In [1]:
```
# Import tensorflow
import tensorflow as tf
tf.__version__
```

Out[1]:
```
'2.12.0'
```

In [2]:
```
# Import library
import math
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
import yfinance as yf
```

In [3]:
```
# Install yfinance
!pip install yfinance
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: yfinance in c:\users\hend\appdata\roaming\python\python310\site-packages (0.2.20)
Requirement already satisfied: pandas>=1.3.0 in c:\anaconda3\lib\site-packages (from yfinance) (1.5.3)
Requirement already satisfied: numpy>=1.16.5 in c:\anaconda3\lib\site-packages (from yfinance) (1.23.5)
Requirement already satisfied: requests>=2.26 in c:\anaconda3\lib\site-packages (from yfinance) (2.28.1)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\hend\appdata\roaming\python\python310\site-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in c:\anaconda3\lib\site-packages (from yfinance) (4.9.1)
Requirement already satisfied: appdirs>=1.4.4 in c:\anaconda3\lib\site-packages (from yfinance) (1.4.4)
Requirement already satisfied: pytz>=2022.5 in c:\anaconda3\lib\site-packages (from yfinance) (2022.7)
Requirement already satisfied: frozendict>=2.3.4 in c:\users\hend\appdata\roaming\python\python310\site-packages (from yfinance) (2.3.8)
Requirement already satisfied: cryptography>=3.3.2 in c:\anaconda3\lib\site-packages (from yfinance) (39.0.1)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\anaconda3\lib\site-packages (from yfinance) (4.11.1)
Requirement already satisfied: html5lib>=1.1 in c:\users\hend\appdata\roaming\python\python310\site-packages (from yfinance) (1.1)
Requirement already satisfied: soupsieve>1.2 in c:\anaconda3\lib\site-packages (from beautifulsoup4>=4.11.1->yfinance) (2.3.2.post1)
Requirement already satisfied: cffi>=1.12 in c:\anaconda3\lib\site-packages (from cryptography>=3.3.2->yfinance) (1.15.1)
Requirement already satisfied: six>=1.9 in c:\anaconda3\lib\site-packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings>=0.4 in c:\anaconda3\lib\site-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\anaconda3\lib\site-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\anaconda3\lib\site-packages (from requests>=2.26->yfinance) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\anaconda3\lib\site-packages (from requests>=2.26->yfinance) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\anaconda3\lib\site-packages (from requests>=2.26->yfinance) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in c:\anaconda3\lib\site-packages (from requests>=2.26->yfinance) (2022.12.7)
Requirement already satisfied: pycparser in c:\anaconda3\lib\site-packages (from cffi>=1.12->cryptography>=3.3.2->yfinance) (2.21)
```

In [4]:
```
# Download BBRI.JK File
A_df = yf.download('BBRI.JK')
```

```
[*********************100%***********************]  1 of 1 completed
```

In [5]:
```
# Show dataset
A_df
```

Out[5]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2003-11-10	105.0	110.0	95.0	97.5	53.242930	5144140000
2003-11-11	97.5	100.0	95.0	100.0	54.710700	2938860000
2003-11-12	97.5	105.0	97.5	105.0	57.442243	2002910000
2003-11-13	105.0	105.0	102.5	105.0	57.442243	1190000000
2003-11-14	105.0	105.0	100.0	105.0	57.442243	1752000000
...	...	...	...	...	...	...
2023-07-03	5500.0	5550.0	5450.0	5475.0	5475.000000	168593000
2023-07-04	5500.0	5550.0	5425.0	5450.0	5450.000000	94462600
2023-07-05	5400.0	5400.0	5400.0	5400.0	5400.000000	109632000
2023-07-06	5450.0	5475.0	5400.0	5425.0	5425.000000	109406200
2023-07-07	5400.0	5425.0	5375.0	5375.0	5375.000000	114843800

4879 rows × 6 columns

In [6]:
```
# Import datetime
import datetime
```

In [7]:
```
# Install Datetime
!pip install datetime
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: datetime in c:\users\hend\appdata\roaming\python\python310\site-packages (5.1)
Requirement already satisfied: zope.interface in c:\anaconda3\lib\site-packages (from datetime) (5.4.0)
Requirement already satisfied: pytz in c:\anaconda3\lib\site-packages (from datetime) (2022.7)
Requirement already satisfied: setuptools in c:\anaconda3\lib\site-packages (from zope.interface->datetime) (65.6.3)
```

In [8]:
```
# Time range
start = datetime.datetime(2019,1,20)
end = datetime.datetime(2023,6,15)
```

In [9]:
```
# Download BBRI.JK File
A_df = yf.download('BBRI.JK', start=start, end=end)
```

```
[*********************100%***********************]  1 of 1 completed
```

In [10]:
```
# Show dataset
A_df
```

Out[10]:

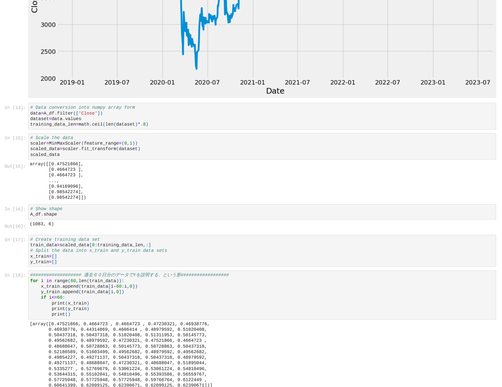
	Open	High	Low	Close	Adj Close	Volume
Date						
2019-01-21	3800.0	3800.0	3800.0	3800.0	3149.417236	80026000
2019-01-22	3770.0	3790.0	3740.0	3770.0	3124.553223	119687600
2019-01-23	3760.0	3810.0	3730.0	3770.0	3124.553223	165406700
2019-01-24	3750.0	3810.0	3750.0	3790.0	3141.129150	176108500
2019-01-25	3820.0	3820.0	3780.0	3780.0	3132.841064	137141300
...	...	...	...	...	...	...
2023-06-08	5375.0	5475.0	5350.0	5475.0	5475.000000	92794100
2023-06-09	5425.0	5450.0	5375.0	5425.0	5425.000000	77251000
2023-06-12	5400.0	5400.0	5375.0	5400.0	5400.000000	92314200
2023-06-13	5400.0	5550.0	5400.0	5550.0	5550.000000	109831400
2023-06-14	5575.0	5575.0	5475.0	5550.0	5550.000000	79137600

1003 rows × 6 columns

In [11]:
```
# Show shape
A_df.shape
```

Out[11]:
```
(1003, 6)
```

In [12]:
```
# Plot the prediction results on the training data
plt.figure(figsize=(16,8))
plt.title('Close Price History')
plt.plot(A_df['Close'])
plt.xlabel('Date',fontsize=18)
plt.ylabel('Close Price RP(RP)',fontsize=18)
plt.show()
```



In [13]:
```
# Data conversion into numpy array form
data=A_df.filter(['Close'])
dataset=data.values
training_data_len=math.ceil(len(dataset)*.8)
```

In [14]:
```
# Scale the data
scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(dataset)
scaled_data
```

Out[14]:
```
array([[0.47521866],
       [0.4864723 ],
       [0.4864723 ],
       ...,
       [0.94168096],
       [0.98542274],
       [0.98542274]])
```

In [15]:
```
# Show shape
A_df.shape
```

Out[15]:
```
(1003, 6)
```

In [16]:
```
# Create training data set
train_data=scaled_data[0:training_data_len,:]
# Split the data into x_train and y_train data sets
x_train=[]
y_train=[]
```

In [18]:
```
################### 過去60日的データ→ 2天分を訓練する。これっ!#################
for i in range(60,len(train_data)):
    # x_train.append(train_data[i-60:i,0])
    x_train.append(train_data[i,0])
    if i<=60:
        print(x_train)
        print(y_train)
        print()
```

```
[array([0.47521866, 0.4864723 , 0.4864723 , 0.4723032 , 0.49939776,
       0.49939776, 0.49939776, 0.49939776, 0.49939776, 0.49939776,
       0.50457516, 0.50457516, 0.51031063, 0.51031063, 0.50149771,
       0.49564282, 0.49979503, 0.47230321, 0.47521866, 0.4664723 ,
       0.46608047, 0.50729862, 0.50149771, 0.50729862, 0.50457516,
       0.52386589, 0.51803499, 0.49562682, 0.49979502, 0.49562682,
       0.49562682, 0.49562682, 0.49562682, 0.49562682, 0.49562682,
       0.49271137, 0.49608047, 0.47230321, 0.49608047, 0.51899594,
       0.5335277 , 0.52706979, 0.53061224, 0.53061224, 0.54818496,
       0.53644315, 0.55325941, 0.54818496, 0.55335166, 0.56535767,
       0.5720948 , 0.57225948, 0.57225948, 0.59768764, 0.6122449 ,
       0.60441905, 0.63599125, 0.62306671, 0.62306671, 0.62306671])]
[0.63568053119933394]
```

In [19]:
```
# Show shape
A_df.shape
```

Out[19]:
```
(1003, 6)
```

In [21]:
```
#convert train data to numpy arrays
x_train,y_train=np.array(x_train),np.array(y_train)
```

In [22]:
```
#Reshape
x_train.shape
```

Out[22]:
```
(807, 60)
```

In [23]:
```
# Train Shape
x_train=np.reshape(x_train,(x_train.shape[0],x_train.shape[1],1))
x_train.shape
```

Out[23]:
```
(807, 60, 1)
```

In [24]:
```
# Build LSTM model
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(x_train.shape[1],1)))
model.add(LSTM(50,return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
```

In [25]:
```
# Compile the model
model.compile(optimizer='adam',loss='mean_squared_error')
```

In [26]:
```
# Train the model
model.fit(x_train,y_train,batch_size=1,epochs=1)
```

```
807/807 [==============================] - 15s 13ms/step - loss: 0.0057
```

Out[26]:
```
<keras.callbacks.History at 0x2aaa32841f0>
```

In [27]:
```
# Create the testing data set
test_data=scaled_data[training_data_len-60:,:]
```

In [28]:
```
# Create the data sets x_test and y_test
x_test=[]
y_test=dataset[training_data_len:,:]
for i in range(60,len(test_data)):
    x_test.append(test_data[i-60:i,0])
```

In [29]:
```
# Convert the data to a numpy array
x_test=np.array(x_test)
```

In [30]:
```
# Reshape
x_test=np.reshape(x_test,(x_test.shape[0],x_test.shape[1],1))
```

In [31]:
```
# Get the models predicted price values
predictions=model.predict(x_test)
predictions=scaler.inverse_transform(predictions)
```

```
7/7 [==============================] - 1s 12ms/step
```

In [32]:
```
# Get the root mean squared error(RMSE)
rmse=np.sqrt np.mean((predictions - y_test)**2))
```

In [33]:
```
# Print RMSE
rmse
```

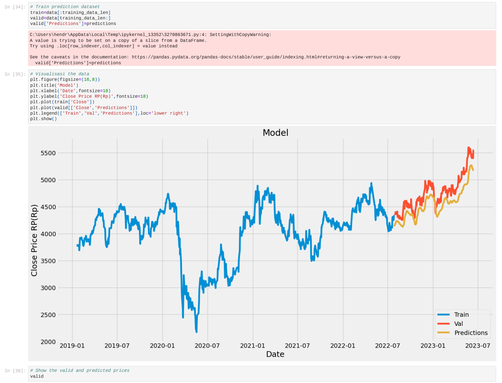
Out[33]:
```
233.1425021064845
```

In [34]:
```
# Train prediction dataset
train=data[:training_data_len]
valid=data[training_data_len:]
valid['Predictions']=predictions
```

```
C:\Users\hend\AppData\Local\Temp\ipykernel_13352\3270863671.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  valid['Predictions']=predictions
```

In [35]:
```
# Visualiseo the data
plt.figure(figsize=(16,8))
plt.title('Model')
plt.xlabel('Date',fontsize=18)
plt.ylabel('Close Price RP(Rp)',fontsize=18)
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Val', 'Predictions'],loc='lower right')
plt.show()
```



In [36]:
```
# Show the valid and predicted prices
valid
```

Out[36]:

	Close	Predictions
Date		
2022-07-20	4360.0	4135.090703
2022-07-21	4360.0	4107.710727
2022-08-01	4360.0	4177.507324
2022-08-02	4400.0	4383.425293
2022-08-03	4360.0	4309.642211
...	...	...
2023-06-09	5475.0	5231.413574
2023-06-12	5425.0	5210.260604
2023-06-13	5400.0	5192.645020
2023-06-13	5550.0	5173.098219
2023-06-14	5550.0	5174.261719

216 rows × 2 columns

In [37]:
```
# Get the quote
apple_quote=A_df = yf.download('BBRI.JK')
```

```
[*********************100%***********************]  1 of 1 completed
```

In [38]:
```
# Create a new dataframe
new_df=apple_quote.filter(['Close'])
```

In [39]:
```
# Get the last 60 day closing price values and convert the dataframe to an array
last_60_days=new_df[-60:].values
```

In [40]:
```
# Scale the data to be values between 0 and 1
last_60_days_scaled=scaler.transform(last_60_days)
```

In [41]:
```
#Create an empty list
X_test=[]
```

In [42]:
```
# Append the past 60 days
X_test.append(last_60_days_scaled)
```

In [43]:
```
# Convert the x_test data to a numpy array
X_test=np.array(X_test)
```

In [44]:
```
# Reshape
X_test=np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
```

In [45]:
```
# Get the predicted scaled price
pred_price=model.predict(X_test)
```

```
1/1 [==============================] - 0s 22ms/step
```

In [46]:
```
# Undo the scaling
pred_price=scaler.inverse_transform(pred_price)
print(pred_price)
```

```
[[5149.7866]]
```

In [47]:
```
# Get the quote
apple_quote2=A_df = yf.download('BBRI.JK')
print(apple_quote2['Close'])
```

```
[*********************100%***********************]  1 of 1 completed
Date
2003-11-10    97.5
2003-11-11   100.0
2003-11-12   105.0
2003-11-13   105.0
2003-11-14   105.0
              ...
2023-07-03  5475.0
2023-07-04  5450.0
2023-07-05  5400.0
2023-07-06  5425.0
2023-07-07  5375.0
Name: Close, Length: 4879, dtype: float64
```

In [ ]: