In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import yfinance as yf
import warnings
import numpy as np
warnings.filterwarnings("ignore")
```

In [2]:
```
df=yf.download("AAPL",period='5y')
```

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

In [3]:
```
df=pd.DataFrame(df)
```

In [4]:
```
df.head()
```

Out[4]:

| | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| **Date** | | | | | | |
| **2018-10-08 00:00:00-04:00** | 55.552502 | 56.200001 | 55.049999 | 55.942501 | 53.578018 | 118655600 |
| **2018-10-09 00:00:00-04:00** | 55.910000 | 56.817501 | 55.562500 | 56.717499 | 54.320263 | 107564000 |
| **2018-10-10 00:00:00-04:00** | 56.365002 | 56.587502 | 54.012501 | 54.090000 | 51.803814 | 167962400 |
| **2018-10-11 00:00:00-04:00** | 53.630001 | 54.875000 | 53.080002 | 53.612499 | 51.346493 | 212497600 |
| **2018-10-12 00:00:00-04:00** | 55.105000 | 55.720001 | 54.209999 | 55.527500 | 53.180557 | 161351600 |

In [5]:
```
df1=df[['Adj Close']]
```

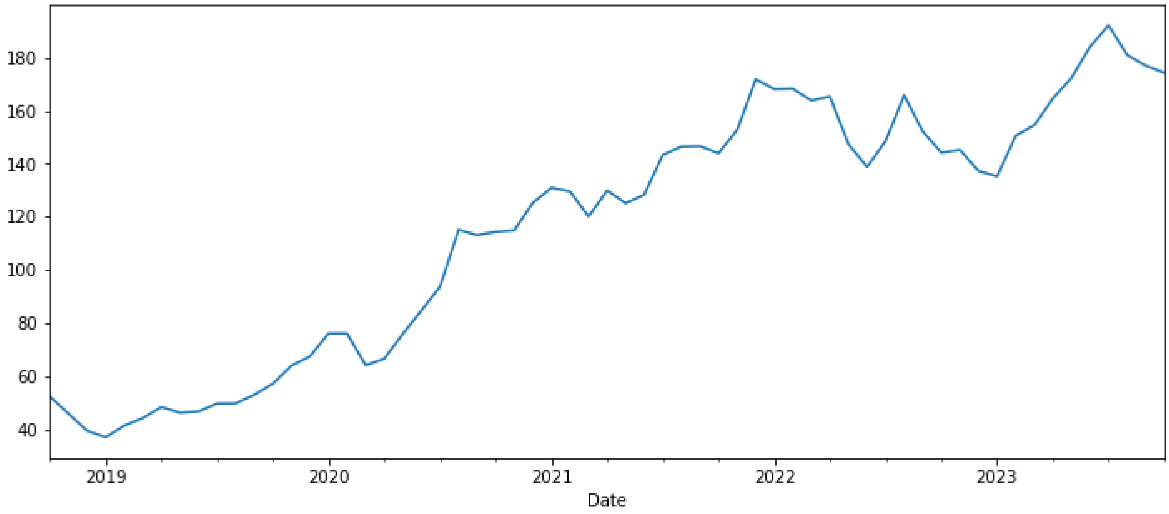
In [6]:
```
df1=df1['Adj Close'].resample('MS').mean()
```

In [7]:
```
df1.head()
```

Out[7]:
```
Date
2018-10-01 00:00:00-04:00    52.390161
2018-11-01 00:00:00-04:00    45.906409
2018-12-01 00:00:00-05:00    39.468077
2019-01-01 00:00:00-05:00    37.041332
2019-02-01 00:00:00-05:00    41.391336
Freq: MS, Name: Adj Close, dtype: float64
```

In [8]:
```
df1.plot(figsize=(12, 5), legend=False)
```

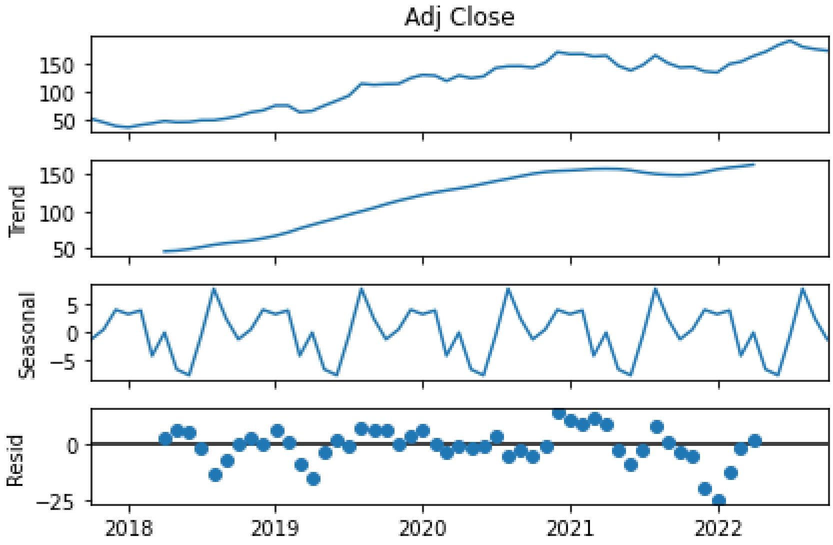
Out[8]:
```
<AxesSubplot:xlabel='Date'>
```



### *Stationary Check*

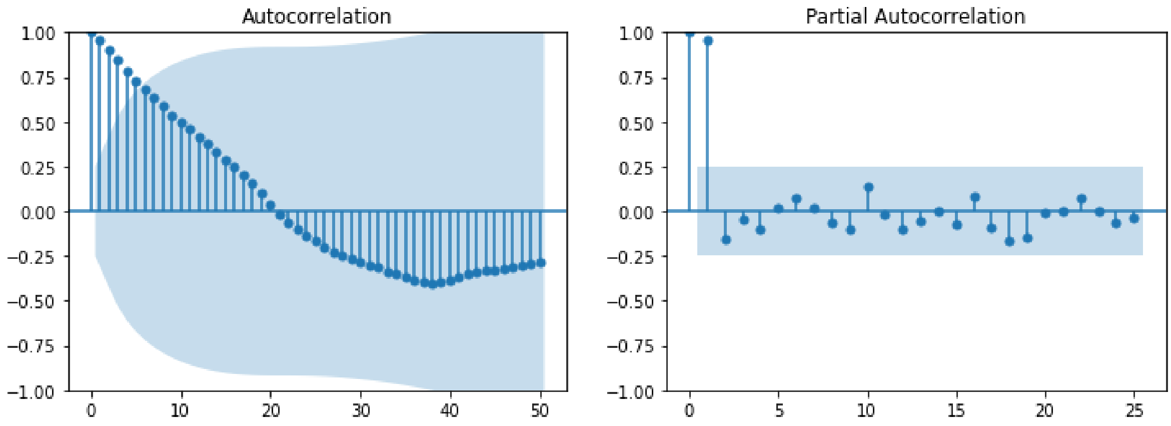
In [9]:
```
import statsmodels.api as sm
```

In [10]:
```
decomposition=sm.tsa.seasonal_decompose(df1,model='additive')
decomposition.plot().show()
```



In [11]:

```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
plot_acf(df1, lags=50, ax=ax1)
plot_pacf(df1, lags=25, ax=ax2)
plt.show()
```



In [12]:

```
from statsmodels.tsa.stattools import adfuller
```

In [13]:

```
adftest=adfuller(df1)
print('pvalue of adfuller test is:', adftest[1])
```

```
pvalue of adfuller test is: 0.8320560253704592
```

### *Remove Stationarity*

In [14]:

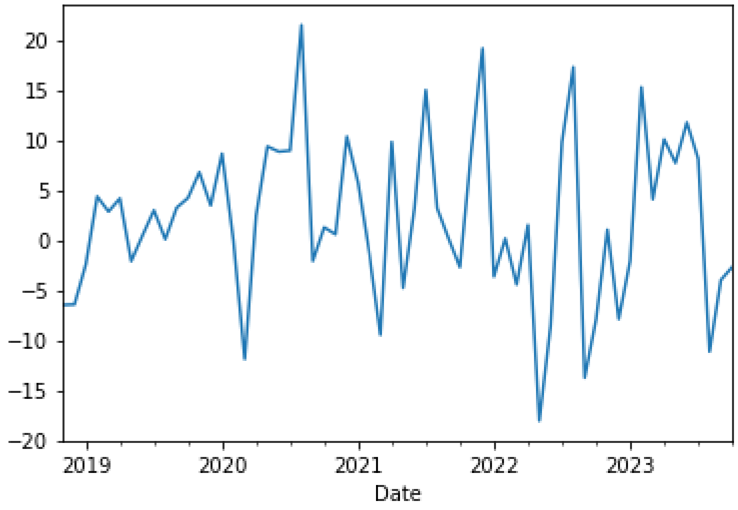
```
diff_data=df1.diff().dropna()
```

In [15]:

```
diff_data.plot()
```

Out[15]:

```
<AxesSubplot:xlabel='Date'>
```



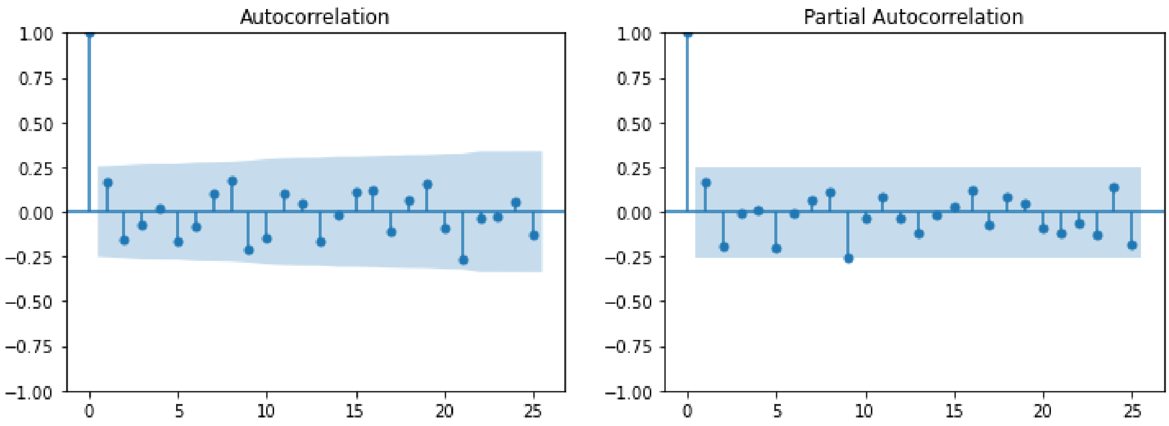
In [16]:

```
adftest2=adfuller(diff_data)
print('pvalue of adfuller test is:', adftest2[1])
```

```
pvalue of adfuller test is: 1.835220163491212e-08
```

### *Plot ACF and PACF*

In [17]:
```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
plot_acf(diff_data, lags=25, ax=ax1)
plot_pacf(diff_data, lags=25, ax=ax2)
plt.show()
```



### *Train Test Split*

In [18]:
```
len(df1)
```

Out[18]:
```
61
```

In [19]:
```
train=df1[:48]
test=df1[48:]
```

### *Find P, D, Q Order Values*

In [30]:

```
import itertools
from sklearn.metrics import mean_squared_error
from statsmodels.tsa.arima.model import ARIMA

p = range(0,5)
d = range(0,2)
q = range(0,5)

pdq_combination = list(itertools.product(p, d, q))

print('No of PDQ combinations: ', len(pdq_combination))

best_rmse = float('inf')
best_order = None

for pdq in pdq_combination:
    try:
        model = ARIMA(train, order=pdq).fit()
        pred = model.predict(start=len(train), end=(len(df1) - 1))
        error = np.sqrt(mean_squared_error(test, pred))
        if error < best_rmse:
            best_rmse = error
            best_order = pdq
    except:
        continue

print('Best PDQ order:', best_order)
print('Best RMSE:', best_rmse)
```

```
No of PDQ combinations:  50
```

```
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "

Best PDQ order: (4, 1, 4)
Best RMSE: 13.79443991215171
```

```
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
```

In [21]:

```
import pmdarima as pm
auto_arima=pm.auto_arima(train,stepwise=False,seasonal=False)
auto_arima
```

Out[21]:

```
ARIMA(order=(0, 1, 0), scoring_args={}, suppress_warnings=True)
```

### *Arima Model*

In [22]:

```
from statsmodels.tsa.arima.model import ARIMA
```

In [26]:

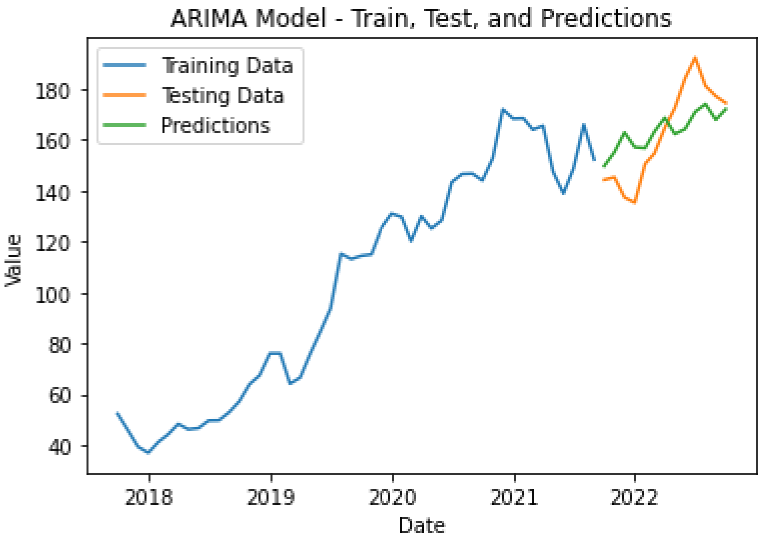
```
model = ARIMA(train, order=(4,1,4)).fit()
pred=model.predict(start=len(train),end=(len(df1)-1))
error=np.sqrt(mean_squared_error(test,pred))
print(error)
```

```
13.79443991215171
```

```
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
```

In [27]:

```
plt.plot(train.index, train, label='Training Data')
plt.plot(test.index, test, label='Testing Data')
plt.plot(test.index, pred, label='Predictions')
plt.legend()
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('ARIMA Model - Train, Test, and Predictions')
plt.show()
```



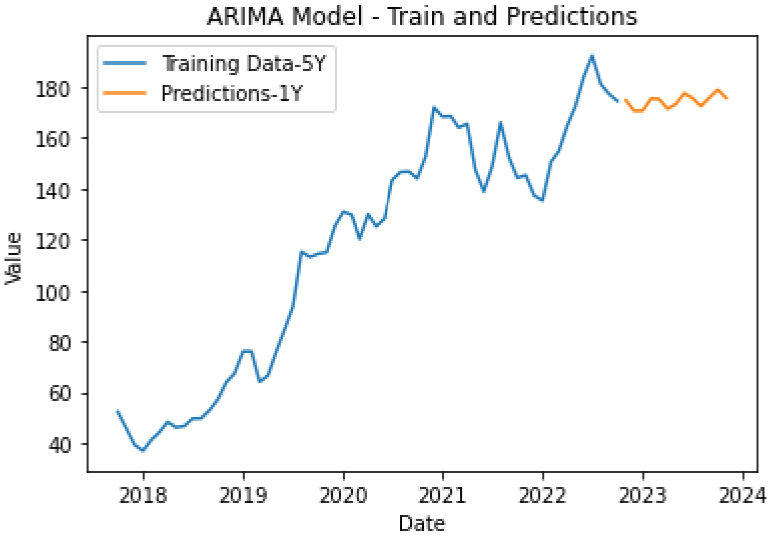
### *Predict Future Data*

In [28]:

```
#Training and predicting entire 5Years Data
final_model=ARIMA(df1,order=(4,1,4)).fit()
prediction=final_model.predict(start=len(df1),end=len(df1)+12)

#plot Graph
plt.plot(df1.index, df1, label='Training Data-5Y')
plt.plot(prediction.index, prediction, label='Predictions-1Y')
plt.legend()
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('ARIMA Model - Train and Predictions')
plt.show()
```

```
C:\vinay\Anaconda\lib\site-packages\statsmodels\base\model.py:607: Convergen
ceWarning: Maximum Likelihood optimization failed to converge. Check mle_ret
vals
  warnings.warn("Maximum Likelihood optimization failed to "
```


ARIMA Model - Train and Predictions
Training Data-5Y
Predictions-1Y
Value
Date