**LAMPIRAN 1**

**Listing Program**

"""skripsi.ipynb

Automatically generated by Colaboratory.

Original file is located at https://colab.research.google.com/drive/1nImoRyOyBl2SEvzwWrm81MpcBECp\_BAt

"""

from google.colab import drive

drive.mount('/content/gdrive')

import pandas as pd

from tqdm import tqdm\_notebook as tqdm

"""# \*\*1. Baca Dataset\*\*"""

dftweet = pd.read\_csv("gdrive/My Drive/skripsi2019/prepro/mrtsemua.csv", encoding = "latin-1", usecols=["Text"])

tweets = pd.read\_csv("gdrive/My Drive/skripsi2019/prepro/mrtsemua.csv", encoding = "latin-1", usecols=["Text"])

tweets.columns = ["tweet"]

dftweet.columns = ["tweet"]

print(dftweet.columns)

dftweet.info()

pd.set\_option('max\_colwidth', 0)

dftweet.head(10)

import re

import string

"""# \*\*2. Casefolding\*\*"""

def casefolding(tweet):

tweet = tweet.lower().strip() #case folding

return tweet

hasilcf = []

for tweet in tqdm(dftweet['tweet']):

pro = casefolding(tweet)

hasilcf.append(pro)

dftweet = pd.DataFrame({"tweet": hasilcf})

dftweet.to\_csv(r'gdrive/My Drive/skripsi2019/prepro/casefolding.csv')

dftweet.rename(columns={"tweet": "Hasil Case Folding"}).head(10)

"""# \*\*3. Cleaning\*\*"""

dftweet = pd.read\_csv("gdrive/My Drive/skripsi2019/prepro/casefolding.csv", encoding = "latin-1", usecols=["tweet"])

def clean(tweet):

tweet = ''.join(re.sub("(@[A-Za-z0-9]+)|(#[A-Za-z0-9]+)|(\w+:\/\/\S+)|(http\S+)", "", tweet)) #hapus #,@,url

tweet = re.sub(r'[^A-Za-z\s\/]' ,' ', tweet) #hapus simbol dan tanda baca

tweet = re.sub(r'\_', '', tweet)

tweet = re.sub(r'/', ' ', tweet)

tweet = re.sub(r'\d+', '', tweet)

tweet = re.sub(r'\n', ' ', tweet)

tweet = re.sub(r'\s{2,}', ' ', tweet)

return tweet

hasilclean = []

for tweet in dftweet['tweet']:

pro = clean(tweet)

hasilclean.append(pro)

dftweet['tweet'] = hasilclean

dftweet.to\_csv(r'gdrive/My Drive/skripsi2019/prepro/clean.csv')

dftweet.rename(columns={"tweet": "Hasil Cleaning"}).head(10)

"""# \*\*4. Negation Handling\*\*"""

dftweet = pd.read\_csv("gdrive/My Drive/skripsi2019/prepro/clean.csv", encoding = "latin-1", usecols=["tweet"])

def neghandling(tweet):

tweet = re.sub(r'\stidak\s' ,' tidak\_', tweet)

tweet = re.sub(r'\stdk\s' ,' tidak\_', tweet)

tweet = re.sub(r'\sga\s', ' tidak\_', tweet)

tweet = re.sub(r'\sgak\s', ' tidak\_', tweet)

tweet = re.sub(r'\senggak\s', ' tidak\_', tweet)

tweet = re.sub(r'\snggak\s' ,' tidak\_', tweet)

tweet = re.sub(r'\sengga\s' ,' tidak\_', tweet)

tweet = re.sub(r'\sngga\s' ,' tidak\_', tweet)

tweet = re.sub(r'\sbelum\s' ,' belum\_', tweet)

return tweet

hasilnh = []

for tweet in dftweet['tweet']:

pro = neghandling(tweet)

hasilnh.append(pro)

dftweet['tweet'] = hasilnh

dftweet.to\_csv(r'gdrive/My Drive/skripsi2019/prepro/neghandling.csv')

dftweet.rename(columns={"tweet": "Hasil Negation Handling"}).head(10)

"""# \*\*5. Stopword Removal\*\*

## Stopword Bahasa Indonesia

"""

dftweet = pd.read\_csv("gdrive/My Drive/skripsi2019/prepro/clean.csv", encoding = "latin-1", usecols=["tweet"])

def cstopword2(tweet, stopwords2):

kata = tweet.split()

kata = ' '.join(word for word in kata if word not in stopwords2)

return kata

osw = open("gdrive/My Drive/skripsi2019/prepro/stopwordtala.txt", "r")

sttwitter2 = []

for word in osw:

w = word.split()

for d in w:

sttwitter2.append(d)

hasilcsw2 = []

for tweet in tqdm(dftweet['tweet']):

procsw = cstopword2(tweet, sttwitter2)

hasilcsw2.append(procsw)

dftweet = pd.DataFrame({"tweet": hasilcsw2})

dftweet.to\_csv(r'gdrive/My Drive/skripsi2019/prepro/cstopword2.csv')

dftweet.rename(columns={"tweet": "Hasil Stopword Removal"}).head(10)

"""# \*\*6. Labelling\*\*"""

def hitung\_sentimen(tweet, tampung\_pos, tampung\_neg, hasilh):

positif = 0

negatif = 0

corpus\_positif = open("gdrive/My Drive/skripsi2019/labelling/katapos.txt").read()

positif\_words = corpus\_positif.split('\n')

corpus\_negatif = open("gdrive/My Drive/skripsi2019/labelling/kataneg.txt").read()

negatif\_words = corpus\_negatif.split('\n')

kata\_list = tweet.split(' ')

for word in kata\_list:

if word in positif\_words:

positif += 1

elif word in negatif\_words:

negatif += 1

tampung\_pos.append(positif)

tampung\_neg.append(negatif)

for i in range(len(tampung\_pos)):

if tampung\_pos[i] > tampung\_neg[i]:

hasil = 'positif'

elif tampung\_pos[i] < tampung\_neg[i]:

hasil = 'negatif'

else:

hasil = 'netral'

hasilh.append(hasil)

dftweet = pd.read\_csv("gdrive/My Drive/skripsi2019/prepro/stemming.csv", encoding = "latin-1", usecols=["tweet"])

tampung\_pos = []

tampung\_neg = []

hasilh = []

for tweet in tqdm(dftweet['tweet']):

hitung\_sentimen(tweet, tampung\_pos, tampung\_neg, hasilh)

print(len(hasilh))

dftweet = pd.DataFrame({"tweet": tweets['tweet'], "tweet\_clean": hasilstem, "positif": tampung\_pos, "negatif": tampung\_neg, "sentimen": hasilh})

df\_ml = dftweet.rename(columns={"tweet": "Tweet", "tweet\_clean":"Hasil Preprocessing", "positif": "Positif", "negatif": "Negatif", "sentimen": "Sentimen"})

df\_ml.to\_csv("gdrive/My Drive/skripsi2019/labelling/lb\_label.csv")

# df\_ml['Numbered Sentimen'] = [1 if i == 'positif' else 0 if i == 'netral' else -1 for i in df\_ml['Sentimen']]

df\_ml.head(10)

import matplotlib.pyplot as plt

count\_Class=pd.value\_counts(df\_ml["Sentimen"], sort= True)

count\_Class.plot(kind= 'bar', color= ["green", "yellow", "red" ])

plt.title('Hasil Labelling')

plt.show()

print(count\_Class)

"""## \*\*Data bersentimen positif dan negatif\*\*"""

n\_class = 2

if n\_class == 2:

df\_pos = df\_ml.copy()[df\_ml['Sentimen'] == 'positif']

df\_neg = df\_ml.copy()[df\_ml['Sentimen'] == 'negatif']

df = pd.concat([df\_pos, df\_neg], ignore\_index=True).reset\_index(drop=True)

df['Numbered Sentimen'] = [1 if i == 'positif' else 0 for i in df['Sentimen']]

df.to\_csv("gdrive/My Drive/skripsi2019/labelling/pntweet.csv")

df.head()

"""# \*\*7. Visualisasi\*\*"""

from wordcloud import WordCloud

from collections import Counter

all\_words = []

for tweet in tqdm(df\_ml['Hasil Preprocessing']):

words = tweet.split(' ')

for word in words:

all\_words.append(word)

wordfreq = Counter(all\_words)

wordcloud = WordCloud(width=900,

height=500,

max\_words=500,

max\_font\_size=100,

relative\_scaling=0.5,

colormap='Blues',

normalize\_plurals=True).generate\_from\_frequencies(wordfreq)

plt.figure(figsize=(10,10))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis("off")

plt.show()

import nltk

all\_words = []

for tweet in tqdm(df\_ml['Hasil Preprocessing']):

words = tweet.split(' ')

for word in words:

all\_words.append(word)

plt.figure(figsize=(12,5))

plt.xticks(fontsize=13, rotation=90)

plt.title('Top 20 Kata')

fd = nltk.FreqDist(all\_words)

fd.plot(25,cumulative=False)

import itertools

positif\_sentimen = df[df["Sentimen"]=="positif"]["Hasil Preprocessing"]

l = [tweet.split() for tweet in positif\_sentimen] #membuat list kata pada kelas positif

kata\_positif = list(itertools.chain.from\_iterable(l))

positif\_word = Counter(kata\_positif) #menhitung jumlah setiap kata pada kelas positif

positif\_word = sorted(positif\_word.items(), key=lambda kv: kv[1], reverse=True) #descending jumlah kata

top20\_positif = positif\_word[:21]

top\_words\_positif = [i[0] for i in top20\_positif] #top 20 positif

top\_occurences\_positif = [i[1] for i in top20\_positif]

plt.bar(top\_words\_positif, top\_occurences\_positif)

plt.title("Top 20 kata pada kelas positif")

plt.xticks(rotation='vertical')

plt.show()

negatif\_sentimen = df[df["Sentimen"]=="negatif"]["Hasil Preprocessing"]

k = [tweet.split() for tweet in negatif\_sentimen]#membuat list kata pada kelas negatif

kata\_negatif = list(itertools.chain.from\_iterable(k))

negatif\_word = Counter(kata\_negatif) #menhitung jumlah setiap kata pada kelas negatif

negatif\_word = sorted(negatif\_word.items(), key=lambda kv: kv[1], reverse=True)

top20\_negatif = negatif\_word[:21]

top\_words\_negatif = [i[0] for i in top20\_negatif] #top 20 negatif

top\_occurences\_negatif = [i[1] for i in top20\_negatif]

plt.bar(top\_words\_negatif, top\_occurences\_negatif)

plt.title("Top 20 Kata Negatif")

plt.xticks(rotation='vertical')

plt.show()

"""# \*\*8. Modelling\*\*"""

from sklearn.model\_selection import train\_test\_split

from sklearn import model\_selection, naive\_bayes, metrics

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import confusion\_matrix, classification\_report

import numpy as np

dfa = pd.read\_csv("gdrive/My Drive/skripsi2019/labelling/pntweet.csv")

from sklearn.model\_selection import train\_test\_split

X = dfa['Hasil Preprocessing']

y = dfa['Numbered Sentimen']

SEED = 123

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=SEED)

print([np.shape(X\_train), np.shape(X\_test)])

train = pd.DataFrame(X\_train)

train.head()

test = pd.DataFrame(X\_test)

test.head()

null\_accuracy = len(X\_test[y\_test == 1])/(len(X\_test))

print(null\_accuracy)

from sklearn.pipeline import Pipeline

from sklearn.feature\_extraction.text import CountVectorizer

vect = CountVectorizer()

bayes = MultinomialNB()

list\_min\_df = np.arange(1,6,1)

jum\_fitur = np.zeros(len(list\_min\_df))

score\_train = np.zeros(len(list\_min\_df))

score\_test = np.zeros(len(list\_min\_df))

count = 0

for min\_df in list\_min\_df:

vect.set\_params(min\_df=min\_df)

val1\_pipeline = Pipeline([

('vect', vect),

('clf', bayes)

])

model = val1\_pipeline.fit(X\_train, y\_train)

kata = vect.get\_feature\_names()

jum\_fitur[count]= len(kata)

score\_test[count]= model.score(X\_test, y\_test)

y\_pred\_class = model.predict(X\_test)

print('jumlah fitur = ',jum\_fitur[count])

print(classification\_report(y\_test,y\_pred\_class))

print('\n')

count = count + 1

matrix = np.matrix(np.c\_[list\_min\_df, jum\_fitur, score\_test])

models = pd.DataFrame(data = matrix, columns =

['min\_df', 'Jumlah Fitur', 'Test Accuracy'])

models

import matplotlib.ticker as mtick

graf = models.plot.bar(x='Jumlah Fitur',y='Test Accuracy', figsize=(5,5))

graf.set\_ylabel("Persentase")

graf.yaxis.set\_major\_formatter(mtick.PercentFormatter(1.0))

graf.set\_xlabel("Jumlah Fitur")

graf.set\_ylim(0.7,1)

"""## \*\*Pelatihan Model Naive Bayes\*\*"""

null\_accuracy = len(X\_test[y\_test == 1])/(len(X\_test))

print(null\_accuracy)

import joblib

vect = CountVectorizer(min\_df=3)

X\_train\_vect = vect.fit\_transform(X\_train).toarray()

model = naive\_bayes.MultinomialNB()

model.fit(X\_train\_vect, y\_train)

filename = 'finalized\_model.pkl'

joblib.dump(model, filename)

"""# \*\*9. Evaluasi Model Naive Bayes\*\*"""

X\_test\_vect = vect.transform(X\_test).toarray()

model = joblib.load(filename)

y\_pred\_class = model.predict(X\_test\_vect)

m\_confusion\_test = metrics.confusion\_matrix(y\_test, y\_pred\_class)

pd.DataFrame(data = m\_confusion\_test, columns = ['Prediksi Negatif', 'Prediksi Positif'],

index = ['Hasil Labelling Negatif', 'Hasil Labelling Positif'])

print(classification\_report(y\_test,y\_pred\_class))

accuracy = m\_confusion\_test.trace()/m\_confusion\_test.sum()

print(accuracy)

"""## Prediksi Kalimat"""

def test\_sample(model, sample):

sample\_counts = vect.transform([sample])

result = model.predict(sample\_counts)[0]

prob = model.predict\_proba(sample\_counts)[0]

print("Prediksi oleh model sebagai %s: probabilitas negatif %f, probabilitas positif %f" % (result, prob[0], prob[1]))

"""\* 1 = positif

\* 0 = negatif

"""

**LAMPIRAN 2**

**Kamus Stopword Sebagian**

ada

adalah

adanya

adapun

agak

agaknya

agar

akan

akankah

akhir

akhiri

akhirnya

aku

akulah

amat

amatlah

anda

andalah

antar

antara

antaranya

apa

apaan

apabila

apakah

apalagi

apatah

artinya

asal

asalkan

atas

atau

ataukah

ataupun

awal

awalnya

bagai

bagaikan

bagaimana

bagaimanakah

bagaimanapun

bagi

bagian

bahkan

bahwa

bahwasanya

baik

bakal

bakalan

balik

banyak

bapak

baru

bawah

beberapa

begini

beginian

beginikah

beginilah

begitu

begitukah

begitulah

begitupun

bekerja

belakang

belakangan

belum

belumlah

benar

benarkah

benarlah

berada

berakhir

berakhirlah

berakhirnya

berapa

berapakah

berapalah

berapapun

berarti

berawal

berbagai

berdatangan

beri

berikan

berikut

berikutnya

berjumlah

berkali-kali

berkata

berkehendak

berkeinginan

berkenaan

berlainan

berlalu

berlangsung

berlebihan

bermacam

bermacam-macam

bermaksud

bermula

bersama

bersama-sama

bersiap

bersiap-siap

bertanya

bertanya-tanya

berturut

berturut-turut

bertutur

berujar

berupa

besar

betul

betulkah

biasa

biasanya

bila

bilakah

bisa

bisakah

boleh

bolehkah

bolehlah

buat

bukan

bukankah

bukanlah

bukannya

bulan

bung

cara

caranya

cukup

cukupkah

cukuplah

cuma

dahulu

dalam

dan

dapat

dari

daripada

datang

dekat

demi

demikian

demikianlah

dengan

depan

di

dia

diakhiri

diakhirinya

dialah

diantara

diantaranya

diberi

diberikan

diberikannya

dibuat

dibuatnya

didapat

didatangkan

digunakan

diibaratkan

diibaratkannya

diingat

diingatkan

diinginkan

dijawab

dijelaskan

dijelaskannya

dikarenakan

dikatakan

dikatakannya

dikerjakan

diketahui

diketahuinya

dikira

dilakukan

dilalui

dilihat

dimaksud

dimaksudkan

dimaksudkannya

dimaksudnya

diminta

dimintai

dimisalkan

dimulai

dimulailah

dimulainya

dimungkinkan

dini

dipastikan

diperbuat

diperbuatnya

dipergunakan

diperkirakan

diperlihatkan

diperlukan

diperlukannya

dipersoalkan

dipertanyakan

dipunyai

diri

dirinya

disampaikan

disebut

disebutkan

disebutkannya

disini

disinilah

ditambahkan

ditandaskan

ditanya

ditanyai

ditanyakan

ditegaskan

ditujukan

ditunjuk

ditunjuki

ditunjukkan

ditunjukkannya

ditunjuknya

dituturkan

dituturkannya

diucapkan

diucapkannya

diungkapkan

dong

dua

dulu

empat

enggak

enggaknya

entah

entahlah

guna

gunakan

hal

hampir

hanya

hanyalah

hari

harus

haruslah

harusnya

hendak

hendaklah

hendaknya

hingga

ia

ialah

ibarat

ibaratkan

ibaratnya

ibu

ikut

ingat

ingat-ingat

ingin

inginkah

inginkan

ini

inikah

inilah

itu

itukah

itulah

jadi

jadilah

jadinya

jangan

jangankan

janganlah

jauh

jawab

jawaban

jawabnya

jelas

jelaskan

jelaslah

jelasnya

jika

jikalau

juga

jumlah

jumlahnya

justru

kala

kalau

kalaulah

kalaupun

kalian

kami

kamilah

kamu

kamulah

kan

kapan

kapankah

kapanpun

karena

karenanya

kasus

kata

katakan

katakanlah

katanya

ke

keadaan

kebetulan

kecil

kedua

keduanya

keinginan

kelamaan

kelihatan

kelihatannya

kelima

keluar

kembali

kemudian

kemungkinan

kemungkinannya

kenapa

kepada

kepadanya

kesampaian

keseluruhan

keseluruhannya

keterlaluan

ketika

khususnya

kini

kinilah

kira

kira-kira

kiranya

kita

kitalah

kok

kurang

lagi

lagian

lah

lain

lainnya

lalu

lama

lamanya

lanjut

lanjutnya

lebih

lewat

lima

luar

macam

maka

makanya

makin

malah

malahan

mampu

mampukah

mana

manakala

manalagi

masa

masalah

masalahnya

masih

masihkah

masing

masing-masing

mau

maupun

melainkan

**LAMPIRAN 3**

**Kamus Leksikon Sebagian**

**Kamus Leksikon Positif Sebagian**

adaptif

adem

adil

afinitas

afirmasi

agilely

agung

ahli

ahlinya

ajaib

aklamasi

akomodatif

akurat

alami

alhamdulillah

allahuakbar

aman

amanah

amaze

amazed

ambisius

amin

ampuh

andal

anggun

angkat

antik

antusias

antusiasme

anyar

apik

apresiasi

artistik

asik

asiknya

asli

aspirasi

asri

asyik

atasi

bagus

bahagia

baik

bakat

bangga

bangun

bantu

bantuan

banyak

baru

bebas

bekerja

benar

benefit

bening

beradaptasi

beralasan

berani

berapi

berarti

berasa

berbakat

bercahaya

bercanda

berdedikasi

berdikari

berempati

bergaya

bergema

bergembira

bergengsi

bergizi

berguna

berharga

berhasil

berhutang

berimbang

berisi

beristirahat

berjanji

berjasa

berjaya

berjenis

berjuang

berkat

berkeinginan

berkelanjutan

berkelas

berkembang

berkeyakinan

berkilau

berkilauan

berkualitas

berlaba

berlimpah

bermanfaat

bermartabat

bermoral

bernilai

bernyali

berpendidikan

berpengalaman

berpengaruh

berpijar

bersahaja

bersatu

bersemangat

berseri

bersih

bersinar

bersorak

bersyukur

bertekun

berterimakasih

bertobat

beruntung

bervariasi

berwawasan

besar

biasa

bijak

bijaksana

bimbingan

bisa

bonus

brilian

bulat

cahaya

cakap

canggih

cantik

capai

carikan

cekatan

cemas

cemerlang

cepat

cepatan

cepet

cepetan

cerah

cerdas

cerdik

ceria

cermat

cetar

cinta

cocok

contoh

cozy

cukup

cukuplah

dahsyat

damai

dermawan

dewasa

diakses

diakui

diampuni

dibaca

dibebaskan

diberikan

dibersihkan

dicapai

**Kamus Leksikon Negatif Sebagian**

abnormal

absurd

acak

acuh

aduh

agresi

agresif

alay

amatir

ambruk

amburadul

amin

amit

ampun

amuk

anarki

anarkis

anarkisme

anarkistis

ancaman

ancur

aneh

anehnya

anjlok

anjloknya

antagonisme

antipati

asam

asem

asing

astaghfirullah

babi

badai

bahaya

bajingan

bandel

bangke

bangkrut

bangsat

barbar

basi

batas

bau

bawahan

bebal

beban

bejat

belum

bencana

benci

bengis

berbahaya

berbisa

berbohong

berbuih

berderak

berdesakan

berdetak

berebut

bergairah

bergelombang

bergerak

berhenti

berisiko

berjangkit

berkurang

berlemak

bermasalah

bermusuhan

bersandiwara

berselisih

bertengkar

bertentangan

berteriak

bertingkah

betul

biadab

bising

blunder

bobrok

bocor

bodoh

bohong

boros

brutal

bukan

buruk

buset

busuk

cabul

cacat

cacian

cape

capek

cekcok

celaan

celaka

cemas

cemberut

cemburu

cengeng

cerewet

ceroboh

copet

cuek

culas

curam

curang

curiga

dampak

dancok

dangkal

deadline

debu

defensif

degenerasi

degradasi

dehumanisasi

delusi

demam

demoralisasi

denda

dendam

dengki

depresi

derita

desak

desis

destruktif

diam

dibajak

dibakar

dibanjiri

dibantai

dibatalkan

dibenarkan

dibenci

dibuang

dicela

dicerca

dicuci

dicuri

diganggu

dihapus

dihentikan

dihukum

dijauhi

dijelaskan

dikaburkan

dikenakan

dikerjain

dikorbankan

diktator

diktatoris

dilecehkan

dilema

dilenyapkan

dimaki

dimarahi

dimengerti

dingin

dipalsukan

dipecat

diperangi

diperkosa

dipermainkan

dipertanyakan

dipikirkan

diragukan

dirampas

direbut diremehkan

dirugikan

dirusak

disalahgunakan

disalahin

disalahpahami

disayangkan

disederhanakan

disengaja

disengketakan

disiram

diskredit

diskriminasi