2502014210 - Eleanor Maritsa Maharani

No. 1a

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.activations import relu
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import GridSearchCV
from keras.layers import Dense, Dropout
from sklearn.metrics import make_scorer
```

Pada kasus ini saya hanya akan menggunakan kolom date dan close saja. Kedua dataset timeseries yang saya dapatkan yaitu dataset AMZN dan dataset CSCO

```
# Membaca dataset AMZN.csv
df_amzn = pd.read_csv('AMZN.csv', usecols=['Date', 'Close'])
df_amzn['Date'] = pd.to_datetime(df_amzn['Date'])  # Mengubah kolom
Date menjadi tipe datetime
df_amzn.set_index('Date', inplace=True)  # Menggunakan kolom Date
sebagai index
```

Pada dataset AMZN kolom close memiliki 5758 records dengan tipe data float

```
# Eksplorasi data AMZN
print("Info dataset AMZN:")
print(df_amzn.info())

Info dataset AMZN:
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 5758 entries, 1997-05-15 to 2020-04-01
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Close   5758 non-null   float64
dtypes: float64(1)
memory usage: 90.0 KB
None
```

```python
print("Statistik deskriptif dataset AMZN:")
print(df_amzn.describe())
```

```
Statistik deskriptif dataset AMZN:
             Close
count  5758.000000
mean    340.417580
std     523.140207
min       1.395833
25%      37.562500
50%      81.599998
75%     334.290001
max    2170.219971
```

```python
print("Lima data teratas dataset AMZN:")
print(df_amzn.head())
```

```
Lima data teratas dataset AMZN:
               Close
Date
1997-05-15  1.958333
1997-05-16  1.729167
1997-05-19  1.708333
1997-05-20  1.635417
1997-05-21  1.427083
```

Pada dataset AMZN tidak ditemukan missing value

```python
# Mengecek jumlah nilai yang hilang pada setiap kolom
print(df_amzn.isnull().sum())
```

```
Close    0
dtype: int64
```

```python
# Visualisasi data time series AMZN
plt.figure(figsize=(12, 6))
plt.plot(df_amzn.index, df_amzn['Close'], label='AMZN', color='b')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.title('Amazon (AMZN) Close Price Time Series')
plt.legend()
plt.show()
```

Disini saya akan memisahkan data time series menjadi input dan output. Input atau window size adalah jumlah langkah waktu dari data historis yang digunakan untuk memprediksi. Window size saya inisialisasi 5, karena saya mau mengambil dari hari senin sampai dengan hari jumat. Output atau horizon adalah jumlah langkah waktu untuk memprediksi ke masa depan. Horizon saya inisialisas 1 karena hanya untuk hari senin saja.

```
# Split function
def split_data_time_series(df, window_size, horizon):
    x, y = [], []
    for i in range(len(df) - window_size - horizon + 1):
        x.append(df.iloc[i:i+window_size]['Close'].values)
        y.append(df.iloc[i+window_size+horizon-1]['Close'])
    return np.array(x), np.array(y)

window_size = 5
horizon = 1

# Menggunakan fungsi split_data_time_series untuk memisahkan data AMZN
x_amzn, y_amzn = split_data_time_series(df_amzn, window_size, horizon)
```

Problem data : scalling data

Scaling data adalah proses transformasi data sehingga nilainya memiliki rentang yang sama atau setidaknya skalanya serupa. Teknik penskalaan data yang akan saya gunakan adalah normalisasi min-max scaling. Alasan menggunakan teknik penskalaan min-max karena teknik ini mengubah nilai numerik menjadi rentang 0 hingga 1, sehingga semua fitur memiliki rentang nilai dan tidak mempengaruhi kinerja model. Scalling data perlu dilakukan karena jika ada variabel numerik dengan skala yang berbeda-beda pada dataset perlu dilakukan scaling data agar skala variabel tersebut seimbang dan dapat digunakan dalam model.

```python
# Skalasi data menggunakan MinMaxScaler
scaler = MinMaxScaler()
x_amzn = scaler.fit_transform(x_amzn)
y_amzn = scaler.fit_transform(y_amzn.reshape(-1, 1))

# Menampilkan contoh input dan output
print("Contoh input AMZN:")
print(x_amzn[1])  # Contoh input pertama dari AMZN
print("\nOutput yang sesuai untuk contoh input AMZN:")
print(y_amzn[1])  # Output yang sesuai untuk contoh input pertama dari
AMZN\
```

```
Contoh input AMZN:
[1.53693076e-04 1.44087293e-04 1.10466888e-04 1.44087293e-05
 0.00000000e+00]

Output yang sesuai untuk contoh input AMZN:
[4.80290795e-05]
```

```python
from sklearn.model_selection import train_test_split
```

Memisahkan dataset menjadi train, test dan validation set dengan ketentuan 80 train, 10 val, 10 test

```python
# Pisahkan data AMZN menjadi train, validation, dan test set
X_train_amzn, X_temp_amzn, y_train_amzn, y_temp_amzn =
train_test_split(x_amzn, y_amzn, test_size=0.2, random_state=42)
X_val_amzn, X_test_amzn, y_val_amzn, y_test_amzn =
train_test_split(X_temp_amzn, y_temp_amzn, test_size=0.5,
random_state=42)

# Menampilkan jumlah data pada setiap set
print("Jumlah data dalam setiap set:")
print("Train set (AMZN):", len(X_train_amzn))
print("Validation set (AMZN):", len(X_val_amzn))
print("Test set (AMZN):", len(X_test_amzn))
```

```
Jumlah data dalam setiap set:
Train set (AMZN): 4602
Validation set (AMZN): 575
Test set (AMZN): 576
```

```python
# Ubah bentuk data agar sesuai dengan input yang diperlukan oleh LSTM
(jumlah data, jumlah time steps, jumlah fitur)
X_train_amzn = X_train_amzn.reshape(X_train_amzn.shape[0],
window_size, 1)
X_test_amzn = X_test_amzn.reshape(X_test_amzn.shape[0], window_size,
1)
X_val_amzn = X_val_amzn.reshape(X_val_amzn.shape[0], window_size, 1)
```

No. 1b - AMZN

Membuat arsitektur baseline dengan LSTM (units=50) dan layer akhir berupa node Perceptron dengan units=1. Activation function untuk LSTM menggunakan ReLU

```
# Membangun model LSTM
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(window_size, 1)))
model.add(Dense(1))

model.compile(optimizer='adam', loss='mean_squared_error',
metrics=['mae'])

print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 50)                10400

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 10,451
Trainable params: 10,451
Non-trainable params: 0
_________________________________________________________________
None
```

```
# Latih model dengan data train
modelamzn=model.fit(X_train_amzn, y_train_amzn, epochs=5,
batch_size=64, validation_data=(X_val_amzn, y_val_amzn))
```

```
Epoch 1/5
72/72 [==============================] - 2s 7ms/step - loss: 0.0332 -
mae: 0.1081 - val_loss: 0.0026 - val_mae: 0.0465
Epoch 2/5
72/72 [==============================] - 0s 4ms/step - loss: 3.8965e-
04 - mae: 0.0135 - val_loss: 1.2959e-04 - val_mae: 0.0070
Epoch 3/5
72/72 [==============================] - 0s 4ms/step - loss: 1.0104e-
04 - mae: 0.0061 - val_loss: 1.0432e-04 - val_mae: 0.0049
Epoch 4/5
72/72 [==============================] - 0s 4ms/step - loss: 8.7378e-
05 - mae: 0.0050 - val_loss: 9.7132e-05 - val_mae: 0.0044
Epoch 5/5
72/72 [==============================] - 0s 4ms/step - loss: 8.3062e-
05 - mae: 0.0045 - val_loss: 9.4783e-05 - val_mae: 0.0044
```

```python
# Evaluate the model on the test data using
print("Evaluate on test data")
results = model.evaluate(X_test_amzn, y_test_amzn, batch_size=64)
print("Test MSE:", results[0])
```

```
Evaluate on test data
9/9 [==============================] - 0s 3ms/step - loss: 6.2500e-05
- mae: 0.0045
Test MSE: 6.249965372262523e-05
```

```python
# Melakukan prediksi pada data test
y_pred_amzn = model.predict(X_test_amzn)

# Menampilkan hasil prediksi
print("Predicted Values:")
print(y_pred_amzn)
```

```
18/18 [==============================] - 0s 2ms/step
Predicted Values:
[[0.16373861]
 [0.03237148]
 [0.01875058]
 [0.02577537]
 [0.03958131]
 [0.02094679]
 [0.03110025]
 [0.01710599]
 [0.40085173]
 [0.4569433 ]
 [0.14848065]
 [0.10972425]
 [0.1390023 ]
 [0.0283286 ]
 [0.23274946]
 [0.0189946 ]
 [0.15536751]
 [0.02408208]
 [0.8236798 ]
 [0.05290263]
 [0.34955913]
 [0.8014701 ]
 [0.10527873]
 [0.01940142]
 [0.89146537]
 [0.0251525 ]
 [0.35063112]
 [0.11297817]
 [0.00478641]
 [0.00348775]
 [0.07738731]
```

```
 [0.02041113]
 [0.04243191]
 [0.02017161]
 [0.00412962]
 [0.00865368]
 [0.16027144]
 [0.03377822]
 [0.82172847]
 [0.03168599]
 [0.01710985]
 [0.19889207]
 [0.19720523]
 [0.00383657]
 [0.00482799]
 [0.31651968]
 [0.00369013]
 [0.75762427]
 [0.11905486]
 [0.03639422]
 [0.03420278]
 [0.2196061 ]
 [0.08201177]
 [0.27850217]
 [0.13857451]
 [0.00277586]
 [0.04227838]
 [0.17772834]
 [0.01474932]
 [0.05896948]
 [0.02236107]
 [0.6459702 ]
 [0.04010871]
 [0.03507708]
 [0.01678898]
 [0.37790522]
 [0.08274665]
 [0.87973285]
 [0.5384527 ]
 [0.18623495]
 [0.019773  ]
 [0.06301866]
 [0.09501254]
 [0.02159155]
 [0.02384833]
 [0.01942723]
 [0.14902557]
 [0.23593584]
 [0.03445841]
 [0.02711789]
```

```
[0.00392839]
[0.02929246]
[0.41995683]
[0.0161845 ]
[0.01845626]
[0.0051023 ]
[0.01567866]
[0.02541699]
[0.5623209 ]
[0.4000822 ]
[0.09628523]
[0.10496039]
[0.12350361]
[0.01870335]
[0.00305852]
[0.08314491]
[0.00783349]
[0.14229655]
[0.08200346]
[0.03397383]
[0.19010459]
[0.9317458 ]
[0.09656854]
[0.00749212]
[0.13361423]
[0.4621458 ]
[0.8867352 ]
[0.00992336]
[0.27638176]
[0.03538875]
[0.02318772]
[0.00510916]
[0.00962198]
[0.01020862]
[0.05866841]
[0.03732515]
[0.01711228]
[0.5900785 ]
[0.03447898]
[0.01549486]
[0.02246568]
[0.01802418]
[0.0429681 ]
[0.01633571]
[0.00526426]
[0.42253792]
[0.0168277 ]
[0.0221216 ]
[0.82865834]
```

```
 [0.10091329]
 [0.08400083]
 [0.00809232]
 [0.02063091]
 [0.07739322]
 [0.93306184]
 [0.02342046]
 [0.02300545]
 [0.03336574]
 [0.01254269]
 [0.08915617]
 [0.00685509]
 [0.13691945]
 [0.08167905]
 [0.02746383]
 [0.01861721]
 [0.02163601]
 [0.70347756]
 [0.00869851]
 [0.14527386]
 [0.03044118]
 [0.11514153]
 [0.93238944]
 [0.0161768 ]
 [0.01803261]
 [0.017637  ]
 [0.75236416]
 [0.01116796]
 [0.04153172]
 [0.4374344 ]
 [0.10057253]
 [0.03319201]
 [0.24394119]
 [0.00880227]
 [0.02894878]
 [0.00873627]
 [0.3562201 ]
 [0.87127477]
 [0.102914  ]
 [0.00495099]
 [0.10475805]
 [0.01899449]
 [0.10118069]
 [0.3948631 ]
 [0.01990227]
 [0.01866442]
 [0.03150182]
 [0.3351326 ]
 [0.00529762]
```

```
[0.11974362]
[0.01109388]
[0.35820448]
[0.8967722 ]
[0.40293923]
[0.00382497]
[0.01607815]
[0.7784206 ]
[0.8364717 ]
[0.00842602]
[0.14410348]
[0.02404088]
[0.01945895]
[0.03781281]
[0.9080614 ]
[0.8244964 ]
[0.39912507]
[0.02094311]
[0.00388053]
[0.03227286]
[0.0178857 ]
[0.03877383]
[0.8144512 ]
[0.7626991 ]
[0.8070912 ]
[0.08825635]
[0.1818167 ]
[0.68293244]
[0.07998852]
[0.02010887]
[0.00392308]
[0.08337921]
[0.15729617]
[0.02566002]
[0.1602053 ]
[0.09613758]
[0.03371062]
[0.04435454]
[0.04013088]
[1.0071641 ]
[0.26225492]
[0.0365838 ]
[0.01696044]
[0.02035609]
[0.060224  ]
[0.01854686]
[0.02356091]
[0.01977631]
[0.0835867 ]
```

```
[0.3599383 ]
[0.02055651]
[0.00630414]
[0.13582525]
[0.03574483]
[0.05910812]
[0.03867008]
[0.01186909]
[0.00772275]
[0.08157688]
[0.64194477]
[0.0124166 ]
[0.02063396]
[0.08984783]
[0.01888793]
[0.02576727]
[0.6751632 ]
[0.06485689]
[0.76579005]
[0.02001525]
[0.08020051]
[0.04283994]
[0.02721488]
[0.03563645]
[0.08267239]
[0.03067447]
[0.22490014]
[0.0167068 ]
[0.02558721]
[0.02872093]
[0.02893595]
[0.46062407]
[0.05825863]
[0.08700383]
[0.00623632]
[0.01842995]
[0.01755215]
[0.0531819 ]
[0.22342561]
[0.01113562]
[0.00875664]
[0.01106949]
[0.03686691]
[0.0099051 ]
[0.01646855]
[0.01102692]
[0.05646102]
[0.11915167]
[0.00679129]
```

```
 [0.04322927]
 [0.02679651]
 [0.0194676 ]
 [0.00623731]
 [0.31539026]
 [0.15887228]
 [0.02832634]
 [0.01845654]
 [0.90954334]
 [0.13077646]
 [0.00535156]
 [0.03075596]
 [0.02001906]
 [0.01918878]
 [0.10135582]
 [0.008633  ]
 [0.1960528 ]
 [0.02312475]
 [0.03612414]
 [0.00481309]
 [0.00776881]
 [0.00882378]
 [0.862577  ]
 [0.01113117]
 [0.0363114 ]
 [0.13601281]
 [0.02622359]
 [0.01660722]
 [0.15331824]
 [0.01987401]
 [0.19449653]
 [0.10105473]
 [0.01778729]
 [0.03836653]
 [0.24365458]
 [0.02309514]
 [0.01091751]
 [0.08500074]
 [0.04069138]
 [0.12052096]
 [0.01025118]
 [0.008472  ]
 [0.7534397 ]
 [0.01945206]
 [0.02096426]
 [0.112209  ]
 [0.2416926 ]
 [0.04044448]
 [0.09926677]
```

```
[0.08376848]
[0.8296354 ]
[0.00991851]
[0.12325343]
[0.00506127]
[0.08411477]
[0.03291276]
[0.27406976]
[0.00816887]
[0.07665189]
[0.07913165]
[0.00613785]
[0.08389762]
[0.13488847]
[0.00614182]
[0.13002355]
[0.00369821]
[0.02021807]
[0.8444131 ]
[0.03973595]
[0.03709438]
[0.01307963]
[0.01609056]
[0.0377534 ]
[0.23585662]
[0.09193055]
[0.03907885]
[0.03020708]
[0.03645257]
[0.0079643 ]
[0.01263347]
[0.11960658]
[0.36215886]
[0.7416828 ]
[0.9190535 ]
[0.05440288]
[0.02372421]
[0.04161705]
[0.03220487]
[0.00887022]
[0.16968043]
[0.01386521]
[0.03929795]
[0.14622691]
[0.0183759 ]
[0.14906354]
[0.06488575]
[0.01893811]
[0.02313308]
```

```
 [0.14051144]
 [0.00395012]
 [0.01108578]
 [0.00949311]
 [0.01770983]
 [0.03660135]
 [0.8357697 ]
 [0.01553625]
 [0.08320685]
 [0.8339136 ]
 [0.0064503 ]
 [0.03510325]
 [0.14757647]
 [0.13711475]
 [0.13525325]
 [0.01611308]
 [0.03581711]
 [0.01816144]
 [0.03333885]
 [0.18166152]
 [0.01662648]
 [0.01524703]
 [0.00652316]
 [0.00732041]
 [0.14052998]
 [0.00812531]
 [0.830839  ]
 [0.4614857 ]
 [0.17979796]
 [0.03935987]
 [0.09469292]
 [0.08206721]
 [0.041781  ]
 [0.00839003]
 [0.02740173]
 [0.0048978 ]
 [0.00676811]
 [0.45816022]
 [0.71963155]
 [0.0206838 ]
 [0.26202726]
 [0.05288532]
 [0.386093  ]
 [0.00309094]
 [0.00562753]
 [0.09653225]
 [0.01636249]
 [0.02107432]
 [0.0182632 ]
```

```
[0.01162842]
[0.05798945]
[0.01044452]
[0.02970853]
[0.82712305]
[0.01530178]
[0.07470316]
[0.0189141 ]
[0.03287639]
[0.00459153]
[0.84114397]
[0.01887733]
[0.00483158]
[0.44733536]
[0.01111624]
[0.00787134]
[0.02234934]
[0.0196323 ]
[0.07296859]
[0.02315419]
[0.03262429]
[0.09696373]
[0.13784476]
[0.06159937]
[0.803158  ]
[0.09723182]
[0.01674588]
[0.897047  ]
[0.04141469]
[0.14931752]
[0.0038023 ]
[0.05977629]
[0.05968872]
[0.09756564]
[0.16755271]
[0.00507394]
[0.14347984]
[0.0087617 ]
[0.31146002]
[0.16199689]
[0.03595842]
[0.01216306]
[0.9955911 ]
[0.01747294]
[0.00870441]
[0.00396325]
[0.13367066]
[0.03556015]
[0.7168819 ]
```

```
 [0.02827901]
 [0.17741156]
 [0.01919372]
 [0.7165225 ]
 [0.00410101]
 [0.02456263]
 [0.03432585]
 [0.4587351 ]
 [0.05832656]
 [0.01095238]
 [0.6860375 ]
 [0.02980424]
 [0.00802065]
 [0.01538529]
 [0.5364409 ]
 [0.14149861]
 [0.24058789]
 [0.01109935]
 [0.01755052]
 [0.03625634]
 [0.0053202 ]
 [0.03326789]
 [0.04354767]
 [0.36895013]
 [0.00508163]
 [0.05654766]
 [0.00577005]
 [0.8074446 ]
 [0.09452778]
 [0.03892666]
 [0.03452144]
 [0.01324016]
 [0.02409693]
 [0.01581893]
 [0.05428535]
 [0.17040448]
 [0.03633562]
 [0.01892377]
 [0.00300511]
 [0.00402918]
 [0.00607626]
 [0.01841486]
 [0.0301069 ]
 [0.02633132]
 [0.02932489]
 [0.08457028]
 [0.35729823]
 [0.07829659]
 [0.174115  ]
 [0.05945431]
```

