

BAB IV

HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Data awal (Data Primer) yang akan digunakan dalam analisis sentimen ini adalah data berupa teks yang dikumpulkan oleh peneliti melalui *IGCommentsExport* dan data teks tersebut berupa *comment* masyarakat terhadap pelayanan Mie Gacoan. Pengumpulan data dari Instagram Mie Gacoan dilakukan pada bulan November 2023-Januari 2024. Data yang dikumpulkan tersebut merupakan *comment* masyarakat pada saat Mie Gacoan menjadi topik yang masih hangat atau sering diperbincangkan. Dalam mengumpulkan data tersebut, digunakan *IGCommentsExport* dengan menggunakan kata kunci “mie gacoan”.

Setelah melakukan *crawling* data bulan November 2023 – Januari 2024, data-data tersebut dijadikan suatu dataset, dengan total data yang didapat adalah 1156 data. Data yang didapat merupakan *comment* yang terdiri dari atribut user id, *username*, comment id, comment text, profil URL, avatar URL dan date. Beberapa *comment* yang dikumpulkan ditampilkan dalam Gambar 4.1 Namun, setelah dilakukan pelabelan data, jumlah data yang dipakai dalam penelitian adalah 424 data. Hal tersebut disebabkan penelitian ini hanya menggunakan *comment* dengan nilai sentimen positif dan negatif. Dataset yang dipakai tersebut disimpan dalam dokumen berformat *Comma Separated Values* (CSV). Jumlah data pakai yang telah dilabeli dapat dilihat dalam Tabel 4.1.

User Id	Username	Comment Id	Comment Text	Profile URL	Avatar URL	Date
2,94E+09	rattnaadaw	1,81112E+16	@kulo_fhitria ayo too	https://www.instagram.com/ra	https://scontent-cgk1-1.12/28/2023, 10:46:12 PM	
8,3E+09	kulo_fhitria	1,81516E+16	@rattnaadaw hayok kapan	https://www.instagram.com/ki	https://scontent-cgk1-1.12/28/2023, 10:47:03 PM	
5,61E+10	dusk_r95	1,79819E+16	Info web untuk lokernya	https://www.instagram.com/di	https://scontent-cgk1-1.12/29/2023, 8:25:47 AM	
5,73E+10	_regaindraa	1,82795E+16	Finally mie gacoan sudah buka di tasik	https://www.instagram.com/_i	https://instagram.fsrg6-12/29/2023, 3:01:39 PM	
4,51E+10	dwkrmdani	1,80106E+16	Minnn gratis oh min 😊	https://www.instagram.com/di	https://scontent-cgk1-1.12/29/2023, 3:33:16 PM	
6,22E+10	ulfah_hasanah	1,7898E+16	Daerah mana kak	https://www.instagram.com/ul	https://instagram.fsrg6-12/29/2023, 5:20:20 PM	
5,69E+10	salummeliyan	1,83061E+16	hihi aku udh ke sono enak tau	https://www.instagram.com/sc	https://scontent-cgk1-1.12/29/2023, 6:28:34 PM	
3,56E+10	masmunari60	1,81086E+16	Ngantri panjang ditasik mah	https://www.instagram.com/m	https://scontent-cgk1-1.12/29/2023, 9:50:26 PM	
6,35E+10	inss4042	1,80083E+16	Buka cabang di Banjarbaru, Kalimantan selatan	https://www.instagram.com/in	https://instagram.fsrg6-12/29/2023, 9:54:52 PM	
5,3E+10	kapten_ekadia	1,79822E+16	Siapa yang pernah makan saya	https://www.instagram.com/ki	https://scontent-cgk1-1.12/30/2023, 7:05:08 AM	
5,62E+09	anisitirohanii	1,8089E+16	ME GACOOAN Bantar gebang - Cimuning	https://www.instagram.com/ai	https://scontent-cgk1-1.12/30/2023, 11:42:30 AM	
5,39E+10	dstiinrainiii	1,8275E+16	Knpaaa sii gabuka cabang d garutt	https://www.instagram.com/di	https://instagram.fsrg6-12/30/2023, 2:58:31 PM	
5,36E+10	miraclegracia2	1,83212E+16	B	https://www.instagram.com/m	https://scontent-cgk1-1.12/30/2023, 6:40:06 PM	
1,11E+10	eka.srvt	1,79151E+16	@ekaaoktariani	https://www.instagram.com/el	https://instagram.fcgk1(12/30/2023, 6:53:12 PM	
5,2E+10	vitry.indriaa	1,80178E+16	@fasidol99 mie gacoan yng d pondok gede juga	https://www.instagram.com/vi	https://scontent-cgk1-1.12/31/2023, 12:11:08 AM	
5,57E+10	heo_hyun273	1,80361E+16	@pitelii _ cepet pulang buk, kita harus serbuu..	https://www.instagram.com/hh	https://instagram.fcgk1(12/31/2023, 12:40:41 AM	
5,75E+10	rismaayuristia	1,82612E+16	Enk nya g main 😊	https://www.instagram.com/ri	https://scontent-cgk1-1.12/31/2023, 9:57:25 AM	
5,46E+10	calon_flighta	1,80517E+16	Ini yang di bali itu kk	https://www.instagram.com/cc	https://instagram.fcgk1(12/31/2023, 10:16:08 AM	
5,46E+10	calon_flighta	1,80189E+16	Aku tau kesitu satu keluarga pada bingung karn	https://www.instagram.com/cc	https://instagram.fcgk1(12/31/2023, 10:18:55 AM	
5,45E+10	flourencectx	1,79079E+16	@salmarqss cobainn bubb	https://www.instagram.com/fl	https://scontent-cgk1-1.12/31/2023, 10:55:36 AM	

