

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

```
In [2]: df = pd.read_csv("Penyebab Kematian di Indonesia yang Dilaporkan - Clean.csv")
df.head(50)
```

```
Out[2]:
```

	Cause	Type	Year	Data Redundancy	Total Deaths	Source	Page at Source	Source URL
0	Abrasi Air Laut	Bencana Alam	2012	1	0	Profil Kesehatan Indonesia Tahun 2012	Lampiran 3.38	https://pusdatin.kemkes.go.id/download.php?fil...
1	AIDS	Bencana Non Alam dan Penyakit	2000	1	47	Profil Kesehatan Indonesia Tahun 2004	78	https://pusdatin.kemkes.go.id/download.php?fil...
2	AIDS	Bencana Non Alam dan Penyakit	2001	1	99	Profil Kesehatan Indonesia Tahun 2004	78	https://pusdatin.kemkes.go.id/download.php?fil...
3	AIDS	Bencana Non Alam dan Penyakit	2002	1	100	Profil Kesehatan Indonesia Tahun 2004	78	https://pusdatin.kemkes.go.id/download.php?fil...
4	AIDS	Bencana Non Alam dan Penyakit	2003	1	261	Profil Kesehatan Indonesia Tahun 2010	115	https://pusdatin.kemkes.go.id/download.php?fil...
		Bencana Non				Profil Kesehatan		

```
In [3]: to_drop = ['Source URL', 'Source', 'Cause', 'Page at Source']
```

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

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 680 entries, 0 to 679
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  ---
0    Type            680 non-null   object
1    Year            680 non-null   int64
2    Data Redundancy 680 non-null   int64
3    Total Deaths   680 non-null   int64
dtypes: int64(3), object(1)
memory usage: 21.4+ KB
```

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

```
Out[6]:
```

	Year	Data Redundancy	Total Deaths
0	2012	1	0
1	2000	1	47
2	2001	1	99
3	2002	1	100
4	2003	1	261
5	2004	1	361
6	2005	1	509
7	2006	1	635
8	2007	1	788
9	2008	1	711
10	2009	1	331

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

```
Out[7]: 0      Bencana Alam
1  Bencana Non Alam dan Penyakit
2  Bencana Non Alam dan Penyakit
3  Bencana Non Alam dan Penyakit
4  Bencana Non Alam dan Penyakit
5  Bencana Non Alam dan Penyakit
6  Bencana Non Alam dan Penyakit
7  Bencana Non Alam dan Penyakit
8  Bencana Non Alam dan Penyakit
9  Bencana Non Alam dan Penyakit
10 Bencana Non Alam dan Penyakit
Name: Type, 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]:
```

	Type	Year	Data Redundancy	Total Deaths
0	Bencana Alam	2012	1	0
1	Bencana Non Alam dan Penyakit	2000	1	47
2	Bencana Non Alam dan Penyakit	2001	1	99
3	Bencana Non Alam dan Penyakit	2002	1	100
4	Bencana Non Alam dan Penyakit	2003	1	261
5	Bencana Non Alam dan Penyakit	2004	1	361
6	Bencana Non Alam dan Penyakit	2005	1	509
7	Bencana Non Alam dan Penyakit	2006	1	635
8	Bencana Non Alam dan Penyakit	2007	1	788
9	Bencana Non Alam dan Penyakit	2008	1	711
10	Bencana Non Alam dan Penyakit	2009	1	331

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

```
Out[10]:
```

	Year	Data Redundancy	Total Deaths
0	2012	1	0
1	2000	1	47
2	2001	1	99
3	2002	1	100
4	2003	1	261
5	2004	1	361
6	2005	1	509
7	2006	1	635
8	2007	1	788
9	2008	1	711
10	2009	1	331

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

```
Out[11]: 0      Bencana Alam
1  Bencana Non Alam dan Penyakit
2  Bencana Non Alam dan Penyakit
3  Bencana Non Alam dan Penyakit
4  Bencana Non Alam dan Penyakit
5  Bencana Non Alam dan Penyakit
6  Bencana Non Alam dan Penyakit
7  Bencana Non Alam dan Penyakit
8  Bencana Non Alam dan Penyakit
9  Bencana Non Alam dan Penyakit
10 Bencana Non Alam dan Penyakit
Name: Type, dtype: object
```

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

Prediksi Naive Bayes :	['Bencana Sosial'	'Bencana Non Alam dan Penyakit'		
'Bencana Non Alam dan Penyakit'	'Bencana Non Alam dan Penyakit'			
'Bencana Non Alam dan Penyakit'	'Bencana Non Alam dan Penyakit'			
'Bencana Non Alam dan Penyakit'	'Bencana Non Alam dan Penyakit'			
'Bencana Non Alam dan Penyakit'	'Bencana Non Alam dan Penyakit'			
'Bencana Non Alam dan Penyakit'	'Bencana Non Alam dan Penyakit'			
'Bencana Non Alam dan Penyakit'	'Bencana Non Alam dan Penyakit'			
'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	
'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	
'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	
'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	
'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	
'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'		
'Bencana Non Alam dan Penyakit'	'Bencana Non Alam dan Penyakit'			
'Bencana Non Alam dan Penyakit'	'Bencana Non Alam dan Penyakit'			
'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	
'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	
'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	'Bencana Sosial'	

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

Akurasi Naive Bayes : 0.4661764705882353

```
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
Bencana Alam	0.17	0.01	0.03	148
Bencana Non Alam dan Penyakit	0.88	0.58	0.70	512
Bencana Sosial	0.06	0.95	0.11	20
accuracy			0.47	680
macro avg	0.37	0.51	0.28	680
weighted avg	0.70	0.47	0.53	680

In [ ]: