BAB 3
METODE PENELITIAN

3.1 METODA PENELITIAN

Penelitian ini merupakan pembuatan aplikasi analisis sentimen positif dan negatif pada data media sosial Twitter. Aplikasi ini menggunakan metode Naïve Bayes Classifier. Penelitian ini membutuhkan data *tweet* yang didapatkan dari Twitter yang berkaitan dengan topik ulasan pelayanan provider internet "indihome", pada selanjutnya melakukan proses *pre-processing* untuk mendapatkan hasil data yang bersih dan sesuai yang diharapkan. Data ini nantinya akan digunakan untuk memetakan informasi atau sentimen dari pengguna Twitter tentang komentar *internet service provider* (ISP) yang diperoleh sehingga menjadi informasi yang sesuai tentang komentar dan penyedia layanan tersebut.

Berikut dibawah ini adalah alat, bahan, dan proses jalannya penelitian analisis sentimen ISP "indihome" serta tahapan untuk menyelesaikan proses analisis sentimen menggunakan data *tweet*.

3.2 BAHAN PENELITIAN

Bahan yang dibutuhkan dalam penelitian ini adalah data *tweet, re-tweet* pada Twitter yang berkatian dengan ulasan pelayanan provider internet "indihome".

3.3 ALAT PENELITIAN

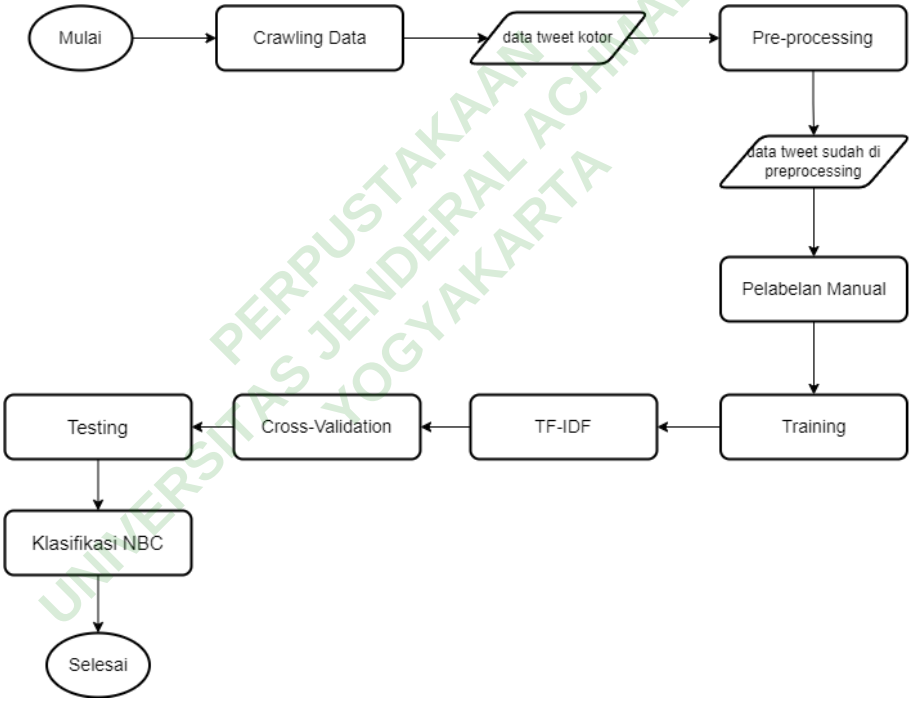
Alat yang diperlukan dalam penelitian ini adalah laptop dengan spesifikasi cukup dalam menjalankan proses pengolahan data dan mampu terkoneksi ke jaringan internet. Berikut adalah sistem operasi dan perangkat lunak yang akan digunakan:

1. Sistem Operasi Windows 10 Pro 64-bit.
2. Python v3.8.6.
3. Jupyter Notebook v6.4.4.
4. Framework Flask.
5. Sublime Text v1.0.0.1.

6. Microsoft Office Excel 2019.
7. XAMPP v3.3.0.
8. phpMyAdmin v5.1.1.

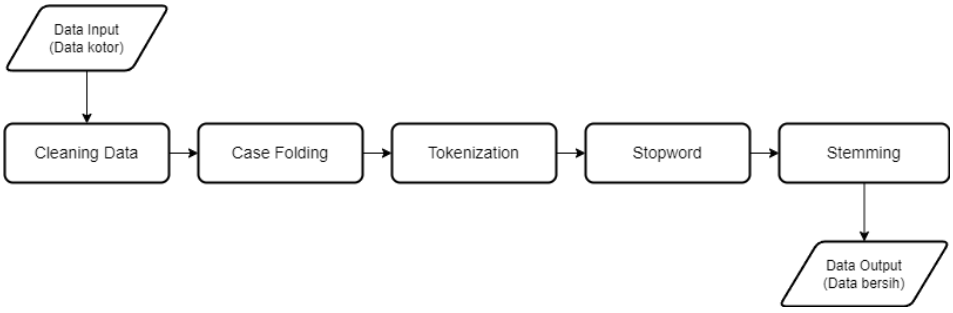
3.4 JALAN PENELITIAN

Penelitian ini menggunakan bahasa pemrograman python dan Jupyter Notebook untuk proses pengambilan data *tweet* dari Twitter dimana data tersebut akan tersimpan di dalam Microsoft Office Excel dalam bentuk format *Comma Separated Values* (*CSV*) dan dimodelkan dengan *library* pada bahasa pemrograman Python. Alur penelitian yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 3.1.



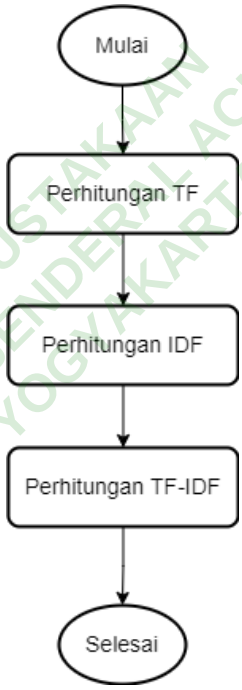
Gambar 3.1 Alur penelitian

Tahap *pre-processing* terdapat subproses terdiri dari *cleaning*, *casefolding*, *tokenization*, *stopword*, *stemming* yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 Alur *Pre-processing*

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

3.4.1 Pengambilan Data

Pengambilan Data adalah tahap dalam pengambilan data *tweet* dari mengenai topik ulasan layanan provider internet “indihome” dengan menggunakan perangkat lunak Command Prompt dan mengeksekusi di *console* Jupyter Notebook. Data *tweet* yang diambil menggunakan kata kunci ”indihome” dengan rentan waktu 1 juli – 11 juli 2022 dengan jumlah *tweet* yang didapatkan adalah 10.000 dengan

campuran data *tweet* dan *re-tweet* yang dimana akan ditampilkan di Microsoft Office Excel dalam format CSV.

1. Pengumpulan Data *Tweet* dan *Re-Tweet*

Pengumpulan data *tweet* dan *re-tweet* dilakukan dengan bantuan beberapa *library* dari Python. *Library* yang membantu dalam penelitian ini adalah snscrape.modules.twitter yang dimana *library* ini berfungsi mengambil data *tweet* dengan jumlah tidak terbatas, tanpa menggunakan *Application Programming Interface* (API) dan dengan rentan waktu yang dapat disesuaikan kebutuhan. Sedangkan untuk *library* tweepy membutuhkan API agar *developer* dapat mengambil data dari Twitter. *Library* pandas yang memiliki fungsi untuk manipulasi serta pembersihan data.

Kemudian untuk kode dalam pengambilan data mulai dari jumlah data *tweet* yang ingin diambil, topik, dan kolom apa saja yang ingin diambil untuk disimpan ke dalam Microsoft Office Excel dengan format CSV. Berikut kode untuk pengambilan *data tweet* dan *re-tweet* terdapat dibawah ini.

```
maxTweets = 10000

tweets_list = []

for   i,tweet   in   enumerate(sntwitter.TwitterSearchScraper('indihome
since:2022-06-01 until:2022-06-12 lang:id').get_items()):
    if i>maxTweets:
        break
    tweets_list.append([tweet.date, tweet.username, tweet.content])

tweets_df = pd.DataFrame(tweets_list, columns=['Datetime', 'Username',
'Text'])
tweets_df = tweets_df.drop_duplicates()
```

Data *tweet* tersebut tersimpan di dalam file yang dimana terdiri dari tanggal, username dan *tweet* dengan pemisah titik koma pada setiap kolomnya. Dan didalam teks *tweet* dan *re-tweet* terdapat berbagai macam variasi bentuk yang ditampilkan mulai dari *username*, angka, *hashtag*, link *Uniform Resource Locator* (URL) yang perlu diolah kembali datanya agar mendapatkan sesuai dengan keinginan. Contoh data *tweet* yang sudah dilakukannya pengambilan pada Tabel 3.1.

