stock price perdiction by using LSTM method

In [1]:
```python
import pandas as pd
```

In [2]:
```python
import pandas as pd

df = pd.read_csv('MSFT.csv')

df
```

Out[2]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	1986-03-13	0.088542	0.101563	0.088542	0.097222	0.060524	1031788800
1	1986-03-14	0.097222	0.102431	0.097222	0.100694	0.062686	308160000
2	1986-03-17	0.100694	0.103299	0.100694	0.102431	0.063767	133171200
3	1986-03-18	0.102431	0.103299	0.098958	0.099826	0.062145	67766400
4	1986-03-19	0.099826	0.100694	0.097222	0.098090	0.061065	47894400
...	...	...	...	...	...	...	...
9405	2023-07-10	334.600006	335.230011	327.589996	331.829987	331.829987	32791400
9406	2023-07-11	331.059998	332.859985	327.000000	332.470001	332.470001	26698200
9407	2023-07-12	336.600006	341.649994	335.670013	337.200012	337.200012	29995300
9408	2023-07-13	339.559998	343.739990	339.019989	342.660004	342.660004	20567200
9409	2023-07-14	347.589996	351.429993	344.309998	345.239990	345.239990	28302200

9410 rows × 7 columns

In [3]:
```python
df = df[['Date', 'Close']]

df
```

Out[3]:

	Date	Close
0	1986-03-13	0.097222
1	1986-03-14	0.100694
2	1986-03-17	0.102431
3	1986-03-18	0.099826
4	1986-03-19	0.098090
...	...	...
9405	2023-07-10	331.829987
9406	2023-07-11	332.470001
9407	2023-07-12	337.200012
9408	2023-07-13	342.660004
9409	2023-07-14	345.239990

9410 rows × 2 columns

In [4]:
```python
df['Date']
```

Out[4]:
```
0       1986-03-13
1       1986-03-14
2       1986-03-17
3       1986-03-18
4       1986-03-19
           ...
9405    2023-07-10
9406    2023-07-11
9407    2023-07-12
9408    2023-07-13
9409    2023-07-14
Name: Date, Length: 9410, dtype: object
```

In [5]:
```python
import datetime

def str_to_datetime(s):
  split = s.split('-')
  year, month, day = int(split[0]), int(split[1]), int(split[2])
  return datetime.datetime(year=year, month=month, day=day)

datetime_object = str_to_datetime('1986-03-19')
datetime_object
```

Out[5]:
```
datetime.datetime(1986, 3, 19, 0, 0)
```

In [6]:
```python
df
```

Out[6]:

	Date	Close
0	1986-03-13	0.097222
1	1986-03-14	0.100694
2	1986-03-17	0.102431
3	1986-03-18	0.099826
4	1986-03-19	0.098090
...	...	...
9405	2023-07-10	331.829987
9406	2023-07-11	332.470001
9407	2023-07-12	337.200012
9408	2023-07-13	342.660004
9409	2023-07-14	345.239990

9410 rows × 2 columns

In [7]:
```python
df['Date'] = df['Date'].apply(str_to_datetime)
df['Date']
```

```
C:\Users\durga prasad\AppData\Local\Temp\ipykernel_97680\2565755782.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Date'] = df['Date'].apply(str_to_datetime)
```

Out[7]:
```
0      1986-03-13
1      1986-03-14
2      1986-03-17
3      1986-03-18
4      1986-03-19
          ...
9405   2023-07-10
9406   2023-07-11
9407   2023-07-12
9408   2023-07-13
9409   2023-07-14
Name: Date, Length: 9410, dtype: datetime64[ns]
```

In [8]:
```python
df.index = df.pop('Date')
df
```

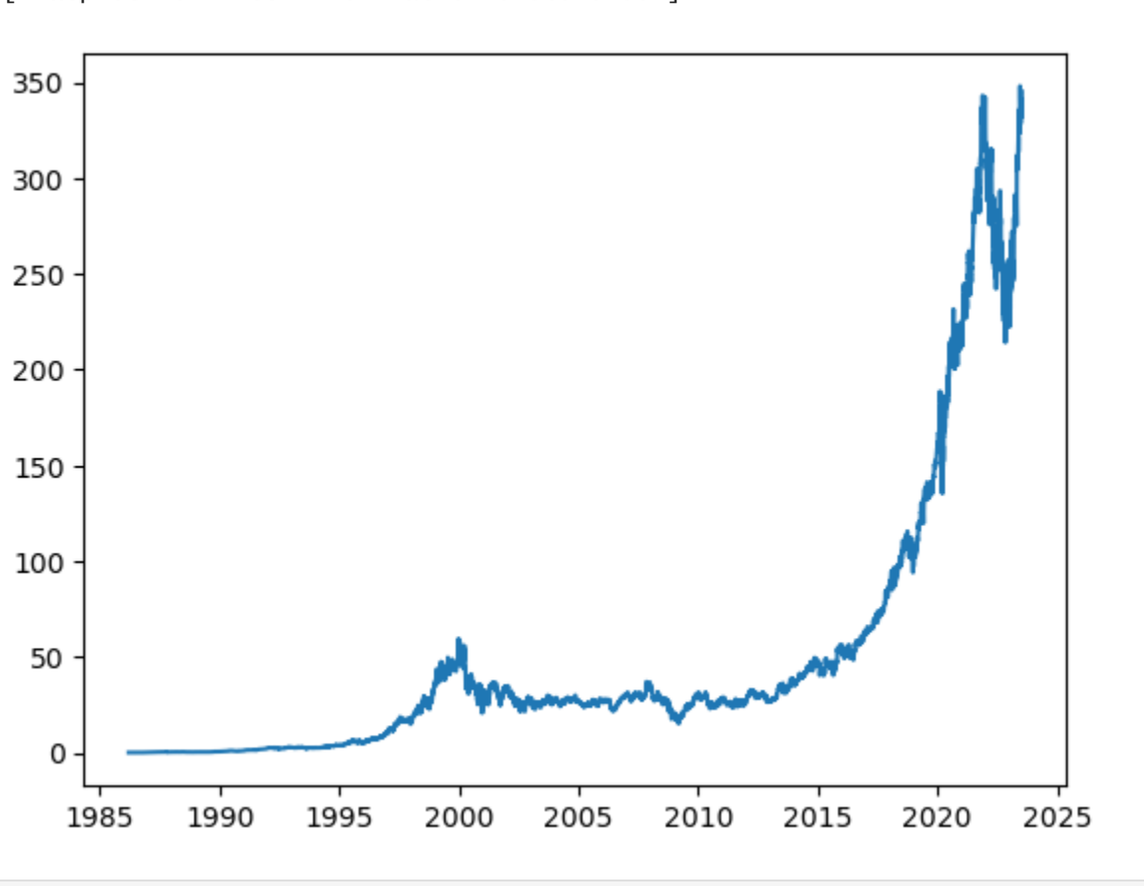
Out[8]:

Date	Close
1986-03-13	0.097222
1986-03-14	0.100694
1986-03-17	0.102431
1986-03-18	0.099826
1986-03-19	0.098090
...	...
2023-07-10	331.829987
2023-07-11	332.470001
2023-07-12	337.200012
2023-07-13	342.660004
2023-07-14	345.239990

9410 rows × 1 columns

In [9]:
```python
import matplotlib.pyplot as plt

plt.plot(df.index, df['Close'])
```

Out[9]:
```
[<matplotlib.lines.Line2D at 0x2029bd19400>]
```



In [10]:
```python
import numpy as np

def df_to_windowed_df(dataframe, first_date_str, last_date_str, n=3):
  first_date = str_to_datetime(first_date_str)
  last_date  = str_to_datetime(last_date_str)

  target_date = first_date

  dates = []
  X, Y = [], []

  last_time = False
  while True:
    df_subset = dataframe.loc[:target_date].tail(n+1)

    if len(df_subset) != n+1:
      print(f'Error: Window of size {n} is too large for date {target_date}')
      return

    values = df_subset['Close'].to_numpy()
    x, y = values[:-1], values[-1]

    dates.append(target_date)
    X.append(x)
    Y.append(y)

    next_week = dataframe.loc[target_date:target_date+datetime.timedelta(days=7)]
    next_datetime_str = str(next_week.head(2).tail(1).index.values[0])
    next_date_str = next_datetime_str.split('T')[0]
    year_month_day = next_date_str.split('-')
    year, month, day = year_month_day
    next_date = datetime.datetime(day=int(day), month=int(month), year=int(year))

    if last_time:
      break

    target_date = next_date

    if target_date == last_date:
      last_time = True

  ret_df = pd.DataFrame({})
  ret_df['Target Date'] = dates

  X = np.array(X)
  for i in range(0, n):
    X[:, i]
    ret_df[f'Target-{n-i}'] = X[:, i]

  ret_df['Target'] = Y

  return ret_df

# Start day second time around: '2021-03-25'
windowed_df = df_to_windowed_df(df,
                                '2021-03-25',
                                '2022-03-23',
                                n=3)
windowed_df
```

Out[10]:

	Target Date	Target-3	Target-2	Target-1	Target
0	2021-03-25	235.990005	237.580002	235.460007	232.339996
1	2021-03-26	237.580002	235.460007	232.339996	236.479996
2	2021-03-29	235.460007	232.339996	236.479996	235.240005
3	2021-03-30	232.339996	236.479996	235.240005	231.850006
4	2021-03-31	236.479996	235.240005	231.850006	235.770004
...	...	...	...	...	...
247	2022-03-17	276.440002	287.149994	294.390015	295.220001
248	2022-03-18	287.149994	294.390015	295.220001	300.429993
249	2022-03-21	294.390015	295.220001	300.429993	299.160004
250	2022-03-22	295.220001	300.429993	299.160004	304.059998
251	2022-03-23	300.429993	299.160004	304.059998	299.489990

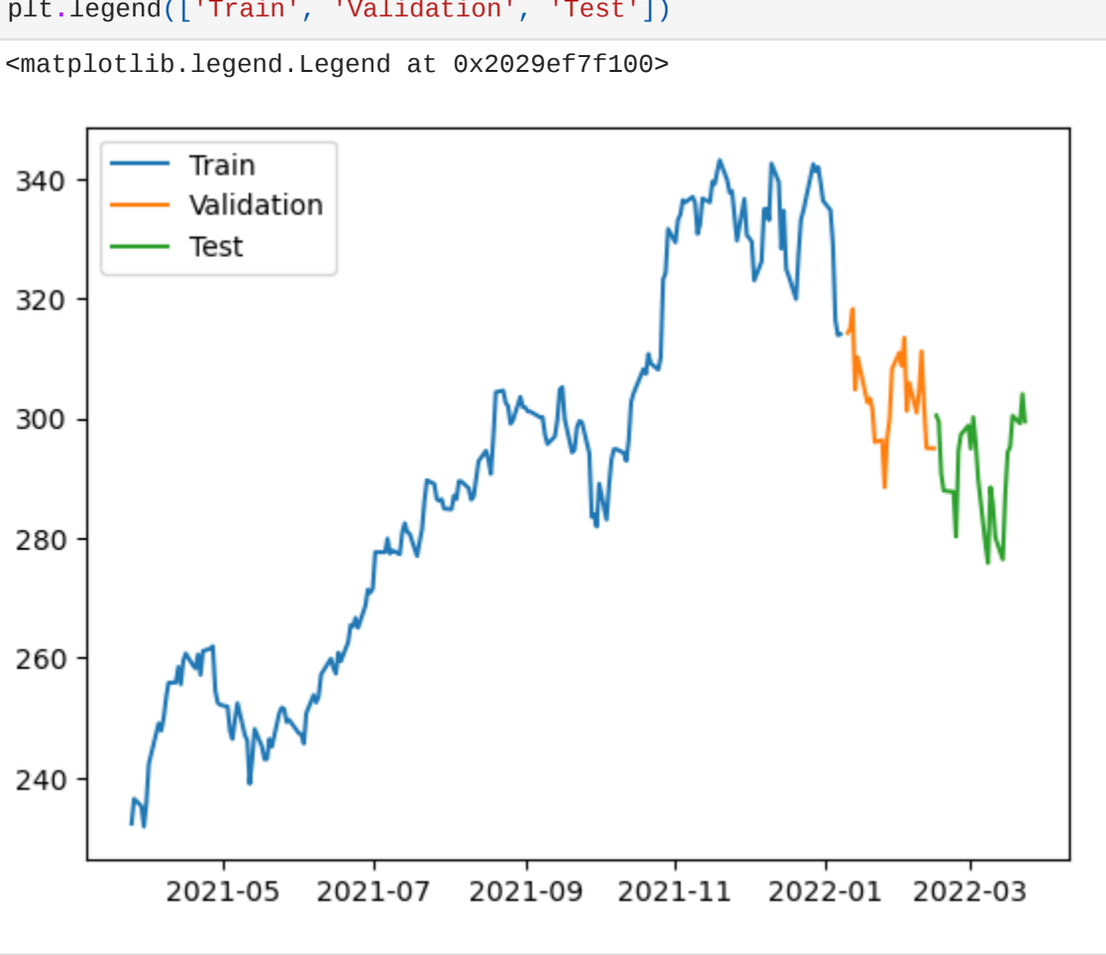
252 rows × 5 columns

In [11]:
```python
def windowed_df_to_date_X_y(windowed_dataframe):
  df_as_np = windowed_dataframe.to_numpy()

  dates = df_as_np[:, 0]

  middle_matrix = df_as_np[:, 1:-1]
  X = middle_matrix.reshape((len(dates), middle_matrix.shape[1], 1))

  Y = df_as_np[:, -1]

  return dates, X.astype(np.float32), Y.astype(np.float32)

dates, X, y = windowed_df_to_date_X_y(windowed_df)

dates.shape, X.shape, y.shape
```

Out[11]:
```
((252,), (252, 3, 1), (252,))
```

In [12]:
```python
q_80 = int(len(dates) * .8)
q_90 = int(len(dates) * .9)

dates_train, X_train, y_train = dates[:q_80], X[:q_80], y[:q_80]

dates_val, X_val, y_val = dates[q_80:q_90], X[q_80:q_90], y[q_80:q_90]
dates_test, X_test, y_test = dates[q_90:], X[q_90:], y[q_90:]

plt.plot(dates_train, y_train)
plt.plot(dates_val, y_val)
plt.plot(dates_test, y_test)

plt.legend(['Train', 'Validation', 'Test'])
```

Out[12]:
```
<matplotlib.legend.Legend at 0x2029ef7f100>
```



In [13]:
```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import layers

model = Sequential([layers.Input((3, 1)),
                    layers.LSTM(64),
                    layers.Dense(32, activation='relu'),
                    layers.Dense(32, activation='relu'),
                    layers.Dense(1)])

model.compile(loss='mse',
              optimizer=Adam(learning_rate=0.001),
              metrics=['mean_absolute_error'])

model.fit(X_train, y_train, validation_data=(X_val, y_val), epochs=100)
```

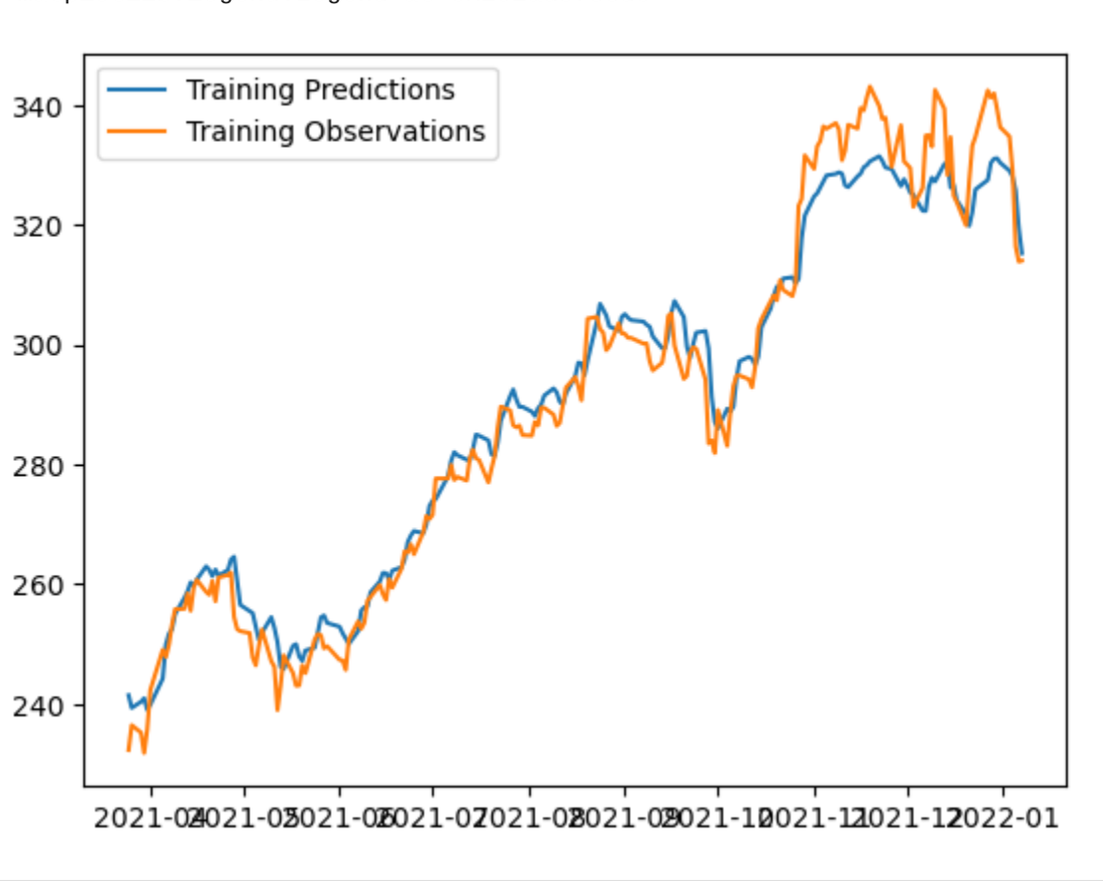
```
Epoch 1/100
7/7 [==============================] - 5s 157ms/step - loss: 84583.1328 - mean_absolute_error: 289.0900 - val_loss: 92300.3281 - val_mean_absolute_error: 303.7225
Epoch 2/100
7/7 [==============================] - 0s 20ms/step - loss: 84402.5000 - mean_absolute_error: 288.7775 - val_loss: 92131.9688 - val_mean_absolute_error: 303.4452
Epoch 3/100
7/7 [==============================] - 0s 13ms/step - loss: 84223.7109 - mean_absolute_error: 288.4669 - val_loss: 91899.6797 - val_mean_absolute_error: 303.0622
Epoch 4/100
7/7 [==============================] - 0s 20ms/step - loss: 83982.1328 - mean_absolute_error: 288.0487 - val_loss: 91620.1484 - val_mean_absolute_error: 302.6007
Epoch 5/100
7/7 [==============================] - 0s 18ms/step - loss: 83699.7578 - mean_absolute_error: 287.5579 - val_loss: 91285.4297 - val_mean_absolute_error: 302.0471
Epoch 6/100
7/7 [==============================] - 0s 13ms/step - loss: 83341.3047 - mean_absolute_error: 286.9342 - val_loss: 90815.9062 - val_mean_absolute_error: 301.2688
Epoch 7/100
7/7 [==============================] - 0s 15ms/step - loss: 82778.0156 - mean_absolute_error: 285.9506 - val_loss: 90007.6328 - val_mean_absolute_error: 299.9244
Epoch 8/100
7/7 [==============================] - 0s 16ms/step - loss: 81956.0234 - mean_absolute_error: 284.5074 - val_loss: 89044.0000 - val_mean_absolute_error: 298.3135
Epoch 9/100
7/7 [==============================] - 0s 13ms/step - loss: 80868.0469 - mean_absolute_error: 282.5819 - val_loss: 87687.7891 - val_mean_absolute_error: 296.0317
Epoch 10/100
7/7 [==============================] - 0s 13ms/step - loss: 79496.8984 - mean_absolute_error: 280.1544 - val_loss: 85935.2812 - val_mean_absolute_error: 293.0567
Epoch 11/100
7/7 [==============================] - 0s 13ms/step - loss: 77665.8594 - mean_absolute_error: 276.8627 - val_loss: 83845.3906 - val_mean_absolute_error: 289.4691
Epoch 12/100
7/7 [==============================] - 0s 14ms/step - loss: 75480.1875 - mean_absolute_error: 272.9048 - val_loss: 81011.7734 - val_mean_absolute_error: 284.5325
Epoch 13/100
7/7 [==============================] - 0s 13ms/step - loss: 72634.7031 - mean_absolute_error: 267.6251 - val_loss: 77911.6250 - val_mean_absolute_error: 279.0316
Epoch 14/100
7/7 [==============================] - 0s 12ms/step - loss: 69557.7656 - mean_absolute_error: 261.8022 - val_loss: 74335.5156 - val_mean_absolute_error: 272.5482
Epoch 15/100
7/7 [==============================] - 0s 12ms/step - loss: 66036.8750 - mean_absolute_error: 255.0155 - val_loss: 70397.9922 - val_mean_absolute_error: 265.2264
Epoch 16/100
7/7 [==============================] - 0s 17ms/step - loss: 62173.0430 - mean_absolute_error: 247.2993 - val_loss: 66033.1953 - val_mean_absolute_error: 256.8662
Epoch 17/100
7/7 [==============================] - 0s 14ms/step - loss: 57940.6172 - mean_absolute_error: 238.5432 - val_loss: 61287.9609 - val_mean_absolute_error: 247.4570
Epoch 18/100
7/7 [==============================] - 0s 23ms/step - loss: 53328.0703 - mean_absolute_error: 228.7318 - val_loss: 56207.8242 - val_mean_absolute_error: 236.9701
Epoch 19/100
7/7 [==============================] - 0s 14ms/step - loss: 48434.1836 - mean_absolute_error: 217.7701 - val_loss: 50694.0547 - val_mean_absolute_error: 225.0355
Epoch 20/100
7/7 [==============================] - 0s 22ms/step - loss: 43021.2227 - mean_absolute_error: 204.8369 - val_loss: 44424.4766 - val_mean_absolute_error: 210.6447
Epoch 21/100
7/7 [==============================] - 0s 13ms/step - loss: 37051.0859 - mean_absolute_error: 189.7482 - val_loss: 38141.3203 - val_mean_absolute_error: 195.1624
Epoch 22/100
7/7 [==============================] - 0s 14ms/step - loss: 31453.0020 - mean_absolute_error: 174.4480 - val_loss: 32244.1426 - val_mean_absolute_error: 179.4187
Epoch 23/100
7/7 [==============================] - 0s 15ms/step - loss: 25931.0098 - mean_absolute_error: 157.7147 - val_loss: 25905.7070 - val_mean_absolute_error: 160.7869
Epoch 24/100
7/7 [==============================] - 0s 14ms/step - loss: 20208.7773 - mean_absolute_error: 138.3627 - val_loss: 20108.3711 - val_mean_absolute_error: 141.6171
Epoch 25/100
7/7 [==============================] - 0s 20ms/step - loss: 15386.4375 - mean_absolute_error: 119.7521 - val_loss: 15201.8301 - val_mean_absolute_error: 123.0807
Epoch 26/100
7/7 [==============================] - 0s 17ms/step - loss: 11289.3828 - mean_absolute_error: 101.2199 - val_loss: 11006.2930 - val_mean_absolute_error: 104.6581
Epoch 27/100
7/7 [==============================] - 0s 18ms/step - loss: 7908.6768 - mean_absolute_error: 83.0129 - val_loss: 7601.9077 - val_mean_absolute_error: 86.8846
Epoch 28/100
7/7 [==============================] - 0s 20ms/step - loss: 5357.4976 - mean_absolute_error: 65.5601 - val_loss: 4977.2061 - val_mean_absolute_error: 70.1729
Epoch 29/100
7/7 [==============================] - 0s 14ms/step - loss: 3480.8350 - mean_absolute_error: 50.1049 - val_loss: 3118.2747 - val_mean_absolute_error: 55.3652
Epoch 30/100
7/7 [==============================] - 0s 20ms/step - loss: 2289.1453 - mean_absolute_error: 38.5179 - val_loss: 1878.6207 - val_mean_absolute_error: 42.7277
Epoch 31/100
7/7 [==============================] - 0s 14ms/step - loss: 1588.2301 - mean_absolute_error: 32.7288 - val_loss: 1113.2419 - val_mean_absolute_error: 32.5619
Epoch 32/100
7/7 [==============================] - 0s 18ms/step - loss: 1230.3042 - mean_absolute_error: 29.7028 - val_loss: 665.3276 - val_mean_absolute_error: 24.7459
Epoch 33/100
7/7 [==============================] - 0s 16ms/step - loss: 1063.2351 - mean_absolute_error: 27.5467 - val_loss: 430.1495 - val_mean_absolute_error: 19.4212
Epoch 34/100
7/7 [==============================] - 0s 13ms/step - loss: 1026.0426 - mean_absolute_error: 27.1047 - val_loss: 300.6086 - val_mean_absolute_error: 15.7367
Epoch 35/100
7/7 [==============================] - 0s 18ms/step - loss: 1006.5313 - mean_absolute_error: 26.8273 - val_loss: 256.0862 - val_mean_absolute_error: 14.3697
Epoch 36/100
7/7 [==============================] - 0s 17ms/step - loss: 1012.1427 - mean_absolute_error: 26.8933 - val_loss: 234.2474 - val_mean_absolute_error: 13.6448
Epoch 37/100
7/7 [==============================] - 0s 14ms/step - loss: 1011.7583 - mean_absolute_error: 26.8960 - val_loss: 237.9800 - val_mean_absolute_error: 13.7717
Epoch 38/100
7/7 [==============================] - 0s 15ms/step - loss: 1011.3625 - mean_absolute_error: 26.8903 - val_loss: 242.6335 - val_mean_absolute_error: 13.9281
Epoch 39/100
7/7 [==============================] - 0s 12ms/step - loss: 1010.7485 - mean_absolute_error: 26.8716 - val_loss: 257.3885 - val_mean_absolute_error: 14.4117
Epoch 40/100
7/7 [==============================] - 0s 12ms/step - loss: 1010.2915 - mean_absolute_error: 26.8664 - val_loss: 267.0926 - val_mean_absolute_error: 14.7203
Epoch 41/100
7/7 [==============================] - 0s 14ms/step - loss: 1009.9299 - mean_absolute_error: 26.8554 - val_loss: 270.4075 - val_mean_absolute_error: 14.8241
Epoch 42/100
7/7 [==============================] - 0s 11ms/step - loss: 1010.0215 - mean_absolute_error: 26.8564 - val_loss: 269.5949 - val_mean_absolute_error: 14.7987
Epoch 43/100
7/7 [==============================] - 0s 13ms/step - loss: 1010.4951 - mean_absolute_error: 26.8657 - val_loss: 277.3927 - val_mean_absolute_error: 15.0402
Epoch 44/100
7/7 [==============================] - 0s 12ms/step - loss: 1009.8690 - mean_absolute_error: 26.8550 - val_loss: 271.8607 - val_mean_absolute_error: 14.8693
Epoch 45/100
7/7 [==============================] - 0s 13ms/step - loss: 1009.9346 - mean_absolute_error: 26.8603 - val_loss: 264.7196 - val_mean_absolute_error: 14.6455
Epoch 46/100
7/7 [==============================] - 0s 14ms/step - loss: 1011.3848 - mean_absolute_error: 26.8768 - val_loss: 257.5962 - val_mean_absolute_error: 14.4184
Epoch 47/100
7/7 [==============================] - 0s 20ms/step - loss: 1009.6398 - mean_absolute_error: 26.8585 - val_loss: 271.3281 - val_mean_absolute_error: 14.8528
Epoch 48/100
7/7 [==============================] - 0s 14ms/step - loss: 1010.1130 - mean_absolute_error: 26.8626 - val_loss: 279.7814 - val_mean_absolute_error: 15.1134
Epoch 49/100
7/7 [==============================] - 0s 12ms/step - loss: 1009.4153 - mean_absolute_error: 26.8455 - val_loss: 266.0143 - val_mean_absolute_error: 14.6863
Epoch 50/100
7/7 [==============================] - 0s 17ms/step - loss: 1010.3193 - mean_absolute_error: 26.8645 - val_loss: 249.9544 - val_mean_absolute_error: 14.1703
Epoch 51/100
7/7 [==============================] - 0s 13ms/step - loss: 1008.7847 - mean_absolute_error: 26.8583 - val_loss: 491.8224 - val_mean_absolute_error: 21.0523
Epoch 52/100
7/7 [==============================] - 0s 12ms/step - loss: 945.1847 - mean_absolute_error: 25.9270 - val_loss: 806.4429 - val_mean_absolute_error: 27.5000
Epoch 53/100
7/7 [==============================] - 0s 13ms/step - loss: 976.0363 - mean_absolute_error: 26.4545 - val_loss: 205.6085 - val_mean_absolute_error: 12.6335
Epoch 54/100
7/7 [==============================] - 0s 11ms/step - loss: 1000.5185 - mean_absolute_error: 26.6736 - val_loss: 332.7031 - val_mean_absolute_error: 16.7716
Epoch 55/100
7/7 [==============================] - 0s 14ms/step - loss: 912.5154 - mean_absolute_error: 25.5116 - val_loss: 554.0905 - val_mean_absolute_error: 22.5095
Epoch 56/100
7/7 [==============================] - 0s 11ms/step - loss: 864.3452 - mean_absolute_error: 24.9828 - val_loss: 210.2141 - val_mean_absolute_error: 12.8704
Epoch 57/100
7/7 [==============================] - 0s 13ms/step - loss: 816.6941 - mean_absolute_error: 24.0462 - val_loss: 431.4639 - val_mean_absolute_error: 19.6344
Epoch 58/100
7/7 [==============================] - 0s 11ms/step - loss: 838.9641 - mean_absolute_error: 24.4802 - val_loss: 239.9106 - val_mean_absolute_error: 13.9461
Epoch 59/100
7/7 [==============================] - 0s 13ms/step - loss: 792.8554 - mean_absolute_error: 23.6960 - val_loss: 219.6216 - val_mean_absolute_error: 13.2944
Epoch 60/100
7/7 [==============================] - 0s 11ms/step - loss: 721.3986 - mean_absolute_error: 22.5956 - val_loss: 258.9904 - val_mean_absolute_error: 14.6643
Epoch 61/100
7/7 [==============================] - 0s 11ms/step - loss: 673.6110 - mean_absolute_error: 21.7303 - val_loss: 216.1108 - val_mean_absolute_error: 13.2058
Epoch 62/100
7/7 [==============================] - 0s 14ms/step - loss: 640.2932 - mean_absolute_error: 21.1473 - val_loss: 336.7569 - val_mean_absolute_error: 17.2011
Epoch 63/100
7/7 [==============================] - 0s 17ms/step - loss: 598.2849 - mean_absolute_error: 20.4839 - val_loss: 312.3014 - val_mean_absolute_error: 16.4897
Epoch 64/100
7/7 [==============================] - 0s 20ms/step - loss: 563.8327 - mean_absolute_error: 19.6832 - val_loss: 143.0245 - val_mean_absolute_error: 10.3973
Epoch 65/100
7/7 [==============================] - 0s 19ms/step - loss: 511.3591 - mean_absolute_error: 18.8916 - val_loss: 164.4601 - val_mean_absolute_error: 11.3605
Epoch 66/100
7/7 [==============================] - 0s 17ms/step - loss: 483.1568 - mean_absolute_error: 18.3049 - val_loss: 278.4445 - val_mean_absolute_error: 15.5376
Epoch 67/100
7/7 [==============================] - 0s 13ms/step - loss: 461.5906 - mean_absolute_error: 17.9690 - val_loss: 250.7678 - val_mean_absolute_error: 14.6513
Epoch 68/100
7/7 [==============================] - 0s 10ms/step - loss: 421.8330 - mean_absolute_error: 16.9020 - val_loss: 165.6838 - val_mean_absolute_error: 11.4635
Epoch 69/100
7/7 [==============================] - 0s 14ms/step - loss: 380.8611 - mean_absolute_error: 16.1738 - val_loss: 124.3627 - val_mean_absolute_error: 9.6524
Epoch 70/100
7/7 [==============================] - 0s 15ms/step - loss: 351.8802 - mean_absolute_error: 15.4520 - val_loss: 73.0049 - val_mean_absolute_error: 6.9866
Epoch 71/100
7/7 [==============================] - 0s 19ms/step - loss: 321.3545 - mean_absolute_error: 14.7426 - val_loss: 349.1490 - val_mean_absolute_error: 17.7690
Epoch 72/100
7/7 [==============================] - 0s 17ms/step - loss: 357.5885 - mean_absolute_error: 16.0009 - val_loss: 115.7226 - val_mean_absolute_error: 9.2953
Epoch 73/100
7/7 [==============================] - 0s 18ms/step - loss: 306.2094 - mean_absolute_error: 14.3646 - val_loss: 182.0842 - val_mean_absolute_error: 12.2130
Epoch 74/100
7/7 [==============================] - 0s 44ms/step - loss: 286.0738 - mean_absolute_error: 14.0789 - val_loss: 195.6739 - val_mean_absolute_error: 12.7757
Epoch 75/100
7/7 [==============================] - 0s 36ms/step - loss: 262.1735 - mean_absolute_error: 13.2243 - val_loss: 69.2411 - val_mean_absolute_error: 6.7991
Epoch 76/100
7/7 [==============================] - 0s 32ms/step - loss: 231.0875 - mean_absolute_error: 12.5787 - val_loss: 186.6207 - val_mean_absolute_error: 12.4507
Epoch 77/100
7/7 [==============================] - 0s 15ms/step - loss: 242.6183 - mean_absolute_error: 12.5827 - val_loss: 59.1748 - val_mean_absolute_error: 6.2486
Epoch 78/100
7/7 [==============================] - 0s 22ms/step - loss: 194.8542 - mean_absolute_error: 11.3740 - val_loss: 49.1339 - val_mean_absolute_error: 5.7120
Epoch 79/100
7/7 [==============================] - 0s 14ms/step - loss: 172.9386 - mean_absolute_error: 10.7021 - val_loss: 89.1523 - val_mean_absolute_error: 7.9828
Epoch 80/100
7/7 [==============================] - 0s 13ms/step - loss: 168.2287 - mean_absolute_error: 10.4822 - val_loss: 138.9902 - val_mean_absolute_error: 10.3976
Epoch 81/100
7/7 [==============================] - 0s 13ms/step - loss: 159.5960 - mean_absolute_error: 10.3491 - val_loss: 119.2254 - val_mean_absolute_error: 9.4548
Epoch 82/100
7/7 [==============================] - 0s 14ms/step - loss: 162.2132 - mean_absolute_error: 10.3353 - val_loss: 49.0909 - val_mean_absolute_error: 5.7568
Epoch 83/100
7/7 [==============================] - 0s 10ms/step - loss: 141.5938 - mean_absolute_error: 9.6782 - val_loss: 96.0615 - val_mean_absolute_error: 8.3234
Epoch 84/100
7/7 [==============================] - 0s 12ms/step - loss: 130.4877 - mean_absolute_error: 9.3682 - val_loss: 41.0309 - val_mean_absolute_error: 5.3191
Epoch 85/100
7/7 [==============================] - 0s 13ms/step - loss: 102.4933 - mean_absolute_error: 7.9910 - val_loss: 33.2426 - val_mean_absolute_error: 4.8306
Epoch 86/100
7/7 [==============================] - 0s 10ms/step - loss: 112.0880 - mean_absolute_error: 8.7677 - val_loss: 33.5161 - val_mean_absolute_error: 4.8456
Epoch 87/100
7/7 [==============================] - 0s 13ms/step - loss: 96.3299 - mean_absolute_error: 7.8662 - val_loss: 31.5771 - val_mean_absolute_error: 4.7918
Epoch 88/100
7/7 [==============================] - 0s 12ms/step - loss: 79.9504 - mean_absolute_error: 7.1125 - val_loss: 31.0036 - val_mean_absolute_error: 4.8416
Epoch 89/100
7/7 [==============================] - 0s 18ms/step - loss: 73.4183 - mean_absolute_error: 6.6117 - val_loss: 30.8512 - val_mean_absolute_error: 4.8393
Epoch 90/100
7/7 [==============================] - 0s 19ms/step - loss: 62.5453 - mean_absolute_error: 5.9890 - val_loss: 32.8843 - val_mean_absolute_error: 4.8334
Epoch 91/100
7/7 [==============================] - 0s 16ms/step - loss: 59.8795 - mean_absolute_error: 6.0491 - val_loss: 34.0722 - val_mean_absolute_error: 4.8741
Epoch 92/100
7/7 [==============================] - 0s 13ms/step - loss: 52.5983 - mean_absolute_error: 5.6302 - val_loss: 32.3857 - val_mean_absolute_error: 4.8091
Epoch 93/100
7/7 [==============================] - 0s 14ms/step - loss: 52.8615 - mean_absolute_error: 5.7127 - val_loss: 30.9612 - val_mean_absolute_error: 4.7778
Epoch 94/100
7/7 [==============================] - 0s 13ms/step - loss: 48.0567 - mean_absolute_error: 5.3874 - val_loss: 31.2946 - val_mean_absolute_error: 4.7617
Epoch 95/100
7/7 [==============================] - 0s 13ms/step - loss: 54.9160 - mean_absolute_error: 5.7622 - val_loss: 32.4474 - val_mean_absolute_error: 4.7977
Epoch 96/100
7/7 [==============================] - 0s 13ms/step - loss: 50.5118 - mean_absolute_error: 5.6702 - val_loss: 33.5074 - val_mean_absolute_error: 4.8432
Epoch 97/100
7/7 [==============================] - 0s 13ms/step - loss: 44.7902 - mean_absolute_error: 5.2378 - val_loss: 34.2496 - val_mean_absolute_error: 4.8672
Epoch 98/100
7/7 [==============================] - 0s 14ms/step - loss: 39.7476 - mean_absolute_error: 4.8117 - val_loss: 31.6423 - val_mean_absolute_error: 4.7499
Epoch 99/100
7/7 [==============================] - 0s 12ms/step - loss: 36.2760 - mean_absolute_error: 4.5492 - val_loss: 41.7896 - val_mean_absolute_error: 5.2624
Epoch 100/100
7/7 [==============================] - 0s 14ms/step - loss: 36.5069 - mean_absolute_error: 4.7170 - val_loss: 40.7952 - val_mean_absolute_error: 5.2138
```

Out[13]:
```
<keras.callbacks.History at 0x202a8ef5cd0>
```

In [14]:
```python
train_predictions = model.predict(X_train).flatten()

plt.plot(dates_train, train_predictions)
plt.plot(dates_train, y_train)
plt.legend(['Training Predictions', 'Training Observations'])
```

```
7/7 [==============================] - 1s 3ms/step
```

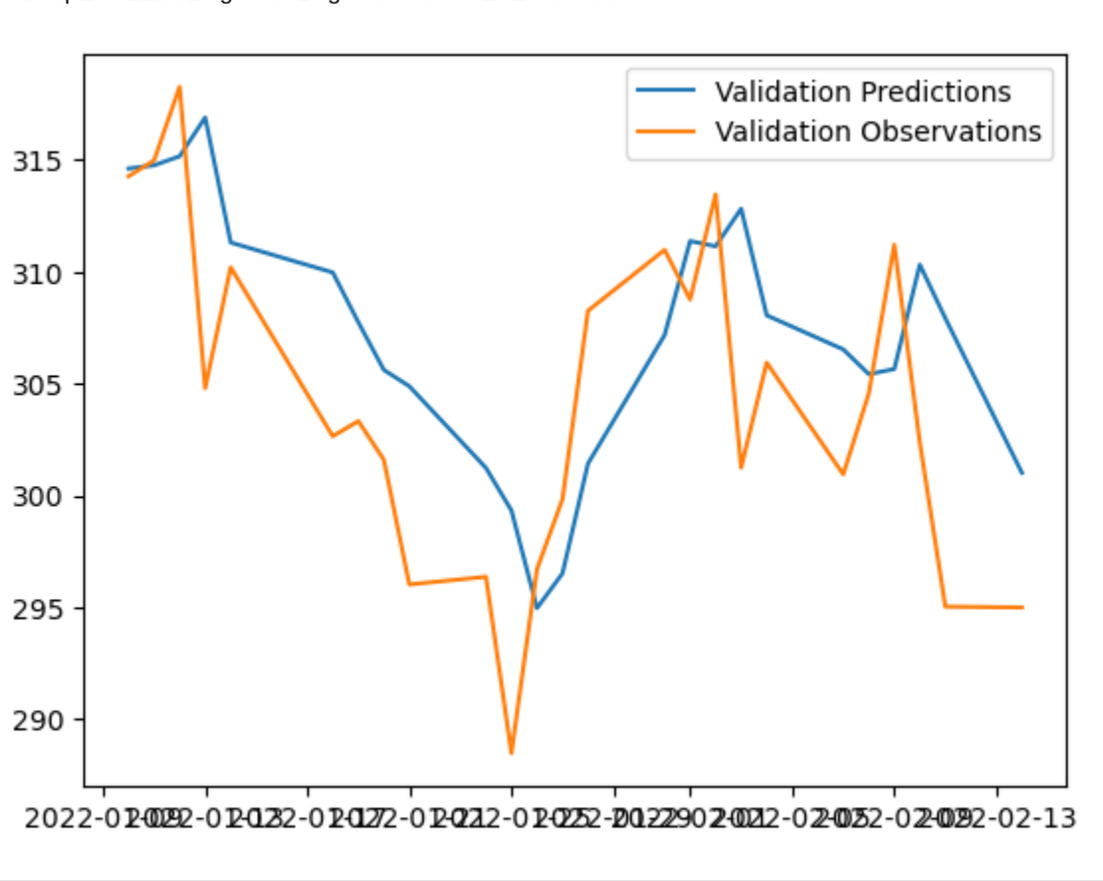
Out[14]:
```
<matplotlib.legend.Legend at 0x202ad5f53d0>
```



In [15]:
```python
val_predictions = model.predict(X_val).flatten()

plt.plot(dates_val, val_predictions)
plt.plot(dates_val, y_val)
plt.legend(['Validation Predictions', 'Validation Observations'])
```

```
1/1 [==============================] - 0s 31ms/step
```

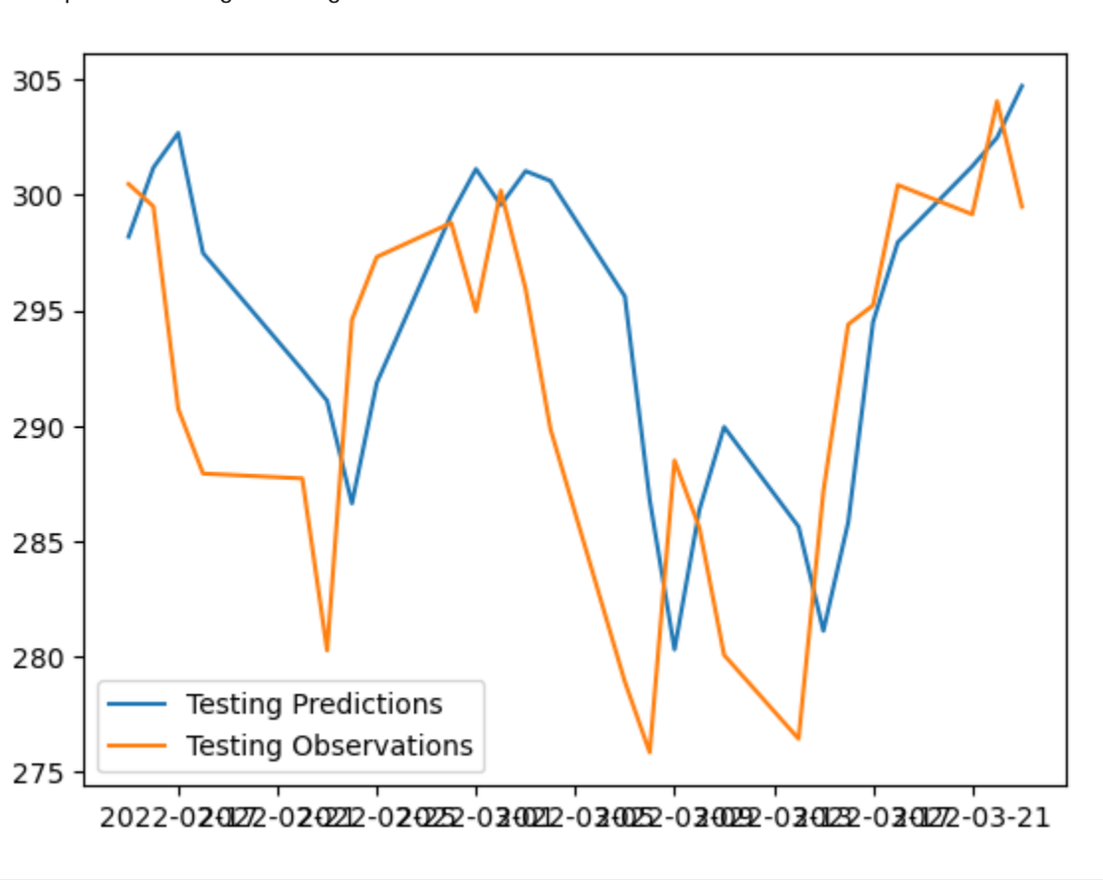
Out[15]:
```
<matplotlib.legend.Legend at 0x202ad7a9e80>
```



In [16]:
```python
test_predictions = model.predict(X_test).flatten()

plt.plot(dates_test, test_predictions)
plt.plot(dates_test, y_test)
plt.legend(['Testing Predictions', 'Testing Observations'])
```

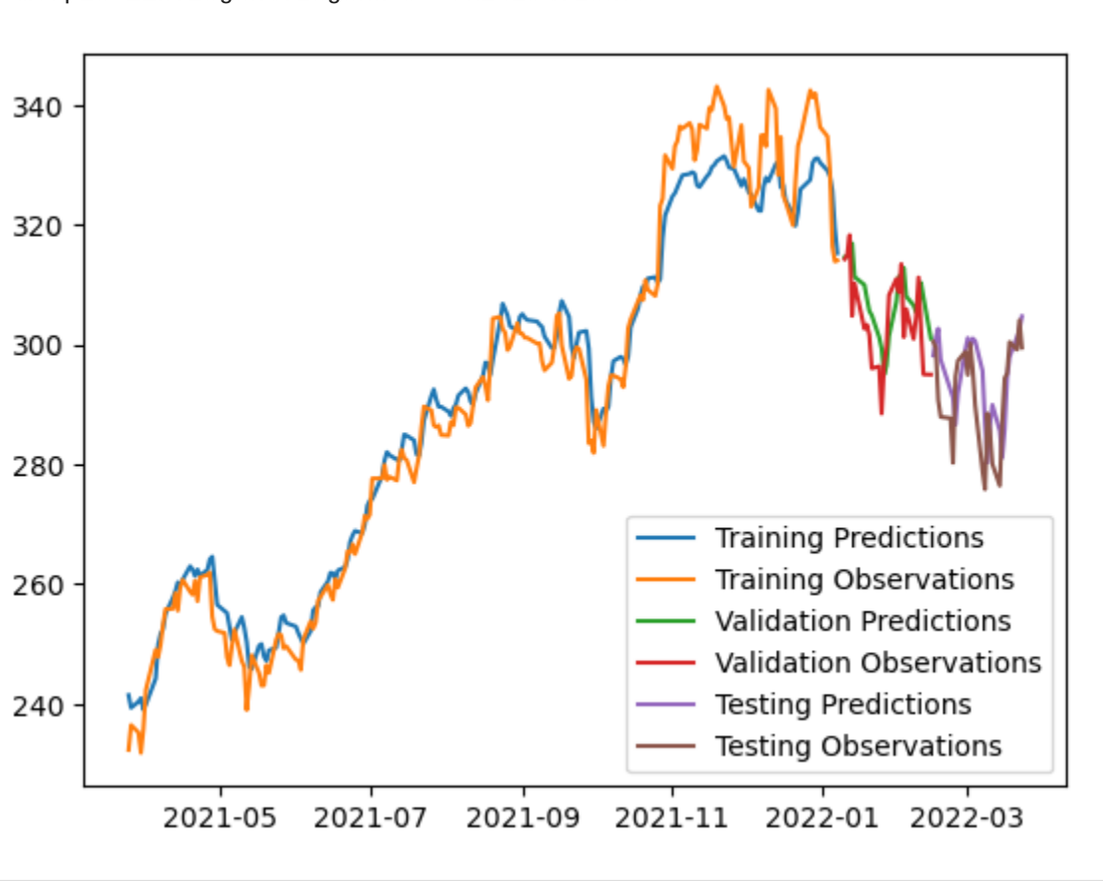
```
1/1 [==============================] - 0s 32ms/step
```

Out[16]:
```
<matplotlib.legend.Legend at 0x202ad6a0df0>
```



In [17]:
```python
plt.plot(dates_train, train_predictions)
plt.plot(dates_train, y_train)
plt.plot(dates_val, val_predictions)
plt.plot(dates_val, y_val)
plt.plot(dates_test, test_predictions)
plt.plot(dates_test, y_test)
plt.legend(['Training Predictions',
            'Training Observations',
            'Validation Predictions',
            'Validation Observations',
            'Testing Predictions',
            'Testing Observations'])
```

Out[17]:
```
<matplotlib.legend.Legend at 0x202ad72cdf0>
```

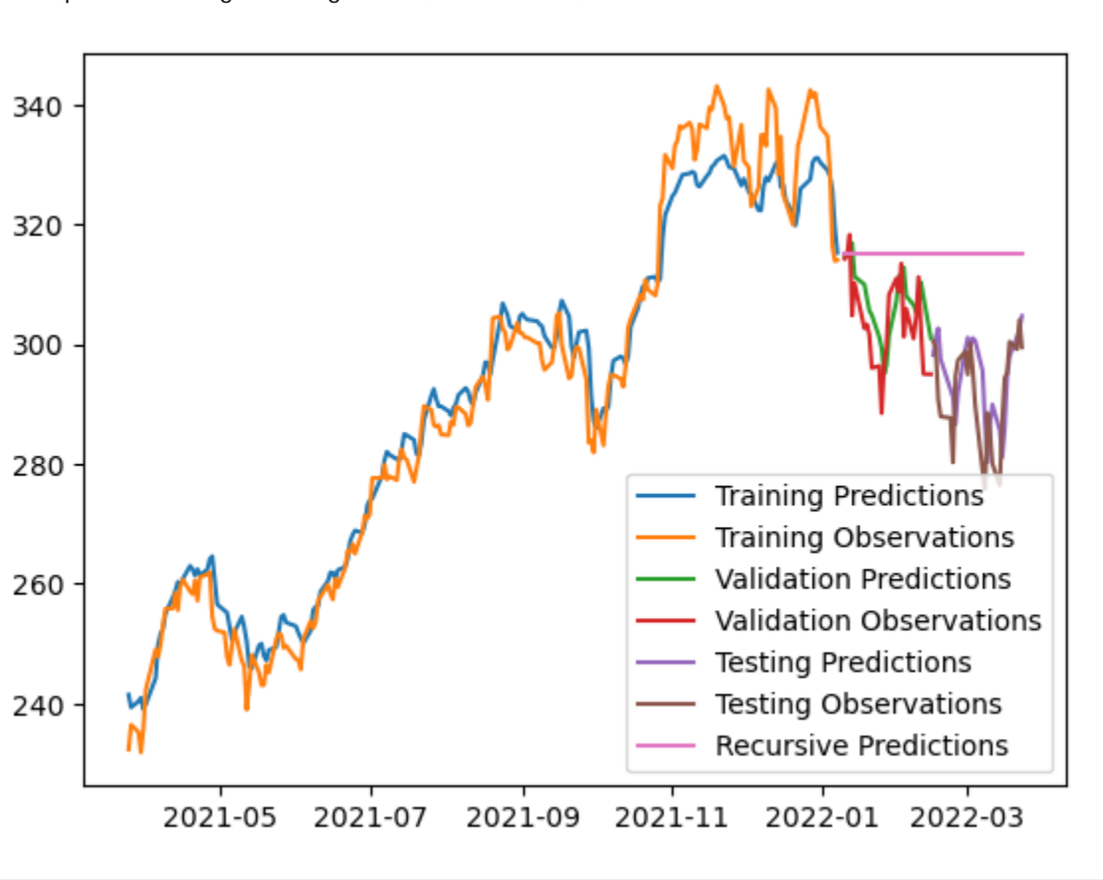


In [18]:
```python
from copy import deepcopy

recursive_predictions = []
recursive_dates = np.concatenate([dates_val, dates_test])

for target_date in recursive_dates:
  last_window = deepcopy(X_train[-1])
  next_prediction = model.predict(np.array([last_window])).flatten()
  recursive_predictions.append(next_prediction)
  last_window[-1] = next_prediction
```

```
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 31ms/step
```

In [19]:
```python
plt.plot(dates_train, train_predictions)
plt.plot(dates_train, y_train)
plt.plot(dates_val, val_predictions)
plt.plot(dates_val, y_val)
plt.plot(dates_test, test_predictions)
plt.plot(dates_test, y_test)
plt.plot(recursive_dates, recursive_predictions)
plt.legend(['Training Predictions',
            'Training Observations',
            'Validation Predictions',
            'Validation Observations',
            'Testing Predictions',
            'Testing Observations',
            'Recursive Predictions'])
```

Out[19]:
```
<matplotlib.legend.Legend at 0x202a8e32f10>
```



In [ ]: