Preprocessing

Preprocessing adalah tahap mempersiapkan data sebelum di proses, dimana pada tahap preprocessing ini terdapat beberapa poses yaitu.

- casefold
- tokenisasi
- stopword removal
- remove symbol dan character

pada tahap ini dibutuhkan beberapa library yang dibutuhkan yaitu.

- **nltk.tokenize**: berfungsi untuk membuat kalimat tweet menjadi token
- **Regular Expresion (re)** : untuk menghapus karakter dan simbol yang tidak dibutuhkan
- **nltk.corpus** : untuk mendapatkan daftar kata yang tidak memiliki makna
- **pandas** : untuk memproses data yang berhubungan dengan csv
- **numpy** : untuk memproses data yang berhubungan dengan array

In [1]:
```
from nltk.tokenize import TweetTokenizer
tknzr = TweetTokenizer()
import re
from nltk.corpus import stopwords
stopWords = set(stopwords.words('indonesian'))
import pandas as pd
import numpy as np
```

Casefold

untuk merubah kalimat tweet menjadi lowercase (huruf kecil)

In [2]:
```
def casefold(str):
    return str.casefold()
```

Tokenisasi

merubah kalimat tweet menjadi sebuah token

In [3]:
```
def tokenisasi(str):
    return tknzr.tokenize(str)
```

Stopword removal

untuk menghapus kata yang tidak memiliki makna. berikut contoh beberapa kata yang terdapat dalam stopword

In [4]:
```
data = list(set(stopWords))
print(data[:30])
```

```
['hendaklah', 'justru', 'depan', 'saya', 'jawab', 'kata', 'agar', 'dapat', 'sem
acam', 'semakin', 'merupakan', 'termasuk', 'aku', 'kemungkinan', 'rasa', 'sebai
k-baiknya', 'ke', 'inilah', 'menjadi', 'daripada', 'entahlah', 'khususnya', 'te
rsebutlah', 'sebelumnya', 'demi', 'jangan', 'sepertinya', 'diperkirakan', 'apat
ah', 'serupa']
```

In [5]:
```
def stopword_removal(token):
    result = []
    for i in range(len(token)):
        if token[i] not in stopWords: #mengecek apakah token tidak ada dalam sto
            result.append(token[i]) #menyimpan dalam result untuk d return
    return result
```

Remove symbol

pada fungsi ini dilakukan tahap penghapusan.

- mention dan hastag
- Link
- Symbol

In [6]:
```
def url_symbol_removal(token):
    new_token = []
    for t in token:
        if t[:1] != "@" and t[:1] != "#": # mengecek token yang tidak diawali de
            url_removed = re.sub(r"http\S+", "", t) #mengubah http menjadi "" ag
            emoji_removed = re.sub(r"([\\x][a-z0-9A-Z]+)","",url_removed)
            symbol_removed = re.sub(r"[^\w]", "", emoji_removed)
            if symbol_removed != '' and symbol_removed != '\xF0': # mengecek aga
                new_token.append(symbol_removed)
    return new_token
```

pada tahap ini fungsi di atas digabung menjadi satu agar mudah di panggil

In [7]:
```
def preprocessing(string):
    string = string[2:-1] # string dimulai dari 2 sampai -1 karena data sebelumn
    a = casefold(string) # merubah ke huruf kecil
    a = tokenisasi(a) # membuat token
    a = stopword_removal(a) # menghapus kata yang tidak penting
    a = url_symbol_removal(a) # menghapus simbol dan link
    a = ' '.join(a) # membuat array menjadi string
    return a
```

Read data

pada tahap ini data yang telah di training atau diberi label akan di baca menggunakan pandas

In [8]:
```
data = pd.read_csv("training_labeled.csv")
```

berikut sample data dari data training

- 1 = positif
- 0 = negatif

In [9]:
```
data[:10]
```

Out[9]:

	ID Tweet	ID User	Screen Name	Tweet	Timestamp
0	b'1127125080545189888'	b'1115202813968015360'	SGanjen	b'@katakitatweet @Aryprasetyo85 @emhaainunnadj...	05-11-19 08:15
1	b'1127125040342786050'	b'973292600017895425'	brothe28	b'@RachlanNashidik @prabowo @jokowi Biasa demo...	05-11-19 08:15
2	b'1127125027827044352'	b'1058630527119323136'	Toyezzz1	b'@DiniKurnia21 @MangiranNing @MahesaTiwi @kan...	05-11-19 08:15
3	b'1127125020407320576'	b'1119035387387600896'	Faqih07380284	b'@permadiaktivis @bawaslu_RI @prabowo @DivHum...	05-11-19 08:15
4	b'1127125017613881344'	b'1018171197350035456'	Jusca07538974	b'Makin muak dgn tingkah2 para Pengumpat dari ...	05-11-19 08:15
5	b'1127125008566837248'	b'887520780354953216'	leenahanwoo	b'@FerdinandHaean2 @jokowi @KPU_ID @prabowo In...	05-11-19 08:15
6	b'1127124908838871040'	b'1100944001115406336'	sasauw_nelson	b'@FerdinandHaean2 @jokowi @KPU_ID @prabowo HA...	05-11-19 08:14
7	b'1127124904405377024'	b'437379680'	Rien_Harbani	b'@RachlanNashidik @prabowo @jokowi Klo drmokr...	05-11-19 08:14
8	b'1127124885279350784'	b'941513645925658624'	beritaemak	b'@FerdinandHaean2 Pa @prabowo gausah turun bi...	05-11-19 08:14
9	b'1127124880598585345'	b'2321810466'	NiswariSejuk	b'@Demokrat_TV @renandabachtar Apa 02 gk suka ...	05-11-19 08:14

In [10]:
```
tweets = data["Tweet"]
label = data["Label"]
```

setiap tweet akan looping dan di klasifikasikan menjadi kelas Positif atau Negatif

In [11]:
```
result = []
for i in range(len(tweets)):
    pre = preprocessing(tweets[i]) if preprocessing(tweets[i]) != "" else "bagus"
    result.append({'Text' : pre,'Label' : label[i]})
df = pd.DataFrame(result) # membuat data frame dari result
df.to_csv('preprocessing result.csv', index=False, header='column_names') # mengu
```

data hasil processing

In [12]:
```
pd.read_csv('preprocessing result.csv')[:10]
```

Out[12]:

	Label	Text
0	1	orang disampingnya
1	0	demokratnya mah gk
2	0	wowo anak buah kayak dancok untung kau yg dian...
3	0	lucu gue liat d yg dukung orng ndtang d ilc bi...
4	1	muak dgn tingkah 2 pengumpat kubu puasa ramadh...
5	1	insya allah ayahanda tulus jujur jujur membela...
6	1	rakyat bodoh dungu sj ndpt dipengaruhi elit po...
7	1	klo drmokrat tdk keprntingan lg kah
8	1	pa gausah turun biar emak aja
9	1	02 gk suka tdk menganggp lbih yg kalah jk kalh...

NBC

Likelihood

pada fungsi ini data training akan dihitung frequensi kata untuk melanjutkan ke tahap berikutnya yaitu menghitung probability setiap kata

In [13]:
```
def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        return False
```

In [14]:
```
def likelihood(texts):
    token=[]
    positive={}
    negative={}
    for i in range(len(texts)):
        t = tokenisasi(texts[i])
        for something in t:
            if len(something) > 2 and not is_number(something):
                if something not in token:
                    token.append(something)
                    if labels[i] == 1:
                        positive[something] = 1
                        negative[something] = 0
                    else:
                        positive[something] = 0
                        negative[something] = 1
                else:
                    if label[i] == 1:
                        positive[something] += 1
                    else:
                        negative[something] += 1
    kata = []
    positif = []
    negatif = []
    for key in positive:
        kata.append(key)
        positif.append(positive[key])
        negatif.append(negative[key])
    res = pd.DataFrame({
        "kata" : kata,
        "positif" : positif,
        "negatif" : negatif
    })
    return res,token,positif,negatif,positive,negative
```

fungsi dibawah untuk menghitung probabilitas dari setiap kata

In [15]:
```
def prob(w,c,token,positif,negatif,positive,negative):
    if c == "positif":
        if w not in token:
            return (0+1)/(sum(positif)+len(token))
        else:
            return (positive[w]+1)/(sum(positif)+len(token))
    elif c == "negatif":
        if w not in token:
            return (0+1)/(sum(negatif)+len(token))
        else:
            return (negative[w]+1)/(sum(negatif)+len(token))
```

fungsi dibawah untuk menghitung probabilitas dari satu tweet

In [16]:
```
def P(text,token,positif,negatif):
    words = tokenisasi(text)
    positive_probability = np.prod([prob(word,"positif",token,positif,negatif,pos
    negative_probability = np.prod([prob(word,"negatif",token,positif,negatif,pos
    return "positif" if positive_probability > negative_probability else "negati
```

Classification berfungsi untuk menghitung akurasi dan F1 score dimana hasil prediksi akan dibandungkan dengan data yang telah di beri label, hasil dari fungsi ini akan menghasilkan nilai TP,FP,TN,FN

In [17]:
```
def Classification(length_training,likelihood,testing):
    TP,FP,TN,FN =0,0,0,0
    precision,recall,F1,accuracy=0,0,0,0
    unique, counts = np.unique(testing['label'],return_counts=True)
    prior_positif = dict(zip(unique,counts))['positif']/100
    prior_negatif = dict(zip(unique,counts))['negatif']/100
    result = {"kalimat":[],"label":[]}
    for i in range(len(testing)):
        prediction = P(texts[i],token,positif,negatif)
        if prediction=="positif":
            if testing['label'][i]=="positif":
                TP = TP+1
            else:
                FP = FP+1
        else:
            if testing['label'][i]=="negatif":
                TN = TN+1
            else:
                FN=FN+1
    precision=TP/(TP+FP)
    recall=TP/(TP+FN)
    F1=(2*precision*recall)/(precision+recall)
    accuracy=(TP+TN)/(TP+FP+TN+FN)
    return TP,FP,TN,FN,precision,recall,F1,accuracy
```

In [18]:
```
data_preprocessing = pd.read_csv('preprocessing result.csv')
texts = data_preprocessing["Text"]
labels = data_preprocessing["Label"]
```

In [19]:
```
data_preprocessing[:10]
```

Out[19]:

	Label	Text
0	1	orang disampingnya
1	0	demokratnya mah gk
2	0	wowo anak buah kayak dancok untung kau yg dian...
3	0	lucu gue liat d yg dukung orng ndtang d ilc bi...
4	1	muak dgn tingkah 2 pengumpat kubu puasa ramadh...
5	1	insya allah ayahanda tulus jujur jujur membela...
6	1	rakyat bodoh dungu sj ndpt dipengaruhi elit po...
7	1	klo drmokrat tdk keprntingan lg kah
8	1	pa gausah turun biar emak aja
9	1	02 gk suka tdk menganggp lbih yg kalah jk kalh...

Memanggil fungsi likekihood untuk menghasilkan data frequency word

In [20]:
```
res,token,positif,negatif,positive,negative   = likelihood(texts)
```

In [21]:
```
res.to_csv("likelihood.csv",index=False)
```

berikut sampel data dari hasil likelihood

In [22]:
```
res[:50]
```

Out[22]:

	kata	negatif	positif
0	orang	1	1
1	disampingnya	0	1
2	demokratnya	1	0
3	mah	1	0
4	wowo	1	0
5	anak	1	2
6	buah	1	0
7	kayak	1	0
8	dancok	1	0
9	untung	1	0
10	kau	3	0
11	diancam	1	0
12	tim	1	0
13	mawar	1	0
14	lucu	1	0
15	gue	2	0
16	liat	1	1
17	dukung	1	1
18	orng	1	0
19	ndtang	1	0
20	ilc	1	0
21	bikin	3	0
22	malu	2	0
23	nklo	1	0
24	dtang	1	0
25	debat	1	0
26	salah	1	2
27	mulu	1	0
28	nheran	1	0
29	muji	2	0
30	mahluq	1	0
31	nwajar	1	0
32	aja	5	1
33	biar	1	1
34	berkembang	1	0

	kata	negatif	positif
35	biak	1	0
36	mentok	1	0
37	muak	0	1
38	dgn	4	4
39	tingkah	0	1
40	pengumpat	0	1
41	kubu	0	2
42	puasa	0	2
43	ramadhan	0	1
44	segan	0	1
45	dikotori	0	1
46	sungguh	0	1
47	mohon	0	1
48	bantu	0	1
49	amankan	0	1

pada tahap ini dilakukan penghitungan prior, yaitu kemungkinan persentasi kelas tertentu

In [23]:
```
unique, counts = np.unique(labels,return_counts=True)
prior_positif = dict(zip(unique,counts))[1]/100
prior_negatif = dict(zip(unique,counts))[0]/100
```

In [24]:
```
print("prior positif ",prior_positif)
print("prior negatif ",prior_negatif)
```

```
prior positif  0.61
prior negatif  0.39
```

pada tahap ini dilakukan looping semua data dan dilakukan klasifikasi

In [25]:
```
result = {"kalimat":[],"label":[]}
for i in range(len(texts)):
    result['kalimat'].append(texts[i])
    result['label'].append(P(texts[i],token,positif,negatif))
```

In [26]:
```
res = pd.DataFrame(result)
```

In [27]:
```
res.to_csv("prediction.csv", index=False, header='column_names')
```

berikut adalah hasil prediksi dari klasifikasi

In [28]:
```
pd.read_csv("prediction.csv")
```

Out[28]:

	kalimat	label
0	orang disampingnya	positif
1	demokratnya mah gk	negatif
2	wowo anak buah kayak dancok untung kau yg dian...	negatif
3	lucu gue liat d yg dukung orng ndtang d ilc bi...	negatif
4	muak dgn tingkah 2 pengumpat kubu puasa ramadh...	positif
5	insya allah ayahanda tulus jujur jujur membela...	positif
6	rakyat bodoh dungu sj ndpt dipengaruhi elit po...	positif
7	klo drmokrat tdk keprntingan lg kah	positif
8	pa gausah turun biar emak aja	positif
9	02 gk suka tdk menganggp lbih yg kalah jk kalh...	positif
10	jebakan	negatif
11	artikan power kekuatan fisik kekuatan moral po...	positif
12	njangan offline side n ndarah merah pemakan da...	positif
13	senin bang mengadakan buka puasa rumah kertane...	positif
14	jd gunanya bawaslu mk dkkp	positif
15	bentuklah om	positif
16	resiko pembangkangan rakyat semesta mayoritas ...	positif
17	pokoknya prabowo menang nharus jd presiden n n...	positif
18	pokeke lawan 2024 yg berani	positif
19	gorengan	positif
20	pake tolong ditambahin pendukung 02 n n	positif
21	ah bodo sih l	negatif
22	nmassa andaanda nkedaulatan konstitusi terarah...	negatif
23	alhamdulillah puisi nw munajat akbar 212 dikab...	positif
24	murka allah diaceh bencana alam sunami 26 des ...	negatif
25	menang 2014 curang klo 2019 tingal diam hadapi...	negatif
26	sok bener dik nmau manganulir pemilu 2019 nnaif	negatif
27	gila keracunan kebanyakan kekuasaan kebodohan ...	negatif
28	ahy ngaku kalah krn urutan ketiga	negatif
29	emang ahmad dani yg ditahan n	negatif
...	...	...
70	penjarakan nmanusia pengangguran politik nhany...	negatif
71	diceneralisasi pilpres curang kpu curang bohon...	negatif
72	win win solution maksud loh grey	positif
73	kerennnn om lanjutkan om semangat tersenyum	positif

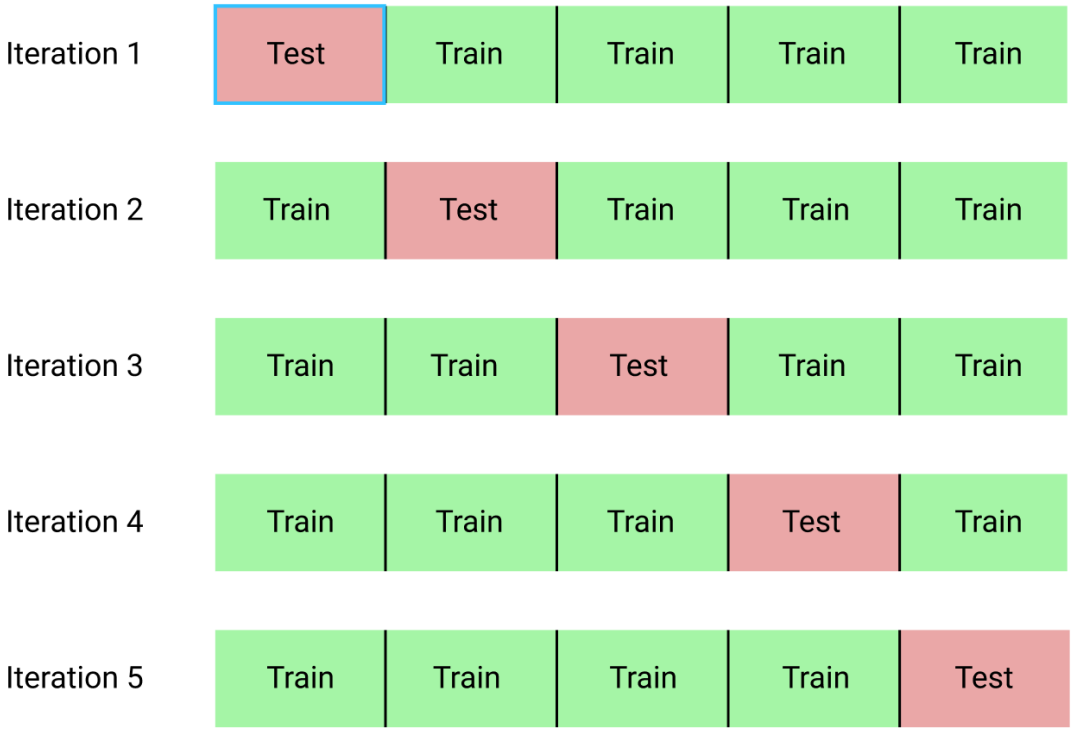
	kalimat	label
74	baca menduga pernyataan hasto fitnah mengarah ...	positif
75	jujur	positif
76	slah calon uang rakyat menggunakn lembaga nega...	positif
77	cc nbang bro	positif
78	cc in ah	positif
79	ga pake ahy bang pideo nya promosi ya	negatif
80	tujuan kubu 02 ngusulin pembentukan tpf utk me...	positif
81	mengaku kalah ga pakai alasan ngeles ngeles pe...	negatif
82	tk sutradara tk rela	positif
83	bikin tpf akal akalan yg kalah aja pa jkw gabu...	negatif
84	persengkolan parpol pendukung prabowo mengalih...	positif
85	bagus	positif
86	pengkhianat nyinyirlah malu mengangkang 02	negatif
87	indonesia kandung njika biarkan kandung perkos...	positif
88	pemarah mudah stroke	negatif
89	salut abang mari berjuang bang anak medan kena...	positif
90	alhamdulillah nbukannya ntu tanah leluhur bpk ...	negatif
91	ukri bocorkan 01 dipaksakan menang n n n n n n...	negatif
92	sampeyan kalah suara coba pakai jurrus cawet a...	negatif
93	kali setuju dgn tweet	positif
94	suaranya g kawan	negatif
95	mantap	positif
96	dibawah dikecamatan dikawal polisi saksidikeca...	positif
97	netral smnjak liat qc n	positif
98	alhamdulillah	positif
99	masjidnya prabowo yg yah	positif

100 rows × 2 columns

Validation

cross validation

pada tahap ini dilakukan pengujian menggunakan cross validation dengan k-fold = 10 untuk menghasilkan nilai yang stabil


Iteration 1
Test
Train
Train
Train
Train
Iteration 2
Train
Test
Train
Train
Train
Iteration 3
Train
Train
Test
Train
Train
Iteration 4
Train
Train
Train
Test
Train
Iteration 5
Train
Train
Train
Train
Test

In [29]:
```
def crossValidations(kfold,data):
    result = {'Fold':[],'TP':[],'FP':[],'TN':[],'FN':[],'precision':[],'recall':
    fold = round(len(data) / kfold)
    for i in range(kfold):
        start = i * fold
        end = (i + 1) * fold
        training_set = {'kalimat': [],'label':[]}
        testing_set = {'kalimat': [],'label':[]}

        if end > len(data):
            end = len(data)
        for j in range(len(data)):
            if start<= j <= end:
                testing_set['kalimat'].append(data['kalimat'][j])
                testing_set['label'].append(data['label'][j])
            else:
                training_set['kalimat'].append(data['kalimat'][j])
                training_set['label'].append(data['label'][j])
        training = pd.DataFrame(training_set)
        testing = pd.DataFrame(testing_set)
        res,token,positif,negatif,positive,negative = likelihood(training['kalim
        frequencyWord = pd.DataFrame(res)
        TP, FP, TN, FN,precision,recall,F1,accuracy = Classification(len(trainin
        result['Fold'].append(i+1)
        result['TP'].append(TP)
        result['FP'].append(FP)
        result['TN'].append(TN)
        result['FN'].append(FN)

        result['precision'].append(precision)
        result['recall'].append(recall)
        result['F1'].append(F1)
        result['accuracy'].append(accuracy)
    return result
```

In [30]:
```
data_testing = pd.read_csv('prediction.csv')
fold =10
```

In [31]:
```
result = crossValidations(10,data_testing)
```

In [32]:
```
res = pd.DataFrame(result)
```

In [33]:
```
res.to_csv("hasil_cross.csv", index=False, header='column_names')
```

In [34]:
```
pd.read_csv("hasil_cross.csv")
```

Out[34]:

	F1	FN	FP	Fold	TN	TP	accuracy	precision	recall
0	1.000000	0	0	1	4	7	1.000000	1.000000	1.000000
1	0.705882	4	1	2	0	6	0.545455	0.857143	0.600000
2	0.200000	2	6	3	2	1	0.272727	0.142857	0.333333
3	0.666667	3	2	4	1	5	0.545455	0.714286	0.625000
4	0.666667	1	3	5	3	4	0.636364	0.571429	0.800000
5	0.666667	3	2	6	1	5	0.545455	0.714286	0.625000
6	0.833333	0	2	7	4	5	0.818182	0.714286	1.000000
7	0.666667	3	2	8	1	5	0.545455	0.714286	0.625000
8	0.769231	1	2	9	3	5	0.727273	0.714286	0.833333
9	0.769231	1	2	10	2	5	0.700000	0.714286	0.833333

hasil pengujian menggunakan cross-validation menghasilkan nilai

In [35]:
```
f1,accuracy=0,0
for i in range(len(result)):
    f1 = f1+result['F1'][i]
    accuracy=accuracy+ result['accuracy'][i]
print("rata-rata F1 ",f1/len(result))
print("rata-rata accuracy ",accuracy/len(result))
```

```
rata-rata F1  0.6861236802413272
rata-rata accuracy  0.6262626262626262
```