

ANALISIS SENTIMEN PADA OPINI PENGGUNA APLIKASI QASIR MENGUNAKAN SUPPORT VECTOR MACHINE DAN RANDOM FOREST

(SENTIMENT ANALYSIS ON USER OPINION OF QASIR APPLICATION USING A SUPPORT VECTOR MACHINE AND RANDOM FORESTS)

Dhana Aulia Ayu Kurniawan¹⁾, Ema Utami²⁾, Hanif Al Fatta³⁾

^{1, 2,3)} Magister Teknik Informatika, Universitas AMIKOM Yogyakarta

Jalan Ring Road Utara, Condong Catur, Depok, Sleman, Yogyakarta 55281

e-mail: dhana8277@students.amikom.ac.id¹⁾, ema.u@amikom.ac.id²⁾, hanif.a@amikom.ac.id³⁾

ABSTRAK

Qasir merupakan aplikasi Point-Of-Sale (POS) berbasis android yang bisa diakses secara gratis pada Google Playstore. Dengan banyaknya aplikasi POS yang tersedia, pengguna akan lebih selektif dalam memilih aplikasi yang akan digunakan. Salah satu aspek yang dapat mempengaruhi keputusan memilih aplikasi adalah opini pada aplikasi tersebut. Opini merupakan informasi yang didapatkan setelah menggunakan aplikasi bisa berisi kritik maupun saran. Sehingga berdasarkan hal tersebut pengguna dapat menyimpulkan bagaimana pengguna lain menggunakan aplikasi tersebut. Selain berguna untuk pengguna, opini jika diolah dengan baik akan menghasilkan sebuah informasi yang dapat digunakan untuk evaluasi bagi tim pengembang. Untuk menganalisa dan menemukan hubungan antar data yang dimiliki dapat menggunakan Data Mining. Penelitian ini akan menggunakan metode Support Vector Machine dan Random Forest, namun masing masing metode memiliki kekurangan dan kelebihannya sehingga kedua metode tersebut akan dibandingkan nilai akurasi. Hasil yang didapat adalah Support Vector Machine memiliki nilai akurasi tertinggi dengan 80,63% sedangkan Random Forest sebesar 80,21%.

Kata Kunci: POS, Klasifikasi, Random Forest, Support Vector Machine, Analisis Sentimen.

ABSTRACT

Qasir is an Android-based Point-Of-Sale (POS) application that can be accessed for free on the Google Playstore. With so many POS applications available, users will be more selective in choosing the application to use. One aspect that can influence the decision to choose an application is the opinion on the application. Opinion is information obtained after using the application that can contain criticism or suggestions. So based on this the user can conclude how other users use the application. Besides being useful for users, opinions if processed properly will produce information that can be used for evaluation for the development team. To analyze and find relationships between owned data, you can use Data Mining. This research will use the Support Vector Machine and Random Forest methods, but each method has its advantages and disadvantages so that the accuracy of the two methods will be compared. The results obtained are that the Support Vector Machine has the highest accuracy value with 80.63% while the Random Forest is 80.21%.

Keywords: POS, Classification, Random Forest, Support Vector Machine, Sentiment Analysis

I. PENDAHULUAN

Google Play Store merupakan sebuah aplikasi pada *smartphone* dimana berbagai jenis kategori aplikasi dapat diunduh secara berbayar maupun gratis. Qasir merupakan salah satu aplikasi *Point-Of-Sale* (POS) berbasis android yang terdapat di Google Play Store. Aplikasi ini diciptakan untuk mencatat penjualan, mengelola produk, mengawasi stok dan memantau laporan transaksi. Akan tetapi banyaknya aplikasi POS membuat pengguna lebih selektif dalam menentukan aplikasi yang akan digunakan. Opini

merupakan informasi yang didapatkan setelah menggunakan aplikasi bisa berisi kritik maupun saran. Sehingga berdasarkan hal tersebut pengguna dapat menyimpulkan bagaimana aplikasi tersebut. Selain berguna untuk pengguna, opini jika diolah dengan baik akan menghasilkan sebuah informasi yang dapat digunakan untuk evaluasi bagi tim pengembang.

Untuk menganalisa dan menemukan hubungan antar data dapat menggunakan *Data Mining*. Di antara banyaknya metode yang dapat digunakan untuk menganalisis opini berbentuk teks, yang akan digunakan dalam penelitian ini adalah

metode *Support Vector Machine* (SVM) dan *Random Forest*.

Metode *Support Vector Machine* (SVM) merupakan teknik prediksi, klasifikasi maupun regresi yang digunakan pada banyak aplikasi seperti bio-informatika, pengenalan tulisan tangan, dan sebagainya. Implementasi dari SVM ini memerlukan data *training* dan data *testing* sehingga digolongkan dalam *supervised learning*. [8] Konsep dasar dari SVM ini adalah mencari *hyperline* terbaik sebagai pemisah antara dua *class* data. Metode ini memiliki kelebihan diantaranya implementasi yang mudah, kemampuan untuk mengklasifikasikan pattern yang tidak masuk dalam *class* dalam *training set*, kemudian dapat meminimalisir error pada *training set*, dan kemampuan untuk menghadapi masalah dalam *pattern*. [5]

Sedangkan *Random Forest* dibangun dari beberapa *decision tree* dengan memilih sejumlah *F* fitur secara random sehingga digunakan sebagai *node* untuk membangun *decision tree*. Nilai *F* akan mempengaruhi hasil akhir dari kinerja algoritma *Random Forest*. Jika nilai *F* terlalu kecil, maka nilai korelasi dari *tree* tersebut semakin kecil. Kemudian jika nilai *F* nya terlalu besar maka nilai korelasi dari *tree* tersebut akan semakin besar. [4]. Metode ini membangun banyak *tree* dalam mengklasifikasikan suatu objek berdasarkan atributnya. Keuntungan menggunakan metode ini dapat mengcover *dataset* besar dengan dimensi tinggi. Mampu mengestimasi *missing* data dan mempertahankan akurasi secara efektif ketika jumlah *missing* data yang banyak hal ini dikarenakan *Random Forest* mempunyai metode tersendiri untuk *balancing error*. [6]

Tujuan dari penelitian ini adalah menganalisis sentiment opini pada aplikasi Qasir menggunakan algoritma *Support Vector Machine* dan *Random Forest* yang kemudian hasil akurasi dari masing masing aplikasi akan dibandingkan untuk menemukan metode mana yang paling cocok dengan data opini yang dimiliki. Dengan menggunakan 2 kelas (positif dan negative), penelitian ini akan menggunakan sebanyak 2767 data opini yang akan di proses terlebih dahulu menggunakan Phyton. Sedangkan untuk proses analisis metode akan menggunakan aplikasi Rapid Miner 9.9.

II. STUDI PUSTAKA

Pada penelitian sebelumnya yang menganalisis sentimen menggunakan *Support Vector Machine*

(SVM) dengan jumlah *dataset* 3000 opini aplikasi pada google playstore dan metode evaluasi menggunakan *k-fold cross validation* yang bernilai 10 mendapatkan nilai akurasi tertinggi sebesar 90,67% dengan menggunakan metode *pre-processing* data yaitu *Tokenization*, *transform case*, *Filter tokens (by length)*, *Filter stopwords*, *Stemming* dan *Labeling*. [1]

Selanjutnya merupakan penelitian terdahulu yang menggunakan metode modifikasi dari *Random Forest* yaitu *Modified Balanced Random Forest*. Hal ini dikarenakan data yang dihadapi merupakan *imbalanced* data yang nantinya akan mempengaruhi performa dari analisis yang akan dibangun. Metode dari *pre-processing* yang digunakan adalah *case folding*, *cleaning*, *tokenization*, *stopword removal*, *stemming* dan *remove unknown word*. Dari penelitian ini menghasilkan nilai akurasi tertinggi sebesar 79% dengan nilai *F1-Scores* 74% [9].

Penelitian sebelumnya membandingkan performa dari metode *Naïve Bayes* dan *Support Vector Machine* dengan menggunakan *Particle Swarm Optimization* (PSO) dan *dataset* yang digunakan sebanyak 1364 opini. Hasil penelitian menunjukkan bahwa performa dari metode PSO - *Support Vector Machine* lebih baik dengan nilai akurasi 93% dan nilai AUC 97% dibandingkan PSO - *Naïve Bayes* dengan nilai akurasi 69% dan nilai AUC 65%. [10]

Kemudian terdapat penelitian sebelumnya yang membandingkan kinerja antara metode *Random Forest*, *Naïve Bayes* dan *Support Vector Machine* dengan tweet sebagai *dataset* dan menggunakan metode pembobotan kata TD-IDF menghasilkan akurasi *Support Vector Machine* tertinggi diantara metode lainnya dengan nilai 77,58% disusul dengan *Random Forest* sebesar 75,81% dan *Naïve Bayes* sebesar 75,22%. Teknik pra - pemrosesan data yang digunakan ada *Case-folding*, *cleaning text*, *Tokenize*, *Stemming* dan *Stopword removal*. [3]

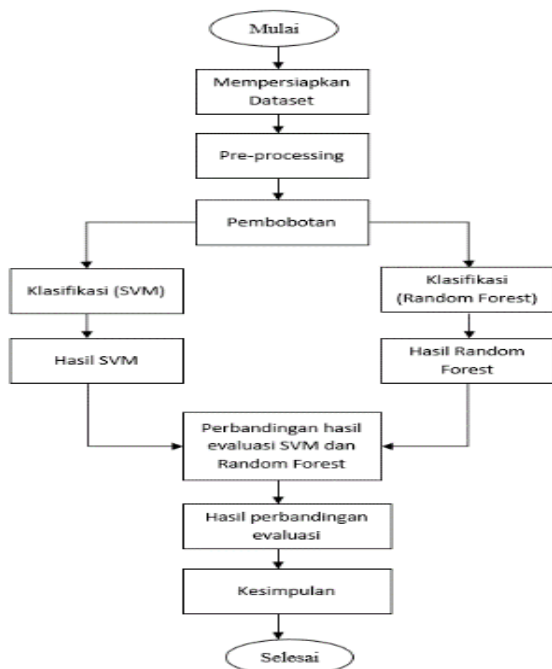
Selanjutnya terdapat penelitian yang membandingkan *Naïve Bayes*, *Support Vector Machine*, *Decision Trees* dan *Random Forest* menggunakan 400.000 opini amazon sebagai *dataset* dengan perbandingan 80% untuk data *training* dan 20% untuk data *testing*. Metode *pre-processing* yang digunakan adalah *Tokenization*, *Removal Stopwords* dan *Stemming*. Menghasilkan metode *Support Vector Machine* dengan akurasi tertinggi 89% dan sebagai metode yang komplit dikarenakan memiliki nilai yang tinggi untuk

semua atribut evaluasi seperti Akurasi, Presisi, *Recall* dan *F1 Score*. Disusul dengan metode *Random Forest* yang menghasilkan akurasi tertinggi 88% kemudian metode *Naïve Bayes* dengan nilai 85% dan terakhir metode *Decision Tree* dengan nilai 82%. [11]

Yang terakhir adalah penelitian sebelumnya yang membandingkan *Naïve Bayes*, *Random Forest* dan *Support Vector Machine* menggunakan dataset opini aplikasi Ruang Guru pada Google Playstore dengan data sebanyak 1629. Dengan menggunakan metode *pre-processing Tokenization*, *Stemming*, *Case Folding* dan *Stopword Removal* menghasilkan bahwa Metode *Random Forest* memiliki akurasi tertinggi dengan nilai 97,16% dan AUC sebesar 0,996 kemudian SVM dengan nilai akurasi 96,01% dengan AUC sebesar 0,543 dan yang terakhir adalah *Naïve Bayes* dengan akurasi 94,16% dengan AUC sebesar 0,999 [2].

III. METODE PENELITIAN

Pada penelitian ini merupakan penelitian eksperimen dengan sifat penelitian deskriptif dan menggunakan pendekatan kuantitatif karena bertolak ukur dari performa model yang digunakan. Tahapan dari penelitian ini dapat dilihat pada gambar 1 dibawah ini:



Gambar 1. Alur Penelitian

Secara keseluruhan, penelitian ini akan dibagi dalam 2 tahap yaitu :

A. Pengumpulan Data

Dengan menggunakan data pada kolom komentar google playstore. Metode yang akan digunakan adalah *web scraping* yang merupakan proses untuk pengambilan atau ekstraksi data dari sebuah *website* yang disimpan dalam format tertentu. Kemudian data tersebut akan dievaluasi. Setelahnya data yang akan disimpan adalah dalam format csv.

