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

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

df
```

Out[1]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	1986-03-13	0.088542	0.101563	0.088542	0.097222	0.060396	1031788800
1	1986-03-14	0.097222	0.102431	0.097222	0.100694	0.062553	308160000
2	1986-03-17	0.100694	0.103299	0.100694	0.102431	0.063632	133171200
3	1986-03-18	0.102431	0.103299	0.098958	0.099826	0.062014	67766400
4	1986-03-19	0.099826	0.100694	0.097222	0.098090	0.060936	47894400
...	...	...	...	...	...	...	...
9437	2023-08-23	323.820007	329.200012	323.459991	327.000000	327.000000	21166400
9438	2023-08-24	332.850006	332.980011	319.959991	319.970001	319.970001	23281400
9439	2023-08-25	321.470001	325.359985	318.799988	322.980011	322.980011	21671400
9440	2023-08-28	325.660004	326.149994	321.720001	323.700012	323.700012	14808500
9441	2023-08-29	321.880005	328.980011	321.880005	328.410004	328.410004	19068500

9442 rows × 7 columns

In [2]:
```
df = df[['Date','Close']]

df
```

Out[2]:

	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
...	...	...
9437	2023-08-23	327.000000
9438	2023-08-24	319.970001
9439	2023-08-25	322.980011
9440	2023-08-28	323.700012
9441	2023-08-29	328.410004

9442 rows × 2 columns

In [3]:
```
df['Date']
```

Out[3]:
```
0       1986-03-13
1       1986-03-14
2       1986-03-17
3       1986-03-18
4       1986-03-19
           ...
9437    2023-08-23
9438    2023-08-24
9439    2023-08-25
9440    2023-08-28
9441    2023-08-29
Name: Date, Length: 9442, dtype: object
```

In [5]:
```
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]:
```
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
...	...	...
9437	2023-08-23	327.000000
9438	2023-08-24	319.970001
9439	2023-08-25	322.980011
9440	2023-08-28	323.700012
9441	2023-08-29	328.410004

9442 rows × 2 columns

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

```
C:\Users\aksha\AppData\Local\Temp\ipykernel_1820\2565755782.py:1: SettingWithCopy
Warning:
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/stabl
e/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
          ...
9437   2023-08-23
9438   2023-08-24
9439   2023-08-25
9440   2023-08-28
9441   2023-08-29
Name: Date, Length: 9442, dtype: datetime64[ns]
```

In [9]:
```
df.index = df.pop('Date')
df
```

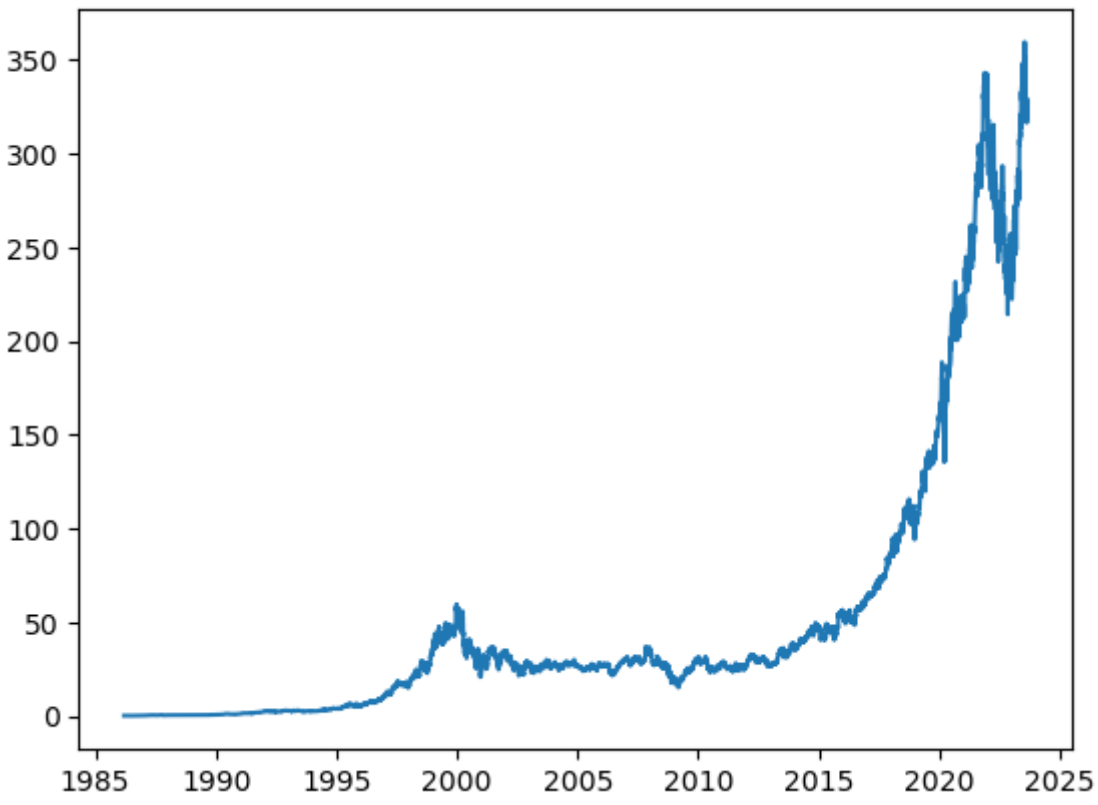
Out[9]:

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
...	...
2023-08-23	327.000000
2023-08-24	319.970001
2023-08-25	322.980011
2023-08-28	323.700012
2023-08-29	328.410004

9442 rows × 1 columns

In [10]:
```
import matplotlib.pyplot as plt

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

Out[10]:
```
[<matplotlib.lines.Line2D at 0x1b4df9c97d0>]
```



In [11]:
```
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: '2022-03-25'
windowed_df = df_to_windowed_df(df,
                                '2022-03-25',
                                '2023-08-29',
                                n=3)
windowed_df
```

Out[11]:

	Target Date	Target-3	Target-2	Target-1	Target
0	2022-03-25	304.059998	299.489990	304.100006	303.679993
1	2022-03-28	299.489990	304.100006	303.679993	310.700012
2	2022-03-29	304.100006	303.679993	310.700012	315.410004
3	2022-03-30	303.679993	310.700012	315.410004	313.859985
4	2022-03-31	310.700012	315.410004	313.859985	308.309998
...	...	...	...	...	...
354	2023-08-23	316.480011	321.880005	322.459991	327.000000
355	2023-08-24	321.880005	322.459991	327.000000	319.970001
356	2023-08-25	322.459991	327.000000	319.970001	322.980011
357	2023-08-28	327.000000	319.970001	322.980011	323.700012
358	2023-08-29	319.970001	322.980011	323.700012	328.410004

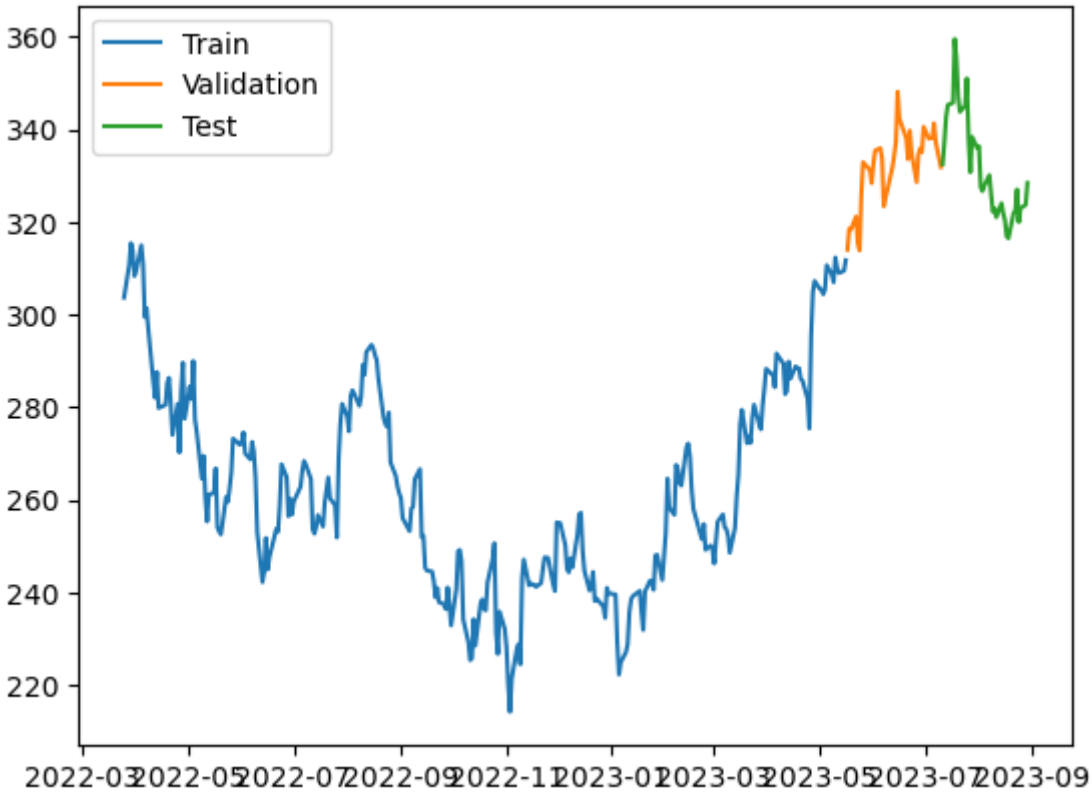
359 rows × 5 columns

In [12]:
```
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[12]:
```
((359,), (359, 3, 1), (359,))
```

In [13]:
```
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[13]:
```
<matplotlib.legend.Legend at 0x1b4e1b91450>
```



In [15]:
```
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
9/9 [==============================] - 3s 58ms/step - loss: 69555.8203 - mean_abs
olute_error: 262.7640 - val_loss: 109809.9141 - val_mean_absolute_error: 331.2732
Epoch 2/100
9/9 [==============================] - 0s 7ms/step - loss: 69299.3203 - mean_abso
lute_error: 262.2763 - val_loss: 109519.4219 - val_mean_absolute_error: 330.8344
Epoch 3/100
9/9 [==============================] - 0s 7ms/step - loss: 69076.9531 - mean_abso
lute_error: 261.8523 - val_loss: 109236.4141 - val_mean_absolute_error: 330.4064
Epoch 4/100
9/9 [==============================] - 0s 7ms/step - loss: 68845.1328 - mean_abso
lute_error: 261.4095 - val_loss: 108917.7188 - val_mean_absolute_error: 329.9238
Epoch 5/100
9/9 [==============================] - 0s 6ms/step - loss: 68557.8047 - mean_abso
lute_error: 260.8593 - val_loss: 108498.6484 - val_mean_absolute_error: 329.2881
Epoch 6/100
9/9 [==============================] - 0s 7ms/step - loss: 68163.8906 - mean_abso
lute_error: 260.1021 - val_loss: 107853.0781 - val_mean_absolute_error: 328.3063
Epoch 7/100
9/9 [==============================] - 0s 6ms/step - loss: 67497.6016 - mean_abso
lute_error: 258.8175 - val_loss: 106830.2109 - val_mean_absolute_error: 326.7447
Epoch 8/100
9/9 [==============================] - 0s 7ms/step - loss: 66507.2344 - mean_abso
lute_error: 256.8984 - val_loss: 105360.5391 - val_mean_absolute_error: 324.4881
Epoch 9/100
9/9 [==============================] - 0s 6ms/step - loss: 65365.6875 - mean_abso
lute_error: 254.6660 - val_loss: 103813.5859 - val_mean_absolute_error: 322.0956
Epoch 10/100
9/9 [==============================] - 0s 7ms/step - loss: 64028.2109 - mean_abso
lute_error: 252.0315 - val_loss: 101850.0703 - val_mean_absolute_error: 319.0330
Epoch 11/100
9/9 [==============================] - 0s 6ms/step - loss: 62297.7969 - mean_abso
lute_error: 248.5620 - val_loss: 99457.1562 - val_mean_absolute_error: 315.2604
Epoch 12/100
9/9 [==============================] - 0s 7ms/step - loss: 60277.0820 - mean_abso
lute_error: 244.4620 - val_loss: 96419.5391 - val_mean_absolute_error: 310.4053
Epoch 13/100
9/9 [==============================] - 0s 6ms/step - loss: 57567.6133 - mean_abso
lute_error: 238.8613 - val_loss: 92616.9297 - val_mean_absolute_error: 304.2185
Epoch 14/100
9/9 [==============================] - 0s 6ms/step - loss: 54492.0898 - mean_abso
lute_error: 232.3090 - val_loss: 88225.9844 - val_mean_absolute_error: 296.9141
Epoch 15/100
9/9 [==============================] - 0s 6ms/step - loss: 50682.0000 - mean_abso
lute_error: 223.9283 - val_loss: 82593.5703 - val_mean_absolute_error: 287.2726
Epoch 16/100
9/9 [==============================] - 0s 7ms/step - loss: 46306.3047 - mean_abso
lute_error: 213.9681 - val_loss: 76567.9844 - val_mean_absolute_error: 276.5861
Epoch 17/100
9/9 [==============================] - 0s 7ms/step - loss: 41436.6602 - mean_abso
lute_error: 202.2794 - val_loss: 69861.6016 - val_mean_absolute_error: 264.1848
Epoch 18/100
9/9 [==============================] - 0s 7ms/step - loss: 36446.4922 - mean_abso
lute_error: 189.5370 - val_loss: 62897.6953 - val_mean_absolute_error: 250.6583
Epoch 19/100
9/9 [==============================] - 0s 7ms/step - loss: 31063.1738 - mean_abso
lute_error: 174.7566 - val_loss: 55131.1641 - val_mean_absolute_error: 234.6554
Epoch 20/100
9/9 [==============================] - 0s 7ms/step - loss: 25715.6797 - mean_abso
lute_error: 158.6736 - val_loss: 47684.1406 - val_mean_absolute_error: 218.2112
```

```
Epoch 21/100
9/9 [==============================] - 0s 7ms/step - loss: 20641.0176 - mean_abso
lute_error: 141.7866 - val_loss: 40396.1719 - val_mean_absolute_error: 200.8188
Epoch 22/100
9/9 [==============================] - 0s 7ms/step - loss: 15921.6270 - mean_abso
lute_error: 124.1208 - val_loss: 33507.4688 - val_mean_absolute_error: 182.8646
Epoch 23/100
9/9 [==============================] - 0s 7ms/step - loss: 11798.0537 - mean_abso
lute_error: 105.9761 - val_loss: 27166.8594 - val_mean_absolute_error: 164.6173
Epoch 24/100
9/9 [==============================] - 0s 7ms/step - loss: 8181.7153 - mean_absol
ute_error: 87.4060 - val_loss: 21213.1074 - val_mean_absolute_error: 145.4126
Epoch 25/100
9/9 [==============================] - 0s 7ms/step - loss: 5087.1831 - mean_absol
ute_error: 67.3364 - val_loss: 15870.8105 - val_mean_absolute_error: 125.7092
Epoch 26/100
9/9 [==============================] - 0s 8ms/step - loss: 3003.6882 - mean_absol
ute_error: 49.6014 - val_loss: 12087.9492 - val_mean_absolute_error: 109.6356
Epoch 27/100
9/9 [==============================] - 0s 8ms/step - loss: 1745.8419 - mean_absol
ute_error: 35.2853 - val_loss: 9378.7695 - val_mean_absolute_error: 96.4923
Epoch 28/100
9/9 [==============================] - 0s 8ms/step - loss: 1044.0393 - mean_absol
ute_error: 25.5096 - val_loss: 7503.6060 - val_mean_absolute_error: 86.2298
Epoch 29/100
9/9 [==============================] - 0s 8ms/step - loss: 689.4307 - mean_absolu
te_error: 20.2928 - val_loss: 5749.2285 - val_mean_absolute_error: 75.3722
Epoch 30/100
9/9 [==============================] - 0s 8ms/step - loss: 523.3865 - mean_absolu
te_error: 18.6147 - val_loss: 4760.4995 - val_mean_absolute_error: 68.5020
Epoch 31/100
9/9 [==============================] - 0s 8ms/step - loss: 510.2256 - mean_absolu
te_error: 18.8133 - val_loss: 4759.6509 - val_mean_absolute_error: 68.4946
Epoch 32/100
9/9 [==============================] - 0s 7ms/step - loss: 514.5592 - mean_absolu
te_error: 18.8572 - val_loss: 4607.3320 - val_mean_absolute_error: 67.3755
Epoch 33/100
9/9 [==============================] - 0s 8ms/step - loss: 513.4595 - mean_absolu
te_error: 18.9798 - val_loss: 4543.2119 - val_mean_absolute_error: 66.8995
Epoch 34/100
9/9 [==============================] - 0s 8ms/step - loss: 508.7667 - mean_absolu
te_error: 18.7950 - val_loss: 4746.2788 - val_mean_absolute_error: 68.3999
Epoch 35/100
9/9 [==============================] - 0s 9ms/step - loss: 508.1140 - mean_absolu
te_error: 18.7146 - val_loss: 4733.9009 - val_mean_absolute_error: 68.3100
Epoch 36/100
9/9 [==============================] - 0s 8ms/step - loss: 508.0721 - mean_absolu
te_error: 18.7764 - val_loss: 4674.6953 - val_mean_absolute_error: 67.8746
Epoch 37/100
9/9 [==============================] - 0s 6ms/step - loss: 507.1964 - mean_absolu
te_error: 18.7178 - val_loss: 4785.0488 - val_mean_absolute_error: 68.6835
Epoch 38/100
9/9 [==============================] - 0s 6ms/step - loss: 506.4106 - mean_absolu
te_error: 18.6622 - val_loss: 4767.2729 - val_mean_absolute_error: 68.5539
Epoch 39/100
9/9 [==============================] - 0s 7ms/step - loss: 506.6581 - mean_absolu
te_error: 18.6565 - val_loss: 4779.4131 - val_mean_absolute_error: 68.6426
Epoch 40/100
9/9 [==============================] - 0s 6ms/step - loss: 505.8952 - mean_absolu
te_error: 18.6659 - val_loss: 4740.8633 - val_mean_absolute_error: 68.3612
```

```
Epoch 41/100
9/9 [==============================] - 0s 7ms/step - loss: 505.6343 - mean_absolu
te_error: 18.6741 - val_loss: 4738.2871 - val_mean_absolute_error: 68.3425
Epoch 42/100
9/9 [==============================] - 0s 6ms/step - loss: 506.0569 - mean_absolu
te_error: 18.6613 - val_loss: 4756.5098 - val_mean_absolute_error: 68.4764
Epoch 43/100
9/9 [==============================] - 0s 6ms/step - loss: 504.8896 - mean_absolu
te_error: 18.5895 - val_loss: 4702.5684 - val_mean_absolute_error: 68.0805
Epoch 44/100
9/9 [==============================] - 0s 6ms/step - loss: 505.6401 - mean_absolu
te_error: 18.7196 - val_loss: 4679.5879 - val_mean_absolute_error: 67.9113
Epoch 45/100
9/9 [==============================] - 0s 7ms/step - loss: 491.9972 - mean_absolu
te_error: 18.3478 - val_loss: 4838.4917 - val_mean_absolute_error: 69.0848
Epoch 46/100
9/9 [==============================] - 0s 8ms/step - loss: 488.1954 - mean_absolu
te_error: 18.1717 - val_loss: 4629.6885 - val_mean_absolute_error: 67.5515
Epoch 47/100
9/9 [==============================] - 0s 10ms/step - loss: 482.2434 - mean_absol
ute_error: 18.1073 - val_loss: 4622.3564 - val_mean_absolute_error: 67.5036
Epoch 48/100
9/9 [==============================] - 0s 9ms/step - loss: 473.5781 - mean_absolu
te_error: 17.9548 - val_loss: 4615.4805 - val_mean_absolute_error: 67.4588
Epoch 49/100
9/9 [==============================] - 0s 10ms/step - loss: 470.9189 - mean_absol
ute_error: 17.9085 - val_loss: 4554.7256 - val_mean_absolute_error: 67.0109
Epoch 50/100
9/9 [==============================] - 0s 8ms/step - loss: 464.2463 - mean_absolu
te_error: 17.8287 - val_loss: 4500.0518 - val_mean_absolute_error: 66.6047
Epoch 51/100
9/9 [==============================] - 0s 9ms/step - loss: 460.4318 - mean_absolu
te_error: 17.7205 - val_loss: 4390.0884 - val_mean_absolute_error: 65.7749
Epoch 52/100
9/9 [==============================] - 0s 7ms/step - loss: 451.7035 - mean_absolu
te_error: 17.5727 - val_loss: 4271.5996 - val_mean_absolute_error: 64.8632
Epoch 53/100
9/9 [==============================] - 0s 7ms/step - loss: 441.8223 - mean_absolu
te_error: 17.3747 - val_loss: 4347.6704 - val_mean_absolute_error: 65.4627
Epoch 54/100
9/9 [==============================] - 0s 7ms/step - loss: 431.0551 - mean_absolu
te_error: 16.9942 - val_loss: 4053.0364 - val_mean_absolute_error: 63.1513
Epoch 55/100
9/9 [==============================] - 0s 7ms/step - loss: 428.9161 - mean_absolu
te_error: 17.0171 - val_loss: 4379.4521 - val_mean_absolute_error: 65.7225
Epoch 56/100
9/9 [==============================] - 0s 8ms/step - loss: 403.1098 - mean_absolu
te_error: 16.3051 - val_loss: 5183.2734 - val_mean_absolute_error: 71.5631
Epoch 57/100
9/9 [==============================] - 0s 7ms/step - loss: 407.5589 - mean_absolu
te_error: 16.3882 - val_loss: 3812.4392 - val_mean_absolute_error: 61.2555
Epoch 58/100
9/9 [==============================] - 0s 7ms/step - loss: 342.1129 - mean_absolu
te_error: 14.9692 - val_loss: 3605.8318 - val_mean_absolute_error: 59.5499
Epoch 59/100
9/9 [==============================] - 0s 7ms/step - loss: 320.8546 - mean_absolu
te_error: 14.5354 - val_loss: 3508.8926 - val_mean_absolute_error: 58.7357
Epoch 60/100
9/9 [==============================] - 0s 8ms/step - loss: 287.9976 - mean_absolu
te_error: 13.6328 - val_loss: 3289.3884 - val_mean_absolute_error: 56.8475
```

```
Epoch 61/100
9/9 [==============================] - 0s 8ms/step - loss: 255.7273 - mean_absolu
te_error: 12.8063 - val_loss: 3237.7224 - val_mean_absolute_error: 56.4222
Epoch 62/100
9/9 [==============================] - 0s 8ms/step - loss: 241.1995 - mean_absolu
te_error: 12.2083 - val_loss: 2866.4360 - val_mean_absolute_error: 53.0037
Epoch 63/100
9/9 [==============================] - 0s 8ms/step - loss: 260.8766 - mean_absolu
te_error: 12.8794 - val_loss: 2715.6733 - val_mean_absolute_error: 51.5608
Epoch 64/100
9/9 [==============================] - 0s 9ms/step - loss: 244.7468 - mean_absolu
te_error: 12.6482 - val_loss: 2601.1252 - val_mean_absolute_error: 50.4634
Epoch 65/100
9/9 [==============================] - 0s 8ms/step - loss: 206.3157 - mean_absolu
te_error: 11.4693 - val_loss: 2691.6296 - val_mean_absolute_error: 51.3764
Epoch 66/100
9/9 [==============================] - 0s 9ms/step - loss: 185.8740 - mean_absolu
te_error: 10.7023 - val_loss: 2471.2925 - val_mean_absolute_error: 49.1967
Epoch 67/100
9/9 [==============================] - 0s 8ms/step - loss: 166.6731 - mean_absolu
te_error: 10.2396 - val_loss: 2406.4248 - val_mean_absolute_error: 48.5419
Epoch 68/100
9/9 [==============================] - 0s 8ms/step - loss: 148.3090 - mean_absolu
te_error: 9.3204 - val_loss: 2211.1604 - val_mean_absolute_error: 46.4815
Epoch 69/100
9/9 [==============================] - 0s 8ms/step - loss: 129.7939 - mean_absolu
te_error: 8.8026 - val_loss: 2045.8628 - val_mean_absolute_error: 44.6644
Epoch 70/100
9/9 [==============================] - 0s 7ms/step - loss: 129.4979 - mean_absolu
te_error: 8.7955 - val_loss: 1980.0404 - val_mean_absolute_error: 43.9485
Epoch 71/100
9/9 [==============================] - 0s 8ms/step - loss: 104.5596 - mean_absolu
te_error: 7.6858 - val_loss: 1756.9685 - val_mean_absolute_error: 41.3011
Epoch 72/100
9/9 [==============================] - 0s 7ms/step - loss: 97.1692 - mean_absolut
e_error: 7.4681 - val_loss: 1676.6454 - val_mean_absolute_error: 40.3345
Epoch 73/100
9/9 [==============================] - 0s 8ms/step - loss: 93.3857 - mean_absolut
e_error: 7.4364 - val_loss: 1535.8431 - val_mean_absolute_error: 38.5534
Epoch 74/100
9/9 [==============================] - 0s 9ms/step - loss: 78.9485 - mean_absolut
e_error: 6.7651 - val_loss: 1586.9152 - val_mean_absolute_error: 39.2578
Epoch 75/100
9/9 [==============================] - 0s 8ms/step - loss: 76.9378 - mean_absolut
e_error: 6.6727 - val_loss: 1427.3835 - val_mean_absolute_error: 37.1514
Epoch 76/100
9/9 [==============================] - 0s 8ms/step - loss: 68.3872 - mean_absolut
e_error: 6.3653 - val_loss: 1309.3983 - val_mean_absolute_error: 35.5196
Epoch 77/100
9/9 [==============================] - 0s 8ms/step - loss: 64.6755 - mean_absolut
e_error: 6.1723 - val_loss: 1211.6750 - val_mean_absolute_error: 34.1087
Epoch 78/100
9/9 [==============================] - 0s 8ms/step - loss: 64.5083 - mean_absolut
e_error: 6.2552 - val_loss: 1141.9634 - val_mean_absolute_error: 33.0823
Epoch 79/100
9/9 [==============================] - 0s 7ms/step - loss: 62.1198 - mean_absolut
e_error: 6.1707 - val_loss: 1071.3477 - val_mean_absolute_error: 31.9988
Epoch 80/100
9/9 [==============================] - 0s 8ms/step - loss: 63.4881 - mean_absolut
e_error: 6.2693 - val_loss: 1082.7872 - val_mean_absolute_error: 32.2171
```

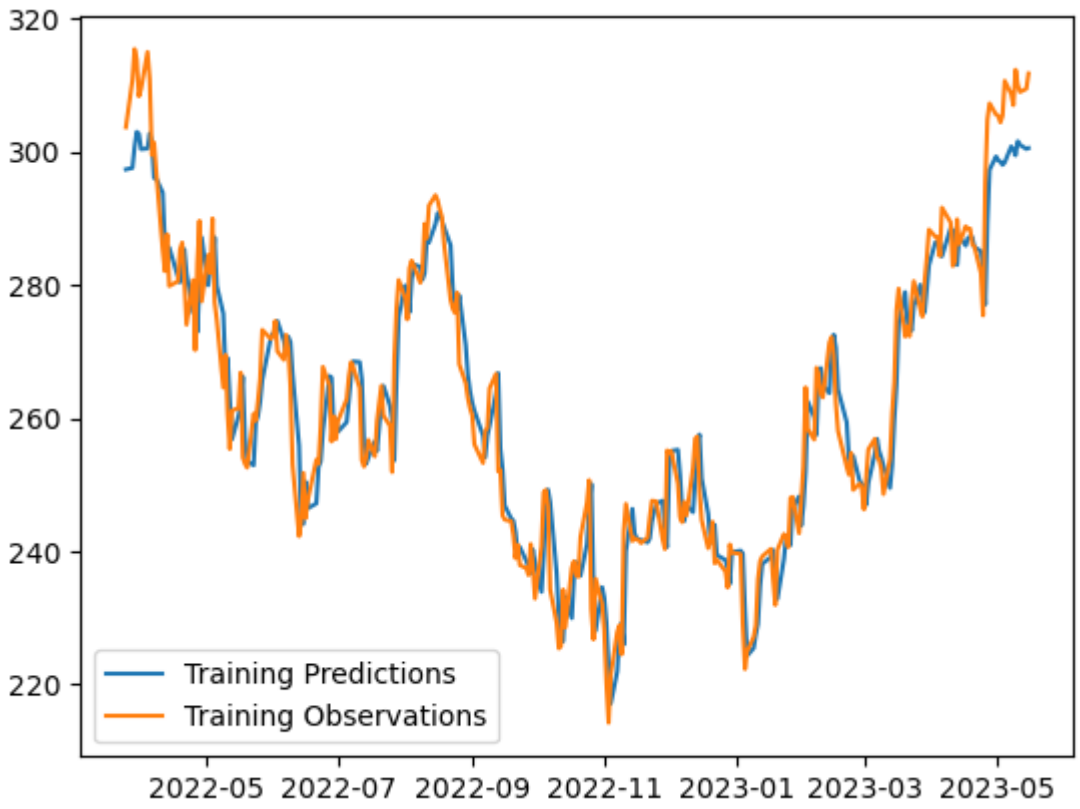
```
Epoch 81/100
9/9 [==============================] - 0s 9ms/step - loss: 52.7728 - mean_absolut
e_error: 5.6269 - val_loss: 999.5151 - val_mean_absolute_error: 30.8794
Epoch 82/100
9/9 [==============================] - 0s 9ms/step - loss: 54.6579 - mean_absolut
e_error: 5.6657 - val_loss: 1018.6453 - val_mean_absolute_error: 31.2376
Epoch 83/100
9/9 [==============================] - 0s 9ms/step - loss: 50.9907 - mean_absolut
e_error: 5.5694 - val_loss: 875.9238 - val_mean_absolute_error: 28.7988
Epoch 84/100
9/9 [==============================] - 0s 8ms/step - loss: 55.5080 - mean_absolut
e_error: 5.9787 - val_loss: 920.1172 - val_mean_absolute_error: 29.6155
Epoch 85/100
9/9 [==============================] - 0s 7ms/step - loss: 47.1564 - mean_absolut
e_error: 5.3461 - val_loss: 879.2391 - val_mean_absolute_error: 28.9134
Epoch 86/100
9/9 [==============================] - 0s 9ms/step - loss: 49.0415 - mean_absolut
e_error: 5.4557 - val_loss: 792.6069 - val_mean_absolute_error: 27.3472
Epoch 87/100
9/9 [==============================] - 0s 8ms/step - loss: 51.8662 - mean_absolut
e_error: 5.7013 - val_loss: 802.5610 - val_mean_absolute_error: 27.5648
Epoch 88/100
9/9 [==============================] - 0s 8ms/step - loss: 53.8337 - mean_absolut
e_error: 5.8043 - val_loss: 864.9278 - val_mean_absolute_error: 28.7352
Epoch 89/100
9/9 [==============================] - 0s 8ms/step - loss: 48.6195 - mean_absolut
e_error: 5.5337 - val_loss: 724.9432 - val_mean_absolute_error: 26.1041
Epoch 90/100
9/9 [==============================] - 0s 8ms/step - loss: 44.3444 - mean_absolut
e_error: 5.2291 - val_loss: 756.7995 - val_mean_absolute_error: 26.7508
Epoch 91/100
9/9 [==============================] - 0s 9ms/step - loss: 43.7969 - mean_absolut
e_error: 5.2164 - val_loss: 717.1555 - val_mean_absolute_error: 25.9957
Epoch 92/100
9/9 [==============================] - 0s 9ms/step - loss: 44.6434 - mean_absolut
e_error: 5.3852 - val_loss: 679.4049 - val_mean_absolute_error: 25.2556
Epoch 93/100
9/9 [==============================] - 0s 8ms/step - loss: 41.2598 - mean_absolut
e_error: 5.1230 - val_loss: 641.1443 - val_mean_absolute_error: 24.4593
Epoch 94/100
9/9 [==============================] - 0s 8ms/step - loss: 41.2248 - mean_absolut
e_error: 5.0353 - val_loss: 686.7388 - val_mean_absolute_error: 25.4407
Epoch 95/100
9/9 [==============================] - 0s 8ms/step - loss: 49.3096 - mean_absolut
e_error: 5.6365 - val_loss: 585.1838 - val_mean_absolute_error: 23.2907
Epoch 96/100
9/9 [==============================] - 0s 9ms/step - loss: 47.6976 - mean_absolut
e_error: 5.3944 - val_loss: 603.1674 - val_mean_absolute_error: 23.7138
Epoch 97/100
9/9 [==============================] - 0s 9ms/step - loss: 49.3545 - mean_absolut
e_error: 5.5182 - val_loss: 681.5872 - val_mean_absolute_error: 25.3901
Epoch 98/100
9/9 [==============================] - 0s 8ms/step - loss: 43.5390 - mean_absolut
e_error: 5.2359 - val_loss: 616.8290 - val_mean_absolute_error: 24.0467
Epoch 99/100
9/9 [==============================] - 0s 7ms/step - loss: 39.4075 - mean_absolut
e_error: 4.9666 - val_loss: 584.0346 - val_mean_absolute_error: 23.3289
Epoch 100/100
9/9 [==============================] - 0s 7ms/step - loss: 39.9053 - mean_absolut
e_error: 4.9912 - val_loss: 586.4802 - val_mean_absolute_error: 23.4058
```

Out[15]: <keras.src.callbacks.History at 0x1b4f10a0a50>

In [16]:
```
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'])
```

```
9/9 [==============================] - 0s 2ms/step
```

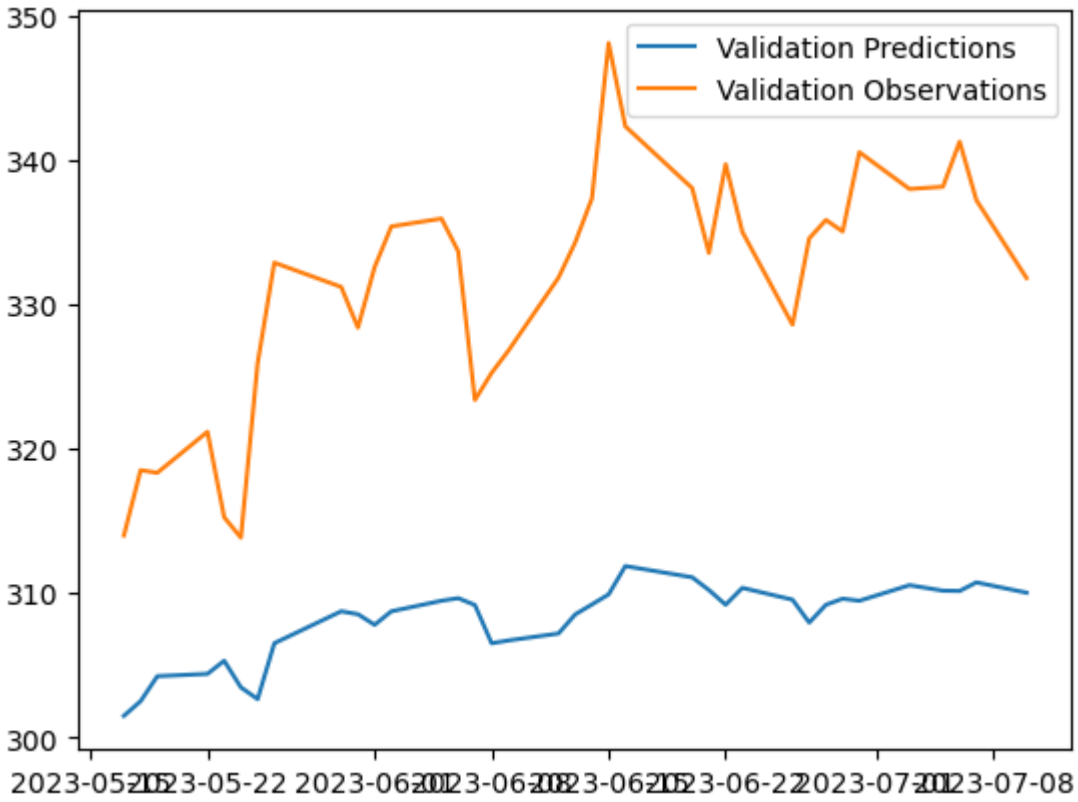
Out[16]: <matplotlib.legend.Legend at 0x1b4f348c1d0>



In [17]:
```
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'])
```

```
2/2 [==============================] - 0s 2ms/step
```

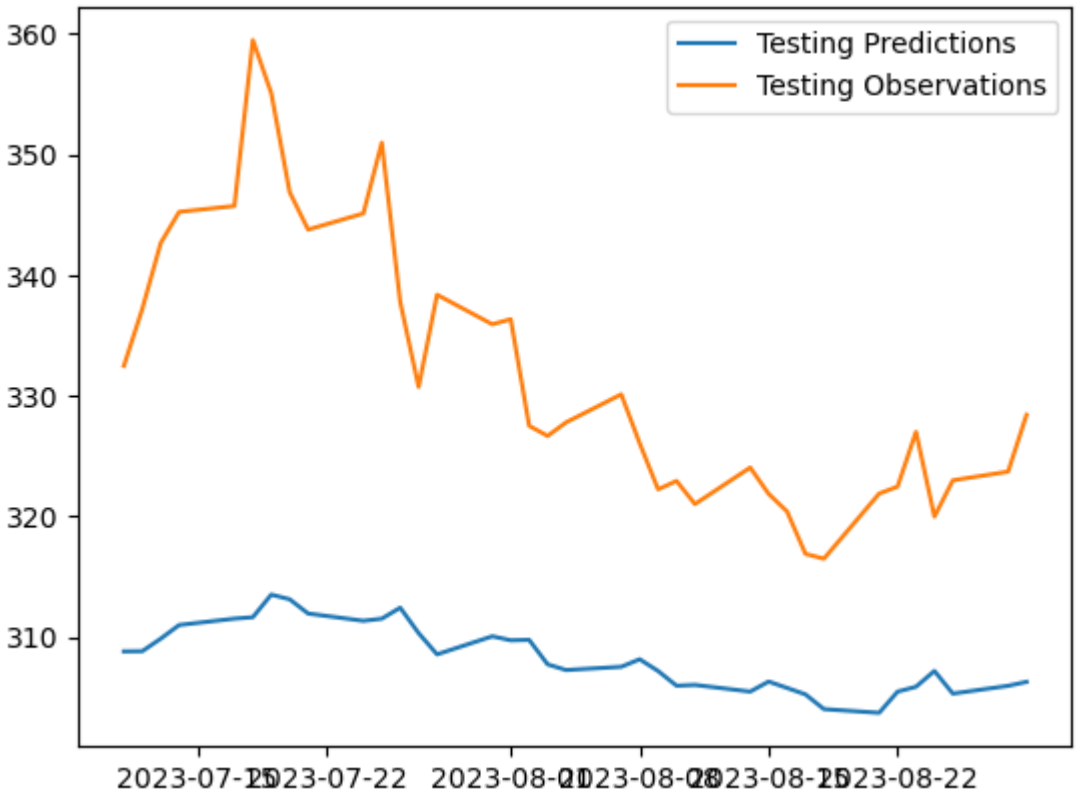
Out[17]: <matplotlib.legend.Legend at 0x1b4f357f410>



In [18]:
```
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'])
```

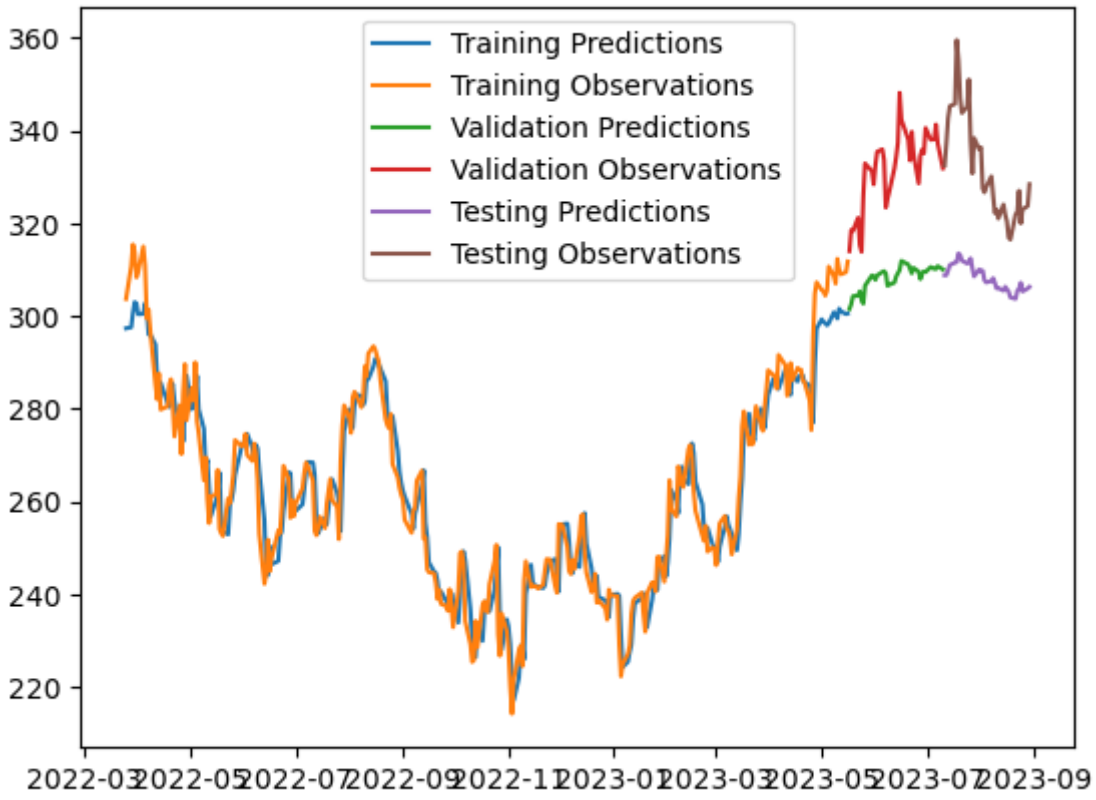
```
2/2 [==============================] - 0s 4ms/step
```

Out[18]:
```
<matplotlib.legend.Legend at 0x1b4f45e0490>
```



In [19]:
```
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[19]: <matplotlib.legend.Legend at 0x1b4f4650f50>



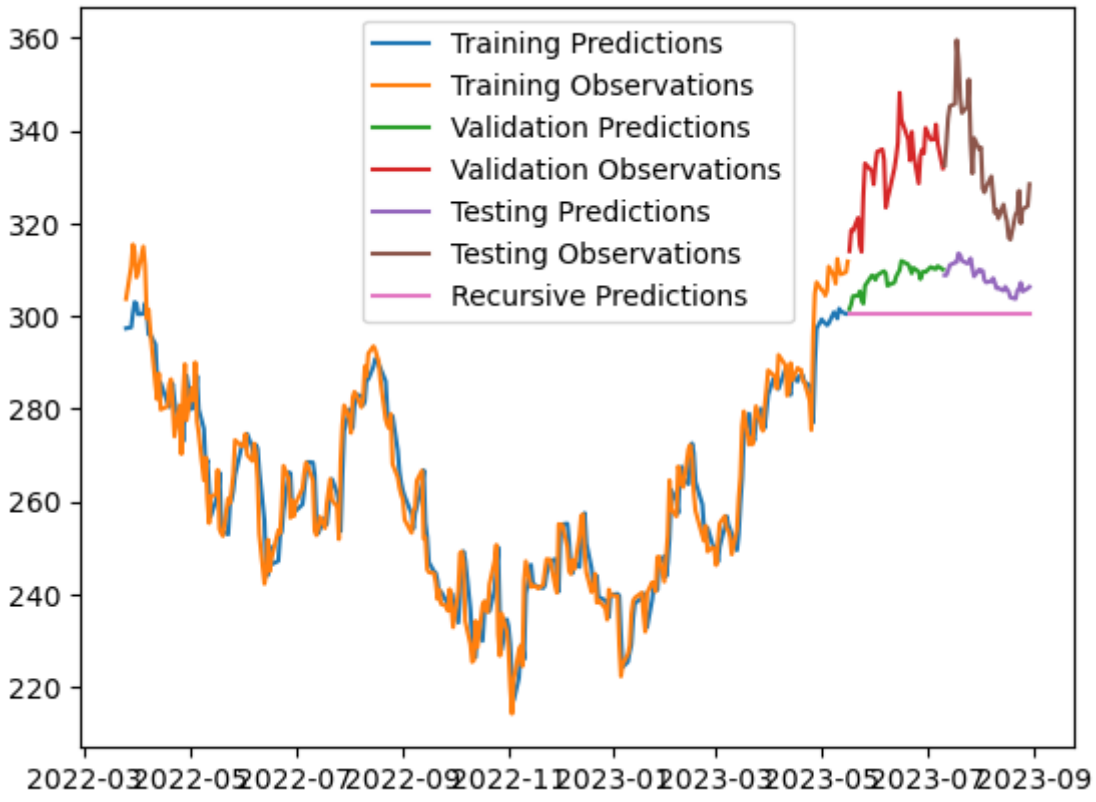
In [20]:
```
from copy import deepcopy

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 28ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 24ms/step
```

```
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 17ms/step
```

In [21]:
```
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[21]: <matplotlib.legend.Legend at 0x1b4f46f9710>



In [ ]: