

APLIKASI METODE ARIMA, LSTM, DAN *HYBRID* ARIMA-LSTM PADA PERAMALAN HARGA *CRUDE PALM OIL* (CPO) DUNIA

ELISABETH KURNIASARI MARCELA USKONO



**DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2023**



@Hak cipta milik IPB University

IPB University



IPB University
— Bogor Indonesia —

Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

Perpustakaan IPB University

PERNYATAAN MENGENAI SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa skripsi dengan judul “Aplikasi Metode ARIMA, LSTM, dan *Hybrid* ARIMA-LSTM pada Peramalan Harga *Crude Palm Oil* (CPO) Dunia” adalah karya saya dengan arahan dari dosen pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, Juni Tahun 2023

Elisabeth Kurniasari Marcela Uskono
G14190091



ABSTRAK

ELISABETH KURNIASARI MARCELA USKONO. Aplikasi Metode ARIMA, LSTM dan *Hybrid* ARIMA-LSTM pada Peramalan Harga Minyak Sawit Mentah Dunia. Dibimbing oleh AKBAR RIZKI dan MOHAMMAD MASJKUR.

Minyak sawit mentah atau *Crude Palm Oil* (CPO) adalah salah satu komoditas unggulan ekspor Indonesia di sektor pertanian. Indonesia juga merupakan produsen CPO terbesar di dunia. Harga CPO dunia berdampak signifikan terhadap volume ekspor CPO serta pertumbuhan ekonomi Indonesia. Peramalan harga CPO secara akurat sangat penting untuk membantu pengusaha, investor, produsen dan juga pemerintah dalam perencanaan ekonomi dengan mempertimbangkan perubahan dan fluktuasi harga. Penelitian ini menggunakan data harga bulanan CPO dunia dari Januari 2003 hingga April 2023 yang diperoleh dari rata-rata harga harian CPO yang bersumber dari kompilasi data ISTA Mielke GmbH, Oil World, US Department of Agriculture dan World Bank. Metode yang digunakan dalam penelitian ini adalah *Autoregressive Integrated Moving Average* (ARIMA), *Long Short-Term Memory* (LSTM) dan *Hybrid* ARIMA-LSTM. Penggunaan ketiga metode ini didasarkan pada kondisi data yang berbentuk linear dan nonlinear. Metode LSTM terbukti lebih efektif dalam memprediksi harga CPO dunia dibanding metode ARIMA dan *Hybrid* ARIMA-LSTM. Model LSTM terbaik adalah model dengan *hyperparameter epoch* 100, *batch size* 1, dan *learning rate* 0,001. Model ini menghasilkan nilai *Mean Absolute Percentage Error* 2,33% dan *Root Mean Squared Error* 34,708. Hal ini karena model LSTM lebih stabil dalam mempelajari pola data deret waktu.

Kata kunci: ARIMA, harga minyak sawit mentah, kombinasi ARIMA-LSTM, LSTM, peramalan.



ABSTRACT

ELISABETH KURNIASARI MARCELA USKONO. Application of ARIMA, LSTM, and Hybrid ARIMA-LSTM Methods in Forecasting World Crude Palm Oil (CPO) Prices. Supervised by AKBAR RIZKI and MOHAMMAD MASJKUR.

Crude Palm Oil (CPO) is one of Indonesia's main export commodities in the agricultural sector. Indonesia is also the largest producer of CPO in the world. The world CPO prices have a significant impact on Indonesia's export volume of CPO as well as its economic growth. It is essential to forecast CPO prices accurately to assist entrepreneurs, investors, producers, and the government in economic planning for the future while considering changes and price fluctuations. This study uses world CPO monthly price data from January 2003 to April 2023 which is compilation of data from ISTA Mielke GmbH, Oil World, US Department of Agriculture and World Bank. The methods used in this research are Autoregressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM), and Hybrid ARIMA-LSTM. The use of these methods is based on linear and nonlinear data conditions. Due to its ability to learn the data pattern well, the LSTM was found to be more effective in forecasting world CPO prices. The best LSTM model in this study is a model with hyperparameter epoch 100, batch size 1, and learning rate 0,001. This model produces a MAPE and an RMSE value of 2,33% and 34,708.

Keywords: ARIMA, crude palm oil prices, forecasting, hybrid ARIMA-LSTM, LSTM.



@Hak cipta milik IPB University

Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

© Hak Cipta milik IPB, tahun 2023¹
Hak Cipta dilindungi Undang-Undang

Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan atau menyebutkan sumbernya. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik, atau tinjauan suatu masalah, dan pengutipan tersebut tidak merugikan kepentingan IPB.

Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apa pun tanpa izin IPB.

APLIKASI METODE ARIMA, LSTM DAN *HYBRID* ARIMA-LSTM PADA PERAMALAN HARGA *CRUDE PALM OIL* (CPO) DUNIA

ELISABETH KURNIASARI MARCELA USKONO

Skripsi
sebagai salah satu syarat untuk memperoleh gelar
Sarjana pada
Program Studi Statistika dan Sains Data

**DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2023**



@Hak cipta milik IPB University

Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.



Judul Skripsi : Aplikasi Metode ARIMA, LSTM, dan *Hybrid* ARIMA-LSTM
pada Peramalan Harga *Crude Palm Oil* (CPO) Dunia

Nama : Elisabeth Kurniasari Marcela Uskono

NIM : G14190091

Disetujui oleh

Pembimbing 1:

Akbar Rizki, S.Stat., M.Si.

Pembimbing 2:

Ir. Mohammad Masjkur, M.S.

Diketahui oleh

Ketua Departemen:

Dr. Bagus Sartono, S.Si., M.Si.

NIP. 197804112005011002

Tanggal Ujian:
26 Juni 2023

Tanggal Lulus:

(tanggal penandatanganan
Fakultas/Sekolah ...)

oleh

Dekan



PRAKATA

Puji dan syukur penulis panjatkan kepada Tuhan yang Maha Esa atas segala karunia-Nya sehingga karya ilmiah ini berhasil diselesaikan. Tema yang dipilih dalam penelitian yang dilaksanakan sejak bulan Januari 2023 sampai bulan Juni 2023 ini ialah peramalan data deret waktu, dengan judul “Aplikasi Metode ARIMA, LSTM, dan *Hybrid* ARIMA-LSTM pada Peramalan Harga *Crude Palm Oil* (CPO) Dunia”.

Karya ilmiah ini dapat diselesaikan tidak terlepas dari dukungan berbagai pihak dalam prosesnya. Oleh karena itu, penulis ingin mengucapkan terimakasih kepada:

1. Ibu Akbar Rizki, S.Stat., M.Si. dan Bapak Ir. Mohammad Masjkur, M.S., selaku pembimbing yang telah bersedia membimbing, memotivasi, meluangkan waktu, dan memberi banyak saran dalam penelitian serta penyusunan karya ilmiah ini.
2. Ibu Sachnaz Desta Oktarina, S.Stat., M.Agr.Sc., Ph.D selaku dosen penguji luar komisi pembimbing yang telah memberi saran serta koreksi dalam penulisan karya ilmiah ini.
3. Seluruh keluarga besar yang telah menjadi penyemangat utama, selalu mendukung dan mendoakan penulis.
4. Grashela, Anggun, Resti, Arie, Rochmat, Hambali, Adel, Christin, Apriana, teman Statistika 56 yang senantiasa mendukung, membantu dan menemani penulis dalam masa penyusunan karya ilmiah ini.
5. Bang Rafi, Kak Putri, kating STK yang telah bersedia meluangkan waktu untuk membantu penulis selama penyusunan karya ilmiah ini.
6. Picen, Ihsan, Jeanne, Sabe, Fida, Dhea, Novia, Retno, Sekar, dan keluarga KKN di Ngambarsari yang selalu mendukung dan memotivasi penulis.
7. Yohana, Justine, Yalia, Chandra, Sely, teman-teman penulis yang selalu mendengarkan dan memberi semangat selama proses penyusunan karya ilmiah ini.
8. Keluarga pendamping IPB yang sudah memberikan penulis banyak pengalaman dan pelajaran hidup.
9. Seluruh dosen pengajar dan pegawai Departemen Statistika IPB yang telah memberikan ilmu yang bermanfaat dan mendukung rangkaian penyelesaian tugas akhir dengan menunjang segala kebutuhan dalam proses administrasi penulis.
10. Semua pihak terkait yang telah mendukung penulis dalam penyusunan karya ilmiah ini yang tidak dapat penulis sebutkan satu per satu.

Semoga karya ilmiah ini bermanfaat bagi pihak yang membutuhkan dan bagi kemajuan ilmu pengetahuan.

Bogor, Juni 2023

Elisabeth Kurniasari Marcela Uskono



DAFTAR ISI

DAFTAR TABEL	x
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN	x
I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	2
II TINJAUAN PUSTAKA	3
2.1 <i>Crude Palm Oil (CPO)</i>	3
2.2 <i>Autoregressive Integrated Moving Average (ARIMA)</i>	3
2.3 <i>Long Short-Term Memory (LSTM)</i>	5
2.4 <i>Hybrid ARIMA-LSTM</i>	6
III METODE	7
3.1 Data	7
3.2 Prosedur Analisis	7
IV HASIL DAN PEMBAHASAN	11
4.1 Eksplorasi Data	11
4.2 Pembagian Data	12
4.3 Pemodelan ARIMA	12
4.4 <i>LSTM Network</i>	16
4.5 <i>Hybrid ARIMA-LSTM</i>	18
4.6 Evaluasi Model	19
4.7 Peramalan	21
V SIMPULAN DAN SARAN	22
5.1 Simpulan	22
5.2 Saran	22
DAFTAR PUSTAKA	23
LAMPIRAN	26
RIWAYAT HIDUP	33



DAFTAR TABEL

1	Ciri plot ACF dan PACF untuk identifikasi model ARIMA	7
2	Nilai pendugaan parameter model tentatif ARIMA	15
3	Nilai pendugaan parameter model <i>Overfitting</i>	15
4	Arsitektur dan <i>hyperparameter</i> model LSTM	16
5	Nilai rata-rata <i>loss</i> kombinasi <i>hyperparameter</i>	17
6	Nilai rata-rata <i>loss</i> kombinasi <i>hyperparameter</i> untuk sisaan ARIMA	19
7	Evaluasi model	19

DAFTAR GAMBAR

1	Algoritma LSTM (Fan <i>et al.</i> 2021)	5
2	Tabel ESACF	7
3	<i>k-fold cross validation</i> dengan $k = 5$	9
4	Grafik fluktuasi harga CPO dunia (Januari 2003 – April 2023)	11
5	Pembagian data latih dan data uji	12
6	Plot ACF harga CPO dunia (a) sebelum pembedaan dan (b) setelah pembedaan satu kali ($d=1$)	13
7	Plot deret waktu harga CPO setelah pembedaan satu kali ($d=1$)	13
8	Plot PACF setelah pembedaan satu kali ($d=1$)	14
9	Identifikasi model tentatif dengan plot ESACF	14
10	Plot data setelah normalisasi	16
11	Plot data sisaan ARIMA (a) sebelum dinormalisasi dan (b) setelah dinormalisasi	18
12	Plot perbandingan data latih dengan data prediksi	20
13	Plot perbandingan data uji dengan data prediksi	21
14	Peramalan harga <i>crude palm oil</i> dunia	21

DAFTAR LAMPIRAN

8	Lampiran 1 Pemodelan ARIMA di <i>Python</i>	27
9	Lampiran 2 Pemodelan LSTM di <i>Python</i>	29
10	Lampiran 3 Pemodelan Hybrid di <i>Python</i>	32

I PENDAHULUAN

1.1 Latar Belakang

Minyak kelapa sawit merupakan energi alternatif utama pengganti bahan bakar fosil (Hilda 2014). Menurut BPS (2022) Indonesia adalah produsen minyak kelapa sawit terbesar di dunia. Data Index Mundi (2016) menunjukkan produksi minyak sawit Indonesia sebesar 36.000.000-ton metrik, lebih tinggi 15.000.000-ton metrik daripada Malaysia. Salah satu jenis minyak kelapa sawit yang dipasarkan adalah CPO atau *crude palm oil*. CPO adalah salah satu dari sembilan komoditas utama ekspor Indonesia pada bidang perkebunan yang menyumbang sebesar 53,37% dari total ekspor di bidang tersebut (Hilda 2014). Fluktuasi harga CPO dunia mempengaruhi volume ekspor kelapa sawit dan perekonomian Indonesia. Dari sisi Indonesia selaku produsen, kenaikan harga CPO mendorong kenaikan volume ekspor komoditas kelapa sawit, laju pertumbuhan ekonomi Indonesia, inflasi dan jumlah uang yang beredar. Meski demikian kenaikan harga CPO justru melemahkan nilai tukar riil rupiah (Iskandar 2015). Syahril *et al.* (2019) dalam penelitiannya mengenai pengaruh CPO terhadap indeks lingkungan hidup menyebutkan bahwa pada jangka panjang harga CPO berpengaruh positif terhadap indeks kualitas lingkungan Indonesia. Oleh karena besarnya pengaruh harga CPO dunia, peramalan harga CPO dunia perlu dilakukan untuk memberikan informasi penting bagi pengusaha, investor, produsen dan juga pemerintah. Peramalan harga CPO dengan model yang baik membantu proses pengambilan keputusan dalam perencanaan ekonomi dengan memperhatikan perubahan dan fluktuasi harga yang mungkin di masa depan (Ariff *et al.* 2015).

Peramalan harga CPO cukup menantang karena fluktuasi harga CPO terus terjadi sepanjang waktu (Khalid *et al.* 2018). Salah satu metode peramalan yang paling terkenal adalah *Autoregressive Integrated Moving Average* (ARIMA) (Rahim *et al.* 2018). ARIMA merupakan pengembangan dari metode *Autoregressive Moving Average* (ARMA) untuk mengatasi ketidakstasioneran data (Chatfield 2003). Model ARIMA yang tepat dan memenuhi kriteria dalam evaluasi model valid digunakan untuk meramalkan harga CPO (Oktiani 2018). Arifin *et al.* (2021) melakukan peramalan harga CPO di spot Medan menggunakan enam metode yaitu *Moving Average*, *Double Moving Average*, *Exponential Moving Average*, *Double Exponential Smoothing*, *Winter* dan ARIMA-SARIMA dengan hasil nilai MSE metode ARIMA paling kecil dibanding metode lainnya (284,155). Sihombing *et al.* (2022) dalam penelitiannya menyimpulkan bahwa peramalan harga CPO dengan menggunakan ARIMA cukup baik dengan kesalahan yang lebih kecil dibanding model eksponensial *smoothing Error*, *Trend*, *Seasonal* (ETS). Hal ini karena metode ARIMA lebih baik dibanding metode ETS untuk peramalan data berbentuk linear dan dalam jumlah besar. Metode ARIMA baik digunakan untuk peramalan data linear namun diketahui tidak cukup baik untuk analisis data nonlinear (Fan *et al.* 2021).

Harga CPO menunjukkan adanya unsur nonlinear dan volatilitas yang tinggi (Ofuoku dan Nginiatedema 2022). Oleh karena itu, penggunaan *machine learning* dalam peramalan data CPO dapat dilakukan untuk mengatasi kelemahan metode ARIMA. Metode *machine learning* bekerja dengan baik untuk penggunaan data nonlinear (Wu *et al.* 2019). Salah satu metode *machine learning* yang

dikembangkan adalah *Long Short-Term Memory* (LSTM). LSTM adalah pengembangan dari metode *Recurrent Neural Network* (RNN) untuk mengatasi masalah penanganan deret waktu pada RNN terkait peristiwa *gradien exploded or vanished* (Xu *et al.* 2021). Peramalan dengan menggunakan LSTM baik *univariate* maupun *multivariate* menunjukkan hasil yang baik dengan tingkat kesalahan yang rendah (Alencar *et al.* 2021). Ofuoku dan Ngniatedema (2022) pada penelitiannya menyimpulkan bahwa metode LSTM bekerja dengan lebih baik dibandingkan *Neural Network* lainnya seperti *Multilayer Perceptron* dan *Simple Recurrent Neural Network* pada peramalan harga CPO. Ofuoku dan Ngniatedema (2022) menyebutkan bahwa metode LSTM andal untuk peramalan dengan memberikan informasi yang berguna dan akurat.

Metode ARIMA dan LSTM memiliki kelebihan dan kekurangannya masing-masing. Oleh karena itu metode lain yang digunakan pada penelitian ini adalah *Hybrid ARIMA-LSTM*. Metode *Hybrid ARIMA-LSTM* merupakan metode yang menggabungkan metode ARIMA dan LSTM sehingga dapat menangani data linear dan nonlinear secara bersama-sama (Choi 2018). Pentingnya peramalan harga CPO dan volatilitasnya yang tinggi menginisiasi penggunaan metode ARIMA, LSTM dan *Hybrid ARIMA-LSTM* pada penelitian ini. Penggunaan metode ini diharapkan akan menghasilkan peramalan harga CPO dunia dengan tingkat akurasi yang tinggi.

1.2 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah:

1. Menentukan metode terbaik antara ARIMA, LSTM dan *Hybrid ARIMA-LSTM* untuk peramalan harga *Crude Palm Oil* (CPO) dunia
2. Melakukan peramalan harga *Crude Palm Oil* (CPO) dunia dengan model terbaik

II TINJAUAN PUSTAKA

2.1 Crude Palm Oil (CPO)

Minyak sawit mentah atau *Crude Palm Oil* (CPO) adalah bagian dari komoditas kelapa sawit yang memiliki potensi tinggi. Selain itu CPO sebagai produk olahan memiliki nilai ekonomis yang lebih tinggi dibanding tandan buah segar kelapa sawit. CPO digunakan pada berbagai industri seperti industri minyak goreng, lemak khusus, margarin dan sabun mandi (BPS 2022).

2.2 Autoregressive Integrated Moving Average (ARIMA)

ARIMA model pertama kali diperkenalkan oleh Box & Jenkins pada tahun 1970. Model yang dibentuk adalah ARIMA(p, d, q). Elemen dari metode ARIMA yaitu:

- Autoregressive* (AR): Proses yang mengasumsikan bahwa data pada waktu saat ini dipengaruhi oleh data pada waktu-waktu sebelumnya dengan ordo (p);
- Integrated* (I): Proses menstasionerkan data (*differencing*= d);
- Moving Average* (MA): Proses yang mengasumsikan bahwa data pada saat ini dipengaruhi oleh sisaan (*residual*) waktu-waktu sebelumnya dengan ordo (q) (Choi 2018).

Model ARIMA diformulasikan sebagai berikut:

$$\phi(B)(1-B)^d y_t = \delta + \theta(B)\varepsilon_t$$

Dengan $\phi(B) = 1 - \phi_1 B^1 - \phi_2 B^2 - \dots - \phi_p B^p$ adalah koefisien parameter AR ke- p , $(1-B)^d y_t$ adalah proses *differencing* ke- d data y pada waktu ke- t , δ adalah intersep, $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$ adalah koefisien parameter MA ke- q dan ε_t adalah sisaan pada waktu ke- t (Montgomery *et al.* 2015).

2.2.1 Kestasioneran Deret Waktu

Data deret waktu dikatakan stasioner jika nilai rata-rata, ragam dan korelasi antar data tidak terpengaruh oleh perubahan waktu. Stasioner menunjukkan adanya kesetimbangan dan stabilitas statistik dalam data. Kestasioneran data dapat diidentifikasi menggunakan plot *Autocorrelation Function* (ACF). Kondisi data tidak stasioner dicirikan dengan data yang terlihat menurun secara perlahan (mengikuti pola eksponensial) dan tidak *cut off* (*tail off*) dekat nol setelah beberapa *lag* (Montgomery *et al.* 2015). Uji kestasioneran yang paling banyak digunakan adalah uji *Augmented Dickey Fuller* (ADF) yang telah mempertimbangkan adanya autokorelasi pada sisaan apabila data deret waktu yang diuji ternyata tidak stasioner (Aktivani 2021).

Hipotesis yang digunakan adalah:

$H_0: \delta = 0$, data mengandung *unit root* (tidak stasioner dalam rata-rata)

$H_1: \delta < 0$, data tidak mengandung *unit root* (sudah stasioner dalam rata-rata)

Statistik uji:

$$\tau = \frac{\delta}{se(\delta)}$$

dengan $Se(\delta)$ sebagai *standard error* dari δ . Jika nilai mutlak dari τ statistik uji ADF lebih besar dari nilai kritis ADF pada tabel MacKinnon atau nilai p uji ADF lebih kecil dari taraf nyata (0,05) maka tolak H_0 atau data diketahui sudah stasioner.

Apabila data tidak stasioner dalam rata-rata dapat dilakukan pembedaan atau *differencing*. Pembedaan dilakukan dengan menerapkan operator *difference* untuk menghitung perubahan nilai data deret waktu untuk mendapatkan data deret waktu baru dimana $x_t = y_t - y_{t-1}$ (Montgomery *et al.* 2015). Setelah dilakukan pembedaan kemudian dilakukan pemeriksaan kestasioneran data lagi. Aktivani (2021) dalam penelitiannya menyampaikan bahwa pembedaan terus dilakukan hingga data stasioner setelah pembedaan ordo (d).

2.2.2 Diagnostik Model

1. Uji kebebasan antar sisaan

Kebebasan antar sisaan diperiksa dengan uji Ljung-Box. Uji ini dilakukan untuk mengetahui ada tidaknya autokorelasi pada sisaan dengan melihat apakah semua nilai autokorelasi K data pertama sama dengan 0. Hipotesis yang diujikan adalah:

$H_0: \rho_1 = \rho_2 = \dots = \rho_k = 0$, tidak ada autokorelasi pada sisaan.

H_1 : Minimal ada satu $\rho_j \neq 0$ ($j = 1, 2, \dots, k$), ada autokorelasi pada sisaan.

Statistik uji:

$$LB = n(n+2) \sum_{k=1}^K \frac{\hat{\rho}_k^2}{n-k}$$

Daerah kritis:

Statistik uji Ljung-Box menunjukkan adanya distribusi Khi-Kuadrat sehingga jika statistik uji lebih tinggi atau sama dengan nilai Khi-Kuadrat $\chi^2_{1-\alpha, K-p-q}$ dengan derajat bebas $K-p-q$ pada tabel maka tolak H_0 . Kondisi ini juga terjadi jika nilai p uji Ljung-Box lebih kecil dari α (0,05) (Abassi *et al.* 2021).

2. Uji kenormalan sisaan

Pengujian sisaan berdistribusi normal dapat dilakukan dengan uji Jarque-Bera. Hipotesis yang digunakan pada uji ini yaitu:

H_0 : Data berdistribusi normal

H_1 : Data tidak berdistribusi normal

Statistik Uji:

$$JB = \frac{(n-k)}{6} \left(S^2 + \frac{(K-3)^2}{4} \right)$$

Dengan S merupakan *skewness*, K merupakan *kurtosis*, n jumlah sampel dan k sebagai jumlah koefisien yang diestimasi.

Daerah kritis:

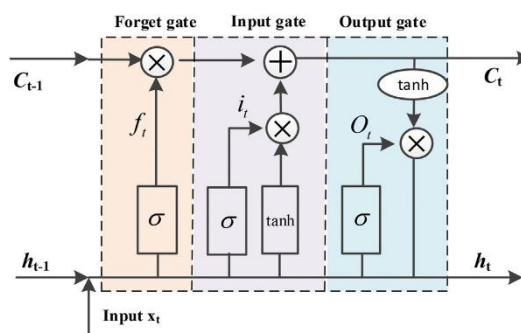
Statistik uji Jarque-Bera menunjukkan adanya distribusi Khi-Kuadrat dengan derajat bebas 2 sehingga jika statistik uji lebih tinggi atau sama dengan nilai Khi-Kuadrat $\chi^2_{\alpha, 2}$ dengan derajat bebas 2 pada tabel maka tolak H_0 . Jika nilai p dari hasil uji Jarque-Bera lebih kecil dari α (0,05) maka tolak H_0 atau data diketahui tidak berdistribusi normal (Abassi *et al.* 2021).

2.3 Long Short-Term Memory (LSTM)

LSTM adalah jenis khusus dari RNN dengan kemampuan untuk merekam urutan data sehingga cocok digunakan untuk analisis data deret waktu. LSTM adalah sistem modul, tempat data ditangkap dan disimpan. Sistem LSTM terdiri dari sel-sel, menyerupai jalur transpor yang menghubungkan modul satu dengan modul lainnya untuk mengumpulkan dan mentransformasi data (Fan *et al.* 2021). Secara umum arsitektur LSTM terdiri dari *memory cell*, *forget gate*, *input gate*, dan *output gate*. LSTM menggunakan dua fungsi aktivasi yaitu *sigmoid* dan *tanh* (*Hyperbolic Tangent*). Fungsi aktivasi *sigmoid* mengubah berapapun nilai input (X) ke dalam range $[0,1]$. Fungsi aktivasi *tanh* mengubah berapapun nilai input (X) ke dalam range $[-1,1]$ (Karlic dan Olgac 2016). Berikut formula untuk fungsi aktivasi sigmoid dan tanh:

$$\sigma = \frac{1}{1 + e^{-x}}$$

$$\tanh = 2\sigma(2x) - 1$$



Gambar 1 Algoritma LSTM (Fan *et al.* 2021)

Contoh Algoritma LSTM dapat dilihat pada Gambar 1. Simbol \oplus dan \otimes menunjukkan penjumlahan dan perkalian pada model sedangkan tanda panah menunjukkan arah aliran. Informasi berupa *hidden state* dan *cell state* dari *time step* sebelumnya menjadi input tambahan pada *memory cell* (neuron) *time step* saat ini. Input saat ini (x_t) dan *hidden state* sebelumnya (h_{t-1}) kemudian di proses di *forget gate* dengan bobot (W_f) dan bias (b_f) untuk menghapus informasi yang tidak diperlukan oleh sel. Selanjutnya ada dua proses terhadap x_t dan h_{t-1} di *input gate* sebelum kemudian *cell state* sebelumnya (C_{t-1}) diperbarui menjadi *cell state* baru (C_t). Proses pertama yaitu dengan fungsi sigmoid (i_t), bobot (W_i) dan bias (b_i) untuk menentukan informasi yang perlu diperbarui pada *cell state*. Proses kedua yaitu dengan fungsi tanh (\hat{C}_t), bobot (W_c) dan bias (b_c) untuk membentuk vektor baru berisi informasi yang dapat ditambahkan ke *cell state*. Setelah mendapat *cell state* baru, data lalu diproses pada *output gate* dengan bobot (W_o) dan bias (b_o) untuk menentukan informasi yang menjadi keluaran. Nilai *output* (h_t) didasarkan pada kondisi *cell state* yang sudah diperbarui. Proses kerja LSTM dapat dilihat melalui persamaan berikut (Fan *et al.* 2021):

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

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

$$\hat{C}_t = \tanh(W_c \times x_t + U_c \times h_{t-1} + b_c)$$

$$C_t = f_t \times C_{t-1} + i_t \times \hat{C}_t$$

$$O_t = \sigma(W_o \times x_t + U_o \times h_{t-1} + b_o)$$

$$h_t = O_t \times \tanh(C_t)$$

2.3.1 Normalisasi Data

Normalisasi data yaitu proses mentransformasi data dengan menyederhanakan semua nilai data ke dalam suatu skala tertentu pada rentang yang lebih kecil. Normalisasi mengharuskan adanya nilai maksimum dan minimum dari data. Brownlee (2017) menyebutkan bahwa pada metode LSTM data dapat dinormalisasi menggunakan MinMaxScaler pada *package scikit-learn* dengan range (0,1). Normalisasi dengan MinMaxScaler dapat diformulasikan sebagai berikut (Ali dan Faraj 2014):

$$x' = \frac{x - \min}{\max - \min}$$

Setelah data diolah dilakukan denormalisasi untuk keperluan *reporting* dan *plotting* dengan formula sebagai berikut (Ali dan Faraj 2014):

$$x = x'(\max - \min) + \min$$

Keterangan:

- x' : data setelah dinormalisasi
- x : data sebelum dinormalisasi
- \max : nilai maksimum dari data
- \min : nilai minimum dari data

2.3.2 Hyperparameter Tuning

Hyperparameter tuning adalah proses menemukan kombinasi nilai *hyperparameter* optimal yang membantu model LSTM memberikan performa terbaik (Kumar *et al.* 2021). Salah satu metode *hyperparameter tuning* adalah *grid search*. Kelebihan utama dari *grid search* adalah kemampuannya mencoba semua kemungkinan kombinasi *hyperparameter* sehingga selalu menemukan kombinasi *hyperparameter* terbaik yang mampu mengoptimalkan model berdasarkan nilai akurasi dan presisi (Soares *et al.* 2019).

2.4 Hybrid ARIMA-LSTM

Hybrid ARIMA-LSTM adalah metode gabungan dari ARIMA dan LSTM untuk menangani kondisi data linear dan nonlinear sekaligus pada peramalan data deret waktu. Peramalan dilakukan terlebih dahulu menggunakan metode ARIMA (L_t) dengan sisaan berupa data yang diasumsikan nonlinear. Sisaan tersebut digunakan sebagai input pada pemodelan LSTM (N_t) dengan sisaan (ε_t). Proses peramalan dengan *Hybrid ARIMA-LSTM* dapat diformulasikan sebagai berikut (Fan *et al.* 2021):

$$x'_t = L_t + N_t + \varepsilon_t$$

III METODE

3.1 Data

Penelitian ini menggunakan data sekunder harga *Crude Palm Oil* (CPO) atau minyak sawit mentah dunia periode bulanan dari Januari 2003 sampai April 2023 yang diperoleh dari kompilasi data ISTA Mielke GmbH, Oil World, US Department of Agriculture dan World Bank dengan satuan US dollar/ ton metrik (USD/ MT).

3.2 Prosedur Analisis

Prosedur analisis pada penelitian ini dijelaskan sebagai berikut:

1. Melakukan eksplorasi data menggunakan plot data deret waktu untuk mengetahui karakteristik dan pola data.
2. Membagi data menjadi data latih dan data uji. Pembagian data didasarkan pada kemiripan pola antara data latih dan data uji.
3. Pemodelan ARIMA
 - a) Melakukan uji kestasioneran data latih dengan identifikasi plot *autocorrelation function* (ACF) dan uji formal *Augmented Dickey Fuller* (ADF).
 - b) Mengidentifikasi kandidat model tentatif berdasarkan ordo p dan q dari model ARIMA (p, d, q) dengan plot ACF dan *Partial Autocorrelation Function* (PACF) data stasioner sesuai dengan ciri pada Tabel 1. Apabila identifikasi sulit dilakukan maka digunakan juga plot *Extended Sample ACF* (Montgomery *et al.* 2015).

Tabel 1 Ciri plot ACF dan PACF untuk identifikasi model ARIMA

ACF	PACF	Kandidat Model
<i>Tails off</i>	<i>Cuts off</i> setelah lag p	AR (p)
<i>Cuts off</i> setelah lag q	<i>Tails off</i>	MA (q)
<i>Cuts off</i> setelah lag q	<i>Cuts off</i> setelah lag p	AR (p) atau MA (q)
<i>Tails off</i>	<i>Tails off</i>	ARMA (p, q)

Sumber : Montgomery *et al.* (2015)

Identifikasi model tentatif ARIMA dengan plot ESACF didasarkan pada pola segitiga "nol" asimtotik dimana titik sudut kiri atas segitiga nol terletak di baris p dan kolom q sehingga model yang terbentuk adalah ARMA (p, q) (Tsay dan Tiao 1984). Contoh terlihat pada Gambar 2 dengan model yang terbentuk adalah ARMA (1, 2).

AR	MA							
	0	1	2	3	4	5	6	7
0	*	X	X	X	X	X	X	X
1	*	X	0	0	0	0	0	0
2	*	X	X	0	0	0	0	0
3	*	X	X	X	0	0	0	0
4	*	X	X	X	X	0	0	0

Gambar 2 Tabel ESACF

- c) Menduga parameter model tentatif dengan *Maximum Likelihood Estimation* (MLE).

- d) Melakukan pengujian parameter pada model tentatif dengan uji t. Uji t merupakan uji signifikansi parameter model ARIMA yang dilakukan terhadap setiap parameter model secara parsial. Uji ini dilakukan untuk melihat nilai dan besarnya pengaruh parameter pada model. Uji ini menunjukkan kebaikan model ARIMA (p, d, q) berdasarkan pengaruh setiap parameternya (Montgomery *et al.* 2015).

Hipotesis yang digunakan adalah:

$H_0: \beta_j = 0$, parameter tidak berpengaruh signifikan terhadap model

$H_1: \beta_j \neq 0$, parameter berpengaruh signifikan terhadap model

Statistik uji:

$$t_0 = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)}$$

Keterangan:

$\hat{\beta}_j$ estimator dari β_j

$se(\hat{\beta}_j)$ standard error dari $\hat{\beta}_j$.

Daerah kritis:

Tolak H_0 jika $|t_0| > t_{\alpha/2, n-p}$ atau nilai $p < \alpha$ (0,05).

- e) Memilih model tentatif ARIMA berdasarkan nilai *Akaike Information Criterion* (AIC) terkecil dan hasil pengujian pada poin (d). AIC diformulasikan dalam Montgomery *et al.* (2015) sebagai berikut:

$$AIC = \ln \left(\frac{\sum_{t=1}^T e_t^2}{T} \right) + \frac{2P}{T}$$

Keterangan:

T : Periode data deret waktu

$\sum_{t=1}^T e_t^2$: Jumlah kuadrat sisaan

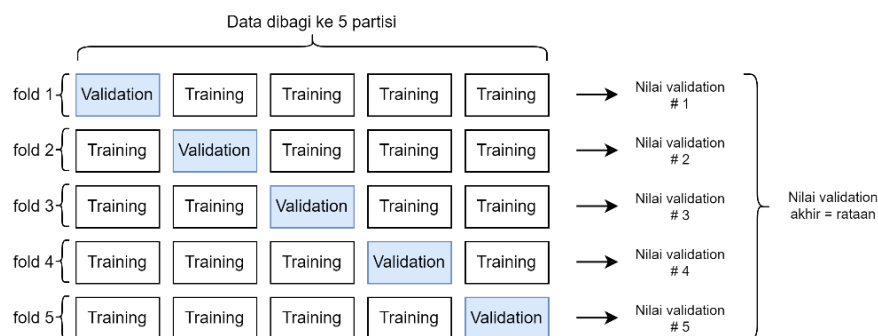
P : Jumlah parameter dalam model

- f) Melakukan uji diagnostik pada model yang diperoleh pada poin (e)
g) Melakukan *overfitting* pada model tentatif ARIMA dengan nilai AIC terkecil.
h) Memilih model terbaik dengan membandingkan model tentatif ARIMA dan model *overfitting* menggunakan kriteria nilai AIC dan hasil signifikansi parameter.

4. Pemodelan LSTM

- a) Melakukan normalisasi data menggunakan MinMaxScaler pada *range scale* (0, 1).
b) Mentransformasi data ke bentuk *supervised learning*.
c) Mengubah data ke bentuk 3 dimensi yang memuat [*samples, time steps, features*]. Data latih (*train*) dan data uji (*test*) diubah menggunakan fungsi *numpy.reshape()* agar memiliki bentuk (*trainX.shape[0], 1, trainX.shape[1]*) dan (*testX.shape[0], 1, testX.shape[1]*).
d) Mendefinisikan fungsi untuk membangun model LSTM dengan memuat LSTM dan *Dense Layer*, neuron, *input*, *loss function* berupa *mean squared error*, *optimizer* berupa ADAM serta fungsi aktivasi berupa Sigmoid dan Tanh.
e) Melakukan *hyperparameter tuning* pada model menggunakan GridSearchCV yaitu metode *grid search* dengan *k-fold cross-validation*

untuk mengevaluasi performa model. Dengan metode ini data dibagi menjadi k partisi dengan ukuran yang sama. Untuk setiap $fold\ i$, model dilatih pada partisi $k - 1$ dan evaluasi dilakukan pada partisi i . Nilai validasi diperoleh dari rata-rata nilai validasi semua $fold$ yang terbentuk (Chollet 2021). Skema k -fold cross-validation dengan $k = 5$ terlihat pada Gambar 3. *Hyperparameter* yang di-*tuning* adalah *epochs*, *batch size* dan *learning rate*. Kombinasi *hyperparameter* terbaik adalah kombinasi yang memberikan *validation loss* atau kesalahan terkecil.



