

# Digital Talent Scholarship 2022

## Time Series 1

Lead a sprint through the Machine Learning Track

# Agenda

- Time Series examples
- Common Pattern
- Introduction to Time Series
- Metrics for Evaluating Performance
- Forecasting

# Objektif Pembelajaran

- Memahami apa itu Time Series
- Mengenal jenis-jenis Time Series
- Mengetahui jenis-jenis kasus penggunaan Time Series yang tepat
- Membagi data menjadi train, validation, dan test
- Membuat Forecasting sederhana

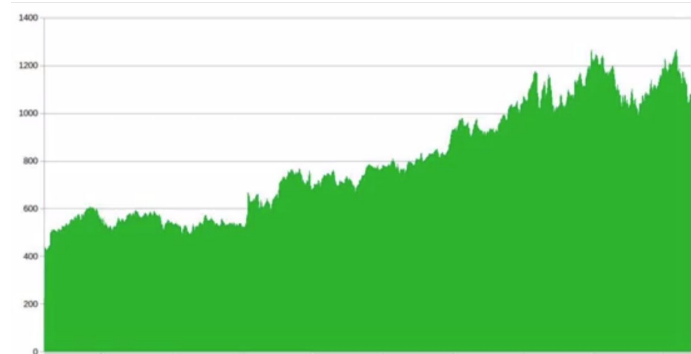
# Are your students ML-ready?

# Apa itu Time Series?

**Time Series** adalah serial dari kumpulan data yang teratur oleh urutan waktu. Frekuensi urutan waktu yang dimiliki oleh Time series data bisa meliputi tahunan, bulanan, jam, atau bahkan mili-detik. Selama data tersebut disimpan dalam urutan waktu, data itu adalah data Time Series.

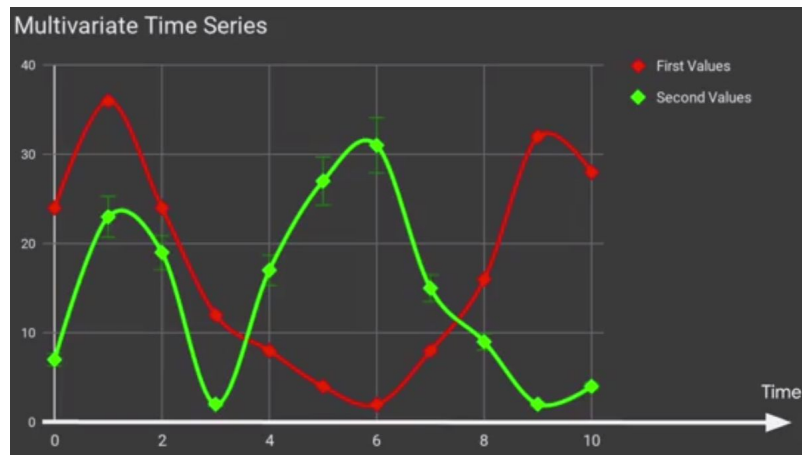
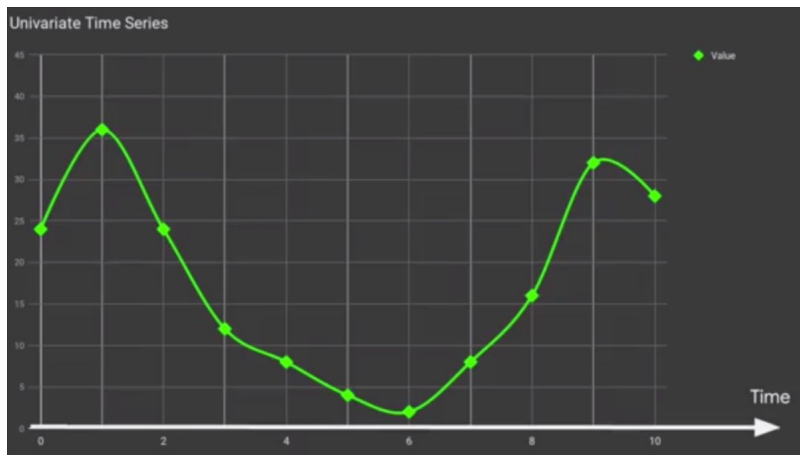
# Contoh Time Series

- Stock
- Cuaca
- Moore's Law



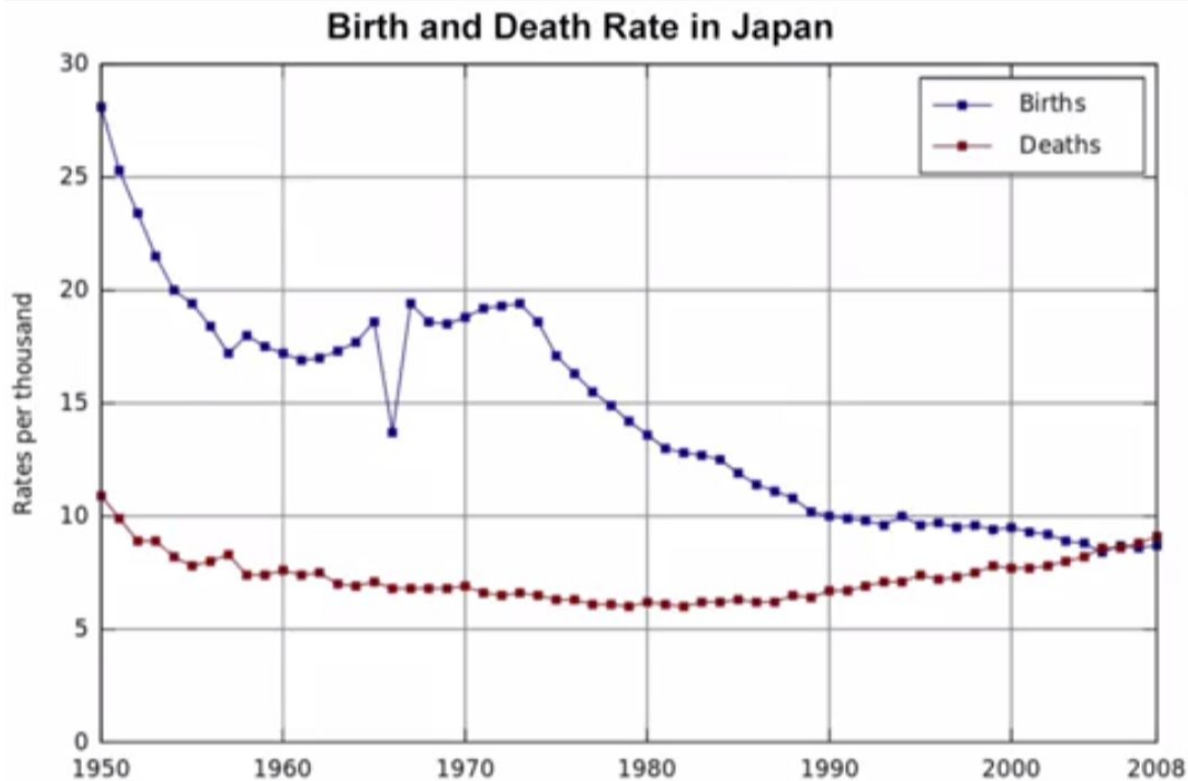
## Introduction to Time Series Data and Stationarity - YouTube

# Univariate & Multivariate Time Series

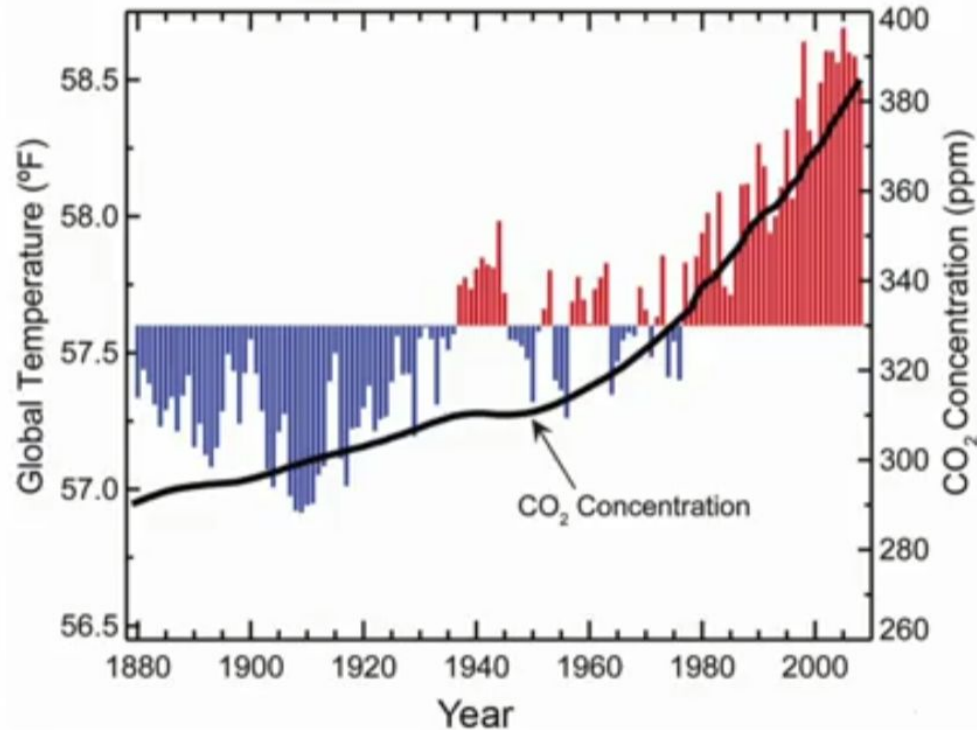




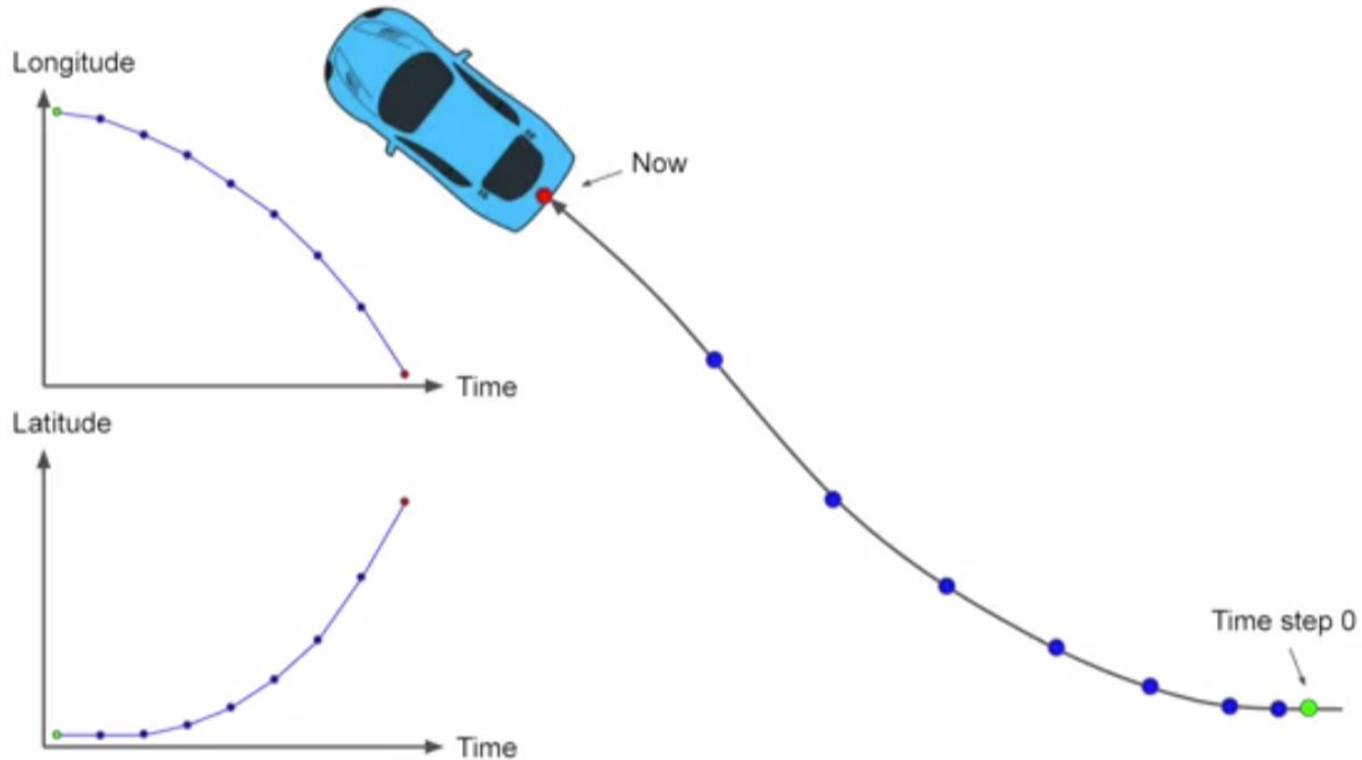
# Contoh Multivariate Time Series



# Contoh Multivariate Time Series



# Contoh Multivariate Time Series



# Time Series dalam Machine Learning

Time Series dengan Machine Learning kita bisa:

- **Forecast** : Memprediksi masa depan
- **Imputation** : Memproyeksikan masa lalu / Mengisi data yang kosong

Birth and Death Rate in Japan

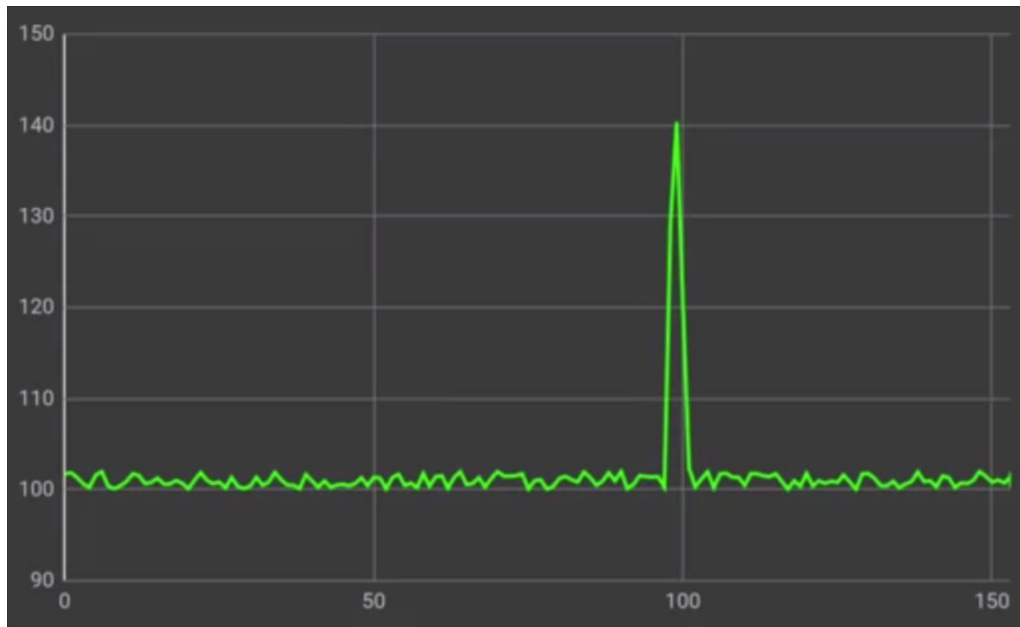


# Time Series dalam Machine Learning



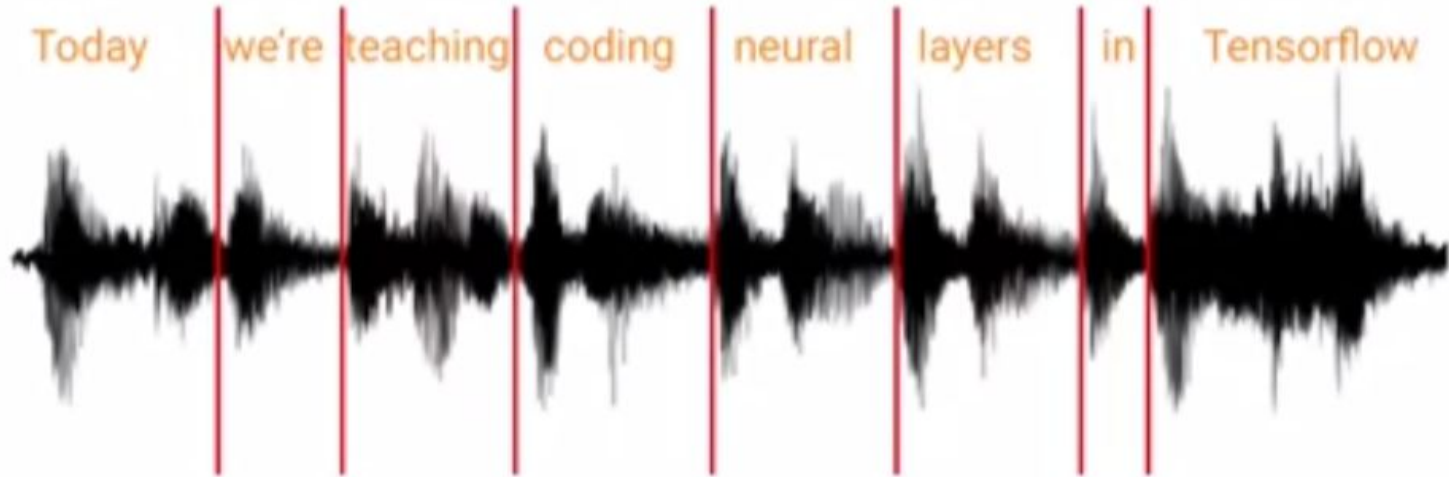
# Time Series dalam Machine Learning

- **Detect Anomalies** : Mendeteksi potensi service attack



# Time Series dalam Machine Learning

- **Sound Recognition** : Menganalisis suara untuk memisahkan kata-kata



# Pattern umum dalam Time Series

- Trend
- Seasonality
- Combination
- White noise
- Auto-correlated



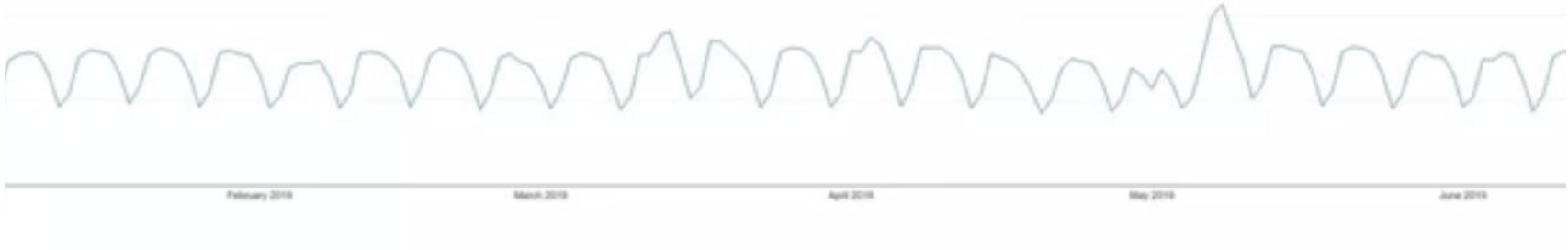
# Pattern umum dalam Time Series

- Trend



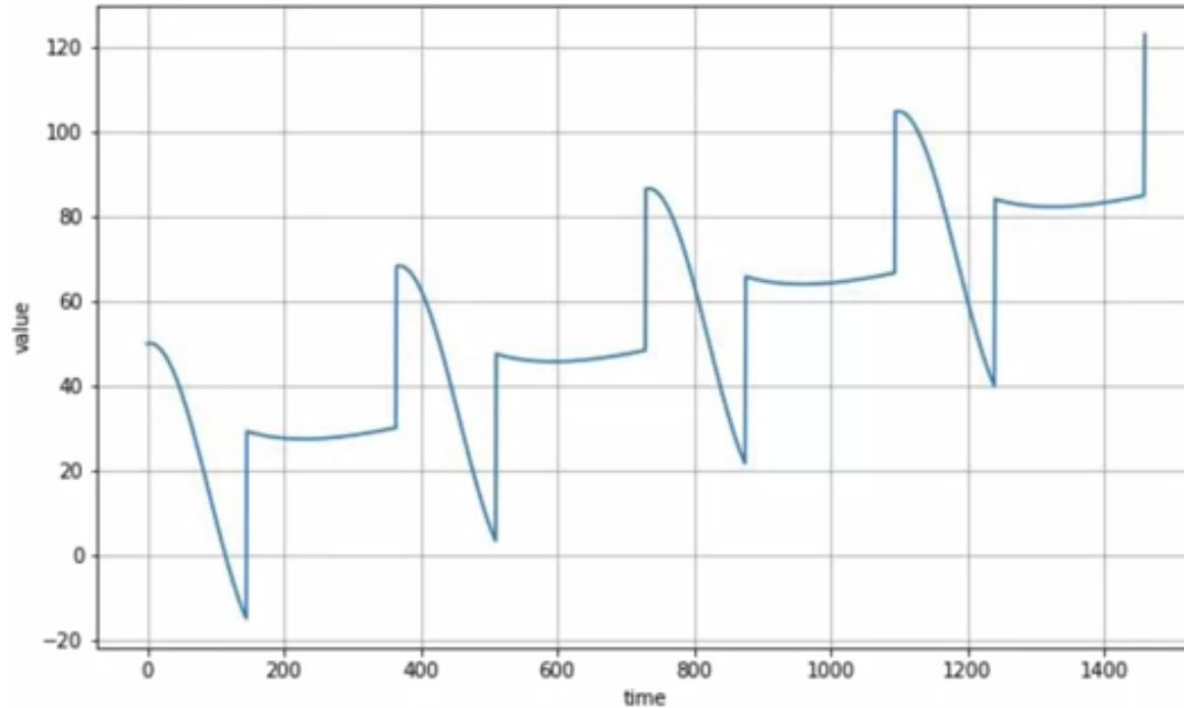
# Pattern umum dalam Time Series

- Seasonality



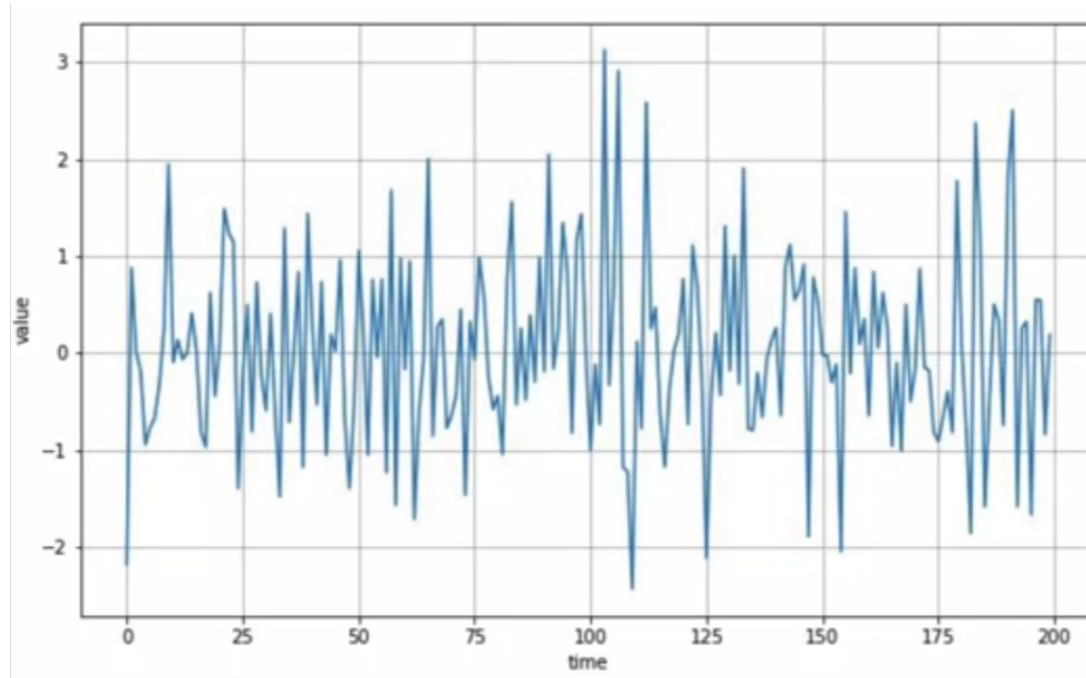
# Pattern umum dalam Time Series

- Trend + Seasonality



# Pattern umum dalam Time Series

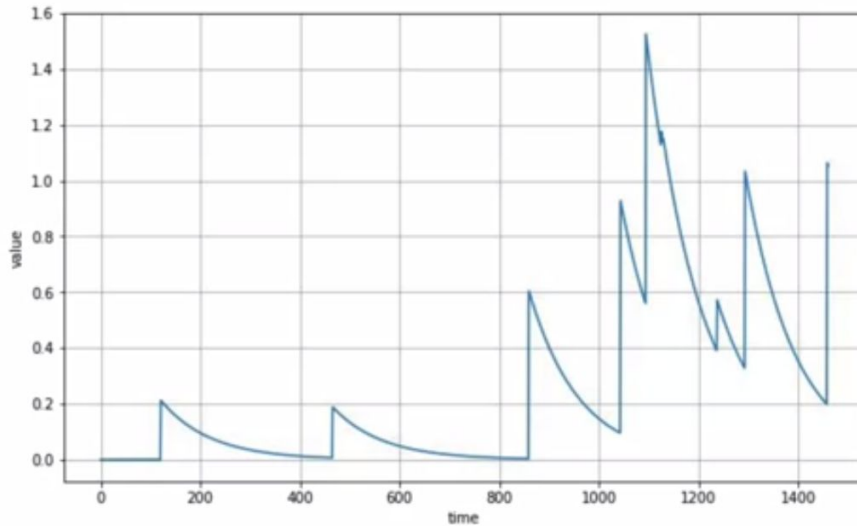
- White Noise



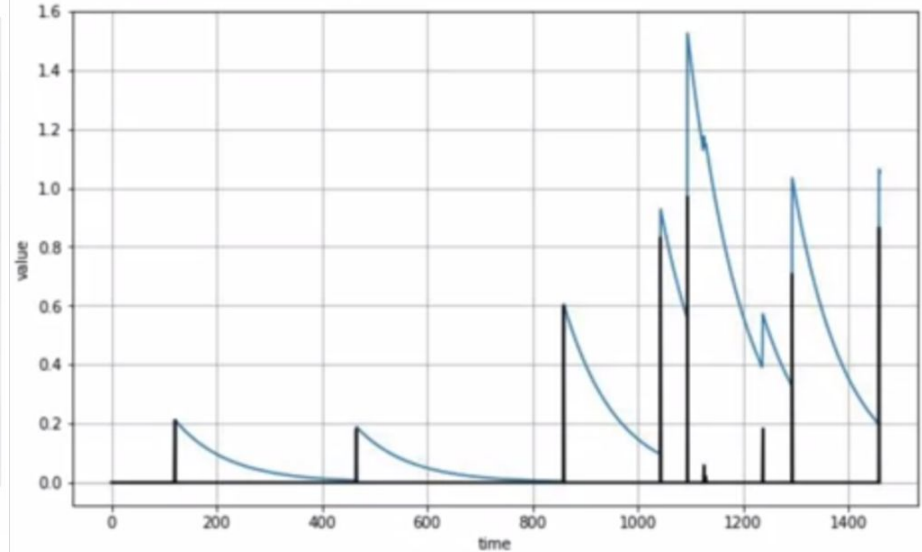
# Pattern umum dalam Time Series

- Autocorrelation

Autocorrelation

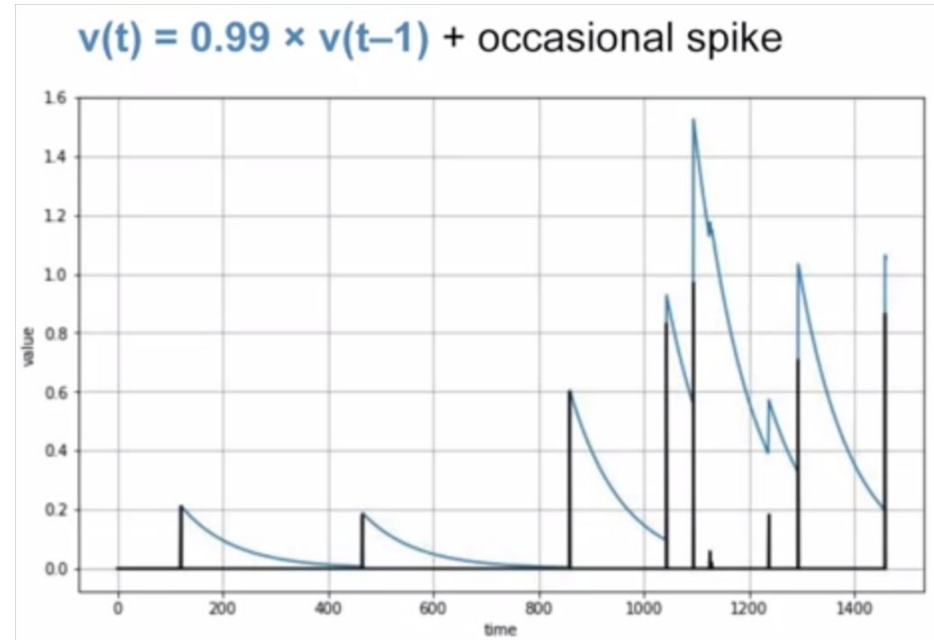


$$v(t) = 0.99 \times v(t-1) + \text{occasional spike}$$



# Pattern umum dalam Time Series

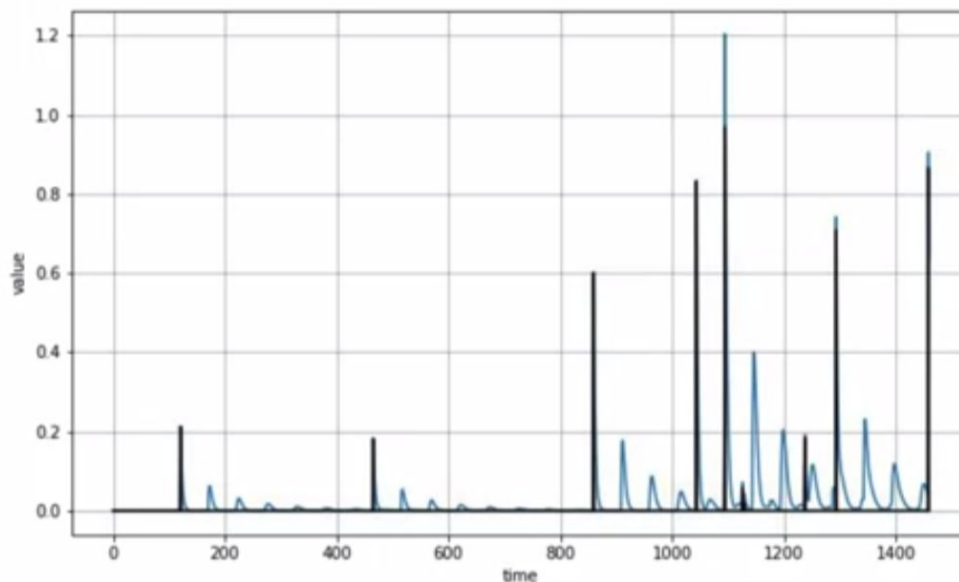
- Autocorrelation
  - **Lag** : Hasil dari Innovation (Garis biru)
  - **Innovation** : Tidak bisa diprediksi (Garis hitam)



# Pattern umum dalam Time Series

- Multiple Autocorrelation

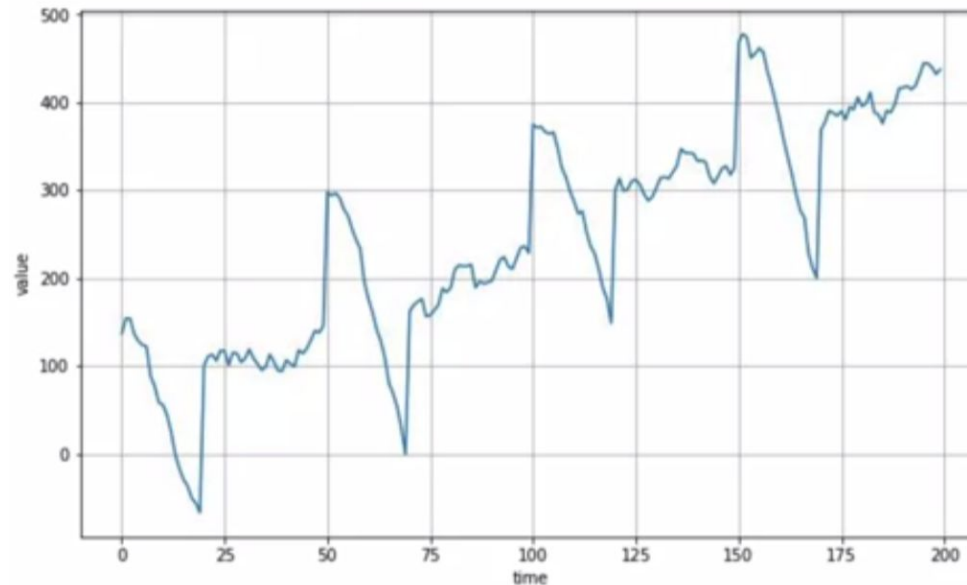
$$v(t) = 0.7 \times v(t-1) + 0.2 \times v(t-50) + \text{occasional spike}$$



# Pattern umum dalam Time Series

- Combination

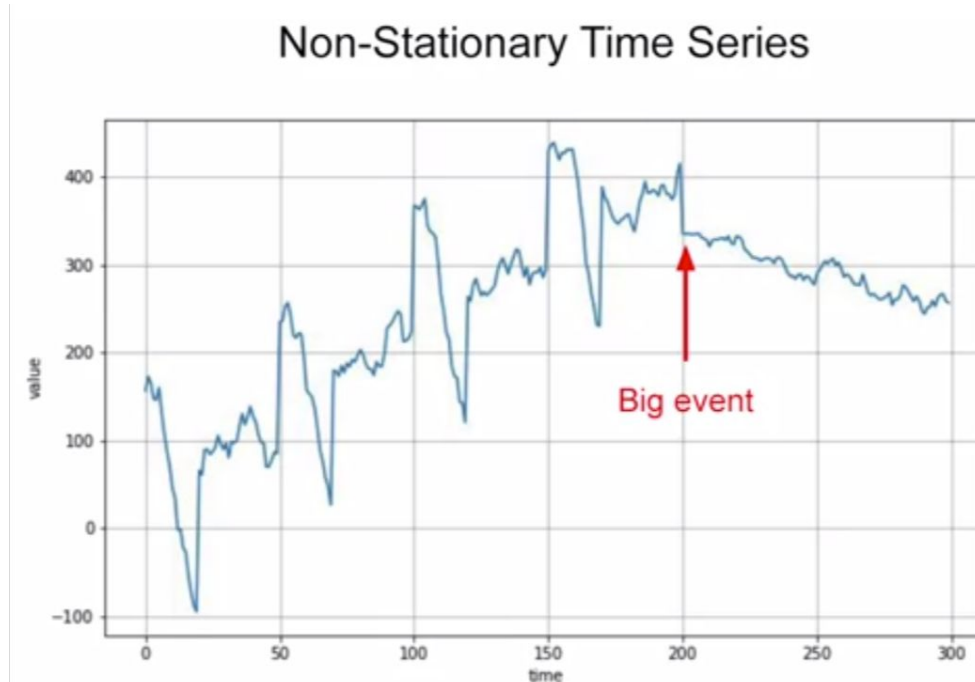
Trend + Seasonality + Autocorrelation + Noise





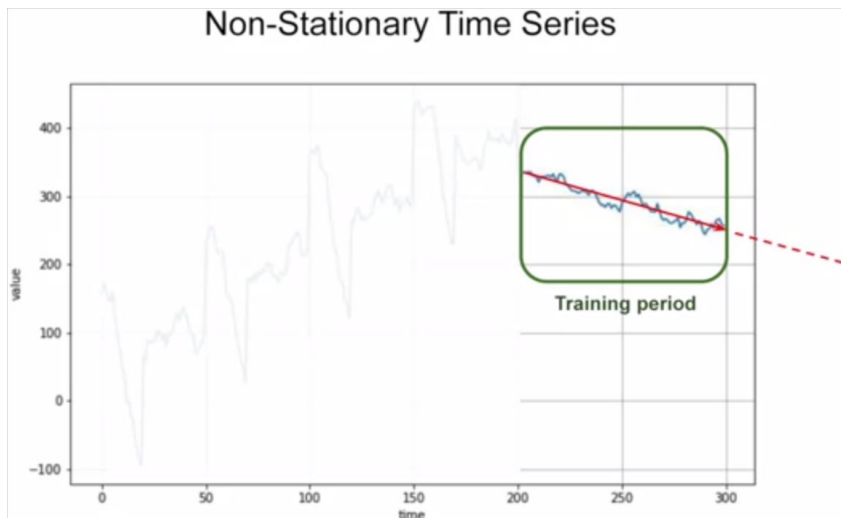
# Pattern umum dalam Time Series

- Non-Stationary Time Series

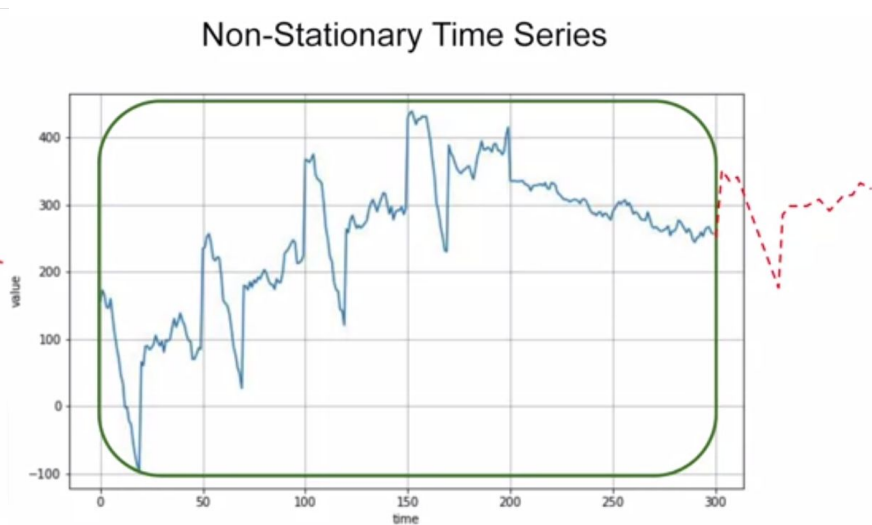


# Menghadapi Non-Stationary Time Series

Ini



Bukan ini



# Notebook Time Series

# Evaluasi Performance

- $\text{errors} = \text{forecasts} - \text{actual}$
- $\text{mse} = \text{np.square(errors).mean()}$
- $\text{rmse} = \text{np.sqrt(mse)}$
- $\text{mae} = \text{np.abs(errors).mean()}$
- $\text{mape} = \text{np.abs(errors/x\_valid).mean()}$

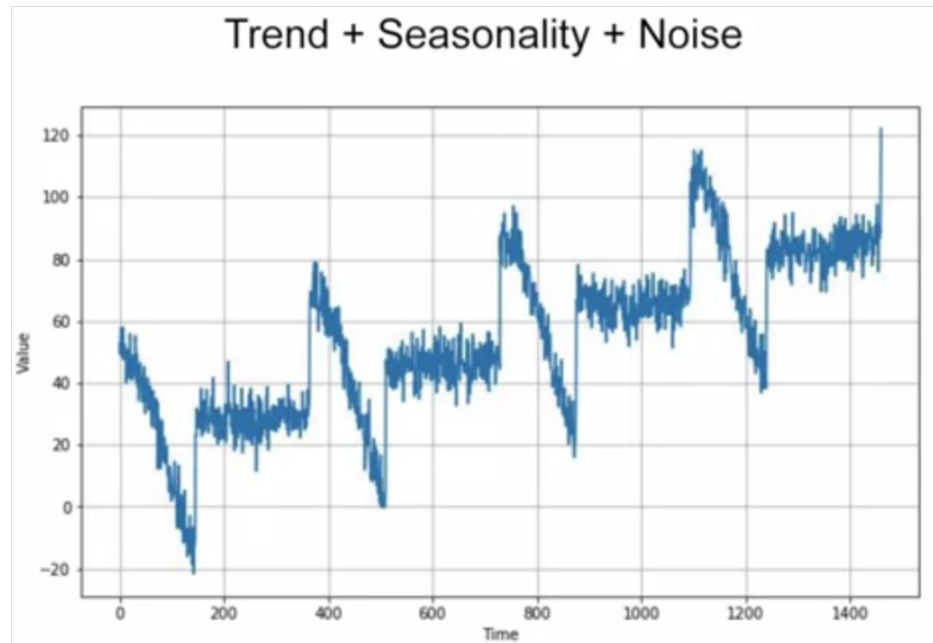
# Teknik

- Naive Forecasting
- Fixed Partitioning
- Roll-Forward Partitioning

# Teknik

- **Naive Forecasting**

Di mulai dari sampel data ini



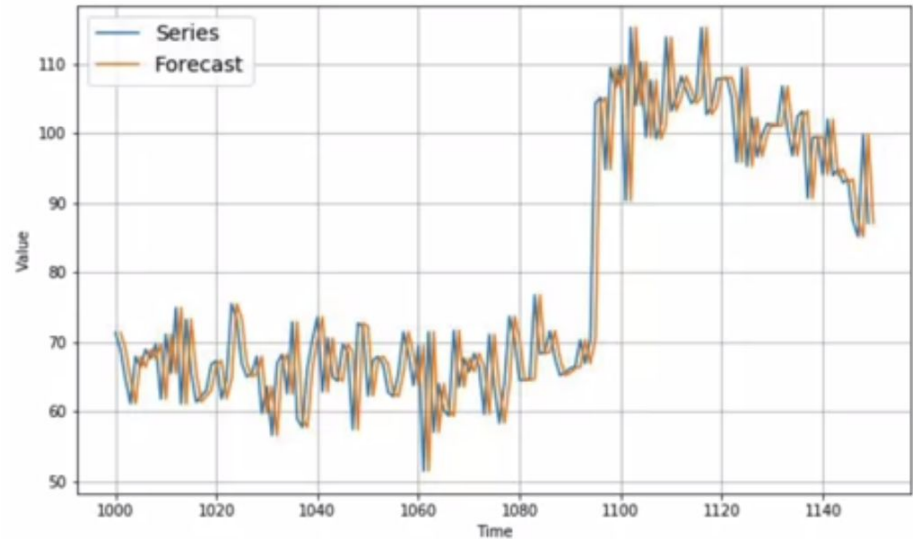
# Teknik

- **Naive Forecasting**

Kita set Forecast

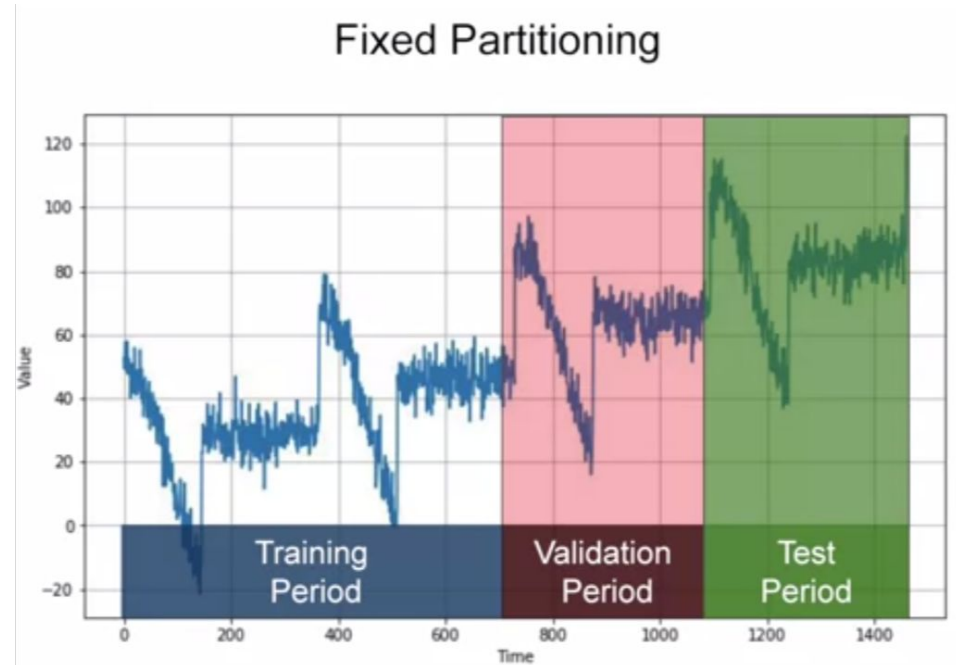
$$v(t + 1) = v(t)$$

Naive Forecasting



# Teknik

- **Fixed Partitioning**  
Langkah selanjutnya
- Retrain using train and val data, setelah hyperparameter tuning
- Kalau sudah bagus, retrain pakai test data

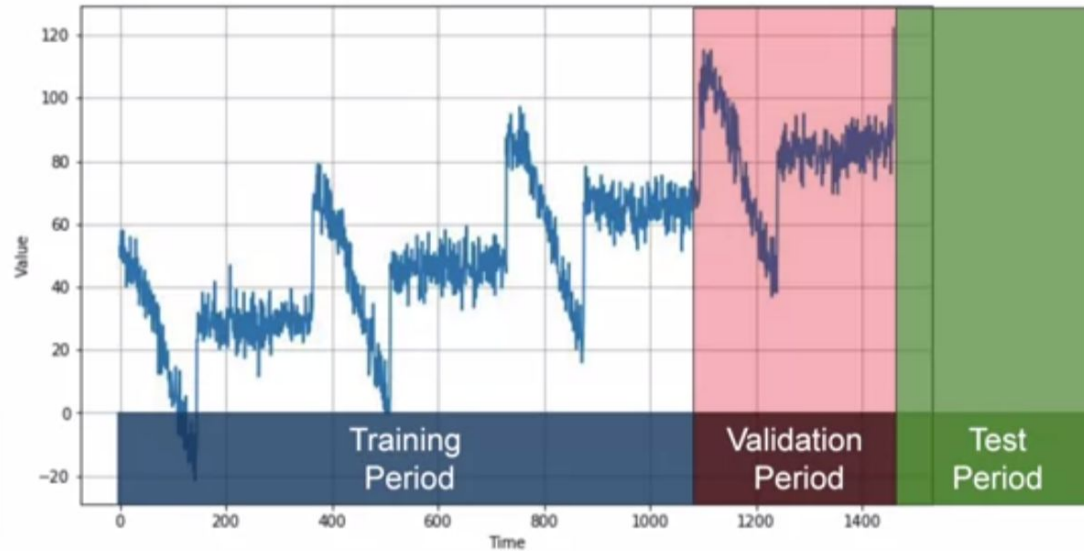




# Teknik

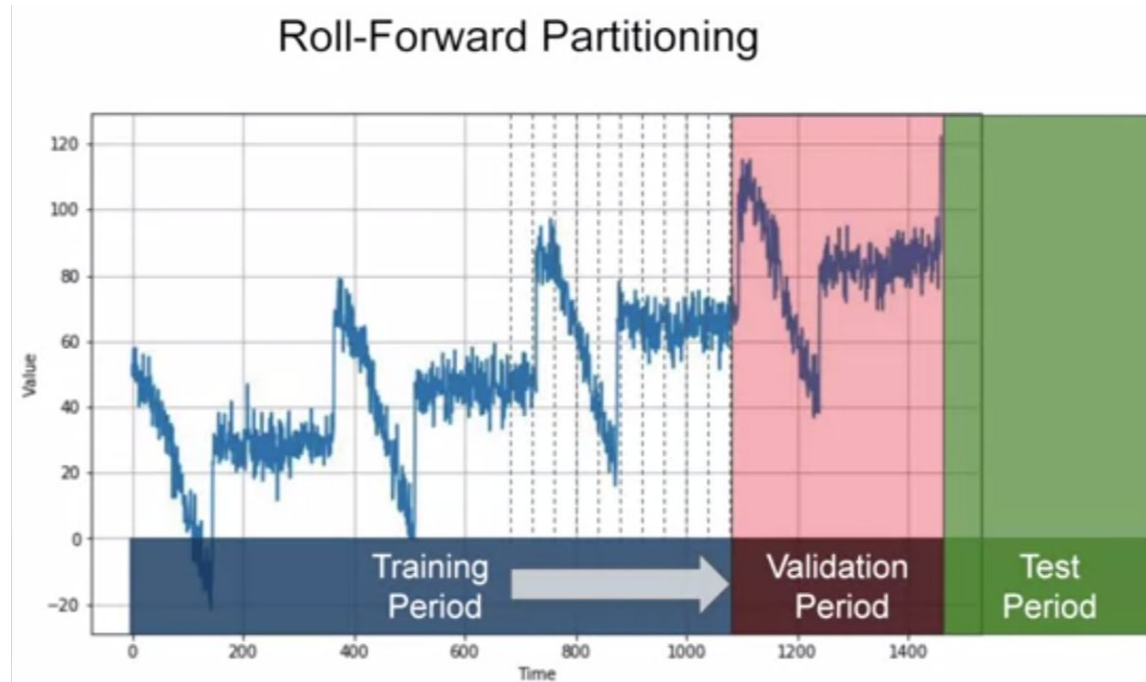
- Fixed Partitioning

Fixed Partitioning



# Teknik

- Roll-forward Partitioning

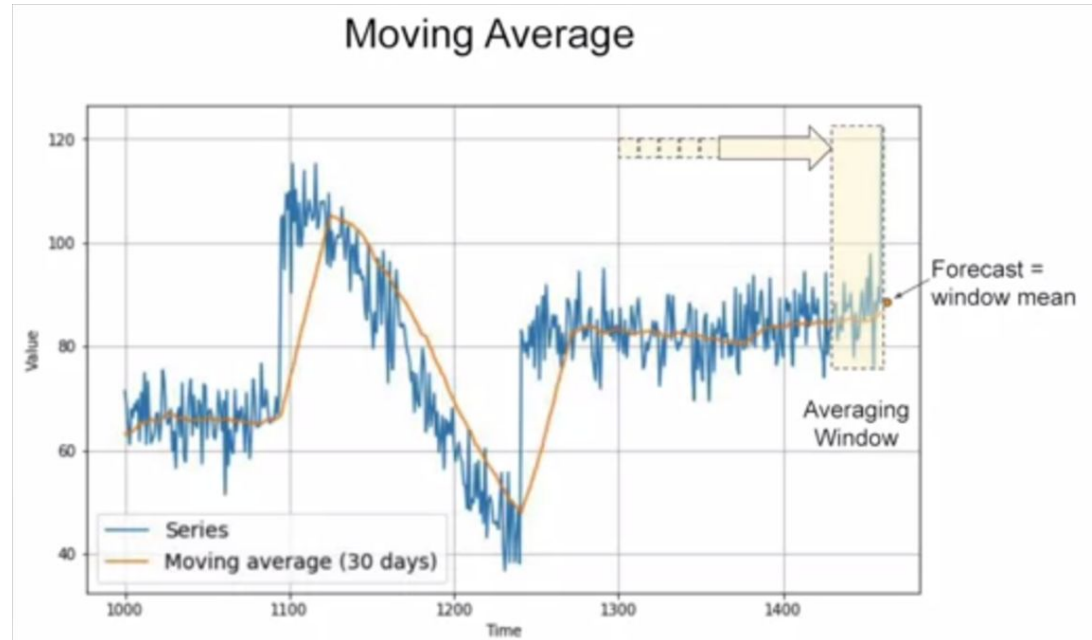


# Train, Validation, dan Test

- Training Period
- Validation Period
- Testing Period
- (Fixed Partitioning, one season)
- Retrain using train and val data, habis hyperparameter tuning
- Kalo dah oke, retrain pakai test data

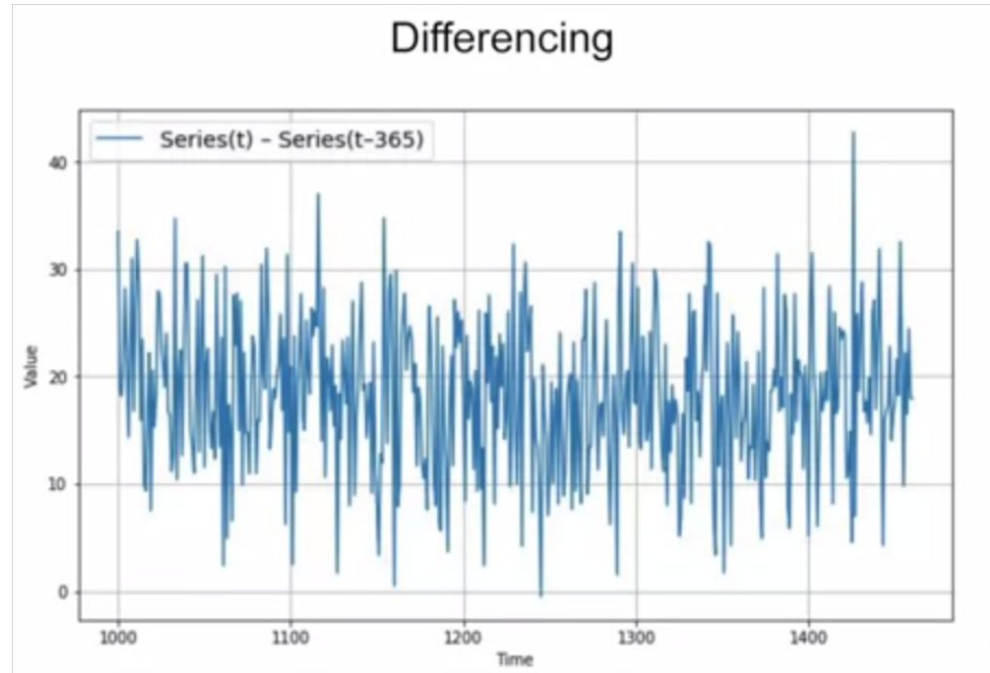
## Forecasting Method - Moving Average and Differencing

- **Moving Average :**  
Mengambil average di window untuk mengurangi noise
- **Window :**  
Range yang akan digunakan untuk mencari rata-rata



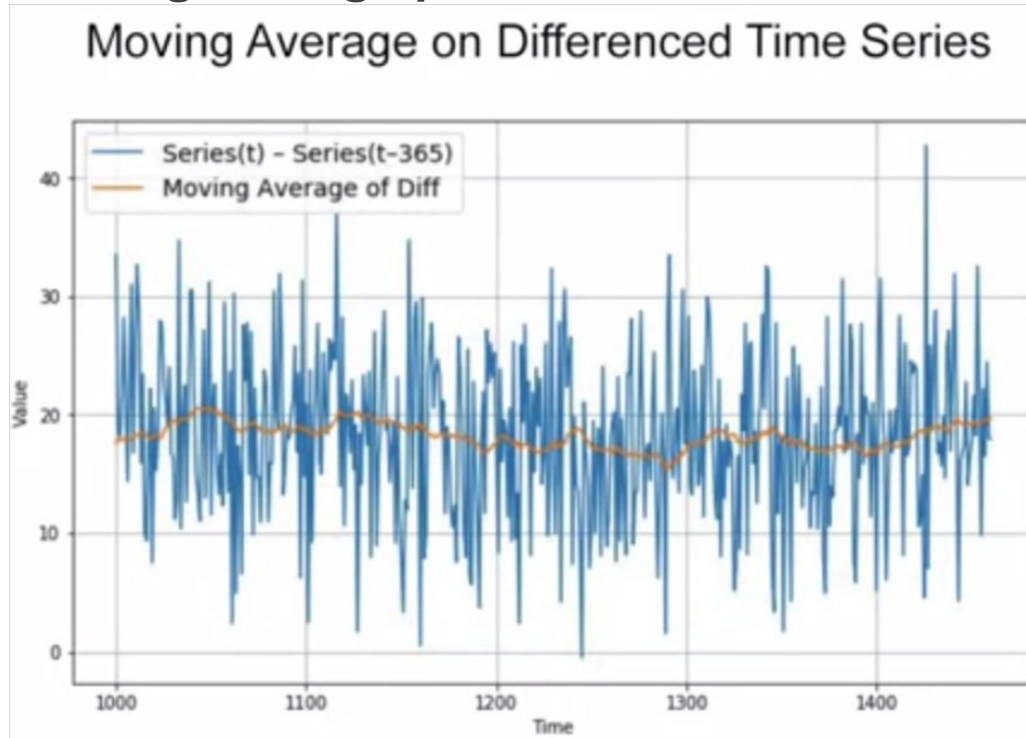
## Forecasting Method - Moving Average and Differencing

- **Differencing :**  
Menghilangkan trend dan seasonality dengan mengurangi season ini dengan season sebelumnya. Di contoh ini, series  $t$  dikurangi oleh  $\text{series}(t - 365)$ , series satu tahun sebelumnya, karena di kasus ini,  $\text{season} = \text{tahun}$



## Forecasting Method - Moving Average and Differencing

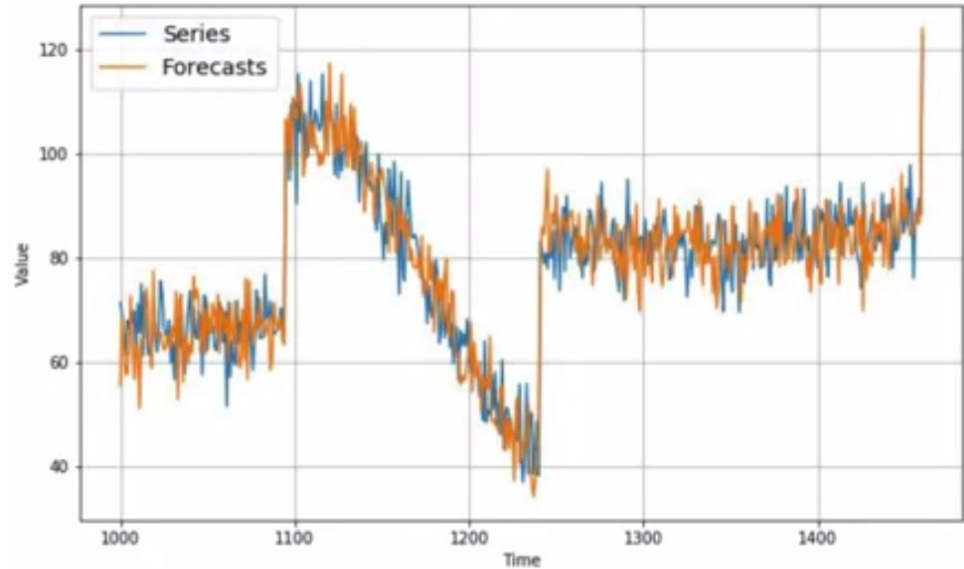
- Mencari Moving average pada Difference



## Forecasting Method - Moving Average and Differencing

- **Mengembalikan Trend dan Seasonality** dengan menambah kembali  $\text{series}(t - 365)$

Restoring the Trend and Seasonality

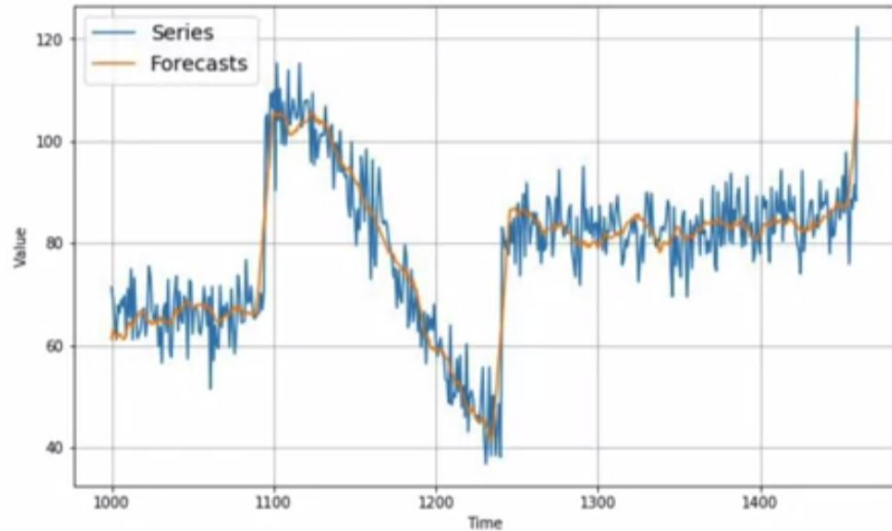


Forecasts = moving average of differenced series +  $\text{series}(t - 365)$

## Forecasting Method - Moving Average and Differencing

- **Smoothing:**  
Apply Moving Average pada hasil terakhir

Smoothing Both Past and Present Values



Forecasts = trailing moving average of differenced series + centered moving average of past series ( $t - 365$ )



# Trailing versus centered windows

- **Trailing Window**

Anggap kita ingin memprediksi series(present), maka kita akan menggunakan series( $t - 20$ ) sampai series( $t - 1$ ). Berguna untuk memprediksikan series masa kini.

- **Centered Window**

Anggap kita ingin memprediksi series(tahun lalu), maka kita akan memprediksinya dengan series(tahun lalu - 10) sampai series(tahun lalu + 10). Lebih akurat dalam memprediksikan series masa lalu.

## Time Series Analysis in Python - YouTube

# Q & A

# Thank You