```
[0.36243537]
[0.06028509]
[0.10956332]
[0.02317446]
[0.00730057]
[0.77706873]
[0.03734623]
[0.00913471]
[0.45038566]
[0.46184486]
[0.02403922]
[0.02812559]
[0.1463976 ]
[0.04112627]
[0.03709799]
[0.03095706]
[0.01687028]
[0.08074464]
[0.02348766]
[0.3695123 ]
[0.16841234]
[0.00405385]
[0.00753647]
[0.1214885 ]
[0.00726883]
[0.02825579]
[0.13691181]
[0.03452606]
[0.02442591]
[0.31331345]
[0.03115149]
[0.01835718]
[0.00343592]
[0.08358245]
[0.74403685]
[0.9204302 ]
[0.05466411]
[0.10084676]
[0.02718118]
[0.31185415]
[0.14184847]
[0.45064193]
[0.01838584]
[0.9285454 ]
[0.8524493 ]
[0.01648919]
[0.02089727]
[0.01330339]
[0.01814021]
```

```
 [0.7819007 ]
 [0.26376122]
 [0.01095093]
 [0.11858483]
 [0.09875701]]
```

```python
# Mengubah hasil prediksi kembali ke skala semula
y_pred_inverse_amzn = scaler.inverse_transform(y_pred_amzn)
y_test_inverse_amzn = scaler.inverse_transform(y_test_amzn)

# Menampilkan hasil prediksi dalam skala semula
print("Predicted Values (Inverse Transform):")
print(y_pred_inverse_amzn)
```

```
Predicted Values (Inverse Transform):
[[ 356.51608  ]
 [  71.60389  ]
 [  42.06254  ]
 [  57.298088 ]
 [  87.24074  ]
 [  46.825745 ]
 [  68.8468   ]
 [  38.495716 ]
 [ 870.7727   ]
 [ 992.4255   ]
 [ 323.4243   ]
 [ 239.36842  ]
 [ 302.86737  ]
 [  62.835598 ]
 [ 506.1885   ]
 [  42.591774 ]
 [ 338.36066  ]
 [  53.62563  ]
 [1787.8125   ]
 [ 116.13233  ]
 [ 759.5281   ]
 [1739.6436   ]
 [ 229.72688  ]
 [  43.4741   ]
 [1934.8275   ]
 [  55.947193 ]
 [ 761.853    ]
 [ 246.42561  ]
 [  11.77671  ]
 [   8.960139 ]
 [ 169.23529  ]
 [  45.663986 ]
 [  93.42318  ]
 [  45.14452  ]
 [  10.352243 ]
```

```
[  20.164146 ]
[ 348.9964   ]
[  74.654854 ]
[1783.5804   ]
[  70.11718  ]
[  38.50409  ]
[ 432.75778  ]
[ 429.0993   ]
[   9.71668  ]
[  11.866887 ]
[ 687.87134  ]
[   9.399085 ]
[1644.5497   ]
[ 259.6049   ]
[  80.3285   ]
[  75.57565  ]
[ 477.68286  ]
[ 179.26492  ]
[ 605.418    ]
[ 301.9396   ]
[   7.416188 ]
[  93.0902   ]
[ 386.85736  ]
[  33.38451  ]
[ 129.29027  ]
[  49.893066 ]
[1402.3917   ]
[  88.38457  ]
[  77.47185  ]
[  37.808174 ]
[ 821.0058   ]
[ 180.85875  ]
[1909.3817   ]
[1169.2051   ]
[ 405.3067   ]
[  44.279984 ]
[ 138.07222  ]
[ 207.46132  ]
[  48.224102 ]
[  53.11868  ]
[  43.53007  ]
[ 324.6061   ]
[ 513.0992   ]
[  76.13005  ]
[  60.20976  ]
[   9.915817 ]
[  64.92603  ]
[ 912.2083   ]
[  36.497173 ]
```

```
[  41.424225 ]
[  12.461819 ]
[  35.400093 ]
[  56.52082  ]
[1220.971    ]
[ 869.10376  ]
[ 210.22156  ]
[ 229.03645  ]
[ 269.25345  ]
[  41.960106 ]
[   8.029228 ]
[ 181.72252  ]
[  18.385303 ]
[ 310.01205  ]
[ 179.24692  ]
[  75.07909  ]
[ 413.69928  ]
[2022.1887   ]
[ 210.83601  ]
[  17.644928 ]
[ 291.1816   ]
[1003.7088   ]
[1924.5686   ]
[  22.91785  ]
[ 600.8193   ]
[  78.14781  ]
[  51.68592  ]
[  12.476702 ]
[  22.264214 ]
[  23.536537 ]
[ 128.6373   ]
[  82.34753  ]
[  38.50937  ]
[1281.1724   ]
[  76.174675 ]
[  35.00146  ]
[  50.11994  ]
[  40.487103 ]
[  94.58609  ]
[  36.825127 ]
[  12.813079 ]
[ 917.8063   ]
[  37.892147 ]
[  49.37369  ]
[1798.6101   ]
[ 220.25902  ]
[ 183.57884  ]
[  18.946657 ]
[  46.140648 ]
```

```
[ 169.24811  ]
[2025.043    ]
[  52.190704 ]
[  51.29062  ]
[  73.760254 ]
[  28.598728 ]
[ 194.75989  ]
[  16.26331  ]
[ 298.35007  ]
[ 178.54332  ]
[  60.960045 ]
[  41.773296 ]
[  48.320526 ]
[1527.115    ]
[  20.261381 ]
[ 316.4693   ]
[  67.4174   ]
[ 251.11755  ]
[2023.5846   ]
[  36.48046  ]
[  40.505386 ]
[  39.647392 ]
[1633.1415   ]
[  25.617178 ]
[  91.470825 ]
[ 950.11414  ]
[ 219.51996  ]
[  73.38345  ]
[ 530.46136  ]
[  20.486406 ]
[  64.18065  ]
[  20.343256 ]
[ 773.97455  ]
[1891.0376   ]
[ 224.59819  ]
[  12.1336565]
[ 228.59761  ]
[  42.591537 ]
[ 220.83894  ]
[ 857.7844   ]
[  44.560352 ]
[  41.875687 ]
[  69.71774  ]
[ 728.2395   ]
[  12.885435 ]
[ 261.09866  ]
[  25.456512 ]
[ 778.2783   ]
[1946.337    ]
```

```
 [ 875.3002   ]
 [   9.69151  ]
 [  36.26651  ]
 [1689.6533   ]
 [1815.5558   ]
 [  19.670391 ]
 [ 313.93097  ]
 [  53.536274 ]
 [  43.598866 ]
 [  83.40516  ]
 [1970.8213   ]
 [1789.5835   ]
 [ 867.0279   ]
 [  46.81776  ]
 [   9.81203  ]
 [  71.38999  ]
 [  40.186768 ]
 [  85.489456 ]
 [1767.7974   ]
 [1655.5562   ]
 [1751.8347   ]
 [ 192.80833  ]
 [ 395.72427  ]
 [1482.5563   ]
 [ 174.87686  ]
 [  45.00843  ]
 [   9.9043   ]
 [ 182.23067  ]
 [ 342.54358  ]
 [  57.047913 ]
 [ 348.85297  ]
 [ 209.90132  ]
 [  74.50823  ]
 [  97.593025 ]
 [  88.43266  ]
 [2185.7578   ]
 [ 570.1806   ]
 [  80.73967  ]
 [  38.180042 ]
 [  45.544624 ]
 [ 132.0111   ]
 [  41.62072  ]
 [  52.4953   ]
 [  44.287167 ]
 [ 182.68068  ]
 [ 782.0387   ]
 [  45.97928  ]
 [  15.068405 ]
 [ 295.97693  ]
```

```
[  78.92008  ]
[ 129.59094  ]
[  85.26443  ]
[  27.137796 ]
[  18.145124 ]
[ 178.32173  ]
[1393.6611   ]
[  28.325266 ]
[  46.147263 ]
[ 196.25998  ]
[  42.360435 ]
[  57.28051  ]
[1465.7062   ]
[ 142.05902  ]
[1662.2598   ]
[  44.8054   ]
[ 175.33662  ]
[  94.30812  ]
[  60.420113 ]
[  78.68502  ]
[ 180.6977   ]
[  67.92338  ]
[ 489.1647   ]
[  37.629948 ]
[  56.89     ]
[  63.686474 ]
[  64.15282  ]
[1000.4084   ]
[ 127.74855  ]
[ 190.09184  ]
[  14.921309 ]
[  41.36715  ]
[  39.46336  ]
[ 116.73802  ]
[ 485.9667   ]
[  25.54703  ]
[  20.387438 ]
[  25.403614 ]
[  81.353676 ]
[  22.878248 ]
[  37.11323  ]
[  25.311287 ]
[ 123.84985  ]
[ 259.81485  ]
[  16.124947 ]
[  95.15251  ]
[  59.512756 ]
[  43.617634 ]
[  14.923469 ]
```

```
 [ 685.4218   ]
 [ 345.96188  ]
 [  62.83069  ]
 [  41.42482  ]
 [1974.0354   ]
 [ 285.027    ]
 [  13.002433 ]
 [  68.1001   ]
 [  44.81366  ]
 [  43.01293  ]
 [ 221.21878  ]
 [  20.119297 ]
 [ 426.5999   ]
 [  51.54936  ]
 [  79.74274  ]
 [  11.834578 ]
 [  18.24502  ]
 [  20.53307  ]
 [1872.1737   ]
 [  25.537384 ]
 [  80.14888  ]
 [ 296.3837   ]
 [  58.27018  ]
 [  37.413986 ]
 [ 333.91614  ]
 [  44.499058 ]
 [ 423.2246   ]
 [ 220.56577  ]
 [  39.973343 ]
 [  84.60608  ]
 [ 529.8398   ]
 [  51.485123 ]
 [  25.074001 ]
 [ 185.74748  ]
 [  89.64827  ]
 [ 262.7846   ]
 [  23.628843 ]
 [  19.770115 ]
 [1635.4741   ]
 [  43.58393  ]
 [  46.86362  ]
 [ 244.75742  ]
 [ 525.5846   ]
 [  89.11279  ]
 [ 216.68799  ]
 [ 183.07494  ]
 [1800.7291   ]
 [  22.907328 ]
 [ 268.71085  ]
```

```
[  12.372829 ]
[ 183.82597  ]
[  72.777824 ]
[ 595.80493  ]
[  19.112684 ]
[ 167.64029  ]
[ 173.01846  ]
[  14.707756 ]
[ 183.35501  ]
[ 293.94522  ]
[  14.716372 ]
[ 283.39407  ]
[   9.416603 ]
[  45.24527  ]
[1832.7794   ]
[  87.57612  ]
[  81.84702  ]
[  29.763254 ]
[  36.29342  ]
[  83.27632  ]
[ 512.92737  ]
[ 200.77704  ]
[  86.15099  ]
[  66.90969  ]
[  80.45503  ]
[  18.669006 ]
[  28.795616 ]
[ 260.80148  ]
[ 786.8547   ]
[1609.9755   ]
[1994.6613   ]
[ 119.38611  ]
[  52.849476 ]
[  91.65589  ]
[  71.24252  ]
[  20.63377  ]
[ 369.40286  ]
[  31.467037 ]
[  86.62618  ]
[ 318.53632  ]
[  41.24992  ]
[ 324.68845  ]
[ 142.12161  ]
[  42.469273 ]
[  51.567413 ]
[ 306.14044  ]
[   9.962955 ]
[  25.438936 ]
[  21.98471  ]
```

```
 [  39.805344 ]
 [  80.77772  ]
 [1814.0334   ]
 [  35.09123  ]
 [ 181.85687  ]
 [1810.0079   ]
 [  15.385392 ]
 [  77.5286   ]
 [ 321.46326  ]
 [ 298.77362  ]
 [ 294.73636  ]
 [  36.342262 ]
 [  79.07684  ]
 [  40.784805 ]
 [  73.701935 ]
 [ 395.38773  ]
 [  37.45575  ]
 [  34.46395  ]
 [  15.543427 ]
 [  17.272522 ]
 [ 306.18066  ]
 [  19.018204 ]
 [1803.3395   ]
 [1002.27716  ]
 [ 391.346    ]
 [  86.76046  ]
 [ 206.76811  ]
 [ 179.38518  ]
 [  92.011475 ]
 [  19.592325 ]
 [  60.82536  ]
 [  12.0183115]
 [  16.074667 ]
 [ 995.06476  ]
 [1562.1501   ]
 [  46.255352 ]
 [ 569.6869   ]
 [ 116.094795 ]
 [ 838.7636   ]
 [   8.099531 ]
 [  13.600945 ]
 [ 210.7573   ]
 [  36.883194 ]
 [  47.10232  ]
 [  41.005497 ]
 [  26.615822 ]
 [ 127.16475  ]
 [  24.048153 ]
 [  65.8284   ]
```

```
 [1795.2803   ]
 [  34.582703 ]
 [ 163.41386  ]
 [  42.417183 ]
 [  72.69895  ]
 [  11.354053 ]
 [1825.6892   ]
 [  42.337437 ]
 [  11.874674 ]
 [ 971.5875   ]
 [  25.505    ]
 [  18.467382 ]
 [  49.86763  ]
 [  43.97485  ]
 [ 159.65187  ]
 [  51.613205 ]
 [  72.152176 ]
 [ 211.69312  ]
 [ 300.35687  ]
 [ 134.99403  ]
 [1743.3043   ]
 [ 212.27455  ]
 [  37.714706 ]
 [1946.933    ]
 [  91.21702  ]
 [ 325.2393   ]
 [   9.642361 ]
 [ 131.0401   ]
 [ 130.85017  ]
 [ 212.99855  ]
 [ 364.7882   ]
 [  12.400326 ]
 [ 312.5784   ]
 [  20.398428 ]
 [ 676.8978   ]
 [ 352.73862  ]
 [  79.38332  ]
 [  27.775362 ]
 [2160.658    ]
 [  39.291565 ]
 [  20.274166 ]
 [   9.991433 ]
 [ 291.304    ]
 [  78.51953  ]
 [1556.1866   ]
 [  62.72804  ]
 [ 386.17032  ]
 [  43.023643 ]
 [1555.4072   ]
```

```
 [  10.290203 ]
 [  54.667854 ]
 [  75.84255  ]
 [ 996.3116   ]
 [ 127.895874 ]
 [  25.149614 ]
 [1489.2905   ]
 [  66.03599  ]
 [  18.791208 ]
 [  34.763813 ]
 [1164.8419   ]
 [ 308.28143  ]
 [ 523.18866  ]
 [  25.468363 ]
 [  39.459816 ]
 [  80.02945  ]
 [  12.9344015]
 [  73.548035 ]
 [  95.84306  ]
 [ 801.58374  ]
 [  12.417    ]
 [ 124.03777  ]
 [  13.910053 ]
 [1752.6012   ]
 [ 206.40996  ]
 [  85.820915 ]
 [  76.26676  ]
 [  30.11142  ]
 [  53.65784  ]
 [  35.70431  ]
 [ 119.13122  ]
 [ 370.9732   ]
 [  80.20139  ]
 [  42.438164 ]
 [   7.9133883]
 [  10.134414 ]
 [  14.574165 ]
 [  41.334415 ]
 [  66.69241  ]
 [  58.503838 ]
 [  64.99635  ]
 [ 184.8139   ]
 [ 776.3128   ]
 [ 171.20735  ]
 [ 379.02066  ]
 [ 130.34177  ]
 [ 787.4544   ]
 [ 132.14359  ]
 [ 239.01941  ]
 [  51.657166 ]
```

```
[  17.229488 ]
[1686.7213   ]
[  82.393234 ]
[  21.20742  ]
[ 978.2031   ]
[1003.0561   ]
[  53.532677 ]
[  62.395287 ]
[ 318.9065   ]
[  90.591484 ]
[  81.85484  ]
[  68.53626  ]
[  37.984493 ]
[ 176.51675  ]
[  52.336445 ]
[ 802.803    ]
[ 366.6526   ]
[  10.187926 ]
[  17.741117 ]
[ 264.88303  ]
[  17.160639 ]
[  62.677685 ]
[ 298.3335   ]
[  76.27678  ]
[  54.371334 ]
[ 680.9176   ]
[  68.95794  ]
[  41.209324 ]
[   8.847737 ]
[ 182.67148  ]
[1615.0809   ]
[1997.6471   ]
[ 119.952675 ]
[ 220.11472  ]
[  60.347027 ]
[ 677.7526   ]
[ 309.04025  ]
[ 978.7589   ]
[  41.27149  ]
[2015.2476   ]
[1850.2085   ]
[  37.15798  ]
[  46.718346 ]
[  30.248545 ]
[  40.73876  ]
[1697.201    ]
[ 573.4475   ]
[  25.146479 ]
```

```
 [ 258.58548  ]
 [ 215.58241  ]]
```

```python
# Menampilkan nilai Close yang sebenarnya (skala semula)
print("Actual Values:")
print(y_test_inverse_amzn)
```

```
Actual Values:
[[3.58739990e+02]
 [6.80000000e+01]
 [3.76899986e+01]
 [5.33125000e+01]
 [9.05599976e+01]
 [5.68100014e+01]
 [7.01699982e+01]
 [4.11250000e+01]
 [8.52969971e+02]
 [9.69859985e+02]
 [3.34380005e+02]
 [2.44850006e+02]
 [2.95059998e+02]
 [6.30625000e+01]
 [5.16890015e+02]
 [3.90999985e+01]
 [3.29750000e+02]
 [4.93400002e+01]
 [1.82354004e+03]
 [1.09260002e+02]
 [7.80000000e+02]
 [1.73565002e+03]
 [2.23229996e+02]
 [4.18600006e+01]
 [1.96394995e+03]
 [5.71599998e+01]
 [7.41200012e+02]
 [2.51250000e+02]
 [6.55729151e+00]
 [4.50000000e+00]
 [1.76429993e+02]
 [4.30999985e+01]
 [9.52099991e+01]
 [4.39599991e+01]
 [4.90625000e+00]
 [1.70100002e+01]
 [3.61790009e+02]
 [7.64599991e+01]
 [1.77342004e+03]
 [7.30000000e+01]
 [3.54700012e+01]
 [4.43510010e+02]
```

```
[4.26950012e+02]
[4.16666651e+00]
[7.05000019e+00]
[6.75890015e+02]
[3.92187500e+00]
[1.65838000e+03]
[2.75589996e+02]
[8.38799973e+01]
[7.29599991e+01]
[4.88269989e+02]
[1.85479996e+02]
[6.26549988e+02]
[3.03910004e+02]
[1.91145837e+00]
[8.23125000e+01]
[3.58690002e+02]
[3.15000000e+01]
[1.30470001e+02]
[4.86250000e+01]
[1.43079004e+03]
[8.95000000e+01]
[7.75699997e+01]
[3.47599983e+01]
[8.28719971e+02]
[1.92029999e+02]
[1.88221997e+03]
[1.17476001e+03]
[4.29309998e+02]
[4.44500008e+01]
[1.45750000e+02]
[2.13210007e+02]
[4.72900009e+01]
[5.24375000e+01]
[4.14000015e+01]
[3.27760010e+02]
[5.43679993e+02]
[7.42099991e+01]
[7.94375000e+01]
[4.94791651e+00]
[6.22500000e+01]
[9.02059998e+02]
[3.27799988e+01]
[3.83499985e+01]
[7.39583349e+00]
[3.25699997e+01]
[5.57700005e+01]
[1.27668005e+03]
[8.52530029e+02]
[2.13500000e+02]
```

