BAB 3
METODE PENELITIAN

Penelitian yang diajukan adalah analisis sentimen menggunakan data dari ulasan aplikasi Tiktok pada Google Play Store. Penelitian ini menggunakan metode Naïve Bayes Classification. Penelitian dimulai dengan latar belakang yang ada, mencari sumber masalah, dan menganalisis data mengenai ulasan pengguna Tiktok di Google Play Store untuk mengatasi permasalahan yang ada. Penelitian ini membutuhkan data ulasan aplikasi Tiktok yang diperoleh dari Google Play Store. Di bawah ini bahan, alat, metode, dan jalannya penelitian analisis sentimen tentang ulasan aplikasi Tiktok serta langkah-langkah untuk menyelesaikan penelitian analisis sentimen menggunakan data ulasan aplikasi Tiktok.

3.1 BAHAN DAN ALAT PENELITIAN

Bahan penelitian berisi bahan-bahan yang digunakan untuk menciptakan solusi. Sumber utama penelitian ini diambil dari ulasan aplikasi Tiktok pada Google Play Store. Data yang didapat mengunakan teknik *scrapping* dengan Google Colab. Selain itu referensi lain yang terkait dalam penelitian ini diperoleh dari literatur yang relevan seperti jurnal, penelitian sebelumnya.

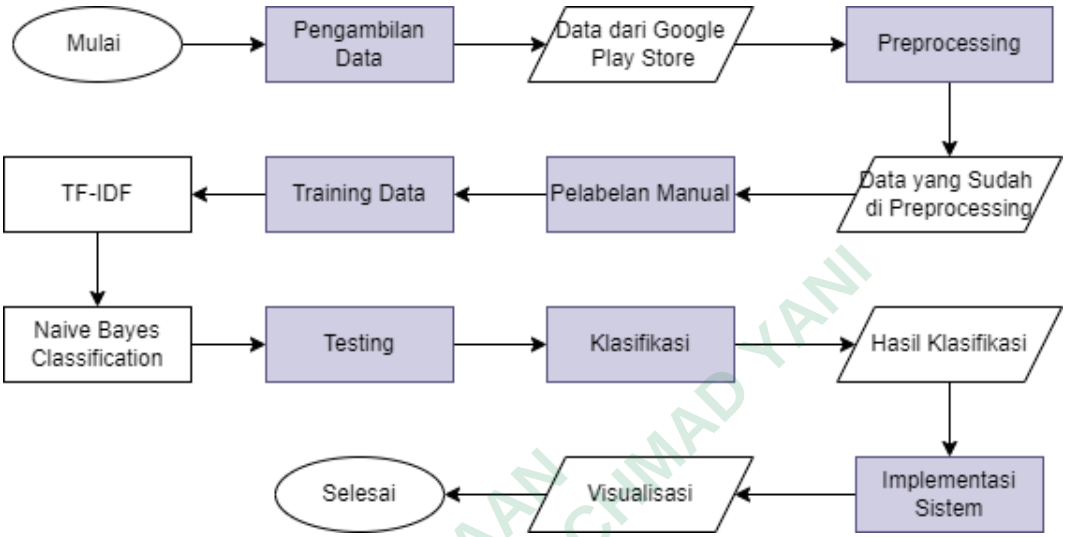
Sistem Operasi dan program-program aplikasi yang dipergunakan dalam dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi : Windows 10
2. Bahasa pemrograman : Python
3. Framework : Flask
4. Text editor : Sublime Text, Google Colab.
5. *Library* : Sastrawi, NLTK, Sklearn

3.2 JALAN PENELITIAN

Naïve Bayes Classification merupakan algoritma klasifikasi yang digunakan untuk melakukan prediksi terhadap probabilitas keanggotaan suatu class dengan menerapkan teorema Bayes (Adnyana, 2020). Alur penelitian

metode Naïve Bayes Classification yang akan digunakan pada penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

1. Tahap pengambilan data, tahap ini melakukan pengambilan di Google Play Store pada ulasan aplikasi Tiktok mengenai kepuasan pengguna. Proses ini dilakukan menggunakan Google Colab, data yang di didapat berformat CSV sehingga dapat di buka di Microsoft Excel.
2. Tahap *preprocessing* data, tahap ini dilakukan proses pengolahan data teks untuk data yang belum terstruktur, sehingga perlu diperbaiki dengan melakukan tahapan-tahapan berikut ini:
   a. *Text cleaning* merupakan proses untuk membersihkan data mentah dari informasi yang tidak relevan.
   b. *Case folding* merupakan proses untuk mengubah setiap huruf pada ulasan menjadi *lowercase* atau huruf kecil.
   c. *Tokenization* merupakan proses untuk pemecahan atau pemisahan kalimat dalam suatu teks yang disebut token.
   d. *Stopword Removal* merupakan tahapan untuk menghapus kata-kata yang tidak memiliki makna penting.
   e. *Stemming* merupakan penghilang infeksi pada kata menjadi bentuk dasarnya

3. Tahap Pelabelan Manual, tahap ini dilakukan proses pelabelan manual untuk memberikan sentimen negatif dan positif terhadap kata yang ada pada dokumen agar dapat dianalisis.
4. Tahap *training* data, tahap ini dilakukan proses *training* data yang di awali dengan ekstraksi menggunaakn TF-IDF pada data teks. Proses *training* data digunakan untuk melakukan klasifikasi sentimen secara otomatis.
5. Tahap *testing.* Tahap ini dilakukan untuk mengukur dan menentukan seberapa akurat model yang telah dibangun, hasil *testing* dihitung dengan rumus *Confusion Matrix* untuk mengetahui persentase dari setiap prediksi pengujian kelas dan label.
6. Tahap Klasifikasi, tahap ini merupakan proses dimana hasil data yang sudah diprediksi kelas dan labelnya secara otomatis akan ditampilkan dalam bentuk *dashboard* menggunakan *framework* Flask.
7. Tahap implementasi sistem, tahap ini merupakan visualisasi yang akan ditampilkan berdasarkan data uji menggunakan metode NBC yang menghasilkan klasifikasi dengan kategori positif dan negatif. Hasil pengklasifikasian selanjutnya akan ditampilkan dalam bentuk *dashboard.*

3.2.1 Pengambilan Data

Pengambilan data dilakukan menggunakan Google Colab dengan melakukan install *package* google-play-scrape. Data yang diambil merupakan ulasan yang ada pada aplikasi Tiktok. Data ulasan diambil pada tanggal 7 Juli 2023 sampai dengan 9 Juli 2023

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

Kode di atas merupakan *library* untuk mengambil data ulasan pada Google Play Store. Setelah terunduh, maka tahap selanjutnya yaitu melakukan pemanggilan *library.*

```
from google_play_scraper import app
import pandas as pd
```

```
import numpy as np
```

Kode di atas digunakan untuk memanggil *library* dengan cara mengimpor *package* yang dibutuhkan. Pada tahap selanjutnya dilakukan *scrapping* data.

```
result, continuation_token = reviews(
	'com.ss.android.ugc.trill',
	lang='id',
	country='id',
	sort=Sort.NEWEST,
	count=5000,
	filter_score_with=None)
```

Kode di atas digunakan untuk melakukan pengambilan data ulasan, *lang* dan *country* digunakan untuk mengambil data dengan bahasa dan negara Indonesia (id). *Sort.NEWEST* digunakan untuk mengambil data ulasan yang terbaru. Fungsi *count* digunakan untuk mengatur jumlah data yang diambil sedangkan filter_score_with = None digunakan untuk mengambil data ulasan pada semua rating. Berikut kode program untuk menyimpan data ulasan ke dalam csv.

```
my_df.to_csv("baru.csv", index = False)
```

Kode di atas digunakan untuk ekspor data ulasan ke dalam file csv. Hasil data ulasan yang telah diekspor, dapat dilihat pada Tabel 3.1.

Tabel 3.1 Contoh Data Hasil *Scraping*

No	Data Ulasan
1.	Sangat bagus dan mengasik kan dan mendidik
2.	Apk nya Cocok buat jomblo kayak kamu
3.	Aplikasi ini bagus karena sangat menghibur
4.	Kamera buat ftonya dikembaliin ke dulu dong jelekk jadinya gtu mahðŸ˜-ðŸ˜ðŸ˜
5.	Banyak Terimakasih. Baru mulai menjadi afiliator di tiktok Semoga sukses
6.	Banyak eror tman saya udah nonton tiap hari even tidak maju maju even

No	Data Ulasan
	nya cuma bohong belaka.
7.	Assalamualaikum
8.	Sya ingin bisa live kenapa sya saat ingin live malah kena pelanggran hah? Aku cmn mau live kayk orangÂ² itu masalah yg mudah kan aku ingin live, tolong tiktok aku ingin masalah/kita live itu tenang apa gk bisa gitu hah aku baru pertama kali live malah kena pelanggaran andai kamu tau aku saat udh bisa live seneng bgt tapi ujung2 nya kamu malah melanggar sya emng ya di disitu ada rasa senang tiba2 orang menghancurkan kesenangan kita sya dari akun @l0velyn_bianra tidak menerima ini semua
9.	Tolong fyp in vidio saya kak. ðŸ™,
10.	Good apk

Pada Tabel 3.1 menunjukkan bahwa data ulasan telah diambil dan masih banyak komponen yang tidak penting dalam proses klasifikasi sentimen. Komponen yang tidak diperlukan harus melalui proses *preprocessing* untuk menghasilkan data yang terstruktur.

3.2.2 Preprocessing

Preprocessing merupakan teknik yang dilakukan untuk menyiapkan data agar lebih mudah dipahami. Berikut kode *library* yang diperlukan untuk *preprocessing.*

```
from nltk.tokenize import word_tokenize
import pandas as pd
import matplotlib.pyplot as plt
import numpy
import re
import nltk
import emoji
```

Kode di atas menampilan *library* untuk melakukan *preprocessing* data. *Library* pandas digunakan untuk memproses data menjadi terstruktur, numpyadalah *library* yang bekerja dengan *array,* nltk (natural language tool kit) berfungsi untuk memudahkan dalam memproses teks yang nanti dilakukan pada tahap *stopword removal* dan *stemming*, *library* re digunakan untuk

mendefinisikan sebuah pola pencarian pada proses *text cleaning*, *library* emoji digunakan untuk mengelola teks yang mengandung karakter emoji pada proses *text cleaning.*

1. Case Folding

Case folding merupakan proses yang dilakukan untuk menyeragaman karakter huruf pada data. Perubahan yang dilakukan dengan mengubah seluruh huruf menjadi huruf kecil. Berikut kode untuk melakukan tahapan *case folding*.

```
def lowercase():
  lower_word = df_data['content'].str.lower()
  return lower_word

lower_ulasan = lowercase()
```

Kode di atas digunakan untuk menampilkan tahapan *case folding.* Fungsi *lowercase* digunakan untuk mengubah huruf besar menjadi huruf kecil. Hal ini bertujuan untuk memudahkan dalam membandingkan kata.

2. Text Cleaning

Text cleaning merupakan proses membersihkan ulasan dari kata-kata yang tidak diperlukan untuk mengurangi proses noise pada proses klasifikasi. Berikut kode untuk *text cleaning*.

```
def clearEmoji(ulasan):
    return ulasan.encode('ascii', 'ignore').decode('ascii')
df_data['content']= df_data['content'].apply(clearEmoji)


def cleaningUlasan(ulasan):
  ulasan = ulasan.strip(' ')
  ulasan = re.sub(r'@[A-Za-a0-9]+',' ',ulasan)
  ulasan = re.sub(r'#[A-Za-z0-9]+',' ',ulasan)
  ulasan = re.sub(r"http\S+",' ',ulasan)
  ulasan = re.sub(r'[0-9]+',' ',ulasan)
  ulasan = re.sub(r"[-()\"#/@;:<>{}'+=~|.!?,_]", " ", ulasan)
  ulasan = re.sub(r'\(cont\)'," ", ulasan)
  ulasan = re.sub(r'#([^\s]+)', '', ulasan)
  ulasan = re.sub(r'\d+', ' ', ulasan)
```

```
        ulasan = re.sub('[!"”#$%&’()*+,-./:;<=>?@[\]^_`{|}~]', ' ',
ulasan)
       ulasan = re.sub(r'''(?i)\b((?:https?://|www\d{0,3}[.]|[a-z0-
9.\-]+[.][a-z]{2,4}/)(?:[^\s()<>]+|
      \(([^\s()<>]+|(\([^\s()<>]+\)))*\))+(?:\(([^\s()<>]+|(\([^\s()
<>]+\)))*\)|[^\s`!()\[\]{};:'".,<>?«»“”‘’]))\n''', '', ulasan)
        return ulasan
      df_data['content']= df_data['content'].apply(cleaningUlasan)
```

Proses pertama yaitu menghapus emoji dengan bantuan *library* emoji dan metode encode dan decode ascii. Selanjutnya kode untuk menghilangkan berbagai karakter dengan bantuan regax (re). Hasil dari tahapan *text cleaning* dapat dilihat pada Tabel 3.2.

Tabel 3.2 Data Hasil *Text Cleaning*

No	Data Ulasan
1.	sangat bagus mengasik kan mendidik
2.	apk nya cocok buat jomblo kayak kamu
3.	aplikasi bagus sangat menghibur
4.	kamera buat ftonya dikembaliin dulu dong jelekk jadinya gtu
5.	banyak terimakasih baru mulai menjadi afiliator tiktok semoga sukses
6.	Assalamualaikum
7.	banyak eror tman udah nonton tiap hari even maju maju even nya cuma bohong belaka
8.	fyp in vidio kak
9.	sya live sya live malah kena pelanggran hah aku cmn mau live kayk orang masalah yg mudah kan aku live tiktok aku ingin masalahkita live tenang apa gk gitu hah aku baru pertama kali live malah kena pelanggaran andai kamu tau aku udh bisa live seneng bgt ujung2 nya kamu malah melanggar sya emng di disitu rasa senang tiba2 orang menghancurkan kesenangan sya akun l0velynbianra menerima semua
10.	good apk

3. Tokenizing

Tokenizing adalah proses pemisahan kata dalam kalimat dengan tujuan untuk analisis teks lebih lanjut, sehingga mempermudah dalam proses analisis. Berikut kode untuk melakukan *tokenizing.*

```
nltk.download('punkt')
def tokenizingText(ulasan):
  ulasan = word_tokenize(ulasan)
  return ulasan
df_data['content']= df_data['content'].apply(tokenizingText)
```

Kode di atas berfungsi untuk melakukan *tokenizing* dengan memecah kalimat menjadi per kata. Hasil dari *tokenizing* dapat dilihat pada Tabel 3.3.

Tabel 3.3 Data Hasil *Tokenizing*

No	Data Ulasan
1.	['karna', 'saya', 'suka', 'tiktok', 'jadi', 'saya', 'kasih', 'bintang', 'lima']
2.	['bagus']
3.	['bagus', 'banget', 'bisa', 'cari', 'video', 'yang', 'menarik']
4.	['sangat', 'bermampat']
5.	['very', 'good']
6.	['sangat', 'menyenangkan']
7.	['duhh', 'kenapa', 'fitur', 'simpan', 'foto', 'nya', 'di', 'hapus', 'tiktokk', 'padahal', 'aku', 'download', 'tiktok', 'cuman', 'buat']
8.	['dear', 'tiktok', 'semua', 'video', 'tiktok', 'nya', 'pada', 'bagus', 'oh', 'ya', 'kan', 'aku', 'mau', 'bikin', 'video', 'tapi', 'akun', 'aku', 'ilang', 'jadi', 'aku', 'ga', 'download', 'tiktok', 'lagi', 'karena', 'mama', 'aku', 'yang', 'bikin', 'akun', 'aku', 'ilang', 'terus', 'hapus', 'tiktok', 'nya']
9.	['bagus', 'apk', 'nya', 'tapi', 'kadang', 'ada', 'vidio', 'yang', 'seharusnya', 'tidak', 'boleh', 'di', 'lihat', 'terutama', 'buat', 'anak', 'anak']
10.	['tik', 'tok', 'sayang', 'apk', 'yang', 'bagus']

4. Stopword Removal

Stopword removal merupakan tahapan untuk menghapus kata-kata yang tidak memiliki makna penting. Pada tahap ini menggunakan *library* Sastrawi untuk melakukan *stopword removal.*

```
pip install sastrawi
```

Kode di atas digunakan untuk *install library* Sastrawi yang akan digunakan untuk melakukan tahapan *stopword removal.* Kode pemanggilan *library* Sastrawi dapat dilihat di bawah ini.

```
From Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()
stopword = stop_factory.create_stop_word_remover()
datastopwords = stop_factory.get_stop_words()
datastopwords
```

Kode di atas digunakan untuk menampilkan kata yang tidak bermakna penting, yang tersedia di *library* Sastrawi. Hasil dari proses *stopword removal* dapat dilihat pada Tabel 3.4.

Tabel 3.4 Hasil dari *Library Sastrawi*

Hasil kata yang tidak memiliki makna dari *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'

Pada Tabel 3.4 menampilkan kata yang akan dihilangkan pada data ulasan. Kode yang digunakan untuk menghilangkan kata-kata di atas sebagai berikut.

```
def removeStopWords(ulasan):
    clean_word_list = [word for word in ulasan.split() if word
    not in stopwords]
     return clean_word_list
stopwords_ulasan = lower_ulasan.apply(removeStopWords)
print(stopwords_ulasan)
```

5. Stemming

Stemming merupakan proses untuk menghilangkan imbuhan pada suatu kata. Proses *stemming* membutuhkan *library* Sastrawi yang sudah disiapkan pada proses *stopword removal*. Berikut kode untuk melakukan *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(len(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 di atas digunakan untuk mengubah kata menjadi bentuk dasar. Sastrawi merupakan *library stemming* yang digunakan untuk memproses teks dalam Bahasa Indonesia dan *class* yang digunakan dalam proses *stemming* adalah *class* StemmerFactory.

3.2.3 Pelabelan Manual

Pada tahap pelabelan ini dilakukan secara manual dengan memberikan label positif dan negatif pada data ulasan yang sudah di *preprocessing.* Pelabelan

manual dilakukan di Microsoft Excel secara manual agar dapat dianalisis. Data ulasan yang diberi label didapatkan 4.737 data ulasan. Contoh data ulasan yang sudah dilakukan pelabelan manual ditunjukkan pada Tabel 3.5.

Tabel 3.5 Pelabelan Manual

No	Ulasan	Kelas	Label
1.	sangat bagus mengasik kan mendidik	positif	1
2.	apk nya cocok buat jomblo kayak kamu	positif	1
3.	kamera buat ftonya dikembaliin dulu dong jelekk jadinya gtu	negatif	-1
4.	banyak terimakasih baru mulai menjadi afiliator tiktok semoga sukses	positif	1
5.	sang at menghibur	positif	1
6.	banyak eror tman udah nonton tiap hari even maju maju even nya cuma bohong belaka	negatif	-1
7.	sya live sya live malah kena pelanggran hah aku cmn mau live kayk orang masalah yg mudah kan aku live tiktok aku ingin masalahkita live tenang apa gk gitu hah aku baru pertama kali live malah kena pelanggaran andai kamu tau aku udh bisa live seneng bgt ujung2 nya kamu malah melanggar sya emng di disitu rasa senang tiba2 orang menghancurkan kesenangan sya akun l0velynbianra menerima semuanegatif	negatif	-1
8.	good apk	positif	1
9.	kecewa berat tiktok q mngunggah vidio ktanya planggran yg buat trus pihak tiktok mnghapus vidio ktanya sih planggran planggran yg mn q jd bngung vidionya jg bysa aja gk da yg aneh trpksa q krangin bntangnya	negatif	-1
10.	aplikasinya bagus	positif	1

Tabel 3.5 menunjukkan hasil pelabelan manual bahwa data ulasan dengan kelas positif diberi label 1 dan kelas negatif diberi label -1. Data pelabelan digunakan untuk memeberikan nilai sentimen yang dihitung akurasinya.

3.2.4 Training

Training data merupakan proses menghasilkan model klasifikasi secara manual dengan menggunakan metode Naïve Bayes Classification dengan fungsi ekstraksi TF-IDF pada data teks. Contoh perhitungan TF-IDF ditunjukkan pada Tabel 3.6.

Tabel 3.6 Contoh Data

Dokumen (d)	Ulasan
d1	sangat bagus mengasikkan mendidik
d2	aplikasi bagus sangat terhibur
d3	tiktok gg sangat bagus tidak mengecewakan
d4	iklan sangat menggagu
d5	kebanyakan iklan udah skip mlah ngebug msih terdengar suara iklan nyo wlaupun udah keluari dri apk nya tetap sama

Tabel 3.6 merupakan contoh data yang nantinya akan digunakan untuk perhitungan TF-IDF, perhitungan TF-IDF dilakukan secara manual menggunakan 5 data ulasan yaitu d1, d2, d3, d4, d5. Perhitungan TF-IDF ditunjukkan pada Tabel 3.7.

Tabel 3.7 Perhitungan TF dan IDF

term/kata	d1	d2	d3	d4	d5	Df	IDF
Apk					1	1	0.69897
Aplikasi		1				1	0.69897
Bagus	1	1	1			3	0.221849
Dri					1	1	0.69897
Gg			1			1	0.69897
Iklan				1	1	2	0.39794
kebanyakan					1	1	0.69897
Keluari					1	1	0.69897
mendidik	1					1	0.69897
mengasikkan	1					1	0.69897

mengecewakan			1			1	0.69897
menggagu				1		1	0.69897
Mlah					1	1	0.69897
term/kata	**d1**	**d2**	**d3**	**d4**	**d5**	**Df**	**IDF**
Msih					1	1	0.69897
Ngebug					1	1	0.69897
Nya					1	1	0.69897
Nyo					1	1	0.69897
Sama					1	1	0.69897
sangat	1	1	1	1		4	0.9691
Skip					1	1	0.69897
Suara					1	1	0.69897
terdengar					1	1	0.69897
Terhibur		1				1	0.69897
Tetep					1	1	0.69897
Tidak			1			1	0.69897
Tiktok			1			1	0.69897
Udah					1	1	0.69897
wlaupun					1	1	0.69897

Tabel 3.7 menjelaskan perhitungan *term frequency* (TF) dan *invers document frequency* (IDF) yang dilakukan secara manual. Perhitungan selanjutnya dapat dilihat pada Tabel 3.8.

Tabel 3.8 Perhitungan TF * IDF

Perhitungan TF* IDF					
term/kata	**d1**	**d2**	**d3**	**d4**	**d5**
Apk					0.69897
Aplikasi		0.69897			
Bagus	0.221849	0.221849	0.221849		
Dri					0.69897
Gg			0.69897		

Iklan				0.39794	0.39794
Kebanyakan					0.69897
Keluari					0.69897
Perhitungan TF* IDF					
Mendidik	0.69897				
Mengasikkan	0.69897				
mengecewakan			0.69897		
Menggagu				0.69897	
Mlah					0.69897
Msih					0.69897
Ngebug					0.69897
Nya					0.69897
Nyo					0.69897
Sama					0.69897
sangat	0.9691	0.9691	0.9691	0.9691	
Skip					0.69897
Suara					0.69897
Terdengar					0.69897
Terhibur		0.69897			
Tetep					0.69897
Tidak			0.69897		
Tiktok			0.69897		
Udah					0.69897
Wlaupun					0.69897

Tabel 3.8 merupakan perhitungan TF-IDF yang dilakukan secara manual. Perhitungan ekstraksi TF-IDF secara otomatis menggunakan *library* sklearn dengan model *tfidfvectorizer*. Berikut kode yang digunakan untuk perhitungan secara otomatis.

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
s1 = "kecewa sama aplikasi kenapa gak masuk daftar akun saya
sangat kecewa sama diselesaikan saya hapus aplikasi"
s2 = "mengundang teman teman selalu menonton hari saya
mendapatkan bonus poin nya"
vect = TfidfVectorizer()
x = vect.fit_transform([s1, s2])
x.toarray()
```

Hasil dari TF-IDF yang dilakukan secara otomatis dapat dilihat pada Gambar 3.2.

```
array([[0.2130798 , 0.4261596 , 0.        , 0.2130798 , 0.2130798 ,
        0.2130798 , 0.2130798 , 0.        , 0.4261596 , 0.2130798 ,
        0.2130798 , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.4261596 , 0.2130798 , 0.30321606, 0.        ,
        0.        ],
       [0.        , 0.        , 0.2827721 , 0.        , 0.        ,
        0.        , 0.        , 0.2827721 , 0.        , 0.        ,
        0.        , 0.2827721 , 0.2827721 , 0.2827721 , 0.2827721 ,
        0.2827721 , 0.        , 0.        , 0.20119468, 0.2827721 ,
        0.56554419]])
```

Gambar 3.2 Hasil Perhitungan TF-IDF

Pada Gambar 3.2 merupakan hasil dari perhitungan TF-IDF secara otomatis. Selanjutnya dilakukan perhitungan *Confusion Matrix* pada data training dengan menggunakan Google Colab.

```
from sklearn.model_selection import ShuffleSplit
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
X = df_data.content
y = df_data.label
ss = ShuffleSplit(n_splits=10, test_size=0.2)
sm = SMOTE()
accs = []
f1s = []
cms = []
```

Perhitungan dilakukan dengan *library* sklearn dengan model ShufflenShit. Perhitungan *cross validation* menghasilkan akurasi model. Kode yang digunakan untuk perhitungan *cross validation* sebagai berikut:

```
for train_index, test_index in ss.split(X):
  X_train, X_test = X.iloc[train_index], X.iloc[test_index]
```

```
        y_train, y_test = y.iloc[train_index], y.iloc[test_index]
        vect = TfidfVectorizer(max_features=1000, binary=True)

        # Fit vectorizer and transform X train, then transform X test
        X_train_vect = vect.fit_transform(X_train)
        X_test_vect = vect.transform(X_test)
        # Oversample
          X_train_res,   y_train_res   =   sm.fit_resample(X_train_vect,
y_train)
        # Fit Naive Bayes on the vectorized X with y train labels,
        # then predict new y labels using X test
        nb = MultinomialNB()
        nb.fit(X_train_res, y_train_res)
        y_pred = nb.predict(X_test_vect)
        # Determine test set accuracy and f1 score on this fold using
the true y labels and predicted y labels
        accs.append(accuracy_score(y_test, y_pred))
        f1s.append(f1_score(y_test, y_pred, average='weighted'))
        cms.append(confusion_matrix(y_test, y_pred))
```

Kode di atas digunakan untuk proses perhitungan *cross validation.* Selanjutnya, data akan ditampilkan berdasarkan akurasi yang dihasilkan. Kode perintah yang digunakan sebagai berikut:

```
      print("\nAverage accuracy across folds: {:.2f}%".format(sum(accs)
/len(accs) * 100))
      print("\nAverage F1 score across folds: {:.2f}%".format(sum(f1s)
/ len(f1s) * 100))
      print("\nAverage    Confusion    Matrix    across    folds:    \n
{}".format(sum(cms) / len(cms)))
```

Kode di atas berfungsi untuk menapilkan hasil nilai *accuracy, f1-score,* dan *Confusion Matrix.* Setelah kode perintah di atas dijalankan, akan dilakukan pengerjaan model *pipeline* dengan *library* sebagai berikut:

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

Fungsi *library* sklearn menggunakan model *pipeline* dan *TfidfTransformer*. Model *pipeline* digunakan untuk mengatur aliran data *input* dan *output* dari sebuah model. Kode yang digunakan dapat dilihat di bawah ini.

```
X = df_train.Text
y = df_train.Label
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 pipeline menggunakan fungsi pickle akan disimpan dan nantinya akan digunakan. file pickle diberi nama model_classifier_nbc.pickle, berikut kode yang digunakan.

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

Setelah melakukan penyimpanan file *pickle*, perlu adanya pemanggilan file dengan mode teks (rb) yang artinya file tersebut sudah ada dan hanya perlu membaca teks *biner* dan mengubah ke objek aslinya. Kode untuk membuka model pickle seperti di bawah ini.

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

Kode di atas digunakan untuk membuka file *pickle* dan menandakan proses *training* data dengan metode NBC telah selesai. Fungsi kode model = open digunakan untuk membuka model pickle.

3.2.5 Testing

Testing merupakan proses untuk mengidentifikasi ketidaksesuai hasil sebuah sistem. Pertama yang dilakukan untuk proses *testing* adalah upload data testing yang bejumlah 156 positif dan 156 negatif. *Testing* menggunakan metode NBC berfungsi untuk melakukan pembobotan dengan data uji dan data latih yang nantinya memprediksi kesamaan dalam data uji. Menghitung akurasi menggunakan model prediction NBC, dapat dilihat sebagai berikut:

```
prediction = nbc_classifier.predict(np.asarray(data_ulasan))
prediction
```

Prediction digunakan untuk melakukan pelabean secara otomatis yang nantinya digunakan untuk menghitung akurasi dalam testing. Hasil data dari prediction akan dibandingkan dengan data yang diberi label secara manual. Perbandingan pelabelan dapat dilihat di bawah pada Gambar 3.3.

	No	Text	Aktual	Prediksi
0	1	ketcheeee aplikasi yg bs mnghibur sekaligus pe...	1	1
1	2	cukup baik ada bug bikin kesel	-1	-1
2	3	mantap semoga centang biru aku	1	1
3	4	bngt	1	1
4	5	mantap ni apk	1	1
...	...	...	...	...
295	296	burukkkk	-1	-1
296	297	plisss dongg nonton live sukanya ngelag muluu ...	1	-1
297	298	bagus kadang ngelag	-1	-1
298	299	tik tok rusak harus kasih iklan sih ganggu	-1	-1
299	300	tik tok asu akun ku dibaned dihapus terus dasa...	-1	-1

300 rows × 4 columns

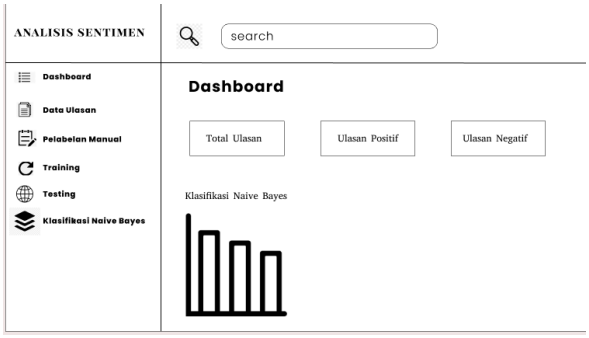
Gambar 3.3 Perbandingan Pelabelan

3.3 DESAIN INTERFACE

Interface merupakan program yang berhubungan langsung dengan pengguna sistem. Berikut adalah deskripsi atau desain-desain *interface* pada Analisis Sentimen Ulasan Aplikasi Tiktok pada Google Play Store Menggunakan Metode Naïve Bayes:

3.3.1 Dashboard Analisis Sentimen

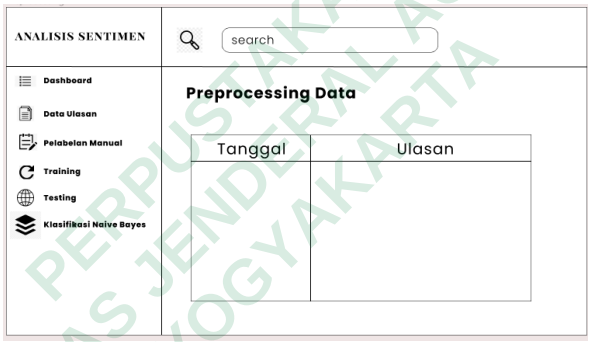
Pada sistem analisis sentimen dapat mengetahui data ulasan yang telah didapatkan kelas dan labelnya berdasarkan prediksi yang sudah dibuat pada model *training* dan ditampilkan pada menu dashboard. Tampilan menu dashboard ditunjukkan pada Gambar 3.4.



Gambar 3.4 Tampilan Dashboard

3.3.2 Preprocessing

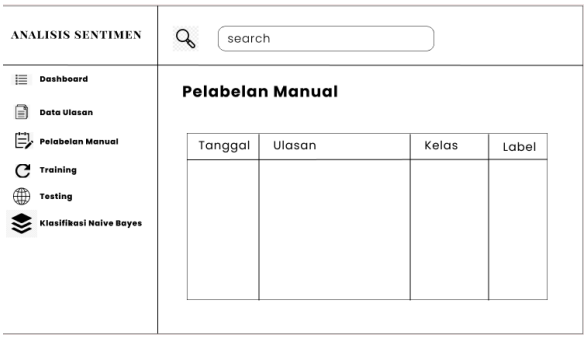
Pada preprocessing terjadi proses pengolahan data ulasan dengan langkah-langkah untuk menyempurnakan data ulasan sehingga menjadi data yang diinginkan. Tampilan menu *preprocessing* data ditunjukkan pada Gambar 3.5.



Gambar 3.5 Tampilan Preprocessing

3.3.3 Pelabelan Manual

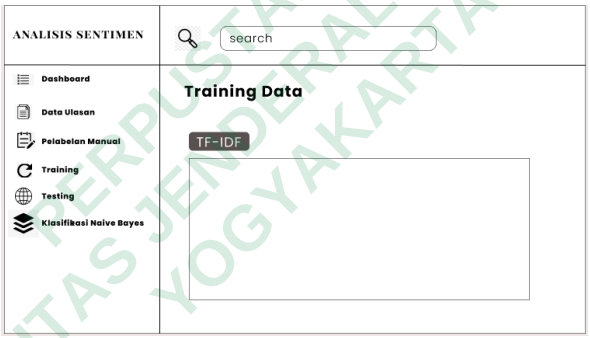
Pada pelabelan manual dilakukan proses untuk menampilkan data ulasan yang sudah diberi kelas dan label terhadap kalimat yang ada pada data ulasan secara manual. Proses pelabelan manual dilakukan di Microsoft Excel. Tampilan menu pelabelan manual ditunjukkan pada Gambar 3.6.



Gambar 3.6 Tampilan Pelabelan Manual

3.3.4 Training

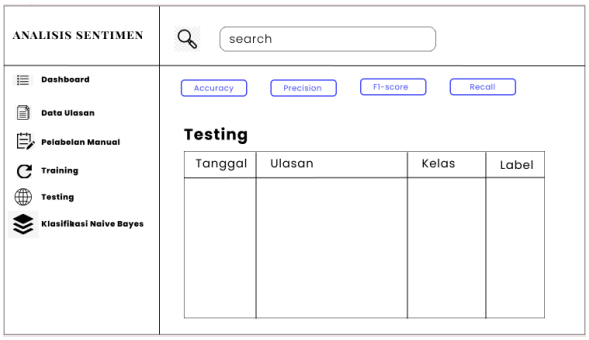
Training pada sistem analisis sentimen mrnggunakan ekstraksi Term Frequency-Invers Document Frequency (TF-IDF) untuk menghasilkan model klasifikasi yang dapat digunakan untuk menampilkan proses klasifikasi sentimen secara otomatis. Tampilan menu training data ditunjukkan pada Gambar 3.7.



Gambar 3.7 Tampilan *Training*

3.3.5 Testing

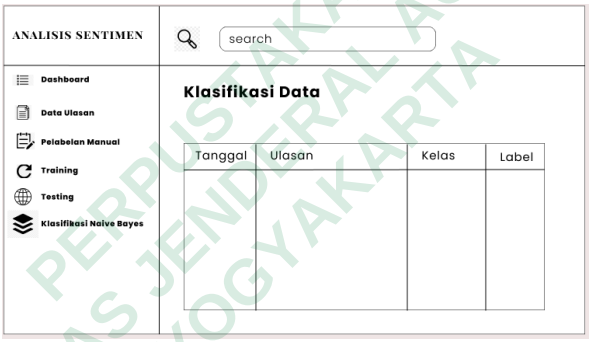
Testing pada sistem analisis sentimen dilakukan untuk mengetahui seberapa akurat model yang dibangun dalam *training* data digunakna untuk memprediksi kelas atau label dari data uji yang tersedia. Tampilan menu *testing* ditunjukkan pada Gambar 3.8.



Gambar 3.8 Tampilan *Testing*

3.3.6 Klasifikasi Naïve Bayes

Klasifikasi naïve bayes dilakukan untuk menampilkan data ulasan yang diprediksi kelas, label yang sudah dibangun pada *training* data. Tampilan menu Klasifikasi Naïve Bayes ditujunjukkan pada Gambar 3.9.



Gambar 3.9 Tampilan Klasifikasi NBC