



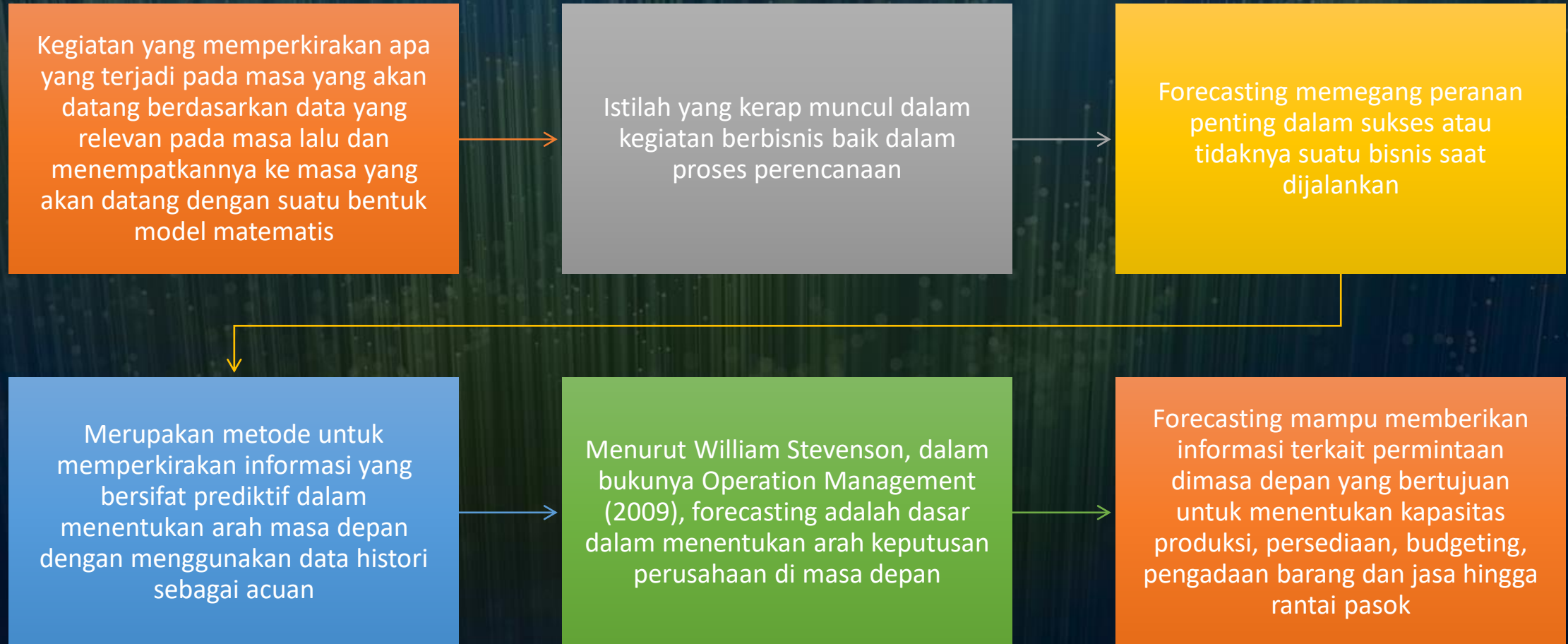
Project UAS PPDS

Mata Kuliah Program Pengolahan Data
Statistik



Teori Forecasting

forecasting



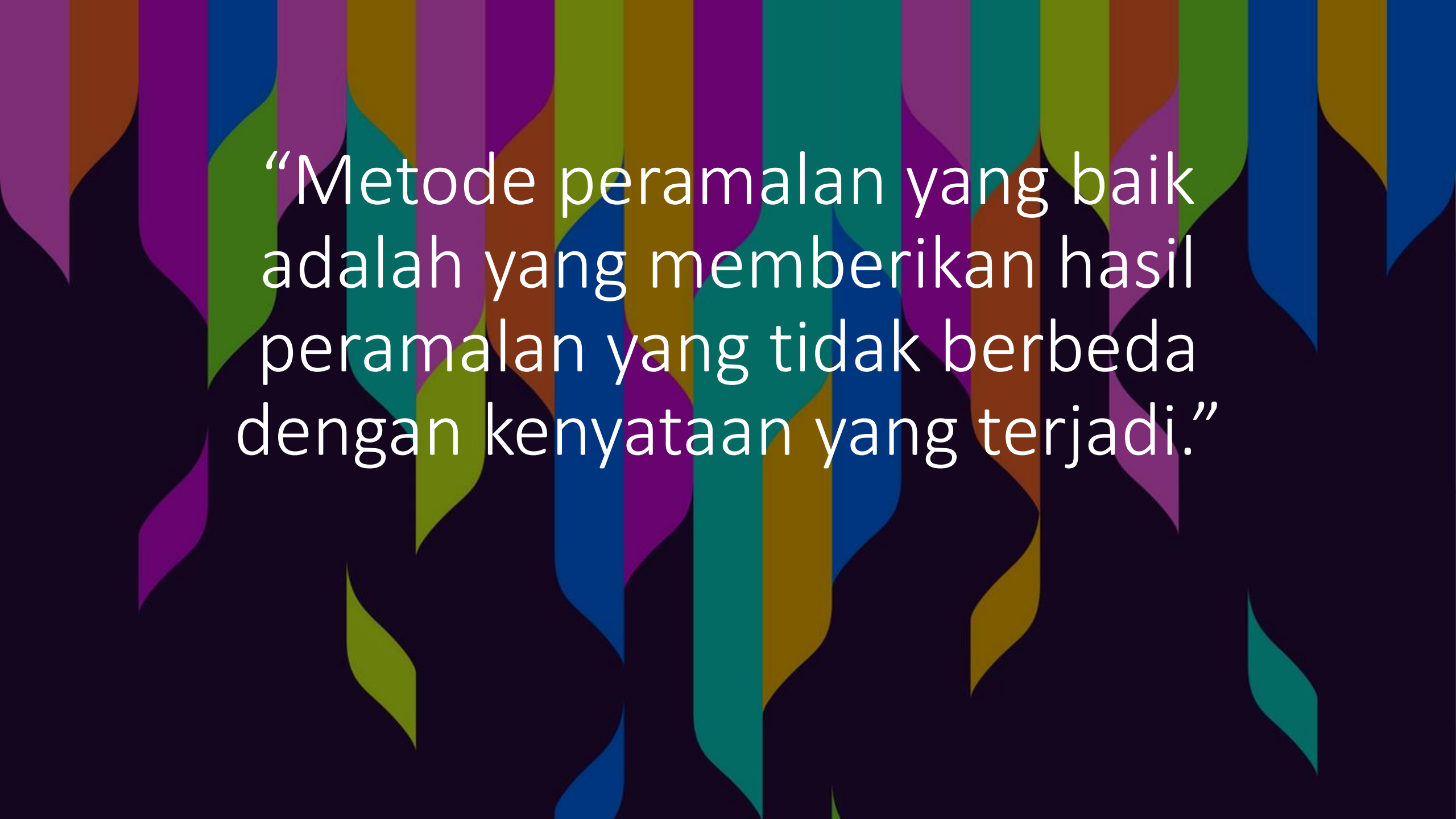
Lanjutan ...

Bisa juga merupakan prediksi intuisi yang bersifat subyektif, atau dengan menggunakan kombinasi model matematis yang disesuaikan dengan pertimbangan yang baik dari seorang manajer

Menurut Satinder Mullick, dkk dalam tulisan di Harvard Business Review, forecasting merupakan alat untuk mengatasi segala jenis potensi masalah yang terjadi dari anomali permintaan baik musiman maupun perubahan ekonomi secara global

Contohnya :

- Dengan adanya forecasting, di masa pandemi COVID 19 yang terjadi diakhir tahun 2019, perusahaan perlu untuk menentukan arah bisnis agar tetap bertahan ditengah pandemic atau pasca pandemic dimasa depan



“Metode peramalan yang baik adalah yang memberikan hasil peramalan yang tidak berbeda dengan kenyataan yang terjadi.”

Tujuan dan Fungsi Forecasting

Fungsi forecasting adalah sebagai pedoman untuk menentukan arah kebijakan dan keputusan perusahaan yang efektif dan efisien di masa depan



Dengan fungsi tersebut, maka forecasting menjadi penting karena beberapa hal berikut :

Sebagai bahan kajian bagi kebijakan perusahaan yang berlaku pada saat ini, masa lalu, serta sejauh mana pengaruhnya di masa depan.

Adanya *time gap* antara kebijakan dengan implementasinya.

Perusahaan memiliki acuan atau arah sehingga setiap kebijakan sejalan dengan tujuan perusahaan.

Memberikan solusi strategis apabila di masa depan menghadapi masalah yang berkaitan dengan bisnis.

Permintaan selalu saja menghadapi anomali atau perubahan dan tentu bisa saja berbeda dari masa lalu dengan masa depan.

Menjaga stabilitas keuangan perusahaan.

Tujuan Forecasting



MENURUT HEIZER DAN RENDER (2009:47), PERKIRAAN ATAU FORECASTING MEMILIKI TUJUAN SEBAGAI BERIKUT:



A. MENGKAJI KEBIJAKAN PERUSAHAAN YANG BERLAKU SAAT INI DAN DI MASA LALU, SERTA MELIHAT SEJAUH MANA PENGARUH DI MASA DATANG.



B. PERKIRAAN DIPERLUKAN KARENA ADANYA TIME LAG ATAU DELAY ANTARA SAAT SUATU KEBIJAKAN PERUSAHAAN DITETAPKAN DENGAN SAAT IMPLEMENTASI.



C. PERKIRAAN MERUPAKAN DASAR PENYUSUNAN BISNIS PADA SUATU PERUSAHAAN SEHINGGA DAPAT MENINGKATKAN EFEKTIVITAS SUATU RENCANA BISNIS.

Faktor-faktor yang mempengaruhi Forecasting

Sifat Produk

- yaitu jangka waktu produk yang akan diproduksi. Apakah produksi jangka panjang atau pendek.

Metode Distribusi

- yaitu jangkauan pasar antara perusahaan dengan konsumen. Kemampuan perusahaan dalam menjangkau konsumennya di dalam pasar.

Posisi Perusahaan dalam pasar

- Apakah perusahaan tersebut *market leader*, penantang baru atau hanya ikut meramaikan pasar tersebut.

Tingkat persaingan

- Bagaimana posisi perusahaan dengan pesaingnya. Apa kekuatan, kelemahan, peluang dan tantangan yang akan dihadapi antara perusahaan dengan pesaing.

Data Historis

- Segala data kejadian yang berhubungan dengan perusahaan di masa lalu. Biasanya data yang diambil setidaknya merupakan data berkembang dari lima tahun lalu.

Jenis – jenis peramalan

Berdasarkan waktu

1

Peramalan jangka panjang yaitu *forecasting* yang menggunakan analisis dengan waktu yang lebih panjang biasanya berlangsung selama dua tahun lebih.

2

Peramalan jangka menengah dengan waktu tiga bulan hingga dua tahun.

3

Peramalan jangka pendek yaitu dengan jangka waktu nol hingga tiga bulan.

Jenis – jenis peramalan

Berdasarkan fungsi dan tujuan

1

General business forecasting

- peramalan bisnis secara keseluruhan mulai dari ekonomi, politik, sosial, budaya dan hal-hal lainnya yang bersifat makro.

2

Sales forecasting

- peramalan jumlah barang yang bisa dijual di masa mendatang berdasarkan data penjualan sebelumnya.

3

Demand forecasting

- peramalan yang bertujuan untuk mengetahui perkiraan permintaan dan kondisi pasar.

4

Financial forecasting

- atau biasa disebut juga dengan capital forecasting. Bertujuan untuk memperkirakan biaya dan modal yang dikeluarkan di masa mendatang.

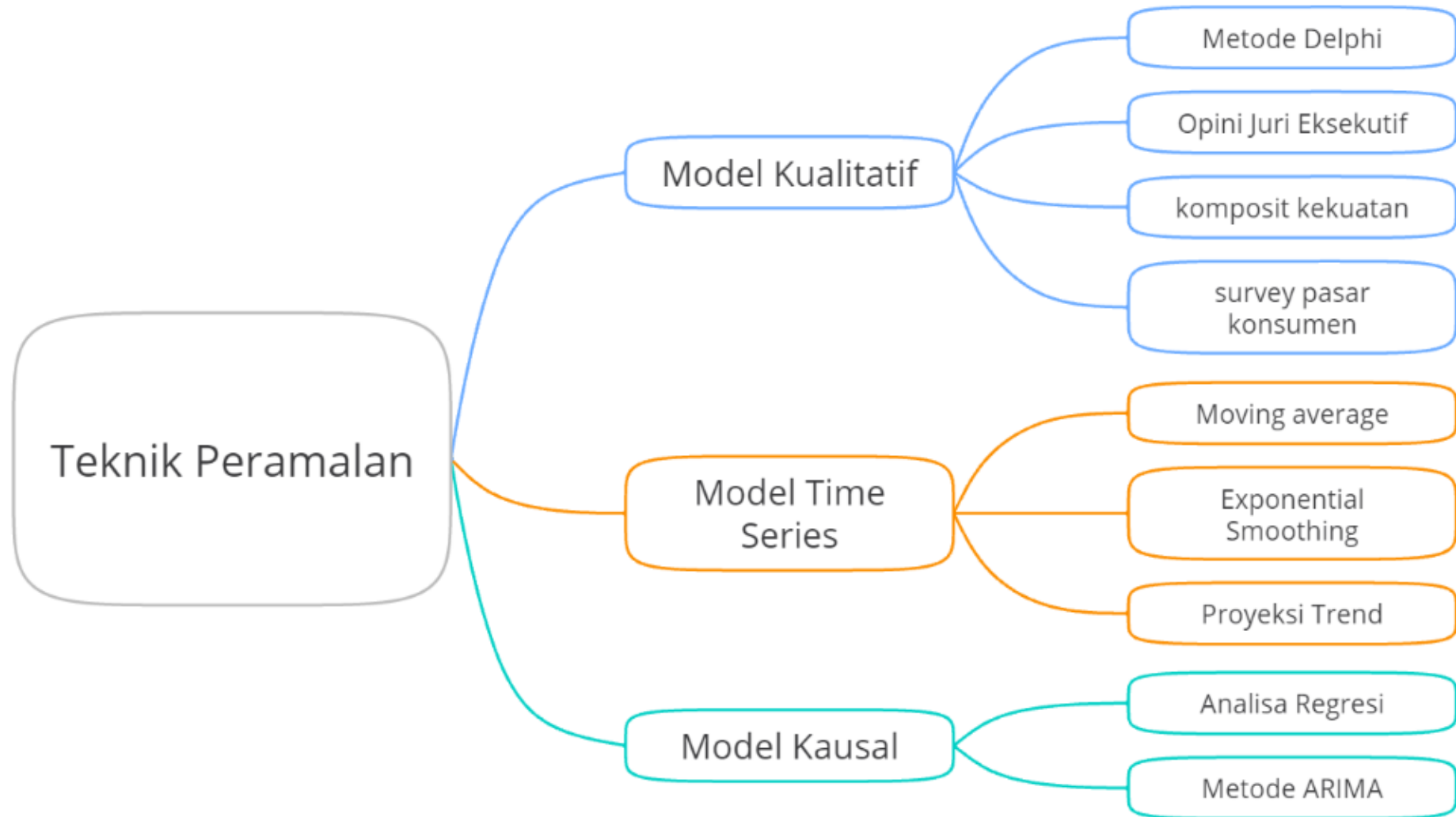
Jenis – jenis peramalan Berdasarkan Ketersediaan Data

Metode Kualitatif

Dimana perusahaan atau organisasi tidak memiliki data yang cukup untuk dianalisis. Hasil peramalan juga akan sangat subyektif karena hasil analisis berbeda-beda. Contoh metode kualitatif: Penyelidikan, wawancara, diskusi.

Metode Kuantitatif

Apabila perusahaan memiliki data yang cukup, maka sebaiknya metode yang digunakan adalah metode kuantitatif. Dimana dalam proses analisisnya menggunakan pendekatan data dan angka.



Teknik Peramalan Model Kualitatif

Model Kualitatif

Model yang berupaya memasukkan faktor-faktor subyektif dalam model peramalan.

Model semacam ini diharapkan akan sangat bermanfaat apabila data kuantitatif yang akurat sulit diperoleh

Metode kualitatif digunakan apabila organisasi atau perusahaan tidak memiliki data di masa lalu. Entah karena data tersebut tidak layak atau tidak sesuai dengan apa yang ingin diramalkan.

Metode kualitatif biasanya umum digunakan oleh para *startup* atau perusahaan yang akan mengalami perubahan dan transisi dalam bisnisnya.

Metode Delphi

Metode dimana sekelompok ahli dari berbagai latar belakang berkumpul untuk memberikan pendapat secara sistematis.

Secara sistematis yang dimaksud ada satu fasilitator yang menengahkan pendapat. Biasanya menggunakan survei tertutup atau kuesioner.

Hal tersebut dilakukan untuk menghindari bias atau *bandwagon effect* yang akan memengaruhi efektivitas hasil *forecasting*.

Metode delphi terbilang cukup efektif dalam peramalan baik dalam jangka waktu pendek maupun panjang.

Riset Pasar

Metode yang sangat umum digunakan dalam melakukan *forecasting*. Dimana peramalan menggunakan sampel penelitian yang diakhiri dengan hipotesis.

Biasanya dilakukan dengan cara melakukan survei kepada konsumen, *stakeholder*, atau investigasi kepada praktisi bahkan pesaing.

Untuk *forecasting* jangka pendek, metode ini sangat efektif namun hasilnya akan sangat fluktuatif untuk peramalan jangka panjang.

Konsensus

Berbeda dengan metode delphi, konsensus dilakukan secara terbuka. Dimana peserta yang hadir melakukan diskusi secara terbuka.

Hasil dari metode konsensus akan sangat bias. Oleh karena itu banyak perusahaan besar yang sudah meninggalkan metode ini untuk melakukan *forecasting*.

Namun metode ini bisa digunakan apabila diikuti dengan metode lainnya seperti diikuti dengan riset pasar atau analogi sejarah.

Analogi Sejarah

Metode yang melihat dan membandingkan pola *life-cycle* suatu produk serupa yang ingin dianalisis.

Metode ini digunakan efektif apabila horizon waktu peramalan dalam jangka menengah dan panjang.

Hal tersebut karena sebuah produk biasanya memiliki daur hidup yang panjang. Anomali nilai produk juga biasanya berlangsung pada jangka waktu yang lama.

Personal Insight



Metode peramalan yang sangat sederhana dan sudah ditinggalkan oleh perusahaan-perusahaan besar.



Metode ini digunakan dengan berdiskusi atau melihat referensi dari orang-orang yang berpengalaman dan ahli di bidangnya.

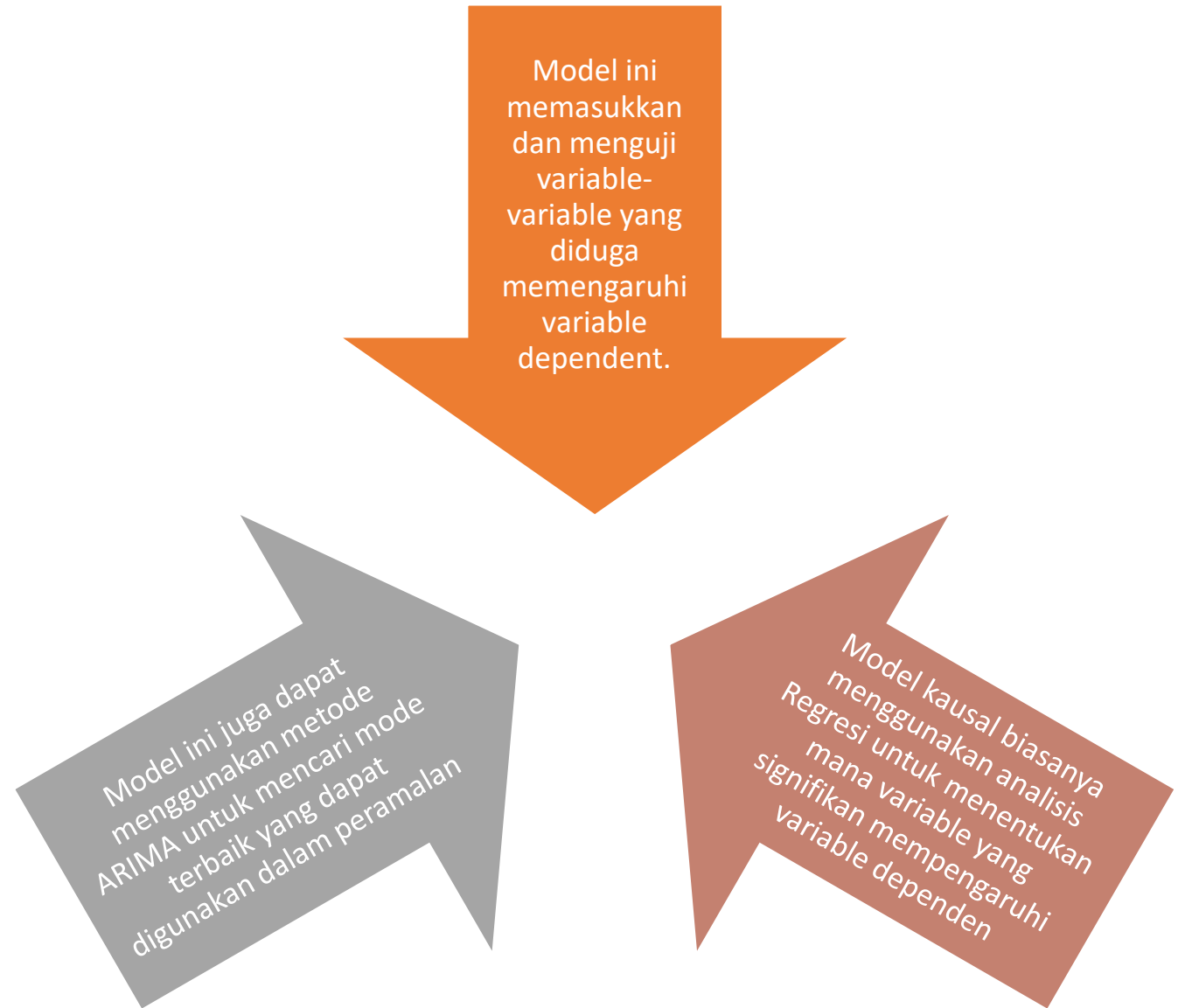


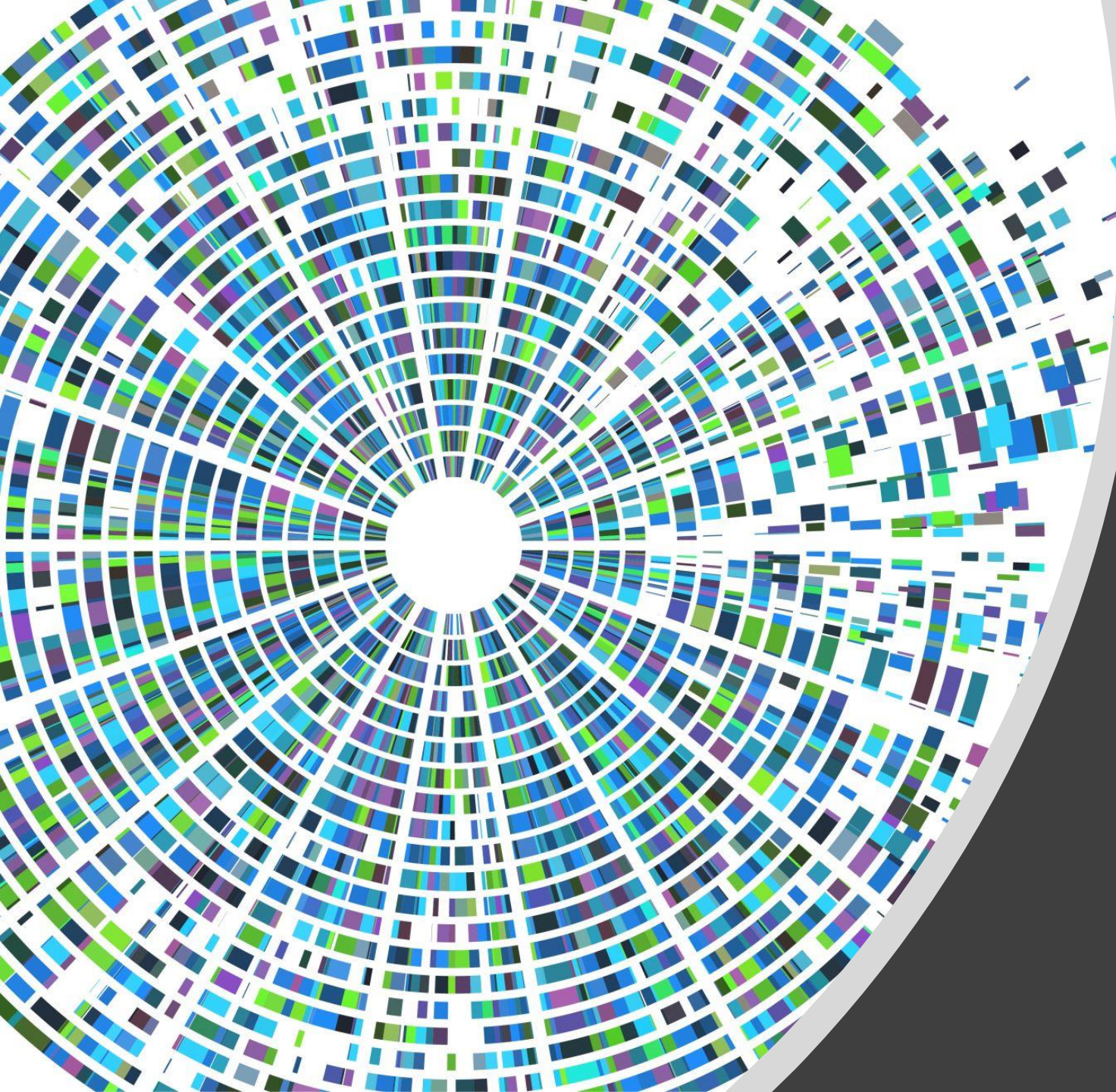
Misalnya melalui jurnal, artikel, wawancara, *mentoring*, atau sumber personal lainnya yang bisa menjadi acuan *forecasting*.



Teknik Peramalan Model Kausal

Model Kausal





Teknik Peramalan Model Time Series

Model ini berusaha
untuk memprediksi
masa depan dengan
menggunakan data
histori

Model Time Series

Data Time Series



Komponen time series

Stationer

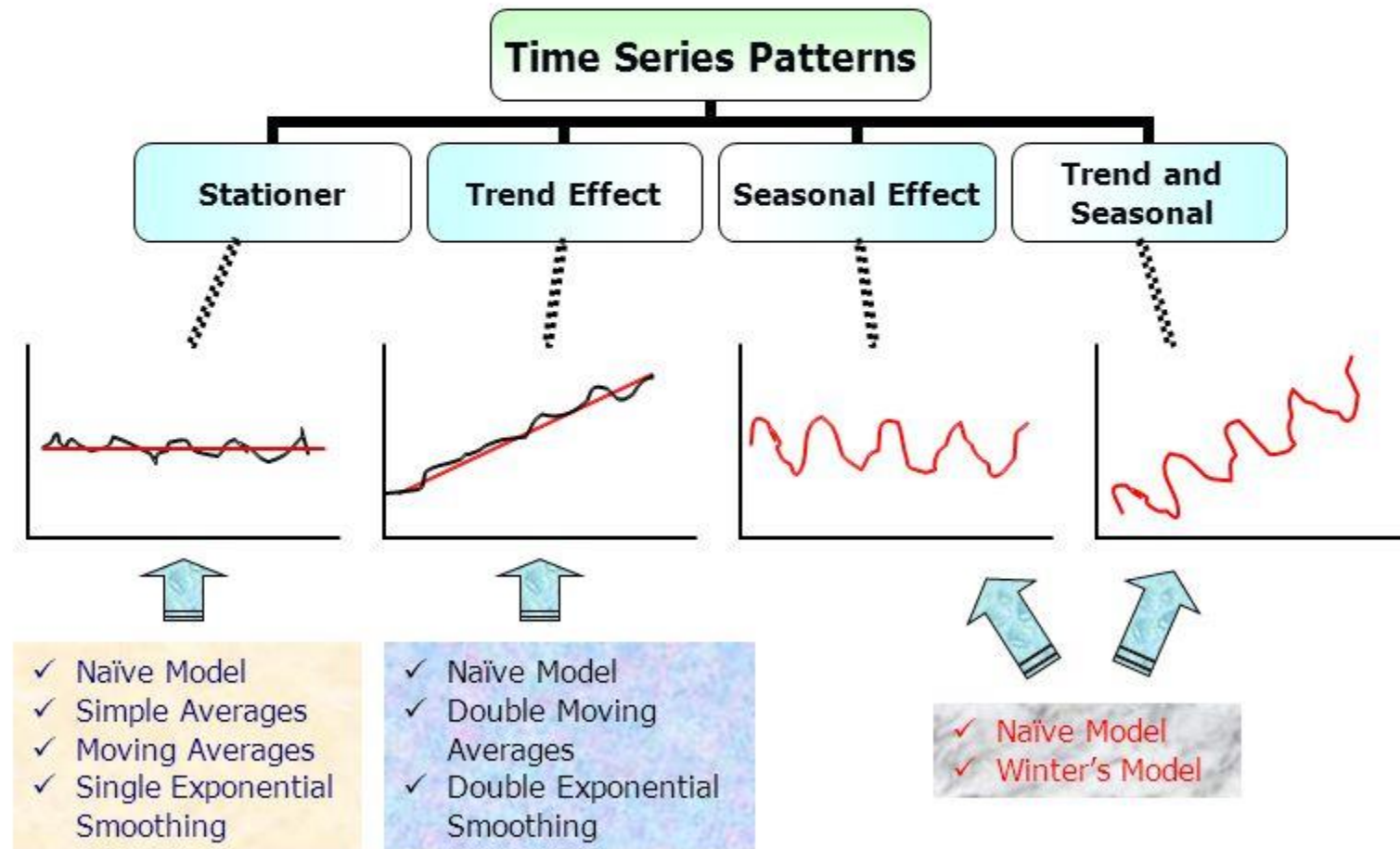
Trend

Siklikal

Musiman (Seasonal)

Tak Beraturan

Kaitan Pola Data dengan beberapa Metode Peramalan



Stationer (Konstan)

Pada pola ini data bergerak di sekitar rata-rata secara stabil. Pola semacam ini cenderung terjadi pada periode waktu yang pendek sampai menengah.

Pola ini terjadi apabila data mengalami fluktuasi secara konstan. Data tidak meningkat maupun turun selama kurun waktu tertentu.

Trend

Yaitu komponen jangka Panjang yang mendasari pertumbuhan (atau penurunan) suatu data runtut waktu.

Merupakan pergerakan data sedikit demi sedikit meningkat atau menurun

Pola ini dipengaruhi oleh faktor budaya, perubahan pendapatan, dan perubahan populasi.

Siklikal

Yaitu suatu pola dalam data yang terjadi setiap beberapa tahun. Fluktuasi atau siklus dari data runtut waktu akibat perubahan kondisi ekonomi

Musiman (seasonal)

Yaitu pola data yang berulang pada kurun waktu tertentu


Fluktuasi musiman yang sering dijumpai pada data kuartalan, bulanan atau mingguan

Tak Beraturan

Yaitu pola acak yang disebabkan oleh peristiwa yang tidak bisa diprediksi atau tidak beraturan

Referensi

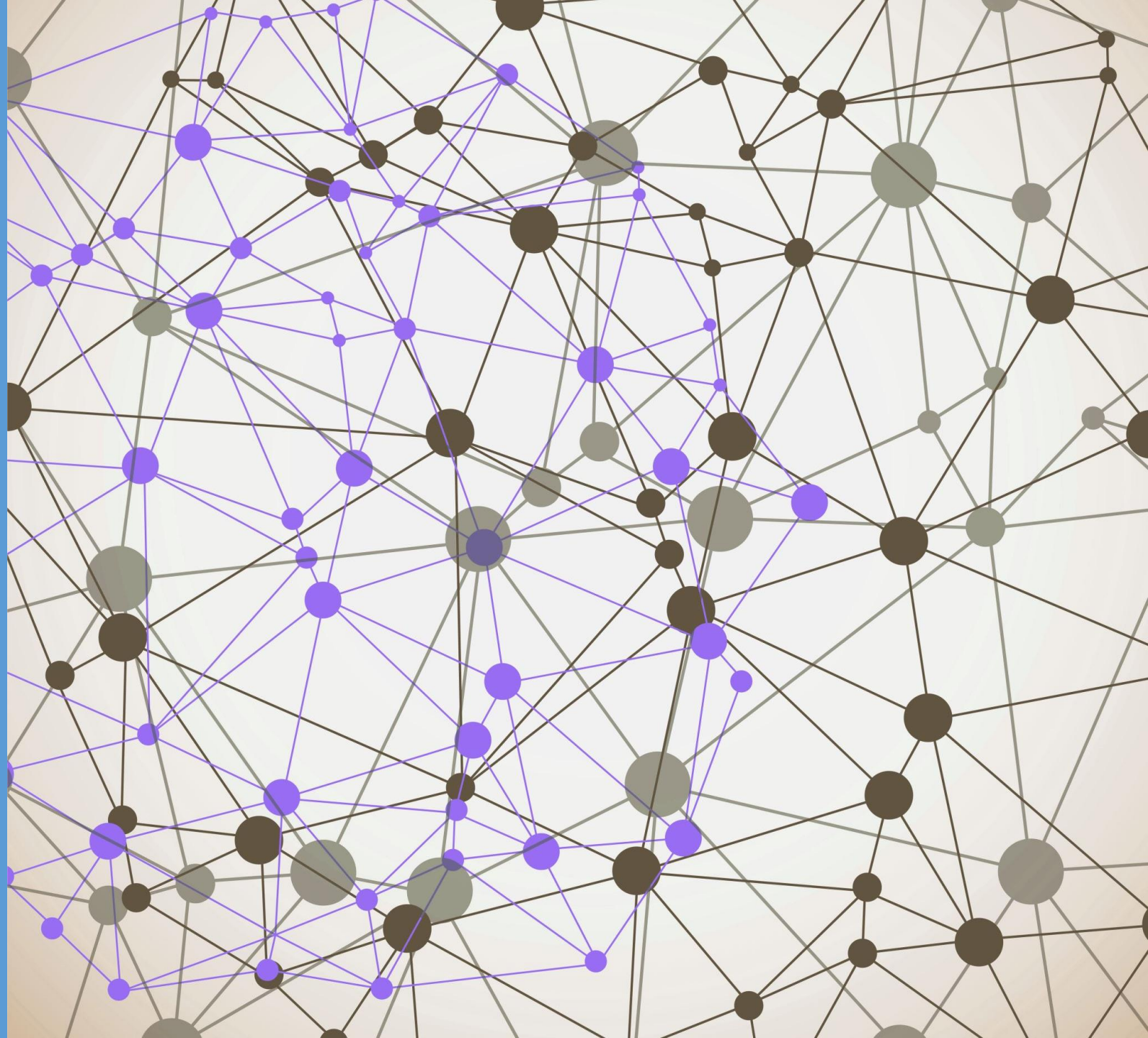
- <https://slideplayer.info/slide/3639303/12/images/8/Kaitan+Pola+Data+dengan+beberapa+Metode+Peramalan.jpg>
- <https://www.rusdionoconsulting.com/forecasting/>
- <https://www.jurnal.id/id/blog/mengenal-metode-forecasting-untuk-kepentingan-bisnis-anda/>

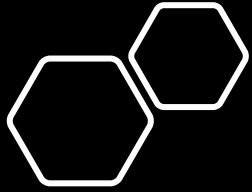


Di seluruh industri, organisasi **umumnya menggunakan** data deret waktu (**Time series**), yang berarti **setiap informasi yang dikumpulkan selama interval waktu reguler**, dalam operasinya. Contohnya antara lain harga saham harian, tingkat konsumsi energi, metrik interaksi media sosial, dan permintaan ritel. **Menganalisis data deret waktu menghasilkan wawasan seperti tren, pola musiman, dan prakiraan peristiwa masa depan** yang dapat membantu menghasilkan **keuntungan**. Misalnya, dengan memahami tren musiman permintaan produk retail, perusahaan dapat merencanakan promosi untuk memaksimalkan penjualan sepanjang tahun.



Praktek Project UAS PPDS

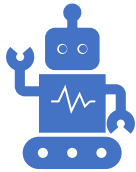




Petunjuk

- Perhatikan setiap perintah yang ada didalam slide berikutnya dan kerjakan sesuai dengan petunjuk codingan
- Setiap petunjuk tersebut kalian akan diberikan keterangan untuk mempermudah kalian dalam memahami project kalian
- Kerjakan hasil UAS kalian pada file word dengan format : **UAS_NIM_NAMA LENGKAP_KELAS A/B.pdf**
- Hasil dari pengerjaan kalian, dikumpulkan di : <https://assess1.ipb.ac.id/course/view.php?id=122> maksimum pengumpulan **Hari Minggu, 11 Desember 2022 pukul 21.00 WIB**. Apabila melewati batas waktu maka **dianggap tidak mengerjakan UAS**
- **TIDAK MENCONTEK!** Yang mencontek dan dicontekkan akan diberikan nilai nol
- Selamat mengerjakan!

Mempersiapkan data



To start, import Library Pandas
dan baca data **airline passenger**
Kedalam data frame:



Download disini : [Download Dataset](#)



Selanjutnya, display 5 baris
pertama menggunakan perintah
method head()



Tampilkan hasilnya

```
import pandas as pd
df = pd.read_csv("AirPassengers.csv")
print(df.head())
```



	Month	#Passengers
0	1949-01	112
1	1949-02	118
2	1949-03	132
3	1949-04	129
4	1949-05	121

Kita dapat melihat bahwa data berisi kolom berlabel “#Passengers ” yang berisi tanggal. Di kolom itu, tanggal diformat sebagai year-month. Kami juga melihat bahwa data dimulai pada tahun 1949.

Kolom kedua diberi label “# Passengers”, dan berisi jumlah penumpang untuk year-month. Mari kita lihat lima data terakhir menggunakan metode tail() :

codingan

```
print(df.tail())
```

Jawablah
pertanyaan
berikut dari
hasil
codingan
Tail()

- Pada bulan dan tahun berapa data terakhir ditunjukkan?
- Pada bulan apa yang memberikan jumlah penumpang terbanyak?
- Berapa total data pada data penumpang ini?

	Month	#Passengers
139	1960-08	606
140	1960-09	508
141	1960-10	461
142	1960-11	390
143	1960-12	432

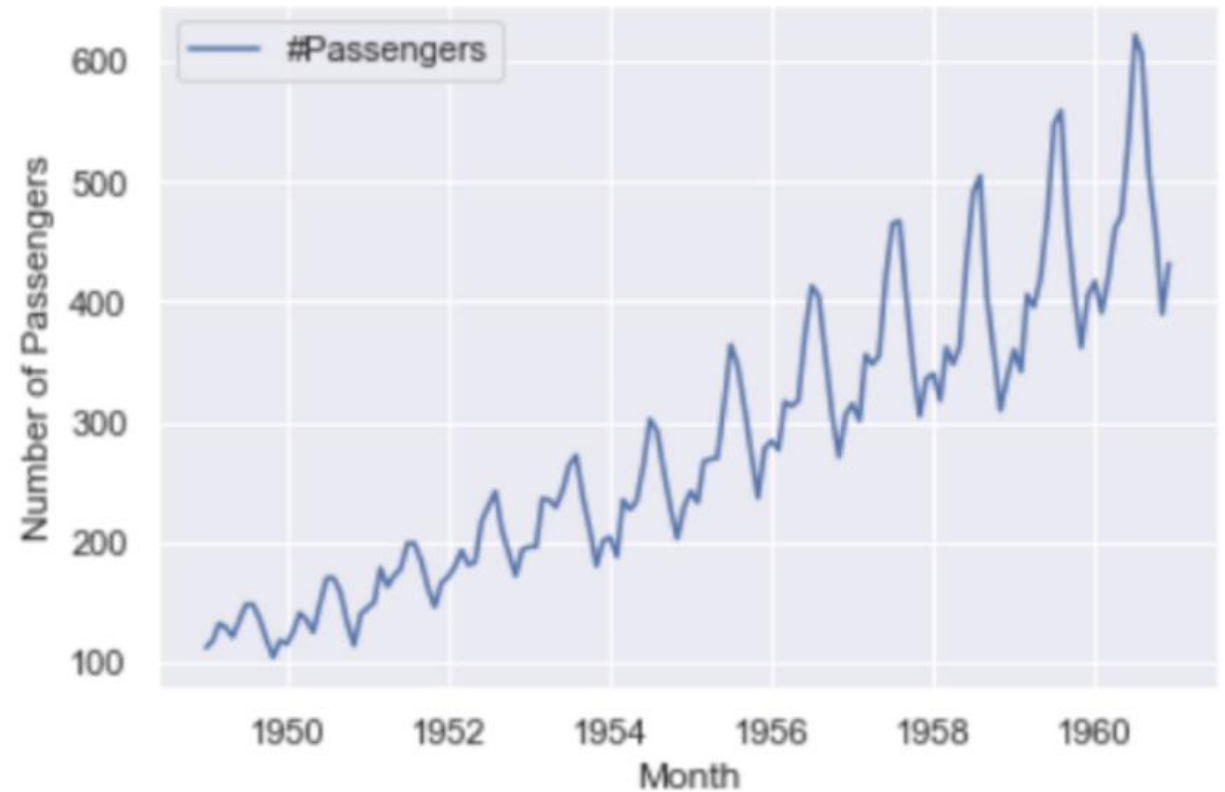
- Selanjutnya, kita convert kolom bulan kedalam objek datetime.
- Ini akan memungkinkannya untuk secara terprogram menarik nilai waktu seperti tahun atau bulan untuk setiap record. Untuk melakuk
- an ini menggunakan method `to_datetime()` library Panda
- Perhatikan bahwa proses ini secara otomatis memasukkan hari pertama setiap bulan, yang pada dasarnya merupakan nilai dummy karena kami tidak memiliki data penumpang harian.

```
df['Month'] = pd.to_datetime(df['Month'],  
format='%Y-%m') print(df.head())
```

	Month	#Passengers
0	1949-01-01	112
1	1949-02-01	118
2	1949-03-01	132
3	1949-04-01	129
4	1949-05-01	121

- Selanjutnya, generate plot time series menggunakan Seaborn dan Matplotlib
- Library ini digunakan untuk melakukan visualisasi pada data time series.

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.lineplot(df)
plt.ylabel("Number of Passengers")
```



Stationarity



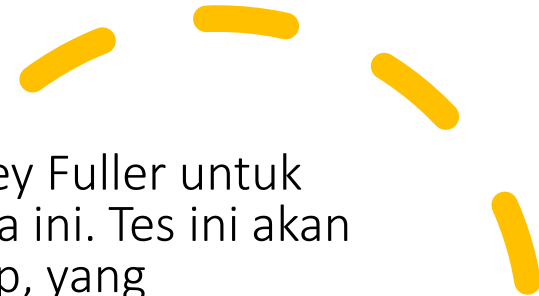
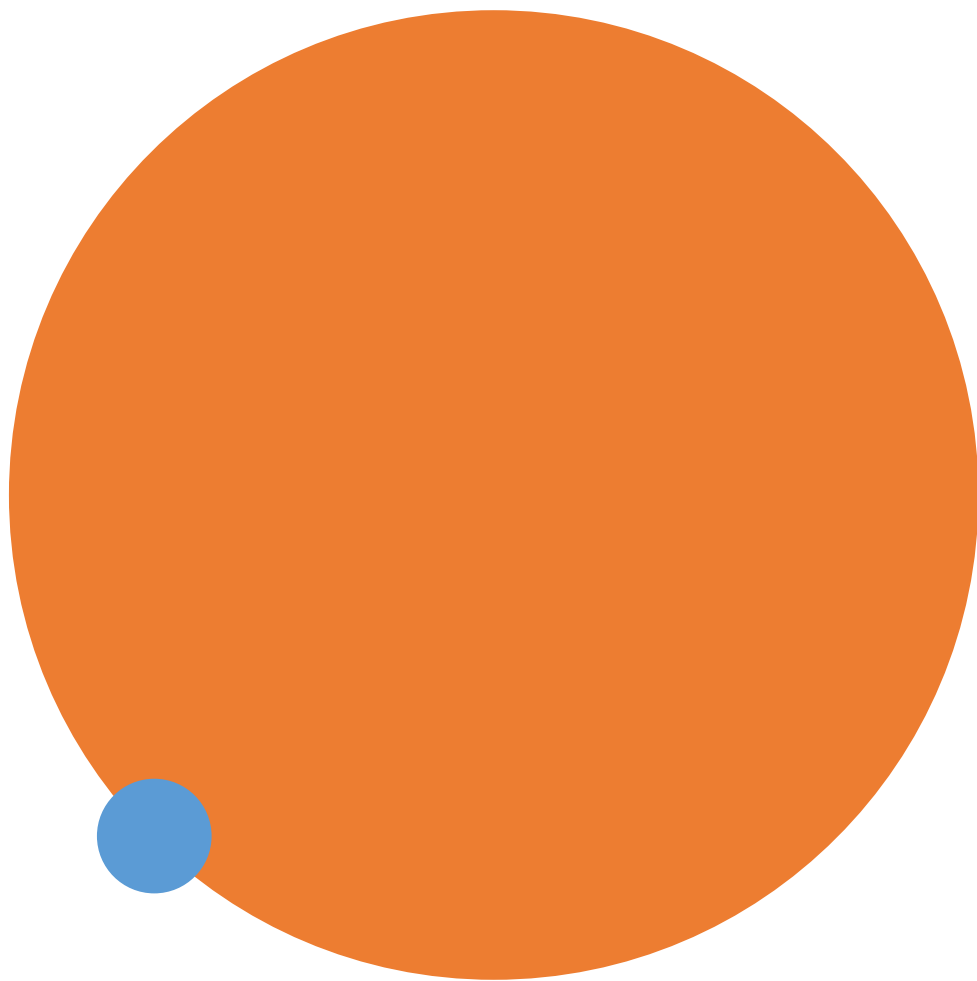
Stationarity adalah bagian penting dari analisis deret waktu. Sederhananya, Stationarity berarti bahwa cara perubahan data deret waktu adalah konstan.



Deret waktu stasioner tidak akan memiliki tren atau pola musiman. Anda harus memeriksa stasioneritas karena hal ini tidak hanya membuat pemodelan deret waktu menjadi lebih mudah, tetapi juga merupakan asumsi dasar dalam banyak metode deret waktu.



Secara khusus, stasioneritas diasumsikan untuk berbagai metode perkiraan deret waktu termasuk autoregressive moving average (ARMA), ARIMA dan Seasonal ARIMA (SARIMA).



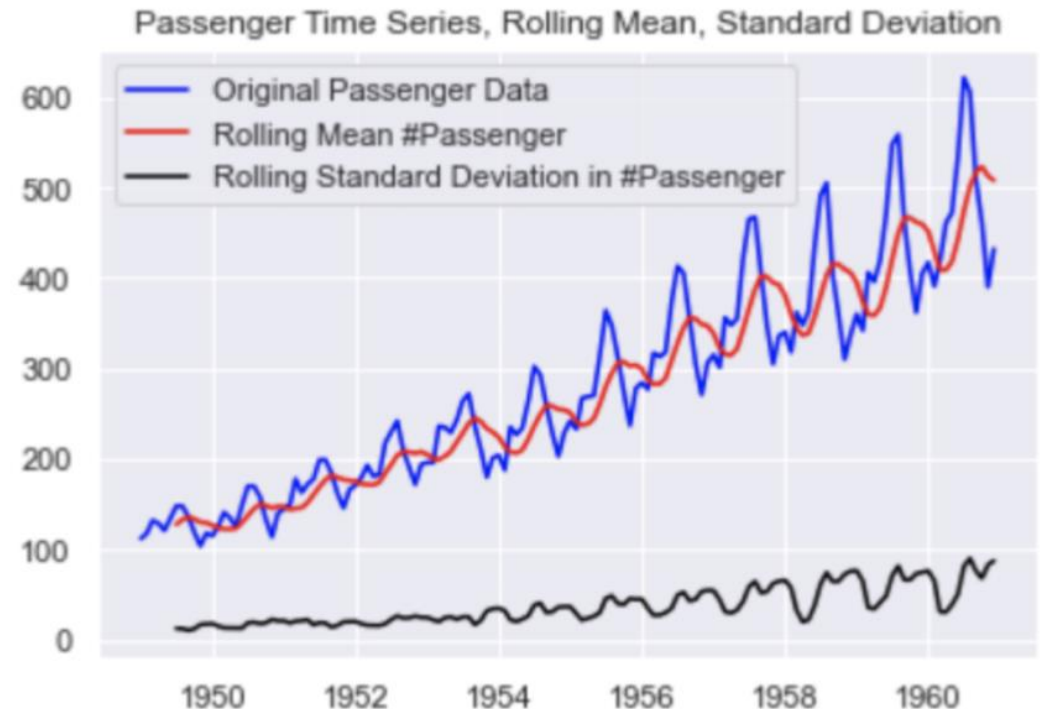
- Kami akan menggunakan tes Dickey Fuller untuk memeriksa Stationarity dalam data ini. Tes ini akan menghasilkan nilai kritis dan nilai-p, yang memungkinkan kita menerima atau menolak hipotesis nol bahwa tidak ada stasioneritas.
- Jika kita menolak hipotesis nol, itu berarti kita menerima alternatifnya, yang menyatakan bahwa ada stasioneritas.
- Nilai-nilai ini memungkinkan kita untuk menguji sejauh mana nilai sekarang berubah dengan nilai masa lalu. Jika tidak ada Stationarity dalam kumpulan data, perubahan nilai sekarang tidak akan menyebabkan perubahan signifikan pada nilai masa lalu.

Untuk memulai, kita hitung rata-rata 7 bulan

Selanjutnya dilakukan overlay dan standar deviasi terhadap data 7 bulan tersebut

Dan gunakan Matplotlib untuk memvisualisasikan hasilnya

```
rolling_mean = df.rolling(7).mean()
rolling_std = df.rolling(7).std()
plt.plot(df, color="blue", label="Original Passenger Data")
plt.plot(rolling_mean, color="red", label="Rolling Mean Passenger Number")
plt.plot(rolling_std, color="black", label = "Rolling Standard Deviation in Passenger Number")
plt.title("Passenger Time Series, Rolling Mean, Standard Deviation")
plt.legend(loc="best")
```



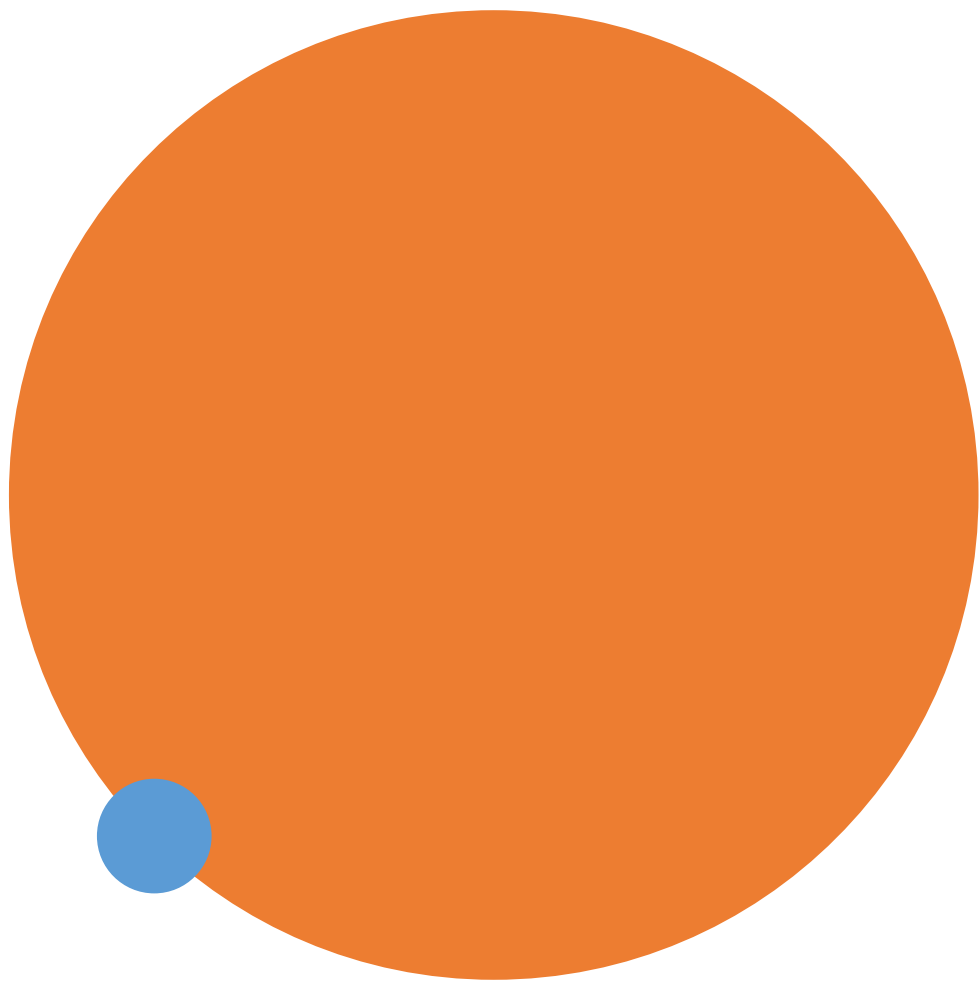
- Selanjutnya, mari impor tes Dickey-Fuller yang ditambah dari paket statsmodels.
- Selanjutnya, mari berikan bingkai data kita ke metode adfuller. Di sini, kami menetapkan parameter autolag sebagai “AIC”, yang berarti lag dipilih untuk meminimalkan kriteria informasi:
- Tampilkan hasilnya

```
from statsmodels.tsa.stattools import adfuller

adft = adfuller(df, autolag="AIC")

output_df = pd.DataFrame({"Values": [adft[0], adft[1], adft[2], adft[3],
adft[4]['1%'], adft[4]['5%'], adft[4]['10%']], "Metric": ["Test
Statistics", "p-value", "No. of lags used", "Number of observations used",
"critical value (1%)", "critical value (5%)", "critical value (10%)"]})
print(output_df)
```

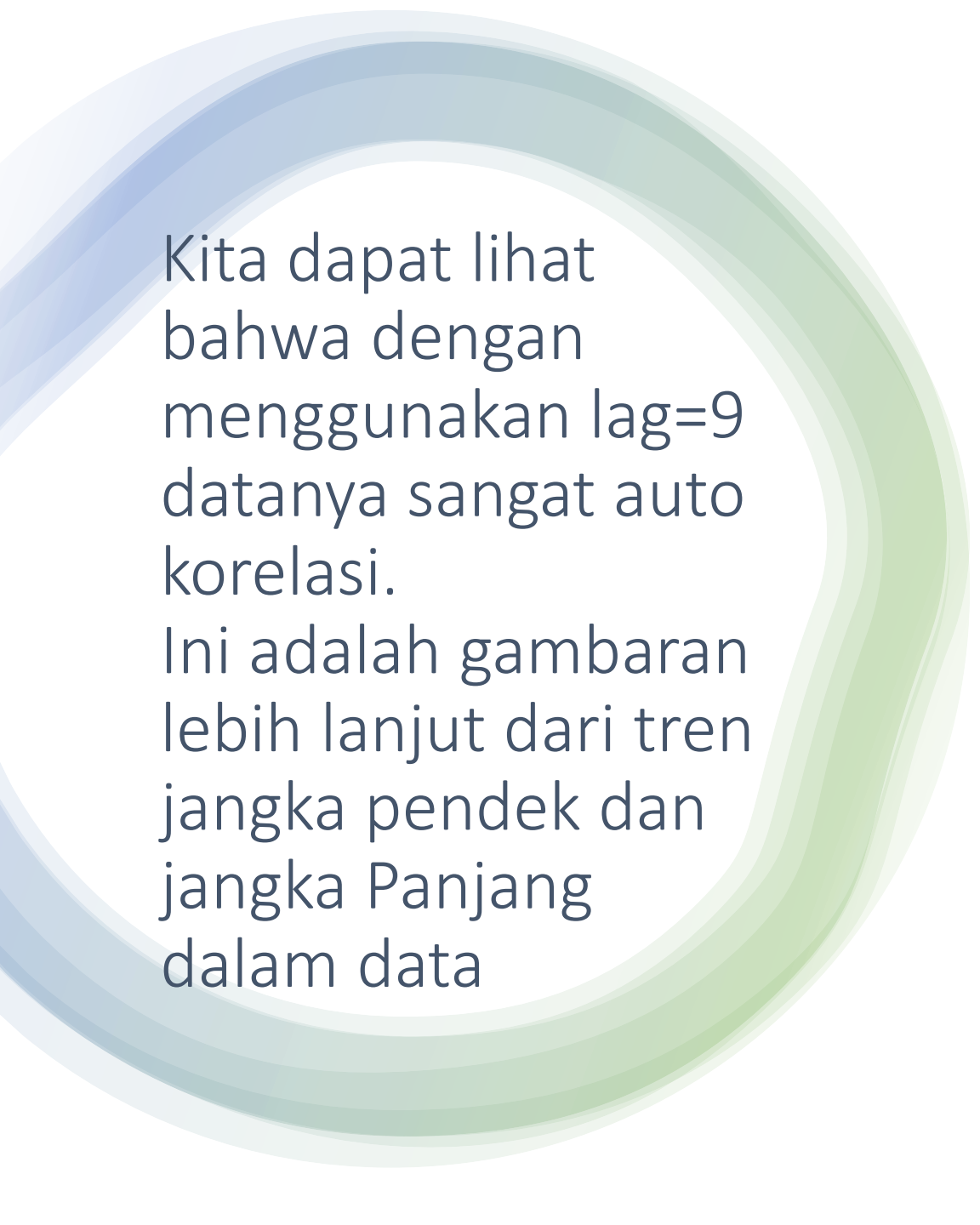
	Values	Metric
0	0.815369	Test Statistics
1	0.991880	p-value
2	13.000000	No. of lags used
3	130.000000	Number of observations used
4	-3.481682	critical value (1%)
5	-2.884042	critical value (5%)
6	-2.578770	critical value (10%)



- Kita dapat melihat bahwa data kita tidak stasioner dari fakta bahwa nilai p kita lebih besar dari 5 persen dan statistik uji lebih besar dari nilai kritis. Kami juga dapat menarik kesimpulan ini dari pemeriksaan data, karena kami melihat tren peningkatan jumlah penumpang yang jelas.
- **Tambahkan Kembali kesimpulan yang bisa kalian berikan!**

Autocorrelation

- Memeriksa data deret waktu untuk autokorelasi dengan Python adalah bagian penting lainnya dari proses analitik. Ini adalah ukuran seberapa korelasi data deret waktu pada titik waktu tertentu dengan nilai masa lalu, yang memiliki implikasi besar di banyak industri.
- Misalnya, jika data penumpang kita memiliki autokorelasi yang kuat, kita dapat mengasumsikan bahwa jumlah penumpang yang tinggi hari ini menunjukkan kemungkinan yang kuat bahwa besok juga akan tinggi.
- Frame data Pandas memiliki metode autokorelasi yang dapat kita gunakan untuk menghitung autokorelasi pada data penumpang kita.
- Mari kita lakukan ini untuk jeda (lag) satu bulan, enam bulan dan Sembilan bulan dengan menggunakan $\text{lag} = 1$, $\text{lag} = 6$, $\text{lag} = 9$



Kita dapat lihat bahwa dengan menggunakan lag=9 datanya sangat auto korelasi.
Ini adalah gambaran lebih lanjut dari tren jangka pendek dan jangka Panjang dalam data

```
autocorrelation_lag1 = df['#Passengers'].autocorr(lag=1)  
print("One Month Lag: ", autocorrelation_lag1)
```

```
autocorrelation_lag3 = df['#Passengers'].autocorr(lag=3)  
print("Three Month Lag: ", autocorrelation_lag3)
```

```
autocorrelation_lag6 = df['#Passengers'].autocorr(lag=6)  
print("Six Month Lag: ", autocorrelation_lag6)
```

```
autocorrelation_lag9 = df['#Passengers'].autocorr(lag=9)  
print("Nine Month Lag: ", autocorrelation_lag9)
```

One Month Lag: 0.9601946480498523

Three Month Lag: 0.837394765081794

Six Month Lag: 0.7839187959206183

Nine Month Lag: 0.8278519011167601

Decomposition

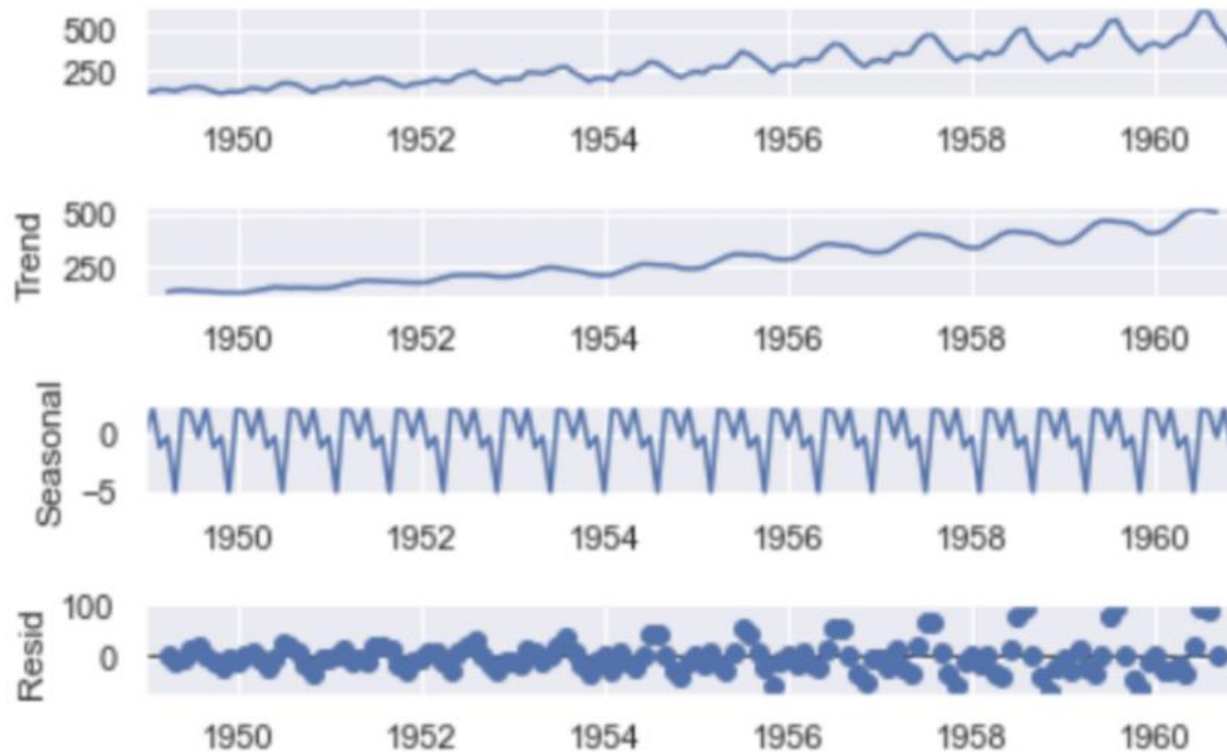
Dekomposisi tren adalah cara lain yang berguna untuk memvisualisasikan tren dalam data deret waktu. Untuk melanjutkan, mari impor `seasonal_decompose` dari paket `statsmodels`

Selanjutnya masukkan data frame yang kita sudah miliki kedalam method `seasonal_decompose`

plotkan hasilnya.

```
from statsmodels.tsa.seasonal import
seasonal_decompose
decompose =
seasonal_decompose(df['#Passengers'], model=
'additive', period=7)
decompose.plot()
plt.show()
```

Hasil



Dari plot ini, kita dapat dengan jelas melihat tren peningkatan jumlah penumpang dan pola musiman dalam naik turunnya nilai setiap tahunnya.

Tambahkan lagi hasil kesimpulan dari yang kalian dapatkan. Boleh melihat dari sisi tipe-tipe forecastingnya, data dibagian tertentu, arah grafik, dst.

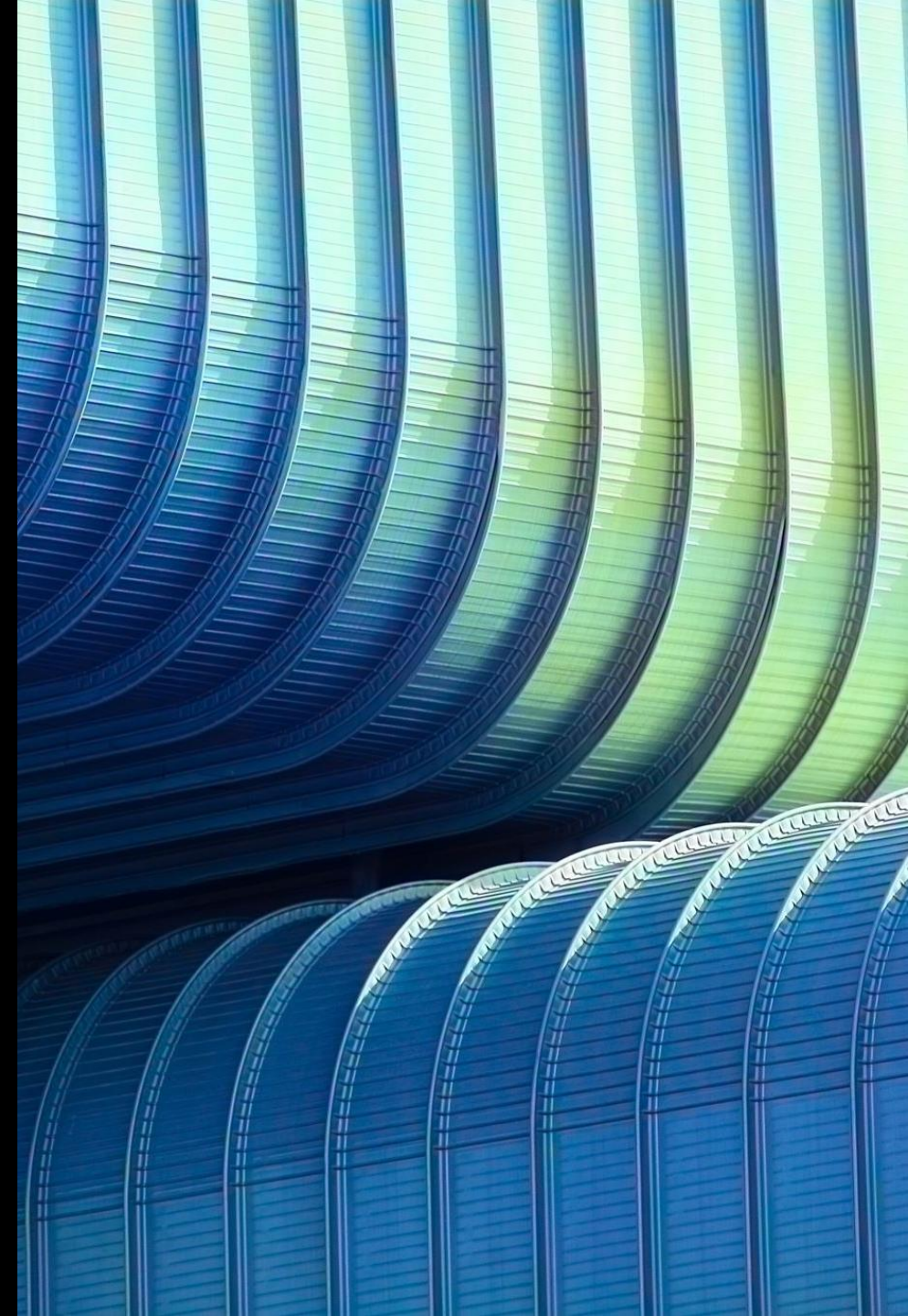
Forecasting

Peramalan deret waktu memungkinkan kita memprediksi nilai masa depan dalam deret waktu berdasarkan data saat ini dan masa lalu.

Di sini, kita akan menggunakan metode ARIMA untuk meramalkan jumlah penumpang, yang memungkinkan kita meramalkan nilai masa depan dalam bentuk kombinasi linear dari nilai masa lalu.

kita akan menggunakan paket `auto_arima`, yang memungkinkan kami melupakan proses penyetelan hyperparameter yang memakan waktu.

Pertama, mari pisahkan data kita untuk pelatihan dan pengujian dan visualisasikan pemisahannya:



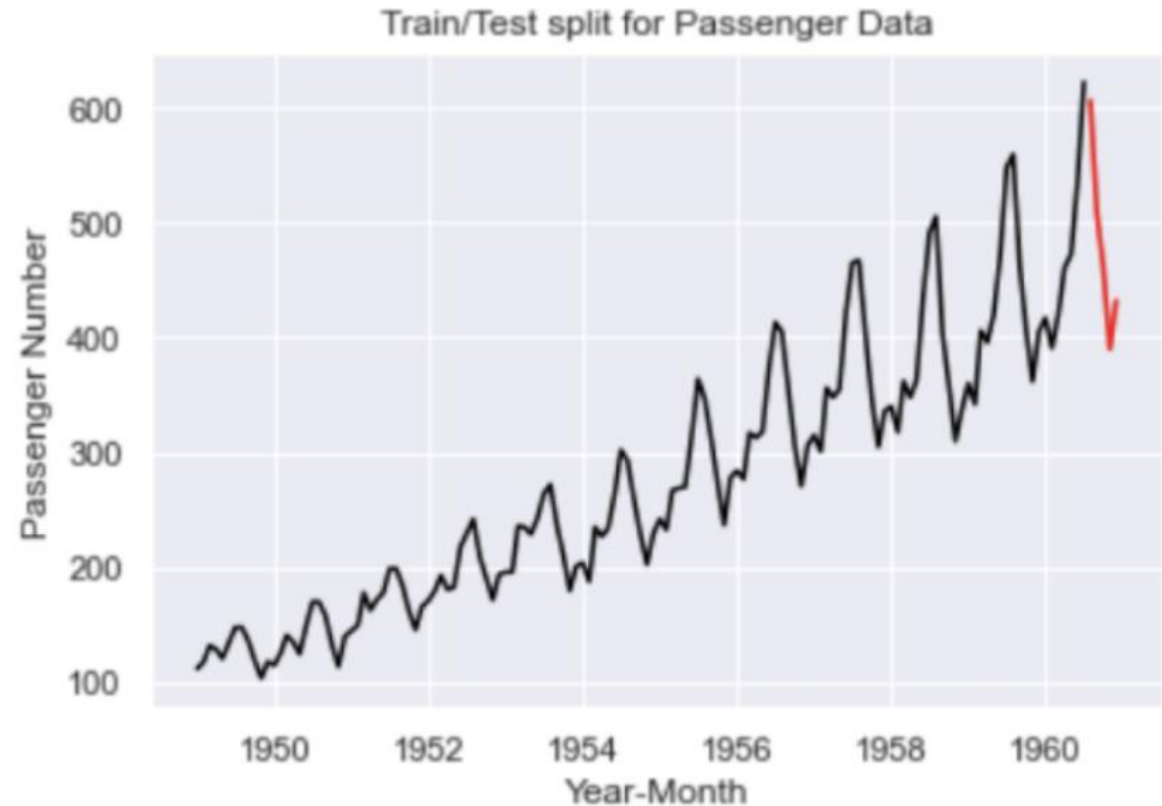
Codingan forecasting

```
df['Date'] = df.index
train = df[df['Date'] < pd.to_datetime("1960-08",
format='%Y-%m')]
train['train'] = train['#Passengers']
del train['Date']
del train['#Passengers']
test = df[df['Date'] >= pd.to_datetime("1960-08",
format='%Y-%m')]
del test['Date'] test['test'] = test['#Passengers']
del test['#Passengers'] plt.plot(train, color =
"black")
plt.plot(test, color = "red")
plt.title("Train/Test split for Passenger Data")
plt.ylabel("Passenger Number")
plt.xlabel('Year-Month')
sns.set()
plt.show()
```

Hasil forecasting

Garis hitam yang ditunjukkan pada grafik adalah data training, dan garis merah adalah data testingnya.

Menurut kalian, apa arti dari garis merah yang ditunjukkan? Jelaskan!



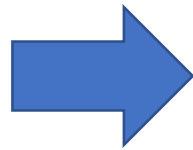
- Selanjutnya lakukan import `auto_arima` dari paket `pmdarima`, latih model kita dan hasilkan prediksinya (lakukan codingannya)

```
from pmdarima.arima
import auto_arima

model = auto_arima(train, trace=True,
error_action='ignore', suppress_warnings=True)
model.fit(train)

forecast = model.predict(n_periods=len(test))
forecast = pd.DataFrame(forecast, index =
test.index, columns=['Prediction'])
```

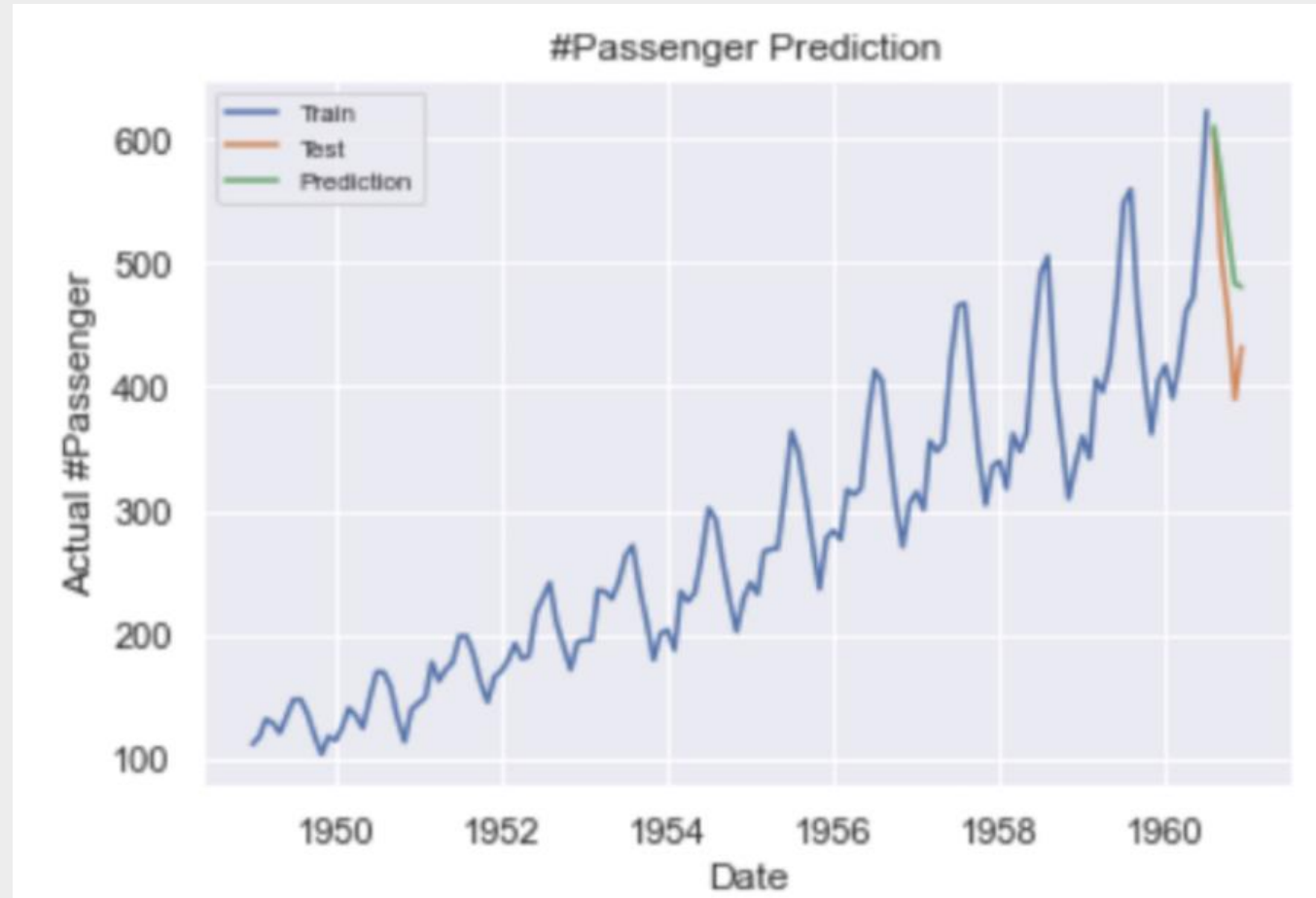
- Berikut ini hasilnya



```
Performing stepwise search to minimize aic
ARIMA(2,1,2)(0,0,0)[0] intercept : AIC=inf, Time=0.30 sec
ARIMA(0,1,0)(0,0,0)[0] intercept : AIC=1352.593, Time=0.01 sec
ARIMA(1,1,0)(0,0,0)[0] intercept : AIC=1340.702, Time=0.03 sec
ARIMA(0,1,1)(0,0,0)[0] intercept : AIC=1336.259, Time=0.05 sec
ARIMA(0,1,0)(0,0,0)[0] intercept : AIC=1352.415, Time=0.01 sec
ARIMA(1,1,1)(0,0,0)[0] intercept : AIC=1329.986, Time=0.07 sec
ARIMA(2,1,1)(0,0,0)[0] intercept : AIC=inf, Time=0.16 sec
ARIMA(1,1,2)(0,0,0)[0] intercept : AIC=inf, Time=0.15 sec
ARIMA(0,1,2)(0,0,0)[0] intercept : AIC=1335.098, Time=0.06 sec
ARIMA(2,1,0)(0,0,0)[0] intercept : AIC=1336.923, Time=0.03 sec
ARIMA(1,1,1)(0,0,0)[0] intercept : AIC=1329.407, Time=0.04 sec
ARIMA(0,1,1)(0,0,0)[0] intercept : AIC=1335.407, Time=0.02 sec
ARIMA(1,1,0)(0,0,0)[0] intercept : AIC=1339.796, Time=0.01 sec
ARIMA(2,1,1)(0,0,0)[0] intercept : AIC=1325.560, Time=0.05 sec
ARIMA(2,1,0)(0,0,0)[0] intercept : AIC=1336.364, Time=0.02 sec
ARIMA(3,1,1)(0,0,0)[0] intercept : AIC=1327.333, Time=0.09 sec
ARIMA(2,1,2)(0,0,0)[0] intercept : AIC=inf, Time=0.20 sec

Best model: ARIMA(4,1,3)(0,0,0)[0]
Total fit time: 3.978 seconds
```

- Tampilan output grafik



- Prediksi kita ditampilkan dalam warna hijau dan nilai sebenarnya ditampilkan dalam warna orange.
- Apa kesimpulan yang dapat kalian gambarkan dari hasilnya grafik tersebut

Terakhir, mari kita hitung root mean squared error (RMSE):

- Terakhir, mari kita hitung root mean squared error (RMSE):

```
from math
import sqrt from sklearn.metrics
import mean_squared_error rms =
sqrt(mean_squared_error(test,forecast))
print("RMSE: ", rms)
```

RMSE: 61.36535942376535

- **Apa fungsi RMSE pada forecasting? Kenapa dibutuhkan RMSE? Tuliskan hasil kesimpulan anda terkait hasil RMSE yang diberikan?**



Terima kasih

gemaparasti@apps.ipb.ac.id

