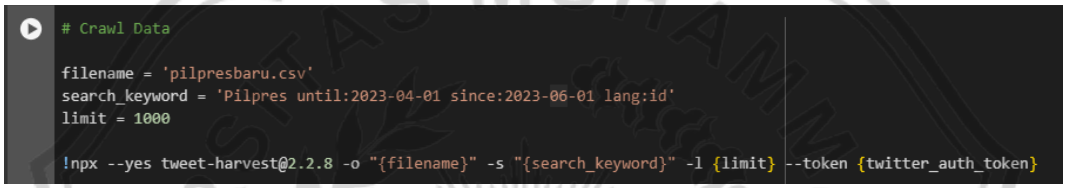
BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Proses pengumpulan data dilakukan sebanyak 2 kali melalui Google Collab dengan menggunakan teknik data crawling pada cuitan atau tweet di media sosial X (Twitter) dengan filter "Pilpres". *Data Crawling* pertama yang dilakukan pada rentang waktu 1 April sampai 1 Juli 2023 menghasilkan 578 data mentah sedangkan *Data Crawling* kedua pada rentang waktu 1 Desember 2023 sampai 1 Januari menghasilkan 246 data mentah.



```
# Crawl Data

filename = 'pilpresbaru.csv'
search_keyword = 'Pilpres until:2023-04-01 since:2023-06-01 lang:id'
limit = 1000

!npx --yes tweet-harvest@2.2.8 -o "{filename}" -s "{search_keyword}" -l {limit} --token {twitter_auth_token}
```

\Gambar 4. 2 Proses Pengumpulan Data

URL	Date	Tweet	ID	Replies	Retweets	Likes
	45097,2903	@Kanseulir Menyejuk	1,67094E+18	0	0	5
https://twit	45097,28896	"Repost dari @partai	1,67094E+18	0	0	0
https://twit	45097,28843	@72Gar7 @tvOneNev	1,67094E+18	2	0	0
https://twit	45097,28837	"Repost dari @partai	1,67094E+18	0	0	0
https://twit	45097,28189	@achmadfauzi_wy Pe	1,67094E+18	9	13	15

Quotes	Conv. ID	Language	Links	Media	Retweeted	Bookmarks	Username
0	1,67077E+18	in				0	one_sach97004
0	1,67094E+18	in				0	mnctrijaya
0	1,67058E+18	in				0	STUDSGUY
0	1,67094E+18	in				0	mnctrijaya
1	1,67094E+18	in				0	FaGtng

Gambar 4. 1 Data yang Dihasilkan

4.2 Pelabelan Data

Data yang telah diperoleh kemudian dikelompokkan menjadi dua kelas, yaitu positif (1), negatif (0). Proses pelabelan dilakukan secara manual dan divalidasi oleh Tia Dwi Putri Hari Nanda, S.Pd. selaku guru Bahasa Indonesia di SMPN 02 Botolinggo Bondowoso. Tabel 3.3 menunjukkan bentuk dataset setelah dilabeli.

4.3 Praproses Teks

Sebelum memulai pelatihan, data harus melalui tahap praproses, yang dalam analisis sentimen sering disebut sebagai text preprocessing. Pada tahap ini, data akan mengalami transformasi tertentu sehingga dapat dimasukkan ke dalam model pelatihan. Beberapa langkah *text preprocessing* yang dilakukan meliputi *case folding*, *tokenizing*, penghapusan *stopword*, dan *stemming*. *Text preprocessing* ini menggunakan beberapa *library* di *Python* seperti *nltk*, *sastrawi*, *re*, dan *pandas.*

4.3.1 Cleansing

Tahap Pembersihan data bertujuan untuk menghapus tanda baca, angka, karakter khusus, spasi berlebih, emoji, URL, mention, hashtag, dan lain sebagainya.

```
#untuk menghilangkan tanda baca, angka, karakter spesial, emotikon, spasi double dll.
def remove_pattern(input_txt, pattern):
    r = re.findall(pattern, input_txt)
    for i in r:
        input_txt = re.sub(i, '', input_txt)
    return input_txt
def remove_tweet_special(text):
    # remove tab, new line, ans back slice
    text = text.replace('\\t'," ").replace('\\n'," ").replace('\\u'," ").replace('\\',"")
    # remove non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')
    # remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)"," ", text).split())
    # remove incomplete URL
    return text.replace("http://", " ").replace("https://", " ")

df['Tweet'] = df['Tweet'].apply(remove_tweet_special)
#menghapus angka
def remove_number(text):
    return  re.sub(r"\d+", "", text)
df['Tweet'] = df['Tweet'].apply(remove_number)
#menghapus tanda baca
def remove_punctuation(text):
    return text.translate(str.maketrans("","",string.punctuation))
df['Tweet'] = df['Tweet'].apply(remove_punctuation)
#menghapus double spasi
def remove_whitespace_LT(text):
    return text.strip()
df['Tweet'] = df['Tweet'].apply(remove_whitespace_LT)
#remove multiple whitespace into single whitespace
def remove_whitespace_multiple(text):
    return re.sub('\s+',' ',text)
df['Tweet'] = df['Tweet'].apply(remove_whitespace_multiple)
# remove single char
def remove_singl_char(text):
    return re.sub(r"\b[a-zA-Z]\b", "", text)
df['Tweet'] = df['Tweet'].apply(remove_singl_char)
df['clean'] = np.vectorize(remove_pattern)(df['Tweet'], "@[\w]*")
df.head(10)
```

Gambar 4. 3 *Source code Cleansing*

4.3.2 *Case folding*

Tujuan dari tahap Case Folding adalah mengubah huruf kapital menjadi huruf kecil.

```
#untuk merubah huruf kapital menjadi huruf kecil
def to_lower(text):
```

Gambar 4. 4 *Source code Case Folding*

4.3.3 Tokenizing

Dalam analisis sentimen dan pemrosesan teks, *Tokenizing* dilakukan untuk membagi teks menjadi unit-unit yang lebih kecil, yang disebut sebagai token. Setiap token mewakili bagian terpisah dari teks, seperti kata, frasa, atau simbol.

```
from nltk.tokenize import RegexpTokenizer

regexp = RegexpTokenizer(r'\w+|$[0-9]+|\S+')
df['Token'] = df['case_folding'].apply(regexp.tokenize)
```

Gambar 4. 5 *Source code Tokenizing*

4.3.4 Stopwords Removal

Penghapusan stopwords, yang dikenal sebagai *Stopwords removal*, adalah tahap dalam pemrosesan teks yang bertujuan untuk menghilangkan kata-kata umum yang tidak memberikan makna khusus dalam teks.

```
# Pastikan kolom 'tidy_text' hanya berisi string
df['Token'] = df['Token'].astype(str)

# Filter elemen yang benar-benar string
df['stopword'] = df['Token'].apply(lambda x: ' '.join([w for w in x.split() if isinstance(w, str) and len(w) > 3]))
```

Gambar 4. 6 *Source code Stopwords Removal*

4.3.5 Stemming

Dalam pemrosesan teks, proses *stemming* dilakukan untuk mengubah kata-kata menjadi bentuk dasarnya atau akar kata. Tujuan utamanya adalah menghilangkan awalan, akhiran, dan imbuhan dari kata-kata sehingga hanya tersisa bentuk dasar atau akar kata.

```
#Stemming the text
factory = StemmerFactory()
ps = factory.create_stemmer()

def simple_stemmer(text):
    text= ' '.join([ps.stem(word) for word in text.split()])
    return text
```

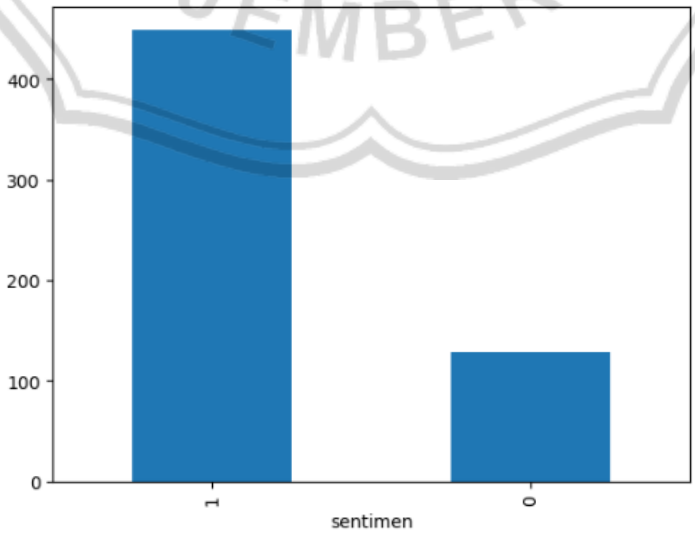
Gambar 4. 7 *Source code Stemming*

Tabel 4. 1 Hasil *Preprocessing*

No	*Cleansing*	*Case folding*	*Tokenizing*	*Stopword*	*Stemming*
1	Menyejukkan Luar Biasa Tapi Bagi Pembenci Anies Baswedan Ini Adalah Pukulan Telak Atas Mereka Yg Ingin Menjegal Anies Baswedan Maju Di Pilpres	menyejukkan luar biasa tapi bagi pembenci anies baswedan ini adalah pukulan telak atas mereka yg ingin menjegal anies baswedan maju di pilpres	[‘menyejukkan’,‘luar’,‘biasa’, ‘pembenci’, ‘anies’, ‘baswedan’, ’pukulan’, ‘telak’ ‘atas’, ‘anies’, ‘baswedan’ ‘maju’,’di’ ‘pilpres’]	[‘menyejukkan’, ‘luar’, ‘biasa’, ‘pembenci’, ‘anies’, ‘baswedan’, ’pukulan’, ‘telak’ ‘atas’, ‘anies’, ‘baswedan’ ‘maju’, ‘pilpres’]	sejuk luar biasa benci anies baswedan pukul telak anies baswedan maju pilpres

4.4 Perbandingan Jumlah Kelas

Proses pelatihan data yang pertama menggunakan masukkan nilai *TF-IDF* dibagi dari hasil *scrapping* periode April sampai Juni 2023 menjadi data latih dan data uji dengan rasio 80:20, 50:50, dan 30:70. Sedangkan perbandingan jumlah data dari total 578 baris data pada setiap kelas adalah 449 untuk kelas 1 (positif) dan 129 untuk kelas 0 (negatif).



Gambar 4. 8 Perbandingan Jumlah Kelas

4.5 Hasil Pelatihan

1. Proses pelatihan pertama menggunakan rasio data 80:20, nilai TF-IDF lalu dilatih memasuki tahap pelatihan dengan rasio 80% data latih yaitu sejumlah 462 data dan 20% data uji sejumlah 116 data menggunakan metode *Naïve Bayes* pada pustaka *sklearn.naive bayes.*

Tabel 4. 2 *Confusion Matrix* 80:20

	Kelas Aktual	
Kelas Prediksi	Positif	Negatif
Positif	TP = 72	FP = 9
Negatif	FN = 17	TN = 18

TP = 72 data yang mempunyai nilai positif. benar dan diprediksi sebagai positif.

FP = 9 data yang mempunyai nilai negatif benar tetapi salah diprediksi sebagai positif.

FN = 17 data yang mempunyai nilai positif benar tetapi salah diprediksi sebagai negatif.

TN = 18 data yang mempunyai nilai negatif benar dan diprediksi dengan benar sebagai negatif

Perhitungan akurasi, presisi, dan *recall:*

$$Akurasi = \frac{(TP + TN)}{TP + FN + FP + TN} = \frac{(72 + 18)}{72 + 17 + 9 + 18} \times 100\% = 77\%$$

$$Presisi = \frac{(TP)}{(TP + FP)} = \frac{(72\ )}{72\ + 9} \times 100\% = 88\%$$

$$Recall = \frac{(TP)}{(TP + FN)} = \frac{(72\ )}{72\ + 17} \times 100\% = 80\%$$

2. Proses pelatihan kedua menggunakan rasio data 50:50, nilai TF-IDF lalu dilatih memasuki tahap pelatihan dengan rasio 50% data latih sejumlah 289 data dan 50% data uji sejumlah 289 data menggunakan metode *Naïve Bayes* pada pustaka *sklearn.naive_bayes.*

Tabel 4. 3 *Confusion Matrix* 50:50

	Kelas Aktual	
Kelas Prediksi	Positif	Negatif
Positif	TP = 195	FP = 27
Negatif	FN = 28	TN = 39

TP = 195 data yang mempunyai nilai positif. benar dan diprediksi sebagai positif.

FP = 27 data yang mempunyai nilai negatif benar tetapi salah diprediksi sebagai positif.

FN = 28 data yang mempunyai nilai positif benar tetapi salah diprediksi sebagai negatif.

TN = 39 data yang mempunyai nilai negatif benar dan diprediksi dengan benar sebagai negatif

Perhitungan akurasi, presisi, dan *recall:*

$$Akurasi = \frac{(TP + TN)}{TP + FN + FP + TN} = \frac{(195 + 39)}{195 + 28 + 27 + 39} \times 100\% = 80\%$$

$$Presisi = \frac{(TP)}{(TP + FP)} = \frac{(195\ )}{195 + 27} \times 100\% = 87\%$$

$$Recall = \frac{(TP)}{(TP + FN)} = \frac{(195\ )}{195\ + 28} \times 100\% = 87\%$$

3. Proses pelatihan ketiga menggunakan rasio data 30:70, nilai TF-IDF lalu dilatih memasuki tahap pelatihan dengan rasio 30% data latih sejumlah 173 data dan 70% data uji sejumlah 405 data menggunakan metode *Naïve Bayes* pada pustaka *sklearn.naive_bayes.*

Tabel 4. 4 *Confusion Matrix* 70:30

	Kelas Aktual	
Kelas Prediksi	Positif	Negatif
Positif	TP = 289	FP = 57
Negatif	FN = 21	TN = 38

TP = 289 data yang mempunyai nilai positif. benar dan diprediksi sebagai positif.

FP = 57 data yang mempunyai nilai negatif benar tetapi salah diprediksi sebagai positif.

FN = 21 data yang mempunyai nilai positif benar tetapi salah diprediksi sebagai negatif.

TN = 38 data yang mempunyai nilai negatif benar dan diprediksi dengan benar sebagai negatif

Perhitungan akurasi, presisi, dan *recall:*

$$Akurasi = \frac{(TP + TN)}{TP + FN + FP + TN} = \frac{(289 + 38)}{289 + 21 + 57 + 38} \times 100\% = 80\%$$

$$Presisi = \frac{(TP)}{(TP + FP)} = \frac{(289)}{289 + 57} \times 100\% = 83\%$$

$$Recall = \frac{(TP)}{(TP + FN)} = \frac{(289)}{289 + 21} \times 100\% = 93\%$$

Tabel 4. 5 Perbandingan Performa pada Setiap Rasio Pembagian Data

Rasio	**Akurasi**	**Presisi**	***Recall***
80:20	77%	88%	80%
50:50	80%	87%	87%
70:30	80%	83%	93%

