

# Peramalan Multivariat Saham Bank Indonesia dengan Model ARIMA dan LSTM

Akhdan Ferdiansyah Ramadhan<sup>\*</sup>, I Made Artha Agastya

Fakultas Ilmu Komputer, Informatika, Universitas Amikom Yogyakarta, Sleman, Indonesia

Email: <sup>1</sup>\*akhdanferdiansyahr@students.amikom.ac.id, <sup>2</sup>artha.agastya@amikom.ac.id

Email Penulis Korespondensi: akhdanferdiansyahr@students.amikom.ac.id

Submitted: 14/05/2025; Accepted: 04/06/2025; Published: 06/06/2025

**Abstrak**—Peramalan harga saham merupakan aspek penting dalam analisis pasar keuangan, terutama dalam mendukung pengambilan keputusan investasi yang lebih tepat dan terinformasi. Penelitian ini membandingkan kinerja model statistik Autoregressive Integrated Moving Average (ARIMA) dengan tiga varian arsitektur Long Short-Term Memory (LSTM), yakni Vanilla LSTM, Bidirectional LSTM, dan Stacked LSTM. Dalam melakukan prediksi harga penutupan dan volume perdagangan saham bank di Indonesia, yaitu BBKA.JK, BBRI.JK, dan BMRI.JK. Data diperoleh melalui Kaggle dan diolah melalui tahapan normalisasi, transformasi, dan pelatihan model menggunakan Google Colab dan TensorFlow. Evaluasi dilakukan menggunakan metrik RMSE, MAE, dan MAPE. Hasil penelitian menunjukkan bahwa ARIMA unggul dalam memprediksi harga penutupan saham dengan rata-rata MAPE sebesar 1,9%, sedangkan Bidirectional LSTM memberikan hasil terbaik pada prediksi volume, terutama untuk saham BBRI dan BMRI. Namun demikian, tingkat kesalahan yang dihasilkan relatif tinggi (rata-rata MAPE sebesar 36,4%) dikarenakan data volume yang fluktuatif. Temuan ini menunjukkan bahwa karakteristik data sangat memengaruhi efektivitas model, di mana pendekatan LSTM memiliki kemampuan yang lebih baik dalam menangani pola non-linier yang kompleks dan memiliki keunggulan dalam menangani peramalan multivariat dibanding model ARIMA. Penelitian ini diharapkan dapat menjadi acuan dalam pemilihan model peramalan yang tepat dalam konteks pasar saham bank Indonesia. Hasil menunjukkan trade-off antara ARIMA yang unggul pada pola linier seperti harga penutupan dan LSTM yang lebih adaptif terhadap pola non-linier seperti volume perdagangan.

**Kata Kunci:** ARIMA; LSTM; Multivariate Forecasting; Financial Time Series; Indonesian Bank Stocks

**Abstract**—Stock price forecasting is a crucial aspect of financial market analysis, particularly in supporting more accurate and informed investment decision-making. This study compares the performance of the statistical Autoregressive Integrated Moving Average (ARIMA) model with three variants of the Long Short-Term Memory (LSTM) architecture, namely Vanilla LSTM, Bidirectional LSTM, and Stacked LSTM, in predicting closing prices and trading volumes of Indonesian bank stocks—specifically BBKA.JK, BBRI.JK, and BMRI.JK. The data were obtained from Kaggle and processed through normalization, transformation, and model training stages using Google Colab and TensorFlow. Evaluation was conducted using RMSE, MAE, and MAPE metrics. The results indicate that ARIMA performs better in forecasting closing prices, achieving an average MAPE of 1.9%, while Bidirectional LSTM yielded the best results in forecasting trading volumes, particularly for BBRI and BMRI stocks. However, the prediction error for volume data remains relatively high (average MAPE of 36.4%) due to its volatile nature. These findings suggest that data characteristics significantly influence model effectiveness. LSTM-based models demonstrate superior capabilities in capturing complex non-linear patterns and exhibit advantages in multivariate forecasting compared to the ARIMA model. This study is expected to serve as a reference for selecting appropriate forecasting models in the context of Indonesian banking stock markets. The results highlight a trade-off between ARIMA, which excels in modeling linear patterns such as closing prices, and LSTM, which is more adaptive to non-linear patterns like trading volumes.

**Keywords:** ARIMA; LSTM; Multivariate Forecasting; Financial Time Series; Indonesian Bank Stocks

## 1. PENDAHULUAN

Pasar saham merupakan komponen fundamental dalam perekonomian serta instrumen investasi yang memiliki peranan krusial [1]. Aktivitas jual beli di pasar saham mencerminkan kondisi ekonomi dan kinerja perusahaan, menjadikannya barometer penting dalam menilai kesehatan ekonomi suatu negara. Bagi para investor, pemahaman terhadap pergerakan harga saham di masa depan menjadi faktor esensial dalam pengambilan keputusan investasi [2]. Oleh karena itu, peramalan harga saham menjadi aspek penting dalam analisis pasar keuangan. Saham sendiri merupakan aset keuangan yang merepresentasikan kepemilikan dalam suatu perusahaan dan menjadi salah satu instrumen investasi yang banyak diminati karena potensi keuntungan yang tinggi [3].

Dalam konteks perekonomian Indonesia, sektor perbankan memiliki peranan strategis dalam mendukung pertumbuhan ekonomi. Bank-bank besar seperti Bank Central Asia (BBCA), Bank Rakyat Indonesia (BBRI), dan Bank Mandiri (BMRI) menjadi fokus utama bagi investor serta analisis keuangan karena stabilitas dan pengaruhnya yang signifikan terhadap perekonomian nasional. Kemampuan untuk melakukan peramalan harga saham dengan akurat memiliki signifikansi tinggi bagi investor, analisis keuangan, serta pembuat kebijakan guna meminimalkan risiko investasi dan memaksimalkan keuntungan [4]. Peramalan yang akurat dapat membantu investor menentukan waktu yang tepat untuk membeli atau menjual saham, serta memberikan gambaran mengenai tren pasar ke depan.

Secara umum, terdapat dua pendekatan utama dalam analisis saham, yaitu analisis fundamental dan analisis teknikal [5]. Analisis fundamental berfokus pada faktor-faktor mendasar perusahaan, seperti laporan keuangan, prospek industri, kondisi makroekonomi, kebijakan moneter, dan faktor eksternal lainnya yang dapat mempengaruhi kinerja perusahaan. Sementara itu, analisis teknikal memanfaatkan pola historis pergerakan harga saham untuk memprediksi harga di masa mendatang dengan menggunakan berbagai indikator seperti *Moving Average*, *Bollinger*

*Bands*, dan *Relative Strength Index* (RSI). Dalam hal ini, pemilihan harga penutupan sebagai target peramalan didasarkan pada signifikansinya sebagai indikator utama dalam evaluasi kinerja saham dan pengambilan keputusan investasi [6].

Dalam beberapa tahun terakhir, penerapan model *deep learning* (DL) dalam analisis data *time series* multivariat (MTS) mengalami perkembangan pesat di berbagai bidang, termasuk sektor perbankan, kesehatan, dan lingkungan [7]. *Time series* multivariat merujuk pada data *time series* yang melibatkan berbagai variabel dengan ketergantungan temporal, di mana setiap variabel (*close* dan *volume*) memberikan informasi tambahan yang bernilai untuk analisis dan peramalan [8]. Dalam konteks pasar saham, tantangan utama dalam peramalan harga saham adalah memahami pola dan hubungan dinamis antar variabel dalam data historis yang dapat dipengaruhi oleh berbagai faktor eksternal dan internal perusahaan.

Metode statistik tradisional, seperti *Autoregressive Integrated Moving Average* (ARIMA), telah banyak digunakan dalam peramalan *time series* karena kesederhanaannya dan interpretabilitasnya. ARIMA mampu menangkap pola linier dalam data historis dan telah digunakan secara luas dalam berbagai aplikasi keuangan dan ekonomi. Namun, model ini memiliki keterbatasan dalam menangkap pola non-linier yang sering kali muncul dalam pergerakan harga saham akibat volatilitas pasar dan faktor eksternal lainnya. Dengan perkembangan *deep learning*, khususnya *Long Short-Term Memory* (LSTM), muncul minat yang semakin besar terhadap potensinya dalam meningkatkan akurasi peramalan harga saham [9]. LSTM memiliki keunggulan dalam menangkap pola temporal yang kompleks dalam data *time series*, termasuk dalam mengakomodasi ketergantungan jangka panjang antar variabel yang sulit ditangkap oleh metode statistik tradisional [10].

