In [145…
```
import sklearn.cluster as cluster
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [146…
```
#Visualisasi data covid-19 di us
df = pd.read_csv('us-covid.csv')
df.head()
```

Out[146]:

| | Data As Of | Start Date | End Date | Group | Year | Month | State | Sex | Age Group | COVID-19 Deaths |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 05/11/2022 | 01/01/2020 | 05/07/2022 | By Total | NaN | NaN | United States | All Sexes | All Ages | 998698.0 |
| **1** | 05/11/2022 | 01/01/2020 | 05/07/2022 | By Total | NaN | NaN | United States | All Sexes | Under 1 year | 275.0 |
| **2** | 05/11/2022 | 01/01/2020 | 05/07/2022 | By Total | NaN | NaN | United States | All Sexes | 0-17 years | 1045.0 |
| **3** | 05/11/2022 | 01/01/2020 | 05/07/2022 | By Total | NaN | NaN | United States | All Sexes | 1-4 years | 138.0 |
| **4** | 05/11/2022 | 01/01/2020 | 05/07/2022 | By Total | NaN | NaN | United States | All Sexes | 5-14 years | 334.0 |

In [147…
```
#1) menampilan data covid setiap negara bagian us pada tahun 2020
#Mengambil data negara bagian tanpa ada duplikasi data
unique_states = df[df['COVID-19 Deaths'] > 0][df['State'] != 'United Stat
unique_states.sort()
print(unique_states)
```

```
['Alabama' 'Alaska' 'Arizona' 'Arkansas' 'California' 'Colorado'
 'Connecticut' 'Delaware' 'District of Columbia' 'Florida' 'Georgia'
 'Hawaii' 'Idaho' 'Illinois' 'Indiana' 'Iowa' 'Kansas' 'Kentucky'
 'Louisiana' 'Maine' 'Maryland' 'Massachusetts' 'Michigan' 'Minnesota'
 'Mississippi' 'Missouri' 'Montana' 'Nebraska' 'Nevada' 'New Hampshire'
 'New Jersey' 'New Mexico' 'New York' 'New York City' 'North Carolina'
 'North Dakota' 'Ohio' 'Oklahoma' 'Oregon' 'Pennsylvania' 'Puerto Rico'
 'Rhode Island' 'South Carolina' 'South Dakota' 'Tennessee' 'Texas' 'Uta
h'
 'Vermont' 'Virginia' 'Washington' 'West Virginia' 'Wisconsin' 'Wyoming']
```

```
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\1081809137.py:3: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  unique_states = df[df['COVID-19 Deaths'] > 0][df['State'] != 'United St
ates']['State'].unique()
```

In [148…
```
#1) menampilan data covid setiap negara bagian us pada tahun 2020
covid_cases = []
for i in unique_states:
    covid_cases.append(df[df['COVID-19 Deaths'] > 0][df['State']==i][df['
                       [df['Age Group'] == 'All Ages'][df['Group'] == 'By

covid_cases = [int(i) for i in covid_cases]
print(covid_cases)
```

```
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\3763128363.py:4: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  covid_cases.append(df[df['COVID-19 Deaths'] > 0][df['State']==i][df['St
ate'] != 'United States'][df['Sex'] == 'All Sexes']
```

```
[6706, 254, 9321, 4027, 33577, 5073, 6298, 1096, 1004, 21827, 10454, 365,
1541, 16721, 9775, 4769, 3510, 4619, 7066, 465, 6729, 10221, 12340, 5834,
5162, 8059, 1269, 2253, 3527, 828, 18164, 2886, 16189, 22282, 8789, 1513,
15097, 5248, 1612, 18407, 1629, 1918, 5705, 1715, 7927, 33542, 1657, 146,
6156, 3703, 1564, 6286, 461]
```

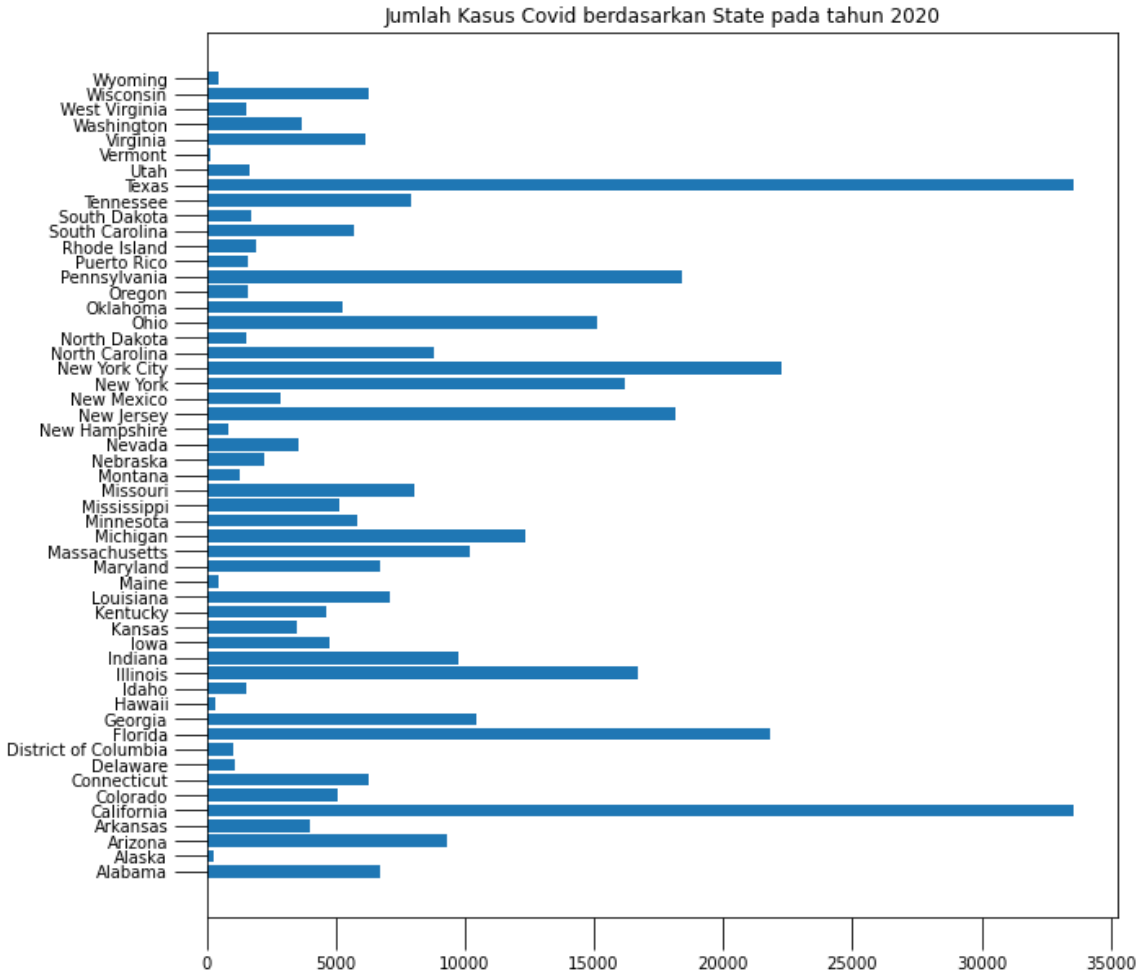
In [149…

```
#1 menampilan data covid setiap negara bagian us pada tahun 2020
print(set(zip(unique_states,covid_cases)))
```

```
{('Nebraska', 2253), ('Hawaii', 365), ('Colorado', 5073), ('Connecticut',
6298), ('New York City', 22282), ('Utah', 1657), ('California', 33577),
('North Carolina', 8789), ('Alaska', 254), ('Indiana', 9775), ('Oklahom
a', 5248), ('Iowa', 4769), ('Oregon', 1612), ('Pennsylvania', 18407), ('V
irginia', 6156), ('Louisiana', 7066), ('Wyoming', 461), ('New Hampshire',
828), ('Alabama', 6706), ('Idaho', 1541), ('Florida', 21827), ('Minnesot
a', 5834), ('South Dakota', 1715), ('Kansas', 3510), ('Puerto Rico', 162
9), ('Montana', 1269), ('Kentucky', 4619), ('Arkansas', 4027), ('New Mexi
co', 2886), ('Maine', 465), ('Mississippi', 5162), ('New York', 16189),
('District of Columbia', 1004), ('New Jersey', 18164), ('South Carolina',
5705), ('Vermont', 146), ('Delaware', 1096), ('Nevada', 3527), ('Arizon
a', 9321), ('Ohio', 15097), ('Wisconsin', 6286), ('Georgia', 10454), ('Te
xas', 33542), ('Rhode Island', 1918), ('West Virginia', 1564), ('Missour
i', 8059), ('Washington', 3703), ('Massachusetts', 10221), ('Maryland', 6
729), ('Illinois', 16721), ('Michigan', 12340), ('North Dakota', 1513),
('Tennessee', 7927)}
```

In [150…

```
#1 menampilan data covid setiap negara bagian us pada tahun 2020
#Menampilkan data negara bagian dalam bentuk grafik batang
plt.figure(figsize=(10,10))
plt.barh(unique_states,covid_cases)
plt.tick_params(size=20,color='black')
plt.title('Jumlah Kasus Covid berdasarkan State pada tahun 2020')
plt.show()
```




In [151…

```
#2 Menampilkan visualisasi kasu covid,pneumonia dan influenza pada tahun
#Memfilter kasus covid,pneumonia dan influenza pada tahun 2020
unique_states = ['Alabama','Indiana','Nevada','Maryland']
covid_cases = []
pneumonia_cases = []
influenza_cases = []

for i in unique_states:
    covid_cases.append(df[df['COVID-19 Deaths'] > 0][df['State']==i][df['
                       [df['Age Group'] == 'All Ages'][df['Group'] == 'By
    pneumonia_cases.append(df[df['Pneumonia Deaths'] > 0][df['State']==i]
                       [df['Age Group'] == 'All Ages'][df['Group'] == 'By
    influenza_cases.append(df[df['Influenza Deaths'] > 0][df['State']==i]
                       [df['Age Group'] == 'All Ages'][df['Group'] == 'By

covid_cases = [int(i) for i in covid_cases]
pneumonia_cases = [int(i) for i in pneumonia_cases]
influenza_cases = [int(i) for i in influenza_cases]

print("Hasil untuk kasus 2020")
#Kasus Covid
print("Jumlah kasus Covid-19")
print(set(zip(unique_states,covid_cases)))

#Kasus Pneumonia
print("Jumlah kasus Pneumonia")
print(set(zip(unique_states,pneumonia_cases)))

#Kasus Influenza
print("Jumlah kasus Influenza")
print(set(zip(unique_states,influenza_cases)))
```

```
print("==================================")

#Memfilter kasus covid,pneumonia dan influenza pada tahun 2021
covid_cases_21 = []
pneumonia_cases_21 = []
influenza_cases_21 = []
for i in unique_states:
    covid_cases_21.append(df[df['COVID-19 Deaths'] > 0][df['State']==i][d
                          [df['Age Group'] == 'All Ages'][df['Group'] == 'By
    pneumonia_cases_21.append(df[df['Pneumonia Deaths'] > 0][df['State']=
                          [df['Age Group'] == 'All Ages'][df['Group'] == 'By
    influenza_cases_21.append(df[df['Influenza Deaths'] > 0][df['State']=
                          [df['Age Group'] == 'All Ages'][df['Group'] == 'By

covid_cases_21 = [int(i) for i in covid_cases_21]
pneumonia_cases_21 = [int(i) for i in pneumonia_cases_21]
influenza_cases_21 = [int(i) for i in influenza_cases_21]

#Hasil untuk kasus 2021
#Kasus Covid 19
print("Hasil untuk kasus 2021")
print("Jumlah kasus Covid-19")
print(set(zip(unique_states,covid_cases_21)))

#Kasus Pneumonia
print("Jumlah kasus Pneumonia")
print(set(zip(unique_states,pneumonia_cases_21)))

#Kasus Influenza
print("Jumlah kasus Influenza")
print(set(zip(unique_states,influenza_cases_21)))


#Memfilter kasus covid,pneumonia,influenza pada tahun 2022
covid_cases_22 = []
pneumonia_cases_22 = []
influenza_cases_22 = []
for i in unique_states:
    covid_cases_22.append(df[df['COVID-19 Deaths'] > 0][df['State']==i][d
                          [df['Age Group'] == 'All Ages'][df['Group'] == 'By
    pneumonia_cases_22.append(df[df['Pneumonia Deaths'] > 0][df['State']=
                          [df['Age Group'] == 'All Ages'][df['Group'] == 'By
    influenza_cases_22.append(df[df['Influenza Deaths'] > 0][df['State']=
                          [df['Age Group'] == 'All Ages'][df['Group'] == 'By

covid_cases_22 = [int(i) for i in covid_cases_22]
pneumonia_cases_22 = [int(i) for i in pneumonia_cases_22]
influenza_cases_22 = [int(i) for i in influenza_cases_22]

#Kasus Covid 19
print("Hasil untuk kasus 2022")
print("Jumlah kasus Covid-19")
print(set(zip(unique_states,covid_cases_22)))

#Kasus Pneumonia
print("Jumlah kasus Pneumonia")
print(set(zip(unique_states,pneumonia_cases_22)))

#Kasus Influenza
print("Jumlah kasus Influenza")
print(set(zip(unique_states,influenza_cases_22)))
```

```
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2812408927.py:9: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  covid_cases.append(df[df['COVID-19 Deaths'] > 0][df['State']==i][df['St
ate'] != 'United States'][df['Sex'] == 'All Sexes']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2812408927.py:11: UserW
arning: Boolean Series key will be reindexed to match DataFrame index.
  pneumonia_cases.append(df[df['Pneumonia Deaths'] > 0][df['State']==i][d
f['State'] != 'United States'][df['Sex'] == 'All Sexes']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2812408927.py:13: UserW
arning: Boolean Series key will be reindexed to match DataFrame index.
  influenza_cases.append(df[df['Influenza Deaths'] > 0][df['State']==i][d
f['State'] != 'United States'][df['Sex'] == 'All Sexes']
Hasil untuk kasus 2020
Jumlah kasus Covid-19
{('Indiana', 9775), ('Nevada', 3527), ('Maryland', 6729), ('Alabama', 670
6)}
Jumlah kasus Pneumonia
{('Nevada', 4354), ('Maryland', 5856), ('Alabama', 5213), ('Indiana', 890
2)}
Jumlah kasus Influenza
{('Indiana', 178), ('Maryland', 151), ('Nevada', 56), ('Alabama', 140)}
==================================
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2812408927.py:41: UserW
arning: Boolean Series key will be reindexed to match DataFrame index.
  covid_cases_21.append(df[df['COVID-19 Deaths'] > 0][df['State']==i][df
['State'] != 'United States'][df['Sex'] == 'All Sexes']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2812408927.py:43: UserW
arning: Boolean Series key will be reindexed to match DataFrame index.
  pneumonia_cases_21.append(df[df['Pneumonia Deaths'] > 0][df['State']==
i][df['State'] != 'United States'][df['Sex'] == 'All Sexes']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2812408927.py:45: UserW
arning: Boolean Series key will be reindexed to match DataFrame index.
  influenza_cases_21.append(df[df['Influenza Deaths'] > 0][df['State']==
i][df['State'] != 'United States'][df['Sex'] == 'All Sexes']
Hasil untuk kasus 2021
Jumlah kasus Covid-19
{('Indiana', 10050), ('Nevada', 5642), ('Maryland', 6091), ('Alabama', 97
15)}
Jumlah kasus Pneumonia
{('Alabama', 6353), ('Maryland', 5693), ('Nevada', 5928), ('Indiana', 949
8)}
Jumlah kasus Influenza
{('Nevada', 0), ('Alabama', 36), ('Maryland', 11), ('Indiana', 23)}
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2812408927.py:72: UserW
arning: Boolean Series key will be reindexed to match DataFrame index.
  covid_cases_22.append(df[df['COVID-19 Deaths'] > 0][df['State']==i][df
['State'] != 'United States'][df['Sex'] == 'All Sexes']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2812408927.py:74: UserW
arning: Boolean Series key will be reindexed to match DataFrame index.
  pneumonia_cases_22.append(df[df['Pneumonia Deaths'] > 0][df['State']==
i][df['State'] != 'United States'][df['Sex'] == 'All Sexes']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2812408927.py:76: UserW
arning: Boolean Series key will be reindexed to match DataFrame index.
  influenza_cases_22.append(df[df['Influenza Deaths'] > 0][df['State']==
i][df['State'] != 'United States'][df['Sex'] == 'All Sexes']
Hasil untuk kasus 2022
Jumlah kasus Covid-19
{('Indiana', 3908), ('Nevada', 1847), ('Maryland', 2668), ('Alabama', 290
9)}
Jumlah kasus Pneumonia
{('Indiana', 3241), ('Alabama', 1838), ('Maryland', 2094), ('Nevada', 187
```

```
7)}
Jumlah kasus Influenza
{('Alabama', 24), ('Nevada', 0), ('Indiana', 47), ('Maryland', 20)}
```

In [152…

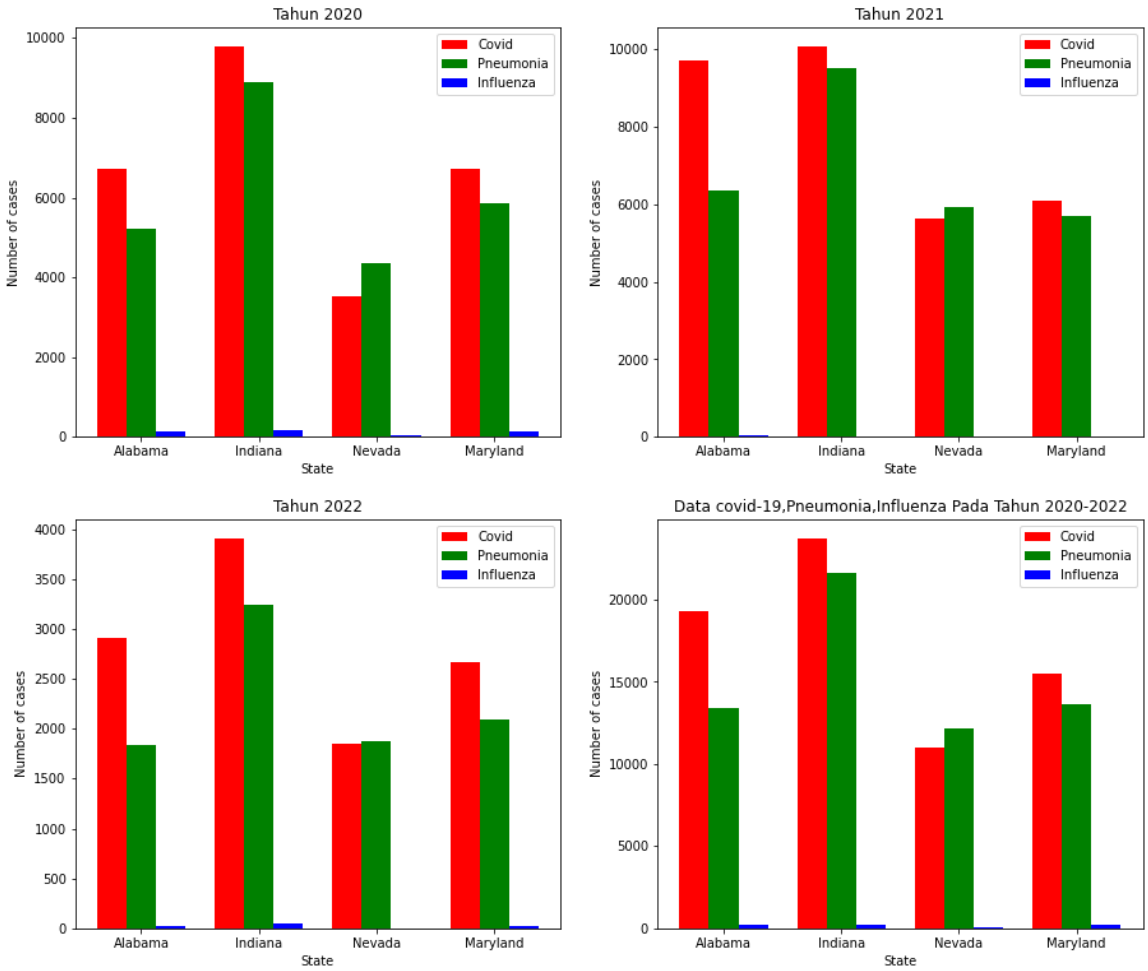
```
#2 Menampilkan visualisasi kasu covid,pneumonia dan influenza pada tahun
#'Alabama','Indiana','Nevada','Maryland'
#memfilter kasus covid,pneumonia dan influenza berdasarkan state untuk 20

covid_cases_total = []
pneumonia_cases_total = []
influenza_cases_total = []
for i in unique_states:
    covid_cases_total.append(df[df['COVID-19 Deaths'] > 0][df['State']==i
                        [df['Age Group'] == 'All Ages'][df['Group'] == 'By
    pneumonia_cases_total.append(df[df['Pneumonia Deaths'] > 0][df['State
                        [df['Age Group'] == 'All Ages'][df['Group'] == 'By
    influenza_cases_total.append(df[df['Influenza Deaths'] > 0][df['State
                        [df['Age Group'] == 'All Ages'][df['Group'] == 'By

covid_cases_total = [int(i) for i in covid_cases_total]
pneumonia_cases_total = [int(i) for i in pneumonia_cases_total]
influenza_cases_total = [int(i) for i in influenza_cases_total]

#Kasus Covid 19
print("Hasil data covid,pneumonia dan influenza dari tahun 2020-2022")
print("Jumlah kasus Covid-19")
print(set(zip(unique_states,covid_cases_total)))

print("\n")

#Kasus Pneumonia
print("Jumlah kasus Pneumonia")
print(set(zip(unique_states,pneumonia_cases_total)))

print("\n")

#Kasus Influenza
print("Jumlah kasus Influenza")
print(set(zip(unique_states,influenza_cases_total)))
```

```
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2269416209.py:9: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  covid_cases_total.append(df[df['COVID-19 Deaths'] > 0][df['State']==i]
[df['State'] != 'United States'][df['Sex'] == 'All Sexes']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2269416209.py:11: UserW
arning: Boolean Series key will be reindexed to match DataFrame index.
  pneumonia_cases_total.append(df[df['Pneumonia Deaths'] > 0][df['State']
==i][df['State'] != 'United States'][df['Sex'] == 'All Sexes']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2269416209.py:13: UserW
arning: Boolean Series key will be reindexed to match DataFrame index.
  influenza_cases_total.append(df[df['Influenza Deaths'] > 0][df['State']
==i][df['State'] != 'United States'][df['Sex'] == 'All Sexes']
Hasil data covid,pneumonia dan influenza dari tahun 2020-2022
Jumlah kasus Covid-19
{('Alabama', 19330), ('Nevada', 11016), ('Indiana', 23733), ('Maryland',
15488)}


Jumlah kasus Pneumonia
{('Nevada', 12159), ('Alabama', 13404), ('Indiana', 21641), ('Maryland',
13643)}
```

```
Jumlah kasus Influenza
{('Maryland', 182), ('Alabama', 200), ('Indiana', 248), ('Nevada', 74)}
```

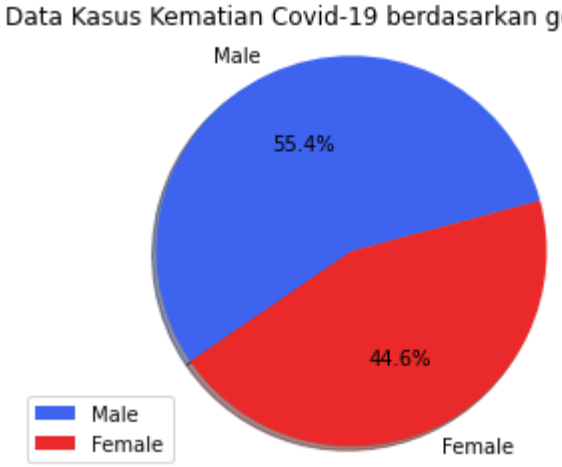
In [186…

```
#2 Menampilkan visualisasi kasu covid,pneumonia dan influenza pada tahun
#N untuk x label
N = 4
ind = np.arange(N)
width = 0.25

#menentukan ukuran baris dan kolom
fig,ax = plt.subplots(nrows=2, ncols=2, figsize=(15,13))

#Untuk tahun 2020
bar1 = ax[0,0].bar(ind,covid_cases,width,color='r')
bar2 = ax[0,0].bar(ind+width,pneumonia_cases,width,color='g')
bar3 = ax[0,0].bar(ind+width*2,influenza_cases,width,color='b')

ax[0,0].set_xlabel('State')
ax[0,0].set_ylabel('Number of cases')
ax[0,0].set_xticks(ind+width,unique_states)
ax[0,0].title.set_text('Tahun 2020')
ax[0,0].legend((bar1,bar2,bar3),('Covid','Pneumonia','Influenza'))

#untuk tahun 2021
bar4 = ax[0,1].bar(ind,covid_cases_21,width,color='r')
bar5 = ax[0,1].bar(ind+width,pneumonia_cases_21,width,color="g")
bar6 = ax[0,1].bar(ind+width*2,influenza_cases_21,width,color='b')

ax[0,1].set_xlabel('State')
ax[0,1].set_ylabel('Number of cases')
ax[0,1].set_xticks(ind+width,unique_states)
ax[0,1].title.set_text('Tahun 2021')
ax[0,1].legend((bar4,bar5,bar6),('Covid','Pneumonia','Influenza'))

#untuk tahun 2022
bar7 = ax[1,0].bar(ind,covid_cases_22,width,color='r')
bar8 = ax[1,0].bar(ind+width,pneumonia_cases_22,width,color='g')
bar9 = ax[1,0].bar(ind+width*2,influenza_cases_22,width,color='b')

ax[1,0].set_xlabel('State')
ax[1,0].set_ylabel('Number of cases')
ax[1,0].set_xticks(ind+width,unique_states)
ax[1,0].title.set_text('Tahun 2022')
ax[1,0].legend((bar7,bar8,bar9),('Covid','Pneumonia','Influenza'))

#Data hasil untuk 2020-2021
bar10 = ax[1,1].bar(ind,covid_cases_total,width,color='r')
bar11 = ax[1,1].bar(ind+width,pneumonia_cases_total,width,color='g')
bar12 = ax[1,1].bar(ind+width*2,influenza_cases_total,width,color='b')

ax[1,1].set_xlabel('State')
ax[1,1].set_ylabel('Number of cases')
ax[1,1].set_xticks(ind+width,unique_states)
ax[1,1].title.set_text('Data covid-19,Pneumonia,Influenza Pada Tahun 2020
ax[1,1].legend((bar10,bar11,bar12),('Covid','Pneumonia','Influenza'))

plt.show()
```




In [154…

```
#3 Menampilkan data gender yang meninggal terkena covid-19 di united Stat
#menampilkan grafik pie chart
gender_labels = ['Male','Female']
total_cases = [553049,445649]

my_colors = ['#3e64ef','#ea2a2a']
my_explode = (0, 0)
plt.pie(total_cases, labels=gender_labels, autopct='%1.1f%%', startangle=
plt.title('Data Kasus Kematian Covid-19 berdasarkan gender di US')
plt.axis('equal')
plt.legend()
plt.show()
```




In [155…

```
#4 menampilkan data gender dan usia yang meninggal karena covid-19
#memfilter data usia tanpa memasukkan all ages
```

```
unique_ages = df[df['COVID-19 Deaths'] > 0][df['State'] != 'United States
```

```
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\1461238252.py:3: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  unique_ages = df[df['COVID-19 Deaths'] > 0][df['State'] != 'United Stat
es'][df['Age Group'] != 'All Ages']['Age Group'].unique()
```

In [156…
```
#4 menampilkan data gender dan usia yang meninggal karena covid-19
print(unique_ages)
```

```
['0-17 years' '15-24 years' '18-29 years' '25-34 years' '30-39 years'
 '35-44 years' '40-49 years' '45-54 years' '50-64 years' '55-64 years'
 '65-74 years' '75-84 years' '85 years and over' '5-14 years'
 'Under 1 year' '1-4 years']
```

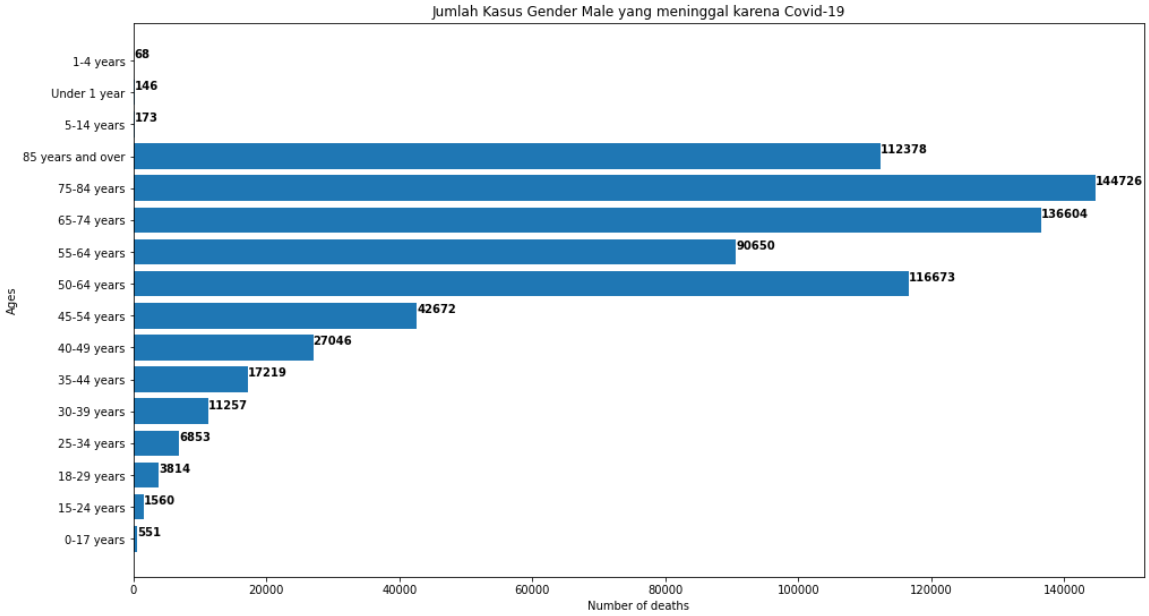
In [157…
```
#4 menampilkan data gender dan usia yang meninggal karena covid-19
#mencari jumlah total male dan female di united states
total_male = []
total_female = []
for i in unique_ages:
    total_male.append(df[df['COVID-19 Deaths'] > 0][df['Age Group']==i][d
                      [df['Age Group'] != 'All Ages'][df['Group'] == 'By
    total_female.append(df[df['COVID-19 Deaths'] > 0][df['Age Group']==i]
                      [df['Age Group'] != 'All Ages'][df['Group'] == 'By

total_male = [int(i) for i in total_male]
total_female = [int(i) for i in total_female]

#Data gender untuk male
print("Male")
print(set(zip(unique_ages,total_male)))

#Data gender untuk female
print("Female")
print(set(zip(unique_ages,total_female)))
```

```
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2692495768.py:6: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  total_male.append(df[df['COVID-19 Deaths'] > 0][df['Age Group']==i][df
['State'] == 'United States'][df['Sex'] == 'Male']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\2692495768.py:8: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  total_female.append(df[df['COVID-19 Deaths'] > 0][df['Age Group']==i][d
f['State'] == 'United States'][df['Sex'] == 'Female']
```

```
Male
{('65-74 years', 136604), ('30-39 years', 11257), ('5-14 years', 173),
('25-34 years', 6853), ('55-64 years', 90650), ('75-84 years', 144726),
('40-49 years', 27046), ('85 years and over', 112378), ('Under 1 year', 1
46), ('1-4 years', 68), ('45-54 years', 42672), ('18-29 years', 3814),
('35-44 years', 17219), ('50-64 years', 116673), ('15-24 years', 1560),
('0-17 years', 551)}
Female
{('55-64 years', 56235), ('30-39 years', 6891), ('65-74 years', 93078),
('18-29 years', 2443), ('35-44 years', 10474), ('85 years and over', 1434
02), ('15-24 years', 1083), ('45-54 years', 23879), ('75-84 years', 11282
7), ('50-64 years', 70599), ('0-17 years', 494), ('5-14 years', 161), ('4
0-49 years', 15915), ('1-4 years', 70), ('Under 1 year', 129), ('25-34 ye
ars', 4311)}
```

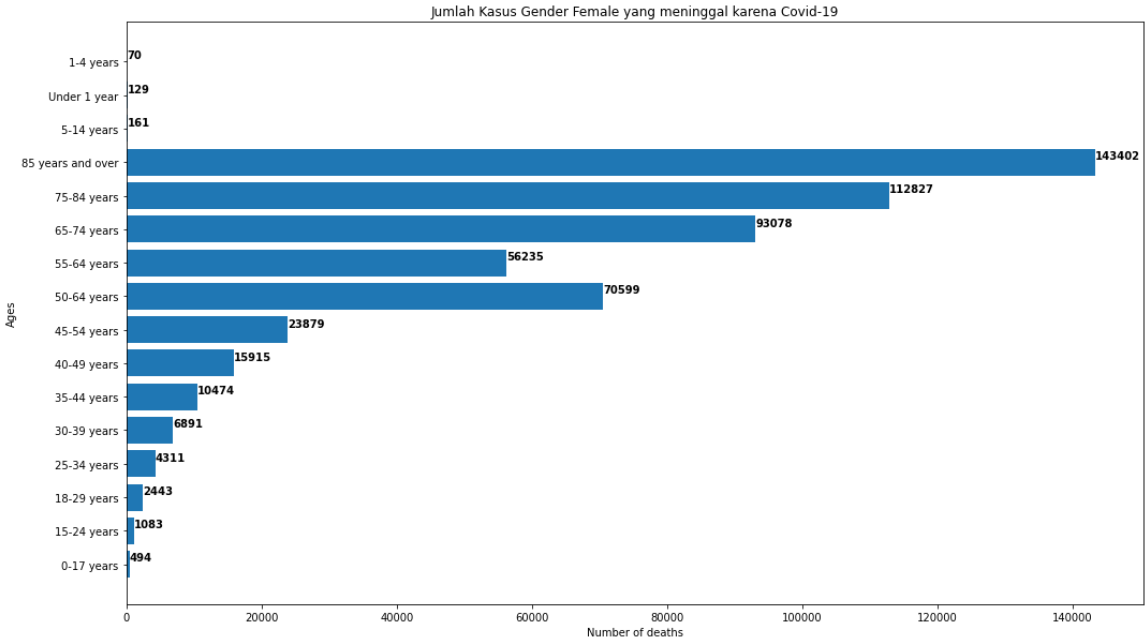
In [158…
```
#4 menampilkan data gender dan usia yang meninggal karena covid-19
#4a menampilkan data gender pria dan usianya yang meninggal karena covid-
fig, ax = plt.subplots(figsize=(16, 9))
```

```
ax.barh(unique_ages,total_male)

for i in ax.patches:
    plt.text(i.get_width()+0.2, i.get_y()+0.5,
             str(round((i.get_width()), 2)),
             fontsize=10, fontweight='bold',
             color='black')

ax.set_xlabel('Number of deaths')
ax.set_ylabel('Ages')

ax.title.set_text('Jumlah Kasus Gender Male yang meninggal karena Covid-1
plt.show()
```




In [159…

```
#4 menampilkan data gender dan usia yang meninggal karena covid-19
#4b menampilkan data gender wanita dan usianya yang meninggal karena covi

fig, ax = plt.subplots(figsize=(17, 10))

ax.barh(unique_ages,total_female)

for i in ax.patches:
    plt.text(i.get_width()+0.2, i.get_y()+0.5,
             str(round((i.get_width()), 2)),
             fontsize=10, fontweight='bold',
             color='black')

ax.set_xlabel('Number of deaths')
ax.set_ylabel('Ages')

ax.title.set_text('Jumlah Kasus Gender Female yang meninggal karena Covid
plt.show()
```




In [160…

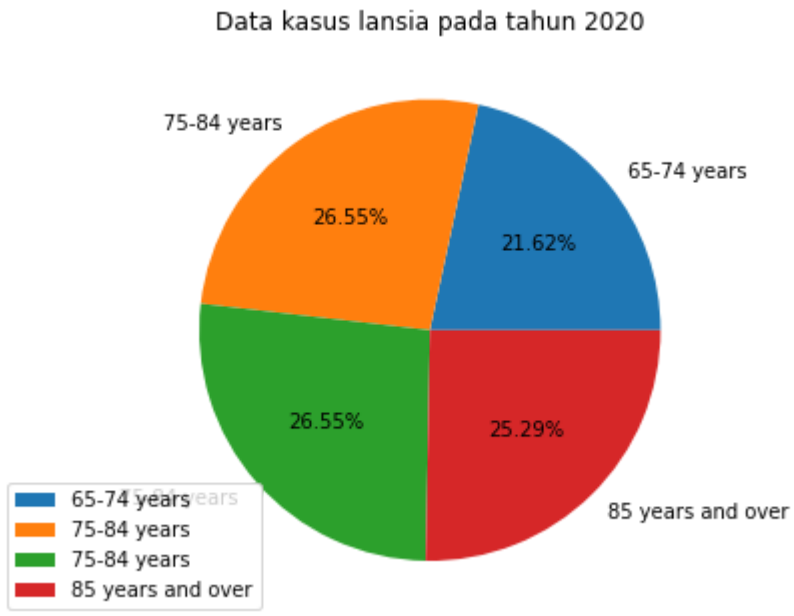
```
#5 Menampilkan data usia lansia yang meninggal karena pneumonia dan covid
count_total_20 = []
count_total_21 = []
count_total_22 = []
unique_ages = ['65-74 years','75-84 years','75-84 years','85 years and ov
for i in unique_ages:
    count_total_20.append(df[df['Pneumonia and COVID-19 Deaths'] > 0][df[
                          [df['Age Group'] != 'All Ages'][df['Group'] == 'By
    count_total_21.append(df[df['Pneumonia and COVID-19 Deaths'] > 0][df[
                          [df['Age Group'] != 'All Ages'][df['Group'] == 'By
    count_total_22.append(df[df['Pneumonia and COVID-19 Deaths'] > 0][df[
                          [df['Age Group'] != 'All Ages'][df['Group'] == 'By

count_total_20 = [int(i) for i in count_total_20]
count_total_21 = [int(i) for i in count_total_21]
count_total_22 = [int(i) for i in count_total_22]

print(count_total_20)
print(count_total_21)
print(count_total_22)
```

```
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\827865417.py:7: UserWar
ning: Boolean Series key will be reindexed to match DataFrame index.
  count_total_20.append(df[df['Pneumonia and COVID-19 Deaths'] > 0][df['A
ge Group']==i][df['State'] == 'United States'][df['Sex'] == 'All Sexes']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\827865417.py:9: UserWar
ning: Boolean Series key will be reindexed to match DataFrame index.
  count_total_21.append(df[df['Pneumonia and COVID-19 Deaths'] > 0][df['A
ge Group']==i][df['State'] == 'United States'][df['Sex'] == 'All Sexes']
C:\Users\yulia\AppData\Local\Temp\ipykernel_25980\827865417.py:11: UserWa
rning: Boolean Series key will be reindexed to match DataFrame index.
  count_total_22.append(df[df['Pneumonia and COVID-19 Deaths'] > 0][df['A
ge Group']==i][df['State'] == 'United States'][df['Sex'] == 'All Sexes']
[42271, 51909, 51909, 49446]
[66608, 62838, 62838, 45248]
[19736, 21132, 21132, 16640]
```

In [189…

```
#5 Menampilkan data usia lansia yang meninggal karena pneumonia dan covid
#Data lansia yang meninggal karena pneumonia dan covid-19 pada tahun 2020
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
```

```
ax.axis('equal')

ax.pie(count_total_20, labels = unique_ages,autopct='%1.2f%%')
plt.title('Data kasus lansia pada tahun 2020')
plt.legend(unique_ages,loc=3)
plt.show()
```




In [190…
```
#5 Menampilkan data usia lansia yang meninggal karena pneumonia dan covid
#Data lansia yang meninggal karena pneumonia dan covid-19 pada tahun 2021
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.axis('equal')

ax.pie(count_total_21, labels = unique_ages,autopct='%1.2f%%')
plt.title('Data kasus lansia pada tahun 2021')
plt.legend(unique_ages,loc=3)
plt.show()
```


In [191…
```
#5 Menampilkan data usia lansia yang meninggal karena pneumonia dan covid
#Data lansia yang meninggal karena pneumonia dan covid-19 pada tahun 2022
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
```

