LAPORAN PROJECT UAS

PEMROSESAN BAHASA ALAMI

Analisis Sentimen Pengunjung Hotel Dengan K-Nearest Neighbors

Studi Kaus Hotel Pop! Surabaya

Disusun Oleh :

Lu'luatul Maknunah	:	210411100048
Niswatul Sifa	:	210411100145
Desti Fitrotun Nisa	:	210411100182

PRODI TEKNIK INFORMATIKA

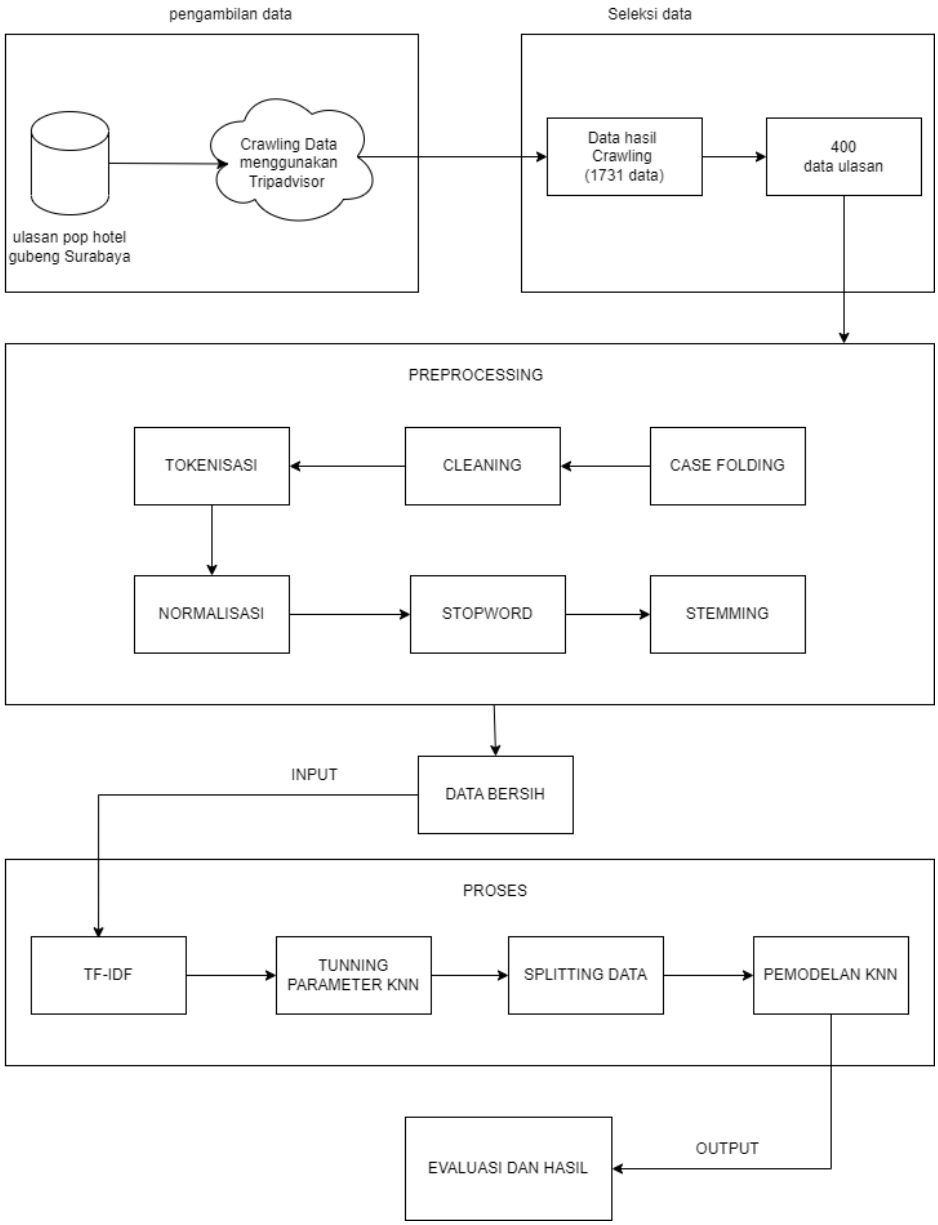
JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