Penelitian terdahulu yang dilakukan oleh Furizal et al. (2024) menunjukkan bahwa model sekuensial berbasis LSTM dapat digunakan untuk meramalkan harga saham dengan tingkat akurasi yang tinggi. Dalam penelitian tersebut, model yang terdiri dari lapisan tersembunyi LSTM dengan 200 unit menunjukkan kemampuan dalam menangkap pola temporal dan hubungan jangka panjang dalam data saham [1][11]. Hasil pengujian menunjukkan bahwa model ini mencapai nilai *Mean Absolute Error* (MAE) terendah sebesar 0,0142, yang mencerminkan tingkat akurasi yang lebih tinggi dibandingkan penelitian sebelumnya. Meskipun demikian, penelitian ini masih memiliki ruang untuk perbaikan guna meningkatkan akurasi peramalan lebih lanjut [1][12]. Selain itu, metode peramalan berbasis data multivariat terbukti lebih unggul dibandingkan metode univariat, dengan memanfaatkan informasi dari berbagai sumber seperti harga pembukaan, harga tertinggi, dan harga terendah, yang berkontribusi pada peningkatan akurasi prediksi [1][13].

Penelitian ini bertujuan untuk melakukan perbandingan kinerja model ARIMA dengan berbagai arsitektur *Long Short-Term Memory* (LSTM)—*Vanilla LSTM*, *Bidirectional LSTM*, dan *Stacked LSTM*—dalam meramalkan harga penutupan dan volume perdagangan saham perbankan di Indonesia. Tidak seperti penelitian sebelumnya yang umumnya hanya mengadopsi satu jenis model LSTM atau berfokus pada data univariat, penelitian ini menggunakan pendekatan multivariat dan membandingkan secara komprehensif berbagai arsitektur LSTM untuk mengevaluasi pengaruh struktur jaringan terhadap akurasi peramalan. Masing-masing arsitektur memiliki keunggulan berbeda: *Vanilla LSTM* sebagai bentuk dasar yang memproses data secara sekuensial; *Bidirectional LSTM* yang membaca data dalam dua arah waktu (*forward* dan *backward*) untuk menangkap konteks temporal yang lebih luas; serta *Stacked LSTM* yang menggunakan beberapa lapisan LSTM bertingkat untuk mengekstraksi representasi fitur yang lebih kompleks.

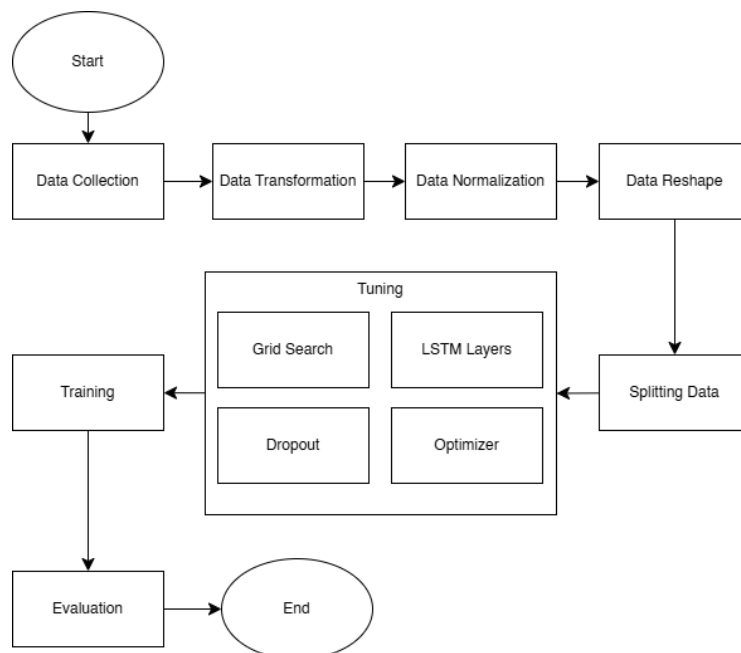
Penelitian ini juga mengkaji trade-off antara model ARIMA dan model LSTM dalam konteks saham bank Indonesia. Tantangan dalam memodelkan saham bank di Indonesia terletak pada kebutuhan untuk menangkap dinamika jangka pendek dan panjang secara bersamaan, terutama dalam menghadapi volatilitas volume perdagangan [14]. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi nyata dalam pengembangan strategi peramalan yang adaptif terhadap karakteristik pasar domestik, serta menjadi referensi bagi investor dan analis dalam pengambilan keputusan investasi yang lebih terinformasi.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Penelitian ini menggunakan metode kuantitatif melalui pendekatan eksperimen berbasis data *time series*, dengan fokus utama pada pengembangan dan evaluasi model peramalan harga saham. Pendekatan ini dipilih karena mampu menangkap pola temporal dan fluktuasi historis yang terjadi pada data pasar saham secara lebih akurat. Tujuan utama dari penelitian ini adalah untuk memperoleh model prediksi yang tidak hanya memiliki tingkat akurasi tinggi, tetapi juga mampu meminimalkan tingkat kesalahan (*error*) dalam peramalan *time series*. Guna mencapai tujuan tersebut, penelitian ini membandingkan berbagai pendekatan model *time series forecasting*, serta melakukan tuning parameter secara sistematis guna menemukan konfigurasi terbaik yang dapat menghasilkan performa optimal dalam meramalkan pergerakan harga saham di masa depan.

Penelitian ini terdiri dari beberapa tahapan utama, sebagaimana pada Gambar 1. dalam diagram tahapan penelitian, dimulai dengan pengumpulan data, transformasi data, hingga pembersihan data. Proses tersebut dilanjutkan dengan persiapan tuning, yang melibatkan *grid search*, *LSTM layers*, *dropout*, *optimizer*, dan pelatihan model untuk menghasilkan performa optimal.



Gambar 1. Tahapan Penelitian

Hasil pelatihan model kemudian diproses untuk melakukan peramalan data historis harga saham, diakhiri dengan evaluasi secara menyeluruh guna mengkaji keluaran akurasi dan error yang sesuai pada tujuan penelitian ini.

## 2.2 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini terdiri dari harga saham harian bank-bank besar Indonesia, termasuk Bank Mandiri (BMRI.JK), Bank Rakyat Indonesia (BBRI.JK), dan Bank Central Asia (BBCA.JK) yang berbentuk CSV (.csv) yang masih mentah. Data diperoleh dari platform Kaggle dan mencakup periode dari Januari 2019 hingga November 2024, sehingga memberikan wawasan mendalam tentang tren pasar saham. Data yang digunakan pada penelitian ini dapat diakses melalui tautan berikut [Bank Central Asia Stock Historical Price](#), [Bank Rakyat Indonesia Stock Historical Price](#), dan [Bank Mandiri Stock Historical Price](#).

## 2.3 Data Transformation

Proses ini dilakukan dengan melakukan konversi data CSV menjadi format numerik. Mengubah nilai harga saham ke dalam skala yang lebih seragam untuk memudahkan pemodelan, serta memilih fitur yang relevan, dalam hal ini hanya kolom “Close” dan “Volume” yang digunakan.

## 2.4 Data Normalization

Proses normalisasi data dilakukan untuk menyetarakan skala nilai pada setiap fitur agar memiliki rentang nilai yang seragam, sehingga proses pelatihan model tidak terpengaruh oleh perbedaan skala antar fitur. Pada penelitian ini, digunakan metode *Min-Max Scaling* yang mengubah nilai data kedalam rentang [0, 1]. Normalisasi ini bertujuan untuk mempercepat proses konvergensi model dan meningkatkan stabilitas perhitungan numerik selama pelatihan model [15]. Secara matematis, proses normalisasi *Min-Max* dapat dirumuskan sebagai berikut.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

Dimana  $x$  merupakan nilai asli,  $x'$  adalah nilai setelah normalisasi,  $x_{\min}$  adalah nilai minimum dalam fitur, dan  $x_{\max}$  adalah nilai maksimum dalam fitur. Normalisasi diterapkan pada kolom *close* dan *volume* dari dataset saham, karena kedua fitur ini digunakan sebagai input utama dalam proses pemodelan *time series*. Dengan normalisasi tersebut, model dapat memahami pola pergerakan data tanpa dipengaruhi oleh skala nilai yang besar atau fluktuasi ekstrem antar saham.

## 2.5 Data Reshape

Tahap ini bertujuan untuk mengubah struktur data agar sesuai dengan format input yang dibutuhkan oleh model *Long Short-Term Memory* (LSTM). Model LSTM memerlukan data dalam bentuk tiga dimensi, yaitu (*samples*, *timesteps*, *features*), agar dapat mempelajari ketergantungan temporal antar observasi secara efektif. Proses ini dilakukan melalui segmentasi *sliding window*, dimana dataset dibagi menjadi urutan-urutan berukuran tetap yang digeser satu langkah pada setiap iterasi. Secara umum, bentuk akhir dari dataset yang telah di-*reshape* adalah sebagai berikut.

$$X_{\text{reshaped}} \in \mathbb{R}^{n \times t \times f} \quad (2)$$

Dengan  $n$  sebagai jumlah sampel,  $t$  adalah panjang jendela waktu (*timesteps*) sebesar 60 hari atau data sebelum prediksi, dan  $f$  sebagai jumlah fitur (dalam penelitian ini, *close* dan *volume*). Dengan transformasi ini, LSTM dapat menangkap pola musiman, tren, dan fluktuasi jangka pendek dalam data historis untuk meningkatkan akurasi prediksi.

## 2.6 Split Data

Data yang telah diproses akan dibagi menjadi dua subset yaitu, data latih dan data uji. Data latih digunakan untuk melatih model dalam mengenali pola data historis, selanjutnya data uji digunakan untuk mengevaluasi kinerja model setelah proses pelatihan. Hal ini dilakukan penulis untuk mengoptimalkan performa pemodelan, rasio pembagian total dataset menjadi 80% data latih dan 20% sisa total dataset digunakan sebagai data uji.

## 2.7 Tuning

Pada proses *tuning* penulis menggunakan *Autoregressive Integrated Moving Average* (ARIMA) model, Vanilla LSTM, Bidirectional LSTM, dan Stacked LSTM model. Setiap model yang digunakan memiliki *tuning* dan keluaran yang berbeda.

### 2.7.1 Grid Search

Model *Autoregressive Integrated Moving Average* (ARIMA) merupakan pendekatan statistik yang umum digunakan dalam analisis dan peramalan data *time series* univariat. Model ini terdiri dari tiga komponen utama, yaitu *Autoregressive* (AR), *Integrated* (I), dan *Moving Average* (MA). Komponen AR menggambarkan ketergantungan linier antara nilai saat ini dengan nilai-nilai sebelumnya; komponen I (*differencing*) digunakan untuk menghilangkan tren dalam data guna mencapai kestasioneran; dan komponen MA merepresentasikan hubungan antara nilai saat ini dengan kesalahan acak (residual) dari model sebelumnya. Model ARIMA secara umum dinotasikan sebagai ARIMA ( $p, d, q$ ), dimana  $p$  menunjukkan orde AR,  $d$  adalah derajat differencing, dan  $q$  adalah orde MA [16].

Dalam penelitian ini, proses *tuning* parameter ARIMA dilakukan menggunakan pendekatan *grid search*, yaitu eksplorasi sistematis terhadap kombinasi parameter ( $p, d, q$ ) dalam ruang parameter yang telah ditentukan [17]. Rentang nilai yang digunakan adalah  $p \in \{1, 2, 3, 4\}$ , dan  $q \in \{1, 2, 3, 4\}$ . Untuk setiap kombinasi parameter, dilakukan proses pelatihan model menggunakan pendekatan *walk-forward validation*. Teknik ini mensimulasikan skenario prediksi nyata dengan cara memperbarui model secara berurutan berdasarkan observasi aktual dari hasil prediksi sebelumnya. Evaluasi performa model dilakukan menggunakan metrik *Root Mean Squared Error* (RMSE), dan kombinasi parameter dengan nilai RMSE terendah dipilih sebagai parameter optimal.

### 2.7.2 LSTM layers

Pada penelitian ini, digunakan tiga arsitektur *Long Short-Term Memory* (LSTM) untuk melakukan pemodelan data *time series*, yaitu Vanilla LSTM, Bidirectional LSTM, dan Stacked LSTM. Ketiga arsitektur ini dirancang menggunakan *framework* TensorFlow dengan antarmuka Keras, yang mendukung konstruksi *neural network* secara modular dan efisien.

Model Vanilla LSTM dibangun menggunakan 2 *layer* LSTM dengan 50 unit *hidden layer* dan diikuti oleh 1 *layer output* bertipe *Dense* yang berfungsi menghasilkan prediksi akhir [18]. Arsitektur ini dipilih karena kesederhanaannya dan kemampuannya dalam menangkap pola temporal dasar dari data historis. Dengan panjang jendela sebagai input (*window size*), model ini dirancang untuk memprediksi nilai pada hari ke- $n$  berdasarkan informasi sebelum hari ke- $n-1$ .

Untuk meningkatkan kompleksitas dan kemampuan dalam menangkap pola temporal dua arah, digunakan Bidirectional LSTM. Dalam arsitektur ini, memungkinkan proses pembelajaran dilakukan secara simultan dalam arah maju (*forward*) dan mundur (*backward*). Secara teknis, pada setiap langkah waktu, keluaran dari kedua arah digabungkan, biasanya melalui operasi konkatenasi atau penjumlahan. Proses ini dapat direpresentasikan secara matematis sebagai berikut:

$$h_t^{\rightarrow} = LSTM(x_t, h_{t-1}^{\rightarrow}) \quad (3)$$

$$h_t^{\leftarrow} = LSTM(x_t, h_{t+1}^{\leftarrow}) \quad (4)$$

$$h_t = \text{concat}(h_t^{\rightarrow}, h_t^{\leftarrow}) \quad (5)$$

Hal ini memungkinkan model untuk memanfaatkan informasi kontekstual masa lalu dan masa depan secara bersamaan, meskipun perlu dicatat bahwa dalam implementasi *time series* univariat yang bersifat kausal. Output dari kedua arah dikombinasikan melalui proses konkatenasi, dan hasilnya diteruskan ke lapisan *Dense* untuk menghasilkan prediksi.

Selanjutnya, untuk menguji kemampuan model dalam menangkap hubungan temporal yang lebih dalam, digunakan Stacked LSTM. Model ini terdiri dari dua lapisan LSTM bertingkat, dimana lapisan pertama dikonfigurasi agar dapat mengirimkan output sekuensial ke lapisan berikutnya. Lapisan kedua menerima *output* tersebut dan menghasilkannya sebagai input ke lapisan *Dense*. Dengan pendekatan ini, Stacked LSTM dapat membentuk representasi fitur yang lebih abstrak dari pola jangka panjang dalam data, yang potensial meningkatkan performa prediktif model pada data sekuensial yang kompleks.

### 2.7.3 Dropout

Untuk mengurangi risiko *overfitting*, yang umum terjadi pada *neural network* berkapasitas tinggi seperti LSTM, mekanisme regulasi *Dropout* dapat diterapkan. Namun demikian, berdasarkan implementasi dalam penelitian ini, penggunaan secara eksplisit dari lapisan *Dropout* parameter pada lapisan LSTM di ketiga arsitektur model. Hal ini menunjukkan bahwa regulasi diterapkan secara langsung dalam arsitektur model.

Penggunaan *dropout* membuat model lebih baik dalam merespons terhadap *overfitting*, terutama pada model yang kompleks seperti Bidirectional dan Stacked LSTM. Secara umum, *dropout* bekerja dengan menonaktifkan sejumlah unit neuron secara acak selama proses pelatihan model, sehingga mencegah model terlalu bergantung pada kombinasi fitur tertentu TensorFlow menyediakan parameter *dropout* untuk regulasi input ke unit LSTM, serta untuk meregulasi koneksi rekuren [19]. Penyesuaian nilai dropout dapat menjadi langkah strategis dalam tuning *hyperparameter* guna memperbaiki performa generalisasi model.

