Modul 8 Praktikum Natural Language Processing

Sentiment Analysist Statistical Base

Implementing Sentiment Analysis

Salah satu aplikasi industri NLP yang populer adalah Analisis Sentimen. Hal ini sangat penting dari sudut pandang bisnis untuk memahami bagaimana umpan balik, baik dari pelanggan produk, maupun opini publik.

Problem

Bagaimana Mengimplementasikan Sentimen Analisis

Solution

Mengimplementasikan library umum dan data emosi eksternal untuk membangun Analisis Sentimen

How It Works

ikuti langkah-langkah di bagian ini untuk menerapkan analisis sentimen pada kasus aplikasi android

1. Memahami mendefinisikan masalah bisnis

2. Mengidentifikasi sumber, pengumpulan, dan pemahaman data yang potensial

3. Implementasi kedalam program

Mount Google Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

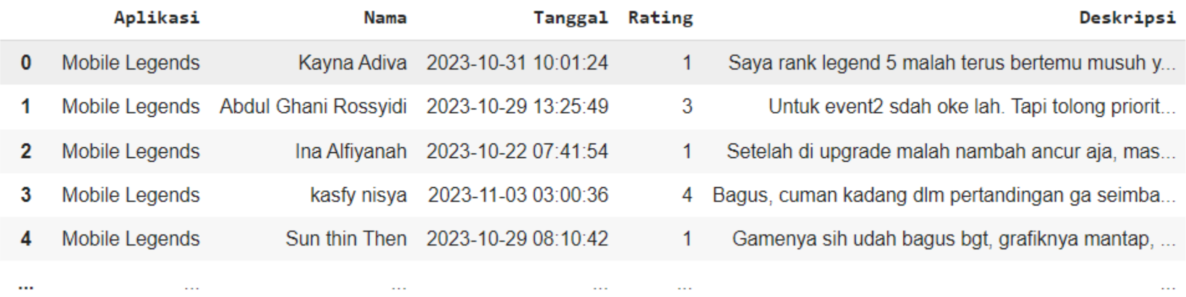
Import library

```
#import modul
import pandas as pd
import numpy as np
import re
import re as reg
import matplotlib.pyplot as plt
%matplotlib inline
```

Input Data Hotel

```
#input data hotel
dataset = pd.read_csv('dataku_lite.csv, sheet_name='Sheet1')

#output data hotel
dataset
```

Output:



	Aplikasi	Nama	Tanggal	Rating	Deskripsi
0	Mobile Legends	Kayna Adiva	2023-10-31 10:01:24	1	Saya rank legend 5 malah terus bertemu musuh y...
1	Mobile Legends	Abdul Ghani Rossyidi	2023-10-29 13:25:49	3	Untuk event2 sdah oke lah. Tapi tolong priorit...
2	Mobile Legends	Ina Alfiyanah	2023-10-22 07:41:54	1	Setelah di upgrade malah nambah ancur aja, mas...
3	Mobile Legends	kasfy nisya	2023-11-03 03:00:36	4	Bagus, cuman kadang dlm pertandingan ga seimba...
4	Mobile Legends	Sun thin Then	2023-10-29 08:10:42	1	Gamenya sih udah bagus bgt, grafiknya mantap, ...
...	...	...	...	...	...

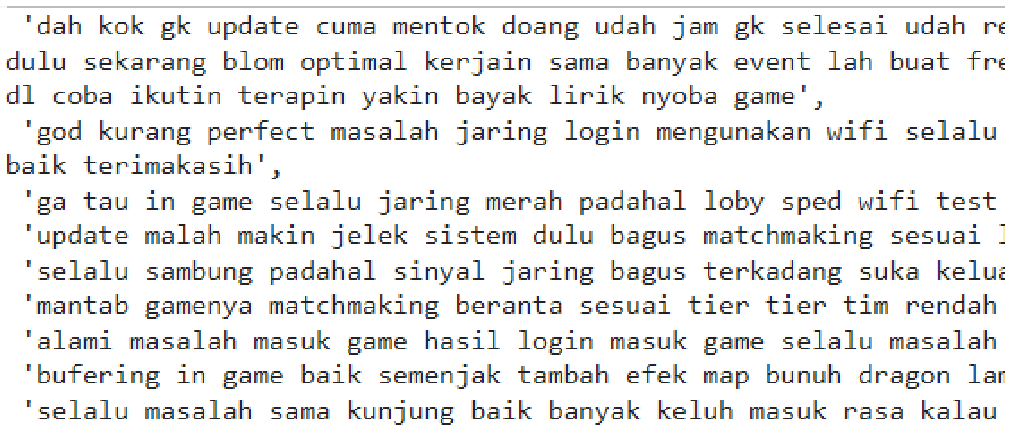
Preprocessing:

```
pip install sastrawi
```

Bersihkan kata

```
#Preprocessing
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

slangs={'yg':'yang', 'tdk':'tidak', 'pd':'pada', 'mlh':'malah',
'jgn':'jangan', 'jg':'juga', 'tp':'tapi', 'blkg': 'belakang',
'dr':'dari', 'klo':'kalo', 'lg':'lagi'}
processed_comments = []

for sentence in dataset['deskripsi']:
    # Remove all the special characters
    processed_comment = re.sub(r'\W', ' ', str(sentence))

    # Converting to Lowercase
    processed_comment = processed_comment.lower()
    #Remove number
    processed_comment = re.sub(r'\d+', ' ', processed_comment)

    # remove all single characters
    processed_comment = re.sub(r'\s+[a-zA-Z]\s+', ' ',
processed_comment)

    #remove duplicate character
    pattern=reg.compile(r"(.)\1{1,}",reg.DOTALL)
    processed_comment=pattern.sub(r"\1",processed_comment)


    #Corrected Slang words
    words = processed_comment.split()
    rfrm=[slangs[word] if word in slangs else word for word in words]
    processed_comment= " ".join(rfrm)

    #remove stopword
    factory = StopWordRemoverFactory()
    more_stopword = ['tak', 'jd', 'per', 'nya'] #menambahkan stopword
    stopwords = factory.get_stop_words() + more_stopword
    temp = [t for t in re.findall(r'\b[a-z]+-?[a-
z]+\b',processed_comment) if t not in stopwords]
    processed_comment = ' '.join(temp)

    #stemming
    stemmer = StemmerFactory().create_stemmer()
    processed_comment = stemmer.stem(processed_comment)

    # Substituting multiple spaces with single space
    processed_comment = re.sub(r'\s+', ' ', processed_comment,
flags=re.I)

    processed_comments.append(processed_comment)

#output data Preprocessing
processed_comments
```

Output:



```
 'dah kok gk update cuma mentok doang udah jam gk selesai udah re
dulu sekarang blom optimal kerjain sama banyak event lah buat fre
dl coba ikutin terapin yakin bayak lirik nyoba game',
 'god kurang perfect masalah jaring login mengunakan wifi selalu
baik terimakasih',
 'ga tau in game selalu jaring merah padahal loby sped wifi test
 'update malah makin jelek sistem dulu bagus matchmaking sesuai l
 'selalu sambung padahal sinyal jaring bagus terkadang suka kelua
 'mantab gamenya matchmaking beranta sesuai tier tier tim rendah
 'alami masalah masuk game hasil login masuk game selalu masalah
 'bufering in game baik semenjak tambah efek map bunuh dragon lan
 'selalu masalah sama kunjung baik banyak keluh masuk rasa kalau
```

Export Data Bersih Kedalam Excell

```
pip install xlsxwriter
```

```
#SAVE HASIL PREPROCESSING
import xlsxwriter
workbook = xlsxwriter.Workbook('hasilpreprocessing.xlsx',
{'nan_inf_to_errors': True})
worksheet=workbook.add_worksheet()
row=0
col=0
x=dataset
hasilakhir=list(zip(x.aplikasi,
x.nama,x.tanggal,x.rating,x.deskripsi,processed_comments))
worksheet.write(row, col, "aplikasi")
worksheet.write(row, col+1, "nama")
worksheet.write(row, col+2, "tanggal")
worksheet.write(row, col+3, "rating")
worksheet.write(row, col+4, "deskripsi")
worksheet.write(row, col+5, "CleanReview")
row+=1
for a,b,c,d,e,f in (hasilakhir):
    worksheet.write(row, col, a)
    worksheet.write(row, col+1, b)
    worksheet.write(row, col+2, c)
    worksheet.write(row, col+3, d)
    worksheet.write(row, col+4, e)
    worksheet.write(row, col+5, f)
    row+=1
workbook.close()
```

Jika Preprocessing Sudah Ada, Maka Proses Diawali Dari Sini

```
dataset = pd.read_excel('hasilpreprocessing.xlsx', sheet_name='Sheet1')
```

```
dataset['Number_of_words'] = dataset['CleanReview'].apply(lambda
x:len(str(x).split()))
dataset.drop(dataset[dataset["Number_of_words"]<2].index, inplace =
True)
```

Load Dictionary

```
dictionary=pd.read_excel('NRC.xlsx', sheet_name='dict')
#Load to memory
positive=[]
negative=[]
anger=[]
anticipation=[]
disgust=[]
fear=[]
joy=[]
sadness=[]
surprise=[]
trust=[]
teksbaru=''
for i in range(1,len(dictionary)):
    kata=dictionary.iloc[i,1]
    #print(kata)
    if (dictionary.iloc[i,6])==1:
        positive.append(kata)
    if (dictionary.iloc[i,7])==1:
        negative.append(kata)
    if (dictionary.iloc[i,8])==1:
        anger.append(kata)
    if (dictionary.iloc[i,9])==1:
        anticipation.append(kata)
    if (dictionary.iloc[i,10])==1:
        disgust.append(kata)
    if (dictionary.iloc[i,11])==1:
        fear.append(kata)
    if (dictionary.iloc[i,12])==1:
        joy.append(kata)
    if (dictionary.iloc[i,13])==1:
        sadness.append(kata)
    if (dictionary.iloc[i,14])==1:
        surprise.append(kata)
    if (dictionary.iloc[i,15])==1:
        trust.append(kata)
```

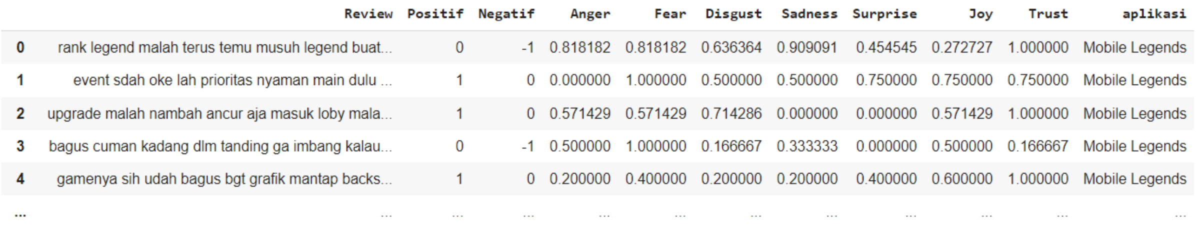
Lakukan Ekstraksi Fitur

```
#Preprocessing
processed_comments=dataset['CleanReview']
print('Panjang :',len(processed_comments))

datasetangkapos=[]
for kalimat in processed_comments:
    baris=[]
    baris.append(kalimat)
    #ekstraksi fitur
    f_sentimenpositif=0
    f_sentimennegatif=0
    f_anger=0
    f_fear=0
    f_disgust=0
    f_sadness=0
    f_surprise=0
    f_joy=0
    f_trust=0
    y=kalimat
    # Untuk setiap kata (teks) dalam kalimat, dilakukan iterasi untuk
mencocokkan dengan kata-kata dalam set emosi dan sentimen
positif/negatif.
    for teks in y.split():
        # Iterasi melalui indeks kata-kata dalam kamus kata positif
(positive)
        for j in range(0,len(positive)-1):
            # Jika kata dalam kalimat sama dengan kata positif
            if teks==positive[j]:
                # (frekuensi sentimen positif) akan ditambah 1.
                f_sentimenpositif+=1
        for j in range(0,len(negative)-1):
            if teks==negative[j]:
                f_sentimennegatif+=1
        for j in range(0,len(anger)-1):
            if teks==anger[j]:
                f_anger+=1
        for j in range(0,len(fear)-1):
            if teks==fear[j]:
                f_fear+=1
        for j in range(0,len(disgust)-1):
            if teks==disgust[j]:
                f_disgust+=1
        for j in range(0,len(sadness)-1):
            if teks==sadness[j]:
                f_sadness+=1
        for j in range(0,len(joy)-1):
            if teks==joy[j]:
                f_joy+=1
        for j in range(0,len(surprise)-1):
            if teks==surprise[j]:
                f_surprise+=1
        for j in range(0,len(trust)-1):
            if teks==trust[j]:
                f_trust+=1

    # Menghitung skor sentimen dengan mengurangkan frekuensi kata-kata negatif dari kata-kata positif.
    # Hasilnya, l_sentimen akan menjadi indikator sentimen keseluruhan dari kalimat.
    l_sentimen=f_sentimenpositif-f_sentimennegatif

    # mencari nilai max dari kelas emosi
    totalemotion=max([f_anger,f_fear,f_disgust,f_sadness,f_surprise,f_joy,f_trust])

    # deklarasi variabel untuk menyimpan nilai sentimen
    l_sentimenne=0 # Indikator sentimen negatif.
    l_sentimenp=0 # Indikator sentimen positif.
    l_sentimenn=0 # Indikator sentimen netral.

    # Jika skor sentimen positif dan negatif sama, maka kalimat dianggap netral.
    if f_sentimenpositif-f_sentimennegatif==0:
        # Skor sentimen negatif diatur menjadi 0 (netral).
        l_sentimenne=0
    # Jika skor sentimen positif dan negatif tidak sama
    else:
        # Jika skor sentimen positif lebih besar dari skor sentimen negatif, maka kalimat dianggap positif.
        if f_sentimenpositif-f_sentimennegatif>0:
            # Skor sentimen positif diatur menjadi 1.
            l_sentimenp=1
        # Jika skor sentimen positif lebih kecil dari skor sentimen negatif, maka kalimat dianggap negatif.
        else:
            # Skor sentimen negatif diatur menjadi -1.
            l_sentimenn=-1

    if totalemotion==0:
        totalemotion=1;
    l_anger=f_anger/totalemotion
    l_disgust=f_disgust/totalemotion
    l_fear=f_fear/totalemotion
    l_sadness=f_sadness/totalemotion
    l_surprise=f_surprise/totalemotion
    l_joy=f_joy/totalemotion
    l_trust=f_trust/totalemotion

    baris.append(l_sentimenp)                            #dataset(3)
    baris.append(l_sentimenn)                            #dataset(3)
    baris.append(l_anger)                            #dataset(4)
    baris.append(l_fear)                            #dataset(5)
    baris.append(l_disgust)                            #dataset(6)
    baris.append(l_sadness)                            #dataset(7)
    baris.append(l_surprise)                            #dataset(8)
    baris.append(l_joy)                            #dataset(9)
    baris.append(l_trust)                            #dataset(10)
    datasetangkapos.append(baris)
```

Buat Dataframe Untuk Menyimpan Nilai

```
datasetanotated=pd.DataFrame(datasetangkapos,
columns=['Review','Positif','Negatif','Anger','Fear','Disgust','Sadness
','Surprise','Joy','Trust'])
datasetanotated['aplikasi']=dataset['aplikasi']
datasetanotated
```

Output:



	Review	Positif	Negatif	Anger	Fear	Disgust	Sadness	Surprise	Joy	Trust	aplikasi
0	rank legend malah terus temu musuh legend buat...	0	-1	0.818182	0.818182	0.636364	0.909091	0.454545	0.272727	1.000000	Mobile Legends
1	event sdah oke lah prioritas nyaman main dulu ...	1	0	0.000000	1.000000	0.500000	0.500000	0.750000	0.750000	0.750000	Mobile Legends
2	upgrade malah nambah ancur aja masuk loby mala...	1	0	0.571429	0.571429	0.714286	0.000000	0.000000	0.571429	1.000000	Mobile Legends
3	bagus cuman kadang dlm tanding ga imbang kalau...	0	-1	0.500000	1.000000	0.166667	0.333333	0.000000	0.500000	0.166667	Mobile Legends
4	gamenya sih udah bagus bgt grafik mantap backs...	1	0	0.200000	0.400000	0.200000	0.200000	0.400000	0.600000	1.000000	Mobile Legends
...	...	...	...	...	...	...	...	...	...	...	...

Rename nama kolom agar bisa diakses

```
datasetfitur = datasetanotated.drop('Review', 1)
```

```
grouped_app = dataset_fitur.rename(columns={'Mobile Legends':
'MobileLegends'})
```

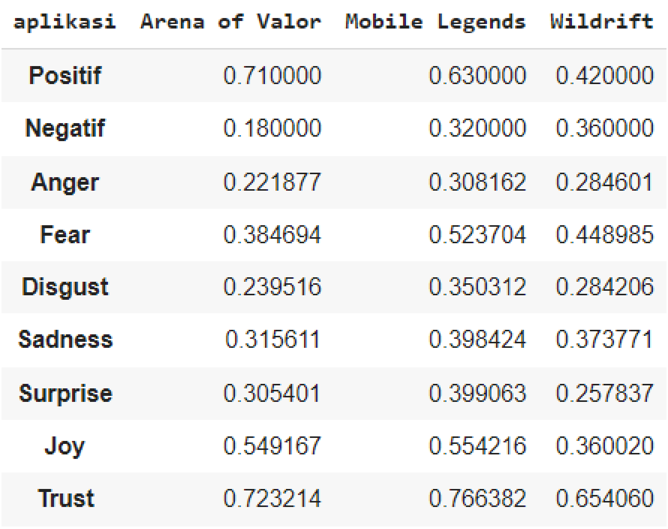
```
grouped_app = grouped_app.rename(columns={'Arena of Valor':
'ArenaOfValor'})
```

Buang Kolom Review dan lakukan groping berdasarkan kolomnya

```
datasetfitur = datasetanotated.drop('Review', 1)
```

```
grouped_obwis = datasetfitur.groupby("aplikasi").mean().abs()
# Lakukan Transpose
grouped_obwis=grouped_obwis.T
```

Output:

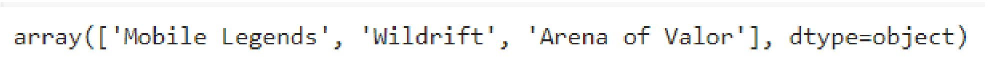


aplikasi	Arena of Valor	Mobile Legends	Wildrift
Positif	0.710000	0.630000	0.420000
Negatif	0.180000	0.320000	0.360000
Anger	0.221877	0.308162	0.284601
Fear	0.384694	0.523704	0.448985
Disgust	0.239516	0.350312	0.284206
Sadness	0.315611	0.398424	0.373771
Surprise	0.305401	0.399063	0.257837
Joy	0.549167	0.554216	0.360020
Trust	0.723214	0.766382	0.654060

Cek nilai unik pada total aplikasi

```
komenaplikasi=datasetanotated.aplikasi.unique()
komenaplikasi
```

Output:

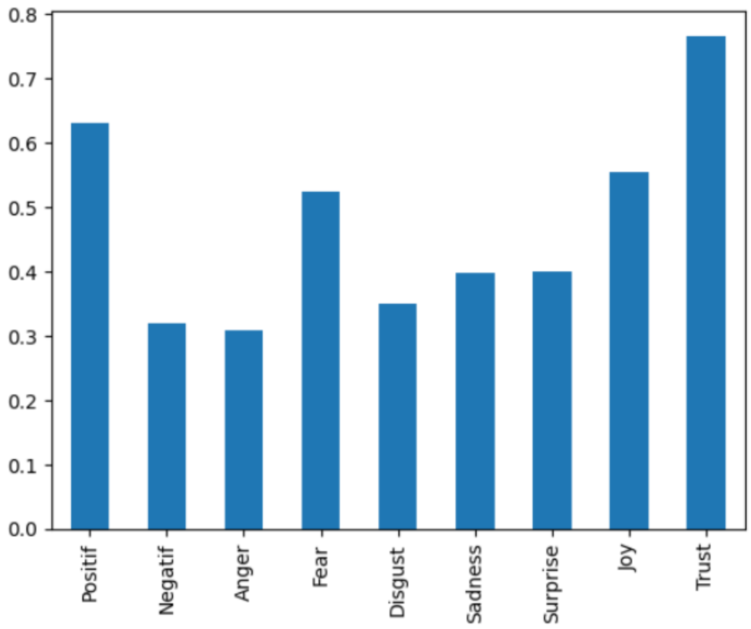


```
array(['Mobile Legends', 'Wildrift', 'Arena of Valor'], dtype=object)
```

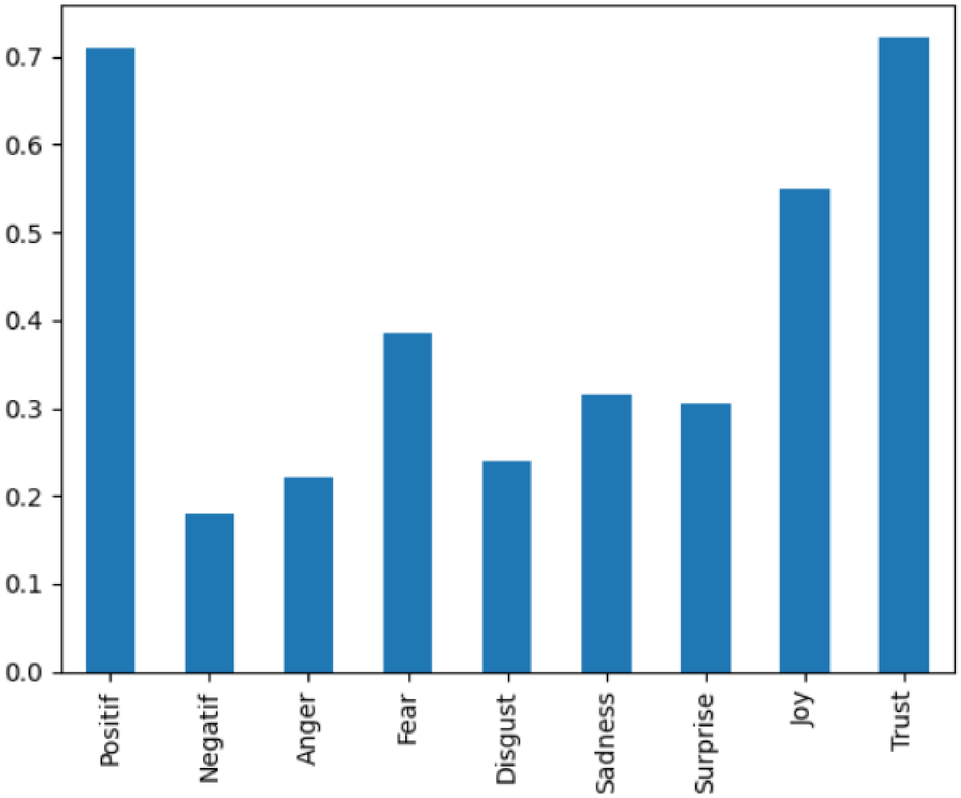
Lakukan Visualisasi

Barchart

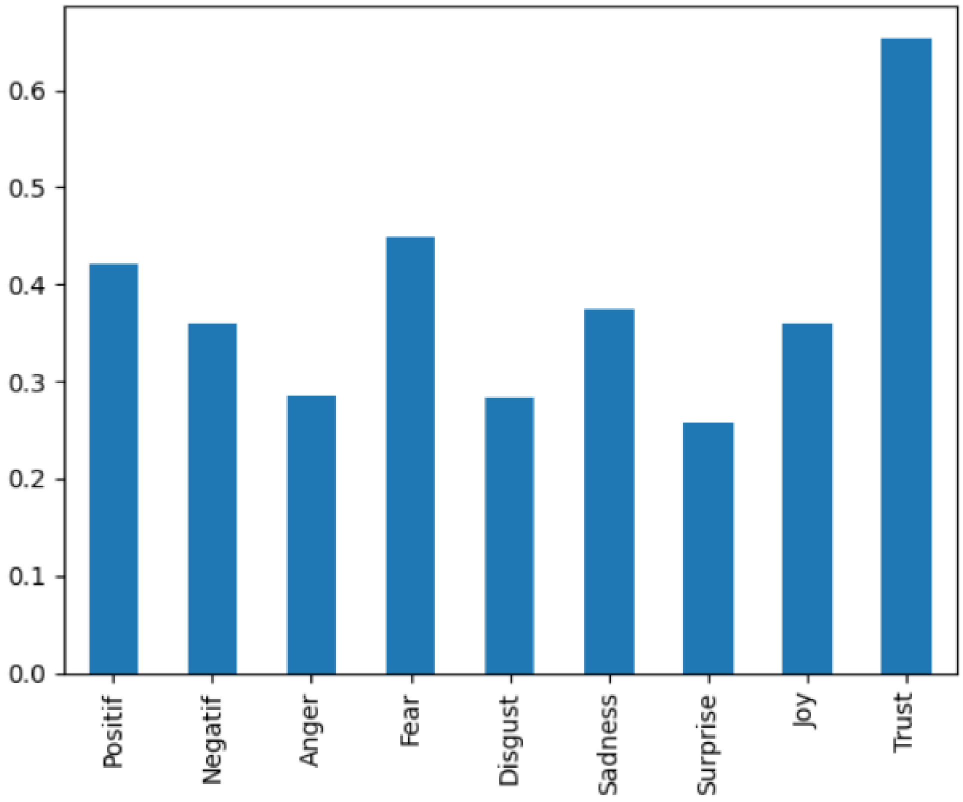
```
grouped_app.MobileLegends.plot(kind='bar')
```



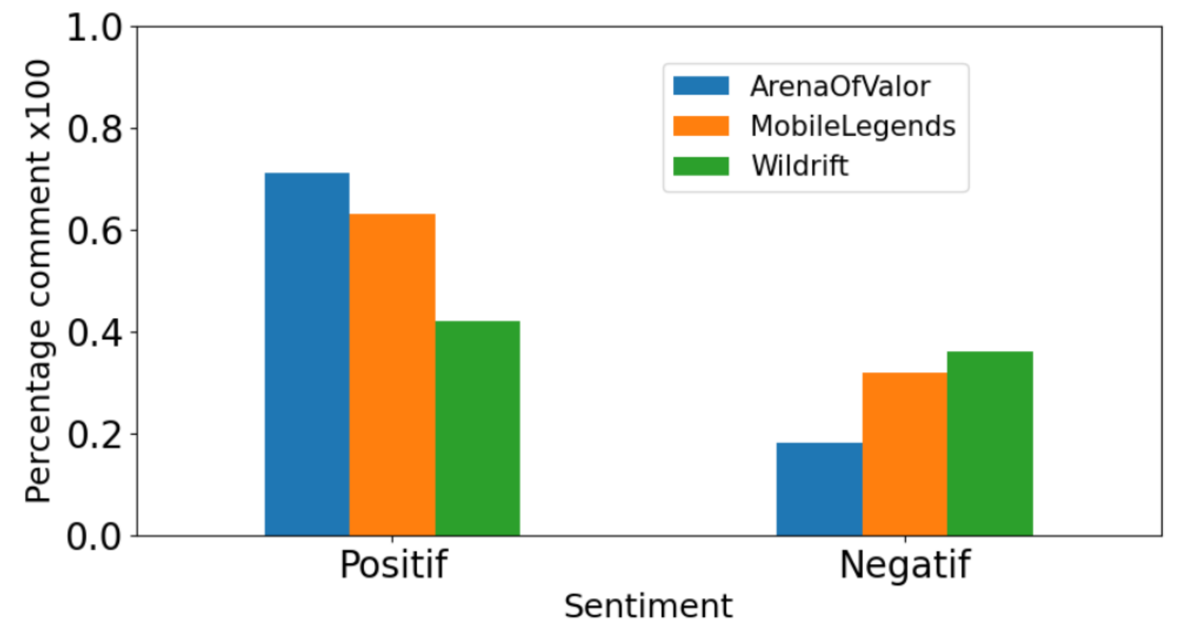
```
grouped_app.ArenaOfValor.plot(kind='bar')
```



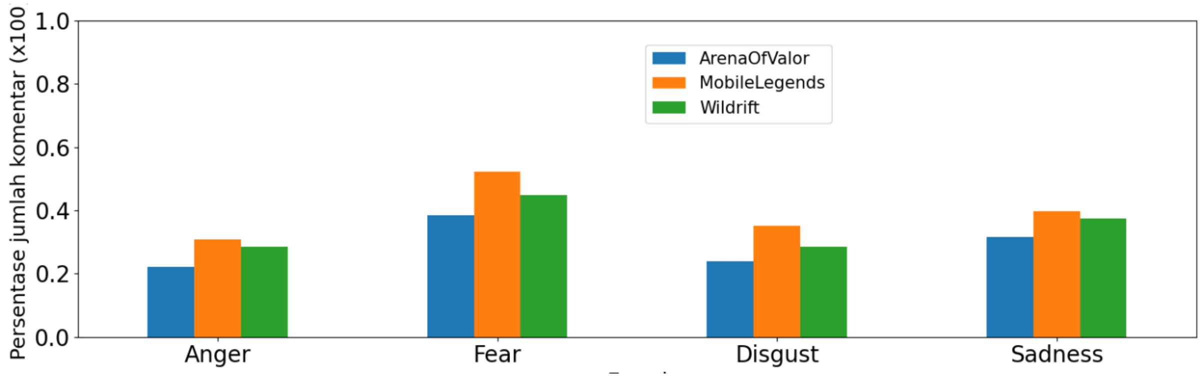
```
grouped_app.Wildrift.plot(kind='bar')
```



```
grouped_app.iloc[0:2,:].plot(kind='bar', figsize=(10,5));
plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True
d = {'Column 1': [i for i in range(10)],
      'Column 2': [i * i for i in range(10)]}
#plt.legend(bbox_to_anchor=(1.0, 1.0))
plt.legend(loc='center left', bbox_to_anchor=(0.5, 0.8),  prop={'size':
15})
plt.ylim([0, 1])
plt.xticks(fontsize = 20, rotation=0)
plt.yticks(fontsize = 20)
plt.xlabel('Sentiment', fontsize=18)
plt.ylabel('Percentage comment x100', fontsize=18)

plt.show()
```

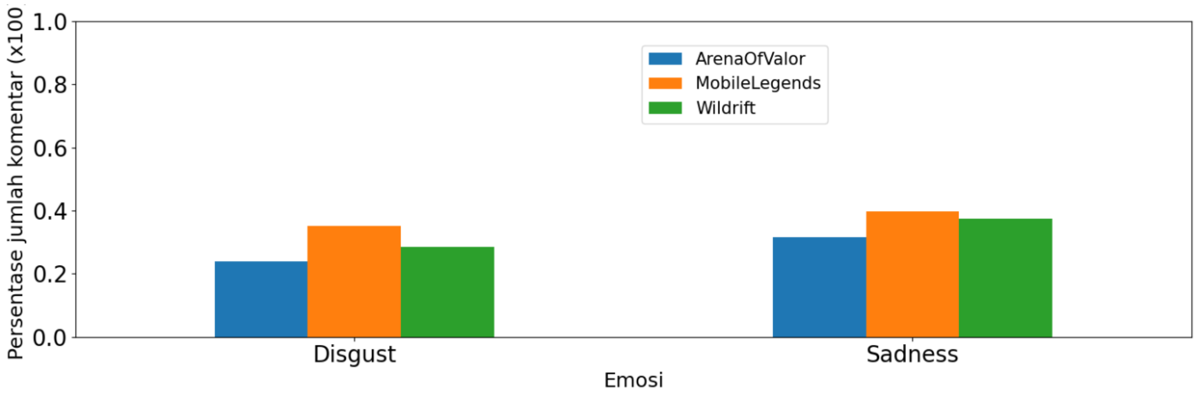


```
grouped_app.iloc[2:6,:].plot(kind='bar', figsize=(15,5));
plt.rcParams["figure.figsize"] = [4.50, 3.50]
plt.rcParams["figure.autolayout"] = True
d = {'Column 1': [i for i in range(10)],
      'Column 2': [i * i for i in range(10)]}
#plt.legend(bbox_to_anchor=(1.0, 1.0))
plt.legend(loc='center left', bbox_to_anchor=(0.5, 0.8),  prop={'size':
15})
plt.ylim([0, 1])
plt.xticks(fontsize = 20, rotation=0)
plt.yticks(fontsize = 20)
plt.xlabel('Emosi', fontsize=18)
plt.ylabel('Persentase jumlah komentar (x100)', fontsize=18)
plt.show()
```

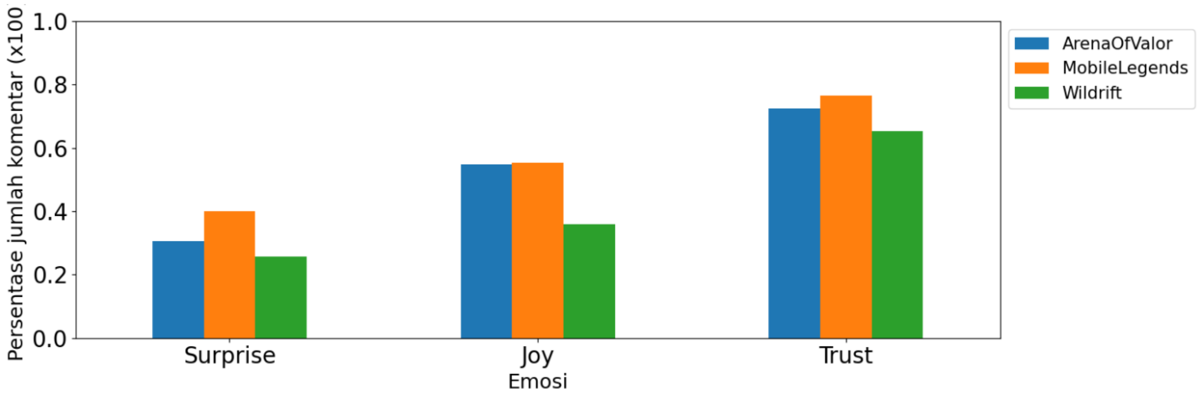


```
grouped_app.iloc[4:6,:].plot(kind='bar', figsize=(15,5));
plt.rcParams["figure.figsize"] = [4.50, 3.50]
plt.rcParams["figure.autolayout"] = True
```

```
d = {'Column 1': [i for i in range(10)],
      'Column 2': [i * i for i in range(10)]}
#plt.legend(bbox_to_anchor=(1.0, 1.0))
plt.legend(loc='center left', bbox_to_anchor=(0.5, 0.8),  prop={'size':
15})
plt.ylim([0, 1])
plt.xticks(fontsize = 20, rotation=0)
plt.yticks(fontsize = 20)
plt.xlabel('Emosi', fontsize=18)
plt.ylabel('Persentase jumlah komentar (x100)', fontsize=18)
plt.show()
```



```
grouped_app.iloc[6:9,:].plot(kind='bar', figsize=(15,5));
plt.rcParams["figure.figsize"] = [4.50, 3.50]
plt.rcParams["figure.autolayout"] = True
d = {'Column 1': [i for i in range(10)],
      'Column 2': [i * i for i in range(10)]}
#plt.legend(bbox_to_anchor=(1.0, 1.0))
plt.legend(loc='upper left', bbox_to_anchor=(1, 1),  prop={'size': 15})
plt.ylim([0, 1])
plt.xticks(fontsize = 20, rotation=0)
plt.yticks(fontsize = 20)
plt.xlabel('Emosi', fontsize=18)
plt.ylabel('Persentase jumlah komentar (x100)', fontsize=18)
plt.show()
```



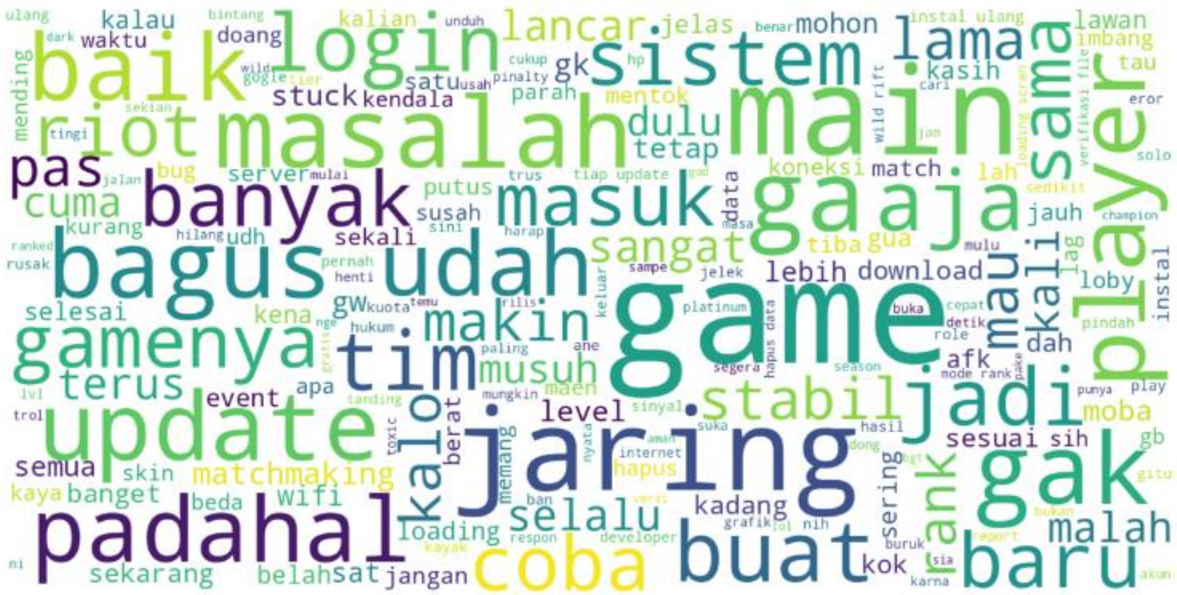
WordCloud

```
from sklearn.feature_extraction.text import
CountVectorizer,TfidfVectorizer, HashingVectorizer

def get_top_n_words(corpus, n=None):

    vec = CountVectorizer().fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in
vec.vocabulary_.items()]
    words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
    return words_freq[:n]
```

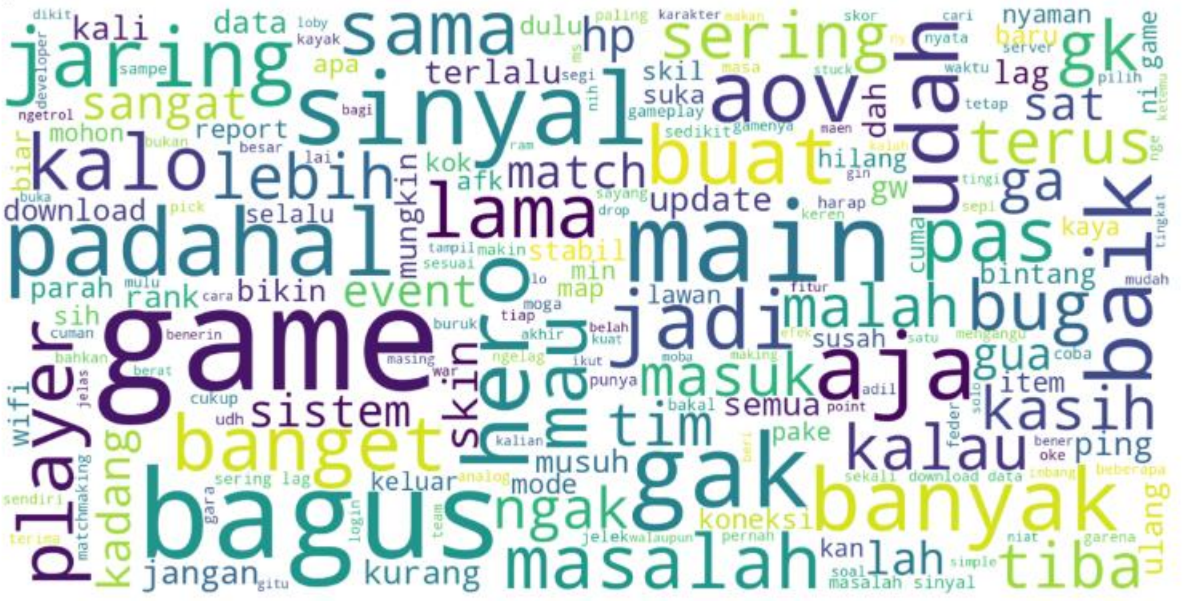
Mobile Legends

main
jaring
bagus
game
padahal
jadi
gak
sering
makin
lag
hero
aja
baik
buat
terus
banyak
monton
player
malah
update
data
sangat
tim
kalo
sinyal
sat
masalah
udah
download
musuh
tanding
sendiri
sama
lebih
imbang
wifi
hilang
keluar
skin
rank
sistem
baru
bug
mohon
kalau
pas
match
berat
masuk
tiba tiba
kurang
ga
beberapa
ping
semua
bintang
stabil
bikin
biar
temu
jangan
banget
lama
rekan tim
mau
grafik
fitur
ulang
soal
gameplay
sekarang
parah
penting
gamenya
turun
lah
tiba
tampil
bagi
nge
sekali
lawan
hp
sehinga
pilih
aku
utama
susah
keluh
ram
buruk
baik masalah
season
langsung
gerak
kadang
loading
mode offline
suka
login
performa
asa
legend
koneksi
gb
resource
afk
satu
skil
merah
event
bukan
tetap
dulu
memori
perlu
banyak bug
cuman
sat main
mode
nyaman main
nyaman
masa
naik

Wild Rift



AoV



Referensi:

https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm