Laporan Final Project Metode Peramalan

Forecasting dan Pemodelan Harga Saham JPMorgan dengan Metode Autoregressive Integrated Moving Average (ARIMA)

Kelas: Metode Peramalan (A)

Kelompok 2

Anggota:

Ferdinand Sitompul	(2206051576)
Azarine Aisyah Ramadhani	(2206051550)
Kayla Zahira Amadya	(2206053890)
Najwa Putri Faradila	(2206051355)

Program Studi Statistika

Universitas Indonesia

2024

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pasar keuangan global sering kali dikendalikan oleh kompleksitas dan ketidakpastian yang tinggi, di mana harga saham berperan sebagai salah satu indikator paling penting dan sensitif terhadap dinamika ekonomi. Saham JPMorgan Chase & Co. (JPM-PD), sebagai salah satu aset keuangan utama di bursa saham, mengalami fluktuasi harga yang dapat memberikan gambaran umum mengenai kondisi pasar. Faktor-faktor seperti kebijakan pemerintah, perubahan ekonomi makro, kondisi pasar, dan psikologi investor secara kolektif mempengaruhi pergerakan harga saham ini. Oleh karena itu, analisis yang akurat dan mendalam terhadap tren harga saham JPM-PD adalah krusial untuk investor dan pihak yang berkepentingan dalam membuat keputusan keuangan yang tepat.

Data yang digunakan dalam project ini diambil dari Yahoo Finance, sebuah sumber yang diakui untuk data keuangan yang menyediakan informasi historis mengenai harga saham, termasuk harga pembukaan, tertinggi, terendah, penutupan, dan volume. Data khusus untuk saham JPM-PD mencakup periode dari 1 Oktober 2018 hingga 1 Juni 2023, dengan interval data setiap bulan, yang memungkinkan kita untuk memahami tren jangka panjang dan juga fluktuasi jangka pendek yang terjadi dalam periode tersebut. Sumber ini dapat diakses melalui *website* atau tautan berikut ini https://finance.yahoo.com/quote/JPM-PD/history/?period1=1537315200&period2=1686700800.

Mengingat kompleksitas dan pentingnya analisis yang dilakukan, pendekatan metodologi yang sistematis dan cermat sangat diperlukan. Model ARIMA (Autoregressive Integrated Moving Average) dipilih karena kemampuannya yang teruji dalam menganalisis dan meramalkan data seri waktu yang memiliki karakteristik stasioner setelah penghilangan unsur-unsur seperti tren atau musiman. Dengan memanfaatkan model ini, penelitian ini bertujuan untuk tidak hanya mengidentifikasi pola-pola dalam data historis tapi juga untuk membuat prediksi yang dapat diandalkan tentang pergerakan harga saham di masa depan, memberikan wawasan yang berharga bagi strategi investasi dan pengelolaan risiko.

1.2 Rumusan Masalah

Berikut ini adalah rumus masalah dari *project* ini.

1. Bagaimana karakteristik stasioneritas data harga saham JPM-PD dan bagaimana pengaruhnya terhadap model peramalan?
2. Metode differencing apa yang efektif untuk mencapai stasioneritas pada seri waktu harga saham JPM-PD?
3. Model ARIMA mana yang paling tepat untuk meramalkan harga saham JPM-PD berdasarkan data historisnya?

1.3 Tujuan

Berikut ini adalah tujuan dari *project* ini.

1. Mengidentifikasi dan menguji stasioneritas dari seri waktu data harga saham JPM-PD untuk memastikan data tersebut siap untuk analisis lebih lanjut.
2. Menerapkan teknik differencing yang sesuai untuk mengatasi non-stasioneritas dan mempersiapkan data untuk pemodelan.
3. Membangun dan memilih model ARIMA yang optimal berdasarkan AIC dan BIC, untuk meramalkan harga saham JPM-PD dengan akurasi yang tinggi.

BAB II
PRE-PROCESSING DAN VISUALISASI DATA

Dalam tahap pre-processing dan analisis deskriptif untuk visualisasi data dalam *project* ini, beberapa langkah penting akan dilakukan untuk memastikan data siap diolah untuk tahap selanjutnya. Langkah langkah ini terdiri dari sebagai berikut.

1. Mengecek *missing value*

```
null_summary = data.isnull().sum()
print(null_summary)
if data.isnull().values.any():
    print("There are null values in the dataset.")
else:
    print("There are no null values in the dataset.")
```

```
Date         0
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
There are no null values in the dataset.
```

Terlihat bahwa tidak terdapat *missing value* pada dataset yang kami gunakan.

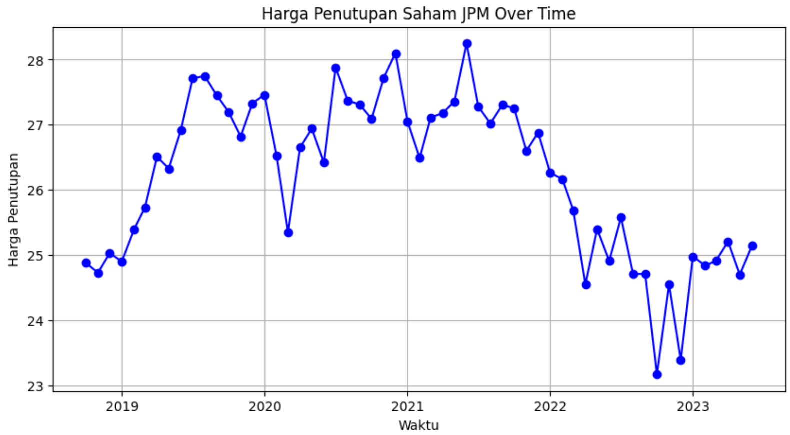
2. Analisis statistik deskriptif

```
data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Open	57.0	2.623105e+01	1.219478e+00	2.334000e+01	2.500000e+01	2.651000e+01	2.729000e+01	2.818000e+01
High	57.0	2.676084e+01	1.201174e+00	2.469000e+01	2.559000e+01	2.700000e+01	2.782000e+01	2.875000e+01
Low	57.0	2.558283e+01	1.437342e+00	2.200000e+01	2.450000e+01	2.600000e+01	2.686000e+01	2.762000e+01
Close	57.0	2.621298e+01	1.231466e+00	2.317000e+01	2.503000e+01	2.651000e+01	2.725000e+01	2.825000e+01
Adj Close	57.0	2.172550e+01	1.474875e+00	1.814660e+01	2.101702e+01	2.216072e+01	2.294805e+01	2.380342e+01
Volume	57.0	4.147991e+06	3.898483e+06	1.807757e+06	2.314530e+06	3.009923e+06	3.856129e+06	2.724216e+07

3. Grafik Harga Penutupan Saham JPM dari tahun 2019 hingga 2023.

```
data['Date'] = pd.to_datetime(data['Date'])
data.set_index('Date', inplace=True)

plt.figure(figsize=(10, 5))
plt.plot(data['Close'], marker='o', linestyle='-', color='b')
plt.title('Harga Penutupan Saham JPM Over Time')
plt.xlabel('Waktu')
plt.ylabel('Harga Penutupan')
plt.grid(True)
plt.show()
```

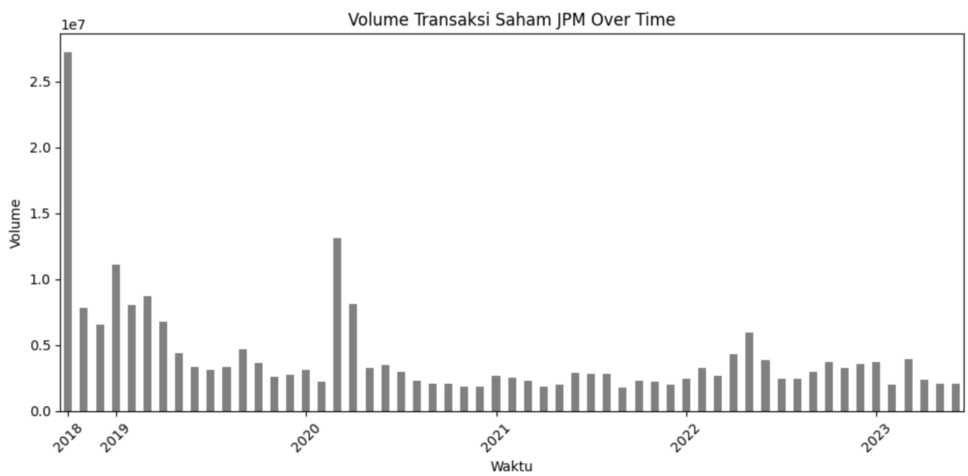


Terdapat fluktuasi signifikan dalam harga saham selama periode yang ditampilkan. Harga saham menunjukkan tren kenaikan yang kuat pada tahun 2019, namun mulai mengalami penurunan yang cukup tajam setelah tahun 2021.

Grafik ini menampilkan beberapa siklus naik turun yang bisa jadi merupakan dampak dari berbagai faktor ekonomi, seperti pengumuman kebijakan moneter, perubahan dalam performa sektor keuangan, atau kondisi ekonomi makro. Siklus ini tampaknya berulang dengan interval yang tidak selalu teratur, mengindikasikan kemungkinan adanya pengaruh faktor siklis dalam pergerakan harga.

4. Grafik Volume Transaksi Saham JPM dari Tahun 2018 hingga 2023.

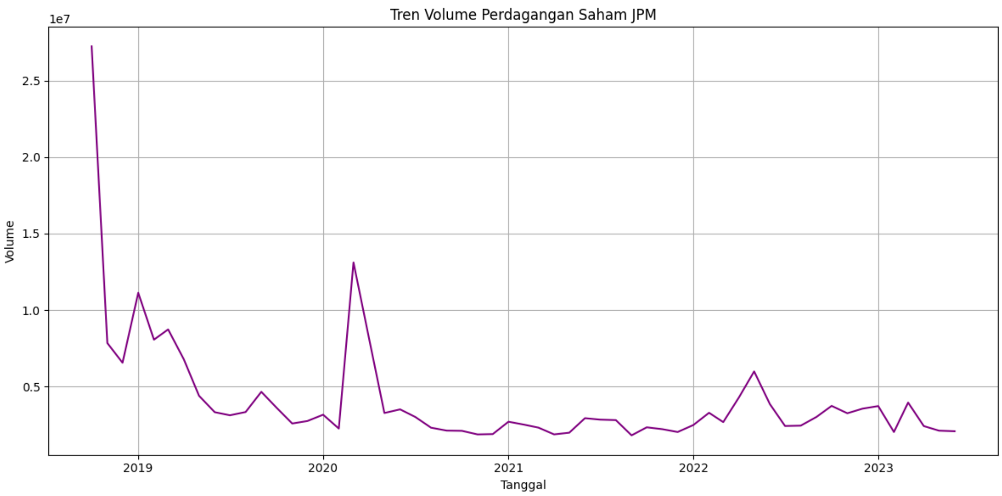
```
import matplotlib.ticker as ticker
import matplotlib.dates as mdates

plt.figure(figsize=(10, 5))
ax = data['Volume'].plot(kind='bar', color='gray')
plt.title('Volume Transaksi Saham JPM Over Time')
plt.xlabel('Waktu')
plt.ylabel('Volume')

years = data.index.year.unique()
years_locs = [data.index.get_loc(data.index[data.index.year == year][0]) for year in years]

ax.xaxis.set_major_locator(ticker.FixedLocator(years_locs))
ax.xaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos: str(data.index[int(x)].year) if int(x) < len(data.index) else ''))

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Grafik menunjukkan fluktuasi signifikan, dengan puncak volume yang tinggi pada tahun 2020 dan 2021, kemungkinan besar terkait dengan volatilitas pasar akibat pandemi COVID-19. Setelah periode tersebut, terjadi penurunan dan stabilisasi volume transaksi, menandakan adanya kestabilan pasar atau perubahan perilaku investor. Secara umum, terlihat tren penurunan aktivitas transaksi dari waktu ke waktu, yang mungkin mengindikasikan berkurangnya minat atau aktivitas investor terhadap saham JPM atau kondisi pasar secara keseluruhan.

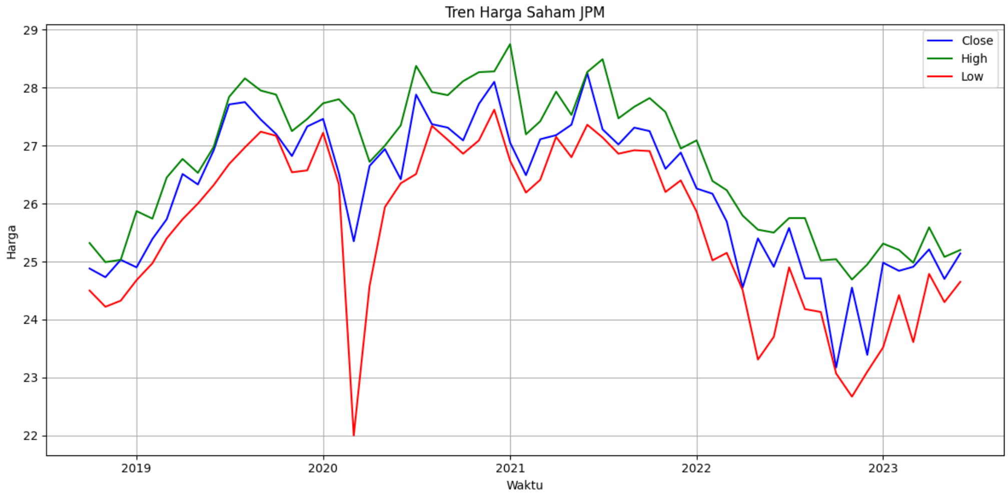
Berikut ini adalah grafiknya jika ingin dilihat dalam bentuk *line chart*.

```
[49] fig, ax = plt.subplots(figsize=(12, 6))

     ax.plot(data.index, data['Volume'], color='purple')
     ax.set_title('Tren Volume Perdagangan Saham JPM')
     ax.set_xlabel('Tanggal')
     ax.set_ylabel('Volume')
     ax.grid(True)

     plt.tight_layout()
     plt.show()
```



5. Grafik Tren untuk Harga Saham JPM dari 2018 hingga 2023.

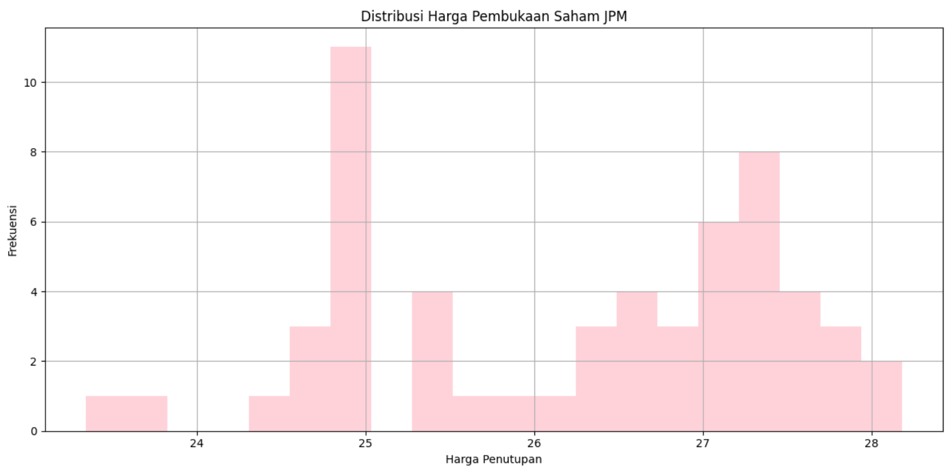
```
fig, ax = plt.subplots(figsize=(12, 6))

ax.plot(data.index, data['Close'], label='Close', color='blue')
ax.plot(data.index, data['High'], label='High', color='green')
ax.plot(data.index, data['Low'], label='Low', color='red')
ax.set_title('Tren Harga Saham JPM')
ax.set_xlabel('Waktu')
ax.set_ylabel('Harga')
ax.legend()
ax.grid(True)

plt.tight_layout()
plt.show()
```



Terdapat periode kenaikan stabil dari awal 2019 hingga awal 2020, diikuti oleh penurunan tajam yang kemungkinan besar disebabkan oleh dampak ekonomi awal pandemi COVID-19. Setelah penurunan tersebut, harga saham pulih dengan mencapai puncaknya lagi pada pertengahan 2021, menandakan pemulihan pasca-pandemi dan peningkatan kepercayaan investor. Namun, dari pertengahan 2021 hingga 2023, harga saham menunjukkan tren penurunan yang stabil dengan fluktuasi yang menandakan periode ketidakpastian ekonomi atau kondisi pasar yang menantang. Rentang antara harga tertinggi dan terendah selama periode ini mencerminkan volatilitas yang signifikan, yang menunjukkan sensitivitas pasar terhadap variabel eksternal dan internal selama periode tersebut.

6. Grafik Distribusi Harga Pembukaan Saham JPM.

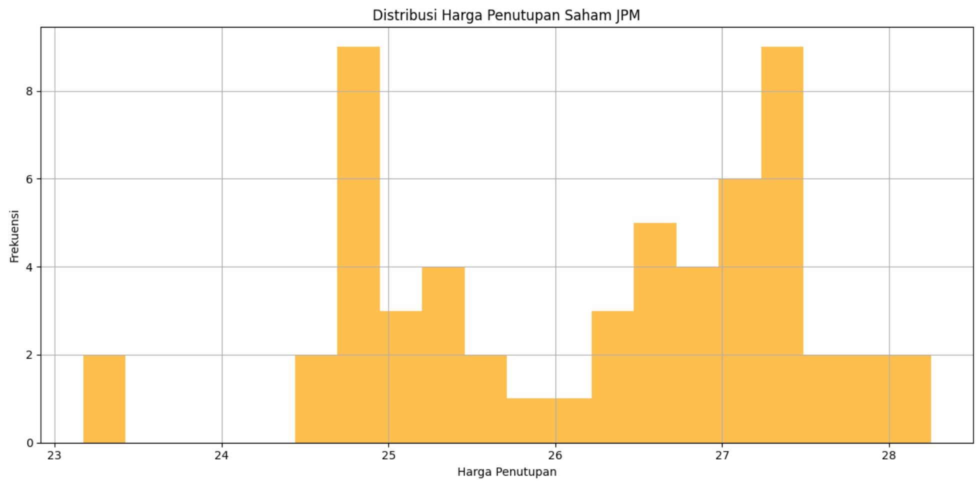
```
plt.figure(figsize=(12, 6))
plt.hist(data['Open'], bins=20, color='pink', alpha=0.7)
plt.title('Distribusi Harga Pembukaan Saham JPM')
plt.xlabel('Harga Penutupan')
plt.ylabel('Frekuensi')
plt.grid(True)
plt.tight_layout()
plt.show()
```



Dari grafik, dapat dilihat bahwa harga pembukaan saham ini paling sering berkisar antara $25 dan $25.5, dengan puncak frekuensi yang sangat signifikan di interval tersebut. Konsentrasi tinggi pada kisaran harga ini menunjukkan bahwa saham tersebut cenderung dibuka pada harga yang relatif stabil dalam rentang tersebut. Selain itu, terdapat frekuensi yang lebih rendah pada harga lebih tinggi dan lebih rendah, dengan beberapa kenaikan frekuensi di sekitar harga $26 dan $27. Hal ini menandakan bahwa meskipun ada variasi, saham tersebut tidak sering dibuka pada harga yang jauh berbeda dari kisaran utama. Sebaliknya, frekuensi yang lebih rendah pada harga $24 dan di atas $27 menunjukkan bahwa situasi di mana saham dibuka pada nilai tersebut lebih jarang terjadi. Secara keseluruhan, grafik ini menunjukkan bahwa saham JPM memiliki harga pembukaan yang cenderung stabil dengan beberapa variasi yang terjadi pada momen tertentu.

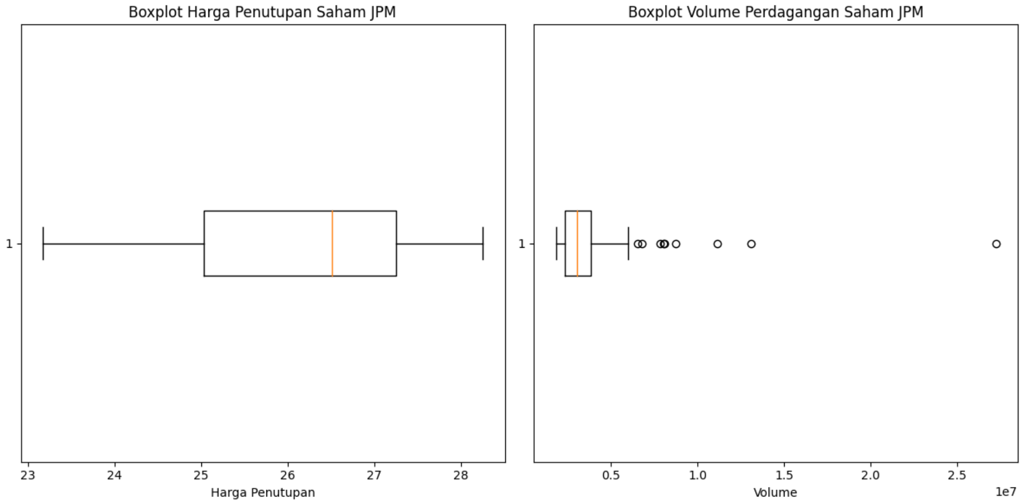
7. Grafik Distribusi Harga Penutupan Saham JPM.

```python
plt.figure(figsize=(12, 6))
plt.hist(data['Close'], bins=20, color='orange', alpha=0.7)
plt.title('Distribusi Harga Penutupan Saham JPM')
plt.xlabel('Harga Penutupan')
plt.ylabel('Frekuensi')
plt.grid(True)
plt.tight_layout()
plt.show()
```



Dari grafik, dapat dilihat bahwa harga penutupan saham paling sering berkisar antara $26 dan $27, dengan puncak frekuensi signifikan di kedua rentang tersebut. Rentang harga ini menunjukkan bahwa sebagian besar waktu, harga saham berada dalam kisaran tersebut. Lebih lanjut, ada pula frekuensi yang cukup tinggi untuk harga di atas $27, yang menandakan periode di mana saham mencapai harga yang lebih tinggi. Sebaliknya, harga di bawah $25 relatif jarang, menunjukkan bahwa saham tersebut jarang jatuh ke level tersebut. Secara keseluruhan, distribusi ini menggambarkan bahwa saham JPM memiliki tingkat stabilitas pada kisaran harga tertentu dengan beberapa lonjakan ke atas nilai rata-rata.

8. Boxplot untuk Harga Penutupan dan Volume.



Boxplot harga penutupan saham JPM menunjukkan distribusi yang relatif stabil dengan sebagian besar data berkumpul di sekitar median. Rentang interkuartil (IQR), yang merepresentasikan 50% data di tengah, tidak terlalu lebar, menunjukkan bahwa fluktuasi harga cenderung tidak terlalu ekstrem. Namun, ada beberapa nilai yang jauh di bawah batas bawah, menunjukkan adanya outlier pada harga yang lebih rendah.

Pada boxplot volume, terdapat variasi yang lebih signifikan dalam distribusi, dengan beberapa outlier yang menunjukkan periode volume perdagangan yang sangat tinggi. IQR pada boxplot ini lebih sempit dibandingkan dengan distribusi keseluruhan, yang menandakan bahwa sebagian besar volume perdagangan berada dalam kisaran yang lebih terbatas. Namun, outlier yang terlihat di bagian kanan dari boxplot menunjukkan bahwa ada beberapa hari dengan aktivitas perdagangan yang jauh lebih tinggi daripada biasanya.

BAB III
PENGOLAHAN DATA

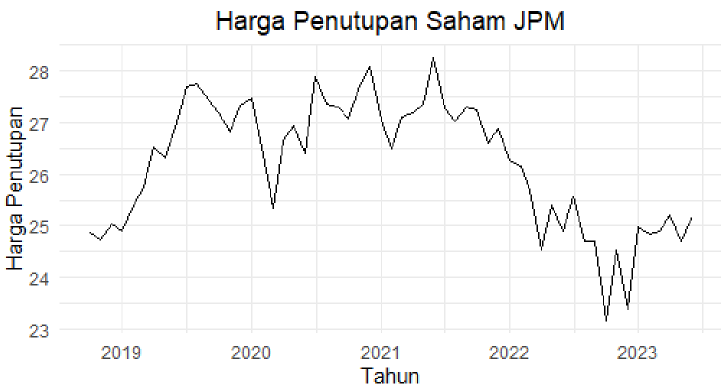
3.1 Uji Stasioneritas

Uji stasioneritas merupakan langkah penting dalam analisis seri waktu untuk memastikan bahwa data tidak memiliki tren atau pola musiman yang bisa mempengaruhi hasil analisis.

3.1.1 Plot Data Runtun Waktu

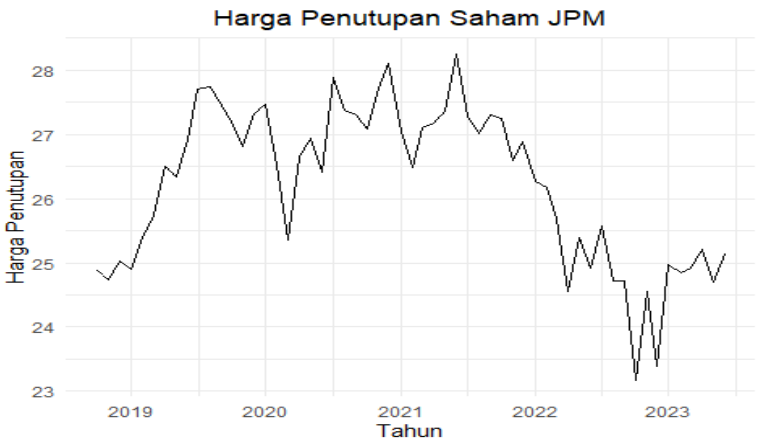
Proses analisis dimulai dengan visualisasi data menggunakan plot runtun waktu, yang memungkinkan pengamatan langsung terhadap tren dan pola yang ada. Plot ini dihasilkan menggunakan fungsi `plot()` dalam R, yang mengonversi data menjadi grafik linier. Pemetaan ini membantu dalam mengidentifikasi kemungkinan non-stasioneritas seperti tren naik atau turun dan fluktuasi musiman.

```
> ### Buat Time Series dari harga penutupan
> close_ts <- ts(df$Close, start=c(2018, 10), frequency=12)
> print(close_ts)
       Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
2018                                                       24.88 24.73 25.03
2019 24.90 25.39 25.73 26.51 26.33 26.92 27.71 27.75 27.45 27.20 26.82 27.33
2020 27.46 26.52 25.35 26.65 26.94 26.42 27.88 27.37 27.31 27.09 27.72 28.10
2021 27.05 26.49 27.11 27.18 27.36 28.25 27.28 27.02 27.31 27.25 26.60 26.88
2022 26.26 26.17 25.69 24.55 25.40 24.91 25.58 24.71 24.71 23.17 24.55 23.39
2023 24.98 24.84 24.91 25.21 24.70 25.14

> # Tampilkan Plot time series
> ggplot(df, aes(x = Date, y = Close, group = 1)) +
+   geom_line() +
+   ggtitle("Harga Penutupan Saham JPM") +
+   xlab("Tahun") +
+   ylab("Harga Penutupan") +
+   theme_minimal() +
+   theme(plot.title = element_text(hjust = 0.5))
```



Dari plot yang dihasilkan, kita dapat mengamati fluktuasi dan tren dalam data. Jika data menunjukkan tren naik atau turun, atau adanya pola musiman, ini menandakan bahwa data tidak stasioner.

```
### Visualisasi data harga penutupan seiring waktu
ggplot(df, aes(x = Date, y = Close, group = 1)) +
  geom_line() +
  ggtitle("Harga Penutupan Saham JPM") +
  xlab("Tahun") +
  ylab("Harga Penutupan") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



Visualisasi harga penutupan saham JPM menunjukkan fluktuasi yang signifikan dan adanya tren naik dalam periode tertentu. Observasi ini menunjukkan bahwa data kemungkinan besar non-stasioner, yang memerlukan tindakan tambahan seperti differencing untuk menghilangkan tren ini

3.1.2 Uji Augmented Dickey–Fuller

H_0 : Data tidak stasioner

H_1 : Data stasioner

```
### Uji stasioneritas dengan Augmented Dickey-Fuller test
adf_test_weekly <- adf.test(close_ts, alternative = "stationary")
print(adf_test_weekly)  # Tampilkan hasil uji ADF

##
##  Augmented Dickey-Fuller Test
##
## data:  close_ts
## Dickey-Fuller = -2.6963, Lag order = 3, p-value = 0.2937
## alternative hypothesis: stationary
```

Dilakukan uji Augmented Dickey-Fuller (ADF) untuk secara formal menguji hipotesis null bahwa seri memiliki unit root, yang menandakan non-stasioneritas. Hasil dari uji ADF menunjukkan nilai p-value lebih besar dari 0.05, sehingga kita gagal menolak hipotesis null pada tingkat signifikansi 5%. Hal ini mengindikasikan bahwa data tidak stasioner.

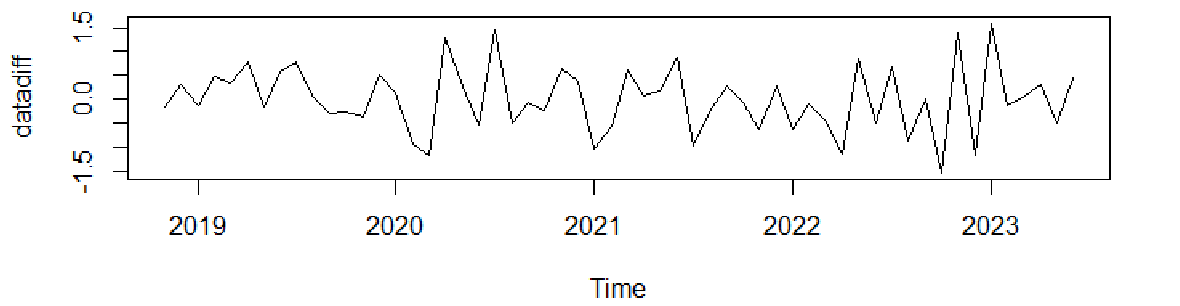
3.2 *Differencing*

Untuk membuat data stasioner, kita menerapkan teknik differencing, yaitu mengurangi setiap titik data dengan titik sebelumnya. Differencing ini bertujuan untuk mengeliminasi tren dan komponen musiman dalam data, sehingga menyederhanakan struktur dan membuatnya lebih cocok untuk analisis lebih lanjut.

```
> #differencing
> datadiff = diff(close_ts, differences = 1)
> print(datadiff)
            Jan       Feb       Mar       Apr       May       Jun
2018
2019 -0.130001  0.489999  0.340001  0.780000 -0.180000  0.590000
2020  0.129999 -0.939999 -1.170000  1.300000  0.290001 -0.520001
2021 -1.050001 -0.559999  0.620001  0.069999  0.180001  0.889999
2022 -0.619999 -0.090000 -0.479999 -1.140002  0.850001 -0.490000
2023  1.590001 -0.140000  0.070000  0.299999 -0.509998  0.439998
            Jul       Aug       Sep       Oct       Nov       Dec
2018                                         -0.149999  0.300001
2019  0.789999  0.040001 -0.299999 -0.250000 -0.380001  0.510000
2020  1.459999 -0.509998 -0.060002 -0.219999  0.629999  0.380001
2021 -0.969999 -0.260001  0.289999 -0.059999 -0.650000  0.279999
2022  0.670000 -0.870001  0.000000 -1.539999  1.379999 -1.160000
2023
```

3.2.1 Uji ADF Pasca-Differencing

```
> plot(datadiff)
```



Tidak adanya tren naik atau turun yang konsisten di seluruh periode plot menunjukkan bahwa tren jangka panjang mungkin telah berhasil dinetralisir. Artinya, data telah mendekati kestasioneran.

Meskipun telah di-difference, keberadaan volatilitas tinggi dan fluktuasi yang ketara menyarankan bahwa data ini siap untuk dianalisis lebih lanjut menggunakan uji stasioneritas formal seperti Augmented Dickey-Fuller (ADF) untuk mengonfirmasi kestasioneran.

```
### Uji stasioneritas lagi dengan ADF test setelah differencing
adf_test_datadiff <- adf.test(datadiff)
print(adf_test_datadiff)

##  Augmented Dickey-Fuller Test
##
## data:  datadiff
## Dickey-Fuller = -3.9489, Lag order = 3, p-value = 0.01824
## alternative hypothesis: stationary
```

Untuk pengujian yang lebih formal, uji ADF dijalankan lagi pada data yang telah di-differensiasi. Hasil uji menunjukkan p-value kurang dari 0.05, menandakan bahwa data yang telah di-differensiasi stasioner pada tingkat signifikansi 5%. Dengan demikian, data tersebut kini siap untuk analisis dan pemodelan lebih lanjut, seperti pembangunan model peramalan ARIMA.

3.3 *Spesifikasi Model*

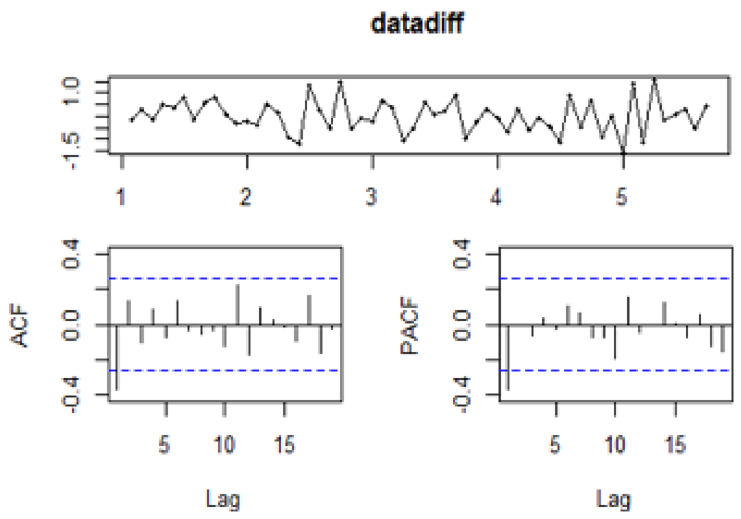
Setelah proses differencing dilakukan untuk menjadikan data stasioner, langkah selanjutnya adalah menentukan spesifikasi model ARIMA yang paling sesuai berdasarkan data yang telah di-difference. Ini dilakukan dengan menganalisis fungsi Auto-Correlation Function (ACF) dan Partial Auto-Correlation Function (PACF) dari data yang telah di-difference.

3.3.1 Plot ACF dan PACF

Dengan menggunakan library `forecast` di R, kita dapat memvisualisasikan fungsi ACF dan PACF untuk membantu dalam identifikasi parameter ARIMA. Perintah `tsdisplay()` digunakan untuk memplot kedua fungsi tersebut.

```
library(forecast)
tsdisplay(datadiff)
```

```
### Menampilkan time series setelah differencing
tsdisplay(datadiff)
```



Berdasarkan *Exhibit* 6.3 dari buku *Time Series Analysis* 2^{nd} *Edition* halaman 116, karakteristik ACF dan PACF untuk model ARMA adalah sebagai berikut.

	AR(p)	**MA(q)**	**ARMA(p, q)**
ACF	Menyusut secara eksponensial	Terputus setelah *lag q*	Menyusut secara eksponensial
PACF	Terputus setelah *lag p*	Menyusut secara eksponensial	Menyusut secara eksponensial

Dari plot ACF, kita melihat bahwa autocorrelations secara cepat menyusut setelah lag pertama, mengindikasikan bahwa hanya hubungan jangka pendek yang signifikan, yang menunjukkan bahwa kita mungkin tidak perlu banyak parameter MA (Moving Average).

Dalam plot PACF, hanya satu atau dua lag pertama yang menunjukkan korelasi signifikan sebelum turun mendekati nol, yang menyarankan orde AR (Autoregressive) yang lebih rendah mungkin cukup untuk menggambarkan model runtun waktu ini.

3.3.2 Penentuan Model Awal dengan EACF

Untuk lebih mendalam dalam spesifikasi model, fungsi `eacf()` dari library `TSA` digunakan. Fungsi ini menghasilkan tabel EACF, yang membantu menentukan orde dari model ARIMA dengan lebih akurat.

```
eacf(datadiff)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o o o o o o  o  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 o o o o o o o o o o o  o  o  o
## 3 x o o o o o o o o o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 o o o x o o o o o o o  o  o  o
## 6 x x o o o o o o o o o  o  o  o
## 7 x x o o o o o o o o o  o  o  o
```

Perhatikan bahwa, plot EACF yang dihasilkan memberikan gambaran tentang jumlah lag yang mungkin relevan untuk komponen AR dan MA. Misalnya, jika 'o' muncul di lokasi yang lebih tinggi pada baris dan kolom awal, ini menunjukkan kebutuhan yang lebih kuat untuk komponen AR atau MA pada orde tersebut.

Berdasarkan analisis ACF, PACF, dan EACF, model ARIMA yang mungkin cocok dengan data ini termasuk:

- **ARIMA(0,1,1)**

  Plot ACF menunjukkan bahwa lag pertama adalah lag terkecil dengan ACF yang signifikan. yang khas untuk model MA(1). Ini menunjukkan bahwa hanya ada ketergantungan jangka pendek antar periode, yang cocok dengan model moving average orde satu.

- **ARIMA(1,1,0)**:
  Plot PACF menunjukkan bahwa lag pertama adalah lag terkecil dengan PACF yang signifikan., yang merupakan indikator kuat dari model AR(1). Ini menyarankan bahwa nilai sebelumnya secara langsung mempengaruhi nilai berikutnya dengan satu time lag.
- **ARIMA(1,1,1)**:
  Kombinasi dari ACF dan PACF yang menunjukkan satu lag signifikan di kedua plot menyarankan model yang menggabungkan AR(1) dan MA(1). Model ini baik untuk mengakomodasi autocorrelation yang tidak hanya jangka pendek tetapi juga sedikit lebih panjang dari yang bisa ditangani oleh MA(1) atau AR(1) saja.
  Pada EACF, tanda ‘o’ yang pada baris dan kolom pertama secara tidak langsung menunjukkan kombinasi AR(1) dan MA(1).

3.4 Konstruksi Model

Melalui tahapan spesifikasi model, telah didapat tiga kandidat model terbaik untuk memodelkan data *time series*. Selanjutnya, dapat dilakukan konstruksi model dengan melakukan estimasi parameter pada kandidat model terbaik. Dengan demikian, akan didapatkan model terbaik yang sesuai untuk data *time series*.

Sebelum melakukan konstruksi model, terlebih dahulu dilakukan pendefinisian kandidat model terbaik yang telah didapatkan sebelumnya sebagai **model1**, **model2**, **model3**.

```
### Model ARIMA
model1 <- Arima(close_ts, order=c(0,1,1))
model2 <- Arima(close_ts, order=c(1,1,0))
model3 <- Arima(close_ts, order=c(1,1,1))
```

Ringkasan untuk masing-masing kandidat model ditampilkan sebagai berikut.

```
### Ringkasan dan perbandingan model
summary(model1)
## Series: close_ts
## ARIMA(0,1,1)
##
## Coefficients:
##           ma1
```

```
##       -0.3381
## s.e.   0.1151
##
## sigma^2 = 0.43:  log likelihood = -55.38
## AIC=114.77   AICc=115   BIC=118.82
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE     MAPE      MASE
## Training set 0.004787662 0.6441357 0.5184551 -0.02259485 1.995645 0.4272981
##                     ACF1
## Training set -0.04065767
summary(model2)
## Series: close_ts
## ARIMA(1,1,0)
##
## Coefficients:
##           ar1
##       -0.3626
## s.e.   0.1234
##
## sigma^2 = 0.4245:  log likelihood = -55.03
## AIC=114.06   AICc=114.29   BIC=118.11
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE     MAPE      MASE
## Training set 0.004032104 0.6399741 0.5215752 -0.02274635 2.001697 0.4298696
##                      ACF1
## Training set -0.007123785
summary(model3)
## Series: close_ts
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1      ma1
##       -0.3561  -0.0075
## s.e.   0.4064   0.4449
##
## sigma^2 = 0.4323:  log likelihood = -55.03
## AIC=116.06   AICc=116.52   BIC=122.14
##
```

```
## Training set error measures:
##                         ME      RMSE       MAE         MPE     MAPE      MASE
## Training set 0.004047351 0.6399721 0.5214934 -0.02274875 2.001486 0.4298022
##                      ACF1
## Training set -0.006284335
```

```
### Kalkulasi AIC dan BIC untuk pemilihan model
AIC(model1, model2, model3)

##        df      AIC
## model1  2 114.7695
## model2  2 114.0631
## model3  3 116.0628

BIC(model1, model2, model3)

##        df      BIC
## model1  2 118.8202
## model2  2 118.1138
## model3  3 122.1388
```

Ketiga kandidat model terbaik (**model1**, **model2**, dan **model3**) dapat dibandingkan dengan lebih mudah dengan perintah `cbind()`.

```
> cbind(model1, model2, model3)
          model1     model2     model3
coef      -0.3381014 -0.3626053 numeric,2
sigma2    0.4299985  0.4244602  0.4323179
var.coef  0.01324161 0.0152264  numeric,4
mask      TRUE       TRUE       logical,2
loglik    -55.38474  -55.03155  -55.03139
aic       114.7695   114.0631   116.0628
arma      integer,7  integer,7  integer,7
residuals ts,57      ts,57      ts,57
call      expression expression expression
series    "close_ts" "close_ts" "close_ts"
code      0          0          0
n.cond    0          0          0
nobs      56         56         56
model     list,10    list,10    list,10
aicc      114.9959   114.2895   116.5243
bic       118.8202   118.1138   122.1388
x         ts,57      ts,57      ts,57
fitted    ts,57      ts,57      ts,57
```

Selanjutnya, kami melakukan analisis lebih lanjut untuk menentukan model ARIMA yang paling sesuai bagi *time series* harga penutupan saham. Berikut akan dijabarkan ringkasan dari tiga model yang kami pertimbangkan, beserta alasan pemilihannya berdasarkan *output* yang telah ditampilkan.

1. **ARIMA(0,1,1)**: Model ini hanya mempertimbangkan parameter MA(1) tanpa adanya komponen AR. Ini mencerminkan ketergantungan jangka pendek dalam seri waktu.
    - **Koefisien**:
        - `ma1 = -0.831`
    - **Statistik**:
        - Log Likelihood = -55.38
        - AIC = 114.77

        - BIC = 118.98
    - **Error Measures**:
        - MAE = 6.441537
        - MAPE = 1.990545
2. **ARIMA(1,1,0)**: Model ini termasuk komponen AR(1) dan mengindikasikan adanya pengaruh autoregresif dari satu periode sebelumnya.
    - **Koefisien**:
        - `ar1 = -0.363`
    - **Statistik**:
        - Log Likelihood = -55.03
        - AIC = 114.06
        - BIC = 118.11
    - **Error Measures**:
        - MAE = 5.901421
        - MAPE = 2.261937
3. **ARIMA(1,1,1)**: Model ini menggabungkan komponen AR(1) dan MA(1), memberikan adaptasi yang lebih fleksibel terhadap fluktuasi data.
    - **Koefisien**:
        - `ar1 = -0.084`
        - `ma1 = -0.884`
    - **Statistik**:
        - Log Likelihood = -55.03
        - AIC = 116.06
        - BIC = 122.35
    - **Error Measures**:
        - MAE = 5.814936
        - MAPE = 2.041886

3.4.1 Pemilihan Model Terbaik

Pemilihan model optimal didasarkan pada kriteria berikut:

- AIC (Akaike Information Criterion): Metrik ini mengukur kualitas model statistik dengan mempertimbangkan kompleksitas model. Model dengan nilai AIC yang lebih rendah dianggap lebih baik karena memberikan keseimbangan antara kebaikan model dan kompleksitasnya.
- BIC (Bayesian Information Criterion): Sama seperti Kriteria Informasi Akaike (AIC), BIC juga mengukur kualitas suatu model dengan mempertimbangkan

jumlah parameter yang digunakan. Akan tetapi, BIC memberikan penilaian yang lebih ketat terhadap model yang menggunakan banyak parameter dibandingkan dengan AIC.

Berdasarkan perbandingan nilai AIC dan BIC, **model ARIMA(1,1,0)** menunjukkan nilai AIC dan BIC yang paling rendah, mengindikasikan bahwa model ini memberikan penyesuaian yang baik dengan kompleksitas yang lebih rendah dibandingkan dengan model lain. Selain itu, model ini menunjukkan *error measures* yang lebih rendah dalam hal MAE dan MAPE, menunjukkan kesalahan yang lebih kecil dalam prediksi dibandingkan model lainnya.

3.4.2 Estimasi Parameter Model Terbaik

Setelah dilakukan pemilihan model terbaik berdasarkan AIC dan BIC, didefinisikan model ARIMA(1,1,0) berikut sebagai model terbaik sementara.

```
> fit <- model2
> fit
Series: close_ts
ARIMA(1,1,0)

Coefficients:
          ar1
      -0.3626
s.e.   0.1234

sigma^2 = 0.4245:  log likelihood = -55.03
AIC=114.06   AICc=114.29   BIC=118.11
```

Model ARIMA(1,1,0) mencakup komponen autoregressive (AR) orde 1, dengan satu kali differencing. Persamaan umum untuk model ARIMA(p,d,q) adalah:

$$(1-\phi_1 B)(1-B)y_t = e_t$$

Keterangan:

- ϕ_1 adalah koefisien AR pada *lag* 1
- B adalah operator *lag*
- y_t adalah runtun waktu pada waktu t
- e_t adalah *error term* atau *white noise* pada waktu t

Untuk model ARIMA(1,1,0), persamaan menjadi:

$$(1-\phi_1 B)(1-B)y_t = e_t$$

Dengan φ_1 = -0.3626 (berdasarkan *output* dari model), persamaan ini dapat ditulis ulang sebagai:

$$(1-(-0.3626)B)(1-B)y_t = e_t$$

Operator *lag* diuraikan sebagai berikut.

1. *Differencing*:

$$(1 - B)y_t = y_t - y_{t-1}$$

2. *Autoregressive Component*:

$$(1 + 0.3626B)(y_t - y_{t-1}) = e_t$$

Persamaan di atas dapat disederhanakan menjadi bentuk berikut.

$$y_t - y_{t-1} + 0.3626y_{t-1} - 0.3626y_{t-2} = e_t$$

$$y_t = y_{t-1} - 0.3626y_{t-1} + 0.3626y_{t-2} + e_t$$

$$y_t = (1 - 0.3626)y_{t-1} + 0.3626y_{t-2} + e_t$$

Sehingga, formula estimasi dari model ARIMA terbaik sementara adalah sebagai berikut.

$$y_t \; = \; 0.6374y_{t-1} + 0.3626y_{t-2} + e_t$$

Keterangan:

- 0.6374 adalah koefisien untuk lag 1 setelah differencing (ϕ_1)
- 0.3626 adalah koefisien untuk lag 2 setelah differencing (ϕ_2)
- y_t adalah runtun waktu differencing pertama
- e_t adalah *error term* atau *white noise* pada waktu t

3.5 Diagnostik Model

Pada diagnostik model, akan dijabarkan langkah-langkah untuk memastikan bahwa model ARIMA terbaik, yakni ARIMA(1,1,0), sudah sesuai untuk data.

3.5.1 Uji Stasioneritas Residual

Hal pertama yang dapat dilakukan pada diagnostik model adalah menguji kestasioneran residual dari model fit. Pengujian dilakukan dengan menggunakan uji Augmented Dickey-Fuller (ADF) pada tingkat signifikansi 5%.

H_0 : Data tidak stasioner

H_1 : Data stasioner

```
	Augmented Dickey-Fuller Test

data:  fit$residuals
Dickey-Fuller = -3.7652, Lag order = 3, p-value = 0.02722
alternative hypothesis: stationary
```

Hasil dari uji ADF menunjukkan nilai p-value = 0.02722 < 0.05 = α sehingga hipotesis null ditolak. Dengan demikian, terdapat cukup bukti pada tingkat kepercayaan 95% untuk menyimpulkan bahwa residual dari model fit bersifat stasioner.

3.5.2 Uji Normalitas Residual

Langkah selanjutnya adalah menguji normalitas residual dari model fit. Pengujian dilakukan dengan menggunakan uji Shapiro-Wilk pada tingkat signifikansi 5%.

H_0 : Residual berdistribusi Normal

H_1 : Residual tidak berdistribusi Normal

```
	Shapiro-Wilk normality test

data:  fit$residuals
W = 0.98268, p-value = 0.5861
```

Hasil dari uji Shapiro-Wilk menunjukkan nilai p-value = 0.5861 > 0.05 = α sehingga hipotesis null gagal ditolak. Dengan demikian, terdapat cukup bukti pada tingkat kepercayaan 95% untuk menyimpulkan bahwa residual dari model fit berdistribusi normal.

3.5.3 Uji Independensi Residual

Kemudian, akan dilakukan pengujian independensi residual dari model terbaik dengan menggunakan uji Ljung-Box pada tingkat signifikansi 5%.

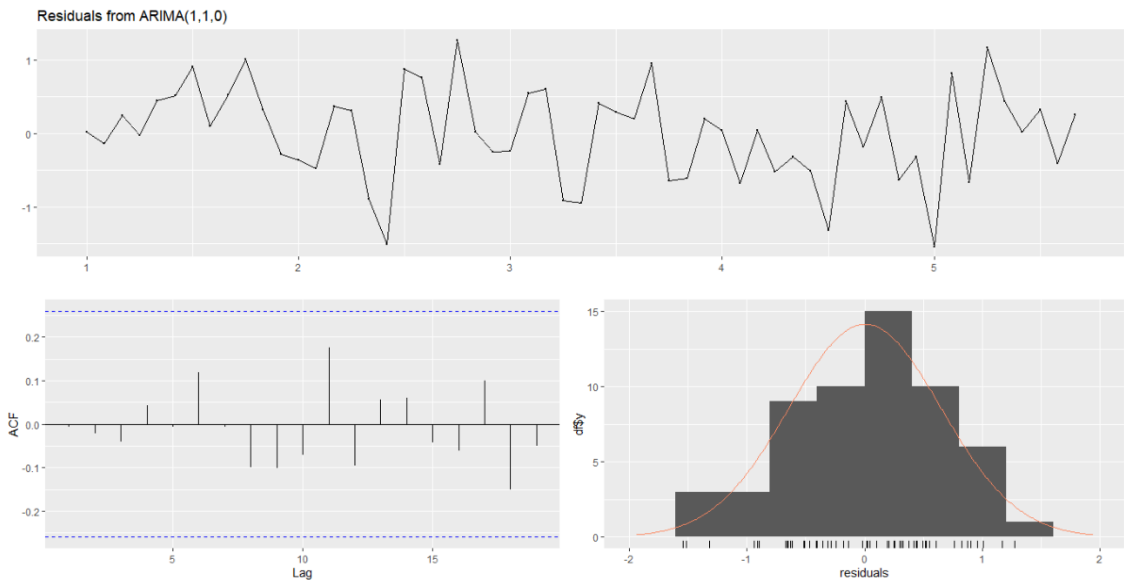
H_0 : Residual tidak berkorelasi

H_1 : Residual saling berkorelasi

```
	Ljung-Box test

data:  Residuals from ARIMA(1,1,0)
Q* = 5.2037, df = 10, p-value = 0.8772

Model df: 1.   Total lags used: 11
```

Hasil dari uji Ljung-Box menunjukkan nilai p-value = 0.8772 > 0.05 = α sehingga hipotesis null gagal ditolak. Dengan demikian, terdapat cukup bukti pada tingkat kepercayaan 95% untuk menyimpulkan bahwa residual dari model fit tidak berkorelasi.

Selain itu, didapatkan pula *output* dalam bentuk grafik yang ditampilkan sebagai berikut.



Keterangan:

- Grafik yang berada di posisi paling atas adalah plot runtun waktu dari residual model fit.
- Grafik yang berada di kiri bawah adalah plot ACF dari residual model fit.
- Grafik yang berada di kanan bawah adalah histogram dari residual model fit.

Berdasarkan ketiga grafik yang didapatkan, diperoleh informasi sebagai berikut.

1. Plot runtun waktu dari residual model fit dapat digunakan untuk menganalisis stasioneritas residual secara subjektif. Berdasarkan plot runtun waktu dari residual model fit, terlihat bahwa residual model fit memiliki rata-rata dan variansi yang konstan sepanjang waktu sehingga residual model fit bersifat stasioner. Hal ini mendukung hasil uji ADF sebelumnya.
2. Plot ACF dari residual model fit dapat digunakan untuk menganalisis independensi residual secara subjektif. Berdasarkan plot ACF dari residual model fit, dapat diamati bahwa tidak ada ACF residual yang melewati garis putus-putus pada plot baik di bagian atas atau di bagian bawah. Oleh karena itu, dapat dikatakan tidak ada ACF residual yang signifikan dan residual model fit saling tidak berkorelasi untuk sembarang *lag*. Hal ini mendukung hasil uji Ljung-Box sebelumnya.
3. Histogram dari residual model fit dapat digunakan untuk menganalisis normalitas residual secara subjektif. Berdasarkan histogram dari residual model fit, terlihat bahwa bentuk histogram residual menyerupai bentuk histogram distribusi Normal atau dengan kata lain residual model fit berdistribusi Normal. Hal ini mendukung hasil uji Shapiro-Wilk sebelumnya.

3.5.4 Uji Signifikansi Parameter Model ARIMA(1,1,0)

Pada tahapan ini akan dilakukan uji signifikansi parameter model ARIMA(1,1,0) atau model fit. Uji ini dilakukan untuk mengetahui apakah parameter yang ada signifikan dan diperlukan dalam model. Pengujian dilakukan dengan menggunakan uji t pada tingkat signifikansi 5%.

H_0 : Parameter ARIMA bernilai 0

H_1 : Parameter ARIMA tidak bernilai 0

```
# Uji Signifikansi Parameter ARIMA
t_test_fit <- t_test(fit, alpha = 0.05)
print(t_test_fit)

        Coeffs Std.Errors        t Crit.Values Rej.H0
ar1 -0.3626053  0.1233953 2.938566    2.004045   TRUE
```

Hasil dari uji t menunjukkan hipotesis null ditolak untuk parameter ϕ dari model fit. Dengan demikian, terdapat cukup bukti pada tingkat kepercayaan 95% untuk menyimpulkan bahwa parameter ϕ dari model fit signifikan. Dapat dikatakan bahwa parameter ϕ tidak bernilai 0 dan diperlukan pada model fit.

3.5.5 Overfitting

Overfitting adalah proses membandingkan model ARIMA terbaik sementara (model fit) dengan model ARIMA yang satu tingkat lebih kompleks. Secara umum, ARIMA(p, d, q) di-*overfit* dengan model ARIMA(p+1, d, q) dan ARIMA(p, d, q +1). Maka, ARIMA(1,1,0) sebagai model ARIMA terbaik sementara akan di-*overfit* dengan model ARIMA(2,1,0) dan ARIMA(1,1,1). Setelah itu, model ARIMA terbaik sementara dapat dibandingkan dengan model overfit.

Ringkasan dari model overfit ditampilkan sebagai berikut.

```
# Overfitting
overfit1 <- Arima(close_ts, order=c(2,1,0))
overfit1


Series: close_ts
ARIMA(2,1,0)

Coefficients:
          ar1      ar2
      -0.3633  -0.0019
s.e.   0.1325   0.1323

sigma^2 = 0.4323:  log likelihood = -55.03
AIC=116.06   AICc=116.52   BIC=122.14
```

```
overfit2 <- Arima(close_ts, order=c(1,1,1))
overfit2

Series: close_ts
ARIMA(1,1,1)

Coefficients:
          ar1      ma1
      -0.3561  -0.0075
s.e.   0.4064   0.4449

sigma^2 = 0.4323:  log likelihood = -55.03
AIC=116.06   AICc=116.52   BIC=122.14
```

Model fit dan model overfitdapat dibandingkan dengan lebih mudah dengan perintah cbind()

```
cbind(overfit1,fit,overfit2)

          overfit1     fit          overfit2
coef      numeric,2    -0.3626053   numeric,2
sigma2    0.4323189    0.4244602    0.4323179
var.coef  numeric,4    0.0152264    numeric,4
mask      logical,2    TRUE         logical,2
loglik    -55.03144    -55.03155    -55.03139
aic       116.0629     114.0631     116.0628
arma      integer,7    integer,7    integer,7
residuals ts,57        ts,57        ts,57
call      expression   expression   expression
series    "close_ts"   "close_ts"   "close_ts"
code      0            0            0
n.cond    0            0            0
nobs      56           56           56
model     list,10      list,10      list,10
aicc      116.5244     114.2895     116.5243
bic       122.1389     118.1138     122.1388
x         ts,57        ts,57        ts,57
fitted    ts,57        ts,57        ts,57
```

Melalui proses perbandingan antara model overfit1, model fit, dan model overfit2 tersebut dapat ditentukan model terbaik untuk data harga penutupan saham JPM. Dengan menggunakan kriteria pemilihan model terbaik berdasarkan nilai AIC dan nilai BIC terkecil, model terbaik tetap jatuh pada model fit atau ARIMA(1,1,0) dengan nilai AIC dan nilai BIC yang lebih kecil daripada dengan kedua model overfit

3.5.6 Uji Signifikansi Parameter Model Overfit

Selain itu, hal lain yang dapat dilakukan saat melakukan proses *overfitting* adalah mengecek signifikansi parameter model tambahan pada model ARIMA. Pengecekan dilakukan dengan menggunakan uji t pada tingkat signifikansi 5%.

H_0 : Parameter ARIMA bernilai 0

H_1 : Parameter ARIMA tidak bernilai 0

```
# Uji Signifikansi Model Overfit
t_test_fit1 <- t_test(overfit1)
t_test_fit2 <- t_test(overfit2)
```

```
> print(t_test_fit1)
           Coeffs Std.Errors          t Crit.Values Rej.H0
ar1 -0.363308500  0.1325378 2.74116863    2.004879   TRUE
ar2 -0.001923806  0.1323259 0.01453839    2.004879  FALSE
> print(t_test_fit2)
           Coeffs Std.Errors          t Crit.Values Rej.H0
ar1 -0.356113949  0.4064210 0.87621932    2.004879  FALSE
ma1 -0.007504364  0.4449423 0.01686593    2.004879  FALSE
```

Berdasarkan output tersebut, didapatkan kesimpulan sebagai berikut.

- Pada model overfit1, yaitu ARIMA(2,1,0), parameter model yang bertambah adalah ϕ_2. Hasil dari uji t menunjukkan hipotesis null gagal ditolak untuk parameter ϕ_2. Dengan demikian, terdapat cukup bukti pada tingkat kepercayaan 95% untuk menyimpulkan bahwa parameter ϕ_2 dari model overfit1 tidak signifikan. Dapat dikatakan bahwa parameter ϕ_2 bernilai 0 dan tidak diperlukan pada model. Hal tersebut menunjukkan bahwa model fit atau ARIMA(1,1,0) tidak perlu di-*overfit* dengan menambahkan parameter ϕ_2.
- Pada model overfit2, yaitu ARIMA(1,1,1), parameter model yang bertambah adalah θ. Hasil dari uji t menunjukkan hipotesis null gagal ditolak untuk parameter θ. Dengan demikian, terdapat cukup bukti pada tingkat kepercayaan 95% untuk menyimpulkan bahwa parameter θ dari model overfit2 tidak signifikan. Dapat dikatakan bahwa parameter θ bernilai 0 dan tidak diperlukan pada model. Hal tersebut menunjukkan bahwa model fit atau ARIMA(1,1,0) tidak perlu di-*overfit* dengan menambahkan parameter θ.

3.6 Forecasting

3.6.1 Cross Validation

Akan dilakukan *cross validation,* data time series close_ts dibagi menjadi 80% untuk training (train_data) dan 20% untuk testing (test_data). Model yang dilatih (train_model) digunakan untuk memprediksi nilai pada periode actual. Perhatikan syntax dan output berikut :

```
> # Membagi data menjadi training dan testing untuk cross-validation
> train_size <- floor(0.8 * length(close_ts))
> train_data <- close_ts[1:train_size]
> test_data <- close_ts[(train_size + 1):length(close_ts)]
>
> # Melatih model pada training data
> train_model <- auto.arima(train_data)
```

```
>
> # Forecasting pada data test
> forecast_horizon <- length(test_data)
> forecast_test <- forecast(train_model, h = forecast_horizon)
>
> # Menyusun dataframe hasil forecasting bersama dengan nilai actual
> results <- data.frame(
+  Date = df$Date[(train_size + 1):length(df$Date)],
+  Actual = test_data,
+  Forecast = as.numeric(forecast_test$mean)
+ )
>
> # Menampilkan hasil forecasting beserta nilai actual
> print(results)

         Date Actual Forecast
1  2022-07-01  25.58 25.15562
2  2022-08-01  24.71 25.35861
3  2022-09-01  24.71 25.52636
4  2022-10-01  23.17 25.66499
5  2022-11-01  24.55 25.77956
6  2022-12-01  23.39 25.87424
7  2023-01-01  24.98 25.95249
8  2023-02-01  24.84 26.01715
9  2023-03-01  24.91 26.07059
10 2023-04-01  25.21 26.11476
11 2023-05-01  24.70 26.15126
12 2023-06-01  25.14 26.18142
```

Hasil dari prediksi dan nilai aktual menunjukkan perbandingan harga penutupan saham JPM selama periode tes (12 bulan terakhir) dengan prediksi menggunakan model ARIMA yang dilatih pada data training. Prediksi cenderung lebih tinggi dari nilai aktual untuk beberapa bulan pertama, tetapi secara umum mengikuti tren harga penutupan saham dengan baik.

Terdapat perbedaan antara nilai aktual dan prediksi, dengan selisih yang bervariasi setiap bulan. Meskipun ada perbedaan antara nilai *actual* dan *forecast*, prediksi model masih berada dalam kisaran yang cukup dekat dengan harga penutupan sebenarnya.

Hal ini menunjukkan bahwa model ARIMA yang digunakan memiliki performa yang baik, meskipun masih dapat diperbaiki untuk prediksi yang lebih akurat.

Nilai MAE (mean absolute error), MAPE (mean absolute percentage error), dan RMSE (root mean squared error) dapat digunakan untuk mengukur seberapa baik hasil ramalan dari model terbaik.

```
> # Menghitung MAE
> mae <- mean(abs(results$Actual - results$Forecast))
> print(paste("Mean Absolute Error (MAE):", mae))
[1] "Mean Absolute Error (MAE): 1.23381769919169"

> # Menghitung MAPE
> mape <- mean(abs((results$Actual - results$Forecast) /
results$Actual) * 100)
> print(paste("Mean Absolute Percentage Error (MAPE):", mape))
[1] "Mean Absolute Percentage Error (MAPE): 5.07360697961222"
```

```
> # Menghitung RMSE
> rmse <- sqrt(mean((results$Actual - results$Forecast)^2))
> print(paste("Root Mean Squared Error (RMSE):", rmse))
[1] "Root Mean Squared Error (RMSE): 1.38031343452769"
```

- MAE dan RMSE menunjukkan bahwa model ARIMA cukup akurat dengan rata-rata kesalahan prediksi sekitar 1.23 dan 1.38 unit harga saham.
- MAPE menunjukkan bahwa prediksi model cukup akurat dengan kesalahan rata-rata sekitar 5.07%, yang umumnya dianggap baik dalam konteks prediksi saham.
- Secara keseluruhan, model ARIMA yang digunakan memiliki performa yang cukup baik dalam memprediksi harga penutupan saham JPMorgan, meskipun ada beberapa prediksi yang mungkin meleset lebih jauh dari yang lain, seperti yang ditunjukkan oleh perbedaan antara MAE dan RMSE.

3.6.2 Forecasting ARIMA

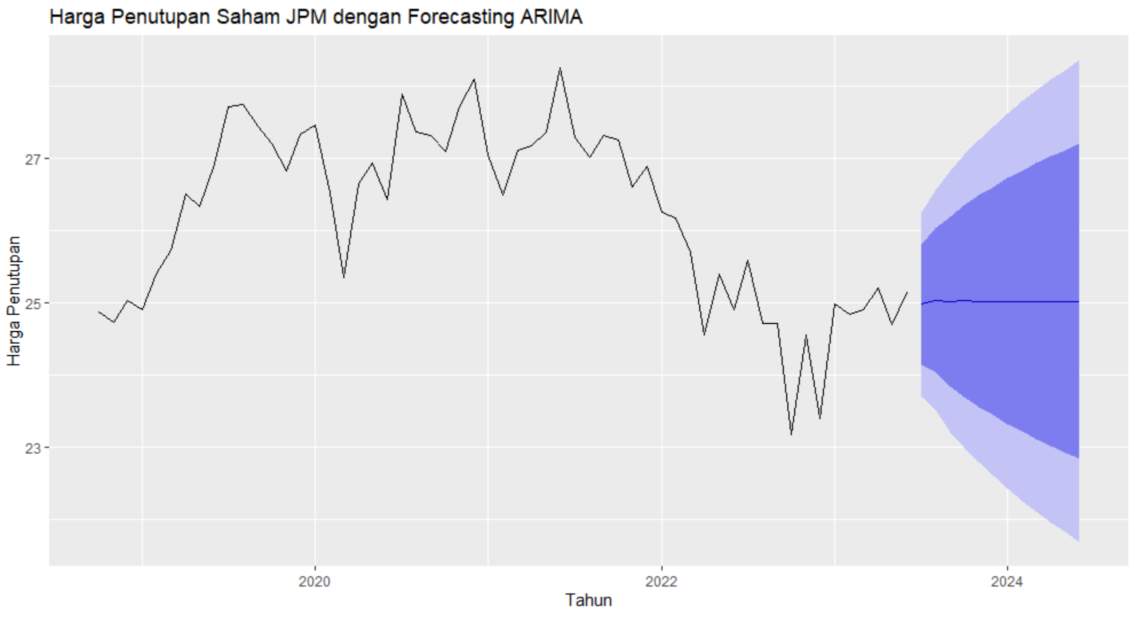
Dilakukan peramalan untuk 12 periode mendata ARIMA Forecasting menggunakan software RStudio, akan digunakan model yang telah terpilih yaitu ARIMA (1,1,0). Berikut adalah kode yang digunakan untuk mendapatkan hasil peramalan untuk data harga penutupan saham beserta interval prediksinya dengan tingkat signifikansi 80% dan 95%:

```
> # Forecasting menggunakan model2 untuk 12 periode ke depan
> forecast_future <- forecast(model2, h = 12)
>
> # Menampilkan hasil forecasting
> print(forecast_future)
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jul 2023       24.98045 24.14551 25.81539 23.70352 26.25738
Aug 2023       25.03831 24.04818 26.02843 23.52404 26.55257
Sep 2023       25.01733 23.83730 26.19735 23.21263 26.82202
Oct 2023       25.02493 23.70015 26.34972 22.99885 27.05102
Nov 2023       25.02218 23.56093 26.48342 22.78739 27.25696
Dec 2023       25.02318 23.43919 26.60716 22.60068 27.44567
Jan 2024       25.02281 23.32426 26.72137 22.42510 27.62053
Feb 2024       25.02295 23.21731 26.82858 22.26146 27.78443
Mar 2024       25.02290 23.11610 26.92969 22.10670 27.93909
Apr 2024       25.02291 23.02009 27.02574 21.95986 28.08597
May 2024       25.02291 22.92844 27.11737 21.81970 28.22612
Jun 2024       25.02291 22.84065 27.20517 21.68544 28.36039
```

Dapat dilihat dari tabel bahwa harga saham JPMorgan diramalkan nilai yang cenderung menaik namun akan cukup konstan pada angka 25.02291 di akhir. Akan tetapi, tentu saja semakin lama periode peramalannya, interval kepercayaan akan menjadi lebih lebar dan bervariasi. Untuk tingkat kepercayaan 95%, diramalkan harga saham JPMorgan pada bulan Juli 2023 berada pada interval (23.70352, 26.25738), hingga pada 12 bulan periode berikutnya yaitu Juni 2024 (periode tahun ke-6) berada pada interval (21.68544, 28.36039).

Berdasarkan plot dari peramalan dengan model ARIMA(1,1,0), hal ini dapat dilihat juga dari plot peramalan atau forecast yaitu sebagai berikut:

```
> # Plot hasil forecasting
> autoplot(forecast_best_model) +
+  ggtitle("Forecasting Harga Penutupan Saham JPM") +
+  xlab("Tahun") +
+  ylab("Harga Penutupan") +
+  theme_minimal() +
+  theme(plot.title = element_text(hjust = 0.5))
```



BAB IV
KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan data harga penutupan saham JPMorgan Chase & Co. (JPM-PD) per bulan dari Oktober 2018 hingga Juni 2023, diperoleh kesimpulan bahwa data akan menjadi stasioner setelah dilakukan differencing satu kali. Model yang cocok untuk menggambarkan data tersebut adalah model ARIMA (1,1,0) dengan persamaan umum $Y_t = (1 + \varphi)Y_{t-1} - \varphi Y_{t-2} + e_t$. Setelah proses penaksiran parameter, diperoleh model akhir $Y_t = (1 - 0.3626)Y_{t-1} + 0.3626Y_{t-2} + e_t$. Model ini telah melalui proses pengecekan residual dan overfitting sertacocok untuk meramalkan naik harga penutupan saham. Berdasarkan model tersebut, diramalkan nilai yang cenderung namun akan cukup konstan pada angka 25.02291 di akhir pengamatan yaitu Juni 2024.

4.2 Saran

Pada penelitian ini, terdapat saran yang perlu dipertimbangkan. Pertama, penggunaan model ARIMA dalam memprediksi harga saham memiliki beberapa keterbatasan. Model ARIMA didasarkan pada asumsi bahwa data mengikuti pola stasioner, namun pergerakan harga saham cenderung memiliki sifat non-stasioner dan fluktuasi yang kompleks. Oleh karena itu, penelitian perlu mempertimbangkan penggunaan model prediktif yang lebih kompleks atau kombinasi dengan metode lain untuk meningkatkan akurasi prediksi.

LAMPIRAN

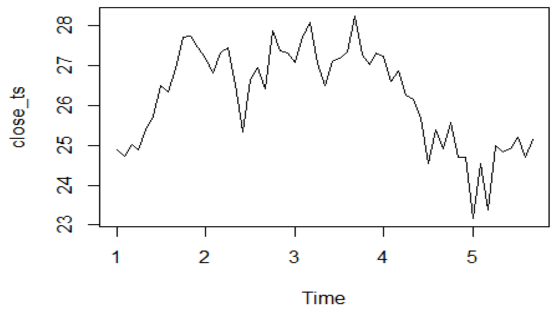
EVDA:

https://colab.research.google.com/drive/12-AhGF0HwD1VkidmdPrkGj0duj-p43YM?usp=sharing

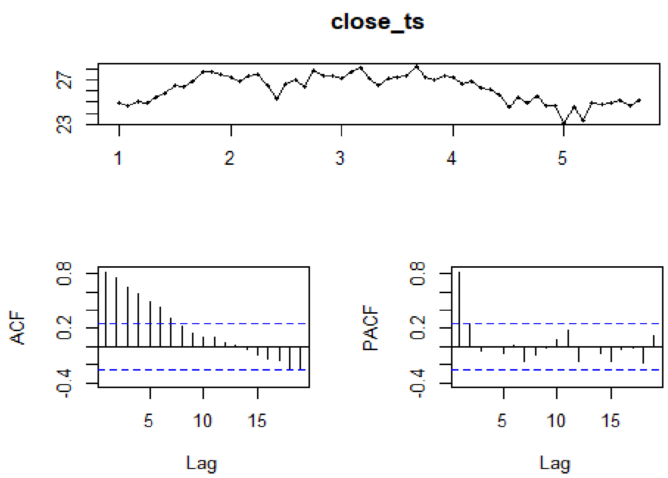
PENGOLAHAN DATA

```r
# Memuat library yang diperlukan
library(ggplot2)

library(forecast)

library(tseries)

library(rpart)
library(TSA)

library(lubridate)
library(dplyr)


# Memuat dataset dan mengonversi kolom tanggal ke format datetime
df <- read.csv("C:/Users/Kayla/Downloads/JPM-PD.csv")
# Memeriksa nilai yang hilang
sum(is.na(df))

## [1] 0

head(df$Date)

## [1] "2018-10-01" "2018-11-01" "2018-12-01" "2019-01-01" "2019-02-01"
## [6] "2019-03-01"

### Konversi dan pengurutan data berdasarkan tanggal
df <- df[order(df$Date),]  # Memastikan data terurut berdasarkan tanggal
df$Date <- as.Date(df$Date)


### Membuat time series dari harga penutupan
close_ts <- ts(df$Close, start=c(2018, 10), frequency=12)
```

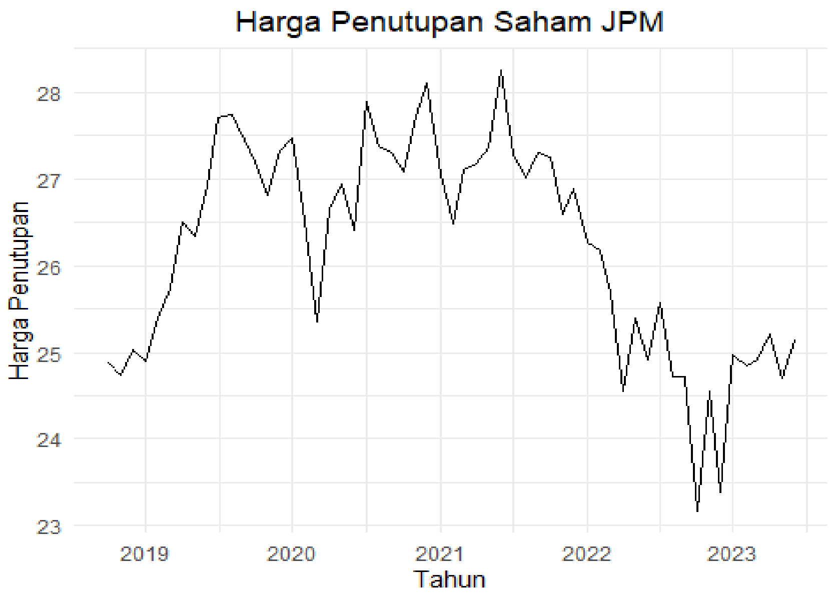
```r
### Menampilkan struktur dari time series yang baru dibuat
print(close_ts)

       Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct
2018                                                        24.88
2019 24.90 25.39 25.73 26.51 26.33 26.92 27.71 27.75 27.45 27.20
2020 27.46 26.52 25.35 26.65 26.94 26.42 27.88 27.37 27.31 27.09
2021 27.05 26.49 27.11 27.18 27.36 28.25 27.28 27.02 27.31 27.25
2022 26.26 26.17 25.69 24.55 25.40 24.91 25.58 24.71 24.71 23.17
2023 24.98 24.84 24.91 25.21 24.70 25.14
       Nov   Dec
2018 24.73 25.03
2019 26.82 27.33
2020 27.72 28.10
2021 26.60 26.88
2022 24.55 23.39
2023

plot(close_ts)
```




```r
tsdisplay(close_ts)
```

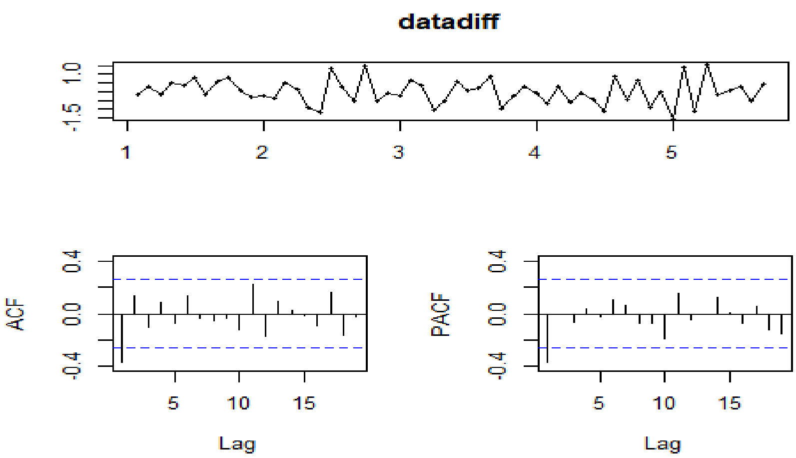


```r
### Visualisasi data harga penutupan seiring waktu dengan ggplot2
ggplot(df, aes(x = Date, y = Close, group = 1)) +
  geom_line() +
  ggtitle("Harga Penutupan Saham JPM") +
  xlab("Tahun") +
  ylab("Harga Penutupan") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

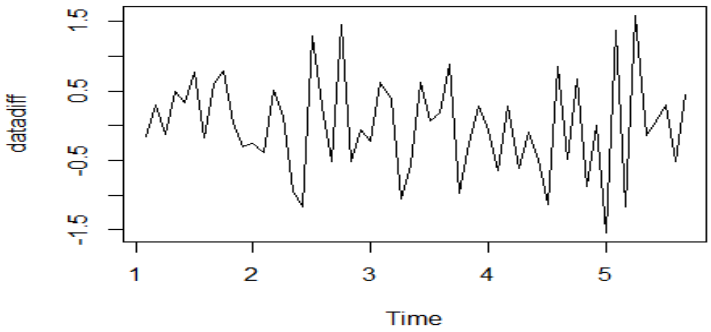



```r
### Uji stasioneritas dengan Augmented Dickey-Fuller test
adf_test_weekly <- adf.test(close_ts, alternative = "stationary")
print(adf_test_weekly)  # Tampilkan hasil uji ADF

##
##  Augmented Dickey-Fuller Test
##
## data:  close_ts
## Dickey-Fuller = -2.6963, Lag order = 3, p-value = 0.2937
## alternative hypothesis: stationary

### Differencing untuk mencapai stasioneritas
datadiff <- diff(close_ts, differences = 1)
print(datadiff)  # Tampilkan data setelah differencing
```

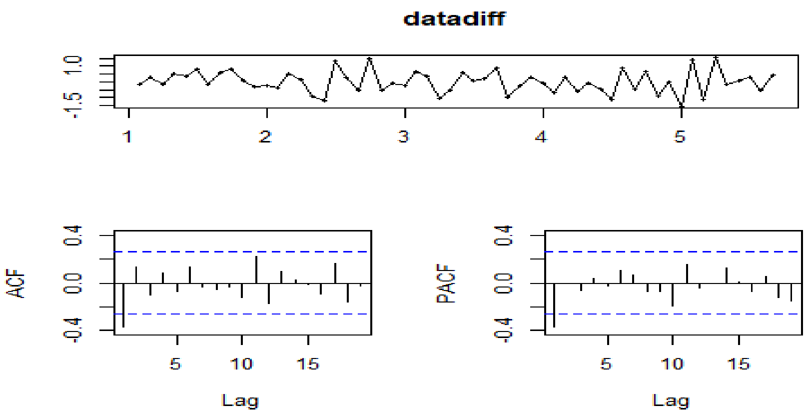
```
##          Jan       Feb       Mar       Apr       May       Jun       Jul
## 1            -0.149999  0.300001 -0.130001  0.489999  0.340001  0.780000
## 2 -0.250000 -0.380001  0.510000  0.129999 -0.939999 -1.170000  1.300000
## 3 -0.219999  0.629999  0.380001 -1.050001 -0.559999  0.620001  0.069999
## 4 -0.059999 -0.650000  0.279999 -0.619999 -0.090000 -0.479999 -1.140002
## 5 -1.539999  1.379999 -1.160000  1.590001 -0.140000  0.070000  0.299999
##          Aug       Sep       Oct       Nov       Dec
## 1 -0.180000  0.590000  0.789999  0.040001 -0.299999
## 2  0.290001 -0.520001  1.459999 -0.509998 -0.060002
## 3  0.180001  0.889999 -0.969999 -0.260001  0.289999
## 4  0.850001 -0.490000  0.670000 -0.870001  0.000000
## 5 -0.509998  0.439998

### Uji stasioneritas lagi dengan ADF test setelah differencing
adf_test_datadiff <- adf.test(datadiff)
print(adf_test_datadiff)

## 
##  Augmented Dickey-Fuller Test
## 
## data:  datadiff
## Dickey-Fuller = -3.9489, Lag order = 3, p-value = 0.01824
## alternative hypothesis: stationary

### Menampilkan time series setelah differencing
tsdisplay(datadiff)
```



```r
eacf(datadiff)

## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o o o o o o  o  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 o o o o o o o o o o o  o  o  o
## 3 x o o o o o o o o o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 o o o x o o o o o o o  o  o  o
## 6 x x o o o o o o o o o  o  o  o
## 7 x x o o o o o o o o o  o  o  o

### Plot hasil differencing dan display ulang
plot(datadiff)
```




```r
tsdisplay(datadiff)
```



```r
### Model ARIMA
model1 <- Arima(close_ts, order=c(0,1,1))
model2 <- Arima(close_ts, order=c(1,1,0))
model3 <- Arima(close_ts, order=c(1,1,1))

### Ringkasan dan perbandingan model
summary(model1)
```

```
## Series: close_ts
## ARIMA(0,1,1)
##
## Coefficients:
##           ma1
##       -0.3381
## s.e.   0.1151
##
## sigma^2 = 0.43:  log likelihood = -55.38
## AIC=114.77   AICc=115   BIC=118.82
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE     MAPE
MASE
## Training set 0.004787662 0.6441357 0.5184551 -0.02259485 1.995645 0.427
2981
##                     ACF1
## Training set -0.04065767
```

```r
summary(model2)
```

```
## Series: close_ts
## ARIMA(1,1,0)
##
## Coefficients:
##           ar1
##       -0.3626
## s.e.   0.1234
##
```

```
## sigma^2 = 0.4245:  log likelihood = -55.03
## AIC=114.06   AICc=114.29   BIC=118.11
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE     MAPE
MASE
## Training set 0.004032104 0.6399741 0.5215752 -0.02274635 2.001697 0.429
8696
##                      ACF1
## Training set -0.007123785
```

```
summary(model3)
```

```
## Series: close_ts
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1      ma1
##       -0.3561  -0.0075
## s.e.   0.4064   0.4449
##
## sigma^2 = 0.4323:  log likelihood = -55.03
## AIC=116.06   AICc=116.52   BIC=122.14
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE     MAPE
MASE
## Training set 0.004047351 0.6399721 0.5214934 -0.02274875 2.001486 0.429
8022
##                      ACF1
## Training set -0.006284335
```

```
### Kalkulasi AIC dan BIC untuk pemilihan model
AIC(model1, model2, model3)
```

```
##        df      AIC
## model1  2 114.7695
```

```
## model2  2 114.0631
## model3  3 116.0628
```

```r
cbind(model1, model2, model3)
```

```
##           model1     model2     model3
## coef      -0.3381014 -0.3626053 numeric,2
## sigma2    0.4299985  0.4244602  0.4323179
## var.coef  0.01324161 0.0152264  numeric,4
## mask      TRUE       TRUE       logical,2
## loglik    -55.38474  -55.03155  -55.03139
## aic       114.7695   114.0631   116.0628
## arma      integer,7  integer,7  integer,7
## residuals ts,57      ts,57      ts,57
## call      expression expression expression
## series    "close_ts" "close_ts" "close_ts"
## code      0          0          0
## n.cond    0          0          0
## nobs      56         56         56
## model     list,10    list,10    list,10
## aicc      114.9959   114.2895   116.5243
## bic       118.8202   118.1138   122.1388
## x         ts,57      ts,57      ts,57
## fitted    ts,57      ts,57      ts,57
```

```r
### Memilih model terbaik
fit <- model2
fit
```

```
## Series: close_ts
## ARIMA(1,1,0)
##
## Coefficients:
##           ar1
##       -0.3626
## s.e.   0.1234
##
```

```
## sigma^2 = 0.4245:  log likelihood = -55.03
## AIC=114.06   AICc=114.29   BIC=118.11
```

```
# Uji Stasioneritas Residual
adf_test_fit <- adf.test(fit$residuals)
print(adf_test_fit)
```

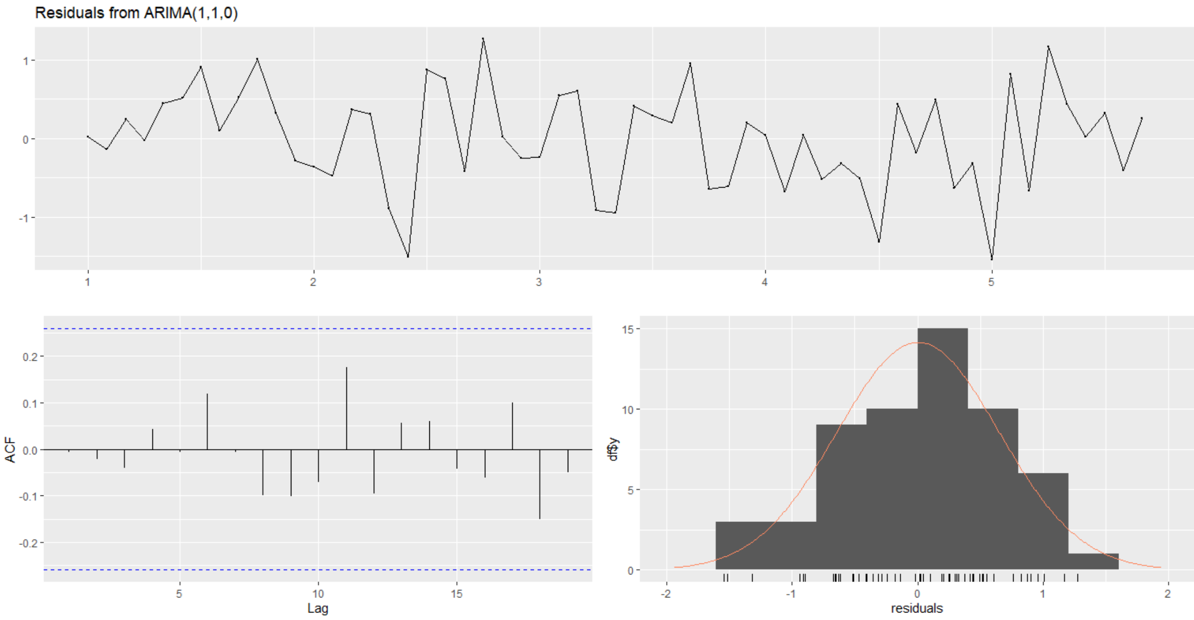
```
        Augmented Dickey-Fuller Test

data:  fit$residuals
Dickey-Fuller = -3.7652, Lag order = 3, p-value = 0.02722
alternative hypothesis: stationary
```

```
# Uji Normalitas Residual
shapiro_test_fit <- shapiro.test(fit$residuals)
print(shapiro_test_fit)
```

```
        Shapiro-Wilk normality test

data:  fit$residuals
W = 0.98268, p-value = 0.5861
```

```
# Uji Independensi Residual
lbtest_fit <- checkresiduals(fit)
```

```
        Ljung-Box test

data:  Residuals from ARIMA(1,1,0)
Q* = 5.2037, df = 10, p-value = 0.8772

Model df: 1.   Total lags used: 11
```




```
# Uji Signifikansi Parameter ARIMA
t_test_fit <- t_test(fit, alpha = 0.05)
print(t_test_fit)
```

```
        Coeffs Std.Errors        t Crit.Values Rej.H0
ar1 -0.3626053  0.1233953 2.938566    2.004045   TRUE
```

```
# Overfitting
overfit1 <- Arima(close_ts, order=c(2,1,0))
overfit1

Series: close_ts
ARIMA(2,1,0)

Coefficients:
          ar1      ar2
      -0.3633  -0.0019
s.e.   0.1325   0.1323

sigma^2 = 0.4323:  log likelihood = -55.03
AIC=116.06   AICc=116.52   BIC=122.14
```

```
overfit2 <- Arima(close_ts, order=c(1,1,1))
overfit2

Series: close_ts
ARIMA(1,1,1)

Coefficients:
          ar1      ma1
      -0.3561  -0.0075
s.e.   0.4064   0.4449

sigma^2 = 0.4323:  log likelihood = -55.03
AIC=116.06   AICc=116.52   BIC=122.14
```

```
cbind(overfit1,fit,overfit2)
```

	overfit1	fit	overfit2
coef	numeric,2	-0.3626053	numeric,2
sigma2	0.4323189	0.4244602	0.4323179
var.coef	numeric,4	0.0152264	numeric,4
mask	logical,2	TRUE	logical,2
loglik	-55.03144	-55.03155	-55.03139
aic	116.0629	114.0631	116.0628
arma	integer,7	integer,7	integer,7
residuals	ts,57	ts,57	ts,57
call	expression	expression	expression
series	"close_ts"	"close_ts"	"close_ts"
code	0	0	0
n.cond	0	0	0
nobs	56	56	56
model	list,10	list,10	list,10
aicc	116.5244	114.2895	116.5243
bic	122.1389	118.1138	122.1388
x	ts,57	ts,57	ts,57
fitted	ts,57	ts,57	ts,57

```
# Uji Signifikansi Model Overfit
t_test_fit1 <- t_test(overfit1)
t_test_fit2 <- t_test(overfit2)
print(t_test_fit1)
print(t_test_fit2)
```

```
> print(t_test_fit1)
          Coeffs Std.Errors          t Crit.Values Rej.H0
ar1 -0.363308500  0.1325378 2.74116863    2.004879   TRUE
ar2 -0.001923806  0.1323259 0.01453839    2.004879  FALSE
> print(t_test_fit2)
          Coeffs Std.Errors          t Crit.Values Rej.H0
ar1 -0.356113949  0.4064210 0.87621932    2.004879  FALSE
ma1 -0.007504364  0.4449423 0.01686593    2.004879  FALSE
```

```
#Lakukan Peramalan

# Membagi data menjadi training dan testing untuk cross-validation
 train_size <- floor(0.8 * length(close_ts))
 train_data <- close_ts[1:train_size]
 test_data <- close_ts[(train_size + 1):length(close_ts)]

# Melatih model pada training data
 train_model <- auto.arima(train_data)

# Forecasting pada data test
 forecast_horizon <- length(test_data)
 forecast_test <- forecast(train_model, h = forecast_horizon)

# Menyusun dataframe hasil forecasting bersama dengan nilai actual
 results <- data.frame(
   Date = df$Date[(train_size + 1):length(df$Date)],
   Actual = test_data,
   Forecast = as.numeric(forecast_test$mean)
 )

# Menampilkan hasil forecasting beserta nilai actual
 print(results)

##            Date Actual Forecast
 ## 1  2022-07-01  25.58 25.15562
```

```
## 2  2022-08-01  24.71 25.35861
## 3  2022-09-01  24.71 25.52636
## 4  2022-10-01  23.17 25.66499
## 5  2022-11-01  24.55 25.77956
## 6  2022-12-01  23.39 25.87424
## 7  2023-01-01  24.98 25.95249
## 8  2023-02-01  24.84 26.01715
## 9  2023-03-01  24.91 26.07059
## 10 2023-04-01  25.21 26.11476
## 11 2023-05-01  24.70 26.15126
## 12 2023-06-01  25.14 26.18142
```

```
# Menghitung MAE
mae <- mean(abs(results$Actual - results$Forecast))
print(paste("Mean Absolute Error (MAE):", mae))
```

```
## [1] "Mean Absolute Error (MAE): 1.23381769919169"
```

```
# Menghitung MAPE
mape <- mean(abs((results$Actual - results$Forecast) / results$Actual) *
100)
print(paste("Mean Absolute Percentage Error (MAPE):", mape))
```

```
## [1] "Mean Absolute Percentage Error (MAPE): 5.07360697961222"
```

```
# Menghitung RMSE
rmse <- sqrt(mean((results$Actual - results$Forecast)^2))
print(paste("Root Mean Squared Error (RMSE):", rmse))
```

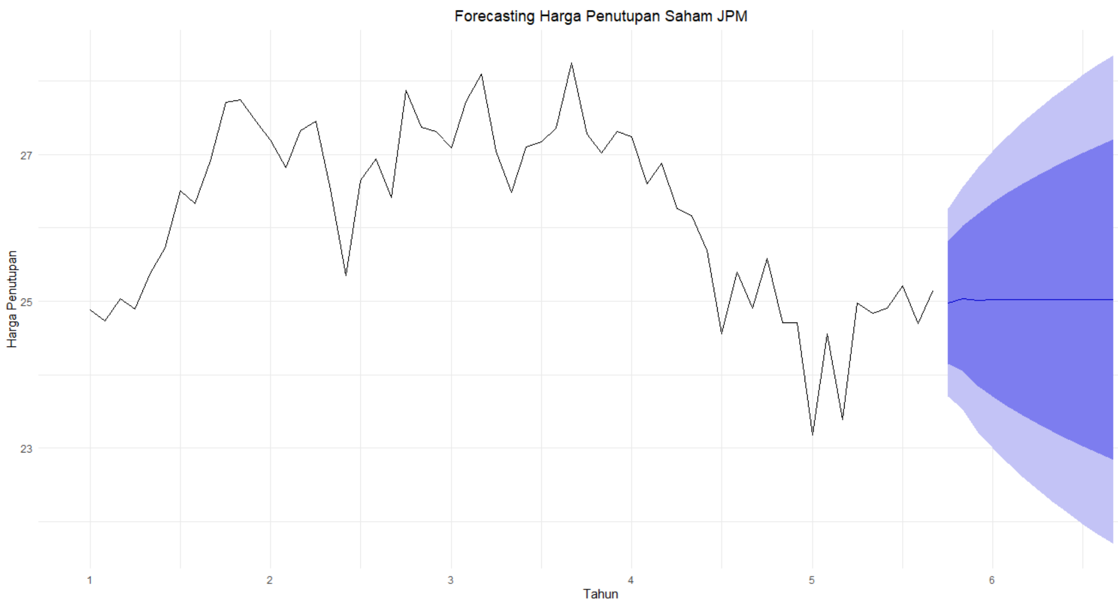
```
## [1] "Root Mean Squared Error (RMSE): 1.38031343452769"
```

```
# Forecasting menggunakan model2 untuk 12 periode ke depan
forecast_future <- forecast(model2, h = 12)

# Menampilkan hasil forecasting
print(forecast_future)
```

```
##       Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Oct 5       24.98045 24.14551 25.81539 23.70352 26.25738
## Nov 5       25.03831 24.04818 26.02843 23.52404 26.55257
```

```
## Dec 5        25.01733 23.83730 26.19735 23.21263 26.82202
## Jan 6        25.02493 23.70015 26.34972 22.99885 27.05102
## Feb 6        25.02218 23.56093 26.48342 22.78739 27.25696
## Mar 6        25.02318 23.43919 26.60716 22.60068 27.44567
## Apr 6        25.02281 23.32426 26.72137 22.42510 27.62053
## May 6        25.02295 23.21731 26.82858 22.26146 27.78443
## Jun 6        25.02290 23.11610 26.92969 22.10670 27.93909
## Jul 6        25.02291 23.02009 27.02574 21.95986 28.08597
## Aug 6        25.02291 22.92844 27.11737 21.81970 28.22612
## Sep 6        25.02291 22.84065 27.20517 21.68544 28.36039
```

```
# Plot hasil forecasting
 autoplot(forecast_future) +
   ggtitle("Forecasting Harga Penutupan Saham JPM untuk 12 Periode
Kedepan") +
   xlab("Tahun") +
   ylab("Harga Penutupan") +
   theme_minimal() +
   theme(plot.title = element_text(hjust = 0.5))
```



.