B. Pengolahan Data

- Text preprocessing* : Tahapan untuk mempersiapkan data sebelum digunakan pada proses selanjutnya dengan cara membersihkan data teks mentah, karena data yang diperoleh biasanya bersifat tidak terstruktur dan memiliki banyak *noise* contohnya imbuhan, tanda baca, karakter khusus, angka, dan lain sebagainya. [5] Pada penelitian ini menggunakan scenario *Case folding*, *cleaning*, *stopword removal*, *remove frequent and rare word*, dan *stemming*.
- Labelling* : Menggunakan lexicon based dengan SentiWordNet yang merupakan hasil dari anotasi otomatis WordNet.
- Pembobotan* : Menggunakan metode TF-IDF dengan menghitung jumlah kemunculan term setiap dokumen dikali dengan jumlah setiap kemunculan kata yang sama pada semua dokumen.
- Klasifikasi Data* : Menggunakan metode SVM dan *Random Forest*.
- Evaluasi* : Menggunakan *Cross Validation*, yang merupakan merupakan teknik validasi untuk mengukur hasil analisa akan menggeneralisasi data independen. Pada dasarnya, teknik ini akan memprediksi model kemudian memperkirakan seberapa akurat hasil dari analisis model. Salah satu teknik dari *cross validation* adalah *k-fold cross validation*. Fokus utamanya adalah menghilangkan bias pada data yang digunakan sehingga pendekatan yang dilakukan adalah memecah data dengan ukuran yang sama berdasarkan k bagian set data. [7]

Setelah keseluruhan proses dilakukan, dataset hasil akan diterjemahkan kedalam *wordcloud* sehingga bisa diketahui kata yang sering muncul dalam komentar aplikasi Qasir ini.

IV. HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan implementasi dari rancangan yang dipaparkan pada bab sebelumnya.

A. Pengumpulan Data

Pada penelitian ini, data berasal dari Google PlayStore, nama *instance* dari aplikasi yang akan diambil datanya adalah 'com.innovecto.etalastic' *language* EN dan *country* ID untuk Indonesia. Untuk *syntax* dapat dilihat pada gambar 2.

```
!pip install google-play-scraper
from google_play_scraper import app
import pandas as pd
import numpy as np
from google_play_scraper import Sort, reviews_all

result = reviews_all(
    'com.innovecto.etalastic',
    sleep_milliseconds=0,
    lang='en',
    country='id',
    sort=Sort.MOST_RELEVANT
)
```

Gambar 2. Crawling Data All

Setelah mengambil seluruh data *review*, data akan di *filter* menggunakan *score*. Jika ingin mengambil seluruh data maka *filter score* *None*, akan tetapi jika ingin mengambil spesifik *score* maka *value None* diganti dengan 1,2,3,4 atau 5. Untuk *syntax* dapat dilihat pada gambar 3.

```
from google_play_scraper import Sort, reviews
result, continuation_token = reviews(
    'com.innovecto.etalastic',
    lang='en',
    country='us',
    sort=Sort.MOST_RELEVANT,
    count=3000,
    filter_score_with=None
)
```

Gambar 3. Crawling Data Filter

Karena tidak akan menggunakan seluruh data, maka perlu di *select column* yang digunakan dalam tahapan selanjutnya. Pada penelitian ini menggunakan 4 *column* yaitu *username*, *at*, *content* dan *score*. Untuk *syntax* dapat dilihat pada gambar 4.

```
df_busu = pd.DataFrame(np.array(result), columns=['review'])
df_busu = df_busu.join(pd.DataFrame(df_busu.pop('review').tolist()))
df_busu.head()
```

Gambar 4. Crawling Data Select Column

Berikut merupakan hasil dari *scraping* data pada gambar 5.

	userName	at	content	score
0	Alvin Sanjaya Putra	2021-01-22 16:32:25	great appl to think that this app's are mostly...	3
1	Niko Sutiono	2020-06-27 01:45:15	For now I give it a 4 stars, but potential to ...	4
2	A Google user	2019-07-31 12:02:10	keep developing your awesome appl you guys are...	5
3	Dwi Permana	2022-01-14 23:56:38	It was a really good app and simple to used. I...	3
4	Debora Oey	2020-12-09 18:29:23	It's very helpful and easy to use. The basic f...	5

Gambar 5. Hasil Akhir Data

Proses terakhir dalam pengumpulan data adalah *download* hasil tersebut kedalam file CSV untuk bisa digunakan dalam proses selanjutnya yang bisa dilihat pada gambar 6.

```
my_df.to_csv("scrapped_data.csv", index = False)
```

Gambar 6. Convert to CSV

B. Text Preprocessing

Pada tahapan ini menggunakan Google Colab dengan Bahasa Phyton. Terdapat beberapa tahapan yang dilakukan yaitu :

- Transform Case* : Mengubah seluruh huruf menjadi huruf kecil yang bisa dilihat pada gambar 7.

```
df["content"] = df["English"].str.lower()
df.head()
```

	English	content
0	Very useful and help me in selling! Thank God ...	very useful and help me in selling! thank god ...
1	These apps really make it easier for me to man...	these apps really make it easier for me to man...
2	Quite helpful in the financial process, I say...	quite helpful in the financial process, I say...
3	Why doesn't the help function work? how to add...	why doesn't the help function work? how to add...
4	The new version is better than the one that ha...	the new version is better than the one that ha...

Gambar 7. Transform Case

- Remove Punctuation / Tokenize* : Menghilangkan karakter tertentu seperti tanda baca, angka, simbol, dan link URL seperti pada gambar 8.

<pre> PUNCT_TO_REMOVE = string.punctuation def remove_punctuation(text): return text.translate(str.maketrans('', '', PUNCT_TO_REMOVE)) df["text_wo_punct"] = df["content"].apply(lambda text: remove_punctuation(text)) df.head() </pre>		
English	content	text_wo_punct
0	Very useful and help me in selling! Thank God ...	very useful and help me in selling! thank god ...
1	These apps really make it easier for me to man...	these apps really make it easier for me to man...
2	Quite helpful in the financial process, I say...	quite helpful in the financial process, i say...
3	Why doesn't the help function work? how to add...	why doesnt the help function work? how to add...
4	The new version is better than the one that ha...	the new version is better than the one that ha...

Gambar 8. Tokenize

- c. *Stopword* : Menghilangkan kata yang dianggap tidak penting pada kalimat dengan menggunakan *library* dari *nltk.corpus* dalam Bahasa Inggris. Proses import terlihat pada gambar 9.

```

from nltk.corpus import stopwords
", ".join(stopwords.words('english'))

```

Gambar 9. Import NLTK Corpus

Setelah daftar kata terimport, kemudian masuk kedalam proses *stopword*. Untuk proses dan hasilnya dapat dilihat pada gambar 10.

<pre> df.drop(["content"], axis=1, inplace=True) STOPWORDS = set(stopwords.words('english')) def remove_stopwords(text): """Custom function to remove the stopwords""" return " ".join([word for word in str(text).split() if word not in STOPWORDS]) df["text_wo_stop"] = df["text_wo_punct"].apply(lambda text: remove_stopwords(text)) df.head() </pre>		
English	text_wo_punct	text_wo_stop
0	Very useful and help me in selling! Thank God ...	useful help selling thank god finally apps like
1	These apps really make it easier for me to man...	apps really make easier manage distributor bus...
2	Quite helpful in the financial process, I say...	quite helpful financial process say good job
3	Why doesn't the help function work? how to add...	doesnt help function work add menu im using as...
4	The new version is better than the one that ha...	new version better one hasnt updated discount...

Gambar 10. Stopword

- d. *Remove Frequent word* : Menghilangkan kata yang sering muncul seperti pada gambar 11.

<pre> df.drop(["text_wo_punct"], axis=1, inplace=True) FREQUWORDS = set([w for (w, uc) in cnt.most_common(10)]) def remove_frequent(text): """Custom function to remove the frequent words""" return " ".join([word for word in str(text).split() if word not in FREQUWORDS]) df["text_wo_stopfreq"] = df["text_wo_stop"].apply(lambda text: remove_frequent(text)) df.head() </pre>		
English	text_wo_stop	text_wo_stopfreq
0	Very useful and help me in selling! Thank God ...	useful help selling thank god finally apps like
1	These apps really make it easier for me to man...	apps really make easier manage distributor bus...
2	Quite helpful in the financial process, I say...	quite helpful financial process say good job
3	Why doesn't the help function work? how to add...	doesnt help function work add menu im using as...
4	The new version is better than the one that ha...	new version better one hasnt updated discount...

Gambar 11. Remove Frequent Word

- e. *Remove Rare word* : Pada tahapan ini akan menghilangkan kata yang jarang muncul seperti pada gambar 12.

