Twitter Sentiment Extraction Analysis, Exploratory Data Analysis and Model

Zahrandika Putra, Fikri Putra Hidayat, Nurul Amelia
School of Electrical Engineering
Telkom University, Indonesia

putrazahrandika@student.telkomuniversity.ac.id, fikzhidayat@student.telkomuniversity.ac.id, nurulamelia@student.telkomuniversity.ac.id

Abstrak – Analisis sentimen dalam data ini merupakan suatu proses klasifikasi dokumen tekstual ke dalam tiga kelas yaitu kelas sentimen positif, negatif dan netral. Data yang diperoleh berasal dari Kaggle yang diambil melalui jejaring sosial *Twitter* berdasarkan query Bahasa Inggris. Pengklasifikasian ini bertujuan untuk menentukan sentimen publik terhadap objek tertentu yang disampaikan di *Twitter* dalam Bahasa Inggris. Data yang sudah terkumpul dibagi menjadi *test* dan *train* yang akan digunakan sebagai proses *preprocessing* untuk menghasilkan model klasifikasi melalui proses pelatihan. Untuk data *train* berjumlah 27.481 dan data *test* berjumlah 3.534. Algoritma yang digunakan untuk *Twitter Sentiment Analysis* yaitu Random Forest, Multinomial Naive Bayes, Bagged Multinomial Naive Bayes, Gaussian Naive Bayes, SVM dan SVM Optimized. Untuk pemodelan paling bagus pada *train* terdapat pada algoritma Random Forest, untuk *test* terdapat pada algoritma SVM. Sedangkan untuk ROC-nya, ROC paling besar pada *train* terletak di algoritma Random Forest dan untuk *test*-nya terletak di algoritma SVM Optimized.

Kata Kunci : Analisis Sentiment, Klasifikasi, Twitter, Random Forest, SVM, Naive Bayes.

1. PENDAHULUAN

Indonesia adalah salah satu negara dengan pengguna sosial media terbanyak. Lebih dari 50% penduduk Indonesia menggunakan media sosial seperti *Twitter*, *Instagram*, YouTube, WhatsApp dan lainnya. *Twitter* menjadi salah satu sosial media yang cukup banyak digunakan masyarakat Indonesia. Bukan hanya di Indonesia saja *Twitter* digunakan, melainkan di negara lain juga. Opini yang diposting pada *Twitter* disebut dengan *tweet. Tweet* dapat digunakan untuk menentukan sentimen publik mengenai berbagai hal. Sentimen publik secara singkat adalah reaksi/emosi dari text yang ada, bisa berupa positif, negatif dan netral. Analisis sentimen dilakukan untuk melihat pendapat seseorang terhadap sebuah permasalahan. Terdapat pengaruh dan manfaat dengan adanya analisis sentimen menyebabkan penelitian mengenai analisis sentimen berkembang. Penelitian analisis sentimen ini dilakukan untuk mengetahui sentimen publik dengan menggunakan pendekatan Machine Learning dengan beberapa metode diantaranya Random Forest, Multinomial Naive Bayes, Bagged Multinomial Naive Bayes, Gaussian Naive Bayes, SVM dan SVM Optimized. Dengan adanya perkembangan penelitian mengenai analisis sentimen, maka peneliti tertarik untuk melakukan penelitian mengenai analisis sentimen pada *Twitter*.

2. RELATED WORK

Sentiment analysis adalah proses penggunaan text *analytics* untuk mendapatkan berbagai sumber data dari internet dan beragam platform media sosial, contohnya adalah *Twitter*. Tujuan-nya adalah untuk memperoleh opini dari pengguna yang terdapat pada platform tersebut, karena setiap waktu internet selalu dibanjiri data dari berbagai sumber.