### 2.7.4 Optimizer

Proses optimasi dalam pelatihan ketiga model LSTM dilakukan menggunakan Adam *optimizer*, sebuah algoritma optimasi berbasis adaptif yang banyak digunakan dalam pelatihan *neural network* mendalam. Adam menggabungkan kelebihan dari metode *Adaptive Gradient Algorithm* (AdaGrad) dan *Root Mean Square Propagation* (RMSProp) dengan memanfaatkan momen pertama (*mean*) dan kedua (*variance*) dari gradien untuk memperbarui bobot.

Dalam implementasi ini, digunakan konfigurasi *default* dari TensorFlow, dimana laju pembelajaran (*learning rate*) dari adam secara implisit diset pada nilai 0.001, kecuali jika ditentukan secara eksplisit. Laju pembelajaran ini tergolong moderat dan umum digunakan untuk memulai pelatihan model LSTM. Pemilihan *optimizer* Adam didasarkan pada keunggulannya dalam konvergensi yang cepat dan kestabilan pada data *time series* yang memiliki distribusi gradien yang fluktuatif [20].

## 2.8 Training

Setelah proses transformasi data, normalisasi data, mengubah bentuk data, hingga tuning. Model dilatih dan diuji dengan *batch* dan *epoch* yang sudah ditentukan untuk mendapatkan keluaran yang optimal dalam melakukan peramalan data *time series*.

## 2.9 Evaluasi

Model yang telah dilatih dan diuji akan dievaluasi menggunakan beberapa metrik yaitu, *Root Mean Square Error* (RMSE) untuk mengukur deviasi antara nilai prediksi dan nilai aktual, *Mean Absolute Error* (MAE) untuk menghitung rata-rata kesalahan absolut dalam prediksi, dan *Mean Absolute Percentage Error* (MAPE) untuk menunjukkan kesalahan prediksi dalam bentuk persentase dari nilai aktual.

### 2.9.1 Root Mean Square Error (RMSE)

Metrik ini sangat sensitif terhadap kesalahan yang besar karena kesalahan dikuadratkan sebelum dihitung rata-ratanya, sehingga memberikan penalti lebih besar terhadap prediksi yang meleset jauh. Proses ini dapat direpresentasikan secara matematis sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (6)$$

Dimana  $y_t$  merupakan nilai atau data aktual pada waktu ke- $t$ ,  $\hat{y}_t$  adalah nilai prediksi pada waktu ke- $t$ , dan  $n$  adalah jumlah total observasi. Metrik yang digunakan untuk mengukur besar deviasi antara nilai prediksi dan nilai aktual pada model *machine learning*. RMSE memberikan perhatian lebih pada kesalahan besar.

### 2.9.2. Mean Absolute Error (MAE)

Metrik ini memberikan gambaran umum mengenai seberapa jauh prediksi model dari nilai sebenarnya tanpa mempertimbangkan arah kesalahan (positif atau negatif). Berikut proses yang direpresentasikan secara matematis sebagai berikut:

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \quad (7)$$

Dengan  $y_t$  sebagai nilai aktual,  $\hat{y}_t$  adalah nilai prediksi, dan  $n$  adalah jumlah total observasi. MAE menghitung rata-rata dari kesalahan nilai absolut antara nilai aktual dan nilai prediksi pada model *machine learning*.

### 2.9.3 Mean Average Percentage Error (MAPE)

Metrik ini berguna untuk memahami seberapa besar kesalahan relatif yang dilakukan oleh model. Proses ini dapat direpresentasikan secara matematis sebagai berikut:

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (8)$$

Dengan  $y_t$  sebagai nilai aktual,  $\hat{y}_t$  adalah nilai prediksi, dan  $n$  adalah jumlah observasi. MAPE menunjukkan kesalahan prediksi relatif dalam bentuk persentase terhadap nilai aktual yang berguna untuk interpretasi skala data pada model *machine learning*.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Pengumpulan Data

Tahap pertama dalam penelitian ini dilakukannya crawling data dari platform Kaggle menggunakan data perbankan di Indonesia antara lain “BBCA.JK”, “BBRI.JK”, dan “BMRI.JK”. Sebanyak 1.436 data diperoleh dari crawling tersebut. Menampilkan dataset yang digunakan sebagai sampel penelitian.

**Tabel 1.** Contoh Isi Dataset BBKA.JK

| Date       | Open   | High   | Low    | Close  | Adj Close | Volume      |
|------------|--------|--------|--------|--------|-----------|-------------|
| 2019-01-02 | 5200,0 | 5245,0 | 5200,0 | 5240,0 | 4642,3    | 35.956.000  |
| 2019-01-03 | 5200,0 | 5220,0 | 5115,0 | 5180,0 | 4589,1    | 72.358.000  |
| 2019-01-04 | 5175,0 | 5205,0 | 5125,0 | 5205,0 | 4611,3    | 51.465.000  |
| 2019-01-07 | 5265,0 | 5325,0 | 5245,0 | 5245,0 | 4646,7    | 73.438.000  |
| 2019-01-08 | 5230,0 | 5320,0 | 5205,0 | 5240,0 | 4642,3    | 66.247.500  |
| 2019-01-09 | 5300,0 | 5320,0 | 5225,0 | 5255,0 | 4655,6    | 84.871.000  |
| 2019-01-10 | 5300,0 | 5300,0 | 5190,0 | 5255,0 | 4655,6    | 114.314.500 |
| 2019-01-11 | 5280,0 | 5300,0 | 5235,0 | 5250,0 | 4651,1    | 84.826.500  |

Pada Tabel 1 dataset berupa file *Coma Separated Vector* (.csv) yang diolah dengan pembersihan dan transformasi data untuk memastikan data sesuai dengan kelengkapan informasi. Data ini digunakan sebagai basis data latih dan validasi model ARIMA serta varian model LSTM. Tipe data pada dataset dapat dilihat pada Tabel 2.

**Tabel 2.** Tipe Data Pada Dataset Bank

| Kolom Data | Tipe Data |
|------------|-----------|
| Open       | Float     |
| High       | Float     |
| Low        | Float     |
| Close      | Float     |
| Adj Close  | Float     |
| Volume     | Integer   |

Kolom-kolom seperti *open*, *high*, *low*, *close*, dan *adj close* merupakan data numerik bertipe *float* untuk merepresentasikan harga yang biasanya memiliki nilai desimal. Sebaliknya, kolom volume adalah satu-satunya yang bertipe *integer* merepresentasikan jumlah unit saham (bilangan bulat, tanpa desimal).

#### 3.2 Transformasi Data

Pada proses ini dilakukan feature selection guna memilih data untuk digunakan dalam pelatihan model *machine learning*. Tujuan dilakukannya transformasi data adalah untuk memilih kolom data yang dibutuhkan. Berikut struktur untuk dataset yang telah dilakukan feature selection pada Tabel 3.

**Tabel 3.** Contoh Bentuk Dataset Setelah Proses Transformasi Data

| Date       | Close  | Volume      |
|------------|--------|-------------|
| 2019-01-02 | 3662,5 | 45.319.200  |
| 2019-01-03 | 3587,5 | 74.174.000  |
| 2019-01-04 | 3725,0 | 78.514.000  |
| 2019-01-07 | 3712,5 | 53.624.800  |
| 2019-01-08 | 3675,0 | 80.073.200  |
| 2019-01-09 | 3787,5 | 104.903.400 |
| 2019-01-10 | 3850,0 | 130.603.200 |
| 2019-01-11 | 3850,0 | 88.899.400  |
| 2019-01-14 | 3850,0 | 71.320.000  |

Dari keseluruhan atribut yang tersedia dalam dataset awal, dua fitur utama yang dipilih adalah *close* dan *volume*, karena keduanya memiliki keterkaitan langsung dengan indikator teknikal dalam analisis pasar saham. Guna menyaring atribut-atribut yang memiliki kontribusi signifikan terhadap proses prediksi harga saham dan mengurangi risiko *overfitting*.

