



Meskipun model algoritma LSTM dan Regresi Linear telah digunakan untuk memprediksi harga saham, namun masih jarang ditemukan perbandingan antara kedua model algoritma tersebut pada prediksi harga saham BBRI. Oleh karena itu, skripsi ini akan membahas dan menganalisis perbandingan antara model algoritma LSTM dan Regresi Linear dalam memprediksi harga saham BBRI periode 2001-2022 menggunakan bahasa pemrograman Python dan aplikasi Orange Data Mining.

1.1 IDENTIFIKASI MASALAH

Berdasarkan hal-hal yang telah dipaparkan pada latar belakang masalah, dapat ditemukan beberapa permasalahan, antara lain:

1. Belum digunakan penggunaan model algoritma LSTM dan Regresi Linear untuk melakukan perbandingan terhadap prediksi harga saham BBRI.
2. Hasil perbandingan komparasi model algoritma LSTM dan Regresi Linear pada bahasa pemrograman Python dan aplikasi Orange Data Mining masih jarang digunakan.

1.2 BATASAN MASALAH

Berdasarkan masalah yang sudah diidentifikasi, berikut ini adalah batasan masalah pada penelitian ini adalah:

1. Belum digunakannya model algoritma LSTM dan Regresi Linear untuk melakukan komparasi terhadap prediksi harga saham BBRI.
2. Belum diketahui hasil komparasi model algoritma LSTM dan Regresi Linear terhadap prediksi harga saham BBRI.

1.3 TUJUAN PENELITIAN

Dalam penelitian ini peneliti bertujuan untuk mengetahui tingkat penyimpangan prediksi harga saham BBRI dengan menggunakan model LSTM dan model Regresi Linear.

2. TINJAUAN PUSTAKA

2.1 Data Mining

Menurut Pan-Ning Tan, et.al (2019:24) definisi *data mining* adalah sebuah proses otomatis mencari informasi yang berguna dalam penyimpanan data yang besar.

Menurut Pan Ning Tan et al (2019:25) ada dua tipe proses dalam *knowledge discovery in database* (KDD) yaitu :

- a. **Data preprocessing** bertujuan untuk mengubah data mentah menjadi format yang digunakan untuk analisis, tahapan dari *data preprocessing* termasuk menggabungkan data dari beberapa sumber, membersihkan data untuk menghapus observasi duplikat dan *noise*, dan menyeleksi catatan dan fitur yang relevan dalam *data mining*.
- b. **Data postprocessing** bertujuan untuk memastikan bahwa hasil valid dan berguna yang akan diintegrasikan pada pengambilan keputusan seperti filter pola, visualisasi dan interpretasi pola.

Menurut Pan Ning Tan, et.al (2019:29), tugas *data mining* dibagi menjadi dua kategori yaitu:

a. **Predictive Tasks**

Tujuan dari tugas tersebut untuk memprediksi nilai dari atribut tertentu berdasarkan nilai atribut lainnya. Atribut yang akan diprediksi biasa disebut dengan variabel tidak bebas atau target sedangkan atribut yang dilakukan untuk prediksi disebut dengan variabel bebas atau bersifat penjelasan.

b. **Descriptive Tasks**

Tujuan dari tugas tersebut untuk mendapatkan pola (korelasi, tren, klaster, lintasan dan anomali) yang merangkum hubungan yang mendasari data. Untuk tugas deskriptif bersifat eksploratif dan sering membutuhkan teknik *postprocessing* untuk memvalidasi dan menjelaskan hasilnya.



Menurut Pan Ning Tan, et.al (2019:29), *data mining* memiliki beberapa metode pengolahan yaitu:

a. **Prediksi (Predictive).**

Teknik prediksi digunakan apabila suatu nilai memiliki atribut yang berbeda, contoh algoritmanya seperti Algoritma *Linear Regression*, *Neural Network* dan lain-lain.

b. **Asosiasi (Association)**

Teknik Asosiasi digunakan untuk hubungan antar data, contoh algoritmanya seperti Algoritma Apriori.

c. **Klustering (Clustering)**

Teknik *clustering* digunakan untuk pengelompokan data dalam suatu kelompok tertentu, contoh algoritmanya seperti *K-Means*, *K-Medoids*, *Self Organization Map(SOM)*, *Fuzzy C-Means*.

d. **Klasifikasi (Classification)**

Teknik klasifikasi mengelompokkan data yang mempunyai variabel tertentu berbeda dengan *clustering* yang tidak memiliki variabel yang dependen, contoh algoritma seperti *ID3* dan *K Nearest Neighbour*.

2.2. Algoritma Long Term-Short Memory (LSTM)

Menurut Ian Goodfellow, Yoshua Bengio, et al (2017:410) definisi long term-short memory yaitu ide cerdas untuk memperkenalkan self-loop untuk menghasilkan jalur di mana gradien dapat mengalir dalam jangka waktu yang lama merupakan kontribusi inti dari jangka pendek awal (LSTM). Dengan membuat bobot dari self-loop ini terjaga (dikontrol oleh unit tersembunyi lainnya), skala waktu integrasi dapat diubah secara dinamis. Meskipun LSTM memiliki parameter yang tetap, skala waktu integrasi dapat berubah berdasarkan urutan input, karena konstanta waktu adalah output dari model itu sendiri.

2.3 Algoritma Regresi Linear

Menurut Ibnu Daqiqil (2022:28) definisi Regresi Linear yaitu analisis regresi setidaknya memiliki 3 kegunaan, yaitu mendeskripsi fenomena data atau kasus yang sedang diteliti; untuk tujuan kontrol; serta untuk tujuan prediksi. Regresi mampu mendeskripsikan fenomena data melalui terbentuknya suatu model hubungan yang bersifatnya numerik.

2.4 Machine Learning

Menurut Andreas C.Muller (2017:1), definisi machine learning yaitu machine learning merupakan suatu proses untuk mengekstraksi pengetahuan berdasarkan data. Bidang penelitian ini berkaitan dengan statistik, artificial intelligence, dan komputer sains dan juga dikenal sebagai analitik prediktif atau pembelajaran statistik.

2.5 Saham

Saham adalah jenis sekuritas yang menunjukkan kepemilikan dalam sebuah perusahaan dan mewakili bagian tertentu (diukur dalam bentuk saham) dari kesuksesan perusahaan di masa depan. Menurut sumber terdapat 2 jenis saham yang digunakan secara umum, yaitu:

a. **Saham Biasa**

Saham yang memberikan hak kepada pemiliknya untuk memberikan suara dalam rapat pemegang saham dan menerima dividen yang dikeluarkan perusahaan.

b. **Saham Preferen**

Saham yang biasanya tidak memberikan hak suara, tetapi memiliki beberapa hak yang melebihi hak-hak saham biasa. Pemegang saham preferen, misalnya, memiliki perlakuan istimewa dalam kondisi tertentu, seperti menerima dividen sebelum pemegang saham biasa jika terjadi likuidasi

atau kebangkrutan perusahaan. Selain itu, saham preferen juga memiliki fungsi yang mirip dengan obligasi bagi investor yang mencari pendapatan stabil.

3. METODE

3.1 Teknik Pengumpulan Data

Dalam penelitian ini data yang digunakan adalah data sekunder dari Kaggle yaitu <https://www.kaggle.com/>. Alasan peneliti menggunakan data sekunder ini adalah untuk mempersingkat waktu pengumpulan data apabila pengumpulan data dilakukan secara manual akan memakan waktu cukup lama. Dalam melakukan pengumpulan data, peneliti menggunakan metode kuantitatif, yaitu penelitian yang lebih sistematis, terencana, terstruktur, jelas dari awal hingga akhir penelitian serta tidak dipengaruhi oleh keadaan yang ada pada lapangan.

3.2 Teknik Analisis Data

Dalam penelitian ini, peneliti menggunakan teknik analisis yang dilakukan berdasarkan metode CRISP-DM (Cross-Industry Standart Process for Data Mining). CRISP-DM menyediakan proses standar yang tidak berpemilik dan tersedia secara bebas agar dapat menyesuaikan hasil data mining ke dalam strategi pemecahan masalah umum bisnis atau unit penelitian.

a. Fase Bisnis/Pemahaman Penelitian (*Business/Research Understanding Phase*)

- (1) Pertama, menyampaikan dengan jelas tujuan dan persyaratan proyek dalam hal bisnis atau unit penelitian secara keseluruhan.
- (2) Kemudian, terjemahkan tujuan dan batasan ini ke dalam perumusan definisi masalah data mining.
- (3) Terakhir, siapkan strategi awal untuk mencapai tujuan yang telah ditetapkan.

b. Fase Pemahaman Data (*Data Understanding Phase*)

- (1) Pertama, mengumpulkan data.
- (2) Kemudian, gunakan analisis data eksploratori untuk membiasakan diri dengan data, dan menemukan wawasan awal.
- (3) Mengevaluasi kualitas data.
- (4) Terakhir, memilih data yang sesuai dan berkemungkinan berisi pola agar dapat ditindaklanjuti.

c. Fase Persiapan Data (*Data Preparation Phase*)

- (1) Pada fase ini mencakup semua aspek mulai dari menyiapkan set data akhir, data awal, data mentah, dan data kotor.
- (2) Pilih kasus dan variabel yang diinginkan untuk melakukan proses analisis.
- (3) Melakukan transformasi pada variabel tertentu, jika diperlukan.
- (4) Bersihkan data mentah sehingga siap untuk alat pemodelan.

d. Fase Model (*Modeling Phase*)

- (1) Pilih dan terapkan teknik pemodelan yang sesuai.
- (2) Mengkalibrasi pengaturan model untuk mengoptimalkan hasil.
- (3) Sering kali, beberapa teknik yang berbeda dapat diterapkan untuk masalah penggalian data yang sama.
- (4) Mungkin memerlukan pengulangan kembali ke tahap persiapan data, untuk menyesuaikan bentuk data dengan persyaratan yang lebih spesifik dari teknik data mining tertentu.

e. Fase Evaluasi (*Evaluation Phase*)

- (1) Fase pemodelan telah menghasilkan satu atau lebih model. Model-model ini harus dievaluasi kualitas dan efektivitasnya, sebelum digunakan di lapangan.
- (2) Menentukan apakah model tersebut sudah mencapai tujuan yang telah ditetapkan pada tahap pertama.





- (3) Menetapkan apakah beberapa aspek penting dari masalah bisnis atau penelitian belum diperhitungkan secara memadai.
- (4) Terakhir, buatlah keputusan mengenai penggunaan hasil data mining.

