

BAB 3

METODE PENELITIAN

Penelitian ini merupakan penelitian mengenai analisis sentimen berdasarkan data yang diperoleh dari media sosial Twitter. Penelitian ini dilakukan berdasarkan dari latar belakang permasalahan yang ada, sehingga dilakukan analisis data mengenai Biznet untuk mengatasi permasalahan yang muncul. Metode Naive Bayes Classifier digunakan dalam penelitian ini untuk melakukan analisis sentimen dengan label positif, negatif dan netral berdasarkan data yang diperoleh dari media sosial Twitter tentang Biznet. Berikut ini merupakan bahan, alat, serta tahapan jalannya penelitian untuk menyelesaikan proses analisis sentimen Biznet.

3.1 BAHAN DAN ALAT PENELITIAN

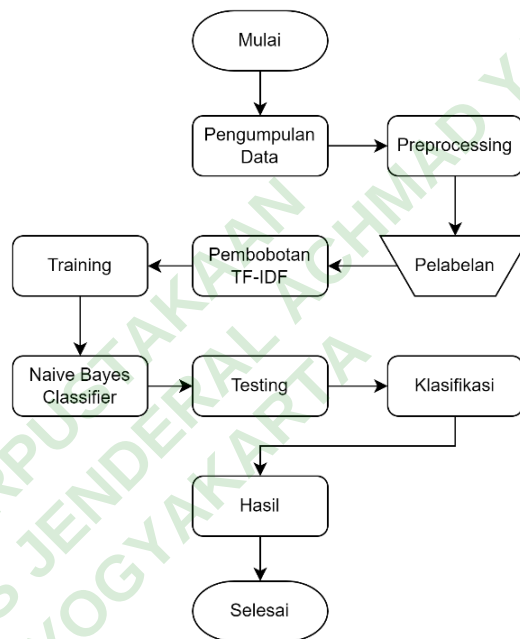
Bahan-bahan yang diperlukan dalam melaksanakan penelitian ini yaitu data *tweet* dan komentar yang berkaitan dengan Biznet pada media sosial Twitter. Pengambilan data pada Twitter ini menggunakan *tools* tweet-harvest dari hasil pencarian di Twitter berdasarkan kata kunci “Biznet” dan “Internet Biznet”. Data *tweet* yang diambil dengan rentang waktu pada tanggal 1 Januari 2024 – 26 Juni 2024.

Alat yang digunakan dalam melaksanakan penelitian adalah komputer yang memiliki spesifikasi yang memadai dalam menjalankan sistem operasi serta *software* pengembangan dan memiliki koneksi internet yang stabil. Sistem operasi serta aplikasi yang digunakan dalam melaksanakan penelitian analisis sentimen ini adalah sebagai berikut:

1. Sistem Operasi : Windows 11
2. *Tools* Analisis : *Google Colaboratory*
3. *Text Editor* : *Sublime Text*
4. Bahasa Pemrograman : Python
5. *Framework* : Flask

3.2 JALAN PENELITIAN

Penelitian ini menerapkan metode Naive Bayes Classifier dalam melaksanakan penelitian analisis sentimen berdasarkan data yang diperoleh dari media sosial Twitter tentang Biznet. Pada penelitian ini tahapan demi tahapan dilakukan dengan berurutan. Berikut merupakan alur penelitian serta tahapan dari analisis sentimen dengan mengimplementasikan metode Naive Bayes Classifier ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

3.2.1 Pengumpulan Data

Tahap *crawling* atau pengumpulan data yang dilakukan pada media sosial Twitter agar memperoleh data *tweet* dan komentar mengenai Biznet dengan kata kunci “Biznet” dan “Internet Biznet”. Data yang dikumpulkan adalah data *tweet* maupun komentar yang diambil mulai bulan Januari hingga bulan Juni 2024. Pada tahap ini dilakukan pengambilan data menggunakan *Google Colaboratory* dengan *tools* *tweet-harvest*. Data yang diperoleh berupa file dengan format *Comma Separated Values* (CSV) dan dapat dibuka melalui *Google Spreadsheets*. Berikut merupakan tampilan proses pengumpulan data pada *Google Colaboratory* dengan menggunakan *tools* *tweet-harvest* ditunjukkan pada Gambar 3.2.

```
# Crawl Data

filename = 'biznetjuni.csv'
search_keyword = 'biznet or internet biznet since:2024-06-01 until:2024-06-27'
limit = 2000

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit} --token {twitter_auth_token}

Opening twitter search page...

Filling in keywords: biznet or internet biznet since:2024-06-01 until:2024-06-27

-- Scrolling... (1) (2)Created new directory: /content/tweets-data

Your tweets saved to: /content/tweets-data/biznetjuni.csv
Total tweets saved: 17
```

Gambar 3.2 Pengumpulan Data

Pengumpulan data Twitter dilakukan dengan menggunakan *tools tweet-harvest* menghasilkan data yang berformat file *Comma Separated Values* (CSV) seperti contoh data *tweet* pada Tabel 3.1.

Tabel 3.1 Hasil Data Twitter

No	Full_text
1	Knp ya provider yg speed nya oke kayak myrepublic sama biznet ini ga bisa dipasang di daerahku coyy butuhh bgtt ini
2	@ColeCalamello Biznet paling mending tapi roket ungu kalau daerahnya waras juga kenceng cuman kalau alasan belum ke jangkau pake starlink ae
3	@lawveys kyaknya sutet wifi biznet lu dien
4	Emg deh wifi biznet di rumah juara bgt kecepatannya ketimbang yg di kostan :)
5	@prematthur biznet padahal wes paling oke menurutku coba iconnet thur
6	@mghtypip aman tah pip? gawe biznet ping e gak stabil anyinggg
7	provider wifi yang oke selain biznet apa kawan?
8	Lah? Biznet knp dah
9	@mbakcim @IndiHomeCare Ganti Biznet palagan tercover koyoke.
10	@beaurorra RORRRRAAAA hi kata aku ganti biznet pls firstmed emang JELEK wifi bapak

Tabel 3.1 menunjukkan contoh perolehan hasil data *tweet* maupun komentar dari proses *crawling* data melalui *platform* media sosial Twitter. Data yang

diperoleh masih dengan berbagai macam bentuk dan variasi seperti username, angka, URL dan simbol maupun karakter sehingga diperlukan *preprocessing* data agar data menjadi lebih terstruktur dan dapat dilakukan untuk proses selanjutnya dalam analisis sentimen.

3.2.2 *Preprocessing*

Preprocessing adalah tahap mengolah data yang diperoleh dari hasil *crawling* pada *platform* media sosial Twitter. Pada tahapan ini dilakukan pengolahan data tidak terstruktur agar dapat diolah selanjutnya untuk proses penelitian dengan menerapkan tahapan seperti *Cleaning*, *Case Folding*, *Number Removal*, *Punctuation Removal*, *Whitespaces Removal*, *Tokenizing*, *Filtering*, *Stemming* dan *Normalization*.

1. *Cleaning*

Tahap *Cleaning* data dilakukan untuk menghapus *mention*, *hashtag* dan *url* yang terdapat pada data *tweet*. Berikut adalah kode program pada tahap *cleaning* data.

```
def clean_twitter(text):
    text = re.sub(r'@[A-Za-z0-9_]+', '', text)
    text = re.sub(r'#\w+', '', text)
    text = re.sub(r'https?:\/\/\S+', '', text)
    return text
```

Hasil dari penerapan kode program pada tahap *cleaning* data ditunjukkan pada Gambar 3.3.

	full_text
0	Knp ya provider yg speed nya oke kayak myrepub...
1	Biznet paling mending tapi roket ungu kalau d...
2	kyaknya sutet wifi biznet lu dien
3	Emg deh wifi biznet di rumah juara bgt kecepat...
4	biznet padahal wes paling oke menurutku coba ...
...	...
5540	hah biznet mau ada FUP ?!!?!
5541	Bulan ini ganti First Media tadinya mau Bizne...
5542	Biznet News Berita baik: Bandwidth naik (7515...
5543	Plzz lah mo ganti biznet tapi keknya gak ada d...
5544	Awali awal tahun dengan wifi biznet kaga ada k...