Gambar 3 k -fold cross validation dengan $k = 5$

- f) Membangun kembali model dengan data latih dan *hyperparameter* terbaik dari hasil *tuning*.
- g) Melakukan prediksi terhadap data latih dan data uji kemudian melakukan denormalisasi data hasil prediksi.
5. *Hybrid ARIMA – LSTM*
 - a) Melakukan prediksi terhadap data latih dan data uji dengan model ARIMA terbaik.
 - b) Menghitung sisaan dari prediksi model ARIMA terbaik dengan formula:
$$\varepsilon_{(L)t} = x_t - L_t$$

Keterangan:

x_t : data aktual pada waktu ke t

L_t : data hasil prediksi pada waktu ke t
 - c) Melakukan pemodelan LSTM dengan input berupa sisaan dari prediksi model ARIMA terbaik ($\varepsilon_{(L)t}$)
 - d) Menjumlahkan hasil prediksi model ARIMA dengan hasil prediksi model LSTM data sisaan ARIMA sebagai hasil prediksi final model *Hybrid*.
6. Memilih model terbaik hasil pada poin (3), (4) dan (5) berdasarkan hasil evaluasi data uji dengan kriteria berupa nilai *Root Mean Squared Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE) terkecil (Fan et al. 2021). RMSE dan MAPE dapat diformulasikan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_t - w_t)^2}$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_t - w_t}{x_t} \right| 100\%$$

Keterangan:

n : jumlah sampel data yang digunakan

x_t : data aktual pada waktu ke t

w_t : data hasil prediksi pada waktu ke t

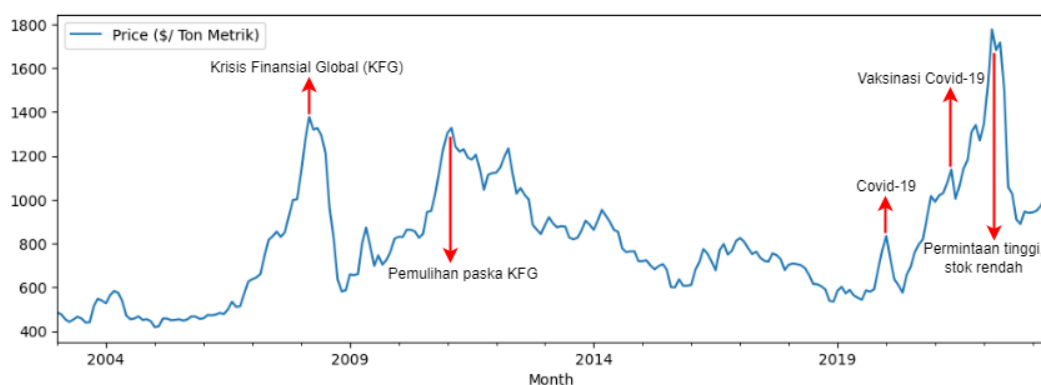
7. Melakukan peramalan harga CPO dengan model terbaik.

- Hak Cipta Dilindungi Undang-undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

IV HASIL DAN PEMBAHASAN

4.1 Eksplorasi Data

Harga *crude palm oil* dunia dengan periode bulanan dari Januari 2003 sampai April 2023 mengalami fluktuasi terlihat pada Gambar 4. Harga CPO terbilang cukup rendah pada tahun 2003 s.d 2006. Harga CPO kemudian naik secara perlahan hingga mencapai nilai tertinggi pada Maret 2008 yaitu \$ 1377,22. Namun kemudian harga CPO sempat turun drastis ke harga \$ 581,25 pada November 2008. Selain itu harga CPO juga cukup stabil dan tinggi sejak akhir tahun 2010 hingga *quarter* tiga tahun 2012 dimana harga selalu lebih dari \$ 1000. Kewinoto *et al.* (2015) dalam penelitiannya menyebutkan bahwa fluktuasi harga yang terjadi pada tahun 2008 hingga 2012 disebabkan oleh Krisis Finansial Global (KFG) yang menimbulkan ketidakpastian bagi pengembangan industri kelapa sawit. Di awal tahun 2008 harga CPO melambung ketika krisis dimulai. Harga CPO kemudian turun drastis di akhir tahun 2008 karena negara pengimpor menunda bahkan membatalkan kesepakatan impor (Dradjat 2011). Go dan Lau (2020) berdasarkan Oil World (2010) menjelaskan bahwa penurunan harga CPO pada tahun 2008 sejalan dengan penurunan pertumbuhan konsumsi CPO dari 9,8% pada 2007/2008 menjadi 8,5% pada tahun 2008/2009. Pertumbuhan ini selanjutnya menurun menjadi 4,5% pada 2009/2010. Hal ini karena di beberapa negara berkembang masyarakat berusaha mengurangi konsumsi makanan yang dianggap mewah seperti produk dengan minyak nabati. Selanjutnya di akhir tahun 2010 sampai 2012 harga CPO cukup tinggi karena sedang masa pemulihan kondisi ekonomi pada setiap negara di dunia. Setelahnya harga CPO dunia berfluktuasi namun tidak pernah menyentuh angka \$ 500 lagi hingga Oktober 2018.



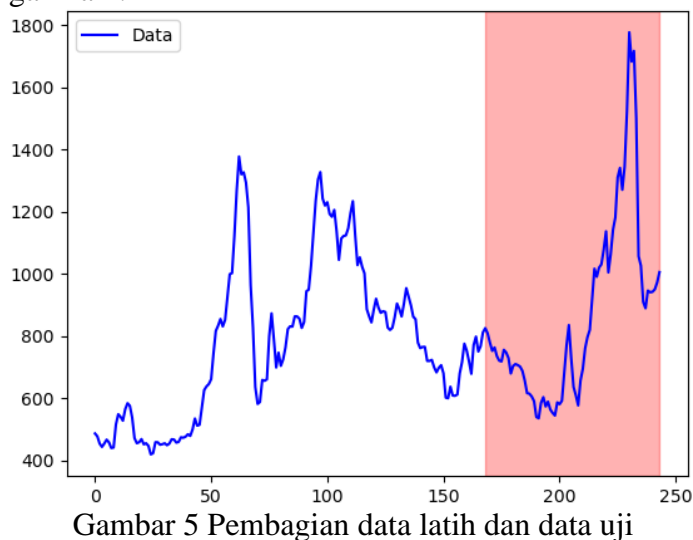
Gambar 4 Grafik fluktuasi harga CPO dunia (Januari 2003 – April 2023)

Harga CPO dunia terus berada pada rentang \$ 500 - \$ 800 dan mengalami fluktuasi pada akhir 2019 s.d. pertengahan 2020. Hal ini disebabkan oleh pandemi Covid-19. Pada awal pandemi permintaan terhadap CPO tinggi sedangkan ketersediaan rendah sehingga terjadi kenaikan harga CPO. Namun kemudian China sebagai salah satu pengimpor CPO terbesar di dunia melakukan *lockdown* sehingga harga CPO kembali turun. Harga CPO mulai meningkat lagi pada *quarter* tiga tahun 2020 karena sudah mulai dilaksanakan vaksin Covid-19 di berbagai negara di dunia

(Ashaari *et al.* 2022). Harga CPO dunia tertinggi selama 20 tahun terakhir terjadi pada bulan Maret 2022 dengan nilai jual \$ 1776,96 per ton metrik CPO. Hal ini terjadi karena Indonesia sebagai negara pengekspor CPO terbesar di dunia membatasi ekspor CPO sehingga persediaan CPO rendah namun permintaan terhadap CPO yang tinggi menyebabkan kenaikan harga CPO yang tinggi (Llewellyn 2022).

4.2 Pembagian Data

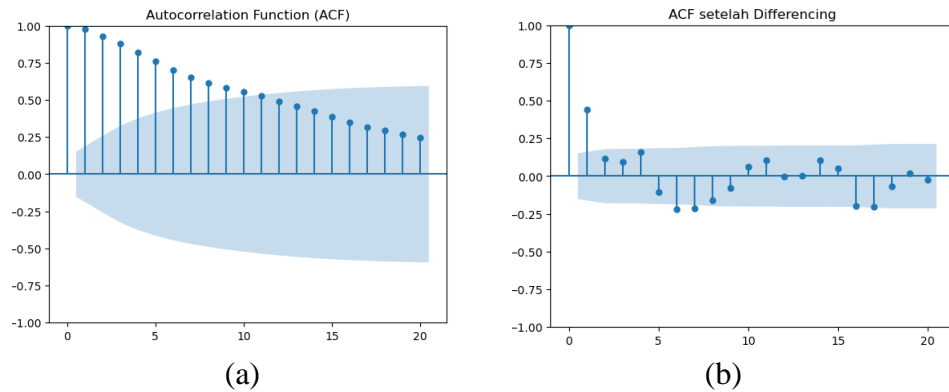
Data yang digunakan merupakan data bulanan sejak Januari 2003 s.d April 2023 (244 amatan). Data bulan Januari tahun 2003 s.d Desember 2016 merupakan data latih (168 amatan) dan digunakan untuk membangun model sedangkan data bulan Januari 2017 s.d April 2023 merupakan data uji (76 amatan) dan digunakan untuk menguji kebaikan model. Pembagian data didasarkan pada kemiripan pola antara data latih dan data uji seperti terlihat pada Gambar 5. Data pada bagian awal terlihat lebih rendah dari bagian lainnya, lalu naik secara perlahan hingga mencapai titik tertinggi. Data pada bagian akhir terlihat sama-sama tidak terlalu tinggi namun ada kecenderungan naik.



Gambar 5 Pembagian data latih dan data uji

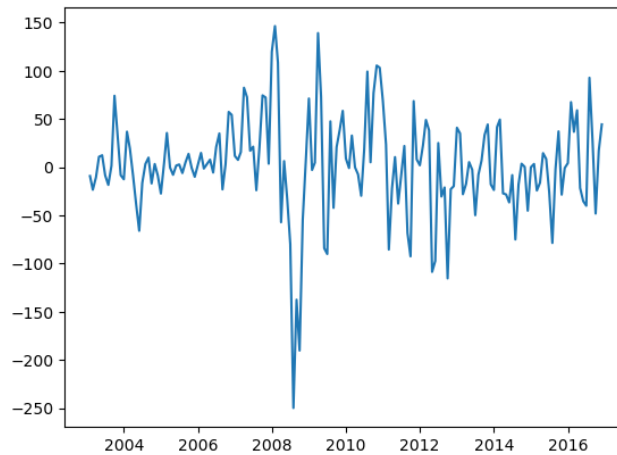
4.3 Pemodelan ARIMA

Pemeriksaan kestasioneran terhadap data *crude palm oil* dunia dengan plot ACF menunjukkan data tidak stasioner. Hal ini terlihat pada Gambar 6(a) yang menunjukkan kondisi garis cenderung menurun secara perlahan mendekati nol dan tidak terpotong (*cut-off*) setelah beberapa *lag*. Pernyataan ini diperkuat dengan hasil uji formal ADF yang menunjukkan nilai p 0,235. Nilai ini lebih besar dari $\alpha = 5\%$, dengan kata lain tak tolak H_0 atau data diketahui tidak stasioner pada taraf nyata 5%. Pembedaan atau *differencing* dilakukan untuk menangani ketidakstasioneran data harga CPO dunia tersebut. Data setelah satu kali pembedaan seperti pada Gambar 6(b) terlihat sudah tidak ada kecenderungan menurun dan terpotong setelah beberapa *lag* saja. Hal ini diperkuat dengan nilai p hasil uji formal ADF sebesar $3,04 \times 10^{-7}$ atau lebih kecil dari $\alpha = 5\%$ (tolak H_0) menunjukkan kondisi sudah stasioner. Hal ini sesuai dengan pernyataan Montgomery *et al.* (2015) bahwa pembedaan satu kali sudah cukup untuk membuat data stasioner dalam rata-rata.