### 3.3 Normalisasi Data

Proses transformasi data pada penelitian ini menghasilkan data masukan yang lebih terstandarisasi dan siap digunakan dalam pelatihan model LSTM. Transformasi dilakukan dengan menerapkan metode normalisasi *Min-Max*, yang mengubah skala fitur numerik *close* dan *volume* ke dalam rentang 0 hingga 1. Transformasi dilakukan menggunakan fungsi *MinMaxScaler* dengan menghasilkan bentuk array. Bentuk dari dataset yang sudah dilakukan transformasi dapat dilihat pada Tabel 4.

**Tabel 4.** Contoh Bentuk Dataset Setelah Proses Normalisasi Data

| Date       | Close | Volume |
|------------|-------|--------|
| 2019-01-02 | 0,29  | 0,05   |
| 2019-01-03 | 0,29  | 0,08   |
| 2019-01-04 | 0,30  | 0,10   |
| 2019-01-07 | 0,30  | 0,05   |
| 2019-01-08 | 0,31  | 0,06   |

Dataset diubah menjadi bentuk numerik berskala kecil agar lebih mudah diproses dengan tetap mempertahankan struktur dua dimensi berupa pasangan fitur yang saling berkaitan, menggunakan *library scikitlearn*. Dengan bentuk data ini, proses pembelajaran oleh model menjadi lebih efisien dan dapat dibandingkan secara langsung setelah dilakukan transformasi balik (*inverse transform*) ke skala aslinya.

### 3.4 Tuning

Setelah data dibagi menjadi data latih dan data tes, selanjutnya dilakukan proses tuning pada model *machine learning* untuk menghasilkan keluaran yang sesuai. Pada model ARIMA sebelum melakukan *tuning*, dilakukan pengecekan menggunakan *ADF test* untuk melihat adakah data yang memiliki nilai *null* atau data yang tidak stasioner. Setelah itu, dilakukan *ACF* dan *PACF Plots* untuk mengidentifikasi parameter *AR* (*p*) dan *MA* (*q*) untuk *fitting* model ARIMA. Kemudian menggunakan *grid search* untuk menemukan kombinasi parameter ARIMA yang optimal (*p*, *d*, *q*). Hasil dari penggunaan *grid search* ditemukan parameter ARIMA yang optimal yaitu *p* = 1, *d* = 1, dan *q* = 1.

Pada model Vanilla LSTM dan Bidirectional LSTM dilakukan tuning dengan membuat 2 *layer* LSTM dengan masing-masing *layer* menggunakan 50 unit *hidden layer*, 1 *layer dropout* sebesar 0,2 untuk menghindari terjadinya *overfitting* dan 2 *dense layer* untuk memprediksi *close* dan *volume* pada model. Kemudian pada model Stacked LSTM dilakukan tuning dengan membuat 3 *layer* LSTM dengan masing-masing *layer* menggunakan 50 unit *hidden layer*, 1 *layer dropout* sebesar 0,2 untuk menghindari *overfitting* dan 2 *dense layer* untuk memprediksi *close* dan *volume*. Selanjutnya model *compiler*, model menggunakan *optimizer adam* dan *loss function mean squared error*.

### 3.5 Training

Pada proses pelatihan model, model ARIMA dan LSTM dilatih untuk memprediksi pattern atau nilai data historis yang telah dibagi saat proses *split data* dengan menggunakan data validasi sebagai acuan utama. Pada model LSTM menggunakan *model\_fit()* dengan parameter *epochs*=10, *batch size*=32, dan *validation data* agar model mendapatkan iterasi *pattern* yang baik. Serta *test loss* untuk menelisik seberapa besar *output* prediksi harga saham yang meleset. Berdasarkan Gambar 2. terlihat bahwa nilai *training loss* mengalami penurunan tajam pada *epoch* pertama, kemudian berangsur stabil hingga akhir *epoch* ke-10.



**Gambar 2.** Grafik Epoch Training dan Validasi Loss Model Vanilla LSTM

Hal ini menunjukkan bahwa model berhasil mempelajari pola data dengan cukup cepat pada tahap awal pelatihan. Sebaliknya, *validation loss* cenderung berfluktuasi dengan nilai yang relatif lebih tinggi dibandingkan *training loss*, meskipun tetap berada dalam kisaran yang stabil.

### 3.6 Evaluasi Model

Setelah semua model dilakukan training dan loss test sudah mendapati minimum kesalahan. Kemudian dilakukan kalkulasi evaluasi model menggunakan 3 metrik yaitu, RMSE, MAE, dan MAPE. Berikut evaluasi dari masing-masing model dan perbankan.

#### a. Evaluasi Model Setiap Perbankan

**Tabel 5.** Close BBCA.JK

| Models             | RMSE  | MAE   | MAPE |
|--------------------|-------|-------|------|
| ARIMA              | 124,1 | 95,0  | 0,9% |
| Vanilla LSTM       | 200,5 | 160,7 | 1,6% |
| Bidirectional LSTM | 227,8 | 192,0 | 1,9% |
| Stacked LSTM       | 231,5 | 188,0 | 1,9% |

Pada Tabel 5. prediksi harga penutupan saham BBCA.JK, model ARIMA menunjukkan performa terbaik dengan nilai RMSE, MAE, dan MAPE terendah secara konsisten. Rata-rata MAPE dari keempat model adalah 1,5%, yang menunjukkan bahwa secara umum prediksi harga penutupan berada dalam kategori error yang rendah.

**Tabel 6.** Volume BBCA.JK

| Models             | RMSE         | MAE          | MAPE  |
|--------------------|--------------|--------------|-------|
| ARIMA              | 52.662.514,0 | 27.808.104,2 | 36,9% |
| Vanilla LSTM       | 55.683.352,3 | 26.871.570,6 | 30,4% |
| Bidirectional LSTM | 53.678.671,9 | 26.862.275,7 | 34,4% |
| Stacked LSTM       | 54.245.496,5 | 28.869.947,1 | 40,7% |

Sebaliknya, pada Tabel 6. prediksi volume. Hasil menunjukkan tingkat kesalahan yang jauh lebih tinggi. Rata-rata MAPE adalah 35,6%, menandakan bahwa seluruh model memiliki kesulitan yang lebih besar dalam memodelkan volume dibandingkan dengan harga penutupan. Meskipun model Vanilla LSTM memiliki MAPE terendah (30,4%), hasil keseluruhan tetap menunjukkan variabilitas prediksi yang tinggi.

**Tabel 7.** Close BBRI.JK

| Models             | RMSE  | MAE   | MAPE |
|--------------------|-------|-------|------|
| ARIMA              | 90,7  | 70,0  | 1,3% |
| Vanilla LSTM       | 232,8 | 196,0 | 2,9% |
| Bidirectional LSTM | 224,7 | 189,8 | 1,9% |
| Stacked LSTM       | 300,8 | 247,3 | 2,5% |

Hasil evaluasi terhadap performa model dalam memprediksi harga penutupan (*close price*) saham BBRI.JK terlihat pada Tabel 7. menunjukkan bahwa model ARIMA memiliki keunggulan dalam hal akurasi, dengan nilai RMSE, MAE, dan MAPE terendah dibandingkan dengan model berbasis LSTM. Rata-rata MAPE untuk seluruh model adalah 2,15%, yang mengindikasikan bahwa model-model yang digunakan cukup andal dalam memodelkan harga penutupan saham ini.

**Tabel 8.** Volume BBRI.JK

| Models             | RMSE         | MAE          | MAPE  |
|--------------------|--------------|--------------|-------|
| ARIMA              | 98.598.402,6 | 68.995.756,3 | 34,0% |
| Vanilla LSTM       | 49.619.362,2 | 35.594.047,8 | 44,6% |
| Bidirectional LSTM | 53.481.919,9 | 27.424.206,4 | 36,4% |
| Stacked LSTM       | 54.349.059,5 | 28.913.150,8 | 40,7% |

Namun demikian, performa model dalam memprediksi volume transaksi pada Tabel. 8 menunjukkan *error* yang lebih besar. Rata-rata MAPE sebesar 38,9% mencerminkan tingginya tingkat kesalahan dalam prediksi volume, terutama pada model Vanilla LSTM yang mencatat nilai MAPE tertinggi (44,6%). Meskipun beberapa model LSTM menunjukkan nilai MAE yang lebih rendah dibanding ARIMA, hal ini tidak mencerminkan peningkatan akurasi secara keseluruhan karena nilai MAPE tetap tinggi.

