
APLIKASI PERBANDINGAN PREDIKSI HARGA BITCOIN MENGUNAKAN DEEP LEARNING DENGAN METODE ARIMA, SARIMA, LTSM DAN GRADIENT BOOSTING REGRESSOR

Agung Slamet Riyadi¹, Ire Puspa Wardhani², Irfan² dan Andi Perdana¹

⁽¹⁾Universitas Gunadarma

Jl. Margonda Raya 100, Depok 16424 Jawa Barat

⁽²⁾STMIK Jakarta STI&K

Jl. BRI No. 17, Radio Dalam, Kebayoran Baru, Jakarta Selatan 12140

agungsr@staff.gunadarma.ac.id, irepuspa@gmail.com, andiutama14@gmail.com,

irfan@staff.jak-stik.ac.id

ABSTRAK

Sebagai mata uang digital pertama di dunia yang muncul pada tahun 2009 Bitcoin menjadi yang paling populer saat ini selain Ethereum, Binance Coin, Tether, Solana dan sebagainya. Berkembangnya metode untuk melakukan prediksi harga mata uang digital termasuk Bitcoin semakin penting. Penelitian ini menggunakan algoritma deep learning dengan memanfaatkan metode Autoregressive Integrated Moving Average (ARIMA), Seasonal Autogresive Integrated Moving Average (SARIMA), Long Short Term Memory (LTSM) dan Gradient Boosting Regressor untuk memprediksi harga Bitcoin. Data diambil menggunakan dataset seperti harga penutupan harian, volume perdagangan, indikator teknis dan model yang digunakan untuk jaringan saraf buatan atau tiruan sehingga dapat menghasilkan prediksi yang lebih akurat dari empat metode tersebut. Dengan membuat aplikasi berbasis Neural Network yang dapat meningkatkan keakuratan prediksi harga bitcoin dalam memberikan informasi lebih akurat sehingga diketahui harga bitcoin kedepannya. Pengolahan data menggunakan aplikasi berbasis pemrograman python. Hasil penelitian ini menunjukkan bahwa dengan pendekatan deep learning mempunyai potensi untuk memiliki kemampuan memprediksi harga bitcoin sehingga dapat dimanfaatkan untuk para pedagang dan investor untuk mengambil keputusan yang lebih tepat dengan hasil penggunaan metode SARIMA lebih baik dibandingkan dengan metode lainnya dengan memiliki nilai lebih rendah atau tingkat kesalahan yang lebih rendah, yaitu dengan nilai RMSE 13692.22 dibandingkan dari metode ARIMA dengan RMSE 14490,55 metode LTSM dengan RMSE 19650,40 dan Gradient Boosting Regressor dengan RMSE 14573,20.

Kata Kunci: *Prediksi, Bitcoin, Deep Learning, ARIMA, SARIMA, LTSM, Gradient Boosting Regressor*

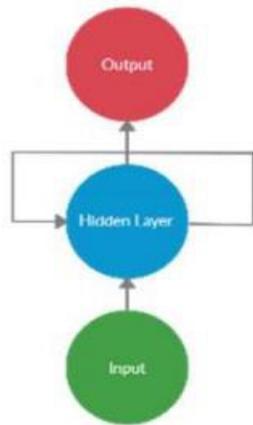
PENDAHULUAN

Mata uang digital Bitcoin mengalami perkembangan yang sangat luar biasa dan sangat populer dari nilai tukarnya. Bitcoin telah menarik perhatian para pedagang dan investor untuk mencari peluang pada pasar kripto yang dinamis. Bitcoin adalah mata uang elektronik yang dibuat pada tahun 2009 oleh Satoshi Nakamoto. Nama Bitcoin pertama kali dikenal oleh publik setelah di publikasikannya paper yang dibuat oleh penemu Bitcoin pada tahun 2009. Paper tersebut berjudul "Bitcoin: A peer-to-peer electronic cash system", yaitu membahas tentang algoritma dan teknologi yang digunakan oleh Bitcoin. Satoshi Nakamoto sendiri merupakan pseudonym atau nama

samaran, tidak diketahui apakah Satoshi Nakamoto adalah seorang individu atau sebuah kelompok. [15]. Bitcoin muncul dari kekecewaan Nakamoto terhadap krisis keuangan yang bermula pada tahun 2007-2008 dan reaksi pemerintah terhadap krisis tersebut. Dengan adanya Bitcoin, Nakamoto berharap untuk memutus pihak ketiga dalam transaksi yang dilakukan secara online, karena menurut Nakamoto selalu ada pihak ketiga yang mengambil keuntungan dalam tiap transaksi online yang dilakukan seseorang baik melalui PayPal, kartu kredit, atau Western Union. Solusi yang ditawarkan untuk mengganti pihak ketiga adalah dengan membuat jaringan sistem yang bernama Blockchain. [2]. Terdapat metode tradisional seperti

