Jurnal Smart Teknologi
Vol. 3, No. 6, September 2022, Halaman 612 – 623
ISSN: 2774-1702, http://jurnal.unmuhjember.ac.id/index.php/JST

Analisis Sentimen Pada Pelayanan Jaringan Internet Indihome Dengan Metode Multinomial Naïve Bayes Masa Pandemi Covid-19

Sentiment Analysis on Indihome Internet Network Services Using The Multinomial Naïve Bayes Method During The Covid-19 Pandemic

Dion Reddy [1)], Deni Arifianto [2)*], Dewi Lusiana [3)]
[1]Mahasiswa Program studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember
email: dionreddys@gmail.com
[2] Dosen Fakultas Teknik, Universitas Muhammadiyah Jember*Koresponden Author
email: deniarifianto@unmuhjember.ac.id
[3] Dosen Fakultas Teknik, Universitas Muhammadiyah Jember
email: dewilusiana@unmuhjember.ac.id

Abstrak

Pada saat pandemi Internet sangat dibutuhkan masyarakat karena semua pekerjaan dilakukan dirumah aja. Salah satu pelayanan jaringan internet di Indonesia adalah Indihome. Pengguna Indihome pun semakin banyak dengan imbasnya pandemi covid-19. Dengan semakin bertambahnya pengguna maka, banyak pula yang berkomentar tentang pelayanan internet di salah satu platform sosial media seperti *Facebook*. Penelitian ini bertujuan melakukan analisis sentimen pada komentar *Facebook* yang diambil dari *fanspage* Indihome. Menganalisis sentimen komentar masyarakat dengan klasifikasi metode *Naïve Bayes*. *Dataset* yang diambil menggunakan aplikasi facepager berjumlah total 854 data yang diambil pada saat masa pandemi. Kemudian dilabelin manual oleh pakar Bahasa dan dilakukan proses *Text Mining*. Sentimen pada dataset mempunyai 3 kelas yaitu Positif, Negatif, dan Netral. *Tools* yang digunakan untuk menghitung *Naïve Bayes* adalah *Python*. Mengklasifikasi dengan metode *Naïve Bayes* di prediksi menggunakan 4 Skenario dengan total 21 Akurasi. Dari 21 kali percobaan menghasilkan akurasi tertinggi menggunakan 10 fold iterasi 8 dengan Nilai sebesar 85.8%. Dari hasil uji coba dapat disimpulkan bahwa rata-rata tingkat akurasi cenderung meningkat dengan bertambahnya data *training*.

***Keywords**: Naïve Bayes, Covid-19, Indihome, Analisis Sentimen, Komentar Facebook.*

Abstract

During the internet pandemic, people really need it because all work is done at home. One of the internet network services in Indonesia is Indihome. Indihome users are also increasing with the impact of the covid-19 pandemic. With the increase in the number of users, many also comment about internet services on social media platforms such as Facebook. This study aims to analyze the sentiment on Facebook comments taken from the Indihome fanspage. Analyzing the sentiment of public comments using the Naïve Bayes classification method. The dataset taken using the Facepager application is 854 data taken during the pandemic. Then it is manually labeled by a linguist and the Text Mining process is carried out. Sentiment in the dataset has 3 classes, namely Positive, Negative, and Neutral. The tool used to calculate Naïve Bayes is Python. Classification using the Naïve Bayes method is predicted to use 4 Scenarios with a total of 21 Accuracy. From 21 trials, the highest accuracy was obtained by using 10-fold iterations 8 times with a value of 85.8%. From the test results, it can be concluded that the average level of accuracy tends to increase with increasing training data.

***Keywords**: Naïve Bayes, Covid-19, Indihome, Sentiment Analysis, Facebook Comments.*

Jurnal Smart Teknologi
Vol. 3, No. 6, September 2022, Halaman 612 – 623
ISSN: 2774-1702, http://jurnal.unmuhjember.ac.id/index.php/JST

1. PENDAHULUAN

Merujuk data yang dikeluarkan *we are social and hootsuite*, jumlah yang memakai sosial media aktif di Indonesia meraih 160 juta pengguna. Dapat disimpulkan bahwa banyak aktivitas yang dihabiskan penduduk Indonesia dalam mengakses sosial media. *Facebook* merupakan urutan ketiga dalam kategori sosial media *platform* yang dikutip *we are social and hootsuite*. Jadi *Facebook* juga salah satu sosial media yang banyak diakses dan terjadi banyak interaksi didalamnya (Simon K, 2020). Apalagi, selama masa pandemi Indonesia yang dimulai pada bulan Maret 2020 hingga sekarang dimana kebijakan pemerintah untuk membatasi segala pergerakan masyarakat membuat semua hal yang biasanya dilakukan secara luring, kini semua harus secara daring. Banyak aktivitas dilakukan dirumah dengan memperlakukan *Work From Home*, hal ini tentu memiliki imbas dengan semakin banyak penggunaan layanan internet pada perusahaan telekomunikasi (Cindy M A, 2020).

Telkom Indihome yang memang sudah memiliki rekam jejak lama tentu ada banyak kendala yang dialami pada perusahaan, contoh kejadian yang terjadi pada Kamis (13/8/2020) dimana terjadi gangguan secara massal, Menurut kutipan dari Tivan Rahmat dalam situsnya https://www.suara.com/tekno/2020/08/13/160106/telkom-temukan-penyebab-gangguan-indihome-minta-maaf-ke-pelanggan bahwa diakses pada 1 Mei 2021 memuat berita bahwa dari salah satu *Domain Name Server*(DNS) itu tidak berjalan wajar pada Telkom Indihome dan menyebabkan banyak komentar dari pengguna layanan.

Proses *crawling* dapat dilakukan untuk mengambil data atau konten yang ada di sosial media. Hasil pengambilan data yang dilakukan tentu masih berupa data mentah dan kotor yang perlu dilakukan pengolahan data seperti tahapan *pre processing* dan berbagai tahapan sehingga dapat dihasilkan informasi baru yang bermanfaat. Sentimen merupakan sebuah proses komputasi dalam membuat kategori dan identifikasi opini-opini berbentuk potongan teks, yang dikhususkan untuk mengukur tujuan dari si pembikin penggalan teks yang ditujukan pada topik tertentu, bisa bersifat positif, negatif, maupun netral (Monarizqa et al., 2014).

Penelitian ini memakai metode *Naïve Bayes* ada beberapa yang menguatkan penulis mengambil cara ini diantaranya dari tulisan Ronny Julianto yang melakukan proses pengklasifikasian menggunakan algoritma *Naïve Bayes* pada data perusahaan *provider* telepon seluler menghasilkan akurasi sebesar 79% (Ronny J, Evi DB, 2017).

2. LANDASAN TEORI

a. Data Mining

Data Mining ialah cara dengan mendapatkan hubungan atau gambaran mulai ratusan hingga ribuan *field* pada sesuatu basis data yang luas. *Data Mining* biasanya disebut juga jajaran proses dengan menggali nilai tambah berwujud keterangan yang semasa ini belum diketahui. Keterangan yang dihasilkan didapat dengan macam mengekstrak dan melihat pola utama atau menarik dari data pada basis data. *Data Mining* biasanya dipakai untuk mendapatkan keahlian pada basis data yang besar maka dari itu sering disebut *Knowledge Discovery Databases* (Hasan, 2017).

b. Text Mining

Text Mining bisa didefinisikan sebagai penemuan hal baru yang belum diketahui, secara otomatis mengekstraksi sebuah informasi dari sumber teks tidak terstruktur. Atau lebih singkatnya *text mining* disebut sebagai suatu proses menganalisa teks untuk mendapatkan hal yang berguna untuk tujuan tertentu. Yang membedakan paling dasar di *text mining* dan *data mining* ialah pada asal data yang dipakai. Pada *data mining*, data yang diekstrak berasal pada pola-pola tertentu dan terstruktur, sedangkan *text mining* sumber data yang dipakai berasal pada teks yang relatif tidak testruktur sebab menggunakan bahasa manusia atau biasa disebut sebagai *natural language*. Tentu ada banyak jenis dari bahasa manusia yang digunakan, sehingga menjadikan teks mining menjadi salah satu ranah tersendiri dari *data mining*. Pada *text mining* terdapat langkah-langkah untuk memproses data teks tersebut (Purbo, 2019).

- **Case Folding**

Data proses *Case Folding* adalah dengan merubah tulisan atau kalimat dalam huruf kecil semua, untuk mempermudah dalam pemrosesan data.

- **Cleansing**

Proses selanjutnya adalah *cleansing* dengan membersihkan data atau kalimat dari karakter yang tidak diperlukan seperti tanda baca dan beberapa karakter lainnya.

- **Tokenizing**

Pada proses *tokenization* dilakukan untuk membagi teks yang berupa kalimat maupun paragraf menjadi token/bagian tertentu.

- **Stopword**

Proses ini membuang daftar kata-kata yang kurang penting untuk di analisis menggunakan *stopword*. *Stoplist* merupakan kata-kata yang tidak deskriptif dan *stopword* digunakan dibawah pendekatan *bag-of-words.*

- **Normalization**

Data komentar yang didapatkan pasti menggunakan kalimat-kalimat yang tidak sesuai dengan tulisan yang tepat. Kata baku untuk normalisasi kata didapatkan dari lampiran penilitian Servasius Dwi Harijiatno (Servasius D H, 2019).

- **Stemming**

Proses *stemming* adalah dengan menemukan kata dasar dari sebuah kalimat. Kalimat yang memiliki imbuhan akan dirubah dalam bentuk kata dasar.

c. Analisis Sentimen

Analisis sentimen dengan menyebutnya *opinion mining* ialah beberapa peranan pada *text mining*. *Text mining* membuat studi tentang pendapat yang muncul dari beberapa orang, sentimen, evaluasi, tingkah laku dan emosi terhadap beberapa entitas misal produk, layanan, organisasi, individu, persoalan, topik, acara dan jenis lainnya (Monarizqa et al., 2014).

d. Klasifikasi

Klasifikasi merupakan sebuah cara untuk melihat model atau fungsi yang membedakan gagasan atau *class* data yang tujuannya memperkirakan *class* yang tidak diketahui dari sebuah objeknya. Ada 2 didalam klasifikasi, yakni proses *train* dan proses *test*. Proses *train* menggunakan *training set* yang telah diketahui labelnya yang berfungsi untuk membangun model. *Testing* digunakan untuk pengujian akurasi model yang sudah dirancang saat proses *training* (Dicky N, Gunadi WN, 2015).

e. Naïve Bayes

Naïve Bayes yaitu model penggolongan statistika prediksi nilai probabilitas keanggotaan suatu kelas. *Naïve Bayes* berdasar dari teorema bayes dengan mempunyai kesanggupan untuk menggolongkan seperti halnya pada *decision tree* atau pun *neural network* (Mochammad H. W, 2019).

f. Klasifikasi Multinomial Naïve Bayes

Klasifikasi *Multinomial Naïve Bayes* ialah pembangunan bentuk dari algoritma *bayes* yang biasa dipakai dalam klasifikasi teks. Dokumen *Multinomial Naïve Bayes* dianggap sebagai "*bag of words*" dimana urutan kejadian munculnya kata dalam dokumen diabaikan, sehingga setiap kata diproses menggunakan distribusi *multinomial* (Lutfiah Maharani Siniwi, et al., 2021).

g. K-Fold Cross Validation

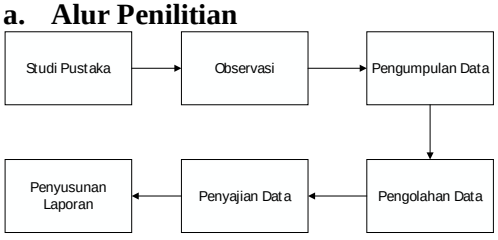
K-Fold Cross Validation sesuatu cara pengambilan sampel yang mengevaluasi model pembelajaran mesin untuk sejumlah sampel data yang terhingga. Proses ini memiliki parameter k untuk mewakili berapa kali kelompok data sampel dibagi. Cara ini umumnya digunakan untuk memprediksi kesanggupan *machine learning* berhadapan dengan *unseen* data (Jason, 2018).

h. Confusion Matrix

Confusion matrix ialah suatu cara untuk membuat menghitung akurasi pada proses data mining (Rosandy, 2016). *Confusion matrix* berisi keterangan mengenai hasil klasifikasi *actual* dan telah diprediksi oleh sistem klasifikasi.

3. Metode Penilitian

a. Alur Penilitian



Gambar 1. Alur Penelitian
Sumber: Hasil Gambar Penelitian Sendiri

Jurnal Smart Teknologi
Vol. 3, No. 6, September 2022, Halaman 612 – 623
ISSN: 2774-1702, http://jurnal.unmuhjember.ac.id/index.php/JST

b. Pengumpulan Data

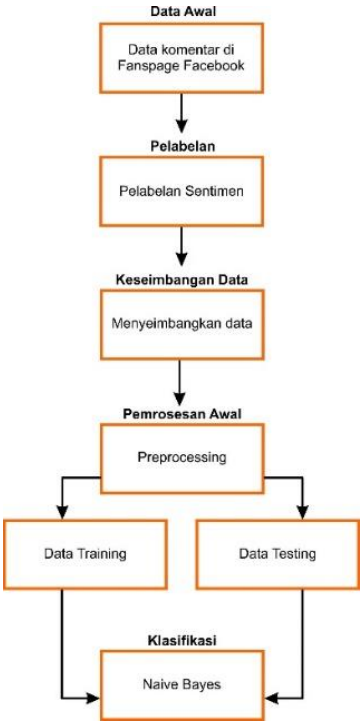
Data yang terkumpul untuk kajian ini melewati cara *crawling* atau mendapatkan data dari sosial media hingga selanjutnya akan dilakukan proses analisis sentimen untuk mendapatkan informasi baru berupa sentimen yang muncul baik sentimen positif, negatif, dan netral. Proses pengumpulan data atau *crawling* data dilakukan dengan menggunakan bantuan *software* facepager. *Software facepager* akan mengambil data dari *fanspage* Facebook, data berupa postingan dari *fanspage* tersebut, beserta komentarnya akan dapat diambil dan dijadikan sebagai sebuah data.

c. Labeling Data

Lanjut dalam proses pelabelan ini peniliti menentukan nilai data komentar *facebook* yang berisi positif yang artinya kata kata mengandung unsur pujian, dukungan, dan tanggapan baik untuk perusahaan. Berisi negatif sebaliknya yang isinya penghinaan atau pun ujaran kebencian, berkata kasar dan membawa ras atau agama. Berisi netral yang isinya komentar bertanya, memberi informasi ataupun komentar yang biasa saja tanpa ada tanggapan positif dan negatif. Dalam proses *labelling* data ini harus dilakukan oleh pakar Bahasa yang sesuai dengan bidangnya. Jika terjadi lebih dari 2 sentimen pada kalimat komentar, Pakar Bahasa melakukan *majority voting* secara manual. Karena proses *labeling* ini dilakukan secara manual, maka butuh waktu yang tidak cepat, perihal ini membuat sebuah kekurangan di saat jumlah data yang diproses *labeling* sangat banyak.

d. Metode Usulan

Dalam penelitian ini peneliti mengusulkan sebuah metode pada analisis sentimen untuk mengetahui sentimen masyarakat dalam layanan perusahaan, dengan menggunakan metode klasifikasi *Naïve Bayes* serta menggunakan pemrosesan data awal dengan beberapa tahapan yang perlu dilalui. Adapun model yang ditawarkan ditunjukan pada Gambar 2.

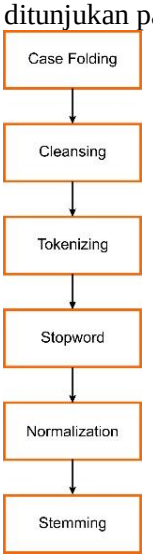


Gambar 2. Gambaran Model
Sumber: Hasil gambar penelitian sendiri

e. Penerapan Konsep

1. Preprocessing

Berikut gambar alur tahapan Preprocessing ditunjukan pada dibawah ini.



Gambar 3. Tahapan *Preprocessing*
Sumber: Purbo, 2019.

Tabel 1. Hasil *Pre-Processing*

No	Komentar	Sentimen
1	terimakasih | informasinya | jelas semoga | tambah | jaya | telkomgroup	Positif

Jurnal Smart Teknologi
Vol. 3, No. 6, September 2022, Halaman 612 – 623
ISSN: 2774-1702, http://jurnal.unmuhjember.ac.id/index.php/JST

No	Komentar	Sentimen
2	teknisinya | tidak | profesional | biaya | teknisi | tidak | jadi | datang	Negatif
3	tanya | lupa | kata | sandi | reset ulang	Netral
4	Tidak profesional jelas	?

Sumber: Contoh data sendiri.

2. TF-IDF

Dari proses *pre-processing* yang sudah dilakukan, maka tahapan berikutnya adalah melakukan perhitungan memakai metode TF-IDF. Metode TF-IDF akan membantu dalam menghitung kata yang muncul.

Tabel 2. TF-IDF

Term	D1	D2	D3	DF	TF*IDF			
					IDF	D1	D2	D3
terimakasih	1			1	0,477	0,477		
informasi	1			1	0,477	0,477		
jelas	1			1	0,477	0,477		
tambah	1			1	0,477	0,477		
semoga	1			1	0,477	0,477		
jaya	1			1	0,477	0,477		
telkom group	1			1	0,477	0,477		
teknisi		2		1	0,477		0,238	
tidak		2		1	0,477		0,238	
profesional		1		1	0,477		0,477	
biaya		1		1	0,477		0,477	
jadi		1		1	0,477		0,477	
datang		1		1	0,477		0,477	
tanya			1	1	0,477121 25			0,477
lupa			1	1	0,477			0,477
kata			1	1	0,477			0,477
sandi			1	1	0,477			0,477
reset			1	1	0,477			0,477
ulang			1	1	0,477			0,477
Jumlah						**3,339**	**2,385**	**2,862**
Ranking						**3**	**1**	**2**

Sumber: Contoh perhitungan sendiri.

3. Klasifikasi Naïve Bayes

Untuk melakukan perhitungan klasifikasi sentimen, diambil dari data training dengan perhitungan berikut ini :

Positif $= \frac{1}{3} = 0{,}33$

Negatif $= \frac{1}{3} = 0{,}33$

Netral $= \frac{1}{3} = 0{,}33$

Menghitung *bags of word* dengan cara berikut ini

$$P(t_k|c) = \frac{N_k + 1}{|V| + N'} \quad (1)$$

dimana:

t_k = Kata dalam semua dokumen yang diberi label (positif/negatif/netral).

c = Kelas (positif/negatif/netral) pada data latih.

$|V|$ = Jumlah semua term atau kata unik (jika berulang, tetap dihitung 1).

Nk = Jumlah kemunculan t_k pada dokumen latih suatu kategori c

N' = Jumlah total term yang terdapat pada c dokumen latih.

+1 = Penambahan angka 1 berfungsi sebagai *Laplace Smoothing*

Misalkan terdapat data *testing* berikut ini :

Tidak profesional jelas

Klasifikan data *testing* ke dalam sentimen positif,negatif,atau netral dengan data *training*.

Sentimen Positif

P(tidak|positif) = $\frac{0+1}{7+19}$ = 0,038462

P(professional|positif) = $\frac{0+1}{7+19}$ = 0,038462

P(jelas|positif) = $\frac{1+1}{7+19}$ = 0,076923

Sentimen Negatif

P(tidak|negatif) = $\frac{2+1}{8+19}$ = 0,111111

P(professional|negatif) = $\frac{1+1}{8+19}$ = 0,074074

P(jelas|negatif) = $\frac{0+1}{8+19}$ = 0,037037

Sentimen Netral

P(tidak|netral) = $\frac{0+1}{6+19}$ = 0,04

P(professional|netral) = $\frac{0+1}{6+19}$ = 0,04

P(jelas|netral) = $\frac{0+1}{6+19}$ = 0,04

Kemudian dicari nilai probabilitas paling tinggi dengan menggunakan *Naïve Bayes* sebagai berikut :

Jurnal Smart Teknologi
Vol. 3, No. 6, September 2022, Halaman 612 – 623
ISSN: 2774-1702, http://jurnal.unmuhjember.ac.id/index.php/JST

Sentimen Positif
p(positif) x p(tidak|positif) x p(profesional|positif) x p(jelas|positif)
1 x 0,038462 x 0,038462 x 0,076923
= 0,000114
0,000114 x 3^{-1} = 0.000038

Sentimen Negatif
p(negatif) x p(tidak|negatif) x p(profesional| negatif) x p(jelas| negatif)
1 x 0,111111 x 0,074074 x 0,037037
= 0,000305
0,000305 x 3^{-1} = 0,000102

Sentimen Netral
p(netral) x p(tidak|netral) x p(profesional| netral) x p(jelas| netral)
1 x 0,04 x 0,04 x 0,04 = 0,000064
0,000064 x 3^{-1} = 0,0000213

Maka sentimen yang terpilih adalah dengan nilai terbesar yaitu sentimen **negatif**.

4. Hasil dan Pembahasan

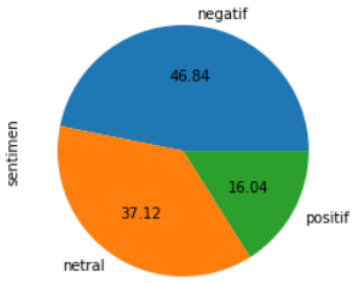
a. Balancing Data

Total Data yang sudah berhasil di *Crawling* berjumlah 854 data. Lanjut data ini akan digunakan proses penilitian. Hasil jumlah *dataset* kelas sentimen ditunjukan pada gambar 4.

```
negatif    400
netral     317
positif    137
Name: sentimen, dtype: int64
```

Gambar 4. Jumlah Kelas Sentimen
Sumber: Hasil perhitungan *python*.

Pada kekurangan dari kesimpulan penilitian Ali imron menyatakan bahwa kinerja sistem dengan *Naïve Bayes* sangat dipengaruhi oleh keseimbang data (Ali Imron, 2019), untuk itu perlu proses untuk menyeimbangkan data. Berikut persentase *dataset* kelas sentimen di gambar 5.



Gambar 5. Persentase Kelas Sentimen
Sumber: Hasil perhitungan *python*.

Pada tahapan *balancing* data digunakan untuk menyeimbangkan *dataset* yang *imbalance* artinya terdapat perbedaan jumlah data yang sangat tinggi antara positif, negatif dan netral. Data *imbalance* hasil yang ditunjukan di gambar 4 dan gambar 5.

Kemudian proses *Over-sampling* adalah metode untuk meningkatkan jumlah sampel mayoritas untuk menyeimbangkan distribusi kelas. Metode ini untuk membantu model pengklasifikasian data yang lebih akurat. Pada tahapan model *over-sampling* disini dibantu dengan pemograman *Python*. Adapun langkah proses dalam melakukan *over-sampling* yang ditunjukan pada gambar 6.

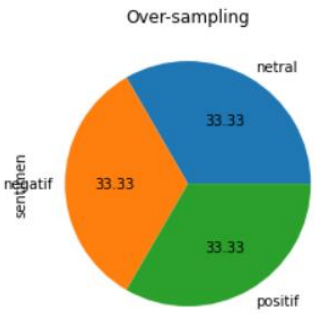
```
from imblearn.over_sampling import RandomOverSampler

#ros = RandomOverSampler(sampling_strategy=1) # Float
ros = RandomOverSampler(sampling_strategy="not majority") # String
X_res, y_res = ros.fit_resample(X, y)

ax = y_res.value_counts().plot.pie(autopct='%.2f')
_ = ax.set_title("Over-sampling")
```

Gambar 6. Kode tahapan *Over-sampling*
Sumber: Kode program *python* sendiri.

Kemudian keluar hasil persentase dan jumlah sentimen *dataset* dari tahapan *Over-sampling* yang terlihat pada gambar 7 dengan gambar 8.

```
netral     400
negatif    400
positif    400
Name: sentimen, dtype: int64
```

Gambar 7. Hasil *Over-sampling*
Sumber: Hasil perhitungan *python*.



Gambar 8. Persentase Setelah *Over-Sampling*
Sumber: Hasil perhitungan *python*.

b. Preprocessing

Seperti yang telah dijelaskan sebelumnya, tujuan dari *pre-processing* untuk menghilangkan data komentar kata,

Jurnal Smart Teknologi
Vol. 3, No. 6, September 2022, Halaman 612 – 623
ISSN: 2774-1702, http://jurnal.unmuhjember.ac.id/index.php/JST

lambang, dan perkara lain yang kurang deskriptif. Tahapan *pre-processing* yang dilakukan pada penelitian dijelaskan singkat secara berurutan seperti dibawah ini:

1. Case Folding

Case Folding dipakai sebagai mengubah huruf jadi kecil semua. Kode *Case Folding* dengan *Python* yang di implementasi tersebut dengan melihat di gambar 9.

```
def lowercase(text):
    text = text.lower()
    return text
data['casefolding'] = data['komentar'].apply(lambda x: lowercase(x))
data
```

Gambar 9. Kode Tahapan *Case Folding*
Sumber: Kode program *python* sendiri.

2. Cleansing

Cleansing membersihkan data atau kalimat dari karakter yang tidak di perlukan seperti tanda baca dan karakter lainnya. Kode program tahapan ini yang digunakan terlihat pada gambar 10.

```
def cleansing(text):
    text = text.translate(str.maketrans("","",string.punctuation))
    text = text.strip()
    return text
data['cleansing'] = data['casefolding'].apply(lambda x: cleansing(x))
data
```

Gambar 10. Kode Tahapan *Cleansing*
Sumber: Kode program *python* sendiri.

3. Tokenizing

Proses ini membagi teks yang berupa kalimat maupun paragraph menjadi token-token atau bagian tertentu. Proses kode programnya seperti ditunjukan pada gambar 11.

```
import nltk
from nltk.tokenize import word_tokenize

def token(text):
    text = nltk.tokenize.word_tokenize(text)
    return text
data['tokenizing'] = data['cleansing'].apply(lambda x: token(x))
data
```

Gambar 11. Kode Tahapan *Tokenizing*
Sumber: Kode program *python* sendiri.

4. Stopword

Stopword membuang kata kata kurang penting. Stoplist terdapat pada file stopwords.txt yang diambil dari www.ranks.nl/stopwords/indonesian. Adapun kode program stopword ditujunkan pada gambar 12.

```
from nltk.corpus import stopwords

list_stopwords = stopwords.words('indonesian')
list_stopwords.extend(["yg", "dg", "rt", "dgn", "ny", "d", 'klo',
                       'kalo', 'amp', 'biar', 'bikin', 'bilang',
                       'gak', 'ga', 'krn', 'nya', 'nih', 'sih',
                       'si', 'tau', 'tdk', 'tuh', 'utk', 'ya',
                       'jd', 'jgn', 'sdh', 'aja', 'n', 't',
                       'nyg', 'hehe', 'pen', 'u', 'nan', 'loh', 'rt',
                       '&amp', 'yah'])

txt_stopword = pd.read_csv("stopwords.txt", names= ["stopwords"], header = None)
list_stopwords.extend(txt_stopword["stopwords"][0].split(' '))
list_stopwords = set(list_stopwords)

def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]

data['stopword'] = data['tokenizing'].apply(stopwords_removal)
data
```

Gambar 12. Kode Tahapan *Stopword*
Sumber: Kode program *python* sendiri.

Stopwords.extend digunakan untuk tambahan *stoplist* apabila ada kata yang kurang di file txt.

5. Normalize

Data komentar yang didapat bukan membentuk kalimat-kalimat yang tepat pada tulisan yang baik. Kerap kali terlihat kata-kata yang tertulis oleh ringkasan dengan bahasa kekinian. Seperti halnya ‘yang’ diringkas jadi ‘yg’, kemudian kata ‘aku’ diringkas jadi ‘ane’ serta lebih banyak lagi. Kata-kata baku untuk normalisasi kata didapatkan dari lampiran penilitian Servasius Dwi Harijiatno (Servasius D H, 2019). Berikut kode program pada tahapan ini terlihat pada gambar 13.

```
normalizad_word = pd.read_excel("normalisasi.xlsx")
normalizad_word_dict = {}

for index, row in normalizad_word.iterrows():
    if row[0] not in normalizad_word_dict:
        normalizad_word_dict[row[0]] = row[1]

def normalized_term(document):
    return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in document]

data['normalisasi'] = data['stopword'].apply(normalized_term)
data
```

Gambar 13. Kode Tahapan *Normalize*
Sumber: Kode program *python* sendiri.

6. Stemming

Tahapan ini dilakukan sebagai pengganti kata-kata komentar ke dalam bentuk kata dasarnya. Implementasi tersebut ditunjukan pada gambar 14.

Jurnal Smart Teknologi
Vol. 3, No. 6, September 2022, Halaman 612 – 623
ISSN: 2774-1702, http://jurnal.unmuhjember.ac.id/index.php/JST

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

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


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

term_dict = {}

for document in data['normalisasi']:
    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]

data['stemming'] = data['normalisasi'].swifter.apply(get_stemmed_term)
data
```

Gambar 14. Kode Tahapan *Stemming*
Sumber: Kode program *python* sendiri.

c. Ekstraksi Fitur

Pada penilitian ini untuk pencarian TF-IDF serta pembobotan kata memakai pertolongan *Python*. Adapun hasil TF-IDF dengan output index = 0 pada kode program *python* yang ditunjukan gambar 15.

```
# Check TF-IDF result
index = 0

print('%20s' % "term", "\t", '%10s' % "TF", "\t", '%20s' % "TF-IDF\n")
for key in data["TF-IDF_dict"][index]:
    print('%20s' % key, "\t\t", data["TF_dict"][index][key] ,"\t" , data["TF-IDF_dict"][index][key])
```

Gambar 15. Kode Program hasil TF-IDF
Sumber: Kode program *python* sendiri.

```
        term             TF                  TF-IDF

    indihome              1       1.4497716469449058
        beda              1       3.6020599913279625
        6jam              1       4.079181246047625
     bengong              1       4.079181246047625
 syaratkeren              1       4.079181246047625
       pokok              1       3.3010299956639813
        lupa              1       3.1249387366083
         tgl              1       3.1249387366083
           5              1       3.380211241711606
      update              1       3.7781512503836434
       tagih              1       2.560667306169737
        yaaa              1       4.079181246047625
```

Gambar 16. Hasil Output kode program TF-IDF
Sumber: Hasil perhitungan *python*.

Melihat hasil pembobotan kata tertinggi dari semua *dataset* akan menghasilkan keluaran seperti gambar 4.14. Jadi dari dataset komentar masyarakat *facebook* paling banyak berkomentar adalah kata "indihome" dengan nilai rank 650.734155.

	term	rank
0	indihome	650.734155
2	min	266.970799
3	pasang	258.851379
1	internet	255.081141
4	jaring	229.124839
...	...	...
144	30mbps	30.791812
143	perangkat	30.791812
139	tangan	30.791812
136	bareng	30.791812
149	orang	30.791812

Gambar 17. Hasil bobot kata
Sumber: Hasil perhitungan *python*.

d. Implementasi Naïve Bayes

Cara ekstraksi fitur dan klasifikasi *Naive Bayes* digabungkan pada satu *class pipeline* bersama rangkaian *vectorizer => transformer => classifier*.

```
pipeline_mnb = Pipeline([
 ('vect', CountVectorizer()),
 ('tfidf', TfidfTransformer(use_idf=True,
smooth_idf=True)),
 ('clf', MultinomialNB(alpha=1))
])
```

Gambar 18. Kode program urutan klasifikasi
Sumber: Kode program *python* sendiri.

Sebelum lanjut proses klasifikasi, penulis melakukan terlebih dahulu pembagian data *training* dan *testing* yang dibantu dengan kode program *python*. Adapun implementasi kode program python yang ditunjukan pada gambar dibawah ini.

```
kf = KFold(n_splits=10) #random_state=0, shuffle=True
i=1
for train_set, test_set in kf.split(X, y):
    print("iteration ", i)
    print("training: ", train_set, "having : ", len(train_set))
    print("testing: ", test_set, "having : ", len(test_set))
    print("----------------------------")
    i += 1
```

Gambar 19. Kode program K-Fold
Sumber: Kode program *python* sendiri.

```
iteration  8
training:  [   0    1    2 ... 1197 1198 1199] having :  1080
testing:  [840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857
 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875
 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893
 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911
 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929
 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947
 948 949 950 951 952 953 954 955 956 957 958 959] having :  120
```

Gambar 20. Contoh hasil pembagian K-Fold
Sumber: Hasil perhitungan *python*.

Setelah pembagian data training dan testing kemudian melakukan proses prediksi *Multinomial Naïve Bayes* dengan bantuan kode program python. Adapun implementasi proses klasifikasi *Multinomial Naïve Bayes* menggunakan python ditunjukan pada gambar 21.

Jurnal Smart Teknologi
Vol. 3, No. 6, September 2022, Halaman 612 – 623
ISSN: 2774-1702, http://jurnal.unmuhjember.ac.id/index.php/JST

```
scores = []
i = 1
for train_set, test_set in kf.split(X, y):
    pipeline_mnb.fit(X.iloc[train_set], y[train_set])
    sco = pipeline_mnb.score(X[test_set], y[test_set])
    scores.append(sco)
    print("------------------")
    print("iterasi ", i)
    i += 1
    predictions_mnb = pipeline_mnb.predict(X[test_set])
    tes = predictions_mnb == y[test_set]
    print(tes)
    print(predictions_mnb)
    print(confusion_matrix(y[test_set], predictions_mnb))
```

Gambar 21. Kode program prediksi *Multinomial Naïve Bayes*
Sumber: Kode program *python* sendiri.

Tabel 3. Prediksi *Multinomial Naïve Bayes* K=10 iterasi 8

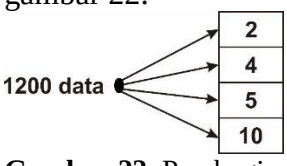
Index	Prediksi	Aktual	Matrix
840	NETRAL	NETRAL	TNR
841	NEGATIF	NEGATIF	TN
842	POSITIF	POSITIF	TP
843	POSITIF	POSITIF	TP
844	NETRAL	POSITIF	FP
…	…	…	…
955	NETRAL	NETRAL	TNR
956	NEGATIF	NEGATIF	TN
957	NEGATIF	NEGATIF	TN
958	NEGATIF	NEGATIF	TN
959	POSITIF	POSITIF	TP

Sumber: Hasil perhitungan *python*.

Pada proses klasifikasi penulis menggunakan beberapa skenario untuk membagi data *testing* dan data *training*. Proses dan hasil pembagian beberapa data *testing* dan *training* dijelaskan pada subab Uji Model.

e. Uji Model

Apabila ingin melihat hasil performa dari Algoritma *Naive Bayes*, selanjutnya membuat pengujian model. Pada proses pengujian model dilakukan sebanyak 21 percobaan dengan 4 skenario. Gambar pembagian skenario ditunjukan pada gambar 22.



Gambar 22. Pembagian 4 skenario
Sumber: Hasil penelitian sendiri.

Hasil pembagian *data training* dan *testing* yang kemudian di klasifikasi bakal ditampilkan pada wujud *confusion matrix*. Tabel ini tersusun dari *predicted class* dan *actual class*. Model ini bisa dilihat di Tabel 4.

Tabel 4. Model *Confusion Matrix*

		Predicted Class		
		Class A	**Class B**	**Class C**
Actual Class	**Class A**	AA	AB	AC
	Class B	BA	BB	BC
	Class C	CA	CB	CC

Sumber: Ali Imron, 2019.

Sementara dengan nilai akurasi model didapat dari jumlah data yang cocok hasil klasifikasi dibagi dengan total data (Ali Imron, 2019), yang bisa di pada gambar 23.

$$Akurasi = \frac{AA+BB+CC}{AA+AB+AC+BA+BB+BC+CA+CB+CC}$$

Gambar 23. Perhitungan Akurasi Model
Sumber: Ali Imron, 2019.

Tahap pertama melakukan uji coba skenario 1 dengan membagi data *training* dan *testing* menjadi 2 bagian. Hasil pembagian ditujukan di tabel 5.

Tabel 5. Hasil 2-Fold *Cross Validation*

K = 2	**DATA**	
Iterasi 1	600	600
Iterasi 2	600	600

Sumber: Hasil penelitian sendiri.

Tahap kedua melakukan uji coba skenario 2 dengan membagi data training dan testing menjadi 4 bagian. Hasil pembagian ditujukan pada tabel 6.

Tabel 6. Hasil 4-Fold *Cross Validation*

K=4	**DATA**			
Iterasi 1	300	300	300	300
Iterasi 2	300	300	300	300
Iterasi 3	300	300	300	300
Iterasi 4	300	300	300	300

Sumber: Hasil penelitian sendiri.

Tahap ketiga melakukan uji coba skenario 3 dengan membagi data *training* dan testing menjadi 5 bagian. Hasil pembagian ditujukan pada tabel 7.

Tabel 7. Hasil 5-Fold *Cross Validation*

K=5	**DATA**				
Iterasi 1	240	240	240	240	240
Iterasi 2	240	240	240	240	240
Iterasi 3	240	240	240	240	240
Iterasi 4	240	240	240	240	240
Iterasi 5	240	240	240	240	240

Sumber: Hasil penelitian sendiri.

Tahap keempat melakukan uji coba skenario 4 dengan membagi data *training* dan testing menjadi 10 bagian. Hasil pembagian ditujukan dengan tabel 8.

Jurnal Smart Teknologi
Vol. 3, No. 6, September 2022, Halaman 612 – 623
ISSN: 2774-1702, http://jurnal.unmuhjember.ac.id/index.php/JST

Tabel 8. Hasil 10-Fold *Cross Validation*

K=10	DATA									
Iterasi 1	120	120	120	120	120	120	120	120	120	120
Iterasi 2	120	120	120	120	120	120	120	120	120	120
Iterasi 3	120	120	120	120	120	120	120	120	120	120
Iterasi 4	120	120	120	120	120	120	120	120	120	120
Iterasi 5	120	120	120	120	120	120	120	120	120	120
Iterasi 6	120	120	120	120	120	120	120	120	120	120
Iterasi 7	120	120	120	120	120	120	120	120	120	120
Iterasi 8	120	120	120	120	120	120	120	120	120	120
Iterasi 9	120	120	120	120	120	120	120	120	120	120
Iterasi 10	120	120	120	120	120	120	120	120	120	120

Sumber: Hasil penelitian sendiri.

f. Evaluasi Model

Evaluasi model merupakan hasil akurasi dari rangkuman semua uji coba yang telah dilakukan. Berikut hasil akurasi dari semua uji coba di tunjukan pada tabel 9.

Tabel 9. Hasil akurasi dari semua uji coba

Jumlah K	Iterasi Fold	Akurasi
K = 2	Iterasi 1	0.75833333
	Iterasi 2	0.785
K = 4	Iterasi 1	0.81333333
	Iterasi 2	0.79333333
	Iterasi 3	0.79
	Iterasi 4	0.84333333
K = 5	Iterasi 1	0.825
	Iterasi 2	0.80833333
	Iterasi 3	0.7875
	Iterasi 4	0.825
	Iterasi 5	0.83333333
K = 10	Iterasi 1	0.81666667
	Iterasi 2	0.85
	Iterasi 3	0.80833333
	Iterasi 4	0.84166667
	Iterasi 5	0.78333333
	Iterasi 6	0.78333333
	Iterasi 7	0.80833333
	Iterasi 8	0.85833333
	Iterasi 9	0.84166667
	Iterasi 10	0.81666667

Sumber: Hasil penelitian sendiri.

Dari nilai akurasi pada 21 percobaan diatas menunjukan angka yang cukup baik. Dengan beberapa uji percobaan 4 model diketahui bahwa hasil akurasi tertinggi terdapat pada K 10 Iterasi 8 yaitu sebesar 0.85833333 atau 85.8%.

Selanjutnya mengetahui hasil rata rata pada setiap perubahan jumlah Kfold *cross validation*. Hasil rata-rata pada pengujian tersebut ditunjukan tabel 10.

Tabel 10. Hasil rata-rata uji K-Fold *Cross Validation*

Jumlah K-FOLD	Hasil Rata-rata
2	0.7716666666666667
4	0.8099999999999999
5	0.8158333333333333
10	0.8208333333333334

Sumber: Hasil penelitian sendiri.

Hasil rata-rata pada pengujian K-Fold *Cross Validation* mengalami kenaikan di setiap pertambahan jumlah *data training*. Jadi ada pengaruh terhadap perubahan peningkatan jumlah data *training* pada nilai akurasi.

5. KESIMPULAN

a. Kesimpulan

Dari hasil pengujian ini algoritma *Multinomial Naive Bayes* yang sudah dikerjakan mememiliki beberapa hal yang didapatkan yaitu mendapatkan hasil dari rata-rata tingkat akurasi yang cenderung meningkat seiring bertambahnya data training. Kemudian Algoritma *Multinomial Naive Bayes* terjamin akurat sebab menimbulkan nilai akurasi tertinggi pada K=10 Iterasi 8 yaitu 0.85833333 atau 85.8%.

b. Saran

1. Menggunakan algoritma klasifikasi yang lain agar bisa membandingkan hasil pengujian untuk mencari algoritma klasifikasi terbaik.
2. Menambah *data training* supaya lebih tinggi akurasi yang dihasilkan.
3. Membuat *interface* dari proses pengujian model dan visualisasi dari performa metode yang digunakan.

6. REFERENSI

Abbas, M., Ali Memon, K., & Aleem Jamali, A. (2019). Multinomial Naive Bayes Classification Model for Sentiment Analysis. *IJCSNS International Journal of Computer Science and Network Security*, *19*(3), 62.

Ali Imron, (2019). Analisis Sentimen Terhadap Tempat Wisata Di Kabupaten Rembang Menggunakan Metode Naïve Bayes Classifier, 45, 40.

Anisa Eka Puridewi, J. N. (2018). Perbandingan metode naive bayes, support vector machine dan id3 dalam penetapan status penanganan kecelakaan kerja. *Seminar Nasional Matematika Dan Pendidikan Matematika*, 130–137.

Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze, Introduction to Information Retrievel, Cambridge University Press. 2008

Cindy, MA. (2020). "*Trafik Internet Naik 20% pada Masa Corona, Operator Kucurkan Rp 1,9 T*", https://katadata.co.id/desysetyowati/digital/5e9d61d7b8736/trafik-internet-naik-20-pada-masa-corona-operator-kucurkan-rp-19-t, diakses pada 10 Maret 2021.

Devita, R. N., Herwanto, H. W., & Wibawa, A. P. (2018). Perbandingan Kinerja Metode Naive Bayes dan K-Nearest Neighbor untuk Klasifikasi Artikel Berbahasa indonesia. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *5*(4), 427. https://doi.org/10.25126/jtiik.201854773

Dicky Nofriansyah, S.Kom., & Gunadi, W. N. (2015). *ALGORITMA DATA MINING DAN PENGUJIAN*. DEEPUBLISH.

Haqqizar, N., & Larasyanti, T. N. (2019). Analisis Sentimen Terhadap Layanan Provider Telekomunikasi Telkomsel Di Twitter Dengan Metode Naïve Bayes. *Prosiding TAU SNAR-TEK 2019 Seminar Nasional Rekayasa Dan Teknologi*, *10*(2), 1–15.

Hasan, M. (2017). *Prediksi Tingkat Kelancaran Pembayaran Kredit Bank Menggunakan Algoritma Naïve Bayes Berbasis Forward Selection*. *9*, 317–324.

Jason, B. (2018). "A Gentle Introduction to k-fold Cross-Validation", https://machinelearningmastery.com/k-fold-cross-validation/, diakses pada 01 Desember 2021.

Lutfiah Maharani Siniwi, Alan Prahutama, Arief Rachman Hakim. (2021). *Query Expansion Ranking* Pada Analisis Sentimen Menggunakan Klasifikasi *Multinomial Naïve Bayes*. *Jurnal Gaussian*, 10(3), 377-387.

Masripah, S. (2015). Evaluasi Penentuan Kelayakan Pemberian Kredit Koperasi Syariah Menggunakan Algoritma Klasifikasi C4.5. *Jurnal Pilar Nusa Mandiri*, *XI*(1), 1–10.

Mochammad HW. (2019). "Algoritma Naïve Bayes", https://binus.ac.id/bandung/2019/12/algoritma-naive-bayes/, diakses pada 20 Juli 2021.

Monarizqa, N., Nugroho, L. E., & Hantono, B. S. (2014). Penerapan Analisis Sentimen Pada Twitter Berbahasa Indonesia Sebagai Pemberi Rating. *Jurnal Penelitian Teknik Elektro Dan Teknologi Informasi*, *1*, 151–155.

Musthofa, GH., & Azriel, Christian Nurcahyo, P. H. S. (2020). PENGARUH SENTIMEN DI SOSIAL MEDIA DENGAN HARGA SAHAM PERUSAHAAN. *Jurnal Ilmiah Edutic*, *6*(2).

Purbo, O. W. (2019). *Text Mining*. Andi.

Ronny Julianto, Evi Dianti Bintari, I. (2017). Analisis Sentimen Layanan Provider Telepon Seluler pada Twitter menggunakan Metode Naïve Bayesian Classification. *Journal of Big Data Analytic and Artificial Intelligence*, *3*(1).

Rosandy, T. (2016). Perbandingan Metode

Naïve Bayes Classifier Dengan Metode Decision Tree (C.45) Untuk Menganalisa Kelancaran Pembiayaan. *Jurnal TIM Darmajaya*, 2(1), 52.

Saiyed, S., Bhatt, N., & Ganatra, A. P. (2016). A Survey on Naive Bayes Based Prediction of Heart Disease Using Risk Factors. *International Journal of Innovative and Emerging Research in Engineering*, *3*(2), 111–115.

Septiani, W. D. (2017). Komparasi Metode Klasifikasi Data Mining Algoritma C4.5 Dan Naive Bayes Untuk Prediksi Penyakit Hepatitis. *Jurnal Pilar Nusa Mandiri*, *13*(1), 76–84. https://doi.org/10.33480/pilar.v13i1.149

Servasius, D. H. (2019). Analisis Sentimen Pada Twitter Menggunakan Multinomial Naïve Bayes, 93.

Simon, K. (2020). "Indonesia *Digital report 2020*. Global Digital Insights", https://datareportal.com/reports/digital-2020-indonesia, diakses pada 10 Maret 2021.