NIM: 2540120074

Nama: Lionel Riyadi

Link Video Penjelasan: https://binusianorg-my.sharepoint.com/personal/lionel_riyadi_binus_ac_id/_layouts/15/guestaccess.aspx?share=ET92mgsv0ahNoyRUjxBhI7oBr3oobkG-vmnFA8dur66iGg&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcmI2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0RpcmVjdCJ9fQ&e=8tVTE0

```
from google.colab import drive
import pandas as pd
import re

drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)
```

```
ROOT = '/content/drive/MyDrive/Odd2324/TextMining/UTS/'
```

```
dataset = pd.read_csv(ROOT+'dataset.csv',encoding='latin1')
print(dataset)
```

```
                                                Teks       Media      Label
0  nasional/read/Pelatih Chelsea, Mauricio Pochet...      Kompas       Bola
1  Laporan Wartawan Tribunnews.com, Aisyah Nursya...  Tribunnews  Kesehatan
2  Laporan Wartawan Tribunnews.com, Aisyah Nursya...  Tribunnews  Kesehatan
3  Laporan Wartawan Tribunnews.com, Aisyah Nursya...  Tribunnews  Kesehatan
4  TEMPO.CO, Jakarta - Copa Interamericana akan k...       Tempo       Bola
5  JAKARTA, KOMPAS.com - Kepala Staf Angkatan Dar...      Kompas    Politik
6  JAKARTA, KOMPAS.com - Pengadilan Tindak Pidana...      Kompas    Politik
7  TEMPO.CO, Jakarta - Hasil sigi lembaga survei ...       Tempo    Politik
8  TEMPO.CO, Jakarta - Kepala Badan Pengawas Obat...       Tempo    Politik
9  KOMPAS.com - Pemain Panama U17, Martin Krug, t...      Kompas       Bola
```

Preprocessing Data tahap ini, akan membersihkan teks dari huruf yang tidak sesuai seperti simbol, https://, .com, spasi yang berlebih ataupun \n dan \t.

```
def remove_special_characters(text):
    return re.sub(r'[^a-zA-Z0-9\s]', '', text)

def convert_to_lowercase(text):
    return text.lower()
def remove_https(text):
    return text.replace('https://','')
def remove_com(text):
    return text.replace('.com','')
def remove_extra_space(text):
    return re.sub(r'\s+', ' ', text)
def remove_enter_and_tab(text):
    cleaned_text = text.replace('\n', ' ').replace('\t', ' ')
    return cleaned_text
```

STOPWORDS Indonesia

kemudian tak hanya simbol dan huruf tetapi juga menghilangkan dari huruf stopwords indonesia

```
import nltk
from nltk.tokenize import word_tokenize
import string
nltk.download('punkt')

nltk.download('stopwords')
from nltk.corpus import stopwords

stop_words = set(stopwords.words('indonesian'))
print(stop_words)
# Buang Stopwordsnya
array_of_tokenized = []
for i,teks in enumerate(dataset['Teks']):
    final = remove_com(teks)
    final = remove_https(final)
```

```
    final = remove_enter_and_tab(final)
    final = remove_special_characters(final)
    final = convert_to_lowercase(final)
    final = remove_extra_space(final)

    tokenized = word_tokenize(final)
    filtered_words = [word for word in tokenized if word.lower() not in stop_words]
    dataset.iat[i, 0] = final
    array_of_tokenized.append(filtered_words)
```

```
{'akulah', 'diperlukannya', 'dibuat', 'dekat', 'biasanya', 'berakhir', 'ikut', 'bagaimanakah', 'ditegaskan', 'sebaliknya', 'be
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
for i in array_of_tokenized:
  print(i)
```

```
mi', 'penurunan', '82', 'persen', '405', 'persen', 'april', '323', 'persen', 'september', '2023', 'survei', 'prc', '7', '12', '
rasi', 'diterbitkan', 'pt', 'gramedia', '2013sharisya', 'kusuma', 'rahmanda', 'i', 'naomy', 'ayu', 'nugraheni', 'pilihan', 'edi
```

Kemudian dibersihkan datanya kita dapat menjadikan data tersebut menjadi vectorization dengan 50 feature dan frekuensinya 3

dan kita melakukan train test slpit 0.4 karena minimnya dataset

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
tfidf_vectorizer = TfidfVectorizer(max_features=50, min_df=3)
documents = [' '.join(tokens) for tokens in array_of_tokenized]
```

```
tfidf_matrix = tfidf_vectorizer.fit_transform(documents)
print(tfidf_matrix)

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(tfidf_matrix, dataset['Label'], test_size=0.4, random_state=122)
```

```
  (0, 4)        0.30442495712972306
  (0, 11)       0.6088499142594461
  (0, 28)       0.27065560176294645
  (0, 14)       0.6088499142594461
  (0, 30)       0.30442495712972306
  (1, 36)       0.2363469155086841
  (1, 24)       0.18870802894256158
  (1, 27)       0.2363469155086841
  (1, 17)       0.21012934433808766
  (1, 8)        0.21012934433808766
  (1, 0)        0.2363469155086841
  (1, 21)       0.1705965648399034
  (1, 9)        0.5661240868276848
  (1, 10)       0.2821382847528781
  (1, 22)       0.2363469155086841
  (1, 2)        0.2363469155086841
  (1, 35)       0.2363469155086841
  (1, 37)       0.2363469155086841
  (1, 13)       0.2363469155086841
  (2, 16)       0.3186077388961825
  (2, 33)       0.15930386944809125
  (2, 29)       0.14163255554066148
  (2, 3)        0.15930386944809125
  (2, 5)        0.7081627777033073
  (2, 18)       0.15930386944809125
  :     :
  (8, 3)        0.12548980901813128
  (8, 5)        0.22313886545409886
  (8, 18)       0.12548980901813128
  (8, 8)        0.11156943272704943
  (8, 0)        0.12548980901813128
  (8, 21)       0.18115853380047287
  (8, 9)        0.3005869714172758
  (8, 10)       0.07490150526005258
  (8, 4)        0.25097961803626256
```

```
(9, 7)        0.12072476842624233
(9, 12)       0.12072476842624233
(9, 6)        0.24144953685248466
(9, 31)       0.10733297018146055
(9, 34)       0.4828990737049693
(9, 1)        0.5366648509073026
(9, 15)       0.12072476842624233
(9, 18)       0.24144953685248466
(9, 24)       0.3855642972150624
(9, 0)        0.12072476842624233
(9, 21)       0.0871398331570485
(9, 9)        0.1927821486075312
(9, 11)       0.12072476842624233
(9, 28)       0.10733297018146055
(9, 14)       0.12072476842624233
(9, 30)       0.24144953685248466
```

Saat sudah dipecah menjadi train test. maka kita langsung ambil model SVM dari library sklearn.svm dan langsung train, test dan prediksi hasil tersebut

```
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
svm_classifier = SVC(kernel='linear')
svm_classifier.fit(X_train, y_train)
y_pred = svm_classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.75
```

```
print(y_test)
print("=======")
print(y_pred)
```

```
7      Politik
8      Politik
9         Bola
1    Kesehatan
Name: Label, dtype: object
=======
['Politik' 'Politik' 'Bola' 'Politik']
```

Random Forest

Untuk model Random Forest, sama juga akan ambil dari sklearn.ensemble dan lakukan hal yang sama yaitu train model dan test hasil model tersebut untuk mengecek akurasinya

```
#Random Forest
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.5
```

1. SVM ini lebih kompatible atau bekerja lebih baik pada dataset yang non-linear kompleks. makanya RandomForest memiliki akurasi yang lebih kecil darip ada SVM
2. Dataset yang dimiliki cukup sedikit sehingga sulit untuk mengidentifikasinya