BAB 3
METODE PENELITIAN

Metode *Naïve Bayes Classifer* dipilih dalam proses penelitian ini. Penggunaan metode *Naïve Bayes Classifer* (NBC) dalam penelitian ini dipilih dikarenakan pada algoritma NBC dapat melakukan proses pengolahan data diskrit dan data kuantitatif (Mustafa et al., 2018). Penelitian ini menggunakan data *tweet,* kemudian dilakukan proses pengolahan data berupa preprocessing. Hasil dari tahap pengolahan data digunakan untuk memetakan informasi opini masyarakat terhadap proses transfer pemain LigaSpanyol.

Penelitian ini bermula dari identifikasi permasalahan, kemudian pengolahan data yang sudah didapat dan menentukan sentimen analisis, sehingga hasil penelitian ini berupa informasi yang tepat dan sesuai.

3.1 BAHAN PENELITIAN

Pada penelitian ini bahan utama yaitu berupa data *tweet* dan *retweet* di media sosial Twitter yang berkaitan tentang tema Transfer Pemain Liga Spanyol. Penelitian ini menggunakan data *tweet* yang didapatkan dari Twitter dengan keyword/hastag "Bursa pemain La Liga Spanyol", "Transfer La Liga", "Transfer Real Madrid", "Transfer Barcelona", "Transfer Liga Spanyol" dan "Transfer Copa Del Ray" serta akun-akun sosial media Twitter yang meliput berita tentang dunia sepak bola seperti @idextratime, @PanditFootball.com, dan @SuperSoccerTV

3.2 ALAT PENELITIAN

Alat pada penelitian ini berupa laptop dengan spesifikasi cukup untuk menjalankan proses pengolahan data dan mampu untuk koneksitas Internet. Adapun Sistem Operasi dan program-program aplikasi yang dipergunakan dalam dalam pengembangan aplikasi ini sebagai berikut:

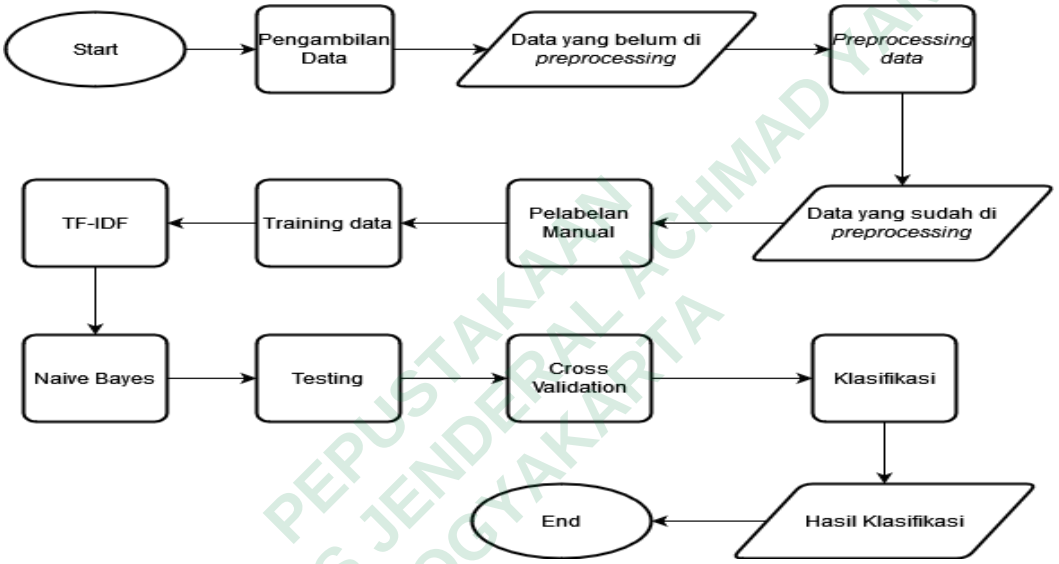
1. OS: *Windows 10 64-bit*
2. Bahasa Pemrograman: *Python 3.10.2*
3. *Microsoft Excel 2019*

4. Jupyter Notebook

3.3 JALAN PENELITIAN

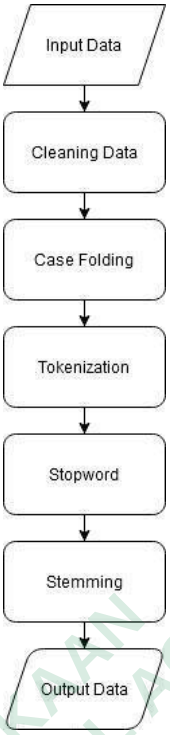
Pada penelitian ini tahap pertama adalah menggunakan Bahasa pemrograman *Python* untuk melakukan pengambilan data yang akan di tampilkan pada Microsoft Office Excel dan divisualisasikan dengan bantuan *library Python*. Pada gambar 3.1 merupakan alur penelitian:

.



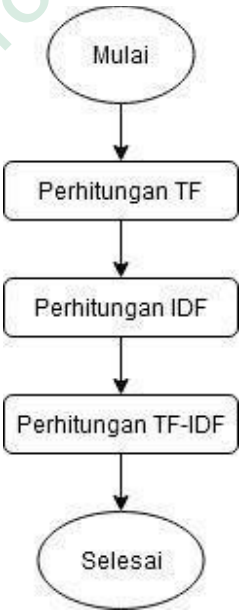
Gambar 3.1 Flowchart Jalan Penelitian

Pada tahap preprocessing terdapat subproses meliputi cleaning, casefolding, tokenization, stopword dan stemming. Pada gambar 3.2 merupakan alur dari subproses pada preprocessing data :



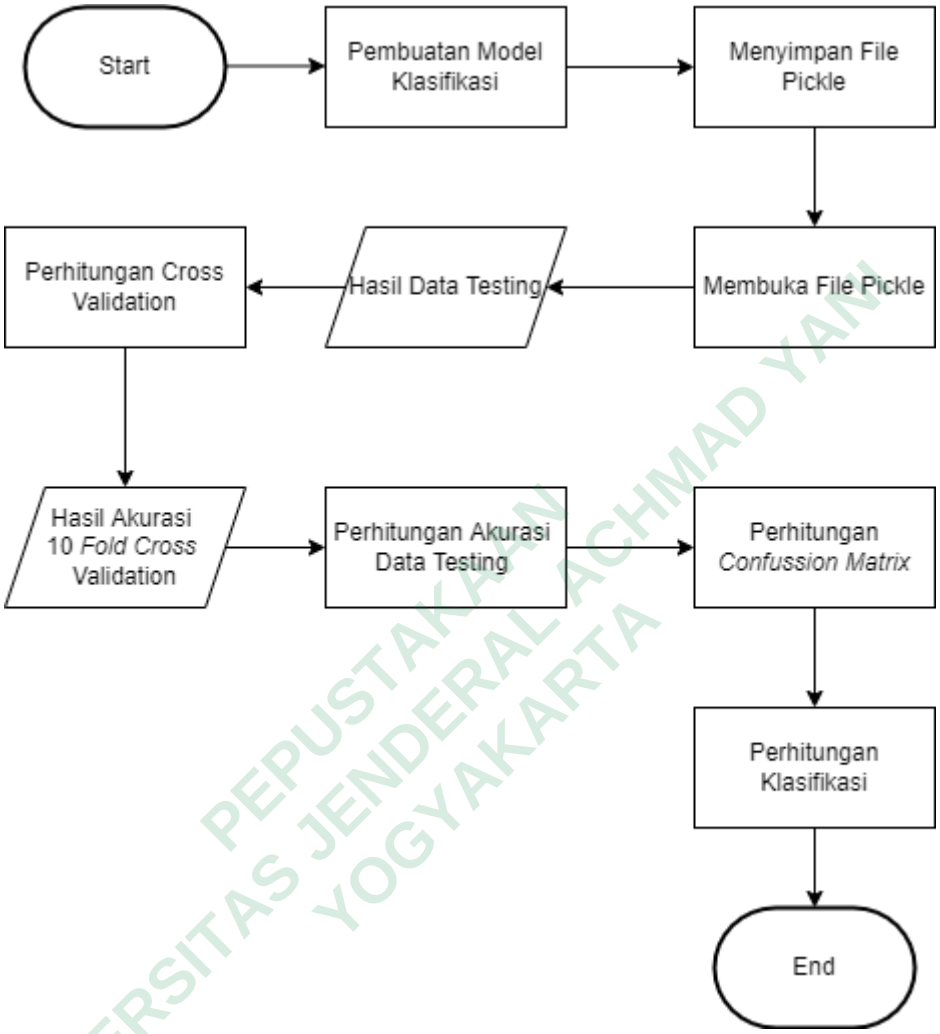
Gambar 3.2 Flowchart Preprocessing

Setelah melalui tahap data training terdapat subproses yaitu, TF-IDF untuk melakukan perhitungan *term* pada setiap dokumen yang dapat dilihat pada gambar 3.3:



Gambar 3.3 Alur perhitungan TF-IDF

Adapun tahapan yang menggambarkan perhitungan metode Naïve Bayes, bisa dilihat pada Gambar 3.4



Gambar 3.4 Flowchart Naive Bayes

3.3.1 Pengambilan Data

Pengambilan data merupakan tahap pertama untuk pengambilan data berupa topik mengenai “Bursa pemain La Liga Spanyol” dengan menggunakan Jupyter Notebook. Data yang di ambil berupa keyword “Transfer La Liga”, “Transfer Real Madrid”, “Transfer Barcelona”, “Transfer Liga Spanyol” dan “Transfer Copa Del Ray”. Data tweet di ambil dari periode 1 Januari 2020 sampai dengan 31 Mei 2022, dengan jumlah data total 11.282.

Proses pengambilan data pada penelitian ini menggunakan *library snscrape.module.twitter*, peneliti menggunakan *library snscrape* dikarenakan tanpa API dan tidak terbatas pada periode waktu tertentu. Kemudian *library pandas* untuk manipulasi dan menampilkan data. Pada gambar 3.4 merupakan source code import *library snscrape* dan *pandas*.

```
import snscrape.modules.twitter as sntwitter
import pandas as pd
```

Gambar 3.5 Import library crawling data

Setelah melakukan import library, kemudian melakukan kode untuk pengambilan data. Dimulai dengan berapa jumlah data yang ingin di ambil, kata kunci/topik yang ingin di ambil serta periode waktu pengambilan data. Pada gambar 3.5 merupakan source code crawling data.

```
# Setting variables to be used below
maxTweets = 5000

# Creating list to append tweet data to
tweets_list2 = []

# Using TwitterSearchScraper to scrape data and append tweets to list
for i,tweet in enumerate(sntwitter.TwitterSearchScraper('transfer Copa Del Rey since:2020-01-01 until:2022-05-31 lang:id').get_items()):
    if i>maxTweets:
        break
    tweets_list2.append([tweet.id, tweet.date, tweet.username, tweet.content])
```


Gambar 3.6 Crawling data

Data yang sudah berhasil diambil kemudian disimpan kedalam format Microsost Office Excel atau dalam bentuk format CSV. Pada tabel 3.1 meruapakan contoh data *tweet* yang diambil sebagai berikut:

Tabel 3.1 Contoh data *tweet* dan *re-tweet*

No	Tweet
1	Deal! Perisic Ke Tottenham Selamat Tinggal Abramovich Real Madrid Juara Berita Transfer Pemain https://t.co/Zk98NPhSFa
2	@Marcelo akan meninggalkan Real Madrid setelah 15 tahun membela klub, dengan status bebas transfer. Berpamitan pada momen yang tepatsetelah memenangkan LaLiga dan Champions League sebagai kapten tim!!. https://t.co/NjZ2ZDAXnm
3	Real Madrid!!! Sapa mau senggol haa?? Terlepas dari si masalah transfer gajelas yang kemarin, kita sudahmenang dan buktikan siapa rajanya
4	Dapat Dorongan Transfer Untuk Bintang Leeds Target Utama Barcelona https://t.co/0hOmBZWSaY https://t.co/GkdGZcZpN7
5	,Xavi Hernandez membenarkan bahwa Robert Lewandowski masuk radar Barcelona. Barcelona sedang mengupayakan transfer Lewandowski ke Camp Nou. https://t.co/CASqypbm52
6	,Kata Barcelona Soal Transfer Frenkie De Jong ke MU https://t.co/0ve6ptR6qL
7	@bolasportdotcom,"Ingin Berkhianat ke Barcelona, Gelandang Real Madrid Siap Ulangi Transfer Haram 14 Tahun Lalu https://t.co/D4naf3kqA2"
8	@KhbrkNews, Barcelona plan bargain transfer deal for Man Utd star Nemanja Matic https://t.co/rWwjIqonHs https://t.co/iz5529v4GE
9	"Terdepan! Chelsea Segera Segel Transfer Dembele dari Barcelona - https://t.co/kpmQt1cweG
10	Mbappe minta Paris Saint-Germain untuk memboyong gelandang Barcelona Frenkie de Jong saat jendela transfer musim panas dibuka pada Juni nanti. https://t.co/QkCqwaTdmz

3.3.2 Preprocessing

Pada tahap *preprocessing*, data teks disiapkan agar dapat dipergunakan dalam proses selanjutnya. Pada proses preprocessing dibutuhkan beberapa *library* untuk membantu jalannya tahap *preprocessing*. Beberapa *library* yang dibutuhkan seperti *library pandas* yang berguna untuk manipulasi data, *library numpy* untuk komputasi serta *library nltk* untuk membantu pengolahan natural language. Pada gambar 3.6 merupakan *import library* yang perlu dilakukan sebelum melakukan *preprocessing*, kemudian perintah untuk menampilkan data *tweet* yang berupa file CSV akan di panggil ke dalam dataframe. Gambar 3.6 merupakan source code untuk melakukan *import library pandas* dan pemanggilan data.

```
import pandas as pd, numpy as np, nltk, string, emoji, re
from pandas import DataFrame

# mebaca data set .csv
def load_data():
    data = pd.read_csv('laligaspanyol.csv', nrows=None, header=0, names=['Tanggal','Username','Text'])
    return data

data = load_data()

data.head()
```

Gambar 3.7 Import library dan pemanggilan data

Berikut merupakan tahapan-tahapan dalam sub proses preprocessing

1. Cleaning Data

Pada proses cleaning data, proses ini berfungsi untuk menghilangkan link url, tanda baca, angka, simbol dan username yang berada pada *tweet* dan *re-tweet*. Pada gambar 3.7 merupakan source code cleaning data :

```
def cleaning_text(text):
    # hapus tab, newline, dan backslash
    tab = text.replace('\t', ' ').replace('\n', ' ').replace('\\', ' ')
    # hapus underscore
    score = tab.replace('_', '')
    # hapus user mention
    user = re.sub('@[A-Za-z0-9]+', '', score)
    # hapus link
    link = re.sub(
        '((https?):((//)|(\\\\))+([\w\d:#@%/;$()~_?\+-=\\\.&](#!)?)*)+', '', user)
    # menghapus url
    url = re.sub(r'http\S+', '',link)
    # menghapus punctuation
    punc = re.sub(r'[^\w\s]','',url)
    # menghapus retweet (rt)
    rt = re.sub(r'RT[\s]+', '', punc)
    # menghapus angka
    no = re.sub('[0-9]+', '', rt)
    # menghapus slang
    slang = re.sub(r'\\n', " ", no)
    # menghapus regex
    reg = re.sub("b'", " ", slang)
    # hapus hashtag
    hashtag = re.sub('/#[\w_]+[ \t]*/', '', reg)
    # menghapus emoticon
    emot = emoji.get_emoji_regexp().sub("", hashtag).strip()
```

Gambar 3.8 Source Code Cleaning Data

Setelah dilakukan proses cleaning data, maka hasil akhir dari data yang setelah di bersihkan seperti pada tanel 3.2 sebagai berikut :

Tabel 3.2 Data tweet bersih

No	Tweet
1	Deal Perisic Ke Tottenham Selamat Tinggal Abramovich Real Madrid Juara Berita Transfer Pemain
2	Marcelo akan meninggalkan Real Madrid setela tahun membela klub, dengan status bebas transfer. Berpamitan pada momen yang tepat setelah memenangkan LaLiga dan Champions League sebagai kapten tim
3	Real Madrid Sapa mau senggol haa Terlepas dari si masalah transfer gajelas yang kemarin kita sudah menang dan buktikan siapa rajanya
4	Dapat Dorongan Transfer Untuk Bintang Leeds Target Utama Barcelona
5	Terdepan Chelsea Segera Segel Transfer Dembele dari Barcelona
6	Mbappe minta Paris Saint Germain untuk memboyong gelandang Barcelona Frenkie de Jong saat jendela transfer musim panas dibuka pada Juni nanti
7	Xavi Hernandez membenarkan bahwa Robert Lewandowski masuk radar Barcelona. Barcelona sedang mengupayakan transfer Lewandowski ke Camp Nou
8	Kata Barcelona Soal Transfer Frenkie De Jong ke MU
9	Ingin Berkhianat ke Barcelona, Gelandang Real Madrid Siap Ulangi Transfer Haram Tahun Lalu

10	KhbrkNews Barcelona plan bargain transfer deal for Man Utd star Nemanja Matic

2. Tokenization

Pada proses *Tokenization* bertujuan untuk melakukan pemecahan dari sebuah kalimat menjadi sebuah potongan kata. Dari sebuah potongan kata dapat disimpak kedalam sebuah dokumen. Pada gambar 3.8 merupakan source code *Tokenization*.

```
def clean_text(tweet):
    for i in tweet:
        cleaned.append(cleaning_text(
            re.sub("[\n\r\t\xa0]"," ",i).strip()))
clean_text(data["Text"])
```

Gambar 3.9 Source code Tokenization

3. Case Folding

Dalam proses penarikan data *tweet* bursa transfer pemain La Liga Spanyol, data masih disimpan dalam keadaan *raw data*/data mentah. Sehingga apabila dilakukan analisis tanpa memiliki standard, maka hasil analisis tidak akurat. Sehingga, diperlukan proses *Case Folding* untuk melakukan konversi dari bentuk awal menjadi bentuk standard. *Case Folding* dalam penelitian bursa transfer pemain La Liga Spanyol berguna untuk merubah kalimat ke bentuk *uppercase* atau *lowercase* secara standard. Pada gambar 3.9 merupakan source code *Case Folding*.

```
def lowercase():
    lower_word = data['Clean_Text'].str.lower()
    return lower_word

lower_tweet = lowercase()

print(lower_tweet)
```

Gambar 3.10 Source code Case Folding

4. Stopword Removal

Stopword Removal merupakan proses penghapusan kata- kata yang tidak memiliki informasi. Pada proses ini mengambil kata- kata penting dan membuang kata-kata yang kurang penting seperti “yang”, “di”, “dan”, “dari”, dll. Peneliti menggunakan proses *Stopword Removal* untuk menghapus kata-kata yang memiliki informasi rendah dari teks *tweet* bursa transfer pemain La Liga Spanyol. Pada gambar 3.10 merupakan source code untuk *stopword removal*.

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
stopwords = factory.get_stop_words()
print(stopwords)
```

Gambar 3.11 source code stopword removal

Pada stopwordremoval diperlukan panghapusan kata-kata yang terdapat pada *library sastrawi*. Pada tabel 3.3 merupakan kata-kata yang terdapat pada *library sastrawi*.

Tabel 3.3 Daftar Kata *Library Sastrawi*

Daftar Kata library Sastrawi
['yang', 'untuk', 'pada', 'ke', 'para', 'namun', 'menurut', 'antara', 'dia', 'dua', 'ia', 'seperti', 'jika', 'jika', 'sehingga', 'kembali', 'dan', 'tidak', 'ini', 'karena', 'kepada', 'oleh', 'saat', 'harus', 'sementara', 'setelah', 'belum', 'kami', 'sekitar', 'bagi', 'serta', 'di', 'dari', 'telah', 'sebagai', 'masih', 'hal', 'ketika', 'adalah', 'itu', 'dalam', 'bisa', 'bahwa', 'atau', 'hanya', 'kita', 'dengan', 'akan', 'juga', 'ada', 'mereka', 'sudah', 'saya', 'terhadap', 'secara', 'agar', 'lain', 'anda', 'begitu', 'mengapa', 'kenapa', 'yaitu', 'yakni', 'daripada', 'itulah', 'lagi', 'maka', 'tentang', 'demi', 'dimana', 'kemana', 'pula', 'sambil', 'sebelum', 'sesudah', 'supaya', 'guna', 'kah', 'pun', 'sampai', 'sedangkan', 'selagi', 'sementara', 'tetapi', 'apakah', 'kecuali', 'sebab', 'selain', 'seolah', 'seraya', 'seterusnya', 'tanpa', 'agak', 'boleh', 'dapat', 'dsb', 'dst', 'dll', 'dahulu', 'dulunya', 'anu', 'demikian', 'tapi', 'ingin', 'juga', 'nggak', 'mari', 'nanti', 'melainkan', 'oh', 'ok', 'seharusnya', 'sebetulnya', 'setiap', 'setidaknya', 'sesuatu',

'pasti', 'saja', 'toh', 'ya', 'walau', 'tolong', 'tentu', 'amat', 'apalagi', 'bagaimanapun']

5. Stemming

Stemming merupakan proses penghilangan infleksi kata menjadi bentuk dasar. Peneliti menggunakan *Pyhton Sastrawi* dalam proses *Stemming*, hal ini dikarenakan data *tweet* yang berbahasa Indonesia semua kata imbuhan sufiks dan prefix harus dihilangkan. Pada gambar 3.11 merupakan source code *stemming*

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

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

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

term_dict = {}

for document in stopwords_tweet:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ""

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

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

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

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

stem_tweet = stopwords_tweet.apply(get_stemmed_term)

print(stem_tweet)
```

Gambar 3.12 Source Code Stemming

6. **Normalisasi**

Normalization merupakan penyeragaman pada term yang mengalami kesalahan penulisan atau menggunakan bahasa yang tidak baku. Peneliti membuat *dataset* tentang *term* baku sehingga *dataset* berguna untuk

menyeragamkan kata yang tidak sesuai. Hasil dari proses *Normalization* berupa data *tweet* yang lebih terstruktur serta dapat dilakukan perhitungan pada proses selanjutnya. Pada gambar 3.12 merupakan source code untuk proses normalisasi

```
normalizad_word = pd.read_excel("normalisasi.xlsx")

normalizad_word_dict = {}

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

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

normal_tweet = stem_tweet.apply(normalized_term).str.join(" ")

print(normal_tweet)
```

Gambar 3.13 Source Code Normalisasi

Berikut contoh data pada file *normalisasi.xlsx* yang telah dibuat berdasarkan topik pembahasan sebagai dataset memperbaiki kata yang salah dapat dilihat pada Tabel 3.5

Tabel 3.4 Dataset Normalisasi

No	Kata Sebelum	Kata Sesudah
1	Yg	Yang
2	Dl	Dahulu
3	Stlh	Setelah
4	Akn	Akan
5	Stju	Setuju
6	Lnjtkn	Lanjutkan
7	Dg	Dengan
8	Lg	Lagi
9	Gt	Gitu
10	Pny	Punya

3.3.3 Pelabelan Manual

Pada tahap pelabelan manual, tahap ini merupakan proses memberikan

lavel terhadap kata pada dokumen sehingga dapat dianalisis lebih lanjut mengenai sifat yang positif atau negatif. Pada proses pelabelan manual peneliti membagi data tweet yang berisi tentang bursa transfer pemain dengan data training label positif dan label negatif. Data *tweet* yang sudah dilabeli dengan jumlah data 600 tweet dari 600 *tweet* dengan masing-masing 300 *tweet* positif dan 300 *tweet* negatif dari data *training.* Pada gambar 3.13 merupakan hasil pelabelan manual

	no	tanggal	username	tweet	label	kelas
0	1	2012-03-07 17:32:43+00:00	agentaruhanbola	account transfer main bola situs taruh bola li...	Positif	1
1	2	2010-06-20 09:15:32+00:00	baskaramp	aduh aduh kangen liga inggris spanyol itali ni...	Positif	1
2	3	2021-04-15 16:30:00+00:00	Bolanet	akhir spekulasi barcelona segera umum transfer...	Negatif	0
3	4	2021-04-15 10:28:14+00:00	M88Indo	akhir spekulasi barcelona segera umum transfer...	Negatif	0
4	5	2020-08-30 15:51:35+00:00	_fireshare	aku masuk opini kalau macam insentif bisnis sp...	Negatif	0

Gambar 3.14 Hasil Pelabelan Manual

Ditunjukan bahwa label positif diberi kelas 1 dan nilai kelas 0 untuk label negatif. Pelabelan manual dilakukan untuk perhitungan akurasi yang telah diberi sentimen positif dan negatif.

3.3.4 Data Training

Pada proses training data menggunakan metode *Naïve Bayes Classifer*. Pada tahap ini diawali dengan fitur ekstrasi pada teks menggunakan TF-IDF, kemudian dilakukan proses training data untuk membuat model klasifikasi sentiment. Berikut contoh perhitungan TF-IDF secara manual dengan Microsoft Office Excel dapat dilihat pada Tabel 3.6 :

Tabel 3.5 Data Training

Dokumen (d)	**Kalimat**
d1	real madrid lineup ganti raphael varane tengah minat transfer manchester united.
d2	real madrid tumbal vinicius junior transfer kylian mbappe
d3	update transfer main sergio aguero sepakat gabung barcelona kontrak tahun depan

d4	rumor transfer sergio aguero sepakat gabung barcelona musim depan

Dalam melakukan perhitungan *Term Frequency* (TF) ini menggunakan beberapa komponen yaitu *term* atau kata, dan d merupakan jumlah data yang akan digunakan terdiri dari d1, d2, d3 dan d4 dan df untuk menghitung jumlah term atau kata yang muncul pada setiap dokumen. Contoh perhitungan TF dapat dilihat pada Tabel 3.6

Tabel 3.6 Perhitungan TF

Term (kata)	d1	d2	d3	d4	df
real	1	1			2
madrid	1	1			2
lineup	1				1
ganti	1				1
raphael	1				1
varane	1				1
tengah	1				1
minat	1				1
transfer	1	1	1	1	4
manchester	1				1
united	1				1
tumbal		1			1
vinicius		1			1
junior		1			1

kylian		1			1
mbappe		1			1
update			1		1
main			1		1
sergio			1	1	2
aguero			1	1	2
sepakat			1	1	2
gabung			1	1	2
barcelona			1	1	2
kontrak			1		1
tahun			1		1
depan			1	1	2
rumor				1	1
musim				1	1

Perhitungan *Invers Document Frequency* (IDF) menggunakan beberapa komponen seperti *term* atau kata, df dan idf yang berhubungan antara ketersediaan suatu *term* di semua dokumen, dihitung dengan N atau jumlah dokumen. Berikut contoh perhitungan IDF dapat dilihat pada Tabel 3.7

Tabel 3.7 Perhitungan IDF

Term (kata)	df	Idf	Idf(N=4)	Idf(N=600)
real	2	0.5	0.30103	2,477121
madrid	2	0.5	0.30103	2,477121

lineup	1	1	0.60206	2,778151
ganti	1	1	0.60206	2,778151
raphael	1	1	0.60206	2,778151
varane	1	1	0.60206	2,778151
tengah	1	1	0.60206	2,778151
minat	1	1	0.60206	2,778151
transfer	4	0.25	0	2,176091
manchester	1	1	0.60206	2,778151
united	1	1	0.60206	2,778151
tumbal	1	1	0.60206	2,778151
vinicius	1	1	0.60206	2,778151
junior	1	1	0.60206	2,778151
kylian	1	1	0.60206	2,778151
mbappe	1	1	0.60206	2,778151
update	1	1	0.60206	2,778151
main	1	1	0.60206	2,778151
sergio	2	0.5	0.30103	2,477121
aguero	2	0.5	0.30103	2,477121
sepakat	2	0.5	0.30103	2,477121
gabung	2	0.5	0.30103	2,477121
barcelona	2	0.5	0.30103	2,477121

kontrak	1	1	0.60206	2,778151
tahun	1	1	0.60206	2,778151
depan	2	0.5	0.30103	2,477121
rumor	1	1	0.60206	2,778151
musim	1	1	0.60206	2,778151

Tabel 3.8 menjelaskan perhitungan TF-IDF secara manual dengan menggunakan Microsof Office Excel dari hasil perkalian tf dan idf

Tabel 3.8 Perhitungan TF-IDF

Term (kata)	d1	d2	d3	d4
real	0.30103	0.30103		
madrid	0.30103	0.30103		
lineup	0.60206			
ganti	0.60206			
raphael	0.60206			
varane	0.60206			
tengah	0.60206			
minat	0.60206			
transfer	0.60206	0.60206	0.60206	0.60206
manchester	0.60206			
united	0.60206			
tumbal		0.60206		

vinicius		0.60206		
junior		0.60206		
kylian		0.60206		
mbappe		0.60206		
update			0.60206	
main			0.60206	
sergio			0.30103	0.30103
aguero			0.30103	0.30103
sepakat			0.30103	0.30103
gabung			0.30103	0.30103
barcelona			0.30103	0.30103
kontrak			0.60206	
tahun			0.60206	
depan			0.30103	0.30103
rumor				0.60206
musim				0.60206

Pada proses perhitungan TF-IDF ini menggunakan *library sklearn.feature_extraction.text* dan *TfidfVectorizer* untuk menjalankan proses perhitungan secara otomatis. Proses perhitungan TF-IDF dibantu menggunakan *library Multinomial Naïve Bayes* yang dimana membantu dalam mengklasifikasi teks pada data Data *Training*

Pada gambar 3.14 merupakan source code perhitungan TF-IDF pada jupyter notebook

```
from sklearn.feature_extraction.text import TfidfVectorizer

d1 = "tim buruk terlalu lebih utk line up yang isi mantan top skor liga prancis jerman main mahal afrika main muda potensial
d2 = "kurang puas barcelona siap belanja banyak main bursa transfer musim panas"

vect = TfidfVectorizer()
X = vect.fit_transform([d1, d2])

X.toarray()
```

Gambar 3.15 Kode perhitungan TF-IDF

Contoh hasil perhitungan TF-IDF dari sistem dapat dilihat pada Gambar 3.15 dan 3.16

```
[(0.15904861514904767, 'afrika'),
 (0.15904861514904767, 'aku'),
 (0.15904861514904767, 'apa'),
 (0.15904861514904767, 'arti'),
 (0.0, 'banyak'),
 (0.0, 'barcelona'),
 (0.0, 'belanja'),
 (0.0, 'bursa'),
 (0.31809723029809533, 'buruk'),
 (0.15904861514904767, 'defend'),
 (0.15904861514904767, 'gagal'),
 (0.15904861514904767, 'gak'),
 (0.15904861514904767, 'isi'),
 (0.15904861514904767, 'jerman'),
```

Gambar 3.16 Hasil Perhitungan TF-IDF Pertama

```
[(0.0, 'afrika'),
 (0.0, 'aku'),
 (0.0, 'apa'),
 (0.0, 'arti'),
 (0.3160304990863645, 'banyak'),
 (0.3160304990863645, 'barcelona'),
 (0.3160304990863645, 'belanja'),
 (0.3160304990863645, 'bursa'),
 (0.0, 'buruk'),
 (0.0, 'defend'),
 (0.0, 'gagal'),
 (0.0, 'gak'),
 (0.0, 'isi'),
 (0.0, 'jerman'),
```

Gambar 3.17 Hasil Perhitungan TF-IDF Pertama

Selanjutnya melakukan pencarian nilai akurasi data *training* yang telah dilakukan pelabelan secara manual. Pada gambar 3.17 merupakan source code untuk mencari nilai akurasi data training.

```
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix

print("Accuracy: {:.2f}%".format(accuracy_score(y_test, y_pred) * 100))
print("\nF1 Score: {:.2f}".format(f1_score(y_test, y_pred, average='weighted') * 100))
print("\nCOnfusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

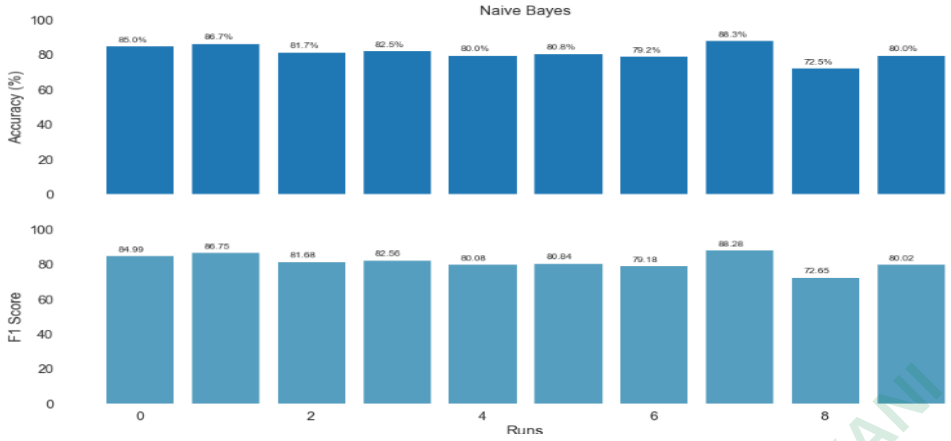
Gambar 3.18 Kode Akurasi Data Training

Cross validation merupakan sebuah metode untuk memperoleh hasil akurasi dengan melakukan percobaan sebanyak K kali agar nilai parameter mempunyai hasil yang sama. Prinsip *cross-validation* membagi data menjadi dua bagian, yaitu data latih dan data uji. Dengan menggunakan *library from sklearn.model_selection* dan import *ShuffleSplit* untuk menghitung rata-rata dalam 10 kali. Pada gambar 3.18 merupakan source code untuk menghitung *cross validation*.

```
fig, (ax1, ax2) = plt.subplots(2, 1, sharex=True, figsize=(16,9))

acc_scores = [round(a * 100, 1) for a in accs]
f1_scores = [round(f * 100, 2) for f in f1s]

x1 = np.arange(len(acc_scores))
x2 = np.arange(len(f1_scores))

ax1.bar(x1, acc_scores)
ax2.bar(x2, f1_scores, color='#559ebf')

# Place values on top of bars
for i, v in enumerate(list(zip(acc_scores, f1_scores))):
    ax1.text(i - 0.25, v[0] + 2, str(v[0]) + '%')
    ax2.text(i - 0.25, v[1] + 2, str(v[1]))

ax1.set_ylabel('Accuracy (%)')
ax1.set_title('Naive Bayes')
ax1.set_ylim([0, 100])

ax2.set_ylabel('F1 Score')
ax2.set_xlabel('Runs')
ax2.set_ylim([0, 100])

sns.despine(bottom=True, left=True)  # Remove the ticks on axes for cleaner presentation

plt.show()
```

Gambar 3.19 Kode Cross-validation

Maka hasil gafik dari cross validation seperti pada gambar 3.19 berikut :



Gambar 3.20 Grafik Cross Validation

Selanjutnya dilakukan pembuatan model klasifikasi dengan variabel X dan Y dengan data *training* yang sudah dilakukan. Model dibuat pada sebuah fungsi agar lebih mudah dalam pemanggilanya dan dijalankan untuk tahap berikutnya, sehingga membuatnya lebih efektif dan efisien. model klasifikasi tersebut menggunakan *library sklearn.pipeline* dengan *import pipeline*. Pada gambar 3.20 merupakan source code import library dan pembuatan model klasifikasi

```
import os
import pickle
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfTransformer

X = df.tweet
y = df.kelas

txt_classifier = Pipeline([('vect', TfidfVectorizer()),
                           ('tfidf', TfidfTransformer()),
                           ('classifier', MultinomialNB(alpha=1.0)),
                           ])
X_train = np.asarray(X)
txt_classifier = txt_classifier.fit(X_train, np.asarray(y))
```

Gambar 3.21 Import library dan pembuatan model klasifikasi

Selanjutnya file *pickle* yang sudah dibuat model klasifikasi akan digunakan untuk eksekusi data *testing* dari data yang digunakan adalah 200 *tweet* yang sudah dilakukan pelabelan secara manual dari data *training* yang berjumlah 600 *tweet* dan 200 *tweet* yang digunakan data *testing* mengambil dari total data 11.283 *tweet.*Pad gambar 3.21 merupakan source code untuk variable pemanggilan hasil dari Naive Bayes.

```
result_tweet=[]
for i in range(len(predicted)):
    if(predicted[i]==1):
        sentiment_result='Positif'
    elif(predicted[i]==0):
        sentiment_result='Negatif'
#     result_tweet.append({'class':prediction_linear[i], 'result_nbc':sentiment_result})
    result_tweet.append({'Cleaned_Text':data_tweet[i], 'class':predicted[i] })
```

Gambar 3.22 Kode Pemanggilan klasifikasi

3.3.5 Testing

Pada tahap testing merupakan tahapan untuk mengetahui tingkat keakuratan pemodelan yang telah dibangun serta untuk memklasifikasi label atau kelas dari data uji yang tersedia. Pada gambar 3.22 merupakan hasil pelabelan manual.

	Unnamed: 0	Cleaned_Text	actual	predicted
0	0	afk transfer deal done everton resmi dapat jam...	0	1
1	1	agen lautaro martinez datang italia barcelona ...	1	0
2	2	akhir saga transfer lionel messi bintang asal ...	0	0
3	3	aku manajemen chelsea jdi lebih bagus skrng so...	0	0
4	4	anchester united pasti dapat untung real madri...	1	1
...	...	...	...	...
195	195	video bursa transfer chelsea pinjam bintang ba...	1	0
196	196	wujud ingin ronald koeman belanja main petingg...	1	1
197	197	wujud ingin ronald koeman belanja main petingg...	1	1
198	198	yang takut kalo saing macam real madridjuventu...	0	0
199	199	yaudah lah ikhlasin aja banyak ujung tombak gu...	0	0

200 rows × 4 columns

Gambar 3.23 Hasil Pelabelan Manual dan Naive Bayes

1. Hasil Klasifikasi

Setelah mendapatkan nilai akurasi yang baik dari proses *training* dan *testing* dari pemodelan klasifiaksi maka dari itu dilakukannya tahap klasifikasi untuk data keseluruhan agar mendaptkan hasil sentimen positi dan negatif yang telah diuji dari tahapan *training* dan *testing.* Pada gambar 3.23 merupakan hasil klasifikasi seluruh tweet.

	tweet	class
0	mgoalcomidid spesial transfer main lima main p...	1
1	kaka liverpool mgoalcomidid spesial transfer m...	1
2	dlvr spesial transfer main bintang la liga spa...	0
3	infobarca spesial transfer main bintang la lig...	0
4	fcbinews spesial transfer main bintang la liga...	0
...	...	...
11278	real madrid transfer buat tahun w aurelien tch...	0
11279	isco umum diri resmi tinggal real madrid statu...	1
11280	pilih transfer real madrid ganti mbappe bintan...	1
11281	mungkin serius mo salah tinggal liverpool stat...	1
11282	depan chelsea segera segel transfer dembele ba...	0

11283 rows × 2 columns

Gambar 3.24 Hasil Klasiikasi