

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Pengumpulan Data

Data dikumpulkan melalui scraping web yang dilakukan pada laman video youtube dengan kata kunci “Kurikulum Merdeka” pada bulan Oktober 2023. Peneliti mengumpulkan data menggunakan bahasa pemrograman Python dan mengeksekusinya menggunakan Google Colab. *Library* yang digunakan untuk melakukan proses *crawling* adalah *googleapiclient*, dimana nantinya diminta untuk memasukkan *API key* dan *id* video *Youtube* yang akan dilakukan *crawling*, seperti terlihat pada gambar 4.1 dibawah ini.

```
import pandas as pd
from googleapiclient.discovery import build
```

**Gambar 4.1 Import Library Python**

Kemudian melakukan scraper komentar dengan memasukkan kunci API yang diperoleh sebelumnya saat membuat kredensial di Google Developer. Skrip tersebut berfungsi untuk mengambil data, komentar video YouTube atau informasi komentar ditampilkan

```
def video_comments(video_id):
    # empty list for storing reply
    replies = []

    # creating youtube resource object
    youtube = build('youtube', 'v3',
        developerKey="AIzaSyDIZQY9AwH3f7rQ-QlriDqywM8dnH4qr2A")

    # retrieve youtube video results
    video_response = youtube.commentThreads().list(part='snippet,replies',
        videoId=video_id).execute()

    # iterate video response
    while video_response:

        # extracting required info
        # from each result object
        for item in video_response['items']:

            # Extracting comments ()
            published = item['snippet']['topLevelComment']['snippet']['publishedAt']
            user =
            item['snippet']['topLevelComment']['snippet']['authorDisplayName']

            # Extracting comments
            comment = item['snippet']['topLevelComment']['snippet']['textDisplay']
            likeCount = item['snippet']['topLevelComment']['snippet']['likeCount']

            replies.append([published, user, comment, likeCount])

            # counting number of reply of comment
            replycount = item['snippet']['totalReplyCount']

            # if reply is there
            if replycount>0:
                # iterate through all reply
                for reply in item['replies']['comments']:
```

```

# Extract reply
published = reply['snippet']['publishedAt']
user = reply['snippet']['authorDisplayName']
repl = reply['snippet']['textDisplay']
likeCount = reply['snippet']['likeCount']

# Store reply is list
#replies.append(reply)
replies.append([published, user, repl, likeCount])

# print comment with list of reply
#print(comment, replies, end = '\n\n')

# empty reply list
#replies = []

# Again repeat
if 'nextPageToken' in video_response:
    video_response = youtube.commentThreads().list(
        part = 'snippet,replies',
        pageToken = video_response['nextPageToken'],
        videoId = video_id
    ).execute()
else:
    break
#endwhile
return replies

#melakukan proses crawling data
# isikan dengan api key Anda
api_key = "AIzaSyDIZQY9AwH3f7rQ-QlriDqywM8dnH4qr2A"

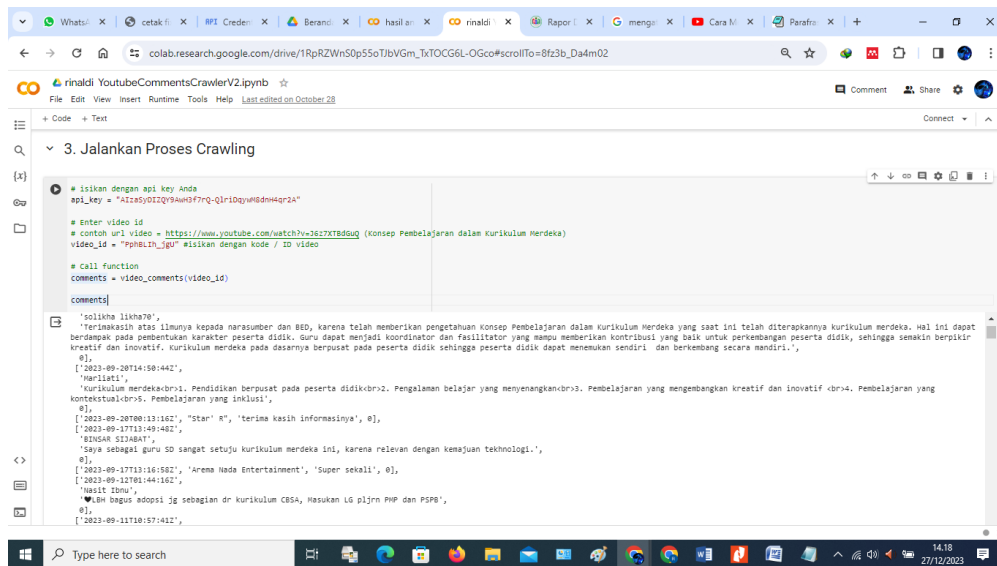
# Enter video id
# contoh url video = https://www.youtube.com/watch?v=J6z7XTBdGuQ
(Konsep Pembelajaran dalam Kurikulum Merdeka)
video_id = "PphBLIh_jgU" #isikan dengan kode / ID video

# Call function
comments = video_comments(video_id)

comments

```

**Gambar 4.2** Kode Pemrograman Proses *Scraping*



```

3. Jalankan Proses Crawling

# isikan dengan api key anda
api_key = "AIzaSyG1Z994wH9F7rQ-Q1r1DqyW@dH4q2A"

# enter video id
# contoh url video = https://www.youtube.com/watch?v=J6z7XT8duQ (Konsep Pembelajaran dalam kurikulum Merdeka)
video_id = "PphBLTh_gu" #isikan dengan kode / ID video

# call function
comments = video_comments(video_id)

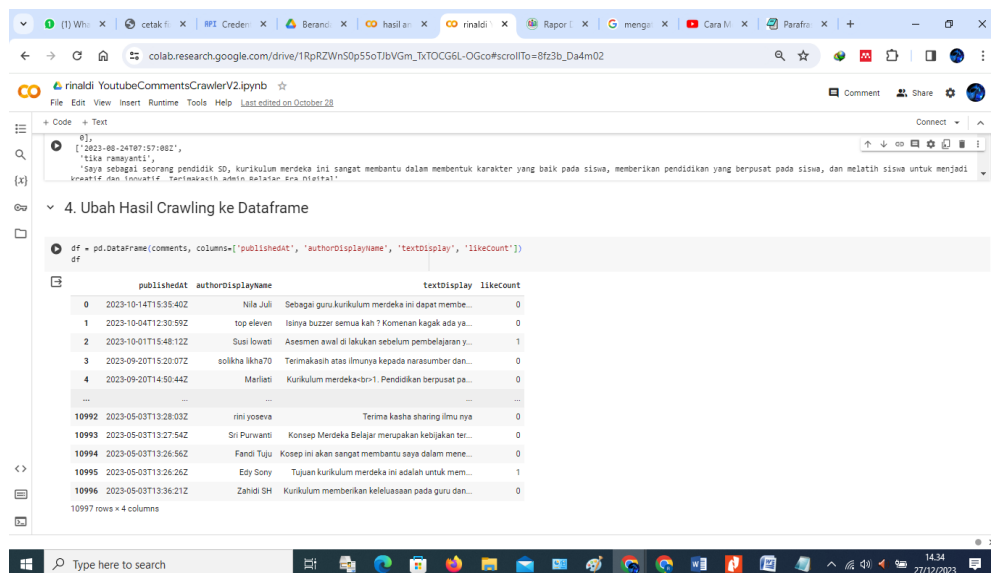
comment[]

['solihha likha70',
 'Terimakasih atas ilmunya kepada narasumber dan BED, karena telah memberikan pengetahuan konsep Pembelajaran dalam kurikulum Merdeka yang saat ini telah diterapkannya kurikulum merdeka. Hal ini dapat berdampak pada pembentukan karakter peserta didik. Guru dapat menjadi koordinator dan fasilitator yang mampu memberikan kontribusi yang baik untuk perkembangan peserta didik, sehingga semakin berpikir kreatif dan inovatif. Kurikulum merdeka pada dasarnya berpusat pada peserta didik sehingga peserta didik dapat menemukan sendiri dan berkembang secara mandiri.',
 0],
 ['2023-09-20T14:50:44Z',
 'Marliati',
 'Kurikulum merdeka-ber1. Pendidikan berpusat pada peserta didik-ber2. Pengalaman belajar yang menyenangkan-ber3. Pembelajaran yang mengembangkan kreatif dan inovatif -ber4. Pembelajaran yang kontekstual-ber5. Pembelajaran yang inklusif',
 0],
 ['2023-09-20T00:13:16Z', 'Star R', 'terima kasih informasinya', 0],
 ['2023-09-17T13:49:48Z',
 'BINSAR SIJABAT',
 'saya sebagai guru SD sangat setuju kurikulum merdeka ini, karena relevan dengan kemajuan teknologi.',
 0],
 ['2023-09-17T13:16:58Z', 'Arena Nada Entertainment', 'Super sekali', 0],
 ['2023-09-12T01:44:16Z',
 'HARIS IDOM',
 'Mau bagus adopsi jg sebagian dr kurikulum CBSA, Masukan LG pljrn PWP dan PSPB',
 0],
 ['2023-09-11T10:57:43Z',
]]

```

**Gambar 4.3 Hasil Scraping Data Komentar**

Selanjutnya mengubah hasil indeks menjadi *dataframe*. Hal ini dilakukan untuk menyusun kumpulan data komentar yang diambil sebelumnya. kedalam tabel dan diurutkan berdasarkan komentar terbaru.



```

4. Ubah Hasil Crawling ke Dataframe

df = pd.DataFrame(comments, columns=['publishedat', 'authordisplayname', 'textdisplay', 'likecount'])
df


```

	publishedat	authordisplayname	textdisplay	likecount
0	2023-10-14T15:35:40Z	Nila Juli	Sebagai guru kurikulum merdeka ini dapat membe...	0
1	2023-10-04T12:30:59Z	top eleven	Isinya buzzer semua kan ? Komenan kayak ada ya...	0
2	2023-10-01T15:48:12Z	Susi Iovati	Asesmen awal di lakukan sebelum pembelajaran y...	1
3	2023-09-20T15:20:07Z	solikha likha70	Terimakasih atas ilmunya kepada narasumber dan...	0
4	2023-09-20T14:50:44Z	Marliati	Kurikulum merdeka-ber1 Pendidikan berpusat pa...	0
...	...	...	...	...
10992	2023-05-03T13:28:03Z	rini yoseva	Terima kasha sharing ilmu nya	0
10993	2023-05-03T13:27:54Z	Siti Purwanti	Konsep Merdeka Belajar merupakan kebijakan ter...	0
10994	2023-05-03T13:26:56Z	Fandi Tuju	Kosep ini akan sangat membantu saya dalam mene...	0
10995	2023-05-03T13:26:26Z	Edy Sony	Tujuan kurikulum merdeka ini adalah untuk mem...	1
10996	2023-05-03T13:36:21Z	Zahidi SH	Kurikulum memberikan keleluasaan pada guru dan...	0

10997 rows x 4 columns

**Gambar 4.4 Hasil Scraping Data Komentar**

Terakhir, menyimpan data hasil crawling ke dalam file CSV. kemudian file tersebut diunggah untuk proses analisis sentimen selanjutnya.

```
df.to_csv('kurikulum.csv', index=False)
```

**Gambar 4.5 Skrip Pemrograman untuk *Convert to Csv***

## 4.2 Preprocessing

Preprocessing terdiri dari beberapa proses karena data komentar memiliki fitur yang tidak terstruktur dan banyak mengandung noise. Proses ini bertujuan untuk mengubah data mentah atau kotor lalu membersihkannya agar dapat diklasifikasikan. Pada langkah ini peneliti melakukan langkah-langkah prapemrosesan teks yang meliputi; *case folding*, *cleaning*, *tokenize*, *normalize*, *stemming*, dan *stopword removal*.

### 4.2.1 Case Folding

*Case Folding* bertujuan untuk merubah semua karakter kata dalam data set menjadi huruf kecil. Pada tahap ini dilakukan *case folding* menggunakan python. Dari dataset yang sudah didapatkan sebelumnya.



```
▼ Case Folding

# merubah jenis huruf menjadi huruf kecil
def lowercase(review_text):
    low = review_text.lower()
    return low

data_sentimen['clean_review'] = data_sentimen['komentar'].apply(lambda low:lowercase(str(low)))

data_sentimen.head()
```

**Gambar 4.6 Kode Case Folding Python**

Setelah dilakukan proses tersebut maka didapatkan hasil data set yang sudah tidak mempunyai lagi huruf kapital.

**Tabel 4.1 Contoh Hasil Case Folding**

<b>Sebelum Case Folding</b>	<b>Sesudah Case Folding</b>
Sebagai guru.kurikulum merdeka ini dapat memberikan motivasi dan karakter siswa dalam pembelajaran	sebagai guru kurikulum merdeka ini dapat memberikan motivasi dan karakter siswa dalam pembelajaran

#### Case Folding

```
[ ] # merubah jenis huruf menjadi huruf kecil
def lowercase(review_text):
    low = review_text.lower()
    return low

data_sentimen['clean_review'] = data_sentimen['koment'].apply(lambda low:lowercase(str(low)))
```

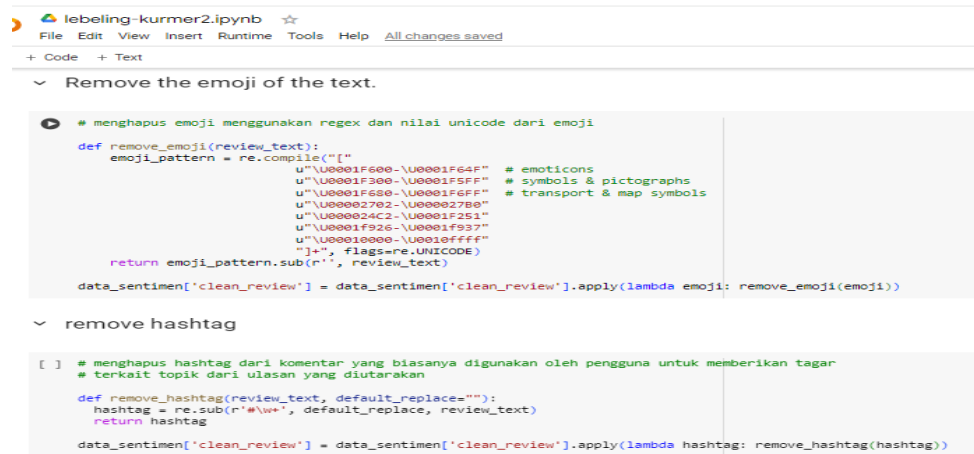
▶ data\_sentimen.head()

	koment	clean_review
0	Sebagai guru.kurikulum merdeka ini dapat membe...	sebagai guru.kurikulum merdeka ini dapat membe...
1	Isinya buzzer semua kah ? Komenan kagak ada ya...	isinya buzzer semua kah ? komenan kagak ada ya...
2	Asesmen awal di lakukan sebelum pembelajaran y...	asesmen awal di lakukan sebelum pembelajaran y...
3	Terimakasih atas ilmunya kepada narasumber dan...	terimakasih atas ilmunya kepada narasumber dan...
4	Kurikulum merdeka 1. Pendidikan berpusat pa...	kurikulum merdeka 1. pendidikan berpusat pa...

**Gambar 4.7 Hasil Case Folding Dataset**

## 4.2.2 Cleaning

Tahap *cleaning* didalam penelitian ini menggunakan python. Untuk menghapus atribut data yang tidak memiliki makna dan tidak dibutuhkan seperti url, tanda baca, *mention*, *whitespaces*, dan lain sebagainya.



```

lebeling-kurmer2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text

Remove the emoji of the text.

# menghapus emoji menggunakan regex dan nilai unicode dari emoji
def remove_emoji(review_text):
    emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F" # emoticons
        u"\U0001F300-\U0001F5FF" # symbols & pictographs
        u"\U0001F5B0-\U0001F6FF" # transport & map symbols
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001F926-\U0001F937"
        u"\U00010000-\U0001FFFF"
    ]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', review_text)

data_sentimen['clean_review'] = data_sentimen['clean_review'].apply(lambda emoji: remove_emoji(emoji))

remove hashtag

# menghapus hashtag dari komentar yang biasanya digunakan oleh pengguna untuk memberikan tagar
# terkait topik dari ulasan yang diutarakan
def remove_hashtag(review_text, default_replace=""):
    hashtag = re.sub(r'#\w+', default_replace, review_text)
    return hashtag

data_sentimen['clean_review'] = data_sentimen['clean_review'].apply(lambda hashtag: remove_hashtag(hashtag))

```

**Gambar 4.8** Proses *Cleaning* dengan Python

**Tabel 4.2** Hasil *Cleaning*

Sebelum <i>Cleaning</i>	Sesudah <i>Cleaning</i>
Saya sebagai manusia hanya bisa berdo&#39	saya sebagai manusia hanya bisa berdo

## 4.2.3 Tokenize

Tahap selanjutnya adalah *tokenize* atau tokenisasi dimana suatu kalimat dari komentar YouTube atau Dataset akan dipecah menjadi potongan kata atau token. Hal ini dilakukan untuk mengetahui kemunculan dari kata tersebut.

Tabel 4.3 Contoh Hasil Tokenisasi

Sebelum Tokenisasi	Sesudah Tokenisasi
Saya sebagai guru SD sangat setuju kurikulum merdeka ini, karena relevan dengan kemajuan tekhnologi.	['saya', 'sebagai', 'guru', 'sudah', 'sangat', 'setuju', 'kurikulum', 'merdeka', 'karena', 'relevan', 'dengan', 'kemajuan',

```
# tokenize
import nltk
nltk.download('punkt')

from nltk.tokenize import word_tokenize

def word_tokenize_wrapper(cleaning):
    return word_tokenize(cleaning)

df['tokenize'] = df['cleaning'].apply(word_tokenize_wrapper)
df
```

Gambar 4.9 Kode Python Tokenisasi

#### 4.2.4 Normalisasi

Tahap berikutnya adalah normalisasi atau *normalize*. Didalam penelitian ini, tahap normalisasi dilakukan untuk menstandarisasi kata yang memiliki makna yang sama dengan melakukan perubahan penulisan kata yang disingkat dan atau tidak baku. Penelitian ini menggunakan sebuah kamus normalisasi yang didapatkan dari kamus NLP (*Neuro Linguistic Programming*) bahasa Indonesia menggunakan *library* dari sastrawi.



Table 4.4 Contoh Hasil Normalisasi

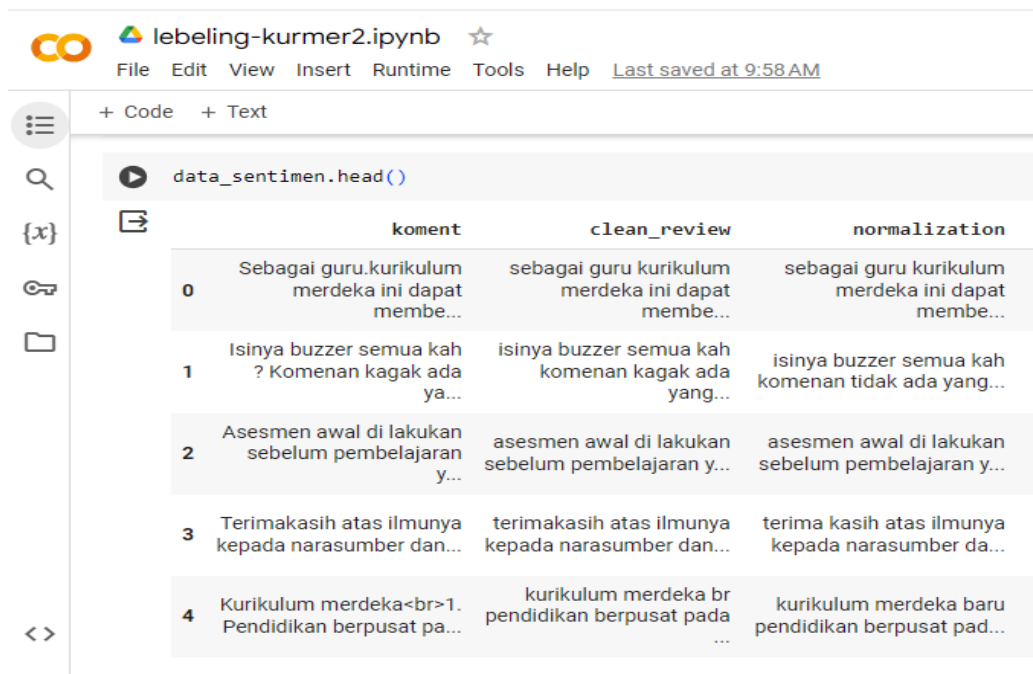
Sebelum Normalisasi	Sesudah Normalisasi
['kasi', 'tips', 'terutama', 'untuk',	['kasih', 'tips', 'terutama', 'untuk',

```

▶ clean_text = []
for review in data_sentimen['clean_review']:
    wordlist = TextBlob(review).words
    for k,v in enumerate(wordlist):
        if v in dict_slangs.keys():
            wordlist[k] = dict_slangs[v]
    clean_text.append(' '.join(wordlist))
clean_text
data_sentimen['normalization'] = clean_text
data_sentimen.head()

```

Gambar 4.10 Kode Normalisasi Python



	koment	clean_review	normalization
0	Sebagai guru.kurikulum merdeka ini dapat membe...	sebagai guru kurikulum merdeka ini dapat membe...	sebagai guru kurikulum merdeka ini dapat membe...
1	Isinya buzzer semua kah ? Komenan kagak ada ya...	isinya buzzer semua kah komenan kagak ada yang...	isinya buzzer semua kah komenan tidak ada yang...
2	Asesmen awal di lakukan sebelum pembelajaran y...	asesmen awal di lakukan sebelum pembelajaran y...	asesmen awal di lakukan sebelum pembelajaran y...
3	Terimakasih atas ilmunya kepada narasumber dan...	terimakasih atas ilmunya kepada narasumber dan...	terima kasih atas ilmunya kepada narasumber da...
4	Kurikulum merdeka 1. Pendidikan berpusat pa...	kurikulum merdeka br pendidikan berpusat pada ...	kurikulum merdeka baru pendidikan berpusat pad...

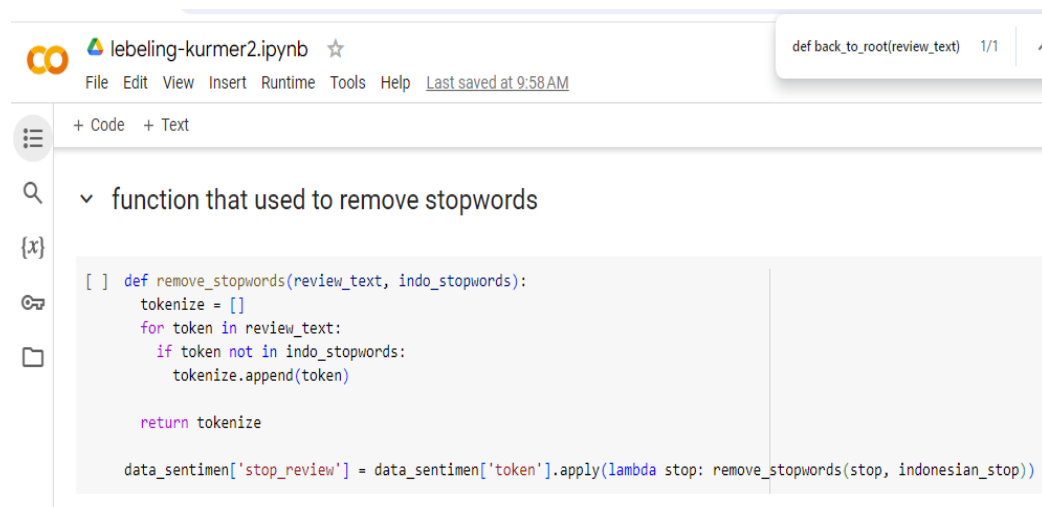
Gambar 4.11 Hasil Normalisasi

#### 4.2.5 Stopword Removal

*Stopword Removal* bertujuan untuk menghapus kata-kata umum yang banyak digunakan namun tidak memberikan pengaruh sentimen pada suatu kalimat. Proses *stopword* yang digunakan pada penelitian ini adalah dengan memanfaatkan *library* dari Sastrawi yang di dalamnya terdapat *corpus stopwords* bahasa Indonesia.

**Tabel 4.5 Contoh Hasil *Stopword Removal***

Sebelum <i>Stopword Removal</i>	Sesudah <i>Stopword Removal</i>
['sebagai', 'guru', 'kurikulum', 'merdeka', 'dapat', 'memberikan', 'motivasi', 'karakter', 'siswa', 'dalam',	['guru', 'kurikulum', 'merdeka', 'motivasi', 'karakter', 'siswa', 'pembelajaran']



```

def back_to_root(review_text) 1/1 ^
lebeling-kurmer2.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at 9:58 AM
+ Code + Text
function that used to remove stopwords
[ ] def remove_stopwords(review_text, indo_stopwords):
    tokenize = []
    for token in review_text:
        if token not in indo_stopwords:
            tokenize.append(token)
    return tokenize
data_sentimen['stop_review'] = data_sentimen['token'].apply(lambda stop: remove_stopwords(stop, indonesian_stop))

```

**Gambar 4.12 Alur Sub Proses *Stopword Removal***

	koment	clean_review	normalization	final_text	token	stop_review
0	Sebagai guru kurikulum merdeka ini dapat membe...	sebagai guru kurikulum merdeka ini dapat membe...	sebagai guru kurikulum merdeka ini dapat membe...	sebagai guru kurikulum merdeka dapat memberik...	[sebagai, guru, kurikulum, merdeka, dapat, mem...	[guru, kurikulum, merdeka, motivasi, karakter,...
1	Isinya buzzer semua kah? Komenan kagak ada ya...	isinya buzzer semua kah komenan kagak ada yang...	isinya buzzer semua kah komenan tidak ada yang...	isinya buzzer semua komenan tidak yang kreat...	[isinya, buzzer, semua, komenan, tidak, yang, ...	[isinya, buzzer, komenan, kreatif, tinggal, co...
2	Asesmen awal di lakukan sebelum pembelajaran y...	asesmen awal di lakukan sebelum pembelajaran y...	asesmen awal di lakukan sebelum pembelajaran y...	asesmen awal lakukan sebelum pembelajaran yan...	[asesmen, awal, lakukan, sebelum, pembelajaran...	[asesmen, lakukan, pembelajaran, bertujuan, ti...
3	Terimakasih atas ilmunya kepada narasumber dan...	terimakasih atas ilmunya kepada narasumber dan...	terima kasih atas ilmunya kepada narasumber da...	terima kasih atas ilmunya kepada narasumber ...	[terima, kasih, atas, ilmunya, kepada, narasum...	[terima, kasih, ilmunya, narasumber, pengetahu...
4	Kurikulum merdeka  1. Pendidikan berpusat pa...	kurikulum merdeka br pendidikan berpusat pada ...	kurikulum merdeka baru pendidikan berpusat pad...	kurikulum merdeka baru pendidikan berpusat pad...	[kurikulum, merdeka, baru, pendidikan, berpusa...	[kurikulum, merdeka, pendidikan, berpusat, pes...

**Gambar 4.13 Hasil Stopword Removal**

#### 4.2.6 Stemming

Tahap terakhir didalam penelitian ini adalah tahap *stemming*. Tahap *stemming* dilakukan oleh peneliti dengan tujuan untuk mendapatkan kata dasar dari suatu kata menggunakan *library* Sastrawi. Dalam kata lain kata pada dataset dari tahap sebelumnya diubah ke kata dasar sesuai KBBI.

```

[ ] from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

[ ] def back_to_root(review_text):
    stop_token = ' '.join(review_text)
    stem = StemmerFactory()
    create_stem = stem.create_stemmer()
    result_stem = create_stem.stem(stop_token)
    return result_stem

data_sentimen['stem_review'] = data_sentimen['stop_review'].apply(lambda stem:back_to_root(stem))

[ ] data_sentimen.head()

```

**Gambar 4.14 Alur Sub Proses Stemming**

### 4.3 Model

Tahap Pemodelan atau tahap model didalam penelitian ini melakukan pelebelan dengan 3 pemodelan yaitu dengan metode *lexicon based* untuk mendapatkan label kelas didalam dataset dan menggunakan 2 pemodelan klasifikasi dengan metode Naïve Bayes dan K-Nearest Neighbor.

#### 4.3.1 *Lexicon Based*

Untuk melakukan pelebelan didalam penelitian ini digunakanlah metode *lexicon based* dilakukan pada kelas dataset[40]. Terdapat 3 kelas yaitu kelas positif dan kelas negatif dan kelas netral didalam suatu komentar atau dokumen pada dataset. Kata kata pada dataset akan dibandingkan dengan sekumpulan kata yang ada dalam kamus *lexicon Indonesian Sentiment (Inset)*[40]. Perbandingan kata tersebut akan diberikan nilai atau *score* jika kata pada komentar atau dokumen didalam dataset terdapat kesamaan dengan kata yang ada didalam kamus *lexicon*. Jumlah nilai atau *score* ini akan menentukan komentar tersebut termasuk didalam label positif, lebel negatif atau lebel netral. Penentuan label sentimen berdasarkan pada total nilai polarity. Jika nilai di atas 0, maka komentar tersebut termasuk sentimen positif. Sebaliknya, jika nilai total di bawah 0, maka komentar termasuk sentimen negatif. Sedangkan jika total nilai polarity-nya adalah 0, maka komentar tersebut termasuk sentimen netral.

Selanjutnya dari hasil pemberian label dengan menggunakan kamus *lexicon based*, terdapat 19.980 komentar positif , 5.107 komentar negatif dan 1397 komentar yang bersifat netral. Dari hasil tersebut menghasilkan kata-kata yang sering diucapkan didalam komentar yang tervisualisasikan dari *word frequency*

### 15 dari 500 Sampel Kata yang Paling Sering Muncul



**Gambar 4.14 Hasil WordFreq**

Dari gambar tersebut tervisualisasikan bahwa kata yang sering terucap dalam siswa didalam dataset adalah kata kurikulum kemudian merdeka, guru, pembelajaran dan sebagainya. Dataset yang sudah diklasifikasikan ke dalam positif dan negatif selanjutnya juga dapat divisualisasikan dalam bentuk grafik dan wordcloud. Frekuensi kemunculan kata dari kelas positif yang sering muncul adalah kreatif, inovasi, dan dampak positif.



**Gambar 4.15 Hasil Word Frequency Positif**

Sedangkan frekuensi kemunculan kata dari kelas negatif yang sering muncul adalah beban, kurang, kurikulum, guru, dan proses ajar .



**Gambar 4.16 Hasil Word Cloud Negatif**

Sedangkan frekuensi kemunculan kata dari kelas netral yang sering muncul adalah konsep kurikulum, konsep ajar, ajar dan guru .



**Gambar 4.17 Hasil Word Cloud Netral**

### 4.3.2 Naïve Bayes

Pada tahap ini peneliti akan membagi dataset menjadi data latih (*training*) dan data uji (*test*). Peneliti akan membagi beberapa rasio perbandingan percobaan data latih dan data uji ini yaitu dari 80% data latih 20% data uji.

**Tabel 4.6 Perbandingan Data Latih dan Data Uji Naïve Bayes**

Data Latih	Data Uji
80%	20%

### 4.3.3 K-Nearest Neighbor

Pada tahap ini peneliti juga akan menggunakan konsep yang sama dengan algoritma sebelumnya yaitu Naïve Bayes. Peneliti akan membagi 2 data menjadi data latih dan data uji, 80% data latih dan data uji sebanyak 20% .

**Tabel 4.7 Perbandingan Data Latih dan Data Uji K-Nearest Neighbor**

Data Latih	Data Uji
80%	20%

Kemudian untuk algoritma K-Nearest Neighbor pada penelitian ini dilakukan uji variasi nilai k. Nilai k akan dicari yang terbaik dari bantuan algoritma KNN itu sendiri, pombobotan word2vec dan *10-fold cross validation* untuk mendapatkannya. Nilai k terbaik akan dilihat dari nilai *error* terkecil. Variasi k yang akan diuji yaitu 10 nilai ganjil pertama dimulai dari k=1 sampai k=19.

#### 4.3.4 Asses

Tahap terakhir yaitu tahap Asses merupakan tahap untuk dilakukannya evaluasi dari setiap model pada penelitian ini. Hasil evaluasi pada penelitian ini berisikan nilai-nilai *confusion matrix*, nilai akurasi, presisi, recall dan f1-score. Metode K-Nearest Neighbor menggunakan *10-fold cross validation* dalam pencarian nilai variasi k terbaik yang selanjutnya akan digunakan didalam model.

##### 4.3.4.1 Tahap pengujian dengan *Naïve Bayes*

Dalam penelitian ini penulis menggunakan algoritma *naive bayes* dengan jenis *Gaussian Naïve Bayes* karena setelah beberapakali uji coba algoritma tersebutlah yang paling sesuai dengan data yang digunakan oleh penulis. seperti yang sudah di utarakan sebelumnya didalam tahap model, penelitian ini membagi 80% untuk data latih dan 20% untuk data uji dan kemudian diolah dengan metode *Gaussian Naïve Bayes*. Berikut merupakan hasil dari pembagian dataset data latih:data uji.

**Tabel 4.8 Hasil Akurasi, Presisi, *Recall*, *f1-score* GNBC**

<b>Perbandingan data latih dan data uji</b>	<b>Akurasi</b>	<b>Presisi</b>	<b><i>Recall</i></b>	<b><i>f1-score</i></b>
80% : 20%	53.82%	72%	53.82%	59%

Kemudian untuk hasil dari *confusion matrix* dengan metode *Naïve Bayes* pada rasio 80%:20% didapat bahwa prediksi benar sentimen positif (*true positive*) ada sebanyak 2298 data dan prediksi benar untuk sentimen negatif (*true*



*negative*) sebanyak 490 data, sedangkan prediksi benar untuk sentimen netral (*true netral*) sebanyak 117 .

**Tabel 4.9 Tabel Confusion Matrix Naïve Bayes**

Hasil Aktual	Nilai Prediksi		
	Negatif	Netral	Positif
Negatif	490	264	268
Netral	84	117	78
Positif	1012	686	2298

Kemudian setelah mendapat rasio dengan nilai akurasi terbaik, maka selanjutnya dilakukan *cross validation*. *Cross validation* diperlukan untuk mengetahui kinerja minimum dan maksimum yang didapat.

**Tabel 4.10 Hasil 10-folds Cross Validation**

n-fold	Akurasi	Presisi	<i>Recall</i>	<i>f1-score</i>
1	54%	54%	54%	54%
2	52%	52%	52%	52%
3	52%	52%	52%	52%
4	53%	53%	53%	53%
5	54%	54%	54%	54%
6	53%	53%	53%	53%
7	51%	51%	51%	51%
8	52%	52%	52%	52%
9	54%	54%	54%	54%
10	52%	52%	52%	52%

#### 4.3.4.2 K-Nearest Neighbor

Metode selanjutnya adalah metode K-Nearest Neighbor (KNN). Pada penelitian ini metode KNN menggunakan 80% data latih dan 20% data uji seperti yang sebelumnya sudah dijelaskan. Untuk menentukan variasi k terbaik, data latih diolah menggunakan algoritma KNN, pembobotan word2vec dan 10-*fold validation*. selanjutnya adalah kembali melakukan uji akurasi, presisi, *recall*, dan *f1-score* dengan rasio data 80%:20% Algoritma perhitungan jarak yang digunakan pada metode K-Nearest Neighbor ini adalah *Euclidean Distance*. Berikut merupakan hasil akurasi, presisi, *recall* dan *f1-score* dari masing- masing rasio.

**Tabel 4.11 Hasil Akurasi, Presisi, *Recall*, dan *f1-score* KNN**

<b>Data Latih : Data uji</b>	<b>k</b>	<b>Akurasi</b>	<b>Presisi</b>	<b><i>Recall</i></b>	<b><i>f1-score</i></b>
80:20	11	79.70%	77.20%	79.70%	77.65%

Nilai akurasi rasio tersebut adalah 79.70%. Kemudian untuk hasil dari *confusion matrix* pada rasio 80%:20% ini didapatkan prediksi benar pada sentimen positif (*true positive*) adalah 3720 data dan prediksi benar untuk sentimen negatif (*true negative*) adalah 454 data, sedangkan prediksi benar untuk sentimen netral (*true netral*) sebanyak 43 .

**Tabel 4.12 Hasil *Confusion Matrix* KNN**

Hasil Aktual	Negative	Netral	Positif
Negative	454	31	537
Netral	76	43	160
Positif	242	34	3720

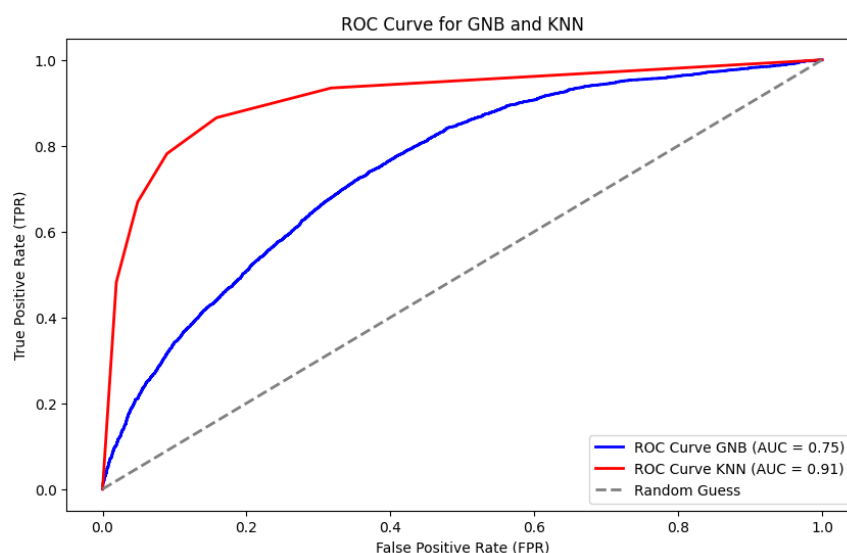
Setelah didapati rasio dengan nilai akurasi terbaik, selanjutnya dilakukan *cross validation*. *Cross validation* dilakukan untuk mengetahui kinerja minimum dan maksimum yang didapat. Penelitian ini menggunakan *10-folds cross validation* dimana nilai optimal didapatkan di  $k=11$ . Berikut tabel hasil dari *10-folds cross validation*.

**Tabel 4.13 Hasil 10-folds Cross Validation**

n-fold	Akurasi	Presisi	<i>Recall</i>	<i>f1-score</i>
1	79,61%	79,61%	79,61%	79,61%
2	79,47%	79,47%	79,47%	79,47%
3	79,42%	79,42%	79,42%	79,42%
4	81,17%	81,17%	81,17%	81,17%
5	79,99%	79,99%	79,99%	79,99%
6	79,57%	79,57%	79,57%	79,57%
7	79,47%	79,47%	79,47%	79,47%
8	79,85%	79,85%	79,85%	79,85%
9	79,41%	79,41%	79,41%	79,41%
10	78,99%	78,99%	78,99%	78,99%

#### 4.3.4.3 Evaluasi Kinerja Algoritma

Untuk mengevaluasi kinerja dari algoritma *Naïve Bayes* dan KNN akan dihitung menggunakan Kurva ROC dan menghitung luas area di bawah Kurva ROC atau disebut juga *Area Under Curve* (AUC). Kurva ROC menggambarkan hubungan antara *observed class* dan *predicted class*[41].



Gambar 4.18 Kurva ROC dengan Nilai AUC

Berdasarkan Gambar 4.18, didapatkan nilai AUC sebesar 0,75 pada algoritma *gaussian naïve bayes* sehingga klasifikasi nya masuk kedalam kategori cukup sedang untuk algoritma KNN mendapatkan nilai AUC sebesar 0,91 maka hasil klasifikasi algoritma KNN masuk kedalam kategori sangat baik. Pernyataan ini dapat kita lihat pada Tabel 2.3. Sehingga dapat kita simpulkan dalam penelitian ini algoritma KNN mampu bekerja lebih baik dibanding dengan algoritma *Gaussian Naïve Bayes*

#### 4.4 Interpretasi Hasil

Hasil analisis dari penelitian ini, telah dilakukan proses pengambilan data dengan menggunakan teknik *web scraping* yang didapatkan dari video YouTube tentang kurikulum merdeka menggunakan *tools* Google *Collaboratory* dimana bahasa pemrograman yang dipakai adalah bahasa pemrograman python.. Proses pengambilan data tersebut menghasilkan file excel yang berisikan 4 kolom rincian yaitu *publishedAt* yaitu waktu saat pengguna melontarkan komentarnya pada video YouTube ini, *authorDisplayedName* yaitu nama dari akun yang berkomentar, *textDisplayed* yaitu isi dari komentar, dan yang terakhir *likeCount* yaitu jumlah pengguna yang menyukai komentar tersebut. Penulis pada penelitian ini hanya akan menggunakan 1 kolom yaitu *textDisplayed* yang berisi komentar.

Setelah *scraping web* selesai maka didapatkan sebanyak 30.083 data komentar dari hasil *web scraping*. Untuk mendapatkan hasil yang lebih baik penulis membersihkan komentar dari kata-kata yang mengandung duplikat sehingga jumlah data yang digunakan pada penelitian ini adalah 26.539 komentar dataset yang diambil tersebut kemudian disimpan didalam file excel Pengambilan data komentar tersebut menggunakan *tools* Google *Collaboratory* dimana bahasa pemrograman yang dipakai adalah bahasa pemrograman python.

Tahapan tahapan yang dilakukan didalam penelitian ini diantaranya adalah *web scraping*, *text pre-processing*, pelabelan dataset dengan menggunakan kamus *lexicon*, klasifikasi data menggunakan 2 metode klasifikasi yaitu dengan Naïve Bayes dan K-Nearest Neighbor, pengujian *confusion matrix* dan visualisasi dimana visualisasi menggunakan *wordcloud*.

Tahap *pre-processing* pada penelitian ini meliputi beberapa tahapan lagi yaitu seperti *case folding*, *cleaning*, *tokenize*, *normalize*, *stemming*, dan *stopword removal*. Dataset yang sudah dilakukan tahapan-tahapan *pre-processing* ini selanjutnya disebut dengan data komentar bersih yang akhirnya berjumlah 26.539 data komentar bersih.

Data yang sudah dibersihkan dan terstruktur selanjutnya dilakukan proses pelabelan yaitu melabelkan data menjadi 3 kelas sentimen yaitu kelas positif, kelas negatif dan kelas netral. Pelabelan kelas didalam dataset ini menggunakan salah satu tahap pemodelan yaitu *lexicon based*. Kata-kata atau data komentar yang ada pada dataset dibandingkan dengan dokumen kamus *lexicon*. Data komentar tersebut akan dibandingkan dan diberikan nilai atau *score* sesuai dengan kamus *lexicon*. Jumlah dari *score* tersebut akan menentukan data komentar tersebut memiliki label kelas positif, kelas negatif atau kelas netral. Kamus *lexicon* yang digunakan bersumber dari *Indonesian Sentiment* (Inset). Hasil dari perbandingan data komentar dengan dokumen kamus *lexicon* ini terdapat 19.979 komentar berlabel positif, 5.108 komentar berlabel negatif dan terdapat 1.398 komentar berlabel netral.

Kemudian penulis juga memvisualisasikan *word frequency* atau frekuensi kata yang sering muncul dan *word cloud* pada masing-masing label kelas positif, kelas negatif dan kelas netral yang ditunjukkan pada gambar 4.17, gambar 4.18 dan gambar 4.19. Data yang sudah diberikan label selanjutnya disimpan dalam file csv untuk proses selanjutnya.

Sebelum memasuki tahap *asses*, penulis membagi terlebih dahulu data uji dan data latih dari dataset yang sebelumnya sudah diproses. penulis membagi rasio pembagian data latih dan sebanyak 80% dan data uji sebanyak 20%. Pembagian rasio ini dilakukan pada kedua model klasifikasi yaitu Naïve bayes dan K-Nearest Neighbor. Model klasifikasi K-Nearest Neighbor (KNN) setelah pembagian rasio tersebut kemudian akan dilakukan uji variasi nilai k. Nilai k terbaik dicari dari bantuan algoritma KNN, pembobotan word2vec dan *10-fold cross validation*. Nilai k terbaik akan terlihat dari nilai *error* terkecil dengan variasi k yang akan diuji yaitu 10 nilai ganjil pertama dimulai dari k=1 sampai k=19.

Tahap terakhir yaitu tahap *asses* pada penelitian ini. Tahap ini dilakukan evaluasi dari 2 model klasifikasi yaitu Naïve Bayes dan K-Nearest Neighbor. Evaluasi tersebut berisikan nilai *confusion matrix*, nilai akurasi, nilai presisi, nilai *recall*, dan *f1-score* dari data uji pada masing masing model klasifikasi yaitu Naïve Bayes dan KNN. Kemudian metode KNN selanjutnya akan dilakukan *10-fold cross validation* dalam pencarian nilai variasi k terbaik. Hasil nilai akurasi, presisi, *recall*, dan *f1-score* dari masing-masing metode klasifikasi ditunjukkan pada tabel 4.8 untuk metode Naïve Bayes dan tabel 4.11 untuk metode KNN. Berikut hasil rangkuman hasil nilai akurasi, presisi, *recall*, dan *f1-score*.

**Tabel 4.14 Hasil Akurasi, Presisi, *Recall*, dan *f1-score* NBC dan KNN**

Rasio	Naïve Bayes				K-Nearest Neighbor			
	Akurasi	Presisi	<i>Recall</i>	<i>f1</i>	Akurasi	Presisi	<i>Recall</i>	<i>f1</i>
80 : 20	53.82%	72%	53.82%	59%	79.70%	77.20%	79.70%	77.65%

Pada tabel diatas dapat terlihat masing-masing nilai akurasi, nilai presisi, nilai *recall*, dan nilai *f1-score* dari rasio yang sebelumnya sudah ditentukan oleh penulis. Berikut definisi dari *confusion matrix* tersebut

1. Akurasi

Akurasi merupakan jumlah dari prediksi data yang benar dibagi oleh jumlah semua data. Dengan kata lain akurasi mengukur sejauh mana model dapat memprediksi dengan benar secara keseluruhan.

2. Presisi

Presisi merupakan pengukuran sejauh mana hasil prediksi positif dari model sudah benar. Presisi menunjukkan berapa data dari prediksi positif yang benar akan dibagi dengan semua prediksi positif.

3. *Recall*

*Recall* merupakan pengukuran sejauh mana indentifikasi dengan benar dari model untuk kasus positif aktual. *Recall* menunjukkan berapa banyak data dari kasus positif aktual yang berhasil diidentifikasi.

4. *F1-Score*

*F1-Score* merupakan gabungan dari nilai presisi dan nilai *recall* dimana untuk memberikan gambaran keseluruhan dari kinerja model yang digunakan. Dapat disimpulkan *f1-score* merupakan nilai rata-rata dari



presisi dan *recall*. Antara presisi dan *recall* *f1-score* dapat memberikan keseimbangan yang berguna ketika terdapat ketidakseimbangan kelas yang signifikan.

Selanjutnya pada metode K-Nearest Neighbor dilakukan *cross validation*. *Cross validation* dilakukan untuk mengetahui kinerja minimum dan maksimum yang didapat. Peneliti menggunakan *10-folds cross validation* dimana nilai optimal didapatkan di  $k=11$  dengan hasil *10-folds cross validation* metode KNN ditunjukkan pada tabel 4.12.

Dari penelitian yang sudah dilakukan, penulis mendapatkan bahwa model terbaik didalam penelitian ini adalah metode KNN pada dataset komentar YouTube tentang kurikulum merdeka pada rasio 80% : 20% data latih : data uji mendapat nilai akurasi sebesar 79,70% sedangkan untuk *Gaussian Naïve Bayes* dengan rasio 80% : 20% data latih : data uji sebesar 53.82%. Pada penelitian ini model KNN lebih baik dibandingkan dengan metode *Gaussian Naïve Bayes* hal ini dapat dilihat pada tabel 4.14 bahwa dari semua rasio nilai akurasi dan nilai rata-rata presisi dan *recall* (*f1-score*) dari metode KNN lebih besar dibandingkan dengan metode *Gaussian Naïve Bayes*. Selain menggunakan beberapa metode diatas penulis juga mengukur kinerja ke dua algoritma *Gaussian Naïve Bayes* dan KNN menggunakan kurva roc dengan membandingkan nilai AUC kedua nya algoritma *gaussian naïve bayes* mendapatkan nilai AUC sebesar 0,75 sehingga klasifikasi nya masuk kedalam kategori cukup. Sedangkan untuk algoritma KNN mendapatkan nilai AUC sebesar 0,91 sehingga hasil klasifikasi algoritma KNN masuk kedalam kategori sangat baik