SENTIMENT ANALYSIS OF AMAZON REVIEWS USING NAÏVE BAYES ALGORITHM

Abigail Tesalonika William, Jeanet Wynne W. Kastilong,
Maria Stefany Phileina F. Samosir, Zefanya Febrina Valentine Mehe
Universitas Multimedia Nusantara, Jalan Scientia Boulevard Gading, Curug Sangereng, Serpong,
Kabupaten Tangerang, Banten 15810

***Abstrak*—Analisis sentimen pada ulasan produk di Amazon menggunakan metode machine learning telah menjadi topik penelitian yang penting dalam domain e-commerce. Artikel ilmiah ini membahas langkah-langkah untuk melakukan analisis sentimen menggunakan algoritma Naive Bayes dan memberikan rekomendasi untuk meningkatkan kualitas analisis sentimen. Pemrosesan data, seperti pembersihan teks dan pembagian data, adalah langkah awal dalam analisis sentimen. Kemudian, algoritma Naive Bayes digunakan untuk mengklasifikasikan sentimen ulasan pengguna. Hasil analisis sentimen dapat memberikan wawasan berharga kepada penjual dan pengguna dalam memahami pandangan pengguna terhadap produk, meningkatkan kualitas produk, dan membantu pengambilan keputusan pembelian.**

***Kata Kunci*—analisis sentimen, ulasan produk, Amazon, metode machine learning, algoritma Naive Bayes.**

I. INTRODUCTION

Dalam dunia e-commerce, kerap kali ditemukan ulasan barang-barang yang dijual, dan walaupun sudah ada sistem rating, sistem rating tidak selalu merepresentasikan kalimat ulasan yang ditulis. Disinilah analisis sentimen berperan dalam analisa kata tersebut, karena sentiment analysis adalah proses yang melibatkan analisis dan pengkategorian informasi dari ulasan pelanggan sebagai pendapat positif, negatif, atau netral berdasarkan tiga tingkat analisis sentimen: dokumen, kalimat, dan aspek. Tujuan tingkat aspek adalah untuk menentukan target dari sentimen atau pendapat yang terwakili dalam ulasan pelanggan, yang dikenal sebagai identifikasi istilah aspect [1]. Dalam konteks Amazon sendiri sebagai e-commerce terbesar di dunia, analisis sentimen di Amazon melibatkan aplikasi analisis dan pengkategorian sentimen dari para penulis ulasan produk di Amazon yang memberikan konteks apakah ulasan tersebut bersifat positif, negatif, ataupun netral. Sistem analisis sentimen diterapkan di berbagai bidang bisnis dan sosial karena pendapat merupakan pengaruh utama terhadap perilaku penjual maupun juga pembeli [2]. Sentiment analysis of Amazon reviews memanfaatkan teknik pengolahan bahasa alami (natural language processing) untuk mengidentifikasi sentimen (positif atau negatif) dalam ulasan pengguna [2].

Dengan menganalisis sentimen dari ulasan-ulasan produk di Amazon, bisnis dapat memahami lebih baik tentang preferensi dan kebutuhan pelanggan mereka dan meningkatkan kualitas produk mereka untuk meningkatkan kepuasan pelanggan dan meningkatkan penjualan. Selain itu, analisis sentimen juga dapat membantu calon pembeli dalam memutuskan apakah suatu produk cocok untuk mereka berdasarkan pendapat pengguna sebelumnya. Hal ini dapat membantu dalam pengambilan keputusan dan mengurangi risiko pembelian produk yang tidak sesuai dengan kebutuhan mereka [3].

II. Literature Study

Penelitian *Sentiment Analysis of Amazon Reviews* menggunakan beberapa landasan teori yang didasarkan dari rumus-rumus algoritma umum yang biasanya digunakan dalam *Natural Language Processing*, yang berguna dalam identifikasi kata-kata dalam ulasan produk Amazon. Berdasarkan telaah literatur yang telah dilakukan, algoritma Naive Bayes, Stopwords, Lemmatization, serta Stemming adalah pilihan yang akan digunakan dalam penelitian ini.

2.1 *Naive Bayes*

Naive Bayes Classification Algorithm adalah salah satu algoritma *Supervised Learning* yang digunakan untuk memprediksi kelas atau label dari data baru berdasarkan kemungkinan terjadinya suatu kejadian. Algoritma ini didasarkan pada teori probabilitas *Bayes* dan asumsi *naive independence assumption*, yaitu bahwa setiap fitur input saling independen satu sama lain [5].

Cara kerja *Naive Bayes Classification Algorithm* adalah dengan menghitung probabilitas posteriori dari setiap kelas atau label berdasarkan fitur input yang diberikan, kemudian memilih kelas dengan probabilitas posteriori tertinggi sebagai prediksi untuk data baru. Algoritma ini cocok untuk digunakan pada data dengan jumlah fitur input yang besar dan tidak terlalu kompleks [5].