**Tabel 9.** Close BMRI.JK

| Models             | RMSE  | MAE   | MAPE |
|--------------------|-------|-------|------|
| ARIMA              | 117,0 | 86,2  | 1,3% |
| Vanilla LSTM       | 197,0 | 156,8 | 3,0% |
| Bidirectional LSTM | 267,2 | 226,0 | 2,2% |
| Stacked LSTM       | 224,6 | 181,6 | 1,8% |

Berdasarkan Tabel. 9 hasil evaluasi terhadap performa model dalam memprediksi harga penutupan saham BMRI.JK, diketahui bahwa model ARIMA kembali menunjukkan performa terbaik dibandingkan ketiga varian LSTM, khususnya pada nilai RMSE dan MAE. Walaupun model Vanilla LSTM menghasilkan nilai MAPE tertinggi sebesar 3,0%, rata-rata MAPE keseluruhan masih tergolong rendah, yaitu 2,1% yang menandakan bahwa prediksi terhadap harga penutupan cukup akurat.

**Tabel 10.** Volume BMRI.JK

| Models             | RMSE          | MAE          | MAPE  |
|--------------------|---------------|--------------|-------|
| ARIMA              | 45.239.211,4  | 31.601.414,1 | 37,5% |
| Vanilla LSTM       | 111.939.015,1 | 73.898.773,5 | 33,3% |
| Bidirectional LSTM | 53.818.758,4  | 26.591.076,9 | 32,9% |
| Stacked LSTM       | 54.734.255,8  | 27.274.337,1 | 35,1% |

Di sisi lain, pada Tabel 10. prediksi volume transaksi menunjukkan tingkat kesalahan yang jauh lebih tinggi dibandingkan prediksi harga penutupan. Rata-rata nilai MAPE untuk seluruh model adalah 34,7%, dengan model ARIMA mencatat nilai tertinggi (37,5%) dan Bidirectional LSTM mencatat nilai terendah (32,9%). Namun demikian, Vanilla LSTM memiliki nilai RMSE tertinggi, nilai MAE dan MAPE-nya tetap kompetitif yang menandakan bahwa fluktuasi *error*-nya cukup besar namun selalu konsisten.

b. Evaluasi Model Rata-rata Perbankan

**Tabel 12.** Rata-rata Error Harga Penutupan Saham Bank

| Saham           | RMSE  | MAE   | MAPE |
|-----------------|-------|-------|------|
| BBCA.JK         | 196,0 | 158,9 | 1,6% |
| BBRI.JK         | 212,3 | 175,8 | 2,1% |
| BMRI.JK         | 201,5 | 162,7 | 2,1% |
| Rata-rata total | 203,3 | 165,8 | 1,9% |

Berdasarkan hasil penggabungan dari ketiga saham, diperoleh bahwa prediksi harga penutupan (*close price*) menghasilkan tingkat kesalahan yang relatif rendah, dengan rata-rata MAPE keseluruhan sebesar 1,9%. Di antara ketiganya, saham BBCA.JK memiliki performa prediksi terbaik, baik dari RMSE, MAE, dan MAPE. Hal ini menandakan kestabilan harga yang relatif tinggi dan pola yang lebih mudah ditangkap oleh model.

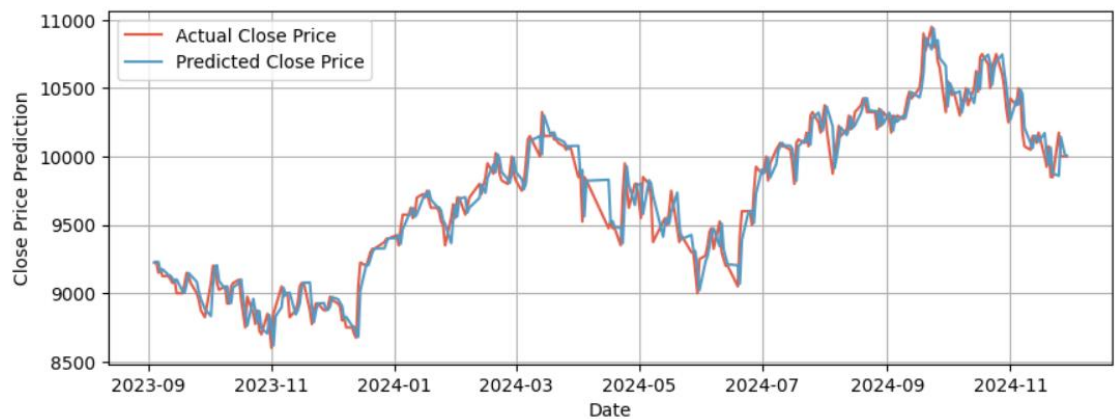
**Tabel 13.** Rata-rata Error Volume Saham Bank

| Saham           | RMSE         | MAE          | MAPE  |
|-----------------|--------------|--------------|-------|
| BBCA.JK         | 54.067.508,7 | 27.602.974,9 | 35,6% |
| BBRI.JK         | 64.512.686,1 | 40.731.790,3 | 38,9% |
| BMRI.JK         | 66.932.810,2 | 39.341.400,4 | 34,7% |
| Rata-rata total | 61.837.668,3 | 35.892.055,2 | 36,4% |

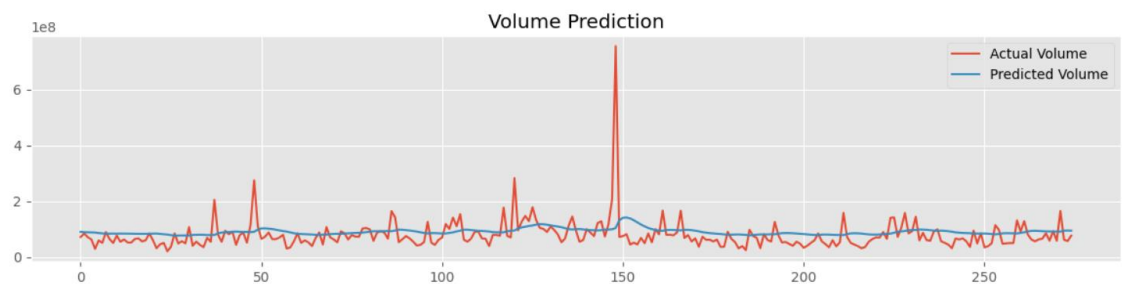
Sebaliknya, prediksi terhadap volume transaksi menunjukkan tantangan yang lebih besar, dengan rata-rata MAPE sebesar 36,4%. Saham BBRI.JK mencatat MAPE tertinggi untuk volume, yaitu 38,9% yang mengindikasikan tingkat fluktuasi volume yang lebih besar dan kemungkinan pengaruh eksternal yang lebih kuat. Sementara BMRI.JK mencatat nilai MAPE volume terendah, yaitu 34,7% namun demikian nilai RMSE-nya tertinggi.

c. Evaluasi Model Terbaik per-Emiten

Berdasarkan hasil evaluasi terhadap ketiga model *deep learning* (Vanilla LSTM, Bidirectional LSTM, dan Stacked LSTM) serta basis model ARIMA pada masing-masing saham perbankan (BBCA.JK, BBRI.JK, dan BMRI.JK), dilakukan analisis performa prediksi terhadap dua fitur utama harga penutupan (*close*) dan volume perdagangan. Evaluasi ini menggunakan tiga metrik umum yang merepresentasikan tingkat akurasi model secara kuantitatif, yaitu dengan mengacu pada metrik *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), dan *Mean Absolute Percentage Error* (MAPE). Setiap model diterapkan pada data historis untuk masing-masing emiten, lalu dibandingkan berdasarkan kinerja prediksi terhadap fitur harga penutupan (*close*) dan volume perdagangan. Tujuan dari evaluasi ini adalah untuk mengidentifikasi model mana yang paling efektif dalam menangkap pola harga penutupan (*close*) dan volume perdagangan, serta menentukan model terbaik secara spesifik untuk tiap emiten dan tiap fitur. Hasil analisis kemudian divisualisasikan dalam bentuk grafik untuk memperkuat interpretasi dan memberikan gambaran intuitif mengenai akurasi pada masing-masing model.

