

**PREDIKSI HARGA JUAL KAKAO DENGAN METODE
LONG SHORT-TERM MEMORY MENGGUNAKAN METODE OPTIMASI ROOT
MEAN SQUARE PROPAGATION DAN ADAPTIVE MOMENT ESTIMATION
DILENGKAPI GUI RSHINY**

Yayan Setiawan¹, Tarno^{2*}, Puspita Kartikasari³

^{1,2,3} Departemen Statistika, Fakultas Sains dan Matematika, Universitas Diponegoro

*email: tarno.stat@gmail.com

ABSTRACT

Cocoa is a leading commodity from Indonesia. Cocoa prices from time to time fluctuate. Accurate Cocoa price predictions are very important to ensure future prices and help decision making. Cocoa price data is non-stationary and nonlinear, so to make accurate predictions, an Artificial Neural Network (ANN) model is applied. One type of ANN is Long Short-Term Memory (LSTM). LSTM has superior performance for time series based prediction. Optimization methods used are Root Mean Square Propagation, and Adaptive Moment Estimation. The best model was selected based on the Means Square Error (MSE) and Mean Absolute Percentage Error (MAPE) values. This study uses the R-Shiny GUI to facilitate the use of LSTM for users who are less proficient in programming languages. Based on the results, the Long Short-Term Memory model with the Adaptive Moment Estimation optimization method is more optimal than the Long Short-Term Memory with Root Mean Square Propagation seen from the smaller MSE and MAPE values. This study used 27 combinations of hyperparameters. Prediction results with LSTM using the R-Shiny GUI have different levels of accuracy in each experiment. The best accuracy value is experiment with MSE value of 491505.1 and MAPE value of 1.739155% . Cocoa Price Forecasting for the period November to December 2021 tends to decline.

Keywords : Cocoa Prices, Forecasting, Long Short-Term Memory, Root Mean Square Propagation, Adaptive Moment Estimation, GUI R-Shiny

1. PENDAHULUAN

Sejak tahun 1930 kakao merupakan salah satu komoditas perkebunan yang mempunyai peran penting dalam perkembangan perekonomian Indonesia. Indonesia merupakan pengekspor biji kakao terbesar ketiga dunia, hal ini mengindikasikan peran penting kakao baik sumber lapangan kerja maupun pendapatan bagi petani dan penghasil devisa bagi negara(Nur, 2021). Harga kakao sering mengalami fluktuasi. Harga historis kakao dari tahun 2009 hingga Desember 2020 terus mengalami perubahan yang signifikan. Harga terendah yaitu pada angka 1.813 di kuartal 4 tahun 2012 tepatnya pada tanggal 31 Oktober 2012. Sementara harga tertinggi dicapai pada kuartal 2 tahun 2015 tepatnya pada tanggal 8 Juni 2015 yaitu sebesar 44.885. Selama dua tahun terakhir harga kakao terus mengalami kenaikan dan penurunan yang signifikan(Bappebti, 2021). Fluktuasi harga kakao dapat diprakirakan dengan peramalan harga kakao. Peramalan harga kakao yang akurat penting guna memastikan harga kakao dimasa depan, selain itu juga untuk membantu pengambilan keputusan. Oleh karena itu sangat penting untuk mengkaji harga kakao dari periode waktu ke waktu, sehingga fluktuasi harga dapat diantisipasi dengan baik. Ketidakpastian harga kakao merupakan data non stasioner dan non linear. Maka digunakan metode *Artificial Neural Network* (ANN) pada peramalan harga kakao.

Salah satu bagian dari ANN adalah *Recurrent Neural Networks* (RNN). RNN dirancang untuk mengenali pola sebagai urutan data dan berguna dalam prediksi dan prakiraan. RNN dapat mengerjakan data teks, gambar, ucapan, dan deret waktu. RNN adalah salah satu ANN

yang kuat dan mewakili otak biologis. RNN mengambil *input* dari *input* saat ini dan *output* yang telah dihitung sebelumnya. RNN memiliki beberapa varian, salah satu diantaranya adalah *Long Short-Term Memory* (LSTM) (Ciaburro & Venkateswaran, 2017).

LSTM memiliki formulasi unik yang memungkinkannya menghindari masalah yang mencegah pelatihan dan penskalaan RNN lainnya. Tantangan historis teknis utama yang dihadapi RNN adalah cara melatihnya secara efektif. Prosedur pembaruan bobot yang sulit pada *vanilla* RNN mengakibatkan perubahan bobot yang dengan cepat menjadi sangat kecil sehingga tidak berpengaruh (*vanishing gradients*) atau sangat besar sehingga menghasilkan perubahan yang sangat besar (*exploding gradients*). Desain dari LSTM dapat mengatasi masalah yang terdapat dalam *vanilla* RNN. *Memory cell* dari LSTM terdiri dari *weights* dan *gates*. Keunggulan lain dari LSTM adalah memiliki memori untuk mengatasi masalah ketergantungan temporal jangka panjang dengan masukan *sequences*, juga memproses *input* dan *output* tiap langkah waktu, memungkinkan input dan output yang panjang. Beberapa aplikasi dari LSTM yaitu *Automatic Image Caption Generation*, *Automatic Translation of Text*, *Automatic Handwriting Generation*, dan *Stock Market Prediction* (Brownlee, 2017).

Untuk meningkatkan tingkat akurasi LSTM, digunakan *optimizer*. *Optimizer* berfungsi untuk menyesuaikan bobot model selama proses pelatihan. *Optimizer* dapat membentuk model ke dalam bentuk yang paling akurat dengan cara menyesuaikan bobot yang didasarkan pada nilai *loss function*. Penelitian ini menggunakan metode optimasi RMSProp dan metode optimasi Adam karena keduanya memiliki kelebihan dibandingkan metode optimasi lain. Metode optimasi Adam melambat saat konvergen ke lokal minimal, Adam berperilaku seperti bola berat dengan gesekan, yang dengan demikian lebih menyukai minimal datar di permukaan kesalahan, atau dengan kata lain lebih cepat untuk melakukan pembelajaran. Sementara RMSProp didefinisikan secara rekursif (perulangan) sebagai rata-rata penurunan dari semua gradien kuadrat masa lalu, selain itu RMSProp juga menjaga tingkat pembelajaran tetap tinggi secara optimal.

Penelitian ini mengaplikasikan *Long Short-Term Memory* (LSTM) pada peramalan data harga kakao. Akan tetapi kendala dari LSTM adalah penggunaan program yang susah bagi pengguna yang tidak mengerti program. Beberapa program yang biasa digunakan untuk menganalisis LSTM adalah *Python* dan *R*. Penelitian ini menggunakan program Rstudio yang merupakan salah satu program pengembangan dari *R* untuk memodelkan *Long Short-Term Memory* (LSTM) dengan paket *keras*. Namun, dalam penggunaannya paket tersebut susah bagi pengguna yang tidak mengerti program terutama program *R*. Oleh karena itu dibuat *user interface* yang dapat mempermudah penggunaan program *R*. Salah satu program dalam *R* yang bisa membuat menu *user interface* adalah *R-Shiny*. *R-Shiny* merupakan *toolkit* dari program *R* yang dapat digunakan untuk membuat program seperti GUI. Penelitian ini akan dibuat *interface* dengan *R-Shiny* untuk melakukan peramalan data harga kakao. Hal ini dimaksudkan untuk mempermudah seseorang yang kurang mengerti pemrograman menjadi lebih mudah untuk melakukan peramalan data dengan *Long Short-Term Memory* (LSTM) utamanya pada harga kakao.

2. TINJAUAN PUSTAKA

2.1. Analisis Runtun Waktu

Data runtun waktu (*time series*) adalah serangkaian nilai-nilai variabel yang disusun berdasarkan waktu (Qiu et al., 2020). Analisis runtun waktu mempelajari pola gerakan nilai-nilai variabel pada satu interval waktu seperti harian, mingguan, bulanan, tahunan yang teratur. Metode *time series* didasarkan pada asumsi bahwa pola lama akan terulang. Manfaat

dari analisis time series dapat diperoleh ukuran-ukuran yang dapat digunakan untuk membuat keputusan pada saat ini, untuk prediksi dan untuk merencanakan masa depan.