Contoh penggunaan *Naive Bayes Classification Algorithm* adalah pada klasifikasi *email spam*. Algoritma ini dapat mempelajari pola dari email yang sudah diberi label *spam* atau *non-spam*, kemudian digunakan untuk memprediksi label dari *email* baru yang belum pernah dilihat sebelumnya. Algoritma ini juga dapat digunakan pada berbagai tugas klasifikasi lainnya, seperti klasifikasi teks, pengenalan wajah, dan sebagainya.

Rumus Naive Bayes Classification Algorithm:

$$P(c\,|x) = \frac{P(x|c)P(c)}{P(x)}$$

- *x = Contoh data yang memiliki kelas (label) yang tidak diketahui*
- *c = Hipotesis bahwa X adalah kelas data (label)*
- *P(c|c) =Probabilitas hipotesis H berdasarkan kondisi*
- *X P(c) = Probabilitas Hipótesis H*
- *P(x|c) = Probabilitas sampel data X berdasarkan kondisi Hipotesis H*
- *P(x) = Probabilitas dari X*

2.2 *Multinomial Naive Bayes*

Multinomial Naive Bayes, yang merupakan algoritma yang akan digunakan untuk menganalisis sentimen review produk Amazon. Algoritma Multinomial Naive Bayes adalah metode pembelajaran probabilitas yang banyak digunakan dalam Natural Language Processing (NLP) [4].

Pengklasifikasi Naive Bayes adalah kumpulan dari banyak algoritme di mana semua algoritma berbagi satu prinsip umum, yaitu setiap fitur yang diklasifikasikan tidak terkait dengan fitur lainnya. Ada atau tidaknya suatu fitur tidak mempengaruhi ada tidaknya fitur yang lain [4].

Rumus Multinomial Naive Bayes Algorithm:

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

- *P(c) adalah probabilitas awal dari kelas c*
- *$P(w_i|$ c) adalah probabilitas bersyarat bahwa kata w_i milik kelas c*
- *f_i^d adalah jumlah kemunculan w_i dalam dokumen d.*

2.3 *Stopwords*

Stopwords adalah kata-kata yang sering muncul dalam suatu bahasa dan mempunyai fungsi sebagai kata sambung, namun tidak memiliki makna atau arah yang signifikan. Contoh dari kata-kata stopwords ini adalah "dan", "atau", "tetapi", atau dalam bahasa Inggris seperti *"and", "or"*, dan *"but"*.

Dalam analisis sentimen, teknik preprocessing seperti menghapus stopwords sering digunakan untuk meningkatkan akurasi model klasifikasi sentimen. Penghapusan stopwords dapat mengurangi kebisingan dalam data teks dan meningkatkan kinerja model pembelajaran mesin [6].

2.4 *Stemming*

Stemming adalah sebuah teknik dalam analisis sentimen yang berguna untuk memetakan atau "memetik" berbagai bentuk awalan dan akhiran kata menjadi kata akarnya. Contoh dari stemming adalah mengubah kata "jumping" atau "jumped" menjadi "jump". Efektivitas dari metode stemming ini sendiri banyak bergantung dari kata-kata dan juga bahasa yang digunakan [7].

Dampak dari stemming sendiri dapat dilihat dari membandingkan classification models dari analisis sentimen yang menggunakan stemming atau tidak [9].

2.5 *Lemmatizing*

Lemmatizing adalah teknik preprocessing text yang digunakan dalam analisis sentimen untuk mengurangi kepadatan dan normalisasi kata-kata kembali ke akar kata secara *grammar*. Lemmatizing melibatkan penggantian token tertentu dengan lemma atau bentuk baku kata yang sesuai, yang merupakan bentuk dasar kata tersebut. Misalnya, lemma untuk kata 'lebih baik' adalah 'baik', dan lemma untuk kata 'membawa' adalah 'bawa' [11].

Lemmatizing sendiri bukan tanpa kendala, sebab proses perubahan sintaks yang diperlukan untuk mencocokkan kata-kata dalam

leksikon afektif (terutama lemmatisasi) rentan terjadi salah prediksi, yang dapat memiliki dampak signifikan dan secara statistik berpengaruh terhadap kinerja model analisis sentimen berbasis kamus yang sederhana [13].

2.6 *Previous Research*

Xu Yun [13] dkk dari Stanford University menggunakan algoritma pembelajaran terawasi yang ada seperti algoritma perceptron, naive bayes, dan mesin vektor pendukung untuk memprediksi peringkat ulasan. Mereka melakukan validasi silang dengan 70% data.

