LAPORAN AKHIR PROYEK

Sentimen Analisis Terhadap Review Kepuasan Pelanggan Shopee Berbahasa Inggris Menggunakan Algoritma SVM (Support Vector Machine)

Disusun oleh:

12S17011	**Astri Monica Sianturi**
12S17013	**Mega Sari Pasaribu**
12S17046	**Pebri Sangmajadi Sinaga**

11S4037 - PEMROSESAN BAHASA ALAMI

PROGRAM STUDI SARJANA SISTEM INFORMASI

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

NOVEMBER 2020

Daftar Isi

Daftar Isi i
Daftar Gambar ii
Daftar Tabel iii
1.1. Latar Belakang 1
1.2. Tujuan 2
1.3. Manfaat 2
1.4. Ruang Lingkup 2
2. Isi 4
2.1 Analisis Data dan Metode 4
2.2 Desain 6
2.3 Implementasi 8
1. Load Library 8
2. Load Data 9
3. Exploratory Data Analysis 10
4. Pre-Processing Data 11
5. Feature Extraction 14
6. TF-IDF 15
7. Implementasi SVM 15
8. Evaluasi Model 17
2.4 Hasil 18
3. Penutup 20
3.1 Pembagian Tugas dan Tanggung Jawab 20
3.2 Kesimpulan 21
3.3 Saran 21
Daftar Pustaka 22

Daftar Gambar

Gambar 1 Tahapan Pengerjaan Proyek.....6

Daftar Tabel

Tabel 1 Atribut pada Dataset yang Digunakan 4
Tabel 2 Pembagian Tugas dalam Kelompok 20

1. Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang pengerjaan proyek, tujuan, manfaat, dan ruang lingkup yang ingin dicapai dalam proyek.

1.1. Latar Belakang

Kebutuhan masyarakat semakin meningkat seiring dengan perkembangan teknologi yang mempengaruhi proses transaksi jual beli dari konvensional berubah menjadi modern dengan pemanfaatan internet. Perubahan transaksi dengan memanfaatkan internet disebut dengan *e-commerce*. *E-commerce* memungkinkan pengguna untuk melakukan pembelian barang dimanapun dan kapanpun. Perkembangan *e-commerce* yang sangat cepat menjadi peluang bagi produsen untuk memasarkan dan mempromosikan produk kepada konsumen. Saat ini telah banyak *e-commerce* yang beredar di kalangan masyarakat. Salah satu *e-commerce* yang sering digunakan masyarakat adalah Shopee.

Shopee adalah salah satu platform e-commerce yang terkemuka di Asia Tenggara dan Taiwan. Shopee diluncurkan pada tahun 2015. Shopee merupakan *platform* yang memberikan pelanggan mengenai pengalaman belanja online yang mudah, aman, dan cepat disertai metode pembayaran yang baik. Pada Shopee terdapat banyak produk yang tersedia, misalnya makanan, minuman, barang-barang elektronik, produk kesehatan dan kecantikan, pakaian, dan lain-lain. Shopee juga memungkinkan pengguna untuk melakukan pengisian pulsa, pembelian paket data, pembelian token listrik, dan lain-lain.

Shopee menyediakan banyak fitur kepada pengguna. Salah satu fitur yang diberikan adalah *review* terhadap transaksi yang dilakukan. Pada Shopee terdapat *review* dari pelanggan yang telah membeli suatu produk untuk memberikan komentarnya terhadap produk tersebut. Dengan adanya *review* akan menjadi masukan terhadap pihak Shopee serta para pembeli yang ingin membeli produk tersebut sehingga membantu pengambilan keputusan untuk membeli barang berdasarkan *review* para pengguna. Pada *review* juga terdapat banyak kata-kata yang menjadi masukan untuk pihak Shopee atau penjual barang agar dapat meningkatkan kualitas barang berdasarkan review para pengguna. Namun, banyaknya review yang ada terkadang pembeli malas untuk membaca satu per satu kolom ulasan tersebut dan hanya melihat komen-komen teratas dengan berbagai rating yang tinggi. Oleh karena itu, *review* dari pelanggan harus diolah menjadi sebuah informasi untuk memudahkan calon pembeli dan pihak Shopee/penjual dalam mengambil keputusan dengan cara melakukan analisis sentimen.

Sentiment analysis merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung di dalam suatu kalimat opini. [1]

Pada proyek ini, tim ingin membagi suatu opini ke dalam opini positif dan opini negatif pada *review* barang di Shopee. Penelitian ini bertujuan untuk menghasilkan informasi sentimen mengenai *review* pengguna terhadap suatu barang yang dipesan di Shopee yang mengarah ke sentimen positif dan negatif menggunakan algoritma *Support Vector Machine* yang merupakan salah satu teknik pembelajaran dari *Text Mining*.

1.2. Tujuan

Adapun tujuan dari pengerjaan proyek dengan teknik *text mining* ini adalah untuk melakukan analisis sentimen terhadap *review* produk yang diberikan pelanggan Shopee yang dapat memudahkan pihak Shopee/penjual dan pembeli dalam mengetahui kepuasan pelanggan terhadap suatu produk yang dapat dimanfaatkan dalam mengambil keputusan.

1.3. Manfaat

Adapun manfaat yang diperoleh dari pengerjaan proyek sentimen analisis terhadap *review* kepuasaan pelanggan Shopee yang ditujukan untuk:

a. Calon Pembeli
    1. Mengetahui ulasan pelanggan terhadap produk.
    2. Membantu pengambilan keputusan apakah pembelian produk akan dilakukan berdasarkan *review* dari pelanggan.
b. Pihak Shopee/Penjual
    1. Mengetahui kepuasan pelanggan terhadap produk yang ditawarkan.
    2. Membantu dalam meningkatkan kualitas barang berdasarkan *review* pelanggan

1.4. Ruang Lingkup

Adapun batasan dari pengerjaan proyek sentimen analisis terhadap *review* kepuasaan pelanggan Shopee adalah sebagai berikut.

1. Algoritma yang digunakan dalam melakukan analisis sentimen adalah *Support Vector Machine* (SVM).
2. Literatur yang digunakan sebagai ruang lingkup pengerjaan proyek ini adalah *review* produk pelanggan Shopee berbahasa Inggris.

3. Data yang digunakan adalah data *review* dari pelanggan Shopee terhadap suatu barang yang dapat diakses di *website* Kaggle.com.

2. Isi

Bab ini berisi penjelasan mengenai analisis data dan metode, desain dari pemrosesan bahasa alami, implementasi sentimen analisis, dan hasil dari implementasi.

2.1 Analisis Data dan Metode

Pada subbab ini akan dibahas mengenai analisis data dan metode yang digunakan dalam mengerjakan proyek sentimen analisis terhadap review kepuasaan pelanggan shopee.

2.1.1 Analisis Data

Pada proyek ini, data yang digunakan merupakan Store Transaction Data (https://www.kaggle.com/shymammoth/shopee-reviews) yang diambil dari *kaggle* untuk mengidentifikasi review kepuasaan dari pelanggan Shopee terhadap suatu produk. Dataset menyimpan data mengenai pendapat pelanggan shopee mengenai kepuasannya belanja suatu produk pada *platform* Shopee.

Data yang akan digunakan merupakan data ulasan pelanggan Shopee. Dataset pada kaggle berukuran 131.28 MB yang dikemas dalam format CSV. Data ini terdiri dari tiga atribut yaitu, label, text, dan Sentimen. Dataset pada kaggle terdiri dari 1502575 baris. Namun pada pengerjaan proyek sentimen analisis ini, kami hanya menggunakan 10000 baris data review tersebut. Data yang digunakan berukuran 234,5 KB.

Tabel 1 Atribut pada Dataset yang Digunakan

No	Attribute	Non-Null Count	Data Type	Tipe Atribut
1	label	10000 non-null	int64	Nominal
2	text	10000 non-null	Object	Categorical
3	Sentimen	10000 non-null	Object	Categorical

Berdasarkan sifatnya, data dapat terbagi menjadi dua jenis, yaitu data kualitatif (non-metrik) dan data kuantitatif (metrik). Data kualitatif dapat disebut data yang bukan berupa angka. Pada data kualitatif tidak bisa dilakukan operasi matematika, seperti penambahan, pengurangan, perkalian dan pembagian. Data kuantitatif dapat disebut sebagai data berupa angka. Berbagai jenis operasi matematika dapat dilakukan pada data kuantitatif. [2]

Data yang terdapat pada dataset terdiri atas data kuantitatif dan kualitatif. Data kuantitatif merupakan data yang dapat diukur (*measurable*) atau dapat dihitung sebagai angka atau bilangan. Data tersebut dapat berupa bilangan diskrit atau bilangan kontinu. Data kuantitatif memiliki kecenderungan dapat dianalisis dengan cara atau teknik statistik. Data yang termasuk kuantitatif pada *dataset* adalah label.

Data kualitatif merupakan data yang berbentuk kata, kalimat atau gambar. Data kuantitatif dapat juga disebut data kategori, yakni nominal dan ordinal. Data yang termasuk kualitatif pada dataset adalah text dan sentimen.

2.1.2 Analisis Metode

Pada proyek ini, pengembangan sentimen analisis dilakukan dengan menggunakan metode Support Vector Machine. SVM (Support Vector Machine) merupakan salah satu metode klasifikasi dengan menggunakan metode machine learning (*supervised learning*) yang memprediksi kelas berdasarkan pola dari hasil proses *training* yang diciptakan oleh Vladimir Vapnik. Klasifikasi dilakukan dengan garis pembatas (*hyperplane*) yang memisahkan antara kelas opini positif dan opini negatif. Secara intuitif, suatu garis pembatas yang baik adalah yang memiliki jarak terbesar ke titik data pelatihan terdekat dari setiap kelas, karena pada umumnya semakin besar margin, semakin rendah error generalisasi dari pemilah. Margin adalah jarak dari suatu titik vektor di suatu kelas terhadap *hyperplane*. [3]
SVM dapat diterapkan dalam berbagai bidang seperti *face detection, text and hypertext categorization* serta *classification of images*. Yang menjadi keuntungan dari *Support Vector Machine* adalah:

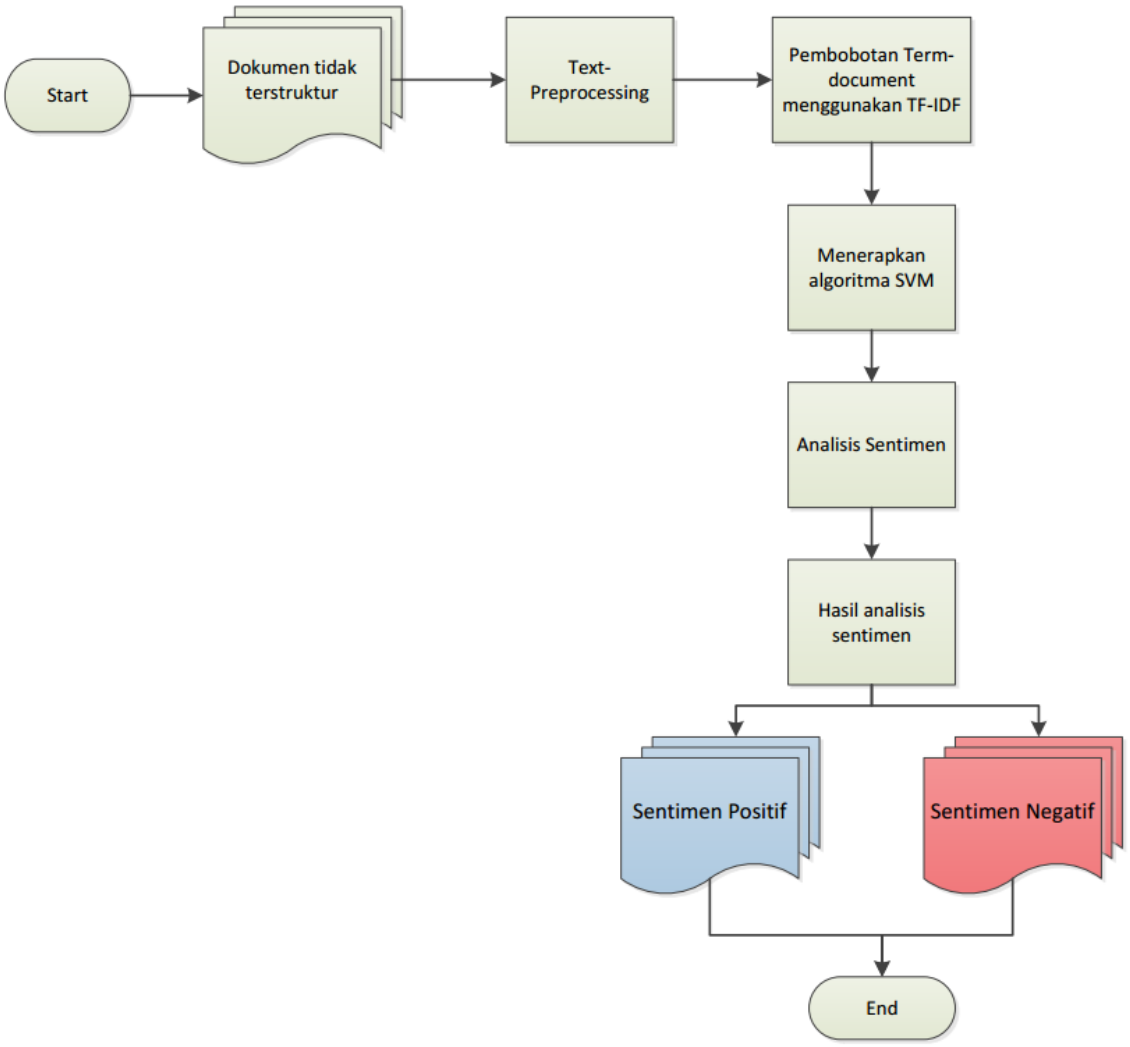
1. Efektif dalam ruang dimensi tinggi.
2. Efektif dalam kasus dimana jumlah dimensi lebih besar dari jumlah sampel.
3. Menggunakan subset poin pelatihan dalam fungsi keputusan *(support vector)* sehingga dapat menghemat memori.
4. Fungsi kernel yang berbeda dapat digunakan untuk pengambilan keputusan.

Sedangkan yang menjadi kerugian dari Support Vector Machines adalah sebagai berikut:

1. Jika jumlah fitur jauh lebih besar dari jumlah sampel, menghindari *overfitting* dalam memilih fungsi kernel dan istilah regularisasi sangat penting.
2. SVM tidak memberikan estimasi probabilitas secara langsung, estimasi tersebut dihitung dengan menggunakan *five-fold cross validation.*

2.2 Desain

Tahapan pengerjaan proyek analisis sentimen yang akan dilakukan dengan teknik *text mining* untuk menganalisis sentimen terhadap *review* produk menggunakan algoritma *Support Vector Machine* (SVM) adalah sebagai berikut:



Gambar 1 Tahapan Pengerjaan Proyek

1. **Membaca Dokumen tidak Terstruktur**

   Unstructured Documents merupakan langkah awal dalam desain yang digunakan untuk membaca keseluruhan *input document* yang terdapat dalam dataset *review* produk pelanggan Shopee. *Dataset* yang terdapat dalam *review* produk pelanggan Shopee masih merupakan *dataset* yang terdapat kalimat atau kata yang belum terstruktur dengan benar sehingga masih perlu dilakukan pada teks pra pemrosesan pada *dataset* tersebut.

2. Teknik Teks Pra Pemrosesan

Pra pemrosesan teks merupakan tahap yang dilakukan untuk mengolah teks sebelum digunakan dalam proses selanjutnya. Pra Pemrosesan teks dilakukan untuk menghilangkan data yang tidak dibutuhkan ataupun data yang terdapat dalam teks yang tidak sesuai dengan proses yang dibutuhkan. Penerapan pra pemrosesan teks ini akan menggunakan bahasa pemrograman *Python*. Beberapa tahapan untuk melakukan pra pemrosesan teks adalah *case folding*, *tokenization*, *stopwords removal*, *stemming*, dan *tagging*.

Berikut ini adalah tahapan proses penerapan *text mining* menurut Feldman dan Sanger (2007): [4]

1. *Case folding* yaitu tahap untuk mengkonversi keseluruhan teks huruf kapital (*uppercase*) pada sebuah kalimat menjadi huruf kecil (*lowercase*) serta menghilangkan seluruh karakter yang dianggap tidak valid seperti angka, tanda baca, dan *Uniform Resources Locator* (URL). Pengimplementasian *case folding* tidak menggunakan library khusus pada Python namun menggunakan modul yang disediakan oleh Python itu sendiri.
2. *Tokenizing* yaitu proses memecah sebuah kalimat dalam dokumen berdasarkan tiap kata yang menyusunnya, atau dapat dikatakan memecah kalimat ke dalam satuan kata yang disebut dengan token. Cara untuk membedakan kata tersebut yaitu dengan menggunakan pemisah (delimiter) seperti spasi, enter, tabulasi, petik tunggal (‘), titik (.), semikolon (;), titik dua (:) dan lain sebagainya. Pengimplementasian *tokenizing* dapat dilakukan dengan menerapkan library NLTK.
3. *Stemming* yaitu merubah berbagai kata berimbuhan menjadi kata dasar. Stemming bertujuan untuk menghilangkan imbuhan-imbuhan seperti awalan kata (*prefixes*), sisipan kata (*infixes*), akhiran kata (*suffixes*) serta awalan dan akhiran kata (*confixes*) pada kata turunan. yang terdapat dalam kata. Tahap ini pada umumnya dilakukan untuk teks dengan bahasa Inggris, karena teks dengan bahasa Inggris memiliki struktur imbuhan yang tetap.
4. *Stopwords removal* bertujuan untuk mengurangi kata yang kurang penting, seperti kata yang, di, ke, dalam, dan, dengan, ini, itu, untuk, dan lain sebagainya, sehingga mempermudah dan mempercepat pengolahan dokumen.
5. *Tagging* yaitu merubah berbagai kata dalam bentuk lampau menjadi kata awalnya, tahap ini pada umumnya dilakukan untuk teks dengan bahasa Inggris atau bahasa lainnya yang memiliki bentuk lampau.

3. Term-document Matrix Using TF-IDF

Pengukuran tingkat *similarity* antara 2 *object* atau *document* harus dimodelkan dalam bentuk tertentu. Pada representasi dokumen *bag of words*, setiap kata yang terdapat dalam dokumen akan dihitung bobotnya. Cara termudah yang sering digunakan adalah dengan menghitung frekuensi kemunculan kata atau *term* dalam sebuah dokumen serta menggunakannya sebagai bobot dalam *term* tersebut. Data yang berhasil melewati tahap *preprocessing* harus berbentuk angka atau numerik. Pengubahan data menjadi numerik dilakukan dengan menggunakan algoritma pembobotan *Terms Frequency - Inverse Document Frequency* (TF- IDF) yang merupakan algoritma yang membahas tentang *Term Frequency* dengan *Inverse Document Frequency*. [5]

4. Support Vector Machine (SVM) Algorithm

Algoritma yang digunakan dalam menganalisis sentimen yaitu dengan menggunakan *Support Vector Machine* (SVM). Metode *Support Vector Machine* (SVM) adalah metode klasifikasi *linier* dengan menemukan *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada *input space* (Saifinnuha, 2015). Prinsip dasar dari SVM adalah pengklasifikasi *linear*, kemudian dikembangkan menjadi pengklasifikasi non*linear* dengan memasukkan *kernel* trik pada ruang dimensi tinggi. [6]

Hasil dari *preprocessing* tersebut akan dilakukan klasifikasi dengan *Support Vector Machine*. Nilai bias yang didapat pada proses ini akan digunakan untuk analisis sentimen.

2.3 Implementasi

Pada subbab ini akan dibahas mengenai implementasi dari sentimen analisis yang dikembangkan oleh tim proyek.

1. **Load Library**

   Pada implementasi sentimen analisis review dari kepuasaan pelanggan Shopee diperlukan beberapa library untuk mendukung keberhasilan program. Berikut adalah library yang digunakan pada sentimen analisis yang dikembangkan oleh tim.

```
import nltk
import string
import re
import pickle
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt


from sklearn.svm import SVC, LinearSVC
from sklearn import svm
from sklearn.model_selection import train_test_split, KFold, cross_val_score,cross_val_predict
from sklearn.decomposition import PCA
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.metrics import classification_report
from sklearn.preprocessing import LabelEncoder
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import KFold, cross_val_score, train_test_split
import nltk
from nltk.tokenize import word_tokenize
from nltk.classify.scikitlearn import SklearnClassifier
```

2. Load Data

Sebelum implementasi model analisis sentimen, tim terlebih dahulu memuat (*load*) data yang akan digunakan. Berikut adalah kode program yang digunakan untuk meload data.

```
1  def load_data(path):
2      data_review = pd.read_csv(path, encoding='utf-8')
3      return  data_review
```

```
1  data_review = pd.read_csv('./shopee_reviews_data.csv')
```

Data yang di-*load* memiliki format_csv dan data diinisialisasi dengan data_review. Setelah data dimuat, maka tim melihat berapa jumlah data dan kolom yang terdapat pada dataset. Jumlah baris data pada dataset sebanyak 10000 baris dan kolom sebanyak tiga.

```
1  # Menampilkan jumlah data dan kolom data pada dataset
2  print(data_review.shape)
```

```
(10000, 3)
```

Kemudian tim menampilkan data yang terdapat pada dataset. Data yang ditampilkan adalah lima data paling atas dan lima data paling bawah.

```
# menampilkan 10 data paling atas
data_review.head(5)
```

	label	text	Sentimen
0	5	Looks ok. Not like so durable. Will have to us...	Positive
1	5	Tried, the current can be very powerful depend...	Positive
2	5	Item received after a week. Looks smaller than...	Positive
3	5	Thanks!!! Works as describe no complaints. Not...	Positive
4	5	Fast delivery considering it's from overseas a...	Positive

```
# menampilkan 5 data paling bawah
data_review.tail()
```

	label	text	Sentimen
9995	5	Hi Product does not work properly :)	Positive
9996	5	Best purchase ever.. never regret and item cam...	Positive
9997	5	Suction is good. It will not fall. Very happy ...	Positive
9998	5	Very goooood suction and very fast delivery	Positive
9999	5	Item received in good condition. Yet to try out.	Positive

3. Exploratory Data Analysis

Setelah data selesai di-*load*, maka data dieksplor terlebih dahulu. Tujuan dilakukannya eksplorasi data adalah untuk mengetahui kondisi atau kualitas dari data.

Pada dataset yang digunakan terdapat data numerik yaitu label. Tim akan melihat analisis statistik dari data label tersebut.

```
# menampilkan info deskripsi dataset
data_review.describe()
```

	label
count	10000.000000
mean	4.764600
std	0.712066
min	1.000000
25%	5.000000
50%	5.000000
75%	5.000000
max	5.000000

Setelah itu, informasi umum pada dataset ditampilkan. Dataset terdiri dari 3 atribut yaitu label, text, sentimen. Total data keseluruhan adalah 10.000 data. Memori penyimpanan yang digunakan oleh dataset adalah 234,5 KB.

```
# menampilkan info dataset
data_review.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 3 columns):
label       10000 non-null int64
text        10000 non-null object
Sentimen    10000 non-null object
dtypes: int64(1), object(2)
memory usage: 234.5+ KB
```

Data yang sudah digunakan load sebelumnya akan diperiksa apakah pada dataset tersebut terdapat data yang duplikat atau berulang. Hasil yang diperoleh adalah bahwa pada dataset yang digunakan tidak terdapat data yang duplikat.

```
1 #memeriksa duplicate data
2 data_duplicate = data_review[data_review['text'].duplicated()]
3 print(f'No. of duplicate reviews on train data: {data_duplicate.shape[0]}')
```

```
No. of duplicate reviews on train data: 0
```

4. Pre-Processing Data

Pra pemrosesan teks merupakan tahap yang dilakukan untuk mengolah teks sebelum digunakan dalam proses selanjutnya. Pra Pemrosesan teks dilakukan untuk menghilangkan data yang tidak dibutuhkan ataupun data yang terdapat dalam teks yang tidak sesuai dengan proses yang dibutuhkan. Penerapan pra pemrosesan teks ini akan menggunakan bahasa pemrograman *Python*. Berikut beberapa tahapan pra pemrosesan teks yang diterapkan pada sentimen analisis.

NER (Named-Entity Recognition) digunakan untuk mencari dan mengelompokkan entitas dalam teks ke dalam kategori yang ditetapkan, misalnya nama orang, organisasi, lokasi, ekspresi, dll.

```
def NER(review):
    for i in range(len(review)):
        text = review.text.iloc[i]
        for sent in nltk.sent_tokenize(text):
            for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(sent))):
                if hasattr(chunk, 'label') and chunk.label:
                    if chunk.label() == 'ORGANIZATION' or  chunk.label() == 'PERSON' or
                    chunk.label() == 'DATE' or  chunk.label() == 'LOCATION':
                        name_value = ' '.join(child[0] for child in chunk.leaves())
                        text = text.replace(name_value, "")
                        review.text.iloc[i] = text
    return review
```

Case folding digunakan untuk mengkonversi keseluruhan teks huruf kapital (*uppercase*) pada sebuah kalimat menjadi huruf kecil (*lowercase*) serta menghilangkan seluruh karakter yang dianggap tidak valid seperti angka, tanda baca, dan *Uniform Resources Locator* (URL).

```
def case_folding(review):
    for i in range(len(review)):
        text = review.text[i].lower()
        review.text.iloc[i] = text
    return review
```

Remove punctuation digunakan untuk menghapus semua tanda baca dari string atau ulasan pelanggan.

```
def remove_punctuation(review):
    remove = string.punctuation
    for i in range(len(review)):
        for kata in remove:
            text = review.text[i].replace(kata,"")
            review.text.iloc[i] = text
    return review
```

Stemming digunakan untuk menghilangkan imbuhan-imbuhan seperti awalan kata (*prefixes*), sisipan kata (*infixes*), akhiran kata (*suffixes*) serta awalan dan akhiran kata (*confixes*) pada kata turunan. yang terdapat dalam kata.

```
def stemming(review):
    ps = PorterStemmer()
    for i in range(len(review)):
        text = review.text.iloc[i]
        text = ps.stem(text)
        review.text.iloc[i] = text
    return review
```

Stopwords removal bertujuan untuk mengurangi kata yang kurang penting, seperti kata

yang, di, ke, dalam, dan, dengan, ini, itu, untuk, dan lain sebagainya, sehingga mempermudah dan mempercepat pengolahan dokumen.

```
def stop_removal(review):
    from nltk.tokenize import sent_tokenize, word_tokenize
    cachedStopWords = set(stopwords.words("english"))
    for i in range(len(review)):
        text = review.text.iloc[i]
        teks =" ".join([word for word in text.split() if word not in cachedStopWords])
        review.text.iloc[i] = teks
    return review
```

Selanjutnya adalah lemmatization, lemmatization adalah proses memetakan token ke dalam bentuk dasar yaitu *lemma*. Proses *lemmatization* mengubah kata ke bentuk dasarnya sesuai dengan kata-kata yang terdapat di kamus.

```
1  def lemmatization (review):
2      lm = WordNetLemmatizer()
3      for i in range(len(review)):
4          text = review.text.iloc[i]
5          text = lm.lemmatize(text)
6          review.text.iloc[i] = text
7      return review
```

Pada program berikut merupakan penerapan semua fungsi pemrosesan data yang telah didefinisikan sebelumnya. Hasil yang akan ditampilkan adalah hasil lematisasi karena lematisasi adalah proses terakhir yang menampung seluruh output tahap-tahap sebelumnya kemudian kemudian menjadi input tahap lematisasi lalu diproses dan menghasilkan output yaitu hasil dari tahap lematisasi tersebut.

```
1  def preprocessing_data(review):
2      hasil_ner = NER(review)
3      hasil_case_folding = case_folding(hasil_ner)
4      hasil_remove_punctuation = NER(review)
5      hasil_stop_removal = stop_removal(hasil_remove_punctuation)
6      hasil_stemming = stemming( hasil_stop_removal)
7      hasil_lemmatization = lemmatization(hasil_stemming)
8      return hasil_lemmatization
```

Selanjutnya akan ditampilkan data hasil preprocessing yang sudah dilakukan sebelumnya. Berikut ini adalah data hasil preprocessing.

```
clean_data.text
```

```
0       looks ok  like durable  use recommend others w...
1       tried  current powerful depending setting  dar...
2       received week  looks smaller expected  can t w...
3       thanks    works describe complaints  really ex...
4       fast delivery considering it s overseas tried ...
                              ...
9995                          hi product work properly
9996    best purchase ever   never regret item came me...
9997                  suction good  fall  happy purchase
9998              goooood suction fast delivery   appreci
9999                received good condition  yet try out
Name: text, Length: 10000, dtype: object
```

Data yang sudah diproses tersebut selanjutnya kami simpan ke file baru yaitu file dengan format excel. Hal ini berguna agar data terpisah dari dataset yang belum dilakukan proses preprocessing.

```
hasil_preprocessing= data_review.to_excel('data_setelah_preprocessing.xlsx', encoding='utf-8')
new_data = pd.read_excel('./data_setelah_preprocessing.xlsx')
```

5. Feature Extraction

Pada *feature extraction*, hal yang pertama perlu diketahui adalah konsep model bag-of-words. *Model bag-of-words* terutama digunakan sebagai alat pembuatan fitur. Setelah mengubah teks menjadi "*bag-of-words*", kita dapat menghitung berbagai ukuran untuk mengkarakterisasi dokumen.

Setiap entri dari daftar mengacu pada frekuensi atau hitungan entri yang sesuai dalam daftar bag-of-words. Ketika kita memiliki kumpulan (baris) vektor, atau matriks, di mana setiap baris merupakan dokumen (vektor), dan setiap kolom merupakan setiap kata dalam daftar bag-of-words, maka kumpulan tersebut dikenal sebagai *term-frequency* (tf) matriks dokumen.

Term-frequency adalah sparse matrix dimana setiap baris merupakan dokumen dalam kumpulan data latih (*D*) dan setiap kolom merupakan istilah / kata dalam daftar *bag-of-words*. Pada sentimen analisis yang dikembangkan, program untuk membuat matriks term-frequency menggunakan kelas *CountVectorizer* pada *library* scikit-learn:

```
from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer(min_df=1)
X_train_counts = count_vect.fit_transform(clean_data.text)
print (X_train_counts.shape)
count_vect.vocabulary_
```

Selanjutnya, fitur yang diekstraksi akan disimpan ke file feature.pkl dengan memanfaatkan *pickle*. *Pickle* adalah sebuah modul pada standard *library python*, yang dapat digunakan untuk menyimpan dan membaca data ke dalam /dari sebuah file.

```
import pickle
with open('./feature.pkl', 'wb') as f:
    pickle.dump(count_vect, f)
print('saved featured in ', './feature.pkl')
```

6. TF-IDF

Proses selanjutnya yang kami lakukan adalah menghitung nilai tf-idf. Hasil dari tahap *countvectorizer* digunakan sebagai data yang akan diolah pada tahap tf-idf. Hasil dari nilai tf-idf menunjukkan kesamaan atau *similarity* antar dokumen dengan memberikan *value* atau bobot pada setiap kata yang terdapat di dalam dataset. Nilai yang dihasilkan merupakan ukuran statistik yang digunakan untuk mengevaluasi seberapa penting sebuah kata di dalam sebuah dataset. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika jarang muncul dalam dokumen.

```
from sklearn.feature_extraction.text import TfidfTransformer
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
print (X_train_tfidf.shape)
print(tfidf_transformer.fit_transform(count_vect.fit_transform(clean_data.text)).toarray())
```

7. Implementasi SVM

Implementasi *Support Vector Machine* yang dikembangkan menggunakan kernel linear. Data yang akan dianalisis menggunakan *Support Vector Machine* telah dilakukan pembobotan t*erm frequency inverse document frequency* (tf-idf) lebih dahulu pada setiap kata.

Pelatihan model menggunakan data training dengan SVM linear mempertimbangkan parameter C. Parameter C yang digunakan adalah 1. Model yang telah dilatih dengan data selanjutnya akan digunakan untuk mengklasifikasikan data.

```
# SVM regularization parameter

y = clean_data.Sentimen
C = 1.0
model_sentimen = SVC(probability=True, kernel='linear', C=C)
model = model_sentimen.fit(X_train_tfidf,y)


```

Setelah pemodelan menggunakan SVM dilakukan, maka kami mengevaluasi kinerja model berdasarkan metrik kesalahan untuk menentukan keakuratan model. Pada proyek ini, kami menggunakan *cross-validation (*CV) untuk mengevaluasi kinerja model atau algoritma dimana data dipisahkan menjadi dua subset yaitu data proses pembelajaran dan data validasi / evaluasi. Model atau algoritma dilatih oleh subset pembelajaran dan divalidasi oleh subset validasi. Selanjutnya pemilihan jenis CV dapat didasarkan pada ukuran dataset. Biasanya K-fold *Cross-Validation* digunakan karena dapat mengurangi waktu komputasi dengan tetap menjaga keakuratan estimasi. [7]

CV K-Fold adalah tempat kumpulan data tertentu dibagi menjadi sejumlah K bagian/lipatan di mana setiap lipatan digunakan sebagai kumpulan pengujian di beberapa titik. Pada proyek ini kami menggunakan k=5 yang berarti terdapat 5 grup pelatihan data.

```
# Implementasi K-fold Cross Validation
# Implementasi tahap ini digunakan untuk membagi dataset kedalam data train dan data test yang akan digunakan pada proses kla

from sklearn.model_selection import KFold
i = 0
kf = KFold(n_splits=5)
for train, test in kf.split(X_train_tfidf):
    X = X_train_tfidf
    y = y
    X_train, X_test, y_train, y_test = X[train], X[test], y[train], y[test]
    model_sentimen = model_sentimen.fit(X_train,y_train)
    print("classification_report", classification_report(model_sentimen.predict(X_test),y_test))
    with open('model_sentimen '+ str(i)+'.pkl ', 'wb') as f:
        pickle.dump(model_sentimen, f)
        print('model_sentimen',i,' saved in', './model.pkl',i)
        i+= 1
```

Pada program diatas dituliskan kode untuk menampilkan hasil klasifikasi dari pelatihan yang dilakukan. Berikut adalah hasil klasifikasi dari sentimen analisis yang dilakukan menggunakan SVM.

```
classification_report              precision    recall  f1-score   support

   Negative       0.05      0.67      0.09         3
   Positive       1.00      0.98      0.99      1997

avg / total       1.00      0.98      0.99      2000

model_sentimen 0  saved in ./model.pkl 0
classification_report              precision    recall  f1-score   support

   Negative       0.07      1.00      0.12         3
   Positive       1.00      0.98      0.99      1997

avg / total       1.00      0.98      0.99      2000

model_sentimen 1  saved in ./model.pkl 1
classification_report              precision    recall  f1-score   support

   Negative       0.04      0.75      0.08         4
   Positive       1.00      0.97      0.98      1996

avg / total       1.00      0.97      0.98      2000

model_sentimen 2  saved in ./model.pkl 2
classification_report              precision    recall  f1-score   support
```

8. Evaluasi Model

Berikut ini adalah proses yang akan menunjukkan hasil evaluasi model yang dibangun. Pada evaluasi ini kami menggunakan confusion matrix yang menghasilkan nilai *precision, recall dan f1-score.*

```
evaluasi_model_sentimen = pd.DataFrame(performance_model)
evaluasi_model_sentimen.columns = ['precision','recall','f1-score']

evaluasi_model_sentimen
```

Berikut ini adalah hasil precision, recall dan f1-score model yang dibangun. Dari hasil tersebut untuk masing-masing model diberikan nilai precision, recall dan f1-score. Nilai precision untuk kelima model sama yaitu 1, nilai recall untuk model 1 adalah 0.98 dan nilai f1-score adalah 0.99. Nilai recall untuk model 2 adalah 0.98 dan nilai f1-score adalah 0.99. Nilai recall untuk model 3 adalah 0.97 dan nilai f1-score adalah 0.98. Nilai recall untuk model 4 adalah 0.98 dan nilai f1-score adalah 0.99. Nilai recall untuk model 5 adalah 0.96 dan nilai f1-score adalah 0.98.

	precision	recall	f1-score
0	1.00	0.98	0.99
1	1.00	0.98	0.99
2	1.00	0.97	0.98
3	1.00	0.98	0.99
4	1.00	0.96	0.98

Untuk menampilkan nilai rata-rata dari kelima model yang dibangun adalah sebagai berikut. Dari hasil tersebut nilai rata-rata precision adalah 1, nilai rata-rata recall adalah 0.973 dan nilai rata-rata f1-score adalah 0.986.

```
sumprecision = evaluasi_model_sentimen['precision'].astype(float)
avgprecision = sum(sumprecision)/len(sumprecision)

sumrecall = evaluasi_model_sentimen['recall'].astype(float)
avgrecall = sum(sumrecall)/len(sumrecall)

sumfscore = evaluasi_model_sentimen['f1-score'].astype(float)
avgfscore = sum(sumfscore)/len(sumfscore)

print ('precision', 'recall', 'fscore' )
print (avgprecision, avgrecall, avgfscore )
```

```
precision recall fscore
1.0 0.9739999999999999 0.986
```

2.4 Hasil

Hasil yang diperoleh dari model yang dibangun adalah kelima model sentimen dapat menganalisis dengan baik review pelanggan shopee yang bersentimen positif dan bersentimen negatif dengan nilai rata-rata precision adalah 1, nilai rata-rata recall adalah 0.973 dan nilai rata-rata f1-score adalah 0.986.

Berikut ini adalah hasil evaluasi dari kelima model sentimen yang dibangun.

- Model sentimen 1

```
classification_report              precision    recall  f1-score   support

   Negative       0.05      0.67      0.09         3
   Positive       1.00      0.98      0.99      1997

avg / total       1.00      0.98      0.99      2000

model_sentimen 0  saved in ./model.pkl 0
```

- Model sentimen 2

```
classification_report              precision    recall  f1-score   support

   Negative       0.07      1.00      0.12         3
   Positive       1.00      0.98      0.99      1997

avg / total       1.00      0.98      0.99      2000

model_sentimen 1  saved in ./model.pkl 1
```

- Model sentimen 3

```
classification_report              precision    recall  f1-score   support

   Negative       0.04      0.75      0.08         4
   Positive       1.00      0.97      0.98      1996

avg / total       1.00      0.97      0.98      2000

model_sentimen 2  saved in ./model.pkl 2
```

- Model sentimen 4

```
classification_report              precision    recall  f1-score   support

   Negative       0.04      1.00      0.08         2
   Positive       1.00      0.98      0.99      1998

avg / total       1.00      0.98      0.99      2000

model_sentimen 3  saved in ./model.pkl 3
```

- Model sentimen 5

```
classification_report              precision    recall  f1-score   support

   Negative       0.03      0.67      0.05         3
   Positive       1.00      0.96      0.98      1997

avg / total       1.00      0.96      0.98      2000

model_sentimen 4  saved in ./model.pkl 4
```

3. Penutup

Bab ini berisi penjelasan mengenai pembagian tugas dan tanggung jawab setiap anggota tim dalam pengerjaan proyek. Pada bab ini juga akan dibahas mengenai kesimpulan dan saran yang diperoleh dari pengerjaan proyek.

3.1 Pembagian Tugas dan Tanggung Jawab

Dalam pengerjaan proyek, kelompok yang terdiri atas 3 orang akan mengerjakan tugas masing-masing untuk menyelesaikan proyek analisis sentimen terhadap *review* pelanggan Shopee. Berikut adalah tabel pembagian tugas dalam kelompok.

Tabel 2 Pembagian Tugas dalam Kelompok

No	Kegiatan	Penanggung Jawab	Luaran
1	Penyusunan Proposal	Astri, Mega, Pebri	Proposal
2	Analisis Proyek dan Perencanaan Teknik Analisis Sentimen	Astri, Mega, Pebri	
3	Load data dan Pre Processing Data	Mega	Program Python
4	Feature Extraction dan TF-IDF	Astri	Program Python
5	Implementasi SVM	Astri, Mega, Pebri	Program Python
6	Evaluasi Model	Pebri	Program Python
7	Penulisan Laporan Akhir	Astri, Mega, Pebri	Laporan Akhir
8	Presentasi	Astri, Mega, Pebri	Video

3.2 Kesimpulan

Berdasarkan proses analisis sentimen yang dilakukan, maka kesimpulan yang didapat adalah:

1. Penggunaan metode *Support Vector Machine* yang digunakan dengan pembobotan TF-IDF dapat menjadi salah satu cara untuk menyelesaikan permasalahan klasifikasi dalam analisis sentimen. Hal tersebut dapat dibuktikan dengan penggunaan hasil pembobotan kata TF-IDF data ulasan dengan nilai label terhadap masing-masing aspek untuk proses klasifikasi sentimen menggunakan metode SVM.
2. Pengujian dilakukan dengan 5 kali iterasi dengan besar precision, recall, dan fscore masing-masing 1.0; 0.9739999999999999; 0.986.
3. Tim proyek berpendapat bahwa SVM merupakan salah satu algoritma yang sudah baik terbukti dengan akurasi yang tinggi, sehingga sangat baik diterapkan untuk melakukan analisis sentimen.
4. Keberhasilan penerapan algoritma SVM didukung dengan penerapan Named-Entity Recognition, Case Folding, Remove punctuation, Stopwords Removal, Stemming, dan Lemmatization.
5. Preprocessing dataset akan memudahkan penerapan SVM dalam melakukan analisis sentimen.

3.3 Saran

Berdasarkan penelitian yang telah dilakukan, dalam analisis sentimen menggunakan metode *Support Vector Machine* akan menimbulkan tingkat akurasi yang tinggi. Semakin banyak data yang diujikan maka tingkat akurasi dari hasil klasifikasi akan semakin tinggi. Kedepannya diharapkan dalam penelitian ini dapat menggunakan metode lain sebagai pembanding dari metode *Support Vector Machine* yang sudah dikerjakan dalam proyek ini.

Daftar Pustaka

[1] A. S. A. F. Alvi Pranandha Syah, "ANALISIS SENTIMEN PADA DATA ULASAN PRODUK TOKO ONLINE DENGAN METODE MAXIMUM ENTROPY2017," *e-Proceeding of Engineering*, vol. 4, no. 3, p. 4633, 3 Desember 2017.

[2] J. I. T. I. Johan Oscar Ong, "IMPLEMENTASI ALGORITMA K-MEANS CLUSTERING UNTUK MENENTUKAN STRATEGI MARKETING PRESIDENT UNIVERSITY," vol. 12, 1 Juni 2013.

[3] G. V. Y. L. Valonia Inge Santoso, "PENERAPAN SENTIMENT ANALYSIS PADA HASIL EVALUASI DOSEN DENGAN METODE SUPPORT VECTOR MACHINE," *JURNAL TRANSFORMATIKA*, no. 2, Januari 2007.

[4] F. d. Sanger, ""Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data,"," 2007.

[5] D. a. J. H. M. Jurafsky, "Speech and Language Processing," *Prentice Hall*, 2008.

[6] L. M. M. A. F. Dimas Joko Haryanto, "Analisis Sentimen Review Barang Berbahasa Indonesia Dengan Metode Support Vector Machine Dan Query Expansion," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2.

[7] O. P. H. Pardomuan Robinson Sihombing, "Perbandingan Metode Artificial Neural Network (ANN) dan Support Vector Machine (SVM) untuk Klasifikasi Kinerja Perusahaan Daerah Air Minum (PDAM) di Indonesia," *Jurnal Ilmu Komputer*, vol. XII, no. 1.