Time Series Analysis (Weather)

Prediksi rata-rata kecepatan angin menggunakan data harian

Novi Handayani

Table of Contents

01

Scrapping Data

Web Extraction from API

Web Extraction from API

Data yang dibutuhkan dalam *task* ini adalah data rata-rata kecepatan angin harian dari tanggal 1 Januari 2020 hingga 1 Januari 2022 yang berasal dari website: https://www.wunderground.com/history/daily/us/ca/burbank/KBUR/date/2022-5-1.

Untuk mendapatkan data rata-rata kecepatan angin harian tersebut, dilakukan *web extraction* menggunakan *API request* untuk setiap tanggal mulai dari tanggal 1 Januari 2020 hingga 1 Januari 2022.

Data yang diperoleh berisi data kecepatan angin yang diukur untuk setiap jam dalam suatu hari. Berikut adalah gambaran data yang diperoleh dengan sudah dilakukan perhitungan rata-rata kecepatan angin untuk hari tersebut.

Untuk data lengkapnya dapat dilihat pada tautan berikut https://drive.google.com/file/d/1H_9ba9L-L2_1L83TVsIkx7H4UO2P5Wru/view?usp=sharing

	Date	Avg Wind Speed
0	2020-01-01	5.294118
1	2020-01-02	6.538462
2	2020-01-03	5.222222
3	2020-01-04	4.333333
4	2020-01-05	6.846154
5	2020-01-06	5.421053
6	2020-01-07	4.214286
7	2020-01-08	5.105263
8	2020-01-09	6.666667
9	2020-01-10	6.000000

02

EDA

Exploratory Data Analysis

Exploratory Data Analysis

Dataset terdiri dari 732 data dengan nilai rata-rata kecepatan angin berada dalam rentang nilai terkecil, yaitu 3.333333 dan nilai terbesar, yaitu 16.583333.

Terdapat *missing value* sejumlah satu data (dapat dilihat pada statistik *'count'* yang berjumlah 731 data rata-rata kecepatan angin).

	Avg Wind Speed
count	731.000000
mean	6.810134
std	1.730566
min	3.333333
25%	5.666667
50%	6.666667
75%	7.500000
max	16.583333

Statistik data

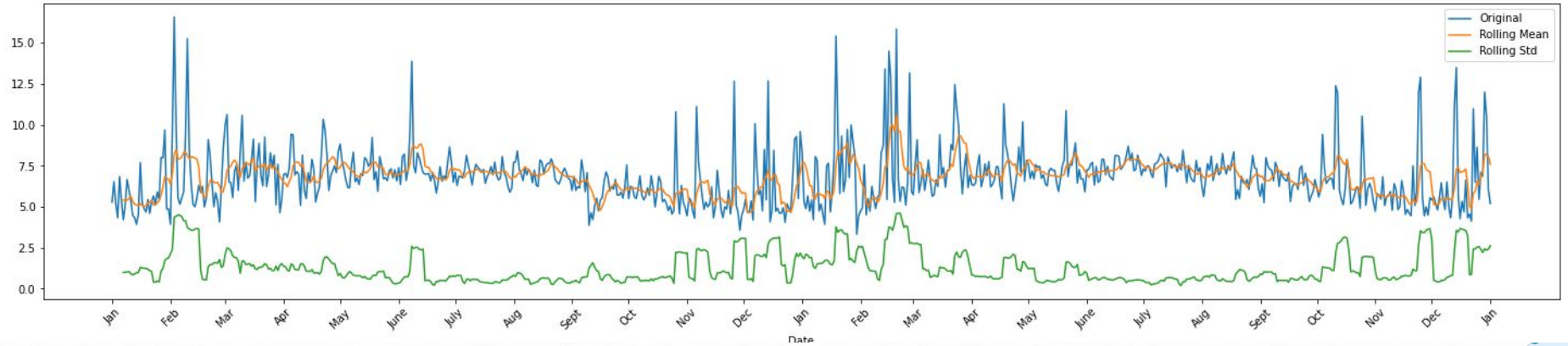
	Date	Avg Wind Speed
0	2020-01-01	5.294118
1	2020-01-02	6.538462
2	2020-01-03	5.222222
3	2020-01-04	4.333333
4	2020-01-05	6.846154
...	...	...
727	2021-12-28	6.866667
728	2021-12-29	12.000000
729	2021-12-30	10.511111
730	2021-12-31	6.043478
731	2022-01-01	5.200000

732 rows × 2 columns

Exploratory Data Analysis

Time series characteristic

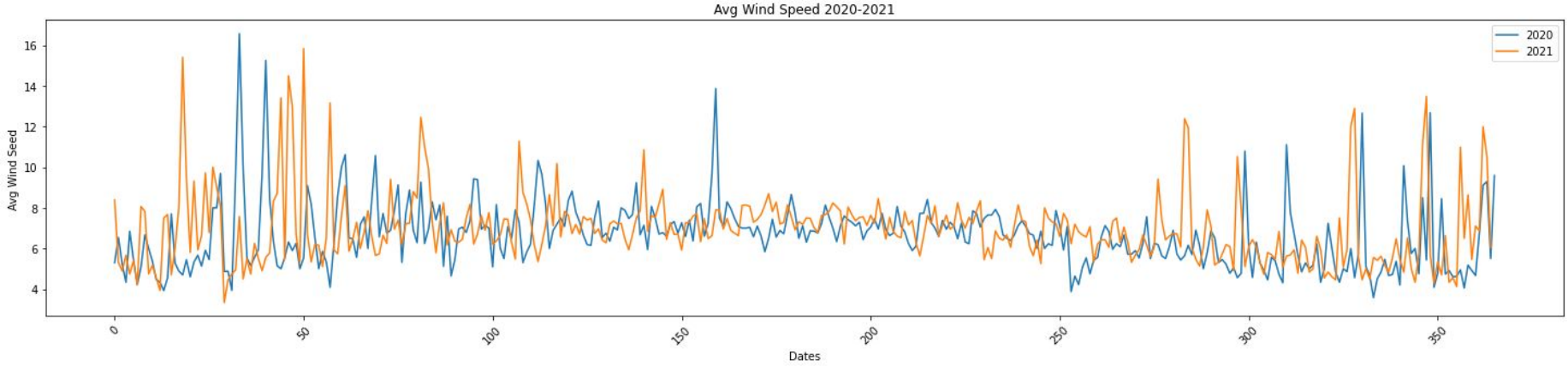
- Rolling Mean dan Rolling Std



Dari grafik di atas, secara sekilas terlihat bahwa data tersebut stasioner. Hal ini dikarenakan grafik tersebut tidak menunjukkan trend yang konsisten, melainkan cukup stabil, sehingga dapat dikatakan stasioner.

Exploratory Data Analysis

- Seasonality



Dari kedua grafik di atas terlihat bahwa terdapat seasonality, dimana rata-rata kecepatan angin pada bulan februari dan akhir november hingga desember cenderung lebih tinggi dibandingkan bulan-bulan lainnya. Sedangkan pertengahan tahun terlihat stabil.

Exploratory Data Analysis

- Stationary

```
Results of Dickey-Fuller Test:
Test Statistic                -7.369066e+00
p-value                        9.072579e-11
#Lags Used                     6.000000e+00
Number of Observations Used    7.250000e+02
Critical Value (1%)           -3.439402e+00
Critical Value (5%)           -2.865535e+00
Critical Value (10%)          -2.568897e+00
dtype: float64
```

Salah satu metode untuk mengetahui apakah time series data bersifat stasioner atau tidak, yaitu dengan melakukan *statistical test,* salah satunya adalah dengan **ADF test**.

Hasil dari ADF test tersebut menunjukkan bahwa nilai Test Statistic lebih kecil dibandingkan dengan seluruh nilai Critical Value, serta nilai p-value lebih kecil dari 0.05. Hal ini mengimplikasikan bahwa data bersifat **stationary**.

03

Preprocessing

Handle Missing Value

Dari hasil eksplorasi data, ditemukan bahwa terdapat satu *instance* data yang memiliki *missing value,* yaitu data pada tanggal 8 November 2020. Untuk menangani *missing value* pada data *time series*, salah satu cara yang dapat dilakukan adalah dengan menggunakan **teknik interpolasi**.

Teknik interpolasi merupakan teknik yang dapat memperkirakan suatu nilai dengan membuat asumsi pada hubungan dalam rentang titik data.

Normalization

Dalam beberapa referensi disebutkan bahwa algoritma cenderung menghasilkan performa yang lebih baik pada data yang memiliki skala atau distribusi yang konsisten. Oleh karena itu, tahap normalisasi merupakan salah satu tahap yang penting dilakukan dalam *preprocessing* jika distribusi data yang dimiliki cukup besar.

Dari hasil eksplorasi data, diketahui bahwa data rata-rata kecepatan angin memiliki nilai terkecil, yaitu 3.333333 dan nilai terbesar, yaitu 16.583333. Distribusi nilai tersebut bisa dikatakan cukup besar, sehingga dapat dilakukan normalisasi terlebih dahulu. Selain itu juga agar dapat dilakukan evaluasi perbandingannya.

Metode yang digunakan dalam normalisasi pada data rata-rata kecepatan angin adalah metode **MinMaxScaler.**

04

Modelling

ARIMA

01

Menentukan parameter d

Dengan melihat p-value dan sifat stationary-nya

```
from statsmodels.tsa.stattools import adfuller
result = adfuller(df['Interpolation'])
print(f'p-value: {result[1]}')
```

```
p-value: 9.072579454025724e-11
```

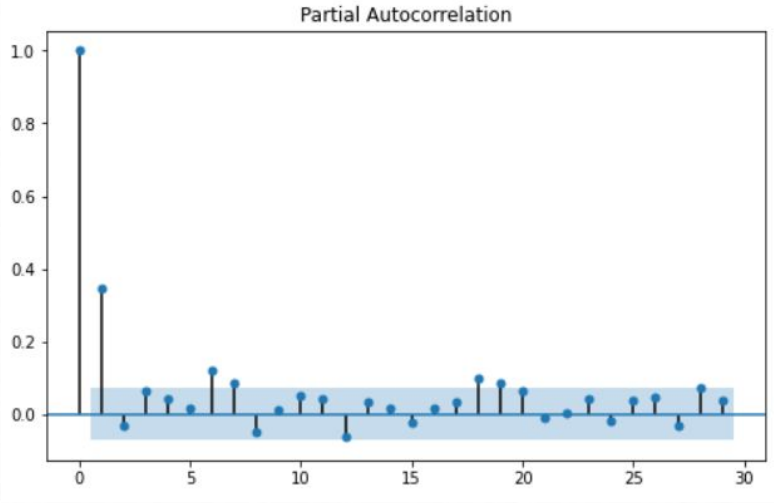
Seperti yang telah ditunjukkan saat tes stationary pada bagian sebelumnya, terlihat bahwa data tersebut stasioner. Oleh karena itu dipilih **value d = 0**

ARIMA

02 Menentukan parameter p

Dengan melihat grafik PACF (Partial Autocorrelation Function)

Karena pada PACF terlihat pada lag 1 mendapatkan hasil diluar significant limit dan merupakan yang paling significant, maka kami memilih **value p = 1**



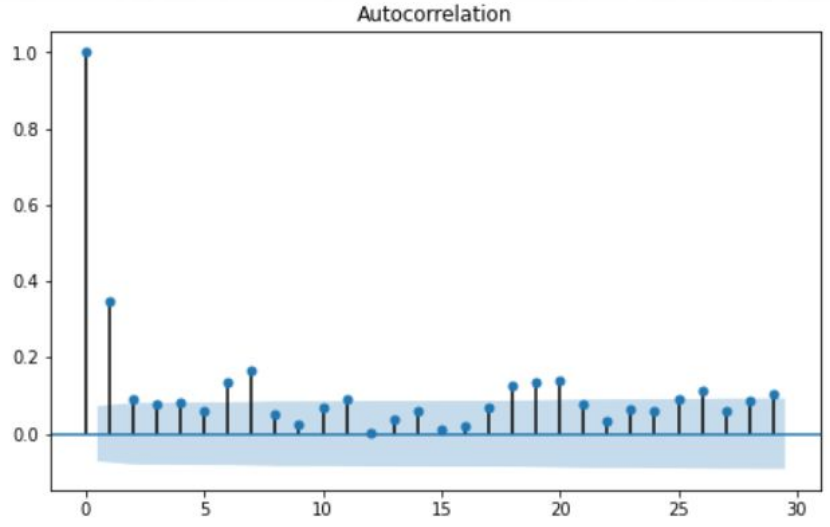
ARIMA

03

Menentukan parameter q

Dengan melihat grafik ACF (Autocorrelation Function)

Sedangkan pada ACF, lag 1 juga terlihat significant, namun karena lag 2 juga masih melebihi limit significat jadi kami memilih untuk mencoba dengan **value q = 2**



ARIMA

04

Membuat model ARIMA(1,0,2)

Summary model arima pada **data asli**

Terlihat bahwa didapatkan nilai **AIC = -992.777** serta memastikan bahwa nilai P>|z| tetap < 0.05 untuk menjaga stationarity

ARMA Model Results

Dep. Variable:	Interpolation	No. Observations:	512
Model:	ARMA(1, 2)	Log Likelihood	-992.777
Method:	css-mle	S.D. of innovations	1.681
Date:	Thu, 16 Jun 2022	AIC	1995.555
Time:	13:57:42	BIC	2016.747
Sample:	0	HQIC	2003.862

	coef	std err	z	P>|z|	[0.025	0.975]
const	6.7923	0.296	22.981	0.000	6.213	7.372
ar.L1.Interpolation	0.9850	0.012	82.989	0.000	0.962	1.008
ma.L1.Interpolation	-0.6794	0.044	-15.380	0.000	-0.766	-0.593
ma.L2.Interpolation	-0.2556	0.042	-6.046	0.000	-0.338	-0.173

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.0153	+0.0000j	1.0153	0.0000
MA.1	1.0540	+0.0000j	1.0540	0.0000
MA.2	-3.7117	+0.0000j	3.7117	0.5000

ARIMA

05 Membuat model ARIMA(1,0,2)

Summary model arima pada **data normalized**

Terlihat bahwa didapatkan nilai **AIC = -650.459**, yang mana lebih besar daripada pada model sebelumnya, serta juga memastikan bahwa nilai P>|z| tetap < 0.05 untuk menjaga stationarity

ARMA Model Results

Dep. Variable:	Interpolation	No. Observations:	512
Model:	ARMA(1, 2)	Log Likelihood	330.229
Method:	css-mle	S.D. of innovations	0.127
Date:	Thu, 16 Jun 2022	AIC	-650.459
Time:	13:57:43	BIC	-629.267
Sample:	0	HQIC	-642.151

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.2611	0.022	11.703	0.000	0.217	0.305
ar.L1.Interpolation	0.9850	0.012	82.988	0.000	0.962	1.008
ma.L1.Interpolation	-0.6794	0.044	-15.380	0.000	-0.766	-0.593
ma.L2.Interpolation	-0.2556	0.042	-6.046	0.000	-0.338	-0.173

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.0153	+0.0000j	1.0153	0.0000
MA.1	1.0540	+0.0000j	1.0540	0.0000
MA.2	-3.7117	+0.0000j	3.7117	0.5000

SARIMA

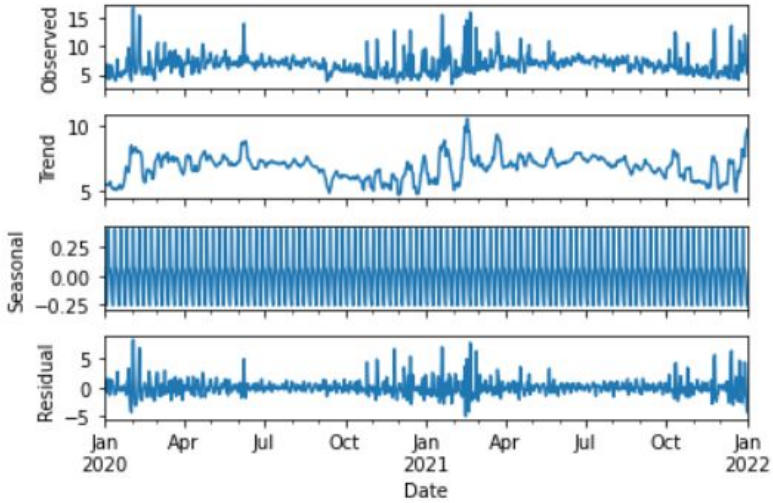
01 **Menentukan parameter (p, d, q)**
Kami memilih nilai p,d,q yang sama dengan yang digunakan pada model ARIMA

02 **Menentukan parameter P, D, Q, dan m**
Menggunakan grafik seasonality decompose serta melakukan beberapa kali percobaan

SARIMA

02 (cont) Menentukan parameter P, D, Q, dan m

Seasonality decompose



Dari grafik tersebut, terlihat bahwa terdapat seasonality pada rentang waktu yang sangat sempit. Mengingat bahwa data yang digunakan adalah data harian, maka kami mengasumsikan seasonality terjadi secara mingguan sehingga dipilih nilai **m = 7**

SARIMA

02 Menentukan parameter P, D, Q, dan m

(cont)

```
[282] from statsmodels.tsa.stattools import adfuller
      import numpy as np
      def check_stationarity(ts):
          dftest = adfuller(ts)
          adf = dftest[0]
          pvalue = dftest[1]
          critical_value = dftest[4]['5%']
          if (pvalue < 0.05) and (adf < critical_value):
              print('The series is stationary')
          else:
              print('The series is NOT stationary')
```

```
[283] seasonal = result.seasonal
      check_stationarity(seasonal)

      The series is stationary
```

Kami menggunakan hasil yang didapat dari pengamatan seasonality pada langkah sebelumnya. Lalu, ketika dicek p-value dengan ADF test, terlihat pula bahwa data bersifat stasioner sehingga digunakan nilai **D = 0**

Sedangkan untuk nilai P dan M, kami melakukan beberapa kali percobaan dan memilih model yang lebih baik, yaitu dengan value **P dan Q = 1**

SARIMA

03

Membuat model SARIMA(1,0,2)x(1,0,1,7)

Summary model arima pada **data asli**

Terlihat bahwa didapatkan nilai **AIC = 2384.812** serta memastikan bahwa nilai P>|z| tetap < 0.05 untuk menjaga stationarity

Statespace Model Results

Dep. Variable:	Interpolation	No. Observations:	640
Model:	SARIMAX(1, 0, 2)x(1, 0, 1, 7)	Log Likelihood	-1186.406
Date:	Thu, 16 Jun 2022	AIC	2384.812
Time:	13:58:01	BIC	2411.581
Sample:	01-01-2020 - 10-01-2021	HQIC	2395.202
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.9998	0.001	870.522	0.000	0.998	1.002
ma.L1	-0.6890	0.023	-29.541	0.000	-0.735	-0.643
ma.L2	-0.2553	0.024	-10.503	0.000	-0.303	-0.208
ar.S.L7	0.9926	0.121	8.228	0.000	0.756	1.229
ma.S.L7	-0.9901	0.132	-7.514	0.000	-1.248	-0.732
sigma2	2.3683	0.068	34.875	0.000	2.235	2.501

Ljung-Box (Q):	76.23	Jarque-Bera (JB):	3376.65
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	0.42	Skew:	2.41
Prob(H) (two-sided):	0.00	Kurtosis:	13.16

SARIMA

04 Membuat model SARIMA(1,0,2)x(1,0,1,7)

Summary model arima pada **data normalized**

Terlihat bahwa didapatkan nilai **AIC = -1000.881**, jauh lebih rendah daripada AIC pada model sebelumnya. Selain itu, disini kami juga memastikan bahwa nilai P>|z| tetap < 0.05 untuk menjaga stationarity

Statespace Model Results

Dep. Variable:	Interpolation	No. Observations:	732
Model:	SARIMAX(1, 0, 2)x(1, 0, 1, 7)	Log Likelihood	506.441
Date:	Thu, 16 Jun 2022	AIC	-1000.881
Time:	13:58:10	BIC	-973.307
Sample:	01-01-2020 - 01-01-2022	HQIC	-990.244
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.9990	0.002	495.877	0.000	0.995	1.003
ma.L1	-0.6578	0.023	-28.970	0.000	-0.702	-0.613
ma.L2	-0.2920	0.022	-13.274	0.000	-0.335	-0.249
ar.S.L7	0.9994	0.026	38.801	0.000	0.949	1.050
ma.S.L7	-0.9972	0.068	-14.667	0.000	-1.130	-0.864
sigma2	0.0145	0.001	20.271	0.000	0.013	0.016

Ljung-Box (Q):	56.08	Jarque-Bera (JB):	2791.05
Prob(Q):	0.05	Prob(JB):	0.00
Heteroskedasticity (H):	0.93	Skew:	2.26
Prob(H) (two-sided):	0.57	Kurtosis:	11.43

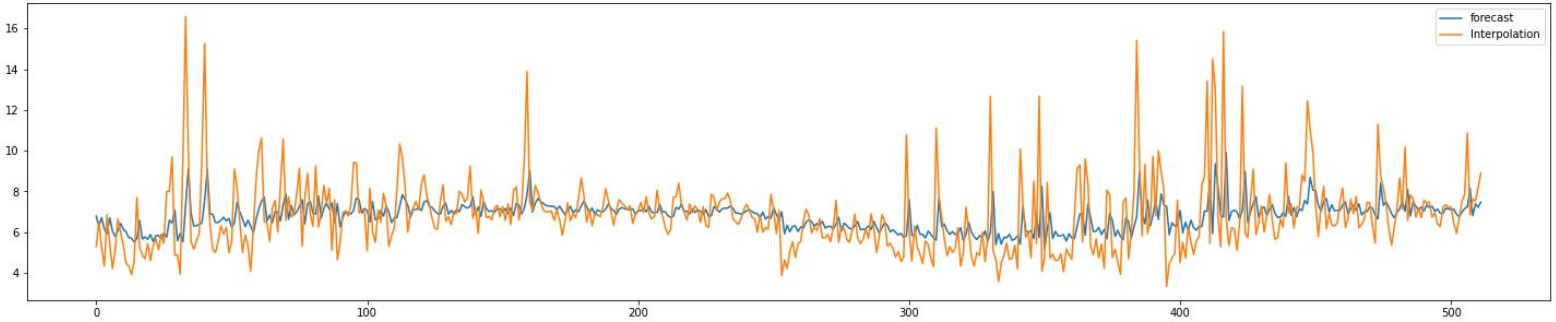
05

Evaluation

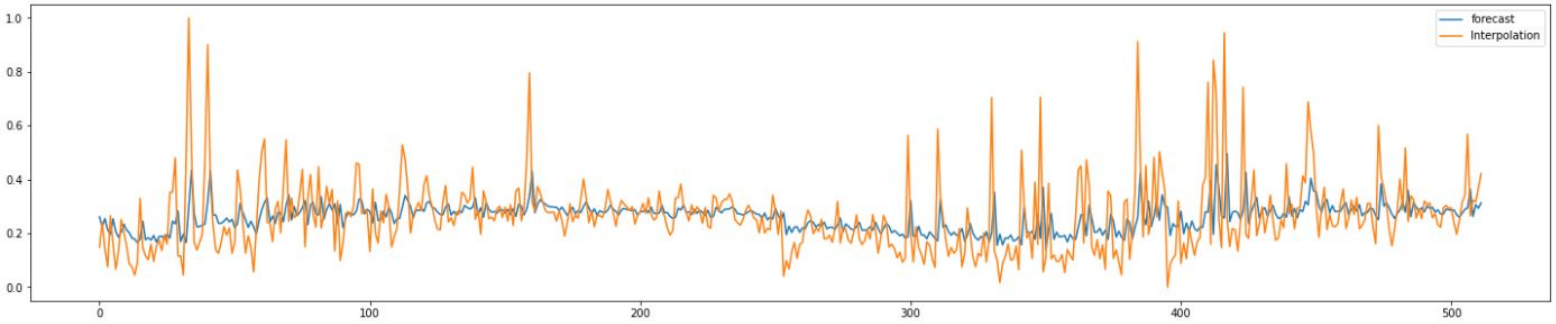
Hasil Prediksi Model ARIMA

Pada data training

Pada data asli



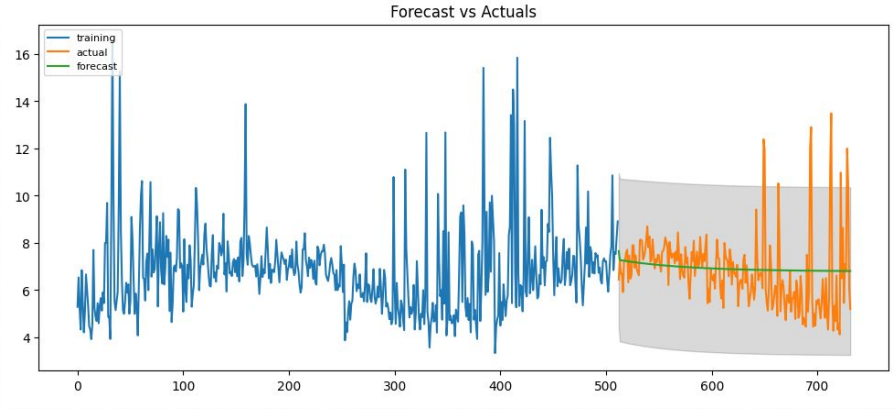
Pada data normalized



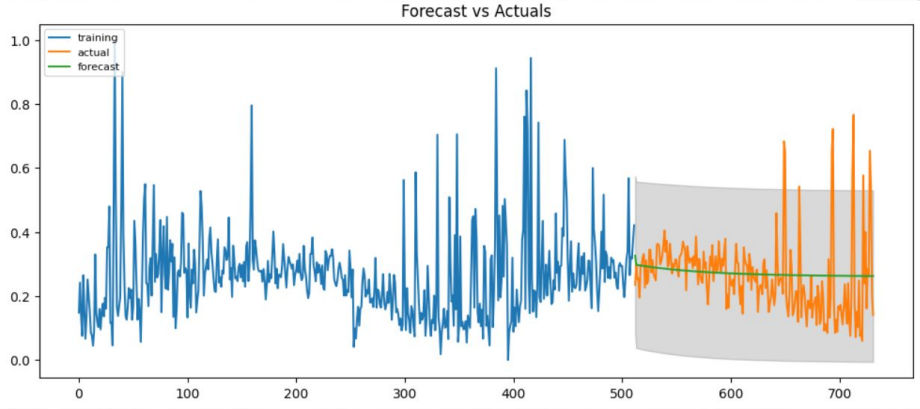
Hasil Prediksi Model ARIMA

Hasil forecasting pada data testing

Pada data asli



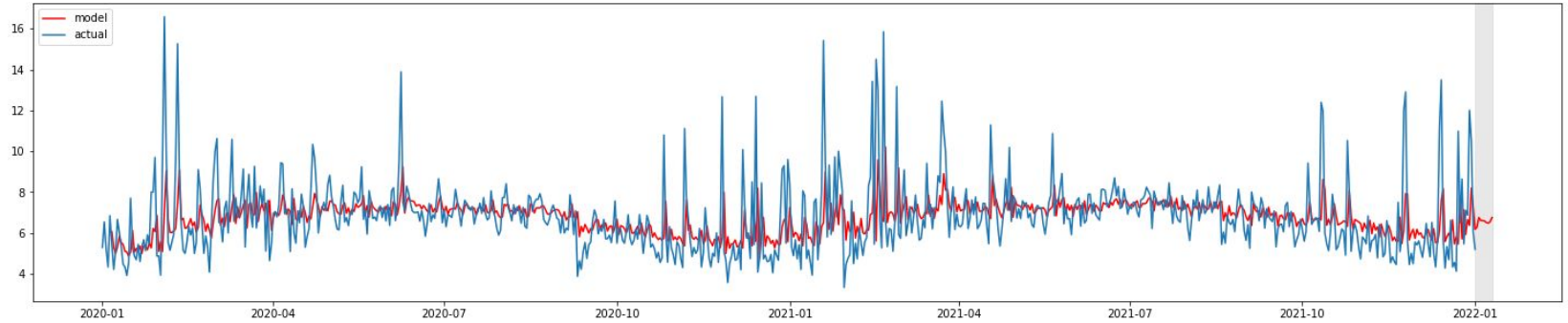
Pada data normalized



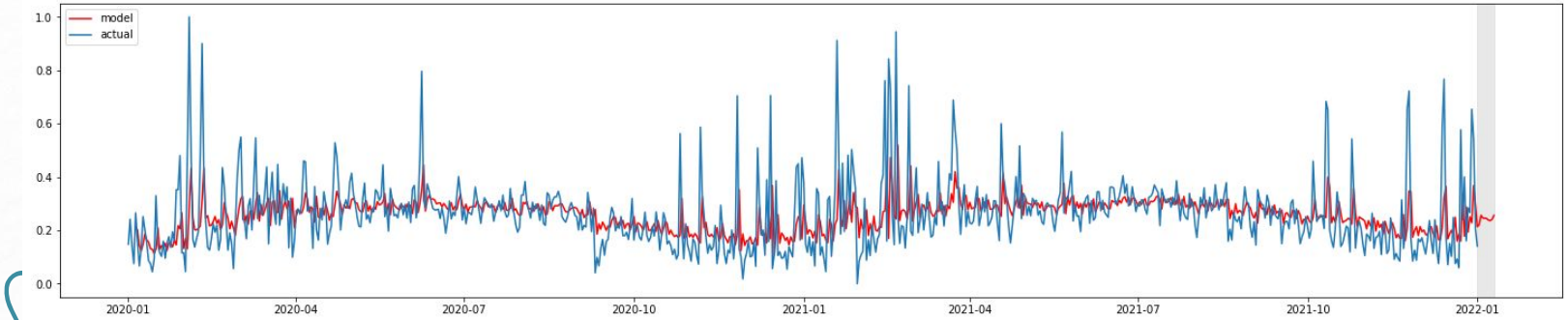
Hasil Prediksi Model SARIMA

Pada data training

Pada data asli



Pada data normalized



Evaluasi RMSE

Kami menggunakan RMSE sebagai evaluasi untuk kedua model.

Didapatkan hasil sebagai berikut

- ARIMA (data asli): 1.5057733312094088
- ARIMA (data normalized): 0.11364327123830775
- SARIMA (data asli): 1.554430204060829
- SARIMA (data normalized): 0.12111227920593934

Dari hasil tersebut, terlihat bahwa model **ARIMA memberikan hasil yang lebih baik** daripada SARIMA dikarenakan nilai RMSE-nya lebih rendah, baik pada data ternormalisasi ataupun tidak.

Kemudian, dari percobaan yang kami lakukan juga dapat terlihat bahwa **normalisasi** pada data kecepatan angin ini juga berpengaruh terhadap prediksi dan menghasilkan **prediksi yang lebih baik** daripada yang dilakukan pada data asli, baik pada model ARIMA maupun SARIMA.