```
[2.41550003e+02]
[2.75790009e+02]
[3.92700005e+01]
[2.33854175e+00]
[1.91729996e+02]
[1.35000000e+01]
[3.08309998e+02]
[1.84710007e+02]
[7.36399994e+01]
[4.29369995e+02]
[1.97131006e+03]
[2.04520004e+02]
[1.32600002e+01]
[3.05720001e+02]
[1.00300000e+03]
[1.89782996e+03]
[2.17291660e+01]
[5.95929993e+02]
[8.27500000e+01]
[4.90000000e+01]
[9.05000019e+00]
[1.91599998e+01]
[2.20104160e+01]
[1.32940002e+02]
[7.96399994e+01]
[3.67700005e+01]
[1.32731006e+03]
[7.14400024e+01]
[2.89375000e+01]
[5.78750000e+01]
[3.75600014e+01]
[1.02937500e+02]
[3.34900017e+01]
[7.80989599e+00]
[8.96229980e+02]
[3.58499985e+01]
[4.78699989e+01]
[1.78588000e+03]
[2.24059998e+02]
[1.84470001e+02]
[1.48000002e+01]
[4.40200005e+01]
[1.80130005e+02]
[1.95276001e+03]
[4.47500000e+01]
[5.41666679e+01]
[7.45899963e+01]
[2.43799992e+01]
[2.00800003e+02]
```

```
 [1.33000002e+01]
 [3.00459991e+02]
 [1.80000000e+02]
 [5.84687500e+01]
 [3.83100014e+01]
 [4.62900009e+01]
 [1.59839001e+03]
 [1.59600000e+01]
 [3.15910004e+02]
 [6.27187500e+01]
 [2.60309998e+02]
 [1.90942004e+03]
 [3.37799988e+01]
 [3.80999985e+01]
 [3.58899994e+01]
 [1.59191003e+03]
 [2.18799992e+01]
 [9.33799973e+01]
 [9.34150024e+02]
 [2.41690002e+02]
 [7.49400024e+01]
 [5.41940002e+02]
 [1.38750000e+01]
 [6.04687500e+01]
 [1.75599995e+01]
 [7.43239990e+02]
 [1.82833997e+03]
 [2.25160004e+02]
 [7.84999990e+00]
 [2.37419998e+02]
 [4.28400002e+01]
 [2.18369995e+02]
 [8.41710022e+02]
 [4.37299995e+01]
 [3.83699989e+01]
 [6.98600006e+01]
 [6.91359985e+02]
 [7.27083349e+00]
 [2.66410004e+02]
 [2.09270840e+01]
 [7.59219971e+02]
 [1.93435999e+03]
 [8.86539978e+02]
 [4.33854151e+00]
 [3.35999985e+01]
 [1.71596997e+03]
 [1.86247998e+03]
 [1.49300003e+01]
 [3.23679993e+02]
```

```
 [5.10800018e+01]
 [4.11250000e+01]
 [7.32500000e+01]
 [2.01741003e+03]
 [1.86877002e+03]
 [8.53000000e+02]
 [4.87187500e+01]
 [4.27083349e+00]
 [6.35625000e+01]
 [3.82099991e+01]
 [8.28700027e+01]
 [1.86369995e+03]
 [1.67056995e+03]
 [1.75435999e+03]
 [2.01160004e+02]
 [4.38559998e+02]
 [1.56613000e+03]
 [1.76270004e+02]
 [4.27900009e+01]
 [4.52083349e+00]
 [1.91300003e+02]
 [3.61079987e+02]
 [5.28750000e+01]
 [3.46760010e+02]
 [2.30250000e+02]
 [8.40100021e+01]
 [1.24639999e+02]
 [7.94687500e+01]
 [1.97273999e+03]
 [5.63909973e+02]
 [8.45400009e+01]
 [3.63199997e+01]
 [4.27999992e+01]
 [1.35559998e+02]
 [3.91500015e+01]
 [5.03199997e+01]
 [4.45499992e+01]
 [1.99270004e+02]
 [7.69159973e+02]
 [4.60099983e+01]
 [1.08199997e+01]
 [2.96519989e+02]
 [8.24375000e+01]
 [1.38470001e+02]
 [8.21500015e+01]
 [2.25799999e+01]
 [1.53900003e+01]
 [1.84320007e+02]
 [1.39000000e+03]
```

```
 [2.54300003e+01]
 [4.35999985e+01]
 [1.97089996e+02]
 [3.94599991e+01]
 [5.62500000e+01]
 [1.52748999e+03]
 [1.41729996e+02]
 [1.62594995e+03]
 [4.20800018e+01]
 [1.78899994e+02]
 [9.34499969e+01]
 [5.95900002e+01]
 [7.57099991e+01]
 [1.84429993e+02]
 [6.91250000e+01]
 [5.18369995e+02]
 [3.46399994e+01]
 [5.67299995e+01]
 [6.38600006e+01]
 [6.88125000e+01]
 [9.90330017e+02]
 [1.31589996e+02]
 [1.96690002e+02]
 [1.24700003e+01]
 [3.91800003e+01]
 [3.70900002e+01]
 [1.17129997e+02]
 [5.29419983e+02]
 [2.25100002e+01]
 [1.61200008e+01]
 [2.07800007e+01]
 [7.94300003e+01]
 [1.94699993e+01]
 [3.39700012e+01]
 [1.98333340e+01]
 [1.22550003e+02]
 [2.68899994e+02]
 [1.44399996e+01]
 [8.11250000e+01]
 [4.87200012e+01]
 [3.58400002e+01]
 [1.16400003e+01]
 [6.76010010e+02]
 [3.38290009e+02]
 [6.05800018e+01]
 [4.04399986e+01]
 [1.94430005e+03]
 [2.99709991e+02]
 [7.28999996e+00]
```

```
 [7.08125000e+01]
 [4.73199997e+01]
 [4.08400002e+01]
 [2.36809998e+02]
 [1.43125000e+01]
 [4.27809998e+02]
 [5.17599983e+01]
 [7.67399979e+01]
 [7.00000000e+00]
 [1.39499998e+01]
 [1.56999998e+01]
 [1.92377002e+03]
 [2.26900005e+01]
 [7.40000000e+01]
 [3.03029999e+02]
 [5.26250000e+01]
 [3.66500015e+01]
 [3.23000000e+02]
 [4.02299995e+01]
 [4.31420013e+02]
 [2.26690002e+02]
 [3.79500008e+01]
 [7.50625000e+01]
 [5.35219971e+02]
 [5.10200005e+01]
 [1.98229160e+01]
 [1.88050003e+02]
 [9.12900009e+01]
 [2.73790009e+02]
 [1.95699997e+01]
 [1.69799995e+01]
 [1.63331006e+03]
 [4.32200012e+01]
 [4.64399986e+01]
 [2.58480011e+02]
 [5.59500000e+02]
 [9.14375000e+01]
 [2.35479996e+02]
 [1.77419998e+02]
 [1.84775000e+03]
 [1.83700008e+01]
 [2.74130005e+02]
 [6.89583349e+00]
 [1.86529999e+02]
 [7.50000000e+01]
 [5.31070007e+02]
 [1.47399998e+01]
 [1.79490005e+02]
 [1.76289993e+02]
```

```
 [1.02299995e+01]
 [1.91630005e+02]
 [3.09660004e+02]
 [9.63000011e+00]
 [2.99660004e+02]
 [3.97916675e+00]
 [4.28125000e+01]
 [1.82255005e+03]
 [8.96500015e+01]
 [7.97900009e+01]
 [2.63600006e+01]
 [3.71666679e+01]
 [8.91500015e+01]
 [4.66369995e+02]
 [1.92339996e+02]
 [9.02799988e+01]
 [6.93499985e+01]
 [7.73099976e+01]
 [1.52700005e+01]
 [2.53750000e+01]
 [2.66880005e+02]
 [7.19070007e+02]
 [1.55147998e+03]
 [1.97018994e+03]
 [1.26839996e+02]
 [5.17799988e+01]
 [8.87300034e+01]
 [7.20400009e+01]
 [1.66399994e+01]
 [3.70559998e+02]
 [3.06700001e+01]
 [8.85625000e+01]
 [3.30540009e+02]
 [3.92599983e+01]
 [3.27619995e+02]
 [1.43630005e+02]
 [3.88499985e+01]
 [4.97700005e+01]
 [3.07359985e+02]
 [4.45833349e+00]
 [2.20499992e+01]
 [1.80520840e+01]
 [3.86100006e+01]
 [7.59375000e+01]
 [1.74160999e+03]
 [3.04375000e+01]
 [1.93729996e+02]
 [1.74962000e+03]
 [1.02500000e+01]
```

```
 [8.30500031e+01]
 [3.32630005e+02]
 [3.00989990e+02]
 [2.96760010e+02]
 [3.27599983e+01]
 [8.73125000e+01]
 [3.88499985e+01]
 [5.65000000e+01]
 [3.97970001e+02]
 [3.46300011e+01]
 [3.29375000e+01]
 [1.22500000e+01]
 [1.25500002e+01]
 [3.07190002e+02]
 [1.43750000e+01]
 [1.68915002e+03]
 [9.76780029e+02]
 [3.84200012e+02]
 [8.60899963e+01]
 [2.18279999e+02]
 [1.83389999e+02]
 [9.43099976e+01]
 [1.51700001e+01]
 [5.75468750e+01]
 [5.96999979e+00]
 [1.18500004e+01]
 [9.82010010e+02]
 [1.58651001e+03]
 [4.34000015e+01]
 [5.60479980e+02]
 [1.17889999e+02]
 [8.21359985e+02]
 [2.33333325e+00]
 [8.01000023e+00]
 [2.16479996e+02]
 [3.38199997e+01]
 [4.40099983e+01]
 [3.77799988e+01]
 [1.82500000e+01]
 [1.33750000e+02]
 [2.23333340e+01]
 [6.52500000e+01]
 [1.76403003e+03]
 [3.27599983e+01]
 [1.71990005e+02]
 [3.93899994e+01]
 [6.85400009e+01]
 [6.95833349e+00]
 [1.80800000e+03]
```

```
 [4.27000008e+01]
 [6.90999985e+00]
 [9.89580017e+02]
 [2.02999992e+01]
 [1.52700005e+01]
 [5.01875000e+01]
 [3.75200005e+01]
 [1.65320007e+02]
 [5.09375000e+01]
 [7.49375000e+01]
 [2.18350006e+02]
 [3.15329987e+02]
 [1.30929993e+02]
 [1.77383997e+03]
 [2.16419998e+02]
 [3.41800003e+01]
 [2.00238000e+03]
 [8.86699982e+01]
 [3.26269989e+02]
 [4.36458349e+00]
 [1.30229996e+02]
 [1.24980003e+02]
 [2.31320007e+02]
 [3.73000000e+02]
 [7.46000004e+00]
 [3.24929993e+02]
 [1.69799995e+01]
 [6.57909973e+02]
 [3.58140015e+02]
 [8.03499985e+01]
 [2.34375000e+01]
 [2.14987012e+03]
 [3.62200012e+01]
 [1.55312500e+01]
 [4.54166651e+00]
 [2.99640015e+02]
 [7.91800003e+01]
 [1.58094995e+03]
 [6.17500000e+01]
 [4.02200012e+02]
 [4.48199997e+01]
 [1.65942004e+03]
 [5.06250000e+00]
 [5.25900002e+01]
 [7.73000031e+01]
 [9.89840027e+02]
 [1.24589996e+02]
 [2.17900009e+01]
 [1.53764001e+03]
```

```
 [7.48125000e+01]
 [1.24899998e+01]
 [3.02199993e+01]
 [1.16946997e+03]
 [3.18119995e+02]
 [5.22369995e+02]
 [2.17999992e+01]
 [3.70499992e+01]
 [7.01875000e+01]
 [7.85416651e+00]
 [7.14400024e+01]
 [9.00000000e+01]
 [7.99020020e+02]
 [7.16666651e+00]
 [1.22769997e+02]
 [8.36999989e+00]
 [1.64281006e+03]
 [1.89250000e+02]
 [8.31800003e+01]
 [7.76999969e+01]
 [2.68999996e+01]
 [6.25000000e+01]
 [3.33199997e+01]
 [1.23830002e+02]
 [3.75429993e+02]
 [8.11600037e+01]
 [4.25699997e+01]
 [2.11458325e+00]
 [4.75000000e+00]
 [1.26000004e+01]
 [4.01399994e+01]
 [8.41875000e+01]
 [6.20400009e+01]
 [5.60000000e+01]
 [1.89679993e+02]
 [7.60770020e+02]
 [1.73529999e+02]
 [3.69510010e+02]
 [1.28360001e+02]
 [7.68489990e+02]
 [1.42250000e+02]
 [2.47490005e+02]
 [4.87500000e+01]
 [1.28333330e+01]
 [1.69918994e+03]
 [7.80999985e+01]
 [1.69375000e+01]
 [9.94130005e+02]
 [1.01003998e+03]
 [5.49099998e+01]
```

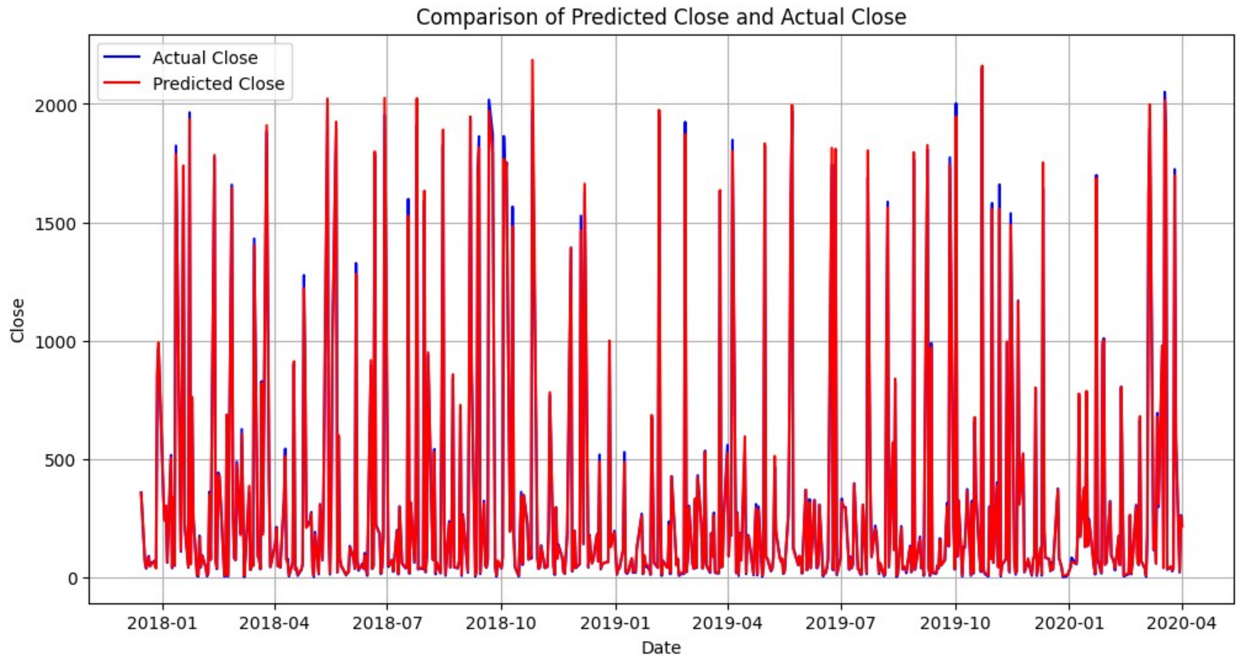
```
 [6.09375000e+01]
 [3.22440002e+02]
 [9.09100037e+01]
 [8.01200027e+01]
 [6.55000000e+01]
 [3.00625000e+01]
 [1.75389999e+02]
 [5.34375000e+01]
 [8.05750000e+02]
 [3.75040009e+02]
 [4.83854151e+00]
 [1.42700005e+01]
 [2.57890015e+02]
 [1.40400000e+01]
 [6.85625000e+01]
 [3.06100006e+02]
 [8.55000000e+01]
 [5.30299988e+01]
 [6.63700012e+02]
 [6.97812500e+01]
 [3.76800003e+01]
 [3.09375000e+00]
 [1.89029999e+02]
 [1.62209998e+03]
 [1.89853003e+03]
 [1.15940002e+02]
 [2.24389999e+02]
 [5.90000000e+01]
 [6.93969971e+02]
 [2.98230011e+02]
 [9.79469971e+02]
 [3.99799995e+01]
 [2.05022998e+03]
 [1.89801001e+03]
 [3.34099998e+01]
 [4.41199989e+01]
 [2.60300007e+01]
 [3.78699989e+01]
 [1.72379004e+03]
 [5.93859985e+02]
 [2.16399994e+01]
 [2.63250000e+02]
 [2.20570007e+02]]
```

```python
import numpy as np

# Menghitung RMSE
rmse = np.sqrt(np.mean((y_pred_inverse_amzn -
y_test_inverse_amzn)**2))
```

```python
# Menghitung MAE
mae = np.mean(np.abs(y_pred_inverse_amzn - y_test_inverse_amzn))

# Menghitung MAPE
mape = np.mean(np.abs((y_test_inverse_amzn - y_pred_inverse_amzn) /
y_test_inverse_amzn)) * 100

# Menampilkan hasil evaluasi
print("RMSE on Test Data:", rmse)
print("MAE on Test Data:", mae)
print("MAPE on Test Data:", mape)
```

```
RMSE on Test Data: 20.650520825819033
MAE on Test Data: 10.060721353731221
MAPE on Test Data: 13.862660145821687
```

```python
import matplotlib.pyplot as plt

# Mengambil tanggal dari data test
test_dates_amzn = df_amzn.index[-len(y_test_inverse_amzn):]

# Plot hasil prediksi dan nilai Close yang sebenarnya
plt.figure(figsize=(12, 6))
plt.plot(test_dates_amzn, y_test_inverse_amzn, label='Actual Close',
color='blue')
plt.plot(test_dates_amzn, y_pred_inverse_amzn, label='Predicted
Close', color='red')
plt.xlabel('Date')
plt.ylabel('Close')
plt.title('Comparison of Predicted Close and Actual Close')
plt.legend()
plt.grid(True)
plt.show()
```



No. 1c - AMAZN

Saya menggunakan metode grid search untuk menemukan nilai hyperparameter yang optimal karena grid search dapat secara sistematis mengevaluasi berbagai kombinasi nilai hyperparameter dan memilih kombinasi yang menghasilkan kinerja model terbaik sehingga membantu menghemat waktu dan sumber daya karena mencari nilai hiperparameter optimal secara manual dapat memakan waktu yang lama. Untuk memperbaiki kinerja model, terdapat beberapa pendekatan yang dapat dilakukan seperti:

1. Menambah atau mengurangi layer pada model
2. Mengubah jumlah neuron pada setiap layer
3. Mengubah learning rate pada optimizer
4. Menggunakan teknik regularisasi seperti dropout atau L2 regularization
5. Memilih metrik evaluasi yang lebih tepat Berikut adalah beberapa perubahan yang dapat dilakukan pada arsitektur baseline untuk meningkatkan kinerjanya:
6. Menambahkan layer Dropout: Layer dropout dapat membantu mencegah overfitting dengan secara acak menghilangkan neuron selama pelatihan.

```
from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import GridSearchCV
from keras.optimizers import Adam
from keras.layers import Dropout

# Fungsi untuk membangun model LSTM
def build_lstm_model(units =50, activation = 'relu', 
learning_rate=0.001, dropout=0.0):
    model = Sequential()
```

```
    model.add(LSTM(units=units, activation=activation,
input_shape=(window_size, 1)))
    model.add(Dropout(dropout))
    model.add(Dense(1))
    optimizer = Adam(learning_rate=learning_rate)
    model.compile(optimizer=optimizer, loss='mean_squared_error')
    return model

# Membungkus model dengan KerasRegressor agar dapat digunakan dengan
GridSearchCV
lstm_model = KerasRegressor(build_fn=build_lstm_model, verbose=0)
```

```
<ipython-input-48-ca58bac11d11>:2: DeprecationWarning: KerasRegressor
is deprecated, use Sci-Keras (https://github.com/adriangb/scikeras)
instead. See https://www.adriangb.com/scikeras/stable/migration.html
for help migrating.
  lstm_model = KerasRegressor(build_fn=build_lstm_model, verbose=0)
```

```
# Menentukan hyperparameter yang akan diuji
param_grid = {
    'units': [32, 64],
    'activation': ['relu', 'tanh'],
    'learning_rate': [0.001, 0.01, 0.1],
    'dropout': [0.0, 0.2],
    'epochs': [10, 20]
}

# Membangun objek GridSearchCV untuk dataset AMZN
grid_search_amzn = GridSearchCV(estimator=lstm_model,
param_grid=param_grid, cv=3)

grid_search_amzn.fit(X_train_amzn, y_train_amzn)
```

```
GridSearchCV(cv=3,
             estimator=<keras.wrappers.scikit_learn.KerasRegressor
object at 0x7fc8adb715d0>,
             param_grid={'activation': ['relu', 'tanh'], 'dropout':
[0.0, 0.2],
                         'epochs': [10, 20],
                         'learning_rate': [0.001, 0.01, 0.1],
                         'units': [32, 64]})
```

```
# Menampilkan hasil pencarian kombinasi terbaik untuk dataset AMZN
print("Best Parameters for AMZN:", grid_search_amzn.best_params_)
print("Best Score (RMSE) for AMZN:", np.sqrt(-
grid_search_amzn.best_score_))
```

```
Best Parameters for AMZN: {'activation': 'tanh', 'dropout': 0.0,
'epochs': 10, 'learning_rate': 0.1, 'units': 32}
Best Score (RMSE) for AMZN: 0.006721167045739181
```

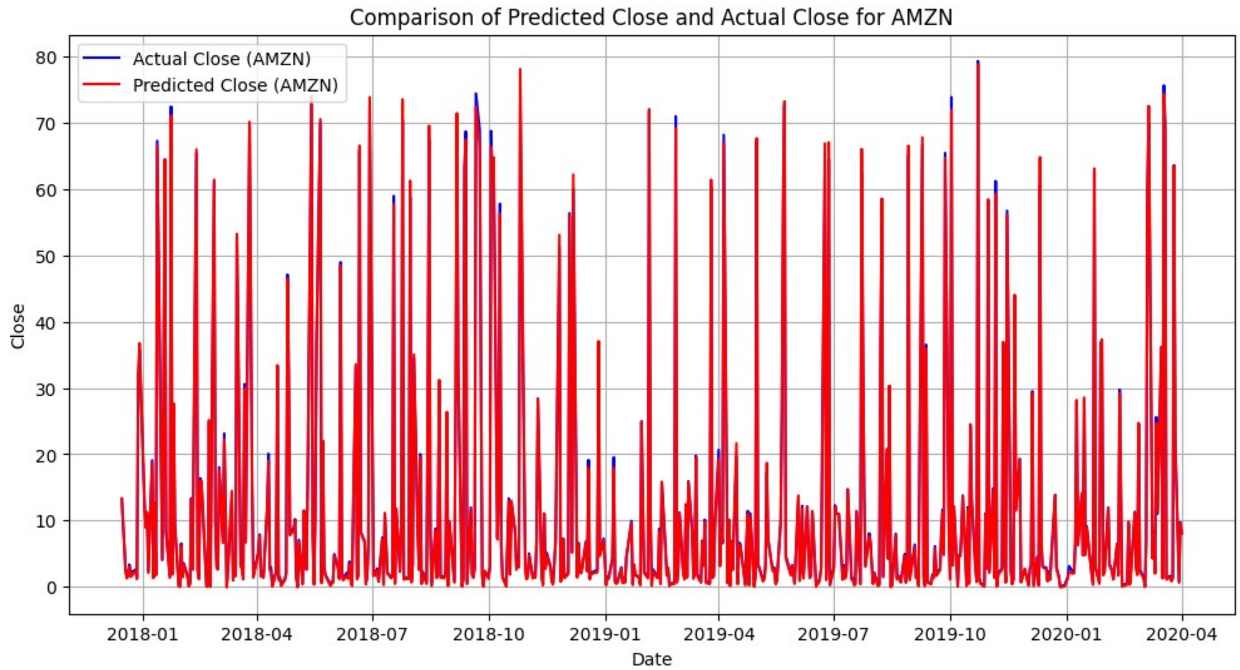
Dari hasil di atas diperoleh bahwa parameter terbaik yang menghasikan akurasi optimal adalah

1. 'epochs': 10,
2. 'optimizer': 'adam' dengan 'learn_rate': 0.01
3. 'Units' : 32
4. 'activation' : 'tanh'

Model arsitektur modifikasi

```
# Membangun model LSTM
model = Sequential()
model.add(LSTM(32, activation='tanh', input_shape=(window_size, 1)))
model.add(Dense(1))

from tensorflow.keras.optimizers import Adam

# Kompilasi model
optimizer = Adam(learning_rate=0.1)
model.compile(optimizer=optimizer, loss='mean_squared_error',
metrics=['mae'])

print(model.summary())

Model: "sequential_147"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_147 (LSTM)             (None, 32)                4352

 dense_147 (Dense)           (None, 1)                 33

=================================================================
Total params: 4,385
Trainable params: 4,385
Non-trainable params: 0
_________________________________________________________________
None

# Latih model dengan data train
modelamzn=model.fit(X_train_amzn, y_train_amzn, epochs=10,
batch_size=32)

Epoch 1/10
144/144 [==============================] - 2s 11ms/step - loss:
6.8733e-05 - mae: 0.0055
Epoch 2/10
144/144 [==============================] - 1s 9ms/step - loss:
1.2039e-04 - mae: 0.0080
Epoch 3/10
144/144 [==============================] - 1s 7ms/step - loss:
1.6743e-04 - mae: 0.0089
```

```
Epoch 4/10
144/144 [==============================] - 1s 5ms/step - loss:
1.0699e-04 - mae: 0.0072
Epoch 5/10
144/144 [==============================] - 1s 5ms/step - loss:
1.1527e-04 - mae: 0.0077
Epoch 6/10
144/144 [==============================] - 1s 4ms/step - loss:
6.2729e-05 - mae: 0.0053
Epoch 7/10
144/144 [==============================] - 1s 5ms/step - loss:
1.0720e-04 - mae: 0.0074
Epoch 8/10
144/144 [==============================] - 1s 5ms/step - loss:
1.0751e-04 - mae: 0.0069
Epoch 9/10
144/144 [==============================] - 1s 5ms/step - loss:
7.9258e-05 - mae: 0.0058
Epoch 10/10
144/144 [==============================] - 1s 5ms/step - loss:
7.9482e-05 - mae: 0.0063

# Melakukan prediksi menggunakan model terbaik untuk dataset AMZN
best_model_amzn = grid_search_amzn.best_estimator_
y_pred_amzn = best_model_amzn.predict(X_test_amzn)

# Menampilkan plot hasil prediksi dan nilai Close yang sebenarnya
untuk dataset AMZN
plt.figure(figsize=(12, 6))
plt.plot(test_dates_amzn, y_test_inverse_amzn, label='Actual Close
(AMZN)', color='blue')
plt.plot(test_dates_amzn, y_pred_inverse_amzn, label='Predicted Close
(AMZN)', color='red')
plt.xlabel('Date')
plt.ylabel('Close')
plt.title('Comparison of Predicted Close and Actual Close for AMZN')
plt.legend()
plt.grid(True)
plt.show()
```



NO. 1D - AMZN

```
from sklearn.metrics import mean_squared_error, mean_absolute_error

# Evaluasi LSTM biasa untuk dataset AMZN
y_pred_lstm_amzn = model.predict(X_test_amzn)
y_pred_inverse_lstm_amzn = scaler.inverse_transform(y_pred_lstm_amzn)
y_test_inverse_amzn = scaler.inverse_transform(y_test_amzn)

rmse_lstm_amzn = np.sqrt(mean_squared_error(y_test_inverse_amzn,
y_pred_inverse_lstm_amzn))
mae_lstm_amzn = mean_absolute_error(y_test_inverse_amzn,
y_pred_inverse_lstm_amzn)
mape_lstm_amzn = np.mean(np.abs((y_test_inverse_amzn -
y_pred_inverse_lstm_amzn) / y_test_inverse_amzn)) * 100

print("Performance of LSTM Architecture for AMZN:")
print("RMSE:", rmse_lstm_amzn)
print("MAE:", mae_lstm_amzn)
print("MAPE:", mape_lstm_amzn)

18/18 [==============================] - 0s 7ms/step
Performance of LSTM Architecture for AMZN:
RMSE: 20.650520825819033
MAE: 10.060721353731221
MAPE: 13.862660145821687
```

```
# Evaluasi LSTM dengan modifikasi GridSearch untuk dataset AMZN
y_pred_gridsearch_amzn = best_model_amzn.predict(X_test_amzn)
y_pred_inverse_gridsearch_amzn =
scaler.inverse_transform(y_pred_gridsearch_amzn.reshape(-1, 1))

rmse_gridsearch_amzn = np.sqrt(mean_squared_error(y_test_inverse_amzn,
y_pred_inverse_gridsearch_amzn))
mae_gridsearch_amzn = mean_absolute_error(y_test_inverse_amzn,
y_pred_inverse_gridsearch_amzn)
mape_gridsearch_amzn = np.mean(np.abs((y_test_inverse_amzn -
y_pred_inverse_gridsearch_amzn) / y_test_inverse_amzn)) * 100

print("Performance of LSTM Architecture with GridSearch for AMZN:")
print("RMSE:", rmse_gridsearch_amzn)
print("MAE:", mae_gridsearch_amzn)
print("MAPE:", mape_gridsearch_amzn)
```

```
Performance of LSTM Architecture with GridSearch for AMZN:
RMSE: 0.6307800071601061
MAE: 0.33040755221125595
MAPE: 11.289919143930744
```

RMSE (Root Mean Squared Error): Nilai RMSE menggambarkan tingkat kesalahan prediksi dalam satuan yang sama dengan variabel target. Semakin rendah nilai RMSE, semakin baik model dalam memprediksi nilai Close yang sebenarnya. Jadi, kita membandingkan nilai RMSE antara kedua arsitektur LSTM. Hasil RMSE LSTM dengan modifikasi GridSearch lebih rendah, itu menunjukkan bahwa model tersebut memberikan prediksi yang lebih akurat. RMSE model arsitektur baseline: 20.65 RMSE model arsitektur modifikasi : 0.63

MAE (Mean Absolute Error): Nilai MAE merupakan nilai rata-rata dari selisih absolut antara nilai prediksi dan nilai Close yang sebenarnya. Semakin rendah nilai MAE, semakin baik model dalam memprediksi nilai Close yang sebenarnya. Kita dapat membandingkan nilai MAE antara kedua arsitektur LSTM dan dapat kita lihat bahwa nilai MAE pada gridsearch lebih rendah (0.3304) jadi model ini lebih memberikan prediksi yang akurat mana yang memiliki MAE yang lebih rendah untuk menentukan model yang lebih baik.

MAPE adalah singkatan dari "Mean Absolute Percentage Error" atau "Kesalahan Persentase Mutlak Rata-Rata". Ini adalah metrik yang digunakan untuk mengukur tingkat kesalahan atau ketidakakuratan dalam ramalan atau perkiraan. Pada arsitektur gridsearch mendapatkan nilai MAPE yang lebih rendah (11.2899) daripada arsitektur baseline (13.86)

Dengan ini maka model arsitektur yang sudah dimodifikasi menggunakan hyperparameter tunning gridsearch lebih cocok daripada menggunakan model arsitektur baseline.

No. 1e

Link Google Colab:
https://drive.google.com/drive/folders/1fVu56rHJ7ojt87U3utHyqixNhaPjlbd7?usp=sharing

2502014210 - Eleanor Maritsa Maharani

No. 1a

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.activations import relu
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import GridSearchCV
from keras.layers import Dense, Dropout
from sklearn.metrics import make_scorer
```

Pada kasus ini saya hanya akan menggunakan kolom date dan close saja. Kedua dataset timeseries yang saya dapatkan yaitu dataset AMZN dan dataset CSCO

```
# Membaca dataset CSCO.csv
df_csco = pd.read_csv('CSCO.csv', usecols=['Date', 'Close'])
df_csco['Date'] = pd.to_datetime(df_csco['Date'])  # Mengubah kolom
Date menjadi tipe datetime
df_csco.set_index('Date', inplace=True)  # Menggunakan kolom Date
sebagai index
```

Pada dataset CSCO, kolom close memiliki 7589 records dengan tipe data float

```
# Eksplorasi data CSCO
print("Info dataset CSCO:")
print(df_csco.info())

Info dataset CSCO:
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 7589 entries, 1990-02-16 to 2020-04-01
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Close   7589 non-null   float64
dtypes: float64(1)
memory usage: 118.6 KB
None

print("Statistik deskriptif dataset CSCO:")
print(df_csco.describe())
```

```
Statistik deskriptif dataset CSCO:
             Close
count  7589.000000
mean     20.399541
std      14.906589
min       0.071181
25%       8.479167
50%      19.680000
75%      27.120001
max      80.062500
```

```
print("Lima data teratas dataset CSCO:")
print(df_csco.head())
```

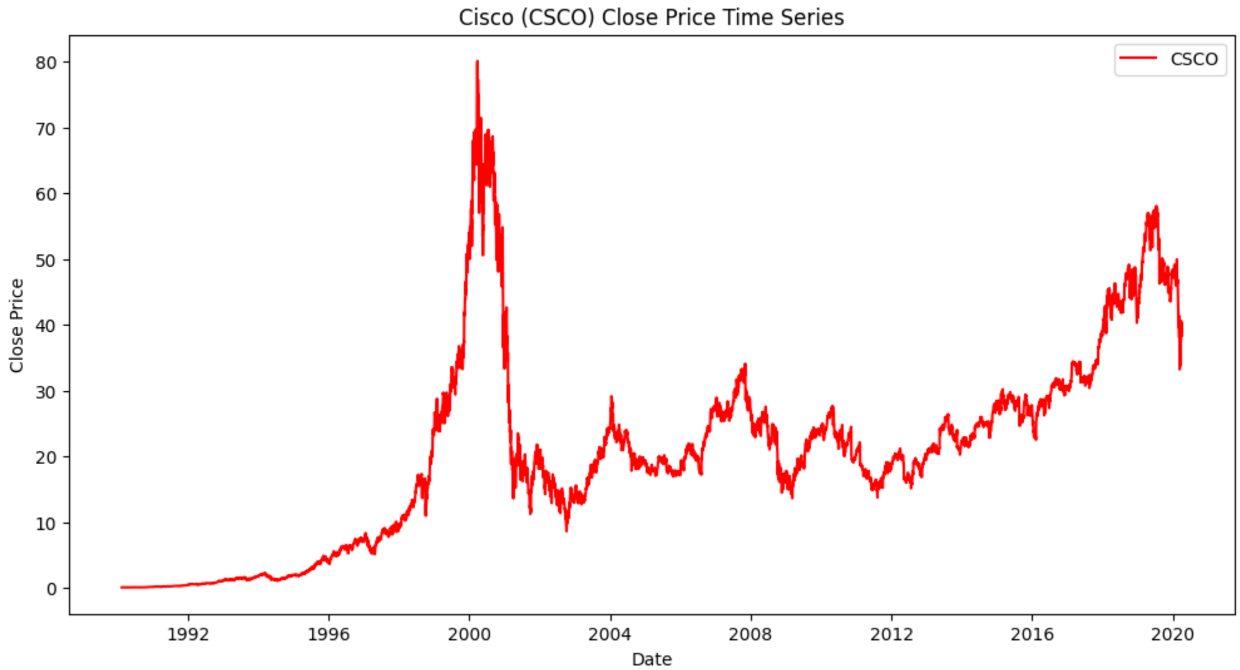
```
Lima data teratas dataset CSCO:
               Close
Date
1990-02-16  0.077257
1990-02-20  0.079861
1990-02-21  0.078125
1990-02-22  0.078993
1990-02-23  0.078559
```

Dataset CSCO juga tidak memiliki missing value

```
# Mengecek jumlah nilai yang hilang pada setiap kolom
print(df_csco.isnull().sum())
```

```
Close    0
dtype: int64
```

```
# Visualisasi data time series CSCO
plt.figure(figsize=(12, 6))
plt.plot(df_csco.index, df_csco['Close'], label='CSCO', color='r')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.title('Cisco (CSCO) Close Price Time Series')
plt.legend()
plt.show()
```



Disini saya akan memisahkan data time series menjadi input dan output. Input atau window size adalah jumlah langkah waktu dari data historis yang digunakan untuk memprediksi. Window size saya inisialisasi 5, karena saya mau mengambil dari hari senin sampai dengan hari jumat. Output atau horizon adalah jumlah langkah waktu untuk memprediksi ke masa depan. Horizon saya inisialisas 1 karena hanya untuk hari senin saja.

```
# Split function
def split_data_time_series(df, window_size, horizon):
    x, y = [], []
    for i in range(len(df) - window_size - horizon + 1):
        x.append(df.iloc[i:i+window_size]['Close'].values)
        y.append(df.iloc[i+window_size+horizon-1]['Close'])
    return np.array(x), np.array(y)

window_size = 5
horizon = 1

# Menggunakan fungsi split_data_time_series untuk memisahkan data CSCO
x_csco, y_csco = split_data_time_series(df_csco, window_size, horizon)
```

Problem data : scalling data

Scaling data adalah proses transformasi data sehingga nilainya memiliki rentang yang sama atau setidaknya skalanya serupa. Teknik penskalaan data yang akan saya gunakan adalah normalisasi min-max scaling. Alasan menggunakan teknik penskalaan min-max karena teknik ini mengubah nilai numerik menjadi rentang 0 hingga 1, sehingga semua fitur memiliki rentang nilai dan tidak mempengaruhi kinerja model. Scalling data perlu dilakukan karena jika ada variabel numerik dengan skala yang berbeda-beda pada dataset perlu dilakukan scaling data agar skala variabel tersebut seimbang dan dapat digunakan dalam model.

```
# Skalasi data menggunakan MinMaxScaler
scaler = MinMaxScaler()
x_csco = scaler.fit_transform(x_csco)
y_csco = scaler.fit_transform(y_csco.reshape(-1, 1))

print("Contoh input CSCO:")
print(x_csco[0])  # Contoh input pertama dari CSCO
print("\nOutput yang sesuai untuk contoh input CSCO:")
print(y_csco[0])  # Output yang sesuai untuk contoh input pertama dari
CSCO

Contoh input CSCO:
[7.59631864e-05 1.08518771e-04 8.68150170e-05 9.76668475e-05
 9.22409323e-05]

Output yang sesuai untuk contoh input CSCO:
[6.51112627e-05]

from sklearn.model_selection import train_test_split
```

Memisahkan dataset menjadi train, test dan validation set dengan ketentuan 80 train, 10 val, 10 test

```
# Pisahkan data CSCO menjadi train, validation, dan test set
X_train_csco, X_temp_csco, y_train_csco, y_temp_csco =
train_test_split(x_csco, y_csco, test_size=0.2, random_state=42)
X_val_csco, X_test_csco, y_val_csco, y_test_csco =
train_test_split(X_temp_csco, y_temp_csco, test_size=0.5,
random_state=42)

# Menampilkan jumlah data pada setiap set
print("Jumlah data dalam setiap set:")
print("Train set (CSCO):", len(X_train_csco))
print("Validation set (CSCO):", len(X_val_csco))
print("Test set (CSCO):", len(X_test_csco))

Jumlah data dalam setiap set:
Train set (CSCO): 6067
Validation set (CSCO): 758
Test set (CSCO): 759

# Ubah bentuk data agar sesuai dengan input yang diperlukan oleh LSTM
(jumlah data, jumlah time steps, jumlah fitur)
X_train_csco = X_train_csco.reshape(X_train_csco.shape[0],
window_size, 1)
X_test_csco = X_test_csco.reshape(X_test_csco.shape[0], window_size,
1)
X_val_csco = X_val_csco.reshape(X_val_csco.shape[0], window_size, 1)
```

