BAB 3
METODE PENELITIAN

Penelitian ini merupakan penelitian sentimen positif dan sentimen negatif pada data *tweet* menggunakan metode SVM yang berkaitan dengan isu Presiden Jokowi tiga Periode dengan pengumpulan datanya menggunakan Snscrape. Selanjutnya dilakukan pengolahan data berupa *preprocessing* untuk mendapatkan sebuah hasil.

Penelitian ini didasari dari latar belakang permsalahan yang muncul dengan mengambil data *tweet* yang kemudian diolah dan didefinisikan sentimen yang sesuai dengan permasalahan sehingga mendapatkan hasil yang sesuai. Berikut ini merupakan bahan penelitian, alat penelitian, dan jalan penelitian dalam melakukan penelitian analisis sentimen isu Presiden Jokowi tiga periode.

3.1 BAHAN DAN ALAT PENELITIAN

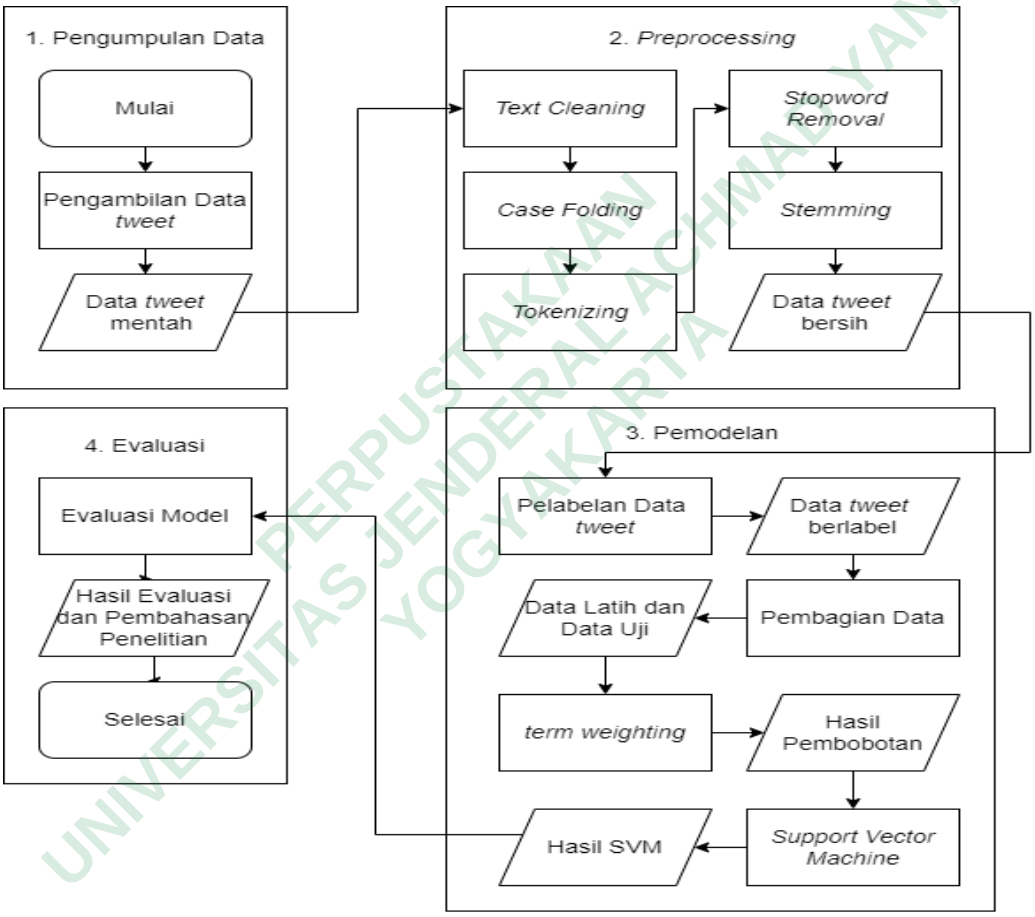
Bahan penelitian yang digunakan adalah data *tweet* yang berada di Twitter yang memiliki hubungan dengan isu Presiden Jokowi tiga periode. Data yang diambil dari Twitter merupakan data yang mengandung kata kunci "jokowi 3 periode". Data tersebut diambil mulai dari tanggal 01 September 2022 sampai 31 Desember 2022.

Alat yang digunakan dalam penelitian ini adalah komputer dengan spesifikasi cukup untuk menjalankan sistem operasi dan perangkat lunak pengembangan serta koneksitas Internet. Sistem operasi dan program-program aplikasi yang dipergunakan dalam dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi: Windows 11.
2. Processor AMD Ryzen 5 3500U with Radeon Vega Mobile Sfx 2.10 GHz.
3. Visual Studio Code 1.76
4. Python 3.8.8.
5. SQLite 3.35.5
6. Figma

3.2 JALAN PENELITIAN

Penelitian ini merupakan penelitian yang dilakukan untuk menentukan sebuah kelas pemisah yaitu kelas positif dan kelas negatif yang data tersebut diambil dari *tweet* yang berasal dari media sosial Twitter. Penelitian ini dilakukan dengan metode SVM dengan melakukan pengumpulan data *tweet* yang memiliki hubungan dengan isu Presiden Jokowi tiga periode. Langkah-langkah metode penelitian ini yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 *Flowchart* penelitian

3.2.1 Pengumpulan Data

Pengumpulan data pada tahap ini dilakukan dengan menggunakan Snscrape dengan kata kunci yang berhubungan dengan isu Presiden Jokowi 3 periode yaitu "jokowi 3 periode". Pengambilan data dimulai dari tanggal 01 September sampai 31 Desember 2022. Dari hasil pengambilan data ini dihasil data sebanyak 7.861 data yang merupakan data yang masih mentah.

```
MAX_TWEETS = 10000s
data_tweets = []

    for i,tweet in enumerate (sntwitter.TwitterSearchScraper
(f'jokowi 3 periode since:{request.form["since"]}
until:{request.form["until"]} lang:id').get_items()):

        if i > MAX_TWEETS:
            break

        data_tweets.append([tweet.date.strftime("%d/%m/%Y"),
tweet.id, tweet.username, tweet.content])
```

Kode program diatas digunakan untuk melakukan pengambilan data dari Twitter. Fungsi dari MAX_TWEETS adalah sebagai pengatur panjang data yang diperbolehkan dalam pengambilan data *tweet.* Fungsi TwitterSearchScrapper dipergunakan untuk mengambil data sesuai dengan kata kunci yang sesuai dengan penelitian.

```
tweets=pd.DataFrame(data_tweets,columns=['datetime', 'tweet_id',
       'username', 'text'])
```

Kode program diatas digunakan untuk membuat tabel dengan colomn yang berisikan *datetime, tweet_id, username,* dan *text.* Dataframe ini diisi oleh data yang berhasil dikumpulkan pada proses diatas.

```
def download():
    model_name = request.args.get("model","tweet")
    download_type = request.args.get("type","csv")

    model_dict = {
        "tweet": Tweet,
        "process": Preprocess,
        "label": LabelModel
    }

    model = model_dict[model_name].query.all()
    model = [row.dict__ for row in model]
    frame = pd.DataFrame(model)

    if download_type == "excel":
        frame.to_excel(f"app/data/{model_name}.xlsx")
        return send_file (f"data/{model_name}.xlsx",
        as_attachment=True)
    else:
        frame.to_csv(f"app/data/{model_name}.csv")
```

```
return send_file(f"data/{model_name}.csv",
as_attachment=True)
```

Kode program di atas digunakan untuk menyimpan data dengan nama penyimpanannya sesuai dengan nama model atau basis data yang akan disimpan. Data dapat disimpan atau diunduh dengan format CSV dan Excel. Hasil dari pengumpulan data *tweet* dapat dilihat pada Gambar 3.2.

username	text	id	datetime	tweet_id
Yenithung1	@CNNIndonesia Rakya	1	1/9/2022	1.56549E+18
SasmayaRespati	@haryopanuntun @W	2	1/9/2022	1.56548E+18
SasmayaRespati	@deditelaumbanu4 @	3	1/9/2022	1.56548E+18
JonasMarzuki	@democrazymedia Sa	4	1/9/2022	1.56548E+18
JonasMarzuki	@susipudjiastuti Salah	5	1/9/2022	1.56548E+18
Tamps_J	@YanHarahap Yes, Jok	6	1/9/2022	1.56548E+18
HantonoYuli	@RamliRizal 3 periode	7	1/9/2022	1.56548E+18
a_djaffar	@rahmani_ima @joko	8	1/9/2022	1.56548E+18
AKUsetiaNKRI	@Syarman59 @qomar	9	1/9/2022	1.56548E+18

Gambar 3.2 Data *tweet*

Pada Gambar 3.2 data *tweet* yang berhasil diambil memiliki banyak komponen yang tidak perlukan dalam proses analisis seperti *username, id, datetime, tweet_id*. Maka dari itu dibutuhkan *preprocessing* data agar data menjadi data yang bersih dan terstruktur.

3.2.2 *Preprocessing*

Tahap ini dilakukan sebelum proses penggolongan sentimen dimana terdapat pengolahan *dataset* mentah menjadi *dataset* bersih yang dipersiapkan untuk proses tahap selanjutnya dimana proses ini dibantu dengan *library* re atau regex untuk membantu mendefinisikan sebuah pola. Langkah–langkah pembersihan data dapat dilihat dibawah ini.

1 *Text Cleaning*

Text cleaning adalah tahap dimana dilakukan penghapusan berupa emotikon, *mention*, simbol-simbol, dan yang lainnya yang tidak memiliki fungsi atau tidak relevan dengan data yang diambil. Berikut merupakan kode program untuk melakukan *text cleaning* pada data.

```
def remove_pattern(self,text, pattern):
        r = re.findall(pattern, str(text))
        for i in r:
            text = re.sub(i, '', str(text))

        return text

    def cleaning(self,text):
        text = re.sub(r'\$\w*', '', text)
        text = re.sub(r'^rt[\s]+', '', text)
        text = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', ' ',
               text)
        text = re.sub('&quot;'," ", text)
        text = re.sub(r"\d+", " ", str(text))
        text = re.sub(r"\b[a-zA-Z]\b", "", str(text))
        text = re.sub(r"[^\w\s]", " ", str(text))
        text = re.sub(r'(.)\1+', r'\1\1', text)
        text = re.sub(r"\s+", " ", str(text))
        text = re.sub(r'#', '', text)
        text = re.sub(r'[^a-zA-z0-9]', ' ', str(text))
        text = re.sub(r'\b\w{1,2}\b', '', text)
        text = re.sub(r'\s\s+', ' ', text)
        text = re.sub(r'^RT[\s]+', '', text)
        text = re.sub(r'^b[\s]+', '', text)
        text = re.sub(r'^link[\s]+', '', text)

        return text

def process(self):
     self.frame['remove_user'] =np.vectorize(self.remove_pattern)
     (self.frame['text'], "@[\w]*")
```

Kode program diatas terdapat fungsi remove_pattern yang dipergunakan untuk membersihkan pola tertentu dalam teks menggunakan regex. Fungsi cleaning dipergunakan untuk membersihkan teks dengan melakukan beberapa substitusi dan penghapusan karakter khusus pada teks. Fungsi proses merupakan fugsi untuk menghapus *user* atau *mention* pada teks. Hasil dari tahapan *text cleaning* dapat dilihat pada Tabel 3.1.

Tabel 3.1 Hasil *text cleaning*

text	*remove_user*	*text_cleaning*
@myputun Katanya sih Ada 3 agenda oligarki Plan a jokowi 3 periode Plan b perpanjangan jabatan/penundaan pemilu Plan c dukung boneka... https://t.co/d9cfPPJOAU.	Katanya sih Ada 3 agenda oligarki Plan a jokowi 3 periode Plan b perpanjangan jabatan/penundaan pemilu Plan c dukung boneka... https://t.co/d9cfPPJOAU.	Katanya sih Ada agenda oligarki Plan jokowi periode Plan perpanjangan jabatan penundaan pemilu Plan dukung boneka
Pilih mana ? PAN pendukung Jokowi, Tunda Pemilu dan 3 periode presiden. PARTAI UMMAT pendukung Perubahan. PILIH MANA, PARTAI UMMAT atau PAN ? https://t.co/uYURJdqsF0	Pilih mana ? PAN pendukung Jokowi, Tunda Pemilu dan 3 periode presiden. PARTAI UMMAT pendukung Perubahan. PILIH MANA, PARTAI UMMAT atau PAN ? https://t.co/uYURJdqsF0	Pilih mana PAN pendukung Jokowi Tunda Pemilu dan periode presiden PARTAI UMMAT pendukung Perubahan PILIH MANA PARTAI UMMAT atau PAN
@OposisiCerdas Pak Amien bicara menekankan kalau Jokowi tidak ngotot 3 periode atau menunda pemilu. Dan turun sesuai waktu😂 😂	Pak Amien bicara menekankan kalau Jokowi tidak ngotot 3 periode atau menunda pemilu. Dan turun sesuai waktu😂 😂	Pak Amien bicara menekankan kalau Jokowi tidak ngotot periode atau menunda pemilu Dan turun sesuai waktu
Tolak Jokowi 3 Periode, Rocky Gerung Langsung Beberkan Strategi PROJO https://t.co/t1XZyqS1NW	Tolak Jokowi 3 Periode, Rocky Gerung Langsung Beberkan Strategi PROJO https://t.co/t1XZyqS1NW	Tolak Jokowi Periode Rocky Gerung Langsung Beberkan Strategi PROJO

Pada Tabel 3.1 dapat dilihat bahwa pola tertentu terhapus, link pada data terhapus, *mention* pada data terhapus, dan emotikon pada data terhapus.

2. *Case Folding*

Case folding adalah tahap dirubahnya sebuah kalimat kedalaman huruf kecil atau huruf besar secara keseluruhan. Tujuan dari *case folding* ini adalah agar data lebih terstruktur. Kode program dapat dilihat dibawah ini.

```
self.frame['case_folding']=self.frame['text_cleaning'].str.lower()
```

Kode program diatas merupakan potongan setelah melakukan *text cleaning* maka data hanya perlu dipanggil. Pada fungsi lower merupakan fungsi untuk mengubah teks menjadi huruf kecil semua. Hasil dari *case folding* dapat dilihat pada Tabel 3.2.

Tabel 3.2 Hasil *case folding*

Text_cleaning	*Case_folding*
Katanya sih Ada agenda oligarki Plan jokowi periode Plan perpanjangan jabatan penundaan pemilu Plan dukung boneka	katanya sih ada agenda oligarki plan jokowi periode plan perpanjangan jabatan penundaan pemilu plan dukung boneka
Pilih mana PAN pendukung Jokowi Tunda Pemilu dan periode presiden PARTAI UMMAT pendukung Perubahan PILIH MANA PARTAI UMMAT atau PAN	pilih mana pan pendukung jokowi tunda pemilu dan periode presiden partai ummat pendukung perubahan pilih mana partai ummat atau pan
Pak Amien bicara menekankan kalau Jokowi tidak ngotot periode atau menunda pemilu Dan turun sesuai waktu	pak amien bicara menekankan kalau jokowi tidak ngotot periode atau menunda pemilu dan turun sesuai waktu
Tolak Jokowi Periode Rocky Gerung Langsung Beberkan Strategi PROJO	tolak jokowi periode rocky gerung langsung beberkan strategi projo

Pada Tabel 3.2 terlihat bahwa data teks memiliki persamaan gaya huruf yaitu huruf kecil sebagai penyalaras kata pada kalimat teks.

3. *Tokenizing*

Tokenizing adalah proses pemecahan kalimat menjadi kata-kata yang disebut dengan token. Berikut merupakan kode program untuk memproses *tokenizing.*

```
def word_tokenize_wrapper(self,text):
        return word_tokenize(text)

self.frame['tokenizing'] = self.frame['case_folding'].apply(lambda
x: self.word_tokenize_wrapper(x.lower()))
```

Kode program diatas merupakan kode program yang digunakan untuk membuat sebuah kalimat menjadi sebuah kata dimana data diambil dari *case folding* dan diproses menjadi potongan-potongan kata. Hasil dari *tokenizing* dapat dilihat pada tabel 3.3.

Tabel 3.3 Hasil *tokenizing*

Case_folding	*tokenizing*
katanya sih ada agenda oligarki plan jokowi periode plan perpanjangan jabatan penundaan pemilu plan dukung boneka	katanya,sih,ada,agenda,oligarki,plan, jokowi,periode,plan,perpanjangan, jabatan,penundaan,pemilu,plan, dukung,boneka
pilih mana pan pendukung jokowi tunda pemilu dan periode presiden partai ummat pendukung perubahan pilih mana partai ummat atau pan	pilih,mana,pan,pendukung,jokowi, tunda,pemilu,dan,periode,presiden, partai,ummat,pendukung,perubahan,pilih, mana,partai,ummat,atau,pan
pak amien bicara menekankan kalau jokowi tidak ngotot periode atau menunda pemilu dan turun sesuai waktu	pak,amien,bicara,menekankan,kalau, jokowi,tidak,ngotot,periode,atau, menunda,pemilu,dan,turun,sesuai, waktu
tolak jokowi periode rocky gerung langsung beberkan strategi projo	tolak,jokowi,periode,rocky,gerung, langsung,beberkan,strategi,projo

Dari Tabel 3.3 kata yang semula masih utuh berbentuk kalimat setelah dilakukan *tokenizing* menjasi pecahan kata–kata sesuai dengan kalimat.

4. *Stopword Removal*

Stopword removal merupakan langkah *filtering* penghapusan kata-kata yang dinilai tidak penting untuk data tersebut. Kode program yang digunakan untuk proses *stopword removal* dapat dilihat dibawah ini.

```
LIST_STOPWORDS = stopwords.words('indonesian')
LIST_STOPWORDS.extend(['amp', 'biar', 'sih', 'si', 'n', 't', 'nyg',
                       'hehe', 'pen', 'u', 'nan', 'loh', 'rt','yah',
                       'no', 'je', 'om', 'pru', 'sch','injirrr',
                       'ah','oena', 'bu', 'eh', 'xac', 'anjir'])

LIST_STOPWORDS = set(LIST_STOPWORDS)
def stopwords_removal(self,text):
        # Menghapus stopwords dari teks
        return [word for word in text if word not in
```

```
self.LIST_STOPWORDS]
```

Pada kode program diatas LIST_STOPWORD digunakan untuk memanggil NLTK *corpus* dan untuk menambahkan kata-kata yang belum ada pada NLTK *corpus.* Fungsi stopword_removal adalah melakukan proses penghapusan kata yang terdapat pada teks sesuai kata yang berada pada LIST_STOPWORD. Proses ini menggunakan NLTK corpus berbahasa Indonesia seperti Gambar 3.3.

```
{'diperlukan', 'hendaknya', 'tapi', 'dimungkinkan', 'hendaklah', 'umumnya', 'tambahnya', 'usai', 'katakan',
'sebagaimana', 'sekali', 'persoalan', 'waduh', 'bermaksud', 'jelaslah', 'ditanyai', 'tiba', 'terdahulu',
'menghendaki', 'tidak', 'sangatlah', 'kalaulah', 'rata', 'tadi', 'sendirinya', 'tersampaikan', 'sekadar',
'mengakhiri', 'mempergunakan', 'sedikit', 'sekali-kali', 'katakanlah', 'karenanya', 'oleh', 'semampunya',
'diakhirinya', 'kapanpun', 'setidaknya', 'disini', 'menaiki', 'tentunya', 'terbanyak', 'tak', 'secara',
'diibaratkannya', 'mengatakan', 'hendak', 'dikarenakan', 'sekarang', 'berturut', 'ditanyakan', 'terlihat',
'diperlukannya', 'sebuah', 'cuma', 'ingat-ingat', 'sesegera', 'mengerjakan', 'keinginan', 'berlebihan', 'apalagi',
'siapapun', 'enggaknya', 'lagi', 'diungkapkan', 'bisa', 'tentu', 'bersiap', 'dia', 'ia', 'ini', 'dituturkan',
'mendatang', 'semacam', 'sebenarnya', 'terutama', 'diibaratkan', 'tunjuk', 'inilah', 'diri', 'seterusnya',
```

Gambar 3.3 List NLTK *corpus* Indonesia

Pada Gambar 3.3 merupakan *corpus* dari NLTK untuk melakukan proses *stopword removal.* Hasilnya dapat dilihat pada Tabel 3.4.

Tabel 3.4 Hasil *stopword removal*

tokenizing	*Stopword removal*
katanya,sih,ada,agenda,oligarki,plan, jokowi,periode,plan,perpanjangan, jabatan,penundaan,pemilu,plan, dukung,boneka	agenda,oligarki,plan,jokowi,periode, plan,perpanjangan,jabatan,penundaan, pemilu,plan,dukung,boneka
pilih,mana,pan,pendukung,jokowi, tunda,pemilu,dan,periode,presiden, partai,ummat,pendukung,perubahan, pilih,mana,partai,ummat,atau,pan	pilih,pan,pendukung,jokowi,tunda, pemilu,periode,presiden,partai,ummat, pendukung,perubahan,pilih,partai, ummat,pan
pak,amien,bicara,menekankan,kalau, jokowi,tidak,ngotot,periode,atau, menunda,pemilu,dan,turun,sesuai, waktu	amien,bicara,menekankan,jokowi, ngotot,periode,menunda,pemilu,turun, sesuai
tolak,jokowi,periode,rocky,gerung, langsung,beberkan,strategi,projo	tolak,jokowi,periode,rocky,gerung, langsung,beberkan,strategi,projo

Pada Tabel 3.4 menunjukan bahwa beberapa kata sudah terhapus setelah dilakukannya proses *stopword removal*. Seperti contoh pada atas kata ”katanya” menghilang setelah dilakukan proses *stopword removal.*

5. *Stemming*

Stemming merupakan pengurangan kata tambahan menjadi kata dasar atau bentuk dasar dari kata yang sudah berimbuhan. Berikut merupakan kode program untuk melakukan proses *stemming.*

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()
def stemming(self,text):
        term_dict = {}

        for text in self.frame['stop_words']:
            for term in text:
                if term not in term_dict:
                    term_dict[term] = ' '

        for term in term_dict:
            term_dict[term] = stemmer.stem(term)

        text = [term_dict[term] for term in text]
        return ' '.join(text)

def fit_stemming(self,text):
    text = np.array(text)
    text = ' '.join(text)

    return text

def stemmed_wrapper(term):
            return stemmer.stem(term)

        term_dict = {}

        for Text in self.frame['stop_words']:
            for term in Text:
                if term not in term_dict:
                    term_dict[term] = ' '

        for term in term_dict:
            term_dict[term] = stemmed_wrapper(term)

        # Memulai stemming
        def apply_stemmed_term(Text):
            return [term_dict[term] for term in Text]
self.frame['stemming']=self.frame['stop_words'].apply
```

```
(apply_stemmed_term)
```

Kode program diatas merupakan kode program yang dipergunakan untuk melakukan proses *stemming* yang menggunakan bantuan *library* Sastrawi. Hasil dari proses *stemming* ini dapat dilihat pada Tabel 3.5.

Tabel 3.5 Hasil *stemming*

no	*tokenizing*	*stemming*
1	agenda,oligarki,plan,jokowi,periode, plan,perpanjangan,jabatan,penundaan, pemilu,plan,dukung,boneka	agenda oligarki plan jokowi periode plan panjang jabat tunda milu plan dukung boneka
2	pilih,pan,pendukung,jokowi,tunda, pemilu,periode,presiden,partai,ummat, pendukung,perubahan,pilih,partai, ummat,pan	pilih pan dukung jokowi tunda milu periode presiden partai ummat dukung ubah pilih partai ummat pan
3	amien,bicara,menekankan,jokowi, ngotot,periode,menunda,pemilu,turun, sesuai	amien bicara tekan jokowi ngotot periode tunda milu turun sesuai
4	tolak,jokowi,periode,rocky,gerung, langsung,beberkan,strategi,projo	tolak jokowi periode rocky gerung langsung kan strategi projo

Dari Tabel 3.5 dapat dilihat bahwa terdapat kata berimbuhan diubah menjadi kata dasar seperti contoh kata ”perpanjangan” menjadi ”panjang”.

3.2.3 Pemodelan

Pada tahap ini dilakukan beberapa langkah untuk proses pemodelan yang dapat dilihat di bawah ini.

1. *Labeling*

Pada tahap *labeling* ini dilakukan pelabelan data secara otomatis menggunakan Vader Lexicon yang bertujuan untuk menganalisis lebih lanjut kata-kata yang bersifat postif atau negatif. Berikut kode untuk melakukan pelabelan menggunakan Vader Lexicon.

```
from googletrans import Translator
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import pandas as pd
import nltk
import json
```

```
nltk.download('vader_lexicon')
sid = SentimentIntensityAnalyzer()

translator = Translator()

class Labeling:
    def __init__(self, frame: pd.DataFrame):
        self.frame = frame

    def labeling(self):
        with open('app/_json_sentiwords_id.txt') as f:
            data2 = f.read()

        sentiment = json.loads(data2)

        sid.lexicon.update(sentiment)


        self.frame['score'] = self.frame['stemming'].apply(lambda
                              x: sid.polarity_scores(str(x)))


        def condition(c):
            if c >= 0.0000:
                return "positif"

            else:
                return 'negatif'


        self.frame['compound'] = self.frame['score'].apply(lambda
                                 score_dict: score_dict
                                 ['compound'])


       self.frame['sentimen'] = self.frame['compound'].apply
                                (condition)


        self.frame["score"] = self.frame["score"].astype(str)


        return self.frame
```

Dari kode program diatas dihasilkan sebuah sentimen positif dan sentimen negatif dari data sebanyak 7.861 data dibagi menjadi dua sentimen yaitu 4.684 data berupa sentimen positif dan 2.270 data berupa sentimen negatif. Hasil dapat dilihat pada Tabel 3.6.

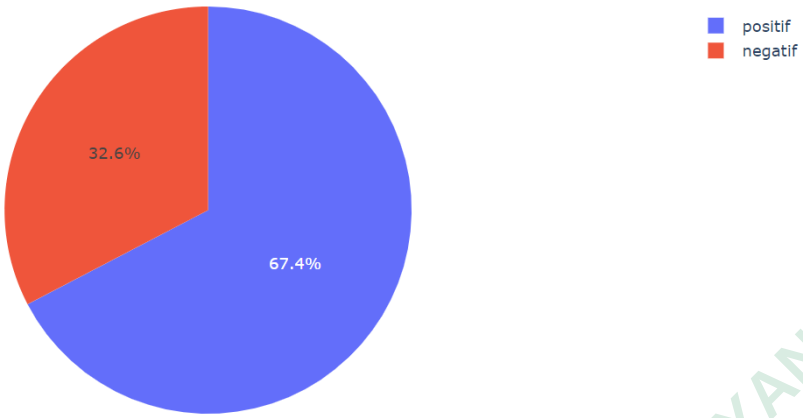
Tabel 3.6 Hasil Vader Lexicon

stemming	*compound*	*score*	sentimen
pilih pan dukung jokowi tunda milu periode presiden partai ummat dukung ubah pilih partai ummat pan	0.8402	{'neg': 0.0, 'neu': 0.636, 'pos': 0.364, 'compound': 0.8402}	positif
amien bicara tekan jokowi ngotot periode tunda milu turun sesuai	-0.2500	{'neg': 0.294, 'neu': 0.471, 'pos': 0.235, 'compound': -0.25}	negatif
Maju jokowi rakyat dukung periode	0.8750	{'neg': 0.0, 'neu': 0.25, 'pos': 0.75, 'compound': 0.875}	positif
maaf amien jokowi upaya perpanjanhan periode pensiun presiden trus tambah guru bangsa	-0.2500	{'neg': 0.154, 'neu': 0.846, 'pos': 0.0, 'compound': -0.25}	negatif

Dari Tabel 3.6 merupakan tampilan dari pelabelabelan otomatis menggunakan Vader Lexicon dimana untuk mendapatkan hasil *compound* digunakan satu data sampel dengan perhitungan pada Persamaan 6 sebagai berikut.

$$\text{nilai } compound = \frac{7}{\sqrt{7^2 + 15}} = 0{,}875$$

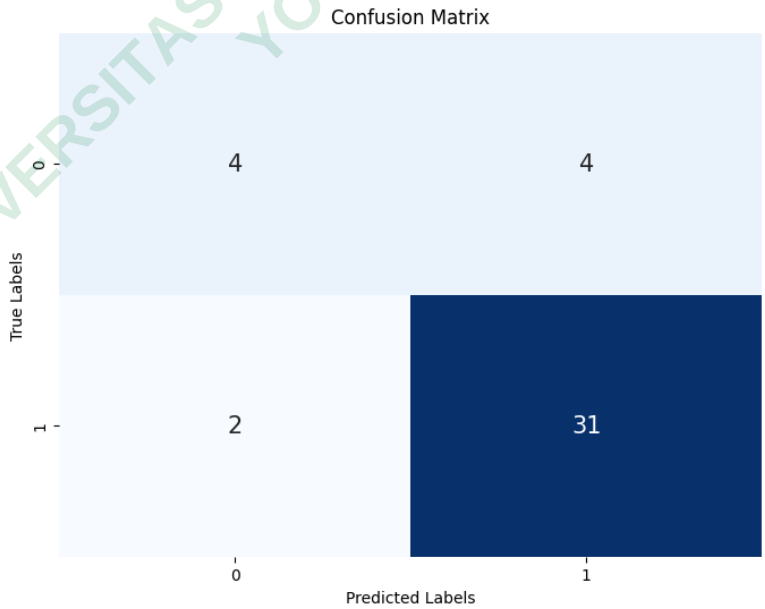
Pada perhitungan di atas data sampel yang digunakan adalah ”aman gue dukung jokowi periode” dengan kata ”aman” mengandung skor 4 dan ”dukung” mengandung skor 3. sehingga setelah dimasukkan kedalam rumus menghasilkan skor *compound* sebesar 0,875.



Gambar 3.4 *Pie Chart* sentimen positif dan negatif

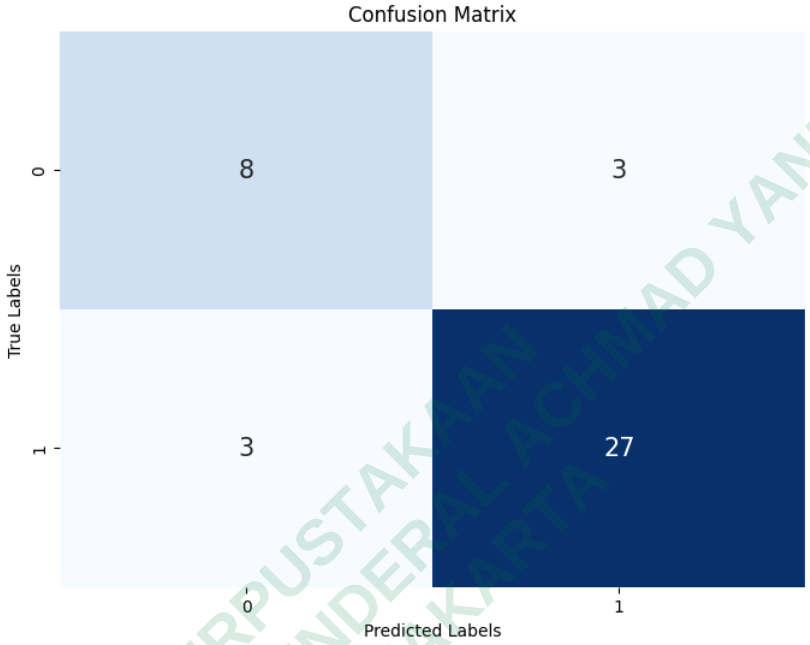
Pada Gambar 3.4 hasil dari pelabelan Vader Lexicon sebesar 67,3% merupakan data bersentimen positif dengan jumlah 4.684 data. Pada bagian data bersentimen negatif data memiliki 32,6% dengan jumlah data 2.270.

Pada bagian proses *labeling* ini dilakukan sebuah pengambilan sampel data sebesar 200 data *labeling* otomatis dan 200 data *labeling* manual untuk mengetahui keakuratan labeling pada Vader Lexicon. Pada bagian data sampel *labeling* otomatis akurasi yang dihasilkan sebesar 85%. Hasil dapat dilihat pada Gambar 3.4.



Gambar 3.5 *Confussion Matrix* data uji Vader Lexicon

Pada Gambar 3.4 ditampilkan sebuah evaluasi data uji Vader Lexicon yang memiliki akurasi sebesar 85%, presisi sebesar 88%, *recall* sebesar 93% dan *f1-score* sebesar 91%. Selanjutnya adalah 200 sampel data pelabelan manual dengan hasil yang dapat dilihat pada Gambar 3.5



Gambar 3.6 *Confussion Matrix* pelabelan manual

Pada Gambar 3.5 merupakan hasil dari evaluasi data uji pelabelan manual yang memiliki akurasi 85%, presisi 90%, *recall* 90%, dan *f1-score* sebesar 90%, dimana dapat disimpulkan bahwa pelabelan menggunakan Vader Lexicon hampir memiliki tingkat akurasi yang sama dengan pelabelan manual.

2. Data Latih dan Data Uji

Pada tahap ini terdapat dua data yaitu data latih yang merupakan data yang dilatih untuk membuat prediksi. Sedangkan data uji merupakan data yang dites untuk melihat keakuratan atau performa algoritma yang sudah dilatih. Kode program untuk melakukan pemisahan data adalah sebagai berikut.

```
def __init__(self,frame: pd.DataFrame):
        self.frame = frame
       self.frame["polarity"]=
label_encoder.fit_transform(self.frame["sentimen"])
        self.sentimen = self.frame["sentimen"].value_counts()
```

```
x = self.frame["stemming"]
y = self.frame["polarity"]
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size = 0.2, random_state = 1396)

text = self.frame['stemming']
```

Kode program tersebut merupakan kode program untuk membagi data menjadi 80% data latih dan 20% data uji dengan hasil 5.563 data merupakan data latih dan 1.391 data merupakan data uji.

3. *Term Weighting*

Tahap ini merupakan tahap pembobotan TD-IDF yang merupakan cara dalam pemberian bobot hubungan pada suatu kata dengan dokumen. Tahap ini dilakukan perkalian dari kata-kata yang terdapat pada *tweet* dengan diberi bobot nilai *term frequency* (TF) dengan *inverse document frequency (*IDF). Pada tahap ini dilakukan perhitungan pada *term-frequency*, *inverse document frequency*, dan t*erm frequency–inverse document frequency* pada data sampel dengan tiga dokumen pada Tabel 3.7 sebagai berikut.

Tabel 3.7 Data setelah *preprocessing*

Dokumen	*text*
Dokumen 1	ngusul jokowi periode
Dokumen 2	Tolak Periode
Dokumen 3	Maturnuwun pak jokowi periode njih

Pada Tabel 3.7 merupakan sampel untuk melakukan perhitungan TF-IDF. Persamaan penerapan perhitungan *term-frequency* dan *inverse document frequency* dapat dilihat pada persamaan (3) dan Persamaan (4).

$$tf_{tolak} = \frac{1}{1} = 1$$

$$idf_{tolak} = \log\binom{3}{1} = 0{,}477$$

Sehingga perolehan perhitungan TF-IDF dapat dihitung dengan persamaan (5) sebagai berikut.

$$tf - idf_{tolak} = 1\ x\ 0{,}477 = 0{,}477$$

Tabel 3.8 merupakan tabel dari hasil perhitungan *term-frequency, inverse document frequency,* dan t*erm frequency–inverse document frequency.*

Tabel 3.8 Perhitungan TF-IDF

corpus	TF			IDF	TF-IDF
	D1	D2	D3		
usul	1	0	0	0,477	0,477
jokowi	1	0	1	0,238	0,476
periode	1	1	1	0.159	0,477
tolak	0	1	0	0,477	0,477
manurnuwun	0	0	1	0,477	0,477
pak	0	0	1	0.477	0,477
njih	0	0	1	0,477	0,477

4. Klasifikasi data dengan metode SVM

Pada tahapan ini dilakukan proses klasifikasi menggunakan metode SVM untuk mengenali pola yang terdapat pada data latih yang sudah melewati tahapan *labeling*. Hasil dari pengklasifikasian ini menghasilkan model yang digunakan dalam proses klasifikasi. Penelitian ini menggunakan SVC dengan perhitungan menggunakan Persamaan (1) dan Persamaan (2) sebagai berikut.

a. Menentukan nilai x dan y yang telah ditransformasikan ke dimensi seperti Tabel 3.9.

Tabel 3.9 TF-IDF pada data uji (x) dan sentimen (y)

Data uji (x)	Label(y)
3,88	1
2,22	-1

b. Mencari *hyperplane* penggunakan persamaan (1) sehingga mendapatkan hasil persamaan sebagai berikut
   1. $3{,}88_{w1} + b \geq 1$
   2. $-2{,}22_{w1} - b \geq 1$
c. Substitusi untuk mencari nilai w1 dengan menjumlahkan persamaan (1) dan persamaan (2) sehingga memperoleh nilai sebesar 1,20.
d. Subtitusi nilai $w1$ dengan persamaan (2) dan persamaan (3) dari tahapan b yaitu 3,88$w1$ + b ≥ 1 atau -2,22$w1$ - b ≥ 1 sehingga diperoleh nilai b yaitu 1 atau -1
e. Menentukan persamaan hyperplane dari nilai $w1$ dan b yang telah dihitung sebelumnya menggunakan rumus $1{,}20x1 + b = 0$ yaitu 1,20$x1$ + 1 = 0 atau 1,20$x1$ - 1 = 0
f. Menentukan hasil klasifikasi melalui persamaan hyperplane melalui Tabel 3.10 dan 3.11 dengan hasil klasfikasi dibulatkan dari belakang koma seperti berikut.

Tabel 3.10 Hasil klasifikasi b=1

Data uji (x)	Kelas=Sign(
3,88	1
2,22	1

Tabel 3.11 Hasil klasifikasi b=-1

Data uji (x)	Kelas=Sign(
3,88	1
2,22	-1

Berdasarkan Tabel 3.10 dan Tabel 3.11 disimpulkan bahwa terdapat kecenderungan kelas 1 dari dua sampel yang telah dilakukan dalam metode ini.

3.2.4 Evaluasi

Tahapan ini dilakukan perhitungan *recall, precision, accuracy,* dan *f1-score.* Pada nilai perhitungan ini nilai didapatkan dari hasil *confusion matrix* yang didapatkan setelah dilakukan pengujian model dan data uji. Pada tahap ini juga dilakukan pembahasan hasil dari keseluruhan penelitian yang dilaksanakan.

Penggunaan metode *confusion matrix* ini digunakan untuk mengetahui presentase pada setiap pengujian. Metode yang digunakan seperti

1. *Accuracy* digunakan untuk mengetahui jumlah klasifikasi dibagi dengan total sampel *testing* yang diuji.
2. *Precision* digunakan untuk mengetahui klasifikasi dari kategori dibagi dengan total sampel klasifikasi kategori tersebut.
3. *Recall* digunakan untuk mengetahui sampel yang diklasifikasikan dalam kategori
   tertentu dibagi total sampel dalam testing yang berkategori tertentu.
4. *F1-score* digunakan untuk menghitung rata-rata dari nilai *precision* dan nilai *recall.*