3. METODE PENELITIAN

3.1 SENTIMENT ANALYSIS

Sentiment analysis berperan sebagai alat yang dapat menghubungkan seluruh data. *Sentiment analysis* salah satu bidang dari *Natural Language Processing* (NLP) yang membangun sistem untuk mengenali dan mengekstraksi opini dalam bentuk teks. Tipe-tipe *Sentiment Analysis* sebagai berikut :

a) Fine-Grained Sentiment Analysis

Tipe analisis sentimen ini akan mengelompokkan respon atau pendapat ke dalam beberapa kategori seperti positif, negatif dan netral.

b) Intent Sentiment Analysis

Tipe analisis sentimen ini bertujuan untuk mengidentifikasikan dan menggali lebih dalam motivasi dibalik pesan pengguna untuk melihat apakah itu termasuk keluhan, saran, pendapat, pertanyaan atau penghargaan dan lainnya.

c) Aspect-Based Sentiment Analysis

Tipe analisis sentiment ini berfokus pada elemen-elemen yang lebih spesifik dari produk atau layanan seseorang yang dipunya.

3.2 PENGUMPULAN DATA

Data didapatkan dari Kaggle, pengambilan data yang didapatkan dilakukan berdasarkan *query* atas *term* objek pada aplikasi yang terhubung. Hasil dari *query* berupa *tweet* untuk data pelatihan dan pengujian mengalami *preprocessing* yang sama.

3.3 PREPROCESSING

Tujuan dilakukannya *preprocessing* adalah untuk menghilangkan sebuah *noise*. Tetapi, data yang didapatkan melalui Kaggle sudah dikatakan ke dalam data bersih karena data yang ada sudah dipisahkan antara *train* dan *test*. Tetapi, dalam kasus ini masih perlu penghilangan tanda baca.

Penghilangan tanda baca di sini menggunakan modul *Regular Expression* untuk menghilangkan tanda baca. Berikut adalah beberapa simbol/tanda baca yang kami coba hilangkan dari data :

```
text = re.sub('\[.*?\]', '', text)
text = re.sub('https?://\S+|www\.\S+', '', text)
text = re.sub('<.*?>+', '', text)
text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
text = re.sub('\n', '', text)
text = re.sub('\w*\d\w*', '', text)
```

Gambar 1. Pembersihan tanda baca dan lainnya.

1	549e992a42	Sooo SAD I will miss you here in San Diego!!!

Gambar 2. Sebelum *preprocessing*

1	549e992a42	sooo sad i will miss you here in san diego

Gambar 3. Setelah *preprocessing*

Dalam proses di atas terdapat kekurangan karena pada saat klasifikasi kata yang paling sering muncul lebih banyak kata tidak bermakna atau *stopwords* seperti *is, am, are, the,* dsb. Maka dari itu perlu dilakukan pemrosesan lebih lanjut dengan modul NLTK *Stopwords*, sehingga kata-kata yang dipilih tidak berdasarkan *stopwords*. Dalam modul NLTK terdapat bahasa Indonesia juga, tetapi kami menyesuaikan dataset yang kami miliki yakni dalam Bahasa Inggris. Setelah menggunakan kedua modul tadi data yang kami miliki sudah terbilang cukup baik dengan *noise* yang minim.

```
def remove_stopword(x):
    return [y for y in x if y not in stopwords.words('english')]
train['temp_list'] = train['temp_list'].apply(lambda x:remove_stopword(x))
```

Gambar 4. Penghilangan kata yang tidak bermakna.

	Common_words	count
0	i	7200
1	to	5305
2	the	4590
3	a	3538

Gambar 5. Kata tidak bermakna belum dihilangkan.

	Common_words	count
1	good	1251
2	day	1058
3	love	909

Gambar 6. Kata tidak bermakna telah dihilangkan.

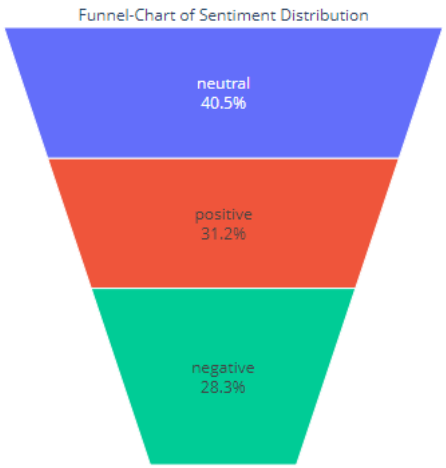
3.4 METODE KLASIFIKASI

Proses dimulai dengan pengambilan data yang dilakukan pada *Kaggle*. Dengan jumlah data yang didapatkan pada *train* adalah 27.481 sedangkan data *test* nya berjumlah 3.534. Kemudian, dilakukan proses dengan menggunakan method.dropna() yang berfungsi untuk menghapus suatu baris atau kolom yang mengandung *missing values*.

Pada bagian berikutnya kami mengelompokkan kembali data yang telah diolah pada *preprocessing* awal. Setelah kami menghilangkan *stopwords*, kami membuat kelompok berdasarkan *unique words* kata ini didapatkan dari train.csv yang kami miliki karena sudah terdapat pengelompokan berdasarkan tiga sentimen, positif, netral, dan negatif.

	sentiment	text
1	neutral	11117
2	positive	8582
0	negative	7781

Gambar 7. Jumlah kata netral, positif dan negatif pada data.



Gambar 8. Visualisasi dalam bentuk Funnel-Chart.

Pada visualisasi Funnel-Chart menampilkan jumlah kata netral, positif dan negatif dalam bentuk persen.

	words	count
0	congratulations	29
1	thnx	10
2	appreciated	8
3	shared	7
4	presents	7
5	greetings	7

Gambar 9. Contoh Kata Positif.

	words	count
0	ache	12
1	suffering	9
2	allergic	7
3	cramps	7
4	saddest	7
5	pissing	7

Gambar 10. Contoh Kata Negatif.

Pada bagian ini hasil klasifikasinya akan digunakan kemudian untuk proses fitting ke model. Akan tetapi, perlu adanya penyesuaian pada sentimen, dalam proses fitting model nanti karena datanya hanya akan ada positif dan negatif saja, jadi untuk yang netral dihapuskan dari dataset.

```
train = train[train['sentiment']!='neutral']
train['cleaned_tweet'] = train['text'].apply(clean_the_tweet)

train.head()
train['sentiment'] = train['sentiment'].apply(lambda x: 1 if x =='positive' else 0)
train.head()
```

Gambar 11. *Syntax* penghapusan sentimen netral.

	textID	text	sentiment	cleaned_tweet
1	96d74cb729	Shanghai is also really exciting (precisely -...	1	also really exciting precisely skyscrapers gal...
2	eee518ae67	Recession hit Veronique Branquinho, she has to...	0	veronique branquinho she has to quit her compa...
3	01082688c6	happy bday!	1	
4	33987a8ee5	http://twitpic.com/4w75p - I like it!!	1	com w p i like it
5	726e501993	that`s great!! weee!! visitors!	1	great weee visitors

Gambar 12. Tabel setelah sentimen netral dihapus.

Setelah melewati beberapa proses yang sudah dijelaskan, langkah selanjutnya adalah melakukan pemodelan dengan menggunakan metode klasifikasi menggunakan beberapa algoritma diantaranya Random Forest, Multinomial Naïve Bayes, Bagged Multinomial Naïve Bayes, Gaussian Naive Bayes, SVM dan SVM Optimized. Beberapa algoritma yang telah disebutkan akan menghasilkan output berupa nilai akurasi dan ROC.

3.5 PEMBOBOTAN

Pada penelitian, pembobotan yang digunakan adalah TF-IDF. Untuk perhitungan bobot yang digunakan adalah sebagai berikut :

Term Frequency - Inverse Document Frequency (TF-IDF)

Ni (d) = dfi . Log D/dfi.

