu terikat dengan data pelatihan dan tidak dapat menggeneralisasi dengan baik pada data baru. Sebaliknya, jika kedua prediksi buruk, bisa menunjukkan underfitting.

Secara keseluruhan, grafik ini memberikan pemahaman visual yang berguna tentang seberapa baik model memprediksi data dalam time series dan bagaimana kinerjanya pada data yang belum dilihat sebelumnya. Dari grafik ini, tampak bahwa secara keseluruhan tren data bergerak naik, meskipun ada beberapa penurunan lokal (minor). Pada bagian akhir (di sekitar prediksi testing), terlihat sedikit penurunan harga, tetapi ini tidak cukup untuk disimpulkan sebagai tren penurunan harga saham secara keseluruhan.

4. KESIMPULAN

Pada bagian kesimpulan dalam penelitian ini merangkum temuan utama dari penelitian serta memberikan interpretasi tentang apa yang dapat diambil dari hasil yang diperoleh. Bagian ini juga memberikan rekomendasi untuk penelitian lebih lanjut atau implikasi praktis dari hasil yang dicapai. Berikut adalah beberapa hal yang dapat dijelaskan:

Algoritma LSTM dapat digunakan secara efektif untuk memprediksi harga saham Apple Inc. berdasarkan data historis. Meskipun ada beberapa kesalahan prediksi, model ini secara umum mampu mengikuti tren harga saham dengan akurasi yang cukup baik. Penelitian ini memberikan kontribusi signifikan dalam pengembangan metode prediksi harga saham menggunakan *deep learning*, khususnya algoritma LSTM. Dengan menggunakan model LSTM, penelitian ini memberikan bukti bahwa teknologi terbaru dalam kecerdasan buatan dapat meningkatkan akurasi dalam memprediksi harga saham, yang penting bagi investor dan pelaku pasar. Mengingat ketepatan prediksi yang lebih tinggi, model LSTM yang dikembangkan dalam penelitian ini dapat berguna bagi investor dalam mengambil keputusan investasi, terutama dalam pasar saham yang dinamis seperti saham Apple Inc. Model ini dapat membantu meminimalkan risiko dengan memberikan prediksi harga yang lebih andal.

Namun penelitian ini memiliki keterbatasan dalam hal data yang digunakan dalam penelitian, seperti periode data yang terbatas, yang dapat mempengaruhi hasil prediksi dalam jangka panjang. Jika hanya menggunakan data harga saham historis tanpa mempertimbangkan faktor eksternal (misalnya, berita atau peristiwa ekonomi global), model mungkin tidak dapat memprediksi harga dengan sempurna. Walaupun LSTM memberikan hasil yang baik, model ini bisa lebih baik lagi jika dikombinasikan dengan teknik lain, seperti model ensemble atau teknik hybrid yang menggabungkan LSTM dengan algoritma lain. Jika ada data yang hilang atau tidak akurat maka dapat mempengaruhi hasil prediksi.

Hasil dari penelitian ini dapat digunakan oleh para analis keuangan, hedge funds, dan investor institusional untuk mengembangkan strategi investasi berbasis prediksi harga saham yang lebih cerdas. Model ini dapat membantu mereka dalam merencanakan investasi jangka pendek dan jangka panjang dengan lebih akurat. Penelitian ini juga dapat diimplementasikan dalam aplikasi atau platform trading untuk memberikan rekomendasi prediksi harga saham secara *real-time*. Misalnya, aplikasi yang menggunakan model LSTM untuk memprediksi harga saham Apple Inc. di masa depan dapat membantu investor mengambil keputusan lebih cepat.

Dari hasil visualisasi tampak bahwa secara keseluruhan tren data bergerak naik, meskipun ada beberapa penurunan lokal (minor). Pada bagian akhir (di sekitar prediksi testing), terlihat sedikit penurunan harga, tetapi ini tidak cukup untuk disimpulkan sebagai tren penurunan harga saham secara keseluruhan.

REFERENSI

1. Arwansyah, Suryani, SY H, Ahyuna, Usman, Alam S. Model Prediksi Deret Waktu Menggunakan Deep Convolutional LSTM. Vol. 2024, Februari. Makassar; 2024 Feb.
2. Cahyani J, Mujahidin S, Fiqar TP. Implementasi Metode Long Short Term Memory (LSTM) untuk Memprediksi Harga Bahan Pokok Nasional. Jurnal Sistem dan Teknologi Informasi (JustIN). 29 Juli 2023;11(2):346.
3. Nur Anisa O, Ery Wibowo Agung R, Nurcahyono N. Faktor-Faktor yang Mempengaruhi Harga Saham: Berdasarkan Signaling Theory. Jurnal Akuntansi Indonesia [Internet]. 2022;11(2):85–95. Tersedia pada: <http://dx.doi.org/10.30659/jai.11.2.85-95>
4. Rahman A, Sri Istiyowati L, Azis Z. Implementasi Data Mining Dalam Prediksi Harga Saham Bbni Dengan Pemodelan Matematika Menggunakan Metode Lstm Dengan Optimasi Adam. Sintang; 2024 Des.
5. Dwirizqy Wimbassa M, Marsyah Noor T, Yasara S, Muhammad Arsyah T. Emotional Text Detection dengan Long Short Term Memory (LSTM). Jurnal Format. 2023;12(2):158–64.