2.2 Pre-Processing Data dan Post-Processing Data

Pre-processing pada data diperlukan agar algoritma machine learning berjalan dengan baik. Proses pembelajaran machine learning tidak berfungsi dengan baik saat atribut numerik masukan memiliki skala yang sangat berbeda. Terdapat dua cara yang umum agar semua atribut memiliki skala yang sama, yaitu menggunakan min-max scaling dan standardization. Min-max scaling atau normalisasi yaitu nilai digeser dan diskalakan ulang sehingga berada dalam range 0 sampai 1. Normalisasi dilakukan dengan mengurangi nilai minimum dan membaginya dengan maksimum dikurangi minimum (Géron, 2019). Persamaan untuk melakukan normalisasi sebagai berikut:

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

Keterangan:

\mathbf{x} : Data input aktual

\mathbf{x}' : Data input setelah dinormalisasi

$\min(\mathbf{x})$: Data input dengan nilai minimum

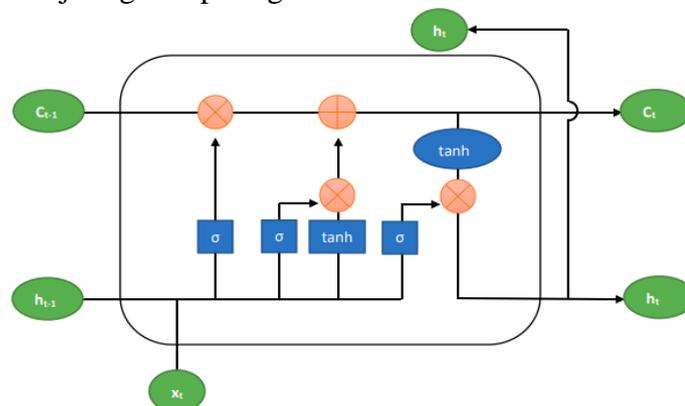
$\max(\mathbf{x})$: Data input dengan nilai maksimum

Setelah proses pelatihan dan didapatkan hasil prediksi menggunakan model yang dibangun, dilakukan post-processing, yaitu pengembalian data ke bentuk atau nilai aslinya. Post-processing yang dilakukan adalah denormalisasi. Denormalisasi menggunakan nilai scaler yang sama dengan proses normalisasi. Persamaan untuk melakukan denormalisasi sebagai berikut:

$$\mathbf{x} = \mathbf{x}' (\max(\mathbf{x}) - \min(\mathbf{x})) + \min(\mathbf{x}) \quad (2)$$

2.3 Long Short-Term Memory (LSTM)

LSTM merupakan salah satu algoritma pengembangan dari *recurrent neural network* (RNN) dengan mengatasi salah satu kekurangan RNN yaitu pengelolaan informasi dalam periode yang lama. Diusulkan oleh Sepp dan Jurgan Schmidhuber pada 1997. LSTM banyak dipilih untuk prediksi berbasis waktu atau *time series* karena dikenal lebih unggul dan handal dalam melakukan prediksi dalam waktu lama dibanding algoritma lain (Zahara et al., 2017). LSTM memiliki struktur jaringan seperti gambar dibawah ini:



Gambar 1. Struktur Jaringan LSTM

2.4 Fungsi Aktivasi

Fungsi aktivasi merupakan salah satu bagian penting dalam implementasi *neural network*. ANN memiliki keunggulan untuk menangani pola nonlinier pada data, disebabkan

oleh fungsi aktivasi pada ANN. Pada dasarnya fungsi aktivasi mentransformasikan nilai *input* untuk disalurkan menjadi *output* (Stewart, 2019). Fungsi aktivasi yang paling sederhana dalam ANN adalah fungsi aktivasi linear. Fungsi ini sebanding dengan input yang diberikan. Fungsi ini dapat digunakan saat mempertimbangkan menggunakan regresi, sebagai layer terakhir (Fabien, 2019).

2.5 Metode Optimasi pada LSTM

Untuk meningkatkan tingkat akurasi, digunakan *optimizer*. *Optimizer* berfungsi untuk menyesuaikan bobot model selama proses pelatihan. *Optimizer* dapat membentuk model ke dalam bentuk yang paling akurat dengan cara menyesuaikan bobot yang didasarkan pada nilai *loss function*. Salah satu algoritma optimasi adalah *Root Mean Square Propagation* (RMSProp). RMSProp bekerja dengan cara mempertahankan rata-rata bergerak gradien kuadrat untuk setiap bobot. Nilai γ yang disarankan adalah 0,9 dan nilai *learning rate* (η) adalah 0,001 (Tieleman & Hinton, 2012). Perhitungan RMSProp sebagai berikut:

$$g_t = \nabla_{\theta_{t-1}} J(\theta_{t-1}) \quad (3)$$

$$n_t = (1 - \gamma)g_t^2 + \gamma n_{t-1} \quad (4)$$

$$\theta_t = \theta_{t-1} - \eta \frac{g_t}{\sqrt{n_t + \epsilon}} \quad (5)$$

Algoritma optimasi yang lain adalah *Adaptive Moment Estimation* (Adam). Adam adalah metode yang menghitung kecepatan pembelajaran adaptif untuk setiap parameter (Ruder, 2016). Metode optimasi Adam menggabungkan ide dari optimasi *Momentum* dan RMSProp (Géron, 2019). Perhitungan metode optimasi Adam sebagai berikut (Kingma & Ba, 2014):

$$m_t = (1 - \beta_1)g_t + \beta_1 m_{t-1} \quad (6)$$

$$v_t = (1 - \beta_2)g_t^2 + \beta_2 v_{t-1} \quad (7)$$

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (8)$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (9)$$

$$\theta_t = \theta_{t-1} - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}} \quad (10)$$

2.6 Ukuran Kinerja Model

Salah satu bagian penting dari ANN adalah pengukuran kinerja dari model atau mengevaluasi performanya. Matrik yang sering digunakan pada permasalahan regresi adalah *Mean Square Error* (MSE). MSE didefinisikan sebagai selisih kuadrat rata-rata dari nilai yang diamati dan nilai hasil prediksi (Frost, n.d.). Persamaan MSE dapat ditulis sebagai berikut:

$$MSE = \frac{\sum_{i=1}^n (x_i - \widehat{x}_i)^2}{n} \quad (11)$$

dengan x_i adalah nilai aktual pada data ke- i dan \widehat{x}_i adalah nilai prediksinya.

Selain MSE, ukuran kinerja model yang sering digunakan adalah *Mean Absolute Percentage Error* (MAPE). MAPE merupakan ukuran ketepatan relatif yang digunakan untuk mengetahui persentase penyimpangan hasil pendugaan (Aindhae, 2019). Persamaannya dapat dirumuskan sebagai berikut:

$$MAPE = \frac{\sum_{i=1}^n |\widehat{x}_i - x_i|}{n} \times 100\% \quad (12)$$

2.7 R-Shiny

R-Shiny merupakan salah satu *toolkit* dari program *R* dengan menggunakan paket *shiny* yang digunakan untuk membuat *interface* seperti GUI maupun web interaktif. Paket *shiny*

lebih sering digunakan pada program *RStudio* yang merupakan salah satu program pengembangan dari *R*. *RStudio* sendiri memiliki kelebihan dibandingkan dengan *R* biasa yaitu memiliki tampilan dan *toolkit* yang *user-friendly* dan lebih mudah digunakan oleh pengguna umum. Komponen program *R-Shiny* dibedakan menjadi dua kelompok besar yaitu *user interface* (UI) dan *server* (Tirta, 2014).

3. METODE PENELITIAN

3.1 Sumber Data dan Variabel Penelitian

Jenis data yang digunakan dalam penelitian ini adalah data skunder. Data yang digunakan adalah harga kakao harian pada jenis perdagangan *spot* dari tanggal 3 Januari 2011 hingga tanggal 29 Desember 2020. Data tersebut diperoleh dari Badan Pengawas Perdagangan Berjangka Komoditi Kementerian Perdagangan. Data tersebut dapat diakses secara daring melalui situs resmi di https://www.bappebti.go.id/harga_komoditi_bursa.

Variabel yang digunakan dalam penelitian ini adalah data harga kakao harian dalam satuan Rupiah per kilogram. Data diubah menjadi data *supervised*. Data dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Pembagian data *training* dan *testing* dibagi menjadi 3 macam, yaitu: 70%, 80%, dan 90% data *training*.

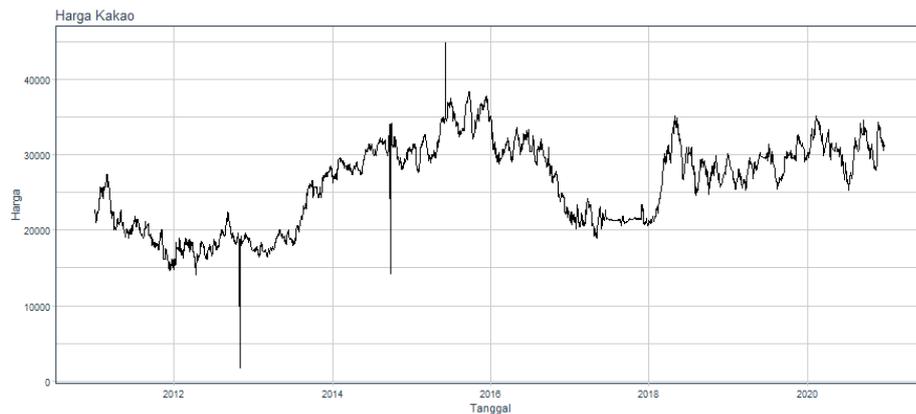
3.2 Langkah-Langkah Analisis Data

1. Membuat desain tata letak *interface R-Shiny* yaitu *ui.R*,
2. Menyusun syntax program analisis LSTM pada file *server.R* dengan langkah-langkah analisis sebagai berikut :
 - a. Melakukan *load* data yang digunakan dalam penelitian,
 - b. Melakukan *Pre-Procesing* data,
 - c. Melakukan pembagian data *training* dan data *testing*,
 - d. Menentukan Jumlah neuron di *hidden layer*,
 - e. Menentukan jumlah *timesteps*,
 - f. Menentukan nilai *batch size*,
 - g. Menentukan metode optimasi,
 - h. Melakukan evaluasi model,
 - i. Melakukan prediksi pada data uji,
 - j. Melakukan denormalisasi pada data hasil prediksi,
 - k. Menampilkan *plot* data hasil prediksi,
3. Menjalankan GUI *RShiny* yang telah dibuat.

4. HASIL DAN PEMBAHASAN

4.1 Analisis Data Eksplorasi Harga Kakao

Data yang digunakan pada penelitian ini adalah data sekunder yaitu data harga harian kakao, yang diperoleh dari Badan Pengawas Perdagangan Berjangka Komoditi Kementerian Perdagangan. Data tersebut terdiri dari dua subdata yaitu Tanggal dan Harga. Data harga harian kakao ini terdiri dari 2222 data dari tanggal 3 Januari 2011 sampai dengan tanggal 29 Desember 2020. Selanjutnya, harga harian kakao akan divisualisasikan dalam plot pada Gambar 2.



Gambar 2. Plot data harga kakao

4.2 Perancangan Arsitektur LSTM

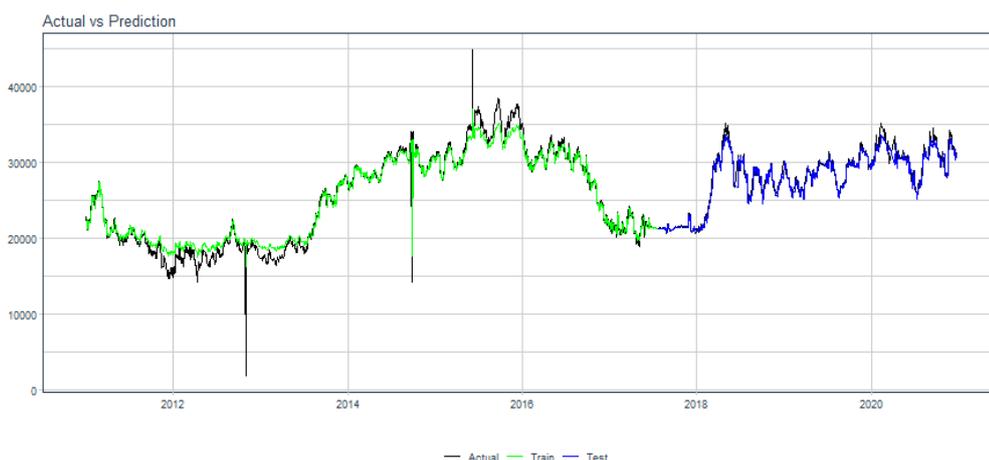
Jaringan syaraf tiruan *long short-term memory* atau LSTM memiliki arsitektur jaringan yang terdiri dari 3 lapisan (*layer*) yaitu lapisan input, lapisan tersembunyi (*hidden*), dan lapisan output. Arsitektur jaringan yang dirancang pada penelitian ini terdiri atas 3 *layer* (*input, hidden, dan output*) dengan *neuron* untuk lapisan input berjumlah satu *neuron*, jumlah *neuron* untuk lapisan *hidden* ditentukan oleh pengguna aplikasi, dan untuk lapisan *output* berjumlah satu *neuron*. Parameter yang digunakan pada perancangan arsitektur LSTM adalah Fungsi aktivasi, *neuron* pada setiap *hidden layer*, *timesteps*, *batch size*, *epoch*, dan metode optimasi. Prediksi harga kakao dengan metode LSTM dilakukan dengan melakukan kombinasi hyperparameter untuk mendapatkan model yang terbaik. Pada bagian input, akan digunakan pilihan *timesteps* dengan nilai 1, 3, dan 6. *Hidden layer* yang digunakan adalah satu, dengan pilihan *unit neuron* di *hidden layer* adalah 6, 8 dan 10. *Batch size* yang digunakan adalah 3, dengan *epochs* 5. Metode optimasi yang digunakan adalah RMSProp dan Adam. Data dibagi menjadi dua yaitu data *training* dan data *testing*. Data *testing* dikategorikan menjadi 3 yaitu 70%, 80% dan 90%.

4.3 Perancangan Interface R-Shiny

Interface R-Shiny untuk Implementasi model *Long Short-Term Memory* pada peramalan harga jual kakao ini dibangun dengan menggunakan paket *shiny* pada aplikasi *Rstudio* versi 2021.09.0. Dalam membuat *interface* atau GUI terdapat dua bagian penting, bagian pertama yaitu merancang tata letak atau tampilan pada dari GUI, dan bagian kedua yaitu merancang program yang akan dijalankan sesuai tata letak GUI. *Interface R-Shiny* ini terdiri dari tujuh menu utama yaitu menu data, menu plot data, menu prediksi, menu hasil, menu, tabel nilai prediksi, menu *plot*, dan menu peramalan, selain itu terdapat *side panel* yang digunakan untuk input data.

4.4 Hasil Peramalan Harga Kakao

Hasil Pelatihan LSTM menggunakan GUI *R-Shiny* yang telah dibuat dengan berbagai variasi arsitektur jaringan yang berbeda-beda pada 27 percobaan. Dari variasi tersebut dipilih model terbaik yang akan digunakan untuk peramalan data periode selanjutnya. Hasil pelatihan dari data *Harga_kakao.csv* dengan variasi arsitektur jaringan yang berbeda yaitu 3 presentase jumlah data *training*, 5 *Timesteps*, 3 *Neuron hidden*. dan 2 metode optimasi menunjukkan bahwa nilai MSE terendah adalah 491505,1. Nilai tersebut diperoleh dengan menggunakan data *training* 70%, *timesteps* 3, 10 *neuron hidden layer*, dan metode optimasi adam. Plot antara data actual dan data prediksi dengan menggunakan data *training* 70%, *timesteps* 3, 10 *neuron hidden layer* dan metode optimasi adam ditampilkan pada Gambar 3.



Gambar 3. Plot Aktual dan Nilai Prediksi

Gambar 3 menampilkan plot data aktual dan data prediksi harga Kakao. Plot yang berwarna merah adalah data aktual, sedangkan plot yang berwarna biru adalah data prediksi. Berdasarkan Gambar 28, terlihat bahwa model yang dibentuk dapat menghasilkan output yang sesuai. Plot hasil prediksi dapat mengikuti plot data aktual. Berikut ditampilkan dalam Tabel 1 lima data pertama hasil prediksi pada *data testing* harga kakao.

Tabel 1. Prediksi Data Testing

Tanggal	Data Aktual	Data Prediksi
7/24/2017	21317,00	21034,80
7/26/2017	21341,00	21054,35
7/27/2017	21311,00	21073,97
7/28/2017	21328,00	21049,46
7/31/2017	21323,00	21063,33
....
12/23/2020	31092,63	31132,33
12/29/2020	31190,22	31036,00

Kinerja dari model yang telah dilatih diukur menggunakan MAPE. Berdasarkan hasil prediksi yang dilakukan nilai MAPE yang diperoleh sebesar 2,00%. Nilai MAPE kurang dari 10%, sehingga model memiliki akurasi peramalan yang tinggi.

4.5 Forecasting

Model terbaik yang telah didapat digunakan untuk melakukan peramalan data pada periode yang akan datang. Pada penelitian ini akan dilakukan peramalan pada tanggal 1 November 2021 sampai dengan 31 Desember 2021, Sebagian datanya dapat dilihat pada Tabel 2.

Tabel 2. Peramalan data

Tanggal	Harga
11/1/2021	29419,62
11/2/2021	29419,62
11/3/2021	28904,52
11/4/2021	29308,7
11/5/2021	28413,57
11/6/2021	29342,65

11/7/2021	29472,94
....
....
....
12/30/2021	26531,27
12/31/2021	27310,66

5. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan tentang prediksi harga kakao menggunakan metode *long short-term memory* (LSTM) dengan *GUI RShiny* ditarik kesimpulan bahwa:

1. Prediksi harga kakao menggunakan LSTM dengan metode optimasi Adam lebih baik dibandingkan dengan metode optimasi RMSProp dilihat dari nilai MSE dan MAPE yang lebih kecil.
2. Model terbaik adalah model dengan data *training* 70%, *timesteps* 3, dan 10 *neuron hidden layer* dengan menggunakan Metode Optimasi Adam.
3. Hasil Peramalan tanggal 1 November 2021 sampai dengan 31 Desember 2021 menunjukkan harga kakao cenderung menurun.

DAFTAR PUSTAKA

- Aindhae. (2019). *CARA MENGHITUNG MEAN ABSOLUTE PERCENTAGE ERROR (MAPE) DENGAN EXCEL*. <https://www.aindhae.com/2019/12/cara-menghitung-mean-absolute.html>
- Amal, I. (2021). *PREDIKSI HARGA MINYAK SAWIT MENTAH MENGGUNAKAN MULTILAYER PERCEPTRON DAN LONG SHORT-TERM MEMORY*.
- Bappebti. (2021). *Harga Bursa (Foward - Future - Spot)*. https://www.bappebti.go.id/harga_komoditi_bursa
- BPS. (2020). *[Seri 2010] PDB Seri 2010 (Milyar Rupiah) 2019*. <https://www.bps.go.id/indicator/11/65/2/-seri-2010-pdb-seri-2010.html>
- Brownlee, J. (2017). *Long short-term memory networks with python: develop sequence prediction models with deep learning*. Machine Learning Mastery.
- Ciaburro, G., & Venkateswaran, B. (2017). Neural network with R: Smart models using CNN, RNN, deep learning, and artificial intelligence principles. In *Packt* (Vol. 91). Packt Publishing.
- De Marchi, L., & Mitchell, L. (2019). *Hands-On Neural Networks: Learn how to build and train your first neural network model using Python*. Packt Publishing Ltd.
- Demuth, H. B., Beale, M. H., De Jess, O., & Hagan, M. T. (2014). *Neural network design*. Martin Hagan.
- Fabien, M. (2019). *Activation Functions*. <https://maelfabien.github.io/deeplearning/act/>
- Frost, J. (n.d.). *Mean Squared Error (MSE)*. <https://statisticsbyjim.com/regression/mean-squared-error-mse/>
- Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*.
- Mallya, A. (2017). *Lstm forward and backward pass*. <https://arunmallya.github.io/writeups/nn/lstm/index.html#/>

- Nur, H. A. (2021). *ANALISIS DAMPAK PENETAPAN HARGA KAKAO OLEH TENGGULAK TERHADAP KESEJAHTERAAN PETANI DITINJAU DALAM PERSPEKTIF EKONOMI ISLAM (Studi Pada Desa Padang Cermin, Kecamatan Padang Cermin, Kabupaten Pesawaran)*. UNIVERSITAS ISLAM NEGERI RADEN INTAN LAMPUNG.
- Puntambekar, A. (2020). *Strengths and Weaknesses of Optimization Algorithms Used for Machine Learning*. <https://medium.com/swlh/strengths-and-weaknesses-of-optimization-algorithms-used-for-machine-learning-58926b1d69dd>
- Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PloS One*, 15(1), e0227222.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *ArXiv Preprint ArXiv:1609.04747*.
- Stewart, M. (2019). *Introduction to Neural Networks*. Toward Data Science. <https://towardsdatascience.com/simple-introduction-to-neural-networks-ac1d7c3d7a2c>
- Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 26–31.
- Tirta, I. M. (2014). Pengembangan E-Modul Statistika Terintegrasi dan Dinamik dengan R-shiny dan mathJax. *Prosiding Seminar Nasional Matematika, Universitas Jember, 2000*(November), 223–232.
- Zahara, S., Sugianto, & Ilmiddafiq, M. B. (2017). Prediksi Indeks Harga Konsumen Menggunakan Metode Long Short Term Memory (LSTM) Berbasis Cloud Computing. *Resti*, 1(1), 19–25.