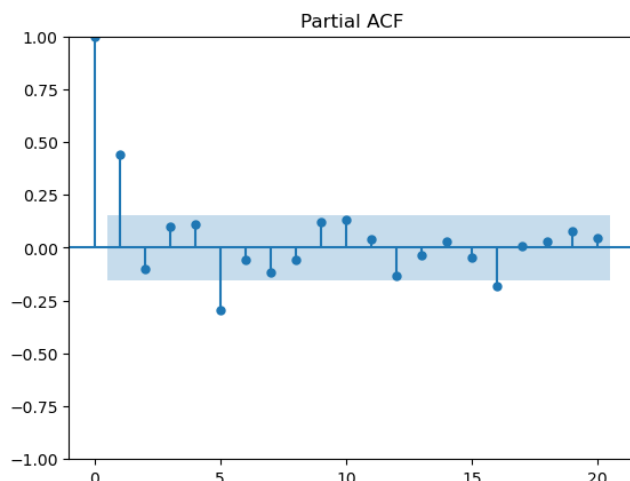
Gambar 6 Plot ACF harga CPO dunia (a) sebelum pembedaan dan (b) setelah pembedaan satu kali ($d=1$)

Gambar 7 menunjukkan plot deret waktu dari data harga *crude palm oil* yang telah stasioner dalam rataannya setelah dilakukan pembedaan pertama dengan rata-rata \$ 1,95 per ton metrik.



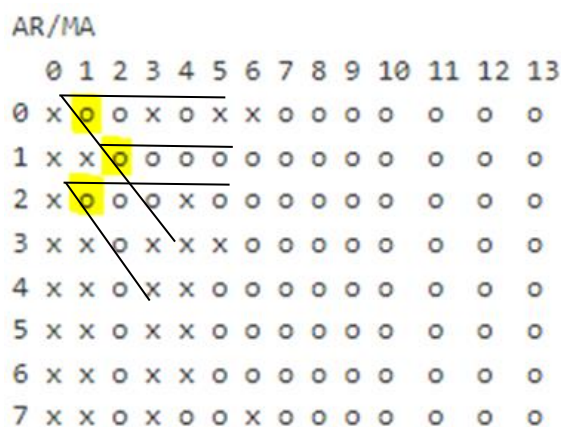
Gambar 7 Plot deret waktu harga CPO setelah pembedaan satu kali ($d=1$)

Identifikasi kandidat model ARIMA dilakukan berdasarkan plot ACF (Gambar 6b) dan PACF setelah *differencing* (Gambar 8). Plot ACF dan PACF pada *python* menunjukkan garis tegak ke atas dimulai dari *lag* 0 sehingga kondisi data dilihat dari garis kedua (*lag* 1). Garis horizontal biru menunjukkan batas signifikansi, dimana ketika pada *lag* tertentu data terpotong oleh garis tersebut maka diketahui data pada waktu t dipengaruhi oleh data pada *lag* tersebut (plot PACF) atau sisaan data pada *lag* tersebut (plot ACF). Plot ACF terlihat tidak lagi terpotong oleh garis horizontal biru setelah *lag* 1 (*cut off* after *lag* 1) sehingga sesuai dengan ciri identifikasi model pada Tabel 1 maka model yang terbentuk adalah model MA (1) atau ARIMA (0, 1, 1). Plot PACF juga menunjukkan kondisi tidak lagi terpotong oleh garis horizontal biru (*cut off*) setelah *lag* 1 sehingga model lain yang terbentuk adalah AR (1) atau ARIMA (1, 1, 0).



Gambar 8 Plot PACF setelah pembedaan satu kali ($d=1$)

Identifikasi model tambahan dilakukan dengan plot *Extended Sample ACF* (ESACF). Berbeda dengan teori identifikasi model ARIMA berdasarkan plot ESACF pada Gambar 2 yang menunjukkan kondisi segitiga nol yang sempurna, identifikasi menggunakan data harga CPO tidak menunjukkan plot segitiga “nol” asimtotik yang sempurna (Gambar 9). Titik sudut segitiga “nol” dari plot tidak dapat dipastikan. Kondisi ini sering terjadi ketika identifikasi dilakukan pada data aktual. Oleh karena itu, setiap kemungkinan titik sudut segitiga “nol” yang terletak di bagian pojok kiri atas plot ESACF diasumsikan sebagai model tentatif yang terbentuk (Chenoweth *et al.* 2000). Berdasarkan kriteria plot ESACF, tidak dapat dipastikan titik sudut terbaik terletak pada ARMA (0,1), ARMA (1, 2) atau ARMA (2,1) sehingga diketahui model tentatif yang terbentuk adalah ARIMA (0, 1, 1), ARIMA (1, 1, 2) dan ARIMA (2, 1, 1).



Gambar 9 Identifikasi model tentatif dengan plot ESACF

Metode AIC yang didasarkan pada *Maximum Likelihood Estimation* digunakan untuk memilih model ARIMA terbaik. Model terbaik yang dipilih adalah model dengan nilai AIC terkecil (Fathurahman 2010). Tabel 2 menunjukkan bahwa model tentatif ARIMA dengan nilai AIC terkecil (1759,9291) adalah ARIMA (1, 1, 2). Semua parameter model ini juga signifikan pada taraf nyata 5%. Selanjutnya dilakukan uji diagnostik pada model ARIMA (1, 1, 2).

Tabel 2 Nilai pendugaan parameter model tentatif ARIMA

Model ARIMA	Parameter	Koefisien Parameter	Nilai-p	AIC
ARIMA (0, 1, 1)	MA(1)*	0,4196	0,0000	1766,8073
ARIMA (1, 1, 0)	AR (1)*	0,4430	0,0000	1765,5759
ARIMA (1, 1, 2)	AR (1)*	-0,6459	0,0000	1759,9291
	MA(1)*	1,2012	0,0000	
	MA(2)*	0,5093	0,0000	
ARIMA (2, 1, 1)	AR (1)*	0,5489	0,0000	1769,4085
	AR (2)*	0,4362	0,0000	
	MA(1)*	0,9952	0,0000	

*Parameter yang signifikan pada taraf nyata 5%

Uji kebebasan antar sisaan dengan Ljung-Box menghasilkan nilai p sebesar 0,84; lebih besar dari $\alpha = 5\%$ sehingga tak tolak H_0 atau sisaan diketahui saling bebas. Uji kenormalan sisaan dengan uji Jarque Bera menghasilkan nilai p sebesar 0,00 atau lebih kecil dari $\alpha = 5\%$ sehingga tolak H_0 atau sisaan tidak normal. Lestari *et al.* (2022) berdasarkan Anderson *et al.* (2011) menyebutkan bahwa untuk penggunaan data lebih dari 30 amatan, ketidaknormalan sebaran sisaan dapat ditoleransi sesuai dengan teorema limit pusat yang berbunyi “suatu distribusi dapat didekati dengan distribusi normal apabila ukuran datanya lebih besar atau sama dengan 30.”

Setelah diperoleh model tentatif ARIMA dengan nilai AIC terkecil akan dilakukan *overfitting*. *Overfitting* adalah proses menambahkan sejumlah nilai parameter (p) pada proses AR(p) atau parameter (q) pada proses MA(q). Hal ini dilakukan untuk mengidentifikasi model ARIMA (p, d, q) lain yang lebih baik dari model tentatif terbaik (Mustapa dan Ismail 2019).

Tabel 3 Nilai pendugaan parameter model *Overfitting*

Model ARIMA	Parameter	Koefisien Parameter	Nilai-p	AIC
ARIMA (2,1,2)	AR (1)*	-0,6282	0,0002	1761,1984
	AR (2)	-0,1647	0,1866	
	MA(1)*	1,1645	0,0000	
	MA(2)*	0,5960	0,0000	
ARIMA (1,1,3)	AR (1)*	-0,5288	0,0120	1761,5462
	MA(1)*	1,0735	0,0000	
	MA(2)*	0,4076	0,0059	
	MA(3)	-0,0634	0,4987	

*Parameter yang signifikan pada taraf nyata 5%

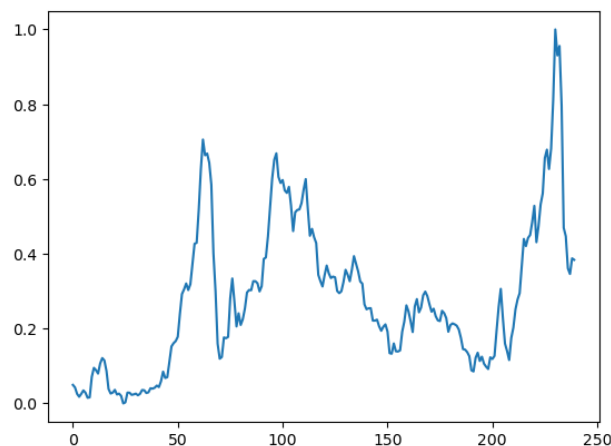
Hasil *overfitting* sesuai Tabel 3 menunjukkan bahwa model tentatif ARIMA dengan nilai AIC terkecil yaitu ARIMA (1, 1, 2) merupakan model terbaik untuk data harga *crude palm oil* dunia karena semua parameternya signifikan dan nilai AIC terkecil. Formula ARIMA (1, 1, 2) dituliskan sebagai berikut:

$$L_t = -0,6459L_{t-1} + 1,2012\varepsilon_{(L)t-1} + 0,30\varepsilon_{(L)t-2} + \varepsilon_{(L)t}$$

Pemodelan dan analisis data dengan metode ARIMA menggunakan *python* dapat dilihat pada Lampiran 1.

4.4.1 Data Preparation

Pemodelan dengan LSTM tidak memerlukan asumsi kestasioneran data seperti pada ARIMA. Hal ini karena sebelum pemodelan LSTM, data dinormalisasi menggunakan MinMaxScaler ke dalam *range* (0,1). Normalisasi data membantu meningkatkan kinerja model dalam menyesuaikan bobot pada model (Brownlee 2017). Plot data harga CPO setelah dinormalisasi terlihat pada Gambar 10.



Gambar 10 Plot data setelah normalisasi

Selanjutnya data diubah ke bentuk *supervised learning* atau data berlabel. Data pada waktu t dilabelkan sebagai data prediksi ($Y = X_t$) sedangkan data pada waktu sebelumnya (X_{t-1}) merupakan *input*. Williams dan Zweig (2016) menyebutkan bahwa bentuk *supervised learning* membantu model mempelajari pola data dan hubungan antara *input* dengan *output* untuk memprediksi data selanjutnya sehingga menghindari performa *out-of-the-box* yang buruk. Data kemudian diubah ke bentuk *array* 3 dimensi sesuai dengan LSTM modelling dan *deep learning* pada umumnya dimana tipikal data inputnya adalah *3-dimensional array* yaitu [*number of samples, time steps, number of features*] (Phaladisailoed dan Numnonda 2018). Pada penelitian ini data latih dan data uji diubah menggunakan fungsi *numpy.reshape()* agar memiliki bentuk (168, 1, 1) dan (76, 1, 1).

4.4.2 Membangun dan Memilih Model Terbaik

Tabel 4 Arsitektur dan *hyperparameter* model LSTM

Karakteristik	Spesifikasi
Arsitektur	3 Layer LSTM dengan 4 unit neuron 1 Layer Dense
Optimizer	ADAM (Adaptive Momentum)
Loss Function	Mean Squared Error
Batch size	1, 32
Epoch	1, 100
Learning Rate	0,001; 0,005

Dimodifikasi dari: Brownlee (2017), Fan *et al.* (2021), Siامي-Namini *et al.* (2018)

Arsitektur LSTM model yang digunakan pada penelitian ini terdiri dari tiga *layer* LSTM dengan empat unit *neuron* dan satu *layer Dense*. *Optimizer* dan *loss function* yang digunakan untuk membantu konvergensi model adalah ADAM dan *mean squared error* (mse). *Optimizer* ADAM dengan *loss function* mse merupakan algoritma yang umum digunakan pada pemodelan LSTM. ADAM (*Adaptive Momentum*) menggabungkan kelebihan dari RMSProp yaitu menggunakan kuadrat gradien untuk menyesuaikan *learning rate* dan kelebihan SGD (*Stochastic Gradient Descent*) dengan momentum yaitu memanfaatkan momentum dengan menggunakan *exponentially weighted moving average* gradien daripada gradien itu sendiri sehingga ADAM bekerja dengan baik sebagai *optimizer* (Naeem dan Maged 2020).

