In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
df = pd.read_csv("covid_19_indonesia_time_series_all.csv")
df.head()
```

Out[2]:

	Date	Location ISO Code	Location	New Cases	New Deaths	New Recovered	New Active Cases	Total Cases	Total Deaths	Total Recovered	...	Latitude	New Cases per Million	Total Cases per Million	New Deaths per Million	Total Deaths per Million	Total Deaths per 100rb	Case Fatality Rate
0	3/1/2020	ID-JK	DKI Jakarta	2	0	0	2	39	20	75	...	-6.204699	0.18	3.60	0.0	1.84	0.18	51.28%
1	3/2/2020	ID-JK	DKI Jakarta	2	0	0	2	41	20	75	...	-6.204699	0.18	3.78	0.0	1.84	0.18	48.78%
2	3/2/2020	IDN	Indonesia	2	0	0	2	2	0	0	...	-0.789275	0.01	0.01	0.0	0.00	0.00	0.00%
3	3/2/2020	ID-RI	Riau	1	0	0	1	1	0	1	...	0.511648	0.16	0.16	0.0	0.00	0.00	0.00%
4	3/3/2020	ID-JK	DKI Jakarta	2	0	0	2	43	20	75	...	-6.204699	0.18	3.96	0.0	1.84	0.18	46.51%

5 rows × 38 columns

In [3]:
```
to_drop = ['Date',
           'Location ISO Code',
           'Latitude','New Cases per Million',
           'Total Cases per Million',
           'New Deaths per Million',
           'Total Deaths per Million',
           'Total Deaths per 100rb',
            'Case Fatality Rate',
           'Case Recovered Rate',
           'Growth Factor of New Cases',
           'Growth Factor of New Deaths',
           'Longitude',
           'Population',
           'Population Density',
           'Area (km2)',
           'Total Rural Villages',
           'Total Urban Villages',
           'Total Districts',
           'Total Cities',
           'Total Regencies',
           'Special Status',
           'Location Level',
           'Time Zone',
           'Continent',
           'Country',
            'City or Regency',
           'Province','Island']
```

In [4]:
```
df.drop(to_drop, inplace=True, axis=1)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31822 entries, 0 to 31821
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Location            31822 non-null  object
 1   New Cases           31822 non-null  int64 
 2   New Deaths          31822 non-null  int64 
 3   New Recovered       31822 non-null  int64 
 4   New Active Cases    31822 non-null  int64 
 5   Total Cases         31822 non-null  int64 
 6   Total Deaths        31822 non-null  int64 
 7   Total Recovered     31822 non-null  int64 
 8   Total Active Cases  31822 non-null  int64 
dtypes: int64(8), object(1)
memory usage: 2.2+ MB
```

In [6]:
```
x = df.drop(["Location"], axis=1)
x.head(11)
```

Out[6]:

	New Cases	New Deaths	New Recovered	New Active Cases	Total Cases	Total Deaths	Total Recovered	Total Active Cases
0	2	0	0	2	39	20	75	-56
1	2	0	0	2	41	20	75	-54
2	2	0	0	2	2	0	0	2
3	1	0	0	1	1	0	1	0
4	2	0	0	2	43	20	75	-52
5	0	0	0	0	2	0	0	2
6	1	1	0	0	1	1	60	-60
7	0	0	0	0	1	0	1	0
8	2	0	0	2	45	20	75	-50
9	0	0	0	0	2	0	0	2
10	1	0	0	1	2	1	60	-59

In [7]:
```
y = df["Location"]
y.head(11)
```

Out[7]:
```
0     DKI Jakarta
1     DKI Jakarta
2       Indonesia
3            Riau
4     DKI Jakarta
5       Indonesia
6      Jawa Barat
7            Riau
8     DKI Jakarta
9       Indonesia
10     Jawa Barat
Name: Location, dtype: object
```

In [8]:
```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB

modelnb = GaussianNB()
```

In [9]:
```
nbtrain = modelnb.fit(x, y)
df.head(11)
```

Out[9]:

	Location	New Cases	New Deaths	New Recovered	New Active Cases	Total Cases	Total Deaths	Total Recovered	Total Active Cases
0	DKI Jakarta	2	0	0	2	39	20	75	-56
1	DKI Jakarta	2	0	0	2	41	20	75	-54
2	Indonesia	2	0	0	2	2	0	0	2
3	Riau	1	0	0	1	1	0	1	0
4	DKI Jakarta	2	0	0	2	43	20	75	-52
5	Indonesia	0	0	0	0	2	0	0	2
6	Jawa Barat	1	1	0	0	1	1	60	-60
7	Riau	0	0	0	0	1	0	1	0
8	DKI Jakarta	2	0	0	2	45	20	75	-50
9	Indonesia	0	0	0	0	2	0	0	2
10	Jawa Barat	1	0	0	1	2	1	60	-59

In [10]:
```
x_test = df.drop(["Location"], axis=1)
x_test.head(11)
```

Out[10]:

	New Cases	New Deaths	New Recovered	New Active Cases	Total Cases	Total Deaths	Total Recovered	Total Active Cases
0	2	0	0	2	39	20	75	-56
1	2	0	0	2	41	20	75	-54
2	2	0	0	2	2	0	0	2
3	1	0	0	1	1	0	1	0
4	2	0	0	2	43	20	75	-52
5	0	0	0	0	2	0	0	2
6	1	1	0	0	1	1	60	-60
7	0	0	0	0	1	0	1	0
8	2	0	0	2	45	20	75	-50
9	0	0	0	0	2	0	0	2
10	1	0	0	1	2	1	60	-59

In [11]:
```
y_uji = df["Location"]
y_uji.head(11)
```

Out[11]:
```
0     DKI Jakarta
1     DKI Jakarta
2       Indonesia
3            Riau
4     DKI Jakarta
5       Indonesia
6      Jawa Barat
7            Riau
8     DKI Jakarta
9       Indonesia
10     Jawa Barat
Name: Location, dtype: object
```

In [12]:
```
Y_predict = nbtrain.predict(x_test)
print("Prediksi Naive Bayes : ",Y_predict)
```

```
Prediksi Naive Bayes :  ['Sulawesi Barat' 'Sulawesi Barat' 'Sulawesi Barat' ... 'Sumatera Selatan'
 'Riau' 'Indonesia']
```

In [13]:
```
from sklearn.metrics import accuracy_score
accuracy= accuracy_score(y_uji, Y_predict)
print("Akurasi Naive Bayes : ",accuracy)
```

```
Akurasi Naive Bayes :  0.2051725221544843
```

In [14]:
```
# Menghitung nilai akurasi dari klasifikasi naive bayes
from sklearn.metrics import classification_report
print(classification_report(y_uji, Y_predict))
```

```
                            precision    recall  f1-score   support

                      Aceh       0.51      0.41      0.45       904
                      Bali       0.11      0.01      0.02       919
                    Banten       0.18      0.11      0.14       924
                  Bengkulu       0.01      0.00      0.01       899
               DKI Jakarta       0.55      0.47      0.50       929
Daerah Istimewa Yogyakarta       0.09      0.03      0.05       914
                 Gorontalo       0.20      0.75      0.32       888
                 Indonesia       0.81      0.69      0.74       929
                     Jambi       0.01      0.00      0.00       910
                Jawa Barat       0.24      0.08      0.12       927
               Jawa Tengah       0.45      0.22      0.30       922
                Jawa Timur       0.47      0.41      0.44       912
          Kalimantan Barat       0.11      0.00      0.00       901
       Kalimantan Selatan       0.38      0.34      0.36       900
        Kalimantan Tengah       0.10      0.05      0.07       904
         Kalimantan Timur       0.27      0.22      0.24       916
         Kalimantan Utara       0.03      0.03      0.03       902
 Kepulauan Bangka Belitung       0.16      0.34      0.21       900
           Kepulauan Riau       0.00      0.00      0.00       911
                  Lampung       0.31      0.04      0.08       904
                   Maluku       0.03      0.03      0.03       908
             Maluku Utara       0.04      0.07      0.05       907
       Nusa Tenggara Barat       0.17      0.39      0.24       899
       Nusa Tenggara Timur       0.02      0.01      0.01       890
                    Papua       0.45      0.38      0.41       908
             Papua Barat       0.42      0.22      0.29       903
                    Riau       0.21      0.37      0.27       928
          Sulawesi Barat       0.04      0.25      0.07       901
        Sulawesi Selatan       0.10      0.04      0.05       911
         Sulawesi Tengah       0.00      0.00      0.00       904
       Sulawesi Tenggara       0.22      0.37      0.28       920
          Sulawesi Utara       0.05      0.04      0.04       904
          Sumatera Barat       0.41      0.38      0.40       904
        Sumatera Selatan       0.48      0.38      0.43       907
          Sumatera Utara       0.02      0.01      0.02       913

                accuracy                           0.21     31822
               macro avg       0.22      0.20      0.19     31822
            weighted avg       0.22      0.21      0.19     31822
```