Tugas Besar Wawasan Global TIK

Analisis Sentimen Kebijakan Pemerintah Mengenai Vaksin COVID-19 Di Indonesia Menggunakan Metode Naive Bayes Classifier Pada Sosial Media Twitter

Oleh Kelompok 152:

Fadhlurrahman Akbar Nasution (1301194258) / IF-42-GAB01

Firdaus Putra Kurniyanto (1301190385) / IF-42-GAB01

Ignasius Nindra Karisma Forestyanto (1301194138) / IF-42-GAB02

Kurniadi Ahmad Wijaya (1301194024) / IF-42-GAB01

Priyan Fadhil Supriyadi (1301190442) / IF-42-GAB01

S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

BANDUNG

2020

Progress Pengerjaan Minggu 1 :

https://www.youtube.com/watch?v=2ogUovUISms

Progress minggu minggu pertama berfokus kepada proses pengumpulan dan pemodelan data serta melakukan klasifikasi menggunakan library textblob

Progress Pengerjaan Minggu 2:

https://www.youtube.com/watch?v=ajvXne5KlvM

Progress akhir berfokus kepada pemodelan menggunakan metode Naive Bayes Classifier dan melakukan kesimpulan dari hasil data uji.

Hasil Pengerjaan Projek:

http://bit.ly/SentimenVaksinCovid_Kelompok152

Merupakan hasil utama dari projek ini dengan jumlah data mencapai 8000an. Melakukan running pada proyek ini akan memakan waktu beberapa lama (30 menit) karena jumlah data yang banyak untuk di translate dan diklasifikasikan.

Hasil Penyederhanaan projek (1000 Data):

https://colab.research.google.com/github/ShinyQ/Analisis-Sentimen-Kebijakan-Vaksinasi-COVID-19-Pemerintah_Naive-Bayes-Classifier/blob/main/Tugas_Besar_WGTIK_Simplified.ipynb

Merupakan hasil proyek yang dapat dicoba karena hanya mengambil 1000an data dalam jangka waktu 1 minggu dari waktu di running.

DAFTAR PUSTAKA

1. Latar Belakang

Menurut *World Health Organization* (WHO) dalam situs webnya, Indonesia menjadi negara dengan 636.154 kasus terindikasi COVID-19 dan 19.248 diantaranya mengalami kematian. Angka dari data yang disebutkan diatas bukanlah merupakan angka yang kecil. Angka tersebut cenderung akan membesar jika tidak ada penanganan khusus dari pemerintah untuk menangani COVID-19.

Lockdown, Pembatasan Sosial Berskala Besar (PSBB), bahkan sosialisasi masker sudah diterapkan di Indonesia, tapi hingga penghujung tahun 2020 ini masih belum terlihat grafik dari penyebaran COVID-19 menurun, tapi kabar baik sudah mulai datang di penghujung tahun 2020, dilansir dari website berita Indonesia, Kompas.com, pada 16 Desember 2020, Indonesia telah mendatangkan 1,2 Juta dosis vaksin corona Sinovac dan masih menunggu 1,8 Juta dosis yang akan menyusul di kemudian hari.

Vaksin sudah didatangkan dan dipesan, masyarakat hanya tinggal menunggu kebijakan yang diambil pemerintah. Proses penungguan kebijakan pemerintah ini tentu membuat masyarakat membuat sentimen yang berbeda beda satu sama lain. Dimana saat berita mengenai vaksin corona Sinovac itu muncul,vaksin tersebut langsung menjadi trending sebagai kata yang paling sering dibicarakan oleh masyarakat Indonesia. Hal ini menarik karena banyak spekulasi dari masyarakat mengenai pemberian vaksin tersebut dan kami membagi spekulasi ini menjadi satu spekulasi yang paling sering dibicarakan yaitu mengenai analisis kebijakan pemerintah untuk memberikan vaksin secara gratis atau tidak.

Analisis sentimen merupakan analisis terhadap suatu peristiwa dari pendapat yang didasarkan pada sikap seseorang tentang suatu objek. Analisis sentimen biasanya dilakukan untuk mengumpulkan dan mengetahui opini masyarakat dalam postingan Blog, Twitter, Facebook, dan yang lainnya. Analisis sentimen dibutuhkan dengan tujuan untuk mengetahui opini publik terhadap suatu objek. Opini-opini tersebut bisa berupa opini negatif atau positif tergantung dari pandangan publik terhadap objek tersebut. Oleh karena itu dibutuhkan suatu analisis terhadap opini-opini tersebut dalam penelitian ini agar bisa dijadikan tolak ukur baik atau buruknya kebijakan yang akan diambil pemerintah perihal vaksin.

2. Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sentimen masyarakat terhadap kebijakan pemerintah mengenai vaksin COVID-19. Proses analisis-nya akan dilakukan berdasarkan tweet yang menyertakan tagar vaksin dan pencarian di twitter dengan keyword vaksin covid 19.

3. Tujuan dan Manfaat

Tujuan dari penelitian ini adalah untuk melakukan analisis sentimen dari masyarakat Indonesia terhadap kebijakan pemerintah mengenai vaksin COVID-19 yang akan dibagikan secara gratis kepada masyarakat Indonesia.

Tujuan ini berguna bagi Pemerintah Indonesia untuk mengetahui kepercayaan masyarakat terhadap vaksin COVID-19 yang akan mulai dibagikan.

4. Metode Penelitian

4.3 Metode yang Digunakan

A. Supervised Learning

Pendekatan supervised learning mempunyai input dan output yang dapat dibuat menjadi suatu model hubungan matematis sehingga mampu melakukan prediksi dan klasifikasi berdasarkan data yang telah ada sebelumnya.

B. Naive Bayes Classifier

Naïve Bayes Classifier merupakan sebuah metode klasifikasi yang berakar pada teorema Bayes. Naive Bayes Classifier bekerja sangat baik dibanding dengan model classifier lainnya seperti Decision Trees ataupun Neural Network. Keuntungan penggunaan metode ini adalah metode ini hanya membutuhkan jumlah data pelatihan (training data) yg kecil untuk menentukan estimasi parameter yg diperlukan dalam proses pengklasifikasian. Adapun Naïve Bayes Classifier memiliki bentuk umum seperti berikut:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

A = hipotesis data B adalah kelas khusus
B = data dengan kelas tidak dikenal
P(A|B) = probabilitas hipotesis A didasarkan pada kondisi B
P(B|A) = probabilitas hipotesis B didasarkan pada kondisi A
P(A) = probabilitas hipotesis A
P(B) = probabilitas B

4.2 Bahan Riset

A. API Twitter

API adalah singkatan dari Application Programming Interface, dan memungkinkan developer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. API yang akan digunakan pada tugas besar ini adalah API Twitter terutama post twitter yang berhubungan dengan vaksin yaitu berupa tagar vaksin dan pencarian vaksin covid. Data yang akan di scrapping ini nantinya akan diolah serta dimodelkan untuk mendapatkan sebuah kesimpulan.

4.3 Peralatan Riset

A. Google Collaboratory

Google Collaboratory atau Google Collab merupakan tools yang berbasis cloud dan free untuk tujuan penelitian. Google collab dibuat dengan environment jupyter dan mendukung semua library yang dibutuhkan dalam lingkungan pengembangan Artificial Intelligence (AI). Google Collab memiliki kelebihan yaitu dapat mengaksesnya secara gratis, karena menggunakan cloud computer maka memiliki spesifikasi yang bagus, tidak perlu melakukan konfigurasi apapun namun kita perlu menginstal library packagenya jika ingin menambahkan library baru, dan memudahkan kita untuk melakukan sharing dengan orang lain.

B. Python

Python merupakan bahasa pemrograman tingkat tinggi. Hal ini disebabkan karena kode yang dituliskan akan di compile menjadi bytecode dan dieksekusi sehingga python cocok digunakan untuk scripting language, data mining dan lain sebagainya. Python memiliki struktur konstruksi yang kuat (blok kode, fungsi, class, module, dan package) serta konsisten menggunakan konsep Object Oriented Programing (OOP).

C. Numpy

NumPy (Numerical Python) adalah library python yang fokus pada scientific computing. Numpy memiliki kemampuan untuk membentuk objek N-dimensional array, yang mirip dengan list pada python. Numpy array memiliki keunggulan yaitu konsumsi memori yang lebih kecil serta runtime yang lebih cepat.

D. Pandas

Pandas (Python Data Analysis Library) merupakan sebuah open source python package/ library dengan lisensi BSD yang menyediakan struktur data dan analisis data yang mudah digunakan dan berkinerja tinggi untuk bahasa pemrograman python. Pandas berfungsi untuk membersihkan data ke dalam sebuah bentuk yang cocok untuk dianalisis (seperti tabel). Pandas dapat menyelaraskan data untuk perbandingan dan penggabungan set data serta penanganan data yang hilang.

E. Seaborn

Seaborn merupakan pustaka visualisasi data pada python yang bersifat open source yang berlisensi BSD dan dibangun berdasarkan pustaka matplotlib. Seaborn mempermudah analisis data karena seaborn memvisualisasikan data dengan indah dan simple tanpa kostumisasi yang rumit

F. TextBlob

TextBlob adalah library Python untuk memproses data tekstual. Ini menyediakan API sederhana untuk menjalankan pemrosesan bahasa alami (NLP) umum seperti penandaan part-of-speech, ekstraksi frasa kata benda, analisis sentimen, klasifikasi, terjemahan, dan banyak lagi.

G. NLTK

Natural Language Toolkit atau disingkat NLTK, adalah library python untuk bekerja dengan pemodelan teks. NLTK menyediakan tools yang baik dalam mempersiapkan teks sebelum digunakan pada machine learning.

H. Tweepy

Tweepy adalah salah satu library python yang mudah digunakan untuk mengakses API dari Twitter. Library Tweepy ini berfungsi untuk mempermudah mendapatkan data tweet dari pengguna twitter berdasarkan keyword yang akan kita gunakan. Jika ingin mendapatkan tweet tentang sebuah topik tertentu, maka buat topik tersebut menjadi sebuah keyword kemudian tweet yang mengandung keyword yang telah diisi akan muncul.

5. Langkah Implementasi

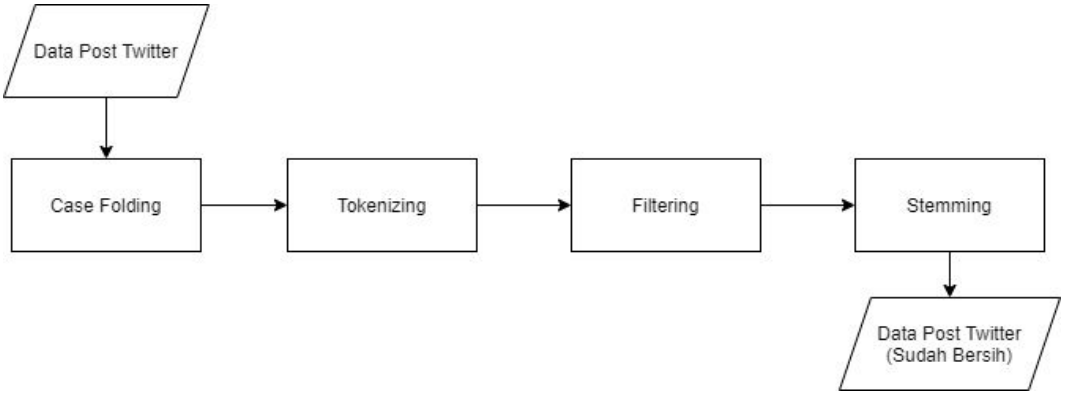
5.1 Proses Pelaksanaan

Proses pertama yang dilakukan dalam melakukan klasifikasi sentimen adalah proses pengumpulan data atau *crawling*. Setelah pengambilan data maka data dibersihkan dengan metode *pre-processing* dengan tahapan-tahapan case folding, tokenizing, filtering dan stemming. Setelah data bersih maka dilakukan pemilihan data latih dan data uji. Kemudian dilakukan pemodelan klasifikasi menggunakan data latih dan pengujian model dengan data uji. Dan terakhir dilakukan perhitungan akurasi dari model yang digunakan.

Data yang didapat dari hasil *crawling* belum bisa langsung diklasifikasikan karena data tersebut masih terdapat banyak simbol dan kata-kata yang tidak diperlukan, oleh karena itu maka memerlukan *pre-processing* data agar data lebih terstruktur dan bersih sehingga bisa diklasifikasikan. Ada beberapa tahapan dalam *preprocessing* data diantaranya, *case folding, tokenizing, stemming, serta filtering.*

1. *Case folding,* pada data *tweet* dilakukan proses perubahan dari huruf besar menjadi huruf kecil dan menghilangkan seluruh tanda baca pada kalimat.
2. *Tokenizing,* pada data *tweet* setiap kata akan dipisahkan berdasarkan spasi yang ditemukan.
3. *Stemming,* yaitu pengubahan kata berimbuhan menjadi kata dasar pada data *tweet.*
4. *Filtering,* yaitu pembuangan kata-kata tidak penting dari hasil pengumpulan data atau *crawling*

Adapun proses tersebut digambarkan dengan diagram alir (flowchart) dibawah ini:

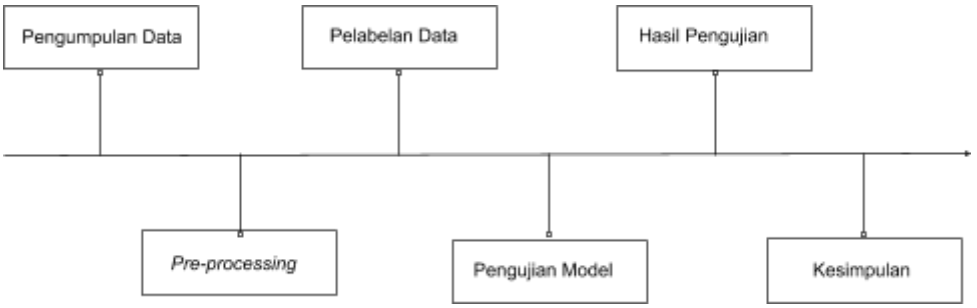


Proses pelabelan dilakukan secara manual, dengan menentukan suatu data masuk kedalam kelas positif atau kelas negatif. Pengelompokan kelas positif dilihat dari isi *tweet* mengandung kata bermakna positif, mendukung dan pernyataan setuju. Kelas negatif merupakan kelas dengan data yang mengandung kata bermakna negatif, ejekan, dan kontra.

Setelah melakukan pelabelan data, data tersebut dibagi menjadi data latih dan data uji secara acak. Data latih digunakan untuk melakukan pelatihan model yang telah dirancang agar program dapat mengerti apa yang ingin dicapai. Data latih terdiri atas *tweet* dan labeling yang dilakukan secara manual. Data uji digunakan sebagai penentu untuk pengujian model *classifier,* setiap data diprediksi berdasarkan 3 kelas, yaitu positif , negatif, dan netral. Adapun acuan dari kelas disebut positif apabila polaritas suatu data yang telah diolah lebih dari 0.0, serta apabila polaritas data kurang dari 0.0 atau negatif maka data bersifat negatif dan jika polaritas data sama dengan 0.0 maka kelas data bersifat netral.

Setelah melakukan tahapan dari pengambilan data sampai pengujian model menggunakan data uji yang telah dipilih secara acak, maka akan diperoleh hasil pengujian, dimana hasil pengujian tersebut akan dilakukan analisis untuk mendapatkan suatu kesimpulan.

5.2 Timeline Kegiatan



Hari / Tanggal	Kegiatan
Minggu, 20 Desember	Pengumpulan Data
Rabu, 23 Desember	*Pre-processing*
Jumat, 25 Desember	Pelabelan Data
Sabtu, 26 Desember	Pengujian Model
Senin, 28 Desember	Menganalisis hasil pengujian
Kamis, 31 Desember	Mendapatkan kesimpulan

6. Pembagian Tugas

Adapun bentuk pembagian tugas pada tugas besar Wawasan Global TIK dijelaskan pada tabel sebagai berikut:

NIM	Nama	Tugas
1301194258	Fadhlurrahman Akbar Nasution	Melakukan Visualisasi Data Dari Hasil Pemodelan
1301190385	Firdaus Putra Kurniyanto	Melakukan Dokumentasi, Merancang Pemanggilan API Twitter pada projek.
1301194138	Ignasius Nindra Karisma Forestyanto	Melakukan Visualisasi Data Dari Hasil Pemodelan
1301194024	Kurniadi Ahmad Wijaya	Memimpin Projek, Memodelkan Dan Merancang Klasifikasi Data Sentimen Postingan Pada Twitter Menggunakan Naive Bayes Classifier
1301190442	Priyan Fadhil Supriyadi	Melakukan Pengetesan Akurasi Pemodelan dari klasifikasi data TextBlob

7. Hasil Implementasi

Hasil pengolahan serta proses klasifikasi data serta kesimpulan dari analisis data dapat dilihat pada link : http://bit.ly/SentimenVaksinCovid_Kelompok152

Berikut merupakan ringkasan proses implementasi yang telah dilakukan:

1. Melakukan crawling data dan berhasil mengumpulkan 8000an data mulai dari tanggal 15 - 24 Desember.
2. Menghapus kolom dan baris yang tidak digunakan dalam data yang dikumpulkan.

3. Melakukan proses pembersihan data dengan library tweet preprocessor, NLTK, dan menghapus duplikasi serta mengecek informasi dari data. Pada akhirnya setelah proses pembersihan tersisa 3780 data.
4. Melakukan penerjemahan data dari bahasa indonesia ke bahasa inggris untuk dapat diolah menggunakan library NLP TextBlob.
5. Melakukan pemodelan data untuk diklasifikasikan positif, negatif, atau netral menggunakan library textblob.
6. Menampilkan serta memvisualisasikan hasil pengolahan data.
7. Melakukan pemodelan dengan metode Naive Bayes Classifier dengan mengambil setengah sampel random dari masing-masing klasifikasi (netral, positif, negatif) dari data yang telah diklasifikasikan sebelumnya menggunakan TextBlob sebagai train dan prediksi data yang akan diprediksi.
8. Memvisualisasikan, menampilkan akurasi, dan mengambil kesimpulan dari hasil pemodelan menggunakan metode Naive Bayes Classifier.

Adapun penjelasan lengkap proses dari implementasi projek ini yaitu:

7.1 Proses Crawling Data (Pengumpulan Data)

Proses crawling (pengambilan data) yang kami lakukan pada projek ini adalah dengan menggunakan library tweepy. Adapun sebelum melakukan pengambilan data dilakukan beberapa proses yaitu:

1. Autentikasi API Twitter menggunakan Consumer Key, Consumer Secret, Access Token, dan Access Token Secret yang didapatkan dari registrasi akun developer pada https://developer.twitter.com/en/application.

2. Setelah melakukan proses autentikasi langkah selanjutnya yang kita lakukan adalah membuat fungsi proses pengambilan data twitter tersebut adapun dalam fungsi tersebut (scraptweets) langkah awal yang dilakukan yaitu membuat kumpulan kolom dengan variabel db_tweets sebagai acuan field data yang akan kita masukkan . Kolom-kolom tersebut sendiri terdiri atas:

    - username : nama dari orang yang melakukan tweet (postingan)
    - acctdesc : deskripsi orang yang melakukan tweet (postingan)
    - location : lokasi daerah pemilik akun
    - following : jumlah pengikut orang yang melakukan tweet (postingan)
    - totaltweets : jumlah postingan orang yang melakukan tweet (postingan)
    - usercreatedts : waktu akun pengguna dibuat
    - tweetcreatedts : waktu tweet (postingan) dibuat
    - retweetcount : jumlah orang yang melakukan retweet
    - text : isi pesan tweet (postingan)
    - hashtags : tagar yang terdapat didalam tweet (postingan)
    - follower : orang yang mengikuti orang yang melakukan tweet (postingan)

```
# Autentikasi API Twitter
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

def scraptweets(search_words, date_since, date_until, numTweets):
    # Membuat Kolom Untuk Di Export Ke Excel
    db_tweets = pd.DataFrame(columns=[
                                      'username', 'acctdesc', 'location', 'following',
                                      'totaltweets', 'usercreatedts', 'tweetcreatedts',
                                      'retweetcount', 'text', 'hashtags', 'followers',
                                     ])

    # Melakukan Query Pencarian Data Tweet Sesuai Kata Kunci Dan Tanggal
    tweets = tweepy.Cursor(
                    api.search, q=search_words, lang="id",
                    since=date_since, until=date_until,  tweet_mode='extended').items()

    # Merubah Kumpulan Item Hasil Tweet Menjadi Kumpulan Data Dalam Array List
    tweet_list = [tweet for tweet in tweets]

    # Melakukan Perulangan Untuk Data Tweet Untuk Dimasukkan Kedalam CSV
    for tweet in tweet_list:
        username = tweet.user.screen_name
        acctdesc = tweet.user.description
        location = tweet.user.location
        following = tweet.user.friends_count
        followers = tweet.user.followers_count
        totaltweets = tweet.user.statuses_count
        usercreatedts = tweet.user.created_at
        tweetcreatedts = tweet.created_at
        retweetcount = tweet.retweet_count
        hashtags = tweet.entities['hashtags']

        try:
            text = tweet.retweeted_status.full_text
        except AttributeError:
            text = tweet.full_text
```

3. Selanjutnya di dalam fungsi tersebut dilakukan pemanggilan data pada API twitter sesuai dengan kata kunci dan jarak waktu yang ditentukan untuk mengambil tweet (postingan yang sesuai) adapun pengambilan data dilakukan secara manual setiap harinya antara tanggal 15 - 24 Desember karena limit pengambilan data pada API Twitter. Pada akhirnya kami berhasil mengumpulkan sekitar 8000-an data.

```
        # Membuat Array Kumpulan Data Sesuai Kolom Dan Memasukkan Kedalam Array Data Tweet
        ith_tweet = [
                        username, acctdesc, location, following, followers, totaltweets,
                        usercreatedts, tweetcreatedts, retweetcount, text, hashtags
                    ]

        db_tweets.loc[len(db_tweets)] = ith_tweet

    # Export Data Kumpulan Tweet Ke File CSV
    filename = 'covid_vaccine_tweets.csv'
    db_tweets.to_csv(filename, index=False)
    print('Scraping has completed!')


# Format Pencarian Data Tweet
search_words = "#vaksin OR #vaksincovid19 OR #vaksincovid OR #VaksinUntukKita OR #vaksingratis"
date_since = "2020-12-23"
date_until = "2020-12-24"
numTweets = 3000

scraptweets(search_words, date_since, date_until, numTweets)
```

4. Melakukan perulangan untuk data yang telah diambil dan mengumpulkannya dalam sebuah array untuk digabungkan nantinya dalam sebuah file csv yang akan diolah sesuai kolom-kolom yang telah ditentukan sebelumnya.

5. Setelah fungsi pemanggilan data tweet telah dibuat dilakukan pemanggilan fungsi dengan parameter kata kunci yaitu "#vaksin OR #vaksincovid19 OR #vaksincovid OR #VaksinUntukKita OR #vaksingratis", kemudian tanggal rentang tweet di posting serta jumlah tweet yang akan diambil.

7.2 Proses Wrangle Data (Pembersihan Data)

Pada projek ini kami melakukan wrangle data (pembersihan data) dengan menggunakan library tweet-preprocessor. Adapun langkah-langkah yang dilakukan dalam proses wrangle data (pembersihan data) ini adalah :

1. Langkah pertama yang kami lakukan adalah mengecek apakah ada data kosong dari data yang sudah didapat dari proses crawling (pengambilan data), data yang dicek pada proses ini adalah :
   - username : nama dari orang yang melakukan tweet (postingan)
   - tweetcreatedts : waktu tweet (postingan) dibuat
   - text : isi pesan tweet (postingan)

```
data.isnull().sum()
```

2. Langkah selanjutnya adalah proses *tokenizing*, *tokenizing* adalah proses memisahkan kata berdasarkan spasi yang ditemukan. Proses *tokenizing* ini bertujuan untuk mempermudah menganalisa setiap kata-kata dari setiap kalimat pada data tweet yang sudah dikumpulkan.

```
def preprocessing_data(x):
    return p.clean(x)

def tokenize_data(x):
    return p.tokenize(x)

data['tweet_clean'] = data['text'].apply(preprocessing_data)
data['tweet_clean'] = data['tweet_clean'].apply(tokenize_data)
data = data.drop_duplicates()
```

3. Langkah selanjutnya adalah menerjemahkan setiap kata-kata dari kalimat yang sudah dipisah tadi menggunakan proses *tokenizing* dari bahasa Indonesia menjadi bahasa Inggris, hal ini dilakukan karena library yang dipakai untuk menganalisis data menggunakan dasar bahasa Inggris.

Disini kami melakukan penerjemahan bahasa menggunakan library google_trans_new.

```
from google_trans_new import google_translator
translator = google_translator()

def convert_eng(tweet):
    return translator.translate(tweet,lang_tgt='en')

data['tweet_english'] = data['tweet_clean'].apply(convert_eng)
```

4. Langkah selanjutnya setelah menerjemahkan kata-kata dari setiap kalimat adalah proses *stemming* data, proses *stemming* ini digunakan untuk menemukan kata dasar dengan cara menghilangkan kata imbuhannya.

```
ps = PorterStemmer()

def stemming_data(x):
    return ps.stem(x)

data['tweet_english'] = data['tweet_english'].apply(stemming_data)
```

7.3 Proses Pemodelan Data Menggunakan TextBlob

Setelah melakukan crawling data dan wrangle data. Kami melakukan pemodelan data menggunakan TextBlob. Adapun langkah-langkah yang kami lakukan dalam proses pemodelan data adalah:

1. Langkah pertama data yang telah crawling (dikumpulkan) dan wrangle (dibersihkan) kami simpan kedalam Github untuk kami gunakan dalam melakukan pemodelan data di Google Collaboratory. Kita memanggil data yang telah dikumpulkan

2. Langkah selanjutnya setelah kita memanggil data kita akan melakukan analisis terhadap data tweet tersebut menggunakan TextBlob. Dalam analisis ini kami membagi kategori data tweet menjadi 3 yaitu Positif, Netral, dan Negatif. Dalam Proses ini kita mendapatkan berapa banyak data tweet yang dimiliki dan dari data tweet tersebut kita menganalisis data tersebut masuk kedalam kategori yang telah kita sediakan. Adapun hasil yang didapatkan dari klasifikasi menggunakan TextBlob tersebut adalah :
   - Positif : 2212
   - Netral : 1317
   - Negatif : 251

   Sehingga untuk sementara sentimen positif lebih banyak jika dibandingkan dengan yang lainnya.

```
data_tweet = list(data['tweet_english'])
polaritas = 0

status = []
total_positif = total_negatif = total_netral = total = 0

for i, tweet in enumerate(data_tweet):
    analysis = TextBlob(tweet)
    polaritas += analysis.polarity

    if analysis.sentiment.polarity > 0.0:
        total_positif += 1
        status.append('Positif')
    elif analysis.sentiment.polarity == 0.0:
        total_netral += 1
        status.append('Netral')
    else:
        total_negatif += 1
        status.append('Negatif')

    total += 1
```

```
Hasil Analisis Data:
Positif = 2212
Netral = 1317
Negatif = 251

Total Data : 3780
```

3. Langkah selanjutnya setelah kita melakukan analisis data tweet maka data tersebut akan menambahkan kolom klasifikasi kedalam datanya.

```
status = pd.DataFrame({'klasifikasi': status})
data['klasifikasi'] = status
data.tail()
```

7.4 Proses Visualisasi Data Yang Telah Dimodelkan

Visualisasi merupakan langkah dalam melakukan penyajian data yang harus selalu diperhatikan, melalui visualisasi kita dapat memperlihatkan apa yang kita olah dan apa yang kita proses dalam suatu data tersebut hingga menjadi suatu sajian yang lebih jelas serta mudah dipahami. Untuk Visualisasi data dalam topik kami yaitu kami menggunakan chart pie yang terdiri dari 3 bagian yaitu Sentimen Positif, Negatif, serta Netral. Disamping menggunakan pie chart yang membagi ketiga status data kami juga menggunakan word cloud untuk melihat kata apa saja yang paling banyak muncul dari data yang telah diolah.

Adapun hasil pengolahan chart tersebut menggunakan library wordcloud, matplotlib dan seaborn untuk melakukan visualisasi data. Adapun proses menampilkanya yaitu dengan membuat fungsi plot_cloud setelah itu kita bagi setiap kata menjadi array untuk diolah oleh wordcloud agar dapat ditampilkan kata yang paling banyak muncul. Adapun proses visualisasinya adalah :

```
from wordcloud import WordCloud, STOPWORDS

def plot_cloud(wordcloud):
    plt.figure(figsize=(12, 8))
    plt.imshow(wordcloud)
    plt.axis("off")

all_words = ' '.join([tweets for tweets in data['tweet_english']])
wordcloud = WordCloud(
                width = 3000, height = 2000, random_state=3,
                background_color='white', colormap='Set2',
                collocations=False, stopwords = STOPWORDS
            ).generate(all_words)

plot_cloud(wordcloud)
```

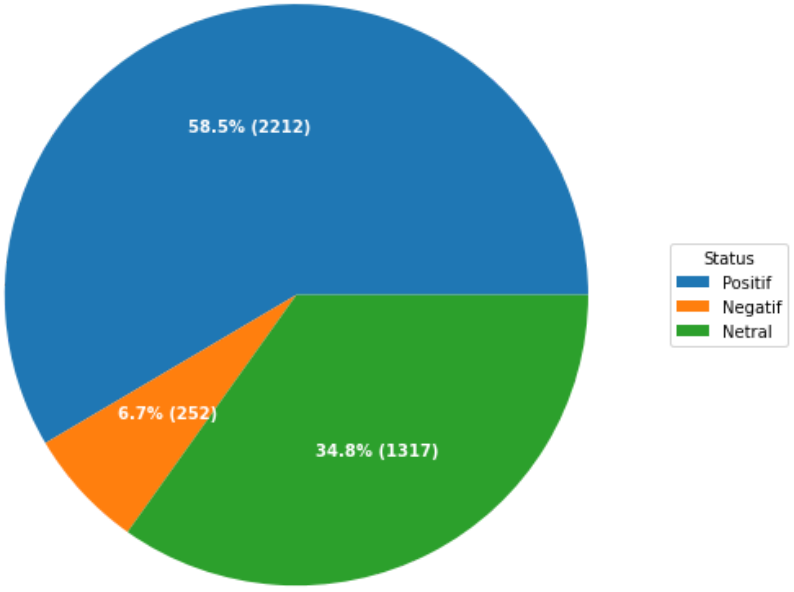
Sehingga hasil dari modelling wordcloud tersebut adalah :

Sehingga dapat kita lihat banyak kata-kata positif yang dihasilkan dari pemodelan *word cloud* yang telah kita lakukan.

Selanjutnya untuk menampilkan pie chart dari data yang telah kita klasifikasikan kita buat function show pie dan melakukan pengkodean untuk letak perdata serta posisi label sesuai data yang ada.

```
def show_pie(label, data, legend_title) :
    fig, ax = plt.subplots(figsize=(8, 10), subplot_kw=dict(aspect='equal'))

    labels = [x.split()[-1] for x in label]

    def func(pct, allvals):
        absolute = int(pct/100.*np.sum(allvals))
        return "{:.1f}% ({:d})".format(pct, absolute)

    wedges, texts, autotexts = ax.pie(data, autopct=lambda pct: func(pct, data),
                                      textprops=dict(color="w"))

    ax.legend(wedges, labels,
              title= legend_title,
              loc="center left",
              bbox_to_anchor=(1, 0, 0.5, 1))

    plt.setp(autotexts, size=10, weight="bold")
    plt.show()

label = ['Positif', 'Negatif', 'Netral']
count_data = [total_positif+1, total_negatif+1, total_netral]

show_pie(label, count_data, "Status")
```

Adapun hasil dari proses pemodelan chart pie tersebut adalah sebagai berikut:



Maka dari itu dari visualisasi data berbentuk chart pie dan dapat disimpulkan sementara bahwa rakyat Indonesia mayoritas menanggapi kebijakan pemerintah mengenai vaksin dengan positif.

7.5 Klasifikasi Data Dengan Metode Naive Bayes Classifier Dan Visualisasinya

Kami melakukan klasifikasi data dengan metode Naive Bayes Classifier menggunakan library Natural Language Toolkit atau disingkat NLTK. Langkah-langkah dalam melakukan klasifikasi data-nya yaitu :

1. Langkah pertama kami menyiapkan data apa saja yang akan kami proses untuk selanjutnya dilakukan pengklasifikasian data, data-data yang kami ambil adalah :

   - username : nama dari orang yang melakukan tweet (postingan)
   - tweetcreatedts : waktu tweet (postingan) dibuat
   - text : isi pesan tweet (postingan)
   - tweet_clean : isi pesan tweet yang sudah melalui proses pembersihan data

```
dataset = data.drop(['username',  'tweetcreatedts', 'text', 'tweet_clean'], axis=1, inplace=False)
dataset = [tuple(x) for x in dataset.to_records(index=False)]
```

2. Selanjutnya pada gambar dibawah ini kami mengambil dataset untuk kemudian dijadikan data training (data yang digunakan untuk melatih algoritma) dengan tujuan data training ini dapat digunakan untuk membantu melakukan prediksi. Pada langkah ini kami mengambil hasil dari metode pengklasifikasian sebelumnya untuk kemudian kami kelompokan ulang pada variable baru berdasarkan label nya dan kami ambil secara acak sampel dari masing masing label nya. Pada langkah ini kami mendapatkan hasil pada variable train_set dan merupakan data training kami yang diambil dari hasil proses data sebelumnya.

```
import random

set_positif = []
set_negatif = []
set_netral = []

for n in dataset:
    if(n[1] == 'Positif'):
      set_positif.append(n)
    elif(n[1] == 'Negatif'):
      set_negatif.append(n)
    else:
      set_netral.append(n)

set_positif = random.sample(set_positif, k=int(len(set_positif)/2))
set_negatif = random.sample(set_negatif, k=int(len(set_negatif)/2))
set_netral = random.sample(set_netral, k=int(len(set_netral)/2))

train = set_positif + set_negatif + set_netral

train_set = []

for n in train:
     train_set.append(n)
```

Setelah itu kami melakukan penghitungan akurasi dari variabel train_set menggunakan Naive Bayes Classifier dari library TextBlob. Hasil akurasinya sampai dengan nilai 0,93 atau 93% yang artinya hasil prediksinya sudah sangat baik.

```
from textblob.classifiers import NaiveBayesClassifier
cl = NaiveBayesClassifier(train_set)
print('Akurasi Test:', cl.accuracy(dataset))
```

```
Akurasi Test: 0.9272486772486772
```

3. Pada langkah ketiga kami melakukan pelabelan data ulang dengan menggunakan metode Naive Bayes Classifier. Kami mengambil data tweet_english yaitu data tweet_clean yang di translate ke bahasa inggris dan memasukkannya ke dalam variabel data_tweet.

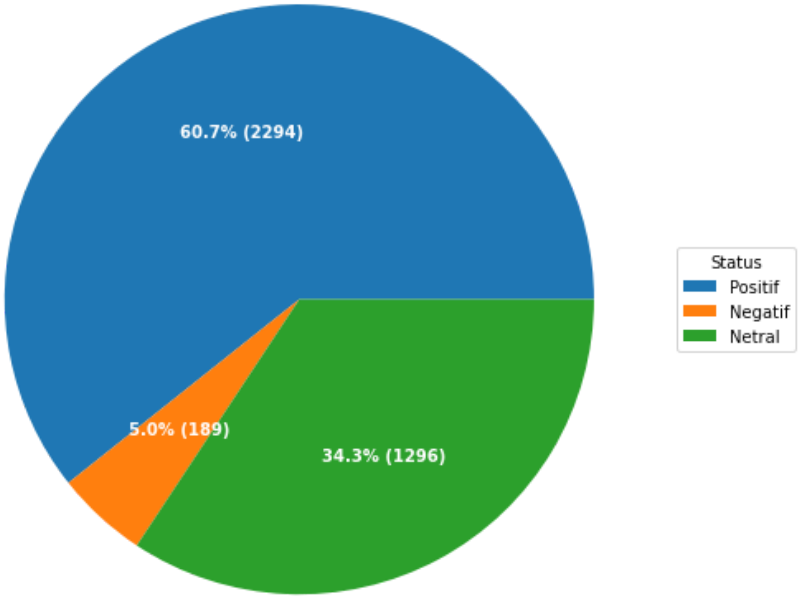
   Dalam proses perulangan kami melakukan analisis untuk mengklasifikasi data ke 3 kelas, yaitu positif , negatif, dan netral dan juga menghitung jumlah masing masing klasifikasinya serta total keseluruhan data yang diuji. Dan hasilnya terdapat 2294 data positif, 1297 data netral, dan 189 data negatif.

```
data_tweet = list(data['tweet_english'])
polaritas = 0

status = []
total_positif = total_negatif = total_netral = total = 0

for i, tweet in enumerate(data_tweet):
    analysis = TextBlob(tweet, classifier=cl)

    if analysis.classify() == 'Positif':
        total_positif += 1
    elif analysis.classify() == 'Netral':
        total_netral += 1
    else:
        total_negatif += 1

    status.append(analysis.classify())
    total += 1

print(f'\nHasil Analisis Data:\nPositif = {total_positif}\nNetral = {total_netral}\nNegatif = {total_negatif}')
print(f'\nTotal Data : {total}')
```

```
Hasil Analisis Data:
Positif = 2294
Netral = 1297
Negatif = 189

Total Data : 3780
```

```
status = pd.DataFrame({'klasifikasi_bayes': status})
data['klasifikasi_bayes'] = status
```

4. Terakhir kami melakukan visualisasi dari hasil data yang sudah kami klasifikasikan menggunakan metode Naive Bayes Classifier menggunakan fungsi chart pie yang telah dibuat sebelumnya. Kami menggunakan diagram lingkaran persen dengan warna biru mewakili data positif, warna orange mewakili data negatif, dan hijau mewakili warna netral.



8. Kesimpulan

Dengan Menggunakan Metode Naive Bayes Classifier Dengan Tingkat Akurasi 0.93 (93%) dan dilakukan pemilihan random data dari setiap klasifikasi. Adapun perbedaan dari klasifikasi sebelumnya adalah sebagai berikut :

- 2294 Sentimen Positif(+82 Data)
- 1297 Sentimen Netral (+20 Data)
- 189 Sentimen Negatif (-62 Data)

Sehingga dengan total tersebut dapat disimpulkan bahwa masyarakat mayoritas menanggapi kebijakan pemerintah mengenai vaksin dengan positif.

9. Analisis Klasifikasi Menggunakan Metode Naive Bayes Classifier

Setelah melakukan klasifikasi, kita dapat melakukan pengecekkan perubahan data yang dilakukan saat menggunakan metode Naive Bayes Classifier dengan membandingkan perbedaan pada kolom klasifikasi sebelumnya menggunakan kode berikut:

```
data_eval = [tuple(x) for x in data.to_records(index=False)]

for n in data_eval:
    if n[5] != n[6]:
       print(f'Text: {n[3]}\nClassifier: {n[5]}\nClassifier Bayes: {n[6]} \n')
```

Adapun beberapa contoh klasifikasi yang diubah yaitu :

Text:
#VaksinCovid19 #Jokowi Presiden Joko Widodo meminta masyarakat tak khawatir soal kehalalan vaksin Covid-19. Sebab, pemerintah telah melibatkan Majelis Ulama Indonesia (MUI) dan Kementerian Agama untuk memastikan kehalalan vaksin.
Classifier: Netral
Classifier Bayes: Positif

Text:
Kalau karena #Corona merasa tidak takut? Mengapa harus takut #Vaksin Padahal #vaksincorona adalah ikhtiar mengahadapi #Covid_19 #VaksinUntukNegeri #vaksinuntukkita
Classifier: Negatif
Classifier Bayes: Positif

Text:
Dengan alasan mencukupi pasokan, China akan impor vaksin Covid19 dari Jerman. Mengapa Indonesia tidak impor juga dari Jerman saja? Mengapa ke China? Apakah ini ladang uang kartel dan pejabat pemburu rente? #KedzalimanPastiTumbang #Vaksin #Sinovac
Classifier: Netral
Classifier Bayes: Negatif

Text:
#vaksingratis #covid19 #pandemi #jokowi #vaksin Jika Tidak Percaya Covid-19 dan Vaksin, Setidaknya Jangan Coba Tularkan
Classifier: Negatif
Classifier Bayes: Positif

Kesimpulan:
Dari data diatas dapat kita lihat data-data yang diubah sudah benar sentimennya sesuai isi teks sehingga dapat simpulkan metode Naive Bayes Classifier cukup akurat dalam melakukan klasifikasi dalam sentimen analisis ada projek ini.

10. Daftar Pustaka

https://www.geeksforgeeks.org/naive-bayes-classifiers

Pemerintah Gratiskan Vaksin Covid-19, Ini Panduan Penerima Vaksin Pfizer-BioNTech (kompas.com)

Indonesia: WHO Coronavirus Disease (COVID-19) Dashboard | WHO Coronavirus Disease (COVID-19) Dashboard

Brata Mas Pintok, Kemas Muslim L. 2018. Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naïve Bayes Classifier.

Sigit, Ema, Emha Taufiq Luthfi. 2018. KLASIFIKASI SENTIMEN PADA TWITTER DENGAN NAIVE BAYES CLASSIFIER.