- Data bentuk teks
- Ekstraksi fitur menggunakan Bag of Word atau TFIDF, dan teknik lainnya
- Lakukan Klasifikasi

SPAM DETECTOR

```
from google.colab import drive
import os
```

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

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

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/dataset_sms_spam _v1.csv")
df.head()
```

	Teks	label
0	[PROMO] Beli paket Flash mulai 1GB di MY TELKO...	2
1	2.5 GB/30 hari hanya Rp 35 Ribu Spesial buat A...	2
2	2016-07-08 11:47:11.Plg Yth, sisa kuota Flash ...	2
3	2016-08-07 11:29:47.Plg Yth, sisa kuota Flash ...	2
4	4.5GB/30 hari hanya Rp 55 Ribu Spesial buat an...	2

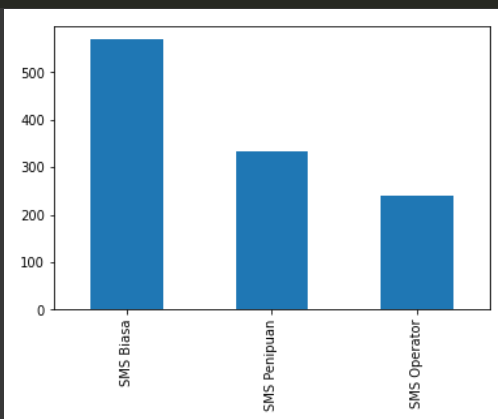
```
df.label = df.label.map({0:"SMS Biasa", 1:"SMS Penipuan", 2:"SMS Operator"})
```

```
# lihat ukuran
df.shape
```

```
(1143, 2)
```

Pembagian Data

```
# lihat distribusi kelas
df.label.value_counts().plot(kind='bar');
```




```
# Split data menjadi data train dan test
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df.Teks, df.label, test_size=0.2)
```

```
X_test.shape
```

```
(229,)
```

Vektorisasi

```
# Inisiasi vectorizer
from sklearn.feature_extraction.text import CountVectorizer

vect = CountVectorizer()
```

```
# Pelajari vocabulary dan ubah data train menjadi matriks
vect.fit(X_train)
X_train_vec = vect.transform(X_train)
```

```
# lihat fitur vektor
X_train_vec
```

```
<914x4301 sparse matrix of type '<class 'numpy.int64'>'
        with 15210 stored elements in Compressed Sparse Row format>
```

```
# lakukan hal yang sama dengan data testing
X_test_vec = vect.transform(X_test)
```

Klasifikasi

Multinomial Naive Bayes

```
# import
from sklearn.naive_bayes import MultinomialNB
```

```
# train dengan melihat waktu eksekusi
nb = MultinomialNB()
%timeit nb.fit(X_train_vec, y_train)
```

```
100 loops, best of 5: 3.09 ms per loop
```

```
# buat prediksi
y_pred = nb.predict(X_test_vec)
```

```
# tampilkan hasil evaluasi model
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
                  precision    recall  f1-score   support

      SMS Biasa       0.98      0.95      0.97       108
   SMS Operator       0.88      0.98      0.93        53
   SMS Penipuan       0.94      0.90      0.92        68

       accuracy                           0.94       229
      macro avg       0.93      0.94      0.94       229
   weighted avg       0.95      0.94      0.94       229
```

Evaluasi/Prediksi

```
# coba sms baru
new_sms = ["Hai bro, apa kabar ?",
           "Dapatkan kuota harian hanya 1000 rupiah per gb, kunjungi aplikasi mygsm",
           "pesugihan halal, lipatgandakan uang anda sekarang bersama ki ....",
           "besok futsal ya"]
```

```
new_sms_vect = vect.transform(new_sms)
```

```
hasil = nb.predict(new_sms_vect)
print(hasil)
```

```
['SMS Biasa' 'SMS Operator' 'SMS Penipuan' 'SMS Biasa']
```