DAFTAR PUSTAKA

Angin, A. S. P., Raj, S., Silitonga, P., Pandiangan, Y. H., Siallagan, H., & Sipayung, R. C. (2023). Pengaruh Motivasi Investasi, Pemahaman Investasi Dan Minat Investasi Pada Mahasiswa. *Innovative: Journal Of Social Science Research*, *3*(5), 2508-2521.

Ahmad, A. (2017). Mengenal Artificial Intelligence, Machine Learning, Neural Network, Dan Deep Learning. J. Teknol. Indones., No. October, 3.

Aulia, N. (2020). Prediksi Harga Ethereum Berdasarkan Informasi Blockchain Menggunakan Metode Long Short Term Memory (Doctoral dissertation, Universitas Islam Indonesia).

Elfajar, A. B., Setiawan, B. D., & Dewi, C. (2017). Peramalan Jumlah Kunjungan Wisatawan Kota Batu Menggunakan Metode Time Invariant *Fuzzy Time series*. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-Ptiik) Universitas Brawijaya, 1(2), 85–94.

Fauzi, A. (2019). Forecasting Saham Syariah Dengan Menggunakan Lstm. Al-Masraf: Jurnal Lembaga Keuangan Dan Perbankan, 4(1), 65-69.

Handayani, L., & Anggriani, D. (2015). Perbandingan Model Chen Dan Model Lee Pada Metode *Fuzzy Time series* Untuk Prediksi Harga Emas. Pseudocode, 2(1), 28-36.

Hansun, S. (2012). Peramalan Data Ihsg Menggunakan *Fuzzy Time series*. Ijccs (Indonesian Journal Of Computing And Cybernetics Systems), 6(2).

Hasinah, M. (2015). Pengaruh larangan ekspor bahan mineral mentah terhadap PT. Antam TBK (analisis peraturan pemerintah nomor 1 tahun 2014).

Idnfinancials. (2023). Https://Www.Idnfinancials.Com/Id/Adro/Pt-Adaro-Energy-Indonesia-Tbk. Diakses Pada 8 November 2023

Khofi, A. F., Arifianto, D., & Saifudin, I. (2022). Perbandingan Model Chen Dan Model Lee Pada Metode *Fuzzy Time series* Untuk Peramalan Harga Beras. Jurnal Smart Teknologi, 3(2), 140-146.

Milniadi, A. D., & Adiwijaya, N. O. (2023). Analisis Perbandingan Model Arima Dan Lstm Dalam Peramalan Harga Penutupan Saham (Studi Kasus: 6 Kriteria Kategori Saham Menurut Peter Lynch). Sibatik Journal:

Jurnal Ilmiah Bidang Sosial, Ekonomi, Budaya, Teknologi, Dan Pendidikan, 2(6), 1683-1692.

Nasution, A. (2019). Metode Weighted Moving Average Dalam M-Forecasting. Jurteksi (Jurnal Teknologi Dan Sistem Informasi), 5(2), 119-124.

Novirman, A. A. (2013). Pengaruh Volume Perdagangan, Frekuensi Perdagangan, Kapitalisasi Pasar Dan Dividendpayoutratioterhadap Return Saham Jakarta Islamic Index (Bachelor's Thesis, Fakultas Ekonomi Dan Bisnis Uin Jakarta).

Nugroho, K. (2016). Model Analisis Prediksi Menggunakan Metode *Fuzzy Time series*. Jurnal Ilmiah Infokam, 12(1).

Oktavia, I., & Nugraha, K. G. S. (2018). Faktor-Faktor Yang Mempengaruhi Harga Saham. Unej E-Proceeding.

Patriya, E., Latif, A., & Handayani, H. (2023). Peramalan Harga Saham Penutupan Indeks Harga Saham Gabungan (Ihsg) Menggunakan Algoritma Long Short Term Memory (Lstm). Jurnal Ilmiah Ekonomi Bisnis, 28(2), 304-314.

Perdani, W. R., Magdalena, R., & Pratiwi, N. K. C. (2022). Deep Learning Untuk Klasifikasi Glaukoma Dengan Menggunakan Arsitektur Efficientnet. *Elkomika: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, *10*(2), 322.

Pipin, S. J., Purba, R., & Kurniawan, H. (2023). Prediksi Saham Menggunakan Recurrent Neural Network (RNN-LSTM) dengan Optimasi Adaptive Moment Estimation. Journal of Computer System and Informatics (JoSYC), 4(4), 806-815.

Puteri, D. I. (2023). Implementasi Long Short Term Memory (LSTM) dan Bidirectional Long Short Term Memory (BiLSTM) Dalam Prediksi Harga Saham Syariah. Euler: Jurnal Ilmiah Matematika, Sains dan Teknologi, 11(1), 35-43.

Rachim F Tarno And Sugito 2020 Perbandingan *Fuzzy Time series* Dengan Metode Chen Dan Metode S.R. Singh. Jurnal Gaussian Volume 9 (3) 306-315.

Rahmawati, A., & Sulistijanti, W. (2023). Peramalan Harga Penutupan Saham Pt. Unilever Indonesia Dengan Menggunakan Metode *Fuzzy Time series* Lee. Jurnal Mirai Management, 8(2), 350-359.

Sumartini, S., Hayati, M. N., & Wahyuningsih, S. (2017). Peramalan Menggunakan Metode *Fuzzy Time series* Cheng. Eksponensial, 8(1), 51-56.

Tambunan, D. (2020). Investasi Saham Di Masa Pandemi Covid-19. Jurnal Khatulistiwa Informatika, 4(2), 117-123.

Trenggana, A. F. M., & Kuswardhana, R. (2017). Pengaruh Informasi Produk, Risiko Investasi, Kepuasaan Investor Dan Minat Mahasiswa Berinvestasi. Jurnal Sekretaris Dan Administrasi Bisnis, 1(1), 8-17.

Tussifah, S. A. Analisis Perbandingan Kinerja Model ARIMA, LSTM Dan GRU Pada Stock Price Forecasting (Bachelor's Thesis, Fakultas Sains Dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta).

Zahroh, A. (2015). Instrumen Pasar Modal. Iqtishodu na: Jurnal Ekonomi Islam, 4(1), 51-65.

Zimmermann, H. J. (2010). *Fuzzy* set theory. Wiley interdisciplinary reviews: computational statistics, 2(3), 317-332.

LAMPIRAN

Nama	:	Pt Adaro Energy Indonesia Tbk
Perusahaan	:	ADRO.JK
Nama Saham	:	28/10/2019 – 25/10/2023
Periode	:	Menara Karya 23th Floor, DKI
Alamat		Jakarta

Lampiran 1. Dataset Harga Close PT. Adaro Energy Indonesia Tbk:

Link: https://bit.ly/DatasetSkripsiNunungUnhas

Date	*Open*	*High*	*Low*	*Close*	*Volume*
28/10/2019	1.350	1.375	1.345	1.370	27468800
29/10/2019	1.370	1.375	1.325	1.335	33546400
30/10/2019	1.320	1.335	1.300	1.305	34870600
31/10/2019	1.305	1.320	1.295	1.310	29065600
01/11/2019	1.300	1.310	1.250	1.270	38229500
04/11/2019	1.280	1.300	1.270	1.280	28986500
05/11/2019	1.290	1.370	1.290	1.370	63048100
06/11/2019	1.375	1.410	1.340	1.390	88847800
07/11/2019	1.390	1.390	1.325	1.385	58536200
08/11/2019	1.400	1.435	1.395	1.415	84463700
⋮	⋮	⋮	⋮	⋮	⋮
11/10/2023	2.800	2.820	2.760	2.760	26295100
12/10/2023	2.760	2.760	2.690	2.690	41046500
13/10/2023	2.730	2.740	2.690	2.710	18962900
16/10/2023	2.750	2.790	2.730	2.780	36979400
17/10/2023	2.770	2.780	2.750	2.770	22750800
18/10/2023	2.750	2.880	2.730	2.840	74460900
19/10/2023	2.840	2.860	2.800	2.820	57235900
20/10/2023	2.830	2.830	2.760	2.800	47189400
23/10/2023	2.780	2.790	2.650	2.660	69223700
24/10/2023	2.640	2.690	2.630	2.660	39300700