menggunakan analisis secara teknis yang telah digunakan untuk memprediksi harga Bitcoin. Namun dengan menggunakan metode tradisional yang mempunyai sifat yang tidak terstruktur dari data pada Bitcoin dapat menyulitkan penerapan dengan metode ini untuk mendapatkan keakuratan yang tinggi. Dalam perkembangannya saat ini pada bidang kecerdasan buatan seperti deep learning yang dapat digunakan untuk mengatasi permasalahan untuk memprediksi harga Bitcoin. [5]. Deep Learning sebagai bagian pengembangan Machine Learning dan Artificial Intelligence (AI) yang merupakan konsep tentang bagaimana membuat mesin-mesin yang dapat berpikir cerdas (smart machine) seperti manusia yang mampu belajar dari berbagai input dengan outcome yang akurat [11].

Recurrent Neural Network (RNN) adalah salah satu algoritma deep learning yang dapat digunakan untuk mengenali pola dan melakukan prediksi pada data numerik berbentuk time series, misalnya ramalan cuaca dan prediksi harga saham.[1].



Gambar 1. *Arsitektur RNN*

LSTM (Long Short Term Memory) merupakan Arsitektur RNN dalam bidang Deep Learning. LSTM memiliki koneksi feedback yang mampu memproses titik data tunggal dengan seluruh urutan data (seperti speech atau video). LSTM pertama kali diperkenalkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997 dan LSTM menjadi salah satu model yang saat ini banyak digunakan Deep Learning untuk Natural Language Processing (NLP) [13]. Penggunaan model LSTM pada arsitektur

Recurrent Neural Network (RNN) bertujuan untuk mempelajari pola data time series yang panjang dan kompleks yang diharapkan mampu menyelesaikan permasalahan vanishing gradient maupun permasalahan lainnya ketika RNN biasa tidak mampu menangani data panjang dan kompleks tersebut. Adapun perbedaan antara topik penelitian yang diambil penulis dengan beberapa penelitian dari peneliti sebelumnya seperti pada Jurnal Nasional yang berjudul Aplikasi Prediksi Jangka Pendek harga Bitcoin Menggunakan Metode ARIMA Jurnal Ilmiah Informatika Komputer p-ISSN: 0853-8638, e-ISSN: 2089-8045 Vol 25, No 3 (2020) terletak pada teknik ataupun model, periode data time series dan platform yang digunakan. [8].

Menurut Meyler, salah satu metode peramalan yang dikenal adalah ARIMA. Metode ARIMA disebutkan memiliki kelemahan dalam beberapa teknik dalam mengidentifikasi model dari beberapa kelas model yang sulit untuk dipahami dan biasanya mahal dilakukan secara komputasional. Selain itu proses ini juga bersifat subjektif dan reliabilitas model yang dipilih bergantung pada keahlian dan pengalaman orang yang melakukan peramalan tersebut.[3]

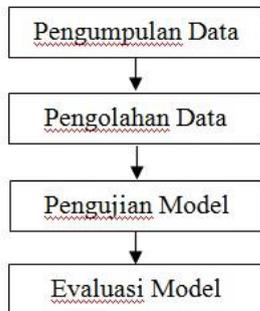
Salah satu metode untuk melakukan peramalan pada data time series yaitu Seasonal Autoregressive Integrated Moving Average (SARIMA). Metode SARIMA merupakan pengembangan metode ARIMA yang memiliki efek musiman. Kelebihan dari metode ini dapat menerima semua jenis pola time series meskipun dalam prosesnya harus distasionerkan terlebih dahulu. Sarima menggunakan nilai pada masa lalu dan masa sekarang dari variabel dependen untuk menghasilkan peramalan yang akurat. [16].

Gradient Boosting Regressor adalah model pembelajaran mesin yang dapat digunakan untuk masalah regresi, klasifikasi dan juga menghasilkan model prediksi yang terdiri dari ensemble model prediksi lemah pada pohon keputusan yang menghasilkan error prediksi yang sangat rendah saat menggunakan median sebagai metode prediksi.[14].

Tujuan dari penelitian ini adalah untuk meningkatkan keakuratan dalam memprediksi harga Bitcoin, sehingga bisa memberikan informasi yang lebih akurat bagaimana harga bitcoin kedepannya. Hasil dari penelitian ini adalah agar dapat memberikan pengetahuan mengenai prediksi harga Bitcoin di masa depan sehingga dapat mengambil keputusan yang lebih baik bagi para pedagang atau investor.

METODE PENELITIAN

Penelitian ini terdapat 4 tahap, yaitu pengumpulan data, pengolahan data, pengujian model, dan evaluasi model.



Gambar 2. Tahapan dalam penelitian

Pengumpulan Data

Pengambilan data dilakukan pada platform berupa harga bitcoin yang diperoleh dari situs indodax.com dari tahun 2012 sampai dengan tahun 2021 dengan total dataset lebih dari 1 juta data. Dataset yang diambil memiliki 7 buah atribut, yaitu Date, Open, High, Low, Close, Vol_btc, Vol_currency.

Adapun data yang dikumpulkan terangkum pada Tabel 1 sebagai berikut :

Tabel 1 : Jenis dan tipe Atribut data Bitcoin

No	Nama variabel Data	Jenis atribut	Type atribut
[1]	Timestamp	Numeric	Interval
[2]	Open	Numeric	Rasio
[3]	High	Numeric	Rasio
[4]	Low	Numeric	Rasio
[5]	Close	Numeric	Rasio
[6]	Vol_btc	Numeric	Rasio
[7]	Vol_currency	Numeric	Rasio

Pengolahan Data

Pada awalnya dilakukan *import library* pada Gambar 3 yaitu proses melakukan import pada data library yang nantinya akan diperlukan berupa dataset. Berikut pada Gambar 3 adalah proses pengolahan data yang dilakukan dengan memanfaatkan pemrograman Python.

```

[1]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

/kaggle/input/bitcoin-historical-data/bitstampUSD_1-min_data_2012-01-01_to_2021-03-31.csv

# Load the dataset
data = pd.read_csv('/kaggle/input/bitcoin-historical-data/bitstampUSD_1-min_data_2012-01-01-
+ Code + Markdown
  
```

Gambar 3 : Proses Import Library

Pada Gambar 4 dapat dilihat beberapa informasi harga pada saat itu seperti harga open, high, low, close, dan lainnya.

```

[3]: # Display the first few rows of the dataset
print(data.head())

Timestamp Open High Low Close Volume_(BTC) Volume_(Currency) \
0 1325179200 4.39 4.39 4.39 4.39 0.455581 2.0
1 1325179800 NaN NaN NaN NaN NaN NaN
2 1325180400 NaN NaN NaN NaN NaN NaN
3 1325181000 NaN NaN NaN NaN NaN NaN
4 1325181600 NaN NaN NaN NaN NaN NaN

Weighted_Price
0 4.39
1 NaN
2 NaN
3 NaN
4 NaN

[4]: # Check the dimensions of the dataset
print('Dataset Shape:', data.shape)

Dataset Shape: (4897377, 8)
  
```

Gambar 4. Informasi Harga Bitcoin

Gambar 5 berikut ini adalah data yang telah disertakan *summary statistics* sehingga dapat diketahui beberapa data seperti min dan max.

```

[4]: # Summary statistics
print('Summary Statistics:')
print(data.describe())

Summary Statistics:
Timestamp Open High Low Close \
count 4.857377e+06 3.613769e+06 3.613769e+06 3.613769e+06 3.613769e+06
mean 1.471361e+09 6.009024e+03 6.813257e+03 6.004482e+03 6.009014e+03
std 6.428023e+07 8.996247e+03 9.800321e+03 8.108773e+03 8.996160e+03
min 1.325114e+09 3.800000e+00 3.800000e+00 1.500000e+00 1.500000e+00
25% 1.398179e+09 4.438000e+02 4.448000e+02 4.435200e+02 4.438000e+02
50% 1.474242e+09 3.596700e+03 3.598100e+03 3.595620e+03 3.597900e+03
75% 1.544288e+09 8.627270e+03 8.632980e+03 8.621890e+03 8.627160e+03
max 1.617196e+09 6.176556e+04 6.178183e+04 6.167355e+04 6.176180e+04

Volume_(BTC) Volume_(Currency) Weighted_Price
count 1.613769e+06 1.613769e+06 3.613769e+06
mean 9.323349e+00 4.176284e+04 6.008935e+03
std 3.054080e+01 1.518244e+05 8.995992e+03
min 0.000000e+00 0.000000e+00 3.800000e+00
25% 4.097759e-01 4.521442e+02 4.438306e+02
50% 1.978812e+00 3.618124e+03 3.596884e+03
75% 7.278216e+00 2.569832e+04 6.627637e+03
max 5.853852e+03 1.390067e+07 6.176212e+04
  
```

Gambar 5. Summary statistics

Dalam persiapan data-data sebelum diproses pada Gambar 6.

```

Data Preprocessing
+ Code + Markdown
[7]: # Check for missing values
missing_values = data.isnull().sum()
print('Missing Values:')
print(missing_values)

Missing Values:
Timestamp      0
Open           1243688
High           1243688
Low            1243688
Close          1243688
Volume_(BTC)   1243688
Volume_(currency) 1243688
Weighted_Price 1243688
dtype: int64

[8]: df=data
    
```

Gambar 6. Data sebelum diproses

Yang kemudian dilakukan konversi data-data menjadi waktu berdasarkan hari, bulan, dan sebagainya. Lalu akan dicari rata-rata dari konversi data tersebut dan Gambar 7. Data yang dapat diketahui beberapa harga open, high, dan lainnya. Sehingga datanya akan muncul hasil harganya yang dimana terdapat harga open, high, low, dan lain-lain.

```

[9]: # Convert Unix time to datetime
df['Timestamp'] = pd.to_datetime(df['Timestamp'], unit='s')

# Resample to daily frequency
df_daily = df.resample('D', on='Timestamp').mean()

# Resample to monthly frequency
df_monthly = df.resample('M', on='Timestamp').mean()

# Resample to annual frequency
df_annual = df.resample('A-DEC', on='Timestamp').mean()

# Resample to quarterly frequency
df_quarterly = df.resample('Q-DEC', on='Timestamp').mean()

Feature Engineering

[11]: # Calculate the daily percentage change in Bitcoin price
df_daily['Price_Pct_Change'] = df_daily['Close'].pct_change()

# Calculate 3-month moving average for df_monthly
df_monthly['Moving_Average_3M'] = df_monthly['Close'].rolling(window=3).mean()

# Calculate 6-month moving average for df_monthly
df_monthly['Moving_Average_6M'] = df_monthly['Close'].rolling(window=6).mean()

# Calculate 12-month moving average for df_annual
df_annual['Moving_Average_12M'] = df_annual['Close'].rolling(window=12).mean()

# Calculate 3-month rolling standard deviation for df_quarterly
df_quarterly['Rolling_Std_3M'] = df_quarterly['Close'].rolling(window=3).std()

# Calculate 6-month rolling standard deviation for df_quarterly
df_quarterly['Rolling_Std_6M'] = df_quarterly['Close'].rolling(window=6).std()
    
```

Gambar 7. Data-data yang akan diproses

Pada Gambar 8 berikut ini menunjukkan hasil dari pengolahan data :

```

[12]: df.head()

[13]:
Timestamp  Open  High  Low  Close  Volume_(BTC)  Volume_(currency)  Weighted_Price
0  2011-12-31 07:52:00  4.39  4.39  4.39  0.455581  2.0  4.39
1  2011-12-31 07:53:00  NaN  NaN  NaN  NaN  NaN  NaN
2  2011-12-31 07:54:00  NaN  NaN  NaN  NaN  NaN  NaN
3  2011-12-31 07:55:00  NaN  NaN  NaN  NaN  NaN  NaN
4  2011-12-31 07:56:00  NaN  NaN  NaN  NaN  NaN  NaN
    
```

Gambar 8. Harga hasil pengolahan data

Analisis Data

Dalam melakukan analisis harga pada data yang digunakan, memungkinkan diketahui prediksi harganya pada Gambar 9.

Exploratory Data Analysis

```

[13]: import matplotlib.pyplot as plt

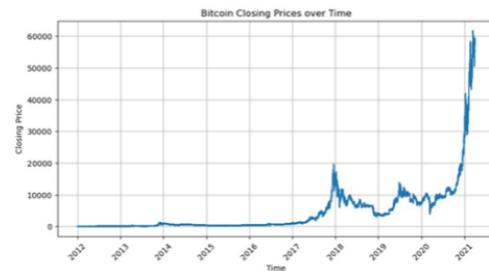
# Convert unix time to datetime
df['Timestamp'] = pd.to_datetime(df['Timestamp'], unit='s')

# Set the Timestamp column as the index
df.set_index('Timestamp', inplace=True)

# Plot 1: Line chart of Bitcoin closing prices over time
plt.figure(figsize=(10, 5))
plt.plot(df.index, df['Close'])
plt.title('Bitcoin Closing Prices over Time')
plt.xlabel('Time')
plt.ylabel('Closing Price')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
    
```

Gambar 9. Prediksi harga

Sehingga bisa diketahui harga dari tahun ke tahun menggunakan grafik terlihat pada Gambar 10.



Gambar 10. Grafik prediksi harga per tahun

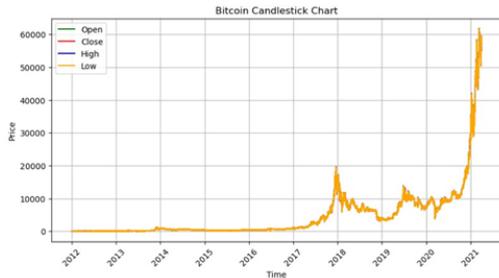
Kemudian akan dilakukan proses analisis prediksi harga sseperti pada Gambar 11 berikut :

```

[14]: # Plot 2: Candlestick chart of Bitcoin prices
plt.figure(figsize=(10, 5))
plt.plot(df.index, df['Open'], color='green', label='Open')
plt.plot(df.index, df['Close'], color='red', label='Close')
plt.plot(df.index, df['High'], color='blue', label='High')
plt.plot(df.index, df['Low'], color='orange', label='Low')
plt.title('Bitcoin Candlestick Chart')
plt.xlabel('Time')
plt.ylabel('Price')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()
    
```

Gambar 11. Analisis prediksi harga

Kemudian dapat ditampilkan dengan grafik seperti pada Gambar 12.



