**LAPORAN PROYEK PENGOLAHAN DATA BESAR**

**Sentiment Analisis pada Streaming Data Twitter untuk Review Produk Miniso**

****

**Disusun oleh:**

**12S18002 – Wiranda Siahaan**

**12S18009 – Novita Enjelia Hutapea**

**12S18041 – Merika H. Manurung**

**12S18046 – Tiara Octavia Situmorang**

**PROGRAM STUDI SARJANA SISTEM INFORMASI FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO INSTITUT TEKNOLOGI DEL**

**MEI 2022**

# 

# 

# **DAFTAR ISI**

[DAFTAR ISI](#_5bc81qw2s2s0) 2

[DAFTAR GAMBAR](#_6aresk655x3t) 5

[DAFTAR TABEL](#_m5yudxp7z2rj) 6

[DAFTAR KODE PROGRAM](#_1v5ujz2g9ufh) 7

[**Bab 1**](#_ochp9exjzb6c) **8**

[**Pendahuluan**](#_yboy4kt0hikr) **8**

[1.1 Latar Belakang](#_i0enmbuigthp) 8

[1.2 Rumusan Masalah](#_uzwe9fk707qq) 9

[1.3 Tujuan](#_fryoi1j5rszt) 9

[1.4 Manfaat](#_s6adzkc3bn4t) 9

[1.5 Ruang Lingkup](#_yynf06pmsrum) 9

[**Bab 2**](#_wtorcs8keng2) **9**

[**Solusi**](#_4jye34aag9kk) **10**

[2.1 Big Data Arsitektur](#_bvsizc9zkzwq) 10

[2.1.1 Data Source](#_xc9h8vje48dt) 10

[2.1.2 Data Preparation](#_rvor7psrw7vx) 10

[2.1.3 Real time data processing](#_qvw81wcba4bi) 11

[2.1.4 Process analysis](#_7o366effgzo8) 11

[2.5 Model Training](#_xw5oai75rmye) 11

[2.1.6 Evaluation](#_vazzxgnvk846) 11

[**Bab 3**](#_xt52j82u6qx6) **12**

[**Implementasi**](#_dljxgtb09yx1) **12**

[3.1 Consume twitter streaming data](#_vd3jtglbvi88) 12

[3.2 Data Preparation](#_loykbehxn05x) 15

[3.3 Text Processing](#_5i29h0z2761i) 17

[3.4 Hasil akhir](#_y92u4zrpbl0d) 23

[3.5 Model](#_2eabqorugb68) 26

[**Bab 4**](#_vkeils1ffuqb) **31**

[**Hasil dan Pembahasan**](#_mschq0jibc52) **32**

[4.1 Hasil Prediksi](#_ftn4hhm6kptc) 32

[4.2 Hasil Visualisasi](#_vgjpde8qtgb9) 32

[4.3 Kesimpulan](#_tflianhrgl3h) 33

[**DAFTAR PUSTAKA**](#_562aieh8622i) **34**

# **DAFTAR GAMBAR**

[Gambar 1. Big Data Arsitektur](#_wq12zgbbthrn) 11

[Gambar 2. Tampilan 5 sampel data](#_1jubzmu50dxc) 16

[Gambar 3. Sampel data 5 baris teratas](#_9xm2zbfecigt) 17

[Gambar 4. Output dari hasil drop kolom](#_3gefwx36v2gj) 18

[Gambar 5. Output berupa tampilan data](#_3ortd15mctfj) 26

[Gambar 6. Ouput berupa berisi subjectivity dan polarity](#_wue1dprrzajv) 27

[Gambar 7. Tampilan nilai teratas](#_g6b79nug43sz) 28

[Gambar 8. Tampilan hasil perhitungan vektorisasi](#_kaa2s6gkrszm) 30

[Gambar 9. Tampilan output dengan nilai future name](#_fjtvp2ysvam1) 31

[Gambar 10. Tampilam kembalian nilai subjectivity dan polaritas](#_5fq2ln9jlwpj) 33

# 

# **DAFTAR KODE PROGRAM**

[Kode program 1. Consume twitter streaming data](#_n5k7i7cpgqzs) 13

[Kode program 2. Import library](#_whwtvhlm3fgf) 13

[Kode program 3. Credential API](#_342005m2h3qh) 14

[Kode program 4. Pembuatan objek otentikasi](#_cur20b4onjur) 14

[Kode program 5. mencari tweet mengandung kata miniso](#_we755az656z9) 14

[Kode program 6. Mengubah data tweet kedalam dataframe](#_ujjnx5av5ixi) 15

[Kode program 7. Stream data](#_bhh6e2z6qz1j) 15

[Kode program 8. Stream data miniso](#_qnh437mts7aq) 15

[Kode program 9. Menampilkan 5 sampel data](#_5olpt9eg7e1c) 16

[Kode program 10. Split data test data train](#_1ogn8nhn1x84) 16

[Kode program 11. Split data kedalam dependen dan independen atribut](#_onns42jr6si9) 17

[Kode program 12. Menampilkan data lima baris teratas](#_qxerlc8vlssh) 17

[Kode program 13. Drop kolom](#_mzpowks8t912) 17

[Kode program 14. Menghapus missing value](#_yaljszn9huc) 18

[Kode program 15. Mengimport semua pustaka](#_chm0ab6hu62m) 19

[Kode program 16. Case folding](#_dfzkonjqfbdx) 19

[Kode program 17. Remove URL](#_bycezm4m041g) 20

[Kode program 18. Remove number](#_arkbqzbadkwp) 20

[Kode program 19. Remove Punctuation](#_z0igmywg5u7r) 22

[Kode program 20. Stop Removal](#_p3s1dboidf8q) 22

[Kode program 21. Stopwords](#_al14dznehh4d) 23

[Kode program 22. Tokenisasi](#_4gjqyv3k4knt) 24

[Kode program 23. final data setelah preparation](#_x6u64wr2erkn) 24

[Kode program 24. Tampilan final data](#_p7qk393y1auj) 24

[Kode program 25. Mengakses dataset](#_m38o2bt8dgjr) 25

[Kode program 26. Membaca data menggunakan library pandas](#_1ynhid5vyinb) 25

[Kode program 27. Penggunaan textblob](#_b25elo4ae0al) 26

[Kode program 28. Mengubah penamaan data final](#_vi3gluaovi23) 26

[Kode program 29. Memanggil data dan mendapatkan nilai subjectivity dan polarity](#_dc49sk18fntc) 26

[Kode program 30. Menghitung nilai sentimen secara keseluruhan](#_14ujnxgco8sr) 27

[Kode program 31. Pemodelan dengan algoritma Naive Bayes](#_rjivz7cpx8tw) 28

[Kode program 32. Membaca dataset berupa dataframe](#_3r46p07rbrgz) 28

[Kode program 33. Menampilkan lima nilai teratas](#_f9lewb3b3i4i) 28

[Kode program 34. split data train dan test](#_e1hgee6vzy2o) 29

[Kode program 35. Menampilkan jumlah data train](#_jci70yjh4zz7) 29

[Kode program 36. Menampilkan jumlah data test](#_6aexmngbt822) 29

[Kode program 37. Fungsi vektorisasi pada data train](#_rr43wboao17q) 29

[Kode program 38. Perintah menghitung vektorisasi](#_axm73ggcan2x) 30

[Kode program 39. Perintah mengembalikan nilai future name](#_2l7murwel9d5) 30

[Kode program 40. Klasifikasi dengan algoritma Naive Bayes](#_rp5nzwljcxlw) 31

[Kode program 41. Menampilkan nilai akurasi, presisi dan recall](#_flsdvq309ayi) 32

# 

# **Bab 1**

# **Pendahuluan**

## **1.1 Latar Belakang**

Twitter sebagai salah satu media sosial yang cukup populer digemari sebagai salah satu media penyebar informasi yang sangat cepat. Informasi yang beredar juga sangat banyak mulai dari berita, opini, komentar, kritik dan semuanya ada yang bersifat positif, negatif dan netral. Menurut data yang dilansir Statista berdasarkan hasil penelitian PeerReach, Indonesia termasuk pengguna twitter yang aktif ke 3 di dunia dibawah Amerika Serikat dan Jepang. Pada Twitter terdapat istilah bernama tweet yang merupakan sebuah pesan atau status yang dibuat oleh penggunanya. Sebuah tweet dapat mengekspresikan sebuah perasaan atau keadaan dari pengguna Twitter. Tweet dapat mengandung sebuah opini dari penggunanya terhadap kejadian yang dialaminya. Opini tersebut dapat dimanfaatkan sebagai penilaian bagi suatu objek.

Twitter dapat diakses melalui web atau perangkat seluler. Untuk berbagi informasi di Twitter seluas mungkin, Twitter juga menyediakan akses terprogram ke data Twitter kepada perusahaan, pengembang, dan pengguna melalui API (antarmuka pemrograman aplikasi). Pada tingkat tinggi, API adalah cara program komputer "berbicara" satu sama lain sehingga mereka dapat meminta dan mengirimkan informasi. Hal ini dilakukan dengan mengizinkan aplikasi perangkat lunak untuk memanggil apa yang dikenal sebagai titik akhir: alamat yang sesuai dengan jenis informasi tertentu yang diberikan. Twitter mengizinkan akses ke bagian layanan kami melalui API untuk memungkinkan orang membangun perangkat lunak yang terintegrasi dengan Twitter, seperti solusi yang membantu perusahaan menanggapi umpan balik pelanggan di Twitter.

Platform API Twitter menyediakan akses luas ke data Twitter publik yang telah dipilih pengguna untuk dibagikan kepada dunia. Twitter juga mendukung API yang memungkinkan pengguna untuk mengelola informasi Twitter non-publik mereka sendiri (mis., Direct Message) dan memberikan informasi ini kepada pengembang yang telah mereka izinkan untuk melakukannya

Analisis sentimen merupakan bagian dari Natural Language Processing (NLP) dan Machine Learning. Cara kerjanya adalah dengan melakukan klasifikasi terhadap opini positif dan negatif. Analisis sentimen menganalisa pendapat orang, sentimen, evaluasi, penilaian, sikap, dan emosi terhadap entitas seperti produk, jasa, organisasi, individu, masalah, peristiwa atau suatu topik. Sentimen analisis ini bertujuan untuk menentukan sentimen publik terhadap objek tertentu yang disampaikan melalui twitter dengan bahasa indonesia yang dapat membantu usaha untuk melihat opini publik terkait dengan objek tersebut sebagai bahan pertimbangan pengambilan keputusan.

Berdasarkan pemaparan yang telah diuraikan, penelitian ini bermaksud untuk membangun analisis sentimen pada twitter dengan objek review yang diberikan pada produk Miniso.

## **1.2 Rumusan Masalah**

1. Bagaimana pengaruh algoritma terhadap proses analisis sentimen untuk review yang diberikan pada produk Miniso
2. Bagaimana proses tweet preprocessing dan sentiment analysis.
3. Bagaimana Melakukan Klasifikasi pada data Twitter yang diperoleh.

## **1.3 Tujuan**

1. Mengetahui pengaruh algoritma terhadap proses analisis sentimen untuk review yang diberikan pada produk Miniso
2. Melakukan tweet preprocessing dan sentiment analysis.
3. Melakukan Klasifikasi pada data Twitter yang diperoleh.

## **1.4 Manfaat**

Manfaat dari pengerjaan proyek :

Proyek ini diharapkan nantinya dapat memberikan pemahaman dan keterampilan bagi tim proyek untuk mengimplementasikan teknik machine learning (scalable machine learning) pada data besar khususnya pada salah satu metode dari pengklasifikasian yaitu Logistic Regression dengan teknologi pengolahan data besar menggunakan Spark Mlib API untuk memproses dan menganalisis data besar.

## **1.5 Ruang Lingkup**

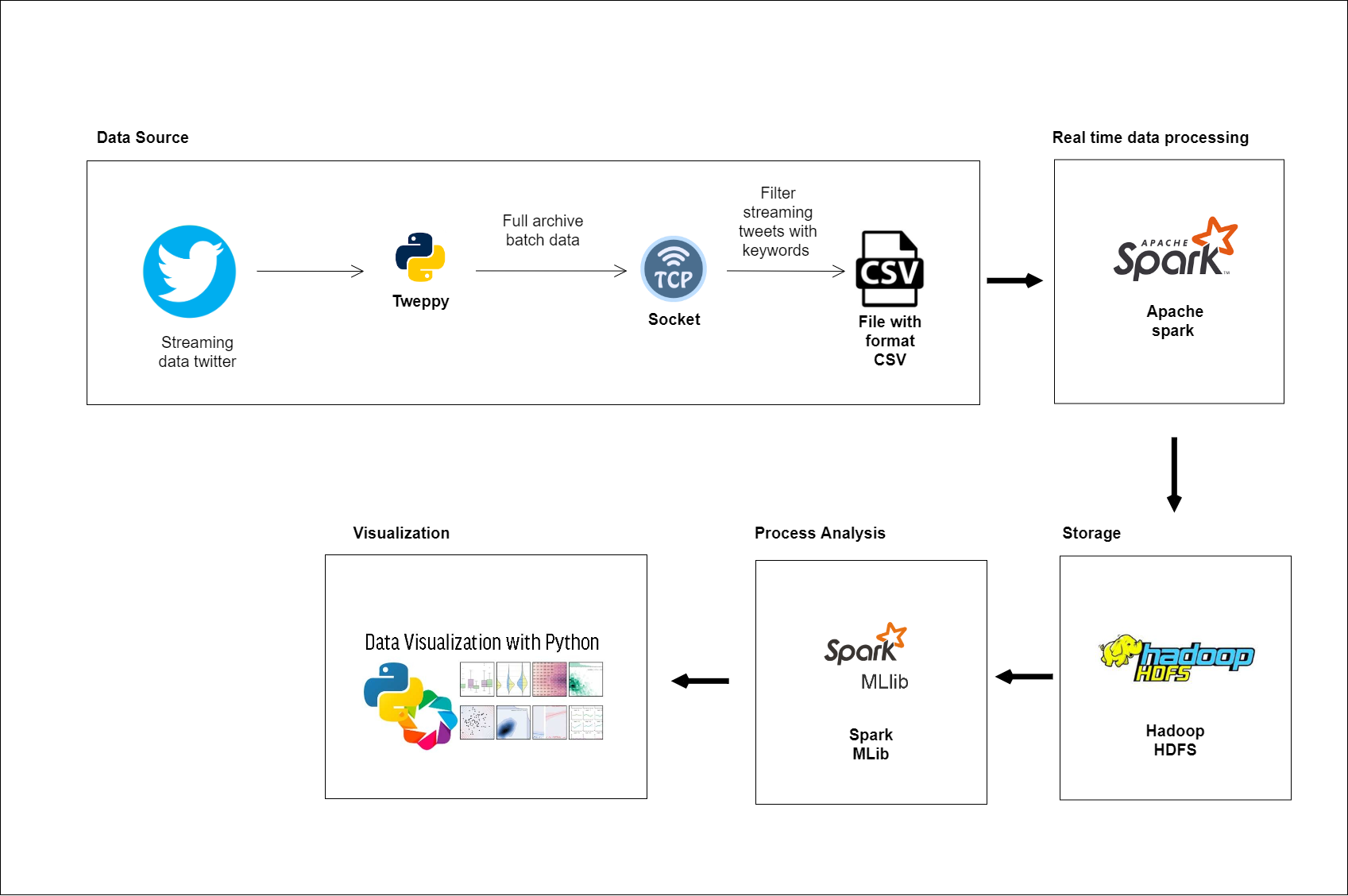
Ruang lingkup dalam pengerjaan proyek ini adalah

1. Melakukan pengolahan data menggunakan salah satu metode yang ada dari *Classification.*
2. Penelitian ini hanya didasarkan pada tanggapan yang diberikan pada platform Twitter tentang review yang diberikan pada produk Miniso.

# **Bab 2**

# **Solusi**

## **2.1 Big Data Arsitektur**



### Gambar 1. Big Data Arsitektur

### 2.1.1 Data Source

Data diambil menggunakan *developer credentials* untuk dapat melakukan autentikasi, dan terhubung dengan API dari twitter. Dengan memanfaatkan pustaka *tweepy* data akan diekstrak dari versi API dengan arsip lengkap, yang kemudian akan di filter sesuai dengan *keyword* yang akan digunakan pada pemodelan, data yang akan diberikan disimpan pada file dengan format CSV. Pada proses pengambilan data, penggunaan *socket* TCP antara *tweepy* dengan spark dimana *socket* TCP akan membantu proses pengiriman tweet ke proses streaming spark.

### 2.1.2 Data Preparation

Data preparation merupakan langkah yang dilakukan untuk mengubah data mentah menjadi sebuah data yang berkualitas untuk nantinya dapat membantu memberikan hasil keputusan, maupun integrasi data yang baik dan konsisten. Hal ini dilakukan karena adanya kemungkinan data *incomplete* yaitu data dengan karakteristik nilai atribut yang kurang maupun mengandung data agregat, selain itu ada juga kemungkinan data bersifat *noisy* yaitu data yang masih mengandung error hingga data *inconsistent*. Beberapa langkah eksplorasi data juga dilakukan untuk mengetahui model ataupun algoritma yang dapat digunakan dalam memproses data.

### 2.1.3 Real time data processing

Dalam pengolahan data real time atau data yang mampu mengambil nilai inputan yang berubah dengan cepat dan memberikan output yang hampir berubah dari waktu ke waktu, proses nya akan menggunakan apache spark sehingga data kemudian akan disimpan pada hadoop HDFS atau Hadoop Distributed File System dalam bentuk database.

### 2.1.4 Process analysis

kemudian dilanjutkan dengan proses analisis menggunakan Machine Learning dengan Spark MLib yang mendukung berbagai penggunaan algoritma.

### 2.5 Model Training

Pada tahapan inilah pembangunan model algoritma akan dilakukan dan dilanjutkan dengan melatih algoritma dengan *train data.* Dalam hal ini algoritma digunakan untuk melakukan klasifikasi yakni dengan Naive Bayes. Adapun dataset yang digunakan adalah dataset yang memiliki tiga kelas yakni sentimen positif, negatif dan netral

### 2.1.6 Evaluation

Setelah dilakukannya pembangunan model dengan menggunakan train data, maka dilanjutkan evaluasi untuk melihat nilai akurasi dan performa dari hasil klasifikasi. Adapun proses evaluasi dilakukan dengan menggunakan test data.

# 

# **Bab 3**

# **Implementasi**

## **3.1 Consume twitter streaming data**

|  |
| --- |
| from google.colab import drive  drive.mount("/content/gdrive") |

### Kode program 1. Consume twitter streaming data

output:

|  |
| --- |
| Mounted at /content/gdrive |

* import library

Tahapan ini adalah proses meng-import dan deklarasi pustaka pustaka yang diperlukan dalam melakukan pengambilan data dari twitter streaming data. Tweepy merupakan library python yang digunakan untuk mengakses data twitter melalui API (Application Programming Interface). Para pengembang biasanya menggunakan pustaka ini dalam merancang dan menyempurnakan aplikasi yang membutuhkan data dari twitter didalamnya. Selanjutnya adalah meng-import tweepy

|  |
| --- |
| import tweepy  from tweepy import OAuthHandler  from tweepy import Stream  import socket  import json  import numpy as np  import pandas as pd |

### Kode program 2. Import library

* Credential API

Untuk mendapatkan izin otentikasi ke API Twitter, perlu dituliskan kode akses nya terlebih dahulu. Kode akses ini bisa didapatkan atau direquest dari link dibawah ini:

https://developer.twitter.com/en/products/twitter-api

|  |
| --- |
| #Kindly put your credentials received from twitter developer account  consumer\_key='2mTmxHwkZramjVDOewE2dybgQ'  consumer\_secret='dJ42E93prTAze5BoF4wAIcxYA6gXfu9rGrHr6gguUt85xcHJSi'  access\_token ='1517128047019040768-YMZCcFIf6VLbZXNkJsa79sZk70xFkM'  access\_secret='Auwa9WPIktjUCrgQmzceZsdMtMA2hbnebSR7QlmYEdrlq' |

### Kode program 3. Credential API

Script berikut digunakan untuk melakukan pembuatan objek otentikasi dengan memasukkan kode token

|  |
| --- |
| auth = tweepy.OAuthHandler(consumer\_key, consumer\_secret)  auth.set\_access\_token(access\_token, access\_secret)  api = tweepy.API(auth) |

### Kode program 4. Pembuatan objek otentikasi

Kemudian dari API twitter yang tersedia, kemudian cari tweet yang mengandung kata “miniso” berbahasa inggris

|  |
| --- |
| for tweet in api.search(q='miniso', lan='en'):  print(tweet.text) |

### Kode program 5. mencari tweet mengandung kata miniso

output:

|  |
| --- |
| @convomfs 3 ini. Tp yang British rose jarang pake.  Yang miniso body mist sih. Gue bukan ahli parfum jadi semuanya s…<https://t.co/slRCTCaRWj>  @Pranaa\_\_ awet gk si parfumnya miniso?  RT @FergieGo: Me encontré a la @JenySweet1 en el Miniso.<https://t.co/f8KSXfhqw3>  miniso has these and theyre so cute 😫<https://t.co/Wr0jqTA4yB>  @convomfs Original Source Vanilla Raspberry, Zwitsal, Miniso Red Fruits &amp; Musk, Heaven Scent yg Black Amber sama Wo…<https://t.co/T0Vd26tkry>  Miniso turned into a gym and he was doing weights infront of me🥲  @tanyakanrl Beli sendok sama sumpit di miniso  @capriesed baru sembuh kemaren abis sakit huhuhu, wah parfum miniso emg lumayan enak2 ya wanginyaa  @OlixYT Yes but I was too shy too ask 😅 it's okay if no for me I appreciate it  Ir a miniso o el mejor parche del mundo?  eriask! Wta figure exo miniso sehun ridi ina harga nya brp ya?  ˗ˏˋ 50฿ รวมส่ง กระเป๋าสะพายข้างเอลโม่ซื้อมาจาก miniso ´ˎ˗  #ส่งต่อกระเป๋า #ส่งต่อ #กระเป๋า #กระเป๋าสะพาย…<https://t.co/1AfbCTbLOA>  @miniso\_ph kailan niyo balak magbenta ng plushie ni anya?? 😩  MINISO Corporation has filed for FCC ID approval on K-346 IPX4 Waterproof Speaker with Suction Cup <https://t.co/aZhSHtmEF3>  Necesito ir a miniso por un ventilador y por cosas que necesito, pero en realidad no necesito jajaja |

* mengubah data tweet kedalam dataframe

|  |
| --- |
| tweetDf = pd.DataFrame(columns = ['User', 'User\_statuses\_count',  'user\_followers','fav\_count','User\_location','Tweets',]) |

### Kode program 6. Mengubah data tweet kedalam dataframe

* Stream Data

Kemudian dilanjutkan dengan stream data stream

|  |
| --- |
| def stream(data, file\_name):  i = 0  for tweet in tweepy.Cursor(api.search, q=data, count=100000, lang='en').items():  print(i, end='\r')  tweetDf.loc[i, 'User'] = tweet.user.name  tweetDf.loc[i, 'User\_statuses\_count'] = tweet.user.statuses\_count  tweetDf.loc[i, 'user\_followers'] = tweet.user.followers\_count  tweetDf.loc[i, 'fav\_count'] = tweet.favorite\_count  tweetDf.loc[i, 'User\_location'] = tweet.user.location  tweetDf.loc[i, 'Tweets'] = tweet.text      tweetDf.to\_csv('{}.csv'.format(file\_name))  i+=1  if i == 200:  break  else:  pass |

### Kode program 7. Stream data

|  |
| --- |
| stream(data = ['miniso'], file\_name = 'miniso\_2021') |

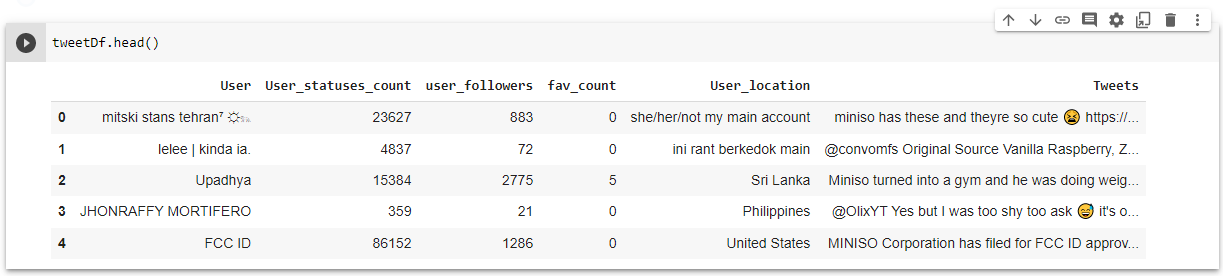
### Kode program 8. Stream data miniso

|  |
| --- |
| tweetDf.head() |

* Menampilkan 5 sampel data

|  |
| --- |
| tweetDf.head() |

### Kode program 9. Menampilkan 5 sampel data



### Gambar 2. Tampilan 5 sampel data

## **3.2 Data Preparation**

* Import library

Dalam proses data preparation, diperlukan beberapa library untuk membantu memproses data, diantaranya:

1. Numpy digunakan dalam membantu melakukan *handling numbers* untuk memudahkan perhitungan multidimensi maupun perhitungan matriks
2. Pandas digunakan dalam membantu handling dataset, dengan disediakannya struktur data dasar dari library tersebut.
3. Simple Imputer digunakan untuk melakukan handling missing data.
4. Label Encoder, One Hot Encoder digunakan untuk membantu melakukan pelabelan data kedalam beberapa kategori, menormalisasi label dan juga membantu melakukan transformasi yang bukan numerik label.
5. Train\_test\_split, library ini digunakan untuk melakukan pemisahan rangkaian data acak ke dalam data train dan test dengan
6. Standard Scaler

|  |
| --- |
| import numpy as np # used for handling numbers  import pandas as pd # used for handling the dataset  from sklearn.impute import SimpleImputer # used for handling missing data  from sklearn.preprocessing import LabelEncoder, OneHotEncoder # used for encoding categorical data  from sklearn.model\_selection import train\_test\_split # used for splitting training and testing data  from sklearn.preprocessing import StandardScaler # used for feature scaling |

### Kode program 10. Split data test data train

* split atribut data ke dalam independent dan dependent attribute

Dataset yang telah disimpan dengan penamaan miniso\_2021 dengan format csv, di import ke dalam sebuah variabel dengan cara melakukan split atribut dependent dan independent

|  |
| --- |
| dataset = pd.read\_csv('miniso\_2021.csv') # to import the dataset into a variable  # Splitting the attributes into independent and dependent attributes  X = dataset.iloc[:, :-1].values # attributes to determine dependent variable / Class  Y = dataset.iloc[:, -1].values # dependent variable / Class |

### Kode program 11. Split data kedalam dependen dan independen atribut

Berikut perintah yang dilakukan untuk menampilkan data lima baris teratas.

|  |
| --- |
| dataset.head() |

### Kode program 12. Menampilkan data lima baris teratas

dan berikut adalah sampel data yang ditampilkan, dimana data berupa nama pengguna, jumlah pengikut, jumlah tweet disukai, lokasi dari pengguna, dan tweets yang di publish oleh pengguna.

|  |
| --- |
|  |

### Gambar 3. Sampel data 5 baris teratas

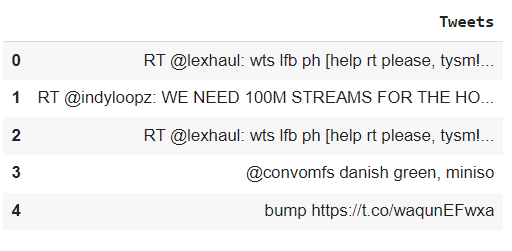
* melakukan **drop column**

mengetahui bahwa tidak semua kolom data akan diproses lebih lanjut, maka akan dilakukan drop kolom untuk atribut unnamed, user, user\_statuses\_count, user followers, fav\_count, dan user\_location dan nantinya hanya akan tersisa kolom tweet saja.

|  |
| --- |
| to\_drop = ['Unnamed: 0', 'User', 'User\_statuses\_count', 'user\_followers', 'fav\_count', 'User\_location'] |

### Kode program 13. Drop kolom

|  |
| --- |
| dataset.drop(to\_drop, inplace=True, axis = 1) |



### Gambar 4. Output dari hasil drop kolom

* Menghapus **missing value**

untuk data tersebut pula, dilakukan penghapusan missing value, menggunakan library yang disediakan oleh numpy dengan mengisi nilai missing value dengan nilai rata rata.

|  |
| --- |
| # handling the missing data and replace missing values with nan from numpy and replace with mean of all the other values  dataset1 = dataset.dropna()  print(dataset1.shape) |

### Kode program 14. Menghapus missing value

output:

|  |
| --- |
| (200, 1) |

## **3.3 Text Processing**

Mengimport semua pustaka yang digunakan dalam melakukan pemrosesan terhadap teks.

|  |
| --- |
| import re, string, unicodedata  import json  import nltk  import json  import time  import itertools  import numpy as np  import pandas as pd  import tensorflow as tf  import tensorflow\_hub as hub  import matplotlib.pyplot as plt  from nltk.corpus import stopwords  from nltk.stem import WordNetLemmatizer  from nltk.tokenize import word\_tokenize  from wordcloud import WordCloud, STOPWORDS  import tensorflow.keras.preprocessing.text as kpt  from sklearn.model\_selection import train\_test\_split  from tensorflow.keras.preprocessing.text import Tokenizer |

### Kode program 15. Mengimport semua pustaka

**Case folding**

Case folding merupakan salah satu bentuk text preprocessing yang berfungsi untuk menyetarakan seluruh karakter ataupun huruf dalam data menjadi huruf kecil.

|  |
| --- |
| # Change text column to lower lower case  dataset1['Tweets'] = dataset1['Tweets'].str.lower()  # Look at the change  print(dataset1) |

### Kode program 16. Case folding

output:

|  |
| --- |
| Tweets  0 miniso has these and theyre so cute 😫 [https://](https://colab.research.google.com/drive/1oC_bDVSSBCIICXuQF2EGLK9fFTF-76RY)...  1 @convomfs original source vanilla raspberry, z...  2 miniso turned into a gym and he was doing weig...  3 @olixyt yes but i was too shy too ask 😅 it's o...  4 miniso corporation has filed for fcc id approv...  .. ...  195 rt @sugarsanriio: hello kitty cooler bag from ...  196 rt @kyvo\_gvr: wts | lfb [help rt] \n\n𓂋 xiumin...  197 rt @kyvo\_gvr: wts | lfb [help rt] \n\n𓂋 xiumin...  198 @wana\_\_\_\_ miniso taught me shege that’s why lool  199 rt @kyvo\_gvr: wts | lfb [help rt] \n\n𓂋 xiumin...  [200 rows x 1 columns] |

**Remove URL**

Remove url dalam text preprocessing merupakan tindakan yang bertujuan untuk menghapus semua teks berbentuk link url dalam data.

|  |
| --- |
| def clean\_data(dataset2):  #replace URL of a text  dataset1['Tweets'] = dataset2['Tweets'].str.replace('http[s]?://(?:[a-zA-Z]|[0-9]|[$-\_@.&+]|[!\*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', ' ')  clean\_data(dataset1)  print(dataset1['Tweets']); |

### Kode program 17. Remove URL

output:

|  |
| --- |
| 0 miniso has these and theyre so cute 😫  1 @convomfs original source vanilla raspberry, z...  2 miniso turned into a gym and he was doing weig...  3 @olixyt yes but i was too shy too ask 😅 it's o...  4 miniso corporation has filed for fcc id approv...  ...  195 rt @sugarsanriio: hello kitty cooler bag from ...  196 rt @kyvo\_gvr: wts | lfb [help rt] \n\n𓂋 xiumin...  197 rt @kyvo\_gvr: wts | lfb [help rt] \n\n𓂋 xiumin...  198 @wana\_\_\_\_ miniso taught me shege that’s why lool  199 rt @kyvo\_gvr: wts | lfb [help rt] \n\n𓂋 xiumin...  Name: Tweets, Length: 200, dtype: object  /usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:3: FutureWarning: The default value of regex will change from True to False in a future version.  This is separate from the ipykernel package so we can avoid doing imports until |

**Remove Number**

Remove number bertujuan untuk menghapus semua data tweet yang masih mengandung angka.

|  |
| --- |
| # removing numbers from strings of speciafied  # column, here 'Name'  dataset1['Tweets']= dataset1['Tweets'].str.replace('\d+', '')    # display output with numbers removed from  # required strings  print(dataset1) |

### Kode program 18. Remove number

output:

|  |
| --- |
| Tweets  0 miniso has these and theyre so cute 😫  1 @convomfs original source vanilla raspberry, z...  2 miniso turned into a gym and he was doing weig...  3 @olixyt yes but i was too shy too ask 😅 it's o...  4 miniso corporation has filed for fcc id approv...  .. ...  195 rt @sugarsanriio: hello kitty cooler bag from ...  196 rt @kyvo\_gvr: wts | lfb [help rt] \n\n𓂋 xiumin...  197 rt @kyvo\_gvr: wts | lfb [help rt] \n\n𓂋 xiumin...  198 @wana\_\_\_\_ miniso taught me shege that’s why lool  199 rt @kyvo\_gvr: wts | lfb [help rt] \n\n𓂋 xiumin...  [200 rows x 1 columns]  /usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:3: FutureWarning: The default value of regex will change from True to False in a future version.  This is separate from the ipykernel package so we can avoid doing imports until |

**Remove punctuation**

Remove punctuation berfungsi untuk menghapus semua tanda baca dalam teks, dimana tanda baca juga dianggap sebagai sebuah delimiter ataupun batas pemisah.

|  |
| --- |
| dataset1['Tweets'] = dataset1['Tweets'].str.replace(r'[^\w\s]','') |

|  |
| --- |
| def clean\_text(text):  '''Make text lowercase, remove text in square brackets,remove links,remove punctuation  and remove words containing numbers.'''  text = str(text).lower()  text = re.sub('\[.\*?\]', '', text)  text = re.sub('https?://\S+|www\.\S+', '', text)  text = re.sub('<.\*?>+', '', text)  text = re.sub('[%s]' % re.escape(string.punctuation), '', text)  text = re.sub('\n', '', text)  text = re.sub('\w\*\d\w\*', '', text)  text = re.sub(r"\d+", "", text)  text = re.sub(r'[0-9]+', '', text)  text = re.sub(r'\s\*\d+\s\*', '', text)  text = re.sub('/[^\w\s]/gi', '', text)  text = re.sub("@[^0-9a-zA-Z]+", '',text)  text = re.sub(r'@[^\s]+', ' ', text)  text = re.sub(r'#[^\w]+', ' ', text)  text = re.sub(r"[.,:;+!\-\_<^/=?&|$\"\(\)\d\\*]", " ", text)  text = re.sub(' +', ' ', text)  text = re.sub('\n+',' ',text)  text = re.sub('\n', text)  text = re.sub('\ ■',' ',text)  text = re.sub(r'&[^\s]+', ' ', text)  return text |

|  |
| --- |
| clean\_data(dataset1) |

### Kode program 19. Remove Punctuation

**Stopword Removal**

* Import Library

|  |
| --- |
| import nltk  nltk.download('stopwords')  print(stopwords.words('english')) |

### Kode program 20. Stop Removal

output:

|  |
| --- |
| [nltk\_data] Downloading package stopwords to /root/nltk\_data...  [nltk\_data] Unzipping corpora/stopwords.zip.  ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"] |

|  |
| --- |
| print(dataset1['Tweets']);  import nltk  nltk.download('stopwords')  nltk.download('punkt')  stop\_words = set(stopwords.words('english'))    dataset1['Tweets'] = dataset1['Tweets'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop\_words)]))  print(dataset1['Tweets']) |

### Kode program 21. Stopwords

output:

|  |
| --- |
| 0 miniso has these and theyre so cute  1 convomfs original source vanilla raspberry zwi...  2 miniso turned into a gym and he was doing weig...  3 olixyt yes but i was too shy too ask its okay...  4 miniso corporation has filed for fcc id approv...  ...  195 rt sugarsanriio hello kitty cooler bag from mi...  196 rt kyvo\_gvr wts lfb help rt \n\n𓂋 xiumin exop...  197 rt kyvo\_gvr wts lfb help rt \n\n𓂋 xiumin exop...  198 wana\_\_\_\_ miniso taught me shege thats why lool  199 rt kyvo\_gvr wts lfb help rt \n\n𓂋 xiumin exop...  Name: Tweets, Length: 200, dtype: object  [nltk\_data] Downloading package stopwords to /root/nltk\_data...  [nltk\_data] Package stopwords is already up-to-date!  [nltk\_data] Downloading package punkt to /root/nltk\_data...  [nltk\_data] Unzipping tokenizers/punkt.zip.  0 miniso theyre cute  1 convomfs original source vanilla raspberry zwi...  2 miniso turned gym weights infront  3 olixyt yes shy ask okay appreciate  4 miniso corporation filed fcc id approval k ipx...  ...  195 rt sugarsanriio hello kitty cooler bag miniso lt  196 rt kyvo\_gvr wts lfb help rt 𓂋 xiumin exoplanet...  197 rt kyvo\_gvr wts lfb help rt 𓂋 xiumin exoplanet...  198 wana\_\_\_\_ miniso taught shege thats lool  199 rt kyvo\_gvr wts lfb help rt 𓂋 xiumin exoplanet...  Name: Tweets, Length: 200, dtype: object |

**Tokenizing**

Sebelum memasuki tahap dimana data akan diklasifikasikan, data terlebih dahulu disegmentasikan menjadi kata kata atau disebut juga dengan proses tokenisasi. Yaitu sebuah proses yang akan melakukan pemotongan string berdasarkan inputan data, atau melakukan pemisahan kalimat menjadi kata.

|  |
| --- |
| dataset1['Tweets'] = dataset1['Tweets'].apply(word\_tokenize)  print(dataset1['Tweets']) |

### Kode program 22. Tokenisasi

output:

|  |
| --- |
| 0 [miniso, theyre, cute]  1 [convomfs, original, source, vanilla, raspberr...  2 [miniso, turned, gym, weights, infront]  3 [olixyt, yes, shy, ask, okay, appreciate]  4 [miniso, corporation, filed, fcc, id, approval...  ...  195 [rt, sugarsanriio, hello, kitty, cooler, bag, ...  196 [rt, kyvo\_gvr, wts, lfb, help, rt, 𓂋, xiumin, ...  197 [rt, kyvo\_gvr, wts, lfb, help, rt, 𓂋, xiumin, ...  198 [wana\_\_\_\_, miniso, taught, shege, thats, lool]  199 [rt, kyvo\_gvr, wts, lfb, help, rt, 𓂋, xiumin, ...  Name: Tweets, Length: 200, dtype: object |

* Info final data after preparation

|  |
| --- |
| tb\_counts = final\_data.TextBlob.value\_counts()  tb\_counts |

### Kode program 23. final data setelah preparation

Setelah melewati proses data preparation, berikut adalah tampilan data tweet yang akan diproses lebih lanjut.

output:

|  |
| --- |
| Tweets  0 [miniso, theyre, cute]  1 [convomfs, original, source, vanilla, raspberr...  2 [miniso, turned, gym, weights, infront]  3 [olixyt, yes, shy, ask, okay, appreciate]  4 [miniso, corporation, filed, fcc, id, approval...  .. ...  195 [rt, sugarsanriio, hello, kitty, cooler, bag, ...  196 [rt, kyvo\_gvr, wts, lfb, help, rt, 𓂋, xiumin, ...  197 [rt, kyvo\_gvr, wts, lfb, help, rt, 𓂋, xiumin, ...  198 [wana\_\_\_\_, miniso, taught, shege, thats, lool]  199 [rt, kyvo\_gvr, wts, lfb, help, rt, 𓂋, xiumin, ...  [200 rows x 1 columns] |

### Kode program 24. Tampilan final data

## **3.4 Hasil akhir**

dataset diakses pada konten drive dengan path yang telah diberikan.

|  |
| --- |
| dataset1.to\_csv('/content/gdrive/My Drive/pdb/preprocessing\_PDB.csv', index = False) |

### Kode program 25. Mengakses dataset

Sedangkan data akan dibaca menggunakan library pandas.

|  |
| --- |
| data = pd.read\_csv('/content/gdrive/MyDrive/pdb/preprocessing\_PDB.csv') |

### Kode program 26. Membaca data menggunakan library pandas

Pada proses sentiment analysis, untuk membantu penguraian data berbentuk teks dan melakukan klasifikasi, dilakukan pemanfaatan sebuah library yaitu TextBlob. Dimana library ini adalah salah satu library pada python yang digunakan untuk melakukan pemrosesan data tekstual, khususnya akan secara aktif menggunakan Natural language ToolKit. Hal tersebut juga akan memudahkan akses sumber daya lexical, dan membuat pemrosesan untuk klasifikasi sangat memungkinkan, penggunaan library ini juga secara umum membantu proses analisis terhadap operasi kompleks untuk data tekstual.

Dan pendekatan sentiment yang digunakan adalah, dengan memberikan orientasi dan identitas pada setiap kata pada kalimat dan kamus kata untuk melakukan klasifikasi kata negatif juga positif.

Pada kode program berikut diperlihatkan penggunaan library TextBlob yang akan mengembalikan nilai subjectivity dan polaritas dari sebuah kalimat. Dimana nilai polaritas ada pada rentang nilai -1 dan 1 yang mendefinisikan bahwa -1 adalah sentimen negatif dan 1 sebagai sentimen positif.

|  |
| --- |
| from textblob import TextBlob  def getSubjectivity(review):  return TextBlob(review).sentiment.subjectivity  def getPolarity(review):  return TextBlob(review).sentiment.polarity  def analyze(score):  if score < 0:  return '-1'  elif score == 0:  return '0'  else:  return '1' |

### Kode program 27. Penggunaan textblob

Proses sentiment analysis akan dilakukan pada data tweets, yang sebelumnya akan dimasukkan ke dalam penamaan final data.

|  |
| --- |
| final\_data = pd.DataFrame(data[['Tweets']])  final\_data.head() |

### Kode program 28. Mengubah penamaan data final

|  |
| --- |
|  |

### Gambar 5. Output berupa tampilan data

Kemudian data dipanggil, dan dihitung nilai subjectivity dan polaritas nya sesuai dengan fungsi untuk deklarasi ambang batas nilai yang telah ditetapkan.

|  |
| --- |
| **final\_data['Subjectivity'] = final\_data['Tweets'].apply(getSubjectivity)**  **final\_data['Polarity'] = final\_data['Tweets'].apply(getPolarity)**  **final\_data['TextBlob'] = final\_data['Polarity'].apply(analyze)**  **final\_data.head()** |

### Kode program 29. Memanggil data dan mendapatkan nilai subjectivity dan polarity

Dan hasil yang diberikan adalah sebagai berikut, dimana untuk setiap kalimat pada tweet akan diberikan kembalian nilai subjectivity dan polaritas, yang berupa nilai rata rata untuk semua sentiment.

|  |
| --- |
|  |

### Gambar 6. Ouput berupa berisi subjectivity dan polarity

dan perintah berikut akan menghitung secara keseluruhan nilai sentiment pada keseluruhan data.

|  |
| --- |
| tb\_counts = final\_data.TextBlob.value\_counts()  tb\_counts |

### Kode program 30. Menghitung nilai sentimen secara keseluruhan

output:

|  |
| --- |
| 0 108  1 81  -1 11  Name: TextBlob, dtype: int64 |

## **3.5 Model**

selanjutnya adalah pemodelan classification dengan algoritma Naive bayes. dimana data akan dipersiapkan terlebih dahulu untuk diolah dengan Machine learning.

Library yang akan digunakan, dalam mendukung proses pengolahan data di akan import terlebih dulu.

|  |
| --- |
| # Naive Bayes Classification  # Importing the libraries  import os  import pandas as pd  import numpy as np  import nltk  import matplotlib.pyplot as plt  import seaborn as sns  from sklearn.feature\_extraction.text import CountVectorizer, TfidfVectorizer  from sklearn import preprocessing  from sklearn.model\_selection import train\_test\_split  from sklearn.naive\_bayes import MultinomialNB  from sklearn import metrics  from imblearn.over\_sampling import SMOTE |

### Kode program 31. Pemodelan dengan algoritma Naive Bayes

kemudian dataset dibaca dalam bentuk dataframe.

|  |
| --- |
| # Importing the dataset  df = pd.read\_csv('/content/gdrive/My Drive/PDB/Final\_Data.csv') |

### Kode program 32. Membaca dataset berupa dataframe

dan berikut adalah perintah yang akan digunakan untuk menampilkan lima nilai teratas pada dataset.

|  |
| --- |
| #looking at the first 5 values of the dataset  df.head() |

### Kode program 33. Menampilkan lima nilai teratas

output:

|  |
| --- |
|  |

### Gambar 7. Tampilan nilai teratas

Kemudian dataset akan di split secara acak menjadi train data dan test data.

|  |
| --- |
| X = df['Tweets']  Y = df['TextBlob']  from sklearn.model\_selection import train\_test\_split as tt  X\_train, X\_test, Y\_train, Y\_test = tt(X, Y,test\_size=0.2, random\_state=100) |

### Kode program 34. split data train dan test

Dimana data train adalah sebanyak 160

|  |
| --- |
| X\_train.shape |

### Kode program 35. Menampilkan jumlah data train

output:

|  |
| --- |
| (160,) |

Dan data test sebanyak 40

|  |
| --- |
| X\_test.shape |

### Kode program 36. Menampilkan jumlah data test

output:

|  |
| --- |
| (40,) |

Sebelumnya, data akan di vektorisasi dengan menyamakan satu dimensi untuk setiap kosakata yang sama dan meninggalkan nilai kosong bila tidak ditemukan kata tersebut sama sekali. pada kode ini ditunjukan fungsi dimana vektorisasi dilakukan pada data train menggunakan pustaka english

|  |
| --- |
| from sklearn.feature\_extraction.text import CountVectorizer  vector = CountVectorizer(stop\_words ='english')  vector.fit(X\_train) |

### Kode program 37. Fungsi vektorisasi pada data train

Dan berikut adalah perintah dimana setiap kosakata akan dihitung vektorisasinya.

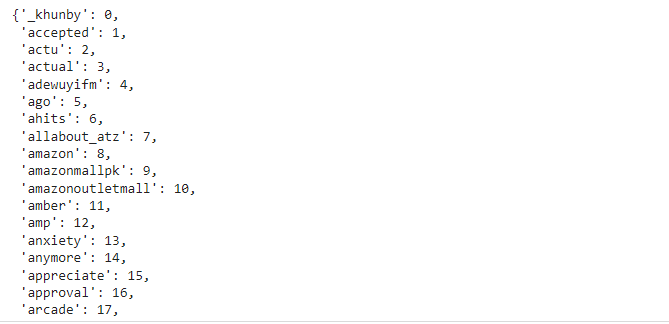
|  |
| --- |
| CountVectorizer(stop\_words='english') |

### 

|  |
| --- |
| vector.vocabulary\_ |

### Kode program 38. Perintah menghitung vektorisasi

output:



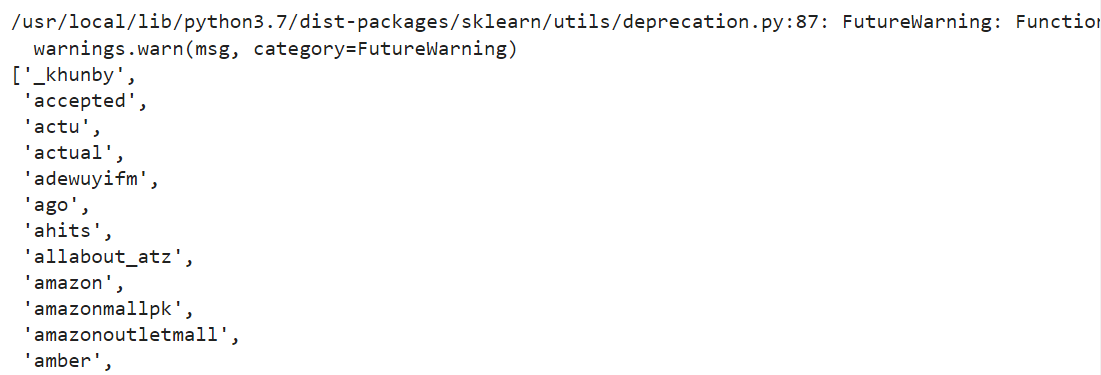
### Gambar 8. Tampilan hasil perhitungan vektorisasi

Dan berikut adalah perintah yang akan mengembalikan nilai future name.

|  |
| --- |
| vector.get\_feature\_names() |

### Kode program 39. Perintah mengembalikan nilai future name

output:



### Gambar 9. Tampilan output dengan nilai future name

Dan data train dan tes, juga akan kembali berikan vectornya.

|  |
| --- |
| X\_train\_transformed =vector.transform(X\_train)  X\_test\_transformed =vector.transform(X\_test) |

### 

Dan model klasifikasi, akan memanfaatkan algoritma Naive bayes untuk model multinomial. Pengklasifikasian akan menghitung jumlah fitur integer dan pada kode berikut akan ditampilkan nilai prediksi untuk untuk data train, juga nilai prediksi probabilitas untuk data test.

|  |
| --- |
| from sklearn.naive\_bayes import MultinomialNB  model = MultinomialNB()  model.fit(X\_train\_transformed,Y\_train)  y\_pred = model.predict(X\_test\_transformed)  y\_pred\_prob = model.predict\_proba(X\_test\_transformed) |

### Kode program 40. Klasifikasi dengan algoritma Naive Bayes

output:

|  |
| --- |
| [[ 0 4 1]  [ 0 23 1]  [ 0 1 10]]  0.825 |

Kemudian akan ditampilkan juga nilai untuk akurasi, precision dan recall

|  |
| --- |
| from sklearn.metrics import confusion\_matrix,accuracy\_score,precision\_score,recall\_score,f1\_score  print(confusion\_matrix(Y\_test,y\_pred))  print()  print(accuracy\_score(Y\_test,y\_pred)) |

|  |
| --- |
| from sklearn.metrics import accuracy\_score  from sklearn.metrics import classification\_report  print(classification\_report(Y\_test, y\_pred)) |

### Kode program 41. Menampilkan nilai akurasi, presisi dan recall

output:

|  |
| --- |
| precision recall f1-score support  -1 0.00 0.00 0.00 5  0 0.82 0.96 0.88 24  1 0.83 0.91 0.87 11  accuracy 0.82 40  macro avg 0.55 0.62 0.58 40  weighted avg 0.72 0.82 0.77 40 |

# 

# 

# **Bab 4**

# **Hasil dan Pembahasan**

## **4.1 Hasil Prediksi**

Berikut adalah hasil yang diberikan dimana untuk setiap kalimat pada tweet akan diberikan kembalian nilai subjectivity dan polaritas.

|  |
| --- |
|  |

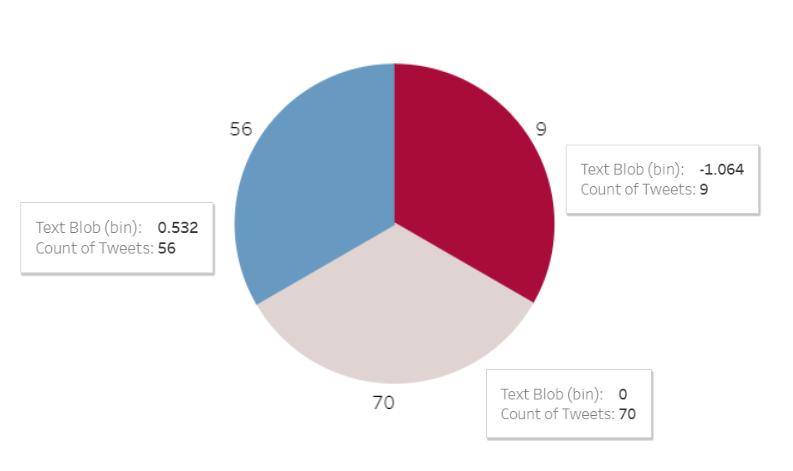
### Gambar 10. Tampilam kembalian nilai subjectivity dan polaritas

Adapun akurasi hasil proses klasifikasi pada sentiment analysis data tweeter adalah sebesar 82% .

|  |
| --- |
| precision recall f1-score support  -1 0.00 0.00 0.00 5  0 0.82 0.96 0.88 24  1 0.83 0.91 0.87 11  accuracy 0.82 40  macro avg 0.55 0.62 0.58 40  weighted avg 0.72 0.82 0.77 40 |

## **4.2 Hasil Visualisasi**

Visualisasi berikut menampilkan Jumlah tweet yang telah dikategorikan sesuai dengan rentang nilai dari Polarisasi yang didukung dari penggunaan TextBlob. Dimana rentang nilai -1 hingga 1 mendefinisikan bahwa -1 adalah sentimen negatif, 0 adalah sentiment netral dan 1 sebagai sentimen positif. Pada visualisasi ini terdapat 9 tweets dengan sentimen negatif, 70 tweet dengan sentimen netral dan 56 tweet lainnya dengan sentimen positif.



### Gambar 11. Tampilan visualisasi berdasarkan kategori sentimen

## **4.3 Kesimpulan**

Dengan dilakukanya proses sentiment analysis menggunakan algoritma *machine learning Classification,* dan mengkategorikan sentiment kepada tiga ambang batas yaitu 1, 0, dan -1 yang mendefinisikan sentiment mengarah pada tweet positif, netral dan negatif. bahwa klasifikasi menggunakan naive bayes dapat memberikan tingkat akurasi yang cukup baik yakni mencapai nilai sebesar 82%.

# 

# 

# **DAFTAR PUSTAKA**