LAPORAN AKHIR PROYEK PENGOLAHAN DATA BESAR

Sentiment Analysis – IMDB Movie Reviews (Klasifikasi)

Disusun oleh:

12S17011	Astri Monica Sianturi
12S17013	Mega Sari Pasaribu
12S17046	Pebri Sangmajadi Sinaga
12S17047	Christina Clara
12S17053	Rommel Parasian Gultom

PROGRAM STUDI SARJANA SISTEM INFORMASI

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

2021

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR POTONGAN KODE PROGRAM

I. PENDAHULUAN

Pada bab ini akan membahas mengenai latar belakang, tujuan, dan ruang lingkup dari pengerjaan proyek yang akan dilakukan.

1.1 Latar belakang

Analisis Sentimen adalah studi komputasional dengan menggunakan opini, sentimen, dan emosi dari objek pengguna (orang) melalui entitas atau atribut yang dimiliki yang diekspresikan dalam bentuk teks [1]. Sentimen analisis yang merupakan pemrosesan bahasa dengan menggunakan pendekatan pembelajaran mesin untuk mendefinisikan apakah suatu penggalan teks dapat dikategorikan pada respon positif atau negatif. Saat ini masyarakat yang sudah lekat dengan teknologi komunikasi sangat mudah dalam menyampaikan tanggapan, baik dalam bentuk ulasan, saran, atau komentar. Tanggapan yang diberikan oleh masyarakat dapat ditemukan dalam berbagai aspek, misalnya pendidikan, ekonomi, hiburan, dsb.

Pada industri hiburan, film menjadi salah satu karya yang banyak diminati sekaligus diulas oleh masyarakat. Masyarakat sebagai penikmat film, dapat memberikan tanggapannya langsung terkait kesan yang diterima selama menonton film. Bahkan saat ini, industri perfilm-an sudah sangat erat dengan kata *rate*. Respon/ulasan masyarakat terhadap sebuah film menjadi penentu apakah film tersebut dapat dikategorikan bagus atau tidak. Akibat kemudahan pemberian ulasan oleh penikmat film dan semakin banyaknya film yang beredar di masyarakat, muncullah permasalahan terkait sulitnya menentukan tanggapan penonton termasuk pada respon positif atau negatif dengan data yang sangat banyak. Untuk mempermudah menentukan *review* movie yang bagus dan yang buruk maka perlu untuk mengklasifikasikan teks ulasan penonton (sentimen) yang mungkin berdasarkan review yang sudah diberikan oleh penonton sebelumnya. Setiap *review* akan diproses sehingga menghasilkan klasifikasi sentimen yang positif dan negatif. Data yang diberikan masih berupa *review* dari berbagai penonton dengan bahasa dan tanda baca yang bebas, oleh karena itu data yang diperoleh terlebih dahulu di*preprocessing* agar mudah untuk diklasifikasikan.

1.2 Tujuan

Adapun tujuan dari pengerjaan proyek ini adalah untuk melakukan analisis sentimen terhadap *dataset* IMDB *Movie Reviews*. Sentimen analisis *review movie* dilakukan untuk mengetahui klasifikasi dari *movie* berdasarkan *review* yang ada, apakah *movie* tersebut termasuk ke dalam kategori positif atau negatif.

1.3 Scope

Adapun batasan dari pengerjaan sentimen analisis dari *dataset* IMDB *Movie Reviews* adalah data *movie review* sebagai berikut.

1. Algoritma yang digunakan dalam melakukan analisis sentimen terhadap IMDB *Movie Reviews* adalah Naive Bayes.
2. Data yang digunakan adalah *dataser* IMDB *Movie Reviews* yang diperoleh dari Kaggle yang dapat diakses dengan menggunakan tautan berikut https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.

II. PERMASALAHAN

Pada bab ini dijelaskan masalah yang akan diselesaikan dalam proyek Pengenalan *Big Data*. Permasalahan proyek pengenalan *Big Data* yang akan diselesaikan adalah melakukan klasifikasi analisis sentimen pada *review IMDB-movies*. Sumber data yang akan digunakan merupakan data yang telah terstruktur dimana data tersebut telah disimpan dalam format *file* .csv.

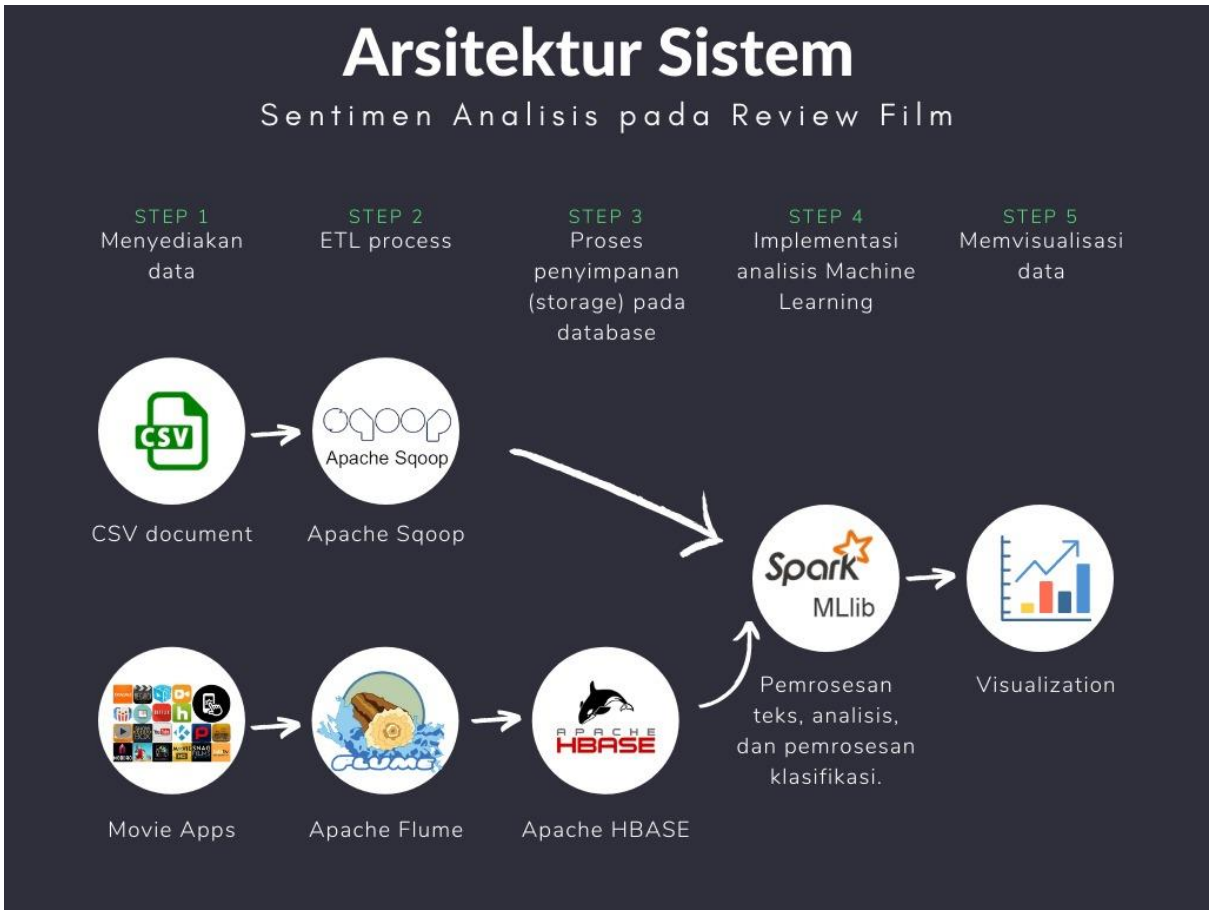
Arsitektur sistem akan didesain dapat mengolah data terstruktur dan data tidak terstruktur. Hal tersebut berguna untuk mengetahui cara klasifikasi yang dapat digunakan apabila data yang digunakan merupakan data *streaming*.

Berdasarkan kebutuhan diatas maka permasalahan yang akan dirumuskan adalah bagaimana suatu sistem dapat mengklasifikasikan suatu film berdasarkan *review* yang diberikan penonton sebagai ulasan yang bernilai positif atau ulasan yang negatif menggunakan algoritma Naive Bayes.

III. SOLUSI

Pada bab ini diberikan sebuah arsitektur sistem yang dapat menjadi solusi yang digunakan dalam masalah pengklasifikasian *sentiment analysis review* IMDB *movie*.

Arsitektur *big data* adalah suatu struktur logis atau fisik yang mampu menangani besar data yang akan disimpan, diakses dan diolah dalam suatu *big data* atau lingkungan TI, untuk mendefinisikan besar dari solusi *big data* maka akan bekerja berdasarkan komponen inti yang digunakan, arus informasi, keamanan dan lainnya. Arsitektur *big data* nantinya akan menjadi referensi dalam merancang infrastruktur *big data* dengan solusi-solusinya.



Gambar 1. Arsitektur Sistem

Pada gambar diatas dijelaskan alur arsitektur sistem yang akan digunakan dalam mengklasifikasi *sentiment analysis* pada *review* IMDB-*movies*. Arsitektur sistem dibentuk agar dapat mengolah data yang terstruktur dan tidak terstruktur kemudian menggunakan spark untuk melakukan implementasi *analysis Machine Learning*, berikut tahapan yang akan digunakan:

1. Menyediakan data

   Hal yang pertama yang harus dipersiapkan adalah data. Data terdiri dari dua jenis yaitu terstruktur dan tidak terstruktur. Data yang akan digunakan yaitu data terstruktur dan data tidak terstruktur. Data yang terstruktur dapat diketahui apabila data tersebut telah tersimpan dalam sebuah format file. Data yang tidak terstruktur atau yang disebut dengan *unstructured data* merupakan data yang tidak mengikuti suatu susunan format tertentu sebagai contoh data yang berasal dari sebuah media sosial atau *website*.

2. ETL Process

   Setelah mengetahui jenis data yang digunakan, kemudian akan dilanjutkan dengan tahapan ETL Process. *Extraction, Transfer, Loading* atau dikenal dengan ETL adalah sebuah proses sebuah fase pemrosesan data dari sumber data ke dalam satu penyimpanan yang konsisten dan dimuat ke dalam gudang data. Data terstruktur akan diproses dengan menggunakan *Apache Sqoop*, sedangkan semi struktur, terstruktur dan tidak terstruktur seperti data *streaming* akan diproses dengan menggunakan *Apache Flume*.

3. Proses penyimpanan (*storage*) pada *database*

   Setelah data diproses kemudian data akan disimpan dalam HBase. HBase adalah *database* terdistribusi yang berorientasi pada kolom. HBase adalah program yang berjalan diatas *Hadoop Distributed File System* (HDFS) yang mampu memproses data dalam skala besar secara interaktif. HBase baik digunakan karena memiliki sifat *fault tolerant*, artinya HBase mampu menangani keutuhan data meskipun terjadi sebuah kegagalan pada sistem yang digunakan, dengan cara mengolah data kembali berdasarkan *historical* pengolahan data tersebut.

4. Implementasi analisis *Machine Learning*

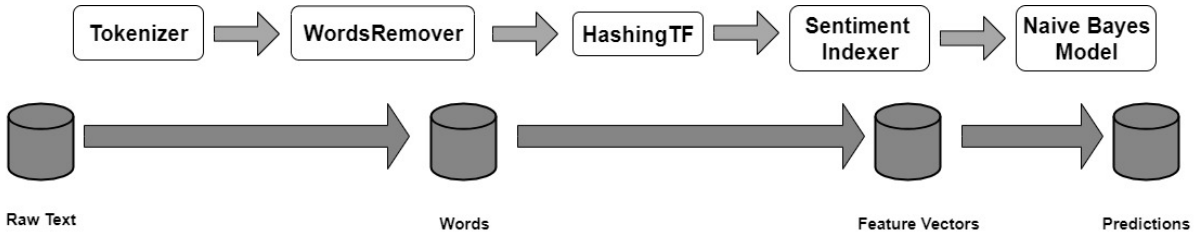
   Pada tahap ini, data yang disimpan akan dilakukan pemrosesan data dimana berguna untuk menghilangkan data yang *noise*, kemudian data yang telah dilakukan *preprocessing*, akan dipakai kembali untuk melakukan klasifikasi analisis sentimen pada *review* IMDB-*movies*.

5. Memvisualisasikan data

   Pada tahapan ini, hasil klasifikasi sentimen analisis pada *review* IMDB-*movie* akan diolah kembali untuk divisualisasikan berdasarkan hasil klasifikasi sentiment analisis untuk mempermudah dalam memahami hasil pengolahan data.

IV. MACHINE LEARNING PIPELINE

Pada bab ini akan dijelaskan *Machine Learning Pipeline* yang akan digunakan dalam *sentiment analysis* - IMDB *Movie Reviews* (Klasifikasi).



Gambar 2. *Machine Learning Pipeline*

Berikut tahapan *Machine Learning Pipeline* yang akan digunakan sebagai berikut:

1. *Tokenizer*

   Tokenizer merupakan proses pemisahan teks menjadi kata, frasa, simbol atau elemen bermakna lainnya yang disebut dengan *token*. Tujuannya adalah mengeksplorasi kata-kata dalam sebuah kalimat. Daftar *token* menjadi masukan untuk diproses lebih lanjut. Tokenisasi adalah proses untuk memotong dokumen menjadi pecahan kecil yang dapat berupa bab, sub-bab, paragraf, kalimat, dan kata (*token*). Pada proses ini akan menghilangkan *whitespace* [2].

2. *Stop Words Removal*

   Banyak kata dalam dokumen yang sangat sering muncul namun pada dasarnya tidak memiliki arti karena digunakan untuk menggabungkan kata dalam kalimat. *Stop Word Removal* merupakan penghapusan kata-kata pada dokumen yang tidak memiliki arti tersebut [2].

3. *Hashing TF*

   Tahap *HashingTF* merupakan proses untuk melakukan *transformer* yang nantinya akan mengambil kata dan mengubah kata tersebut menjadi suatu vektor dengan panjang yang tepat.

4. *String Indexer*

   StringIndexer merupakan sebuah proses yang dilakukan untuk meng-*encode* kolom dengan label *string* menjadi label indeks. Indeks dibuat dalam bentuk [0, nomor label]

yang diurutkan berdasarkan frekuensi label, dimana indeks 0 menunjukkan bahwa label tersebut sering muncul.

5. *Naive Bayes Model*

   Naive Bayes model dibangun dengan menggunakan algoritma Naive Bayes. Naive Bayes merupakan algoritma yang digunakan untuk klasifikasi multi kelas. Naive Bayes menerapkan fungsi statistik sederhana berdasarkan teorema bayes dengan asumsi keberadaan dari suatu fitur tertentu terhadap suatu kelas yang tidak berhubungan dengan fitur lainnya [3].

   Naive Bayes classifier menggunakan *prior probability* (yaitu nilai probabilitas yang diyakini benar sebelum melakukan eksperimen) dari setiap label yang merupakan frekuensi masing-masing label pada *training set* dan kontribusi dari masing-masing fitur. Klasifikasi Naive Bayes dapat dilatih dengan sangat efisien dalam bentuk *supervised learning* [4]. Spark.mllib mendukung Naive Bayes Multinomial dan Naive Bayes Bernoulli. Model Naive Bayes ini digunakan untuk mengklasifikasikan dokumen.

V. IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai lingkungan implementas*i sentiment analysis* - IMDB *Movie Reviews* (Klasifikasi).

5.1 Implementasi *Start Spark Session*

Proses dalam pengerjaan proyek dengan memulai sesi dari spark itu sendiri yang digunakan untuk proses implementasi proyek. Langkah pertama yang dilakukan adalah dengan melakukan *import* pyspark (Apache Spark *deployment* untuk python) dan *library* lain yang dibutuhkan seperti numpy, pandas, matplotlib, seaborn, plotly, collections, *library* MultiClassMetrics, *library* MulticlassClassificationEvaluator

```
import pyspark
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from collections import Counter
from plotly import graph_objs as go
import plotly.express as px
import plotly.figure_factory as ff
from pyspark.sql.types import StringType
from pyspark.sql.functions import col, udf
from pyspark.ml.feature import Tokenizer, StringIndexer, Word2Vec,
StopWordsRemover, HashingTF
from pyspark.ml import Pipeline, Transformer
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.mllib.evaluation import MulticlassMetrics
```

Potongan Kode 1. *Library* yang digunakan

Setelah semua *library* yang dibutuhkan dalam pengerjaan proyek sudah di-*import* maka dapat melakukan *start* Spark Session dan juga menentukan nama aplikasi yang akan digunakan. Implementasi untuk memulai Spark *Session* dapat dilihat pada potongan program dibawah ini.

```
#memulai Spark Session
from pyspark.sql import SparkSession
spark = SparkSession.builder \
```

```
    .master("local[2]") \
    .appName("Proyek_Sentiment Analysis") \
    .getOrCreate()
```

Potongan Kode 2. Memulai Spark Session

5.2 Implementasi *Convert* dan *Load Dataset*

Dataset IMDB *Movie Review* yang sebelumnya masih dalam bentuk format *.csv* akan diubah kedalam format file Parquet. Pengimplementasian ini terdiri dari tiga tahapan yaitu melakukan *load dataset* melalui pandas, yang akan dilanjut dengan melakukan *load dataset* kedalam Spark dan yang terakhir adalah melakukan konversi *dataset* ke dalam format Parquet.

```
# melakukan load dataset menggunakan library pandas (dikarenakan Spark load data tidak sesuai)
df = pd.read_csv("D:/semester8/PDB/Proyek/IMDB Dataset.csv")

# melakukan load dataset kedalam Spark dan menampilkannya
reviews = spark.createDataFrame(df)
reviews.show(10)

# melakukan Konversi Data ke format Parquet
reviews.write.format("parquet").mode("overwrite").save("D:/semester8/PDB/Proyek/Data_Parquet")
```

Potongan Kode 3. *Convert* dan *Load Dataset*

Dataset yang sudah berhasil di konversi kedalam format Parquet akan dimuat dengan fungsi spark.read. Potongan kode program untuk melakukan *load* data parquet dapat dilihat dibawah ini.

```
data_parq=spark.read.format("parquet").load("D:/semester8/PDB/Proyek/Data_Parquet")
```

Potongan Kode 4. Fungsi spark.read

5.3 Implementasi Exploratory Data

Pada tahap ini akan dilakukan *exploratory data* untuk lebih memahami isi data yang akan digunakan. Adapun potongan program *code* yang dapat digunakan untuk menampilkan informasi

data pada tahapan implementasi sebagai berikut. Pada potongan kode program di bawah ini akan menampilkan deskripsi dari *dataset* yang dimiliki.

```
df.describe()
```

Potongan Kode 5. Menampilkan deskripsi dataset

Pada potongan kode program di bawah ini akan menampilkan visualisasi dari *dataset* yang dimiliki berupa grafik *bar chart* yang akan menampilkan jumlah *review* yang positif dan negatif.

```
plt.figure(figsize=(12,6))
sns.countplot(x='sentiment',data=df)
```

Potongan Kode 6. *Bar Chart*

Pada potongan kode program di bawah ini akan menampilkan visualisasi dari *dataset* yang dimiliki berupa grafik *funnel chart* yang akan menampilkan jumlah *review* yang positif dan negatif.

```
fig = go.Figure(go.Funnelarea(
    text =temp.sentiment,
    values = temp.review,
    title = {"position": "top center", "text": "Funnel-Chart of
Sentiment Distribution"}
    ))
fig.show()
```

Potongan Kode 7. *Funnel Chart*

Setelah data diubah ke format Parquet, maka akan dilakukan analisis kembali. Pada potongan kode program di bawah ini akan menampilkan 10 data teratas, menampilkan skema data dan menampilkan deskripsi *dataset* Parquet yang dimiliki.

```
data_parq.show(10)
data_parq.printSchema()
data_parq.describe().show()
```

Potongan Kode 8. EDA pada dataset parquet

Pada *exploratory* data dapat dilihat *review* yang bermakna positif dalam sebuah kalimat *review* yang sudah diberikan oleh penonton. Kode Program untuk melihat *review* yang bermakna positif dapat dilihat pada potongan kode program dibawah.

```
data_parq.where(fn.col('sentiment') == "positive").first()

# contoh dari kata yang bermakna positif
data_parq.where(fn.col('sentiment') == "positive").show(5)
```

Potongan Kode 9. Mencari *review* positif

Selain itu, dapat dilihat *review* yang bermakna negatif dalam sebuah kalimat *review* yang sudah diberikan oleh penonton. Kode Program untuk melihat *review* yang bermakna negatif dapat dilihat pada potongan kode program dibawah.

```
data_parq.where(fn.col('sentiment') == "negative").first()

# contoh dari kata yang bermakna negatif
data_parq.where(fn.col('sentiment') == "negative").show(5)
```

Potongan Kode 10. Mencari *review* negatif

Sentimen yang terdapat pada setiap kategori dalam data parquet juga dapat dihitung jumlahnya dengan menggunakan *count()*.

```
data_parq.groupBy('sentiment').agg(fn.count('*')).show()
```

Potongan Kode 11. Menghitung jumlah *sentiment* dalam data parquet

5.4 Implementasi *Data Cleaning*

Pada tahap ini akan dilakukan pengecekan pada data yang memiliki nilai *null* yang nantinya akan dihapus dari *dataframe*. Kode program untuk melakukan *data cleaning* dapat dilihat pada potongan kode program berikut

```
# melakukan pengecekan dan menghapus nilai null
from pyspark.sql.functions import count

def my_count(df_in):
    df_in.agg( *[ count(c).alias(c) for c in df_in.columns ] ).show()
```

Potongan Kode 12. *Data Cleaning*

5.5 Implementasi *Text Preprocessing*

Tahap *text preprocessing* yang merupakan proses pengubahan bentuk data menjadi data yang terstruktur agar sesuai dengan kebutuhan untuk proses sentimen analisis.

