Jurnal Strategi
Volume 4 Nomor 1 Mei 2022

Prediksi dan Analisis *Time Series* pada Data COVID-19

Kristanto Tanuwidjaja[#1], Andreas Widjaja[*2]

[#]*Program Studi Sistem Informasi, Universitas Kristen Maranatha*
Jl. Surya Sumantri No.65 Bandung 40164, Indonesia.
[1]1873001@maranatha.ac.id
[2]andreas.widjaja@it.maranatha.edu

Abstract — **Data processing on a large scale is currently needed for various needs in the field of information technology to create conclusions that are easy to understand and analyze based on existing data. With methods in data science, large-scale data processing makes it easier to present data to be understood and analyzed. This research was conducted by processing COVID-19 data based on the Time Series. The steps in this research are to apply Exploratory Data Analysis first, then visualize the data, and make predictions using the ARIMA and Prophet methods. The research was conducted with the aim of processing COVID-19 data into informative and easy-to-analyze data, visualizing the COVID-19 dataset to make it easier to understand, and making predictions for the future. The dataset used was obtained from Kaggle entitled "COVID-19 data from John Hopkins University" which contains confirmed data and death data from around the world. From the dataset, several countries in Southeast Asia were selected for exploration, including Indonesia, the closest country to Indonesia, and Southeast Asian countries. From the exploration results obtained various information from data in the form of a DataFrame which is easy to analyze, a variety of graphic plots that are easy to understand, and get the prediction results of confirmed cases in Indonesia from the ARIMA and Prophet methods which are then determined that the optimal prediction is using the Prophet.**

Keywords— **ARIMA, COVID-19, Data, Forecast, Prophet.**

I. PENDAHULUAN

Perkembangan teknologi saat ini dalam pengelolaan data dengan jumlah besar diperlukan untuk berbagai kebutuhan dalam bidang teknologi informasi dengan tujuan menciptakan suatu kesimpulan yang mudah untuk dipahami dan dianalisis berdasarkan data yang sudah dipersiapkan sebelumnya. Metode yang terdapat dalam *data science* untuk mengelola data skala besar dapat mempermudah penyajian data sehingga mudah dipahami dan dianalisis.

Data science merupakan penggabungan antara inferensi data, pengembangan algoritmik, dengan teknologi untuk memecahkan masalah analitik yang kompleks dari data mentah yang dapat dikelola [1]. *Data science* merupakan pendekatan yang berwawasan ke depan, menggunakan cara eksplorasi yang berfokus pada analisis data masa lalu atau saat ini bertujuan untuk menentukan keputusan yang tepat berdasarkan data yang ada yang disebut dengan dataset [2]. *Dataset* merupakan sekumpulan data yang menjadi sumber informasi dari eksplorasi dan analisis menggunakan metode yang terdapat pada *data science*.

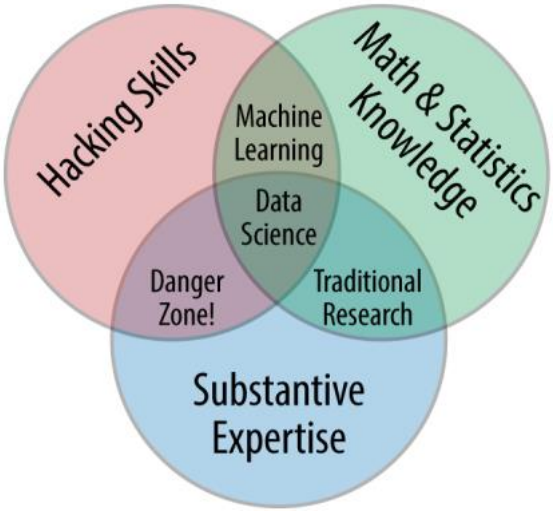
Dataset yang digunakan yaitu "COVID-19 data from John Hopkins University" yang diperoleh dari Kaggle [3] dengan mempertimbangkan situasi pandemi. Dalam dataset tersebut, terdapat berbagai macam informasi dalam bentuk data dengan format file CSV (*Comma Separated Values*) yang berisi data COVID-19 dari berbagai negara dengan angka kasus terkonfirmasi dan kematian perhari.

Eksplorasi dilakukan untuk mengetahui jumlah kasus terkonfirmasi dan kematian per-hari dari berbagai negara yang kemudian dilakukan visualisasi dalam berbagai bentuk grafik agar mempermudah untuk dipahami serta melakukan prediksi untuk mengetahui jumlah kasus untuk masa yang akan datang.

II. KAJIAN TEORI

A. Data Science

Data Science merupakan ilmu yang terdiri dari gabungan antara bidang statistik, metode ilmiah, dan analisis data yang saling tumpang tindih. Drew Conway's memberi ilustrasi mengenai *data science* dalam bentuk Diagram Venn yang menggambarkan bahwa *data science* terdiri dari kemampuan *hacking*, pengetahuan matematika dan statistika, dan keahlian substantif [4].



Gambar 1. Diagram Venn Drew Conway's mengenai data science

B. Exploratory Data Analysis

Exploratory Data Analysis (EDA) adalah tahap awal dalam membangun sebuah model yang merupakan bagian penting dalam proses data science dan mewakili cara melakukan statistik yang dilakukan dalam menganalisis data konfirmatori yang berkaitan dengan pemodelan dan hipotesis [5]. Dalam melakukan EDA, terdapat beberapa tahapan terkait analisis data, antara lain [6]:

1. Persyaratan data.
2. Pengumpulan data.
3. Pengolahan data.
4. Pembersihan data.
5. Analisis data eksplorasi.
6. Pemodelan dan algoritma.
7. Produk data.
8. Komunikasi.

C. Time Series

Time series atau deret waktu adalah sekumpulan data yang memiliki urutan pengamatan dengan berorientasi waktu atau kronologis pada variabel yang diminati [7]. Deret waktu berkaitan dengan pengamatan atau pengukuran data pada banyak titik waktu yang membentuk pola berdasarkan frekuensi tetap yang muncul secara berkala [8]. Apabila diamati lebih dalam, deret waktu memiliki karakteristik yang unik, antara lain [6]:

1. *Trend.*
2. *Outlier.*
3. *Seasonality.*
4. *Abrupt change.*
5. *Constant variance.*

D. Forecasting

Forecast adalah prediksi dari beberapa peristiwa yang kemungkinan dapat terjadi di masa depan. Prediksi merupakan masalah penting yang mencakup berbagai bidang termasuk bisnis dan industri, pemerintahan, ekonomi, ilmu lingkungan, kedokteran, ilmu sosial, politik, dan keuangan. Proses prediksi diklasifikasikan menjadi prediksi jangka pendek, prediksi jangka menengah, dan prediksi jangka panjang. Tahap yang perlu dilakukan dalam proses prediksi yaitu [7]:

1. *Problem definition*.
2. *Data collection*.
3. *Data analysis*.
4. *Model selection and fitting.*
5. *Model validation*.
6. *Forecasting model deployment*.
7. *Monitoring forecasting model performance*.

