

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

In [2]: df=pd.read_csv('owid-covid-data_2.csv')
df.tail()
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	
Out[2]:	76406	ZWE	Africa	Zimbabwe	2021-03-17	36552.0	17.0	30.143	1508.0	1.0	2.714	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791
	76407	ZWE	Africa	Zimbabwe	2021-03-18	36611.0	59.0	33.429	1509.0	1.0	2.429	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791
	76408	ZWE	Africa	Zimbabwe	2021-03-19	36652.0	41.0	32.714	1510.0	1.0	2.000	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791
	76409	ZWE	Africa	Zimbabwe	2021-03-20	36662.0	10.0	27.286	1510.0	0.0	1.286	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791
	76410	ZWE	Africa	Zimbabwe	2021-03-21	36665.0	3.0	25.857	1512.0	2.0	1.286	...	1899.775	21.4	307.846	1.82	1.6	30.7	36.791

5 rows × 59 columns

Menerapkan syntax"Conditional Loop"

```
In [3]: for i in range (len(df)):
        if df.at[i,'location']=='United Kingdom':
            print("United Kingdom berada pada baris ke-",i)
            break

United Kingdom berada pada baris ke- 72007

In [4]: df_uk=df.loc[df['location']=='United Kingdom']

In [5]: df_uk=df_uk.reset_index(drop=True)
df_uk.tail()
```

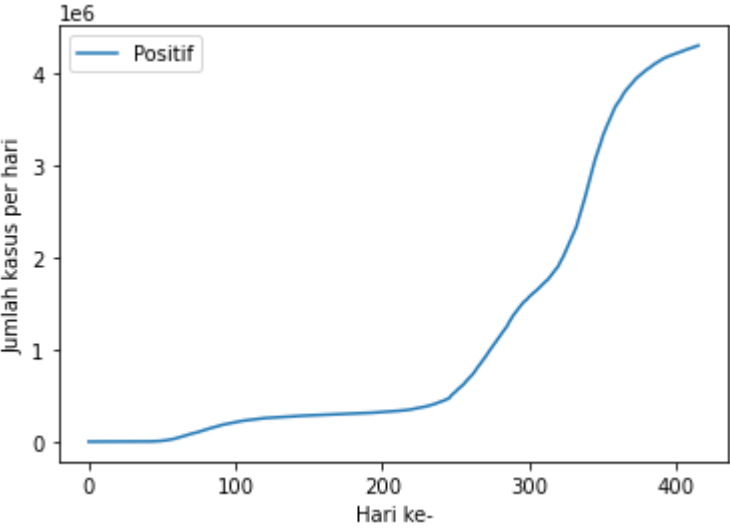
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_death_rate	diabetes_prevalence	female_smokers	male_smokers	handwashing_facilities	hos
Out[5]:	411	GBR	Europe	United Kingdom	2021-03-17	4287996.0	5793.0	5731.000	126068.0	141.0	120.857	...	39753.244	0.2	122.137	4.28	20.0	24.7	NaN
	412	GBR	Europe	United Kingdom	2021-03-18	4294328.0	6332.0	5659.143	126163.0	95.0	108.571	...	39753.244	0.2	122.137	4.28	20.0	24.7	NaN
	413	GBR	Europe	United Kingdom	2021-03-19	4299200.0	4872.0	5400.286	126263.0	100.0	97.714	...	39753.244	0.2	122.137	4.28	20.0	24.7	NaN
	414	GBR	Europe	United Kingdom	2021-03-20	4304839.0	5639.0	5403.429	126359.0	96.0	94.000	...	39753.244	0.2	122.137	4.28	20.0	24.7	NaN
	415	GBR	Europe	United Kingdom	2021-03-21	4310195.0	5356.0	5497.857	126393.0	34.0	91.429	...	39753.244	0.2	122.137	4.28	20.0	24.7	NaN

5 rows × 59 columns

```
In [6]: df_uk=df_uk[['date','total_cases']]
df_uk.head()
```

	date	total_cases	
Out[6]:	0	2020-01-31	2.0
	1	2020-02-01	2.0
	2	2020-02-02	2.0
	3	2020-02-03	8.0
	4	2020-02-04	8.0

```
In [7]: plt.plot(df_uk['total_cases'],label='Positif')
plt.xlabel('Hari ke-')
plt.ylabel('Jumlah kasus per hari')
plt.legend(loc='best')
plt.show()
```



Fungsi Sigmoid

```
In [8]: x=list(df_uk.index)
y=list(df_uk['total_cases'])

In [9]: def kurva_sigmoid(t,a,t0,c):
        return c/(1+np.exp(-(t-t0)/(a)))

In [10]: from scipy.optimize import curve_fit, fsolve

In [11]: fit=curve_fit(kurva_sigmoid,x,y,method='lm')
varA,varB=fit

In [12]: std_er=np.zeros(len(varA))
for i in range(len(varA)):
    std_er[i]=np.sqrt(varB[i][i])

In [13]: a=varA[0]+std_er[0]
t0=varA[1]+std_er[1]
c=varA[2]+std_er[2]

In [14]: def puncak(x):
        return kurva_sigmoid(x,a,t0,c)-int(c)

In [15]: n_puncak=int(fsolve(puncak,t0))

In [16]: print("Hari puncak wabah adalah hari ke-"+str(n_puncak))

Hari puncak wabah adalah hari ke-1044

In [17]: n_0=max(x)+1
pred_x=list(range(n_0,n_puncak))

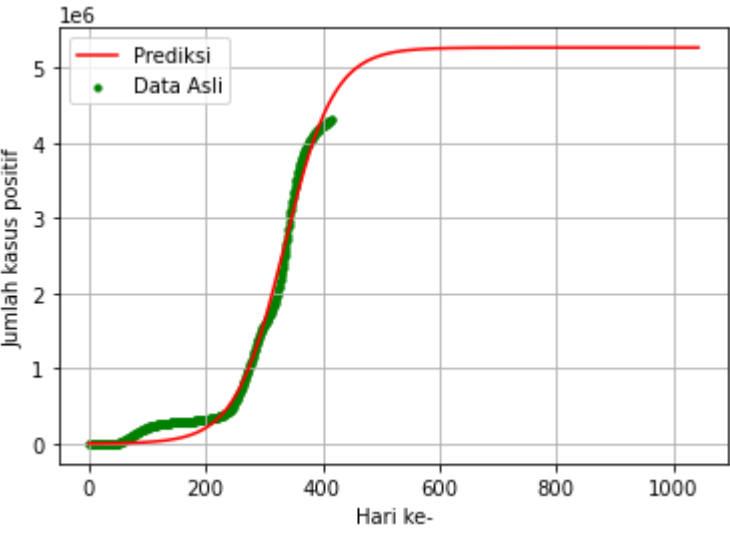
In [18]: pred_y=np.zeros(len(x+pred_x))
for i in range(n_puncak):
    pred_y[i]=kurva_sigmoid(i,a,t0,c)

In [19]: print('Prediksi jumlah puncak', int(pred_y[-1]),'orang')

Prediksi jumlah puncak 5268746 orang
```

membuat kurva

```
In [20]: plt.plot(x+pred_x,pred_y,label='Prediksi',color='red')
plt.scatter(x,y,label='Data Asli',s=10, color='green')
plt.xlabel('Hari ke-')
plt.ylabel('Jumlah kasus positif')
plt.grid()
plt.legend(loc='best')
plt.show()
```



```
In [27]: def akurasi_r2(y_asli,y_prediksi,x):
        atas=sum((y_asli-y_prediksi[0:len(x)])**2)
        bawah=sum((y_asli-(np.mean(y)))**2)
        r=1-(atas/bawah)
        return r

In [28]: akurasi=akurasi_r2(y,pred_y,x)
print(akurasi*100)

98.92698402316105

Package date time
```

```
In [29]: from datetime import datetime, timedelta
df_uk.head()
```

	date	total_cases	
Out[29]:	0	2020-01-31	2.0
	1	2020-02-01	2.0
	2	2020-02-02	2.0
	3	2020-02-03	8.0
	4	2020-02-04	8.0

```
In [30]: fmt="%Y-%m-%d"
n1=df_uk['date'][0]
tgl_0=datetime.strptime(n1,fmt)

In [31]: tgl_puncak=tgl_0+timedelta(days=n_puncak)

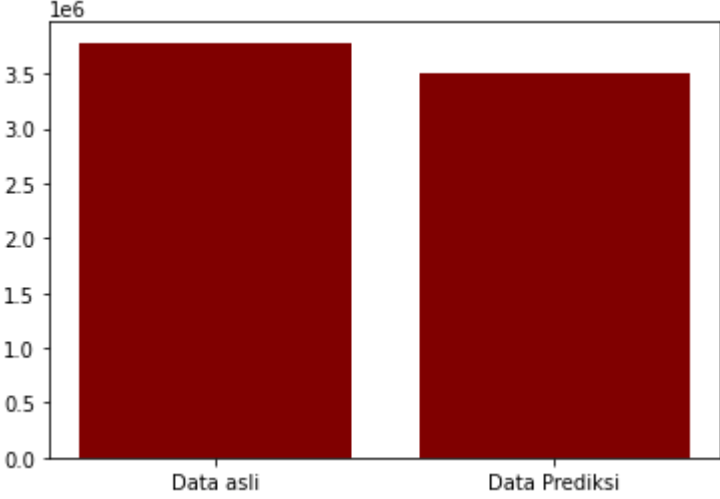
In [32]: h_puncak=tgl_puncak.strftime(fmt)

In [36]: print("Tanggal puncak jumlah kasus COVID-19 di UK",h_puncak)

Tanggal puncak jumlah kasus COVID-19 di UK 2022-12-10
```

Membuat kurva bar dengan "Matplotlib"

```
In [39]: label=['Data asli','Data Prediksi']
value=[df_uk['total_cases'].iloc[364],pred_y[364]]
plt.bar(label,value, color='maroon')
plt.show()
```



```
In [ ]:
```