LSTM (maupun metode *deep learning* lainnya) menggunakan *hyperparameter* yang tidak dipelajari langsung oleh algoritma untuk menemukan nilai terbaik bagi model. Salah satu cara untuk menemukan kombinasi *hyperparameter* terbaik bagi model yaitu dengan *hyperparameter tuning*. Beberapa nilai dari setiap *hyperparameter* dimasukkan ke *dictionary* lalu di-*tuning* dengan metode tertentu untuk menemukan kombinasi *hyperparameter* yang paling optimal terhadap model (Goodfellow *et al.* 2016). Beberapa *hyperparameter* yang akan di-*tuning* adalah *batch size* (1, 32), *epoch* (1, 100) dan *learning rate* (0,001; 0,005) (Brownlee 2017, Siarni-Namini *et al.* 2018, Fan *et al.* 2021). Ketiga *hyperparameter* tersebut dipilih untuk di-*tuning* karena secara langsung mempengaruhi kemampuan dan kecepatan model untuk mencapai akurasi yang optimal. Penelitian ini menggunakan *grid search* yang terdapat pada package *scikit-learn* yaitu GridSearchCV dengan 10-fold *cross validation* untuk melakukan *hyperparameter tuning*. Ahmad *et al.* (2022) menyampaikan bahwa model dengan penggunaan GridSearchCV memberikan akurasi yang lebih tinggi dibanding model yang tidak menggunakan GridSearchCV. *k-fold cross validation* umumnya menggunakan 3, 5, atau 10-fold. Varma dan Simon (2006) menyarankan penggunaan *k-fold* yang besar untuk data dengan ukuran yang kecil (< 1000) sehingga pada penelitian ini *k* yang digunakan adalah 10 dimana sembilan partisi pada setiap *fold* digunakan untuk *training* dan satu partisi untuk *validation*. *Hyperparameter tuning* menunjukkan hasil kombinasi *hyperparameter* yang paling baik untuk pemodelan dengan LSTM Network adalah *epochs* 100, *batch size* 1 dan *learning rate* 0,001. Model ini menghasilkan rata-rata nilai *loss* atau kesalahan sebesar 0,0015.

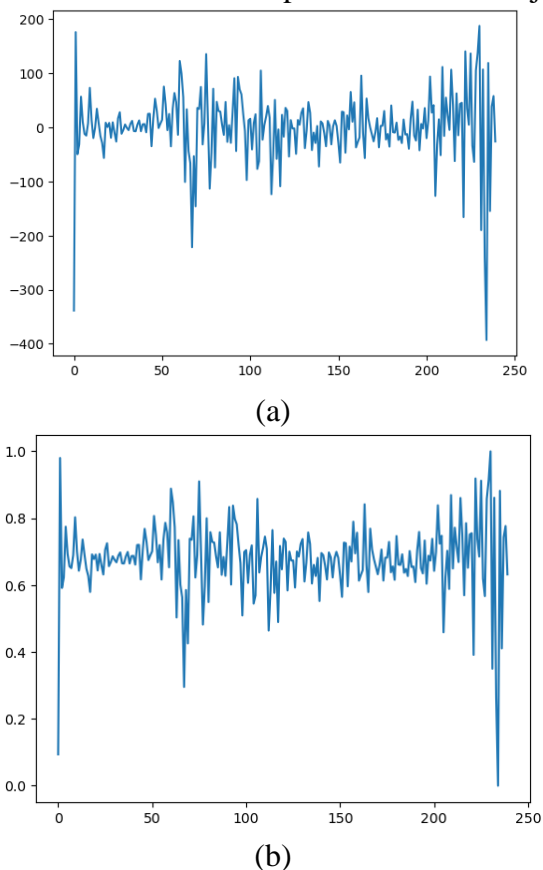
Tabel 5 Nilai rata-rata *loss* kombinasi *hyperparameter*

Model	Batch Size	Epoch	Learning Rate	Rata-rata loss
1	1	1	0,001	0,0294
2			0,005	0,0281
3		100	0,001	0,0015
4			0,005	0,0017
5	32	1	0,001	0,0993
6			0,005	0,0917
7		100	0,001	0,0140
8			0,005	0,0016

Setelah diperoleh kombinasi *hyperparameter* terbaik kemudian model dibangun kembali dan dilatih dengan kombinasi tersebut menggunakan data latih. Model dilatih dengan tujuan mengurangi tingkat kesalahan (*loss*) saat melakukan prediksi karena terus menyesuaikan dan memperbarui bobot dan bias pada model hingga optimal (Chollet 2021). Langkah berikutnya adalah melakukan prediksi terhadap data uji dengan menggunakan model LSTM. Hasil prediksi lalu diubah ke *range* asli data (denormalisasi) untuk bisa dievaluasi. Pemodelan dan analisis data dengan metode LSTM menggunakan *python* dapat dilihat pada Lampiran 2.

4.5 Hybrid ARIMA-LSTM

Peramalan data dengan metode *Hybrid ARIMA-LSTM* diawali dengan pemodelan ARIMA (1, 1, 2) pada langkah 4.3. Selanjutnya sisaan hasil prediksi dengan metode ARIMA (1, 1, 2) yaitu $(\varepsilon_{(L)}t)$ dijadikan suatu dataset baru. Pemodelan LSTM seperti pada langkah 4.4 kemudian diterapkan pada dataset baru tersebut. Gambar 11 menunjukkan data sisaan prediksi metode ARIMA (1, 1, 2) sebelum dan sesudah dinormalisasi untuk proses LSTM selanjutnya.



Gambar 11 Plot data sisaan ARIMA (a) sebelum dinormalisasi dan (b) setelah dinormalisasi

Pembangunan model dan *hyperparameter tuning* dengan arsitektur dan *hyperparameter* seperti Tabel 4 diterapkan pada data sisaan ARIMA. Hasil kombinasi *hyperparameter* yang paling baik untuk pemodelan dengan LSTM *Network* pada data sisaan ARIMA adalah *epochs* 100, *batch size* 1 dan *learning rate* 0.005 dengan rata-rata nilai *loss* atau kesalahan sebesar 0,0077.

Tabel 6 Nilai rata-rata *loss* kombinasi *hyperparameter* untuk sisaan ARIMA

Model	Batch Size	Epoch	Learning Rate	Rata-rata loss
1	1	1	0,001	0,0146
2			0,005	0,0081
3		100	0,001	0,0080
4			0,005	0,0077
5	32	1	0,001	0,1084
6			0,005	0,0959
7		100	0,001	0,0082
8			0,005	0,0078

Model terbaik digunakan untuk melakukan prediksi dan setelahnya data didenormalisasi untuk memperoleh hasil yang sebenarnya dengan sisaan U_t . Hasil prediksi pemodelan ARIMA sebelumnya (\hat{L}_t) dan pemodelan LSTM dataset sisaan ARIMA ($\hat{\epsilon}_{(L)t}$) dijumlahkan untuk menghasilkan nilai prediksi final dari pemodelan *Hybrid* ARIMA-LSTM dengan formula sebagai berikut:

$$Y = \hat{L}_t + \hat{\epsilon}_{(L)t} + U_t$$

Pemodelan dan analisis data dengan metode *Hybrid* ARIMA-LSTM menggunakan *python* dapat dilihat pada Lampiran 3.

4.6 Evaluasi Model

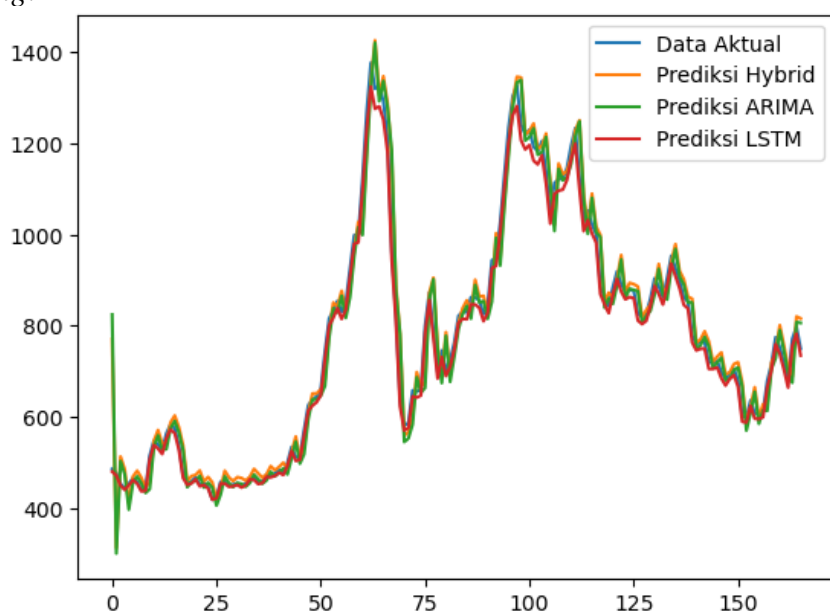
Pada tahap ini hasil prediksi dari model ARIMA, LSTM dan *Hybrid* ARIMA-LSTM dievaluasi terhadap data aktual menggunakan MAPE dan RMSE. RMSE digunakan untuk mengevaluasi kebaikan model dengan menghitung selisih nilai prediksi dan nilai sebenarnya. RMSE memberikan bobot kesalahan yang lebih tinggi untuk hasil prediksi dengan *error* lebih besar. RMSE tidak memiliki nilai maksimal atau minimal, namun dalam perbandingan model diketahui model semakin baik jika memiliki nilai RMSE lebih kecil. MAPE digunakan untuk mengevaluasi model dengan menunjukkan tingkat kesalahan atau selisih nilai prediksi dan nilai sebenarnya dalam bentuk persentase sehingga lebih mudah dipahami. Nilai MAPE dibawah 10% menunjukkan model memiliki kemampuan peramalan sangat baik (Hutasuhut *et al.* 2014). Berdasarkan hasil evaluasi seperti pada Tabel 7 diketahui model yang paling baik adalah LSTM. Model tersebut menghasilkan nilai RMSE data uji sebesar 34,708 dan MAPE data uji sebesar 2,33% atau masuk ke kategori sangat baik untuk peramalan.

Tabel 7 Evaluasi model

Model	RMSE	RMSE – P	MAPE – P (%)
ARIMA	56,469	84,840	5,71
LSTM	17,850	34,708	2,33
<i>Hybrid</i> ARIMA-LSTM	53,406	76,830	5,59

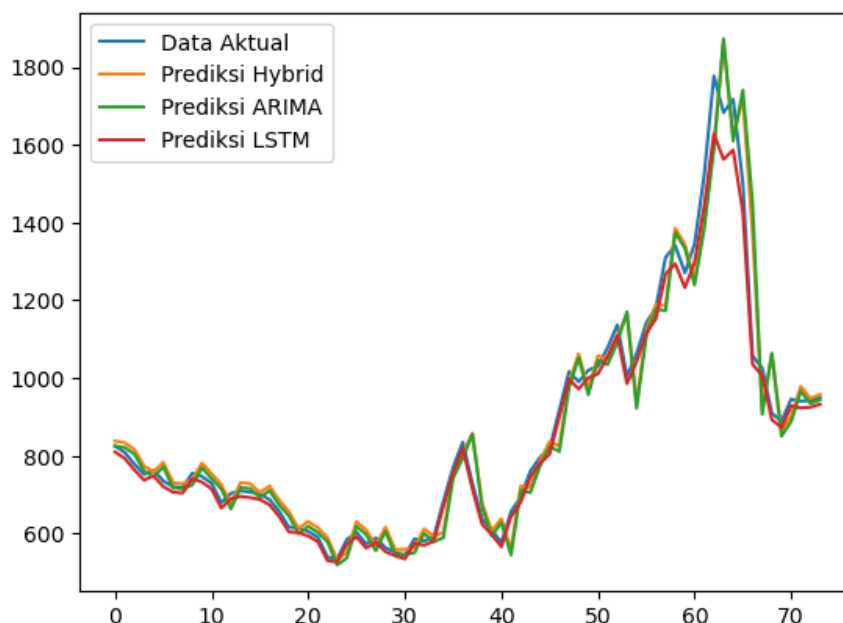
*RMSE: evaluasi model terhadap data latih; RMSE-P dan MAPE-P: evaluasi model terhadap data uji

Perbandingan hasil evaluasi model ARIMA, LSTM, dan *Hybrid* ARIMA-LSTM juga terlihat pada Gambar 12 dan Gambar 13. Gambar 12 menunjukkan perbandingan data latih aktual dengan hasil prediksi data latih dari ketiga model. Pelatihan model membutuhkan waktu cukup lama tergantung data dan kompleksitas model. Hal ini dapat menyebabkan peristiwa *overfitting*. Salah satu cara mengidentifikasi adanya *overfitting* pada model yaitu dengan melakukan prediksi dan evaluasi model terhadap data latih. Apabila hasil prediksi baik maka diketahui tidak terjadi *overfitting* pada model (Brownlee 2017). Berdasarkan plot perbandingan pada Gambar 12 diketahui data latih hasil prediksi ketiga model berada cukup dekat dengan data latih aktual. Hal ini juga didukung dengan nilai RMSE data latih pada Tabel 7 yang sangat kecil sehingga diketahui ketiga model bekerja dengan baik untuk memprediksi data latih atau tidak ada peristiwa *overfitting*.