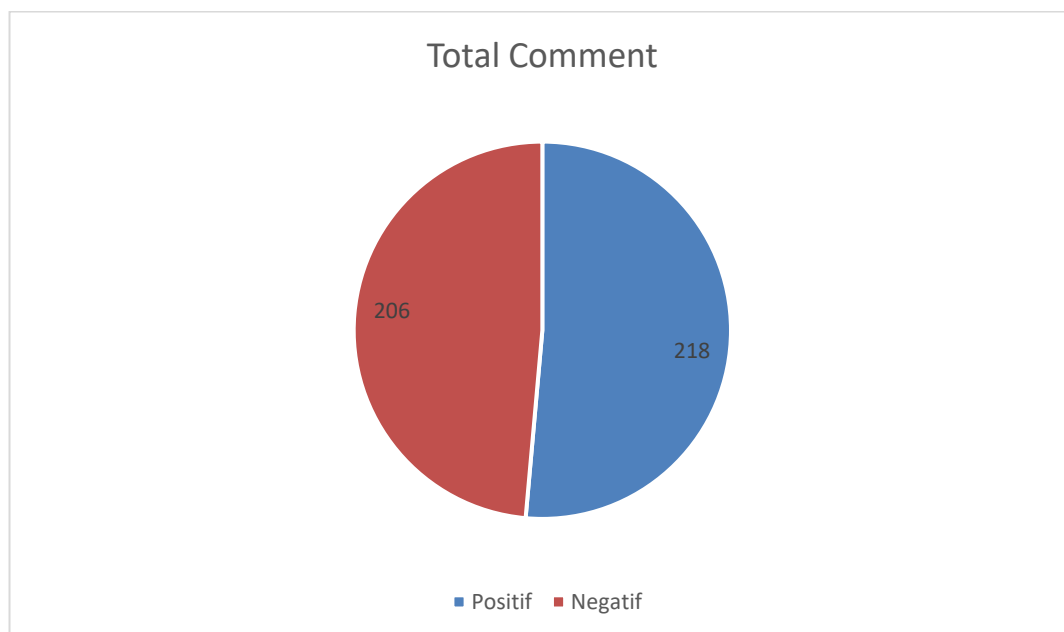
Gambar 4.1. Dataset *Comment* tentang Mie Gacoan

Gambar 4.1. menampilkan beberapa *comment* yang didapatkan. *comment* tersebut masih mentah atau belum dilakukan *preprocessing* data yang akan dilakukan nantinya. Setiap *comment* tersebut memiliki atribut *user*, *id*, *date* dan teks *comment*.

Tabel 4.1. Jumlah Data Instagram yang Dipakai

Kelas	Total Data
Positif	218
Negatif	206

Dari Tabel 4.1 diketahui bahwa dari total *comment* yang dipakai, 218 data termasuk dalam *comment* bersentimen positif dan 206 data termasuk dalam *comment* bersentimen negatif. Berdasarkan data tersebut, didapatkan bahwa 51.4% opini masyarakat di Instagram pada bulan November 2023 – Januari 2024 bersentimen positif terhadap pelayanan Mie Gacoan. Sedangkan, terdapat 48.6% opini masyarakatnya yang bersentimen negatif. Hal ini dapat dilihat pada gambar 4.2



Gambar 4.2. Grafik Total *Coment* Yang Dipakai

4.2. Pelabelan Sentimen

Pelabelan sentimen pada dataset dilakukan secara manual oleh pelabel. Pelabel memberi label berupa positif untuk sentimen positif, negatif untuk sentimen negatif dan netral untuk sentimen netral (selain positif dan negatif). Selanjutnya, data yang dipakai hanya yang berlabel positif dan negatif. Beberapa *comment* tentang pelayanan Mie Gacoan yang telah dilabeli ditampilkan dalam Tabel 4.2 dan Tabel 4.3.

Tabel 4.2. Sentimen Positif

Comment	Label
hihi aku udh ke sono enak tau	Positif
Malahan aku suka bangetttt sama mie gacoan	Positif
Favorit aku udang keju @megacoangatsu	Positif
mie hompimpa, kenapa jarang di pesen padahal enak banget??	Positif
Udang keju krna itu enak banget tp aku selalu pesan	Positif

Tabel 4.3. Sentimen Negatif

Comment	Label
Ngantri panjang ditasik mah	Negatif
@fasidol99 mie gacoan yng d pondok gede juga karyawannya kurang stroong...mie'y udh mau hbs d tambah nahan pedas minumnya bru dtng??	Negatif
pelayanan ny g bngt	Negatif
Mie ini bikin ngantri yg gw GK suka perut dh laper duluan	Negatif
Gacoan cabang tugu yogya super ancurrrrr!!!! Makanan dingin, pangsit dingin lembek kaya pangsit rebus jdnya	Negatif

Dalam Tabel 4.2 dan Tabel 4.3 menampilkan *comment* yang memiliki sentimen positif dan negatif, hasil dari sentimen yang paling dominan dari penilaian pelabel. Pada Tabel 4.2 terdapat *comment* yang mengandung kata yang mengarah ke sentimen positif seperti kata “buka”, “enak”, “suka”, dan lain-lain. Kemudian pada Tabel 4.3 terdapat *comment* yang mengandung kata yang mengarah ke sentimen negatif seperti kata “ngantri”, “lama”, “buruk”, dan lain-lain.

4.3. Preprocessing Data

Tahap selanjutnya adalah *preprocessing* data, yaitu rangkaian proses mengubah data sesuai dengan format yang dibutuhkan. Persiapan data ini menggunakan fitur *text preprocessing* yang ada pada *library* NLTK (*Natural Language Toolkit*). Tahapan data *preprocessing* yang dilakukan dijelaskan sebagai berikut.

1. *Cleaning*

Tahapan ini untuk menghilangkan atribut yang ada pada data seperti simbol, tanda baca, *hashtag*, URL, angka dan *noise* lainnya. Proses *cleaning* ini dilakukan dengan menggunakan kode program pada

gambar 4.3. Kemudian, contoh data yang telah melalui tahap *cleaning* sehingga simbol, tanda baca, *hashtag*, URL, angka dan *noise* lainnya sudah hilang ditampilkan dalam Tabel 4.4.

```
def clean_text(text):
    #menghapus new line
    text = text.replace('\n', " ")
    #menghapus https dan http
    text = re.sub(r'http\S+', '', str(text))
    #menghapus Username
    text = re.sub('@[\s]+', '', text)
    #menghapus single char
    text=re.sub(r"\b[a-zA-Z]\b", " ", text)
    # Remove tagger
    text = re.sub(r'#([\s]+)', '', text)
    #mengganti karakter html dengan tanda petik
    text=re.sub('<.*?>', ' ', text)
    # Remove angka termasuk angka yang berada dalam string
    # Remove non ASCII chars
    text = re.sub(r'^[\x00-\x7f]', r'', text)
    text = re.sub(r'(\u[0-9A-Fa-f]+)', r'', text)
    text = re.sub(r"^[A-Za-z0-9^!.\/'+-=]", " ", text)
    text = re.sub(r'\u\w\w\w\w', '', text)
    # Remove simbol, angka dan karakter aneh
    text = re.sub(r"[.,;+!-\_<^/?\"'(\)\d\*]", " ", text)
    #menghilangkan tanda baca
    text=text.translate(str.maketrans(' ', ' ', string.punctuation))
    #mempertimbangkan huruf dan angka
    text=re.sub('[^a-zA-Z]', ' ',text)
```

Gambar 4.3. Kode Program Tahapan *Cleaning*

Tabel 4.4. Contoh Data Hasil *Cleaning*

<i>Input</i>	Favorit aku udang keju @megacoangatsu
<i>Output</i>	Favorit aku udang keju

2. Case Folding

Tahap ini merupakan proses mengubah semua data huruf kapital yang ada pada dataset menjadi huruf kecil (nonkapital). Proses *case folding* ini dilakukan dengan menggunakan Kode Program pada gambar 4.4. Selanjutnya, contoh data yang telah diubah menjadi huruf kecil/*lowercase* ditampilkan pada Tabel 4.5.

```
#mengecilkan huruf
text=text.lower()
text=' '.join(text.split())
return text
```

Gambar 4.4. Kode Program Tahapan *Case Folding*

Tabel 4.5. Contoh Data Hasil *Case Folding*

<i>input</i>	GACOAN GADA NIATAN MAU BUKA CABANG DI SERANG APA????????
<i>output</i>	gacoan gada niatan mau buka cabang di serang apa

3. *Tokenizing*

Tahapan ini merupakan proses memecahkan kalimat menjadi satuan kata yang menyusunnya. Metode ini akan memotong kata berdasarkan spasi yang selanjutnya kata tersebut akan menjadi token. *Tokenizing* dilakukan dengan menerapkan Kode Program 4.5 dan akan menghasilkan token kata seperti yang ditampilkan dalam Tabel 4.6.

```
# memecah teks menjadi token berdasarkan pola kata, angka dan karakter non-spasi
regexp = RegexpTokenizer(r'\w+|$\d+|\S+')
# menerapkan tokenisasi pada kolom clean
data_gacoan['token'] = data_gacoan['clean'].apply(regexp.tokenize)
```

Gambar 4.5. Kode Program Tahapan *Tokenizing*

Tabel 4.6. Contoh Data Hasil *Tokenizing*

<i>input</i>	favorit aku udang keju
<i>output</i>	['favorit', 'aku', 'udang', 'keju']

4. Normalisasi

Tahap normalisasi adalah tahap penormalan bentuk kata. Pada tahap normalisasi yaitu mengubah bentuk kata yang kurang normal seperti singkatan, bahasa asing dan kata yang kurang baku dalam *dataset* menjadi kata baku. Normalisasi dilakukan dengan cara pencatatan menggunakan excel disimpan dengan format (.xlsx) mengenai kata-kata yang kurang normal, kemudian diubah menjadi kata yang lebih normal. Contoh penerapan normalisasi pada Tabel 4.7.

```
# Membaca file Excel dengan daftar kata-kata untuk normalisasi
normalizad_word = pd.read_excel("normalisasi.xlsx", engine='openpyxl')
# Membuat kamus kosong untuk menyimpan pasangan kata yang akan dinormalisasi dan kata hasil normalisasi
normalizad_word_dict={}
# Iterasi melalui setiap baris dalam dataframe normalizad_word
for index, row in normalizad_word.iterrows():
    # Memeriksa apakah kata sudah ada dalam kamus normalizad_word_dict
    if row[0] not in normalizad_word_dict:
        # Menambahkan pasangan kata ke kamus normalizad_word_dict
        normalizad_word_dict[row[0]] = row[1]
# Mendefinisikan fungsi normalized_term untuk melakukan normalisasi pada dokumen
def normalized_term(document):
    # Mengembalikan daftar kata hasil normalisasi
    return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in document]
# Mengaplikasikan fungsi normalized_term pada kolom 'token' dalam dataframe comment gacoan
data_gacoan['normalisasi'] = data_gacoan['token'].apply(normalized_term)
```

Gambar 4.6. Kode Program Tahapan *Tokenizing*

Tabel 4.7. Contoh Normalisasi

Input	gak banget apa lagi pelayanan banget
Ouput	['tidak ', 'banget ', 'layan ', 'banget ']

5. Stopword Removal

Tahap ini untuk menghapus kata-kata yang terkandung dalam daftar kata *stopword*, yang dalam hal ini peneliti menggunakan *stopword* yang terdapat dalam *library* NLTK (*Natural Language Toolkit*) dengan dataset bahasa Indonesia. Kata yang diperoleh dari tahap *tokenizing* diperiksa

dalam suatu daftar *stopword*, apabila ada kata masuk di dalam *stopword* maka kata tersebut tidak akan diproses lebih lanjut. *Stopword removal* juga dapat digunakan untuk menghilangkan kata yang tidak relevan atau yang tidak bermakna dalam dataset. Daftar beberapa kata *stopword* yang digunakan dalam penelitian ini dapat dilihat dalam Tabel 4.8.

Tabel 4.8. Contoh Daftar Kata *Stopword*

Kata <i>Stopword</i>
Bikin
Nya
Sih
Tu
Hehe
Anjir
Lho
Yang
Ya
Wah

Pada penelitian ini menggunakan 2162 kata sebagai daftar *stopword* yang diperoleh dari *library* NLTK (*Natural Language Toolkit*) dan penambahan secara manual ke dalam sistem. Penghapusan *stopword* ini dilakukan dengan menerapkan Kode Program 4.7.


```

# Mengimpor stopwords dari library nltk untuk bahasa Indonesia
list_stopwords = stopwords.words('indonesian')
# Menambahkan kata-kata tambahan ke dalam list_stopwords
list_stopwords.extend(["amp", "biar", "bikin", "nya", "nih", "sih", 'dll', 'tuh', 'eh', 'kang', 'hehe', 'anjir', 'btw', 'oh', 'h')
# Membaca file stopwords.txt yang berisi kata-kata tambahan untuk dijadikan stopwords
txt_stopword = pd.read_csv("stopwords.txt", names=["stopwords"], header=None)
# Menambahkan kata-kata dari file stopwords.txt ke dalam list_stopwords
list_stopwords.extend(txt_stopword["stopwords"][0].split(' '))
# Mengubah list_stopwords menjadi set untuk efisiensi pencarian
list_stopwords = set(list_stopwords)
# Mendefinisikan fungsi stopwords_removal untuk menghapus stopwords dari kata-kata
def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]

```

Gambar 4.7. Kode Program Tahapan *Stopword Removal*

Tabel 4.9. Contoh Hasil *Stopword Removal*

Input	sayang nya yg punya alergi udang hanya bisa menatapi sambil ngiler bikin dimsum ayam nya donk min
ouput	['sayang', 'alergi', 'udang', 'bisa', 'menatapi', 'ngiler', 'dimsum', 'ayam', 'donk', 'min']

6. *Stemming*

Stemming merupakan tahapan yang bertujuan untuk mengembalikan suatu kata ke bentuk akar katanya atau menghilangkan imbuhan (awalan dan akhiran) yang terdapat pada kata tersebut. Proses *stemming* ini dilakukan dengan menggunakan *library* NDETCStemmer yang ditunjukkan dalam Kode Program 4.8, yang kemudian akan menghasilkan hasil *stemming* seperti yang ditampilkan dalam Tabel 4.10.

```

# Membuat objek stemmer menggunakan NDETCStemmer
stemmer = NDETCStemmer()
# Mendefinisikan fungsi wrapped untuk melakukan stemming pada suatu term
def stemmed_wrapper(term):
    return stemmer.stem(term)
# Membuat kamus kosong untuk menyimpan pasangan term dan term hasil stemming
term_dict = {}
# Mengambil daftar data dari kolom 'stopwords' dalam dataframe
data = data_gacoan['stopwords']
# Iterasi melalui setiap dokumen dalam daftar
for doc in data:
    # Iterasi melalui setiap term dalam dokumen
    for term in doc:
        # Menambahkan term ke kamus term_dict jika belum ada
        if term not in term_dict:
            term_dict[term] = ' '
# Menampilkan jumlah term yang ada dalam kamus term_dict
print(len(term_dict))
print("-----")

```

Gambar 4.8. Kode Program Tahapan *Stemming*

Tabel 4.10. Contoh Data Hasil *Stemming*

<i>Comment</i>
Suka
Minum favorit green thai tea
Kangen gaco buka batam
Enak banget
nyari sakit pakai tunggu superwin akun menang
Mie suit udang keju best combo

7. *Wordcloud*

Dataset yang sudah melewati beberapa proses *preprocessing* data, dicari frekuensi kemunculan kata yang paling sering muncul pada tiap sentimen yang divisualisasikan dalam bentuk *wordcloud*. *Wordcloud* ini merupakan suatu gambar yang berisi kumpulan kata yang besarnya suatu kata merepresentasikan kemunculan atau tingkat kepentingan. Pembuatan *wordcloud* ini menggunakan *library* dari *wordcloud*. Untuk visualisasi

wordcloud dari data bersentimen negatif pada dataset ditampilkan dalam Gambar 4.10.

```
all_text_s0 = ' '.join(word for word in train_s0["stemmer"])
wordcloud = WordCloud(colormap='Reds', width=1000, height=1000, mode='RGBA', background_color='white').generate(all_text_s0)
plt.figure(figsize=(20,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title("Ulasan Negatif")
plt.margins(x=0, y=0)
plt.show()
```

Gambar 4.9. Kode Program Menampilkan Wordcloud Sentimen Negatif



Gambar 4.10. *Wordcloud* Sentimen Negatif dalam Dataset.

Berdasarkan *wordcloud* dalam Gambar 4.10 menunjukkan bahwa beberapa kata seperti “makan”, “mau”, “nomor”, “antri”, “lama”, “kurang”, “kotor”, “tidak”, “teror”, dan “bayar” merupakan kata yang sering muncul dalam dataset yang berlabel sentimen negatif. Bererapa dari kata tersebut mengandung konotasi negatif. Hal tersebut dapat diartikan banyak opini masyarakat terhadap pelayanan mie gacoan. Kemudian untuk *wordcloud* dari data sentimen positif pada dataset ditampilkan dalam Gambar 4.12.

```
all_text_s1 = ' '.join(word for word in train_s1["stemmer"])
wordcloud = WordCloud(colormap='Blues', width=1000, height=1000, mode='RGBA', background_color='white').generate(all_text_s1)
plt.figure(figsize=(20,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title("Ulasan Positif")
plt.margins(x=0, y=0)
plt.show()
```

Gambar 4.11. Kode Program Menampilkan Wordcloud Sentimen Positif

kosakata yang muncul dari dataset dapat dilakukan dengan menggunakan kode program pada gambar 4.13.

```
#pembobotan Tfidf
tf = TfidfVectorizer()
text_tf = tf.fit_transform(data_gacoan['stemmer'])
print("TF-IDF", type(text_tf), text_tf.shape)
```

Gambar 4.13. Kode Program Kosakata TF-IDF

Hasil dari dataset yang telah dilatih menggunakan TF-IDF didapatkan kosakata yang berjumlah 1513 kata. Selanjutnya, pada Tabel 4.11. ditampilkan perbandingan perhitungan manual dan otomatis.

Tabel 4.11. Perbandingan TF-IDF

Term	Manual				Otomatis	
	TF	DF	IDF	e	TF-IDF	TF-IDF
Dimsum	1	11	4,567	20,857	0,536410618	0,536410618
Enak	1	32	3,556	12,645	0,417665022	0,417665022
rambutan	1	5	5,26	27,668	0,617805967	0,617805967
Udang	1	39	3,363	11,31	0,394996476	0,394996476

4.5. Pembagian Data

Pembagian data dalam penelitian ini dilakukan secara acak agar pembagian dataset menjadi data latih dan data uji yang dapat dilakukan dapat menghasilkan hasil yang optimal dengan pembagian 80% data latih dan 20% data uji. Proses pembagian data ini menggunakan *library* Sklearn untuk membagi data. Untuk menjalankan proses ini dilakukan dengan kode program pada gambar 4.14.

```

# Membagi data menjadi data Latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(text_tf, data_gacoan['label'], test_size=1-split_size, stratify=data_g
<

# Menghitung jumlah kelas pada data Latih dan data uji
train_counts = y_train.value_counts()
test_counts = y_test.value_counts()

# Menampilkan informasi tentang ukuran data Latih serta data uji
print("Train size: {} | Test size: {}".format(X_train.shape[0], X_test.shape[0]))
print("Train pos: {} | Train neg: {}".format(train_counts["positif"], train_counts["negatif"]))
print("Test pos: {} | Test neg: {}".format(test_counts["positif"], test_counts["negatif"]))

```

Gambar 4.14. Kode Program Proses Pembagian Data

4.6. Klasifikasi Naïve Bayes

Pemodelan klasifikasi yaitu menggunakan algoritma *Naïve Bayes* dengan *library Sklearn* untuk menjalankan algoritma pada *python* dan melakukan `import MultinomialNB`. Algoritma *Multinomial Naïve Bayes* adalah model Naïve Bayes untuk mengatasi data berbasis teks, seperti dokumen teks atau ulasan. Fungsi *Multinomial Naïve Bayes* untuk mengklasifikasikan teks atau kata-kata ke dalam kategori atau sentimen yang berbeda, seperti positif dan negatif. Selanjutnya dilakukan evaluasi performa klasifikasi *Naïve Bayes* untuk penghitungan nilai akurasi, presisi, *recall*, dan *F1-score* dengan memanfaatkan *Confusion Matrix*. Berikut adalah Kode program 4.15 digunakan untuk klasifikasi menggunakan *Naïve Bayes*.

```

# Melatih model dengan data Latih
model.fit(X_train, y_train)

MultinomialNB()

# Memprediksi Label pada data uji
y_pred = model.predict(X_test)

```

Gambar 4.15. Kode Program Klasifikasi Naïve Bayes

4.7. Evaluasi Hasil Klasifikasi

Evaluasi performa model menggunakan *Multinomial Naïve Bayes* dengan memperhitungkan beberapa metrik yang memberikan wawasan tentang seberapa baik model dapat melakukan klasifikasi pada data. *Confusion*

matrix menjadi dasar evaluasi, menyajikan informasi mengenai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Akurasi mengukur persentase keseluruhan prediksi yang benar, sedangkan presisi fokus pada proporsi positif yang benar dari semua hasil positif yang diprediksi. *Recall*, juga dikenal sebagai *Sensitivity* atau *True Positive Rate*, menilai sejauh mana model dapat mendeteksi semua nilai sebenarnya yang positif. *F1 Score*, sebagai metrik gabungan presisi dan *recall*, memberikan gambaran holistik tentang kinerja model dengan mempertimbangkan *trade-off* antara *precision* dan *recall*. Kode program 4.16 untuk mengevaluasi model *Naïve Bayes*.

```
# Menghitung metrik evaluasi
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = conf_matrix.ravel()
clasif_report = classification_report(y_test, y_pred)
presisi = precision_score(y_test, y_pred, average="binary", pos_label="positif")
recall = recall_score(y_test, y_pred, average="binary", pos_label="positif")
f1 = f1_score(y_test, y_pred, average="binary", pos_label="positif")

# Menampilkan hasil evaluasi
print("Accuracy: {:.2f}%".format(accuracy*100))
print("Precision: {:.2f}%".format(presisi*100))
print("Recall: {:.2f}%".format(recall*100))
print("F1-Score: {:.2f}%".format(f1*100))
print("Confusion Matrix:\n", conf_matrix)
```

Gambar 4.16. Kode Program Untuk Evaluasi Performa

Dalam pengujian menggunakan model *Naive Bayes*, data dipecah menjadi dua subset, yaitu 80% data latih untuk melatih model dan 20% data uji untuk mengukur kinerja. Hasil pengujian ditampilkan dalam gambar 4.17 yang mencakup metrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1 Score*, yang memberikan gambaran menyeluruh tentang kemampuan model dalam melakukan klasifikasi pada dataset.

```

Accuracy: 90.59%
Precision: 87.50%
Recall: 95.45%
F1-Score: 91.30%
Confusion Matrix:
[[35  6]
 [ 2 42]]
TN: 35, FP: 6, FN: 2, TP: 42
Classification Report:
              precision    recall  f1-score   support

negatif      0.95      0.85      0.90        41
positif      0.88      0.95      0.91        44

accuracy          0.91
macro avg         0.91      0.90      0.91        85
weighted avg     0.91      0.91      0.91        85