Tabel 3.1 Contoh Data *Tweet* dan *Re-tweet*

No	*Tweet*
1	Nikmatin hari libur nga ada aktivitas lain selain online dengan IndiHome dong https://t.co/WxU2F30HkU
2	"@RrQuinn55 @IndiHome Surabaya jawa Timur, Norak lu"
3	"@Kinantia321 Sama-sama, Kak Kinan. Semoga informasi yang disampaikan bermanfaat, ya. Selalu ikuti timeline kami untuk
4	"@Naura1012 Hai, Kak Naura. Terima kasih apresiasinya. Selamat menikmati kembali layanan IndiHome. Semoga IndiHome selalu membantu menjalani aktivitasnya, ya Kak. Stay safe, Stay healthy dan bersama IndiHome. ... https://t.co/ct1qQbH4BH"
5	"@cokyfv Sama-sama, Kak Cooky. Semoga informasi yang disampaikan bermanfaat, ya. Selalu ikuti timeline kami untuk informasi menarik lainnya seputar layanan dan produk IndiHome. Selamat beraktivitas kembali.... https://t.co/TneVtUT4R6"
6	,"@JauhariNA112 Sama-sama, Kak Jauhari. Semoga informasi yang disampaikan bermanfaat, ya. Selalu ikuti timeline kami untuk informasi menarik lainnya seputar layanan dan produk IndiHome. Selamat beraktivitas kembali. ... https://t.co/kgEq45Euj6"
7	eyeskha,Harus bersabar sebanyak apa lagi sama indihome
8	"Ada yg disuruh kirim ktp ga sm @IndiHome? Katanya utk kontrak digital. Lah data2 yg dulu kemana? Nih kerjaannya @IndiHomeCare, gmn nih Bang @erickthohir
9	"Halo min @IndiHome @IndiHomeCare lampu indikator LOS menyala kedap kedip warna merah, internet ga bisa diakses sama sekali. Ini penyelesaiannya gimana??"
10	"@IndiHomeCare @IndiHome halloooo...indihome ngga kelar kelar nih problem... Putus nyambung putus nyambung, lampu pon kedip kedip"

3.4.2 Pre-processing

Pre-processing menjadi tahapan awal klasifikasi teks untuk menyiapkan data teks agar dapat digunakan nantinya pada proses yang berikutnya. dan membuat teks menjadi sebuah informasi dengan kualitas yang baik. Proses dari tahapan *pre-processing* ini memerlukan beberapa *library* dari python untuk membantu dalam proses *pre-processing* yaitu *Library* pandas yang memiliki fungsi untuk manipulasi serta pembersihan data, numpy untuk mendukung proses komputasi atau perhitungan numerik, nltk untuk mendukung proses pengolahan bahasa natural

pada data teks agar mempermudah dalam proses *classification*, *tokenization* dan *stemming*,

1. Cleaning

Cleaning yaitu proses untuk menghilangkan *link* URL, tanda baca, angka, simbol, slang dan username pada *re-tweet,* Berikut kode untuk proses data *cleaning* terdapat dibawah ini.

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

    return emot
```

Setelah proses *cleaning* sudah dilakukan maka mendapatkan sebuah data *tweet* yang bersih, berikut beberapa contoh dari hasil *cleaning* dapat dilihat pada Tabel 3.2.

Tabel 3.2 Contoh Data *Tweet* Bersih

No	Data *Tweet* Bersih
1	nikmat nonton bioskop rumah indihome cema ayo download sekarang lalu indihome apps store harga mulai film indihome keren banget.
2	terimakasih indihome karena mau baik wifi skrng hidup internet.
3	asyik banget temen aku main game online kantor biar menang tentu dukung koneksi kuat pakai indihome stabil banget sinyal.
4	mantap indihome
5	min bantu dong sudah minggu wifi tidak bisa sudah dilaporin via tetapi gada ubah please help..
6	yang penting internetan lancar bang kalau pakai indihome.

7	baca buku enak streaming musik jaringan indihome lancar nyaman.
8	streaming musik cek cek online shop apa jaringan nya pakai indihome stabil lancer.
9	aku lelah indihome masa jam gini lag padahal biasa tidak ganggu main game
10	min indihome lemot banget.

2. Tokenization

Tokenization adalah pemecahan teks pada data *tweet* menjadi potongan-potongan kata, berikut kode dalam proses *tokenization* terdapat dibawah ini.

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

Tokenization bertujuan untuk memisahkan karakter-karakter dalam sebuah kalimat sehingga dapat disimpan menjadi sebuah istilah atau setiap kata dalam sebuah dokumen.

3. Case Folding

Case Folding yaitu proses merubah setiap kata dalam dataset menjadi huruf kecil (*lower case*) atau menyamaratakan penggunaan huruf, berikut kode dalam proses dalam *case folding* dapat dilihat dibawah ini.

```
def casefolding():
    lower_text = data['Clean_Text'].str.lower()
    return lower_text
```

4. Stopword Removal

Stopword removal yaitu melakukan penghapusan kata tidak penting dari sebuah teks, misalnya "di", "oleh", "pada", "sebuah", "karena" dan lain sebagainya, dan menggunakan *library* sastrawi yang dimana fungsi dari *library* tersebut mengenali teks Bahasa Indonesia berikut kode yang digunakan dalam pemrosesan *stopoword removal* terdapat dibawah ini.

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

Dalam *stopword removal* diperlukan penghapusan kata-kata yang terdapat pada *library* Sastrawi. Daftar kata-kata yang digunakan terdapat pada Tabel 3.3.

Tabel 3.3 Daftar Kata Stopword Removal

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']

Setelah *list* dari kata-kata *stopword* muncul, kemudian menghilangkan kata-kata yang terdapat pada *term* sesuaikan dengan kata-kata yang terdapat pada data list *stopwod removal*. Berikut kode untuk menghilangkan kata-kata yang terdapat pada list *stopword removal* terdapat pada Gambar 3.12.

```
def removeStopWords(text):
    clean_word_list = [word for word in text.split() if word not
in stopwords]
    return clean_word_list
```

Fungsi dengan nama removeStopword memiliki sebuah fungsi dalam mengecek kata yang ada pada *term* sesuai atau tidak pada list kata *stopword*. Jika memang sudah sesuai dapat melakukan penghilangan kata pada variable clean_word_list.

5. Stemming

Stemming yaitu proses pemetaan dan penguraian bentuk kata ke bentuk kata dasarnya, Berikut kode yang digunakan dalam proses *stemming* terdapat dibawah ini.

```
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)
```

Stemming dilakukan dengan mengubah bentuk Dataframe menjadi bentuk *dictionary* yang nantinya akan menggunakan *library* StemmerFactory untuk mengubah istilah atau kata yang terdapat dalam *term* menjadi kata dasar.

6. Normalisasi

Normalisasi adalah memperbaiki kata-kata yang salah dalam teks berdasarkan korpus yang dibuat, Berikut kode yang digunakan dalam proses normalisasi terdapat dibawah ini.

```
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(" ")
```

normalisasi merupakan proses untuk mengubah kata yang tidak sesuai dengan *list* kata yang sudah dibuat pada file normalisasi.xlsx dan kata tersebut menjadi sebuah Dataframe dengan nama variabel `normal_tweet` yang nantinya akan menghasilkan data *tweet* yang sudah di *pre-processing*. Berikut contoh data pada file normalisasi.xlsx yang telah dibuat berdasarkan topik pembahasan sebagai dataset memperbaiki kata yang salah dapat dilihat pada Tabel 3.4.

Tabel 3.4 Daftar Kata Normalisasi

Kata sebelum	Kata sesudah
ga	tidak
lancaaaar	lancar
ngonsep	konsep
tau	tahu
tp	tetapi
yg	yang
krn	karena
tdk	tidak
knp	kenapa

Data yang sudah dilakukan *pre-processing* nantinya akan menghasilkan data *tweet* yang siap digunakan dan datanya lebih terstruktur untuk tahapan selanjutnya, Berikut data *tweet* yang sudah di *pre-processing* dapat dilihat pada Tabel 3.5.

Tabel 3.5 Contoh *Tweet Pre-processing*

No	Hasil *Pre-processing*
1	nikmat nonton bioskop rumah indihome cema ayo download sekarang lalu indihome apps store harga mulai film indihome keren banget.

2	terimakasih indihome karena mau baik wifi skrng hidup internet.
3	asyik banget temen aku main game online kantor biar menang tentu dukung koneksi kuat pakai indihome stabil banget sinyal.
4	mantap indihome
5	min bantu dong sudah minggu wifi tidak bisa sudah dilaporin via tetapi gada ubah please help..
6	yang penting internetan lancar bang kalau pakai indihome.
7	baca buku enak streaming musik jaringan indihome lancar nyaman.
8	streaming musik cek cek online shop apa jaringan nya pakai indihome stabil lancer.
9	aku lelah indihome masa jam gini lag padahal biasa tidak ganggu main game
10	min indihome lemot banget.

Dapat dilihat pada Tabel 3.5 bahwa data *tweet* menjadi data yang lebih baik dan terstruktur setelah dilakukan *pre-processing* untuk tahapan selanjutnya yaitu, pelabelan manual.

3.4.3 Pelabelan Manual

Pelabelan manual adalah proses memberikan label terhadap teks atau kalimat suatu dokumen sehingga sifat positif atau negatifnya dapat dianalisis lebih lanjut. Data *tweet* yang sudah dilabeli dengan jumlah data 1000 *tweet* dari 10000 *tweet* dengan masing-masing 500 *tweet* positif dan 500 *tweet* negatif dari data *training*, berikut hasil pelabelan manual pada Gambar 3.4.

	Username	cleaned_text	label	kelas
0	Marmiati161	nikmatin hari libur nga aktivitas online indih...	positif	1
1	BangTAJIR8	surabaya jawa timur norak kamu	negatif	0
2	IndiHomeCare	samasama kak kan semoga informasi sampai manfa...	positif	1
3	IndiHomeCare	hai kak naura terima kasih apresiasi selamat n...	positif	1
4	IndiHomeCare	samasama kak cooky semoga informasi sampai man...	positif	1

Gambar 3.4 Hasil Pelabelan Manual

Berikut contoh *tweet* yang sudah dilabeli secara manual dapat dilihat pada Tabel 3.6.

Tabel 3.6 Pelabelan Secara Manual

No	*Tweets*	label	kelas
1	nikmatin hari libur ngga aktivitas online indihome dong	positif	1
2	samasama kak kan semoga informasi sampai manfaat selalu ikut timeline informasi tarik lain putar layanan produk indihome selamat aktivitas	positif	1
3	hai kak naura terima kasih apresiasi selamat nikmat layanan indihome semoga indihome selalu bantu jalan aktivitas kak stay safe stay healthy sama indihome	positif	1
4	samasama kak cooky semoga informasi sampai manfaat selalu ikut timeline informasi tarik lain putar layanan produk indihome selamat aktivitas	positif	1
5	samasama kak jauhari semoga informasi sampai manfaat selalu ikut timeline informasi tarik lain putar layanan produk indihome selamat aktivitas	positif	1
6	maaf kak wifi los hari kenapa	negatif	0
7	sabar banyak apa sama indihome	negatif	0
8	hari minggu saat olahraga sekarang olahraga rumah lho misal pingin senam cari instruktur via youtube banyak pilih internetnya pakai indihome	positif	1
9	halo min lampu indikator los nyala kedap kedip warna merah internet ga akses sama sekali selesai gimana	negatif	0
10	hallooooindihome tidak kelar kelar nih problem putus nyambung putus nyambung lampu pon kedip kedip	negatif	0

Pada Table 3.6 ditunjukan bahwa label positif diberi kelas 1 dan nilai kelas 0 untuk label negatif. Pelabelan manual dilakukan untuk perhitungan akurasi yang telah diberi sentimen posiitif dan negatif.

3.4.4 Training

Pada proses *training* diawali dengan fitur ekstraksi pada data teks menggunakan TF-IDF dan pada tahap ini menggunakan metode Naive Bayes Classifier, kemudian dilakukan data *training* untuk membuat model klasifikasi

dalam bentuk format *pickle* yang dimana dapat digunakan untuk melakukan *klasifikasi* sentimen secara otomatis.

Perhitungan TF-IDF secara manual dengan Microsoft Office Excel dapat dilihat pada Tabel 3.7.

Tabel 3.7 Dokumen TF-IDF

Dokumen (d)	**Kalimat**
d1	kerja dengerin lagu bikin mood naik banget tinggal sambungin wifi indihome beres dah semuanyaa.
d2	mantap indihome meski hujan deras sekal petir internet wifi kuliner rumah pohon tetap stabil lancar
d3	sudah waktu nya makan siang tani streaming youtube koneksi indihome lancar asik banget.
d4	selalu asik game onlinenya main game gini dukung koneksi lancar indihome biar gamenya lancar seru bikin happy.

Pada Tabel 3.7 terdapat beberapa kalimat yang dapat digunakan untuk perhitungan manual TF-IDF dengan 4 dokumen yaitu d1, d2, d3 dan d4. 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 kemunculan kata pada di setiap kalimat dapat dilihat pada Tabel 3.8.

Tabel 3.8 Perhitungan TF

***term* (kata)**	**d1**	**d2**	**d3**	**d4**	**df**
kerja	1				1
dengerin	1				1
lagu	1				1
bikin	1			1	2
mood	1				1

naik	1				1
banget	1		1		2
tinggal	1				1
wifi	1	1			2
indihome	1	1	1	1	4
beres	1				1
semua	1				1
mantap		1			1
meski		1			1
hujan		1			1
deras		1			1
sekali		1			1
petir		1			1
internet		1			1
kuliner		1			1
rumah		1			1
pohon		1			1
tetap		1			1
stabil		1			1
lancar		1	1	2	4
sudah			1		1
waktu			1		1
makan			1		1
siang			1		1

tani			1		1
streaming			1		1
youtube			1		1
koneksi			1		1
asik			1	1	1
selalu				1	1
game				3	3
online				1	1
gini				1	1
dukung				1	1
Biar				1	1
seru				1	1
happy				1	1
main				1	1

Pada Tabel 3.8 menjelaskan kemunculan *term* atau kata pada kalimat setiap dokumen. 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.

Contoh perhitungan IDF kemunculan kata dapat dilihat pada Tabel 3.9.

Tabel 3.9 Perhitungan IDF

term (kata)	df	idf	idf (N=4)	idf (N=1000)
kerja	1	1	0.60206	3
dengerin	1	1	0.60206	3
lagu	1	1	0.60206	3
bikin	2	0.5	0.30103	2.69897
mood	1	1	0.60206	3
naik	1	1	0.60206	3
banget	2	0.5	0.30103	2.69897

tinggal	1	1	0.60206	3
wifi	2	0.5	0.30103	2.69897
indihome	4	0.25	0	2.39794001
beres	1	1	0.60206	3
semua	1	1	0.60206	3
mantap	1	1	0.60206	3
meski	1	1	0.60206	3
hujan	1	1	0.60206	3
deras	1	1	0.60206	3
sekali	1	1	0.60206	3
petir	1	1	0.60206	3
internet	1	1	0.60206	3
kuliner	1	1	0.60206	3
rumah	1	1	0.60206	3
pohon	1	1	0.60206	3
tetap	1	1	0.60206	3
stabil	1	1	0.60206	3
lancar	4	0.25	0	2.39794001
sudah	1	1	0.60206	3
waktu	1	1	0.60206	3
makan	1	1	0.60206	3
siang	1	1	0.60206	3
tani	1	1	0.60206	3
streaming	1	1	0.60206	3
youtube	1	1	0.60206	3
koneksi	1	1	0.60206	3
asik	1	0.5	0.30103	2.69897
selalu	1	1	0.60206	3
game	3	0.333333	0.124939	2.52287875
online	1	1	0.60206	3
gini	1	1	0.60206	3
dukung	1	1	0.60206	3
Biar	1	1	0.60206	3
seru	1	1	0.60206	3
happy	1	1	0.60206	3
main	1	1	0.60206	3

Tabel 3.9 menunjukan mengenai perhitungan idf secara manual dengan rumus persamaan (1).

Berikut penjelasan mengenai perhitungan TF-IDF dapat dilihat pada Tabel 3.10.

Tabel 3.10 Perhitungan TF-IDF

term (kata)	d1	d2	d3	d4
kerja	0.60206			
dengerin	0.60206			
lagu	0.60206			
bikin	0.30103			0.30103
mood	0.60206			
naik	0.60206			
banget	0.30103		0.30103	
tinggal	0.60206			
wifi	0.30103	0.30103		
indihome	0	0	0	0
beres	0.60206			
semua	0.60206			
mantap		0.60206		
meski		0.60206		
hujan		0.60206		
deras		0.60206		
sekali		0.60206		
petir		0.60206		
internet		0.60206		
kuliner		0.60206		
rumah		0.60206		

pohon		0.60206		
tetap		0.60206		
stabil		0.60206		
lancar		0	0	0
sudah			0.60206	
waktu			0.60206	
makan			0.60206	
siang			0.60206	

Tabel 3.10 menjelaskan perhitungan TF-IDF secara manual dengan menggunakan Microsof Office Excel dari hasil perkalian TF dan IDF.

Klasifikasi yang dilakukan dalam penelitian ini menggunakan ekstraksi TF-IDF, yang secara otomatis menghasilkan perhitungan pada pembobotan kata dalam sebuah dokumen. 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*. Berikut kode dalam perhitungan TF-IDF didalam sistem terdapat dibawah ini dan contoh hasil perhitungan TF-IDF dari sistem dapat dilihat pada Gambar 3.5 dan Gambar 3.6.

```
s1 = "indihome jlek bener anjr kagak ngapa ngapain"
s2 = "hadeh indihome ganggu"

vect = TfidfVectorizer()
X = vect.fit_transform([s1, s2])

X.toarray()
```

```
[(0.3920440146223274, 'anjr'),
 (0.3920440146223274, 'bener'),
 (0.0, 'ganggu'),
 (0.0, 'hadeh'),
 (0.2789425453258252, 'indihome'),
 (0.3920440146223274, 'jlek'),
 (0.3920440146223274, 'kagak'),
 (0.3920440146223274, 'ngapa'),
 (0.3920440146223274, 'ngapain')]
```

Gambar 3.5 Hasil Perhitungan TF-IDF Pertama

```
[(0.0, 'anjr'),
 (0.0, 'bener'),
 (0.6316672017376245, 'ganggu'),
 (0.6316672017376245, 'hadeh'),
 (0.4494364165239821, 'indihome'),
 (0.0, 'jlek'),
 (0.0, 'kagak'),
 (0.0, 'ngapa'),
 (0.0, 'ngapain')]
```

Gambar 3.6 Hasil Perhitungan TF-IDF Kedua

Setelah melakukan perhitungan TF-IDF maka dilanjutkan untuk mencari nilai akurasi dari data *training* yang telah dilakukan pelabelan secara manual untuk mengetahui keakuratan data. Kode untuk mencari nilai akurasi data *training* dapat dilihat dibawah ini.

```
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))
```

Cross-validation merupakan sebuah metode untuk memperoleh hasil akurasi dengan percobaan sebanyak k kali dengan parameter yang sama, untuk menguji akurasi metode Naive Bayes agar diketahui akurasinya. 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. Berikut kode untuk menghitung cross validation terdapat dibawah ini.

```
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)
```

Pada Tahap ini merupakan langkah menghitung keakuratan pemodelan yang dibangun pada tahap *training* yang digunakan memprediksi label(kelas) dari data yang tersedia. Maka dilanjutkan dengan 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.

Dalam proses pembuatan model klasifikasi tersebut menggunakan *library* sklearn.pipeline dengan *import* pipeline yang berfungsi *testable* pada proses cross-validation, kemudian *import* pickle mempunyai fungsi untuk menyimpan dan membaca data ke dalam atau dari suatu file berformat .pkl. setelah mengimport sebuah *library* maka dapat membuat sebuah model klasifikasi, Berikut kode untuk pembuatan model klasifikasi dengan menggunakan nama variabel `text_classifier` terdapat dibawah ini.

```
X = df.cleaned_text
y = df.kelas

text_classifier = Pipeline([('vect', TfidfVectorizer()),
```

```
                    ('tfidf', TfidfTransformer()),
                    ('classifier', MultinomialNB(alpha=1.0)),
                    ])
X_train = np.asarray(X)
text_classifier = text_classifier.fit(X_train, np.asarray(y))
```

Model yang sudah diberi nama text_classifier disimpan dalam bentuk file .pickle sehingga dapat dibuka kembali dan digunakan kembali. Kode untuk penyimpanan file *pickle* adalah sebagai berikut.

```
files = open('model_klasifikasi_nbc.pickle', 'wb')
pickle.dump(text_classifier, files)
files.close()
```

Berikut kode untuk membuka file *pickle* yang sudah dibuat dapat dilihat dibawah ini.

```
model = open('model_klasifikasi_nbc.pickle', 'rb')
nbc_classifier = pickle.load(model)
print(nbc_classifier)
```

File pickle yang sudah dibuat model klasifikasi akan digunakan untuk eksekusi data *testing* yang dimana data yang digunakan adalah 200 *tweet* yang sudah dilakukan pelabelan secara manual dari data *training* yang berjumlah 1000 *tweet* dan 200 *tweet* yang digunakan data *testing* mengambil dari total data 10.000 *tweet*.

Pada langkah selanjutnya adalah untuk mengetahui hasil prediksi dari Naïve Bayes. Berikut kode pemanggilan untuk hasil prediksi dari Naïve Bayes terdapat dibawah ini.

```
predicted = nbc_classifier.predict(np.asarray(data_tweet))
predicted
```

Kemudian hasil prediksi dari Naïve Bayes akan ditampung ke dalam variabel `result_tweet` dalam bentuk data *list*. Berikut kode pemanggilan untuk menampung hasil prediksi dari Naïve Bayes terdapat dibawah ini.

```
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({'cleaned_text':data_tweet[i],
'class':predicted[i] })
```

3.4.5 Testing

Pada tahap *testing* adalah sebuah proses menentukan tingkat akurasi dari pemodelan yang dibuat pada tahap pelatihan, digunakan untuk memprediksi label atau kelas dari data uji yang tersedia. Maka ditampilkan hasil dari penggabungan pelabelan secara manual (*actual*) dan Naïve Bayes(*predicted*) terdapat pada Gambar 3.10.

	Text	actual	predicted
0	bikin video enak pakai internetnya indihome su...	1	1
1	sksks indihome biasa banget kalau abis ujannnn...	0	0
2	weekday makin semangat streaming karna indihom...	1	1
3	weekday makin semangat streaming karna indihom...	1	1
4	indihome kamu napasii anjirr jaringan putus	0	0
...	...	...	...
195	indihome lola	0	0
196	duh mana wifi indihome lagi dc lagi barusan	0	0
197	indihome kenapa sih	0	0
198	lengkap sudah siap banget nih awal hari senin ...	0	1
199	kenapa si mau bayar malah tulis bayar aplikasi...	0	0

Gambar 3.7 Hasil Pelabelan Manual dan Naive Bayes

Kemudian mencari nilai pemodelan data pada Gambar 3.7 meliputi *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *True Negative* (TN). Agar bisa dilakukan penghitungan nilai *testing* dari *accuracy*, *precision*, *recall* dan *f-measure*.

3.4.6 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*. Berikut hasil data sentimen dari keseliuruhan data dapat dilihat pada Gambar 3.8.

	cleaned_text	class
0	nikmatin hari libur nga aktivitas online indih...	1
1	surabaya jawa timur norak kamu	0
2	samasama kak kan semoga informasi sampai manfa...	1
3	hai kak naura terima kasih apresiasi selamat n...	1
4	samasama kak cooky semoga informasi sampai man...	1
...	...	...
9995	happy kamis manis temen jangan lupa awal basma...	1
9996	beli jajan buat sajen biar mata tetep melek mo...	1
9997	terima kasih kak nav semoga indihome selalu ba...	1
9998	kantor boring engga tahu mau ngapain ah buka y...	1
9999	metode wedding mau bikin apa inti kalao rumah ...	1

10000 rows × 2 columns

Gambar 3.8 Hasil Sentimen Data Keseluruhan