Gambar 3.3 Hasil *Cleaning*

2. Case Folding

Case Folding merupakan suatu tahap penyeragaman huruf untuk merubah data *tweet* menjadi huruf kecil dengan kode program seperti berikut.

```
df['full_text'] = df['full_text'].str.lower()
```

Hasil penerapan tahap *case folding* ditunjukkan pada Gambar 3.4.

	full_text
0	knp ya provider yg speed nya oke kayak myrepub...
1	biznet paling mending tapi roket ungu kalau d...
2	kyaknya sutet wifi biznet lu dien
3	emg deh wifi biznet di rumah juara bgt kecepat...
4	biznet padahal wes paling oke menurutku coba ...
...	...
5540	hah biznet mau ada fup ?!?!?
5541	bulan ini ganti first media tadinya mau bizne...
5542	biznet news berita baik: bandwidth naik (7515...
5543	plzz lah mo ganti biznet tapi keknya gak ada d...
5544	awali awal tahun dengan wifi biznet kaga ada k...

Gambar 3.4 Hasil Case Folding

3. Number Removal

Number Removal merupakan proses untuk menghapus karakter angka yang terdapat pada data teks dengan kode program seperti berikut.

```
def number_removal(text):
    text = re.sub(r'\d+', '', text)
    return text
```

Hasil dari tahap *number removal* ditunjukkan pada Gambar 3.5.

	full_text
0	knp ya provider yg speed nya oke kayak myrepub...
1	biznet paling mending tapi roket ungu kalau d...
2	kyaknya sutet wifi biznet lu dien
3	emg deh wifi biznet di rumah juara bgt kecepat...
4	biznet padahal wes paling oke menurutku coba ...
...	...
5540	hah biznet mau ada fup ?!?!?
5541	bulan ini ganti first media tadinya mau bizne...
5542	biznet news berita baik: bandwidth naik () ad...
5543	plzz lah mo ganti biznet tapi keknya gak ada d...
5544	awali awal tahun dengan wifi biznet kaga ada k...

Gambar 3.5 Hasil Number Removal

4. Punctuation Removal

Punctuation Removal merupakan proses untuk menghapus simbol atau tanda baca pada teks dengan kode program seperti berikut.

```
def punct_removal(text):
    text = text.translate(str.maketrans('', '', string.punctuation))
    return text
```

Hasil tahap *punctuaction removal* ditunjukkan pada Gambar 3.6.

full_text	
0	knp ya provider yg speed nya oke kayak myrepub...
1	biznet paling mending tapi roket ungu kalau d...
2	kyaknya sutet wifi biznet lu dien
3	emg deh wifi biznet di rumah juara bgt kecepat...
4	biznet padahal wes paling oke menurutku coba ...
...	...
5540	hah biznet mau ada fup
5541	bulan ini ganti first media tadinya mau bizne...
5542	biznet news berita baik bandwidth naik ada p...
5543	plzz lah mo ganti biznet tapi keknya gak ada d...
5544	awali awal tahun dengan wifi biznet kaga ada k...

Gambar 3.6 Hasil Punctuation Removal

5. Whitespaces Removal

Whitespace Removal merupakan proses untuk menghapus spasi yang tidak diperlukan pada data teks dengan kode program seperti berikut.

```
def whitespaces(text):
    return text.strip()
```

Hasil tahap *whitespace removal* ditunjukkan pada Gambar 3.7.

full_text	
0	knp ya provider yg speed nya oke kayak myrepub...
1	biznet paling mending tapi roket ungu kalau da...
2	kyaknya sutet wifi biznet lu dien
3	emg deh wifi biznet di rumah juara bgt kecepat...
4	biznet padahal wes paling oke menurutku coba i...
...	...
5540	hah biznet mau ada fup
5541	bulan ini ganti first media tadinya mau biznet...
5542	biznet news berita baik bandwidth naik ada p...
5543	plzz lah mo ganti biznet tapi keknya gak ada d...
5544	awali awal tahun dengan wifi biznet kaga ada k...

Gambar 3.7 Hasil Whitespace Removal

6. Tokenizing

Tokenizing, merupakan tahap untuk memisahkan suatu kalimat kedalam bentuk kata atau biasa disebut dengan token. Tahap *tokenizing* dapat dilakukan dengan menerapkan kode program seperti berikut.

```
def tokenizing(text):
    return word_tokenize(text)
```

Tahap *tokenizing* dilakukan dengan menggunakan *Natural Language Toolkit* (NLTK) yang merupakan *library* Python yang dirancang agar mempermudah pekerjaan dalam pemrosesan bahasa alami. Hasil dari tahap *tokenizing* ditunjukkan pada Gambar 3.8.

	full_text
0	[knp, ya, provider, yg, speed, nya, oke, kayak...
1	[biznet, paling, mending, tapi, roket, ungu, k...
2	[kyaknya, sutet, wifi, biznet, lu, dien]
3	[emg, deh, wifi, biznet, di, rumah, juara, bgt...
4	[biznet, padahal, wes, paling, oke, menurutku,...
...	...
5540	[hah, biznet, mau, ada, fup]
5541	[bulan, ini, ganti, first, media, tadinya, mau...
5542	[biznet, news, berita, baik, bandwidth, naik, ...
5543	[plzz, lah, mo, ganti, biznet, tapi, keknya, g...
5544	[awali, awal, tahun, dengan, wifi, biznet, kag...

Gambar 3.8 Hasil Tokenizing

7. Stopword Removal (Filtering)

Stopword Removal merupakan tahap untuk menghilangkan kata yang tidak memiliki arti atau makna penting pada teks. Berikut merupakan kode program untuk melakukan tahap *Stopword Removal (Filtering)* dengan menggunakan *library* Sastrawi.

```
stop_factory = StopWordRemoverFactory().get_stop_words()
more_stopword = []

data = stop_factory + more_stopword

dictionary = ArrayDictionary(data)
```

```
str_remover = StopWordRemover(dictionary)

def stopwordremoval(token_list):
    return [word for word in token_list if word not in data]
```

Stopword merupakan kata umum yang banyak muncul pada teks namun biasanya dianggap tidak memiliki nilai makna yang penting atau berarti dalam proses analisis. Hasil dari tahap stopword removal ditunjukkan pada Gambar 3.9.

	full_text
0	[knp, provider, yg, speed, nya, oke, kayak, my...
1	[biznet, paling, mending, roket, ungu, kalau, ...
2	[kyaknya, sutet, wifi, biznet, lu, dien]
3	[emg, deh, wifi, biznet, rumah, juara, bgt, ke...
4	[biznet, padahal, wes, paling, oke, menurutku,...
...	...
5540	[hah, biznet, mau, fup]
5541	[bulan, ganti, first, media, tadinya, mau, biz...
5542	[biznet, news, berita, baik, bandwidth, naik, ...
5543	[plzz, lah, mo, ganti, biznet, keknya, gak, wi...
5544	[awali, awal, tahun, wifi, biznet, kaga, konek...

Gambar 3.9 Hasil Stopword Removal

8. Stemming

Stemming merupakan proses untuk mengubah kata yang berimbuhan menjadi kata dasarnya. Berikut merupakan kode program untuk melakukan tahap *stemming*.

```
def stemming(text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    do = []
    for w in text:
        dt = stemmer.stem(w)
        do.append(dt)
    data = []
    data = " ".join(do)
    print(data)
    return data
```

Tahap *stemming* menggunakan *library* Sastrawi digunakan untuk penyederhanaan kata menjadi bentuk dasarnya dalam konteks bahasa Indonesia. Hasil dari tahap *stemming* ditunjukkan pada Gambar 3.10.

	full_text
0	knp provider yg speed nya oke kayak myrepublic...
1	biznet paling mending roket ungu kalau daerah ...
2	kyaknya sutet wifi biznet lu dien
3	emg deh wifi biznet rumah juara bgt cepat keti...
4	biznet padahal wes paling oke turut coba iconn...
...	...
5540	hah biznet mau fup
5541	bulan ganti first media tadi mau biznet karna ...
5542	biznet news berita baik bandwidth naik pilih ...
5543	plzz lah mo ganti biznet kek gak wilayah ku
5544	awal awal tahun wifi biznet kaga koneksi

Gambar 3.10 Hasil Stemming

9. Normalization

Normalization merupakan proses merubah kata untuk dikembalikan dalam ejaan yang benar dan mengubah menjadi kata yang baku dengan menggunakan kode seperti berikut.

```
norm = {'knp ':'':' kenapa ', ' yg ':'':' yang ', 'kyaknya ':'':' seperti'
        ', 'emg ':'':' memang ', ' wes ':'':' sudah ', ' tbtb ':'':' tiba tiba '}

def normalisasi(str_text):
    for i in norm:
        str_text = str_text.replace(i, norm[i])
    return str_text
```

Hasil dari tahap *normalization* ditunjukkan pada Gambar 3.11.

	full_text
0	kenapa provider yang speed nya oke kayak myre...
1	biznet paling mending roket ungu kalau daerah ...
2	sepertinya sutet wifi biznet kamu dien
3	memang deh wifi biznet rumah juara banget cep...
4	biznet padahal sudah paling oke turut coba ico...
...	...
5540	hah biznet mau fup
5541	bulan ganti first media tadi mau biznet karna ...
5542	biznet news berita baik bandwidth naik pilih ...
5543	plis lah mau ganti biznet kaya tidak wilayah ku
5544	awal awal tahun wifi biznet tidak koneksi

Gambar 3.11 Hasil Normalization

3.2.3 Pelabelan

Pelabelan merupakan tahap untuk melakukan proses pelabelan yang dilakukan secara manual terhadap sebagian data *tweet* maupun komentar yang sebelumnya sudah dilakukan *preprocessing* untuk memberikan label dengan kategori sentimen positif, negatif dan netral supaya dapat dilakukan analisis. Contoh hasil dari proses pelabelan manual ditunjukkan pada Tabel 3.2.

Tabel 3.2 Hasil Pelabelan

No	Full_text	Kelas	Label
1	abai aja mba domain email kyknya bukan milik biznet gio	Netral	0
2	abis lebaran ganti wifi aja kali biznet bagus tidak sih	Netral	0
3	abis telfon biznet ktanya memang ganggu jaring international jdi kaya ig netflix twt lagiganggu	Negatif	-1
4	acara biznet teh	Netral	0
5	acara kamu mba biznet	Netral	0
6	adaa makanya kesini gabisa bales dm nya langsung aja kesini besok serah jam berapa kutitipin org biznet nya bilang aja mau ambil voucher golden lam udah	Netral	0
7	adit biznet	Netral	0
8	admin biznet yang hormat balas dm ku terima kasih	Positif	1
9	admin pln biznet indihome sangat rilet	Netral	0
10	aduh internet biznet mati	Negatif	-1

3.2.4 Training Data

Tahap *training data* dilakukan untuk pembelajaran atau melatih model agar dapat menemukan pola dan nantinya dapat melakukan prediksi terhadap data baru. Pada tahap *training*, data yang digunakan berjumlah 720 data dengan jumlah masing-masing label positif, negatif dan netral yang sama. Berikut merupakan tahapan pada saat melakukan proses *training data*.

1. Membagi data *training* dan data *test*.

Sebelum melakukan perhitungan TF-IDF, dilakukan pembagian data untuk data *training* dan data *test* dengan menggunakan kode seperti berikut.

```
X = df.full_text
y = df.label

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Pada kode tersebut, parameter *test_size=0.2* menentukan persentase dari total dataset yang akan dialokasikan sebagai data *test*, sementara sisanya, yaitu 80% akan digunakan sebagai data *training*. Sedangkan *random_state=42* adalah nilai yang digunakan untuk mengontrol proses pembagian dataset secara acak. Penerapan nilai yang sama dalam *random_state* memungkinkan untuk membagi dataset dengan cara yang konsisten setiap kali kode dijalankan sehingga hasilnya dapat direproduksi dengan mudah (Fadilla et al., 2023).

2. Ekstraksi fitur dengan *TfidfVectorizer*

Untuk melakukan ekstraksi fitur pada teks dapat menggunakan fungsi *TfidfVectorizer* dari *library sklearn* yang dapat dilakukan dengan menerapkan kode seperti berikut.

```
vect = TfidfVectorizer(max_features=1000, binary=True)
X_train_vect = vect.fit_transform(X_train)
```

Kode tersebut digunakan untuk melakukan *vectorization* atau pengubahan teks menjadi representasi numerik menggunakan TF-IDF *Vectorizer* dari *library Scikit-Learn*. Selanjutnya untuk membuat fitur TF-IDF kedalam bentuk array dapat dilakukan dengan menggunakan kode seperti berikut.

```
X_train_vect.toarray()
```

Kode tersebut digunakan untuk mengonversi hasil transformasi TF-IDF yang disimpan dalam format *sparse matrix* menjadi array. *Sparse matrix* adalah format penyimpanan yang efisien untuk data yang sebagian besar elemennya bernilai nol, yang sering terjadi dalam representasi teks menggunakan TF-IDF. Namun, format ini bisa sulit dibaca dan dianalisis langsung. Dengan mengubahnya menjadi array, dapat diketahui nilai-nilai TF-IDF untuk setiap kata dalam bentuk yang lebih intuitif dan mudah dibaca. Hasil dari penerapan kode untuk melakukan transformasi TF-IDF ditunjukkan pada Gambar 3.12.

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

Gambar 3.12 Hasil TF-IDF Dalam Bentuk Array

Untuk memetakan nilai-nilai yang dihasilkan dari representasi TF-IDF ke dalam kata atau fitur yang sesuai dapat menggunakan kode seperti berikut.

```
list(zip(X_train_vect.toarray()[1], vect.get_feature_names_out()))
```

Kode program tersebut digunakan untuk menampilkan nilai TF-IDF beserta *term* atau kata. Hasil dari penerapan kode tersebut ditunjukkan pada Gambar 3.13.

```
[(0.0, 'abangkuu'),
 (0.0, 'abis'),
 (0.0, 'acara'),
 (0.0, 'ada'),
 (0.0, 'admin'),
 (0.0, 'ae'),
 (0.0, 'ah'),
 (0.0, 'aja'),
 (0.0, 'ajg'),
 (0.0, 'akhir'),
 (0.0, 'akses'),
 (0.0, 'aku'),
 (0.0, 'akun'),
 (0.0, 'alami'),
 (0.0, 'alas'),
 (0.0, 'alih'),
 (0.0, 'allah'),
 (0.0, 'ama'),
 (0.0, 'aman'),
 (0.0, 'ambil'),
```

Gambar 3.13 Nilai TF-IDF Beserta Term

3. Membangun model klasifikasi dengan Naive Bayes

Untuk melakukan pengklasifikasian data dengan menerapkan metode Naive Bayes Classifier dapat menggunakan kode seperti berikut.

```
#Naive Bayes Classifier
nb = MultinomialNB()

nb.fit(X_train_res, y_train_res)
```

Kode tersebut digunakan untuk membangun model klasifikasi Naive Bayes yang nantinya dapat digunakan untuk melakukan prediksi label kategori teks yang baru atau belum dikenal. Proses ini melibatkan pembelajaran pola dari data training yang sudah ada, sehingga model dapat mengidentifikasi dan membedakan pola teks yang sesuai dengan kategori atau label.

3.2.5 *Testing*

Pada tahap *testing* data dilakukan proses untuk dapat mengetahui keakuratan model yang sudah dibangun sebelumnya. Hasil yang diperoleh dari proses *testing* data berupa prediksi atau klasifikasi yang selanjutnya dapat dilakukan perhitungan dengan menggunakan *confusion matrix*. Untuk melakukan prediksi terhadap data *testing* dengan menggunakan kode seperti berikut.

```
# Fungsi predict
X_test_vect = vect.transform(X_test)

y_pred = nb.predict(X_test_vect)
```

Kode diatas digunakan untuk menghasilkan prediksi dari model klasifikasi terhadap data *test* yang telah dipersiapkan dalam format vektor fitur yang sesuai.

3.2.6 **Klasifikasi**

Setelah model berhasil dibuat, langkah selanjutnya adalah melakukan klasifikasi dengan menggunakan model yang sudah dibuat sebelumnya terhadap keseluruhan data yang belum diketahui kelas dan labelnya. Berikut merupakan kode program untuk melakukan klasifikasi secara otomatis terhadap keseluruhan data.

```
predicted = nbc.predict(np.asarray(dataset_klasifikasi))
predicted
```