

BAB 3 METODE PENELITIAN

Penelitian ini adalah penelitian analisis sentimen ulasan Aplikasi Blibli dari Google Play Store. Penelitian ini menggunakan algoritma NBC serta melakukan proses pencarian data yang akan diolah untuk menghasilkan klasifikasi sentimen positif dan sentimen negatif. Berikut ini adalah bahan dan alat yang akan digunakan untuk melakukan analisis sentimen terhadap ulasan pengguna aplikasi Blibli di Google Play Store menggunakan metode *Naïve Bayes Classifier*.

3.1 BAHAN DAN ALAT PENELITIAN

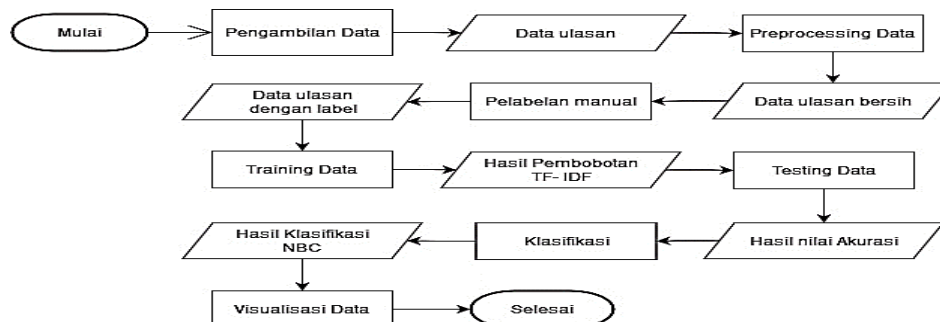
Pada penelitian ini menggunakan data ulasan pengguna aplikasi blibli pada Google Play Store sebanyak 1000 data.

Serta dalam prosesnya menggunakan beberapa alat yaitu laptop dengan spesifikasi yang cukup untuk menjalankan sistem operasi dan perangkat lunak pengembangan dan koneksi internet. Sistem Operasi dan program-program aplikasi yang digunakan dalam pembuatan sistem ini adalah:

1. Sistem Operasi: Windows 10 64 bit.
2. *Google Colaboratory*
3. Bahasa Pemrograman Python version 3.11.4
4. Visual Studio Code version 1.81.0

3.2 JALAN PENELITIAN

Adapun langkah-langkah dalam penelitian ini dapat dilihat pada Gambar 3.1



Gambar 3.1 Jalan Penelitian

Pada Gambar 3.1 menampilkan Jalan Penelitian pada penelitian ini dijelaskan sebagai berikut:

1. Pengambilan data

Pengambilan data merupakan tahapan pertama dalam penelitian ini, pada tahap ini akan dilakukan pengumpulan data yang diperoleh dari ulasan aplikasi Blibli di Google Play Store dengan teknik *Scraping* data menggunakan Google Colab.

2. *Preprocessing*

Pada tahap *preprocessing* ini, data yang telah didapatkan akan diolah untuk menghilangkan permasalahan pada data.

Pada bagian ini ada beberapa tahapan yang dilakukan :

- a. *Cleaning*

Digunakan untuk menghapus beberapa karakter dari huruf abjad yang tidak sesuai seperti menghapus *emoticon*, angka, *mention*, *url* dan tanda baca.

- b. *Tokenizing*

Merupakan proses untuk memisahkan kalimat menjadi beberapa bagian kata.

- c. *Case Folding*

Merupakan proses untuk mengubah semua huruf yang terdapat pada kalimat menjadi huruf kecil.

- d. *Stopword Removal*

Merupakan proses untuk menghilangkan kata-kata yang dianggap tidak berpengaruh yang terdapat pada kalimat.

- e. *Stemming*

Merupakan proses untuk mengubah kata-kata yang ada menjadi bentuk kata dasar.

3. Pelabelan Manual

Tahap ini untuk memberikan sentimen positif dan sentimen negatif terhadap kalimat yang terdapat dalam dokumen agar dapat dilakukan analisis.

4. Training Data

Tahap ini digunakan untuk pembobotan kata menggunakan TF-IDF bertujuan untuk mengubah data hasil *preprocessing* ditransformasi dari kata-kata menjadi angka dengan melakukan pembobotan untuk digunakan sebagai inputan *Naïve Bayes Classifier*.

5. Testing Data

Tahap ini digunakan untuk pengujian performa klasifikasi yang menghasilkan nilai akurasi, *precision*, *recall* dan *f1-score* menggunakan *Confusion Matrix*.

6. Klasifikasi

Tahap ini menggunakan algoritma NBC untuk melakukan klasifikasi yang akan menghasilkan probabilitas dari komentar pengguna yang diklasifikasikan ke negatif dan positif.

7. Visualisasi Data

Tahap untuk melakukan implementasi sistem yang akan ditampilkan dalam bentuk *dashboard*.

3.2.1 Pengambilan Data

Data yang digunakan dalam penelitian ini adalah data ulasan pengguna aplikasi Blibli yang didapatkan dari Google Play Store. Data yang didapatkan berjumlah 1000 data menggunakan teknik *scraping* pada Google Colab.

```
pip install google-play-scraper
```

Kode diatas merupakan library yang digunakan dalam proses pengambilan data ulasan. Setelah terinstall selanjutnya melakukan import library.

```
from google_play_scraper import Sort, reviews
from google_play_scraper import app
import pandas as pd
import numpy as np
```

Kode diatas berfungsi untuk mengimport library agar dapat digunakan pada tahap selanjutnya.

```
result, continuation_token = reviews(
    'blibli.mobile.commerce',
    lang='id',
    country='id',
    sort=Sort.MOST_RELEVANT,
    count=1000,
    filter_score_with= None
)
```

Kode diatas digunakan untuk melakukan pengambilan data ulasan. fungsi dari `filter_score_with= None` yaitu untuk mengatur berapa rating yang akan diambil dengan memasukkan angka, `None` untuk mengambil data ulasan pada semua rating.

```
data.to_csv("Ulasan Blibli.com 1000 Data.csv", index = False , encoding='utf-8')
```

Kode diatas digunakan untuk mengekspor dataset kedalam file csv. Hasil dari data ulasan yang telah berhasil diekspor dapat dilihat pada Gambar 3.2

	A	B	C	D	E	F	G	H	I	J	K	L
1	reviewId	userName	content	score	at							
2	dc0c993c-a46c-490f-b63f-f5185f3dea74	Eric Lin æž—	"Mantap, plg murah, cepat, juga aftersales service-nya bagus. Toped, shopee, apalagi Lazada, ga bisa menang. Plg baik untuk beli barang besar/berat, karena grati									
3	3e408b61-91e5-430f-8f9c-3f408e19bd60	Heri Priyanto,	"Aplikasinya ok, hanya perlu lebih ditingkatkan user interfacenya, terutama dim hal refresh rate dan jangan lupa tetap berikan penawaran cashback maupun dis									
4	aa4b1b6a-9e67-431e-817b-ae148745cc7d	Herman Halim,Barang yg di jual sangat mahal harganya. Sy adalah pecinta olah raga tenis meja. Harga karet bet dan kayu bet sy tau semua. Sangat jauh berbeda harganya dgn										
5	31893631-3668-498a-9c76-ca71692b50b	Yanti Lisnawati,	"Aku download apk ini gara2 dapat free voucher sunscreen seharga 35rb.. pas mau checkout dipersulit katanya hrs sesuai max nominal 35rb dan harus official s									
6	0015492d-27de-4898-9288-f4519094b9a	Muhammad Nur Aziz,	"Ribet banget, rugl jd seller disini lujur, mana masuk uang ke dompet seller lambat otomatis dan masuk ke saldo nya klo jumat jd snin, njr saldo gw ngen									
7	7d46ada6-76e9-42b0-11e2-9350f06c7d8f	simply cooking	tetep kasih bintang 5 walaupun sekarang pengiriman nya g secepat sebelum lebaran kemarin karena langganan tiap minggu.. biasanya order dibawah jam 9 ny									
8	1900d2b6-545a-4d98-926d-ae4cc3c36dd	vienadhi settawan,	terima kasih sudah bisa dan kembali normal untuk semua menu nya..5.2023-06-14 14:13:36									
9	55f0a111-ef13-4400-9c3f-025bc1c2bce	Ferdian Saputra,	"Semenjak belanja di Bilibli gak pernah ada kendala sih, aplikasi enteng bgt gak berat. Tapi kedepannya mungkin bisa ditingkatkan UI/UX nya, biar dilatnya lebi									
10	f3ac0d44-d32e-47d9-b2e2-c5210f9f1d19	Nurie Bandung,	"Kecewa lah... dapat Voucher bilibli 500.000 baru di pake beli produk 40 Rb, eee... udah ga bisa di pake lagi, katanya cuma sekali pake, padahal di fisik voucher r									
11	78f9ec67-6621-4585-acee-e68fbd70c34	Insa Allah Berkah,	"Sebagai pengguna baru saya kecewa dengan pelayanan oknum seler yg tdk profesional,pesanan dari tanggal 20 mei 2023 hingga ulasan tertulis belum ada p									
12	e46d200c-7894-4c18-a876-f954b90fcb0	snz rfm,	"kecewa sih sekarang akses bilibli lewat aplikasi gak bisa karena dengan alasan koneksi internet saya lewat vpn, padahal sebelum bilibli versi terbaru gak rewel walaup									
13	9de9d88e-4be8-43d9-9d24-2edc01f5c917	Zaujatu Amar,	"Biasanya saya belanja bulanan di bilibli, karna meski harga g murah2 bgt tp gratis ongkir full. jd masih lebih hemat dn g capek dbanding ngube2 mp lain. Tp ker									
14	367dd97e-a3be-42a0-90d1-d69ea7a2052d	Ati Nur,	"Setelah muter muter nyari barang yg di inginkan tidak ada di aplikasi langganan saya belanja, saya iseng iseng buka bilibli dan dapatkan barang yg saya butuhkan, lan									
15	beb80069-9a17-4fd1-bc83-0ee5e5aa0ca2	Abdul Rauf,	"Awalnya saya mau belanja di bil2. karena dapat e voucher dari salah satu bank BUMN. Setelah install pendaftaran dll. Applikasi terlihat friendly, akhirnya saya be									
16	6a2b301e-1045-499b-b4e4-dbc0af07a87c	Fadlan Farabi,	"Bnyk problem aplikasinya. Buka notif tp ngkk buka aplikasinya (malah muncul pilihan default Chrome Browser atau Bilibli), tekan tombol back berkali-kali tel									
17	c5fe8e54-9b63-43ef-9187-d522bce2a551	Yayang Supandi,	Susah kalau make aplikasi ini.. Kalau akun kita pindah ke hape baru.. produk yg sangat terbatas jika dibandingkan dengan e-commerce sebelah mungkin perbandingannya 1:1000"									
18	b4fe6236-3484-46ca-9ee8-e27797a35e50	Erlly zuzu,	Penyelesaian masalah terlalu bertele tele ..lama ... Ga pasti!!! Intiny...saya d anjurkan ign byr transaksi yg gagal dijadikan cicilan pdhl kita buat cicilan sehangka kartu									
19	c30c62be-e069-49b7-a5b5-fe1dfcd45245	Rina Mayasari,	"Aplikasi terthebest desain ui/ux yg sangat buruk, produk yg sangat terbatas jika dibandingkan dengan e-commerce sebelah mungkin perbandingannya 1:1000"									
20	ae9fe603-07e2-434c-ac34-5862a0d16e5b	Fajar Bagasty Akbar,	"Beli barang tidak berfungsi, mengajukan retur dana malah tidak di refund, malah di kembalikan ke cashback, sangat tidak menyenangkan sekali. Parah!,"									
21	89472b33-f835-409a-9f63-84f7f2265e6c	Hans Hermawan,	Gak niat jualan. Dapat email katanya ada vocer. Dipakai belanja malah dibatalin. Padahal krmrin2 belanja brg yg sama juga dikirim dan berhasil. Ini mau belan									
22	54c1ce97-f9b5-4c2e-b0c3-2ff0d26cea77	Vlaminora Inke Callista,	"Ini sebenarnya apa sih yang sedang diperbaiki. ? Dari tadi malam jam 11 lho sampe skrg buka aplikasi mau belanja keterangannya ""server lagi dalam pi									
23	a80c06ad-et1d74c32-8cd4-c791f8ed2316	Ricky A,	"setiap buka pesan, loadingnya lama banget sampek beberapa menit dan itupun ngga jaminan bisa kebuka. susah mau chat sama seller karena buka pesannya saja tid									

Gambar 3.2 Data Ulasan

Pada Gambar 3.2 merupakan data ulasan yang telah berhasil diambil. Selanjutnya tahap *preprocessing* untuk membersihkan data agar menjadi data yang terstruktur.

3.2.2 Preprocessing

Tujuan dari tahap ini adalah untuk menghilangkan permasalahan pada data yang akan mengganggu hasil dari proses data. Berikut adalah urutan serta kode program dari tahapan *preprocessing* yang dilakukan.

3.2.2.1 Cleaning

Pada tahap ini digunakan untuk menghapus *emoji*, *hashtag*, *mention*, *url* dan tanda baca.

```
def emoji_remove(ulasan):
    if not ulasan:
        return ''
    t = ulasan.encode('ascii', 'ignore').decode('utf-8')
    return t
```

Kode diatas digunakan untuk melakukan untuk menghapus *emoticon* yang terdapat pada dokumen.

```
def url_remove(ulasan):
    if not ulasan:
        return ''
    t = re.sub(r'http\S+', '', ulasan)
    return t
```

Kode diatas digunakan untuk menghapus *url* yang terdapat pada dokumen.

```
def hashtag_remove(ulasan):
    if not ulasan:
        return ''
    reg = r'#\S+'
    t = re.sub(reg, '', ulasan)
    return t
```

Kode diatas digunakan untuk menghapus *hashtag* yang terdapat pada dokumen.

```
def remove_user(ulasan):
    if not ulasan:
        return ''
    t = re.sub(r'@[^\s]+', '', ulasan)
    return t
```

Kode diatas digunakan untuk menghapus *mention* yang terdapat pada dokumen.

```
def punc_remove(ulasan):
    if not ulasan:
        return ''
    t = re.sub(r'[\^\w\s]+', '', ulasan)
    return t
```

Kode diatas digunakan untuk menghapus tanda baca yang terdpat pada dokumen.
Hasil dari *cleaning* data dapat dilihat pada Tabel 3.1

Tabel 3.1 Data Hasil *Cleaning*

No	Sebelum	Sesudah
0	Mantap, plg murah, cepat, juga aftersales serv...	Mantap plg murah cepat juga aftersales service...
1	Aplikasinya ok, hanya perlu lebih ditingkatkan...	Aplikasinya ok hanya perlu lebih ditingkatkan ...
2	Barang yg di jual sangat mahal harganya. Sy ad...	Barang yg di jual sangat mahal harganya Sy ada...
3	Aku download apk ini gara2 dapet free voucher ...	Aku download apk ini gara dapet free voucher s...
4	Ribet banget, rugi jd seller dsini jujur, mana...	Ribet banget rugi jd seller dsini jujur mana m...
5	tetep kasih bintang 5 walaupun sekarang pengir...	tetep kasih bintang walaupun sekarang pengiri...
6	terima kasih sudah bisa dan kembali normal unt...	terima kasih sudah bisa dan kembali normal unt...
7	Semenjak belanja di Blibli gak pernah ada kend...	Semenjak belanja di Blibli gak pernah ada kend...
8	Kecewa lah,.. dapat Voucher blibli 500.000 bar...	Kecewa lah dapat Voucher blibli baru di pake ...
9	Sebagai pengguna baru saya kecewa dengan pelay...	Sebagai pengguna baru saya kecewa dengan pelay...

Pada Tabel 3.1 menampilkan hasul dari ulasan yang telah bersih dari *emoji*, *hashtag*, *mention*, *url* dan tanda baca.

3.2.2.2 Tokenizing

Pada tahap ini memecah kalimat menjadi bagian-bagian kata. Berikut kode untuk melakukan *tokenizing*.

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

tokenized = []

def tokenize_content(ulasan):
    for ulasan in cleaned:
```

```
tokens = word_tokenize(ulasan)
tokenized.append(tokens)
```

Kode diatas digunakan untuk melakukan tokenisasi teks menggunakan library Natural Language Toolkit (NLTK) dengan memecah kalimat menjadi per-kata. Hasil dari tokenizing dapat dilihat pada Tabel 3.2

Tabel 3.2 Data hasil *tokenizing*

No	Sebelum	Sesudah
0	Mantap plg murah cepat juga aftersales service...	[Mantap, plg, murah, cepat, juga, aftersales, ...
1	Aplikasinya ok hanya perlu lebih ditingkatkan ...	[Aplikasinya, ok, hanya, perlu, lebih, ditingk...
2	Barang yg di jual sangat mahal harganya Sy ada...	[Barang, yg, di, jual, sangat, mahal, harganya...
3	Aku download apk ini gara dapet free voucher s...	[Aku, download, apk, ini, gara, dapet, free, v...
4	Ribet banget rugi jd seller dsini jujur mana m...	[Ribet, banget, rugi, jd, seller, dsini, jujur...
5	tetep kasih bintang walaupun sekarang pengiri...	[tetep, kasih, bintang, walaupun, sekarang, pe...
6	terima kasih sudah bisa dan kembali normal unt...	[terima, kasih, sudah, bisa, dan, kembali, nor...
7	Semenjak belanja di Blibli gak pernah ada kend...	[Semenjak, belanja, di, Blibli, gak, pernah, a...
8	Kecewa lah dapat Voucher blibli baru di pake ...	[Kecewa, lah, dapat, Voucher, blibli, baru, di...
9	Sebagai pengguna baru saya kecewa dengan pelay...	[Sebagai, pengguna, baru, saya, kecewa, dengan...

Pada Tabel 3.2 menampilkan data hasil proses *tokenizing* yaitu dengan memecah kalimat menjadi per-kata.

3.2.2.3 Case Folding

Case folding bertujuan untuk menyeragamkan karakter pada data. Proses dari *case folding* yaitu mengubah seluruh huruf menjadi huruf kecil. Berikut kode program untuk melakukan tahapan *case folding*.

```
def lowercase(content):
```

```

lower_word = []
for tokens in tokenized:
    text = ' '.join(tokens)
    text = text.lower()
    lower_word.append(text)
return lower_word

tokenized_data = tokenize_content(data["ulasan"])
lower_data = lowercase(tokenized_data)
lower_data=pd.Series(lower_data).drop_duplicates().astype(object)
print(lower_data)

```

Kode diatas digunakan untuk melakukan tahap *case folding* yang bertujuan untuk menghasilkan data ulasan dalam bentuk yang telah diubah menjadi huruf kecil, menghapus duplcat dan dicetak menggunakan objek Pandas Series. Hasil dari tahapan diatas dapat dilihat pada Tabel 3.3

Tabel 3.3 Data hasil *case folding*

0	mantap plg murah cepat juga aftersales service...
1	aplikasinya ok hanya perlu lebih ditingkatkan ...
2	barang yg di jual sangat mahal harganya sy ada...
3	aku download apk ini gara dapet free voucher s...
4	ribet banget rugi jd seller dsini jujur mana m...
	...
995	aplikasi bagus banget gila barang nya murah se...
996	hati dgn blibli paylater saya pernah beli hard...
997	saya kasih full bintang karena barang yg saya ...
998	pesanan status terkirim produk belum saya teri...
999	kuarang nyaman produk yang di jual itu itu saj...

Pada Tabel 3.3 menampilkan data hasil proses *case folding* yaitu dengan mengubah semua data menjadi huruf kecil.

3.2.2.4 Stopword Removal

Stopword Removal digunakan untuk menghapus beberapa kata yang tidak perlu.

```
pip install sastrawi
```

Kode di atas merupakan instalasi library Sastrawi yang akan digunakan untuk melakukan tahapan *stopword removal*. Berikut adalah kode untuk melakukan pemanggilan library sastrawi.


```

from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory

factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
stopwords = factory.get_stop_words()
stopwords_str = ", ".join([f"'{word}'" for word in
stopwords])
print(stopwords_str)

```

Kode di atas menggunakan library Sastrawi yang digunakan untuk menampilkan kata-kata yang tidak memiliki makna yang penting. Hasil dari proses tersebut dapat dilihat pada Tabel 3.4

Tabel 3.4 Data hasil *stopword removal*

<pre> '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', 'nant', 'melainkan', 'oh', 'ok', 'seharusnya', 'sebetulnya', 'setiap', 'setidaknya', 'sesuatu', 'pasti', 'saja', 'toh', 'ya', 'walau', 'tolong', 'tentu', 'amat', 'apalagi', 'bagaimanapun' </pre>
--

Pada Tabel 3.4 menampilkan kata yang akan dihilangkan dari data ulasan. Berikut kode yang digunakan untuk menghapus kata-kata yang tidak memiliki makna dari setiap ulasan dalam objek Pandas Series yang telah diubah menjadi huruf kecil.

```

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

stopwords_ulasan = lower_data.apply(removeStopWords)
print(stopwords_ulasan)

```

Kode diatas digunakan untuk melakukan *stopword removal* pada data ulasan.

3.2.2.5 Stemming

Stemming digunakan untuk mengubah kata tambahan menjadi kata dasar. Berikut kode program untuk melakukan proses *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_ulasan:
    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_ulasan = stopwords_ulasan.apply(get_stemmed_term)
print(stem_ulasan)

```

Kode diatas menggunakan library Sastrawi untuk melakukan *stemming* pada dokumen berfungsi mengubah kata tambahan menjadi bentuk kata dasar dengan menghapus kata impuhan di depan maupun imbuhan di belakangnya.

3.2.3 Pelabelan Manual

Pada tahapan pelabelan data ini dilakukan secara manual untuk menghasilkan *sentimen positif* dan *sentimen negatif*.

Data ulasan dari pelabelan manual dapat dilihat pada Tabel 3.5

Tabel 3.5 Hasil pelabelan manual

No	Kelas	Ulasan
1	Positif	mantap plg murah cepat aftersales servicenya bagus oped hopee lazada ga menang plg baik beli barang besarberat gratis ongkirnya ampe rb mantap
2	Positif	aplikasi perlu pebih tingg user interfacenya utama dlm refresh rate jangan lupa tetap ikan tawar cashback maupun diskon belanja terima kasih blibli
3	Negatif	barang yg jual sangat mahal harga sy cinta olah raga tenis meja harga karet bet kayu bet sy tau semua sangat jauh beda harga dgn aplikasi onlineshop lain
4	Negatif	aku download apk gara dapet free voucher sunscreen harga rb pas mau checkout sulit kata hrs sesuai max nominal rb official store semua udah sesuai tp ttp sulit akhir aku coba lg besok nya malah keluar tulis kalo voucher nya pakai pakai mane org kemarin aja gagal mulu padahal semua udh sesuai sm termcondition sistem km baaaanyak yg hrs baik aku uninstall bye
5	Negatif	ribet banget rugi jd seller dsini jujur mana masuk uang dompet seller lambat otomatis masuk saldo nya klo jumat jd snin njir saldo gw ngendep distu ecommerce paling ribet gak bakal gw pake app rugi seller nyesel banget

Pada Tabel 3.5 menampilkan hasil pelabelan secara manual. Pada tahap pelabelan ini dari jumlah data ulasan sebanyak 1000 data didapat dapat ulasan sebanyak 600 dengan rincian 300 data ulasan bernilai positif dan 300 data ulasan bernilai negatif. Data tersebut nantinya akan digunakan untuk tahapan training data.

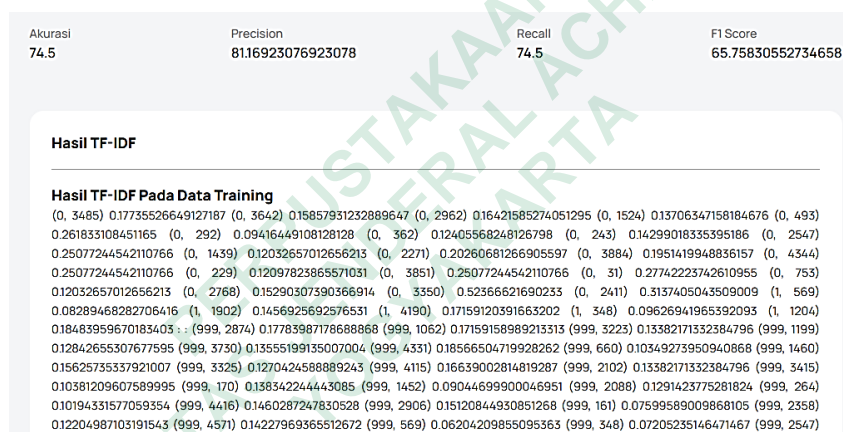
3.2.4 Training Data

Training data merupakan proses dalam melatih data menggunakan algoritma NBC. Proses pelatihan ini dimulai dengan ekstraksi data menggunakan TF-IDF. setelah itu, dilakukan pelatihan data untuk menciptakan model klasifikasi yang dapat digunakan secara otomatis untuk analisis sentimen. Berikut ini

merupakan kode program untuk mendapatkan hasil TF-IDF dari training dan testing.

```
from sklearn.model_selection import train_test_split
text = pd.read_excel('files/hasil_pelabelan.xlsx')
text.dropna()
text = datas['ulasan']
label = datas['label']
X_train, X_test, Y_train, Y_test = train_test_split(text,
label, test_size=0.2, random_state=42)
```

Kode diatas digunakan untuk membagi data ulasan dan label ke dalam set pelatihan dan pengujian menggunakan modul ‘**train_test_split**’ dari library Scikit-learn(sklearn). Hasil TF-IDF data training dan testing dapat dilihat pada Gambar 3.3 sebagai berikut :



Gambar 3.3 Hasil TF-IDF

Gambar 3.3 menampilkan hasil dari TF-IDF pada data training dan hasil TF-IDF pada data testing.

Berikut ini adalah kode untuk melakukan pemrosesan teks, pembentukan model dan evaluasi model klasifikasi teks serta untuk menghitung nilai akurasi, precision, recall dan f1-score untuk data training.

```
vectorizer = TfidfVectorizer(analyzer="word", min_df=0.0004,
max_df=0.115, ngram_range=(1, 3))
X_train_idf = vectorizer.fit_transform(datas['ulasan'])
X_test_idf = vectorizer.transform(X_test)
vectorizer.fit(X_train)
X_train_dtm = vectorizer.transform(X_train)
X_test_dtm = vectorizer.transform(X_test)
nbmodel = MultinomialNB(alpha=0.1)
```

```

nbmodel = nbmodel.fit(X_train_dtm, Y_train)
Y_pred = nbmodel.predict(X_test_dtm)

akurasi = accuracy_score(Y_test, Y_pred) * 100
f1score = f1_score(Y_test, Y_pred, average='weighted') *
100
precision = precision_score(Y_test, Y_pred,
average='weighted') * 100
recall = recall_score(Y_test, Y_pred, average='weighted') * 100
if os.path.exists('files/ta.pickle') :
    os.remove('files/ta.pickle')
file_data = pickle.dump(pipeline, open('files/ta.pickle',
'wb'))

```

3.2.5 Testing Data

Testing data merupakan tahapan untuk menguji performa model yang telah dilatih pada data pelatihan. Data testing adalah data yang tidak digunakan selama proses pelatihan model dan digunakan untuk mengevaluasi seberapa baik model dapat melakukan prediksi pada data yang belum pernah dilihat sebelumnya.

```

import pickle
vect = pickle.load(open('files/ta.pickle', 'rb'))
text = pd.read_excel('files/hasil_pelabelan.xlsx')
text.dropna()
text = text['ulasan'].fillna(' ')
prediction = vect.predict(text)
result = []
sentiment = 'Positif'
for i in range(len(prediction)):
    if (prediction[i] == 1):
        sentiment = 'Positif'
    elif (prediction[i] == 0):
        sentiment = 'Negatif'
    result.append({'ulasan': text[i], 'label': prediction[i],
        'kelas': sentiment})

```

Kode diatas untuk melakukan prediksi sentimen teks pada data ulasan menggunakan model yang telah disimpan dalam file pickle.

```

X = akurasi['ulasan']
Y = akurasi['label']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=42)

vect = TfidfVectorizer(analyzer="word", min_df=0.0004,
max_df=0.115, ngram_range=(1, 3))
vect.fit(X_train)
X_train_dtm = vect.transform(X_train)
X_test_dtm = vect.transform(X_test)

```

```

nbmodel = MultinomialNB(alpha=0.1)
nbmodel = nbmodel.fit(X_train_dtm, Y_train)
Y_pred = nbmodel.predict(X_test_dtm)
akurasi = accuracy_score(Y_test, Y_pred) * 100
measure = f1_score(Y_test, Y_pred, average='weighted') *100
presision = precision_score(Y_test, Y_pred,
                             average='weighted') * 100
recall = recall_score(Y_test, Y_pred, average='weighted')
*100

```

Kode diatas untuk melakukan klasifikasi menggunakan model Niave Bayes Multinomial dan menghitung nilai akurasi, precission, recall dan f1-score pada data testing. Dari kode untuk melakukan testing data diatas hasil yang didapat sebagai berikut:



The screenshot displays the following metrics:

- Akurasi: 77.5
- Precision: 77.43589743589745
- Recall: 77.5
- F1 Score: 76.99797160243408

Below the metrics is a table titled "Hasil Testing" with the following data:

Nomor	Label	Kelas	Ulasan
1	0	Negatif	muas aplikasi jual beli paling bagus customer service bibi bagus responsive beli butuh hari murah bingit semua gratis on
2	1	Positif	thanks bibi belanja bulan ga mesti ribet cukup dirumah ketik psn besok dtng deh bayar jg banyak pilihan barang selalu ses
3	1	Positif	sangat puas bibi kerja orang gudang cepat kirim barang kilat sampai kurir sangat cepat antar barang lebih gratis ongkos
4	0	Negatif	yg buat malas bibi sering update aplikasi wajib masuk beda dg aplikasi belanja spesifikasi barang ambarudul tdk dulu awa
5	1	Positif	suka sekali beli aplikasi bibi udah barang ori layan oke cs cepat tanggap kendala salah langsung direspon by phone sm pi
6	1	Positif	bibi memang wokeh harga barang nya saing callcenter ramah no ongkir antara ka'au saran kuota promo nya kasih acak p
7	0	Negatif	sejak update baru ki mau isi alamat pakai gps provinsi kota pilih akhir ga input alamat mohon baik seperti bug aplikasi bar.

Gambar 3.4 Hasil testing data

Gambar 3.4 menampilkan hasil testing berupa Akurasi, Precision, Recall dan F1-Score.

3.2.6 Klasifikasi

Tahap klasifikasi ini menampilkan hasil perhitungan data menggunakan algoritma NBC. Data yang akan digunakan yaitu data bersih setelah melewati tahapan *preprocessing* sebanyak 1000 data. Kode untuk menampilkan hasil dari klasifikasi sentimen. Berikut kode programnya:

```

text = pd.read_excel('files/coba_data_keseluruhan.xlsx')
text.dropna()
text = text['Ulasan'].fillna(' ')

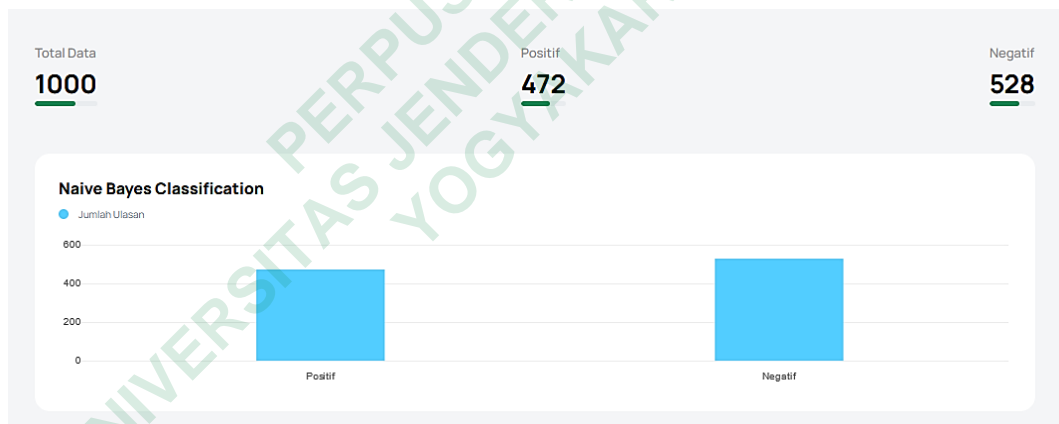
```

```

prediction = vect.predict(text)
result = []
for i in range(len(prediction)):
    if (prediction[i] == 1):
        sentiment = 'Positif'
    elif (prediction[i] == 0):
        sentiment = 'Negatif'
    result.append({'Ulasan': text[i], 'Kelas':
                  sentiment, 'Label': prediction[i]})
text = pd.DataFrame(result)
text = text.dropna()
if os.path.exists('files/hasil_data_keseluruhan.xlsx'):
    os.remove('files/hasil_data_keseluruhan.xlsx')
text.to_excel('files/hasil_data_keseluruhan.xlsx')
hasil =
pd.read_excel('files/hasil_data_keseluruhan.xlsx')
hasil.drop(['Unnamed: 0'], axis=1, inplace=True)
return render_template('klasifikasi.html', datas=hasil)

```

Kode diatas digunakan untuk melakukan prediksi sentimen teks pada data ulasan yang berasal dari file excel menggunakan model yang telah dilatih sebelumnya. Berikut adalah hasil dari proses klasifikasi.



Gambar 3.3 Grafik hasil Klasifikasi