BAB 4
HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Pada bab ini merupakan hasil dari analisis sentiment menggunakan *Naïve Bayes Classification* untuk melakukan analisis data *tweet* label sentiment positif dan netral dengan kata kunci "Eril Ridwan Kamil" menggunakan data yang diambil pada tanggal 3 Juni – 14 Juni 2022 kemudian mencari kata kunci "Pemilu Ridwan Kamil" dan "Elektabilitas Ridwan Kamil" dengan jumlah seluruh data 11.016 data tweet dan re-tweet. Data yang digunakan dalam tahap training data sebanyak 450 data tweet dengan masing-masing sentiment sebanyak 150. Kemudian menerapkan model klasifikasi Naïve Bayes yang dibuat pada data uji 11.016 data. Didapatkan hasil analisis sentiment dengan sentiment positif sebanyak 5701 dengan nilai presisi sebesar 67.8% dan Negatif 2702 data dengan nilai presisi sebesar 86% dengan nilai rata-rata akurasi sebesar 89.84%.

4.2 HASIL SCRAPING DATA

Dengan menggunakan program scraping yang sudah dibuat dengan dibatu tools Snscrape untuk menggambil data dari Twitter. Data yang diambil merupakan data tweet dan re-tweet dengan keyword "Eril Ridwan Kamil", "Pemilu Ridwan Kamil", dan "Elektabilitas Ridwan Kamil" yang menggunakan Bahasa Indonesia. Dalam data tersebut memiliki beberapa atribut seperti tanggal, username, id tweet, da nisi tweet. berikut merupakan program hasil scraping data dapat dilihat pada Gambar 4.1 dan Gambar 4.2.

```
tweets_list2 = []

for i,tweet in enumerate(sntwitter.TwitterSearchScraper(
    'Eril Ridwan Kamil since:2022-05-27 until:2022-06-14 lang:"id"').get_items()):
    if i>10000:
        break
    tweets_list2.append([tweet.date, tweet.id, tweet.user.username, tweet.content ])

tweets_df2 = pd.DataFrame(tweets_list2, columns=['Datetime', 'Tweet Id', 'Username', 'Text'])
```

Gambar 4.1 Program Scraping

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA

	Datetime	Tweet Id	Username	Text
0	2022-06-13 23:59:30+00:00	1536498529937461248	MarSya82963560	ke Swiss untuk mencari ilmu dan belajar, akan ...
1	2022-06-13 23:59:14+00:00	1536498458928218112	nct_ff	@ridwankamil Turut berduka ya pak semoga amal ...
2	2022-06-13 23:58:45+00:00	1536498337356173312	0713dump	galau bgt liat nabila ma eril:(
3	2022-06-13 23:57:41+00:00	1536498069830660097	MarSya82963560	cukup jadi diri kita sendiri dan lakukan denga...
4	2022-06-13 23:56:38+00:00	1536497808512954370	TSrikandi3	@Srikandi41 @Srikandi53 @Srikandi114 @Srikandi...
...	...	...	...	...
9996	2022-06-13 02:34:00+00:00	1536175022016200704	Hi__nessa	Beneran terharu, liat banyak orang yang melaya...
9997	2022-06-13 02:33:57+00:00	1536175008212365312	norinuri22	Tiba-tiba nangis lihat siaran di tv jelang pem...
9998	2022-06-13 02:33:53+00:00	1536174990093328384	ravika_heddy	@txtdaribandung Merinding masyaallah eril keba...
9999	2022-06-13 02:33:51+00:00	1536174982522232833	croissamericano	Kek howww? Gimana bisa sampai yg nganterin kel...
10000	2022-06-13 02:33:48+00:00	1536174970724052992	cakrxwala	oh iya macet karena banyak yang kasih bunga ke...

Gambar 4.2 Hasil Scraping Eril Ridwan Kamil

Kemudian pada Gambar 4.3 dan 4.4 merupakan program dan hasil scraping menggunakan keyword "Pemilu Ridwan Kamil" dari tanggal 1 Januari – 14 Juni 2022 dan didapatkan data sebanyak 489 *tweet*.

```
tweets_list2 = []

for i,tweet in enumerate(sntwitter.TwitterSearchScraper(
    'Pemilu Ridwan Kamil since:2022-01-01 until:2022-06-14 lang:"id"').get_items()):
    if i>10000:
        break
    tweets_list2.append([tweet.date, tweet.id, tweet.user.username, tweet.content ])

tweets_df2 = pd.DataFrame(tweets_list2, columns=['Datetime', 'Tweet Id', 'Username', 'Text'])
```

Gambar 4.3 Program Scraping Keyword Pemilu

Unnamed: 0	Datetime	Tweet Id	Username	Text
485	2022-01-11 04:55:06+00:00	1480765188290740224	21BeritaTerkini	Pemilu 2024, Ridwan Kamil Kaget dan Merasa Ter...
486	2022-01-09 11:32:44+00:00	1480140480822140928	CakNur971	@Rkri_Indonesia @walian76 @ridwankamil @rksant...
487	2022-01-08 15:57:55+00:00	1479844827986468871	AfiikaZ	Pemilu Kian Dekat, Para Tokoh dan Politikus Po...
488	2022-01-07 10:04:18+00:00	1479393447630888960	tadanoryandesu	@ridwankamil kalo @mas_triadhianto sampe kenak...
489	2022-01-01 09:43:28+00:00	1477213877054050306	AfiikaZ	Era Baru atau Status Quo Pemilu Presiden? http...

Gambar 4.4 Hasil Scraping Keyword Pemilu

Pada Gambar 4.5 dan Gambar 4.6 merupakan program dan hasil scraping dengan keyword "Elektabilitas Ridwan Kamil" dan didapatkan sebanyak 525 data *tweet*.

UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA PERPUSTAKAAN

```
tweets_list2 = []

for i,tweet in enumerate(sntwitter.TwitterSearchScraper(
    'Elektabilitas Ridwan Kamil since:2022-01-01 until:2022-06-14 lang:"id"').get_items()):
    if i>10000:
        break
    tweets_list2.append([tweet.date, tweet.id, tweet.user.username, tweet.content ])

tweets_df2 = pd.DataFrame(tweets_list2, columns=['Datetime', 'Tweet Id', 'Username', 'Text'])
```

Gambar 4.5 Program Scraping Keyword Rlektabilitas

Unnamed: 0	Datetime	Tweet Id	Username	Text
521	2022-01-03 11:24:27+00:00	1477964068489097218	AfiikaZ	Survei PSI, Nama Airlangga raih Elektabilitas ...
522	2022-01-03 10:38:18+00:00	1477952453320458240	AfiikaZ	Beda Nasib PKB dan Cak Imin Soal Elektabilitas...
523	2022-01-03 04:27:20+00:00	1477859096564940804	dhenandar0770	@ridwankamil Bahagia na jelma nu boga anak mah...
524	2022-01-02 10:33:18+00:00	1477588806073151488	AfiikaZ	Elektabilitas Prabowo dan Ganjar Bersaing Keta...
525	2022-01-02 05:34:09+00:00	1477513522732879876	AfiikaZ	Elektabilitas Anies Baswedan di Pilpres 2024 T...

Gambar 4.6 Hasil Scraping Keyword Elektabilitas

Hasil dari scraping menggunakan Snscrape dengan jumlah seluruh data *tweet* sebanyak 11.016 dan memiliki atribut Datetime, Tweet id, Text, dan Username.

4.3 HASIL PREPROCESSING DATA

Setelah melakukan scraping data, kemudian tweet-tweet yang sudah di dapatkan akan dilakukan pembersihan data menggunakan metode preprocessing data. Data yang akan di preprocessing merupakan data yang berada di atribut Text yang merupakan isi dari konten *tweet*. Dalam proses preprocessing data ada beberapa tahapan yang perlu dilakukan seperti cleaning data, data reduction, dan lainnya. Tahapan preprocessing data dilakukan satu per satu terhadap data *tweet* yang sudah diambil.

4.3.1 Casefolding

Berikut hasil setelah dilakukannya casefolding dengan program yang sudah dibuat dapat dilihat pada Gambar 4.7 dan Gambar 4.8

UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA PERPUSTAKAAN

```
def lowercase():
    lower_word = df['Text'].str.lower()
    return lower_word

df["Tweet"] = lowercase()
df
```

Gambar 3.7 Kode Program Casefolding

Dengan menggunakan perintah str.lowercase untuk mengubah kata dari huruf kapital menjadi kecil. Pada gambar 3.5 berikut merupakan program untuk melakukan *casefolding* :

	Text	Tweet
0	ke Swiss untuk mencari ilmu dan belajar, akan ...	ke swiss untuk mencari ilmu dan belajar, akan ...
1	@ridwankamil Turut berduka ya pak semoga amal ...	@ridwankamil turut berduka ya pak semoga amal ...
2	galau bgt liat nabila ma eril:(	galau bgt liat nabila ma eril:(
3	cukup jadi diri kita sendiri dan lakukan denga...	cukup jadi diri kita sendiri dan lakukan denga...
4	@Srikandi41 @Srikandi53 @Srikandi114 @Srikandi...	@srikandi41 @srikandi53 @srikandi114 @srikandi...

Gambar 4.8 Hasil Casefolding

4.3.2 Cleaning Data

Kemudian melakukan cleaining data menggunakan beberapa perintah untuk menghapus angka, huruf spesial, username Twitter, dll. Dengan menggunakan program yang dibuat pada Gambar 4.9

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA

```
def url_remove(tweet):
    t = re.sub(r'http\S+', '',tweet)
    return t
df['Tweet'] = df['Tweet'].apply(lambda x: url_remove(x))

def hashtag_remove(tweet):
    reg = "#(\w+:\/\/\S+)"
    return re.sub(reg, " ", tweet)
df['Tweet'] = df['Tweet'].apply(lambda x: hashtag_remove(x))

def remove_user(tweet):
    t = re.sub('@[^\s]+','',tweet)
    return t
df['Tweet'] = df['Tweet'].apply(lambda x: remove_user(x))

def remove_tweet_special(tweet):
    tweet = tweet.replace('\\t', " ").replace('\\n', " ").replace('\\u', " ").replace('\\', "")
    tweet = tweet.encode('ascii','replace').decode('ascii')
    tweet = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)"," ", tweet).split())
    return tweet.replace("http://", " ").replace("https://", " ")
df['Tweet'] = df['Tweet'].apply(lambda x: remove_tweet_special(x))

def rt_remove(tweet):
    t = re.sub(r'RT[\s]+', '', tweet)
    return t
df['Tweet'] = df['Tweet'].apply(lambda x: rt_remove(x))

def remove_number(tweet):
    return re.sub(r"\d+", "", tweet)
df['Tweet'] = df['Tweet'].apply(lambda x: remove_number(x))
```

Gambar 4.9 Kode Program Cleaning Data

Didapatkan hasil cleaning data yang dapat dilihat pada Gambar 3.8 didapatkan dokumen yang lebih bersih dari sebelunya yang masih terdapat symbol-simbol, angka, username, dll.

Tweet	Cleaned
ke swiss untuk mencari ilmu dan belajar, akan ...	ke swiss untuk mencari ilmu dan belajar akan t...
@ridwankamil turut berduka ya pak semoga amal ...	turut berduka ya pak semoga amal ibadah mas er...
galau bgt liat nabila ma eril:(	galau bgt liat nabila ma eril
cukup jadi diri kita sendiri dan lakukan denga...	cukup jadi diri kita sendiri dan lakukan denga...
@srikandi41 @srikandi53 @srikandi114 @srikandi...	seriuus itu info nya mbak aku baru tahu maklum...

Gambar 4.10 Hasil Cleaning Data

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA

4.3.3 Tokenizing

Pada tahap *Tokenizing* bertujuan untuk melakukan pemecahan atau pemisahan karakter dalam suatu teks yang didefinisikan sebagai pemisah kata atau bukan dan nantinya akan mempermudah melakukan Stopword Removal. Berikut pada Gambar 4.11 merupakan program untuk melakukan tokenizing

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

df['Text_Tokens'] = df['Tweet'].apply(lambda x: word_tokenize_wrapper(x))
```

Gambar 4.11 Kode Program Tokenizing

Hasil dari proses tokenizing dapat dilihat pada Gambar 4.12 dengan memisahkan setiap kata dalam dokumen

	Cleaned	Text_Tokens
0	ke swiss untuk mencari ilmu dan belajar akan t...	[ke, swiss, untuk, mencari, ilmu, dan, belajar...
1	turut berduka ya pak semoga amal ibadah mas er...	[turut, berduka, ya, pak, semoga, amal, ibadah...
2	galau bgt liat nabila ma eril	[galau, bgt, liat, nabila, ma, eril]
3	cukup jadi diri kita sendiri dan lakukan denga...	[cukup, jadi, diri, kita, sendiri, dan, lakuka...
4	seriuus itu info nya mbak aku baru tahu maklum...	[seriuus, itu, info, nya, mbak, aku, baru, tah...

Gambar 4.12 Hasil Tokenzing

4.3.4 Stopword Removal

Setelah melakukan penyelisihan kata yang terdapat dalam teks yang sudah di tokenize. Kode program stopword removal yang dapat dilihat pada Gambar 4.13 dan hasilnya nya dapa Gambar 4.14.

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from nltk.corpus import stopwords

list_stopwords = stopwords.words('Indonesian')

txt_stopword = pd.read_csv("id.stopwords.02.01.2016.txt", names = ["Stopword"], header = None)
list_stopwords.extend(txt_stopword["Stopword"])

list_stopwords = set(list_stopwords)

def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]

df["Stopword"] = df["Text_Tokens"].apply(lambda x:stopwords_removal(x))

df.head()
```

Gambar 4.13 Program Stopword Removal

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA

Dengan menggunakan stopword removal beberapa kata yang termasuk dalam stopword akan dihilangkan.

	Text_Tokens	Stopword
0	[ke, swiss, untuk, mencari, ilmu, dan, belajar...	[swiss, mencari, ilmu, belajar, allah, menakdi...
1	[turut, berduka, ya, pak, semoga, amal, ibadah...	[berduka, ya, semoga, amal, ibadah, mas, eril,...
2	[galau, bgt, liat, nabila, ma, eril]	[galau, bgt, liat, nabila, ma, eril]
3	[cukup, jadi, diri, kita, sendiri, dan, lakuka...	[lakukan, hati, tuhan, membedabedakan, profesi...
4	[seriuus, itu, info, nya, mbak, aku, baru, tah...	[seriuus, info, nya, mbak, maklum, mbak, jaran...

Gambar 4.14 Hasil Stopword Removal

4.3.5 Stemming

Dengan melakukan penghilangan infleksi kata menjadi bentuk dasarnya, misalkan kata “menjadi” dan dilakukan *stemming* akan menghasilkan kata “jadi”. Kode program stemming dapat dilihat pada Gambar 4.15:

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter

factory = StemmerFactory()
stemmer = factory.create_stemmer()

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

term_dict = {}

for document in df["Stopword"]:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

print(len(term_dict))
print("------------------------------")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term, ":", term_dict[term])

print(term_dict)
print("------------------------------")

def stemmed_term(document):
    return [term_dict[term] for term in document]

df["Stopword"] = df["Stopword"].apply(lambda x:stemmed_term(x))

print(df["Stopword"])
```

Gambar 4.15 Kode Program Stemming

UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA PERPUSTAKAAN

Setelah melakukan penghapusan infleksi kata atau imbuhan kata didapatkan hasil seperti yang terlihat pada Gambar 4.16

```
swiss : swiss
mencari : cari
ilmu : ilmu
belajar : ajar
allah : allah
menakdirkan : takdir
eril : eril
guru : guru
kehidupan : hidup
```

Gambar 4.16 Hasil Stemming

4.3.6 Normalization

Normalization yaitu mengubah kata yang sebelumnya merupakan singkatan dan kata tidak baku diubah menjadi kata baku. Dapat dilihat pada Gambar 4.17 merupakan program untuk melakukan nomalisasi data

```
normalized_word = pd.read_excel("normalization.xlsx")

normalized_word_dict = {}

for index, row in normalized_word.iterrows():
    if row[0] not in normalized_word_dict:
        normalized_word_dict[row[0]] = row[1]

def normalized_term(document):
    return [normalized_word_dict[term] if term in normalized_word_dict else term for term in document]

normal_tweet = df["Stopword"].apply(normalized_term).str.join(" ")

print(normal_tweet)
```

Gambar 4.17 Kode Program Normalization

Berikut merupakan hasil setelah dilakukannya normalisasi data, dapat dilihat pada Gambar 4.18.

```
0       swiss cari ilmu ajar allah takdir eril guru hi...
1       duka ya semoga amal ibadah mas eril terima ama...
2                       galau banget lihat nabila ma eril
3       laku hati tuhan membedabedakan profesi selamat...
4       seriuus info nya mbak maklum mbak jarang nonto...
```

Gambar 4.18 Hasil Normalisasi

UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA PERPUSTAKAAN

Berdasarkan Gambar 4.19 merupakan hasil akhir setelah dilakukannya berbagai hal dalam *preprocessing data*.

```
0        swiss cari ilmu ajar allah takdir eril guru hi...
1        duka ya semoga amal ibadah mas eril terima ama...
2                        galau banget lihat nabila ma eril
3        laku hati tuhan membedabedakan profesi selamat...
4        seriuus info nya mbak maklum mbak jarang nonto...
                                ...
9996     beneran haru lihat orang layat trus nganterin ...
9997         tibatiba nang lihat siar tv jelang makam eril
9998     merinding masyaallah eril baik yang buat sampe...
9999     kaya howww gimana yang nganterin liang lahat d...
10000                        oh iya macet kasih bunga eril
Name: Stopword, Length: 10001, dtype: object
```

Gambar 4.19 Hasil Preprocessing Data

4.4 HASIL PELABELAN DATA

Tahapan ini nantinya akan menentukan sentiment sebuah kalimat apakah bersifat positif, netral, atau negative. Pada tahap pelabelan data ini akan dilakukan secara otomatis dan manual terhadap 450 data dan dibagi menjadi sentiment positif sebanyak 150, netral 150 dan negative 150. Saat pelabelan data dibantu menggunakan library VaderSentiment supaya mendapatkan hasil yang obyektif dan data tersebut digabungkan dari data tweet masing-masing keyword 150 *tweet*, masing-masing sentiment 50 *tweet*. Kemudian mengedit beberapa hasil nya supaya mendapatkan sentiment yang sesuai. Kode program untuk *VaderSentiment* dapat dilihat pada Gambar 4.20.

UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA PERPUSTAKAAN

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()

scores = [analyzer.polarity_scores(x) for x in tweet_df['Tweet']]
print(scores)
tweet_df['Compound_Score'] = [x['compound'] for x in scores]

tweet_df.loc[tweet_df['Compound_Score'] < 0, 'Sentiment'] = 'Negatif'
tweet_df.loc[tweet_df['Compound_Score'] == 0, 'Sentiment'] = 'Netral'
tweet_df.loc[tweet_df['Compound_Score'] > 0, 'Sentiment'] = 'Positif'

tweet_df
```

Gambar 4.20 Kode Program Vader Sentimen

Setelah melakukan labeling menggunakan VaderSentimen kemudian melakukan cek terhadap sentimen yang sudah dilabel, sehingga mendapatkan sentimen yang diinginkan