```

Gambar 4 17. Hasil Pengujian Evaluasi Performa

Tabel 4.12. *Confusion Matrix*

Fakta	Prediksi	
	Negatif	Positif
Negatif	35	6
Positif	2	42

Pada Tabel 4.12 menampilkan hasil *Confusion Matrix* pada setiap skema dengan metode *Naïve Bayes*. Sebanyak 35 data diprediksi secara benar sebagai data negatif (*True Negative*). Sebanyak 42 data diprediksi secara benar sebagai data positif (*True Positive*). Sebanyak 6 data diprediksi secara salah sebagai data positif, seharusnya data tersebut adalah data negatif (*False Positive*). Sebanyak 2 data diprediksi secara salah sebagai data negatif, seharusnya data tersebut adalah data positif (*False Negative*).

a. Akurasi

$$\text{Akurasi (accuracy)} = \frac{35 + 42}{42 + 35 + 2 + 6} \times 100\% = 90.59\%$$

Akurasi mengacu pada sejauh mana model dapat mengklasifikasikan dengan benar sentimen atau perasaan yang terkandung dalam teks atau data. Sentimen analisis bertujuan untuk menentukan apakah suatu teks atau dokumen bersifat positif atau negatif. Akurasi dalam sentimen analisis diukur dengan membandingkan jumlah prediksi yang benar (positif atau negatif) dengan total jumlah prediksi yang dilakukan oleh model.

Akurasi pada penelitian ini sebesar 90.59% berarti bahwa model yang digunakan berhasil mengklasifikasikan sentimen dengan benar sebanyak 90.59% dari total prediksinya.

b. Presisi

$$\text{Presisi (precision)} = \frac{42}{42 + 6} \times 100\% = 87.50\%$$

Presisi (*precision*) adalah metrik evaluasi yang mengukur sejauh mana prediksi positif yang dibuat oleh model benar-benar relevan dan relevansi tersebut memiliki tingkat ketepatan yang tinggi. Presisi memberikan gambaran tentang berapa persen dari semua prediksi positif yang benar-benar positif. Presisi memberikan informasi tentang seberapa andal model dalam mengidentifikasi sentimen positif. Tingkat presisi yang tinggi menunjukkan bahwa model cenderung memberikan prediksi positif yang benar-benar positif. Presisi sebesar 87.50% pada analisis sentimen klasifikasi dengan Naive Bayes berarti bahwa dari semua prediksi sentimen positif yang dibuat oleh model, sebanyak 87.50% di antaranya adalah benar-benar relevan dan tepat. Presisi dihitung dengan membagi jumlah True Positive (TP), yaitu kasus di mana model dengan benar memprediksi sentimen positif, dengan total dari True

Positive dan False Positive (FP), yaitu kasus di mana model memprediksi sentimen positif tetapi sebenarnya sentimennya negatif.

c. *Recall*

$$Recall = \frac{42}{42 + 2} \times 100\% = 95.45\%$$

Recall adalah metrik evaluasi yang mengukur sejauh mana model dapat mengenali dan menangkap semua kasus positif yang sebenarnya. Recall sebesar 95.45% dalam analisis sentimen berarti bahwa model berhasil mengidentifikasi dengan sangat baik sebagian besar kasus sentimen positif yang sebenarnya ada dalam dataset. Dengan persentase recall sebesar ini, model dapat "mengingat" dan mengklasifikasikan sebanyak 95.45% dari keseluruhan kasus sentimen positif yang sebenarnya ada. Recall yang tinggi seperti ini menunjukkan bahwa model memiliki kemampuan yang baik untuk mengenali dan mendeteksi sebagian besar sentimen positif, yang dapat menjadi indikasi kinerja yang bagus dalam mengatasi kasus positif.

d. *F1-score*

$$F1\ Score = 2 \times \frac{(95.45 \times 87.50)}{(95.45 + 87.50)} = 91.30\%$$

F1 Score memiliki rentang nilai antara 0 hingga 1, di mana nilai 1 menunjukkan kinerja yang sempurna dan nilai 0 menunjukkan kinerja yang sangat buruk. Semakin tinggi nilai *F1 Score*, semakin

baik model dalam menjaga keseimbangan antara presisi dan *recall*. *F1-Score* sebesar 91.30% menunjukkan bahwa model analisis sentimen memiliki kinerja yang sangat baik dalam menggabungkan presisi (*precision*) dan *recall*. *F1-Score* adalah metrik yang berguna karena mencoba mencari keseimbangan antara presisi dan *recall*, dan hasil sebesar 91.30% menandakan bahwa model dapat memberikan prediksi positif yang relevan presisi dan *recall*.