Maria Soledad Elli [14] melakukan sentimen dari mempertimbangkan ulasan pelanggan. Mereka telah menganalisis hasil untuk mengembangkan model bisnis. Penulis mempresentasikan bahwa alat memberikan akurasi yang lebih baik. Penelitian telah memanfaatkan Multinomial Naive Bayesian. Itu bertindak sebagai pengklasifikasi. Mekanisme juga mendukung mesin vektor.

Callen Rain [15] telah memperluas penelitian di daerah pengolahan bahasa alami. Penelitian memanfaatkan Naive Bayesian serta pengklasifikasi daftar keputusan. Mekanisme ini telah digunakan untuk mengkategorikan review yang diberikan. Ulasan ini bisa positif atau negatif. Penelitian memanfaatkan jaringan saraf Deep-learning. Neural network telah ditemukan terkenal di bidang analisis sentimen

III. METHODOLOGY

Metodologi penelitian dalam *Sentiment Analysis of Amazon Reviews menggunakan Naïve Bayes Algorithm* dengan metode *CRISP-DM (Cross-Industry Standard Process for Data Mining)*. *Cross Industry Standard Process for Data Mining* (CRISP-DM) merupakan sebuah model dalam data mining yang digunakan untuk menggambarkan sebuah siklus, dimana CRISP-DM terdiri atas 6 tahapan yaitu *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment* [8].

3.1 *Business Understanding*

Merupakan tahapan pertama dimana tahapan ini digunakan untuk memahami lebih dalam tentang substansi suatu kegiatan. Dalam hal ini, sentiment analysis yang dilakukan pada Amazon digunakan untuk meningkatkan kepuasan pelanggan serta meningkatkan penjualan.

3.2 *Data Understanding*

tahapan kedua dimana pada tahapan ini data dari Amazon yang berupa ulasan dan penilaian dari pengguna dikumpulkan untuk dieksplorasi dan dianalisis lebih lanjut.

3.3 *Data Preparation*

tahapan dimana data mulai dirapikan dan dibersihkan dari *missing values* dan lainnya, untuk nantinya siap dalam proses pemodelan.

3.4 *Modeling*

merupakan tahapan dalam menentukan model, dalam hal ini algoritma yang digunakan oleh peneliti yaitu Multinomial Naive Bayes [12].

3.5 *Evaluation*

tahapan ini berisikan fase untuk menganalisa model yang digunakan sebelumnya, apakah sesuai dengan tujuan awal.

3.6 *Deployment*

tahapan akhir ini merupakan tahapan presentasi dari hasil evaluasi model sebelumnya, dan melakukan pengujian terhadap salah satu review apakah termasuk positif atau negatif.

IV. Result and Analysis

4.1.1 *Data Understanding*

```
In [3]: amazon.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34660 entries, 0 to 34659
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    34660 non-null  object
 1   name                  27900 non-null  object
 2   asins                 34658 non-null  object
 3   brand                 34660 non-null  object
 4   categories            34660 non-null  object
 5   keys                  34660 non-null  object
 6   manufacturer          34660 non-null  object
 7   reviews.date          34621 non-null  object
 8   reviews.dateAdded     24039 non-null  object
 9   reviews.dateSeen      34660 non-null  object
 10  reviews.didPurchase   1 non-null      object
 11  reviews.doRecommend   34006 non-null  object
 12  reviews.id            1 non-null      float64
 13  reviews.numHelpful    34131 non-null  float64
 14  reviews.rating        34627 non-null  float64
 15  reviews.sourceURLs    34660 non-null  object
 16  reviews.text          34659 non-null  object
 17  reviews.title         34655 non-null  object
 18  reviews.userCity      0 non-null      float64
 19  reviews.userProvince  0 non-null      float64
 20  reviews.username      34658 non-null  object
dtypes: float64(5), object(16)
memory usage: 5.6+ MB
```

Gambar 4.1 Dataset Info

Kode “amazon.info()” digunakan untuk mendapatkan informasi detail tentang dataset yang digunakan. Metode ini adalah bagian dari library Pandas dalam bahasa pemrograman Python dan digunakan untuk menampilkan informasi statistik dan struktur data dalam dataframe (dataset) yang sedang dianalisis.

```
In [4]: review = amazon[["reviews.text", "reviews.rating"]].sample(10000, random_state=23)
        review.head()
```

Out[4]:

	reviews.text	reviews.rating
21536	Bought as a Mother's Day Gift. This is great f...	4.0
20669	I can hold this next to my Kindle Paperwhite a...	5.0
30656	Love this device and went on to buy 2 as gifts...	5.0
25297	With some technical savvy, you can quickly hav...	5.0
9016	bought for grandkids they love them. wise choi...	5.0

Gambar 4.2 Kolom Review Text & Rating

Menggunakan metode “sample()” dari library Pandas untuk mengambil sampel acak dari kolom “reviews.text” dan “reviews.rating”. Sampel ini diambil dengan menggunakan random_state 23, hasil dari langkah ini disimpan dalam variabel “review” yang merupakan subset dataset asli. Variabel ini berisi 10.000 baris acak dari kolom “reviews.text” yang berisi teks ulasan pengguna dan kolom “reviews.rating” yang berisi peringkat ulasan.

4.1.2 *Data Preparation*

```
In [5]: review.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 10000 entries, 21536 to 29020
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   reviews.text    10000 non-null  object
 1   reviews.rating  9993 non-null   float64
dtypes: float64(1), object(1)
memory usage: 234.4+ KB

In [6]: review.dropna(inplace=True)
        review.isnull().sum()

Out[6]: reviews.text    0
        reviews.rating  0
        dtype: int64
```

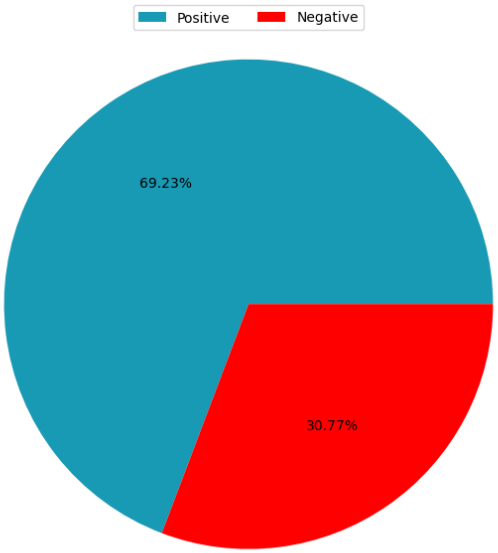
Gambar 4.3 Kolom Review Info

Menggunakan kode “review.info()” untuk memperoleh informasi detail tentang dataset yang diambil sampel sebanyak 10.000 dari Amazon, setelah itu dilakukan penghapusan data null dengan metode “review.dropna(inplace=True)” untuk menghapus baris yang mengandung nilai null (missing values) dalam dataset, setelah itu, menggunakan kode “review.isnull().sum()” untuk memeriksa apakah ada nilai null yang tersisa dalam dataset

```
In [7]: review['reviews.rating'].value_counts().sort_index(ascending=False)

Out[7]: 5.0    6891
        4.0    2455
        3.0     420
        2.0     120
        1.0     107
        Name: reviews.rating, dtype: int64

In [8]: review5 = review[review['reviews.rating']==5].sample(150,random_state=43)
        review4 = review[review['reviews.rating']==4].sample(150,random_state=43)
        review3 = review[review['reviews.rating']==3].sample(150,random_state=43)
        review2 = review[review['reviews.rating']==2].sample(100,random_state=43)
        review1 = review[review['reviews.rating']==1].sample(100,random_state=43)

        review = pd.concat([review5,review4,review3,review2,review1])

In [9]: review['reviews.rating'].value_counts().sort_index(ascending=False)

Out[9]: 5.0    150
        4.0    150
        3.0    150
        2.0    100
        1.0    100
        Name: reviews.rating, dtype: int64
```

Gambar 4.4 Kolom Sampling Review

Menghitung distribusi peringkat ulasan pada dataset dengan kode “review['reviews.rating'].value_counts().sort_index(ascending=False)”, sentimen positif berjumlah ribuan dan sentimen negatif hanya berjumlah ratusan. Oleh karena itu harus melakukan pengambilan sampel yang seimbang dari setiap peringkat ulasan agar dataset yang digunakan untuk pengujian memiliki representasi yang setara dari berbagai sentimen, sehingga mengurangi bias yang mungkin timbul pada saat pengujian algoritma Naive Bayes.



Distribusi Sentiment

Gambar 4.5 Distribusi Rating

Diagram pie ini memberikan gambaran visual yang jelas tentang sebaran sentimen positif dan negatif dalam dataset review. Dengan melihat proporsi relatif dari setiap sentimen, kita dapat memahami bagaimana sentimen ulasan terdistribusi dalam dataset yang digunakan dalam analisis sentimen.

Convert to csv

```
# export cleaned data to csv
review.to_csv('cleaned_review.csv', index=False)
```

Import dataset

```
clean_review = pd.read_csv('cleaned_review.csv')
clean_review.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 650 entries, 0 to 649
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   reviews.text     650 non-null    object
 1   reviews.rating   650 non-null    float64
 2   sentiment_score  650 non-null    int64
 3   sentiment        650 non-null    object
 4   text             650 non-null    object
dtypes: float64(1), int64(1), object(3)
memory usage: 25.5+ KB
```

Gambar 4.6 Cleaned Dataset Info

```
clean_review[clean_review['text'].isnull()]
clean_review
```

	reviews.text	reviews.rating	sentiment_score	sentiment	text
0	The product was easy to set up, and is easy to...	5.0	1	Positive	product be easy set up easy use i love hand free
1	I came in to Best Buy to get a game for my BF ...	5.0	1	Positive	i come in best buy get game for my bf end up w...
2	Bought this for my toddler and teen they both ...	5.0	1	Positive	bought for my toddler teen they both love
3	The product works great. If you need help usin...	5.0	1	Positive	product work great if you need help use just l...
4	It may be cheap but this is a great tablet and...	5.0	1	Positive	may be cheap but great tablet work awesome
...	...	...	...	...	...
645	So I was already not too pleased with Amazon's...	1.0	0	Negative	so i be already not too pleased with amazon s ...
646	We started out great. We had a great thing goi...	1.0	0	Negative	we start out great we have great thing go i lo...
647	Pretty dumb that you have to buy a charger port	1.0	0	Negative	pretty dumb you have buy charger port
648	I HATE this machine. First, I specifically ask...	1.0	0	Negative	i hate machine first i specifically ask salesp...
649	When you buy an Amazon product from Best buy t...	1.0	0	Negative	when you buy amazon product from best buy they...

650 rows × 5 columns

```
clean_review.dropna(inplace=True)
clean_review.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 650 entries, 0 to 649
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   reviews.text     650 non-null    object
 1   reviews.rating   650 non-null    float64
 2   sentiment_score  650 non-null    int64
 3   sentiment        650 non-null    object
 4   text             650 non-null    object
dtypes: float64(1), int64(1), object(3)
memory usage: 25.5+ KB
```

Gambar 4.7 Cleaned Dataset Missing Value

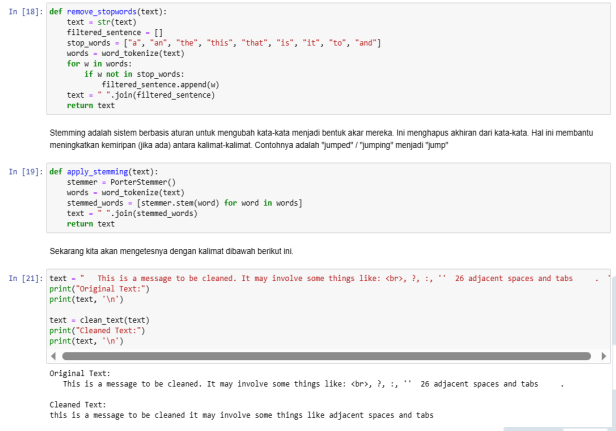
Setelah membersihkan data dan menghapus nilai null, kita dapat mengekspor dataset yang telah dibersihkan menjadi file CSV, Setelah file CSV berhasil diekspor, diimport dataset yang telah dibersihkan kembali ke dalam DataFrame. Kode "clean_review[clean_review['text'].isnull()]" digunakan untuk mencari baris-baris di dataset clean_review yang memiliki nilai null pada kolom 'text', Selanjutnya, kode "clean_review.dropna(inplace=True)" digunakan untuk menghapus baris-baris yang memiliki nilai null dalam dataset clean_review

4.1.3 *Modeling*

```
In [17]: def clean_text(text):
             text = str(text)
             text = text.lower()
             text = text.strip()
             text = re.sub(' \d+', ' ', text)
             text = re.compile('<.*?>').sub('', text)
             text = re.compile('[%s]' % re.escape(string.punctuation)).sub(' ', text)
             text = re.sub('\s+', ' ', text)
             text = text.strip()
             return text
```

Gambar 4.8 Text Cleaning Code Module

Fungsi ini akan mengembalikan teks ulasan yang telah melalui serangkaian langkah pembersihan, termasuk mengubah menjadi huruf kecil, menghapus angka, menghapus tag HTML, menghapus tanda baca, dan menggabungkan spasi berulang menjadi satu.



```
In [18]: def remove_stopwords(text):
             text = str(text)
             filtered_sentence = []
             stop_words = ["a", "an", "the", "this", "that", "is", "it", "to", "and"]
             words = word_tokenize(text)
             for w in words:
                 if w not in stop_words:
                     filtered_sentence.append(w)
             text = " ".join(filtered_sentence)
             return text
```

Stemming adalah sistem berbasis aturan untuk mengubah kata-kata menjadi bentuk akar mereka. Ini menghapus akhiran dari kata-kata. Hal ini membantu meningkatkan kemiripan (jika ada) antara kalimat-kalimat. Contohnya adalah "jumped" / "jumping" menjadi "jump"

```
In [19]: def apply_stemming(text):
             stemmer = PorterStemmer()
             words = word_tokenize(text)
             stemmed_words = [stemmer.stem(word) for word in words]
             text = " ".join(stemmed_words)
             return text
```

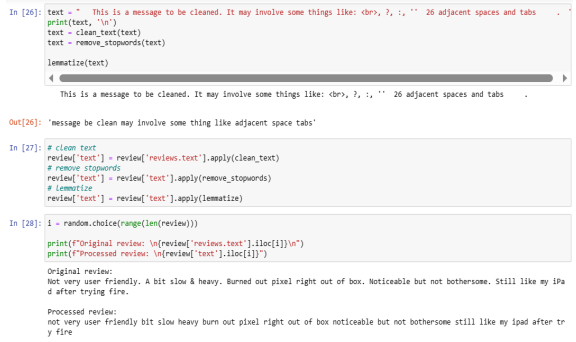
Sekarang kita akan mengetesnya dengan kalimat dibawah berikut ini.

```
In [21]: text = "   This is a message to be cleaned. It may involve some things like: <br>, ?, :, ''  26 adjacent spaces and tabs     .
         print("Original Text:")
         print(text, '\n')

         text = clean_text(text)
         print("Cleaned Text:")
         print(text, '\n')
```

```
Original Text:
   This is a message to be cleaned. It may involve some things like: <br>, ?, :, ''  26 adjacent spaces and tabs     .

Cleaned Text:
this is a message to be cleaned it may involve some things like adjacent spaces and tabs
```

Gambar 4.9 Remove Stopwords and Stemming Code Module

Fungsi “remove_stopwords(text)” yang diberikan adalah untuk menghapus kata-kata yang tidak signifikan (stop words) dari teks ulasan dalam dataset, selanjutnya fungsi “apply_stemming()”, dapat menerapkan stemming pada teks ulasan dalam dataset. Stemming membantu mengurangi variasi kata ke bentuk dasarnya sehingga kata-kata yang serupa dapat diperlakukan secara konsisten. Pada tahap testing, dilakukan pembersihan teks ulasan menggunakan fungsi “clean_text()”, hasilnya adalah teks ulasan yang telah terstruktur dengan baik



```
In [26]: text = "   This is a message to be cleaned. It may involve some things like: <br>, ?, :, ''  26 adjacent spaces and tabs     .
         print(text, '\n')
         text = clean_text(text)
         text = remove_stopwords(text)

         lemmatize(text)
```

```
   This is a message to be cleaned. It may involve some things like: <br>, ?, :, ''  26 adjacent spaces and tabs     .
```

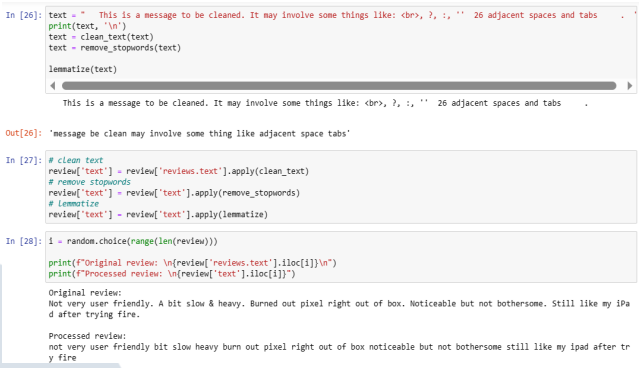
```
Out[26]: 'message be clean may involve some thing like adjacent space tabs'
```

```
In [27]: # clean text
         review['text'] = review['reviews.text'].apply(clean_text)
         # remove stopwords
         review['text'] = review['text'].apply(remove_stopwords)
         # Lemmatize
         review['text'] = review['text'].apply(lemmatize)
```

```
In [28]: i = random.choice(range(len(review)))

         print(f"Original review: \n{review['reviews.text'].iloc[i]}\n")
         print(f"Processed review: \n{review['text'].iloc[i]}")
```

```
Original review:
Not very user friendly. A bit slow & heavy. Burned out pixel right out of box. Noticeable but not bothersome. Still like my iPa
d after trying fire.

Processed review:
not very user friendly bit slow heavy burn out pixel right out of box noticeable but not bothersome still like my ipad after tr
y fire
```

Gambar 4.10 Lemmatize Code Module



```
In [26]: text = "   This is a message to be cleaned. It may involve some things like: <br>, ?, :, ''  26 adjacent spaces and tabs     .
         print(text, '\n')
         text = clean_text(text)
         text = remove_stopwords(text)

         lemmatize(text)
```

```
   This is a message to be cleaned. It may involve some things like: <br>, ?, :, ''  26 adjacent spaces and tabs     .
```

```
Out[26]: 'message be clean may involve some thing like adjacent space tabs'
```

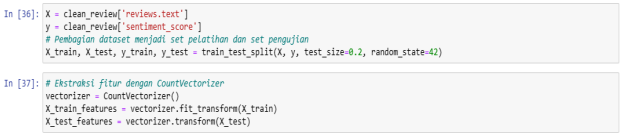
```
In [27]: # clean text
         review['text'] = review['reviews.text'].apply(clean_text)
         # remove stopwords
         review['text'] = review['text'].apply(remove_stopwords)
         # Lemmatize
         review['text'] = review['text'].apply(lemmatize)
```

```
In [28]: i = random.choice(range(len(review)))

         print(f"Original review: \n{review['reviews.text'].iloc[i]}\n")
         print(f"Processed review: \n{review['text'].iloc[i]}")
```

```
Original review:
Not very user friendly. A bit slow & heavy. Burned out pixel right out of box. Noticeable but not bothersome. Still like my iPa
d after trying fire.

Processed review:
not very user friendly bit slow heavy burn out pixel right out of box noticeable but not bothersome still like my ipad after tr
y fire
```

Gambar 4.11 Applying Clean Text, Remove Stopwords and Lemmatize to the Review dataset

Dengan menggunakan fungsi “”lemmatize()”, kita dapat melakukan lemmatisasi pada teks ulasan dalam dataset. Lemmatisasi membantu mengubah kata-kata menjadi bentuk dasar mereka (lemma), sehingga kata-kata dengan bentuk yang berbeda namun memiliki makna yang sama akan dianggap serupa. Dengan menerapkan fungsi preprocessing menggunakan “apply()” pada kolom 'reviews.text' dari DataFrame 'review', kita dapat membersihkan, menghapus stopwords, dan melakukan lemmatisasi pada teks ulasan secara efisien, sehingga mempersiapkannya untuk langkah-langkah selanjutnya dalam analisis sentimen



```
In [36]: X = clean_review['reviews.text']
         y = clean_review['sentiment_score']
         # Pembagian dataset menjadi set pelatihan dan set pengujian
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```



```
In [37]: # Ekstraksi fitur dengan CountVectorizer
         vectorizer = CountVectorizer()
         X_train_features = vectorizer.fit_transform(X_train)
         X_test_features = vectorizer.transform(X_test)
```

Gambar 4.12 Splitting cleaned review dataset to train and test data

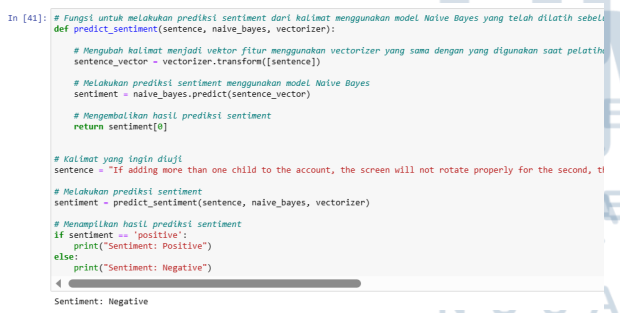
Kode ini membagi dataset menjadi training set dan testing set. Variabel X berisi teks ulasan, sedangkan variabel y berisi skor sentimen yang menjadi target. Pembagian dataset dilakukan dengan rasio 80:20, di mana 80% data digunakan sebagai training set dan 20% sebagai testing set. Pengaturan random_state sebesar 42, lalu digunakan CountVectorizer untuk mengubah teks ulasan menjadi representasi numerik.

4.1.4 *Evaluation*

Gambar 4.13 Melatih Model Dengan Naive Bayes

Melatih model Naive Bayes dengan menggunakan algoritma MultinomialNB, lalu memprediksi sentimen pada test set menggunakan model Naive Bayes yang telah dilatih sebelumnya, dan terakhir dievaluasi kinerja model dengan menghitung akurasi, presisi, recall, dan F1-Score

4.1.5 *Deployment*



Gambar 4.14 Menguji Kalimat Review

Kode pada gambar diatas menggunakan fungsi "predict_sentiment()" untuk melakukan prediksi sentimen dari sebuah kalimat. Hasil prediksi sentimen kemudian ditampilkan sebagai "Sentiment: Positive" jika hasilnya positif, dan "Sentiment: Negative" jika hasilnya negatif.

V. CONCLUSIONS

Artikel ilmiah ini membahas tentang cara melakukan analisis sentimen pada ulasan produk di Amazon menggunakan metode machine learning, dengan fokus pada rekomendasi untuk meningkatkan kualitas analisis sentimen tersebut. Dalam artikel ini, telah dijelaskan beberapa langkah yang perlu dilakukan untuk melakukan analisis sentimen pada ulasan produk di Amazon menggunakan metode machine learning. Langkah-langkah tersebut meliputi pemrosesan data, seperti membersihkan teks dan mengubah kata-kata menjadi bentuk dasar, serta pembagian data menjadi set pelatihan dan set pengujian. Metode machine learning yang digunakan adalah algoritma Naive Bayes.

Hasil evaluasi menunjukkan bahwa model Naive Bayes memiliki kinerja yang baik dalam mengklasifikasikan sentimen ulasan produk di Amazon. Dengan akurasi sebesar 81%, model mampu mengidentifikasi sentimen dengan tingkat keberhasilan sekitar 81%. Presisi sebesar 80.67% menunjukkan bahwa model memberikan proporsi yang baik dalam mengenali ulasan yang sesuai dengan sentimen tertentu. Recall sebesar 81.54% menunjukkan bahwa model mampu menemukan sebagian besar ulasan yang memiliki sentimen positif atau negatif secara keseluruhan. F1-score sebesar 0.8089 menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan recall. Dalam keseluruhan, analisis sentimen menggunakan algoritma Naive Bayes pada ulasan pengguna Amazon dapat membantu dalam memahami pandangan pengguna terhadap produk dan memberikan informasi berharga kepada penjual.

REFERENCES

[1] E. H. Hajar and B. Mohammed, *Using Synonym and Definition WordNet Semantic relations for implicit aspect identification in Sentiment Analysis*. 2019. doi: 10.1145/3320326.3320406.

[2] R. Alroobaea, "Sentiment Analysis on Amazon Product Reviews using the Recurrent Neural Network (RNN)," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, Jan. 2022, doi: 10.14569/ijacsa.2022.0130437.

[3] R. K. Sinha, "Data Analysis and Sentiment Analysis on Amazon Reviews," *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. 12, pp. 2200–2206, Dec. 2021, doi: 10.22214/ijraset.2021.39725.

[4] J. Winahyu and I. Suharjo, "Aplikasi Web Analisis Sentimen Dengan Algoritma Multinomial Naïve Bayes," *Karmapati (Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika)*, vol. 10, no. 2, p. 206, Aug. 2021, doi: 10.23887/karmapati.v10i2.36609.

[5] J. C. Tesoro, "A Semantic Approach of the Naïve Bayes Classification Algorithm," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3287–3294, Jun. 2020, doi: 10.30534/ijatcse/2020/125932020.

[6] K. Ghag and K. A. Shah, *Comparative analysis of effect of stopwords removal on sentiment classification*. 2015. doi: 10.1109/ic4.2015.7375527.

[7] S. Al-Saqqa, A. Awajan, and S. Ghoul, *Stemming Effects on Sentiment Analysis using Large Arabic Multi-Domain Resources*. 2019. doi: 10.1109/snams.2019.8931812.

[8] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Computer Science*, vol. 181, pp. 526–534, 2021. doi:10.1016/j.procs.2021.01.199

[9] A. T. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts," *Kinetik : Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 375–380, Oct. 2019, doi: 10.22219/kinetik.v4i4.912.

[10] M. Uddin, Md. F. B. Hafiz, Md. S. Hossain, and S. Md. A. Islam, "Drug Sentiment Analysis using Machine Learning Classifiers," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 1, Jan. 2022, doi: 10.14569/ijacsa.2022.0130112.

[11] M. Vassallo, G. Gabrieli, V. Basile, and C. Bosco, "The Tenuousness of Lemmatization in Lexicon-based Sentiment Analysis.," *Italian Conference on Computational Linguistics*, vol. 2481, pp. 1–6, Jan. 2019, [Online]. Available: http://ceur-ws.org/Vol-2481/paper74.pdf

[12] A. A. Farisi, Y. Sibaroni, and S. A. Faraby, "Sentiment Analysis on hotel reviews using multinomial naïve Bayes classifier," *Journal of Physics: Conference Series*, vol. 1192, p. 012024, 2019. doi:10.1088/1742-6596/1192/1/012024

[13]Y. Xu, X. Wu, and Q. Wang. Sentiment analysis of yelps ratings based on text reviews, 2015.

[14]M. S. Elli and Y.-F. Wang. Amazon reviews, business analytics with sentiment analysis

[15]C. Rain. Sentiment analysis in amazon reviews using probabilistic machine learning. Swarthmore College, 2013.

Pembagian Kerja

1). Abigail Tesalonika
- laporan

2). Jeannet Kastilong
- laporan
- codingan

3). Maria Stefany
- laporan
- ppt

4). Zefanya Febrina
- laporan
- codingan