Gambar 12 Plot perbandingan data latih dengan data prediksi

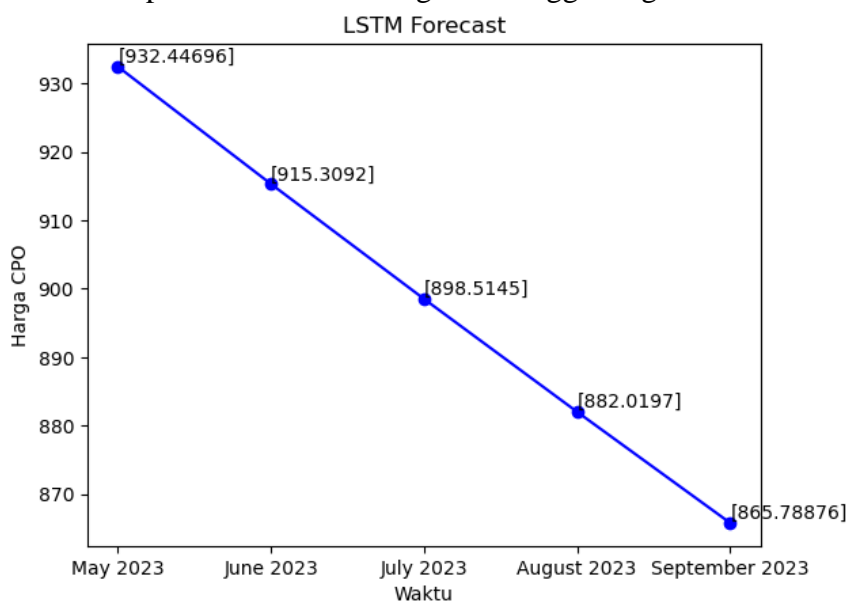
Gambar 13 menunjukkan perbandingan data uji aktual dengan hasil prediksi data uji dari ketiga model. Data hasil prediksi cukup dekat dengan data aktual sehingga ketiga model diketahui bekerja sangat baik untuk peramalan. Meski demikian, plot LSTM terlihat lebih mulus dibandingkan metode ARIMA dan *Hybrid* ARIMA-LSTM. Plot LSTM juga terlihat lebih stabil dalam mengikuti pola data aktual dibanding dua metode lainnya. Hal ini didukung dengan nilai RMSE dan MAPE data uji metode LSTM yang paling kecil sesuai evaluasi pada Tabel 7 menunjukkan bahwa metode LSTM lebih baik untuk peramalan harga CPO dunia dibanding metode ARIMA dan *Hybrid* ARIMA-LSTM.



Gambar 13 Plot perbandingan data uji dengan data prediksi

4.7 Peramalan

Peramalan harga CPO periode Mei 2023 hingga September 2023 dilakukan dengan metode LSTM. Hasil peramalan harga CPO dunia terlihat pada Gambar 14. Harga CPO hasil peramalan memiliki kecenderungan menurun dengan harga tertinggi terjadi pada bulan Mei 2023 yaitu senilai \$ 932,447/ ton metrik. Kecenderungan ini sesuai dengan prediksi dari Fitch Ratings (2022) yang menyebutkan bahwa harga CPO berpotensi turun di 2023. Hal ini dikarenakan kondisi cuaca yang menguntungkan dan peningkatan ketenagakerjaan asing di salah satu negara penghasil CPO terbesar di dunia yaitu Malaysia. Dengan demikian, supply dan volume produksi CPO meningkat sehingga harga CPO menurun.

Gambar 14 Peramalan harga *crude palm oil* dunia

V SIMPULAN DAN SARAN

5.1 Simpulan

Metode LSTM lebih baik digunakan untuk peramalan harga CPO dibanding dua metode lainnya yaitu ARIMA dan *Hybrid* ARIMA-LSTM. Model LSTM terbaik adalah model dengan *hyperparameter epoch* 100, *batch size* 1, dan *learning rate* 0,001. Hal ini karena metode ini dapat mempelajari *trend* dan pola data dengan baik. Berdasarkan hasil evaluasi data prediksi terhadap data aktual diperoleh nilai MAPE metode LSTM sebesar 2,33% atau masuk ke kategori sangat baik untuk peramalan data harga *crude palm oil* dunia.

5.2 Saran

Metode tertentu seperti *Cross Validation* dan *Walk-Forward Validation* perlu digunakan untuk pembagian data latih dan data uji yang lebih tepat sehingga model lebih representatif terhadap data. Hal ini mempengaruhi hasil peramalan dan kebaikan model. Penerapan ketiga metode ini cukup memakan waktu sehingga diperlukan perbandingan peramalan harga CPO dengan metode *machine learning* lainnya seperti Prophet dan XBoost untuk menemukan metode yang lebih baik dan efisien. Penerapan metode ARIMA dan LSTM untuk data *multivariate* perlu dilakukan karena ada faktor selain data harga CPO sendiri yang mempengaruhi perubahan harga CPO. Penerapan metode tersebut dapat menghasilkan peramalan harga CPO yang lebih akurat dan relevan.

DAFTAR PUSTAKA

- Abassi AR, Mahmoudi MR, Arefi MM. 2021. Transformer winding faults detection based on time series analysis. *IEEE Transactions on Instrumentation and Measurement*. 70:1-10.
- Ahmad GN, Fatima H, Ullah S dan Saidi AS. 2022. Efficient medical diagnosis of human heart diseases using machine learning techniques with and without GridSearchCV. *IEEE Access*. 10: 80151-80173.
- Aktivani S. 2021. Uji stasioneritas data inflasi kota Padang periode 2014-2019. *Jurnal Statistika Industri dan Komputasi*. 6(1): 26-33.
- Alencar VA, Pessamilio LR, Rooke F, Bernardino HS, Vieira AB. 2021. Forecasting the car sharing service demand using uni and multivariable models. *Journal of Internet Services and Applications*. 12(1): 1-20.
- Ali PJM, Faraj RH. 2014. Data normalization and standardization: a technical report. *Mach Learn Tech Rep*. 1(1): 1-6.
- Anwar S. 2017. Peramalan suhu udara jangka pendek di Kota Banda Aceh dengan metode Autoregressive Integrated Moving Average (ARIMA). *Malikussaleh Journal of Mechanical Science and Technology*. 5(1): 6-12.
- Ariff NM, Zamhawari NH, Abu Bakar MA. 2015. Time series ARIMA models for daily price of palm oil. *AIP Conference Proceedings*. 1643(1): 283-288.
- Arifin B, Tanaya IGLP, Usman A. 2021. Peramalan harga kelapa sawit dunia pada tahun 2020-2024. *Prosiding SAINTEK*. 3(1): 349-368.
- Ashaari A, Abdullah M, dan Fuzi NM. 2022. Malaysian palm oil industry performance during epidemic Covid 19. *International Journal of Academic Research in Business and Social Sciences*. 12(1): 622-628.
- [BPS] Badan Pusat Statistik. 2022. Statistik Kelapa Sawit Indonesia 2021. Jakarta: Badan Pusat Statistik.
- Brownlee J. 2017. *Long short-term memory networks with python: develop sequence prediction models with deep learning*. Machine Learning Mastery.
- Chatfield C. 2003. *The Analysis of Time Series: An Introduction 6th ed*. Boca Raton (FL): CRC Pr.
- Chenoweth T, Hubata R, Louis RDS. 2000. Automatic ARMA identification using neural networks and the extended sample autocorrelation function: a reevaluation. *Decision Support Systems*. 29(1): 21-30.
- Choi HK. 2018. Stock correlation coefficient prediction with ARIMA-LSTM Hybrid model. *arXiv preprint arXiv:1808.01560*.
- Chollet F. 2021. *Deep Learning with Python*. Rockefeller Center (NY): Simon & Schuster.
- Dradjat B. 2011. Dampak krisis finansial global dan kebijakan antisipatif pengembangan industri kelapa sawit. *Analisis Kebijakan Pertanian*. 9(3): 237-260.
- Fan D, Sun H, Yao J, Zhang K, Yan X, Sun Z. 2021. Well production forecasting based on ARIMA-LSTM model considering manual operations. *Energy*. 220: 119708.
- Fitch Ratings. 2022 Nov 22. Asians palm oil outlook 2023. Outlook Report. [diakses 2023 Juni 30]. <https://www.fitchratings.com/research/corporate-finance/asian-palm-oil-outlook-2023-22-11->

2022#:~:text=Fitch%20Ratings%20expects%20average%20crude%20pal
m%20oil%20%28CPO%29,a%20material%20improvement%20in%20for
eign-labour%20availability%20in%20Malaysia.

- Go YH, Lau WY. 2020. The impact of global financial crisis on information efficiency: Evidence from price-volume relation in crude palm oil futures market. *Journal of Commodity Markets*. 17: 100081.
- Goodfellow I, Bengio Y, Courville A. 2016. *Deep learning*. Cambridge (UK): MIT Press.
- Hilda A. 2014. The impact of crude palm oil price on rupiah's rate. *Bulletin of Monetary Economics and Banking*. 16(4): 295-314.
- Iskandar A. 2015. The impact of exchange of crude palm oil (CPO) price on export value of palm oil commodity and economy of indonesia (Vector Autoregression Analysis Approach). *Jurnal Info Artha Sekolah Tinggi Akuntansi Negara (STAN)*. 13(1): 0852-6737.
- Kanchymalay K, Salim N, Sukprasert A, Krishnan R, Hashim URA. 2017. Multivariate time series forecasting of crude palm oil price using machine learning techniques. *IOP Conference Series: Materials Science and Engineering*. 226(1): 012117.
- Karlic B, Olgac AV. 2011. Performance analysis of various activation functions on generalized MLP architectures of neural network. *International Journal of Artificial Intelligent and Expert System (IJAE)*. 1(4):112.
- Kewinoto JW, Sjahrudin MM. 2015. Analisis pengaruh harga komoditas minyak kelapa sawit (CPO), tingkat inflasi, nilai tukar Rupiah dan volume penjualan komoditas (CPO) terhadap harga saham pada perusahaan penghasil kelapa sawit (CPO) yang terdaftar di Bursa Efek Indonesia. *JOM FEKON*. 2(1)
- Kumar S, Damaraju A, Kumar A, Kumari S, Chen CM. 2021. LSTM network for transportation mode detection. *Journal of Internet Technology*. 22(4): 891-902.
- Khalid N, Hamidi HNA, Thinagar S, Marwan NF. 2018. Peramalan harga minyak sawit mentah Malaysia: satu pendekatan ekonometrik. *Jurnal Ekonomi Malaysia*. 52(3): 263-278.
- Lestari I, Sumargo B, Ladayya F. 2022. Pemodelan ARIMA intervensi untuk meramalkan harga minyak mentah dunia. *Statistika*. 22(2): 133-146.
- Llewellyn A. 2022 Apr 25. Hurting the vulnerable: Indonesia's palm oil ban ignites prices. Al Jazeera. Economy. [diakses 2023 Mei 10]. <https://www.aljazeera.com/economy/2022/4/25/hurting-most-vulnerable>
- Montgomery DC, Jennings CL, Kulahci M. 2015. *Introduction to Time Series Analysis and Forecasting 2nd ed*. Hoboken (NJ): Jhon Wiley and Sons.
- Mustapa FH, Ismail MT. 2019. Modelling and forecasting S&P 500 stock prices using Hybrid Arima-Garch Model. *Journal of Physics: Conference Series*. 1366(1): 012130.
- Naeem M, Maged SA. 2020. Linear time invariant state space system identification using ADAM optimization. *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*. 196-204.
- Ofuoku M, Ngriatedema T. 2022. Prediction of the price of crude palm oil: a deep learning approach. *International Journal of Strategic Decision Sciences*. 13(1): 1-15.