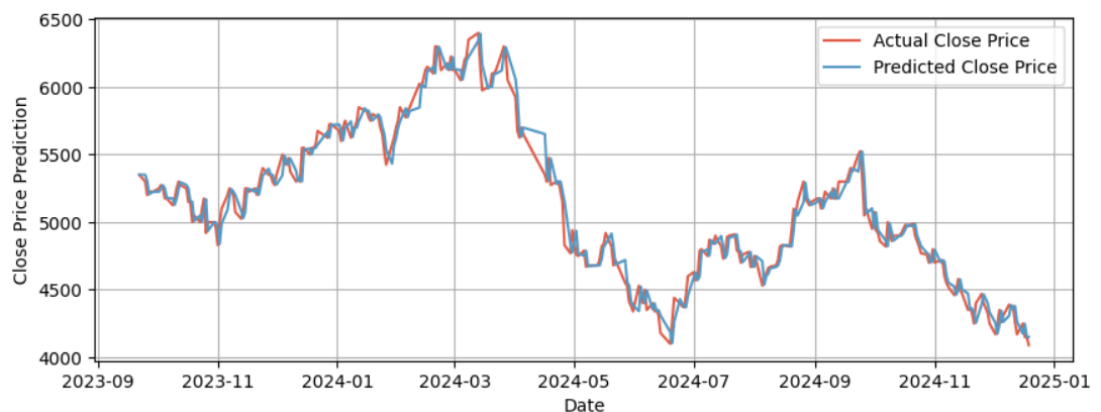


**Gambar 3.** Visualisasi Prediksi Close Emiten BBKA.JK Model ARIMA

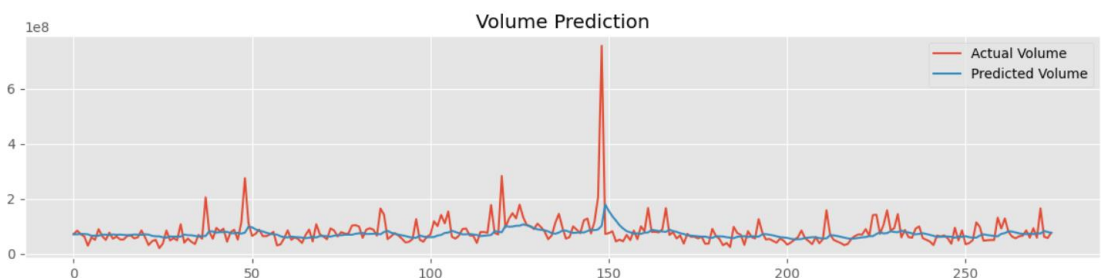


**Gambar 4.** Visualisasi Prediksi Volume Emiten BBKA.Jk Model Vanilla LSTM

Gambar 3 emiten BBKA.JK fitur *close*, model ARIMA menunjukkan performa terbaik dengan nilai RMSE sebesar 124,1, MAE sebesar 95,0, dan MAPE sebesar 0,9%, mengungguli seluruh varian LSTM. Sementara itu, pada Gambar 4. fitur volume model Vanilla LSTM memberikan hasil yang paling optimal dengan MAPE sebesar 30,4%, namun demikian ARIMA memiliki RMSE dan MAE yang sedikit lebih baik. Hal ini mengindikasikan bahwa model LSTM lebih adaptif terhadap fluktuasi volume perdagangan.



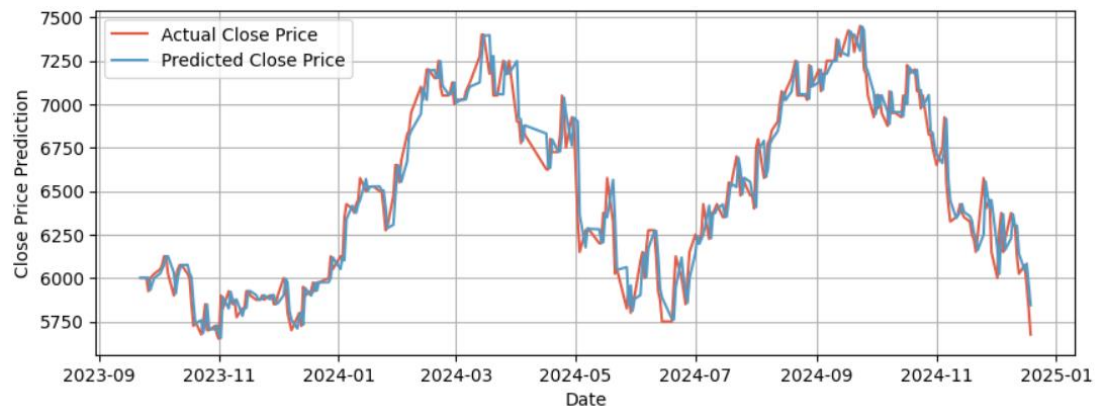
**Gambar 5.** Visualisasi Prediksi Close Emiten BBRI.JK Model ARIMA



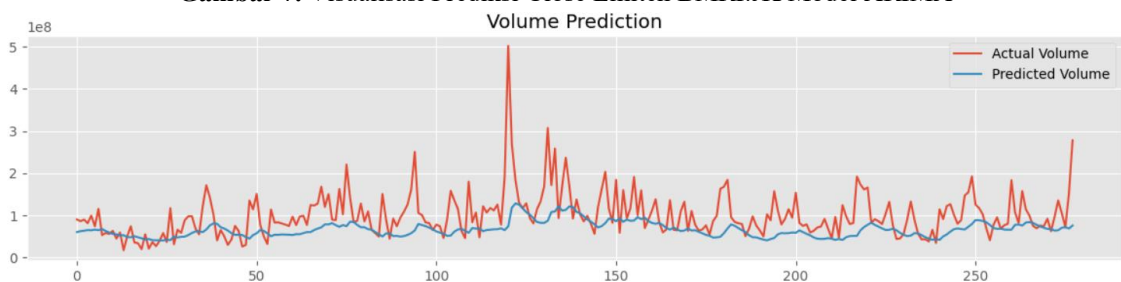
**Gambar 6.** Visualisasi Predksi Volume Emiten BBRI.JK Model Bidirectional LSTM

Untuk Gambar 5 emiten BBRI.JK fitur *close*, kembali model ARIMA mencatatkan hasil prediksi terbaik dengan MAPE sebesar 1,3%, mengungguli semua model LSTM yang cenderung memiliki error lebih besar. Sedangkan pada Gambar 6. fitur volume, performa terbaik ditunjukkan oleh Bidirectional LSTM dengan nilai MAPE sebesar

36,4% dan MAE terendah di antara model lainnya, ini menunjukkan bahwa arsitektur dua arah mampu menangkap pola volume lebih efisien.



**Gambar 7.** Visualisasi Prediksi Close Emiten BMRI.JK Model ARIMA



**Gambar 8.** Visualisasi Prediksi Volume Emiten BMRI.JK Model Bidirectional LSTM

Pada Gambar 7 prediksi *close*, model ARIMA masih mempertahankan keunggulan dengan nilai MAPE sebesar 1,3%. Hal ini memperkuat pola bahwa model statistik konvensional tetap kompetitif untuk prediksi harga penutupan yang cenderung stasioner. Akan tetapi, berbeda dari dua saham sebelumnya pada Gambar 8, prediksi volume pada BMRI.JK menunjukkan bahwa Bidirectional LSTM kembali unggul dengan nilai MAPE terendah yaitu 32,9%, mengindikasikan arsitektur ini cukup efektif menangani pola volume yang kompleks dan volatil.

Hasil evaluasi menunjukkan bahwa model ARIMA secara konsisten memberikan performa terbaik dalam memprediksi harga penutupan (*close*) pada ketiga saham bank, dengan nilai error yang lebih kecil dibandingkan model LSTM. Sebaliknya, untuk prediksi volume yang secara umum lebih volatil dan non-linear, model Bidirectional LSTM cenderung memberikan hasil yang lebih akurat pada dua dari tiga emiten. Disisi lain, Vanilla LSTM unggul pada satu kasus (BBCA.JK).

