

BAB 2

LANDASAN TEORI

Penelitian ini menerapkan 2 model dan akan dilakukan evaluasi berupa perbandingan terkait performa prediksi arah gerak saham BCA sebagai objek penelitian.

2.1 *Autoregressive Integrated Moving Average (ARIMA)*

ARIMA dipakai guna memproses data non-stasioner karena memiliki komponen *Integrated (I)* yang melakukan proses *differencing* agar data menjadi stasioner sehingga efektif dalam memproses pola linear dan tren pada data finansial [12], [13]. Adapun bentuk dari model dapat dijabarkan sebagai berikut:

$$X_t = c + \varepsilon_t + \sum_{i=1}^p \varphi_i X_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \delta t \quad (2.1)$$

dengan:

- X_t : nilai variabel deret waktu harga pada periode ke- t .
- c : konstanta atau *intercept*, yang merepresentasikan nilai dasar dari model.
- ε_t : *error term (white noise)*, yaitu komponen acak yang tidak dapat diprediksi.
- φ_i : koefisien *autoregressive (AR)* yang berarti efek nilai lampau pada nilai sekarang.
- θ_i : koefisien MA.
- δt : komponen tren deterministik, yang menangkap kecenderungan naik atau turun secara sistematis.

2.2 *Long-short Term Memory (LSTM)*

Vanishing gradient dapat diatasi dengan LSTM yang memiliki gerbang *input*, *output* dan *forget*. Dengan struktur ini, LSTM dapat dengan jangka panjang menahan informasi penting sekaligus membuang informasi yang tidak relevan. Mekanisme gate pada LSTM memudahkan model untuk menyesuaikan bobot

informasi tersimpan dan terlupakan [10], [11], [14]. Proses komputasi dalam LSTM dapat dituliskan sebagai:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2.2)$$

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (2.3)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2.4)$$

$$C_t = i_t \cdot \tilde{C}_t + f_t \cdot C_{t-1} \quad (2.5)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o C_t + b_o) \quad (2.6)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (2.7)$$

dengan:

- x_t : *Input* pada waktu t
- h_{t-1} : *Hidden state* dari waktu sebelumnya
- i_t : *Input gate* yakni memproses informasi yang masuk *cell state*)
- f_t : *Forget gate* (mengatur informasi lama yang dibuang)
- o_t : *Output gate* (mengatur informasi yang keluar sebagai *hidden state*)
- C_t : *Cell state* (memori internal LSTM)
- $\tilde{C}_t$: Kandidat *cell state* baru
- W_i, W_c, W_f, W_o : Bobot untuk input x_t

- U_i, U_c, U_f, U_o : Bobot untuk hidden state h_{t-1}
- V_o : Bobot tambahan untuk *cell state* C_t pada output gate
- b_i, b_c, b_f, b_o : Bias masing-masing *gate*
- σ : Fungsi sigmoid
- $\tanh$: Fungsi aktivasi hiperbolik

2.3 Min-Max Scaler

Pengubahan nilai variabel dilakukan ke dalam rentang 0 hingga 1 agar setiap variabel berada pada skala yang sebanding, sehingga algoritma tidak memberikan perlakuan lebih terhadap variabel dengan nilai yang besar [15].

Rumus umum Min-Max Scaler dapat dituliskan sebagai berikut:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (2.8)$$

dengan:

- x : nilai asli variabel
- $x_{\min}$: nilai minimum variabel
- $x_{\max}$: nilai maksimum variabel
- x' : nilai hasil normalisasi dalam rentang $[0,1]$

Jika diperlukan, rentang normalisasi dapat diubah menjadi $[a, b]$ dengan rumus:

$$x' = a + \frac{(x - x_{\min})(b - a)}{x_{\max} - x_{\min}} \quad (2.9)$$

Normalisasi dengan Min-Max Scaler menjaga hubungan proporsional antar data, sehingga pola tren tetap terjaga meskipun skala data diubah.

2.4 Augmented Dickey-Fuller Test (ADF Test)

Merupakan salah satu bentuk *Unit Root Test* yang merupakan pengujian apakah *time series* bersifat stasioner atau non-stasioner [16]. *ADF test* menguji keberadaan *unit root* dalam suatu deret waktu. Uji ini dikembangkan dari Dickey-Fuller test untuk menangani kemungkinan adanya autokorelasi pada residual dengan menambahkan lag dari variabel dependen [16], [17]. Hipotesisnya sebagai berikut:

- H_0 : Deret waktu memiliki *unit root* (tidak stasioner).
- H_1 : Deret waktu tidak memiliki *unit root* (stasioner).

