

SENTIMEN ANALISIS PADA APLIKASI E-BRANCH BCA MENGGUNAKAN METODE NAÏVE BAYES

Vitri Tundjungsari¹ Samson Wijaya²

¹Universitas Esa Unggul

Informatics Engineering Study Program, Faculty of Computer Science,
9 Arjuna Utara Street, Duri Kepa, West Jakarta, Special Capital Region of Jakarta 11510, Indonesia.
e-mail: Vitri.tundjungsari@gmail.com, Samsonwijaya5@gmail.com

Abstrak

Aplikasi eBranch BCA adalah layanan perbankan digital yang disediakan oleh pihak Bank Central Asia (BCA), dengan lebih dari 1 juta unduhan dari Google Play Store, aplikasi ini menawarkan layanan perbankan digital. Meski populer, pertumbuhan ini menimbulkan anomali dalam ulasan pengguna yang tidak dapat dijadikan panduan bagi pengembangan aplikasi untuk mengembangkan aplikasi yang sesuai dengan keinginan pengguna. Identifikasi masalah melibatkan evaluasi tingkat akurasi metode Naïve Bayes dalam sentimen analisis dan pemahaman terhadap ulasan yang tidak konsisten. Tujuan penelitian adalah menganalisis sentimen pengguna dan mengklasifikasikan ulasan menggunakan metode Naïve Bayes. Penelitian melibatkan pengumpulan data, pra-pemrosesan, dan pengujian model. Hasil menunjukkan bahwa skenario 80% Data Training 20% Data Testing memberikan keseimbangan terbaik dan model dapat mengklasifikasikan sentimen dengan baik. Berdasarkan analisis confusion matrix, mayoritas nasabah mengekspresikan sentimen positif terhadap eBranch BCA, dengan sedikit ulasan netral dan kritik terhadap kemudahan penggunaan. Kesimpulan menunjukkan keberhasilan model Naïve Bayes dalam mengklasifikasikan sentimen, memberikan wawasan berharga untuk pengembangan aplikasi.

Kata kunci— Sentimen Analisis, eBranch, Aplikasi, Naïve Bayes, Google Play Store

Abstract

The BCA eBranch application is a digital banking service provided by Bank Central Asia (BCA), with more than 1 million downloads from the Google Play Store, this application offers digital banking services. Despite its popularity, this growth has led to anomalies in user reviews that cannot be used as a guide for application development to develop applications that match the user's wishes. Problem identification involves evaluating the level of accuracy of the Naïve Bayes method in sentiment, analysis and understanding of inconsistent reviews. The purpose of the study was to analyze user sentiment and classify reviews using the Naïve Bayes method. Research involves data collection, pre-processing, and testing of models. Results show that the 80% Data Training 20% Data Testing scenario provides the best balance and the model can classify sentiment well. Based on confusion matrix analysis, the majority of customers expressed positive sentiment towards eBranch BCA, with few neutral reviews and criticism of ease of use. The conclusions show the success of the Naïve Bayes model in classifying sentiment, providing valuable insights for application development

Keywords— Sentiment Analysis, eBranch, Application, Naïve Bayes, Google Play Store

1. PENDAHULUAN

Meningkatnya perkembangan teknologi pada aplikasi perbankan menjadi semakin populer dalam beberapa tahun terakhir termasuk aplikasi *eBranch* BCA. Aplikasi *eBranch* BCA adalah layanan perbankan digital yang disediakan oleh pihak Bank Central Asia (BCA), melalui perangkat *mobile* seperti *smartphone* atau *tablet*, sehingga nasabah dapat melakukan pengisian formulir transaksi dimana saja langsung dari *smartphone*. Sampai saat ini jumlah pengunduh aplikasi *eBranch* BCA sudah lebih dari 1 juta kali data ini bisa dilihat pada *platform Google Play Store*[1].

Jumlah ini terus bertambah mengingat BCA sendiri memiliki jumlah rekening nasabah sebanyak 37 juta dan ada 1.251 kantor cabang per 31 Juli 2023 data ini bisa dilihat langsung pada *website* resmi BCA[2], jika dihitung baru sekitar 500 cabang yang sudah merepakan penggunaan slip digital atau 39.97%, serta bertambahnya nasabah yang melakukan transaksi di cabang ada sekitar 100-250 di setiap cabangnya, jadi jumlah pengguna aplikasi *eBranch* terus meningkat setiap harinya dikarenakan pengalihan dari menggunakan slip transaksi manual yang menggunakan kertas menjadi slip digital yang melalui *smartphone*, hal ini menyebabkan anomali pada ulasan *Google Play Store*.

Ketika banyak pengguna memberikan ulasan dengan rating rendah seperti satu atau dua bintang tetapi teks ulasan yang diberikan bernilai positif atau pengguna memberikan rating tinggi tapi umpan baliknya bernilai negatif. Akibatnya *Google Play Store* tidak dapat membedakan ulasan positif dari ulasan negatif berdasarkan teks ulasan yang diberikan pengguna sehingga mempengaruhi tingkat kepercayaan pada aplikasi sebenarnya dan menjadi masalah ketika data tersebut masih belum terstruktur. Untuk menyelesaikan permasalahan tersebut diperlukan analisis yang dapat digunakan untuk menggali informasi dari data komentar dan ulasan pengguna aplikasi, Mengacu pada hal tersebut dilakukanlah penelitian ini yang bertujuan untuk menganalisis sentimen

pengguna terhadap aplikasi *eBranch* BCA dengan metode *Naïve Bayes* analisis ini memanfaatkan data ulasan pengguna aplikasi *eBranch* BCA.

Dalam prosesnya ada banyak jenis klasifikasi selain *Naïve Bayes* seperti algoritma Genetika, *K-nearest neighbors* (KNN), K-means, dan lainnya juga dapat digunakan untuk klasifikasi teks[3]. Tetapi *Naïve Bayes* secara luas digunakan dalam berbagai bidang, terutama untuk klasifikasi teks dan termasuk dalam *Supervised Learning* yang membutuhkan data pelatihan yang telah dilabeli[4]. Sistem klasifikasi menggunakan metode *Naïve Bayes*, khususnya *Multinomial Naïve Bayes* (MNB), digunakan untuk mendapatkan sentimen dari ulasan dan membandingkannya dengan rating bintang yang diberikan, *Multinomial Naïve Bayes* (MNB) sendiri difokuskan untuk klasifikasi teks dan probabilitas. Berdasarkan latar belakang yang telah peneliti jabarkan.

Maka adapun rumusan masalah yang akan diteliti yaitu, Bagaimana tingkat akurasi metode *Naïve Bayes* pada sentimen analisis pada aplikasi *eBranch* BCA menggunakan metode *Multinomial Naïve Bayes*. Dan terjadinya anomali pada ulasan *Google Play Store* sehingga data ulasan tidak bisa menjadi patokan dalam mengambil keputusan untuk pengembangan aplikasi oleh pihak developer.

2. METODE PENELITIAN

2.1 Naïve Bayes.

Dalam mendukung penelitian ini gunakan algoritma *Naïve Bayes* adalah salah satu Teknik klasifikasi yang menggunakan model probabilistik yang menggunakan model probabilistik yang menggunakan model probabilistik dan statistik. Algoritma ini didasarkan pada teorema Bayes yang disederhanakan dengan asumsi bahwa setiap atribut adalah independen. Artinya, *Naïve Bayes* menganggap bahwa keberadaan atau ketiadaan suatu karakteristik pada satu kelas tidak terkait dengan karakteristik kelas lainnya [5].

2.1.1 Multinomial Naïve Bayes

Multinomial Naïve Bayes adalah turunan dari algoritma *Naïve Bayes* dengan fitur yang unik, yaitu hasil klasifikasi untuk setiap kelas bersifat independen satu sama lain. Dalam kata lain, hasil klasifikasi untuk satu dokumen tidak bergantung pada dokumen lainnya, sehingga hasil yang didapat murni berasal dari dokumen yang diproses. Formula untuk *Multinomial Naïve Bayes* dapat dituliskan seperti yang tertera pada rumus (1) dan (2)[5].

$$P(c|d) \propto P(c) \prod_{i=1}^{n_d} P(w_i|c) \quad (1)$$

Keterangan:

$P(c|d)$ = probabilitas suatu kelas c pada dokumen / teks d
 $P(c)$ = probabilitas prior c
 $P(w_i|c)$ = probabilitas suatu kata pada kelas c

Perhitungan probabilitas sentimen positif atau sentimen negatif dari suatu ulasan yaitu:

$$P(S|d) =$$

Banyaknya kemunculan kata sentimen positif atau negatif pada text
Banyaknya kemunculan kata sentimen positif atau negatif pada text
 (2)

2.2 Metode Pengumpulan Data

Pada penelitian ini media pengumpulan data diambil dari aplikasi *eBranch* BCA pada *Google Play Store* menggunakan cara *scraping* yang menghasilkan data ±1000 data ulasan dalam bahasa Indonesia, data ulasan pengguna diunduh dari *platform* dalam bentuk teks berikut.

2.2.1 Preprocessing

Dataset yang belum diproses kemudian dilakukan *Preprocessing* melalui beberapa tahapan, di antaranya adalah pembersihan data (*Cleansing*), *Stemming*, penghapusan kata-kata penghubung (*stopwords-removal*), dan pemisahan kata (*tokenization*). Tujuan dari *Preprocessing* ini adalah untuk membersihkan dan mempersiapkan data agar siap digunakan dalam pelatihan model klasifikasi *Naïve Bayes*. Berikut merupakan contoh hasil dari *Preprocessing* data yang akan dilakukan[4][5].

2.3.2 Pembagian Data

Dari data tersebut nantinya akan masuk pada tahap Pembagian dataset dilakukan untuk memisahkan data ulasan menjadi tiga kategori: negatif, positif, dan netral. Selanjutnya, setiap kategori akan dibagi menjadi dataset *training* dan dataset *testing*. Dataset *training* digunakan untuk melatih model klasifikasi sebanyak 80%, sementara dataset *testing* digunakan untuk menguji performa model sebanyak 20%. Tujuan pembagian dataset ini adalah untuk melatih model secara optimal dan menguji kemampuannya dalam mengklasifikasikan sentimen ulasan.

2.3.3 Term Frequency - Inverse Document Frequency (TF - IDF)

Setelah dataset *training* melalui proses *Preprocessing*, langkah selanjutnya adalah mengubah ulasan yang telah diproses menjadi representasi numerik menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Metode TF-IDF digunakan untuk memberikan bobot pada setiap kata dalam ulasan berdasarkan frekuensi kemunculannya dan pentingnya dalam keseluruhan dataset. Hal ini memungkinkan model klasifikasi untuk mengidentifikasi kata-kata yang lebih relevan dalam proses klasifikasi sentiment bisa dilihat pada rumus (1)[6].

$$TF - IDF(d, t) = TD(d, t) * IDF(t) \quad (1)$$

Dimana:

$$TF(d, t) = \frac{\text{Jumlah kata } t \text{ pada dokumen } d}{\text{total kata pada dokumen } d}$$

$$IDF(t) = \log \frac{\text{total dokumen}}{\text{jumlah dokumen yang mengandung kata } t}$$

Dan:

t = kata

d = dokumen

2.3.4 Klasifikasi Naïve Bayes

Dataset yang telah melalui *Preprocessing* dan pembobotan kata menggunakan TF-IDF kemudian diolah menggunakan klasifikasi *Naive Bayes*. *Naive Bayes* menggunakan *training Multinomial Naïve Bayes* dan pertama-tama pembuatan *bag of words* untuk mengumpulkan *term* pada ulasan aplikasi

eBranch BCA berdasarkan frekuensi kemunculan ulasan tersebut. Setelah itu dilakukan tahap *training* untuk mendapatkan model klasifikasi dalam bentuk probabilitas. Agar hasil *testing* nantinya tidak terpengaruh oleh peluang 0 yang terjadi pada *training*, diperlukan Laplace smoothing. Hal ini dilakukan dengan menambahkan angka 1 dibagi dengan jumlah semua fitur yang ditambahkan ke semua fitur, sehingga tidak ada fitur bernilai 0.

2.3.5 Confusion Matrix

Pada tahap *Confusion Matrix* digunakan tabel untuk mengevaluasi performa model dengan menunjukkan seberapa akurat klasifikasi yang dilakukan dengan melihat seberapa baik model dapat membedakan sentimen positif dan negatif. Dengan melihat nilai TP, TN, FP, dan FN, serta dapat menghitung metrik evaluasi yang memberikan informasi tentang tingkat keberhasilan dan akurasi model dalam melakukan prediksi berikut adalah gambaran model tabelnya. Kemudian data akan diuji pada model dan model memprediksi kelas dari data uji tersebut. Prediksi ini kemudian dibandingkan dengan kelas sebenarnya dari data uji untuk menghasilkan matrik evaluasi seperti tabel 1.

Table 1. Confusion Matrix

Classification	Predicted class	
	Class = Yes	Class = No
Class = Yes	a (True Positive)	b (False Negative)
Class = No	c (False Positive)	d (True Negative)

2.3.6 Analisis

Setelah model klasifikasi *Naive Bayes* diuji dan menghasilkan *Confusion Matrix*, hasil tersebut akan dianalisis dan diinterpretasikan dalam bentuk diagram pie yang berisikan nilai persentase ulasan pendapat nasabah Bank Central Asia pada *Google Play Store* yang menggunakan aplikasi *eBranch* BCA.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Proses pengambilan data ulasan *eBranch* melibatkan penggunaan *library* *Google-Play Scraper* yang terintegrasi dengan bahasa pemrograman *Python* implementasi bisa dilihat pada gambar 1.

```
import json
import pandas as pd
from tqdm import tqdm

import seaborn as sns
import matplotlib.pyplot as plt

from pygments import highlight
from pygments.lexers import JsonLexer
from pygments.formatters import TerminalFormatter

from google_play_scraper import Sort, reviews, app
```

Gambar 1. Import Library Google Play

Scraper

```
app_packages = [
    'com.bca.smartbranch'
]

app_reviews = []

for ap in tqdm(app_packages):
    for score in list(range(1, 6)):
        for sort_order in [Sort.MOST_RELEVANT, Sort.NEWEST]:
            rvs, _ = reviews(
                ap,
                lang='id',
                country='id',
                sort=sort_order,
                filter_score_with=score
            )
            for r in rvs:
                r['sortOrder'] = 'most_relevant' if sort_order == Sort.MOST_RELEVANT else 'newest'
                r['appId'] = ap
                app_reviews.extend(rvs)
```

Gambar 2. Scraping Data

Setelah tahap pengaturan *library* selesai, langkah berikutnya adalah memasuki proses pengambilan data atau *scraping*. Pada Gambar 2, dapat dilihat alamat aplikasi *eBranch* yang digunakan dalam crawling, yaitu *"com.bca.smartbranch"*.

```
app_reviews_df = pd.DataFrame(app_reviews)
app_reviews_df.to_csv('ulasan2.csv', index=None, header=True)
```

Gambar 3. Penambahan Fungsi Simpan

Selanjutnya, akan ditambahkan fungsi penyimpanan untuk hasil *crawling* dengan format file CSV dan dinamakan

“ulasan2.csv” seperti yang terlihat pada Gambar 3

userName	content	score	at
pariani gusti ayu kade	Sangat puas pelayanan jadi cepat	5	11/10/2023 10:00
pariani gusti ayu kade	Sangat puas pelayanan jadi cepat	5	11/10/2023 10:00
Aji Yoey	Baru install sudah tunggu beberapa saat lagi.	1	11/10/2023 08:58
Aji Yoey	Baru install sudah tunggu beberapa saat lagi.	1	11/10/2023 08:58
Asiang kepalatoko	Gimana sih udah di ganti namanya tetap saja dikomen dari teller. Suruh update akun terus	1	10/10/2023 13:29
Asiang kepalatoko	Gimana sih udah di ganti namanya tetap saja dikomen dari teller. Suruh update akun terus	1	10/10/2023 13:29
roni @caRonid@ abadi	Sangat lemah dan sangat disayangkan, Aplikasi masih perlu pembenahan, 1 minggu pakai aplika	1	11/09/2023 12:26
roni @caRonid@ abadi	Sangat lemah dan sangat disayangkan, Aplikasi masih perlu pembenahan, 1 minggu pakai aplika	1	11/09/2023 12:26
Toni Istanto	Luar biasa aplikasi sangat membantu g usah lama2 nulis	5	08/09/2023 10:42
Toni Istanto	Luar biasa aplikasi sangat membantu g usah lama2 nulis	5	08/09/2023 10:42
Vianite	Detail menu di aplikasi tidak selengkap yg di web. Di web ada pilihan No. Rekening, ktp, paspor, i	2	01/09/2023 09:07
Vianite	Detail menu di aplikasi tidak selengkap yg di web. Di web ada pilihan No. Rekening, ktp, paspor, i	2	01/09/2023 09:07
jamur keras	skrgbkok jd susah kbuka nya y.mloading teroooooss...sampe brjam2...harap perbaiki	3	25/08/2023 10:37
jamur keras	skrgbkok jd susah kbuka nya y.mloading teroooooss...sampe brjam2...harap perbaiki	3	25/08/2023 10:37
Mucisun nganjuk	Aplikasinya sangat bagus untuk keamanan para nasabah BCA simpel dan mudah di gunakan	4	25/08/2023 10:23
Mucisun nganjuk	Aplikasinya sangat bagus untuk keamanan para nasabah BCA simpel dan mudah di gunakan	4	25/08/2023 10:23
Amir Moe	masih harus di perbaiki dari segi keamanan sangat kurang.karna aplikasi dapat dengan mudah d	1	22/08/2023 12:34
Amir Moe	masih harus di perbaiki dari segi keamanan sangat kurang.karna aplikasi dapat dengan mudah d	1	22/08/2023 12:34
Navawi	Bagi saya Dg adanya aplikasi ini sangat membantu. Untuk kedepannya tlg ditambah fitur penged	3	21/08/2023 14:40
Navawi	Bagi saya Dg adanya aplikasi ini sangat membantu. Untuk kedepannya tlg ditambah fitur penged	3	21/08/2023 14:40
Irtan Kart	Teller tidak bisa mendeteksi kesalahan penulisan perbedaan nama dengan no rekening dari aplik	1	08/08/2023 09:32
Irtan Kart	Teller tidak bisa mendeteksi kesalahan penulisan perbedaan nama dengan no rekening dari aplik	1	08/08/2023 09:32
Tukang Upload	bank swasta terbaik service nya memuaskan	5	30/07/2023 15:18
Tukang Upload	bank swasta terbaik service nya memuaskan	5	30/07/2023 15:18
Yusuf Mhmmid	Setelah di update kok gabis dibuka sih	2	29/07/2023 21:44
Yusuf Mhmmid	Setelah di update kok gabis dibuka sih	2	29/07/2023 21:44
tedi bahdir	harus ada update lagi biar setoran kurang dari Rp 50.000 bisa di setor lewat aplikasi ebranch. k	5	28/07/2023 21:14

Gambar 4. Hasil Scraping

Pada proses pengambilan data ulasan sebanyak 1000 data ulasan terkumpul untuk data yang digunakan mulai dari tanggal 13/09/2018 hingga 11/10/2023 bisa dilihat pada Gambar 4.

3.2 Preprocessing

Tahap ini untuk memilih kata yang akan digunakan sebagai indeks, serta membuang data yang tidak diperlukan seperti tabel 'at', 'reviewId', 'userName', 'userImage', 'thumbsUpCount', 'reviewCreatedVersion', 'replyContent', 'repliedAt', 'sortOrder', 'appId' pada file data csv.

```
#Menghapus kolom yang tidak diperlukan
df_preprocessed = review.copy()
df_preprocessed = df_preprocessed.drop(columns=['at', 'reviewId', 'userName', 'userImage', 'thumbsUpCount', 'reviewCreatedVersion', 'replyContent', 'repliedAt', 'sortOrder', 'appId'])
df_preprocessed.head()
```

Gambar 5. Penambahan Fungsi Hapus data

Setelah dilakukan penghapusan data yang tidak diperlukan maka yang tersisa dari data ulasan hanyalah tabel content dan label seperti gambar 5 Data yang sudah didapat akan dilakukan balancing dataset agar setiap data yang ada memiliki jumlah yang sama atau setara sebelum nanti diolah pada tahap selanjutnya pada gambar 6

```
s_1 = df_preprocessed[df_preprocessed['label']==0].sample(330,replace=True)
s_2 = df_preprocessed[df_preprocessed['label']==1].sample(330,replace=True)
s_3 = df_preprocessed[df_preprocessed['label']==-1].sample(330,replace=True)
df_preprocessed = pd.concat([s_1, s_2, s_3])

print(df_preprocessed.shape)
print(df_preprocessed['label'].value_counts(normalize=True))
```

Gambar 6. Balancing Dataset

3.2.1 Case Folding

Pada proses dilakukan penyeragaman huruf atau mengubah bentuk semua bentuk huruf menjadi lowercase atau mengubah bentuk huruf kapital menjadi huruf kecil, implementasi program dan hasil dari proses ini seperti gambar 7 dan hasilnya pada tabel 3.

```
df_preprocessed.loc[:, "content"] = df_preprocessed.content.apply(lambda x : str.lower(x))

print('Case Folding Result : \n')
print(df_preprocessed["content"].iloc[75])
```

Gambar 7. Case Folding

Tabel 1. Case Folding

Sebelum Case Folding	Sesudah Case Folding
"Aplikasi EBRANCH sangat BAIK dan MUDAH digunakan!"	"aplikasi ebranch sangat baik dan mudah digunakan!"
"Pengalaman menggunakan EBranch luar biasa."	"pengalaman menggunakan ebranch luar biasa."
"Tolong perbaiki bug pada Aplikasi EBRANCH."	"tolong perbaiki bug pada aplikasi ebranch."
"Aplikasi EBranch susah diakses."	"aplikasi ebranch susah diakses."

3.2.2 Cleansing

Pada langkah ini data akan diolah untuk menghilangkan karakter yang kurang penting. Karakter-karakter tersebut meliputi angka (0,1,2,3,4,5,6,7,8,9), tanda pagar (#), akun pengguna (@username), alamat situs ("http://", " "), tanda baca (~!@#\$%^&*:"<>?()), serta karakter lain yang tidak relevan implementasi pada gambar 8 dan hasilnya pada tabel 4.


```
def remove_word_special(text):
    # remove tab, new line, ans back slice
    text = text.replace('\t', " ").replace('\n', " ").replace('\u', " ").replace('\ ', " ")
    # remove non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')
    # remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\w+\/\S+)", "", text).split())
    # remove incomplete URL
    return text.replace("http://", " ").replace("https://", " ")

df_preprocessed["content"] = df_preprocessed["content"].apply(remove_word_special)
```

Gambar 8. Cleansing Data

Tabel 2 Hasil Cleansing Data

Sebelum Cleansing	Sesudah Cleansing
"E-Branch suka hang dan sering error."	"ebranch suka hang dan sering error."
"E-Branch nggak responsif, susah banget diakses"	"ebranch nggak responsif, susah banget diakses"
"E-Branch banyak bugnya, kesel deh!"	"ebranch banyak bugnya, kesel deh"
"Ok Mantap Sekali EBranch Teruskan!!!1"	"ok mantap sekali ebranch teruskan"

3.2.3 Tokenizing

Setelah proses pembersihan, pada tahap *Tokenizing*, teks yang datang dari satu kalimat akan dipecah menjadi komponen-komponen yang lebih kecil. memisahkannya berdasarkan spasi dan tanda baca yang ada implementasi pada gambar 9 dan hasilnya pada tabel 5.

```
# NLTK word tokenize
def word_tokenize_wrapper(text):
    return word_tokenize(text)

df_preprocessed["content_tokens"] = df_preprocessed["content"].apply(word_tokenize_wrapper)

print('Tokenizing Result : \n')
print(df_preprocessed["content_tokens"].head())
print('\n\n')
```

Gambar 9. Tokenizing

Tabel 3. Hasil Tokenizing

Sebelum Tokenizing	Sesudah Tokenizing
aplikasi ebranch sangat membantu.	"aplikasi", "ebranch", "sangat", "membantu"
ebranch memberikan layanan yang baik.	"ebranch", "memberikan", "layanan", "baik"
ebranch keren	"ebranch", "keren"

3.2.4 Filtering

Filtering merupakan tahap dimana dihilangkannya kata-kata tidak penting atau stopword yang memiliki frekuensi kemunculan yang tinggi seperti kata hubung, kata-kata ini biasanya disimpan ke dalam stop list, implementasi dapat dilihat pada Gambar 10.

```
from nltk.corpus import stopwords

# ----- get stopword from NLTK stopword -----
# get stopword indonesia

nltk.download('stopwords')

list_stopwords = stopwords.words('indonesian')
list_stopwords = set(list_stopwords)

#remove stopword pada list token
def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]

df_preprocessed['content_tokens_SW'] = df_preprocessed['content_tokens'].apply(stopwords_removal)

print(df_preprocessed['content_tokens_SW'].head())
```

Gambar 10. Filtering

Terlihat perbedaan pada Tabel 6 antara teks sebelum dan sesudah dimana pada teks yang belum diproses terdapat kata ‘Jadi, ‘harusnya’, ‘tetap’, dan ‘ada’ sedangkan pada teks result kata-kata tersebut dihilangkan karena termasuk dalam daftar kata *stopwords*.

Tabel 4. Hasil Stopwords

Sebelum Stopwords	Sesudah Stopwords
'jadi', 'ribet', 'keterlaluhan', 'harusnya', 'opsi', 'manual', 'tetap', 'ada'	'ribet', 'keterlaluhan', 'opsi', 'manual'

3.2.5 Stemming

Pada penelitian ini proses *Stemming* dibangun dengan menggunakan *library* dari *sastrawi*, Pengimplementasian dari proses ini dapat dilihat implementasi program dan hasil dari proses seperti Gambar 11.

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter

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

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

term_dict = {}

for document in df_preprocessed['content_tokens_SW']:
    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("-----")

# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

df_preprocessed['content_stemmed'] = df_preprocessed['content_tokens_SW'].swifter.apply(get_stemmed_term)
print(df_preprocessed['content_stemmed'])

df_preprocessed.to_csv("Text_Preprocessing.csv")
```

Gambar 11. Stemming

Dibuat sebuah *function* yang yang mengambil *raw text* (teks yang belum di *preprocess*) sebagai inputnya dan mengembalikan teks yang sudah melalui proses *Stemming* sebagai outputnya.

3.3 Pembagian Dataset

Pembagian dataset menggunakan fungsi *train_test_split* dari *library sklearn* dengan mengambil sebagian dari dataset sebagai data train dan sebagian lagi sebagai data test. pembagian dataset seperti gambar 12.

```
df_preprocessed = pd.read_csv("Text_Preprocessing.csv")
X = df_preprocessed['content_stemmed']
y = df_preprocessed['label']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.40, stratify=df_preprocessed["label"], random_state=42)
```

Gambar 12. Pembagian Dataset

Pada implementasi pembagian dataset hasil pembagian dataset disimpan ke dalam 4 objek yaitu *X_train* berisi feature untuk *training*, *X_test* berisi feature untuk *testing*, *Y_train* berisi label untuk *training*, dan *Y_test* berisi label untuk *testing*.

3.4 Term Frequency - Inverse Document Frequency

Proses *Term Frequency - Inverse Document Frequency* dilakukan dengan menggunakan *CountVectorizer* untuk mengubah ulasan yang menjadi representasi *bag of words*. Dapat dilihat implementasi program seperti ini gambar 13.

```
from sklearn.feature_extraction.text import TfidfVectorizer

# contoh
d0 = df_preprocessed['content_tokens_SW'].iloc[25]
d1 = df_preprocessed['content_tokens_SW'].iloc[12]

# menggabungkan dokumen jadi satu corpus
corpus = [d0, d1]

# Buat object
vectorizer = TfidfVectorizer()

#tf-idf values
contoh = vectorizer.fit_transform(corpus)

#idf values
print('\nidf values:')
for ele1, ele2 in zip(vectorizer.get_feature_names_out(), vectorizer.idf_):
    print(ele1, ':', ele2)

#indexing
print('\nWord indexes:')
print(vectorizer.vocabulary_)

# Menampilkan tf-idf values
print('\ntf-idf value:')
print(contoh)

# dalam bentuk matrix
print('\ntf-idf values in matrix form:')
print(contoh.toarray())
```

Gambar 13. Term Frequency - Inverse

Document Frequency

Pada implementasinya ulasan diubah ke dalam representasi *bag of words* kemudian diubah menjadi *vector* angka.

3.5 Confusion Matrix

Untuk mengetahui tingkatan dari performa Algoritma Klasifikasi yang telah dilakukan, maka perlu melakukan pengujian terhadap model. Hasil dari implementasinya akan terlihat seperti gambar 14.

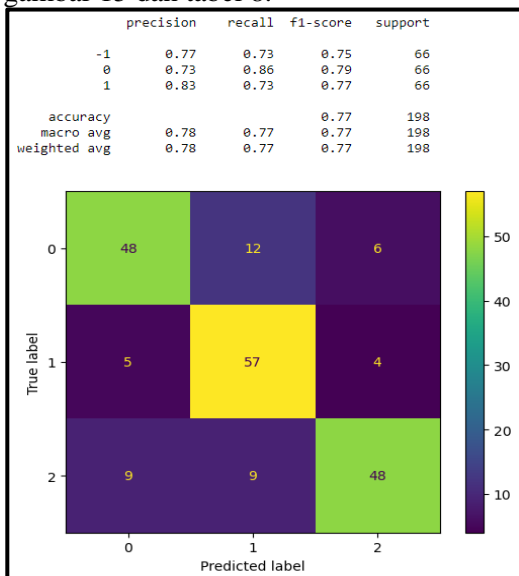
```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.metrics import classification_report
predict = clf.predict(X_test)
cm = confusion_matrix(y_test, predict)

cm_display = ConfusionMatrixDisplay(cm).plot()
print(classification_report(y_test, predict))
```

Gambar 14. Confusion Matrix

3.5.1 Skenario 80% Data Training dan 20% Data Testing

Pada skenario pertama dengan sentimen positif, negatif, dan netral serta dataset yang digunakan sejumlah 80% *data training* dan 20% *data testing*, dengan *data testing* sebanyak 200 dan *data training* sebanyak 800 bisa dilihat hasilnya pada gambar 15 dan tabel 8.



Gambar 15. Hasil Pengujian Skenario

Pertama

Dari hasil ini, dapat diambil beberapa kesimpulan:

1. Sentimen Negatif:

- Presisi (0.77): Ini berarti 77% dari yang diklasifikasikan sebagai sentimen negatif adalah benar-benar negatif.
- Recall (0.73): Hanya 73% dari semua sentimen negatif yang sebenarnya berhasil ditemukan oleh model.
- F1-Score (0.75): Ini adalah rata-rata harmonik dari presisi dan recall, yang menunjukkan performa yang baik dalam mengklasifikasikan sentimen negatif.

2. Sentimen Netral:

- Presisi (0.73): Model memiliki presisi sebesar 73% dalam mengklasifikasikan sentimen netral.
- Recall (0.86): Ini berarti model dapat mendeteksi 86% dari sentimen netral yang sebenarnya.
- F1-Score (0.79): Skor F1 yang baik (0.79) mengonfirmasi performa yang baik dalam mengklasifikasikan sentimen netral.

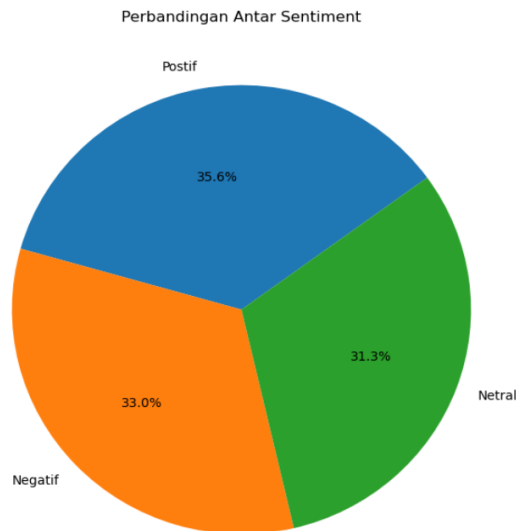
3. Sentimen Positif:

- Presisi (0.83): Model memiliki presisi sebesar 83% dalam mengklasifikasikan sentimen positif.
- Recall (0.73): 73% dari sentimen positif yang sebenarnya berhasil ditemukan oleh model.
- F1-Score (0.77): F1-Score yang baik (0.77) menunjukkan performa yang baik dalam mengklasifikasikan sentimen positif.

Secara keseluruhan, hasil "*Macro averaging*" menunjukkan rata-rata performa yang sangat baik dan seimbang di antara ketiga sentimen.

3.6 Hasil Analisis

Berdasarkan tiga hasil percobaan *confusion matrix*, dipilih skenario 80/20 sebagai hasil dari penelitian ini, dipilih karena memberikan keseimbangan yang paling baik diantara tiga skenario dalam ukuran data pelatihan dan pengujian. Dengan begitu model yang di buat dengan menggunakan algoritma *Naïve Bayes* dinyatakan berhasil dan dapat mengklasifikasi sentiment dengan baik, kemudian hasil penelitian yang didapat terkait sentiment analisis pada aplikasi *eBranch* BCA bisa dilihat pada gambar 16.



Gambar 16. Hasil Analisa

Hasil akhir dari penelitian ini menunjukkan bahwa dari 1000 data yang dimiliki, sentimen positif tercatat sebesar 35.6% atau 356 data, yang menyatakan bahwa nasabah senang dengan hadirnya aplikasi *e-Branch* BCA ini. sementara sentimen netral mencapai 31.3% atau 313 data, yang menyatakan bahwa nasabah merasa biasa saja dan tidak ada perubahan dengan hadirnya aplikasi ini dan sentimen negatif mencapai 33% atau 330 data. Namun, perlu dicatat bahwa data negatif memiliki variasi, dimana sebanyak 12% atau 129 data ulasan yang menyatakan keberatan terhadap fitur reservasi aplikasi *e-Branch* karena tidak sesuai dan aplikasi yang sering memunculkan informasi update aplikasi yang bisa dilihat pada gambar 28. Sementara itu, sisanya sebanyak 21% atau 210 menyampaikan keluhan terkait dengan keberadaan bug atau kesalahan teknis yang mengakibatkan mereka kesulitan dalam menggunakan aplikasi yang bisa dilihat pada gambar 29.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Hasil penerapan dari penelitian yang telah dilakukan dengan menggunakan metode Naïve Bayes dapat dijadikan pilihan sebagai

penyelesaian masalah mengenai klasifikasi analisis sentimen pada aplikasi *eBranch* BCA. Hal ini dibuktikan berdasarkan nilai dari akurasi dan confusion matrix yang dihasilkan.

2. Dari hasil implementasi klasifikasi analisis sentimen diketahui bahwa nasabah melakukan transaksi merasa senang dengan adanya aplikasi *eBranch* BCA, hal ini dibuktikan dengan hasil sentimen terbesar yaitu positif diikuti dengan negatif dan terkecil pada sentimen netral.
3. Sentimen negatif yang dikemukakan oleh nasabah bukan karena penerapan penggunaan aplikasi dalam bertransaksi saja namun juga karena adanya kekurangan dalam aplikasi seperti fitur yang dianggap kurang efektif dan bugs pada aplikasi yang masih sering terjadi.

5. SARAN

Penelitian ini telah memberikan wawasan tentang kinerja model klasifikasi sentimen dalam konteks data ini. Namun, masih ada beberapa aspek yang dapat dieksplorasi lebih lanjut untuk penelitian selanjutnya. Beberapa saran untuk penelitian selanjutnya meliputi:

1. Melakukan peningkatan dalam pengumpulan data pelatihan yang lebih representatif, serta melakukan eksperimen dengan berbagai model klasifikasi seperti K-Means dan lainnya untuk mendapatkan hasil yang lebih akurat.
2. Mengeksplorasi teknik pra-pemrosesan yang lebih canggih untuk meningkatkan kinerja model. Pengembangan terhadap model agar dapat meminimalisir tingkat kesalahan. Sehingga penelitian

selanjutnya dapat memperdalam pemahaman tentang analisis sentimen dalam konteks yang lebih luas dan beragam.

DAFTAR PUSTAKA

- [1] “<https://play.google.com/store/apps/details?id=com.bca.smartbranch> Accessed on 12/05/2023.”
- [2] “<https://www.bca.co.id/id/tentang-bca> Accessed on 12/05/2023.”
- [3] M. T. Anwar and D. R. A. Permana, “Perbandingan Performa Model Data Mining untuk Prediksi Dropout Mahasiswa,” *J. Teknol. dan Manaj.*, vol. 19, no. 2, 2021, doi: 10.52330/jtm.v19i2.34.
- [4] B. M. Pintoko and K. M. L., “Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naive Bayes Classifier,” *e-Proceeding Eng.*, vol. 5, no. 3, pp. 8121–8130, 2018.
- [5] M. I. Fikri, T. S. Sabrila, and Y. Azhar, “Perbandingan Metode Naive Bayes dan Support Vector Machine pada Analisis Sentimen Twitter,” *SMATIKA J.*, vol. 10, no. 02, pp. 71–76, 2020, doi: 10.32664/smatika.v10i02.455.
- [6] F. V. Sari and A. Wibowo, “Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naive Bayes Classifier Berbasis Konversi Ikon Emosi,” *J. SIMETRIS*, vol. 10, no. 2, pp. 681–686, 2019.