Lampiran 2. Perhitungan Metode Fuzzy Time Series Model Lee:

Link: https://bit.ly/FuzzyTimeSeriesLeeNunungUnhas

Date	Close	Fuzzyfikasi	FLR			Prediksi	MAPE	MSE
2019-10-28	1370	A3	Na	->	A3			
2019-10-29	1335	A3	A3	->	A3	1415,927217	0,06061963817	6549,214444
2019-10-30	1305	A3	A3	->	A3	1415,927217	0,08500169882	12304,84746
2019-10-31	1310	A3	A3	->	A3	1415,927217	0,08086047096	11220,57529
2019-11-01	1270	A2	A3	->	A2	1415,927217	0,1149033204	21294,75265
2019-11-04	1280	A2	A2	->	A2	1133,061208	0,1147959314	21591,00867
2019-11-05	1370	A3	A2	->	A3	1133,061208	0,1729480235	56139,99128
2019-11-06	1390	A3	A3	->	A3	1415,927217	0,01865267407	672,220579
2019-11-07	1385	A3	A3	->	A3	1415,927217	0,02233012054	956,4927486
2019-11-08	1415	A3	A3	->	A3	1415,927217	0,0006552769999	0,8597312813
⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮
2023-10-11	2760	A7	A7	->	A7	2773,033827	0,004722400956	169,8806368
2023-10-12	2690	A7	A7	->	A7	2773,033827	0,03086759355	6894,616366
2023-10-13	2710	A7	A7	->	A7	2773,033827	0,02325971463	3973,263301
2023-10-16	2780	A7	A7	->	A7	2773,033827	0,002505817756	48,5275713
2023-10-17	2770	A7	A7	->	A7	2773,033827	0,001095244274	9,204104072
2023-10-18	2840	A7	A7	->	A7	2773,033827	0,02357963851	4484,468375
2023-10-19	2820	A7	A7	->	A7	2773,033827	0,0166546714	2205,82144
2023-10-20	2800	A7	A7	->	A7	2773,033827	0,009630776201	727,1745058
2023-10-23	2660	A7	A7	->	A7	2773,033827	0,04249391979	12776,64596
2023-10-24	2660	A7	A7	->	A7	2773,033827	0,04249391979	12776,64596

Lampiran 3. Perhitungan Code Metode Long Short-Term Memory:

Link: https://bit.ly/LSTMNunungUnhas

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
import math
from warnings import simplefilter
import plotly.graph_objects as go

# Mute sklearn warnings
simplefilter(action='ignore', category=FutureWarning)
simplefilter(action='ignore', category=DeprecationWarning)
# Load Dataset
data = pd.read_excel('/content/Saham ADRO.JK periode 28 oktober 2019 sampai dengan 25 oktober 2023.xlsx')
# Mengatur kolom 'Date' sebagai indeks
data.set_index('Date', inplace=True)
print(data.columns)
print(data.shape)
data.info()
data
```

#EDA

```
#Normalisasi Data: Min-Max Scaler
close_prices = data['Close'].values.reshape(-1, 1)
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(close_prices)
# Extract original and scaled close prices
original_close_prices = data['Close'].values
scaled_close_prices = scaled_data.flatten()
# Calculate statistics
original_stats = pd.Series(original_close_prices).describe()
scaled_stats = pd.Series(scaled_close_prices).describe()
# Print the summary
print("Original Close Prices Statistics:")
print(original_stats)
print("\nScaled Close Prices Statistics:")
print(scaled_stats)
```

#Visualisasi Data

```
import plotly.graph_objects as go
# Plot Close price
fig_close = go.Figure()
fig_close.add_trace(go.Scatter(x=data.index, y=data['Close'], mode='lines', name='Closing
Price', line=dict(color='green')))
fig_close.update_layout(
    title='Closing Stock Price History',
    xaxis=dict(title='Date'),
    yaxis=dict(title='Closing Stock Price'),)
fig_close.show()

ma_day = [10, 20, 50] #the most common moving averages
for ma in ma_day:
    column_name = f"MA for {ma} days"
    data[column_name] = data['Close'].rolling(ma).mean()
fig, axes = plt.subplots(nrows=3, ncols=1)
fig.set_figheight(10)
fig.set_figwidth(17)

plt.figure(figsize=(15, 5))
plt.subplots_adjust(top=1.25, bottom=1.2)

data[['Close', 'MA for 10 days']].plot(ax=axes[0])
axes[0].set_title('ADRO.JK')
data[['Close','MA for 20 days']].plot(ax=axes[1])
data[['Close','MA for 50 days']].plot(ax=axes[2])
plt.figure(figsize=(15,6));
p1=data[['Open', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']].plot()
plt.title('ADRO.JK')
```

#Data Preparation

```
#Membuat data 'Close' untuk dapat di olah selanjutnya di model LSTM
values = data["Close"].values
value = np.reshape(values, (-1, 1))
#Mengubah data menjadi rentan 0-1
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(value)
#Membuat data train 80% dari nilai value
l = math.ceil(len(scaled_data) * 0.8)
train_size = scaled_data[0:l, :]
print(values.shape)
print(value.shape)
print(scaled_data.shape)
print(train_size.shape)
print(values[:5])
print(value[:5])
print(scaled_data[:5])
print(train_size[:5])
#Membuat X_train dan y_train dengan timesteps yang digunakan sebanyak 60 hari data awal
n_steps=60
X_train=[]
y_train=[]
for i in range(n_steps,len(train_size)):
    X_train.append(train_size[i-n_steps:i,0])
    y_train.append(train_size[i,0])
X_train=np.array(X_train)
y_train=np.array(y_train)
#Mengubah shape X_tran dan y_train agar bisa di gunakan pada model LSTM
X_train_lstm=np.reshape(X_train,(X_train.shape[0],X_train.shape[1],1))
y_train_lstm = y_train.reshape(-1, 1)

#Membuat data test yaitu 20% dari dataset ADRO.JK
test_size=scaled_data[l-n_steps:,:]
X_test=[]
y_test=scaled_data[l:,:]
for i in range(n_steps,len(test_size)):
    X_test.append(test_size[i-n_steps:i,0])
X_test=np.array(X_test)
X_test_lstm=np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
y_test_lstm = np.array(y_test).reshape(-1, 1)
```

#Hypertuning parameter

```
# Define hyperparameter search space
space = {
    'units': hp.choice('units', [25, 50, 75, 100, 125]),
    'dropout_rate': hp.uniform('dropout_rate', 0.2, 0.5),
    'learning_rate': hp.loguniform('learning rate', np.log(0.001),
np.log(0.01)),
    'epochs': hp.choice('epochs', [50, 75, 100, 125]),
    'batch_size': hp.choice('batch_size', [16, 32, 64, 128])}
# Define early stopping callback
early_stopping = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dropout, Dense
from sklearn.metrics import mean_squared_error
from hyperopt import fmin, tpe, space_eval, Trials
from keras.callbacks import EarlyStopping
from tensorflow.keras.optimizers import Adamax

def objective(params, model_type, X_train, y_train, X_test, y_test):
    model = Sequential()
    if model_type == 'lstm':
        model.add(LSTM(units=params['units'], return_sequences=True,
activation='tanh', input_shape=(X_train.shape[1], 1)))
        model.add(Dropout(params['dropout_rate']))
        model.add(LSTM(units=params['units'], activation='tanh'))
        model.add(Dense(units=1))

    model.compile(optimizer=Adamax(learning_rate=params['learning_rate']),
loss='mean_squared_error')

    history = model.fit(X_train, y_train, epochs=params['epochs'],
batch_size=params['batch_size'],validation_split=0.1,callbacks=[early_stopping]
, verbose=0)

    y_pred = model.predict(X_test)
    y_pred = scaler.inverse_transform(y_pred)
    y_test_orig = scaler.inverse_transform(y_test.reshape(-1, 1))

    mse = mean_squared_error(y_test_orig, y_pred)
    return mse, history
```

```
# Set up the Trials object to keep track of the optimization process
trials_lstm = Trials()
# Use the fmin function for Bayesian optimization for LSTM and GRU
best_lstm = fmin(fn=lambda params: objective(params, 'lstm', X_train_lstm,
y_train, X_test_lstm, y_test)[0],space=space, algo=tpe.suggest, max_evals=2,
trials=trials_lstm)
# Get the best hyperparameters from the optimization results
best_params_lstm = space_eval(space, best_lstm)
```

```
# Build the final LSTM model using the best hyperparameters
final_model_lstm = Sequential([
    LSTM(units=best_params_lstm['units'], return_sequences=True,
input_shape=(X_train_lstm.shape[1], 1)),
    Dropout(best_params_lstm['dropout_rate']),
    LSTM(units=best_params_lstm['units']),
    Dense(units=1)])

# Kemudian, gunakan Adam sebagai parameter
final_model_lstm.compile(optimizer=tf.keras.optimizers.Adamax(learning_rate=b
est_params_lstm['learning_rate']),
                         loss='mean_squared_error')
# Train the final LSTM model
history_lstm = final_model_lstm.fit(X_train_lstm, y_train,
epochs=best_params_lstm['epochs'],batch_size=best_params_lstm['batch_size'],
verbose=2,validation_split=0.2, callbacks=[early_stopping])


dot_img_file = '/tmp/model_1.png'
tf.keras.utils.plot_model(final_model_lstm, to_file=dot_img_file,
show_shapes=True, show_dtype=False, show_layer_names=True,rankdir='TB',
expand_nested=False, dpi=96, layer_range=None,show_layer_activations=True)
```

```
# Model evaluation for LSTM
y_pred_lstm = final_model_lstm.predict(X_test_lstm)
y_pred_lstm = scaler.inverse_transform(y_pred_lstm)
y_test_orig_lstm = scaler.inverse_transform(y_test.reshape(-1, 1))
# Calculate metrics for LSTM
mse_lstm = mean_squared_error(y_test_orig_lstm, y_pred_lstm)
rmse_lstm = math.sqrt(mse_lstm)
mape_lstm = np.mean(np.abs((y_test_orig_lstm - y_pred_lstm) /
y_test_orig_lstm)) * 100
# Display results for LSTM
print("Performance Metrics for LSTM:")
print("Mean Squared Error (MSE):", mse_lstm)
print("Root Mean Squared Error (RMSE):", rmse_lstm)
print("Mean Absolute Percentage Error (MAPE):", mape_lstm)
```

```
# Visualisasi nilai loss dari riwayat pelatihan
plt.figure(figsize=(8, 6))
plt.plot(history_lstm.history['loss'], label='Loss Pelatihan')
plt.plot(history_lstm.history['val_loss'], label='Loss Validasi')
plt.title('Nilai Loss pada Pelatihan dan Validasi')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

#Visualisasi Hasil Prediksi

```
# Menggunakan indeks dari X_train untuk sumbu x
index_train = data.index[:len(X_train)]
index_test = data.index[-len(X_test):]
# Melakukan plotting nilai sebenarnya, prediksi LSTM, dan data latih
plt.figure(figsize=(10, 6))
plt.plot(index_test, y_test_orig_lstm, label='Nilai Sebenarnya',
color='blue')
plt.plot(index_test, y_pred_lstm, label='Nilai Prediksi LSTM',
color='red')
plt.title('Perbandingan Nilai Sebenarnya, Prediksi LSTM')
plt.xlabel('Sampel')
plt.ylabel('Nilai')
plt.legend()
plt.show()
```

```
from keras.models import load_model, save_model
final_model_lstm.save('saham_lstm.h5')
```