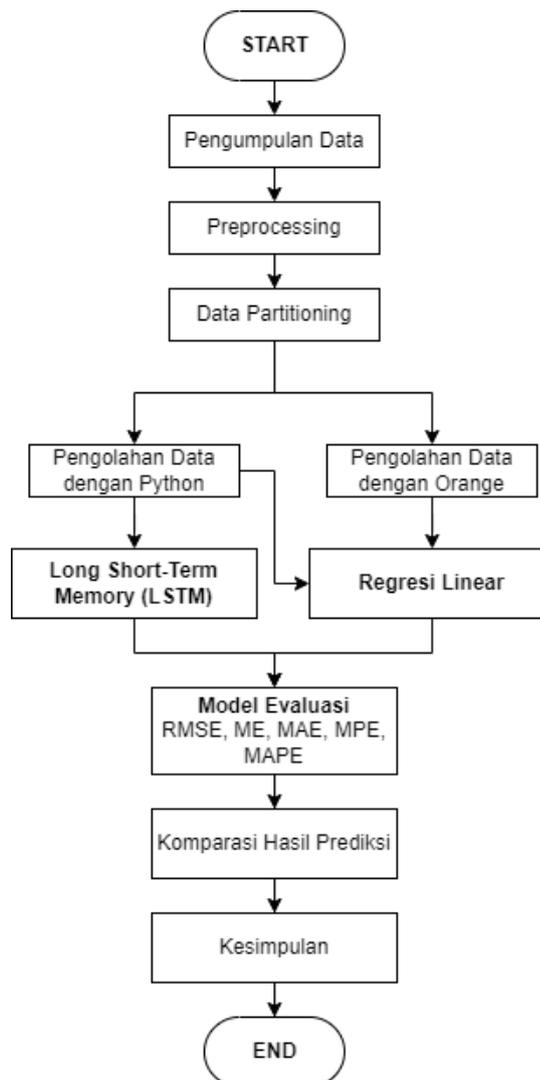
f. Fase Penerapan (Development Phase)

- (1) Pembuatan model tidak menandakan selesainya proyek. Perlu menggunakan model yang telah dibuat.
- (2) Contoh penerapan sederhana: Membuat laporan.
- (3) Contoh penerapan yang lebih kompleks: Menerapkan proses penggalian data paralel di departemen lain.
- (4) Untuk bisnis, pelanggan sering kali melakukan penerapan berdasarkan model yang telah digunakan.

3.3 Penerapan Algoritma

Pada penelitian ini, peneliti melakukan pembahasan tentang bagaimana penerapan sebuah algoritma didalam melakukan *Text Mining* dan menilai isi sentimen yang terkandung didalam sebuah teks. Selain itu, penerapan algoritma digunakan untuk membandingkan hasil akhir data sentimen yang didapatkan dari aplikasi Orange Data Mining dan kode manual Python. Berikut penerapan algoritma tersebut yang dapat digambarkan kedalam bentuk diagram:

Gambar 1
Alur Penerapan Algoritma





3.4 Teknik Pengukuran Data

3.4.1 Pengukuran Kelayakan Metode Prediksi

Peneliti menggunakan Root Mean Square Error (RMSE) sebagai pengukuran kelayakan algoritma, untuk mengukur penyimpangan data prediksi, untuk menentukan tingkat kesalahan prediksi data berdasarkan pengaruh pada tiap variabel yang digunakan. Keakuratan pada pengukuran diestimasikan dengan hasil RMSE memiliki nilai kecil yang mendekati nol. Pada rumus dibawah ini merupakan rumus dasar RMSE.

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

y_t = data prediksi
 y = data actual
 n = jumlah data

3.4.2 Pengukuran Penyimpangan Prediksi

Peneliti menggunakan Mean Error (ME) dan Mean Absolute Error (MAE) untuk mengukur penyimpangan data prediksi. Berikut adalah penjelasan dari masing-masing metode pengukuran penyimpangan prediksi yang peneliti gunakan:

(1) Mean Error (ME)

Rata-rata dari margin selisih antara data yang diprediksi dengan data aktual yang menghasilkan seberapa besar nilai kesalahan prediksi. Rumus perhitungan Mean Error dapat dilihat pada rumus dibawah ini:

$$ME = \frac{\sum E}{n}$$

E = Data Aktual – Data Prediksi
 n = Jumlah Data

(2) Mean Absolute Error (MAE)

Rata-rata absolut dari margin selisih antara data yang diprediksi dengan data aktual yang menghasilkan seberapa besar nilai kesalahan prediksi. Rumus perhitungan Mean Absolute Error dapat dilihat pada rumus dibawah ini:

$$MAE = \left| \frac{\sum E}{n} \right|$$

E = Data Aktual – Data Prediksi
 n = Jumlah Data

3.4.3 Algoritma Long Short-Term Memory

Alih-alih unit yang hanya menerapkan nonlinieritas elementer pada transformasi affine dari input dan unit berulang, jaringan berulang LSTM memiliki "sel LSTM" yang memiliki perulangan internal (loop mandiri), di samping perulangan eksternal RNN. Setiap sel memiliki input dan output yang sama dengan jaringan berulang biasa, tetapi memiliki lebih banyak parameter dan sistem unit gating yang mengontrol aliran informasi. Gambar 3.5 merupakan formula dari LSTM:

$$f_i^{(t)} = \sigma \left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)} \right),$$

$x^{(t)}$ adalah vektor input saat ini dan $h^{(t)}$ adalah vektor lapisan tersembunyi saat ini, yang berisi output dari semua sel LSTM, dan b^f, U^f, W^f adalah bias, bobot masukan, dan bobot berulang untuk *forget gates*. Status internal sel LSTM dapat diperbarui, tetapi dengan bobot loop mandiri bersyarat $f_i^{(t)}$. Gambar 3.6 merupakan rumus *self-loop weight*:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma \left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)} \right),$$

b, U dan W masing-masing menunjukkan bias, bobot masukan, dan bobot berulang ke dalam sel LSTM. Unit gerbang input eksternal $g_i^{(t)}$ dihitung dengan cara yang sama dengan *forget gates* (dengan unit sigmoid untuk mendapatkan nilai gerbang antara 0 dan 1), tetapi dengan parameternya sendiri. Gambar 3.7 merupakan formula untuk *external input gate*:

$$g_i^{(t)} = \sigma \left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)} \right).$$

Keluaran $h_i^{(t)}$ dari sel LSTM juga dapat dimatikan, melalui gerbang keluaran $q_i^{(t)}$, yang juga menggunakan unit sigmoid untuk gating. Gambar 3.8 merupakan formula rumus *output gate*:

$$h_i^{(t)} = \tanh \left(s_i^{(t)} \right) q_i^{(t)}$$

$$q_i^{(t)} = \sigma \left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)} \right)$$

Parameter b^o, U^o, W^o memiliki bias, bobot input, dan bobot berulang. Di antara variannya, dapat memilih untuk menggunakan status sel $i - th$. Namun untuk melakukan hal ini 3 parameter tambahan diperlukan. Jaringan LSTM dapat mempelajari ketergantungan jangka panjang dengan lebih mudah dibandingkan dengan arsitektur berulang yang sederhana, pertama-tama pada kumpulan data buatan yang dirancang untuk menguji kemampuan mempelajari ketergantungan jangka panjang.





3.4.4 Algoritma Regresi Linear

Regresi Linear Sederhana adalah sebuah metode sederhana untuk memprediksi nilai variabel dependen Y berdasarkan satu variabel dependen X. Berikut adalah penulisan formula regresi linier sederhana:

$$Y \approx b_0 + b_1X$$

Y atau dependent variable, adalah akibat dari suatu sebab, misalnya: kenaikan gaji karyawan (terhadap lama pengalaman kerja), atau tinggi/rendah ranking murid (terhadap lama waktu belajarnya).

X atau independent variable, adalah hal yang diasumsikan menjadi sebab atas suatu hal, yang nantinya bisa mengakibatkan Y.

b_1 atau coefficient, adalah suatu unit / proporsi yang dapat mengubah nilai “x”

b_0 atau constant, adalah nilai awal “x” pada suatu kejadian.

4. HASIL

4.1 Exploratory Data Analysis (EDA)

Berikut adalah atribut data saham Twitter yang didapat melalui kegggle.com pada bulan September 2022:

Tabel 2
Nilai Sentiment Rating

No	Field	Keterangan
1	Timestamp	Waktu ketika harga saham tercatat
2	Open	Harga pembukaan saham
3	Low	Harga terendah saham
4	High	Harga tertinggi saham
5	Close	Harga penutupan saham
6	Volume	Total jumlah saham yang diperdagangkan

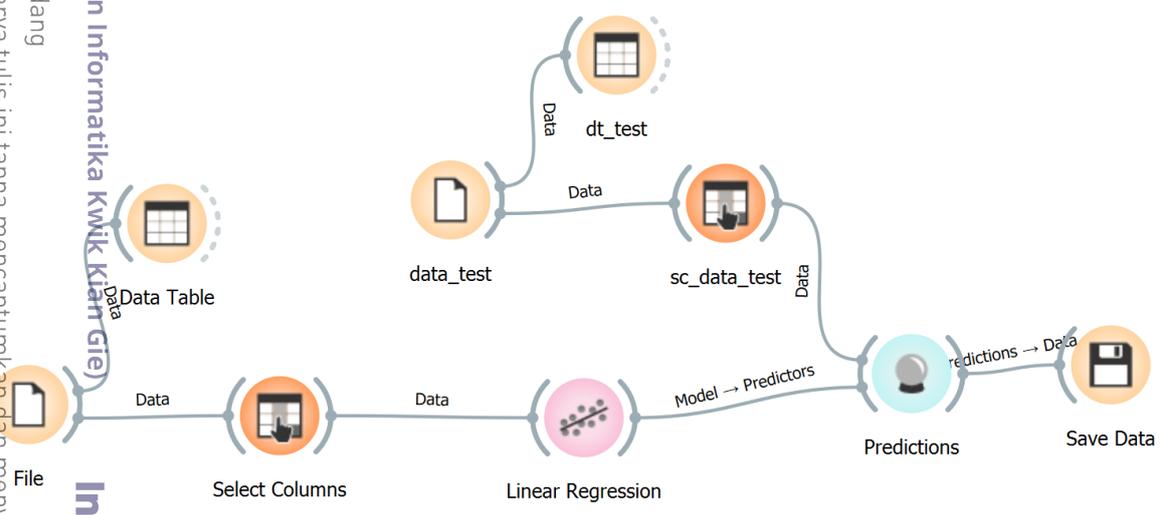
Data yang telah diperoleh akan dilakukan preprocess terlebih dahulu di aplikasi Microsoft Excel, setelah data melalui tahap preprocess data kemudian dapat digunakan sebagai data train dan data test pada bahasa pemrograman Python, maupun aplikasi Orange. Langkah terakhir peneliti

akan melakukan proses perancangan GUI (Graphic User Interface) menggunakan Google Site untuk melakukan visualisasi terhadap data agar dapat mudah dibaca.

4.2 Workflow Regresi Linear

Pada gambar 2, merupakan gambar workflow dari algoritma regresi linear yang digunakan pada aplikasi Orange. *Widget file* digunakan untuk melakukan impor terhadap *data train* untuk melakukan pelatihan model, dan *data test* untuk melakukan pengujian data, selanjutnya *widget select columns* digunakan untuk melakukan pemilihan fitur dan melakukan penyeleksian attribut pada *file* yang telah diimpor. Kemudian, *widget linear regression* akan melakukan pemrosesan prediksi menggunakan algoritma regresi linear yang disambungkan dengan widget *predictions* untuk mendapatkan hasil dari prediksi harga saham BBRI. Data kemudian disimpan menggunakan *widget save data*. Proses-proses tersebut dilakukan dengan memanfaatkan *widget* yang tersedia didalam aplikasi.

Gambar 2
Workflow Regresi Linear



- a. Penulisan kritik dan tinjauan suatu masalah.
- b. Penugutan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



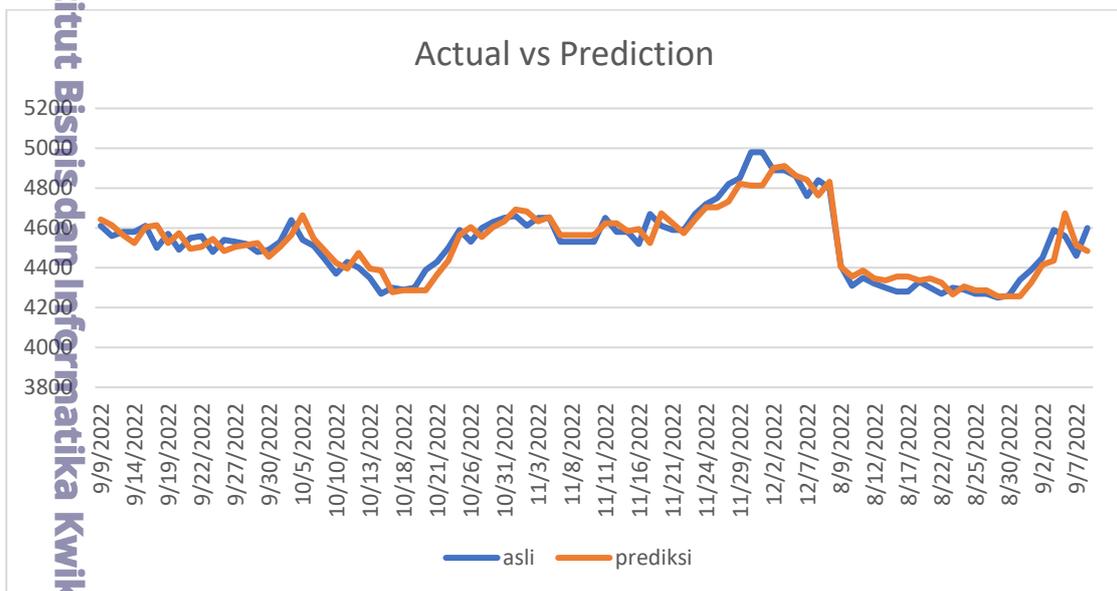
4.2.1 Hasil Regresi Linear Orange

Hasil prediksi dan evaluasi metriks yang telah dilakukan kemudian disimpan kedalam file berformat excel dan dijadikan tabel, kemudian direpresentasikan menggunakan visual *Line Chart* sehingga mudah untuk dilakukan analisis. Dapat dilihat pada tabel 3, 4 dan gambar 3 merupakan hasil dari proses yang telah dilakukan.

Tabel 3
Tampilan Sampel Tabel Regresi Linear - Orange

open	close	timestam	Linear Reg
continuou	continuou	string	continuou
	class	meta	meta
4640	4610	9/9/2022	4643.346
4610	4560	9/12/2022	4613.608
4560	4580	9/13/2022	4564.046
4520	4580	9/14/2022	4524.396
4600	4610	9/15/2022	4603.696
4610	4500	9/16/2022	4613.608
4520	4570	9/19/2022	4524.396
4570	4490	9/20/2022	4573.958
4490	4550	9/21/2022	4494.658
4500	4560	9/22/2022	4504.571
4540	4480	9/23/2022	4544.221
4480	4540	9/26/2022	4484.746
4500	4530	9/27/2022	4504.571
4510	4520	9/28/2022	4514.483
4520	4480	9/29/2022	4524.396
4450	4490	9/30/2022	4455.008
4500	4530	10/3/2022	4504.571

Gambar 3
Line Chart Regresi Linear - Orange





Tabel 4
Hasil Evaluasi Regresi Linear Orange

Deskripsi	Scoring
	Regresi Linear
RMSE	286.992
ME	-
MAE	226.076
MAPE	-
MPE	-

4.3 Python

Peneliti menggunakan bahasa pemrograman Python untuk melakukan prediksi harga BBRI selama 1 bulan kedepan, peneliti menggunakan algoritma Long Short-Term Memory (LSTM) dan Regresi Linear dengan Google Collabs sebagai Intergrated Development Environment (IDE). Berikut ini adalah pembahasan tahapan yang peneliti lakukan beserta dengan source code:

4.3.1 Source Code LSTM

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM

# Load the training and testing data
data_train = pd.read_csv('data_train.csv')
data_test = pd.read_csv('data_test.csv')
# Define the number of previous days to use as input features
look_back = 60
# Create input features by selecting 'close' column from the training data
train_data = data_train['close'].values.reshape(-1, 1)
# Split the data into training and testing sets
test_data = data_test['close'].values.reshape(-1, 1)
# Normalize the training data
sc = MinMaxScaler(feature_range=(0, 1))
train_data_norm = sc.fit_transform(train_data)
# Create input features and target variables
X_train, y_train = [], []
for i in range(look_back, len(train_data_norm)):
    X_train.append(train_data_norm[i-look_back:i, 0])
    y_train.append(train_data_norm[i, 0])
```

1. Dilarang mengutip sebagian atau seluruh karya atau hasil penelitian, baik berbentuk tulisan, gambar, audio, video, atau bentuk lainnya, tanpa izin IBIKKG.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



```
X_train, y_train = np.array(X_train), np.array(y_train)
# Reshape the input features for LSTM layer
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))

# Define the LSTM model
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1], 1)))
model.add(LSTM(units=50))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the LSTM model
model.fit(X_train, y_train, epochs=50, batch_size=32)

# Create input features for the testing data by using look_back days from the training data
inputs = np.concatenate((train_data[-look_back:, :], test_data), axis=0)
inputs_norm = sc.transform(inputs)
X_test = []
for i in range(look_back, len(inputs_norm)):
    X_test.append(inputs_norm[i-look_back:i, 0])
X_test = np.array(X_test)

# Reshape the input features for LSTM layer
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))

# Make predictions on the testing data
y_pred = model.predict(X_test)
y_pred = sc.inverse_transform(y_pred)
y_test = test_data

# Evaluate the model
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
me = np.mean(y_test - y_pred)
mae = mean_absolute_error(y_test, y_pred)
mpe = np.mean((y_test - y_pred) / y_test) * 100
mape = np.mean(np.abs((y_test - y_pred) / y_test)) * 100
print('RMSE:', rmse)
print('ME:', me)
print('MAE:', mae)
print('MPE:', mpe)
print('MAPE:', mape)

# Plot the actual and predicted prices
plt.figure(figsize=(16,8))
plt.plot(data_test['timestamp'], y_test, color='blue', label='Actual')
plt.plot(data_test['timestamp'], y_pred, color='red', label='Predicted Price')
plt.title('Real vs Predicted Prices - LSTM')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.show()
```

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

```
from numpy import savetxt
savetxt('lstm_predicted.csv', y_pred, fmt="%s")
```

4.3.2 Hasil LSTM Python

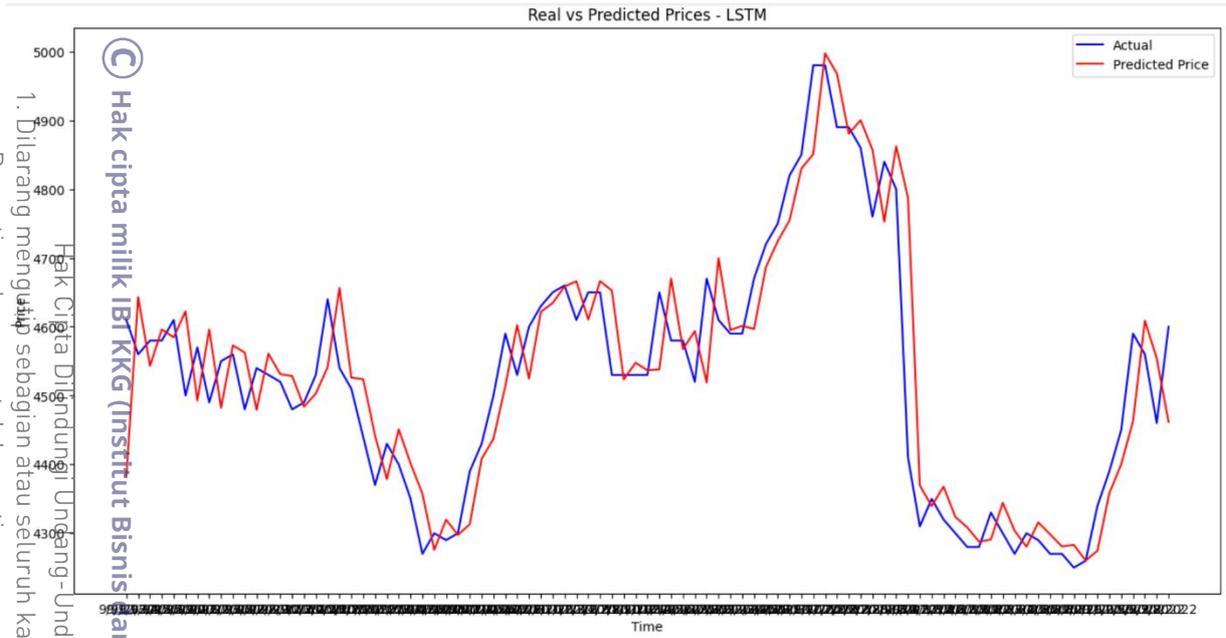
Hasil prediksi dan evaluasi metrik yang telah dilakukan kemudian disimpan kedalam file berformat excel dan dijadikan tabel, kemudian direpresentasikan menggunakan visual Line Chart sehingga mudah untuk dilakukan analisis. Dapat dilihat pada tabel 5, tabel 6 dan gambar 4 merupakan hasil dari proses yang telah dilakukan.

Tabel 5
Tampilan Sampel Tabel LSTM - Python

4381.931	4562.441
4643.048	4479.167
4543.092	4560.853
4596.171	4531.125
4584.677	4528.489
4622.497	4483.9
4492.886	4502.894
4596.101	4541.401
4482.285	4656.515
4573.005	4526.148



Gambar 4
Line Chart LSTM - Python



Tabel 6
Hasil Evaluasi LSTM - Python

Deskripsi	Scoring
	LSTM
RMSE	77.909
ME	-5.085
MAE	56.561
MAPE	1.245
MPE	-0.127

4.3.3 Source Code Regresi Linear

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error,
mean_absolute_percentage_error
import matplotlib.pyplot as plt
# Load the training and testing datasets
train_df = pd.read_csv('data_train.csv')
test_df = pd.read_csv('data_test.csv')
# Drop rows with missing values
train_df.dropna(inplace=True)
test_df.dropna(inplace=True)
# Select only the close price column
X_train = train_df.drop(['timestamp', 'close'], axis=1).values
y_train = train_df['close'].values
```

Hak Cipta milik IBIKKG (Institut Bisnis dan Informatika Kwik Kian Gie)

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber atau mengutipnya tanpa izin IBIKKG.

a. Penguatiran hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Penguatiran tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



```
X_test = test_df.drop(['timestamp', 'close'], axis=1).values
y_test = test_df['close'].values
# Train the model
reg = LinearRegression()
reg.fit(X_train, y_train)

# Make predictions
y_pred = reg.predict(X_test)
# Evaluate the model
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
mae = mean_absolute_error(y_test, y_pred)
mape = mean_absolute_percentage_error(y_test, y_pred)

print(f'RMSE: {rmse}')
print(f'MAE: {mae}')
print(f'MAPE: {mape}')

plt.figure(figsize=(20,10))
plt.plot(test_close, 'red', label='Real Prices')
plt.plot(predicted_value, 'blue', label='Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Prices')
plt.title('Real vs Predicted Prices')
plt.legend(loc='best', fontsize=20)

# Plot the results
plt.figure(figsize=(20,10))
plt.plot(y_test, 'red', label='Actual')
plt.plot(y_pred, 'blue', label='Predicted')
plt.xlabel('Time')
plt.ylabel('Prices')
plt.title('Real vs Predicted Prices - Linaer Regression')
plt.legend()
plt.show()

from numpy import savetxt
savetxt('regresiLinear_predicted.csv', y_pred, fmt="%s")
```

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.
- b. Penguji tidak merugikan kepentingan yang wajar IBIKKG.
- penulisan kritik dan tinjauan suatu masalah.

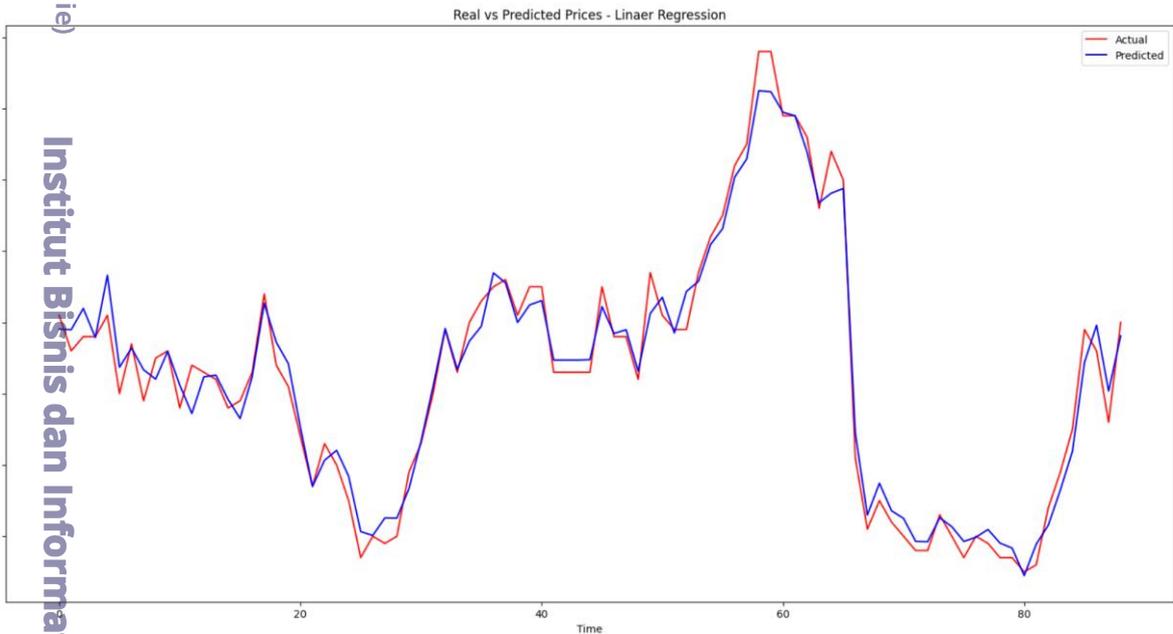
4.3.4 Hasil Regresi Linear Python

Hasil prediksi dan evaluasi metriks yang telah dilakukan kemudian disimpan kedalam file berformat excel dan dijadikan tabel, kemudian direpresentasikan menggunakan visual Line Chart sehingga mudah untuk dilakukan analisis. Dapat dilihat pada tabel 7, tabel 8 dan gambar 5 merupakan hasil dari proses yang telah dilakukan:

Tabel 7
Tampilan Sampel Tabel Regresi Linear - Python

4590.641	4512.072
4589.736	4472.16
4619.623	4523.725
4578.763	4525.874
4666.084	4492.138
4537.045	4465.144
4564.133	4523.58
4533.524	4626.799
4520.441	4572.228
4559.465	4542.057

Gambar 5
Line Chart Regresi Linear - Python





Tabel 8
Hasil Evaluasi LSTM - Python

Deskripsi	Scoring
	Regresi Linear
RMSE	26.634
ME	-
MAE	21.649
MAPE	0.004
MPE	-

5. PEMBAHASAN

Hasil dari penelitian ini adalah penerapan metode *Data Mining* untuk melakukan prediksi harga saham BBRI pada periode 2001 – 2022 dengan menggunakan algoritma Long Short-Term Memory (LSTM), dan Regresi Linear. Hal ini digunakan untuk melihat atribut naik atau turunnya saham. Berdasarkan jurnal-jurnal penelitian terdahulu, peneliti menjadikan sumber data sekunder (Khalis Sofi, Aswan Supriyadi Sunge, Sasmitoh Rahmad Riady, dan Antika Zahrotul, 2021; Steven Sen, Dedy Sugiarto, dan Abdul Rochman, 2020; Adhib Arfan, dan Lussiana ETP, 2020; Syakirah Fachid, dan Agung Triayudi, 2022; Ahmad Ashril Rizal, Siti Soraya, 2018) aplikasi data mining, dataset, atribut dan lain-lain yang disajikan dalam penelitian ini berbeda, tetapi kesamaan dengan penelitian-penelitian terdahulu adalah algoritma data mining yang sama dengan yang peneliti gunakan saat ini.

