

## **BAB 3**

### **METODE PENELITIAN**

Penelitian ini merupakan analisis sentimen terhadap tokoh yang akan maju di pemilihan presiden 2024. Metode Penelitian yang dilakukan adalah pengambilan data Twitter, *preprocessing*, pelabelan manual, *training* dan *testing*, klasifikasi Naive Bayes. Berikut merupakan bahan, alat, dan metode pengembangan sistem serta tahapan penelitian untuk merancang sebuah sistem sentimen bakal calon presiden.

#### **3.1 BAHAN DAN ALAT PENELITIAN**

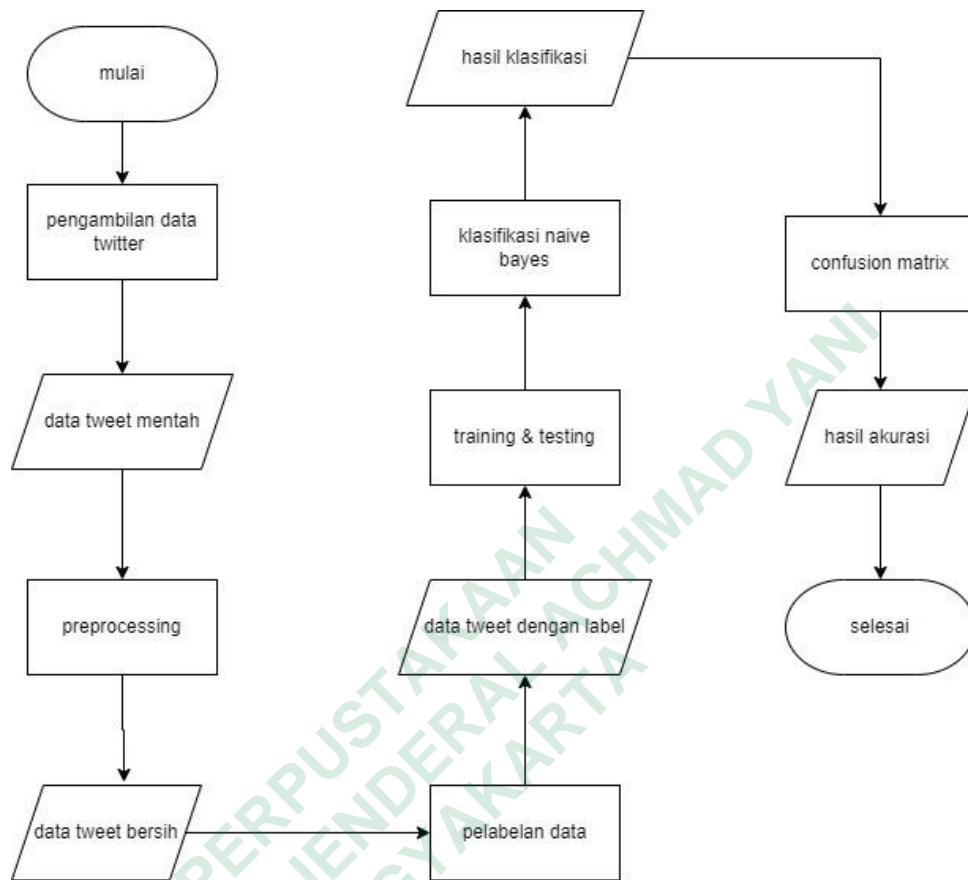
Bahan yang dipergunakan pada penelitian ini adalah data *tweet* dan komentar para pengguna di media sosial Twitter dengan hal hal yang berkaitan dengan pemilihan calon presiden 2024.

Alat penelitian yang digunakan adalah komputer dengan spesifikasi yang cukup untuk menjalankan sistem operasi dan perangkat lunak pengembangan, serta koneksi internet. Sistem Operasi dan program-program aplikasi yang digunakan dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi: Windows 10.
2. Python 3.11.0
3. Jupyter Notebook 6.5.3
4. Visual Studio Code 1.75.1

#### **3.2 JALAN PENELITIAN**

Pada penelitian menggunakan Bahasa pemrograman *Python*, *Jupyter Notebook*, Anaconda 3 yang bertujuan mengolah data yang ada di *csv* menggunakan klasifikasi Naive Bayes. Kemudian akan melakukan proses klasifikasi nilai sentimen dari setiap topik mengenai pemilihan presiden 2024, data yang sudah didapat yang nantinya ditampilkan melalui website dashboard yang dibuat dengan menggunakan bahasa pemrograman Python. Adapun langkah-langkah pada jalan penelitian ini dapat dilihat pada Gambar 3.1



Gambar 3.1 Alur Penelitian

### 3.2.1 Pengambilan Data

Pengambilan data dilakukan menggunakan metode *scraping library* *snsrape*. Data yang sudah di siapkan sebanyak 7500 data yang diambil berdasarkan Menggunakan *keyword* *capres2024*, *presiden2024*, dan *pilpres2024*. Kemudian dilakukan penghilangan *duplicate* data sehingga jumlah datanya menjadi 1211 data *tweet*. Data ini kemudian disimpan dengan format *.xlsx*. Pada Tabel 3.1 adalah beberapa data yang sudah didapatkan dari proses *scraping*.

Tabel 3.1 Data tweet hasil scraping

No	Tweet
1	Komitmen ciptakan Pemilu 2024 damai #PemiluSerentak2024 #Pemilu2024SesuaiTahapan #TolakPenundaanPemilu #AntiPolitikIdentitas #PemilihPemula #PemilihMuda #Pancasila #Pemilu2024 #Pilpres2024 #Capres2024 #KPU #Bawaslu #IndonesiaMaju <a href="https://t.co/GokPFUDprv">https://t.co/GokPFUDprv</a>
2	FKPPI Siap Kawal & Sukseskan Pemilu 2024 #PemiluSerentak2024 #Pemilu2024SesuaiTahapan #TolakPenundaanPemilu #AntiPolitikIdentitas #PemilihPemula #PemilihMuda #Pancasila #Pemilu2024 #Pilpres2024 #Capres2024 #KPU #Bawaslu #IndonesiaMaju <a href="https://t.co/N96zMfxMZ5">https://t.co/N96zMfxMZ5</a>
3	Semua Pihak Harus Terlibat Kawal Pemilu 2024 Damai #PemiluSerentak2024 #Pemilu2024SesuaiTahapan #TolakPenundaanPemilu #AntiPolitikIdentitas #PemilihPemula #PemilihMuda #Pancasila #Pemilu2024 #Pilpres2024 #Capres2024 #KPU #Bawaslu #IndonesiaMaju <a href="https://t.co/mVqpSM20BY">https://t.co/mVqpSM20BY</a>
4	Mari kita Kawal Pemilu 2024 Aman & Damai #PemiluSerentak2024 #Pemilu2024SesuaiTahapan #TolakPenundaanPemilu #AntiPolitikIdentitas #PemilihPemula #PemilihMuda #Pancasila #Pemilu2024 #Pilpres2024 #Capres2024 #KPU #Bawaslu #IndonesiaMaju <a href="https://t.co/qQI9MGLSMN">https://t.co/qQI9MGLSMN</a>
5	Wujud Sikap Netral Aparat dalam Pemilu 2024 #PemiluSerentak2024 #Pemilu2024SesuaiTahapan #TolakPenundaanPemilu #AntiPolitikIdentitas #PemilihPemula #PemilihMuda #Pancasila #Pemilu2024 #Pilpres2024 #Capres2024 #KPU #Bawaslu #IndonesiaMaju <a href="https://t.co/I19LIVVVVoW">https://t.co/I19LIVVVVoW</a>
6	Mari Kawal Pelaksanaan Pemilu 2024 Agar Berjalan dengan sukses. #PemiluSerentak2024 #Pemilu2024SesuaiTahapan #TolakPenundaanPemilu #AntiPolitikIdentitas #PemilihPemula #PemilihMuda #Pancasila #Pemilu2024 #Pilpres2024 #Capres2024 #KPU #Bawaslu #IndonesiaMaju <a href="https://t.co/wljg8Jb9bF">https://t.co/wljg8Jb9bF</a>
7	KPU menegaskan, kegiatan Ganjar, Prabowo, Anies, ketika mendatangi masyarakat, bukan kampanye karena mereka belum mendaftar sebagai Bacapres Pemilu 2024. #PemiluSerentak2024 #TolakPenundaanPemilu #PemilihPemula #PemilihMuda #Pancasila #Pemilu2024 #Capres2024 #KPU #Bawaslu <a href="https://t.co/jdHMzoJMWR">https://t.co/jdHMzoJMWR</a>

No	Tweet
8	Kawal pemilu dengan damai #PemiluSerentak2024 #Pemilu2024SesuaiTahapan #TolakPenundaanPemilu #AntiPolitikIdentitas #PemilihPemula #PemilihMuda #Pancasila #Pemilu2024 #Pilpres2024 #Capres2024 #KPU #Bawaslu #IndonesiaMaju <a href="https://t.co/KZdiXBg34g">https://t.co/KZdiXBg34g</a>
9	Kawal ketat pemilihan umum kita #PemiluSerentak2024 #Pemilu2024SesuaiTahapan #TolakPenundaanPemilu #AntiPolitikIdentitas #PemilihPemula #PemilihMuda #Pancasila #Pemilu2024 #Pilpres2024 #Capres2024 #KPU #Bawaslu #IndonesiaMaju <a href="https://t.co/BrjR164hSA">https://t.co/BrjR164hSA</a>
10	Komitmen kawal pemilu #PemiluSerentak2024 #Pemilu2024SesuaiTahapan #TolakPenundaanPemilu #AntiPolitikIdentitas #PemilihPemula #PemilihMuda #Pancasila #Pemilu2024 #Pilpres2024 #Capres2024 #KPU #Bawaslu #IndonesiaMaju <a href="https://t.co/PKyRgzpIQP">https://t.co/PKyRgzpIQP</a>

Pada Tabel 3.1 menunjukkan bahwa data hasil scraping masih banyak memiliki komponen yang tidak perlu dan tidak penting dalam proses klasifikasi.

Komponen atau karakter yang tidak dibutuhkan ini akan di bersihkan di dalam *preprocessing*.

### 3.2.2 Preprocessing

*Preprocessing* adalah tahap yang dilakukan untuk menjadikan data terstruktur sehingga dapat di olah ke dalam proses berikutnya. Berikut ini adalah library yang di gunakan untuk *preprocessing*.

```
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

Library nltk digunakan untuk pengolahan bahasa alami, seperti tokenisasi dan penghapusan kata yang tidak bermakna. `nltk.tokenize.word_tokenize` digunakan untuk membagi teks menjadi token (kata-kata). `nltk.corpus.stopwords` digunakan untuk mendapatkan daftar kata-kata tidak bermakna (kata-kata umum yang dianggap tidak relevan dalam analisis).

Sastrawi adalah pustaka untuk pemrosesan bahasa alami dalam bahasa Indonesia. `Sastrawi.Stemmer.StemmerFactory` digunakan untuk melakukan *stemming* pada kata-kata.

#### 3.2.2.1 Text Cleaning

Proses ini adalah bagian dari tahap *preprocessing* yang bertujuan untuk membersihkan teks dari karakter-karakter khusus, tautan, dan mengonversi teks menjadi huruf kecil. Berikut ini adalah kode yang digunakan untuk proses *text cleaning*.

```
def clean_text(text):
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'^a-zA-Z\s]', '', text)
    text = text.lower()
    return text

data['cleaned_text'] = data['Text'].apply(clean_text)
```

Dalam langkah pertama, fungsi `re.sub` digunakan untuk menghapus tautan dari teks menggunakan ekspresi reguler. `r'http\S+'` adalah pola ekspresi reguler yang cocok dengan URL apa pun yang dimulai dengan "http" atau "https", diikuti oleh karakter-karakter lainnya. `''` (string kosong) digunakan sebagai pengganti untuk menghapus tautan dari teks.

Dalam langkah kedua, fungsi `re.sub` digunakan lagi untuk menghapus semua karakter yang bukan huruf (alphabet) dan tidak termasuk spasi dari teks. `[^a-zA-Z\s]` adalah pola ekspresi reguler yang cocok dengan karakter apa pun yang bukan huruf (baik huruf besar maupun huruf kecil) atau spasi. `''` (string kosong) digunakan sebagai pengganti untuk menghapus karakter-karakter non-alfabet.

Setelah menghilangkan tautan dan karakter non-alfabet, teks dikonversi menjadi huruf kecil menggunakan metode `.lower()`. Ini membantu memastikan konsistensi dalam analisis teks, sehingga huruf besar dan kecil dianggap sama.

Fungsi `clean_text` diterapkan pada setiap baris dari kolom `'Text'` dalam DataFrame data menggunakan metode `.apply()`. Hasil dari fungsi ini disimpan dalam kolom baru yang disebut `'cleaned_text'`. Proses *text cleaning* menunjukkan data yang bersih dari karakter yang tidak diinginkan di dalam proses berikutnya. Pada Tabel 3.2 adalah beberapa data yang sudah dilakukan proses *text cleaning*.

Tabel 3.2 Data Hasil Proses Text Cleaning

No	cleaned_text
1	komitmen ciptakan pemilu damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju
2	fkppi siap kawal amp sukseskan pemilu pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju
3	semua pihak harus terlibat kawal pemilu damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju
4	mari kita kawal pemilu aman amp damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas

No	cleaned_text
	pemilihpemula pemilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju
5	wujud sikap netral aparat dalam pemilu pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pemilihpemula pemilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju
6	mari kawal pelaksanaan pemilu agar berjalan dengan sukses pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pemilihpemula pemilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju
7	kpu menegaskan kegiatan ganjar prabowo anies ketika mendatangi masyarakat bukan kampanye karena mereka belum mendaftar sebagai bacapres pemilu pemiluserentak tolakpenundaanpemilu pemilihpemula pemilihmuda Pancasila pemilu capres kpu bawaslu
8	kawal pemilu dengan damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pemilihpemula pemilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju
9	kpu menegaskan kegiatan ganjar prabowo anies ketika mendatangi masyarakat bukan kampanye karena mereka belum mendaftar sebagai bacapres pemilu pemiluserentak tolakpenundaanpemilu pemilihpemula pemilihmuda Pancasila pemilu capres kpu bawaslu
10	komitmen ciptakan pemilu damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pemilihpemula pemilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju

### 3.2.2.2 Tokenizing

Tahap ini adalah untuk melanjutkan tahap *preprocessing* teks dengan melakukan tokenisasi. Di bawah ini adalah kode yang digunakan untuk tahap ini.

```
def preprocess_text(text):
    text = clean_text(text)
    words = word_tokenize(text)
    words_without_stopwords = [
        word for word in words if word not in stop_words]
    return words_without_stopwords, [word for word in words
if word in stop_words]
data['preprocessed_text'], data['stopword'] = zip(
    *data['Text'].apply(preprocess_text))
```

Pertama, fungsi `clean_text` dipanggil untuk membersihkan teks dari tautan dan karakter non-alfabet. Kemudian, teks yang telah dibersihkan di-tokenisasi menjadi kata-kata individu menggunakan fungsi `word_tokenize` dari pustaka `nltk`. Kata-kata dalam teks yang telah ditokenisasi diperiksa satu per satu. Data hasil *tokenizing* dan sekaligus sudah dilakukan proses *stopword removal* bisa dilihat pada Tabel 3.3.

Tabel 3.3 Data hasil proses tokenizing

No	tokenized_text
1	['komitmen', 'ciptakan', 'pemilu', 'damai', 'pemiluserentak', 'pemilusesuaitahapan', 'tolakpenundaanpemilu', 'antipolitikidentitas', 'pemilihpemula', 'pemilihmuda', 'pancasila', 'pemilu', 'pilpres', 'capres', 'kpu', 'bawaslu', 'indonesiamaju']
2	['fkppi', 'kawal', 'amp', 'sukseskan', 'pemilu', 'pemiluserentak', 'pemilusesuaitahapan', 'tolakpenundaanpemilu', 'antipolitikidentitas', 'pemilihpemula', 'pemilihmuda', 'pancasila', 'pemilu', 'pilpres', 'capres', 'kpu', 'bawaslu', 'indonesiamaju']
3	['terlibat', 'kawal', 'pemilu', 'damai', 'pemiluserentak', 'pemilusesuaitahapan', 'tolakpenundaanpemilu', 'antipolitikidentitas', 'pemilihpemula', 'pemilihmuda', 'pancasila', 'pemilu', 'pilpres', 'capres', 'kpu', 'bawaslu', 'indonesiamaju']
4	['mari', 'kawal', 'pemilu', 'aman', 'amp', 'damai', 'pemiluserentak', 'pemilusesuaitahapan', 'tolakpenundaanpemilu', 'antipolitikidentitas', 'pemilihpemula', 'pemilihmuda', 'pancasila', 'pemilu', 'pilpres', 'capres', 'kpu', 'bawaslu', 'indonesiamaju']
5	['wujud', 'sikap', 'netral', 'aparatus', 'pemilu', 'pemiluserentak', 'pemilusesuaitahapan', 'tolakpenundaanpemilu', 'antipolitikidentitas', 'pemilihpemula', 'pemilihmuda', 'pancasila', 'pemilu', 'pilpres', 'capres', 'kpu', 'bawaslu', 'indonesiamaju']
6	['mari', 'kawal', 'pelaksanaan', 'pemilu', 'berjalan', 'sukses', 'pemiluserentak', 'pemilusesuaitahapan', 'tolakpenundaanpemilu', 'antipolitikidentitas', 'pemilihpemula', 'pemilihmuda', 'pancasila', 'pemilu', 'pilpres', 'capres', 'kpu', 'bawaslu', 'indonesiamaju']
7	['kpu', 'kegiatan', 'ganjar', 'prabowo', 'anies', 'masyarakat', 'kampanye', 'mendaftar', 'bacapres', 'pemilu', 'pemiluserentak', 'tolakpenundaanpemilu', 'pemilihpemula', 'pemilihmuda', 'pancasila', 'pemilu', 'capres', 'kpu', 'bawaslu']
8	['kawal', 'pemilu', 'damai', 'pemiluserentak', 'pemilusesuaitahapan', 'tolakpenundaanpemilu', 'antipolitikidentitas', 'pemilihpemula',



No	tokenized_text
	‘pemilih muda’, ‘pancasila’, ‘pemilu’, ‘pilpres’, ‘capres’, ‘kpu’, ‘bawaslu’, ‘indonesiamaju’]
9	[‘kawal’, ‘ketat’, ‘pemilihan’, ‘pemiluserentak’, ‘pemilusesuaitahapan’, ‘tolakpenundaanpemilu’, ‘antipolitikidentitas’, ‘pemilihpemula’, ‘pemilih muda’, ‘pancasila’, ‘pemilu’, ‘pilpres’, ‘capres’, ‘kpu’, ‘bawaslu’, ‘indonesiamaju’]
10	[‘komitmen’, ‘kawal’, ‘pemilu’, ‘pemiluserentak’, ‘pemilusesuaitahapan’, ‘tolakpenundaanpemilu’, ‘antipolitikidentitas’, ‘pemilihpemula’, ‘pemilih muda’, ‘pancasila’, ‘pemilu’, ‘pilpres’, ‘capres’, ‘kpu’, ‘bawaslu’, ‘indonesiamaju’]

### 3.2.2.3 Stopword Removal

Kata-kata yang termasuk dalam daftar kata tidak bermakna dihilangkan dari daftar kata-kata. Ini membantu menghilangkan kata-kata umum yang cenderung tidak memiliki makna khusus dalam analisis. Hasil dari penghapusan kata yang tidak bermakna dan kata-kata asli yang telah di-tokenisasi dipisahkan menjadi dua kelompok: kata-kata tanpa *stopwords* dan kata-kata dengan *stopwords*.

Hasil ini akan diambil dalam dua kolom terpisah dalam *DataFrame* data, yaitu ‘preprocessed\_text’ dan ‘stopword’. Fungsi preprocess\_text diterapkan pada setiap baris dari kolom ‘Text’ dalam *DataFrame* data menggunakan metode .apply(). Hasil dari fungsi ini, yaitu kata-kata tanpa *stopwords* dan kata-kata dengan *stopwords*, dipisahkan dan disimpan dalam kolom-kolom baru ‘preprocessed\_text’ dan ‘stopword’. Hasil dari proses pemisahan *stopword* tersebut bisa di lihat pada Tabel 3.4.

Tabel 3.4 Data kata yang di hapus pada Stopwords removal

No	Stopword Removal
1	[]
2	[‘siap’]
3	[‘semua’, ‘pihak’, ‘harus’]
4	[‘kita’]
5	[‘dalam’]
6	[‘agar’, ‘dengan’]

No	Stopword Removal
7	['menegaskan', 'ketika', 'mendatangi', 'bukan', 'karena', 'mereka', 'belum', 'sebagai']
8	['dengan']
9	['umum', 'kita']
10	[]

Tabel 3.4 adalah sebgain *stopword* yang di ambil dari setiap data yang ada pada dataset. Jika data tidak memiliki *stopword* maka akan mengembalikan *array* kosong.

#### 3.2.2.4 Stemming

Tahap yang akan dijalankan setelah *tokenizing* dan *stopword* adalah *stemming*. Tahap ini bertujuan untuk mengubah suatu kata menjadi kata dasar. Data yang telah melalui proses *stemming* bisa dilihat pada Tabel 3.5.

Tabel 3.5 Data hasil stemming

No	stemming
1	komitmen cipta milu damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pemilihpemula pemilihmuda pancasila milu pilpres capres kpu bawaslu indonesiamaju
2	fkppi kawal amp sukses milu pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pemilihpemula pemilihmuda pancasila milu pilpres capres kpu bawaslu indonesiamaju
3	libat kawal milu damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pemilihpemula pemilihmuda pancasila milu pilpres capres kpu bawaslu indonesiamaju
4	mari kawal milu aman amp damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pemilihpemula pemilihmuda pancasila milu pilpres capres kpu bawaslu indonesiamaju
5	wujud sikap netral aparat milu pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pemilihpemula pemilihmuda pancasila milu pilpres capres kpu bawaslu indonesiamaju
6	mari kawal laksana milu jalan sukses pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pemilihpemula pemilihmuda pancasila milu pilpres capres kpu bawaslu indonesiamaju

No	stemming
7	kpu giat ganjar prabowo anies masyarakat kampanye daftar bacapres milu pemiluserentak tolakpenundaanpemilu pilihpemula pilihmuda pancasila milu capres kpu bawaslu
8	kawal milu damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila milu pilpres capres kpu bawaslu indonesiamaju
9	kawal ketat pilih pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila milu pilpres capres kpu bawaslu indonesiamaju
10	komitmen kawal milu pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila milu pilpres capres kpu bawaslu indonesiamaju

Tabel 3.5 menunjukkan beberapa data yang berhasil melalui proses *stemming*. Kode yang digunakan untuk tahapan ini adalah sebagai berikut.

```
def stem_and_save_to_csv(input_csv, output_csv):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()

    dataset = pd.read_csv(input_csv)
    limited_dataset = dataset.head(200)

    stemmed_text = []
    for text in limited_dataset['stemming']:
        words = word_tokenize(text)
        words_stemmed = [stemmer.stem(word) for word in words]
        stemmed_text.append(' '.join(words_stemmed))

    limited_dataset['stemming'] = stemmed_text
    limited_dataset.to_csv(output_csv, index=False)
```

Terlebih dahulu, objek *stemmer* dibuat menggunakan Sastrawi untuk melakukan *stemming* pada kata-kata. Fungsi membaca file CSV input yang diberikan dan memuatnya ke dalam DataFrame dataset. Setiap teks dalam kolom

'stemming' dari `limited_dataset` di-tokenisasi menjadi kata-kata individu menggunakan `word_tokenize`. Kata-kata tersebut kemudian di-stem menggunakan stemmer yang telah dibuat sebelumnya. Hasil kata-kata yang telah di-stem di-join kembali menjadi teks yang telah di-stem.

Hasil *stemming* disimpan dalam list `stemmed_text`. Kolom 'stemming' dari `limited_dataset` digantikan dengan hasil *stemming* yang telah disimpan dalam `stemmed_text.DataFrame` `limited_dataset` dengan kolom 'stemming' yang telah di-stem disimpan ke dalam file CSV yang ditentukan oleh `output_csv`. Parameter `index=False` digunakan untuk menghindari menyimpan indeks baris dalam file CSV.

### 3.2.3 Pelabelan Data

Tahapan ini adalah tahap dimana data akan dilabeli dengan cara manual. Tujuan dari pelabelan data (*data labeling*) adalah untuk memberikan label atau kategori tertentu kepada setiap contoh dalam dataset. Label ini mengindikasikan informasi atau makna tertentu tentang contoh tersebut. Pelabelan data sangat penting dalam berbagai jenis tugas analisis data, pembelajaran mesin, dan pengembangan model prediksi. Pelabelan dilakukan pada 1211 data *tweet* setelah melakukan proses *preprocessing*. Pada pelabelan manual di dapatkan hasil positif sebanyak 614 data, netral sebanyak 487 data dan negatif sebanyak 110 data. Data hasil pelabelan manual dapat dilihat pada Tabel 3.6.

Tabel 3.6 Data Hasil Pelabelan Manual

No	cleaned_text	sentiment_label	sentiment
1	komitmen ciptakan pemilu damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila pemilu pilpres capres kpu bawasl indonesiamaju	netral	0.0
2	fkppi siap kawal amp sukseskan pemilu pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila	netral	0.0

No	cleaned_text	sentiment_label	sentiment
	pemilu pilpres capres kpu bawaslu indonesiamaju		
3	semua pihak harus terlibat kawal pemilu damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju	positif	0.1
4	mari kita kawal pemilu aman amp damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju	positif	0.1
5	wujud sikap netral aparat dalam pemilu pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju	netral	0.0
6	mari kawal pelaksanaan pemilu agar berjalan dengan sukses pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju	netral	0.0
7	kpu menegaskan kegiatan ganjar prabowo anies ketika mendatangi masyarakat bukan kampanye karena mereka belum mendaftar sebagai bacapres pemilu pemiluserentak tolakpenundaanpemilu pilihpemula pilihmuda Pancasila pemilu capres kpu bawaslu	netral	0.0
8	kawal pemilu dengan damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju	netral	0.0

No	cleaned_text	sentiment_label	sentiment
9	kpu menegaskan kegiatan ganjar prabowo anies ketika mendatangi masyarakat bukan kampanye karena mereka belum mendaftar sebagai bacapres pemilu pemiluserentak tolakpenundaanpemilu pilihpemula pemilihmuda Pancasila pemilu capres kpu bawaslu	netral	0.0
10	komitmen ciptakan pemilu damai pemiluserentak pemilusesuaitahapan tolakpenundaanpemilu antipolitikidentitas pilihpemula pilihmuda pancasila pemilu pilpres capres kpu bawaslu indonesiamaju	positif	0.1

### 3.2.4 Training Data

*Training* data merupakan proses melatih data menggunakan algoritma Naive Bayes. Pelatihan ini di mulai dengan proses ekstraksi data menggunakan *TF-IDF*. Selanjutnya dilakukan pelatihan data guna mendapatkan model kalsifikasi yang akan di gunakan untuk pengujian. Kode yang digunakan pada proses ini adalah sebagai berikut.

```
# Latih model klasifikasi Naive Bayes
vectorizer = TfidfVectorizer()
X_tfidf = vectorizer.fit_transform(X)

classifier = MultinomialNB()
classifier.fit(X_tfidf, y)

# Pisahkan data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.2, random_state=42)

# Prediksi sentimen pada data uji
X_test_tfidf = vectorizer.transform(X_test)
y_pred = classifier.predict(X_test_tfidf)
```

```
# Cetak laporan klasifikasi
classification_rep = classification_report(
    y_test, y_pred, output_dict=True)
```

Objek `TfidfVectorizer` dibuat untuk mengonversi teks menjadi representasi vektor TF-IDF. Setelah itu pemanggilan `fit_transform` pada vektorisasi ini pada kolom 'preprocessed\_text' dalam `DataFrame` data (yang disimpan dalam `X`) untuk mendapatkan matriks *TF-IDF*. Selanjutnya matriks *TF-IDF* ini digunakan sebagai fitur masukan (`X`) dalam pelatihan model.

Langkah berikutnya objek `MultinomialNB()` dibuat sebagai model klasifikasi Naive Bayes. Pemanggilan `fit` pada model ini dilakukan dengan matriks TF-IDF (`X_tfidf`) sebagai fitur dan kolom 'sentiment\_label' dari `DataFrame` data (yang disimpan dalam `y`) sebagai label. Selanjutnya adalah pemisahan data latih dan uji. Data dibagi menjadi data latih (`X_train`, `y_train`) dan data uji (`X_test`, `y_test`) menggunakan `train_test_split`. 80% data digunakan untuk pelatihan dan 20% digunakan untuk pengujian, `random_state` digunakan untuk memastikan hasil yang konsisten saat kode dijalankan kembali. Berikutnya adalah memprediksi Sentimen pada data uji dengan matriks TF-IDF dihitung untuk data uji (`X_test`) menggunakan `transform` pada vektorisasi. Model Naive Bayes digunakan untuk memprediksi label sentimen pada data uji (`y_pred`) berdasarkan matriks TF-IDF yang telah dihitung.

### 3.2.5 Testing

*Testing* data adalah tahapan untuk menguji performa model yang telah di latih pada *training data*. *Testing* adalah data yang tidak digunakan selama proses *training data* dan digunakan untuk mengevaluasi seberapa baik model dapat melakukan prediksi pada data yang belum pernah dilihat sebelumnya.

