

Code dilakukan pada google collab

```
import pandas as pd

real = pd.read_csv("realnews.csv")
fake = pd.read_csv("fakenews.csv")

real['News Category'] = "RealNews"
fake['News Category'] = "FakeNews"
Table = pd.concat([real, fake])

import seaborn as sea
sea.countplot(Table['News Category']);

Table.isnull().sum()

import matplotlib.pyplot as plt
data['isu'].value_counts()
plt.figure(figsize = (10,10))
sea.countplot(data['isu']);

plt.figure(figsize = (10,10))
chart = sea.countplot(x = "News Category", hue = "isu", data = Table , palette = 'muted')
sea.set_style("darkgrid")
chart.set_xticklabels(chart.get_xticklabels(), rotation=90)

Table['isu'] = Table['berita_jakarta'] + " " + Table['isu']
Table = Table.drop(['berita_jakarta', 'link/page', 'tanggal'], axis=1)

from wordcloud import WordCloud

import nltk
nltk.download('stopwords')

wordcloud = WordCloud(width = 800, height = 800,
                      background_color = 'white',
                      stopwords = stopwords.words('english'))
```

```
        , min_font_size = 10).generate(" ".join(Table[Table['News  
Category']] == "FakeNews"].isu))
```

```
plat.figure(figsize = (8, 8), facecolor = None)  
plat.imshow(wordcloud)  
plat.axis("off")  
plat.tight_layout(pad = 0)  
plat.show()
```

```
wordcloud = WordCloud(width = 800, height = 800,  
                        background_color = 'white',  
                        stopwords = stopwords.words('english')  
                        , min_font_size = 10).generate(" ".join(Table[Table['News  
Category']] == "RealNews"].isu))
```

```
plat.figure(figsize = (8, 8), facecolor = None)  
plat.imshow(wordcloud)  
plat.axis("off")  
plat.tight_layout(pad = 0)  
plat.show()
```

```
import sklearn  
from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test = train_test_split(data['isu'],data['News Ca  
tegory'],test_size=0.2, random_state = 1)
```

```
from sklearn.feature_extraction.text import TfidfTransformer  
from sklearn.feature_extraction.text import CountVectorizer  
from sklearn.feature_extraction.text import TfidfVectorizer  
from sklearn.pipeline import Pipeline  
from sklearn.naive_bayes import MultinomialNB  
from sklearn.metrics import accuracy_score  
import sklearn.metrics as metrics
```

```
from mlxtend.plotting import plot_confusion_matrix  
from sklearn.metrics import confusion_matrix
```

```
pipe = Pipeline([  
    ('vect', CountVectorizer()),  
    ('tfidf', TfidfTransformer()),
```

```

        ('clf', MultinomialNB())
    ])

model = pipe.fit(x_train, y_train)
prediction = model.predict(x_test)

score = metrics.accuracy_score(y_test, prediction)
print("accuracy:    %0.3f" % (score*100))

fig, ax = plot_confusion_matrix(conf_mat=confusion_matrix(y_test, prediction),
                                show_absolute=True,
                                show_normed=True,
                                colorbar=True)

plt.show()

```

link collab :

https://colab.research.google.com/drive/1oN8MrtVyBYI-kizpq5N7N_X8YqzBC8mr?usp=sharing