2023

1. Arsitektur



Penjelasan :

a. Mengambil data ulasan dari Hotel Pop! Gubeng Surabaya.
b. Melakukan crawling data menggunakan TripAdvisor untuk mengumpulkan data ulasan. Sehingga didapatkan hasil data ulasan sebanyak 1700 ulasan yang kemudian disimpan dalam file berformat csv.
c. Kemudian dilakukan seleksi data dari total 1731 data yang diperoleh, sehingga diambil 400 data dengan diambil 200 ulasan positif dan 200 ulasan negatif yang diberikan pelabelan data secara manual.

d. Dari 400 data tadi akan dilakukan preprocessing terlebih dahulu.
e. Preprocessing

   Terdapat 5 tahapan preprocessing yang dilakukan, yaitu:

   - Case folding

     Merupakan proses mengubah semua huruf dalam data ulasan menjadi huruf kecil.
   - Cleaning

     Merupakan proses membersihkan dan mempersiapkan data teks sebelum dilakukan analisis atau pengolahan lebih lanjut seperti menghapus tanda baca (, . '?!), angka dan huruf tak berarti.
   - Tokenisasi

     Merupakan proses memecah data teks menjadi unit-unit yang lebih kecil seperti kata,frasa, atau kalimat. Unit-unit tersebut disebut dengan token.
   - Normalisasi

     Proses mengubah kata-kata yang salah atau typo menjadi kata yang benar atau sesuai dengan standar bahasa indonesia.
   - Stopwords

     Proses menghapus kata-kata yang umum dan tidak memiliki arti yang signifikan dalam analisis teks.
   - Stemming.

     Proses mengubah kata-kata dalam teks menjadi kata dasar, biasanya dilakukan dengan menghapus awalan atau akhiran kata sehingga menyisakan bentuk dasarnya.

f. Setelah tahapan preprocessing selesai, maka diperoleh data bersih hasil preprocessing yang merupakan data input untuk diproses ke tahapan selanjutnya.
g. TF-IDF (Term Frequency-Inverse Document Frequency)

   Selanjutnya masuk pada tahapan proses, yaitu menghitung bobot setiap kata dalam suatu dokumen berdasarkan seberapa penting kata tersebut dalam konteks keseluruhan korpus dokumen pada data ulasan

menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency).

h. Tunning Parameter

Langkah berikutnya adalah Tunning Parameter yaitu proses pengoptimalan nilai-nilai parameter dalam suatu model atau algoritma untuk mencapai performa yang lebih baik. Sebelum dilakukan pemodelan menggunakan algoritma KNN maka dilakukan tunning parameter terlebih dahulu untuk menentukan jumlah pembagian data uji dan data latih serta nilai k yang optimal agar menghasilkan nilai akurasi yang terbaik dari model tersebut.

i. Splitting Data

Setelah dilakukan tunning parameter maka hasil dari pembagian data uji dan data latih dengan akurasi terbaik akan ditetapkan untuk parameter pada splitting data. Biasanya pembagian data latih dan data uji yaitu 80:20, 70:30 dan 60:40.

j. Pemodelan KNN

Setelah didapatkan nilai K dengan hasil akurasi terbaik dari proses tunning parameter, maka langkah selanjutnya yaitu melakukan pemodelan KNN dengan memasukkan nilai k terbaik yang diperoleh dari hasil tunning parameter tadi.

k. Evaluasi dan Hasil

Kemudian langkah terakhir yaitu dilakukan proses pelatihan menggunakan data latih dari model yang telah dibuat tadi dengan melakukan prediksi dan menghitung akurasi. Sehingga didapatkan output berupa hasil akurasi, presisi, recal dan F1-score dengan jumlah TP, TN, FP,dan FN.

2. Code Program

2.1 Preprocessing

- Case folding

```
# mengubah semua huruf dalam data ulasan menjadi huruf
kecil menggunakan fungsi lower() yang disimpan dalam
fungsi casefolding.
```

```
def casefolding(ulasan):
    ulasan = ulasan.lower()
    return ulasan
# mengimplementasikan fungsi case folding pada data tepatnya pada kolom Ulasan
data['Ulasan'] = data['Ulasan'].apply(casefolding)
data.head(15)
```

- Cleaning

```
#proses cleansing remove regex (cleansing) seperti tanda baca dan angka angka
#menggunakan libary re
import re
import string
def cleaning(Ulasan):
    Ulasan = Ulasan.strip(" ")
    Ulasan = re.sub(r'[?|$|.|!_:")(-+,]', ' ', Ulasan)
    Ulasan = re.sub(r'\d+', ' ', Ulasan)
    Ulasan = re.sub(r"\b[a-zA-Z]\b", " ",Ulasan)
    Ulasan = re.sub('\s+',' ', Ulasan)
    return Ulasan
# mengimplementasikan fungsi cleaning pada data tepatnya pada kolom Ulasan
data['Ulasan'] = data['Ulasan'].apply(cleaning)
data.head(15)
```

- Tokenisasi

```
#tokenisasi atau memisahkan kalimat menurut spasi menggunakan class word_tokenize dari modul nltk.toknize
from nltk.tokenize import word_tokenize
```

```
def word_tokenize_wrapper(text):
    return word_tokenize(text)
# mengimplementasikan fungsi tokenisasi pada data tepatnya pada kolom Ulasan
data['Ulasan']=data['Ulasan'].apply(word_tokenize_wrapper)
data.head(15)
```

- Normalisasi

```
#normalisasi atau merubah kata yang typo ke dalam penulisan yang benar sesuai dengan yang ada di kamus normalisasi.xlsx
normalizad_word = pd.read_excel("normalisasi.xlsx")
normalizad_word_dict = {}
for index, row in normalizad_word.iterrows():
    if row[0] not in normalizad_word_dict:
        normalizad_word_dict[row[0]] = row[1]
def normalized_term(document):
    return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in document]
# mengimplementasikan fungsi normalisasi kata pada data tepatnya pada kolom Ulasan
data['Ulasan'] = data['Ulasan'].apply(normalized_term)
data.head(15)
```

- Stopwords

```
# Menghapus kata yang dianggap tidak mempunyai arti penting menggunakan modul nltk.corpus class stopwords
from nltk.corpus import stopwords
import nltk
nltk.download('stopwords')
```

```
nltk.download('punkt')

# Mendefinisikan stopword Bahasa Indonesia

stop_words = set(stopwords.words('indonesian'))

# Fungsi untuk menghapus stopword dari sebuah teks

def remove_stopwords(Ulasan):

    return [token for token in Ulasan if token not in stop_words]

# Menghapus stopword pada kolom ulasan

data['Ulasan']= data['Ulasan'].apply(remove_stopwords)

data.head(15)
```

- Stemming

```
#Mengubah kata menjadi kata dasar menggunakn modul sastrawi class stemmer factory

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

factory = StemmerFactory()

stemmer = factory.create_stemmer()

#fungsi yang akan digunakan untuk stemming data

def stem_text(text):

    stemmed_tokens = [stemmer.stem(token) for token in text]

    return ' '.join(stemmed_tokens) #Menggabungkan kata kembali

# Melakukan stemming pada kolom ulasan

data['Ulasan'] = data['Ulasan'].apply(stem_text)

#menyimpan data bersih atau data yang telah di lakukan preprocessing

data.to_csv('PBA.csv', index=False)
```

2.2 TF-IDF (Term Frequency-Inverse Document Frequency)

```
#MENGHITUNG TF-IDF dengan modul sklearn.feature extraction text
```

```
fromsklearn.feature_extraction.text import CountVectorizer
fromsklearn.feature_extraction.text import TfidfVectorizer
#melihat kemunculan kata
vectorizer = CountVectorizer()
k_kata=
vectorizer.fit_transform(data_clean['Ulasan'].astype('U'))
print(k_kata)
#menghitung bobot kata atau tf-idf
tfidf= TfidfVectorizer()
h_tfidf=tfidf.fit_transform(data_clean['Ulasan'].astype('U
'))
print(h_tfidf)
```

2.3 Tunning Parameter

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# X = fitur
# y = label
# Menentukan split data
test_sizes = [0.2, 0.3, 0.4]
# menentukan nilai K
nK_values = [1,2,3,4,5,6,7,8,9,10]
# perulangan test size dan nilai k
for test_size in test_sizes:
    for nK in nK_values:
        # SPLIT DATA (DATA UJI & LATIH)
        X_train,     X_test,     y_train,     y_test     =
train_test_split(h_tfidf,data_clean['Label'],test_size=tes
t_size, random_state=42)
        # Klasifikasi knn dengan nilai k
        knn = KNeighborsClassifier(n_neighbors=nK)
```

```
        # latih klasifikasi
        knn.fit(X_train, y_train)
        # melakukan prediksi pada data uji
        y_pred = knn.predict(X_test)
        #menghitung akurasi
        accuracy = accuracy_score(y_test, y_pred)
        # Cetak hasil
        print(f"Test Size: {test_size}, NILAI K: {nK},
Accuracy: {accuracy}")
```

2.4 Splitting Data

```
#Pembagian data
import collections, numpy
from sklearn.model_selection import train_test_split
#pembagian data X dan Y dengan data training 20%
X_train,X_test, y_train, y_test = train_test_split(h_tfidf,
data_clean['Label'], test_size=0.2, random_state=42)
#Melihat jumlah pembagian data
print("Jumlah Data Uji:", X_test.shape)
print("Jumlah Data Latih:",X_train.shape)
# JUMLAH DATA UJI NEGATIF DAN POSITIF
pos = (y_test == 'positif').sum()
neg = (y_test == 'negatif').sum()
#JUMLAH DATA LATIH NEGATIF DAN POSITIF
postrain = (y_train == 'positif').sum()
negtrain = (y_train == 'negatif').sum()
total = pos + neg
# MENCETAK JUMLAH DATA UJI NEGATIF DAN POSITIF, JUMLAH DATA
LATIH NEGATIF DAN POSITIF
print("Jumlah data uji dengan sentimen positif:", pos)
print("Jumlah data uji dengan sentimen negatif:",neg)
```

```
print("Jumlah data latih dengan sentimen positif:", postrain)

print("Jumlah data latih dengan sentimen negatif:",negtrain)

#melihat banyak nya keseluruhan data positif & negatif pada data label

data_clean['Label'].value_counts()
```

2.5 Pemodelan KNN

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

from sklearn.metrics import classification_report

from sklearn.metrics import confusion_matrix

from sklearn.neighbors import KNeighborsClassifier

clf = KNeighborsClassifier(n_neighbors=5).fit(X_train, y_train)

predicted = clf.predict(X_test)
```

2.6 Evaluasi dan Hasil

```
print(f'confusion matrix:\n {confusion_matrix(y_test, predicted)}')

print('================================================\n')

tn,fp,fn,tp = confusion_matrix(y_test, predicted).ravel()

print("TN:", tn)

print("FP:", fp)

print("FN:", fn)

print("TP:", tp)

print(classification_report(y_test,predicted,zero_division=0))

print('================================================\n')

print("Hasil Klasifikasi Sentimen Analisis:")

print("Accuracy:" , accuracy_score(y_test,predicted))
```

```
print("Precision:"   ,   precision_score(y_test,predicted,
average="binary", pos_label="positif"))

print("Recall:"      ,      recall_score(y_test,predicted,
average="binary", pos_label="positif"))

print("f1_score:"       ,       f1_score(y_test,predicted,
average="binary", pos_label="positif"))

print("error_rate:", 1-accuracy_score(y_test,predicted))
```

3. Skenario Pengujian

Pada tahapan skenario uji coba dilakukan pengujian untuk pembagian data latih dan data uji serta nilai k. adapun tahapan skenario uji coba yaitu dengan dilakukan perulangan sebanyak 10 kali untuk masing-masing test_size (data uji yang akan digunakan) dimana untuk test_size yang digunakan 0.2 , 0.3 dan 0.4 . Sedangkan untuk nilai k yang ingin di uji coba yaitu nilai k dari 1,2,3,4,5,6,7,8,9, dan 10. Sehingga di dapatkan hasil sebagai berikut:

Text_size	**Jumlah k**	**Hasil akurasi**
0.2	1	0.7625
	2	0.75
	3	0.825
	4	0.85
	5	0.8625
	6	0.825
	7	0.85
	8	0.85
	9	0.85
	10	0.825

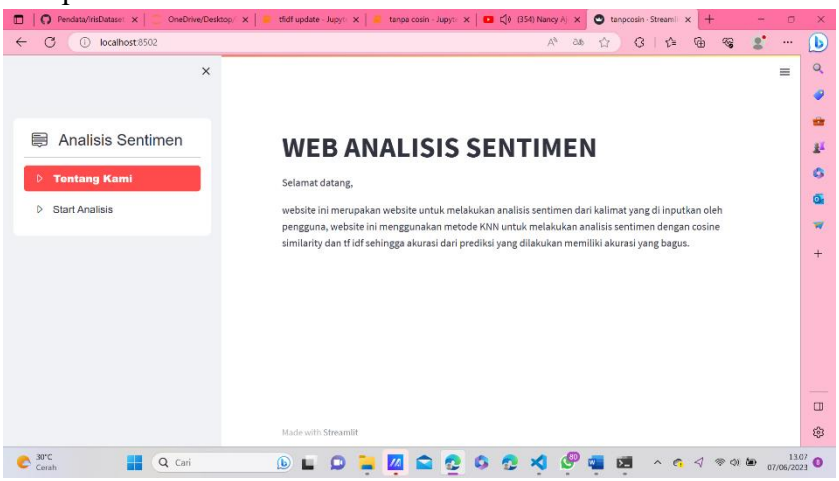
Dari hasil uji coba yang telah dilakukan dengan Tunning Parameter maka di dapatkan hasil akurasi yang terbaik dengan test_size = 0.2 dan nilai k = 5 mendapatkan hasil akurasi 0.8625.

Text_size	Jumlah k	Hasil akurasi
0.3	1	0.7166666666666667
	2	0.75
	3	0.825
	4	0.825
	5	0.85
	6	0.8166666666666667
	7	0.8166666666666667
	8	0.8
	9	0.7833333333333333
	10	0.8

Text_size	Jumlah k	Hasil akurasi
0.4	1	0.71875
	2	0.74375
	3	0.85
	4	0.80625
	5	0.80625
	6	0.8
	7	0.8
	8	0.8125
	9	0.79375
	10	0.8125

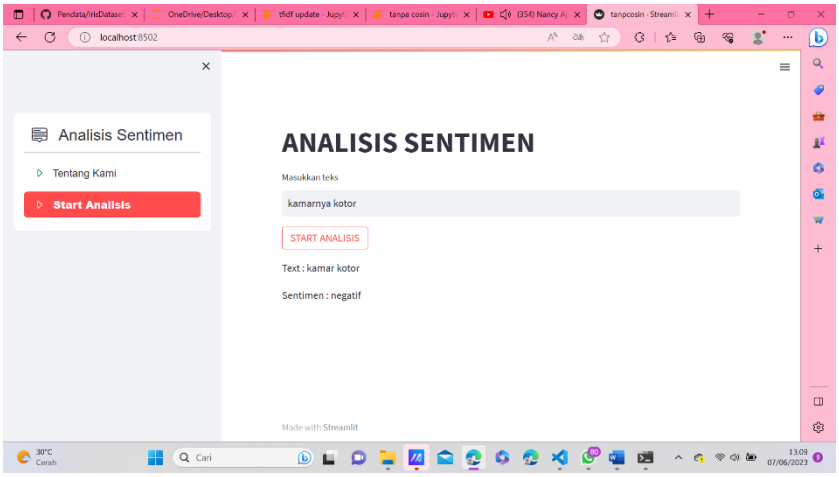
4. Implementasi Interface

4.1 Tampilan Dashboard

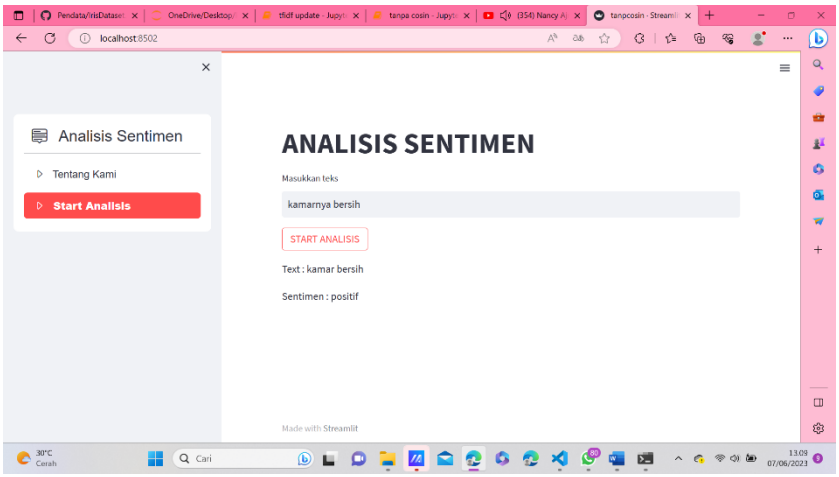


4.2 Tampilan Halaman Analisis

Melakukan analsisis pada kata ‘Kamarnya kotor’.



Melakukan analsisis pada kata ‘Kamarnya bersih’



5. Analisa dan Hasil Uji Coba

5.1 Data False Positif dan Data False Negatif

Dari hasil evaluasi model KNN diperoleh data false positif sebanyak 10 data dan false negatif sebanyak 1 data.

Berikut merupakan data false positif dan false negatif :

Index	Ulasan	Label	Prediksi
209	hotel nyaman lengkap fasilitas kolam renang anak anak keponakan karyawan helpfull kamar bersih kamar mandi nya tabung nyaman mandi nya over bagus	negatif	positif
210	kali nginap kamar pas rekomendasi suka traveling bonus kolam renang view pagi nya dukung abis hujan hehe betah	negatif	positif
278	bersih kamar bersih resto bersih fasilitas kamar kurang lemari ac dingin makan lumayan jenis makan layan check in cepat tugas ramah	negatif	positif
376	sarap pagi tamu sarap front office check out durasi sarap sisa menit jam ludes	negatif	positif
385	kali inap hotels negeri salah supervisor pop hotel amatir seragam kena sandal celana pendek tegur tamu foto lobby hotel heran kaget era digital mana format horisontal marketing butuh dongkrak awareness optimal low anggar salah supervisor amatir larang tamu foto niat publikasi pop hotel socialmedia gaya sombong sopan tamu kapok tinggal pop hotel sbg karyawan hotel wajib wow layan tulus layan tamu tindak spt tugas kasar paham	negatif	positif
391	alam trip surabaya tarik harga nya jangkau stasiun gubeng layan nya wifinya loading nya suka hilang sinyal	negatif	positif
271	fasilitas bagus tidur nyenyak saran kamar mandi pengap mandi nya hehehehee inti joss dahh bagus foto sunset hehee sukses dah hotel pop	negatif	positif
296	kamar sempit lemari bersih lebih nya guling kamar sarap menu layan nya oke minimarket atm hotel lobby nya unik warn warni kesan ceria the best	negatif	positif
223	staycation pop hotel gubeng haris gubeng kecewa layan staf nya satpam resepsionis hk layan jempol check in kendala lampu mati ac dingin dan lain lain salut sigap ramah layan bikin kangen ken good job	negatif	positif
268	nyaman bagus hospitalitynya juara makan enak kamar mandi nya ya mungin tingkat layan nya maximal time kunjung disisni	negatif	positif
42	harga tawar banding fasilitas kamar dar business trip perlu hari backpacking bersih kamar kamar mandi recommended	positif	negatif

Dari 10 data FP (False Positif) dan 1 data FN (False Negatif)diatas maka kami melakukan beberapa analisa yang mempengaruhi terhadap kesalahan prediksi yang terjadi diantaranya yaitu :

a. Disebabkan karena pada tahap preprocessing, yaitu pada penggunaan Stopwords, kami menggunakan modul Sastrawi yang menghapus kata-kata seperti "Tidak, kurang, aku, kamu, dll." Oleh karena itu, dalam konteks ini, kata "Tidak" yang sebenarnya memiliki makna "negatif" juga terhapus karena proses stopword. Hal ini mengakibatkan ulasan yang seharusnya diklasifikasikan sebagai "negatif" karena mengandung kata "Tidak" justru diprediksi sebagai "positif".
b. Banyaknya ulasan yang mengandung beberapa kata dalam bahasa inggris sehingga saat dilakukan proses klasifikasi, kata dalam bahasa inggris tadi diklasifikasikan sebagai “negatif”.
c. Banyaknya ulasan yang memberikan komentar negatif dan juga komentar positif dalam satu kali ulasan. Sehingga dalam satu dokumen ulasan setelah melalui tahapan preprocessing dan TF-IDF banyak kata-kata positif dan negatif yang terdapat dalam satu dokumen, hal ini dapat menyebabkan kebingungan saat dilakukan prediksi.

5.2 Saran

Dari beberapa hasil analisa kami diatas ada beberapa saran untuk pengembangan selanjutnya yaitu:

a. Melakukan penanganan terhadap Masalah Stopwords yaitu seperti dengan membuat daftar stopwords yang disesuaikan dengan konteks data kami. Pastikan kata-kata penting seperti "Tidak" jika berpengaruh terhadap pengklasifikasian tidak terhapus secara otomatis.
b. Melakukan penanganan kata dalam bahasa inggris yaitu dengan memperluas model atau algoritma klasifikasi bisa dengan cara mengenali dan memproses kata-kata dalam bahasa Inggris secara lebih baik