## 4. KESIMPULAN

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja model statistik tradisional ARIMA dengan tiga varian model Long Short-Term Memory (LSTM), yaitu Vanilla LSTM, Bidirectional LSTM, dan Stacked LSTM, dalam melakukan peramalan harga penutupan (*close price*) dan volume perdagangan saham tiga bank besar di Indonesia, yaitu BBKA.JK, BBRI.JK, dan BMRI.JK. Berdasarkan hasil evaluasi, diperoleh bahwa model ARIMA secara konsisten menunjukkan performa terbaik dalam memprediksi harga penutupan saham, yang ditunjukkan oleh nilai RMSE, MAE, dan MAPE yang paling rendah. Hal ini mengindikasikan bahwa model statistik linier lebih efektif dalam menangkap pola pergerakan yang relatif stasioner. Sebaliknya, untuk prediksi volume transaksi yang memiliki karakteristik data yang lebih fluktuatif dan kompleks, model Bidirectional LSTM menunjukkan hasil yang lebih unggul, khususnya pada saham BBRI dan BMRI, sementara Vanilla LSTM memberikan hasil terbaik pada saham BCA. Meskipun demikian, secara keseluruhan tingkat kesalahan pada prediksi volume masih tergolong tinggi (rata-rata MAPE sebesar 36,4%), yang menunjukkan bahwa perlu adanya eksplorasi lebih lanjut terhadap pemilihan fitur dan strategi regulasi model. Dengan demikian, pemilihan model peramalan perlu mempertimbangkan karakteristik data target, di mana ARIMA lebih sesuai untuk data yang stabil dan linier, sementara model *deep learning* seperti LSTM lebih adaptif terhadap pola non-linier dan volatil. Penelitian ini memberikan kontribusi signifikan dalam pemilihan model peramalan berbasis data historis saham dan membuka peluang pengembangan lebih lanjut melalui integrasi fitur eksternal serta tuning *hyperparameter* yang lebih mendalam.

## REFERENCES

- [1] Furizal, A. Ritonga, A. Ma'arif, and I. Suwarno, "Stock Price Forecasting with Multivariate Time Series Long Short-Term Memory: A Deep Learning Approach," *J. Robot. Control*, vol. 5, no. 5, pp. 1322–1335, 2024, doi: 10.18196/jrc.v5i5.22460.



- [2] K. Attaluri, M. K. Tripathi, S. Reddy, and Shivendra, “News-Driven Stock Price Forecasting in Indian Markets: A Comparative Study of Advanced Deep Learning Models,” *2024 3rd Int. Conf. Adv. Technol. ICONAT 2024*, p. 7, 2024, doi: 10.1109/ICONAT61936.2024.10774891.
- [3] D. Kobiela, D. Krefta, W. Król, and P. Weichbroth, “ARIMA vs LSTM on NASDAQ stock exchange data,” *Procedia Comput. Sci.*, vol. 207, no. Kes, pp. 3830–3839, 2022, doi: 10.1016/j.procs.2022.09.445.
- [4] C. R. Ko and H. T. Chang, “LSTM-based sentiment analysis for stock price forecast,” *PeerJ Comput. Sci.*, vol. 7, pp. 1–23, 2021, doi: 10.7717/peerj-cs.408.
- [5] Kenrick, S. C. Ivancio, E. Isabellini, C. G. S. Adhi, and A. Prasetyo, “Multivariate stock market forecasting using lstm on Indonesian banking stock market,” *Procedia Comput. Sci.*, vol. 245, no. C, pp. 1047–1056, 2024, doi: 10.1016/j.procs.2024.10.333.
- [6] Z. Zhang, *Comparison of LSTM and ARIMA in Price Forecasting: Evidence from Five Indexes*, no. Esfct 2023. Atlantis Press International BV, 2024. doi: 10.2991/978-94-6463-268-2\_6.
- [7] M. a, U. Marfuah, R. Nopianti, and A. Tri Panudju, “Lstm Algorithm Analysis of Banking Sector Stock Price Predictions,” *Int. J. Adv. Res.*, vol. 10, no. 01, pp. 627–634, 2022, doi: 10.21474/ijar01/14082.
- [8] Preeti, R. Bala, and R. P. Singh, “A dual-stage advanced deep learning algorithm for long-term and long-sequence prediction for multivariate financial time series,” *Appl. Soft Comput.*, vol. 126, p. 109317, 2022, doi: 10.1016/j.asoc.2022.109317.
- [9] C. M. Liapis, A. Karanikola, and S. Kotsiantis, “A multi-method survey on the use of sentiment analysis in multivariate financial time series forecasting,” *Entropy*, vol. 23, no. 12, 2021, doi: 10.3390/e23121603.
- [10] P. Ray, B. Ganguli, and A. Chakrabarti, “Multivariate Bayesian Time-Series Model with Multi-temporal Convolution Network for Forecasting Stock Market During COVID-19 Pandemic,” *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, 2024, doi: 10.1007/s44196-024-00525-5.
- [11] M. Jarrah and M. Derbali, “Predicting Saudi Stock Market Index by Using Multivariate Time Series Based on Deep Learning,” *Appl. Sci.*, vol. 13, no. 14, 2023, doi: 10.3390/app13148356.
- [12] J. Zhang, H. Liu, W. Bai, and X. Li, “A hybrid approach of wavelet transform, ARIMA and LSTM model for the share price index futures forecasting,” *North Am. J. Econ. Financ.*, vol. 69, no. PB, p. 102022, 2024, doi: 10.1016/j.najef.2023.102022.
- [13] T. Sharif, A. Bouteska, M. Zoynul Abedin, and S. Cotturone, “An enquiry into the monetary policy and stock market shocks in the US,” *Int. Rev. Econ. Financ.*, vol. 98, p. 103925, Mar. 2025, doi: 10.1016/j.iref.2025.103925.
- [14] Y. Ji, A. W.-C. Liew, and L. Yang, “A Novel Improved Particle Swarm Optimization With Long-Short Term Memory Hybrid Model for Stock Indices Forecast,” *IEEE Access*, vol. 9, pp. 23660–23671, 2021, doi: 10.1109/ACCESS.2021.3056713.
- [15] A. Omar and T. Abd El-Hafeez, “Optimizing epileptic seizure recognition performance with feature scaling and dropout layers,” *Neural Comput. Appl.*, vol. 36, no. 6, pp. 2835–2852, 2024, doi: 10.1007/s00521-023-09204-6.
- [16] Jason Brownlee, *Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future*. 2017. [Online]. Available: [https://www.google.co.id/books/edition/Introduction\\_to\\_Time\\_Series\\_Forecasting/-AiqDwAAQBAJ?hl=id&gbpv=0](https://www.google.co.id/books/edition/Introduction_to_Time_Series_Forecasting/-AiqDwAAQBAJ?hl=id&gbpv=0)
- [17] A. Avinash, A. Widjaja, and O. Karnalim, “Analisis Perbandingan Algoritma Machine Learning untuk Forecasting Persediaan Produk Barang Pokok,” *J. Tek. Inform. dan Sist. Inf.*, vol. 10, no. 2, pp. 361–378, 2024, doi: 10.28932/jutisi.v10i2.9357.
- [18] J. Korstanje, *Advanced Forecasting with Python*. Berkeley, CA: Apress, 2021. doi: 10.1007/978-1-4842-7150-6.
- [19] Yudi Widhiyasana, Transmissia Semiawan, Ilham Gibran Achmad Mudzakir, and Muhammad Randi Noor, “Penerapan Convolutional Long Short-Term Memory untuk Klasifikasi Teks Berita Bahasa Indonesia,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 10, no. 4, pp. 354–361, 2021, doi: 10.22146/jnteti.v10i4.2438.
- [20] A. Mire, S. Malik, and A. K. Tyagi, Eds., *Advanced Analytics and Deep Learning Models*. Wiley, pp. 1-412, 2022. doi: 10.1002/9781119792437.