<pre> df.drop(["text_wo_stop"], axis=1, inplace=True) n Rare words = 10 RAREWORDS = set([w for (w, uc) in cnt.most_common()[1:-n Rare words-1:-1]]) def remove_rarewords(text): """Custom function to remove the rare words""" return " ".join([word for word in str(text).split() if word not in RAREWORDS]) df["text_wo_stopfreq"] = df["text_wo_stopfreq"].apply(lambda text: remove_rarewords(text)) df.head() </pre>		
English	text_wo_stopfreq	text_wo_stopfreqrare
0	Very useful and help me in selling! Thank God ...	useful help selling god finally apps like
1	These apps really make it easier for me to man...	apps really make easier manage distributor bus...
2	Quite helpful in the financial process, I say...	quite financial process say job
3	Why doesn't the help function work? how to add...	doesnt help function work add menu im using as...
4	The new version is better than the one that ha...	new version better one hasnt updated discount...

Gambar 12. Remove Rare Word

- f. *Stemming* : Mengubah kata menjadi kata dasarnya yang prosesnya dapat dilihat pada gambar 13.

<pre> from nltk.stem.porter import PorterStemmer stemmer = PorterStemmer() def stem_words(text): return " ".join([stemmer.stem(word) for word in text.split()]) df["text_stemmed"] = df["text_wo_stopfreq"].apply(lambda text: stem_words(text)) df.head() </pre>		
English	text_wo_stopfreqrare	text_stemmed
0	Very useful and help me in selling! Thank God ...	useful help selling god finally apps like
1	These apps really make it easier for me to man...	apps really make easier manage distributor bus...
2	Quite helpful in the financial process, I say...	quite financial process say job
3	Why doesn't the help function work? how to add...	doesnt help function work add menu im using as...
4	The new version is better than the one that ha...	new version better one hasnt updated discount fe...

Gambar 13. Stemming

- g. *Labeling* : Untuk labeling akan menggunakan *sentiwordnet* dengan *output* yang akan diambil hanya Positif dan Negatif saja. Untuk prosesnya dapat dilihat pada gambar 14.

<pre> posnegdf['count'] = 0 data['pos_tag'] = [nltk.pos_tag(word_tokenize(sentence)) for sentence in data['text_stemmed']] def pos_tag(tag): return un_ADT if tag.startswith('') else un_NOUN if tag.startswith('N') else un_AJW if tag.startswith('A') else un_VEB if tag.startswith('V') else None def get_sentiment(word, tag): if pos_tag(tag) not in (un_NOUN, un_ADT, un_AJW): return [] synsets = un_synset(word, pos_tag(tag)) if not synsets: return [] synset = synsets[0] sum_synset = sum_sentiment(synset.name()) return [synset.name(), sum_synset.pos_score(), sum_synset.neg_score(), sum_synset.obj_score()] posnegdf['count'] += 1 sent_score = [] for pos_val in data['pos_tag']: senti_val = [get_sentiment(x,y) for (x,y) in pos_val] for score in senti_val: try: pos = pos + score[1] neg = neg + score[2] except: continue senti_score.append(pos - neg) posnegdf['sent_score'] = senti_score overall = [] for i in range(len(data)): label = 'Positive' if data['sent_score'][i] >= 0.85 else 'Negative' if data['sent_score'][i] <= -0.85 else 'Neutral' data['sentiment'] = overall data = data[data.sentiment == 'Neutral'] data = data.reset_index(drop=True) </pre>		
---	--	--

Gambar 14. Labeling

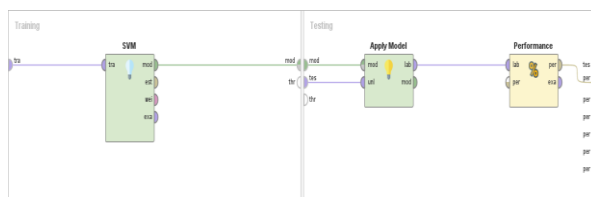
Data mentah yang awalnya 2767, setelah diproses menjadi 1213. Hal ini membuktikan bahwa separuh dari data mentah tidak dapat digunakan lebih lanjut karena tidak memenuhi kriteria dari setiap proses *data pre-processing* yang dilakukan. Setelah itu data akan diexport ke csv. Hasil akhir bisa dilihat pada gambar 15 dibawah ini.

English	text_wo_stopregare	sentimen
These apps really make it easier for me to manage my distributor business. Forward etalastic!	these apps really make easier me manage my distributor business. forward etalastic!	Positive
Quite helpful in the financial process. i say, good job (y)	quite in financial process. i say, job (y)	Positive
Why doesn't the help function work? how to add menu how? i'm using asus z4c. thx :)	why doesn't help function work? how add menu how? i'm using asus z4c. thx :)	Negative
The new version is better than the one that hasn't been updated, so the discount feature can be synchronized without manual input to make it easier	new version better than one that hasn't been updated, so discount feature can be synchronized without manual input make easier	Positive
The apps are really cool	apps are really cool	Positive
Helpful enough	enough	Positive
Very helpful for my financial report,	my financial report,	Positive
Very helpful, good luck to the founders.	helpful, luck founders.	Positive
The app is good. Only after the update it keeps crashing. i hope the next update will be good.	app good. only after update keeps crashing. i hope next update will be good.	Positive
If the purchase could choose the debt option would be better. There is only cash option	if purchase could choose debt option would be better. there only cash option	Positive
Simple. Powerful. Free	simple. powerful. free	Positive
Very cool!!!!	cool!!!!	Positive

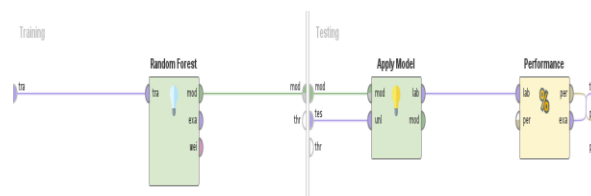
Gambar 15. Data Akhir

C. Pengolahan data

Selanjutnya adalah memproses data di Rapid Miner menggunakan metode *Support Vector Machine* (SVM) dan *Random Forest* yang dapat dilihat pada gambar 16 dan 17. Pada proses tersebut dilakukan didalam *cross validation* yang meliputi proses *training* dan *testing* dengan menggunakan proses pemodelan untuk mendapatkan nilai akurasi. Untuk nilai k pada masing masing metode menggunakan 5.



Gambar 16. Proses SVM



Gambar 17. Proses Random Forest

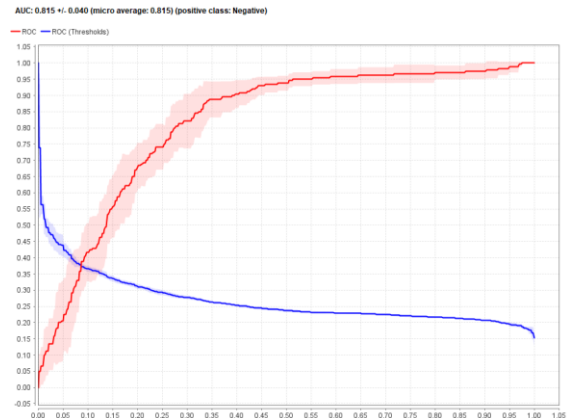
Nilai akurasi yang didapat metode SVM adalah 80,63% dengan *margin* +/- 1.55%. Dan rata rata micro yang didapatkan adalah 80,63%. Sehingga nilai maksimal untuk akurasinya adalah 82.18% dan minimalnya adalah 79.08%. Sedangkan untuk presisi positif bernilai 81.38% dan *negative* sebesar 56.75%. Untuk *recall* positif sebesar 98.36% dan recall negative sebesar 8.75%. Untuk nilai AUCnya sebesar 0.815, sehingga analisis sudah dilakukan dengan baik. Pada gambar 18 terdapat hasil akurasi dan *confusion matrix* yang dihasilkan oleh metode SVM.

accuracy: 80.63% +/- 1.55% (micro average: 80.63%)

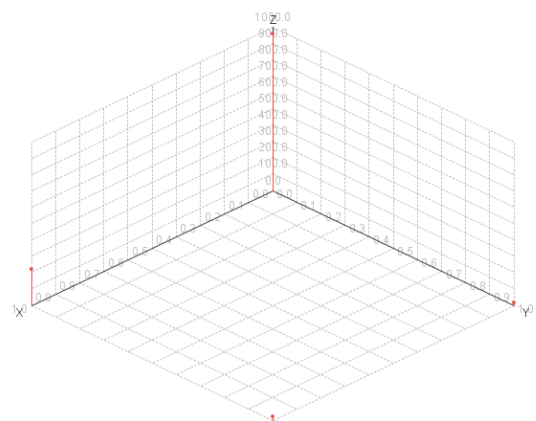
	true Positive	true Negative	class precision
pred. Positive	957	219	81.38%
pred. Negative	16	21	56.76%
class recall	98.36%	8.75%	

Gambar 18. Hasil SVM

Kemudian pada gambar 19 terdapat Grafik ROC dan gambar 20 terdapat *Plot View* yang dihasilkan oleh metode SVM.



Gambar 19. Grafik AUC SVM



Gambar 20. Plot View SVM

Sedangkan nilai akurasi yang didapatkan metode *Random Forest* sebesar 80.21% dengan *margin* +/- 0.04%. Rata rata micro yang didapat adalah 80.21%. Sehingga nilai maksimal untuk akurasinya sebesar 80.25% dan minimal sebesar 80.17%. Untuk nilai presisi positif nya sebesar 80.21% dan presisi *negative* nya sebesar 0%. Untuk nilai *recall* positif sebesar 100% dan recall negative sebesar 0%. Untuk nilai AUCnya sendiri sebesar 0.820, sehingga analisis sudah dilakukan dengan baik. Pada gambar 21 terdapat hasil akurasi dan *confusion matrix* yang dihasilkan oleh metode *Random Forest*.

Secara keseluruhan, dalam komentar positif rata rata pengguna menuliskan tentang penggunaan aplikasi Qasir yang simple, useful, dan bagus. Dan pada komentar negatif, pengguna mendapatkan error pada saat menggunakan aplikasi di fitur print, transaksi dan pada saat menggunakan aplikasi Pro (Qasir-Pro/ layanan berbayar).

V. KESIMPULAN

Berdasarkan analisis dan pengujian yang dilakukan pada opini aplikasi Qasir di Google Playstore menggunakan metode SVM dan *Random Forest*, maka kesimpulan yang diambil adalah akurasi SVM lebih baik 0,42% dibandingkan *Random Forest* dengan menggunakan tahapan text *preprocessing* yang sama dan k yang sama. Walaupun memiliki banyak kekurangan, penelitian ini berhasil mengklasifikasikan sentimen pengguna aplikasi Qasir. Dengan nilai AUC yang diatas 8 menunjukkan bahwa analisis dilakukan dengan baik.

DAFTAR PUSTAKA

- [1] M. I. Ahmadi, F. Apriani, M. Kurniasari, S. Handayani and D. Gustian, "SENTIMENT ANALYSIS ONLINE SHOP ON THE PLAY STORE USING METHOD SUPPORT VECTOR MACHINE (SVM)," *Seminar Nasional Informatika 2020*, 2020.
- [2] R. D. Himawan and Eliyani, "Perbandingan Akurasi Analisis Sentimen Tweet," *JEPIN ((Jurnal Edukasi dan Penelitian Informatika)*, 2021.
- [3] E. Fitri, Y. Yuliani, S. Rosyida and W. Gata, "Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine," *TRANSFORMATIKA*, 2020.
- [4] L. Muflikah, Widodo, W. F. Mahmudi and Solimun, *Machine Learning Dalam Bioinformatika*, Malang: UB Press, 2021.
- [5] F. A. Nugraha, R. Habibi and N. H. Harani, *Analisis Sentimen Terhadap Pembatasan Sosial Menggunakan Deep Learning*, Bandung: Kreatif industri Nusantara, 2020.
- [6] F. A. Sianturi, P. M. Hasugian, A. Simangunsong and B. Nadeak, *Data Mining Teori dan Aplikasi Weka*, Indonesia: IOCS Publisher, 2020, p. 113.
- [7] F. Tempola, M. Muhammad and A. Khairan, "Perbandingan Klasifikasi Antara KNN dan Naive Bayes pada Penentuan Status Gunung Berapi dengan K-Fold Cross Validation," *Jurnal Teknologi Informasi dan Ilmu Komputer*, 2018.
- [8] I. Werdiningsih, B. Nuqoba and M. , *Data Mining Menggunakan Android, Weka dan SPSS*, Surabaya: Pusat Penerbitan dan Percetakan UNAIR, 2020.
- [9] F. N. Zamzami and M. D. P. Adiwijaya, "Analisis Sentimen Terhadap Review Film Menggunakan Metode Modified Balanced Random Forest dan Mutual Information," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 2021.
- [10] A. Mustopa, Hermanto, Anna, E. B. Pratama, A. Hendini and D. Risdiansyah, "Analysis of User Reviews for the PeduliLindungi Application on Google Play Using the Support Vector Machine and Naive Bayes Algorithm Based on Particle Swarm Optimization," *IEEE*, 2020.
- [11] M. Guia, R. R. Silva and J. Bernardino, "Comparison of Naïve Bayes, Support Vector Machine, Decision Tree and Random Forest on Sentiment Analysis," *SCITEPRESS – Science and Technology Publications*, 2019.