Tabel 4.4 menjelaskan bahwa pola terbaik dihasilkan oleh rasio pembagian data 50:50. Pernyataan ini didasarkan pada evaluasi performa model menggunakan beberapa metrik, yaitu akurasi, presisi, dan *recall*. Rasio 50:50 memberikan akurasi sebesar 80%, presisi 87%, dan *recall* 87%. Akurasi mencerminkan persentase total prediksi yang benar, sedangkan presisi dan *recall* menggambarkan kemampuan model dalam mengidentifikasi data positif dengan benar. Dengan nilai yang tinggi pada ketiga metrik tersebut, rasio 50:50 dianggap memiliki pola terbaik dalam konteks evaluasi kinerja model klasifikasi.

Sedangkan pada hasil *scrapping* periode Desember 2023 sampai Januari 2024 dengan jumlah data 246 baris menghasilkan Akurasi sebesar 60%, presisi sebesar 50% dan recall sebesar 70%. Dengan kata lain, model cenderung memberikan prediksi positif yang lebih banyak, dalam kasus ini, menghasilkan banyak *False Positive*.

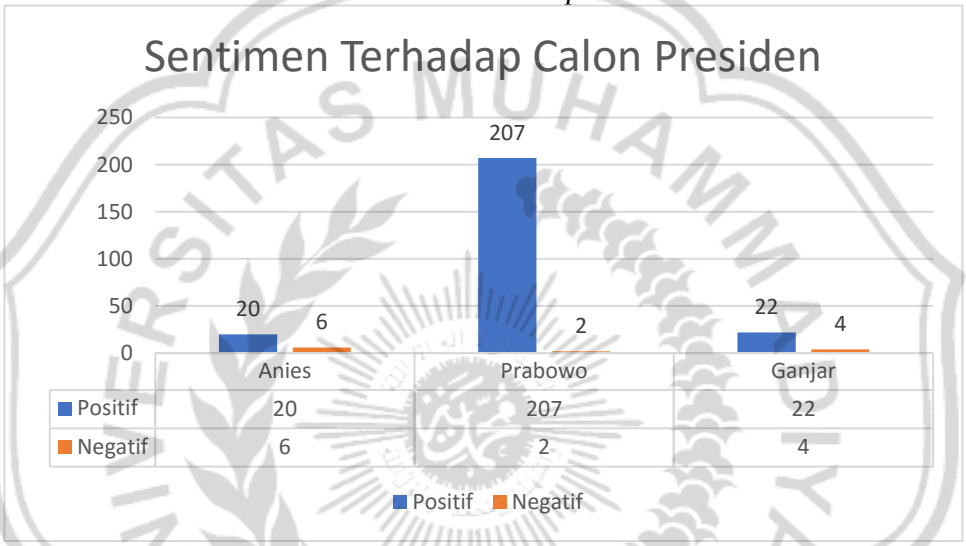
4.6 Analisis Pilpres

Dari hasil *scrapping* data twitter yang telah dilakukan, ada beberapa analisis menarik yang dapat disajikan dalam penelitian ini meliputi:

a. Sentimen Terhadap Calon Presiden

Berdasarkan 578 data, kata ‘anies’ muncul 26 kali, ‘prabowo’ muncul 209 kali, dan ‘ganjar’ muncul 26 kali dengan perincian sentimen pada gambar 4.3.

Gambar 4. 9 Sentimen Terhadap Calon Presiden



b. Distribusi Kata

Dari data keseluruhan, kata yang paling sering keluar adalah kata ‘Pilpres’ sebanyak 331 kali. Disusul dengan kata ‘Pemilu’ sebanyak 279 kali. Dan di nomer urut ke-tiga ada kata ‘Prabowo’ sebanyak 232 kali.

Tabel 4. 6 Distribusi Kata

Kata	**Jumlah**
Pilpres	331
Pemilu	279
Prabowo	232
Dekade	176
Mendingprabowo	176
Terusmajubersamaprabowo	176
Partai	118
Presiden	112
Subianto	79
Politik	68