E. Autoregressive Integrated Moving Average Models

Autoregressive Integrated Moving Average (ARIMA) *models* atau *Box-Jenkins models* merupakan gabungan dari *autoregressive models* dengan *moving average models* [9]. ARIMA merupakan kelas model yang menjelaskan deret waktu tertentu berdasarkan nilai dari masa lalu yang dimiliki, yaitu *lags* dirinya sendiri dan *lags* dari prediksi yang *error* [10]. Model ARIMA memiliki 3 *hyperparameter* yaitu: *p, d*, dan *q*. *p* yaitu istilah yang merujuk pada "*Autoregressive*" (AR), *q* yaitu istilah yang merujuk pada "*Moving Average*" (MA), dan *d* yaitu istilah yang merujuk pada "*Differencing*" [11].

F. Prophet

Prophet merupakan metode yang digunakan untuk memprediksi deret waktu berdasarkan model aditif dimana tren-non linier sesuai dengan musim tahunan, mingguan, dan harian, ditambah efek liburan yang secara singkat merupakan menjumlahan dari beberapa komponen yaitu *trend, seasonality,* dan *holidays*. Prophet menghasilkan prediksi dengan performa baik sehingga mendapatkan prediksi hanya dalam beberapa setelah melakukan *fitting models.* Prophet juga mampu mengatasi *missing value* dan *outliers,* serta perubahan drastis pada deret waktu dengan hanya menyesuaikan terhadap perubahan tren untuk mengatasi *outliers* dan dengan menetapkan nilai *null* ke dalam nilai yang kosong serta meninggalkan variabel *date* di masa yang akan datang untuk mengatasi *missing value* [12].

III. METODOLOGI PENELITIAN

A. Dataset COVID-19 data from John Hopkins University

Dataset yang digunakan diperoleh dari Kaggle yang diunggah oleh pengguna dengan nama "Anthony Goldbloom" yang merupakan CEO dari Kaggle. *Dataset* tersebut diberi judul "COVID-19 data from John Hopkins University" oleh pengguna yang merupakan repositori data COVID-19 Center of Systems Science and Engineering (CSSE) dari John Hopkins University yang pertama kali diunggah pada tanggal 2 November 2020 dan diperbarui setiap hari pada pukul 06.00 UTC dengan format *file* CSV (*Comma Separated Values*).

B. Exploratory Data Analysis

Dalam melakukan EDA, terdapat beberapa tahap yang harus dilakukan agar hasil dari analisis yang dilakukan dapat berjalan dengan baik dan benar seperti melakukan *import library*, memuat *dataset* yang terdiri dari kasus terkonfirmasi dan kasus kematian, mengidentifikasi karakteristik dari *dataset*, dan melakukan eksplorasi dan analisis dari *dataset*. Eksplorasi dan analisis mencakup eksplorasi kasus terkonfirmasi di 3 negara terdekat dengan Indonesia, eksplorasi kasus terkonfirmasi di negara Indonesia, eksplorasi kasus terkonfirmasi di negara Asia Tenggara, eksplorasi kasus kematian di Asia Tenggara, dan eksplorasi kasus kematian di Indonesia.

C. Visualisasi Data

Setelah melakukan eksplorasi terhadap *dataset* COVID-19, tahap yang dapat dilakukan selanjutnya yaitu membuat visualisasi data. Visualisasi data dilakukan agar hasil dari *DataFrame* yang sudah dibuat, dapat divisualisasikan dalam bentuk grafis yang mudah dipahami. Visualisasi data mencakup visualisasi kasus terkonfirmasi dan kematian di 3 negara dekat Indonesia, negara Indonesia, dan negara Asia Tenggara dengan menggunakan *line plot* dan *bar plot*.

D. Prediksi Data

Dengan berdasarkan data yang ada sebelumnya, prediksi dapat dilakukan untuk memprediksi peningkatan kasus terkonfirmasi dalam waktu 7 hari kedepan dengan menggunakan metode ARIMA dan Prophet.

1) *Auto-ARIMA*: Metode Auto-ARIMA menentukan parameter *p, d, q* secara otomatis yang ditentukan oleh *library* pmdarima dengan mencari *aic*, *aicc*, *bic*, *hqic*, dan *oob* yang optimal.
2) *ARIMA:* Metode ARIMA yang dipakai yaitu menggunakan *library* Statsmodels.
3) *Prophet:* Metode Prophet dilakukan dengan menggunakan *library* Prophet.

E. Komparasi Performa

Untuk mengukur performa yang dihasilkan dari ketiga pustaka yang digunakan untuk melakukan prediksi, maka dilakukan pengujian performa dengan menggunakan metode RMSE (*Root Mean Square Error*) dan MAE (*Mean Absolute Error*). RMSE merupakan aturan penilaian kuadrat yang juga mengukur besarnya rata-rata kesalahan sedangkan MAE mengukur besarnya rata-rata kesalahan dalam serangkaian prediksi, tanpa ada pertimbangan arahnya. Semakin hasil dari RMSE dan MAE mendekati nol, maka semakin baik modelnya [13].

IV. HASIL DAN DISKUSI

A. Hasil Eksplorasi Kasus Terkonfirmasi di 3 Negara

Lakukan pemilihan negara yang terdekat dengan Indonesia, yaitu negara Malaysia dan Singapura untuk melihat kasus terkonfirmasi. Kemudian lakukan konversi tipe data dari kolom *Date* menjadi tipe data *datetime*. Gambar 2 merupakan hasil dari eksplorasi kasus terkonfirmasi di 3 negara dimana terlihat bahwa 3 negara tersebut memiliki angka kasus terkonfirmasi yang beragam setiap harinya dengan kasus awal pada tanggal 23 Januari 2020 dan akhir dari data pada tanggal 29 November 2020.

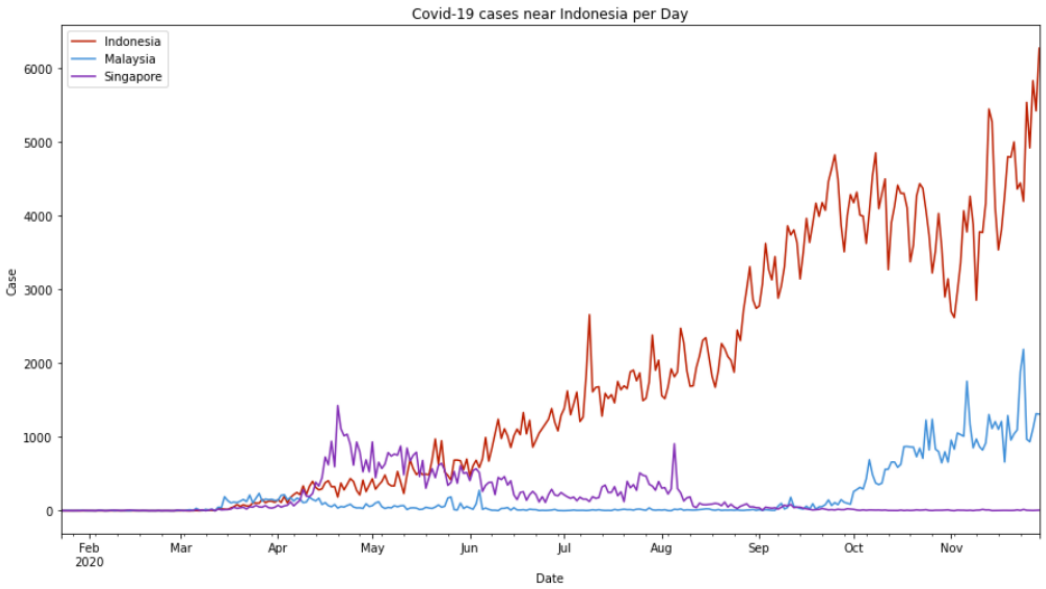
```
asia_confirm=confirm_cases[['Date', 'Indonesia', 'Malaysia', 'Singapore']]
asia_confirm['Date']=pd.to_datetime(asia_confirm.Date)
asia_confirm
```

	Date	Indonesia	Malaysia	Singapore
1	2020-01-23	0.0	0.0	1.0
2	2020-01-24	0.0	0.0	2.0
3	2020-01-25	0.0	3.0	0.0
4	2020-01-26	0.0	1.0	1.0
5	2020-01-27	0.0	0.0	1.0
...	...	...	...	...
308	2020-11-25	5534.0	970.0	7.0
309	2020-11-26	4917.0	935.0	5.0
310	2020-11-27	5828.0	1109.0	4.0
311	2020-11-28	5418.0	1315.0	6.0
312	2020-11-29	6267.0	1309.0	8.0

312 rows × 4 columns

Gambar 2. Kasus Terkonfirmasi di 3 Negara

```
asia_confirm.plot(kind='line', x='Date', y=['Indonesia', 'Malaysia', 'Singapore'], figsize=(15,8))
plt.title('Covid-19 cases near Indonesia per Day')
plt.ylabel('Case')
plt.show()
```



Gambar 3 *Line Plot* Kasus Terkonfirmasi di 3 Negara

B. Hasil Eksplorasi Kasus Terkonfirmasi Harian Tertinggi di 3 Negara

Setelah didapatkan data angka kasus harian di 3 negara, maka dicari angka kasus harian tertinggi di 3 negara. Gambar **4** merupakan tampilan dari angka kasus harian tertinggi di 3 negara dimana negara Indonesia mendapatkan angka kasus harian tertinggi di angka 6267 pertambahan kasus dalam waktu 1 hari, disusul negara Malaysia di angka 2188 pertambahan kasus, dan negara Singapura di angka 1426 pertambahan kasus.

```
total_max_value=pd.DataFrame(asia_confirm.max().drop(['Date']), columns=['Max Value'])
total_max_value
```

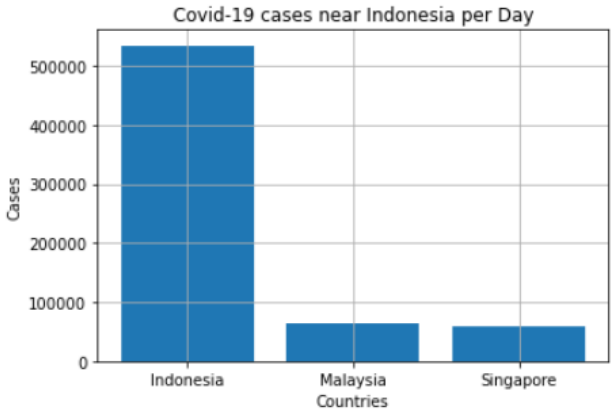
	Max Value
Indonesia	6267.0
Malaysia	2188.0
Singapore	1426.0

Gambar 4. Kasus Terkonfirmasi Harian Tertinggi di 3 Negara

C. Hasil Visualisasi Total Kasus Terkonfirmasi di 3 Negara dengan Bar Chart

Setelah melihat angka kasus harian tertinggi di 3 negara, lakukan visualisasi untuk melihat kasus terkonfirmasi di 3 negara dengan menggunakan *bar chart*.

```
total_id=asia_confirm['Indonesia'].sum()
total_my=asia_confirm['Malaysia'].sum()
total_sg=asia_confirm['Singapore'].sum()

exp_vals = [total_id, total_my, total_sg]
exp_labels = ["Indonesia", "Malaysia", "Singapore"]

plt.bar(exp_labels, exp_vals)
plt.grid()

plt.xlabel('Countries')
plt.ylabel('Cases')
plt.title('Covid-19 cases near Indonesia per Day')

plt.show()
```

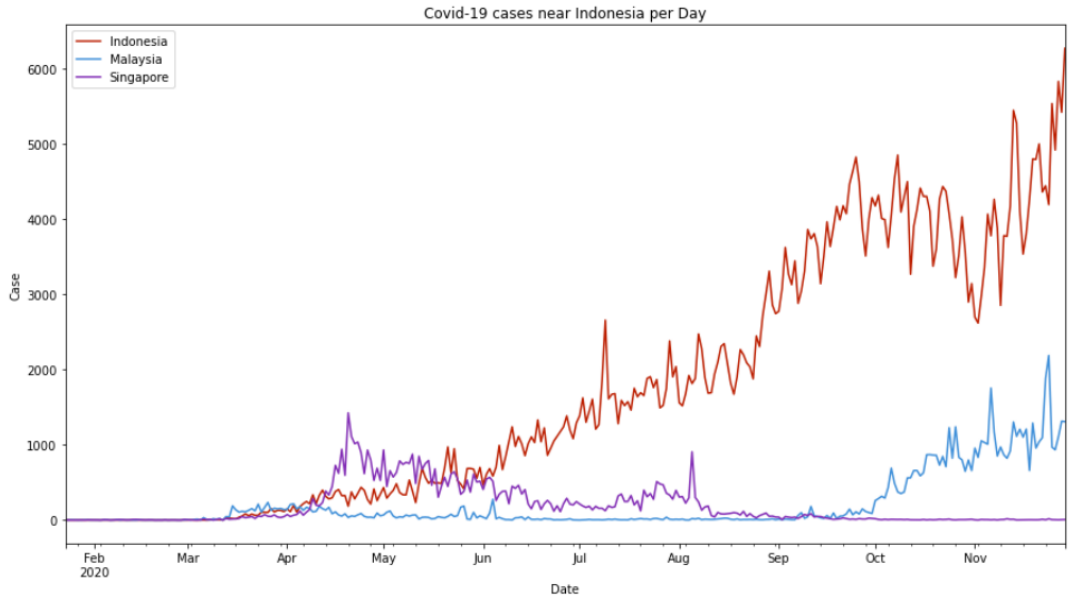


Gambar 5. Visualisasi Total Kasus Terkonfirmasi di 3 Negara dengan Bar Chart

D. Hasil Visualisasi Pergerakan Kasus Terkonfirmasi di 3 Negara dengan Line Chart

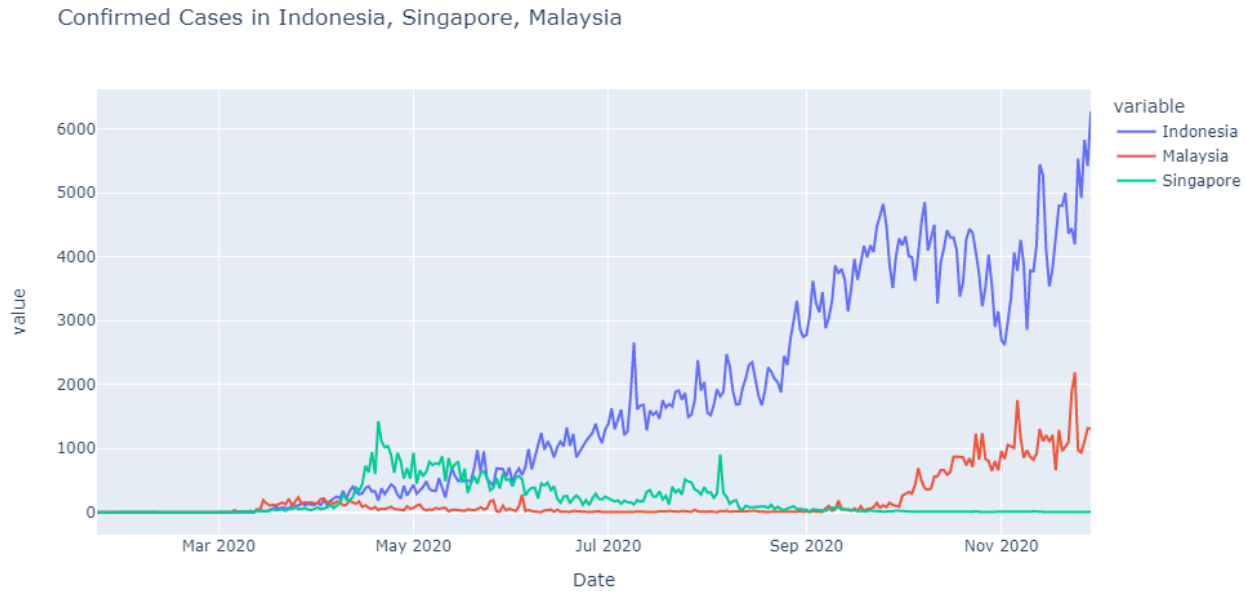
Visualisasi lain yang digunakan untuk melihat kasus terkonfirmasi di 3 negara yaitu dengan menggunakan *line chart*. *Line chart* dibuat dengan menggunakan *library* Matplotlib dan Plotly. Gambar 6 dan Gambar 7 menunjukkan negara Indonesia, memiliki pola yang mengarah naik setiap harinya dengan cukup signifikan dibandingkan dengan negara Malaysia dan Singapura.

```
asia_confirm.plot(kind='line', x='Date', y=['Indonesia', 'Malaysia', 'Singapore'], figsize=(15,8))
plt.title('Covid-19 cases near Indonesia per Day')
plt.ylabel('Case')
plt.show()
```



Gambar 6. Visualisasi Data Pergerakan Kasus Terkonfirmasi 3 Negara dengan *Line Chart* menggunakan Matplotlib

```
px.line(asia_confirm,
        x='Date',
        y=['Indonesia', 'Malaysia', 'Singapore'],
        labels={'x': "Date", 'y':'Indonesia'},
        title='Confirmed Cases in Indonesia, Singapore, Malaysia')
```



Gambar 7. Visualisasi Data Pergerakan Kasus Terkonfirmasi 3 Negara dengan *Line Chart* menggunakan Plotly

E. Hasil Eksplorasi Kenaikan Angka Kasus Terkonfirmasi dan Kasus Kematian di Indonesia per Bulan

Dari Gambar 8 terlihat bahwa kenaikan angka kasus terkonfirmasi di Indonesia per bulan terus terjadi dimulai pada bulan Maret 2020, disaat kasus positif COVID-19 pertama kali terdeteksi di Depok, Jawa Barat. Kenaikan berikutnya terjadi hingga November 2020 dengan kenaikan yang cukup signifikan.

```
unique_year = pd.DatetimeIndex(asia_confirm['Date']).year.unique()
unique_month = pd.DatetimeIndex(asia_confirm['Date']).month.unique()
list_sum = []
region = 'Indonesia'

d = pd.DatetimeIndex(asia_confirm['Date'])
for y in unique_year :
    for m in unique_month :
        tmp = asia_confirm[(d.month == m) & (d.year == y)]
        print_date = str(y) + '/' + str(m)
        list_sum.append([print_date, tmp[region].sum()])

list_day = [i[0] for i in list_sum]
list_sum = [i[1] for i in list_sum]
preview = pd.DataFrame(list_day, columns=['Date'])
preview['Total Confirmed Cases per Month'] = list_sum

print('Region : ' + region)
preview
```

```
Region : Indonesia
```

	Date	Total Confirmed Cases per Month
0	2020/1	0.0
1	2020/2	0.0
2	2020/3	1528.0
3	2020/4	8590.0
4	2020/5	16355.0
5	2020/6	29912.0
6	2020/7	51991.0
7	2020/8	66420.0
8	2020/9	112212.0
9	2020/10	123080.0
10	2020/11	124178.0

Gambar 8. Kenaikan Angka Kasus Terkonfirmasi di Indonesia per Bulan

F. Hasil Eksplorasi Kasus Terkonfirmasi dan Kasus Kematian di Negara Asia Tenggara

Eksplorasi dilakukan di negara yang berada di Asia Tenggara untuk melihat lebih luas kasus terkonfirmasi dan kasus kematian dari berbagai negara dalam Asia Tenggara. Eksplorasi dilakukan ke 12 negara yang berada dalam wilayah Asia Tenggara.

```
se_asia_confirm=confirm_cases[['Date', 'Brunei', 'Burma', 'Cambodia', 'Timor-Leste', 'Indonesia', 'Laos', 'Malaysia',
            'Philippines', 'Singapore', 'Thailand', 'Vietnam', 'India']]
se_asia_confirm['Date']=pd.to_datetime(se_asia_confirm.Date)
se_asia_confirm
```

	Date	Brunei	Burma	Cambodia	Timor-Leste	Indonesia	Laos	Malaysia	Philippines	Singapore	Thailand	Vietnam	India
1	2020-01-23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	2.0	0.0
2	2020-01-24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	0.0	0.0
3	2020-01-25	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	2.0	0.0	0.0
4	2020-01-26	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0
5	2020-01-27	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
308	2020-11-25	0.0	1330.0	0.0	0.0	5534.0	0.0	970.0	1193.0	7.0	16.0	5.0	44489.0
309	2020-11-26	0.0	1639.0	0.0	0.0	4917.0	0.0	935.0	1382.0	5.0	19.0	10.0	43082.0
310	2020-11-27	0.0	1428.0	1.0	0.0	5828.0	0.0	1109.0	1621.0	4.0	5.0	8.0	41322.0
311	2020-11-28	0.0	1344.0	7.0	0.0	5418.0	0.0	1315.0	1879.0	6.0	11.0	2.0	41810.0
312	2020-11-29	0.0	1509.0	8.0	0.0	6267.0	0.0	1309.0	2067.0	8.0	21.0	2.0	38772.0

312 rows × 13 columns

Gambar 9. Kasus Terkonfirmasi di Asia Tenggara

```
se_asia_death=death_cases[['Date', 'Brunei', 'Burma', 'Cambodia', 'Timor-Leste', 'Indonesia', 'Laos', 'Malaysia',
            'Philippines', 'Singapore', 'Thailand', 'Vietnam', 'India']]
se_asia_death
```

	Date	Brunei	Burma	Cambodia	Timor-Leste	Indonesia	Laos	Malaysia	Philippines	Singapore	Thailand	Vietnam	India
1	1/23/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1/24/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1/25/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1/26/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	1/27/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
308	11/25/20	0.0	26.0	0.0	0.0	114.0	0.0	4.0	30.0	0.0	0.0	0.0	524.0
309	11/26/20	0.0	36.0	0.0	0.0	127.0	0.0	3.0	27.0	0.0	0.0	0.0	492.0
310	11/27/20	0.0	19.0	0.0	0.0	169.0	0.0	2.0	13.0	0.0	0.0	0.0	485.0
311	11/28/20	0.0	22.0	0.0	0.0	125.0	0.0	4.0	78.0	1.0	0.0	0.0	496.0
312	11/29/20	0.0	31.0	0.0	0.0	169.0	0.0	3.0	40.0	0.0	0.0	0.0	443.0

312 rows × 13 columns

Gambar 10. Kasus Kematian di Asia Tenggara

G. *Hasil Prediksi Auto-ARIMA*

Setelah melakukan eksplorasi dan visualisasi, maka dilakukan prediksi dengan menggunakan metode Auto-ARIMA memakai *library* pmdarima. Gambar 11 merupakan hasil dari rangkuman dan *parameter* yang secara otomatis ditentukan oleh pmdarima dengan mencari *parameter* yang optimal. Gambar 12 merupakan hasil dari prediksi yang dilakukan oleh *library* pmdarima.

```
automodel = arimamodel(train)
print(automodel.summary())
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=4241.323, Time=1.11 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=4316.026, Time=0.03 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=4315.861, Time=0.05 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=4313.766, Time=0.16 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=4314.764, Time=0.02 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=4270.198, Time=0.63 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=4266.466, Time=0.64 sec
 ARIMA(3,1,2)(0,0,0)[0] intercept   : AIC=inf, Time=0.56 sec
 ARIMA(2,1,3)(0,0,0)[0] intercept   : AIC=4216.371, Time=1.12 sec
 ARIMA(1,1,3)(0,0,0)[0] intercept   : AIC=inf, Time=0.79 sec
 ARIMA(3,1,3)(0,0,0)[0] intercept   : AIC=4252.212, Time=1.13 sec
 ARIMA(2,1,4)(0,0,0)[0] intercept   : AIC=4253.040, Time=1.08 sec
 ARIMA(1,1,4)(0,0,0)[0] intercept   : AIC=4272.374, Time=1.09 sec
 ARIMA(3,1,4)(0,0,0)[0] intercept   : AIC=4211.136, Time=1.46 sec
 ARIMA(4,1,4)(0,0,0)[0] intercept   : AIC=4206.226, Time=1.36 sec
 ARIMA(4,1,3)(0,0,0)[0] intercept   : AIC=4204.461, Time=1.30 sec
 ARIMA(4,1,2)(0,0,0)[0] intercept   : AIC=4208.086, Time=1.11 sec
 ARIMA(5,1,3)(0,0,0)[0] intercept   : AIC=4204.395, Time=1.48 sec
 ARIMA(5,1,2)(0,0,0)[0] intercept   : AIC=4206.400, Time=1.31 sec
 ARIMA(5,1,4)(0,0,0)[0] intercept   : AIC=4204.142, Time=1.45 sec
 ARIMA(5,1,5)(0,0,0)[0] intercept   : AIC=4206.075, Time=1.88 sec
 ARIMA(4,1,5)(0,0,0)[0] intercept   : AIC=4207.094, Time=1.63 sec
 ARIMA(5,1,4)(0,0,0)[0]             : AIC=4207.327, Time=1.22 sec

Best model:  ARIMA(5,1,4)(0,0,0)[0] intercept
Total fit time: 22.697 seconds
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  305
Model:               SARIMAX(5, 1, 4)   Log Likelihood               -2091.071
Date:                Sun, 24 Oct 2021   AIC                           4204.142
Time:                        15:21:33   BIC                           4245.030
Sample:                             0   HQIC                          4220.498
                                  - 305
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     33.5743     17.977      1.868      0.062      -1.660      68.809
ar.L1          0.3529      0.523      0.675      0.500      -0.672       1.378
ar.L2         -0.1955      0.491     -0.398      0.691      -1.158       0.767
ar.L3         -0.6759      0.496     -1.362      0.173      -1.649       0.297
ar.L4         -0.0901      0.132     -0.684      0.494      -0.348       0.168
ar.L5         -0.1859      0.197     -0.941      0.346      -0.573       0.201
ma.L1         -0.7579      0.525     -1.443      0.149      -1.788       0.272
ma.L2          0.1071      0.700      0.153      0.878      -1.265       1.479
ma.L3          0.7847      0.587      1.337      0.181      -0.366       1.935
ma.L4         -0.3243      0.162     -2.000      0.046      -0.642      -0.006
sigma2      6.326e+04   4076.338     15.518      0.000    5.53e+04    7.12e+04
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):               243.61
Prob(Q):                              1.00   Prob(JB):                         0.00
Heteroskedasticity (H):              34.41   Skew:                             0.19
Prob(H) (two-sided):                  0.00   Kurtosis:                         7.37
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Gambar 11. Summary Model Auto-ARIMA

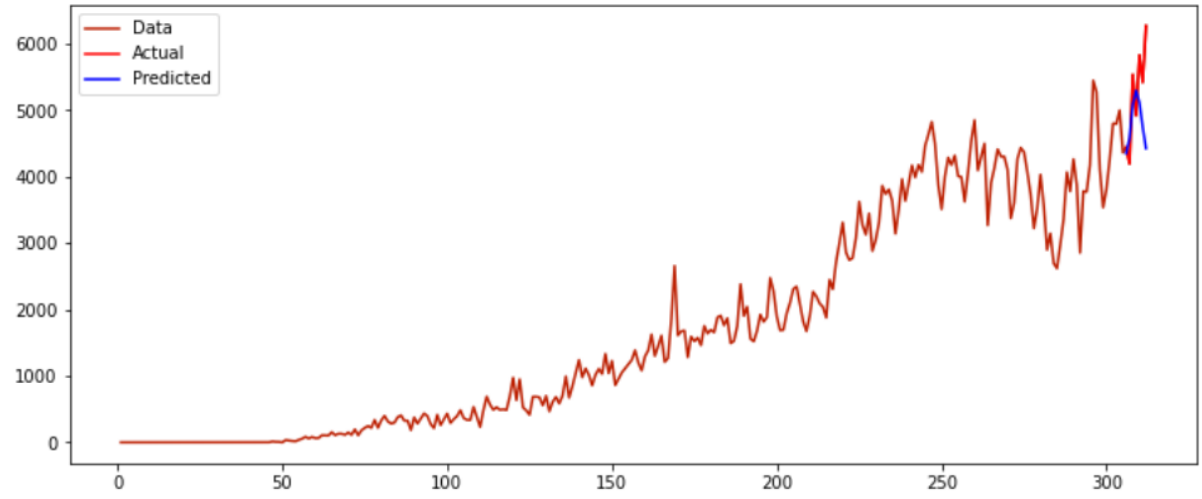
```
pred_arima=automodel.predict(n_periods=7)
pred_arima
```

```
array([4011.55922246, 4043.82528538, 4312.91554267, 4760.04985449,
       5027.01519246, 4947.38420589, 4568.19658238])
```

Gambar 12. Tampilan Prediksi 7 Tahap/Hari Kedepan dengan Auto-ARIMA

```
plt.figure(figsize=(12,5))
plt.plot(id, label="Data")
plt.plot(test.index, test, label="Actual", color='red')
plt.plot(test.index, pred_arima, label="Predicted", color='blue')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1a09a037a00>
```



Gambar 13 *Line Chart* Data Aktual dengan Data Prediksi Auto-ARIMA

H. Hasil Prediksi ARIMA

Penggunaan Statsmodels sebagai *library* untuk memprediksi dengan metode ARIMA memiliki tahap yang serupa dengan menggunakan *library* Pmdarima. Perlu dilakukan pembuatan model dengan tujuan menentukan parameter *p*, *d*, *q* yang optimal. Dalam kasus ini, *p*, *d*, *q* yang digunakan sama dengan yang diperoleh dari *library* Pmdarima, yakni 5, 1, 4. Gambar 14 merupakan hasil dari rangkuman *model* yang dimiliki oleh Statsmodels. Gambar 15 merupakan hasil dari prediksi yang dilakukan oleh *library* Statsmodels.

```
arima = ARIMA(train, order=(5, 1, 4)).fit()
arima.summary()
```

SARIMAX Results

Dep. Variable:	Indonesia	No. Observations:	305
Model:	ARIMA(5, 1, 4)	Log Likelihood	-2093.663
Date:	Sun, 24 Oct 2021	AIC	4207.327
Time:	15:21:36	BIC	4244.497
Sample:	0	HQIC	4222.196
	- 305		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.4268	0.583	0.732	0.464	-0.716	1.569
ar.L2	-0.2525	0.558	-0.452	0.651	-1.346	0.841
ar.L3	-0.6086	0.554	-1.099	0.272	-1.694	0.477
ar.L4	-0.1031	0.139	-0.741	0.459	-0.376	0.170
ar.L5	-0.1474	0.206	-0.715	0.475	-0.552	0.257
ma.L1	-0.8094	0.584	-1.385	0.166	-1.955	0.336
ma.L2	0.1977	0.783	0.252	0.801	-1.338	1.733
ma.L3	0.7067	0.658	1.074	0.283	-0.583	1.996
ma.L4	-0.2962	0.176	-1.685	0.092	-0.641	0.048
sigma2	6.429e+04	4167.658	15.427	0.000	5.61e+04	7.25e+04

Ljung-Box (L1) (Q):	0.15	Jarque-Bera (JB):	231.34
Prob(Q):	0.70	Prob(JB):	0.00
Heteroskedasticity (H):	51.70	Skew:	0.14
Prob(H) (two-sided):	0.00	Kurtosis:	7.26

Warnings:
[1] Covariance matrix calculated using the outer product of gradie

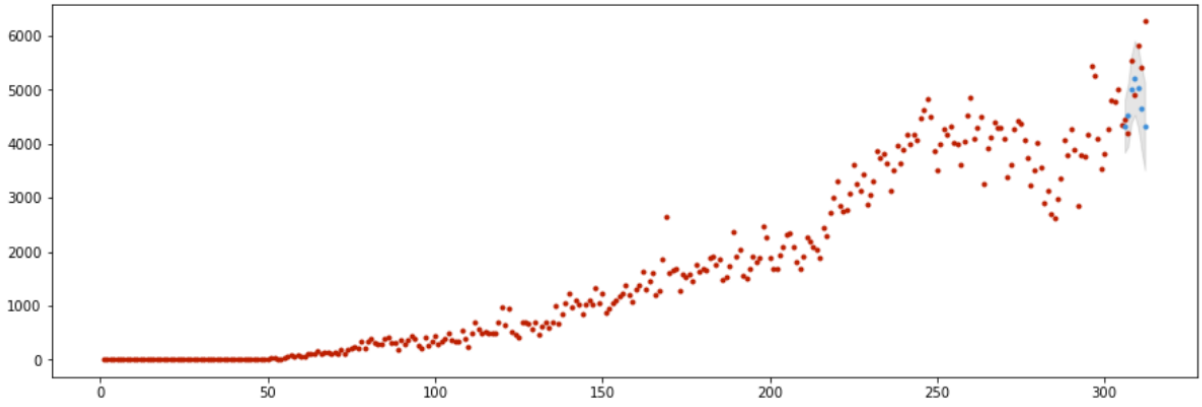
Gambar 14. *Summary* dari Model ARIMA

```
predict_arima = arima.predict(len(id), len(id)+6, typ='levels')
predict_arima
```

```
312    4045.334522
313    3955.348136
314    4222.752652
315    4641.573802
316    4903.711953
317    4811.425869
318    4436.641827
Name: predicted_mean, dtype: float64
```

Gambar 15. Tampilan Prediksi 7 Tahap/Hari Kedepan dengan ARIMA

```
fig, ax = plt.subplots(figsize=(15, 5))

id.loc['1':].plot(ax=ax, style='.')

fcast_f = arima.get_forecast(n_pred).summary_frame()
fcast_f['mean'].plot(ax=ax, style='.')
ax.fill_between(fcast_f.index, fcast_f['mean_ci_lower'], fcast_f['mean_ci_upper'], color='k', alpha=0.1);
```



Gambar 16 *Scatter Plot* Data Aktual dengan Data Prediksi ARIMA

I. Hasil Prediksi Prophet

Metode lain yang digunakan untuk melakukan prediksi yaitu dengan menggunakan *library* Prophet. Gambar 17 menunjukan hasil prediksi yang berada dalam kolom “yhat”.

```
future = model_prophet.make_future_dataframe(periods=7)

forecast_prophet = model_prophet.predict(future)
forecast_prophet[['ds', 'trend', 'weekly', 'yhat', 'yhat_lower', 'yhat_upper']]
```

	ds	trend	weekly	yhat	yhat_lower	yhat_upper
0	2020-01-24	-194.460623	110.930298	-83.530325	-558.790533	425.939462
1	2020-01-25	-188.427997	110.041711	-78.386286	-560.840361	379.680318
2	2020-01-26	-182.395371	-32.504228	-214.899600	-721.533791	245.949695
3	2020-01-27	-176.362746	-250.220180	-426.582926	-912.426085	5.540018
4	2020-01-28	-170.330120	-118.055732	-288.385852	-799.755710	172.944132
...	...	...	...	...	...	...
313	2020-12-02	4730.930873	49.167902	4780.098775	4311.539312	5297.923501
314	2020-12-03	4748.184064	130.640230	4878.824294	4427.377874	5361.385824
315	2020-12-04	4765.437255	110.930298	4876.367552	4383.624596	5361.696720
316	2020-12-05	4782.690445	110.041711	4892.732156	4416.240963	5351.398433
317	2020-12-06	4799.943636	-32.504228	4767.439408	4261.565791	5237.002332

318 rows × 6 columns

Gambar 17. Tampilan Proses Prediksi Prophet

J. Hasil Komparasi Performa

Dari ketiga *library* yang digunakan untuk melakukan prediksi, maka dilakukan komparasi dari hasil performa masing-masing metode prediksi. Gambar 18 merupakan hasil dari komparasi antara Auto-ARIMA, ARIMA, dan Prophet.

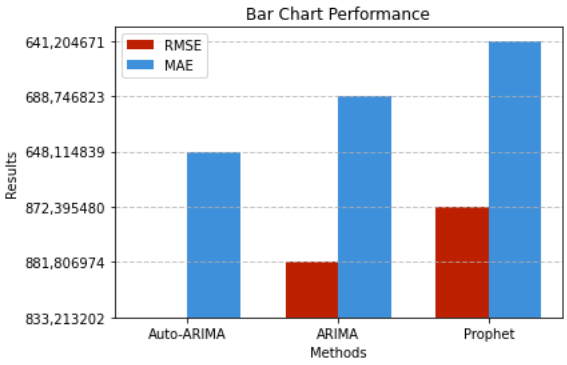
```
rmse_prophet = cc_p.iloc[6]['rmse']
mae_prophet = cc_p.iloc[6]['mae']

data = {'Methods':['Auto-ARIMA', 'ARIMA', 'Prophet'],
        'RMSE':[rmse_autoarima, rmse_arima, rmse_prophet],
       'MAE': [mae_autoarima, mae_arima, mae_prophet]}

compare = pd.DataFrame(data)
compare
```

	Methods	RMSE	MAE
0	Auto-ARIMA	884.138366	703.864873
1	ARIMA	969.611735	797.315891
2	Prophet	872.395480	641.204671

Gambar 18. Perbandingan Auto-ARIMA, ARIMA, dan Prophet

```
width = 0.35

plt.bar(x-width/2, data_1, width, label='RMSE')
plt.bar(x+width/2, data_2, width, label='MAE')

plt.xticks(x, kategori)

plt.grid(linestyle='--',
        linewidth=1,
        axis='y',
        alpha=0.75)

plt.xlabel('Methods')
plt.ylabel('Results')
plt.title('Bar Chart Performance')

plt.legend()
plt.show()
```

```
INFO:matplotlib.category:Using categorical units to plo
trings should be plotted as numbers, cast to the appropr
INFO:matplotlib.category:Using categorical units to plo
trings should be plotted as numbers, cast to the appropr
INFO:matplotlib.category:Using categorical units to plo
trings should be plotted as numbers, cast to the appropr
```



Gambar 19. *Bar Chart* Perbandingan Auto-ARIMA, ARIMA, dan Prophet

Berdasarkan Gambar 18 perbandingan masing-masing metode prediksi dengan menggunakan beragam *library*, terlihat bahwa penggunaan Prophet sebagai metode prediksi mendapatkan nilai RMSE dan MAE yang paling mendekati nol sehingga penggunaan Prophet dalam prediksi termasuk dalam kategori optimal, melihat dari data yang tidak memiliki pola *seasonal* dan memiliki sebaran titik data yang kurang beraturan.

K. Kombinasi ARIMA dan Prophet

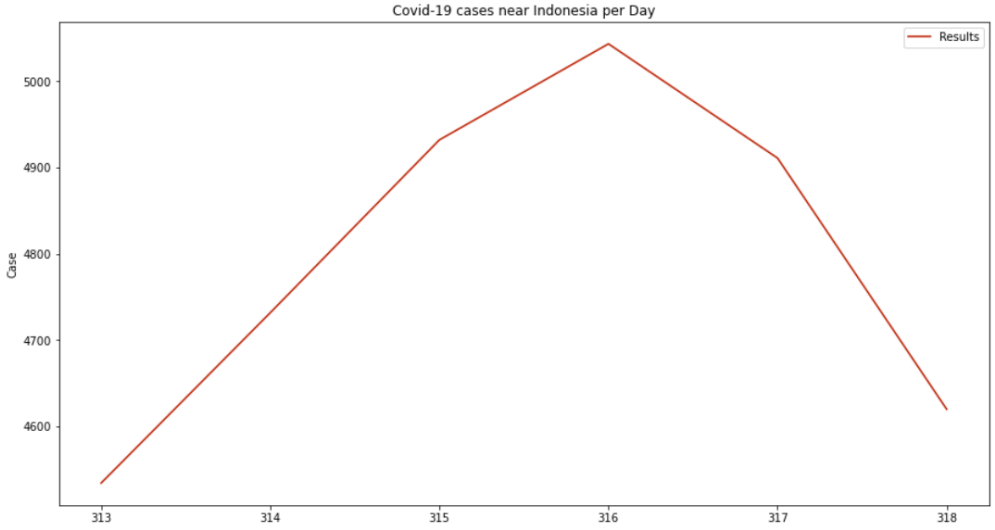
Metode ARIMA dan Prophet memiliki hasil yang tidak jauh berbeda apabila berdasarkan dari hasil uji performa menggunakan RMSE dan MAE. Maka dari itu, terbesit ide untuk melakukan penggabungan antara ARIMA dan Prophet. Namun, karena ARIMA dan Prophet merupakan pendekatan yang berbeda, maka hal yang dapat dilakukan untuk membuat kombinasi antara ARIMA dan Prophet yaitu dengan mengambil rata-rata dari hasil prediksi keduanya. Pengambilan rata-rata dilakukan dengan memulai pada indeks yang sama antara ARIMA dan Prophet. Gambar 20 merupakan hasil dari kombinasi antara ARIMA dengan Prophet.

```
combine = ((predict_arima+fcast_prophet)/2).to_frame("Results").dropna()
combine
```

	Results
313	4534.253394
314	4731.601866
315	4931.881336
316	5043.199956
317	4910.812502
318	4619.831203

Gambar 20. Hasil Kombinasi ARIMA dan Prophet

```
combine.plot(kind='line', y=['Results'], figsize=(15,8))
plt.title('Covid-19 cases near Indonesia per Day')
plt.ylabel('Case')
plt.show()
```



Gambar 21 *Line Plot* Hasil Kombinasi ARIMA dan Prophet

V. KESIMPULAN

Dataset COVID-19 yang terdiri dari data *confirmed cases* dan *deaths cases* dalam jumlah 312 baris dan 272 kolom dapat diolah menjadi data dalam bentuk tabel sederhana dengan pemilihan kolom dan baris tertentu sehingga menjadi informatif dan mudah dilakukan analisis serta eksplorasi mendalam. Dari *dataset* COVID-19 yang sudah diolah, dilakukan visualisasi menggunakan berbagai *plot* seperti *line plot* dan *bar plot* dari beragam *library* agar mempermudah dalam pemahaman informasi dari *dataset*.

Sesuai dengan karakteristik dari keseluruhan *dataset* COVID-19 yaitu *time series data* yang berisi data waktu dan nilai dari data, maka dilakukan prediksi untuk mengetahui pergerakan dari kasus COVID-19 di masa yang akan datang dengan menggunakan metode ARIMA dan Prophet, dengan hasil Prophet lebih optimal untuk melakukan prediksi dengan memperoleh nilai RMSE dan MAE yang lebih mendekati nol dibandingkan dengan ARIMA sehingga hasil lebih optimal.

DAFTAR PUSTAKA

[1] F. Lo, "What is Data Science?," Data Jobs, [Online]. Available: https://datajobs.com/what-is-data-science. [Diakses 11 Maret 2021].
[2] H. Sharma, "What Is Data Science? A Beginner's Guide To Data Science," eureka!, 25 November 2020. [Online]. Available: https://www.edureka.co/blog/what-is-data-science/. [Diakses 3 Maret 2021].
[3] A. Goldbloom, "COVID-19 data from John Hopkins University | Kaggle," 2 November 2020. [Online]. Available: https://www.kaggle.com/antgoldbloom/covid19-data-from-john-hopkins-university. [Diakses 07 April 2021].
[4] J. VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, Sebastopol: O'Reilly Media, Inc., 2017.
[5] R. Schutt dan C. O'Neil, Doing Data Science: Straight Talk from the Frontline, Sebastopol: O'Reilly Media, Inc., 2013.
[6] S. K. Mukhiya dan U. Ahmed, Hands-On Exploratory Data Analysis with Python, Birmingham: Packt Publishing Ltd., 2020.
[7] D. C. Montgomery, C. L. Jennings dan M. Kulahci, Introduction to Time Series Analysis and Forecasting; Second Edition, Hoboken: John Wiley & Sons, Inc., 2015.
[8] W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, Sebastopol: O'Reilly Media, Inc., 2017.
[9] R. J. Hyndman dan G. Athanasopoulos, Forecasting: Principles and Practice, Australia: Monash University, 2018.
[10] S. Prabhakaran, "ARIMA Model – Complete Guide to Time Series Forecasting in Python," Machine Learning Plus, 18 Februari 2019. [Online]. Available: https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/. [Diakses 17 Maret 2021]
[11] L. A. Tofani dan A. Mauludiyanto, "Peramalan Trafik Sms Area Jabotabek dengan Metode Arima," *JURNAL TEKNIK ITS,* vol. I, no. 1, pp. 139-144, 2012.
[12] Prophet, "Prophet: Forecasting at scale," Facebook Open Source, [Online]. Available: https://facebook.github.io/prophet/. [Diakses 2021 Maret 17].
[13] B. V. Vishwas dan A. Patel, Hands-on Time Series Analysis with Python From Basics to Bleeding Edge Techniques, Bengaluru: Apress Media, 2020.