```
ax.axis('equal')

ax.pie(count_total_22, labels = unique_ages,autopct='%1.2f%%')
plt.title('Data kasus lansia pada tahun 2022')
plt.legend(unique_ages,loc=3)
plt.show()
```


In [164…
```
# Melakukan cluster berdasarkan jumlah kasus dan kematian di eropa
df = pd.read_csv('data_eropa.csv')
df.head()
```

Out[164]:

| | dateRep | day | month | year | cases | deaths | countriesAndTerritories | geoId | countryter |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 09/05/2022 | 9 | 5 | 2022 | 4787.0 | 3.0 | Austria | AT | |
| **1** | 08/05/2022 | 8 | 5 | 2022 | 3957.0 | 15.0 | Austria | AT | |
| **2** | 07/05/2022 | 7 | 5 | 2022 | 3521.0 | 3.0 | Austria | AT | |
| **3** | 06/05/2022 | 6 | 5 | 2022 | 5164.0 | 4.0 | Austria | AT | |
| **4** | 05/05/2022 | 5 | 5 | 2022 | 6149.0 | 5.0 | Austria | AT | |

In [165…
```
df.info()
```

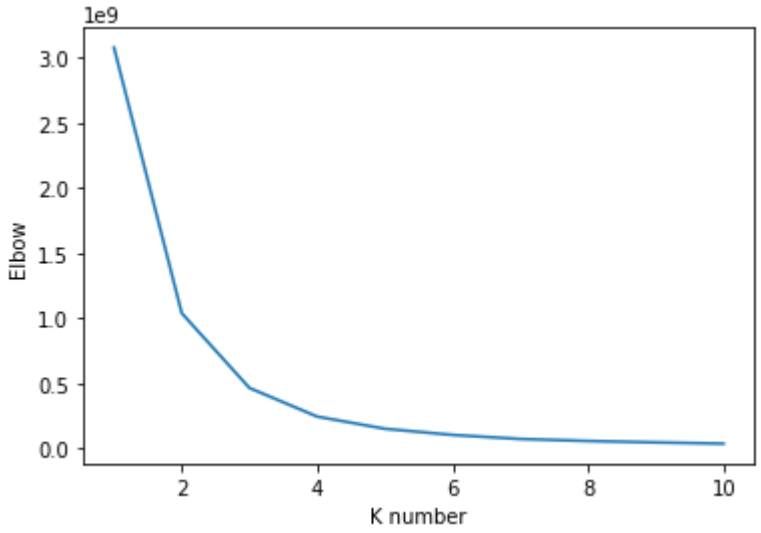
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23427 entries, 0 to 23426
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   dateRep                 23427 non-null  object
 1   day                     23427 non-null  int64
 2   month                   23427 non-null  int64
 3   year                    23427 non-null  int64
 4   cases                   23374 non-null  float64
 5   deaths                  23150 non-null  float64
 6   countriesAndTerritories 23427 non-null  object
 7   geoId                   23427 non-null  object
 8   countryterritoryCode    23427 non-null  object
 9   popData2020             23427 non-null  int64
 10  continentExp            23427 non-null  object
dtypes: float64(2), int64(4), object(5)
memory usage: 2.0+ MB
```

In [166…
```
#mengubah tipe data kolum cases dan deaths menjadi tipe int
df['cases'] = df['cases'].fillna(0)
df['cases'] = df['cases'].astype(dtype='int64')

df['deaths'] = df['deaths'].fillna(0)
df['deaths'] = df['deaths'].astype(dtype='int64')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23427 entries, 0 to 23426
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   dateRep                 23427 non-null  object
 1   day                     23427 non-null  int64 
 2   month                   23427 non-null  int64 
 3   year                    23427 non-null  int64 
 4   cases                   23427 non-null  int64 
 5   deaths                  23427 non-null  int64 
 6   countriesAndTerritories 23427 non-null  object
 7   geoId                   23427 non-null  object
 8   countryterritoryCode    23427 non-null  object
 9   popData2020             23427 non-null  int64 
 10  continentExp            23427 non-null  object
dtypes: int64(6), object(5)
memory usage: 2.0+ MB
```

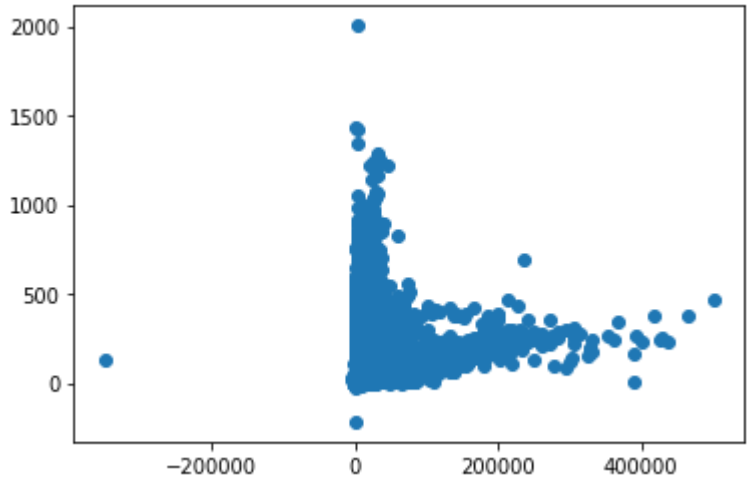
In [182…
```
#mencari nilai k yang optimal dengan menggunakan metode elbow
kluster = []
for i in range(1,11):
    km = KMeans(n_clusters=i)
    km.fit(df[['cases','deaths']])
    kluster.append(km.inertia_)

plt.xlabel('K number')
plt.ylabel('Elbow')
plt.plot(range(1,11),kluster)
plt.show
```

Out[182]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```




In [168…
```
#memvisualisasi data cases dan deaths menggunakan scatter
plt.scatter(df['cases'],df['deaths'])
```

Out[168]:
```
<matplotlib.collections.PathCollection at 0x239de93a800>
```




In [169…
```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['cases','deaths']])
df['cluster'] = y_predicted
y_predicted
```

Out[169]:
```
array([0, 0, 0, ..., 0, 0, 0])
```

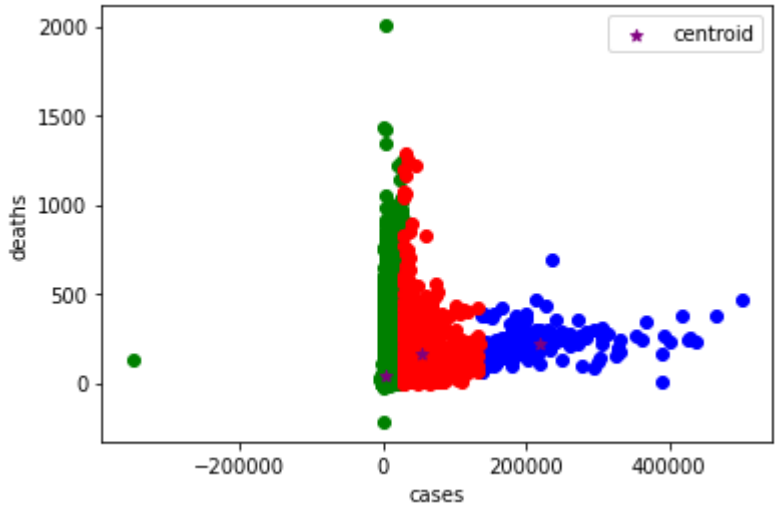
In [170…
```
df.head()
```

Out[170]:

| | dateRep | day | month | year | cases | deaths | countriesAndTerritories | geoId | countryter |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 09/05/2022 | 9 | 5 | 2022 | 4787 | 3 | Austria | AT | |
| **1** | 08/05/2022 | 8 | 5 | 2022 | 3957 | 15 | Austria | AT | |
| **2** | 07/05/2022 | 7 | 5 | 2022 | 3521 | 3 | Austria | AT | |
| **3** | 06/05/2022 | 6 | 5 | 2022 | 5164 | 4 | Austria | AT | |
| **4** | 05/05/2022 | 5 | 5 | 2022 | 6149 | 5 | Austria | AT | |

In [171…
```
#menampilan visualisasi cluster beserta centroidnya
df1 = df[df.cluster ==0]
df2 = df[df.cluster ==1]
df3 = df[df.cluster ==2]

plt.scatter(df1.cases,df1['deaths'],color="green")
plt.scatter(df2.cases,df2['deaths'],color="blue")
plt.scatter(df3.cases,df3['deaths'],color="red")

plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purp

plt.xlabel('cases')
plt.ylabel('deaths')
plt.legend()
```

Out[171]:
```
<matplotlib.legend.Legend at 0x239de897b50>
```




In [172…
```
df.sort_values(by=['cluster'])
```

Out[172]:

| | dateRep | day | month | year | cases | deaths | countriesAndTerritories | geoId | cour |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 09/05/2022 | 9 | 5 | 2022 | 4787 | 3 | Austria | AT | |
| **15454** | 30/01/2022 | 30 | 1 | 2022 | 191 | 2 | Malta | MT | |
| **15453** | 31/01/2022 | 31 | 1 | 2022 | 169 | 3 | Malta | MT | |
| **15452** | 01/02/2022 | 1 | 2 | 2022 | 249 | 1 | Malta | MT | |
| **15451** | 02/02/2022 | 2 | 2 | 2022 | 227 | 5 | Malta | MT | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **21873** | 02/01/2022 | 2 | 1 | 2022 | 110483 | 102 | Spain | ES | |
| **21871** | 04/01/2022 | 4 | 1 | 2022 | 130285 | 139 | Spain | ES | |
| **21869** | 06/01/2022 | 6 | 1 | 2022 | 82859 | 115 | Spain | ES | |
| **21883** | 23/12/2021 | 23 | 12 | 2021 | 80122 | 82 | Spain | ES | |
| **21855** | 20/01/2022 | 20 | 1 | 2022 | 109537 | 207 | Spain | ES | |

23427 rows × 12 columns

In [183…
```
#memprediksi kasus covid dengan linear regression
dt = pd.read_csv('data_eropa.csv')
dt.head()
```

Out[183]:

| | dateRep | day | month | year | cases | deaths | countriesAndTerritories | geoId | countryter |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 09/05/2022 | 9 | 5 | 2022 | 4787.0 | 3.0 | Austria | AT | |
| **1** | 08/05/2022 | 8 | 5 | 2022 | 3957.0 | 15.0 | Austria | AT | |
| **2** | 07/05/2022 | 7 | 5 | 2022 | 3521.0 | 3.0 | Austria | AT | |
| **3** | 06/05/2022 | 6 | 5 | 2022 | 5164.0 | 4.0 | Austria | AT | |
| **4** | 05/05/2022 | 5 | 5 | 2022 | 6149.0 | 5.0 | Austria | AT | |

In [184…
```
#mengkelompokkan data berdasarkan wilayah
grouped = dt.groupby('countriesAndTerritories')
```

In [175…
```
#memfilter data berdasarkan wilyah yang sudah ditentukan
df = pd.DataFrame()
```

```
for name,group in grouped:
    if name == 'Bulgaria':
        df = group
```

In [176…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 796 entries, 1604 to 2399
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   dateRep                 796 non-null    object 
 1   day                     796 non-null    int64  
 2   month                   796 non-null    int64  
 3   year                    796 non-null    int64  
 4   cases                   795 non-null    float64
 5   deaths                  793 non-null    float64
 6   countriesAndTerritories  796 non-null    object 
 7   geoId                   796 non-null    object 
 8   countryterritoryCode    796 non-null    object 
 9   popData2020             796 non-null    int64  
 10  continentExp            796 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 74.6+ KB
```

In [177…
```
#menukar tipe data float pada cases dan deaths menjadi int
df['cases'] = df['cases'].fillna(0)
df['cases'] = df['cases'].astype(dtype='int64')

df['deaths'] = df['deaths'].fillna(0)
df['deaths'] = df['deaths'].astype(dtype='int64')

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 796 entries, 1604 to 2399
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   dateRep                 796 non-null    object 
 1   day                     796 non-null    int64  
 2   month                   796 non-null    int64  
 3   year                    796 non-null    int64  
 4   cases                   796 non-null    int64  
 5   deaths                  796 non-null    int64  
 6   countriesAndTerritories  796 non-null    object 
 7   geoId                   796 non-null    object 
 8   countryterritoryCode    796 non-null    object 
 9   popData2020             796 non-null    int64  
 10  continentExp            796 non-null    object 
dtypes: int64(6), object(5)
memory usage: 74.6+ KB
```

In [178…
```
df = df[['day','month','year','cases','deaths']]
df = df.iloc[::-1]
xFrame = df.drop('deaths',axis = 1)
yFrame = df['deaths']
```

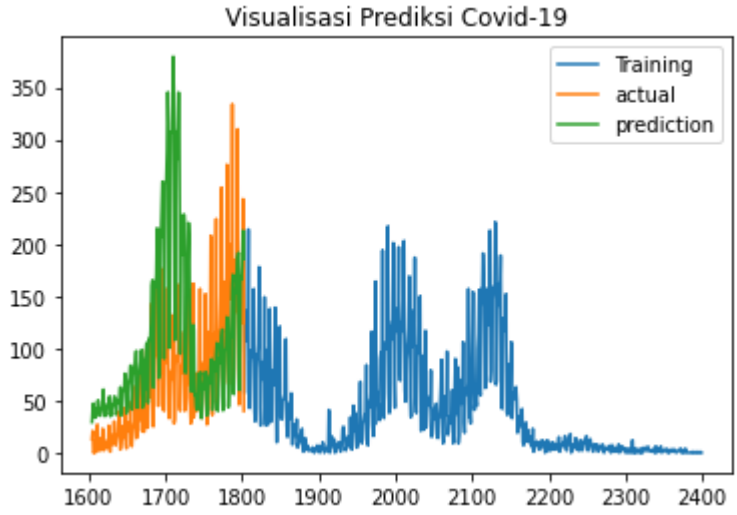
In [179…
```
#membagi dara menjadi data latih dan data uji
xtrain,xtest,ytrain,ytest = train_test_split(xFrame.values,yFrame.values,
```

In [180…
```
#memprediksi
```

```
model = LinearRegression()
model.fit(xtrain,ytrain)
prediction = model.predict(xtest)
```

In [193…

```
plt.plot(df.index[:-len(prediction)],ytrain,label="Training")
plt.plot(df.index[-len(prediction):],ytest,label="actual")
plt.plot(df.index[-len(prediction):],prediction,label="prediction")
plt.title("Visualisasi Prediksi Covid-19")
plt.legend(loc=1)
plt.show()
```




In [ ]: