


## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Web Scraping

Data yang digunakan pada penelitian ini merupakan ulasan dari aplikasi Starbucks Indonesia dari Google Playstore. Pengumpulan data menggunakan *library google-play-scraper* dan *app\_store\_scraper*. *Google-play-scraper* memberikan kemudahan pengguna dalam pengambilan data dari Google Play Store. Sedangkan *app\_store\_scraper* memudahkan pengambilan data dari App Store. Pengambilan data ini menggunakan *software* Jupyter Notebook dengan bahasa pemrograman python. Setelah data terkumpul dilakukan penggabungan antara data dari App Store dan Google Play Store.

	source	review_id	user_name	review_title	review_description	rating	thumbs_up	review_date	developer_response
0	Google Play	d711228e-cdb1-4c69-8cda-a5999b3314f2	Periskha Bunda Syafiie	NaN	No, not again	1	0	2024-05-03 08:32:27	Hi, Really apologize to hear your review. Woul...
1	Google Play	59725f02-489d-4001-8e46-a40224392c9b	Ronny Sugianto	NaN	App sucks. No information regarding how many r...	1	1	2024-05-02 11:00:05	Hi, Really apologize for the inconvenience. Fo...
2	Google Play	3593575b-89fb-4057-b39a-ed6686d3f086	Ferry	NaN	Can't use this bloody app for a weeks!!	1	0	2024-05-01 03:06:28	good things happen: www.starbucks.co.id 
3	Google Play	59b6df70-8c0f-4933-88cb-72f502b83c41	Sahat Hamonangan	NaN	Udah 1 minggu aplikasinya error, muncul tulis...	1	0	2024-05-01 02:51:25	Hi, Terima kasih untuk feedbacknya. Kami sanga...
4	Google Play	793c0264-c32e-4f8b-8c2d-8f85b430f102	Rein Valentine	NaN	Yeaa no, was told to register for rewards. Reg...	1	1	2024-04-30 05:51:12	Hi, Really apologize for the inconvenience.If ...
5	Google Play	fb260ab2-a17a-4e50-a76a-1f5af464489b	Ando Malano	NaN	Unable to connect.. what ..... ?	1	0	2024-04-29 21:33:13	Hi, Really apologize for the inconvenience. Fo...

Gambar 4.7 Scraping data Google Play Store

Dari proses web scrapping pada situs Google Play pada aplikasi Starbucks Indonesia seperti yang dapat dilihat yang nampak pada Gambar 4.7, didapatkan data kumpulan kolom berupa tabel dengan kolom yang berisi informasi berupa source, review\_id, user\_name, review\_title, review\_description, rating, thumbs\_up, review\_date, dan developer\_response.

	source	review_id	review_date	developer_response	review_description	rating	user_name
0	App Store	b516031e-aa43-4722-9f00-0f859e9e1d77	2023-07-01 09:15:27	{'id': 37526026, 'body': 'Hi, Mohon maaf atas ...	Tolong diperbaiki untuk order online Via aplik...	2	Takatiki90
1	App Store	38956518-e4f1-418c-a031-e287aef70b59	2023-03-07 18:18:44	{'id': 35327591, 'body': 'Hi, Mohon maaf atas ...	perasaan ps wrd udah bener, pas log in coba aku...	1	novitsrx
2	App Store	9b2b4724-fbc8-4992-a305-e491e2ca1cf6	2023-11-22 06:01:03	{'id': 40335099, 'body': 'Hi, Mohon maaf atas ...	Halo admin \nKenapa aplikasinya sejak 2 minggu...	2	mantap93
3	App Store	c5a10e5f-dab0-4e31-b1d1-f6bca44bdb1d	2023-02-28 11:26:54	NaN	very very bad 🤔	1	maaboiiii
4	App Store	8ac64cb2-3322-4ff8-97f5-b26ebe22d1c6	2024-02-03 06:25:11	NaN	Kode refal : IMA-5B5A39	5	Arik_12
5	App Store	11516f70-1a70-4a07-9f82-ee7a672a713e	2023-11-12 08:04:59	{'id': 40207612, 'body': 'Hi, Mohon maaf atas ...	Sudah 1 minggu lebih tidak bisa log in	1	rhzphlv

**Gambar 4.8 Scraping data App Store**

Dari proses web scrapping pada situs App Store pada aplikasi Starbucks Indonesia seperti yang nampak pada Gambar 4.8, didapatkan data berupa kumpulan kolom yang menjadi tabel dengan kolom yang berisi informasi berupa source, review\_id, user\_name, review\_title, review\_description, rating, thumbs\_up, review\_date, dan developer\_response.

	source	review_id	review_date	developer_response	review_description	rating	user_name	review_title
0	App Store	b516031e-aa43-4722-9f00-0f859e9e1d77	2023-07-01 09:15:27	{'id': 37526026, 'body': 'Hi, Mohon maaf atas ...	Tolong diperbaiki untuk order online Via aplik...	2	Takatiki90	Internal Server Error
1	App Store	38956518-e4f1-418c-a031-e287aef70b59	2023-03-07 18:18:44	{'id': 35327591, 'body': 'Hi, Mohon maaf atas ...	perasaan ps wrd udah bener, pas log in coba aku...	1	novitsrx	app eror
2	App Store	9b2b4724-fbc8-4992-a305-e491e2ca1cf6	2023-11-22 06:01:03	{'id': 40335099, 'body': 'Hi, Mohon maaf atas ...	Halo admin \nKenapa aplikasinya sejak 2 minggu...	2	mantap93	Aplikasi tidak berjalan
3	App Store	c5a10e5f-dab0-4e31-b1d1-f6bca44bdb1d	2023-02-28 11:26:54	NaN	very very bad 🤔	1	maaboiiii	🤔
4	App Store	8ac64cb2-3322-4ff8-97f5-b26ebe22d1c6	2024-02-03 06:25:11	NaN	Kode refal : IMA-5B5A39	5	Arik_12	Coffe terbaik
...	...	...	...	...	...	...	...	...
14127	Google Play	131f21bc-8c6a-454b-9498-d0274276c8c8	2019-05-02 08:18:12	NaN	good	5	A Google user	NaN

**Gambar 4.9 Hasil penggabungan data dari Google Play Store dengan App Store**

Hasil scrapping data dari kedua situs tersebut lalu digabungkan menjadi satu tabel seperti yang dapat dilihat pada Gambar 4.9 sehingga menghasilkan jumlah total data sebanyak 14.000 data.

	<b>review_description</b>
0	Tolong diperbaiki untuk order online Via aplik...
1	perasaan pswrd udah bener, pas log in coba aku...
2	Halo admin \nKenapa aplikasinya sejak 2 minggu...
3	very very bad 🤔
4	Kode refal : IMA-5B5A39
...	...
14127	good
14128	good
14129	cool
14130	i am happy to have it!
14131	My card cant be used, always failed when the b...

**Gambar 4.10 Hasil data Scraping berupa *table* review\_description**

Hasil dari penggabungan tadi dilakukan filterisasi untuk menghilangkan kolom data yang tidak diperlukan pada penelitian ini seperti yang bisa dilihat pada Gambar 4.10 sehingga menghasilkan kolom review\_description yang berisi ulasan dari pengguna aplikasi Starbucks Indonesia.

#### **4.2 Machine Translation**

Proses ini dimulai dengan memuat dataset teks bahasa Inggris yang ingin diterjemahkan. EasyNMT mendukung berbagai model, termasuk Opus-MT. Setelah model dipilih, inisialisasi dilakukan dengan menetapkan bahasa sumber sebagai bahasa Inggris dan bahasa target sebagai bahasa Indonesia. Teks yang berbahasa Inggris akan diumpankan ke model, yang kemudian memprosesnya melalui jaringan neural untuk menghasilkan terjemahan dalam bahasa Indonesia. Proses ini menggabungkan analisis konteks dan makna kata, sehingga menghasilkan terjemahan yang lebih alami dan akurat yang tersaji pada Tabel 4.4

di bawah. Hasil terjemahan kemudian disimpan dalam *dataset* baru untuk analisis atau penggunaan lebih lanjut.

**Tabel 4.5 hasil translate menggunakan EasyNMT**

Sebelum	Sesudah
<i>I bought a new card registered successfully but can't login and can't use my card to pay. Please starbucks, you are a big company, invest more on your software and server. Well i know your answer will be "i'm sorry for the inconvenience etc...." so dissapointed.</i>	Aku membeli kartu baru yang terdaftar dengan sukses tapi tidak bisa login dan tidak bisa menggunakan kartu saya untuk membayar. Silakan starbucks, Anda adalah perusahaan besar, berinvestasi lebih pada perangkat lunak Anda dan server. Yah saya tahu jawaban Anda akan "saya minta maaf atas ketidakbecusan dll...." jadi kecewa.
<i>Good</i>	Baik
<i>Its good for relax and coffee time</i>	ini bagus untuk bersantai dan waktu kopi
<i>Cannot open the app on busy hours.</i>	Tidak dapat membuka aplikasi pada jam sibuk.
<i>App wont open, it's been a month</i>	App tidak bisa terbuka, sudah satu bulan
<i>After update can't Open the app</i>	Setelah pemutakhiran tidak dapat membuka aplikasi

### 4.3 Vader Lexicon

Proses pelabelan sentimen menggunakan VADER (*Valence Aware Dictionary and Sentiment Reasoner*) Lexicon yang bisa dilihat pada Gambar 4.11 melibatkan beberapa langkah. Pertama, setiap kata dalam teks yang dianalisis dicocokkan dengan kamus VADER, yang berisi penilaian sentimen untuk banyak

kata dan frasa umum. VADER memberikan skor positif dan negatif untuk setiap kata atau frasa.

```

nlTK.download('vader_lexicon')

sia = SentimentIntensityAnalyzer()

def label_sentiment(text):

    scores = sia.polarity_scores(text)

    if scores['compound'] >= 0.05:
        return 'positive'
    elif scores['compound'] <= -0.05:
        return 'negative'
    else:
        return 'neutral'

df['sentiment'] = df['translated_text'].apply(label_sentiment)
df.head(20)

```

**Gambar 4.11** Proses *Labelling* menggunakan Vader Lexicon

Proses *labelling* menggunakan Vader Lexicon seperti pada Gambar 4.11 dilakukan berdasarkan nilai compound yaitu jika kurang dari atau sama dengan 0,05 maka teks diklasifikasikan sebagai positif. Berikutnya, jika nilai compound kurang dari atau sama dengan -0,05 akan diklasifikasikan sebagai negatif.

	review_description	translated_text	sentiment
0	Swear to God this app is made by amateurs. The...	Sumpah, aplikasi ini dibuat oleh para amatir. ...	negative
1	The idea of this app is great and its very use...	Gagasan tentang aplikasi ini sangat bagus dan ...	positive
2	After the last 2 updates this app became annoy...	Setelah 2 pemutakhiran terakhir aplikasi ini m...	negative
3	The input pin is very annoying. Because when y...	Pin inputnya sangat mengganggu. Karena ketika ...	positive
4	The apps keeps kicking me out from the apps. E...	Aplikasi terus menendang saya keluar dari apli...	negative
5	App sucks. No information regarding how many r...	App menyebalkan. Tidak ada informasi tentang b...	negative
6	This app is useful, but once logged out, can't...	Aplikasi ini berguna, tetapi sekali keluar, ti...	positive
7	Traveling in Indonesia and my US Sbox doesn't ...	Bepergian di Indonesia dan Sbox AS saya tidak ...	negative
8	It was good, now it is just rubbish. From the ...	Itu bagus, sekarang itu hanya sampah. Dari yan...	negative
9	I can't login! They ask me to relogin like onc...	Aku tidak bisa masuk! Mereka meminta saya untu...	negative
10	This app just like a beginner who starting lea...	Aplikasi ini seperti pemula yang mulai belajar...	positive
11	Absolutely garbage. The only people giving it ...	Benar-benar sampah. Satu-satunya orang memberi...	positive
12	Need many improvements. Oftentimes won't even ...	Perlu banyak perbaikan. Seringkali bahkan tida...	positive
13	When I open the app it won't even reach the me...	Ketika saya membuka aplikasi itu bahkan tidak ...	positive
14	Some concern and suggestions: - fix the load t...	Beberapa kekhawatiran dan saran: - perbaiki wa...	positive
15	Persistently periodically kicks me out and forc...	Secara berkala terus-menerus menendang saya ke...	negative
16	Guys don't limit passwords to max 12 character...	Laki-laki tidak membatasi kata sandi hingga ma...	negative
17	Yeea no, was told to register for rewards. Reg...	Yeea tidak, diberitahu untuk mendaftar untuk h...	negative
18	I thought it was my device but turns out to be...	Saya pikir itu perangkat saya tetapi ternyata ...	negative

**Gambar 4.12** Hasil *Labelling* menggunakan Vader Lexicon

Hasil *labelling* menggunakan Vader Lexicon seperti yang dapat dilihat pada Gambar 4.11 menghasilkan kolom baru yaitu kolom sentiment yang berisi “positive” atau “negative”.

#### 4.4 Pre-Processing Data

*Pre-processing* data dilakukan setelah data yang dikumpulkan menggunakan teknik scraping menggunakan library google-play-scraper dan app-store-scraper yang diterjemahkan ke bahasa Indonesia dan diberi label pada tahap pelabelan sebelumnya. Data ulasan yang telah diambil memiliki kata-kata yang kurang penting. Maka harus dilakukan praproses data agar data tersebut bersih dari kata yang kurang penting. Langkah-langkah dari praproses data adalah *case folding*, *data cleaning*, *stopword*, *stemming*, *tokenizing*.

##### 4.4.1 Case Folding

Langkah pertama dalam praproses data adalah *case folding*. Semua huruf yang ada pada tiap dokumen akan dijadikan huruf kecil semua agar terhindar dari *case sensitive*. Contoh dari proses *case folding* adalah pada data teks pertama adalah pada kata “Setelah” menjadi “setelah” dan “Saya” menjadi “saya”. Hasil dari *case folding* yang sudah dilakukan tertampil pada Tabel 4.5.

**Tabel 4.6 hasil proses Case Folding**

Sebelum	Sesudah
Setelah 2 pemutakhiran terakhir aplikasi ini menjadi menjengkelkan. Aku tidak bisa memeriksa transaksi bulan lalu hanya bulan sekarang. Update terbaru hanya membuatnya	setelah pemutakhiran terakhir aplikasi ini menjadi menjengkelkan aku tidak bisa memeriksa transaksi bulan lalu hanya bulan sekarang update terbaru hanya membuatnya
Saya pikir itu perangkat saya tetapi ternyata menjadi kesalahan adalah dari aplikasi	saya pikir itu perangkat saya tetapi ternyata menjadi kesalahan adalah dari aplikasi
Aku tidak bisa membuka aplikasi selama berminggu-minggu.	aku tidak bisa membuka aplikasi selama berminggu-minggu.

Setiap kali saya membuka aplikasi, saya harus kembali login, memeriksa OTP pada email dan mengatur nomor PIN berkali-kali.	setiap kali saya membuka aplikasi, saya harus kembali login, memeriksa otp pada email dan mengatur nomor pin berkali-kali
--	---

#### 4.4.2 Data Cleansing

Tahap selanjutnya adalah *Data Cleaning* di mana pada tahap ini data yang sebelumnya sudah disesuaikan semua menjadi lowercase akan dibersihkan lagi dengan menghilangkan URL, tanda baca, angka, simbol, emoticon, emoji, dll. Contoh dari tahap *Data Cleansing* adalah pembersihan noise pada data tersebut seperti penghapusan emoji (☹️). Proses *Data Cleansing* yang sudah dilakukan tertampil pada Tabel 4.6.

**Tabel 4.7 Hasil proses Data Cleaning**

Sebelum	Sesudah
sangat sangat buruk ☹️.	sangat sangat buruk
tidak bisa log in 🙄	tidak bisa log in
aplikasi bagus, banyak promo untuk member terbaik lah 👍	aplikasi bagus, banyak promo untuk member terbaik lah
setiap kali saya membuka aplikasi, saya harus kembali login, memeriksa otp pada email dan mengatur nomor pin berkali-kali	setiap kali saya membuka aplikasi saya harus kembali login memeriksa otp pada email dan mengatur nomor pin berkali kali
menyebalkan! aku bahkan tidak bisa mendaftar. setiap kali saya mencoba untuk melakukannya, aplikasi berhenti memproses dan meminta saya untuk memulai kembali.	menyebalkan aku bahkan tidak bisa mendaftar. setiap kali saya mencoba untuk melakukannya aplikasi berhenti memproses dan meminta saya untuk memulai kembali

#### 4.4.3 Stopwords Removal

Tahap selanjutnya adalah *Stopwords Removal* di mana pada tahap ini kata-kata yang tidak memiliki makna atau arti penting akan dihapuskan. Contoh dari kata-kata tersebut adalah “dapat”, “ini”, “sangat”, “cukup”, dll. Proses *Stopwords Removal* dapat dilihat pada Tabel 4.7.

**Tabel 4.8** hasil proses *Stopwords Removal*

Sebelum	Sesudah
tidak dapat terhubung ke server	tidak terhubung server
sudah beberapa kali buka app ini selalu keluar tidak bisa digunakan	sudah kali buka app keluar tidak digunakan
tidak bisa memuat account saya jadi saya tidak bisa menebus hadiah saya sangat mengecewakan tolong perbaiki segera	tidak memuat account tidak menebus hadiah mengecewakan tolong perbaiki
menyebabkan aku bahkan tidak bisa mendaftar. setiap kali saya mencoba untuk melakukannya aplikasi berhenti memproses dan meminta saya untuk memulai kembali	menyebabkan mendaftar kali mencoba melakukannya aplikasi berhenti memproses

#### 4.4.4 Tokenization

Setelah tahap *Stopwords Removal*, berikutnya akan dilakukan tahap *Tokenization* yang akan membagikan kalimat-kalimat pada data tersebut menjadi kata satuan. Proses *Tokenization* dapat dilihat pada Tabel 4.8.

**Tabel 4.9** Hasil proses *Tokenization*

Sebelum	Sesudah
permisi tidak menghubungi server tolong diperbaiki	['permisi', 'tidak', 'menghubungi', 'server', 'tolong', 'diperbaiki']
tidak menemukan server	['tidak', 'menemukan', 'server']
lambat tolong perbaiki	['lambat', 'tolong', 'perbaiki']



harap tambahkan sandi berulang lupa sandi	['harap', 'tambahkan', 'sandi', 'berulang', 'lupa', 'sandi']
menyebabkan mendaftar kali mencoba melakukannya aplikasi berhenti memproses	['menyebabkan', 'mendaftar', 'kali', 'mencoba', 'melakukannya', 'aplikasi', 'berhenti', 'memproses']

#### 4.4.5 Stemming

Proses berikutnya setelah *Stopword Removal* adalah *Stemming*, pada tahap ini dilakukan penyederhanaan kata dari kata sifat yang memiliki imbuhan menjadi bentuk dasarnya sehingga nantinya bisa diolah dengan menggunakan model Naïve Bayes. Proses *Stemming* dapat dilihat pada Tabel 4.9 dan untuk data lengkapnya bisa dilihat pada Lampiran 8.

**Tabel 4.10 Hasil proses *Stemming***

Sebelum	Sesudah
['harap', 'tambahkan', 'sandi', 'berulang', 'lupa', 'sandi']	harap <b>tambah</b> sandi ulang lupa sandi
['kali', 'menekan', 'menu', 'hadiah', 'kesalahan', 'lambat']	kali tekan menu hadiah <b>salah</b> lambat
['tidak', 'mendaftar', 'server', 'tidak', 'berguna']	tidak <b>daftar</b> server tidak guna
['membuka', 'menu', 'rewards', 'memuat', 'tidak', 'merespon', 'tidak', 'menampilkan', 'meluncurkan', 'aplikasi', 'toko', 'bermain', 'peringkat', 'buruk', 'aplikasi', 'mempengaruhi', 'merek']	<b>buka</b> menu rewards <b>muat</b> tidak merespon tidak <b>tampil</b> luncur aplikasi toko main peringkat buruk aplikasi <b>pengaruh</b> merek

#### 4.5 Pembobotan TF-IDF

Tahap selanjutnya setelah mendapatkan kata-kata yang sudah dipecah dari satu kalimat tertentu, maka perlu dilakukan pembobotan dari tiap kata sebagai nilai

dari fitur agar dapat diproses oleh *machine learning* karena *machine learning* hanya dapat menerima *input* berupa angka. Pada tahap pembobotan ini, metode yang digunakan adalah TF-IDF (*Term Frequency – Inverse Document Frequency*). Pada metode ini, akan dilakukan perhitungan TF (*Term Frequency*) dan IDF (*Inverse Document Frequency*) pada setiap token dalam setiap dokumen.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

# Menghitung TF-IDF
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(df['stemming'])

# Mendapatkan daftar kata-kata fitur (terms)
terms = tfidf_vectorizer.get_feature_names_out()

# Membuat DataFrame untuk nilai TF-IDF
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=terms)

# Mengurutkan DataFrame berdasarkan nama kolom
tfidf_df_sorted = tfidf_df.sort_index(axis=1)

# Menampilkan DataFrame yang sudah diurutkan
print(tfidf_df_sorted)
```

**Gambar 4.13** Proses Pembobotan TF-IDF

Gambar 4.13 merupakan *source code* dari proses TF-IDF dengan memanfaatkan *library* dari `sklearn.feature_extraction.text`. Dengan menggunakan *library* python yang di dalamnya memiliki proses penghitungan menggunakan Rumus (1.1) tersebut proses dari pembobotan TF-IDF dilakukan. Kata “aplikasi” muncul pada 4.447 dokumen dari 13.000 dokumen. Berdasarkan data tersebut dilakukan perhitungan menggunakan Rumus (1.1) dengan implementasi sebagai berikut.

$$w_{ij} = tf_{ij} \times \log\left(\frac{D}{df_j}\right) + 1$$

Diketahui:

$$tf_{ij} = 1$$

$$df_j = 4.447$$

$$\text{IDF}_{fj} = \log\left(\frac{D}{df_j}\right) + 1 = \log\left(\frac{13.000}{4.447}\right) + 1 = \log(2.9233) + 1 = 0.46587$$

Jadi, untuk menghitung bobot kata “aplikasi” menggunakan TF-IDF adalah sebagai berikut.

$$\begin{aligned} \text{TF-IDF} &= tf_{ij} \times \text{IDF}_{fj} \\ &= 1 \times 0,46587 \\ &= \mathbf{0,46587} \end{aligned}$$

Dari hasil pembobotan tersebut didapatkan bahwa bobot kata “aplikasi” yang muncul satu kali pada sebuah dokumen memiliki bobot nilai 0,46587. Berikutnya akan dihitung nilai akumulasi dari seluruh bobot dari suatu kata sehingga didapatkan nilai yang tertampil pada Tabel 4.11.

**Tabel 4.11 Hasil Pembobotan TF-IDF**

No.	Kata	Bobot TF-IDF
1	aplikasi	960,550
2	hadiah	175,827
3	baik	664,287
4	bagus	449,292
...	...	...
5204	buruk	151,567

Hasil dari proses pemberian nilai menggunakan TF-IDF terlampir pada Tabel 4.11 yang lengkapnya dapat dilihat pada Lampiran 9. Setiap kata yang ada pada teks dibobotkan banyaknya kemunculan suatu kata yang ada di dalam *dataset*. Hasil tersebut dihasilkan dari Rumus (1.1).

#### 4.6 Pembagian Data Latih dan Data Uji

Dalam proses membagi data latih dan uji, model klasifikasi dibentuk sehingga hasil dari data latih dan uji dapat memberikan informasi yang berguna untuk memprediksi kategori data baru. Secara umum, semakin tinggi proporsi data latih dibandingkan data uji, semakin baik akurasi hasil evaluasi.

```

tfidf_vectorizer = TfidfVectorizer()
X = tfidf_vectorizer.fit_transform(nb['stemming'])
y = nb['sentiment']

# Membagi data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=30)

```

**Gambar 4.14 Pembagian Data Latih dan Data Uji**

Data latih dan data uji yang digunakan pada proses penelitian ini mencakup sentimen negatif dan positif. Pembagian data latih dan juga data uji pada *dataset* penelitian ini adalah 80% untuk data latih dan 20% untuk data uji. Proses pembagian data latih dan data uji dapat dilihat pada Gambar 4.14.

#### 4.7 Pembentukan Model Naïve Bayes

Proses pemodelan menggunakan algoritma Naïve Bayes pada *dataset* Starbucks Indonesia dapat dilakukan setelah dilakukan pembobotan pada tiap kata menggunakan TF-IDF dan dipecah menjadi data latih dan data uji.

```

MNB = MultinomialNB()
MNB.fit(X_train, y_train)

# Memperkirakan sentimen berdasarkan model yang dilatih
predicted = MNB.predict(X_test)

# Menghitung probabilitas posterior
probs = MNB.predict_proba(X)

# Membuat DataFrame untuk probabilitas
probs_df = pd.DataFrame(probs, columns=['prob_negative', 'prob_positive'])

# Menggabungkan dengan DataFrame asli
nb = pd.concat([nb, probs_df], axis=1)

# Menentukan klasifikasi berdasarkan probabilitas
nb['klasifikasi_NB'] = nb[['prob_negative', 'prob_positive']].idxmax(axis=1)
nb['klasifikasi_NB'] = nb['klasifikasi_NB'].apply(lambda x: 'negative' if x == 'prob_negative' else 'positive')

nb

```

**Gambar 4.15 Proses pembentukan Model Naive Bayes**

Pada tahap ini, akan memanfaatkan beberapa *library* seperti `sklearn.naive_bayes` dan `sklearn.metrics`. Penggunaan `library sklearn.naive_bayes` adalah untuk membentuk *modelling* pada Multinomial Naïve Bayes yang di dalamnya terdapat Rumus (1.2) untuk melakukan proses penghitungan dan menggunakan model tersebut untuk dimuatkan dalam suatu variabel. Sedangkan pada `library sklearn.metrics` digunakan untuk menghitung nilai akurasi, nilai presisi, nilai *recall*, nilai *f1-score*, membuat *classification report*, dan membuat

confusion matrix. Proses pembentukan model Naive Bayes pada penelitian ini ditampilkan pada Gambar 4.15. Berikut merupakan implementasi dari Rumus (1.2).

Implementasi yang dilakukan akan menggunakan salah satu dari ulasan pengguna aplikasi, yaitu ulasan dengan kalimat yang berbunyi “aplikasi bagus” Digunakan Rumus (1.2) yaitu sebagai berikut.

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

$$P(H) =$$

$$P(\text{positif}) = 8320/13000$$

$$P(\text{negatif}) = 4680/13000$$

Berikutnya dilakukan penghitungan untuk dokumen dengan kalimat “aplikasi bagus” dengan sentimen positif:

1.  $P(\text{"aplikasi bagus"}|\textit{positive}) = P(\text{"aplikasi"}|\textit{positive}) \times P(\text{"bagus"}|\textit{positive})$
2.  $P(\text{"aplikasi"}|\textit{positive}) = 0,4$  dan  $P(\text{"bagus"}|\textit{positive}) = 0,8$
3.  $P(\text{"aplikasi bagus"}|\textit{positive}) = 0,4 \times 0,8 = 0,32$

Berikutnya dilakukan penghitungan untuk dokumen dengan kalimat “aplikasi bagus” dengan sentimen negatif:

1.  $P(\text{"aplikasi bagus"}|\textit{negative}) = P(\text{"aplikasi"}|\textit{negative}) \cdot P(\text{"bagus"}|\textit{negative})$
2.  $P(\text{"aplikasi"}|\textit{negative}) = 0,4$  dan  $P(\text{"bagus"}|\textit{negative}) = 0,1$
3.  $P(\text{"aplikasi bagus"}|\textit{negative}) = 0,4 \times 0,1 = 0,04$

Selanjutnya menghitung nilai dari  $P(X)$  dengan implementasi sebagai berikut.

$$P(X) = P(\text{"aplikasi bagus"})$$

$$= P(X|\textit{positive}) \times P(\textit{positive}) + P(X|\textit{negative}) \times P(\textit{negative})$$

$$= (0,32 \times \frac{8320}{13000}) + (0,04 \times \frac{4680}{13000}) = 0,2048 + 0,0144 = 0,2192$$

Langkah terakhir menghitung nilai probabilitas untuk menentukan apakah dokumen “aplikasi bagus” akan diklasifikasikan menjadi positif atau negatif dengan menggunakan Rumus (1.2).

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

$$\begin{aligned} P(\text{positive}|"aplikasi bagus") &= \frac{P(X|\text{positive}) \cdot P(\text{positive})}{P(X)} \\ &= \frac{0,32 \cdot \frac{8320}{13000}}{0,2192} \\ &= 0,9342 \end{aligned}$$

$$\begin{aligned} P(\text{negative}|"aplikasi bagus") &= \frac{P(X|\text{negative}) \cdot P(\text{negative})}{P(X)} \\ &= \frac{0,04 \cdot \frac{4680}{13000}}{0,2192} \\ &= 0,0656 \end{aligned}$$

**Tabel 4.12 Hasil dari perhitungan menggunakan rumus**

No.	<i>stemming</i>	prob_negative	prob_positive	klasifikasi_NB
340	aplikasi bagus	0,0656	0,9342	positive

Berdasarkan hasil penghitungan pada Tabel 4.12 dan dari Lampiran 10 yang terdapat pada *line* 340 tersebut didapatkan nilai probabilitas positif lebih besar daripada nilai probabilitas negatif sehingga bisa disimpulkan bahwa kalimat aplikasi bagus diklasifikasikan sebagai sentimen positif.

	sentiment		stemming	prob_negative	prob_positive	klasifikasi_NB
0	0	sumpah aplikasi amatir verifikasi otp rusak ki...		0.905987	0.094013	negative
1	1	gagas aplikasi bagus guna langgan hadap saatsa...		0.238538	0.761462	positive
2	0	mutakhir aplikasi jengkel periksa transaksi up...		0.705515	0.294485	negative
3	1	pin inputnya ganggu etik digit keyboard tibati...		0.126576	0.873424	positive
4	0	aplikasi tendang aplikasi pasuk passcode gangg...		0.821187	0.178813	negative
...	...		...	...	...	...
13061	1		baik	0.110370	0.889630	positive
13062	1		baik	0.110370	0.889630	positive
13063	1		keren	0.040863	0.959137	positive
13064	1		senang milik	0.163361	0.836639	positive
13065	0	kartu gagal barista gesek coba starbucks cafe ...		0.719984	0.280016	negative

**Gambar 4.16 Hasil Model Naive Bayes**

Hasil *modelling* Naive Bayes yang dilakukan menghasilkan nilai probabilitas negatif dan probabilitas positif yang tertampil pada Gambar 4.16. Dari nilai tersebut akan dibandingkan apakah nilai probabilitas positif atau negatif yang lebih tinggi, jika nilai probabilitas positif lebih tinggi maka akan diklasifikasikan menjadi positif dan sebaliknya untuk diklasifikasikan sebagai negatif.

	sentiment		stemming	prob_negative	prob_positive	klasifikasi_NB
1	1	gagas aplikasi bagus guna langgan hadap saatsa...		0.238538	0.761462	positive
3	1	pin inputnya ganggu etik digit keyboard tibati...		0.126576	0.873424	positive
10	1	aplikasi mula ajar aplikasi mobile uiux paksa ...		0.385695	0.614305	positive
11	1	benarbenar sampah satusatunya orang beri binta...		0.302689	0.697311	positive
12	1	baik seringkali muat homepage segar halaman se...		0.370205	0.629795	positive
...	...		...	...	...	...
13060	1		baik	0.110370	0.889630	positive
13061	1		baik	0.110370	0.889630	positive
13062	1		baik	0.110370	0.889630	positive
13063	1		keren	0.040863	0.959137	positive
13064	1		senang milik	0.163361	0.836639	positive

5099 rows × 5 columns

**Gambar 4.17 Hasil klasifikasi Naive Bayes dengan sentimen positif**

Hasil dari klasifikasi dokumen menggunakan Naive yang memiliki sentimen positif yang tertampil pada Gambar 4.17. Dari hasil klasifikasi tersebut terdapat 5099 dokumen yang diklasifikasikan sebagai sentimen positif.

	sentiment	stemming	prob_negative	prob_positive	klasifikasi_NB
0	0	sumpah aplikasi amatir verifikasi otp rusak ki...	0.905987	0.094013	negative
2	0	mutakhir aplikasi jengkel periksa transaksi up...	0.705515	0.294485	negative
4	0	aplikasi tendang aplikasi pasuk passcode gangg...	0.821187	0.178813	negative
5	0	aplikasi sebal informasi imbal lambat kali lan...	0.741733	0.258267	negative
6	1	aplikasi guna login pilih password lupa jutida...	0.754685	0.245315	negative
...	...	...	...	...	...
13044	0	berhenti	0.510267	0.489733	negative
13048	0	menjatidaklog hilang account imbang kartu bu c...	0.906508	0.093492	negative
13049	0	terjang	0.571375	0.428625	negative
13052	0	ya update baru reload pake kartu kredit hilang...	0.928673	0.071327	negative
13065	0	kartu gagal barista gesek coba starbucks cafe ...	0.719984	0.280016	negative

7967 rows × 5 columns

### Gambar 4.18 Hasil klasifikasi Naive Bayes dengan sentimen positif

Hasil dari klasifikasi dokumen menggunakan Naive yang memiliki sentimen negatif yang tertampil pada Gambar 4.18. Dari hasil klasifikasi tersebut terdapat 7967 dokumen yang diklasifikasikan sebagai sentimen negatif.

## 4.8 Evaluasi

Setelah melewati tahap *modelling* serta mendapatkan hasil dari model pada aplikasi Starbuck Indonesia, maka tahap selanjutnya adalah untuk mengevaluasi hasil dari model algoritma Naive Bayes tersebut menggunakan *confusion matrix*. Tahap ini juga bertujuan untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *f1-score*.

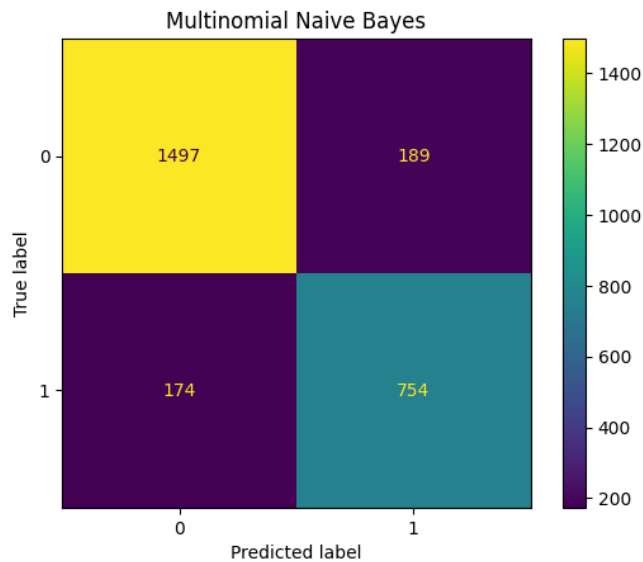
### 4.8.1 Hasil Evaluasi menggunakan Naïve Bayes

Dalam tahap pemodelan menggunakan Naïve Bayes pada aplikasi Starbucks Indonesia didapatkan nilai akurasi sebesar 86%. Hal ini menunjukkan bahwa persentase untuk mengidentifikasi data sebagai kelas positif atau kelas negatif dari total keseluruhan data adalah cukup baik. Sedangkan untuk mengukur kinerja dari algoritma Naïve Bayes itu sendiri dapat diukur dari nilai *f1-score* yang telah dihasilkan. Dalam hal ini, nilai *f1-score* yang didapatkan adalah 81% pada term positif dan 89% pada *term* negatif, pada aplikasi Starbucks Indonesia. Hasil *confusion matrix* dari pemodelan tersebut adalah sebagai berikut:



**Tabel 4.13 Hasil *Confusion Matrix* menggunakan Naive Bayes pada Aplikasi Starbucks Indonesia**

Aktual	Prediksi	
	Negatif	Positif
Negatif	1497	189
Positif	174	754



**Gambar 4.19 Hasil *Confusion Matrix* Model Naive Bayes pada Aplikasi Starbucks Indonesia**

Untuk menghitung nilai akurasi dari hasil klasifikasi tersebut digunakan Rumus (1.5) yang digunakan sebagai berikut:

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+FN+TN}$$

$$\text{Akurasi} = \frac{754+1497}{754+189+174+1497}$$

$$\text{Akurasi} = \frac{2251}{2614}$$

$$\text{Akurasi} = 0,8611$$

#### 4.8.2 Hasil Evaluasi ulasan positif menggunakan Naïve Bayes

Untuk menghitung nilai *precision* dari hasil klasifikasi tersebut digunakan Rumus (1.3) yang digunakan sebagai berikut:

$$Precision = \frac{TP}{TP+FP}$$

$$Precision = \frac{754}{754+189}$$

$$Precision = \frac{754}{943}$$

$$Precision = 0,7995$$

Untuk menghitung nilai *recall* dari hasil klasifikasi tersebut digunakan Rumus (1.4) yang digunakan sebagai berikut:

$$Recall = \frac{TP}{TP+FN}$$

$$Recall = \frac{754}{754+174}$$

$$Recall = \frac{754}{928}$$

$$Recall = 0,8125$$

Untuk menghitung nilai *f1-score* dari hasil klasifikasi tersebut digunakan Rumus (1.6) yang digunakan sebagai berikut:

$$F1-Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$$

$$F1-Score = 2 \times \frac{(0,8125 \times 0,7995)}{(0,8125 + 0,7995)}$$

$$F1-Score = 2 \times \frac{0,6495}{1,6120}$$

$$F1-Score = 0,8058$$

#### 4.8.3 Hasil Evaluasi ulasan negatif menggunakan *Naïve Bayes*

Untuk menghitung nilai *precision* dari hasil klasifikasi tersebut digunakan kebalikan dari Rumus (1.3) yang digunakan sebagai berikut:

$$Precision = \frac{TN}{TN+FN}$$

$$Precision = \frac{1497}{1497+174}$$

$$Precision = \frac{1497}{1671}$$

$$Precision = 0,8958$$

Untuk menghitung nilai *recall* dari hasil klasifikasi tersebut digunakan kebalikan dari Rumus (1.4) yang digunakan sebagai berikut:

$$Recall = \frac{TN}{TN+FP}$$

$$Recall = \frac{1497}{1497+189}$$

$$Recall = \frac{1497}{1,686}$$

$$Recall = 0,8879$$

Untuk menghitung nilai *f1-score* dari hasil klasifikasi tersebut digunakan kebalikan dari Rumus (1.6) yang digunakan sebagai berikut:

$$F1-Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$$

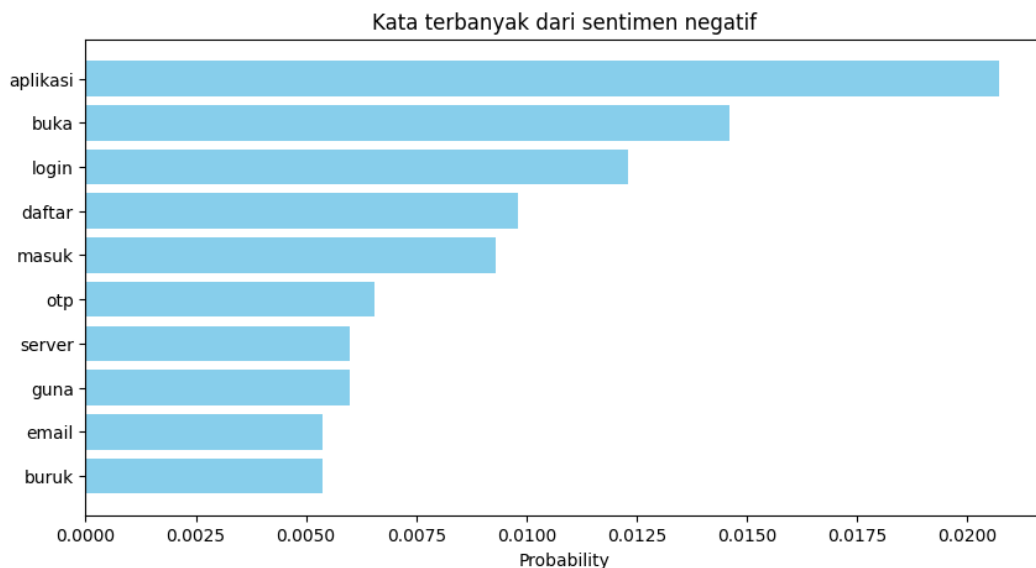
$$F1-Score = 2 \times \frac{(0,8879 \times 0,8958)}{(0,8879 + 0,8958)}$$

$$F1-Score = 2 \times \frac{0,7953}{1,7837}$$

$$F1-Score = 0,8917$$

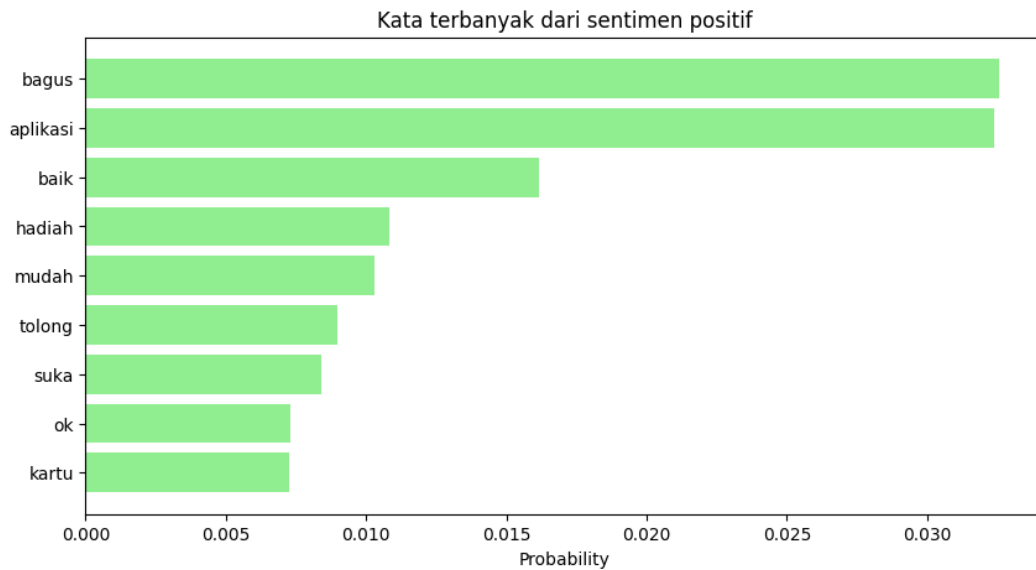
#### 4.9 Hasil dan Visualisasi

Hasil dari proses pengolahan data dari aplikasi Starbucks Indonesia memiliki visualisasi terhadap kata terbanyak dari kata ‘positif’ dan ‘negatif’.



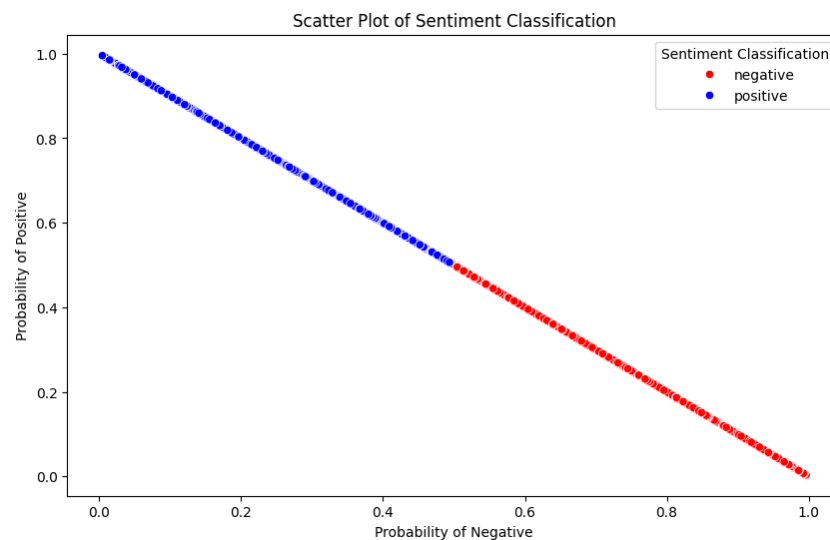
**Gambar 4.20 Hasil visualisasi data terhadap nilai “0” atau “Negatif”**

Dari kata terbanyak pada *term* negatif yang ada pada penelitian ini ditampilkan pada Gambar 4.20, yaitu terdapat kata “aplikasi”, “buka”, “login”, “masuk”, “buruk” dan “daftar” yang merupakan fitur dari aplikasi Starbucks Indonesia.



**Gambar 4.21 Hasil visualisasi data terhadap nilai “1” atau “Positif”**

Berikutnya pada kata terbanyak dari *term* positif yang ada pada penelitian ini ditampilkan pada Gambar 4.21, pada gambar tersebut ada kata “hadiah”, “kartu”, “baik”, dan “bagus” yang merupakan fitur dari aplikasi Starbucks Indonesia.



**Gambar 4.22 Hasil persebaran antara data positif dan negatif**

Pada Gambar 4.22 menampilkan persebaran data antara teks yang memiliki sentimen positif dan negatif. Dari diagram Scatter Plot tersebut dapat dilihat bahwa sentimen positif memiliki nilai probabilitas positif antara 0,5 sampai dengan 0,9 dan sentimen negatif memiliki nilai probabilitas 0,4 sampai dengan 0,1 berdasarkan pada perbandingan terhadap nilai probabilitas positif.

**Tabel 4.14 Hasil Nilai Probabilitas Pada Setiap Kata**

No.	word	avg_prob_negative	avg_prob_positive	classification
1	aplikasi	0.543050	0.456950	negative
2	login	0.742959	0.257041	negative
3	hadiah	0.232563	0.767437	positive
4	baik	0.315545	0.684455	positive
5	otp	0.794369	0.205631	negative
6	kartu	0.460156	0.539844	positive
...	...	...	...	...
5204	tombol	0.705384	0.294616	negative

Daftar nilai probabilitas dari tiap kata berdasarkan klasifikasi menggunakan Naive Bayes dapat dilihat pada Tabel 4.14 yang menampilkan nilai probabilitas tiap kata dalam dataset yang sudah diolah sehingga menghasilkan pengklasifikasian apakah kata yang diolah termasuk ke dalam sentimen positif atau negatif.

**Tabel 4.15 Hasil Nilai Probabilitas Pada Setiap Kata Dengan Klasifikasi Positif**

No.	word	avg_prob_negative	avg_prob_positive	classification
1	hadiah	0.232563	0.767437	positive
2	baik	0.315545	0.684455	positive
3	kartu	0.460156	0.539844	positive
4	bagus	0.13306	0.86693	positive
5	mudah	0.18271	0.817289	positive
...	...	...	...	...
1633	suka	0.234658	0.765342	positive

Hasil dari nilai probabilitas pada tiap kata yang memiliki sentimen positif ditampilkan pada Tabel 4.15. Dari tabel tersebut menampilkan terdapat 1633 kata yang terklasifikasikan sebagai sentimen positif.

**Tabel 4.16 Hasil Nilai Probabilitas Pada Setiap Kata Dengan Klasifikasi Negatif**

No.	word	avg_prob_negative	avg_prob_positive	classification
1	aplikasi	0.543050	0.456950	negative
2	login	0.742959	0.257041	negative
3	otp	0.794369	0.205631	negative
4	daftar	0.745819	0.254180	negative
5	email	0.783868	0.216132	negative
...	...	...	...	...
3571	buruk	0.740294	0.259705	negative

Hasil dari nilai probabilitas pada tiap kata yang memiliki sentimen negatif ditampilkan pada Tabel 4.16. Dari tabel tersebut menampilkan terdapat 3571 kata yang terklasifikasikan sebagai sentimen negatif.