```
data = pd.read_excel(uploaded_file)

# Lakukan prediksi pada setiap teks dalam file Excel
predictions = []
for text in data['Text']:
    cleaned_text = clean_text(text)
```

```

preprocessed_text, stopwords = preprocess_text(cleaned_text)
processed_text = ' '.join(preprocessed_text)

# Lakukan vektorisasi menggunakan vectorizer yang sudah dimuat
X_tfidf = vectorizer.transform([processed_text])

# Lakukan prediksi menggunakan model yang sudah dimuat
predicted_sentiment = classifier.predict(X_tfidf)[0]

predictions.append(predicted_sentiment)

```

Kode tersebut digunakan untuk melakukan prediksi sentimen pada setiap teks dalam file Excel yang telah diunggah. Berikut adalah langkah-langkah yang dilakukan dalam kode tersebut:

1. Membaca File Excel:

File Excel yang diunggah dibaca menggunakan `pd.read_excel(uploaded_file)`, yang menghasilkan DataFrame data yang berisi data dari file Excel.

2. Melakukan Prediksi pada Setiap Teks:

Iterasi dilakukan melalui setiap teks dalam kolom 'Text' dari DataFrame data. Setiap teks diawali dengan membersihkan teks (`cleaned_text`), melakukan *preprocessing* (`preprocessed_text`) untuk menghasilkan teks yang telah diolah. Teks yang telah diolah digabungkan kembali menjadi satu string (`processed_text`) karena vectorizer membutuhkan input berupa daftar teks.

3. Vektorisasi Menggunakan Vectorizer yang Sudah Dimuat:

Teks yang telah diolah kemudian diubah menjadi representasi vektor TF-IDF menggunakan vectorizer yang telah dimuat sebelumnya. `vectorizer.transform([processed_text])` digunakan untuk menghasilkan matriks TF-IDF untuk teks tersebut.

4. Prediksi Menggunakan Model yang Sudah Dimuat:

Matriks TF-IDF yang telah dihitung digunakan untuk melakukan prediksi sentimen menggunakan model klasifikasi yang telah dimuat sebelumnya. `classifier.predict(X_tfidf)` digunakan untuk memprediksi sentimen dan mengambil nilai prediksi pertama.



## 5. Penyimpanan Prediksi:

Hasil prediksi sentimen untuk setiap teks disimpan dalam list `predictions`.

### 3.2.6 Klasifikasi

Pada bagian ini akan dilakukan pengujian data, pengujian dilakukan dengan data dari *twitter* yang memiliki tendensi pada salah satu bakal calon presiden. Kode yang di gunakan untuk tahapan pengujian dan klasifikasi adalah sebagai berikut.

```
with open('model/classifier_model.pkl', 'rb') as clf_file:
    classifier = pickle.load(clf_file)

with open('model/vectorizer_model.pkl', 'rb') as vec_file:
    vectorizer = pickle.load(vec_file)
```

Kode diatas adalah untuk memuat model klasifikasi dan vectorizer yang telah disimpan sebelumnya menggunakan modul `pickle`. Hal ini memungkinkan untuk menggunakan model dan vectorizer yang telah dilatih pada tahap sebelumnya untuk melakukan prediksi pada data baru. Selanjutnya adalah membaca file yang di upload oleh user beserta `target_name` yang di gunakan untuk nama dari dataset. Setelah file dan `target_name` di simpan, selanjutnya adalah dapat dilakukan *preprocessing* pada data uji berikut ini adalah kodenya

```
target_name = request.form.get('target_name')
uploaded_file = request.files['file']
```