	Sentiment	Label	Tweet
0	Positif	1	simulasi elektabilitas wakil presiden sandiaga...
1	Positif	1	harap pilih caprescawapres anies baswedan ridw...
2	Positif	1	kpu pangandaran laksana tahap milu siap tingga...
3	Positif	1	drpd jd presiden mending jaauuh jd presiden cu...
4	Positif	1	hijab gamis sorban peci budaya arab budaya jel...
...	...	...	...
445	Negatif	-1	eril makam sandiaga doa ridwan kamil keluarga ...
446	Negatif	-1	ribu ziarah sambang makam eril cimaung jawa
447	Negatif	-1	ribu ziarah sambang makam eril cimaung jawa
448	Negatif	-1	masyaallahsaya warga jabar kemarin ngikutin be...
449	Negatif	-1	ridwan kamil cerita eril makam

450 rows × 3 columns

Gambar 4.21 Hasil Pelabelan Manual

Hasil sentimen dapat dilihat pada Gambar 4.21 kalimat-kalimat yang dipilih untuk sentimen positif yaitu kalimat-kalimat yang memiliki nilai yang baik didalam kalimat, sentimen negatif berisi kalimat tweet yang menjelek-jelekan elektabilitas Ridwan Kamil, sedangkan dokumen Netral tidak mempunyai nilai didalamnya headline sebuah berita.

UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA PERPUSTAKAAN

```
Sentiment
Negatif    150
Netral     150
Positif    150
dtype: int64
```

Gambar 4.22 Jumlah Label

Pada Gambar 4.22 merupakan jumlah dengan masing-masing sentiment memiliki jumlah yang sama yaitu 150 data positif, 150 data netral dan 150 data negatif.

4.5 HASIL TRAINING DATA

Setelah melakukan labeling data, tahap penelitian selanjutnya adalah training data, menggunakan modul scikitlearn untuk menghitung bobot kata setiap tweet, yang nantinya akan dijadikan model klasifikasi pada testing data. Dengan melakukan import beberapa modul dari library sklearn untuk melakukan training data seperti TfidfVectorizer untuk melakukan perhitungan bobot pada setiap kalimat tweet, kemudian MultinomialNB untuk melakukan perhitungan Naïve Bayes dan yang terakhir roc_auc_score untuk melihat informasi kinerja algoritma klasifikasi.

4.5.1 TF-IDF

dengan membuat fungsi untuk menjalankan TfidfVectorizer dengan nama feature_extraction untuk mengkonversi dokumen menjadi data metrix fitur TF-IDF kemudian menghitung bobot dokumen keseluruhan dari sebuah kata, kemudian menyesuaikan data yang diambil dan mengkonversi data tersebut menggunakan fungsi ‘fit.transform’ untuk mendapatkan hasil data metric. Dengan melakukan perhitungan TF untuk mencari nilai kemunculan sebuah kata dalam dokumen, kemudian perhitungan IDF untuk mencari nilai kemunculan kata dari keseluruhan dokumen.Dapat dilihat pada gambar 4.23 merupakan Program perhitungan TF-IDF.

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA

```
def feature_extraction(data, method = "tfidf"):
    tfv=TfidfVectorizer(sublinear_tf=True)
    features=tfv.fit_transform(data)
    return features
```

Gambar 4.23 Kode Program Perhitungan TF-IDF

```
(0, 1515)     0.40247515972626635
(0, 1123)     0.35273376311714727
(0, 631)      0.11027086146397752
(0, 1211)     0.11150461677298808
(0, 1517)     0.38183061486698044
(0, 1251)     0.36581746181957003
(0, 1136)     0.15682232552482592
(0, 1550)     0.40247515972626635
(0, 381)      0.19117731251327985
(0, 1325)     0.43157201147609947
(1, 52)       0.4319528937623994
(1, 1570)     0.256531319879888
(1, 70)       0.3821675980672404
(1, 881)      0.1252818778167525
(1, 173)      0.32392253592927545
(1, 88)       0.2782426432629353
(1, 259)      0.402830362693417
(1, 1102)     0.30325977130309884
(1, 479)      0.3530450669982579
(1, 631)      0.11036818060588159
(1, 1211)     0.11160302476108776
(2, 1457)     0.26928580611542197
(2, 1310)     0.36461376860772293
(2, 1398)     0.36461376860772293
(2, 757)      0.5856787159749144
```

Gambar 4.24 Hasil TF-IDF Training

Berdasarkan Gambar 4.24 merupakan hasil dari fungsi tf-idf yang sudah dikalankan didapatkan hasil ekstrasi tf-idf dan bobot pada masing-masing kata dalam dokumen tweet.

4.5.2 Naïve Bayes Classification

Setelah melakukan perhitungan Naïve Bayes menggunakan program yang sudah di buat pada Gambar 4.25

```
def train_classifier(features, label, classifier = "naive_bayes"):
    model = MultinomialNB()
    model.fit(features, label)
    probability_to_be_positive = model.predict_proba(features)[:,1]
```

Gambar 4.25 Kode Program Naïve Bayes

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA

dengan menghitung nilai probabilitas pada setiap kata yang sudah dilakukan pembobotan menggunakan TF-IDF dan menghitung prediksi nilai positif dari nilai negatif dan didapatkan hasil seperti yang terlihat pada Gambar 4.26

```
train = train_classifier(features, label, "naive_bayes")

print(train)
```

```
auc (train data): 0.9968888888888889
top 10 scores:  [0.5503447  0.53195915 0.74991115 0.6198192  0.45264322 0.91416601
 0.81472604 0.5764107  0.80051685 0.74383355]
None
```

Gambar 4.26 Hasil Program Naïve Bayes

Berdasarkan Gambar 4.26 ditampilkan hasil setelah menjalankan fungsi perhitungan Naïve Bayes Classification terhadap kata yang sudah dilakukan pembobotan yang bersifat positif sebesar 99.6% dengan beberapa nilai 10 tertinggi dari fitur yang dijalankan.

4.5.3 Klasifikasi Training Data

Setelah melakukan klasifikasi pada proses *training data* menggunakan metode Naïve Bayes sebanyak 450 data tweet yang sudah diberi label masing-masing 150 data *tweet* positif dan 150 data *tweet* netral. Untuk mengetahui apakah pemodelan yang sudah dibuat sudah akurat kemudian perlu dilakukannya pengujian menggunakan confusion matrix dengan memetakan sentimen menjadi kelas actual dan kelas prediksi.

Berikut merupakan kode program untuk pembuatan model klasifikasi dapat dilihat pada Gambar 4.27.

```
X_train,X_test,y_train,y_test = train_test_split(data['Tweet'],data['Polarity'],test_size=0.1,random_state=100)

pipeline = Pipeline([
    ('bow',CountVectorizer()),
    ('tfidf',TfidfTransformer()),
    ('classifier',MultinomialNB())
    ])

X_train = np.asarray(X)
pipeline = pipeline.fit(X_train, np.asarray(y))
```

Gambar 4.27 Membuat Model Klasifikasi

Kemudian model akan disimpan kedalam bentuk file pickle agar dapat dibuka kembali dan digunakan lagi. Kode program untuk menyimpan file pickle dapat

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA

dilihat pada Gambar 4.28 dan untuk membuka file pickle dapat dilihat pada Gambar 4.29.

```
file_data = 'eril.pickle'
f = open(file_data, 'wb')
pickle.dump(pipeline, f)
f.close()
```

Gambar 4.28 Simpan Pickle

```
file_dataset = 'eril.pickle'
f = open(file_dataset, 'rb')
training = pickle.load(f)
f.close()
print(training)
```

```
Pipeline(steps=[('bow', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('classifier', MultinomialNB())])
```

Gambar 4.29 Membuka File Pickle

Dengan melakukan perhitungan akurasi terhadap kelas prediksi dan kelas actual sehingga didapatkan nilai akurasi 74.4%. Dengan rincian tabel hasil klasifikasi yang dapat dilihat pada Gambar 4.230

```
print(accuracy_score(Y_test,Y_pred))
```

```
0.7444444444444445
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, Y_pred)
print(cm)
```

```
[[15  3  5]
 [ 4 29  2]
 [ 5  4 23]]
```

```
from sklearn.metrics import classification_report
report = classification_report(Y_test, Y_pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.62      0.65      0.64        23
           1       0.81      0.83      0.82        35
           2       0.77      0.72      0.74        32

    accuracy                           0.74        90
   macro avg       0.73      0.73      0.73        90
weighted avg       0.75      0.74      0.74        90
```

Gambar 4.30 Klasifikikasi Training Data

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA

Dapat dilihat pada Gambar 4.30 dengan hasil nilai presisi masing-masing setiment mendapatkan nilai presisi netral 62%, positif 81%, dan negatif 77%. Nilai recall netral 65%, positif 83% dan negatif 72%.

4.6 HASIL TESTING DAN ANALISIS

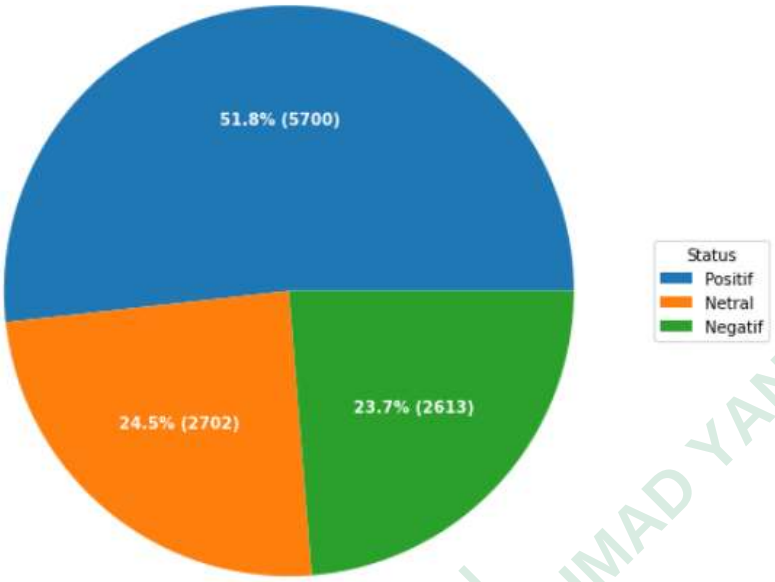
Sebelum dilakukan testing data tweet digabung menjadi satu dari setiap keyword yang diambil kemudian melakukan prediksi, berikut merupakan kode program untuk melakukan prediksi dapat dilihat pada Gambar 4.31.

```
prediction = training.predict(np.asarray(data_tweet))

prediction

result = []

for i in range(len(prediction)):
    if(prediction[i]==1):
        sentiment = 'Positif'
    elif(prediction[i]==0):
        sentiment = 'Netral'
    else:
        sentiment = 'Negatif'
--------
    result.append({'Tweet':data_tweet[i],'Label':prediction[i],'Kelas':sentiment})
----
data = pd.DataFrame(result)

data
```

Gambar 4.31 Kode Program Prediksi

Setelah melakukan prediksi data *tweet* dengan jumlah 11.016 dan didapatkan hasil sentimen positif sebanyak 5700 atau 51.8%, Netral 2702 atau 24.5%, dan negatif sebanyak 2613 atau 23.7%, diagram hasil prediksi dapat dilihat pada Gambar 4.32.

UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA PERPUSTAKAAN



Gambar 4.32 Hasil Prediksi

Setelah didapatkan hasil prediksi kemudian data sentimen Netral dihapus supaya mendapatkan klasifikasi yang baik, karena analisis yang diperlukan hanya sentimen Positif dan Negatif.

Pada langkah hasil analisis yaitu melakukan perhitungan pada seluruh data *tweet* yang sudah diambil sebanyak 11.016 data *tweet*. Pada pengujian data *tweet* yang dilakukan terhadap 11.016 data prediksi berhasil mendapatkan nilai rata-rata akurasi yang cukup bagus sebesar 87.9% dan dapat diketahui bahwa hasil prediksi menggunakan Naïve Bayes yang sudah dibuat memiliki nilai yang baik dalam memberikan nilai sentimen pada data *tweet* yang disediakan. Mengenai rincian hasil klasifikasi data uji dapat dilihat pada Gambar 4.33

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, Y_pred)
print(cm)
```

```
[[1039  112]
 [  91  439]]
```

```
print(accuracy_score(Y_test,Y_pred))
```

```
0.8792385484830458
```

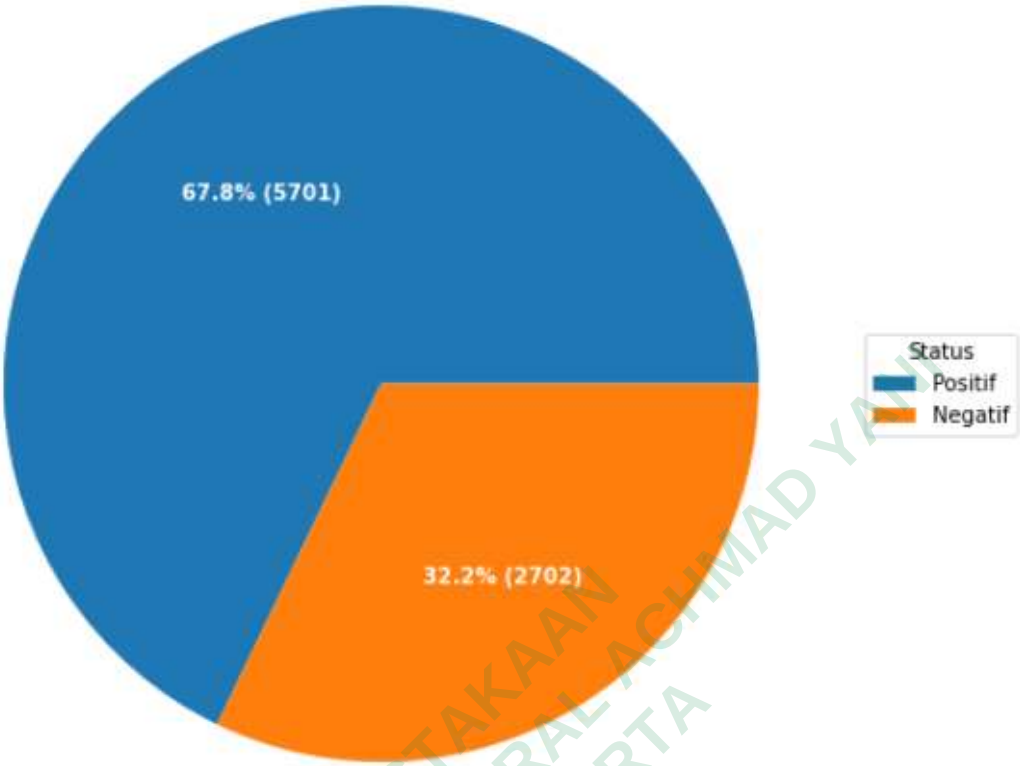
```
from sklearn.metrics import classification_report
report = classification_report(Y_test, Y_pred)
print(report)
```

```
              precision    recall  f1-score   support

           1       0.92      0.90      0.91      1151
           2       0.80      0.83      0.81       530

    accuracy                           0.88      1681
   macro avg       0.86      0.87      0.86      1681
weighted avg       0.88      0.88      0.88      1681
```

Gambar 4.33 Klasifikasi Data Testing

Dengan melakukan confussion matrix didapatkan hasil TP (True Positif) = 1039, FP(Fasle Positif) = 112, TN (True Negatif) = 439, FN (False Negatif) = 91. Dengan melakukan perhitungan prediksi pada *confussion matrix* rumus TP/(TP+FP) kemudian menerapkan rumus tersebut pada data sentimen positif didapatkan perhitungan 1039/(1039+112) didapatkan hasil *precision* sentimen positif 0,92, menghitung nilai recall dengan rumus TP/(TP+FN) dan jika dihitung pada sentimen positif maka 1039/(1039+91) maka didapatkan hasil *recall* sentimen positif 0.90 Kemudian hasil dari prediksi pada data *tweet* keseluruhan dan sudah diketahui data yang berlabel positif dan data yang berlabel netral dengan rincian yang dapat dilihat pada Gambar 4.34

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA



Gambar 4.34 Histogram Sentimen Positif dan Negatif

Berdasarkan pada Gambar 4.34 hasil klasifikasi yang sudah dilakukan didapatkan data berlabel positif sebanyak 5701 data atau sebesar 67.8% dan data berlabel negatif 2702 data atau sebesar 32.2%. Berikut beberapa contoh *tweet* yang masuk dalam kategori positif dan negatif. Dapat dilihat pada Gambar 4.35 Sentimen positif pada klasifikasi ini membahas tentang beberapa hal seperti masyarakat yang turut berduka dan berbelasungkawa terhadap Eril sekeluarga, kebaikan Eril yang pernah ia lakukan selama hidupnya, dan beberapa mengenai Elektabilitas Ridwan Kamil yang tinggi.

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA

```
0       pan jalin komunikasi dgn ridwan kamil salah ka...
1       kang emil jgn jadi dari naik elektabilitas mmg...
2       dapat amat peter gonta dorong pasang ganjarrid...
3       jamiluddin amin elektabilitas ganjar ridwan ka...
4       ridwan kamil dinggap layak usung milik potensi...
                                ...
5696          ridwan kamil jenazah eril indonesia minggu
5697    allahu akbar alhamdulillah ya allah jenazah er...
5698    antar istirahat bangga bahagia putera sayang s...
5699                 kondisi jasad eril khan ridwan kamil
5700    innalillahi wainnailaihi rojiun duka cita insy...
Name: Tweet, Length: 5701, dtype: object
```

Gambar 4.35 Sentimen Positif

Sedangkan pada sentimen negatif data tweet beberapa membahas tentang hal-hal seperti eril hilang saja dibilang naikin elektabilitas, hal apa yang terjadi saat pemakaman Eril dan pemberitaan mengenai Ridwan Kamil elektabilitas Ridwan Kamil. Berikut beberapa contoh tweet dengan sentimen negatif dapat dilihat pada Gambar 4.36.

```
5701    jahat banget sm orang yang bilang kalau ridwan...
5702    eril tinggal saja netizen bdoh ngatain naikin ...
5703    eril hilang saja bilang naikin elektabilitas r...
5704    elektabilitas ridwan kamil dedi mulyadi is mou...
5705    kepala pin ridwan kamil ahaaa sy hanyut anak s...
                                ...
8398    ridwan kamil jasad eril wangi syukur belai man...
8399       momen ridwan kamil adzankan eril telinga lahir
8400    kemendagri benar ridwan kamil cuti jemput jena...
8401    banget keluarga ridwan kamil ketemu udah tingg...
8402    momen haru video call ataliaridwan kamil mandi...
Name: Tweet, Length: 2702, dtype: object
```

Gambar 4.36 Sentimen Negatif

Berdasarkan hasil klasifikasi sentiment menggunakan Naïve Bayes pada data *tweet* mengenai Meninggalnya Almarhum Emmeril Kahn dan Elektabilitas Pemilu Ridwan Kamil. Banyak opini masyarakat mengenai Almarhum Eril dan Ridwan Kamil diklasifikasikan kedalam sentimen positif, dengan hasil data tersebut maka dapat digunakan untuk mengetahui *opini* dari *netizen* mengenai Emmeril Kahn dan Elektabilitas Ridwan Kamil.

PERPUSTAKAAN UNIVERSITAS JENDERAL ACHMAD YANI YOGYAKARTA