1. **Tokenisasi dan Text Vektor**

Pada tahap ini akan dilakukan pemisahan setiap kata yang terdapat dalam satu kalimat *review*, agar setiap *string* teks dapat dipahami oleh mesin. Kode program implementasi tokenisasi dan teks vektor dapat dilihat sebagai berikut

```
tokenizer = Tokenizer(inputCol="review", outputCol="tokens")
output_tokenizer = tokenizer.transform(data_parq)
output_tokenizer.show()
```

Potongan Kode 13. Tokenisasi dan Text Vektor

2. ***Stop Words Removal***

Tahap ini akan menyaring kata-kata menggunakan konstruktor *StopWordsRemover()*. Tahap ini akan menyaring kata-kata yang sering maupun jarang muncul agar Kinerja klasifikasi akan menjadi lebih optimal dengan menghapus kata-kata yang jarang muncul. Kode program implementasi *stop words removal* dapat dilihat sebagai berikut.

```
WordsRemover = StopWordsRemover(inputCol= 'tokens', outputCol=
'filtered_words')
output_WordsRemover = WordsRemover.transform(output_tokenizer)
output_WordsRemover.show()
```

Potongan Kode 14. *Stop Words Removal*

3. ***Hashing TF***

Tahap *HashingTF* merupakan proses untuk melakukan *transformer* yang nantinya akan mengambil kata dan mengubah kata tersebut menjadi suatu vektor dengan panjang yang tepat. Implementasi dari *HashingTF* dapat dilihat pada potongan kode program dibawah ini.

```
word_hash=HashingTF(inputCol="filtered_words",
outputCol="features")
output_word_hash = word_hash.transform(output_WordsRemover)
output_word_hash.show()
```

Potongan Kode 15. *Hashing TF*

5.6 Implementasi Klasifikasi *Sentiment*

Implementasi dilakukan dengan menggunakan *string indexing* untuk memetakan kolom *string* pada label ke kolom ML dari indeks label, yang bertujuan untuk mengubah *sentiment* menjadi bernilai *integer*.

```
sentiment_indexer = StringIndexer(inputCol="sentiment",
outputCol="label")
```

Potongan Kode 16. *String Indexer*

Kemudian menggunakan salah satu algoritma *supervised learning* yaitu Naive Bayes untuk melakukan klasifikasi prediksi terhadap sentimen analisis.

```
from pyspark.ml.classification import NaiveBayes
# Gunakan Algoritma Naive Bayes untuk memprediksi sentiment
naive_bayes = NaiveBayes(featuresCol="features", labelCol="label")
```

Potongan Kode 17. Klasifikasi *Sentiment* dengan Naive Bayes

Data parquet yang digunakan akan dibagi menjadi *data testing* dan *data training*, pada implementasi ini *data training* yang digunakan sebesar 80% dan *data testing* yang digunakan 20%. Pemecahan ini dilakukan secara rata pada tiap labelnya.

```
# Lakukan pembagian data menjadi data training dan data testing
train_set, test_set = data_parq.randomSplit([0.8, 0.2], 3)
```

Potongan Kode 18. Pembagian *data training* dengan *data testing*

Berikut merupakan implementasi pembuatan *pipeline*. *Pipeline* yang mampu menggabungkan beberapa *transformer* dan *estimator* untuk menentukan *workflow* dari sebuah ML.

```
# Membangun Pipeline
pipeline = Pipeline(stages=[tokenizer, WordsRemover, word_hash,
sentiment_indexer, naive_bayes])
```

Potongan Kode 19. Pembuatan *Pipeline*

Pipeline merupakan *estimator* yang terdiri atas beberapa *transformer*. Setelah *pipeline* didefinisikan maka *pipeline* akan disesuaikan dengan *data_train* untuk melatih model yang telah dibuat sebelumnya.

```
# Lakukan Fit pada model
```

```
model = pipeline.fit(train_set)
```

Potongan Kode 20. Fit pada model

Setelah model dilatih menggunakan *data_train*, maka model akan ditransformasikan dengan menggunakan data uji yaitu *test_set* untuk menghasilkan sebuah prediksi dari *review movie*.

```
# Membuat Prediksi
predictions = model.transform(test_set).select(col("label"),
col("prediction"))
predictions.show()
```

Potongan Kode 21. Membuat Prediksi

5.7 Implementasi Evaluasi Sentimen Analisis

Tahap ini dilakukan untuk melakukan pengujian untuk menilai seberapa baik proses klasifikasi pada sentimen analisis *review* film. Implementasi *evaluator* yang digunakan adalah *MulticlassClassificationEvaluator().* Potongan kode program evaluasi klasifikasi sentimen analisis dapat dilihat pada kode program dibawah ini.

```
# Gunakan Evaluator untuk mengukur performa dari model yang dibangun
evaluator = MulticlassClassificationEvaluator(labelCol="label",
predictionCol="prediction", metricName="accuracy")
evaluator.evaluate(predictions)
```

Potongan Kode 22. Evaluasi Sentimen Analisis

5.8 Impmenetasi Visualisasi

Pada tahap ini, akan dilakukan visualisasi sentiment analisis untuk menunjukkan kata-kata yang paling sering digunakan dalam menyampaikan pendapatnya. Implementasi visualisasi menggunakan library *WorldCloud().* Potongan kode program untuk *import* dan *setting wordcloud* dapat dilihat pada potongan kode berikut.

```
from wordcloud import WordCloud
def plot_wordCloud(words):
    wordCloud        =        WordCloud(width=800,        height=500,
background_color='white',                                random_state=21,
max_font_size=120).generate(words)

    plt.figure(figsize=(10, 7))
    plt.imshow(wordCloud, interpolation='bilinear')
```

```
    plt.axis('off')
```

Untuk membuat visualisasi sentiment positif, dilakukan dengan cara melakukan join antara kolom sentiment dan kategori sentiment. Setelah kolom dilakukan join, selanjutnya akan di select kategori positif.

```
#untuk melihat kata kata positif
normal_words = ' '.join(text for text in data_review['text'][data_review['Sentimen'] == 'Positive'])
plot_wordCloud(normal_words)
```

VI. HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil dari implementasi yang sudah dilakukan.

6.1 Hasil Implementasi *Exploratory Data*

Implementasi *exploratory data* untuk mengetahui karakteristik dari data yang akan digunakan, seperti nama dan tipe data dari setiap kolom yang tersedia pada data. Selain itu dalam implementasi *exploratory* data dapat dilihat ringkasan mengenai jumlah *review* dan sentimen yang terdapat pada data tersebut, nilai *mean*, standar deviasi, *minimum* dan *maximum*. Berikut ini adalah hasil dari *load data* yang digunakan, pada hasil tersebut menampilkan atribut *review* dan *sentiment* sebanyak 10 data.

```
+--------------------+---------+
|              review|sentiment|
+--------------------+---------+
|Jess Franco makes...| negative|
|Really enjoyed th...| positive|
|I read several mi...| positive|
|Absolutely one of...| positive|
|True stories make...| positive|
|I saw The Big Bad...| positive|
|Actually I'm stil...| negative|
|Let me say first ...| positive|
|I write this revi...| positive|
|The odd mixture o...| negative|
+--------------------+---------+
only showing top 10 rows
```

Gambar 3. *Load Data Parquet*

Berikut ini adalah hasil untuk menampilkan atribut dan tipe data atribut yang terdapat pada *dataset*. Pada *dataset* memiliki atribut *review* dengan tipe *data string* dan atribut *sentiment* dengan tipe data *string*.

```
root
 |-- review: string (nullable = true)
 |-- sentiment: string (nullable = true)
```

Gambar 4. Atribut dan tipe data atribut

Pada hasil berikut ini akan ditampilkan hasil dari *summary* deskripsi dari *dataset* yang digunakan.

```
+-------+--------------------+---------+
|summary|              review|sentiment|
+-------+--------------------+---------+
|  count|               50000|    50000|
|   mean|                null|     null|
| stddev|                null|     null|
|    miA Turkish Bat...| negative|
|    max|ý thýnk uzak ýs t...| positive|
+-------+--------------------+---------+
```

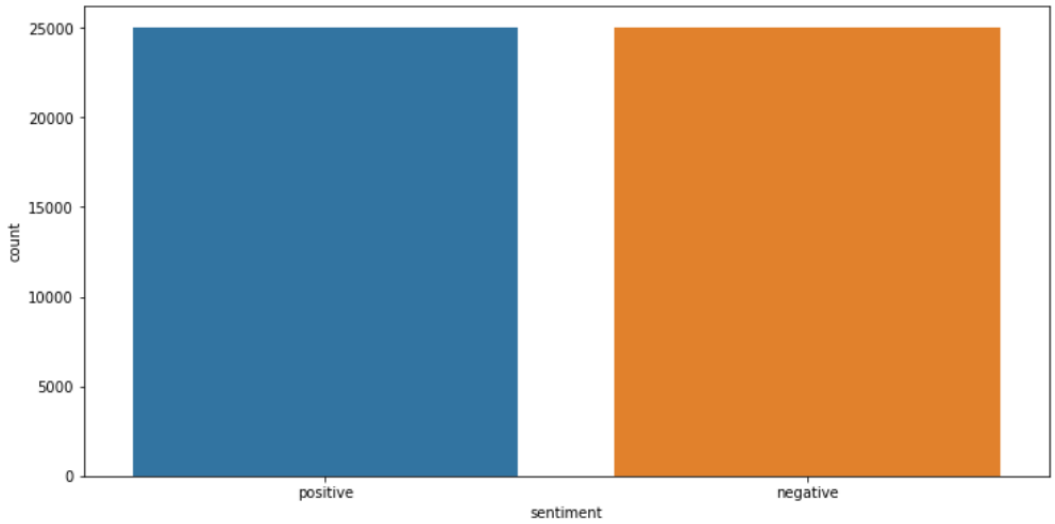
Gambar 5. Deskripsi dari data yang digunakan

Berikut ini adalah hasil dari analisis jumlah data yang memiliki sentimen positif dan negatif.

	sentiment	review
0	negative	25000
1	positive	25000

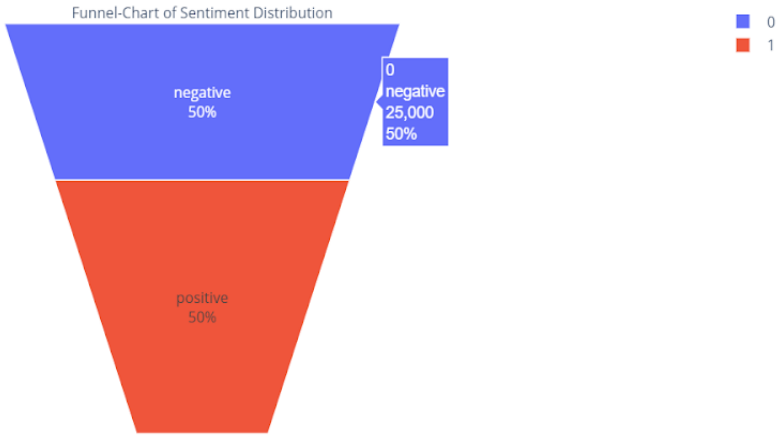
Gambar 6. Jumlah data sentiment positif dan negatif

Berikut ini adalah hasil dari menampilkan visualisasi *dataset* dalam bentuk *bar chart.*



Gambar 7. *Bar chart* yang menunjukkan jumlah data sentiment positif dan negatif

Berikut ini hasil dari visualisasi *dataset* dalam bentuk *funnel chart.*



Gambar 8. *Funnel chart* yang menunjukkan jumlah data sentiment positif dan negative

Hasil implementasi dari *review* positif yang diberikan oleh pengguna dapat dilihat seperti berikut

```
Row(review="Really enjoyed this little movie. It's a moving film about struggle, sacrifice and especially the bonds of friendsh
ip between different peoples (the child actor who plays Miki is especially good). There's so many large scale impersonal films
set around WW2, that this convincingly told little story is a real break from the norm, and an original one at that. I'll also
add that this film is far from boring, very far!! Of course the Horses are wonderful and the scenery breathtaking. To anyone wh
o really treats their animal as part of the family (I do), you'll find this film especially rewarding. Recommended to movie fan
s who look for something a little different.", sentiment='positive')
```

Gambar 9. Hasil implementasi *review* positif

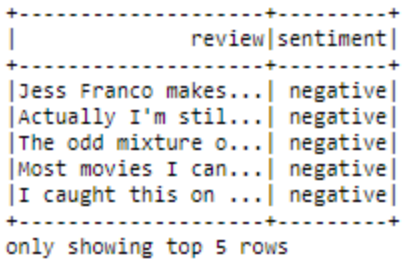
```
+--------------------+---------+
|              review|sentiment|
+--------------------+---------+
|Really enjoyed th...| positive|
|I read several mi...| positive|
|Absolutely one of...| positive|
|True stories make...| positive|
|I saw The Big Bad...| positive|
+--------------------+---------+
only showing top 5 rows
```

Gambar 10. Contoh dari kata yang bermakna positif

Hasil implementasi dari *review* negatif yang diberikan oleh pengguna dapat dilihat seperti berikut

```
Row(review='Jess Franco makes exploitation films, and he has made tons of them. Franco is responsible for some of the most shoc
king films in cinema history, and god bless him for it. Unfortunately, The Diamonds of Kilominjaro is a truly awful movie that
is not up to his usual standards.<br /><br />Exploitation films should be judged on story, sex, and gore. What else is there? T
his film fails on most of those benchmarks. The plot is paper thin, placing a nubile young girl in the jungle among cannibals.
We really don\'t get information on why she and her father were there in the first place. As expected, her father is the "Big W
hite Chief" and she becomes a goddess, sitting in trees, naked. Add fortune hunters and precious stones, and you have your basi
c rescue the girl for greedy intentions plot line. The characters are stock, not adding an ounce of believability to the procee
dings.<br /><br />Gore? None, or at least very little. This film is often mentioned in the same vein as the classic Italian can
nibal movies. Those seeking that type of gore need to run the other way. Save for one cheap be-heading, this movie features sur
prisingly little blood and guts.<br /><br />As best I can tell the only reason this movie exists is so Katja Bienert, Aliene Me
ss, and Mari Carmen Neieto could run around naked. Actually "Lita" (Mari Carmen Neieto) does the full frontal heavy lifting, wh
ile the two jungle ladies are bare chested throughout. Yes, there are love scenes....probably the most sterile Franco has ever
supervised. The women are beautiful, but nothing here to really make this movie an erotic classic either.<br /><br />This movie
just reeks of low budget buffoonery. The sets are laughable. The acting is horrid, and the editing is confusing. There is no re
al story to hold this together, and not enough of a budget (or effort) to shock or titillate. I think Franco fans have come to
expect more out of the master of exploitation.', sentiment='negative')
```

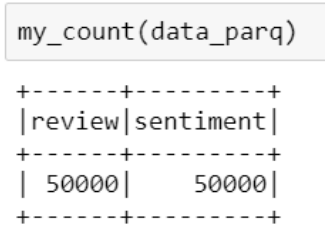
Gambar 11. Hasil implementasi *review* negatif



Gambar 12. Contoh dari kata yang bermakna negative

6.2 Hasil Implementasi Data Cleaning

Berikut ini adalah hasil implementasi dari data cleaning. Pada kode program untuk melihat data apakah terdapat null atau tidak yang sudah disertakan pada Bab 5.4, hasil dari kode tersebut menunjukkan bahwa tidak terdapat data null pada dataset yang digunakan, sehingga tidak perlu untuk melakukan pengurangan data null pada data yang digunakan tersebut. Berikut adalah hasil dari kode program yang menunjukkan bahwa tidak terdapat data null pada dataset.

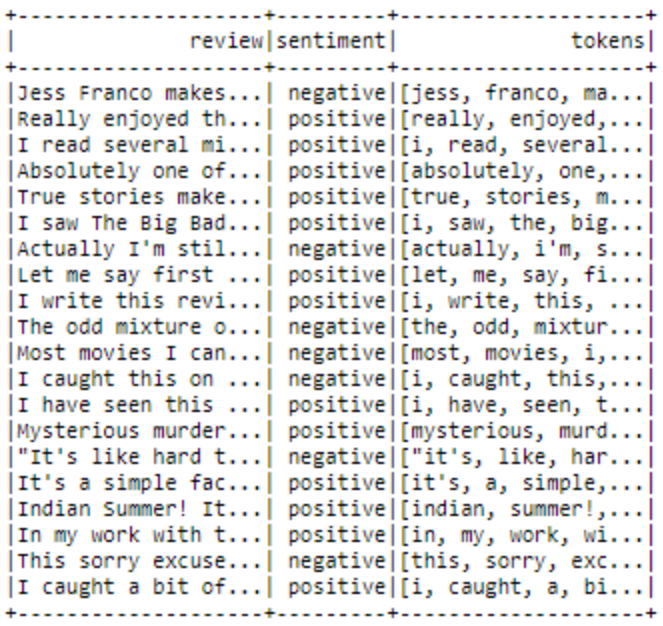


Gambar 13. Tidak terdapat data *null* pada dataset

6.3 Hasil Implementasi Text Preprocessing

1. Hasil Implementasi Tokenisasi

   Proses tokenisasi yang akan digunakan untuk melakukan pemisahan kata yang terdapat pada setiap kalimat *review*. Sehingga akan diperoleh sebuah kolom dengan nama *tokens* yang berisi hasil dari pemisahan setiap kata yang terdapat pada *review* film.

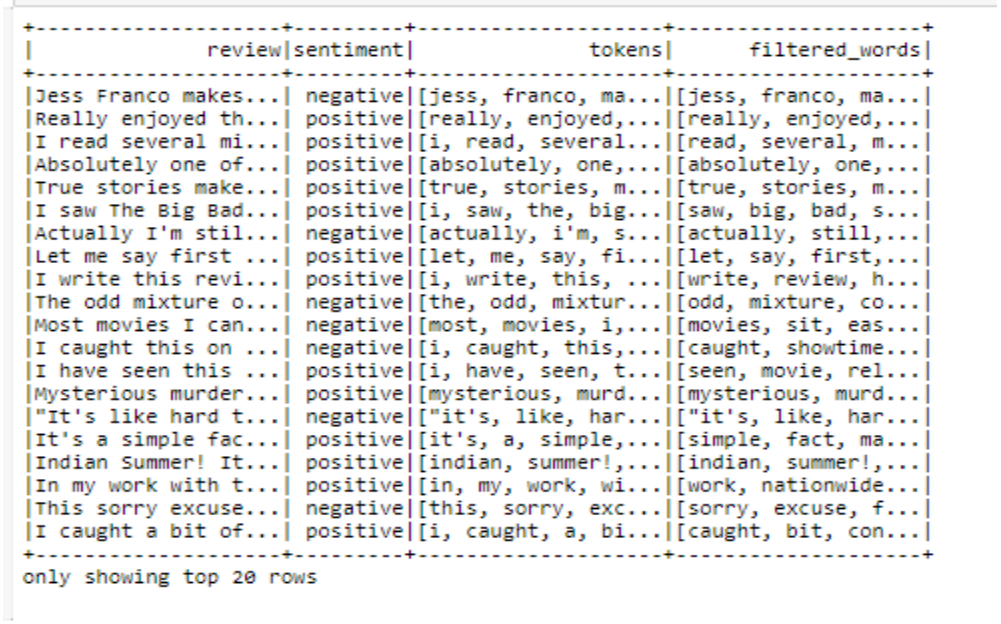


review	sentiment	tokens
Jess Franco makes...	negative	[jess, franco, ma...
Really enjoyed th...	positive	[really, enjoyed,...
I read several mi...	positive	[i, read, several...
Absolutely one of...	positive	[absolutely, one,...
True stories make...	positive	[true, stories, m...
I saw The Big Bad...	positive	[i, saw, the, big...
Actually I'm stil...	negative	[actually, i'm, s...
Let me say first ...	positive	[let, me, say, fi...
I write this revi...	positive	[i, write, this, ...
The odd mixture o...	negative	[the, odd, mixtur...
Most movies I can...	negative	[most, movies, i,...
I caught this on ...	negative	[i, caught, this,...
I have seen this ...	positive	[i, have, seen, t...
Mysterious murder...	positive	[mysterious, murd...
"It's like hard t...	negative	["it's, like, har...
It's a simple fac...	positive	[it's, a, simple,...
Indian Summer! It...	positive	[indian, summer!,...
In my work with t...	positive	[in, my, work, wi...
This sorry excuse...	negative	[this, sorry, exc...
I caught a bit of...	positive	[i, caught, a, bi...

Gambar 14. Hasil Implementasi Tokenisasi

2. Hasil Implementasi *Stop Words Removal*

   Proses *Stop Words Removal* yang dilakukan untuk menyaring kata-kata yang jarang muncul atau tidak terlalu memiliki makna, maka akan dihasilkan kolom filtered_words yang berisi hasil dari *stop words removal*. Disini dapat dilihat bahwa kata seperti *i, this, it's* sudah tidak terdapat lagi pada kolom *filtered_words.*



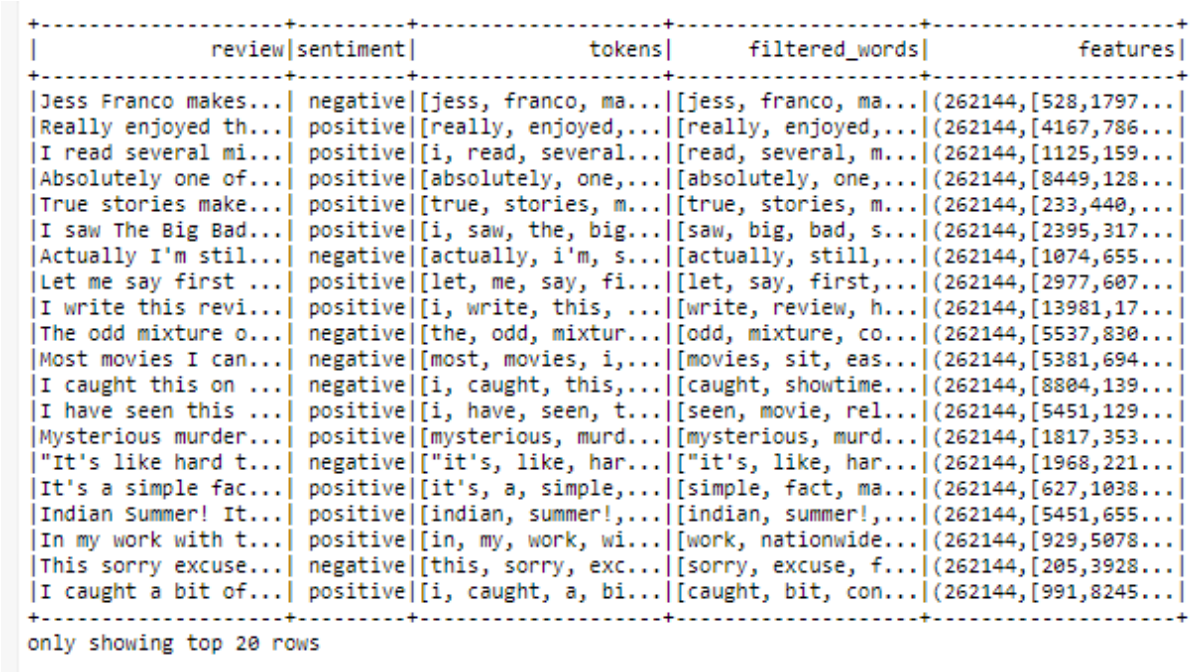
review	sentiment	tokens	filtered_words
Jess Franco makes...	negative	[jess, franco, ma...	[jess, franco, ma...
Really enjoyed th...	positive	[really, enjoyed,...	[really, enjoyed,...
I read several mi...	positive	[i, read, several...	[read, several, m...
Absolutely one of...	positive	[absolutely, one,...	[absolutely, one,...
True stories make...	positive	[true, stories, m...	[true, stories, m...
I saw The Big Bad...	positive	[i, saw, the, big...	[saw, big, bad, s...
Actually I'm stil...	negative	[actually, i'm, s...	[actually, still,...
Let me say first ...	positive	[let, me, say, fi...	[let, say, first,...
I write this revi...	positive	[i, write, this, ...	[write, review, h...
The odd mixture o...	negative	[the, odd, mixtur...	[odd, mixture, co...
Most movies I can...	negative	[most, movies, i,...	[movies, sit, eas...
I caught this on ...	negative	[i, caught, this,...	[caught, showtime...
I have seen this ...	positive	[i, have, seen, t...	[seen, movie, rel...
Mysterious murder...	positive	[mysterious, murd...	[mysterious, murd...
"It's like hard t...	negative	["it's, like, har...	["it's, like, har...
It's a simple fac...	positive	[it's, a, simple,...	[simple, fact, ma...
Indian Summer! It...	positive	[indian, summer!,...	[indian, summer!,...
In my work with t...	positive	[in, my, work, wi...	[work, nationwide...
This sorry excuse...	negative	[this, sorry, exc...	[sorry, excuse, f...
I caught a bit of...	positive	[i, caught, a, bi...	[caught, bit, con...

only showing top 20 rows

Gambar 15. Hasil Implementasi *Stop Words Removal*

3. Hasil Implementasi *HasingTF*

   Proses *HashingTF* yang sudah berhasil diimplementasikan akan menghasilkan suatu vektor dengan panjang yang sesuai dari setiap kata yang tersedia. Hasil dari *HashingTF* dapat dilihat pada kolom *features.*



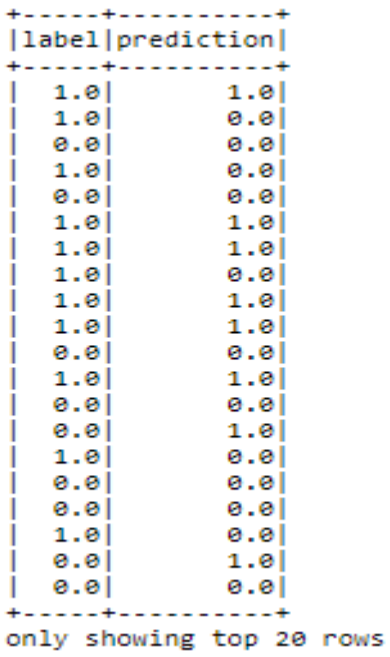
review	sentiment	tokens	filtered_words	features
Jess Franco makes...	negative	[jess, franco, ma...	[jess, franco, ma...	(262144,[528,1797...
Really enjoyed th...	positive	[really, enjoyed,...	[really, enjoyed,...	(262144,[4167,786...
I read several mi...	positive	[i, read, several...	[read, several, m...	(262144,[1125,159...
Absolutely one of...	positive	[absolutely, one,...	[absolutely, one,...	(262144,[8449,128...
True stories make...	positive	[true, stories, m...	[true, stories, m...	(262144,[233,440,...
I saw The Big Bad...	positive	[i, saw, the, big...	[saw, big, bad, s...	(262144,[2395,317...
Actually I'm stil...	negative	[actually, i'm, s...	[actually, still,...	(262144,[1074,655...
Let me say first ...	positive	[let, me, say, fi...	[let, say, first,...	(262144,[2977,607...
I write this revi...	positive	[i, write, this, ...	[write, review, h...	(262144,[13981,17...
The odd mixture o...	negative	[the, odd, mixtur...	[odd, mixture, co...	(262144,[5537,830...
Most movies I can...	negative	[most, movies, i,...	[movies, sit, eas...	(262144,[5381,694...
I caught this on ...	negative	[i, caught, this,...	[caught, showtime...	(262144,[8804,139...
I have seen this ...	positive	[i, have, seen, t...	[seen, movie, rel...	(262144,[5451,129...
Mysterious murder...	positive	[mysterious, murd...	[mysterious, murd...	(262144,[1817,353...
"It's like hard t...	negative	["it's, like, har...	["it's, like, har...	(262144,[1968,221...
It's a simple fac...	positive	[it's, a, simple,...	[simple, fact, ma...	(262144,[627,1038...
Indian Summer! It...	positive	[indian, summer!,...	[indian, summer!,...	(262144,[5451,655...
In my work with t...	positive	[in, my, work, wi...	[work, nationwide...	(262144,[929,5078...
This sorry excuse...	negative	[this, sorry, exc...	[sorry, excuse, f...	(262144,[205,3928...
I caught a bit of...	positive	[i, caught, a, bi...	[caught, bit, con...	(262144,[991,8245...

only showing top 20 rows

Gambar 16. Hasil Implementasi *HasingTF*

6.4 Hasil Implementasi Sentimen Analisis

Proses pengklasfikisian sentimen analisis yang sudah dimplementasikan dengan algoritma Naive Bayes nantinya akan menghasilkan kolom label dan kolom prediksi, dimana label merupakan *class* yang sesungguhnya sedangkan pada kolom prediksi merupakan hasil prediksi algoritma itu sendiri. Apabila hasil label dan prediksi memberikan nilai yang sama baik itu bernilai 1.0 atau 0.0 maka prediksi yang dihasilkan sudah tepat sesuai *class* label.



```
+-----+----------+
|label|prediction|
+-----+----------+
|  1.0|       1.0|
|  1.0|       0.0|
|  0.0|       0.0|
|  1.0|       0.0|
|  0.0|       0.0|
|  1.0|       1.0|
|  1.0|       1.0|
|  1.0|       0.0|
|  1.0|       1.0|
|  1.0|       1.0|
|  0.0|       0.0|
|  1.0|       1.0|
|  0.0|       0.0|
|  0.0|       1.0|
|  1.0|       0.0|
|  0.0|       0.0|
|  0.0|       0.0|
|  1.0|       0.0|
|  0.0|       1.0|
|  0.0|       0.0|
+-----+----------+
only showing top 20 rows
```

Gambar 17. Hasil Implementasi Sentimen Analisis

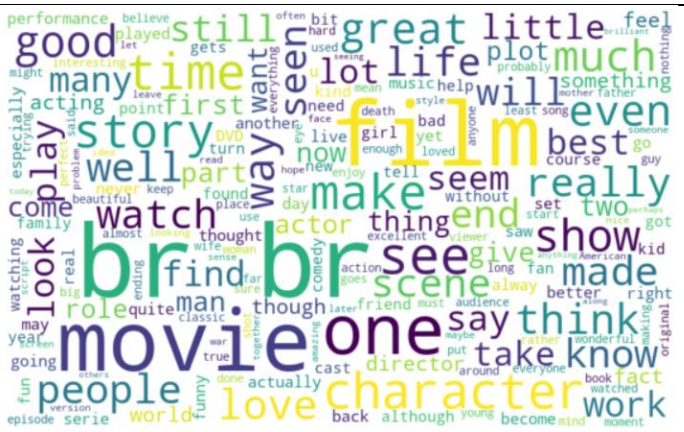
6.5 Hasil Implementasi Evaluasi

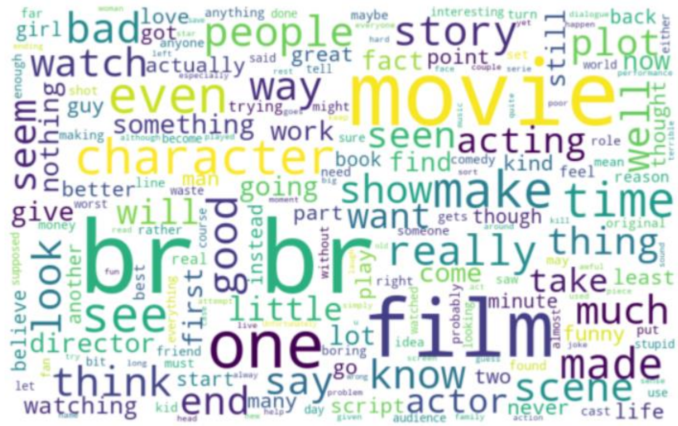
Berdasarkan hasil pengujian menggunakan model *Naive Bayes* pada *data training* dan *data testing* yang diuji dengan rasio 80:20. Dapat dilihat bahwa performa yang dihasilkan dari model yang dibangun adalah **0.8617343844754397.**

6.6 Hasil Implementasi Visualisasi

Berdasarkan hasil implementasi visualisasi sentiment analisis menggunakan WordCloud, didapatkan 3 visualisasi, yakni sentiment netral, positif dan negative. Berikut adalah hasil visualisasi yang telah dilakukan.

Table 1 Visualisasi Sentimen Analisis

No	Kategori Sentimen	Hasil Visualisasi
1	Netral	
2	Positif	

3	Negatif	

movie
br
film
one
character
even
make
time
people
story
plot
bad
watch
way
show
good
really
see
look
acting
well
seen
seem
thing
made
scene
know
actor
think
say
end
much
take
little
first
will
want
find
something
work
going
director
give
nothing
still
now
fact
point
great
love
got
girl
actually
guy
trying
better
instead
another
believe
lot
come
minute
least
funny
thought
back
book
kind
feel
part
though
watching
script
never
life
two
start
many
go
cast
reason
role
play
man
real
best
right
anything
interesting
maybe
far
done
world
turn
anyone
said
tell
might
making
become
line
waste
worst
money
rather
someone
course
without
friend
must
boring
idea
probably
almost
stupid
put
use
audience
kid
day
sure
need
comedy
mean
poor
shot
enough
everything
supposed
original
found
let
bit
long
head
given
action
family
problem

VII. KESIMPULAN

Pada industri hiburan, film menjadi salah satu karya yang banyak diminati sekaligus diulas oleh masyarakat. Respon/ulasan masyarakat terhadap sebuah film menjadi penentu apakah film tersebut dapat dikategorikan bagus atau tidak. Akibat kemudahan pemberian ulasan oleh penikmat film dan semakin banyaknya film yang beredar di masyarakat, muncullah permasalahan terkait sulitnya menentukan tanggapan penonton termasuk pada respon positif atau negatif dengan data yang sangat banyak. Untuk mempermudah menentukan *review movie* yang bagus dan yang buruk maka perlu untuk mengklasifikasikan teks ulasan penonton (sentimen) yang mungkin berdasarkan *review* yang sudah diberikan oleh penonton sebelumnya. Setiap *review* akan diproses sehingga menghasilkan klasifikasi sentimen yang positif dan negatif.

Pada proyek sentimen analisis *dataset movie* yang telah dikerjakan menggunakan algoritma Naïve Bayes menggunakan library spark.mllib. Dataset yang semula adalah format .csv akan diubah ke format .parquet hal ini agar mendukung pemrosesan data secara terdistribusi. Data yang digunakan terlebih dahulu dilakukan *text preprocessing* agar menghasilkan data yang lebih baik sebelum diklasifikasikan. Data yang sudah di-*preprocessing* selanjutnya akan diklasifikasikan menjadi *sentiment* yang positif dan negatif.

Selanjutnya data akan dibagi menjadi data *train* dan data *test*. Berdasarkan hasil pengujian menggunakan model Naive Bayes pada data *train* dan data *test* yang diuji dengan rasio 80:20. Dapat dilihat bahwa performa yang dihasilkan dari model yang dibangun untuk mengklasfikasikan *review* positif dan negatif adalah 0.8617343844754397 atau akurasi sebesar 86%.

DAFTAR PUSTAKA

[1] B., Effendi, S., & Sitompul, O. S Kurniawan, "KLasifikasi Konten Berita dengan Metode Text Mining," *JURNAL DUNIA TEKNOLOGI INFORMASI*, vol. 1, no. 1, pp. 14-19, 2012.

[2] S. K. Vairaprakash Gurusamy, "Preprocessing Techniques for Text Mining," 2015.

[3] Retno Sari and Ratih Yulia Hayuningtyas, "Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Pada Wisata TMII Berbasis Website," *IJSE – Indonesian Journal on Software Engineering*, vol. 2, no. 5, Desember 2015.

[4] "Implementasi Algoritma Naive Bayes Terhadap Analisis Sentimen Opini Film Pada Twitter," *JURNAL INOVTEK POLBENG - SERI INFORMATIKA*, vol. 3, no. 1, Juni 2018.