Adapun bentuk umum persamaan ADF dapat dituliskan sebagai berikut:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^p \delta_i \Delta y_{t-i} + \varepsilon_t \quad (2.10)$$

dengan:

- y_t : nilai deret waktu pada periode t
- Δy_t : *differencing* pertama dari y_t
- α : konstanta
- βt : komponen tren deterministik
- γ : parameter yang diuji untuk menentukan keberadaan *unit root*
- δ_i : koefisien lag dari *differencing*
- ε_t : *error term* (*white noise*)

Penolakan hipotesis nol dilakukan bila nilai uji tidak lebih dibanding *critical value* di signifikansi tertentu.

Jika nilai uji ADF lebih kecil dari *critical value* pada tingkat signifikansi tertentu, maka *null hypothesis* ditolak dan deret waktu dianggap stasioner.

2.5 Differencing

Differencing mengatasi masalah ketidakstasioneran. Proses ini menghitung perbedaan nilai saat ini dengan sebelumnya sehingga tren atau pola musiman dapat dihilangkan sehingga data menjadi stasioner dan sesuai untuk dimodelkan menggunakan ARIMA [6].

Bentuk umum differencing pertama dapat dituliskan sebagai berikut:

$$\Delta y_t = y_t - y_{t-1} \quad (2.11)$$

dengan:

- y_t : nilai deret waktu pada periode t
- y_{t-1} : nilai deret waktu pada periode sebelumnya
- Δy_t : hasil differencing pertama

Differencing dilakukan lagi jika tidak stasioner:

$$\Delta^2 y_t = \Delta y_t - \Delta y_{t-1} \quad (2.12)$$

Proses differencing ini akan terus dilakukan hingga data mencapai kondisi stasioner, yang ditandai dengan mean dan varians yang relatif konstan sepanjang waktu.

2.6 Autocorrelation Function (ACF) dan Partial Autocorrelation Function (PACF)

ACF digunakan untuk mengidentifikasi pola keterkaitan antar periode, misalnya apakah nilai sekarang terpengaruh nilai sebelumnya [18].

Di bawah merupakan ACF ke k :

$$r_k = \frac{\sum_{t=k+1}^T (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^T (y_t - \bar{y})^2} \quad (2.13)$$

dengan r_k adalah nilai autokorelasi ke- k , y_t adalah *time series value* pada periode t , dan $\bar{y}$ *average time series*.

Interpretasi ACF adalah sebagai berikut:

- Jika ACF signifikan pada lag tertentu, maka terdapat pola autoregresif atau *moving average*.
- Grafik ACF digunakan untuk menentukan komponen *Moving Average* (MA) dalam model ARIMA.

Partial Autocorrelation Function (PACF) digunakan untuk mengukur hubungan suatu *time series* dengan lag tertentu setelah pengaruh dari lag-lag lain dihilangkan. Dengan demikian, PACF berfungsi untuk mengidentifikasi hubungan langsung antar periode tanpa dipengaruhi oleh keterkaitan lag lainnya [19].

Dalam metodologi Box-Jenkins, ACF dan PACF digunakan pada tahap identifikasi model:

- ACF untuk orde q .
- PACF untuk orde p .

Kombinasi keduanya membantu peneliti dalam merumuskan model ARIMA yang sesuai dengan pola data [20].

2.7 Maximum Likelihood Estimation (MLE)

Fungsi likelihood dimaksimalkan dalam MLE. Fungsi likelihood sendiri menggambarkan probabilitas munculnya data yang diamati berdasarkan parameter tertentu. Inti dari pendekatan MLE adalah memilih parameter yang membuat data aktual memiliki kemungkinan terbesar untuk terjadi [21].

Secara umum, fungsi *likelihood* dapat dituliskan sebagai:

$$L(\theta) = \prod_{t=1}^n f(y_t | \theta) \quad (2.14)$$

dengan:

- y_t : data observasi pada waktu t
- θ : vektor parameter model
- $f(y_t | \theta)$: fungsi probabilitas dari data y_t dengan parameter θ

Estimasi parameter diperoleh dengan mencari nilai θ yang memaksimalkan $L(\theta)$ atau *log-likelihood*:

$$\hat{\theta} = \arg \max_{\theta} \ln L(\theta) \quad (2.15)$$

Dalam konteks ARIMA, MLE digunakan untuk mengestimasi parameter *autoregressive* (ϕ_i), *moving average* (θ_i), serta konstanta (c), sehingga model dapat merepresentasikan pola data historis secara optimal.

2.8 Root Mean Squared Error (RMSE)

RMSE menilai tingkat kesalahan prediksi dan mengukur akar dari *average* kuadrat perbedaan aktual dan prediksi. Indikator ini memberikan gambaran mengenai seberapa besar deviasi hasil prediksi terhadap data sebenarnya, sekaligus memberikan penalti lebih besar pada kesalahan dengan nilai tinggi akibat proses pengkuadratan. Rumus RMSE dapat dituliskan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (2.16)$$

dengan:

- y_t : nilai aktual pada t
- $\hat{y}_t$: prediksi pada waktu t
- n : nilai pengamatan

2.9 Mean Absolute Error (MAE)

MAE menilai tingkat kesalahan ramalan dengan mengukur *average* dari selisih antara data aktual dan *predicted* dan menunjukkan besar error data [22].

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \quad (2.17)$$

2.10 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) merupakan salah satu Metrik ini menyajikan nilai kesalahan (error) dalam bentuk persentase yang mudah diinterpretasikan, sehingga sangat membantu ketika melakukan perbandingan performa model di berbagai himpunan data (*datasets*) yang berbeda skala. Nilai persentase yang lebih rendah menunjukkan bahwa model memiliki tingkat akurasi yang lebih baik. Secara matematis, formula untuk menghitung MAPE dapat dinyatakan seperti: [22].

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (2.18)$$

Dimana n merepresentasikan jumlah sampel data, y_i merupakan nilai aktual ke- i , dan $\hat{y}_i$ menyatakan nilai hasil prediksi atau peramalan ke- i .

2.11 Cross Industry Standard Process for Data Mining (CRISP-DM)

Cross Industry Standard Process for Data Mining (CRISP-DM) bersifat fleksibel. Kerangka ini terdiri dari tahapan-tahapan berikut [23]:

1. *Business Understanding*,
2. *Data Understanding*,
3. *Data Preparation*,
4. *Modeling*,
5. *Evaluation*,
6. *Deployment*.

Keunggulan CRISP-DM adalah sifatnya yang industri-independen dan dapat diterapkan pada berbagai domain, termasuk analisis finansial dan peramalan harga saham, memastikan proses peramalan harga saham BBKA dengan ARIMA dan LSTM dilakukan secara terstruktur.

2.12 *Exploratory Data Analysis (EDA)*

EDA berarti pendekatan sistematis untuk mendapatkan informasi terkait karakteristik data eksplorasi data dengan deskripsi statistik, visualisasi, dan identifikasi pola yang mendasari. Ini membantu mengungkap informasi dan karakteristik data seperti outlier, korelasi dalam dataset, dan tren [24]. EDA sangat penting dalam mengembangkan model peramalan deret waktu karena menunjukkan dan mengeksplorasi hubungan antara variabel input dan output. Oleh karena itu, EDA harus dilakukan sebelum pemodelan. Beberapa metodologi visual statistik dapat digunakan untuk tujuan ini, seperti plot autokorelasi, *box plot* dan lainnya. [?].

2.13 **Interquartile Range (IQR)**

Interquartile Range (IQR) dapat digunakan untuk keberadaan (*outlier*). IQR mengisolasi 50% populasi data bagian tengah sehingga batas penalti tidak mudah terdistorsi oleh nilai ekstrem. Adapun bentuk penentuan jangkauan antar kuartil dan ambang batas pagar (*fences*) pencilan dirumuskan sebagai berikut [25]:

$$IQR = Q_3 - Q_1 \quad (2.19)$$

$$\text{Batas Bawah} = Q_1 - (1,5 \times IQR) \quad (2.20)$$

$$\text{Batas Atas} = Q_3 + (1,5 \times IQR) \quad (2.21)$$

Dimana Q_1 merupakan nilai kuartil pertama (25) dan Q_3 merupakan nilai kuartil ketiga (75) dari distribusi data. Setiap baris observasi yang memiliki nilai di luar batas atas maupun batas bawah secara otomatis diklasifikasikan sebagai *outlier*.