No. 1B - CSCO

Membuat arsitektur baseline dengan LSTM (units=50) dan layer akhir berupa node Perceptron dengan units=1. Activation function untuk LSTM menggunakan ReLU

```
# Membangun model LSTM
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(window_size, 1)))
model.add(Dense(1))

# Kompilasi model
model.compile(optimizer='adam', loss='mean_squared_error',
metrics=['mae'])

print(model.summary())
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_1 (LSTM)               (None, 50)                10400

 dense_1 (Dense)             (None, 1)                 51

=================================================================
Total params: 10,451
Trainable params: 10,451
Non-trainable params: 0
_________________________________________________________________
None
```

```
# Latih model dengan data train
model.fit(X_train_csco, y_train_csco, epochs=5, batch_size=32,
validation_data=(X_val_csco, y_val_csco))
```

```
Epoch 1/5
190/190 [==============================] - 2s 8ms/step - loss:
9.2633e-05 - mae: 0.0058 - val_loss: 7.8968e-05 - val_mae: 0.0052
Epoch 2/5
190/190 [==============================] - 1s 7ms/step - loss:
9.5913e-05 - mae: 0.0063 - val_loss: 8.0703e-05 - val_mae: 0.0052
Epoch 3/5
190/190 [==============================] - 1s 5ms/step - loss:
8.7541e-05 - mae: 0.0056 - val_loss: 1.0395e-04 - val_mae: 0.0067
Epoch 4/5
190/190 [==============================] - 1s 5ms/step - loss:
9.2041e-05 - mae: 0.0060 - val_loss: 1.1933e-04 - val_mae: 0.0079
Epoch 5/5
190/190 [==============================] - 1s 5ms/step - loss:
9.3112e-05 - mae: 0.0059 - val_loss: 8.8984e-05 - val_mae: 0.0063
```

```
<keras.callbacks.History at 0x7f66c4178190>
```

```
# Evaluate the model on the test data using
print("Evaluate on test data")
results = model.evaluate(X_test_csco, y_test_csco, batch_size=64)
print("Test MSE:", results[0])
```

```
Evaluate on test data
12/12 [==============================] - 0s 3ms/step - loss: 6.8436e-
05 - mae: 0.0050
Test MSE: 6.843644223408774e-05
```

```
# Melakukan prediksi pada data test
y_pred_csco = model.predict(X_test_csco)

# Menampilkan hasil prediksi
print("Predicted Values:")
print(y_pred_csco)
```

```
24/24 [==============================] - 0s 2ms/step
Predicted Values:
[[0.05585746]
 [0.24428965]
 [0.01920731]
 [0.7018409 ]
 [0.20582461]
 [0.3139723 ]
 [0.7124329 ]
 [0.54325926]
 [0.8325493 ]
 [0.40888888]
 [0.6252383 ]
 [0.8096414 ]
 [0.29978436]
 [0.6948901 ]
 [0.40078574]
 [0.02316147]
 [0.19943959]
 [0.333094  ]
 [0.02646684]
 [0.16987573]
 [0.38284248]
 [0.30562997]
 [0.2569818 ]
 [0.24457268]
 [0.06316809]
 [0.01250701]
 [0.7977876 ]
 [0.00845774]
 [0.30592692]
```

```
 [0.07940407]
 [0.00679491]
 [0.3197508 ]
 [0.24871954]
 [0.40199816]
 [0.00560926]
 [0.00569728]
 [0.3725223 ]
 [0.32460576]
 [0.2767042 ]
 [0.2169679 ]
 [0.05179054]
 [0.26348028]
 [0.35249275]
 [0.35508698]
 [0.01085307]
 [0.36694533]
 [0.3615694 ]
 [0.00578414]
 [0.00557354]
 [0.380104  ]
 [0.869349  ]
 [0.00556642]
 [0.5955955 ]
 [0.00711731]
 [0.62724775]
 [0.35254377]
 [0.02874692]
 [0.36238492]
 [0.81911117]
 [0.00589304]
 [0.35054433]
 [0.30365586]
 [0.02182649]
 [0.19804536]
 [0.33167022]
 [0.04452807]
 [0.3014384 ]
 [0.46606123]
 [0.02888734]
 [0.02245275]
 [0.34220046]
 [0.00927647]
 [0.19653638]
 [0.02023125]
 [0.11426175]
 [0.35642827]
 [0.01029081]
 [0.01162249]
```

```
 [0.29301852]
 [0.2873088 ]
 [0.1843908 ]
 [0.08549786]
 [0.02142222]
 [0.18882388]
 [0.26743007]
 [0.02289924]
 [0.01509859]
 [0.6105197 ]
 [0.2414743 ]
 [0.00570103]
 [0.04283491]
 [0.02912011]
 [0.29220814]
 [0.2740882 ]
 [0.41198403]
 [0.01266942]
 [0.09965491]
 [0.34323412]
 [0.00743733]
 [0.01160552]
 [0.08456751]
 [0.65745634]
 [0.20796223]
 [0.05212244]
 [0.16847333]
 [0.35720444]
 [0.24406849]
 [0.00584737]
 [0.22275622]
 [0.15248193]
 [0.4543516 ]
 [0.32288867]
 [0.02852679]
 [0.24123698]
 [0.22002666]
 [0.3109017 ]
 [0.18942653]
 [0.40652174]
 [0.00591391]
 [0.23843594]
 [0.39655644]
 [0.42822278]
 [0.21975137]
 [0.01123919]
 [0.29901022]
 [0.3337441 ]
 [0.29098403]
```

```
 [0.26501843]
 [0.33003318]
 [0.02638359]
 [0.3433739 ]
 [0.24911991]
 [0.3119952 ]
 [0.28890103]
 [0.1256932 ]
 [0.22575708]
 [0.11388406]
 [0.0232314 ]
 [0.22759356]
 [0.34874737]
 [0.27237177]
 [0.22824351]
 [0.23669457]
 [0.38524866]
 [0.26304874]
 [0.30479497]
 [0.55872065]
 [0.2394418 ]
 [0.19712676]
 [0.32501888]
 [0.16101202]
 [0.2469641 ]
 [0.1078844 ]
 [0.7167613 ]
 [0.22824802]
 [0.5397329 ]
 [0.31566536]
 [0.56580824]
 [0.35833502]
 [0.15600255]
 [0.00761415]
 [0.59163266]
 [0.33532697]
 [0.6002961 ]
 [0.58299905]
 [0.3263473 ]
 [0.30209917]
 [0.26437593]
 [0.30651885]
 [0.6664807 ]
 [0.5653224 ]
 [0.4343679 ]
 [0.22526535]
 [0.02123069]
 [0.30861503]
 [0.5786014 ]
```

```
 [0.03185887]
 [0.6512602 ]
 [0.28757823]
 [0.25170743]
 [0.01969649]
 [0.0945197 ]
 [0.31148547]
 [0.34909374]
 [0.20962831]
 [0.72060937]
 [0.372393  ]
 [0.27539366]
 [0.6601399 ]
 [0.30414814]
 [0.35094988]
 [0.53188944]
 [0.3375162 ]
 [0.09956246]
 [0.8152439 ]
 [0.3135808 ]
 [0.30726492]
 [0.27747172]
 [0.15849066]
 [0.4294235 ]
 [0.0215518 ]
 [0.27432662]
 [0.5500302 ]
 [0.0523959 ]
 [0.36665875]
 [0.26164982]
 [0.30970693]
 [0.31590998]
 [0.70840305]
 [0.00617669]
 [0.00684357]
 [0.19998899]
 [0.00916039]
 [0.0253617 ]
 [0.70189   ]
 [0.31181234]
 [0.07718633]
 [0.00755509]
 [0.3064909 ]
 [0.21073554]
 [0.00572741]
 [0.02205363]
 [0.17776887]
 [0.7753012 ]
 [0.2874418 ]
```

```
[0.5187942 ]
[0.02712148]
[0.28315917]
[0.09479076]
[0.27630544]
[0.1997179 ]
[0.01686994]
[0.30034655]
[0.31965798]
[0.309771  ]
[0.44015557]
[0.6026523 ]
[0.3564375 ]
[0.31818426]
[0.11864345]
[0.2936504 ]
[0.29904056]
[0.00575609]
[0.00774361]
[0.33679897]
[0.17650671]
[0.32909834]
[0.6579774 ]
[0.27832985]
[0.22860761]
[0.35189533]
[0.22311318]
[0.2078653 ]
[0.21907601]
[0.16349119]
[0.2998764 ]
[0.02161698]
[0.02344961]
[0.5331016 ]
[0.25818217]
[0.02813052]
[0.08886237]
[0.23847723]
[0.23802254]
[0.09479801]
[0.02323562]
[0.04964801]
[0.2418421 ]
[0.79971814]
[0.00577222]
[0.01096496]
[0.35648507]
[0.56481385]
[0.42442638]
```

```
 [0.2683465 ]
 [0.2687658 ]
 [0.10545537]
 [0.37244582]
 [0.11151705]
 [0.29468584]
 [0.01156518]
 [0.2859497 ]
 [0.1927885 ]
 [0.21932903]
 [0.18489511]
 [0.34618396]
 [0.29445404]
 [0.4319532 ]
 [0.02785247]
 [0.28479552]
 [0.21743947]
 [0.02308925]
 [0.20228583]
 [0.02375237]
 [0.23897617]
 [0.22234723]
 [0.39277345]
 [0.161078  ]
 [0.2424594 ]
 [0.38867623]
 [0.2053892 ]
 [0.60903376]
 [0.28825212]
 [0.22735463]
 [0.32315052]
 [0.20187753]
 [0.26328737]
 [0.00579423]
 [0.27320498]
 [0.39827764]
 [0.61465627]
 [0.18521766]
 [0.4318599 ]
 [0.05517704]
 [0.2958545 ]
 [0.6251953 ]
 [0.8163874 ]
 [0.01119143]
 [0.29057765]
 [0.21707633]
 [0.3737257 ]
 [0.36021918]
 [0.6496166 ]
```

```
 [0.53734165]
 [0.35239542]
 [0.02106236]
 [0.08345599]
 [0.4726696 ]
 [0.00589791]
 [0.24275114]
 [0.33654976]
 [0.00739701]
 [0.31181532]
 [0.5406317 ]
 [0.6826127 ]
 [0.19619963]
 [0.22511075]
 [0.5026187 ]
 [0.34145474]
 [0.01748915]
 [0.00849306]
 [0.53221446]
 [0.28447053]
 [0.03219065]
 [0.02659572]
 [0.8256638 ]
 [0.01144822]
 [0.27302092]
 [0.08317459]
 [0.65121835]
 [0.02334957]
 [0.03021534]
 [0.4176175 ]
 [0.03627409]
 [0.28015125]
 [0.2210965 ]
 [0.03988322]
 [0.07688487]
 [0.00684224]
 [0.03193456]
 [0.35374528]
 [0.27377883]
 [0.44915116]
 [0.28648856]
 [0.3259477 ]
 [0.48427874]
 [0.20593601]
 [0.02462937]
 [0.26758415]
 [0.34349078]
 [0.2805517 ]
 [0.30230373]
```

```
 [0.02854202]
 [0.7780965 ]
 [0.27138698]
 [0.34994936]
 [0.31183136]
 [0.02835027]
 [0.6080187 ]
 [0.2368238 ]
 [0.18796591]
 [0.35807234]
 [0.38804924]
 [0.22835137]
 [0.28921413]
 [0.57763565]
 [0.11678614]
 [0.02376769]
 [0.00648754]
 [0.77343774]
 [0.30292946]
 [0.00585499]
 [0.18162477]
 [0.30777955]
 [0.32353717]
 [0.2857371 ]
 [0.24474336]
 [0.3249426 ]
 [0.29995668]
 [0.28421515]
 [0.2303116 ]
 [0.05314554]
 [0.30178976]
 [0.61454505]
 [0.2885006 ]
 [0.09314805]
 [0.4141239 ]
 [0.5763136 ]
 [0.23830105]
 [0.11271156]
 [0.3420818 ]
 [0.3568176 ]
 [0.11222443]
 [0.6002505 ]
 [0.51022184]
 [0.30112803]
 [0.02751153]
 [0.24667343]
 [0.17289385]
 [0.55591595]
 [0.31054306]
```

```
 [0.14703777]
 [0.21958718]
 [0.56866574]
 [0.21159707]
 [0.2500006 ]
 [0.05419482]
 [0.02811743]
 [0.1987357 ]
 [0.34364492]
 [1.0027865 ]
 [0.02089276]
 [0.01520547]
 [0.25899315]
 [0.04389047]
 [0.00693532]
 [0.00890225]
 [0.24655281]
 [0.01102273]
 [0.64192903]
 [0.1369681 ]
 [0.07153174]
 [0.36030817]
 [0.23983863]
 [0.3571722 ]
 [0.22463995]
 [0.12681922]
 [0.02606482]
 [0.00727895]
 [0.6591463 ]
 [0.26591313]
 [0.01134688]
 [0.33235472]
 [0.22641782]
 [0.29168802]
 [0.2634772 ]
 [0.66270125]
 [0.26385635]
 [0.03567922]
 [0.1656492 ]
 [0.1973339 ]
 [0.56388026]
 [0.2930007 ]
 [0.3331645 ]
 [0.09859293]
 [0.33411896]
 [0.24580589]
 [0.41285336]
 [0.16382544]
 [0.33393592]
```

```
[0.03907522]
[0.00636396]
[0.02291998]
[0.32010323]
[0.02040261]
[0.50608146]
[0.11296721]
[0.16234387]
[0.36522692]
[0.00637669]
[0.37724203]
[0.3290677 ]
[0.350671  ]
[0.5804529 ]
[0.19815661]
[0.62244755]
[0.24993357]
[0.2876492 ]
[0.7239246 ]
[0.23300448]
[0.3856073 ]
[0.27272546]
[0.3217963 ]
[0.16478108]
[0.2952382 ]
[0.07997853]
[0.39292562]
[0.02013711]
[0.5267645 ]
[0.16398968]
[0.07068156]
[0.2289604 ]
[0.29196703]
[0.3372934 ]
[0.2549526 ]
[0.0074126 ]
[0.3720641 ]
[0.07700454]
[0.00820796]
[0.6011271 ]
[0.02808502]
[0.43253398]
[0.23756862]
[0.22777681]
[0.30016285]
[0.00561261]
[0.28716123]
[0.22343363]
[0.39605707]
[0.19628496]
```

```
 [0.19201046]
 [0.32260537]
 [0.29584247]
 [0.21938412]
 [0.11482549]
 [0.22044767]
 [0.09545452]
 [0.26426917]
 [0.28142288]
 [0.17397512]
 [0.2958098 ]
 [0.02769155]
 [0.20457074]
 [0.30849236]
 [0.2526433 ]
 [0.5834621 ]
 [0.16848794]
 [0.04419667]
 [0.39983892]
 [0.14210522]
 [0.30957955]
 [0.01680957]
 [0.3360675 ]
 [0.32098818]
 [0.31069708]
 [0.64584404]
 [0.35462463]
 [0.2942692 ]
 [0.21860106]
 [0.23586081]
 [0.26363415]
 [0.08333868]
 [0.2992487 ]
 [0.12330579]
 [0.3291899 ]
 [0.38353825]
 [0.6766922 ]
 [0.2221815 ]
 [0.24971752]
 [0.33356482]
 [0.21373612]
 [0.54090244]
 [0.10876717]
 [0.18840127]
 [0.31479138]
 [0.32735002]
 [0.01096168]
 [0.2923749 ]
 [0.45867908]
```

```
 [0.332116  ]
 [0.09480778]
 [0.33268493]
 [0.09564608]
 [0.21192582]
 [0.01331379]
 [0.00565259]
 [0.22801445]
 [0.3514886 ]
 [0.5939934 ]
 [0.2590836 ]
 [0.1073412 ]
 [0.07822425]
 [0.06211027]
 [0.21281514]
 [0.06733958]
 [0.83079755]
 [0.42654186]
 [0.7117811 ]
 [0.01952223]
 [0.02920026]
 [0.39197165]
 [0.24399766]
 [0.01719476]
 [0.08368947]
 [0.21355039]
 [0.48638272]
 [0.2275803 ]
 [0.19243197]
 [0.0821106 ]
 [0.37113637]
 [0.40762258]
 [0.69331604]
 [0.15842867]
 [0.38438052]
 [0.22707719]
 [0.52573323]
 [0.34369278]
 [0.18428148]
 [0.2858209 ]
 [0.3672365 ]
 [0.24035044]
 [0.5041681 ]
 [0.3499903 ]
 [0.7060912 ]
 [0.33367562]
 [0.21941781]
 [0.5773533 ]
 [0.29252553]
```

```
 [0.53172946]
 [0.31603742]
 [0.58675665]
 [0.0548683 ]
 [0.01167235]
 [0.14081109]
 [0.2480684 ]
 [0.0921912 ]
 [0.2236009 ]
 [0.35695857]
 [0.5037882 ]
 [0.33278942]
 [0.20422894]
 [0.36295635]
 [0.11767267]
 [0.07625259]
 [0.12624569]
 [0.22800592]
 [0.54989046]
 [0.35660928]
 [0.55215025]
 [0.54878193]
 [0.6759507 ]
 [0.23997462]
 [0.27246058]
 [0.36546463]
 [0.30761844]
 [0.01089287]
 [0.05419075]
 [0.3882084 ]
 [0.24965446]
 [0.24817963]
 [0.03448752]
 [0.01048338]
 [0.17170016]
 [0.11288168]
 [0.01341261]
 [0.30461746]
 [0.11288147]
 [0.27424535]
 [0.09595844]
 [0.6568641 ]
 [0.34319174]
 [0.22377633]
 [0.35032386]
 [0.02382595]
 [0.6104346 ]
 [0.34634882]
 [0.39849287]
```

```
[0.02375589]
[0.05439846]
[0.02627625]
[0.15709083]
[0.23869844]
[0.24931073]
[0.22226581]
[0.23945878]
[0.19591598]
[0.24511021]
[0.06657118]
[0.696073  ]
[0.00586648]
[0.07613945]
[0.40221715]
[0.2440707 ]
[0.30420393]
[0.580908  ]
[0.24048859]
[0.09532064]
[0.23783477]
[0.25404975]
[0.28227118]
[0.38564175]
[0.22830723]
[0.23069212]
[0.01452719]
[0.59236795]
[0.19735987]
[0.22211532]
[0.01481622]
[0.21989733]
[0.42792225]
[0.40022224]
[0.2662571 ]
[0.33152568]
[0.7187822 ]
[0.2916227 ]
[0.70827293]
[0.30563474]
[0.18628931]
[0.82985383]
[0.57192945]
[0.67226046]
[0.00847254]
[0.2741363 ]
[0.0708601 ]
[0.33725494]
[0.00808233]
```

```
 [0.1837208 ]
 [0.45308638]
 [0.32644838]
 [0.21305838]
 [0.3354522 ]
 [0.25284326]
 [0.03009237]
 [0.0056347 ]
 [0.35580522]
 [0.41204882]
 [0.16552238]
 [0.2251598 ]
 [0.27647156]
 [0.8369106 ]
 [0.2969334 ]
 [0.28123057]
 [0.20674667]
 [0.2988112 ]
 [0.0066633 ]
 [0.07358755]
 [0.2955715 ]
 [0.4013651 ]
 [0.02318408]
 [0.23846568]
 [0.00930729]
 [0.09559461]
 [0.23272751]
 [0.02917055]
 [0.14025776]
 [0.02610333]
 [0.00841342]
 [0.21886736]
 [0.00758416]
 [0.3707705 ]
 [0.01071058]
 [0.37546688]
 [0.27115926]
 [0.07564224]
 [0.37250566]
 [0.00727576]
 [0.01336871]
 [0.06833091]
 [0.00566799]]

# Mengubah hasil prediksi kembali ke skala semula
y_pred_inverse_csco = scaler.inverse_transform(y_pred_csco)
y_test_inverse_csco = scaler.inverse_transform(y_test_csco)

# Menampilkan hasil prediksi dalam skala semula
```

```python
print("Predicted Values (Inverse Transform):")
print(y_pred_inverse_csco)
```

```
Predicted Values (Inverse Transform):
[[ 4.5392923 ]
 [19.612232  ]
 [ 1.6075987 ]
 [56.212357  ]
 [16.535362  ]
 [25.18624   ]
 [57.059628  ]
 [43.527206  ]
 [66.66789   ]
 [32.778744  ]
 [50.084816  ]
 [64.835464  ]
 [24.051329  ]
 [55.656353  ]
 [32.13056   ]
 [ 1.9238974 ]
 [16.024616  ]
 [26.71581   ]
 [ 2.1882975 ]
 [13.659764  ]
 [30.695257  ]
 [24.518927  ]
 [20.627495  ]
 [19.634872  ]
 [ 5.124079  ]
 [ 1.0716325 ]
 [63.887264  ]
 [ 0.74772596]
 [24.542679  ]
 [ 6.4228168 ]
 [ 0.6147143 ]
 [25.648468  ]
 [19.966585  ]
 [32.227547  ]
 [ 0.519873  ]
 [ 0.5269135 ]
 [29.869732  ]
 [26.036825  ]
 [22.205114  ]
 [17.42673   ]
 [ 4.2139745 ]
 [21.147316  ]
 [28.267542  ]
 [28.475058  ]
 [ 0.9393319 ]
 [29.423622  ]
```

```
[28.993595  ]
[ 0.53386146]
[ 0.51701516]
[30.476202  ]
[69.61155   ]
[ 0.5164455 ]
[47.713646  ]
[ 0.64050364]
[50.245556  ]
[28.271624  ]
[ 2.3706844 ]
[29.058828  ]
[65.592964  ]
[ 0.5425726 ]
[28.111685  ]
[24.361015  ]
[ 1.8171105 ]
[15.91309   ]
[26.601921  ]
[ 3.63304   ]
[24.183636  ]
[37.352036  ]
[ 2.3819168 ]
[ 1.8672054 ]
[27.444248  ]
[ 0.81321734]
[15.792384  ]
[ 1.6895047 ]
[ 9.211128  ]
[28.58235   ]
[ 0.89435565]
[ 1.0008788 ]
[23.51012   ]
[23.053392  ]
[14.820844  ]
[ 6.910267  ]
[ 1.784772  ]
[15.175451  ]
[21.463266  ]
[ 1.9029212 ]
[ 1.2789366 ]
[48.907455  ]
[19.387028  ]
[ 0.52721316]
[ 3.4976013 ]
[ 2.4005365 ]
[23.445295  ]
[21.99586   ]
[33.02633   ]
```

```
 [ 1.0846243 ]
 [ 8.042707  ]
 [27.526932  ]
 [ 0.6661025 ]
 [ 0.9995211 ]
 [ 6.8358474 ]
 [52.66198   ]
 [16.706354  ]
 [ 4.2405233 ]
 [13.547585  ]
 [28.644436  ]
 [19.594542  ]
 [ 0.53891915]
 [17.889744  ]
 [12.268411  ]
 [36.415367  ]
 [25.899473  ]
 [ 2.3530762 ]
 [19.368046  ]
 [17.671404  ]
 [24.940619  ]
 [15.223659  ]
 [32.589394  ]
 [ 0.54424167]
 [19.143986  ]
 [31.792253  ]
 [34.325287  ]
 [17.649384  ]
 [ 0.9702183 ]
 [23.989403  ]
 [26.767813  ]
 [23.347378  ]
 [21.270355  ]
 [26.470972  ]
 [ 2.1816387 ]
 [27.538113  ]
 [19.998611  ]
 [25.02809   ]
 [23.180756  ]
 [10.1255455 ]
 [18.129787  ]
 [ 9.180917  ]
 [ 1.929491  ]
 [18.27669   ]
 [27.967945  ]
 [21.858559  ]
 [18.32868   ]
 [19.004692  ]
 [30.88773   ]
```

```
 [21.112797  ]
 [24.452133  ]
 [44.76398   ]
 [19.224445  ]
 [15.83961   ]
 [26.069872  ]
 [12.950745  ]
 [19.826164  ]
 [ 8.700996  ]
 [57.40586   ]
 [18.32904   ]
 [43.245125  ]
 [25.32167   ]
 [45.330925  ]
 [28.734873  ]
 [12.550031  ]
 [ 0.68024635]
 [47.396656  ]
 [26.894428  ]
 [48.089653  ]
 [46.706043  ]
 [26.176132  ]
 [24.236492  ]
 [21.21896   ]
 [24.590029  ]
 [53.38385   ]
 [45.292065  ]
 [34.81684   ]
 [18.090454  ]
 [ 1.7694514 ]
 [24.757706  ]
 [46.35427   ]
 [ 2.6196134 ]
 [52.16634   ]
 [23.074944  ]
 [20.205591  ]
 [ 1.6467291 ]
 [ 7.6319366 ]
 [24.987316  ]
 [27.99565   ]
 [16.839626  ]
 [57.713673  ]
 [29.85939   ]
 [22.100285  ]
 [52.876644  ]
 [24.400393  ]
 [28.144125  ]
 [42.617718  ]
 [27.069548  ]
```

```
 [ 8.035313  ]
 [65.283615  ]
 [25.154924  ]
 [24.649708  ]
 [22.26651   ]
 [12.749058  ]
 [34.421333  ]
 [ 1.7951372 ]
 [22.01493   ]
 [44.068817  ]
 [ 4.262398  ]
 [29.400698  ]
 [21.000896  ]
 [24.845047  ]
 [25.341238  ]
 [56.737274  ]
 [ 0.56526226]
 [ 0.6186068 ]
 [16.068563  ]
 [ 0.8039324 ]
 [ 2.0998962 ]
 [56.216286  ]
 [25.013462  ]
 [ 6.245417  ]
 [ 0.67552215]
 [24.587793  ]
 [16.928194  ]
 [ 0.529324  ]
 [ 1.8352797 ]
 [14.291147  ]
 [62.088547  ]
 [23.06403   ]
 [41.57021   ]
 [ 2.2406638 ]
 [22.721457  ]
 [ 7.6536183 ]
 [22.173218  ]
 [16.046879  ]
 [ 1.4206291 ]
 [24.096298  ]
 [25.641045  ]
 [24.850172  ]
 [35.279808  ]
 [48.278133  ]
 [28.583088  ]
 [25.52316   ]
 [ 9.561626  ]
 [23.560663  ]
 [23.99183   ]
```

```
 [ 0.5316176 ]
 [ 0.6906021 ]
 [27.012175  ]
 [14.190186  ]
 [26.396193  ]
 [52.70366   ]
 [22.335154  ]
 [18.357805  ]
 [28.219753  ]
 [17.918299  ]
 [16.6986    ]
 [17.59536   ]
 [13.149056  ]
 [24.05869   ]
 [ 1.800351  ]
 [ 1.9469458 ]
 [42.71468   ]
 [20.723515  ]
 [ 2.3213778 ]
 [ 7.179399  ]
 [19.14729   ]
 [19.110918  ]
 [ 7.6541986 ]
 [ 1.929828  ]
 [ 4.04259   ]
 [19.41645   ]
 [64.04169   ]
 [ 0.5329079 ]
 [ 0.9482822 ]
 [28.586893  ]
 [45.251385  ]
 [34.021606  ]
 [21.536572  ]
 [21.570114  ]
 [ 8.506695  ]
 [29.863615  ]
 [ 8.991576  ]
 [23.643492  ]
 [ 0.99629444]
 [22.944675  ]
 [15.492587  ]
 [17.615599  ]
 [14.861184  ]
 [27.762894  ]
 [23.624949  ]
 [34.623688  ]
 [ 2.2991362 ]
 [22.852352  ]
 [17.46445   ]
```

```
 [ 1.9181198 ]
 [16.25229   ]
 [ 1.9711641 ]
 [19.187199  ]
 [17.857029  ]
 [31.489649  ]
 [12.956022  ]
 [19.465828  ]
 [31.161905  ]
 [16.500534  ]
 [48.788593  ]
 [23.128849  ]
 [18.257578  ]
 [25.920418  ]
 [16.219631  ]
 [21.131886  ]
 [ 0.53466874]
 [21.925209  ]
 [31.929935  ]
 [49.238346  ]
 [14.886986  ]
 [34.616226  ]
 [ 4.4848647 ]
 [23.736975  ]
 [50.08138   ]
 [65.375084  ]
 [ 0.96639776]
 [23.31487   ]
 [17.435402  ]
 [29.965994  ]
 [28.88559   ]
 [52.034866  ]
 [43.05385   ]
 [28.259756  ]
 [ 1.7559868 ]
 [ 6.746935  ]
 [37.880646  ]
 [ 0.54296213]
 [19.489164  ]
 [26.99224   ]
 [ 0.6628774 ]
 [25.0137    ]
 [43.317024  ]
 [54.67427   ]
 [15.765448  ]
 [18.078087  ]
 [40.27631   ]
 [27.384598  ]
 [ 1.4701608 ]
```

```
[ 0.75055164]
[42.643715  ]
[22.826355  ]
[ 2.6461527 ]
[ 2.1986072 ]
[66.11712   ]
[ 0.986939  ]
[21.910486  ]
[ 6.7244253 ]
[52.162994  ]
[ 1.9389437 ]
[ 2.4881454 ]
[33.476955  ]
[ 2.9727926 ]
[22.48085   ]
[17.75698   ]
[ 3.2614918 ]
[ 6.2213025 ]
[ 0.6185    ]
[ 2.625668  ]
[28.367733  ]
[21.971111  ]
[35.999374  ]
[22.98778   ]
[26.144169  ]
[38.809277  ]
[16.544273  ]
[ 2.0413163 ]
[21.47559   ]
[27.547462  ]
[22.512882  ]
[24.252855  ]
[ 2.3542943 ]
[62.312145  ]
[21.779785  ]
[28.064093  ]
[25.014984  ]
[ 2.3389559 ]
[48.707397  ]
[19.015028  ]
[15.106822  ]
[28.713861  ]
[31.111753  ]
[18.337307  ]
[23.205801  ]
[46.277016  ]
[ 9.413058  ]
[ 1.972389  ]
[ 0.59012735]
```

```
[61.939484  ]
[24.30291   ]
[ 0.53952926]
[14.599586  ]
[24.690874  ]
[25.951347  ]
[22.92767   ]
[19.648525  ]
[26.063768  ]
[24.065113  ]
[22.805927  ]
[18.49411   ]
[ 4.322362  ]
[24.211742  ]
[49.22945   ]
[23.148726  ]
[ 7.5222163 ]
[33.1975    ]
[46.171265  ]
[19.133196  ]
[ 9.087127  ]
[27.434755  ]
[28.613493  ]
[ 9.048161  ]
[48.086006  ]
[40.8845    ]
[24.15881   ]
[ 2.2718637 ]
[19.802914  ]
[13.901188  ]
[44.53963   ]
[24.91193   ]
[11.832926  ]
[17.636248  ]
[45.5595    ]
[16.997108  ]
[20.06906   ]
[ 4.406296  ]
[ 2.320331  ]
[15.968311  ]
[27.559792  ]
[80.2854    ]
[ 1.7424202 ]
[ 1.2874863 ]
[20.788385  ]
[ 3.582037  ]
[ 0.62594575]
[ 0.783283  ]
[19.793264  ]
```

```
 [ 0.9529034 ]
 [51.41993   ]
 [11.02744   ]
 [ 5.793099  ]
 [28.892708  ]
 [19.25619   ]
 [28.641857  ]
 [18.040426  ]
 [10.215617  ]
 [ 2.1561399 ]
 [ 0.65343314]
 [52.79716   ]
 [21.341925  ]
 [ 0.9788325 ]
 [26.656675  ]
 [18.18264   ]
 [23.403692  ]
 [21.147072  ]
 [53.081528  ]
 [21.177399  ]
 [ 2.9252088 ]
 [13.321679  ]
 [15.856179  ]
 [45.176704  ]
 [23.508694  ]
 [26.72145   ]
 [ 7.957759  ]
 [26.797798  ]
 [19.733519  ]
 [33.095867  ]
 [13.175794  ]
 [26.783155  ]
 [ 3.1968586 ]
 [ 0.5802424 ]
 [ 1.9045802 ]
 [25.67666   ]
 [ 1.7032119 ]
 [40.553303  ]
 [ 9.107576  ]
 [13.0572815 ]
 [29.286165  ]
 [ 0.58126   ]
 [30.247269  ]
 [26.393742  ]
 [28.121817  ]
 [46.502373  ]
 [15.921989  ]
 [49.86158   ]
 [20.063698  ]
```

```
 [23.080622  ]
 [57.978863  ]
 [18.709517  ]
 [30.916418  ]
 [21.88685   ]
 [25.812092  ]
 [13.252236  ]
 [23.687675  ]
 [ 6.468768  ]
 [31.50182   ]
 [ 1.681974  ]
 [42.207767  ]
 [13.188931  ]
 [ 5.7250915 ]
 [18.386024  ]
 [23.42601   ]
 [27.051725  ]
 [20.465178  ]
 [ 0.6641243 ]
 [29.83308   ]
 [ 6.230875  ]
 [ 0.72774595]
 [48.156128  ]
 [ 2.3177383 ]
 [34.670147  ]
 [19.074608  ]
 [18.291348  ]
 [24.081604  ]
 [ 0.52014035]
 [23.041588  ]
 [17.943932  ]
 [31.75231   ]
 [15.772274  ]
 [15.43035   ]
 [25.87681   ]
 [23.736012  ]
 [17.620007  ]
 [ 9.256223  ]
 [17.70508   ]
 [ 7.7067137 ]
 [21.21042   ]
 [22.58257   ]
 [13.98768   ]
 [23.733398  ]
 [ 2.2862642 ]
 [16.435064  ]
 [24.747892  ]
 [20.28045   ]
 [46.743084  ]
 [13.548753  ]
```

```
[ 3.6065304 ]
[32.054825  ]
[11.438365  ]
[24.834858  ]
[ 1.4158003 ]
[26.953665  ]
[25.74745   ]
[24.924252  ]
[51.733097  ]
[28.438074  ]
[23.610163  ]
[17.557367  ]
[18.937998  ]
[21.159624  ]
[ 6.737551  ]
[24.00848   ]
[ 9.934573  ]
[26.403517  ]
[30.750912  ]
[54.20068   ]
[17.843771  ]
[20.046413  ]
[26.753471  ]
[17.168215  ]
[43.33868   ]
[ 8.77161   ]
[15.141646  ]
[25.25176   ]
[26.256342  ]
[ 0.9480195 ]
[23.458637  ]
[36.761528  ]
[26.637579  ]
[ 7.65498   ]
[26.683088  ]
[ 7.722037  ]
[17.023407  ]
[ 1.136168  ]
[ 0.523339  ]
[18.310358  ]
[28.187218  ]
[47.5855    ]
[20.79562   ]
[ 8.657545  ]
[ 6.3284416 ]
[ 5.0394626 ]
[17.094543  ]
[ 5.4577627 ]
[66.52777   ]
```

```
 [34.190826  ]
 [57.00749   ]
 [ 1.6327895 ]
 [ 2.4069479 ]
 [31.42551   ]
 [19.588875  ]
 [ 1.4466122 ]
 [ 6.765611  ]
 [17.153358  ]
 [38.977577  ]
 [18.275629  ]
 [15.464067  ]
 [ 6.6393156 ]
 [29.75887   ]
 [32.67745   ]
 [55.530445  ]
 [12.744099  ]
 [30.818287  ]
 [18.235384  ]
 [42.125275  ]
 [27.563622  ]
 [14.812099  ]
 [22.934372  ]
 [29.446913  ]
 [19.297129  ]
 [40.40025   ]
 [28.067368  ]
 [56.55235   ]
 [26.762335  ]
 [17.6227    ]
 [46.254433  ]
 [23.470684  ]
 [42.60492   ]
 [25.35143   ]
 [47.00662   ]
 [ 4.4601684 ]
 [ 1.0048672 ]
 [11.334845  ]
 [19.9145    ]
 [ 7.445677  ]
 [17.957312  ]
 [28.62477   ]
 [40.36986   ]
 [26.691446  ]
 [16.407722  ]
 [29.104538  ]
 [ 9.483973  ]
 [ 6.1707253 ]
 [10.16974   ]
```

```
 [18.309675  ]
 [44.057644  ]
 [28.596828  ]
 [44.238407  ]
 [43.96897   ]
 [54.14137   ]
 [19.267067  ]
 [21.865664  ]
 [29.30518   ]
 [24.677986  ]
 [ 0.9425156 ]
 [ 4.40597   ]
 [31.124483  ]
 [20.04137   ]
 [19.923397  ]
 [ 2.8298826 ]
 [ 0.90976   ]
 [13.805703  ]
 [ 9.100735  ]
 [ 1.144073  ]
 [24.437935  ]
 [ 9.100718  ]
 [22.008429  ]
 [ 7.7470226 ]
 [52.614605  ]
 [27.523542  ]
 [17.971344  ]
 [28.09405   ]
 [ 1.9770497 ]
 [48.900646  ]
 [27.776081  ]
 [31.947153  ]
 [ 1.9714457 ]
 [ 4.4225855 ]
 [ 2.1730525 ]
 [12.637083  ]
 [19.164984  ]
 [20.013876  ]
 [17.850515  ]
 [19.225805  ]
 [15.742759  ]
 [19.67787   ]
 [ 5.3962965 ]
 [55.750977  ]
 [ 0.54044825]
 [ 6.1616755 ]
 [32.245064  ]
 [19.594717  ]
 [24.404856  ]
```

```
 [46.538776  ]
 [19.30818   ]
 [ 7.6960044 ]
 [19.095898  ]
 [20.392956  ]
 [22.650425  ]
 [30.919174  ]
 [18.333776  ]
 [18.524548  ]
 [ 1.2332299 ]
 [47.455475  ]
 [15.858257  ]
 [17.838478  ]
 [ 1.2563492 ]
 [17.661058  ]
 [34.301247  ]
 [32.085487  ]
 [21.369438  ]
 [26.590359  ]
 [57.567516  ]
 [23.398466  ]
 [56.726868  ]
 [24.519308  ]
 [14.972709  ]
 [66.452286  ]
 [45.82057   ]
 [53.84618   ]
 [ 0.7489102 ]
 [21.999706  ]
 [ 5.7393737 ]
 [27.048649  ]
 [ 0.71769655]
 [14.767249  ]
 [36.31416   ]
 [26.184217  ]
 [17.114002  ]
 [26.904446  ]
 [20.296448  ]
 [ 2.478309  ]
 [ 0.52190804]
 [28.53251   ]
 [33.03151   ]
 [13.311534  ]
 [18.08201   ]
 [22.186506  ]
 [67.01676   ]
 [23.823277  ]
 [22.567186  ]
 [16.60912   ]
```

```
 [23.973484  ]
 [ 0.604187  ]
 [ 5.9575458 ]
 [23.714336  ]
 [32.176907  ]
 [ 1.9257058 ]
 [19.146364  ]
 [ 0.81568277]
 [ 7.7179193 ]
 [18.68736   ]
 [ 2.404571  ]
 [11.290584  ]
 [ 2.1592202 ]
 [ 0.7441815 ]
 [17.578669  ]
 [ 0.6778477 ]
 [29.729605  ]
 [ 0.9279343 ]
 [30.105272  ]
 [21.761568  ]
 [ 6.1219034 ]
 [29.8684    ]
 [ 0.6531785 ]
 [ 1.140561  ]
 [ 5.5370603 ]
 [ 0.5245703 ]]
```

```python
# Menampilkan nilai Close yang sebenarnya (skala semula)
print("Actual Values:")
print(y_test_inverse_csco)
```

```
Actual Values:
[[4.29166651e+00]
 [1.98899994e+01]
 [1.18055558e+00]
 [5.55800018e+01]
 [1.59899998e+01]
 [2.42999992e+01]
 [5.65299988e+01]
 [4.22900009e+01]
 [6.86250000e+01]
 [3.27000008e+01]
 [4.99687500e+01]
 [6.27500000e+01]
 [2.13700008e+01]
 [5.62500000e+01]
 [3.15100002e+01]
 [1.43055558e+00]
 [1.63500004e+01]
 [2.61299992e+01]
```

```
[1.75000000e+00]
[1.15468750e+01]
[3.01250000e+01]
[2.37199993e+01]
[2.08600006e+01]
[1.95400009e+01]
[4.69444466e+00]
[5.97222209e-01]
[6.52500000e+01]
[3.07291657e-01]
[2.37500000e+01]
[5.97222233e+00]
[1.77083328e-01]
[2.47656250e+01]
[1.97299995e+01]
[3.16599998e+01]
[8.07291642e-02]
[9.02777761e-02]
[2.99400005e+01]
[2.65000000e+01]
[2.24099998e+01]
[1.69899998e+01]
[3.87500000e+00]
[2.17900009e+01]
[2.88125000e+01]
[2.82299995e+01]
[4.73958343e-01]
[2.97700005e+01]
[2.88899994e+01]
[1.05034724e-01]
[7.72569478e-02]
[2.92999992e+01]
[6.60625000e+01]
[7.98611119e-02]
[4.70499992e+01]
[2.09201396e-01]
[4.94099998e+01]
[2.76718750e+01]
[1.97916663e+00]
[2.87199993e+01]
[6.38750000e+01]
[1.08506948e-01]
[2.81499996e+01]
[2.43500004e+01]
[1.52083337e+00]
[1.58199997e+01]
[2.67600002e+01]
[3.09722233e+00]
[2.37700005e+01]
```

```
 [3.55999985e+01]
 [1.92361116e+00]
 [1.37847221e+00]
 [2.72999992e+01]
 [3.75000000e-01]
 [1.58800001e+01]
 [1.27951384e+00]
 [8.38194466e+00]
 [2.84099998e+01]
 [4.65277791e-01]
 [5.55555582e-01]
 [2.27000008e+01]
 [2.31200008e+01]
 [1.36499996e+01]
 [6.50000000e+00]
 [1.38194442e+00]
 [1.47700005e+01]
 [2.04899998e+01]
 [1.37500000e+00]
 [8.50694418e-01]
 [4.92099991e+01]
 [1.91399994e+01]
 [8.94097239e-02]
 [2.90277767e+00]
 [1.95138884e+00]
 [2.36000004e+01]
 [2.07500000e+01]
 [3.40800018e+01]
 [6.52777791e-01]
 [7.58333349e+00]
 [2.80900002e+01]
 [2.37847224e-01]
 [5.31250000e-01]
 [6.52777767e+00]
 [5.31599998e+01]
 [1.64899998e+01]
 [3.57986116e+00]
 [1.30416670e+01]
 [2.75400009e+01]
 [1.93400002e+01]
 [1.08506948e-01]
 [1.73999996e+01]
 [1.21800003e+01]
 [3.77099991e+01]
 [2.59599991e+01]
 [1.95833337e+00]
 [1.88199997e+01]
 [1.76000004e+01]
 [2.49500008e+01]
```

```
 [1.51000004e+01]
 [3.31875000e+01]
 [1.17621526e-01]
 [1.83099995e+01]
 [3.17999992e+01]
 [3.39500008e+01]
 [1.73500004e+01]
 [5.03472209e-01]
 [2.33600006e+01]
 [2.67800007e+01]
 [2.28600006e+01]
 [2.08700008e+01]
 [2.63199997e+01]
 [1.75000000e+00]
 [2.73906250e+01]
 [2.08299999e+01]
 [2.44799995e+01]
 [2.34799995e+01]
 [9.18750000e+00]
 [1.72399998e+01]
 [8.48958302e+00]
 [1.44791663e+00]
 [1.79400005e+01]
 [2.78099995e+01]
 [2.19899998e+01]
 [1.85799999e+01]
 [1.86100006e+01]
 [3.11800003e+01]
 [2.08999996e+01]
 [2.40100002e+01]
 [4.45600014e+01]
 [1.90599995e+01]
 [1.60100002e+01]
 [2.57199993e+01]
 [1.21599998e+01]
 [1.88437500e+01]
 [8.18055534e+00]
 [5.79500008e+01]
 [1.85499992e+01]
 [4.21699982e+01]
 [2.50599995e+01]
 [4.40600014e+01]
 [2.96875000e+01]
 [1.24895830e+01]
 [2.48263896e-01]
 [4.84500008e+01]
 [2.60400009e+01]
 [4.80099983e+01]
 [4.60499992e+01]
```

```
 [2.59699993e+01]
 [2.38600006e+01]
 [2.06200008e+01]
 [2.45799999e+01]
 [5.33600006e+01]
 [4.47799988e+01]
 [3.32812500e+01]
 [1.72000008e+01]
 [1.38194442e+00]
 [2.44200001e+01]
 [4.57000008e+01]
 [2.15972233e+00]
 [5.32299995e+01]
 [2.28600006e+01]
 [1.97199993e+01]
 [1.13541663e+00]
 [7.22222233e+00]
 [2.43999996e+01]
 [2.80100002e+01]
 [1.70000000e+01]
 [5.73600006e+01]
 [2.94899998e+01]
 [2.21100006e+01]
 [5.37812500e+01]
 [2.42500000e+01]
 [2.78500004e+01]
 [4.23400002e+01]
 [2.71399994e+01]
 [7.41666651e+00]
 [6.43125000e+01]
 [2.48799992e+01]
 [2.47399998e+01]
 [2.17600002e+01]
 [1.09687500e+01]
 [3.54687500e+01]
 [1.36111116e+00]
 [2.16800003e+01]
 [4.37500000e+01]
 [3.72569442e+00]
 [2.93600006e+01]
 [2.08500004e+01]
 [2.48500004e+01]
 [2.52199993e+01]
 [5.68800011e+01]
 [1.26736104e-01]
 [1.92708328e-01]
 [1.54099998e+01]
 [3.80208343e-01]
 [1.68055558e+00]
```

```
 [5.41899986e+01]
 [2.54400005e+01]
 [5.86111116e+00]
 [2.30902776e-01]
 [2.43199997e+01]
 [1.75200005e+01]
 [9.37500000e-02]
 [1.36805558e+00]
 [1.36599998e+01]
 [6.03125000e+01]
 [2.32700005e+01]
 [4.14062500e+01]
 [1.75694442e+00]
 [2.20000000e+01]
 [7.30555534e+00]
 [2.18999996e+01]
 [1.59270830e+01]
 [1.02777779e+00]
 [2.40799999e+01]
 [2.39687500e+01]
 [2.51000004e+01]
 [3.35937500e+01]
 [4.64399986e+01]
 [2.86599998e+01]
 [2.54400005e+01]
 [8.50000000e+00]
 [2.34500008e+01]
 [2.45100002e+01]
 [9.20138881e-02]
 [2.60416657e-01]
 [2.68700008e+01]
 [1.37200003e+01]
 [2.58700008e+01]
 [5.30937500e+01]
 [2.20100002e+01]
 [1.83799992e+01]
 [2.77099991e+01]
 [1.84400005e+01]
 [1.62900009e+01]
 [1.71200008e+01]
 [1.22604170e+01]
 [2.38299999e+01]
 [1.35416663e+00]
 [1.52083337e+00]
 [3.93125000e+01]
 [2.07700005e+01]
 [1.82986116e+00]
 [7.02083349e+00]
 [1.89599991e+01]
```

```
 [1.87999992e+01]
 [7.40277767e+00]
 [1.53472221e+00]
 [3.49305558e+00]
 [1.93799992e+01]
 [6.79687500e+01]
 [9.28819478e-02]
 [5.18229187e-01]
 [2.86399994e+01]
 [4.42500000e+01]
 [3.37999992e+01]
 [2.10300007e+01]
 [2.13700008e+01]
 [8.25000000e+00]
 [2.90400009e+01]
 [8.60416698e+00]
 [2.30300007e+01]
 [5.77256918e-01]
 [2.20499992e+01]
 [1.51700001e+01]
 [1.75400009e+01]
 [1.49791670e+01]
 [2.68099995e+01]
 [2.36100006e+01]
 [3.42099991e+01]
 [1.93750000e+00]
 [2.01499996e+01]
 [1.69300003e+01]
 [1.48958337e+00]
 [1.54600000e+01]
 [1.52777779e+00]
 [1.81900005e+01]
 [1.77600002e+01]
 [3.14500008e+01]
 [1.25600004e+01]
 [1.95799999e+01]
 [3.04799995e+01]
 [1.63299999e+01]
 [4.88699989e+01]
 [2.27999992e+01]
 [1.80400009e+01]
 [2.58400002e+01]
 [1.56899996e+01]
 [2.12199993e+01]
 [1.05468750e-01]
 [2.11800003e+01]
 [3.18700008e+01]
 [4.99300003e+01]
 [1.43400002e+01]
```

```
 [3.42900009e+01]
 [3.93402767e+00]
 [2.37500000e+01]
 [5.07900009e+01]
 [6.75000000e+01]
 [5.59027791e-01]
 [2.31499996e+01]
 [1.68199997e+01]
 [2.94300003e+01]
 [2.88199997e+01]
 [4.80625000e+01]
 [4.25699997e+01]
 [2.84699993e+01]
 [1.28125000e+00]
 [6.52777767e+00]
 [3.73100014e+01]
 [1.10243052e-01]
 [1.87600002e+01]
 [2.73999996e+01]
 [2.29166672e-01]
 [2.47999992e+01]
 [4.35800018e+01]
 [5.22500000e+01]
 [1.49300003e+01]
 [1.75000000e+01]
 [4.12299995e+01]
 [2.77900009e+01]
 [1.02430558e+00]
 [3.06423604e-01]
 [4.22099991e+01]
 [2.21700001e+01]
 [2.20833325e+00]
 [1.73697913e+00]
 [6.63750000e+01]
 [5.50347209e-01]
 [2.23400002e+01]
 [6.06944466e+00]
 [5.27200012e+01]
 [1.52083337e+00]
 [2.07638884e+00]
 [3.30800018e+01]
 [2.43055558e+00]
 [2.22299995e+01]
 [1.71700001e+01]
 [2.80902767e+00]
 [5.76388884e+00]
 [1.71875000e-01]
 [2.22222233e+00]
 [2.82999992e+01]
```

```
[2.15699997e+01]
[3.62187500e+01]
[2.28299999e+01]
[2.56800003e+01]
[3.85299988e+01]
[1.67199993e+01]
[1.47569442e+00]
[2.07600002e+01]
[2.77700005e+01]
[2.21200008e+01]
[2.42800007e+01]
[1.94444442e+00]
[6.28750000e+01]
[2.10499992e+01]
[2.81000004e+01]
[2.36200008e+01]
[1.84027779e+00]
[4.83300018e+01]
[2.07600002e+01]
[1.44899998e+01]
[2.72968750e+01]
[3.07199993e+01]
[1.80499992e+01]
[2.31100006e+01]
[4.44900017e+01]
[9.09722233e+00]
[1.61111116e+00]
[1.45833328e-01]
[6.43750000e+01]
[2.41800003e+01]
[9.89583358e-02]
[1.35699997e+01]
[2.47099991e+01]
[2.55200005e+01]
[2.30200005e+01]
[1.97900009e+01]
[2.56299992e+01]
[2.42600002e+01]
[2.31800003e+01]
[1.83099995e+01]
[3.93750000e+00]
[2.39799995e+01]
[4.87999992e+01]
[2.27700005e+01]
[6.48611116e+00]
[3.26199989e+01]
[4.67099991e+01]
[1.85799999e+01]
[8.61805534e+00]
```

```
 [2.73199997e+01]
 [2.83600006e+01]
 [8.77777767e+00]
 [4.84199982e+01]
 [4.00400009e+01]
 [2.39400005e+01]
 [1.78472221e+00]
 [1.92099991e+01]
 [1.30400000e+01]
 [4.43400002e+01]
 [2.40799999e+01]
 [1.17799997e+01]
 [1.72500000e+01]
 [4.53100014e+01]
 [1.63999996e+01]
 [2.01700001e+01]
 [3.96527767e+00]
 [1.88888884e+00]
 [1.53299999e+01]
 [2.69300003e+01]
 [7.78750000e+01]
 [1.09722221e+00]
 [8.92361104e-01]
 [2.03099995e+01]
 [3.25694442e+00]
 [1.92708328e-01]
 [3.69791657e-01]
 [1.97000008e+01]
 [5.06944418e-01]
 [5.19199982e+01]
 [1.07291670e+01]
 [5.38888884e+00]
 [2.94218750e+01]
 [1.97299995e+01]
 [2.83600006e+01]
 [1.78400002e+01]
 [9.07999992e+00]
 [1.68055558e+00]
 [1.96180552e-01]
 [5.30625000e+01]
 [2.15699997e+01]
 [5.17361104e-01]
 [2.63299999e+01]
 [1.84099998e+01]
 [2.29300003e+01]
 [2.10699997e+01]
 [5.30937500e+01]
 [2.04400005e+01]
 [2.52430558e+00]
```

```
[1.31899996e+01]
[1.55900002e+01]
[4.37999992e+01]
[2.23799992e+01]
[2.64300003e+01]
[7.66666651e+00]
[2.59400005e+01]
[1.91599998e+01]
[3.22900009e+01]
[1.31199999e+01]
[2.71399994e+01]
[2.81250000e+00]
[1.42361104e-01]
[1.50000000e+00]
[2.55400009e+01]
[1.29513884e+00]
[3.96800003e+01]
[9.07638931e+00]
[1.30312500e+01]
[2.93600006e+01]
[1.43229172e-01]
[2.97099991e+01]
[2.57000008e+01]
[2.82500000e+01]
[4.77812500e+01]
[1.65625000e+01]
[4.94099998e+01]
[2.05900002e+01]
[2.24699993e+01]
[5.72099991e+01]
[1.80900002e+01]
[3.05599995e+01]
[2.14200001e+01]
[2.55000000e+01]
[1.42343750e+01]
[2.37099991e+01]
[5.75000000e+00]
[3.03400002e+01]
[1.19097221e+00]
[4.17000008e+01]
[1.33984375e+01]
[4.97222233e+00]
[1.72299995e+01]
[2.31900005e+01]
[2.65900002e+01]
[2.03600006e+01]
[2.30902776e-01]
[2.96299992e+01]
[5.77777767e+00]
[3.07291657e-01]
```

```
[4.78499985e+01]
[1.90972221e+00]
[3.42999992e+01]
[1.91000004e+01]
[1.87700005e+01]
[2.38899994e+01]
[7.72569478e-02]
[2.24200001e+01]
[1.77500000e+01]
[3.16000004e+01]
[1.58900003e+01]
[1.55100002e+01]
[2.51399994e+01]
[2.34300003e+01]
[1.71900005e+01]
[9.04166698e+00]
[1.75400009e+01]
[7.33333349e+00]
[2.26093750e+01]
[2.39300003e+01]
[1.35699997e+01]
[2.28799992e+01]
[1.94444442e+00]
[1.55208330e+01]
[2.46599998e+01]
[1.98600006e+01]
[4.50900002e+01]
[1.30799999e+01]
[3.04166675e+00]
[3.14799995e+01]
[1.09791670e+01]
[2.47099991e+01]
[1.00520837e+00]
[2.65300007e+01]
[2.55100002e+01]
[2.48500004e+01]
[5.22187500e+01]
[2.84699993e+01]
[2.31599998e+01]
[1.72708340e+01]
[1.86100006e+01]
[2.03199997e+01]
[6.31944466e+00]
[2.37399998e+01]
[1.00486107e+01]
[2.70000000e+01]
[3.06800003e+01]
[5.32500000e+01]
[1.70799999e+01]
```

```
[1.92999992e+01]
[2.68600006e+01]
[1.62099991e+01]
[4.32999992e+01]
[8.47916698e+00]
[1.59899998e+01]
[2.51700001e+01]
[2.62900009e+01]
[5.26909709e-01]
[2.32199993e+01]
[3.64900017e+01]
[2.56875000e+01]
[7.48611116e+00]
[2.65300007e+01]
[7.06944466e+00]
[1.74599991e+01]
[6.80555582e-01]
[8.76736119e-02]
[1.85599995e+01]
[2.88437500e+01]
[4.71100006e+01]
[2.00937500e+01]
[8.11805534e+00]
[5.94444466e+00]
[4.76388884e+00]
[1.67399998e+01]
[5.24305534e+00]
[6.65000000e+01]
[3.42400017e+01]
[5.58600006e+01]
[1.19097221e+00]
[1.99305558e+00]
[3.13899994e+01]
[1.87299995e+01]
[1.05208337e+00]
[6.51388884e+00]
[1.64799995e+01]
[4.18750000e+01]
[1.79300003e+01]
[1.51300001e+01]
[6.22222233e+00]
[2.92999992e+01]
[3.24900017e+01]
[5.52099991e+01]
[1.28699999e+01]
[3.06299992e+01]
[1.79400005e+01]
[4.21699982e+01]
[2.69899998e+01]
```

```
 [1.43900003e+01]
 [2.46800003e+01]
 [2.98899994e+01]
 [1.97000008e+01]
 [3.82500000e+01]
 [2.87999992e+01]
 [5.61699982e+01]
 [2.63700008e+01]
 [1.75200005e+01]
 [4.57799988e+01]
 [2.31700001e+01]
 [4.20099983e+01]
 [2.60300007e+01]
 [4.69000015e+01]
 [3.84722233e+00]
 [5.62500000e-01]
 [1.12600002e+01]
 [1.88500004e+01]
 [7.18055534e+00]
 [1.74699993e+01]
 [2.87600002e+01]
 [3.87700005e+01]
 [2.64699993e+01]
 [1.67700005e+01]
 [2.87700005e+01]
 [8.93055534e+00]
 [5.77777767e+00]
 [9.65625000e+00]
 [1.80300007e+01]
 [4.42099991e+01]
 [2.90100002e+01]
 [4.46899986e+01]
 [4.37799988e+01]
 [5.34700012e+01]
 [1.91700001e+01]
 [2.22399998e+01]
 [2.89500008e+01]
 [2.41399994e+01]
 [5.05208313e-01]
 [3.79861116e+00]
 [2.99062500e+01]
 [1.98299999e+01]
 [1.99400005e+01]
 [2.43055558e+00]
 [5.08680582e-01]
 [1.40500002e+01]
 [8.88888931e+00]
 [7.15277791e-01]
 [2.42900009e+01]
```

```
[8.52777767e+00]
[2.17999992e+01]
[7.20833349e+00]
[5.27400017e+01]
[2.69200001e+01]
[1.74799995e+01]
[2.73700008e+01]
[1.59027779e+00]
[4.80999985e+01]
[2.76299992e+01]
[3.28750000e+01]
[1.50000000e+00]
[3.95138884e+00]
[1.70138884e+00]
[1.29600000e+01]
[1.91000004e+01]
[1.99531250e+01]
[1.76900005e+01]
[1.88999996e+01]
[1.58599997e+01]
[1.95300007e+01]
[5.06944466e+00]
[5.64199982e+01]
[1.08506948e-01]
[5.62500000e+00]
[3.22200012e+01]
[1.97800007e+01]
[2.37000008e+01]
[4.67900009e+01]
[1.91900005e+01]
[7.04166651e+00]
[1.81499996e+01]
[1.99099998e+01]
[2.21700001e+01]
[3.05200005e+01]
[1.76900005e+01]
[1.86700001e+01]
[8.10763896e-01]
[4.78699989e+01]
[1.55625000e+01]
[1.74699993e+01]
[8.12500000e-01]
[1.72700005e+01]
[3.58800011e+01]
[3.18500004e+01]
[2.13600006e+01]
[2.62099991e+01]
[5.77400017e+01]
[2.36100006e+01]
```

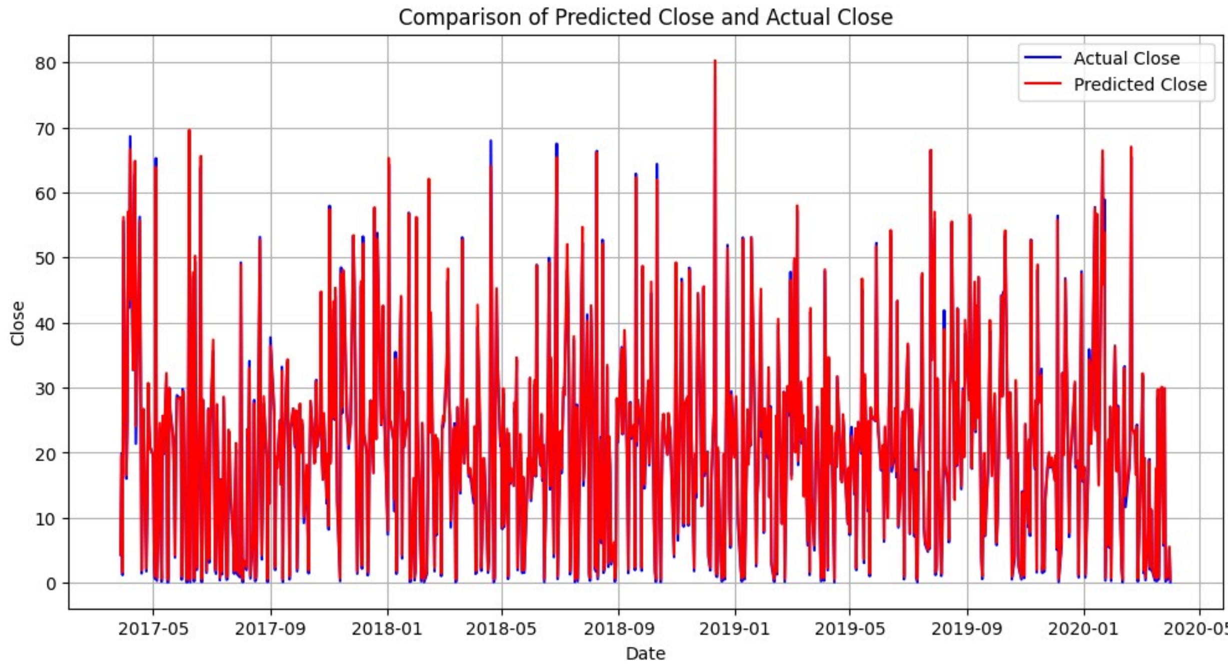
```
[5.61899986e+01]
[2.52031250e+01]
[1.51400003e+01]
[6.52500000e+01]
[4.61399994e+01]
[5.89062500e+01]
[3.29861104e-01]
[2.16700001e+01]
[5.38888884e+00]
[2.70300007e+01]
[2.69965291e-01]
[1.52500000e+01]
[3.64375000e+01]
[2.66800003e+01]
[1.71599998e+01]
[2.72399998e+01]
[2.00699997e+01]
[2.06250000e+00]
[7.85590261e-02]
[2.86900005e+01]
[3.32799988e+01]
[1.16099997e+01]
[1.74699993e+01]
[2.17600002e+01]
[6.55000000e+01]
[2.35300007e+01]
[2.31299992e+01]
[1.66700001e+01]
[2.42900009e+01]
[1.83159724e-01]
[5.37500000e+00]
[2.28999996e+01]
[3.17800007e+01]
[1.47222221e+00]
[1.91599998e+01]
[3.88888896e-01]
[7.19444466e+00]
[1.89899998e+01]
[1.98611116e+00]
[1.11250000e+01]
[1.83333337e+00]
[2.98611104e-01]
[1.73799992e+01]
[2.36111104e-01]
[2.85312500e+01]
[4.93055552e-01]
[2.99200001e+01]
[2.16299992e+01]
[5.70833349e+00]
[2.98899994e+01]
```

```
 [1.99652776e-01]
 [7.04861104e-01]
 [5.12500000e+00]
 [9.28819478e-02]]
```

```
import numpy as np

# Menghitung RMSE
rmse = np.sqrt(np.mean((y_pred_inverse_csco -
y_test_inverse_csco)**2))

# Menghitung MAE
mae = np.mean(np.abs(y_pred_inverse_csco - y_test_inverse_csco))

# Menghitung MAPE
mape = np.mean(np.abs((y_test_inverse_csco - y_pred_inverse_csco) /
y_test_inverse_csco)) * 100

# Menampilkan hasil evaluasi
print("RMSE on Test Data:", rmse)
print("MAE on Test Data:", mae)
print("MAPE on Test Data:", mape)
```

```
RMSE on Test Data: 0.7052914338561512
MAE on Test Data: 0.5036216228905874
MAPE on Test Data: 26.95446903377363
```

```
import matplotlib.pyplot as plt

# Mengambil tanggal dari data test
test_dates_csco = df_csco.index[-len(y_test_inverse_csco):]

# Plot hasil prediksi dan nilai Close yang sebenarnya
plt.figure(figsize=(12, 6))
plt.plot(test_dates_csco, y_test_inverse_csco, label='Actual Close',
color='blue')
plt.plot(test_dates_csco, y_pred_inverse_csco, label='Predicted
Close', color='red')
plt.xlabel('Date')
plt.ylabel('Close')
plt.title('Comparison of Predicted Close and Actual Close')
plt.legend()
plt.grid(True)
plt.show()
```



No. 1C - CSCO

Saya menggunakan metode grid search untuk menemukan nilai hyperparameter yang optimal karena grid search dapat secara sistematis mengevaluasi berbagai kombinasi nilai hyperparameter dan memilih kombinasi yang menghasilkan kinerja model terbaik sehingga membantu menghemat waktu dan sumber daya karena mencari nilai hiperparameter optimal secara manual dapat memakan waktu yang lama. Untuk memperbaiki kinerja model, terdapat beberapa pendekatan yang dapat dilakukan seperti:

1. Menambah atau mengurangi layer pada model
2. Mengubah jumlah neuron pada setiap layer
3. Mengubah learning rate pada optimizer
4. Menggunakan teknik regularisasi seperti dropout atau L2 regularization
5. Memilih metrik evaluasi yang lebih tepat Berikut adalah beberapa perubahan yang dapat dilakukan pada arsitektur baseline untuk meningkatkan kinerjanya:
6. Menambahkan layer Dropout: Layer dropout dapat membantu mencegah overfitting dengan secara acak menghilangkan neuron selama pelatihan.

```
from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import GridSearchCV
from keras.optimizers import Adam
from keras.layers import Dropout

# Fungsi untuk membangun model LSTM
def build_lstm_model(units =50, activation = 'relu',
learning_rate=0.001, dropout=0.0):
    model = Sequential()
```

```
    model.add(LSTM(units=units, activation=activation,
input_shape=(window_size, 1)))
    model.add(Dropout(dropout))
    model.add(Dense(1))
    optimizer = Adam(learning_rate=learning_rate)
    model.compile(optimizer=optimizer, loss='mean_squared_error')
    return model

# Membungkus model dengan KerasRegressor agar dapat digunakan dengan
GridSearchCV
lstm_model = KerasRegressor(build_fn=build_lstm_model, verbose=0)
```

```
<ipython-input-29-ca58bac11d11>:2: DeprecationWarning: KerasRegressor
is deprecated, use Sci-Keras (https://github.com/adriangb/scikeras)
instead. See https://www.adriangb.com/scikeras/stable/migration.html
for help migrating.
  lstm_model = KerasRegressor(build_fn=build_lstm_model, verbose=0)
```

```
# Menentukan hyperparameter yang akan diuji
param_grid = {
    'units': [32, 64],
    'activation': ['relu', 'tanh'],
    'learning_rate': [0.001, 0.01, 0.1],
    'dropout': [0.0, 0.2],
    'epochs': [10, 20]
}

# Membangun objek GridSearchCV untuk dataset CSCO
grid_search_csco = GridSearchCV(estimator=lstm_model,
param_grid=param_grid, cv=3)
grid_search_csco.fit(X_train_csco, y_train_csco)
```

```
GridSearchCV(cv=3,
             estimator=<keras.wrappers.scikit_learn.KerasRegressor
object at 0x7f12912e4af0>,
             param_grid={'activation': ['relu', 'tanh'], 'dropout':
[0.0, 0.2],
                         'epochs': [10, 20],
                         'learning_rate': [0.001, 0.01, 0.1],
                         'units': [32, 64]})
```

```
# Menampilkan hasil pencarian kombinasi terbaik untuk dataset AMZN
print("Best Parameters for AMZN:", grid_search_csco.best_params_)
print("Best Score (RMSE) for AMZN:", np.sqrt(-
grid_search_csco.best_score_))
```

```
Best Parameters for AMZN: {'activation': 'relu', 'dropout': 0.0,
'epochs': 20, 'learning_rate': 0.01, 'units': 32}
Best Score (RMSE) for AMZN: 0.009588723998627991
```

Dari hasil di atas diperoleh bahwa parameter terbaik yang menghasikan akurasi optimal adalah

1. 'epochs': 20,
2. 'optimizer': 'adam' dengan 'learn_rate': 0.01
3. 'Units' : 32
4. 'activation' : 'relu'

Model arsitektur LSTM yang dimodifikasi

```
# Membangun model LSTM
model = Sequential()
model.add(LSTM(32, activation='relu', input_shape=(window_size, 1)))
model.add(Dense(1))

from tensorflow.keras.optimizers import Adam

# Kompilasi model
optimizer = Adam(learning_rate=0.01)
model.compile(optimizer=optimizer, loss='mean_squared_error',
metrics=['mae'])

print(model.summary())

Model: "sequential_147"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_147 (LSTM)             (None, 32)                4352

 dense_147 (Dense)           (None, 1)                 33

=================================================================
Total params: 4,385
Trainable params: 4,385
Non-trainable params: 0
_________________________________________________________________
None

# Latih model dengan data train
modelcsco=model.fit(X_train_csco, y_train_csco, epochs=20,
batch_size=64, validation_data=(X_val_csco, y_val_csco))

Epoch 1/20
95/95 [==============================] - 2s 8ms/step - loss: 0.0054 -
mae: 0.0371 - val_loss: 1.1204e-04 - val_mae: 0.0070
Epoch 2/20
95/95 [==============================] - 0s 5ms/step - loss: 1.1550e-
04 - mae: 0.0064 - val_loss: 1.1204e-04 - val_mae: 0.0066
Epoch 3/20
95/95 [==============================] - 0s 5ms/step - loss: 1.2320e-
04 - mae: 0.0069 - val_loss: 1.0781e-04 - val_mae: 0.0065
Epoch 4/20
95/95 [==============================] - 1s 6ms/step - loss: 1.2424e-
```

```
04 - mae: 0.0070 - val_loss: 9.6961e-05 - val_mae: 0.0055
Epoch 5/20
95/95 [==============================] - 1s 9ms/step - loss: 1.1868e-
04 - mae: 0.0066 - val_loss: 9.6290e-05 - val_mae: 0.0058
Epoch 6/20
95/95 [==============================] - 1s 8ms/step - loss: 1.1622e-
04 - mae: 0.0066 - val_loss: 1.1686e-04 - val_mae: 0.0071
Epoch 7/20
95/95 [==============================] - 1s 8ms/step - loss: 1.2335e-
04 - mae: 0.0070 - val_loss: 9.3193e-05 - val_mae: 0.0054
Epoch 8/20
95/95 [==============================] - 1s 7ms/step - loss: 1.1496e-
04 - mae: 0.0068 - val_loss: 9.7922e-05 - val_mae: 0.0063
Epoch 9/20
95/95 [==============================] - 0s 5ms/step - loss: 1.1110e-
04 - mae: 0.0065 - val_loss: 1.2161e-04 - val_mae: 0.0079
Epoch 10/20
95/95 [==============================] - 0s 5ms/step - loss: 1.0638e-
04 - mae: 0.0062 - val_loss: 1.0857e-04 - val_mae: 0.0070
Epoch 11/20
95/95 [==============================] - 0s 5ms/step - loss: 1.1139e-
04 - mae: 0.0066 - val_loss: 9.5765e-05 - val_mae: 0.0054
Epoch 12/20
95/95 [==============================] - 0s 5ms/step - loss: 1.2424e-
04 - mae: 0.0072 - val_loss: 1.0500e-04 - val_mae: 0.0069
Epoch 13/20
95/95 [==============================] - 0s 5ms/step - loss: 1.0862e-
04 - mae: 0.0066 - val_loss: 9.8510e-05 - val_mae: 0.0062
Epoch 14/20
95/95 [==============================] - 0s 5ms/step - loss: 1.2039e-
04 - mae: 0.0073 - val_loss: 9.4415e-05 - val_mae: 0.0057
Epoch 15/20
95/95 [==============================] - 0s 5ms/step - loss: 1.0374e-
04 - mae: 0.0066 - val_loss: 9.2918e-05 - val_mae: 0.0057
Epoch 16/20
95/95 [==============================] - 0s 5ms/step - loss: 1.0840e-
04 - mae: 0.0068 - val_loss: 9.2046e-05 - val_mae: 0.0055
Epoch 17/20
95/95 [==============================] - 0s 5ms/step - loss: 9.4903e-
05 - mae: 0.0062 - val_loss: 1.3743e-04 - val_mae: 0.0088
Epoch 18/20
95/95 [==============================] - 0s 5ms/step - loss: 1.0815e-
04 - mae: 0.0068 - val_loss: 1.1080e-04 - val_mae: 0.0077
Epoch 19/20
95/95 [==============================] - 1s 6ms/step - loss: 9.0365e-
05 - mae: 0.0059 - val_loss: 8.1766e-05 - val_mae: 0.0052
Epoch 20/20
95/95 [==============================] - 1s 5ms/step - loss: 9.4941e-
05 - mae: 0.0062 - val_loss: 8.2510e-05 - val_mae: 0.0052
```

```
# Melakukan prediksi menggunakan model terbaik untuk dataset CSCO
best_model_csco = grid_search_csco.best_estimator_
y_pred_csco = best_model_csco.predict(X_test_csco)
```

NO. 1D - CSCO

```
from sklearn.metrics import mean_squared_error, mean_absolute_error

# Evaluasi LSTM biasa untuk dataset AMZN
y_pred_lstm_csco = model.predict(X_test_csco)
y_pred_inverse_lstm_csco = scaler.inverse_transform(y_pred_lstm_csco)
y_test_inverse_csco = scaler.inverse_transform(y_test_csco)

rmse_lstm_csco = np.sqrt(mean_squared_error(y_test_inverse_csco,
y_pred_inverse_lstm_csco))
mae_lstm_csco = mean_absolute_error(y_test_inverse_csco,
y_pred_inverse_lstm_csco)
mape_lstm_csco = np.mean(np.abs((y_test_inverse_csco -
y_pred_inverse_lstm_csco) / y_test_inverse_csco)) * 100

print("Performance of LSTM Architecture for AMZN:")
print("RMSE:", rmse_lstm_csco)
print("MAE:", mae_lstm_csco)
print("MAPE:", mape_lstm_csco)

24/24 [==============================] - 0s 3ms/step
Performance of LSTM Architecture for AMZN:
RMSE: 0.7052914338561512
MAE: 0.5036216228905874
MAPE: 26.95446903377363

# Evaluasi LSTM dengan modifikasi GridSearch untuk dataset AMZN
y_pred_gridsearch_csco = best_model_csco.predict(X_test_csco)
y_pred_inverse_gridsearch_csco =
scaler.inverse_transform(y_pred_gridsearch_csco.reshape(-1, 1))

rmse_gridsearch_csco = np.sqrt(mean_squared_error(y_test_inverse_csco,
y_pred_inverse_gridsearch_csco))
mae_gridsearch_csco = mean_absolute_error(y_test_inverse_csco,
y_pred_inverse_gridsearch_csco)
mape_gridsearch_csco = np.mean(np.abs((y_test_inverse_csco -
y_pred_inverse_gridsearch_csco) / y_test_inverse_csco)) * 100

print("Performance of LSTM Architecture with GridSearch for AMZN:")
print("RMSE:", rmse_gridsearch_csco)
print("MAE:", mae_gridsearch_csco)
print("MAPE:", mape_gridsearch_csco)
```

```
Performance of LSTM Architecture with GridSearch for AMZN:
RMSE: 0.6191041361225815
MAE: 0.42538168365246215
MAPE: 8.7894907523707
```

RMSE (Root Mean Squared Error): Nilai RMSE menggambarkan tingkat kesalahan prediksi dalam satuan yang sama dengan variabel target. Semakin rendah nilai RMSE, semakin baik model dalam memprediksi nilai Close yang sebenarnya. Jadi, kita membandingkan nilai RMSE antara kedua arsitektur LSTM. Hasil RMSE LSTM dengan modifikasi GridSearch lebih rendah, itu menunjukkan bahwa model tersebut memberikan prediksi yang lebih akurat. RMSE model arsitektur baseline: 0.705 RMSE model arsitektur modifikasi : 0.619

MAE (Mean Absolute Error): Nilai MAE merupakan nilai rata-rata dari selisih absolut antara nilai prediksi dan nilai Close yang sebenarnya. Semakin rendah nilai MAE, semakin baik model dalam memprediksi nilai Close yang sebenarnya. Kita dapat membandingkan nilai MAE antara kedua arsitektur LSTM dan dapat kita lihat bahwa nilai MAE pada gridsearch lebih rendah (0.425) jadi model ini lebih memberikan prediksi yang akurat mana yang memiliki MAE yang lebih rendah untuk menentukan model yang lebih baik.

MAPE adalah singkatan dari "Mean Absolute Percentage Error" atau "Kesalahan Persentase Mutlak Rata-Rata". Ini adalah metrik yang digunakan untuk mengukur tingkat kesalahan atau ketidakakuratan dalam ramalan atau perkiraan. Pada arsitektur gridsearch mendapatkan nilai MAPE yang lebih rendah (26.95) daripada arsitektur baseline (8.789)

Dengan ini maka model arsitektur yang sudah dimodifikasi menggunakan hyperparameter tunning gridsearch lebih cocok daripada menggunakan model arsitektur baseline.

No. 1E

Link Google Colab:
https://drive.google.com/drive/folders/1fVu56rHJ7ojt87U3utHyqixNhaPjlbd7?usp=sharing