

Nama : Elen Yunisa

Nim : 2202020052

Kelas : A2 Informatika/ 4

## Prediksi pendapatan penjualan jpmorgan

### 1. Import library

```
import pandas as pd
import datetime
from datetime import date, timedelta
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.graphics.tsaplots import plot_pacf
import statsmodels.api as sm
import warnings

[ ] df=pd.read_csv("JPMorgan Chase (2023 - 02.05.2024).csv")
```

### 2. Untuk menggambarkan pendapatan penjualan (harga penutupan disesuaikan) JP Morgan selama periode waktu tertentu

```
330 2024-04-26 193.570007 194.869995 193.059998 193.490005 193.490005 6413700
331 2024-04-29 193.479996 194.259995 192.429993 193.279999 193.279999 5387800
332 2024-04-30 192.809998 194.990005 191.639999 191.740005 191.740005 8153700
333 2024-05-01 192.270004 194.460007 190.789993 191.860001 191.860001 7445300
334 rows x 7 columns

#pendapatan penjualan JPMorgan
import plotly.express as px
figure = px.line(df, x="Date", y="Adj Close", title="Pendapatan penjualan JPMorgan kuartal")
figure.show()
```

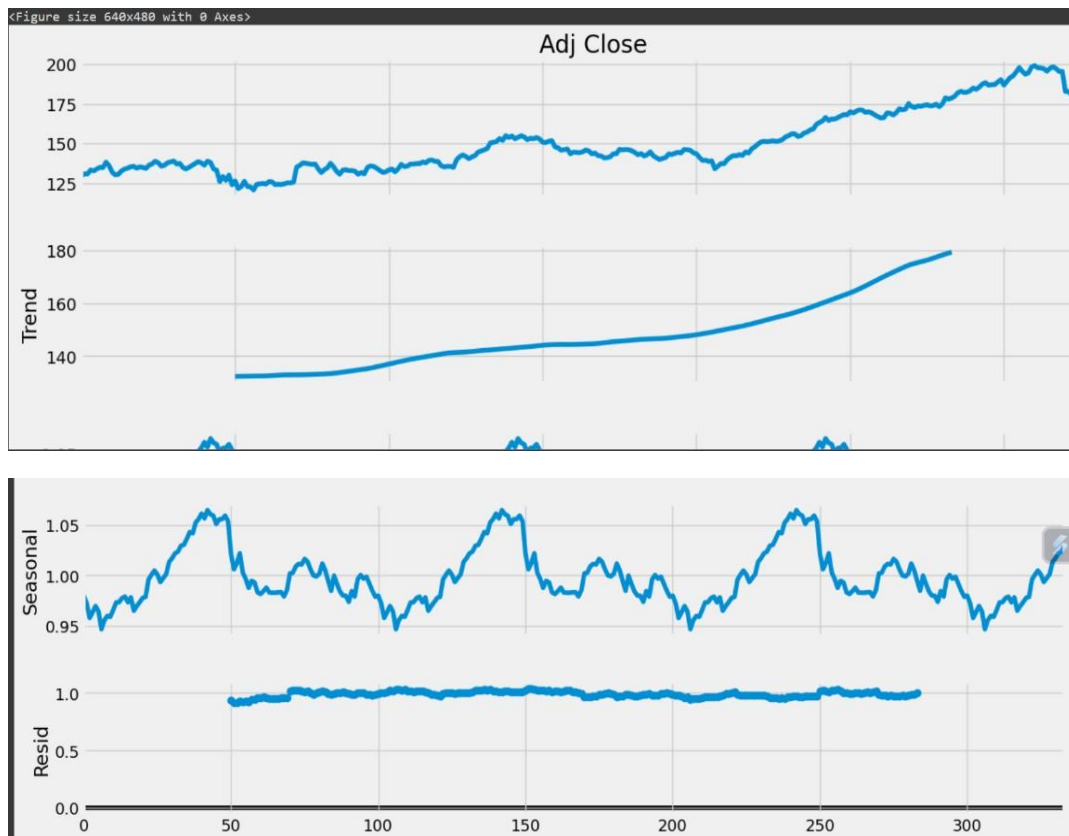
Pendapatan penjualan JPMorgan kuartal



### 3. Melakukan dekomposisi musiman pada data harga penutupan yang disesuaikan dari JPMorgan, membagi data menjadi komponen tren, musiman, dan residu, kemudian menampilkan grafik dari komponen-komponen tersebut dengan ukuran yang lebih besar. Ini membantu dalam

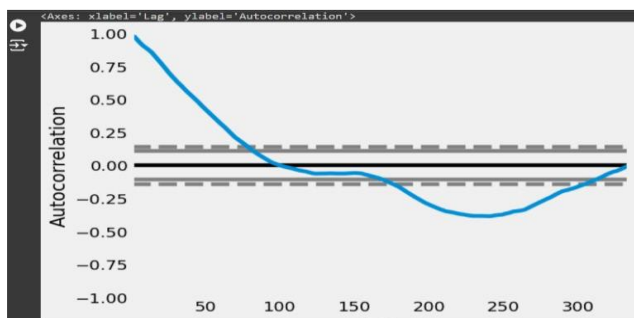
Mengidentifikasi pola musiman, tren, dan fluktuasi yang tidak dapat dijelaskan oleh model musiman.

```
#pendaptan sifatnya musiman dan naik turun , kategori seaonal
#cek grafik
result = seasonal_decompose(df["Adj Close"], model='multiplicative', period=100)
fig = plt.figure()
fig = result.plot()
fig.set_size_inches(15,10)
```



#### 4. Menacar nilai p,d, dan q untuk membuat model ARIMA

```
#jelas menggunakan seasonal ARIMA (musiman)
#maka menentukan d = 1
# mencari p dan q
#mencari nilai p dengan autocorrelation
pd.plotting.autocorrelation_plot(df["Adj Close"])
```



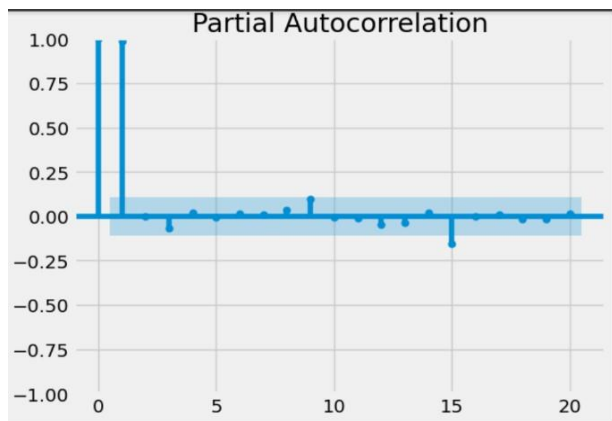
```

d = 1 #karena seasonal
p = 6 #karena ada 6 garis di lewati hingga naik

] #mencari nilai q menggunakan autocorellation partial
plot_pacf(df["Adj Close"], lags =20)
#20 lebar sumbu x

```

5. Dari  $q = 2$  karena ada 2 garis jatuh yang melewati garis yang lain.



```

[ ] q = 2 #ada 2 garis terjauh

```

6. Membuat model SARIMAX berdasarkan data pendapatan (adj close). Model ini memperhitungkan komponen musiman dan non musiman dalam data.

```

#membuat model
#12 month
from statsmodels.tools.sm_exceptions import ConvergenceWarning
warnings.simplefilter('ignore', ConvergenceWarning)
model = sm.tsa.statespace.SARIMAX(df["Adj Close"],
                                order=(p,d,q),
                                seasonal_order=(p,d,q,12))
model = model.fit()
print(model.summary())

```

**SARIMAX Results**

Dep. Variable:	Adj Close	No. Observations:	334
Model:	SARIMAX(6, 1, 2)x(6, 1, 2, 12)	Log Likelihood:	-658.192
Date:	Mon, 10 Jun 2024	AIC:	1350.384
Time:	02:35:53	BIC:	1414.499
Sample:	- 334	HQIC:	1375.984

Covariance Type: opg

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	-1.3320	0.192	-6.954	0.000	-1.707	-0.957
ar.L2	-0.6189	0.219	-2.830	0.005	-1.047	-0.190
ar.L3	0.2271	0.111	2.042	0.041	0.009	0.445
ar.L4	0.1168	0.120	0.970	0.332	-0.119	0.353
ar.L5	-0.0896	0.111	-0.803	0.422	-0.308	0.129
ar.L6	-0.0788	0.068	-1.161	0.246	-0.212	0.054
ma.L1	1.3580	0.187	7.256	0.000	0.991	1.725
ma.L2	0.8084	0.191	4.229	0.000	0.434	1.183
ar.S.L12	-0.3439	0.368	-0.934	0.350	-1.066	0.378
ar.S.L24	0.1423	0.127	1.120	0.263	-0.107	0.391
ar.S.L36	0.0056	0.090	0.062	0.951	-0.172	0.183
ar.S.L48	-0.1618	0.090	-1.792	0.073	-0.339	0.015
ar.S.L60	-0.1758	0.125	-1.407	0.159	-0.421	0.069
ar.S.L72	0.1398	0.135	1.038	0.299	-0.124	0.404

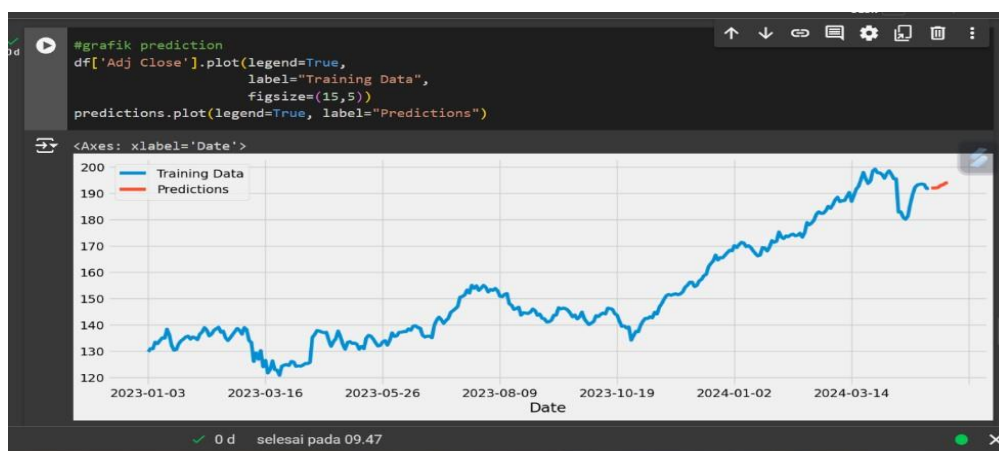
7. Menghasilkan prediksi pendapatan JPMorgan untuk 8 kuartal ke depan menggunakan model prediksi yang telah dilatih sebelumnya

```
[ ] #menentukan prediksi pendapatan JPMorgan 8 kuartal ke depan
predictions = model.predict(len(df), len(df)+7)
print(predictions)

334    191.956415
335    191.926281
336    191.971625
337    192.193045
338    192.966914
339    193.192461
340    193.786671
341    194.120330
Name: predicted_mean, dtype: float64
```

## 8. Grafik prediction

membuat grafik yang menampilkan data historis dan prediksi pendapatan JPMorgan



## 9. KESIMPULAN

Proses prediksi pendapatan penjualan JPMorgan melibatkan beberapa langkah seperti mengimpor library yang diperlukan, menggambarkan pendapatan penjualan menggunakan harga penutupan yang disesuaikan, melakukan dekomposisi musiman untuk mengidentifikasi komponen tren, musiman, dan residu, serta membuat model SARIMAX untuk memprediksi pendapatan JPMorgan selama 8 kuartal ke depan. Hasil analisis ini divisualisasikan dalam bentuk grafik yang menampilkan data historis dan prediksi pendapatan tersebut.