Dimana :

Dfi adalah banyaknya dokumen yang mengandung fitur i (kata) yang dicari.

D adalah jumlah dokumen.

Pada proses pemodelan algoritma SVM, terdapat pengoptimalan model algoritma SVM menjadi algoritma SVM Optimized. Pada algoritma ini parameter terbaiknya adalah sebagai berikut :

```
Best parameters are:
{'C': 1, 'gamma': 'scale', 'kernel': 'linear'}
```

Gambar 13. Parameter terbaik dari SVM Optimized dengan TF- IDF.

3.6 VALIDASI DAN EVALUASI

Kami menggunakan validasi dan evaluasi dengan menggunakan AUC ROC Curve. *Area Under the Curve* (AUC) *of Receiver Characteristic Operator* (ROC). Kurva Receiver Operator Characteristic (ROC) adalah matrik evaluasi untuk masalah klasifikasi biner. Ini adalah kurva probabilitas yang memplot TPR terhadap FPR pada berbagai nilai ambang batas dan pada dasarnya memisahkan 'sinyal' dari 'noise'. Area Under the Curve (AUC) adalah ukuran kemampuan *classifier* untuk membedakan antar kelas dan digunakan sebagai ringkasan dari kurva ROC. Jadi semakin besar AUC-nya maka semakin bagus juga hasil akurasinya,

Dalam ROC-AUC terdapat kurva TPR (*True Negative Rate*) dan FNR (*False Negative Rate*). Sensitivitas / Tingkat Positif Benar / Ingat :

$$Sensitivity = \frac{TP}{TP + FN}$$

Sensitivitas memberi tahu kita berapa proporsi kelas positif yang diklasifikasikan dengan benar.

Contoh sederhana adalah menentukan berapa proporsi kata bersentimen positif yang sebenarnya terdeteksi dengan benar oleh model.

Spesifisitas / Tingkat Negatif Benar :

$$Specificity = \frac{TN}{TN + FP}$$

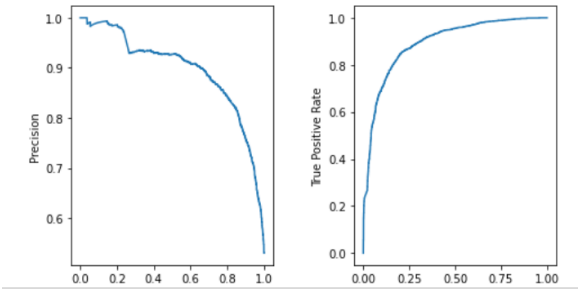
Spesifisitas memberi tahu kita berapa proporsi kelas negatif yang diklasifikasikan dengan benar. Mengambil contoh yang sama seperti dalam Sensitivitas, Spesifisitas berarti menentukan proporsi kata bersentimen positif yang diidentifikasi dengan benar oleh model.

Contohnya adalah sebagai berikut :

```
Train accuracy score:  0.9722946544980443
Test accuracy score:  0.8203373258372036

Train ROC-AUC score:  0.9871867188069052
Test ROC-AUC score:  0.8928635035720545

Are under Precision-Recall curve: 0.8263642806520197
Area under ROC-AUC: 0.8966119488325524
```



Gambar 14. Contoh hasil pemodelan.

Pada hasil di atas dapat dilihat hasil ROC-AUC dari *train* dan *test* lebih besar dari akurasinya maka hasilnya bagus.

3.7 Random Forest

Random Forest adalah algoritma dalam machine learning yang digunakan untuk pengklasifikasian data set dalam jumlah besar. Karena fungsinya bisa digunakan untuk banyak dimensi dengan berbagai skala dan performa yang tinggi. Klasifikasi ini dilakukan melalui penggabungan tree dalam decision tree dengan cara training dataset yang dimiliki.

Random Forest bekerja dengan membangun beberapa decision tree dan menggabungkannya demi mendapatkan prediksi yang lebih stabil dan akurat. 'Hutan' yang dibangun oleh Random Forest adalah kumpulan decision tree dimana biasanya dilatih dengan metode bagging. Ide umum dari metode bagging adalah kombinasi model pembelajaran untuk meningkatkan hasil keseluruhan. Algoritma Random Forest meningkatkan keacakan pada model sambil menumbuhkan tree.

3.8 Multinomial Naive Bayes

Multinomial Naive Bayes adalah salah satu tipe metode Naive Bayes yang digunakan untuk mengklasifikasi kategori dokumen. Sebuah dokumen dapat dikategorikan bertema olahraga, politik, teknologi, atau lain-lain berdasarkan frekuensi kata-kata yang muncul dalam dokumen.

3.9 Bagged Multional Naive Bayes

Pada Bagged Multional Naive Bayes algoritma program dapat menebak tag teks, seperti email atau cerita surat kabar, menggunakan teorema Bayes. Ini menghitung kemungkinan setiap tag untuk sampel yang diberikan dan menampilkan tag dengan peluang terbesar.

3.10 Gaussian Naive Bayes

Gaussian Naive Bayes adalah asumsi pendistribusian nilai kontinu yang terkait dengan setiap fitur berisi nilai numerik. Ketika diplot, akan muncul kurva berbentuk lonceng yang simetris tentang rata-rata nilai fitur.

3.11 SVM

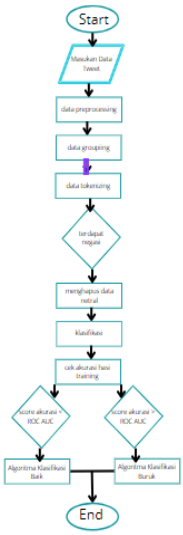
Support Vector Machine (SVM) merupakan salah satu metode dalam *Supervised Learning* yang biasanya digunakan untuk klasifikasi (seperti *Support Vector Classification*) dan regresi (*Support Vector Regression*). Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya. SVM juga dapat mengatasi masalah klasifikasi dan regresi dengan *linier* maupun *non linear*.

3.12 SVM Optimized

Support Vector Machine (SVM) yang dioptimalkan berdasarkan Particle Swarm Optimization (PSO) diperkenalkan dalam memperkirakan harga cryptocurrency di masa depan. Ini adalah bagian dari Artificial Intelligence (AI) yang menggunakan pengalaman sebelumnya untuk memperkirakan harga di masa depan.

4. SYSTEM DESIGN DAN OVERVIEW

4.1 *System Overview*



Gambar 15. *System Design.*

4.2 Data Twitter

	textID	text	selected_text	sentiment
0	cb774db0d1	I'd have responded, if I were going	I'd have responded, if I were going	neutral
1	549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative
2	088c60f138	my boss is bullying me...	bullying me	negative
3	9642c003ef	what interview! leave me alone	leave me alone	negative
4	358bd9e861	Sons of ****, why couldn`t they put them on t...	Sons of ****,	negative
5	28b57f3990	http://www.dothebouncy.com/smf - some shameles...	http://www.dothebouncy.com/smf - some shameles...	neutral
6	6e0c6d75b1	2am feedings for the baby are fun when he is a...	fun	positive
7	50e14c0bb8	Soooo high	Soooo high	neutral
8	e050245fbd	Both of you	Both of you	neutral
9	fc2cbefa9d	Journey!? Wow... u just became cooler. hehe....	Wow... u just became cooler.	positive

Gambar 16. Contoh Data *Twitter*.

Pada gambar diatas adalah contoh dari data *Twitter* yang di dapatkan melalui Kaggle. Dimana terdappat textID, text, selected_text (inti kata dari text) dan sentiment.

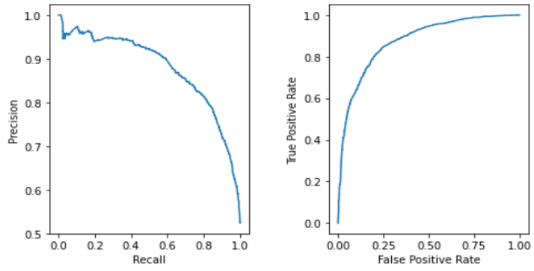
5. HASIL

5.1 VISUALISASI RANDOM FOREST

```
Train accuracy score:  0.9898956975228161
Test accuracy score:  0.8000488878024933

Train ROC-AUC score:  0.999289390175195
Test ROC-AUC score:  0.8763346729135726

Are under Precision-Recall curve: 0.8111726685133887
Area under ROC-AUC: 0.8792233725585481
```



Gambar 17. Visualisasi Random Forest,

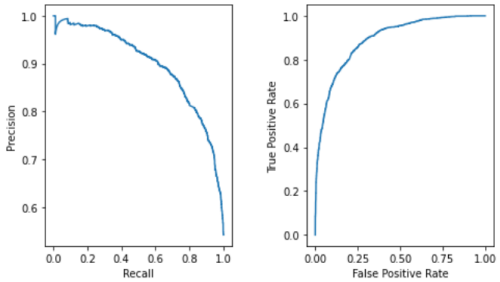
Train Accuracy Score : 0.98, Test Accuracy Score : 0.80, Train ROC-AUC Score : 0.99, Test ROC-AUC Score : 0.87, Area under Precision-Recall Curve : 0.81, dan Area Under ROC-AUC : 0.87.

5. 2 VISUALISASI MULTINOMIAL NAIVE BAYES

```
Train accuracy score:  0.9095501955671447
Test accuracy score:  0.8044487900268883

Train ROC-AUC score:  0.9688667940477889
Test ROC-AUC score:  0.8916858650825341

Are under Precision-Recall curve: 0.8227735932653967
Area under ROC-AUC: 0.8978840252081584
```



Gambar 18. Visualisasi Multinomial Naive Bayes.

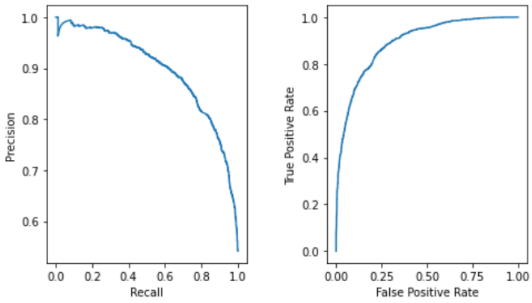
Train Accuracy Score : 0.90, Test Accuracy Score : 0.80, Train ROC-AUC Score : 0.96, Test ROC-AUC Score : 0.89, Area under Precision-Recall Curve : 0.82, dan Area Under ROC-AUC : 0.89.

5.3 VISUALISASI BAGGED MULTIONAL NAÏVE BAYES

```
Train accuracy score:  0.9059647979139505
Test accuracy score:  0.8054265460767539

Train ROC-AUC score:  0.9658887975853248
Test ROC-AUC score:  0.8918129767885886

Are under Precision-Recall curve: 0.8236597252990696
Area under ROC-AUC: 0.8980365588349125
```



Gambar 19. Visualisasi Bagged Multional Naïve Bayes

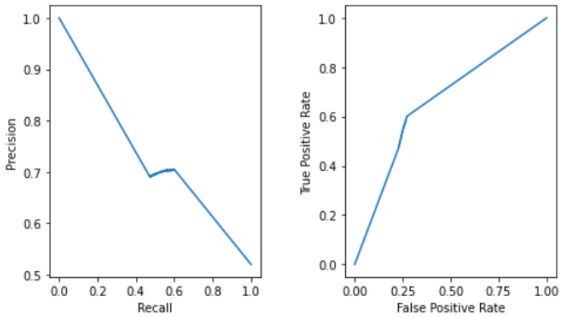
Train Accuracy Score : 0.90, Test Accuracy Score : 0.80, Train ROC-AUC Score : 0.96, Test ROC-AUC Score : 0.89, Area under Precision-Recall Curve : 0.82, dan Area Under ROC-AUC : 0.89.

5.4 VISUALISASI GAUSSIAN NAIVE BAYES

```
Train accuracy score:  0.8825782268578879
Test accuracy score:  0.6169640674651674

Train ROC-AUC score:  0.926081051731836
Test ROC-AUC score:  0.6599290281005442

Are under Precision-Recall curve: 0.5638741998330087
Area under ROC-AUC: 0.7336742557531714
```



Gambar 20. Visualisasi Gaussian Naïve Bayes.

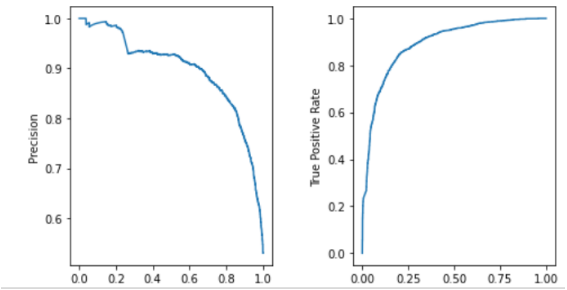
Train Accuracy Score : 0.88, Test Accuracy Score : 0.61, Train ROC-AUC Score : 0.92, Test ROC-AUC Score : 0.65, Area under Precision-Recall Curve : 0.56, dan Area Under ROC-AUC : 0.73.

5.5 VISUALISASI SVM

```
Train accuracy score:  0.9722946544980443
Test accuracy score:  0.8203373258372036

Train ROC-AUC score:  0.9871867188069052
Test ROC-AUC score:  0.8928635035720545

Are under Precision-Recall curve: 0.8263642806520197
Area under ROC-AUC: 0.8966119488325524
```



Gambar 21. Visualisasi SVM.

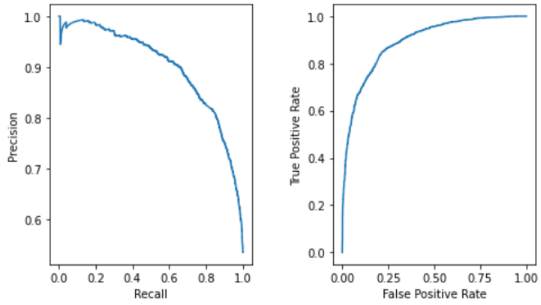
Train Accuracy Score : 0.97, Test Accuracy Score : 0.82, Train ROC-AUC Score : 0.98, Test ROC-AUC Score : 0.89, Area under Precision-Recall Curve : 0.82, dan Area Under ROC-AUC : 0.89.

5.6 VISUALISASI SVM OPTIMIZED

```
Train accuracy score:  0.9235658409387223
Test accuracy score:  0.8164263016377414

Train ROC-AUC score:  0.9680741308595351
Test ROC-AUC score:  0.8932632710845047

Are under Precision-Recall curve: 0.8265989378896329
Area under ROC-AUC: 0.9005616580780089
```



Gambar 22. Visualisasi SVM Optimized.

Train Accuracy Score : 0.92, Test Accuracy Score : 0.81, Train ROC-AUC Score : 0.96, Test ROC-AUC Score : 0.89, Area under Precision-Recall Curve : 0.82, dan Area Under ROC-AUC : 0.90.

5.7 HASIL EVALUASI *TESTING*

Model Name	Train Accuracy	Test Accuracy	Train ROC	Test ROC
Random Forest	0.989896	0.800049	0.999289	0.876335
MultinomialNB	0.909550	0.804449	0.968867	0.891686
Bagged MultinomialNB	0.905965	0.805427	0.965889	0.891813
Gaussian Naive Bayes	0.882578	0.616964	0.926081	0.659929
SVM	0.972295	0.820337	0.987187	0.892864
SVM Optimized	0.923566	0.816426	0.968074	0.893263

Gambar 23. Hasil Evaluasi *Testing.*

Dari hasil pemodelan klasifikasi dengan menggunakan 6 algoritma yang berbeda, hasil yang paling bagus untuk akurasi *train*-nya adalah menggunakan algoritma Random Forest dengan nilai akurasinya 0.98. Untuk nilai akurasi *test* yang paling bagus adalah menggunakan algoritma SVM dengan nilai akurasinya 0.82. ROC sendiri adalah infografis yang menggambarkan kemampuan diagnosa model biner dengan cara membuat plot di antara FPR (*False Positive Rate*) dan TPR (*True Positive Rate*) untuk setiap threshold. Nilai ROC paling besar untuk *train* pada model algoritma Random Forest dengan nilai ROC 0.99, sedangkan untuk *train*-nya nilai ROC paling besar terletak pada model algoritma SVM Optimized dengan nilai ROC 0.893.

6. KESIMPULAN

Setelah dilakukan pemodelan dengan 6 algoritma yang berbeda yaitu algoritma Random Forest, Multinomial Naive Bayes, Bagged Multional Naive Bayes, Gaussian Naive Bayes, SVM dan SVM Optimized akurasi yang paling bagus pada *train* terdapat pada algoritma Random Forest, untuk *test* terdapat pada algoritma SVM. Sedangkan untuk ROC-nya, ROC paling besar pada *train* terletak di algoritma Random Forest dan untuk *test*-nya terletak di algoritma SVM Optimized.

7. REFERENSI

[1] Adminlp2m. 2022. Analisis Sentimen (Sentiment Analysis) : Definisi, Tipe dan Cara Kerjanya. https://lp2m.uma.ac.id/2022/02/21/analisis-sentimen-sentiment-analysis-definisi-tipe-dan-cara-kerjanya/#:~:text=Sentiment%20analysis%20adalah%20proses%20penggunaan,miliaran%20data%20dari%20berbagai%20sumber , diakses 8 Juli 2022.

[2] Algoritma, 2022. Tujuan & Contoh Penerapan Naive Bayes. https://algorit.ma/blog/naive-bayes-2022/ , diakses 8 Juli 2022.

[3] Algoritma, 2022. Cara Kerja Algoritma Random Forest. https://algorit.ma/blog/cara-kerja-algoritma-random-forest-2022/ , dikases 8 Juli 2022.

[4] Arifin, Dian Rudi. 2022. Pengertian Twitter Beserta Sejarah, Fitur, Fungsi, Manfaat, dll. https://dianisa.com/pengertian-twitter/ , diakses 6 Juli 2022.

[5] Bhandari, Aniruddha. 2020. AUC-ROC Curve in Machine Learning Clearly Explained. https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/ , diakses 8 Juli 2022.

[6] Heronius, Albertus. 2019. Twitter Sentiment Analysis Bahasa Indonesia dengan TextBlob. https://medium.com/@albertusheronius/twitter-sentiment-analysis-bahasa-indonesia-dengan-textblob-f34e1ffdcdaa , diakses 6 Juli 2022.

[7] Samsudiney, 2019. Penjelasan sederhana tentang apa itu svm. https://medium.com/@samsudiney/penjelasan-sederhana-tentang-apa-itu-svm-149fec72bd02 , diakses 8 Juli 2022.

[8] ScienceDirect, 2019. An Optimized Support Vector Machine (SVM) based on Particle Swarm Optimization (PSO) for Cryptocurrency Forecasting. An Optimized Support Vector Machine (SVM) based on Particle Swarm Optimization (PSO) for Cryptocurrency Forecasting - ScienceDirect , diakses 8 Juli 2022.

[9] Tutorial, Python. 2022. NLTK stop words. https://pythonspot.com/nltk-stop-words/ , diakses 8 Juli 2022.