

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Pada hasil penelitian ini akan menjelaskan hasil dari pengambilan data dari komentar masyarakat dalam aplikasi mobile legend di dalam aplikasi google playstore dan menghasilkan sebuah data yang akan dilakukan pemrosesan seperti preprocessing, labeling kemudian pengolahan sentiment menggunakan algoritma Support Vector Machine dan juga Adaboost untuk mendapatkan nilai akurasi pada pada sentiment masyarakat mengenai game mobile legend yang akan di implementasikan menggunakan tool RapidMiner.

4.1.1. Pengumpulan Data

Dalam pengumpulan data dalam penelitian ini di ambil langsung dari aplikasi mobile legendnya yang terdapat dalam play store. Proses pengambilan dataset menggunakan metode scraping data. Dimana pada scraping data ini akan menggunakan tool Python untuk mendapatkan data tentang komentar masyarakat di dalam aplikasi mobile legend pada Google Play store. Berikut adalah tahapan dan pngambilan hasil scraping data menggunakan python.

a. Instalasi dan import Library

Pada tahapan ini melakukan penginstalan pada library yang akan digunakan dalam yaitu library google Scraper, library pandas dan library numpy. Berikut adalah hasil dari proses penginstalan dan library yang digunakan dalam pemrosesan.

```
pip install google-play-scraper  
Collecting google-play-scraper  
  Downloading google_play_scraper-1.2.4-py3-none-any.whl (28 kB)  
Installing collected packages: google-play-scraper  
Successfully installed google-play-scraper-1.2.4  
  
#Import Library Python yang digunakan  
from google_play_scraper import app  
import pandas as pd  
import numpy as np
```

Gambar 4. 1 Library yang digunakan dalam pemrosesan Scraping Data

b. Penginputan Variable dan Proses pengambilan Data

Proses yang selanjutnya adalah penginputan variable yang dimana pada proses ini juga digunakan untuk melakukan scraping data sesuai dengan variable dan data yang kita masukan untuk mendapatkan data dalam google play store. Proses pengambilan data meliputi token yang kita gunakan yang mengarah pada aplikasi yang kita tuju. Dalam penelitian ini kita akan mengambil data komentar dari aplikasi Mobile Legend dimana token yang digunakan berasal dari url aplikasi seperti “*com.mobile.legends*” kemudian terdapat bahasa apa yang akan kita ambil dan juga dari server negara mana komentar yang akan kita pilih dan beberapa variable yang lain. Berikut adalah program untuk proses crawling data aplikasi mobile legend.

```
#Melakukan Import variable dalam library yang akan digunakan
from google_play_scraper import Sort, reviews

#Pemrosesan pengambilan data pada Google Play Store
result, continuation_token = reviews(
    'com.mobile.legends',
    #Bahasa komentar yang akan diambil yaitu indonesia
    lang='id',
    #Komentar yang dilakukan di ambil di negara apa
    country='id',
    # menggunakan most_relevant untuk mendapatkan ulasan yang paling relevant
    sort=Sort.MOST_RELEVANT,
    # Jumlah Data yang akan diambil
    count=10000,
    # menentukan rating komentar yang di ambil (rating bintang 1 sampai 5 atau None = ALL
    filter_score_with=None
)
```

Gambar 4. 2 Program Proses Scraping Data

c. Data Frame

Data frame adalah jenis struktur data yang dirancang untuk representasi table, yang terdiri dari banyak kolom dengan tiap kolom berisi list ataupun vector dengan jumlah data yang sama. Dalam pemrosesan ditampilkan hasil dari proses scraping dalam data frame dimana proses scraping sebenarnya mengambil semua data pada setiap komentar seperti username, content, user image, waktu, app version dan lain-lain yang sebenarnya tidak semua data itu kita butuhkan. Berikut adalah hasil dari tampilan data frame dari hasil scraping data pada aplikasi mobile legend.

	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt	appVersion
0	1187a6c6-049d-47b5-9649-0ecabdfaf8b	Galileo Raga Putra	lh.googleusercontent.com/a/ACg8oc...	https://play- Untuk developer tolong di perbaiki sistem matc...	1	255	1.8.45.9182	2023-12-20 04:53:44	None	NaT	1.8.45.9182
1	acc2df8a-7dc0-4094-ad59-5abb4a572e10	Arfan Alkafa	lh.googleusercontent.com/a-/ALV-U...	https://play- Gak ada bintang, sistemnya bobrok, kena dark s...	1	767	1.8.34.9055	2023-12-17 07:13:21	None	NaT	1.8.34.9055
2	5d87a75c-77ff-4ef5-aff1-ac0ab1bfc8d3	kariadi orbit	lh.googleusercontent.com/a-/ALV-U...	https://play- Makin kesini makin ndak menurun kualitasnya. J...	1	3409	1.8.34.9055	2023-12-16 08:12:06	Dear Hero, Kami bertekad untuk menciptakan lin...	2023-12-17 08:10:01	1.8.34.9055
3	92402098-5584-4e0d-b01d-d83eb1154b57	Amelia Azzahra	lh.googleusercontent.com/a/ACg8oc...	https://play- Kok aku baru mau masuk udah eror saja padahal j...	1	301	1.8.34.9055	2023-12-16 08:16:22	None	NaT	1.8.34.9055
4	ac5993c8-0c54-4692-aeb0-d5071c5bac35	Efa Alfa	lh.googleusercontent.com/a/ACg8oc...	https://play- Sering lag tiba- tiba, padahal wifi atau data L...	1	4	1.8.34.9055	2023-12-20 13:24:48	None	NaT	1.8.34.9055

Gambar 4. 3 Data Frame Hasil Scraping

d. Menghitung Data yang di dapatkan

Setelah melihat data frame dari informasi komentar yang di dapat selanjutnya adalah menghitung kembali jumlah data yang didapatkan dari hasil scraping data untuk memastikan bahwa data yang di ambil jumlahnya sesuai dengan yang kita butuhkan. Dalam scraping data kali ini peneliti mengambil 10.000 data komentar yang terdapat dalam aplikasi mobile legend. Berikut adalah hasil tampilan dari pemrosesannya.

```
len(df_busu.index)

10000
```

Gambar 4. 4 Jumlah data yang berhasil di scraping

e. Pengelompokan data berdasarkan kolom

Pada pengelompokan ini melakukan pengelompokan data sesuai dengan kolom yang kita butuhkan seperti username, score/rating, waktu dan Content. Berikut adalah hasil dari pengelompokan berdasarkan kolom yang dibutuhkan.

```
df_busu[['userName', 'score', 'at', 'content']].head() #preview userName, rating, date-time,
```

	userName	score	at	content
0	Kenzo Gaming	3	2023-12-26 15:31:12	Untuk moonton, dari gameplay ml udah bagus , d...
1	Giemael Aditiya	2	2024-01-24 13:28:40	Dari segala aspek seperti grafik,suara,musik,l...
2	Budi Suwarno	5	2024-01-16 16:44:25	Saya suka dengan game ini,tapi ada masalah yan...
3	Wafa Bf	4	2024-01-23 13:37:36	Gamenya keren, dari segi grafik juga bagus, da...
4	Muhyid 224	2	2024-01-15 14:30:55	Game nya sudah bagus... Akan tetapi sayang di ...

Gambar 4. 5 Pengelompokan Kolom

f. Penyortiran Kolom

Jika kita melihat kembali pada gambar 4.5 dapat terlihat bahwa waktu dalam proses scraping tidak berurutan. Dalam proses penyortiran ini akan mengurutkan komentar berdasarkan kometar terbaru dari proses scraping data. berikut adalah hasil dari pemrosesan penyortiran data.

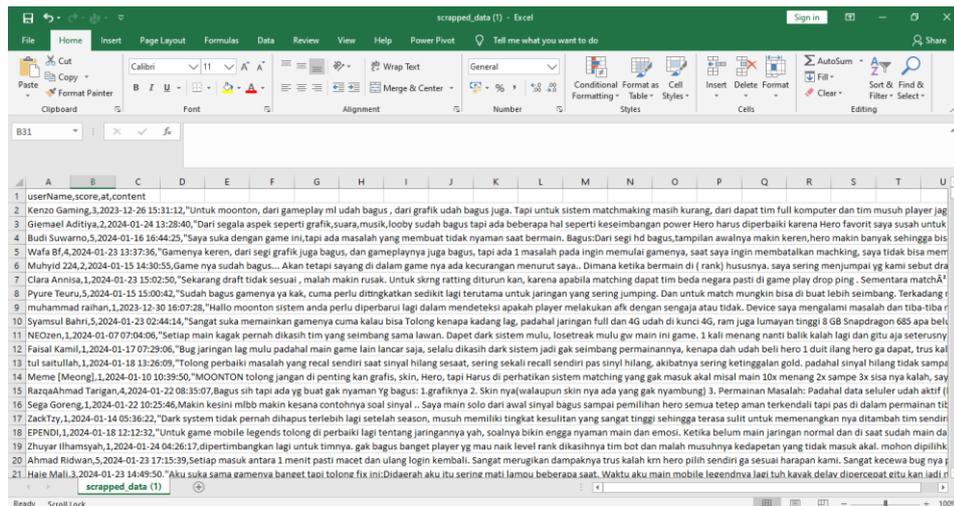
```
#Run This Code to Sort the Data By Date
new_df = df_busu[['userName', 'score', 'at', 'content']]
sorted_df = new_df.sort_values(by='at', ascending=False) #Sort by Newest, change to True if you wa
sorted_df.head()
```

	userName	score	at	content
2368	Banun Yusup	1	2024-01-25 06:35:07	Buat para player baru mending gak usah instal ...
489	tendou pain	1	2024-01-25 02:28:37	Sistem matchmaking perbaiki. Masa tiap maen da...
2549	Fayy Villain	1	2024-01-24 23:40:09	gim terbaik makasih aku udah main 10 match bin...
4553	Gatra Perdana	5	2024-01-24 23:24:37	Game rusak & menyebalkan, sangat tidak adil da...
2123	Siti Rohimah	2	2024-01-24 23:07:42	Kredit skor gua turun lagi kena pa monton jel...

Gambar 4. 6 Hasil Penyortiran Data

g. Menyimpan data

Setelah seluruh pemrosesan selesai dalam proses scraping data, maka yang terakhir adalah melakukan penyimpanan data hasil dari proses scraping data. data yang disimpan dalam penelitian ini berbentuk file csv agar kita dapat dengan mudah mengolah data hasil dari scraping data aplikasi Mobile Legend dari Google Play Store. Berikut adalah hasil pemrosesannya.



Gambar 4. 7 Hasil Penyimpanan Scraping Data

4.1.1. Preprocessing Data

Proses selanjutnya adalah preprocessing pada dataset yang telah di scraping dan disimpan dalam bentuk csv data ini kembali diolah karna masih banyaknya data ulasan yang didapat masih menggunakan kata yang tidak berstruktur seperti singkatan, emotikon, simbol dan angka hal ini diperlukan teknik preprocessing. Preprocessing berguna untuk mengekstrak informasi dari ulasan, mengubah kata-kata yang tidak terstruktur itu menjadi bentuk standar. Pada pemrosesan preprocessing kali ini akan menggunakan python dalam melakukan preprocessing. Tahapan yang digunakan sesuai dengan langkah dan tahapan yang telah di jelaskan dalam bab tiga yaitu diantaranya dilakukan pemrosesan case folding, penghapusan simbol dan karakter, Lemmatization, Stopwrod dan Tokenize. Berikut adalah hasil dari pemrosesan preprocessing penelitian ini.

a. Case Folding

Pada langkah pertama, dilakukan proses "Case Folding" yang bertujuan untuk mengubah seluruh teks dalam kolom "content" menjadi huruf kecil. Hal ini bertujuan untuk menghindari ambiguitas yang disebabkan oleh perbedaan huruf besar dan kecil dalam analisis teks. Berikut adalah hasil dari pemrosesanya.

```
# 1. Case Folding
df_content['content'] = df_content['content'].str.lower()
df_content['content'].head()

0    untuk moonton, dari gameplay ml udah bagus , d...
1    dari segala aspek seperti grafik,suara,musik,l...
2    saya suka dengan game ini,tapi ada masalah yan...
3    gamenya keren, dari segi grafik juga bagus, da...
4    game nya sudah bagus... akan tetapi sayang di ...
Name: content, dtype: object
```

Gambar 4. 8 Pemrosesan Case Folding

Case folding memberikan konsistensi dalam representasi teks, sehingga proses selanjutnya dapat dilakukan dengan lebih efektif. Ini membantu mencegah kata yang sebenarnya sama dianggap berbeda karena huruf kapital.

b. Penghapusan Simbol, Angka dan Emoji

Langkah kedua melibatkan penghapusan simbol, angka, dan emoji dari teks dalam kolom "content". Tujuan dari langkah ini adalah untuk memfokuskan analisis pada informasi teks murni dan menghilangkan elemen-elemen yang mungkin tidak relevan. Berikut adalah hasil dari pemrosesan penghapusan simbol, angka dan emoji.

```
# 2. Penghapusan Simbol, Angka, dan Emoji
import re
df_content['content'] = df_content['content'].apply(lambda x: re.sub(r'^a-z\s', '', x))
df_content['content'].head()

0    untuk moonton dari gameplay ml udah bagus dar...
1    dari segala aspek seperti grafiksuaramusikloob...
2    saya suka dengan game initapi ada masalah yang...
3    gamenya keren dari segi grafik juga bagus dan ...
4    game nya sudah bagus akan tetapi sayang di dal...
Name: content, dtype: object
```

Gambar 4. 9 Pemrosesan Penghapusan Simbol, Angka dan Emoji

Penghapusan simbol dan angka dapat membantu menghilangkan noise dan mempertahankan hanya informasi teks yang mendasar. Ini mendukung pembentukan representasi teks yang lebih bersih dan lebih konsisten.

c. Lemmatization

Proses lemmatization dilakukan pada langkah ketiga untuk mengubah kata-kata dalam teks menjadi bentuk dasar atau kata dasar mereka. Tujuannya adalah untuk mengurangi variasi kata sehingga analisis dapat lebih fokus pada makna inti. Berikut adalah hasil dari pemrosesan Lemmatization.

```
# 3. Lemmatization
from nltk.stem import WordNetLemmatizer
import nltk

nltk.download('wordnet')
lemmatizer = WordNetLemmatizer()
df_content['content'] = df_content['content'].apply(lambda x: ' '.join([lemmatizer.lemmatize(word) for word in x.split()]))
df_content['content'].head()

[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Package wordnet is already up-to-date!
0    untuk moonton dari gameplay ml udah bagus dari...
1    dari segala aspek seperti grafiksuaramusikloob...
2    saya suka dengan game initapi ada masalah yang...
3    gamenya keren dari segi grafik juga bagus dan ...
4    game nya sudah bagus akan tetapi sayang di dal...
Name: content, dtype: object
```

Gambar 4. 10 Pemrosesan Lemmatization

Lemmatization membantu dalam mengidentifikasi makna dasar dari kata-kata, sehingga memungkinkan analisis yang lebih baik terhadap inti pesan yang ingin disampaikan dalam teks. Hal ini memfasilitasi pemahaman yang lebih baik terhadap sentimen atau pesan yang terkandung.

d. Stopword Removal

Langkah keempat melibatkan penghapusan stopwords, yaitu kata-kata umum yang seringkali tidak memberikan informasi signifikan. Hal ini bertujuan untuk menyederhanakan teks dan meningkatkan fokus pada kata-kata kunci. Berikut adalah hasil dari pemrosesan Stopword Removal.

```
# 4. Stopword Removal
from nltk.corpus import stopwords

nltk.download('stopwords')
stop_words = set(stopwords.words('indonesian'))
df_content['content'] = df_content['content'].apply(lambda x: ' '.join([word for word in x.split() if word not in stop_words]))
df_content['content'].head()

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
0    moonton gameplay ml udah bagus grafik udah bag...
1    aspek grafiksuaramusiklooby bagus keseimbangan...
2    suka game initapi nyaman bermain bagusdari seg...
3    gamenya keren segi grafik bagus gameplaynya ba...
4    game nya bagus sayang game nya kecurangan dima...
Name: content, dtype: object
```

Gambar 4. 11 Pemrosesan Stopword Removal

e. Tokenize

Terakhir, dilakukan proses tokenization untuk memecah teks menjadi unit-unit yang lebih kecil, seperti kata-kata. Tujuannya adalah untuk mempersiapkan teks agar dapat diolah lebih lanjut dalam analisis. Berikut adalah hasil dari pemrosesan teknizenya.

```
# 5. Tokenization
from nltk.tokenize import word_tokenize

nltk.download('punkt')
df_content['content'] = df_content['content'].apply(word_tokenize)
df_content['content'].head()
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
0      [moonton, gameplay, ml, udah, bagus, grafik, u...
1      [aspek, grafiksuaramusiklooby, bagus, keseimba...
2      [suka, game, initapi, nyaman, bermain, bagusda...
3      [gamenya, keren, segi, grafik, bagus, gameplay...
4      [game, nya, bagus, sayang, game, nya, kecurang...
Name: content, dtype: object
```

Gambar 4. 12 Pemrosesan Tokenize

Tokenization memfasilitasi representasi teks dalam bentuk yang lebih terstruktur, memungkinkan pengolahan lebih lanjut seperti pembentukan vektor kata atau model pembelajaran mesin. Ini menjadi langkah penting dalam persiapan data untuk tugas analisis sentimen.

Secara keseluruhan, proses preprocessing data ini memberikan kontribusi penting dalam mempersiapkan dataset untuk analisis sentimen. Setiap tahapan memiliki perannya sendiri dalam membersihkan, menyederhanakan, dan mempersiapkan teks komentar pengguna sehingga dapat diolah secara lebih efektif. Dengan dataset yang telah melalui tahapan preprocessing ini, diharapkan analisis sentimen pada aplikasi Mobile Legends dapat memberikan hasil yang lebih akurat dan bermakna. Preprocessing data menjadi tahap kritis dalam menyediakan fondasi yang kuat untuk analisis sentimen lebih lanjut.

4.1.2. Pelabelan Data

Setelah melalui serangkaian tahapan preprocessing, setiap komentar telah diolah dan dibersihkan untuk memastikan representasi teks yang konsisten. Proses selanjutnya adalah memberikan label sentimen pada setiap komentar. Label sentimen diberikan berdasarkan interpretasi positif atau negatif dari konten setelah tahapan preprocessing. Komentar yang dianggap mengandung sentimen positif akan diberi label "Positif", sedangkan komentar yang mengandung sentimen negatif akan diberi label "Negatif". proses pelabelan data dilakukan dengan tujuan untuk mengidentifikasi apakah setiap komentar bersifat positif atau negatif. Proses pelabelan ini dilakukan pada 1000 data dari total 10,000 data sebagai data uji untuk mengevaluasi kinerja model analisis sentimen. Berikut adalah contoh kalimat yang telah diberi label data positif dan negatif pada komentar masyarakat dalam aplikasi mobile legend di google play store.

Tabel 4. 1 Pelabelan content pada dataset.

No	Teks Komentar	Sentiment
1	developer sangat baik. game yang bisa menghilangkan stress sehingga membuat pemain sangat ramah.	Positive
2	menurut saya game lumayan lah tapi masalah pemberian tim tidak sesuai dengan musuh saya jadi frustrasi terus	Positive
3	seru banget game nya tapi sayang nya kalo udah ke master dan naik rank nya musuh nya tambah susah. tapi saya senang dengan game ini	Positive
4	orang bodoh kalo bikin game gini nih main rank ketemu tim poke main magic chees Hero yg di inginkan gk keluar keluar buat teman teman kalo gk mau stress main game lebih baik jangan main game ini	Negative
5	game ga jelas tiba tiba lag ga gerak sama sekali. Padahal jaringan bagus bangettt	Negative
6	game buat orang stress dapet tim sampah terus montoon emang ga niat mikirin duit doang menang sekali kalah 3 kali	Negative

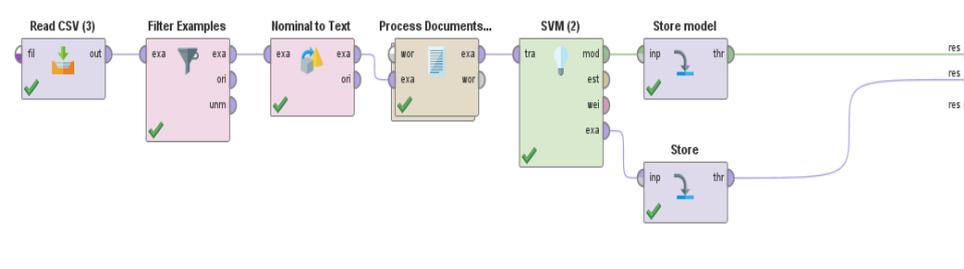
	gara gara timnya tolol semua game dengan matchmaking tertolol yg pernah gw mainin	
--	--	--

4.1.3. Pemrosesan Algoritma

Setelah melalui tahapan preprocessing dan pelabelan data sebagaimana yang telah dijelaskan pada Bab 4.1.3, langkah selanjutnya adalah melakukan pemrosesan algoritma untuk membangun model analisis sentimen. Dalam penelitian ini, algoritma yang digunakan adalah Support Vector Machine (SVM) yang akan di-boosting menggunakan algoritma Adaboost. Algoritma SVM dipilih karena kemampuannya dalam menangani klasifikasi data pada ruang berdimensi tinggi dan memiliki keakuratan yang tinggi. Dalam konteks analisis sentimen, SVM sangat efektif dalam mengklasifikasikan teks menjadi dua kelas sentimen, yaitu positif dan negatif. Langkah awal dalam pemrosesan algoritma adalah melatih model SVM menggunakan dataset yang telah dipreprocess dan dilabeli. Setelah model SVM terlatih, akan dilakukan boosting menggunakan algoritma Adaboost untuk meningkatkan kinerja model. Adaboost akan membantu meningkatkan akurasi klasifikasi dengan memberikan bobot lebih pada sampel yang salah diklasifikasikan pada iterasi sebelumnya. Pemrosesan algoritma ini bertujuan untuk menghasilkan model analisis sentimen yang akurat dan dapat diandalkan dalam mengklasifikasikan komentar pengguna aplikasi Mobile Legends. Dengan mengintegrasikan SVM dan Adaboost, diharapkan model yang dihasilkan mampu mengatasi kompleksitas dari data teks dan meningkatkan performa dalam memprediksi sentimen dari komentar pengguna.

a. Modeling Algoritma

Dalam tahapan Modeling algoritma dalam penelitian ini menggunakan RapidMiner, serangkaian operator yang digunakan untuk membangun model analisis sentimen berdasarkan dataset yang telah dipreproses dan dilabeli. Berikut adalah pemrosesan pada tahapan modeling algoritma menggunakan SVM.



Gambar 4. 13 Modeling Algoritma

Berikut adalah penjelasan untuk setiap operator yang digunakan dalam tahapan modeling pada gambar 4.13 diatas.

1. *Read CSV*

Operator "Read CSV" digunakan untuk membaca dataset yang telah dipreprocess dan dilabeli. Dataset ini berisi komentar pengguna aplikasi Mobile Legends yang telah dibersihkan dan diberi label sentimen.

2. *Filter Example*

Operator "Filter Example" digunakan untuk memilih hanya 1000 data dari total 10,000 data sebagai data uji untuk evaluasi model. Ini dilakukan untuk mempercepat proses dan menghindari overfitting pada model.

3. *Nominal to Text*

Operator "Nominal to Text" digunakan untuk mengkonversi kolom teks dalam dataset menjadi format teks. Hal ini diperlukan untuk memastikan bahwa teks dapat diproses dengan benar dalam langkah-langkah selanjutnya.

4. *Process Documents*

Operator "Process Documents" digunakan untuk melakukan serangkaian langkah preprocessing teks yang meliputi tokenisasi, transformasi kasus teks, penghapusan stopwords, dan penyaringan token. Langkah-langkah ini bertujuan untuk membersihkan dan mempersiapkan teks untuk pemodelan.

5. *SVM*

Operator "SVM" digunakan untuk membangun model Support Vector Machine (SVM) untuk klasifikasi sentimen. Model SVM akan dilatih menggunakan dataset yang telah dipreprocess dan dilabeli. SVM dipilih

karena kemampuannya dalam menangani klasifikasi data pada ruang berdimensi tinggi dan memiliki keakuratan yang tinggi.

6. Store

Operator "Store" digunakan untuk menyimpan model SVM yang telah dilatih serta dataset yang telah dipreprocess dan dilabeli. Model yang disimpan dapat digunakan untuk melakukan prediksi sentimen pada data baru yang belum terlabel.

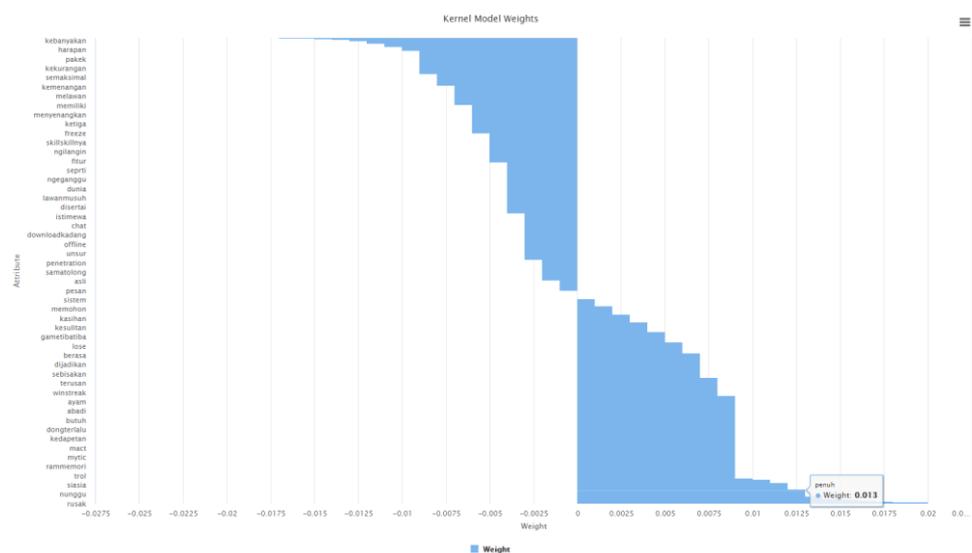
Dengan menggunakan serangkaian operator ini, nantinya dapat digunakan untuk membangun model analisis sentimen yang akurat dan dapat diandalkan untuk mengklasifikasikan sentimen pada komentar pengguna aplikasi Mobile Legends. Proses ini memberikan dasar yang kuat untuk evaluasi kinerja model dan pengambilan keputusan yang informatif terkait dengan pengembangan dan perbaikan aplikasi. Berikut adalah hasil dapemrosesan modeling algoritma.

coenter	label	function...	alpha	abs(alpha)	support vector	aaangat	abadi	abis	abistu	acak	acakabel	adif	aditapi	aduan	advance	afkalo	afkimal	afknya	afksekali	afkturn	ahaha...
0	positf	-1.000	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	positf	-1.000	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	positf	-0.999	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	positf	-1.000	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	positf	-1.000	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	negatif	0.964	0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	positf	-1.000	-0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	positf	-1.001	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	positf	-1.000	-0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	positf	-1.000	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	negatif	0.447	0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	positf	-1.000	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	positf	-0.591	-0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	positf	-1.001	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	positf	-0.935	-0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	positf	-1.000	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	positf	-1.001	-0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	negatif	0.376	0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	positf	-1.001	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	positf	-0.999	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	positf	-1.000	-0.000	0.000	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	negatif	0.206	0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	negatif	0.600	0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	negatif	0.817	0.001	0.001	support vector	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...

Gambar 4. 14 Hasil pengolahan Modeling Algoritma

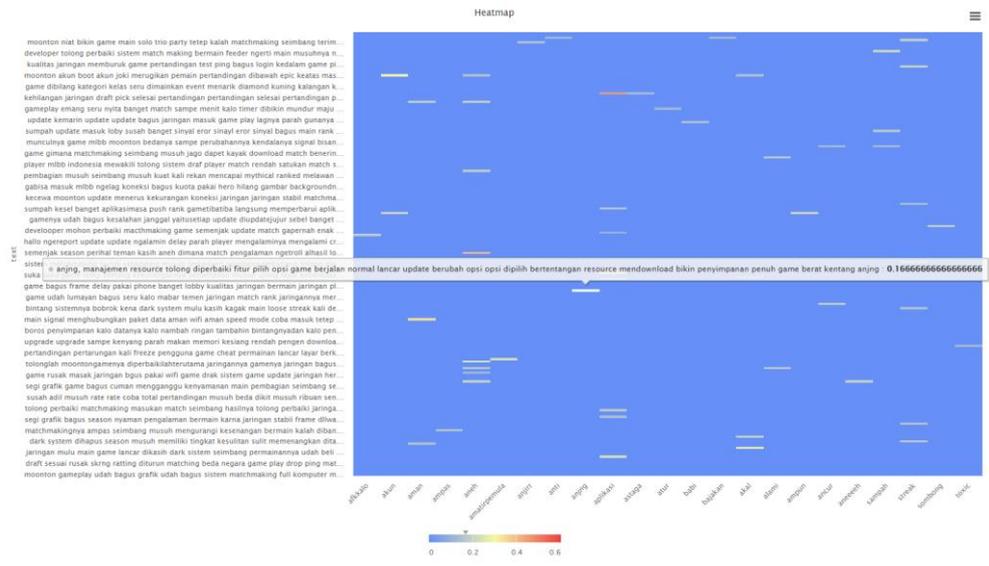
Hasil kernel model SVM yang dihasilkan dari pemrosesan algoritma menggunakan RapidMiner menyediakan informasi penting tentang model yang telah dilatih. Hasil dari proses tersebut adalah data yang telah dipreprocess dan siap digunakan untuk pemodelan atau analisis selanjutnya, seperti pemrosesan validasi analisis sentimen menggunakan algoritma Support Vector Machine (SVM). Data ini memiliki representasi yang lebih

terstruktur dan relevan, memfasilitasi analisis yang lebih akurat dan informatif terkait dengan sentimen dalam komentar pengguna aplikasi Mobile Legends. Pada kernel model SVM dalam hasil diatas menentukan batas keputusan yang memisahkan dua kelas dengan menggunakan hyperplane dalam ruang fitur yang diperbesar (dalam kernel space). Informasi tentang persamaan batas keputusan, termasuk koefisien dan konstanta, seringkali disertakan dalam hasil kernel. Ini memungkinkan untuk memvisualisasikan batas keputusan dan memahami bagaimana model SVM mengklasifikasikan data. Kemudian terdapat nilai Weights pada pemrosesan algoritma svm dalam penelitian ini. berikut adalah hasil visualisasinya.



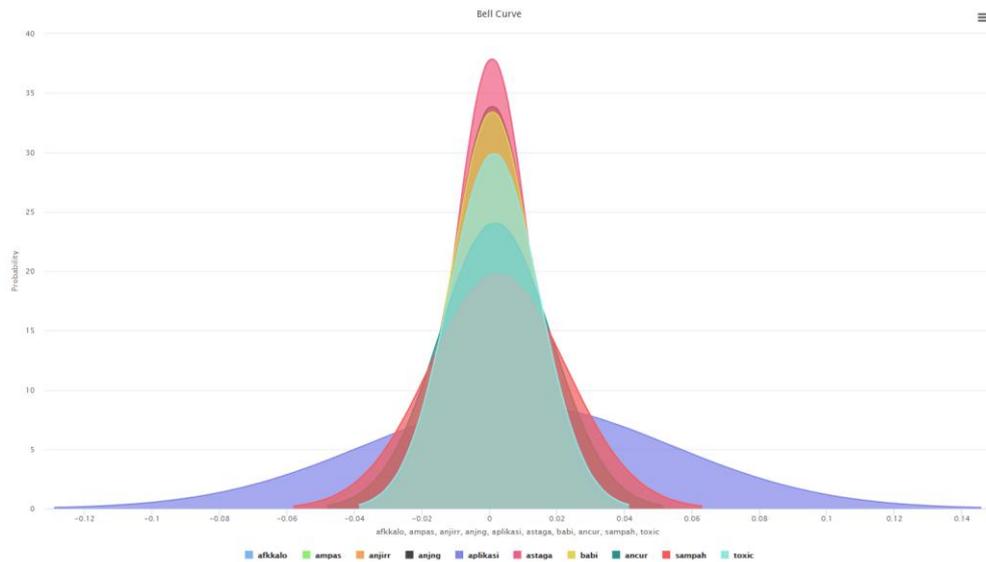
Gambar 4. 15 Visualisasi Kernel Model Weights

Setelah itu terdapat visualisasi dalam bentuk heatmap. Heatmap adalah salah satu jenis visualisasi yang digunakan untuk memvisualisasikan data dalam bentuk matriks dengan menggunakan warna. Dalam konteks RapidMiner, heatmap biasanya digunakan untuk memvisualisasikan hubungan atau pola dalam data tabular yang diolah. Berikut adalah visualisasinya.



Gambar 4. 16 Visualisasi Heatmap

Dalam visualisasi heatmap diatas dapat terlihat pembagian teksnya berdasarkan berdasarkan kata kata yang sering muncul dalam datase dimana beberapa komentar menunjukan adanya sentiment negatif dalam dataset tersebut. Selanjutnya adalah visualisasi dalam bentuk Bell Curve

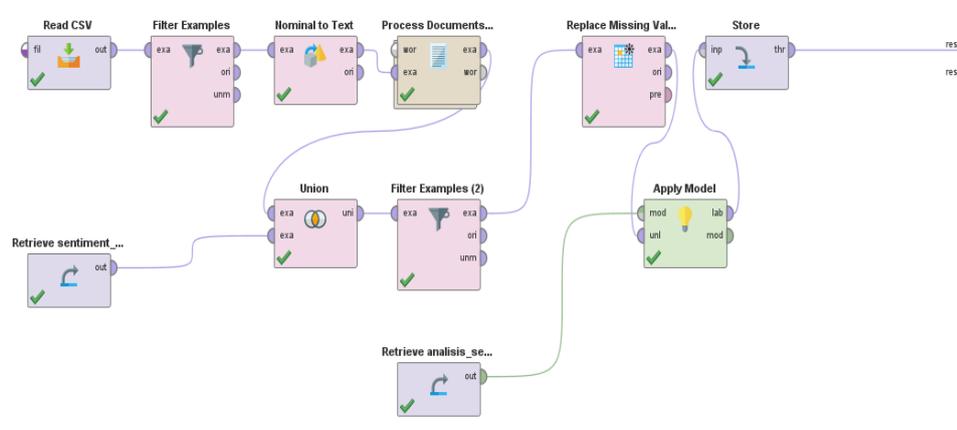


Gambar 4. 17 Visualisasi Bell Curve

Dalam visualisasi Bell Curve diatas dapat dilihat dari 10 kata yang diambil dari sampel dataset pada beberapa kata menunjukan nilai yang tinggi seperti kata astaga dengan nilai probality sebesar 37.84 dari nilai rentang 40.

b. Validasi Model

Dalam proses pengembangan model analisis sentimen, tahap validasi model memainkan peran penting dalam mengevaluasi kinerja model pada data yang belum terlihat sebelumnya. Pada tahapan ini, data yang belum terlabel diproses menggunakan model yang telah dilatih sebelumnya untuk memprediksi sentimen yang terkandung dalam teks. validasi model dalam RapidMiner memungkinkan evaluasi kinerja model pada data yang belum terlihat sebelumnya. Hasil evaluasi performa model ini dapat digunakan untuk memahami sejauh mana model dapat menggeneralisasi pada data baru, serta untuk mengidentifikasi dan mengatasi masalah yang mungkin muncul dalam pemodelan analisis sentimen. Pada intinya berdasarkan pada modeling untuk membentuk store model dan dari modeling algoritma sebagai acuan dan di terapkan pada validasi model untuk melakukan pelabelan pada label data yang belum terlabeli berdasarkan label sebelumnya. Berikut adalah alur pemrosesan validasi model.



Gambar 4. 18 Pemrosesan Validasi Model

Alur pemrosesan yang kompleks pada gambar diatas dirancang untuk melakukan pemodelan masal secara otomatis dan berikut fungsi pemrosesan pada setiap operator yang digunakan dalam rapidminer.

1. *Pembacaan Data Awal (Read CSV)*

Data yang berisi teks yang belum terlabel dibaca dari file CSV menggunakan operator "Read CSV". Langkah ini membentuk basis awal untuk pemrosesan data lanjutan.

2. *Pemilihan Data yang Belum Terlabel (Filter Examples)*

Operator "Filter Examples" digunakan untuk memilih hanya teks yang belum terlabel, sehingga hanya data yang belum dianotasikan yang akan diproses lebih lanjut. Hal ini memastikan bahwa model hanya diterapkan pada data yang belum diberi label sentimen.

3. *Konversi Nominal ke Teks (Nominal to Text):*

Data yang dipilih, yang masih dalam bentuk nominal, dikonversi menjadi format teks menggunakan operator "Nominal to Text". Langkah ini memungkinkan data teks untuk menjalani proses preprocessing selanjutnya.

4. *Pemrosesan Teks (Process Documents):*

Data teks yang telah dikonversi diproses menggunakan operator "Process Documents", yang melibatkan serangkaian langkah preprocessing seperti tokenisasi, transformasi kasus teks, penghapusan stopwords, dan filtering token. Hal ini bertujuan untuk membersihkan dan mempersiapkan teks agar sesuai dengan kebutuhan pemodelan algoritma.

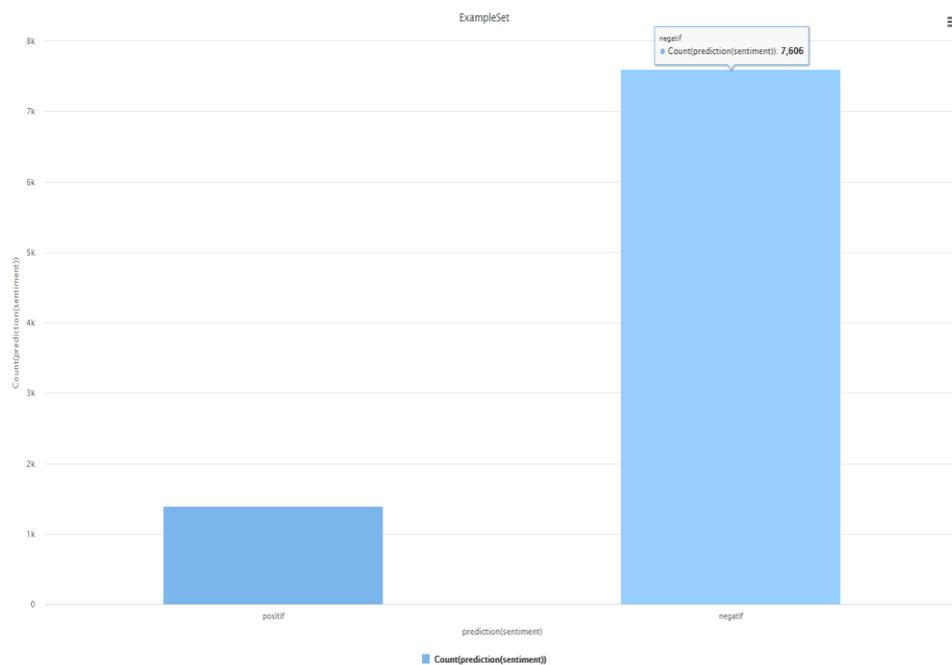
5. *Penggabungan Data (Union):*

Operator "Union" digunakan untuk menggabungkan output dari pemrosesan teks dengan data yang telah dilatih sebelumnya. Input dari "Union" meliputi data teks yang telah diproses dan data yang telah dilatih sebelumnya yang disimpan dalam "store latih". Langkah ini mempersiapkan data untuk tahap prediksi sentimen.

6. *Pemilihan Sentimen yang Belum Terlabel (Filter Examples):*

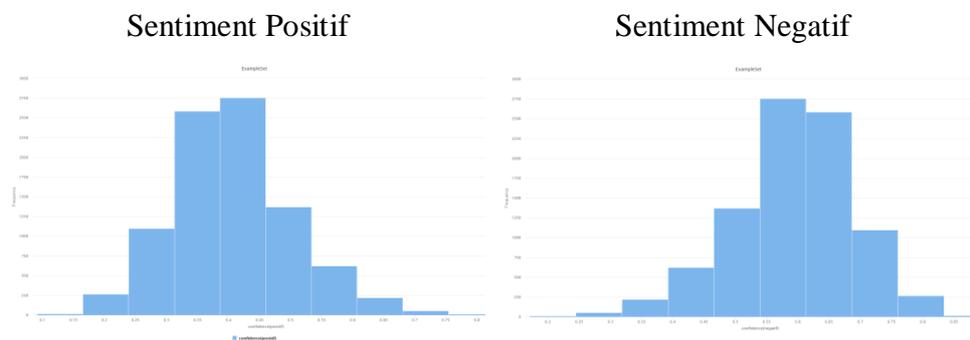
Setelah penggabungan data, operator "Filter Examples" digunakan untuk memilih hanya data dengan sentimen yang masih kosong (belum terlabel). Langkah ini memastikan bahwa model hanya diterapkan pada data yang belum diberi label sentimen.

Pada hasil pelabelan dapat dilihat sentiment yang belum terisi label sebelumnya dapat di prediksi berdasarkan pendekatan positif dan negatif dimana nilai yang cenderung lebih besar akan melabeli text tersebut menjadi label positif maupun negatif. Kemudian pada jumlah hasil yang terlabeli positif dan negatif pada 9000 data yang belum terlabeli adalah sebagai berikut.



Gambar 4. 19 Perbandingan Sentiment Positif dan Negatif

Dalam gambar 4.19 dapat kita lihat bahwa jumlah Sentiment negatif lebih banyak daripada sentiment positif. Dimana sentiment negatif berjumlah 7.606 komentar dan sentiment positif sebesar 1.934 Komentar. Dalam hasil ini menunjukan banyaknya ketidakpuasan masyarakat terhadap aplikasi tersebut berdasarkan dari dataset yang telah di ambil melalui scraping data pada google playstore.



Gambar 4. 20 Perbandingan Sentiment Negatif dan Sentiment Positif

Dapat dilihat pada gambar 4.20 bahwa perbandingan jumlah sentiment negatif dan sentiment positif lebih besar sentiment negatif dimana nilai confiden yang melebihi 0.5 adalah nilai dari sentiment negatif.

c. Hasil Pemrosesan

Setelah melalui tahapan validasi model dengan alur pemrosesan yang telah dijelaskan sebelumnya berikut adalah hasil pemrosesan dan evaluasi model yang mencakup nilai akurasi, presisi, dan recall. Setelah memproses data yang belum terlabel dan menerapkan model pada data tersebut, hasil evaluasi performa model telah diperoleh. Pada tahapan ini melanjutkan dari pemrosesan sebelumnya dimana saat hasil pemrosesan di simpan dalam bentuk format csv kemudian dalam menghitung akurasinya menggunakan pemrosesan algoritma svm dalam menemukan hasil akurasi presisi dan recall. Evaluasi ini mencakup metrik-metrik sebagai berikut:

1. Akurasi (Accuracy):

Akurasi merupakan ukuran yang menggambarkan seberapa baik model dapat mengklasifikasikan data secara keseluruhan dengan benar. Nilai akurasi dihitung sebagai persentase dari jumlah prediksi yang benar (positif dan negatif) dibagi dengan jumlah total prediksi.

2. Presisi (Precision):

Presisi mengukur tingkat ketepatan dari model dalam mengklasifikasikan data positif. Secara khusus, presisi menghitung persentase dari prediksi positif yang benar dibagi dengan jumlah total prediksi positif.

3. Recall (Sensitivity atau True Positive Rate):

Recall mengukur kemampuan model dalam menemukan semua instansi dari kelas yang positif. Secara khusus, recall menghitung persentase dari prediksi positif yang benar dibagi dengan jumlah total instansi yang sebenarnya positif dalam dataset.

4. Interpretasi Hasil

Hasil akurasi, presisi, dan recall memberikan gambaran yang komprehensif tentang kinerja model analisis sentimen. Akurasi mengukur tingkat keseluruhan keberhasilan model, sedangkan presisi dan recall memberikan wawasan tambahan tentang kinerja model dalam mengklasifikasikan sentimen secara spesifik. Berikut adalah pemrosesan dalam mencari nilai akurasi menggunakan algoritma SVM.

accuracy: 90.79% +/- 1.64% (micro average: 90.79%)

	true positif	true negatif	class precision
pred. positif	1364	0	100.00%
pred. negatif	308	1672	84.44%
class recall	81.58%	100.00%	

Gambar 4. 21 Hasil Akurasi

Gambar 4.21 diatas adalah hasil accuracy pada pengolahan sentiment analisis menggunakan algoritma SVM dengan cross validation. Pada hasil tersebut nilai accuracy yang diperoleh mendapat nilai akurasi sebesar 90.79%. selanjutnya hasil Precision adalah sebagai berikut.

precision: 84.44% +/- 2.70% (micro average: 84.44%) (positive class: negatif)

	true positif	true negatif	class precision
pred. positif	1364	0	100.00%
pred. negatif	308	1672	84.44%
class recall	81.58%	100.00%	

Gambar 4. 22 Hasil Precision

Gambar 4.22 diatas adalah hasil Precision pada pengolahan sentiment analisis menggunakan algoritma SVM dengan cross validation. Pada hasil tersebut

nilai Presisi yang diperoleh mendapat nilai akurasi sebesar 84.44%. selanjutnya hasil Recall adalah sebagai berikut.

recall: 100.00% +/- 0.00% (micro average: 100.00%) (positive class: negatif)

	true positif	true negatif	class precision
pred. positif	1364	0	100.00%
pred. negatif	308	1672	84.44%
class recall	81.58%	100.00%	

Gambar 4. 23 Hasil Recall

Gambar 4.22 diatas adalah hasil Recall pada pengolahan sentiment analisis menggunakan algoritma SVM dengan cross validation. Pada hasil tersebut nilai Recall yang diperoleh mendapat nilai akurasi sebesar 100%.

Dengan memperhatikan nilai-nilai metrik evaluasi ini, peneliti dapat mengevaluasi sejauh mana model dapat mengklasifikasikan sentimen dengan tepat dan mengidentifikasi area di mana model mungkin perlu diperbaiki atau disempurnakan. Dari pemrosesan tersebut dapat dilihat bahwa nilai akurasi dan juga presisi masih dapat di tingkatkan. Untuk melakukan peningkatan tersebut dibutuhkan algoritma bosting yaitu adaboost. Berikut adalah hasil pengujian menggunakan adaboost.

accuracy: 91.06% +/- 1.79% (micro average: 91.06%)

	true positif	true negatif	class precision
pred. positif	1375	2	99.85%
pred. negatif	297	1670	84.90%
class recall	82.24%	99.88%	

Gambar 4. 24 Hasil Akurasi SVM dengan Adaboost

Gambar 4.24 diatas adalah hasil Akurasi pada pengolahan sentiment analisis menggunakan algoritma SVM dengan dengan Adaboost. Pada hasil tersebut nilai accuracy yang diperoleh mendapat nilai akurasi sebesar 90.79%. selanjutnya hasil Precision adalah sebagai berikut.

precision: 84.94% +/- 2.91% (micro average: 84.90%) (positive class: negatif)

	true positif	true negatif	class precision
pred. positif	1375	2	99.85%
pred. negatif	297	1670	84.90%
class recall	82.24%	99.88%	

Gambar 4. 25 Hasil Presisi SVM dengan Adaboost

Gambar 4.25 diatas adalah hasil Presision pada pengolahan sentiment analisis menggunakan algoritma SVM dengan dengan Adaboost. Pada hasil tersebut nilai accuracy yang diperoleh mendapat nilai akurasi sebesar 84.91%. selanjutnya hasil Recall adalah sebagai berikut.

recall: 99.87% +/- 0.40% (micro average: 99.88%) (positive class: negatif)

	true positif	true negatif	class precision
pred. positif	1375	2	99.85%
pred. negatif	297	1670	84.90%
class recall	82.24%	99.88%	

Gambar 4. 26 Hasil Presisi SVM dengan Adaboost

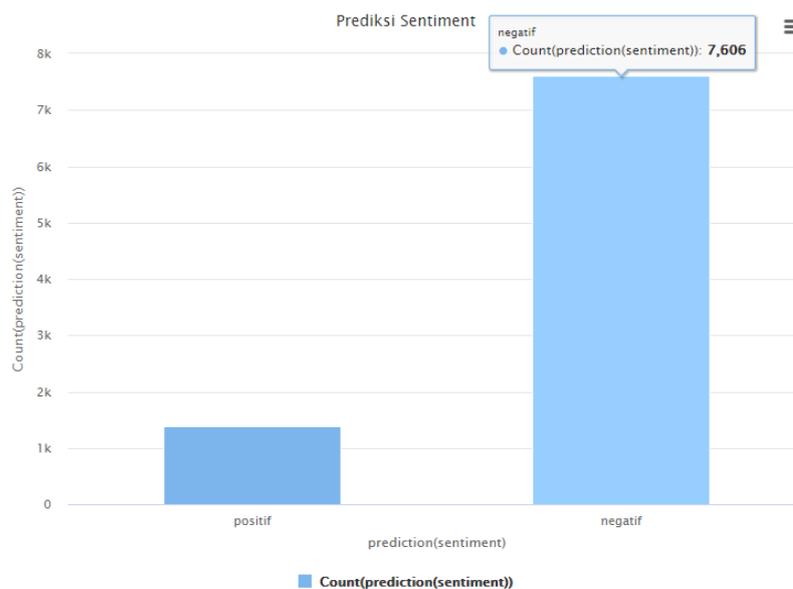
Gambar 4.26 diatas adalah hasil Recall pada pengolahan sentiment analisis menggunakan algoritma SVM dengan dengan Adaboost. Pada hasil tersebut nilai Recall yang diperoleh mendapat nilai akurasi sebesar 99.85%. selanjutnya hasil Recall adalah sebagai berikut.

Dengan demikian, hasil pemrosesan dan evaluasi model ini memberikan pandangan yang mendalam tentang kinerja model analisis sentimen yang telah dikembangkan, dan menyediakan dasar yang kuat untuk interpretasi hasil dan pengambilan keputusan lebih lanjut terkait dengan penggunaan model dalam konteks aplikasi Mobile Legends dimana dalam menggunakan algoritma SVM dengan di bosting menggunakan adaboost dapat meningkatkan akurasi dari pemrosesan model pada algoritma tersebut

4.2. Pembahasan Hasil Pnelitian

Pembahasan pada hasil penelitian membahas tentang penelitian yang telah dilakukan. Penelitian ini melakukan prediksi pada sentiment analisis dimana beberapa komentar pada aplikasi mobile legend yang terdapat di playstore telah di

scraping untuk menjadi sebuah dataset yang kemudian dari ulasan review masyarakat tersebut akan di dilakukan analisis sentiment untuk mengetahui seberapa banyak ulasan positif maupun negative dalam penelitian ini. Pada peneltian ini berhasil mengumpulkan dataset sebanyak 10.000 ulasan komentar baik dari rating 1 sampai rating 5. Kemudian dalam memprediksi ulasan tersebut dilakukan pelabelan manual sebanyak 1000 data komentar yang nantinya 1000 data tersebut akan dilakukan sebuah modeling sebagai data acuan untuk melabeli sisa data dalam ulasan yang berjumlah 9000 tersebut secara otomatis. Dan berikut adalah hasil dari pelabelan secara otomatis dimana data sentiment negatif lebih banyak dibanding dengan data sentiment positif dengan 7.606 sentiment negatif dan 1.394 sentiment positif. Berdasarkan data yang telah dimodeling dan di latih menggunakan algoritma SVM menunjukkan lebih banyak sentiment negatif dalam ulasan Mobile legends di aplikasi play store. Berikut adalah visualisasi hasilnya.



Gambar 4. 27 Hasil Prediksi Sentiment

Kemudian setelah mendapat hasil dari sentiment positif dan negatif pada ulasan menggunakan algoritma svm. Selanjutnya adalah menganalisis kata-kata yang paling sering muncul dalam teks yang telah diproses. Analisis ini dapat dilakukan dengan menggunakan teknik visualisasi yang disebut "Word Cloud". Word cloud merupakan representasi visual dari frekuensi kemunculan kata dalam teks, di mana kata-kata yang muncul lebih sering akan ditampilkan dengan ukuran yang lebih

Tabel 4. 3 Hasil Perbandingan Nilai Akurasi

NO	Performance	Support Vector Machine	SVM & Adaboost
1	Akurasi	90,79 %	91,06 %
2	Presisi	84,44 %	84,94 %
3	Recall	100 %	99, 87 %

Pada hasil perhitungan diatas dapat dilihat pada pemrosesan SVM dengan SVM dikombinasikan dengan adaboost dapat meningkatkan nilai akurasi yang cukup baik. Begitupun dengan nilai presisi namun justru turun pada nilai recall ketika menggunakan adaboost.