- Oktiani D. 2018. Pemodelan harga CPO Indonesia tahun 2018 dengan ARIMA. *Indonesian Journal of Industrial Research*. 10(2): 33:41.
- Phaladisailoed T, Numnonda T. 2018. Machine learning models comparison for bitcoin price prediction. *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*. 506-511.
- Rahim NF, Mahmud O, Sokkalingan R. 2018. A Comparative review on various method of forecasting crude palm oil prices. *Journal of Physics Conference Series*. 1123(1): 012043.
- Siami-Namini S, Tavakoli N, Namin AS. 2018. A comparison of ARIMA and LSTM in forecasting time series. *2018 17th IEEE international conference on machine learning and applications (ICMLA)*. 1394-1401.
- Sihombing PR, Lestari WP, Nursaskiawati MA, Indryani E. 2022. Perbandingan performa ETS dan ARIMA dalam pemodelan harga CPO. *Jurnal Ekonomi dan Statistik Indonesia*. 2(2): 207-211.
- Soares EFDS, Campos CAV, de Lucena SC. 2019. Online travel mode detection method using automated machine learning and feature engineering. *Future Computer Systems*. 101: 1201-1212.
- Syahril S, Majid SA, Zulham T, Saputra J dan Badli S. 2019. The effect of global prices of crude palm oil, marketing margins and palm oil plantations on the environmental destruction: An application of Johansen cointegration approach.
- Tsay RS, Tiao GC. 1984. Consistent estimates of autoregressive parameters and extended sample autocorrelation function for stationary and nonstationary ARMA models. *Journal of the American Statistical Association*. 79(385): 84-96.
- Varma S, Simon R. 2006. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*. 7(1): 1-8.
- Williams JD, Zweig G. 2016. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. arXiv preprint arXiv:1606.01269.
- Wu XY, Wu QB, Zhu CQ. 2019. Improved EEMD-based crude oil price forecasting using LSTM networks. *Physica A: Statistical Mechanics and its Applications*. 516: 114-124.
- Xu D, Zhang Q, Ding Y, Zhang D. 2022. Application of a Hybrid ARIMA-LSTM model based on the SPEI for drought forecasting. *Environmental Science and Pollution Research*. 29(3): 4128-4144.

LAMPIRAN

- Hak Cipta Dilindungi Undang-undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

Lampiran 1 Pemodelan ARIMA di *Python*

```
# Library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.stats.stattools import jarque_bera
from scipy.stats import t
from statsmodels.tsa.stattools import adfuller
from pandas import DataFrame
from pandas import Series
from pandas import concat
from pandas import read_csv
from pandas import datetime
from matplotlib import pyplot
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_percentage_error
from math import sqrt

# Eksplorasi
data = pd.read_csv('palm-oil-240.csv', index_col='Month', parse_dates=True)
data.plot(figsize=(12,4))

# Pembagian Data
train_size = int(len(data) * 0.7)
train_data = data.iloc[:train_size]
test_data = data.iloc[train_size:]

# Kestasioneran Data
sm.graphics.tsa.plot_acf(train_data['Price ($/ Ton Metrik)'], lags=20)
plt.title('Autocorrelation Function (ACF)')
plt.show()
adf = adfuller(train_data['Price ($/ Ton Metrik)'])
print(adf)
# Differencing on training data
diff = train_data['Price ($/ Ton Metrik)'].diff().dropna()
plt.plot(diff)
adf_diff = adfuller(diff)
print(adf_diff)

# Identifikasi Model
plot_acf(diff, lags=20)
plt.title('ACF setelah Differencing')
```

```
plt.show()
plot_pacf(diff, lags=20)
plt.title('Partial ACF')
plt.show()
```

```
# Pendugaan dan Pengujian Parameter
model1 = ARIMA(train_data, order=(0, 1, 1))
results1 = model1.fit(method_kwargs={'warn_convergence': False})
params1 = results1.params
# Model 1 (0, 1, 1)
alpha = 0.05
t_crit1 = t.ppf(1 - alpha/2, df=results1.df_resid)
params1 = results1.params
std_errors1 = results1.bse
for i in range(len(params1)):
    t_stat1 = params1[i] / std_errors1[i]
    p_value1 = 2 * (1 - t.cdf(np.abs(t_stat1), df=results1.df_resid))
    print(f"Parameter: {i}, Estimate: {params1[i]:.4f}, "
          f"Std. Error: {std_errors1[i]:.4f}, t-stat: {t_stat1:.4f}, "
          f"p-value: {p_value1:.4f}, Significance: {'Significant' if np.abs(t_stat1) >
t_crit1 else 'Not Significant'}")
# Nilai AIC
aic1 = results1.aic
print(f"AIC ARIMA(0,1,1): {aic1:.4f}")

# Uji diagnostik model
print(results3.summary())
# plot residual errors
residuals = DataFrame(results3.resid)
residuals.plot()
plt.show()
residuals.plot(kind='kde')
plt.show()
print(residuals.describe())
```


Lampiran 2 Pemodelan LSTM di *Python*

Library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_percentage_error
from keras.optimizers import Adam
from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import GridSearchCV
from pandas import read_csv
from pandas import DataFrame
```

Set random seed

```
np.random.seed(7)
```

Eksplorasi

```
dataframe = pd.read_csv('palm-oil-240.csv', usecols=[1], engine='python')
dataset = dataframe.values.astype('float32')
plt.plot(dataset)
```

Normalisasi

```
scaler = MinMaxScaler(feature_range=(0,1))
dataset = scaler.fit_transform(dataset)
plt.plot(dataset)
```

Pembagian data

```
train_size = int(len(dataset)*0.7)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
```

Supervised Learning

```
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)
look_back = 1
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)
```



```
# reshape input to be [samples, time steps, features]
trainX = np.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
testX = np.reshape(testX, (testX.shape[0], 1, testX.shape[1]))

# Mendefinisikan fungsi untuk lstm model
def create_model(learning_rate=0.001, look_back=1, units=4):
    model = Sequential()
    model.add(LSTM(units, input_shape=(1, look_back)))
    model.add(Dense(1))
    optimizer = Adam(learning_rate=learning_rate)
    model.compile(loss='mean_squared_error', optimizer=optimizer)
    return model

# Membentuk model dengan KerasRegressor
model = KerasRegressor(build_fn=create_model, verbose=2)

# Hyperparameter tuning
param_grid = {'batch_size': [1, 32],
              'epochs': [1, 100],
              'learning_rate': [0.001, 0.005]}
grid = GridSearchCV(estimator=model, param_grid=param_grid, cv=10)
grid_result = grid.fit(trainX, trainY)
print('Best Score: ', grid_result.best_score_)
print('Best Params: ', grid_result.best_params_)
results = grid_result.cv_results_
for mean_score, std_score, params in zip(results['mean_test_score'],
    results['std_test_score'], results['params']):
    print("Hyperparameters:", params)
    print("Mean Score:", mean_score)
    print("Standard Deviation:", std_score)
    print()

# Membangun dan melatih kembali model dengan hyperparameter terbaik
best_lr = grid_result.best_params_['learning_rate']
best_bs = grid_result.best_params_['batch_size']
best_ep = grid_result.best_params_['epochs']
best_units = 4 # neuron
best_model = create_model(learning_rate=best_lr, units=best_units)
history = best_model.fit(trainX, trainY, epochs=best_ep, batch_size=best_bs,
    verbose=2)

# Prediksi data uji
predictions = []
for i in range(len(testX)):
    X = testX[i]
    X = np.reshape(X, (1, 1, look_back))
    yhat = best_model.predict(X)
    yhat = scaler.inverse_transform(yhat)
```

```

    predictions.append(yhat)
for i in range(len(predictions)):
    expected = scaler.inverse_transform(testY[i].reshape(-1, 1))
    print('Month=%d, Predicted=%f, Expected=%f' % (i+1, predictions[i],
    expected))
# Prediksi data latih
predictions_train = []
for i in range(len(trainX)):
    X = trainX[i]
    X = np.reshape(X, (1, 1, look_back))
    yhat = best_model.predict(X)
    yhat = scaler.inverse_transform(yhat)
    predictions_train.append(yhat)
for i in range(len(predictions_train)):
    expected_train = scaler.inverse_transform(trainY[i].reshape(-1, 1))
    print('Month=%d, Predicted=%f, Expected=%f' % (i+1, predictions_train[i],
    expected_train))

# Evaluasi
predictions = np.array(predictions)
testY = np.array(testY)
testY = scaler.inverse_transform([testY])
testY = np.array(testY).reshape(-1, 1) # Reshape testY_coba to have 2
dimensions
predictions = np.array(predictions).reshape(-1, 1)
rmse = np.sqrt(mean_squared_error(testY, predictions))
print('RMSE test: %.4f' % rmse)
mape = mean_absolute_percentage_error(testY, predictions)
print('MAPE test: %.4f' % mape)
plt.plot(testY)
plt.plot(predictions)
plt.show
predictions_train = np.array(predictions_train)
trainY = np.array(trainY)
trainY = scaler.inverse_transform([trainY])
trainY = np.array(trainY).reshape(-1, 1) # Reshape testY_coba to have 2
dimensions
predictions_train = np.array(predictions_train).reshape(-1, 1)
rmse_train = np.sqrt(mean_squared_error(trainY, predictions_train))
print('RMSE test: %.4f' % rmse_train)
mape_train = mean_absolute_percentage_error(trainY, predictions_train)
print('MAPE test: %.4f' % mape_train)
plt.plot(trainY)
plt.plot(predictions_train)
plt.show

```



Lampiran 3 Pemodelan Hybrid di *Python*

```
# Prediksi model ARIMA
# Data uji
history = [x for x in train]
predictions_test = []
residuals_test = []
for t in range(len(test)):
    model = ARIMA(history, order=(1, 1, 2))
    model_fit = model.fit()
    output = model_fit.forecast()
    yhat_test = output[0]
    predictions_test.append(yhat_test)
    obs_test = test[t]
    residuals_test.append(obs_test - yhat_test) # store residuals
    history.append(obs_test)
    print('predicted=%f, expected=%f' % (yhat_test, obs_test))
residuals_test = pd.DataFrame({'residuals': residuals_test})
# Data latih
history = [x for x in train]
predictions_train = []
residuals_train = []
for t in range(len(train)):
    model = ARIMA(history, order=(1, 1, 2))
    model_fit = model.fit()
    output = model_fit.forecast()
    yhat_train = output[0]
    predictions_train.append(yhat_train)
    obs_train = train[t]
    residuals_train.append(obs_train - yhat_train) # store residuals
    history.append(obs_train)
    print('predicted=%f, expected=%f' % (yhat_train, obs_train))
residuals_train = pd.DataFrame({'residuals': residuals_train})

# Dataset residuals ARIMA
dataframe = pd.concat([residuals_train, residuals_test])
dataset = dataframe.values.astype('float32')
plt.plot(dataset)
```

RIWAYAT HIDUP

Penulis dilahirkan di kota Atambua pada tanggal 10 bulan Maret tahun 2001 sebagai anak pertama dari tiga bersaudara pasangan bapak Vinsensius Ignatius Theodorus Uskono dan ibu Maria Yustina Mako Bau. Pendidikan sekolah menengah atas (SMA) ditempuh di sekolah SMA Negeri 1 Atambua, dan lulus pada tahun 2019. Pada tahun yang sama penulis diterima sebagai mahasiswa program sarjana (S-1) di Program Studi Statistika dan Sains Data/, Fakultas Matematika dan Ilmu Pengetahuan Alam, IPB *University* melalui jalur Afirmasi 3T.

Selama mengikuti program S-1, penulis pernah mengikuti beberapa kegiatan kepanitiaan di Departemen Statistika dan di luar Departemen Statistika. Penulis lebih aktif mengikuti *online course* di beberapa media seperti Coursera, DataCamp, dll untuk meningkatkan kemampuan penulis. Pengalaman paling berkesan selama penulis kuliah di IPB *University* adalah Ketika penulis mengikut kegiatan KKN di Desa Ngambarsari, kecamatan Karangtengah, Kabupaten Wonogiri, Jawa Tengah.

@Hak cipta milik IPB University

IPB University

Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.