6. Kesimpulan

Hasil yang diperoleh pada penggunaan model Long Short-Term Memory dan Regresi Linear pada Python maupun aplikasi Orange untuk melakukan prediksi terhadap saham BBRI dapat disimpulkan sebagai berikut:

1. Algoritma Regresi Linear menggunakan Python lebih akurat dalam melakukan prediksi terhadap harga saham BBRI dibandingkan algoritma Long Short-Term Memory (LSTM) yang diolah melalui Python, maupun algoritma Regresi Linear menggunakan Orange.
2. Berdasarkan data train pada harga saham BBRI dapat disimpulkan bahwa algoritma Regresi Linear menggunakan Python memiliki hasil Root Mean Square Error (RMSE) yang lebih kecil dengan skor sebesar 26.634 apabila dibandingkan dengan algoritma Long Short-Term Memory (LSTM).

Selama proses penelitian berlangsung peneliti menemukan saran yang dapat dilakukan oleh peneliti yang akan datang, antara lain:

1. Untuk peneliti berikutnya, dapat menggunakan algoritma lain sebagai pembanding. Seperti Neural Network, atau Gated Recurrent Unit (GRU).
2. Aplikasi untuk peneliti berikutnya dapat menggunakan aplikasi serupa selain Orange Data Mining, seperti RapidMiner, Knime dan WEKA.



DAFTAR PUSTAKA

- Arfan, A. & Etp, L. (2020). Perbandingan Algoritma Long Short-Term Memory dengan SVR Pada Prediksi Harga Saham di Indonesia, Universitas Gunadarma, Jakarta.
- Aggarwal, C. C. (2015). Data Mining: The Textbook. Cham: Springer International Publishing.
- Dubey, U. K. B., & Kothari, D. P. (2022). Research Methodology: Techniques and Trends (1st ed.). Boca Raton: Chapman and Hall/CRC.
- Fachid, S., & Triayudi, A. (2022). Perbandingan Algoritma Regresi Linier dan Regresi Random Forest Dalam Memprediksi Kasus Positif Covid-19. Jurnal Media Informatika Budidarma, Universitas Nasional, Jakarta.
- Goodfellow, Ian. et al. (2017). Deep Learning. MIT University
- Ibnu Daqiqil ID. (2021). : Teori, Studi Kasus dan Implementasi Menggunakan Python (Version 1). UR Press, Riau.
- Khalis Sofi, Aswan Supriyadi Sunge, Sasmitoh Rahmad Riady, & Antika Zahrotul Kamalia. (2021). Perbandingan Algoritma Linear Regression, LSTM, dan GRU Dalam Memprediksi Harga Saham Dengan Model Time Series, Universitas Pelita Bangsa, Bekasi.
- Marakas, G. M., O'Brien, J. A., & O'Brien, J. A. (2013). Introduction to information systems (16th ed). New York, NY: McGraw-Hill/Irwin
- Mladjenovic, Paul. Stock Investing for Dummies. New Jersey: John Wiley & Sons, Inc.
- Muller, Andreas C., Sarah Guido. Introduction To With Python. United States: O'Reilly Media, Inc.
- Ning-Tan, Pang. et al. (2019). Introduction to Data Mining Second Edition. United Kingdom: Pearson Education Limited.
- Orangedatamining.com (2023). About Orange. Diakses pada 15 Februari 2023, dari <https://orangedatamining.com/faq/>
- Orangedatamining.com (2023). Add-ons Extend Functionality. Diakses pada 15 Februari 2023, dari https://orangedatamining.com/home/add-ons_Extend_Functionality
- Orangedatamining.com (2023). Interactive Data Visualization. Diakses pada 15 Februari 2023, dari https://orangedatamining.com/home/interactive_data_visualization/
- Orangedatamining.com (2023). Visual Programming. Diakses pada 15 Februari 2023, dari https://orangedatamining.com/home/visual-_programming/
- Pajankar, A., & Joshi, A. (2022). Hands-on with Python: Implement Neural Network Solutions with Scikit-learn and PyTorch. Berkeley, CA: Apress.
- Reynolds, G. W. (2017). Fundamentals of information systems (9th edition). Mason, OH: Cengage.
- Rizal, A. A., & Soraya, S. (2018). Multi Time Steps Prediction dengan Recurrent Neural Network Long Short Term Memory, STMIK Bumigora Mataram.
- Sen, S., Sugiarto, D., & Rochman, A. (n.d.). Komparasi Metode Multilayer Perceptron (MLP) dan Long Short Term Memory (LSTM) dalam Peramalan Harga Beras, Universitas Trisakti, Jakarta.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

© Hak Cipta dilindungi undang-undang. Dilarang diperjualbelikan atau digunakan untuk kepentingan komersial tanpa izin tertulis dari Universitas Kwik Kian Gie.

T. Larose, Daniel. dan D. Larose, Chantal. (2015). Data Mining and Predictive Analytics. New Jersey: John Wiley & Sons, Inc.

Zainal A. Hasibuan, PhD. (2007). Analisis Data Untuk Riset Manajemen dan Bisnis. Indonesia: USU Press 2014

Hak Cipta Dilindungi Undang-Undang

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 dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Institut Bisnis dan Informatika Kwik Kian Gie