Gambar 12. Grafik Candle Bitcoin

Grafik pada Gambar 12 tersebut mempunyai keterangan seperti open, close, high, dan low dengan menggunakan warna yang mewakili setiap keterangan.

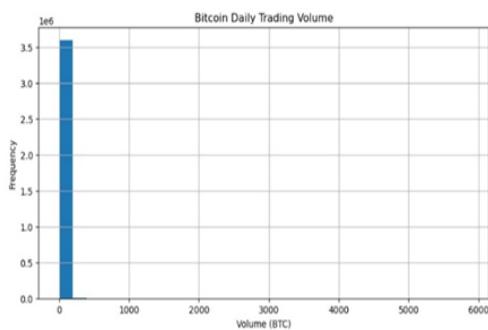
Histogram Daily

Langkah berikutnya adalah membuat histogram daily untuk menentukan trading daily dari volume pada bitcoin dimana pada Gambar 13 terlihat proses histogram tersebut.

```
[15]: # Plot 3: Histogram of Bitcoin daily trading volume
plt.figure(figsize=(10, 5))
plt.hist(df['Volume_(BTC)'], bins=30)
plt.title('Bitcoin Daily Trading Volume')
plt.xlabel('Volume (BTC)')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

Gambar 13. Proses Histogram daily

Dan menghasilkan grafik seperti Gambar 14 berikut :



Gambar 14. Grafik Histogram Daily

Grafik pada gambar 14 tersebut akan ditentukan oleh frekuensi dari volume yang ada.

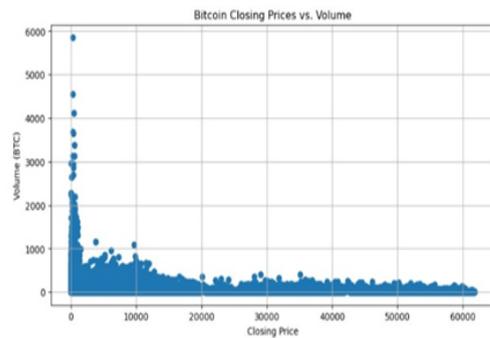
Closing price

Proses *closing price* dari volume yang ada terlihat pada gambar 15.

```
[16]: # Plot 4: Scatter plot of Bitcoin closing prices and volume
plt.figure(figsize=(10, 5))
plt.scatter(df['Close'], df['Volume_(BTC)'])
plt.title('Bitcoin Closing Prices vs. Volume')
plt.xlabel('Closing Price')
plt.ylabel('Volume (BTC)')
plt.grid(True)
plt.show()
```

Gambar 15. Proses closing price

Kemudian dapat dilihat grafiknya pada Gambar 16. Grafik *closing price* berikut :



Gambar 16. Grafik closing price

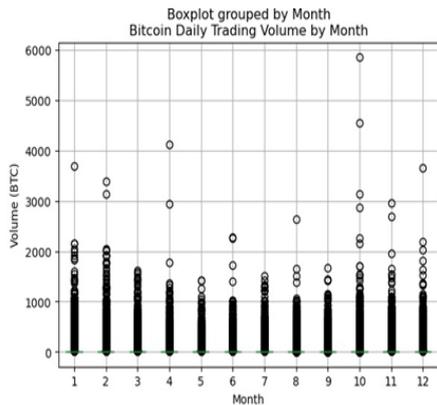
Membuat grafik *closing price* ini untuk mengetahui volume setiap bulan seperti Gambar 17 berikut :

```
[17]: # Plot 5: Box plot of Bitcoin daily trading volume by month
df['Month'] = df.index.month
plt.figure(figsize=(10, 5))
df.boxplot(column='Volume_(BTC)', by='Month')
plt.title('Bitcoin Daily Trading Volume by Month')
plt.xlabel('Month')
plt.ylabel('Volume (BTC)')
plt.grid(True)
plt.show()
```

(figure size 1000x500 with 0 Axes)

Gambar 17. Proses Box plot of Bitcoin daily trading volume by Month

Kemudian grafiknya dapat dilihat pada Gambar 18 berikut :



Gambar 18. Hasil proses Boxplot grouped by Month

Pengujian Model

Mean Square Error (RMSE) merupakan turunan dari Mean Squared Error (MSE). Menghitung dengan cara menjumlahkan semua kuadrat kesalahan prediksi. Kemudian membagi jumlah tersebut dengan banyaknya data waktu prediksi, kemudian menarik akarnya. [9].

Persamaan RMSE :

$$RMSE = \sqrt{(1/n) * \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Di mana:

n adalah jumlah sampel dalam data

y_i adalah nilai aktual

\hat{y}_i adalah nilai prediksi

Penentuan Model Terbaik

Setelah pengujian berhasil dilakukan pada semua model, kemudian menentukan model yang terbaik yaitu model yang memiliki performa yang bagus dalam memprediksi harga bitcoin dengan tingkat error rate yang kecil dan memiliki akurasi yang tinggi.

Berikut dilakukan proses import pada library yang digunakan seperti terdapat statsmodels, sklearn, dan tensorflow pada Gambar 19.

```
[18]: from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
```

Gambar 19. Import statsmodels, sklearn, dan tensorflow

Dilakukan pengaplikasian pada setiap model. Model yang digunakan adalah

model ARIMA dan SARIMA seperti pada Gambar 20 berikut :

```
[19]: # Split data into train and test sets
train_size = int(len(df_daily) * 0.8)
train_data, test_data = df_daily[:train_size], df_daily[train_size:]

[20]: # Scale the data
scaler = MinMaxScaler()
scaled_train_data = scaler.fit_transform(train_data[['Close']])
scaled_test_data = scaler.transform(test_data[['Close']])

[21]: # ARIMA model
arima_model = ARIMA(train_data[['Close']], order=(1, 1, 1))
arima_model_fit = arima_model.fit()
arima_predictions = arima_model_fit.predict(start=len(train_data), end=len(train_data) + len
```

Gambar 20. Permodelan ARIMA dan SARIMA

Maka akan dihasilkan output dari setiap model yang dibuat seperti terlihat pada Gambar 21 berikut :

```
[22]: # SARIMA model
sarima_model = SARIMAX(train_data[['Close']], order=(1, 1, 1), seasonal_order=(0, 1, 1, 12))
sarima_model_fit = sarima_model.fit()
sarima_predictions = sarima_model_fit.predict(start=len(train_data), end=len(train_data) + len

RUNNING THE L-BFGS-B CODE
***
Machine precision = 2.220446e-16
N = 4      n = 10
At NB 0 variables are exactly at the bounds
At Iterate 0  f = 6.855740e+00  |proj g|= 5.562590e-02
This problem is unconstrained.
At Iterate 5  f = 6.822550e+00  |proj g|= 3.413660e-03
At Iterate 10  f = 6.734760e+00  |proj g|= 9.180900e-04
At Iterate 15  f = 6.734880e+00  |proj g|= 6.217500e-04
***
Tit = total number of iterations
Tnf = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip = number of BFGS updates skipped
Nact = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F = final function value
***
N  Tit  Tnf  Tnint  Skip  Nact  Projg  F
4  19  30  1  0  0  3.4950e-06  6.7330e+00
F = 6.73310431508094684

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT <= PGTOL
```

Gambar 21. Hasil output proses dengan model ARIMA dan SARIMA

Model ARIMA memiliki RMSE 14490,55.
Model SARIMA memiliki RMSE 13692,22

Ujicoba lain yaitu model LSTM yang akan digabungkan hasilnya dengan model sebelumnya dengan RMSE yang terlihat pada Gambar 22 berikut :

```
[23]: # LSTM model
def create_lstm_model():
    model = Sequential()
    model.add(LSTM(128, input_shape=(1, 1)))
    model.add(Dense(1))
    model.compile(loss='mean_squared_error', optimizer='adam')
    return model

lstm_model = create_lstm_model()
lstm_history = lstm_model.fit(np.expand_dims(scaled_train_data, axis=-1),
                             np.expand_dims(scaled_train_data, axis=-1),
                             epochs=10, batch_size=1, verbose=0)
lstm_predictions = lstm_model.predict(np.expand_dims(scaled_test_data, axis=-1))

22/22 [-----] 0s 2ms/step

[24]: # Gradient Boosting Regressor model
gbr_model = GradientBoostingRegressor()
gbr_model.fit(np.arange(len(train_data)).reshape(-1, 1), train_data['Close'])
gbr_predictions = gbr_model.predict(np.arange(len(train_data), len(train_data) + len(test_data)))
```

Gambar 22. Proses model LSTM

Model LSTM memiliki RMSE 19650,40.

Hasilnya akan keluar dari setiap model yang digunakan termasuk model Gradient Boosting Regressor pada RMSE seperti pada Gambar 23 berikut :

```
[25]: # Calculate RMSE for each model
arima_rmse = np.sqrt(mean_squared_error(test_data['Close'], arima_predictions))
sarima_rmse = np.sqrt(mean_squared_error(test_data['Close'], sarima_predictions))
lstm_rmse = np.sqrt(mean_squared_error(test_data['Close'], lstm_predictions))
gbr_rmse = np.sqrt(mean_squared_error(test_data['Close'], gbr_predictions))

# Print the RMSE values
print(f"ARIMA RMSE: {arima_rmse}")
print(f"SARIMA RMSE: {sarima_rmse}")
print(f"LSTM RMSE: {lstm_rmse}")
print(f"Gradient Boosting Regressor RMSE: {gbr_rmse}")

ARIMA RMSE: 14490.546813975131
SARIMA RMSE: 13692.218178958887
LSTM RMSE: 19650.40562928348
Gradient Boosting Regressor RMSE: 14573.106791655091
```

Gambar 23. Proses Model Gradient Boosting Regressor

Model Gradient Boosting Regressor memiliki RMSE 14573,20

Evaluasi Model

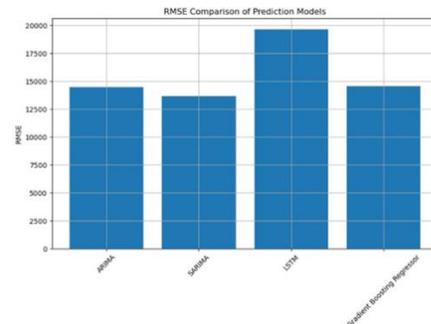
Untuk mengetahui kualitas keakuratan dari keempat model yang telah diuji selanjutnya dilakukan evaluasi dengan melakukan proses RSME value pada model ARIMA, SARIMA, LSTM dan Gradient Boosting Regressor seperti pada Gambar 24. Berikut :

```
[26]: # Define the models and RMSE values
models = ['ARIMA', 'SARIMA', 'LSTM', 'Gradient Boosting Regressor']
rmse_values = [arima_rmse, sarima_rmse, lstm_rmse, gbr_rmse]

# Plot the bar graph
plt.figure(figsize=(10, 6))
plt.bar(models, rmse_values)
plt.title('RMSE Comparison of Prediction Models')
plt.xlabel('Model')
plt.ylabel('RMSE')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

Gambar 24. Proses RSME value pada model ARIMA, SARIMA, LSTM dan Gradient Boosting Regressor

Sehingga dari proses RSME value tersebut didapatkan hasil histogram model ARIMA, SARIMA, LSTM dan Gradient Boosting Regressor seperti pada Gambar 25. Berikut :



Gambar 25. Histogram ARIMA, SARIMA, LSTM dan Gradient Boosting Regressor

HASIL DAN PEMBAHASAN

Sebagaimana langkah proses yang dilakukan pada penelitian ini dalam melakukan prediksi harga Bitcoin melalui dataset yang digunakan dan melakukan pengujian dan evaluasi pada empat model yang telah ditentukan dengan memanfaatkan deep learning yang mampu memaksimalkan kinerja data yang tidak terstruktur dalam sebuah aplikasi.

Berdasarkan pengujian dari keempat model tersebut, yaitu model ARIMA, SARIMA, LSTM dan Gradient Boosting Regressor didapatkan hasil sebagai berikut :

- Berdasarkan nilai RMSE yang diberikan model ARIMA memiliki RMSE 14490,55. Hal ini menunjukkan bahwa rata-rata prediksi model ARIMA menyimpang dari nilai sebenarnya sekitar yaitu 14490.55 unit.
- Model SARIMA memiliki RMSE 13692,22. Hal ini menunjukkan kinerjanya lebih baik dibandingkan model ARIMA dengan kesalahan prediksi yang lebih rendah.
- Model LSTM memiliki RMSE 19650,40. Hal ini menunjukkan bahwa memiliki kesalahan prediksi yang jauh lebih tinggi dari model ARIMA dan SARIMA.
- Model Gradient Boosting Regressor memiliki RMSE 14573,20. Hal ini menunjukkan bahwa performanya lebih

mirip dengan ARIMA dalam hal kesalahan prediksi.

Berdasarkan hasil pengujian keempat model tersebut dapat disimpulkan bahwa SARIMA menjadi model yang terbaik diantara model yang diuji, karena memiliki nilai RMSE terendah atau kesalahan lebih rendah dibandingkan ketiga model lainnya atau dapat dinyatakan memiliki posisi lebih akurat dalam melakukan prediksi harga Bitcoin.

PENUTUP

Penggunaan Deep Learning dan pemrograman Python dengan model SARIMA mampu memberikan hasil prediksi harga Bitcoin yang lebih akurat dengan memiliki RMSE 13692.22 dibandingkan dengan model ARIMA, LSTM ataupun Gradient Boosting Regressor yang memiliki nilai RMSE lebih tinggi, dan tentunya bila dibandingkan dengan metode prediksi harga Bitcoin yang tradisional. Hal ini memberikan perkembangan teknologi untuk memprediksi harga Bitcoin dengan model yang dipilih dan parameter masih dapat dioptimalkan lebih lanjut untuk meningkatkan hasil yang lebih akurat lagi. Nilai RMSE merupakan salah satu parameter untuk memberikan penilaian kinerja model tersebut, tentu diperlukan parameter lain yang lebih lengkap yang dapat lebih direkomendasikan ke depan untuk mempertimbangkan proses evaluasi yang lebih baik sehingga dapat membantu menghasilkan prediksi harga Bitcoin yang sangat dibutuhkan para pedagang dan investor. Untuk pengembangan penelitian ini sarannya antara lain supaya menggunakan lebih banyak dataset dengan lebih bervariasi frekuensinya, model yang lebih kompleks, pengujian yang lebih luas, mengkombinasikan model-model yang ada dengan model atau metode lainnya seperti sentiment analysis agar dapat menghasilkan tingkat akurasi yang lebih optimal lagi.

DAFTAR PUSTAKA

- [1] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena (2016), "Jain_Structural-RNN_Deep_Learning_CVPR_2016_paper," *Cvpr*, pp. 5308–5317, doi: 10.1109/CVPR.2016.57
- [2] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun (2013), "Evaluating user privacy in Bitcoin", *Lecture Notes in Computer Science*, V0 7859, Springer. Eviews. Semarang: FMIPA Unnes.
- [3] G. Lilipaly, D. Hatidja & J. Kekunesa (2014), "Prediksi Harga Saham PT. BRI, Tbk. Menggunakan Metode ARIMA", *Jurnal Ilmiah Sains*, vol 14 No. 2.
- [4] Hendikawati, P (2015), *Peramalan Data Runtun Waktu Metode dan Aplikasinya dengan Minitab dan*
- [5] HM. Sularno dan Eka Suci Budiasih (2022), "Analisis Keabsahan bitcoin sebagai mata uang virtual menurut prespektif hukum positif di Indonesia". *Jurnal Manajemen dan Bisnis*.17, 1. Sekolah Tinggi Manajemen IMMI, Jakarta Selatan.
- [6] I. Miers, C. Garman, M. Green, and A. D. Rubin. *ZeroCoin* (2013), "Anonymous distributed e-cash from bitcoin", *Proceedings - IEEE Symposium on Security and Privacy*, 397-411.
- [7] J. C. Paul, S. Hoque, M.M Rahman (2013), "Selection of best ARIMA Model for Forecasting Average Daily Share Price Index of Pharmaceutical Companies in Bangladesh: A Case Study on Square Pharmaceutical Ltd", *Global Journal of Management and Business Research*, 13(3(1)):14-25.
- [8] Jalili Ghazi Zade, M. and Noori, R (2008), Prediction of Municipal Solid Waste Generation by Use of Artificial Neural Network: A Case Study of Mashhad. *Int. J. Environ. Res.*, 2(1): 13-22, Winter. ISSN: 1735-6865.
- [9] N. Gandal, & H. Halaburda (2014), "Competition in the Cryptocurrency Market", *SSRN Electronic Journal*.
- [10] Nur Fitriani Bintang Pradana, Sri Lestanti (2020), *Aplikasi Prediksi Jangka Pendek harga Bitcoin Menggunakan Metode ARIMA* *Jurnal Ilmiah Informatika Komputer*

- p-ISSN: 0853-8638, e-ISSN: 2089-8045 Vol 25, No. 3.
- [11] Primartha, Rifkie (2018), “Belajar Machine Learning Teori dan Praktik. Bandung: Informatika Bandung.
 - [12] R. Albariqi (2018), Prediksi Harga Bitcoin Menggunakan Jaringan Syaraf Tiruan, Universitas Gajah Mada.
 - [13] R.A. Juanda, Jondri, & A.A. Rohmawati (2018), “Prediksi Harga Bitcoin Dengan Menggunakan Recurrent Neural Network”, e-Proceeding of Engineering, Vol 5, No. 2.
 - [14] S. B. Koduri, L. Guniseti, C. R. Ramesh, K. Mutyalu, and D. Ganesh (2019), “Prediction of crop production using adaboost regression method Prediction of crop production using adaboost regression method,” J. Phys. Conf. Ser.
 - [15] S. Nakamoto (2009), “Bitcoin: A Peer-to-Peer Electronic Cash System”, Cryptography Mailing list.
 - [16] Yuhana, Aulia (2018), SARIMA (Seasonal Autoregressive Integrated Moving Average) using Eviews 9.