

**KLASIFIKASI TWEET INFLUENCER NU DENGAN GNPF-ULAMA
MENGUNAKAN NAIVE BAYES DAN SUPPORT VECTOR MACHINE****CLASSIFICATION OF NU INFLUENCER TWEET WITH GNPF-ULAMA
USING NAIVE BAYES AND SUPPORT VECTOR MACHINE****Andi Taufik¹⁾, Robi Sopandi²⁾**¹Program Studi Sistem Informasi, Universitas Nusa Mandiri²Program Studi Teknik Informatika, Universitas Nusa Mandiri¹Email: robi.rbs@nusamandiri.ac.id**ABSTRAK**

Legitimasi sebagai kelompok yang paling mewakili umat terus menjadi hal yang diperebutkan oleh kelompok Islam saat ini. Di satu sisi ada Nahdlatul Ulama (NU) dengan wacana Islam dan nasionalisme, di sisi lain ada Gerakan Nasional Pembela Fatwa Ulama (GNPF-Ulama) dengan wacana yang hanya menekankan pada aspek Islam. Pertarungan wacana kedua kelompok besar ini termasuk di media sosial *Twitter* sontak menimbulkan kebingungan pada umat Islam terutama yang awam, ditambah lagi banyaknya *Buzzer* dan *Influencer* yang berafiliasi kepada masing-masing kelompok. Penelitian ini akan melakukan klasifikasi *tweet* dari *Influencer* yang berafiliasi dengan NU dan GNPF-Ulama di media sosial *Twitter*. Algoritma yang digunakan adalah *Naive Bayes* dan *Support Vector Machine*. Data *Twitter* diambil menggunakan *library tweepy*, proses *preprocessing* menggunakan *Python* dengan penggunaan *Library Sastrawi* untuk melakukan *stemming* kata bahasa Indonesia. Klasifikasi dengan metode *Naive Bayes* dan *Support Vector Machine* dilakukan menggunakan aplikasi *RapidMiner*. Dari rangkaian proses yang dilakukan, metode *Support Vector Machine* menghasilkan nilai akurasi dan AUC yang lebih baik dari *Naive Bayes* yakni sebesar 77.28% dan AUC sebesar 0.863.

Kata kunci: Klasifikasi, Nahdlatul ulama, GNPF-Ulama, *Naive bayes*, *Support vector machine*

ABSTRACT

Legitimacy as the group that best represents the ummah continues to be contested by Islamic groups today. On the one hand there is the Nahdlatul Ulama (NU) with the discourse of Islam and nationalism, on the other hand there is Gerakan Nasional Pembela Fatwa Ulama (GNPF-Ulama) with a discourse that only emphasizes the Islamic aspect. The battle between the two major groups, including on social media, Twitter, suddenly caused confusion among Muslims, especially the layman, plus the large number of Buzzers and Influencers affiliated with each group. This research will classify tweets from influencers affiliated with NU and GNPF-Ulama on twitter. The algorithm used is Naive Bayes and Support Vector Machine. Twitter data is taken using the Tweepy library, the preprocessing process uses python with the use of the Sastrawi library to stem Indonesian words. Classification using the Naive Bayes method and Support Vector Machine is done using the RapidMiner application. From the circuit process carried out, the Support Vector Machine method produced better accuracy and AUC values than Naive Bayes, namely 77.28% and AUC of 0.863.

Keywords: *Classification, Naive bayes, Support vector machine, Nahdlatul ulama, GNPF-Ulama*

PENDAHULUAN

Populisme Islam belakangan ini merupakan salah satu topik sentral dalam kajian politik di Indonesia. Legitimasi sebagai kelompok yang paling mewakili umat menjadi hal yang diperebutkan oleh kelompok Islam saat ini. Pertarungan wacana khususnya yang terjadi antara kelompok GNPF-Ulama yang dimotori oleh FPI dengan kelompok NU sebagai organisasi Islam terbesar di Indonesia terus terjadi di berbagai tempat dan media termasuk media sosial (Jayanto, 2019).

Nahdlatul Ulama (NU) atau dalam bahasa Indonesia artinya “Kebangkitan Para Ulama” merupakan ormas Islam terbesar di Indonesia saat ini. Sebagai organisasi yang memiliki nasionalisme sebagai salah satu motif berdirinya, militansi dan semangat kelompok NU dalam membela negara sangat tinggi. NU aktif mensosialisasikan NKRI sebagai bentuk final dari sistem kebangsaan di Indonesia. Sehingga dalam pergerakannya saat ini, NU aktif melawan gerakan separatisme, radikalisme, dan konflik antar ras karena dianggap bisa menimbulkan disintegrasi dan memecah belah bangsa (Farih, 2016).

Berbeda dengan NU menggunakan dimensi keagamaan dan nasionalisme dalam wacananya, Gerakan Nasional Pembela Fatwa Ulama (GNPF-Ulama) merupakan gerakan yang menekankan aspek wacana keagamaan sehingga membentuk legitimasi populisme Islam. Gerakan yang dimotori oleh Front Pembela Islam (FPI) dengan Habib Rizieq sebagai tokohnya ini membingkai aktivitas dan gerakan berbasiskan pada identitas kesamaan agama. Hal ini tidak terlepas dari kegeraman atas janji-janji penguasa yang tidak kunjung terwujud sehingga menjadi alternatif untuk menyelesaikan persoalan dengan merebut sumber daya dan kekuasaan dalam negeri (Jayanto, 2019).

Saat ini, baik NU dan GNPF-Ulama terus melakukan kontestasi dalam memperebutkan wacana sebagai perwakilan Islam di Indonesia dalam bentuk narasi yang berbeda. NU dengan Islam dan nasionalisme sebagai wacana sedangkan GNPF-Ulama hanya menekankan pada aspek Islam (Jayanto, 2019). Salah satu contoh pertarungan wacana antara kedua kelompok ini adalah ketika pada tahun pemilihan Presiden 2019, ketua umum NU yaitu KH. Ma'ruf Amin

yang mencalonkan diri sebagai wakil presiden berhadapan langsung dengan Prabowo yang didukung langsung oleh GNPF-Ulama melalui Ijtima Ulama mereka. Pertarungan wacana kedua kelompok besar ini termasuk di media sosial sontak menimbulkan kebingungan pada umat Islam terutama yang awam, ditambah lagi banyaknya *Buzzer* dan *Influencer* yang berafiliasi kepada masing-masing kelompok meski tidak menampakkan keterkaitannya secara langsung.

Meskipun demikian, kita dapat menggunakan data dari salah satu media sosial yang digunakan yaitu *Twitter* sebagai bahan untuk mengklasifikasikan *tweet* mereka menggunakan teknik data mining sehingga bisa terlihat afiliasinya meskipun tidak disebut secara langsung. Metode yang digunakan pada penelitian ini adalah *Naive Bayes* dan *Support Vector Machine*. Penelitian ini akan membandingkan akurasi pada penggunaan metode *Naive Bayes* dan *Support Vector Machine* setelah sebelumnya data dibersihkan melalui tahap *preprocessing* terlebih dahulu. Kedua metode ini sudah banyak digunakan oleh para peneliti terutama pada data yang berasal dari *Twitter* untuk analisa sentimen dengan mengelompokkan teks apakah bermuatan positif atau negatif.

Selain untuk analisa sentimen, penggunaan kedua metode untuk klasifikasi *tweet* juga pernah dilakukan seperti yang dilakukan oleh Baydhowi, Apriliah, dan Kurniawan yang menggunakan metode *Naive Bayes* untuk mengklasifikasikan *tweet* berdasarkan keterkaitan *tweet* terhadap topik retail, proyek, dan pendidikan. Hasil dari percobaan yang dilakukan memiliki tingkat lebih dari 40% sehingga akurasi yang didapat termasuk ke dalam kategori kurang baik (Baydhowi et al, 2019).

Metode *Support Vector Machine* juga banyak digunakan untuk klasifikasi *tweet* seperti yang dilakukan oleh Purnamasari dkk pada penelitian yang menggunakan metode *Support Vector Machine* untuk mengklasifikasikan *tweet* yang mengandung konten *bullying*. Hasil akurasi yang didapatkan dengan metode *Support Vector Machine* adalah 75% (Purnamasari et al, 2018).

Dengan adanya variasi akurasi yang didapat dari kedua metode ini, mendorong penulis untuk melakukan penelitian dengan membandingkan metode *Naive Bayes* dan *Support Vector Machine*. Penelitian ini akan membahas tahapan

yang dilalui untuk melakukan klasifikasi *tweet* dari *Influencer* NU dan GNPf-Ulama melalui media sosial *Twitter*. Proses dimulai dari tahap *preprocessing* menggunakan Python sampai tahap klasifikasi dengan *Naive Bayes* dan *Support Vector Machine* menggunakan aplikasi RapidMiner.

METODE PENELITIAN

Klasifikasi Bayes

Klasifikasi *Bayes* merupakan metode yang dikembangkan dari teorema *Bayes*. Pada penerapan suatu model, kadangkala kita sudah memiliki *prior belief* (keyakinan awal) pada suatu parameter tentang kemungkinan nilainya. Dalam metode *Bayes*, untuk mengestimasi parameter keyakinan awal dimasukkan ke dalam perhitungan. Peluang pada teorema *Bayes* dinyatakan sebagai berikut:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Di mana **H** adalah hipotesis, **X** adalah bukti, **P(H|X)** adalah probabilitas posterior H dengan syarat **X** atau dengan kata lain peluang hipotesis **H** benar untuk bukti **X**. **P(X|H)** adalah probabilitas posterior **X** dengan syarat **H** atau peluang bukti **X** benar untuk hipotesis **H**, **P(X)** adalah probabilitas *prior* bukti **X** dan **P(H)** adalah probabilitas *prior* hipotesis **H** (Suyanto, 2019). Di antara kelebihan pada klasifikasi *Naive Bayes* antara lain:

1. Mudah dipahami
2. Mampu menangani data diskrit dan kuantitatif
3. Hanya memerlukan sedikit data latih untuk estimasi parameter yang dibutuhkan dalam klasifikasi
4. Kokoh untuk titik *noise* yang diisolasi dan atribut yang tidak relevan (Jollyta *et al*, 2020).

Support Vector Machine

SVM berkaitan dengan memaksimalkan jarak pemisah antar kelas data (*margin*). Mencari *hyperplane* terbaik dengan cara memisahkan dua kelas data merupakan konsep dasar dari SVM.

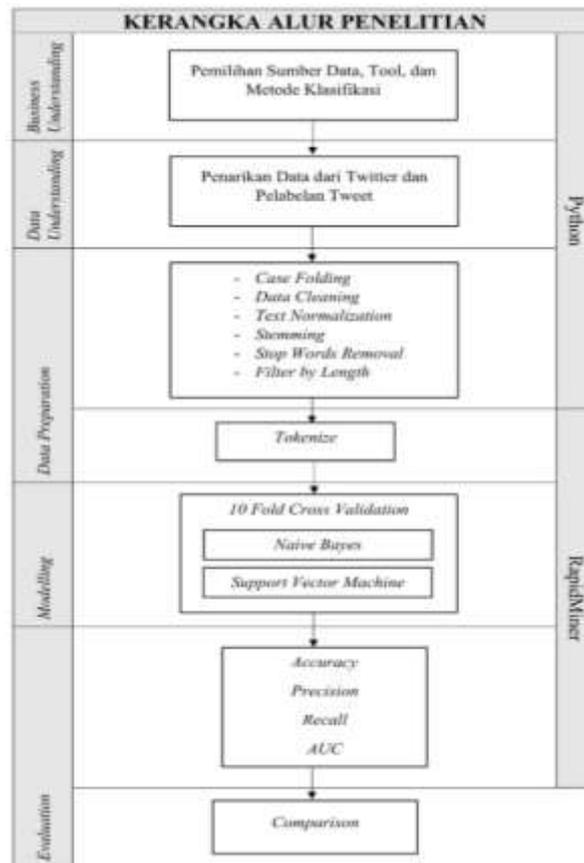
Sebuah *hyperplane* yang paling baik adalah yang mampu mengklasifikasikan kedua kelas dengan jarak yang paling lebar. Sehingga pada kedua gambar di atas, *Hyperplane 1* lebih baik karena memiliki margin yang lebih besar. Adapun yang disebut *Support Vector* adalah objek-objek data terluar yang paling dekat dengan *hyperplane*. Karena posisinya yang hampir tumpang tindih, *support vector* inilah yang diperhitungkan untuk menemukan *hyperplane* yang paling optimal. Kelebihan dari metode SVM antara lain:

1. Relatif mudah diimplementasikan.
2. Kemampuan generalisasi yang tinggi terhadap data.
3. Meskipun dilatih dengan himpunan data yang relatif sedikit, SVM dapat menghasilkan model klasifikasi yang baik (Suyanto, 2019).

Tahapan Penelitian

Penelitian ini terdiri dari 5 tahapan yang mengacu pada metodologi CRISP-DM (*Cross Industry Standard Process for Data Mining*). Pada ada metodologi CRISP-DM terdapat 6 tahapan yaitu, *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modelling*, *Evaluation*, dan *Deployment*. Namun pada penelitian ini, penulis tidak melakukan proses *Deployment*.

Berikut ini adalah kerangka alur pada penelitian ini:



Gambar 1. Kerangka Alur Penelitian

HASIL DAN PEMBAHASAN

1. *Business Understanding*

Tahapan pada penelitian ini dimulai dengan *business understanding*. Pada tahap ini, dilakukan pemahaman terhadap objek yang diteliti. Dimulai dari mengenali akun yang cocok dijadikan sampel untuk diambil *tweetnya*, yakni akun yang memiliki basis pengikut tinggi, status pemiliknya yang merupakan tokoh maupun pengurus dari kelompok NU ataupun GNPf-Ulama.

Pada tahap ini juga dilakukan penentuan algoritma yang digunakan untuk melakukan klasifikasi. Karena proses pada penelitian ini adalah mengklasifikasikan *tweet* dari dua kelompok yang berbeda. Maka dipilih algoritma yang masuk ke dalam kategori algoritma untuk melakukan klasifikasi yaitu *Naive Bayes* dan *Support Vector Machine*. Termasuk juga pada fase ini menentukan *tool* yang akan digunakan untuk mempersiapkan data dan pemodelan.

2. Data Understanding

Pada tahap ini, dilakukan pengambilan yang dibutuhkan untuk penelitian. Data diambil menggunakan *library tweepy* dengan bahasa pemrograman Python.

```
import tweepy #import library tweepy

#memasukkan kredensial yang dari akun developer
api_key = "JW0VvD1ZuSublPPKnZT6fJXwZ"
api_secret_key = "ul30DTglTlrRMDKp6Ni10iEFotRvfb5Qwg9eBko0V9oTC8x35E"
access_token = "1324217910747828225-EDQB5EkjEtbp4tEWFLn0qTftF1dkb4"
access_token_secret = "K9GUg91ulfjaLnYd8KojKI8xiJEAqqJjArx4M8nMY39da"

#memanggil api
auth = tweepy.OAuthHandler(api_key, api_secret_key)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
```

Gambar 2. Mengimpor *Library Tweepy*

Sebelum bisa digunakan, sebuah *library* harus diimpor terlebih dahulu. Selanjutnya kode kredensial yang sudah didapatkan melalui *developer.twitter.com* dimasukkan agar kemudian bisa menggunakan *twitter api* untuk mendapatkan data *tweet*.

```
ListTweet = []
tokoh = ['haikal_hassan', 'ustadtengkuzul', 'na_dirs', 'ulil'] #username yang dipilih
for orang in tokoh:
    hasilUser = api.user_timeline(id=orang, lang="id", count=100, tweet_mode='extended')
    for tweet in hasilUser:
        tweet = ' '.join(tweet.full_text.split('\n'))
        ListTweet.append(tweet)
```

Gambar 3. Mengambil Data *Tweet*

Untuk mengambil pertama ditentukan *username* yang ingin diambil data *tweetnya*. Selanjutnya dilakukan *looping* untuk pengambilan data pada masing-masing *username*. Data yang didapat bervariasi mulai dari *id*, *created_at*, *url*, *location*, *user_mentions*, *tweet*, dan lain sebagainya. Karena pada penelitian ini hanya menggunakan *tweet* maka data yang diambil hanya data *full_text* untuk menampilkan teks penuh dari *tweet* yang diambil. Setiap *tweet* yang didapat lalu dimasukkan ke dalam variabel *ListTweet* untuk kemudian diekspor ke dalam format *excel*.

```
import pandas as pd
pd.DataFrame(ListTweet).to_excel('nu_gnpf.xlsx', index=False)
```

Gambar 4. Ekspor *Tweet* ke dalam Bentuk *Excel*.

Untuk mengekspor kumpulan *tweet* pada variabel *ListTweet* ke dalam bentuk *excel*, penulis menggunakan *library* bernama *pandas*. Variabel *ListTweet* dikonversi terlebih dahulu ke dalam bentuk *DataFrame* untuk kemudian diekspor ke dalam bentuk *excel*. Total data yang diambil sejumlah 400 *tweet* dalam rentang tanggal 23 hingga 30 November 2020, masing-masing 200 *tweet* dari kelompok NU dan 200 *tweet* dari kelompok GNPF-Ulama dengan rincian 100 *tweet* untuk tiap *username*. Setelah diekspor ke dalam bentuk *excel*, setiap *tweet* lalu dilabeli secara manual untuk kemudian dilakukan *preprocessing* pada tahap selanjutnya.

Tabel 1. Contoh Data *Tweet* yang diambil

Label	<i>Tweet</i>
GNPF-U	Heran!!! Sekedar general check-up diframming sedemikian rupa terpapar Covid19. Lalu harus laporkan hasil lah Lalu gak ikut prosedur lah Lalu dituduh kabur lah Bangsa ini terlalu berani menyudutkan, memojokkan, mencari2 kesalahan, menyusahkan ulama habaib. Tunggu!
GNPF-U	Semua masalah negara, korupsi, omnibus law, ciptakerja, partai, pertumbuhan ekonomi minus, hutang ugal2an, reklamasi, dll semua lenyap, tenggelam berganti dengan RS.UMMI Bogor. Sampai kapan begini?
NU	Sedikit tentang Kiai Miftachul Akhyar yg didaulat sebagai Ketum MUI yg baru. Beliau adalah Rois Aam PBNU saat ini. Dari segi pandangan keagamaan, beliau adalah sosok konservatif yg moderat, khas para kiai pada umumnya.
NU	Gus Dur itu melampaui standar pemahaman kita ttg sosok Kiai. Sampai sekarang saja masih banyak yang kaget kalau Kiai bicara soal bola, musik, politik, puisi, atau bahkan tik-tok Mereka sangka urusannya Kiai itu cuma soal khutbah dan khitbah hehehhe https://t.co/boZQsTpkIy

3. Data Preparation

Setelah data diambil, tahap selanjutnya adalah melakukan persiapan data. Pada tahap ini dilakukan beberapa langkah yang disebut dengan *text preprocessing*. Tahap ini terdiri dari 7 langkah, 6 tahap dilakukan menggunakan Python dan 1 tahap dilakukan dengan menggunakan RapidMiner.

a. Case Folding

Dalam suatu teks, tidak semua kata konsisten dituliskan dalam bentuk yang sama. Pada tahap ini, teks dikonversi menjadi bentuk standar antara huruf kecil atau huruf besar secara keseluruhan. Proses ini dilakukan menggunakan fungsi bawaan dari Python.

```
def lowercase(text):
    return text.lower()
```

Gambar 5. Merubah Teks menjadi Huruf Kecil

Tabel 2. Sebelum dan Sesudah Proses *Case Folding*

Sebelum	Sesudah
JANGAN POLITIK TERUS.... SEKARANG KITA SEMINAR KHUSUS UTK WANITA AGAR SEGERA DPT PASANGAN YANG JUJUR DAN TG.JAWAB...	jangan politik terus.... sekarang kita seminar khusus utk wanita agar segera dpt pasangan yang jujur dan tg.jawab...

b. Data Cleaning

Setelah semua kata sama rata dari sisi penggunaan huruf besar dan kecil. Tahap selanjutnya adalah membersihkan data dari karakter-karakter yang tidak penting antara lain simbol *retweet* (RT), tanda *mention* kepada pengguna lain (@*username*), URL, spasi berlebih, dan alfanumerik.

```
def remove_unnecessary_char(text):
    text = re.sub('rt ', ' ', text)
    text = re.sub('@[^\s]+', '', text)
    text = re.sub("([A-Za-z0-9+])|([^\0-9A-Za-z \t])|(\w+:\V\/\S+)", " ", text)
    text = re.sub(' +', ' ', text)
    text = re.sub('[^\0-9a-zA-Z]+', ' ', text)
    return text
```

Gambar 6. Membersihkan Data dari Karakter yang Tidak Penting

Tabel 3. Sebelum dan Sesudah Proses *Data Cleaning*

Sebelum	Sesudah
Kenapa babe Haikal minta maaf? Ada di Podcast Deddy Corbuzier https://t.co/J4248gfch4 https://t.co/a2TS3USCR4	Kenapa babe Haikal minta maaf Ada di Podcast Deddy Corbuzier

c. *Text Normalization*

Dalam menulis sesuatu, terkadang seseorang menggunakan cara penulisan yang bermacam-macam baik menggunakan kata tidak baku ataupun singkatan. Pada tahap ini kata dirubah dari bentuk tidak formal menjadi formal. Penelitian ini menggunakan dataset yang dibuat oleh Muhammad Okky Ibrohim dan Indra Budi untuk melakukan normalisasi teks (Ibrohim and Budi, 2019).

```
alay_dict = pd.read_csv('new_kamusalay.csv', encoding='latin-1', header=None)
alay_dict = alay_dict.rename(columns={0: 'original', 1: 'replacement'})
alay_dict_map = dict(zip(alay_dict['original'], alay_dict['replacement']))
def normalize_alay(text):
    return ' '.join([alay_dict_map[word] if word in alay_dict_map else word for word in text.split(' ')])
```

Gambar 7. Normalisasi Teks

Tabel 4. Sebelum dan Sesudah *Text Normalization*

Sebelum	Sesudah
Semua survey <i>tiba2</i> rontok dengan berita menggelegar ini	Semua survei <i>tiba tiba</i> rontok dengan berita menggelegar ini

d. *Stemming*

Pada proses *Stemming*, data ditransformasikan menjadi kata dasarnya. Hal ini dilakukan dengan menghilangkan awalan dan akhiran pada turunan kata. Untuk melakukan proses *stemming*, penelitian ini menggunakan *library* Python Sastrawi yang merupakan pengembangan dari *library* Sastrawi pada bahasa pemrograman PHP.

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()
def stemming(text):
    return stemmer.stem(text)
```

Gambar 8. *Stemming* menggunakan *Library* Sastrawi

Tabel 5. Sebelum dan Sesudah Proses *Stemming*

Sebelum	Sesudah
Jangan <i>bercerai</i> berai... <i>Berbeda</i> itu nggak apa apa	jangan <i>cerai</i> berai <i>beda</i> itu nggak apa apa

e. *Stop Words Removal*

Pada tahap ini kata umum yang seringkali muncul dalam jumlah besar dan tidak memiliki pengaruh besar terhadap makna dihapus. Penelitian ini menggunakan dataset dari *kaggle* bernama *Indonesian Stoplist* (Oswin Rahadiyan Hartono 2016) yang dibuat berdasarkan penelitian oleh Fadillah Tala (Tala, 2003).

```
id_stopword_dict = pd.read_csv('stopwordbahasa.csv', header=None)
id_stopword_dict = id_stopword_dict.rename(columns={0: 'stopword'})
def remove_stopword(text):
    text = ' '.join(['' if word in id_stopword_dict.stopword.values else word for word in text.split(' ')])
    text = re.sub(' +', ' ', text) # Remove extra spaces
    text = text.strip()
    return text
```

Gambar 9. Menghapus *Stop Words*

Tabel 6. Sebelum dan Sesudah *Stop Words Removal*

Sebelum	Sesudah
Siapa yang kasih izin? Siapa yang tetap izinkan? Siapa yang perpanjang izin?	Siapa kasih izin? Siapa izinkan? Siapa perpanjang izin?

f. *Filter by Length*

Ketika menulis di media sosial khususnya *Twitter*, seseorang tidak selalu menulis *tweet* yang berkaitan dengan pemikirannya. Namun terkadang *tweet* yang ditulis hanya berupa balasan singkat terhadap *tweet* orang lain ataupun sekadar *mention* singkat. Pada penelitian ini, data yang memiliki jumlah karakter kurang dari 15 akan dibuang sehingga diolah hanya data yang memiliki panjang lebih dari 15 karakter. Data *tweet* yang difilter adalah data yang sudah melalui serangkaian proses di atas.

```
def preprocess(text): #mengumpulkan proses dalam 1 fungsi
    text = lowercase(text)
    text = remove_unnecessary_char(text)
    text = normalize_alay(text)
    text = stemming(text)
    text = remove_stopword(text)
    return text

data['Tweet_New'] = data['tweet'].apply(preprocess) #menerapkan ke data tweet

data = data[data['Tweet_New'].apply(lambda x: len(x)>=15)] # filter data lebih dari 15 karakter
```

Gambar 10. Menghilangkan *Tweet* yang Kurang dari 15 Karakter

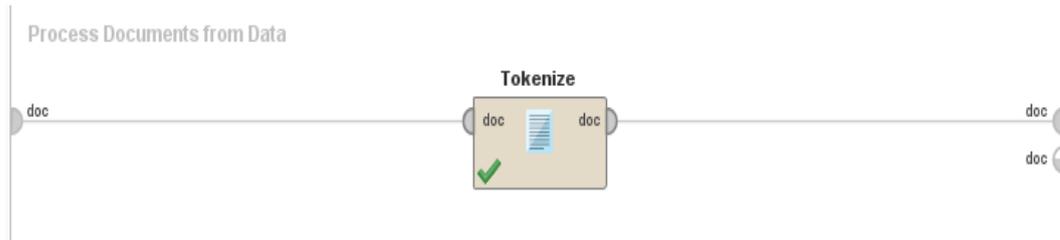
Tabel 7. Contoh Teks yang Dibuang

<i>Preprocessed Tweet</i>	
1	sepakat mas
2	haha
3	terima kasih
4	usul bagus oke
5	iya kalem

g. *Tokenize*

Tokenize adalah proses memecah teks pada suatu kalimat menjadi token-token kata. Sebagai contoh kalimat “Budi bermain bola” akan menghasilkan tiga token yaitu “Budi”, “Bermain”, dan “Bola”. Setelah rangkaian sebelumnya dilakukan menggunakan Python, tahap *tokenize* pada penelitian ini dilakukan

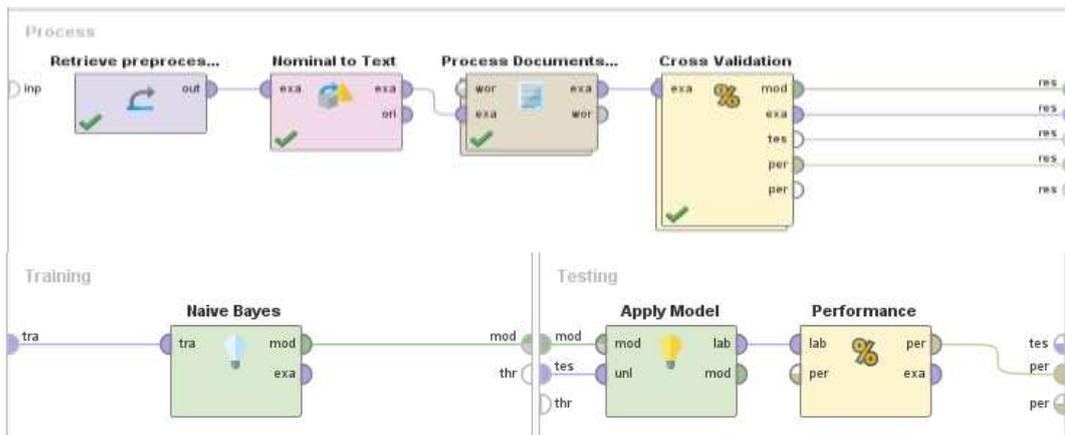
menggunakan ekstensi *text processing* pada aplikasi RapidMiner.



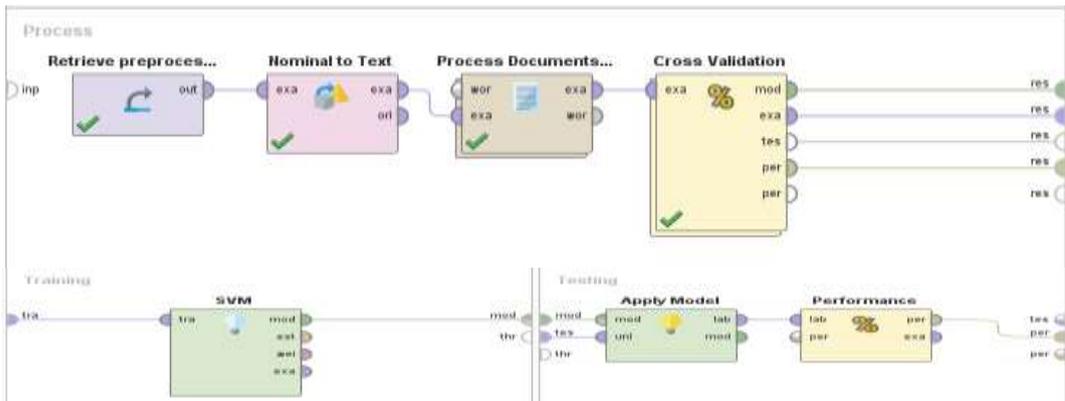
Gambar 11. *Tokenize* menggunakan RapidMiner

4. Modelling

Setelah data melalui proses *preprocessing*, tahap selanjutnya adalah *Modelling*. Tahap ini merupakan fase pemodelan menggunakan algoritma *Naive Bayes* dan *Support Vector Machine*. Proses ini menggunakan metode *10-fold cross validation*. Yang mana dataset dibagi menjadi 10 bagian, kemudian data dipisahkan menjadi dua buah subset, 1 bagian menjadi data uji dan 9 bagian sisanya menjadi data latih. Proses ini diulang ke setiap bagian pada data sehingga diperoleh hasil terbaik dari model yang digunakan. Berikut desain pemodelan yang dilakukan dengan aplikasi RapidMiner:



Gambar 12. *Modelling* menggunakan *Naive Bayes*



Gambar 13. Modelling menggunakan Support Vector Machine

5. Evaluation

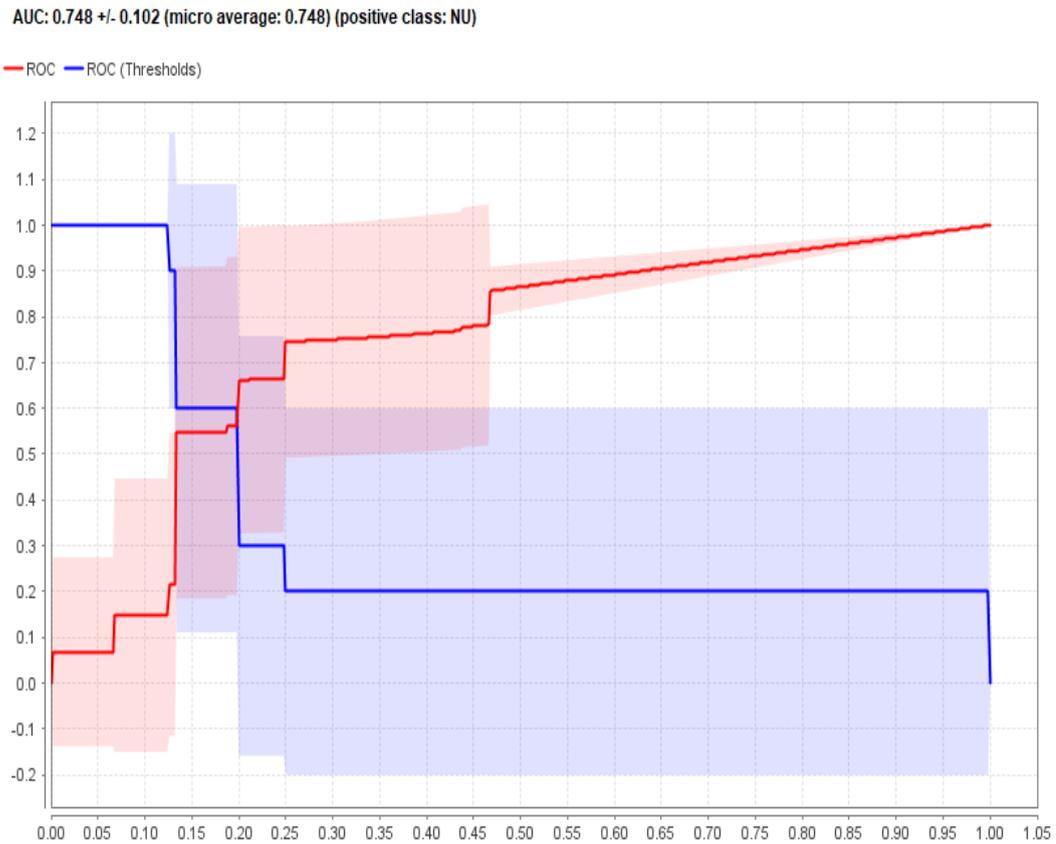
Setelah dilakukan pemodelan, tahap selanjutnya adalah melakukan evaluasi dari model yang telah dimasukkan dengan melihat nilai *AUC*, *Accuracy*, *Precision*, dan *Recall* dari masing-masing metode.

a. Model Naive Bayes

Hasil pengujian menggunakan metode *Naive Bayes* menghasilkan *AUC* sebesar 0.748. Akurasi yang didapatkan adalah 76.92%, *precision* pada prediksi GNPf-Ulama sebesar 73.96% dan *recall* sebesar 82.24%, *precision* pada prediksi NU sebesar 80.58% dan *recall* sebesar 71.79%

Tabel 8. Confusion Matrix Naive Bayes

	true GNPf-U	true NU	class precision
pred. GNPf-U	125	44	73,96%
pred NU	27	112	80,58%
class recall	82,24%	71,79%	



Gambar 14. Kurva ROC Naive Bayes

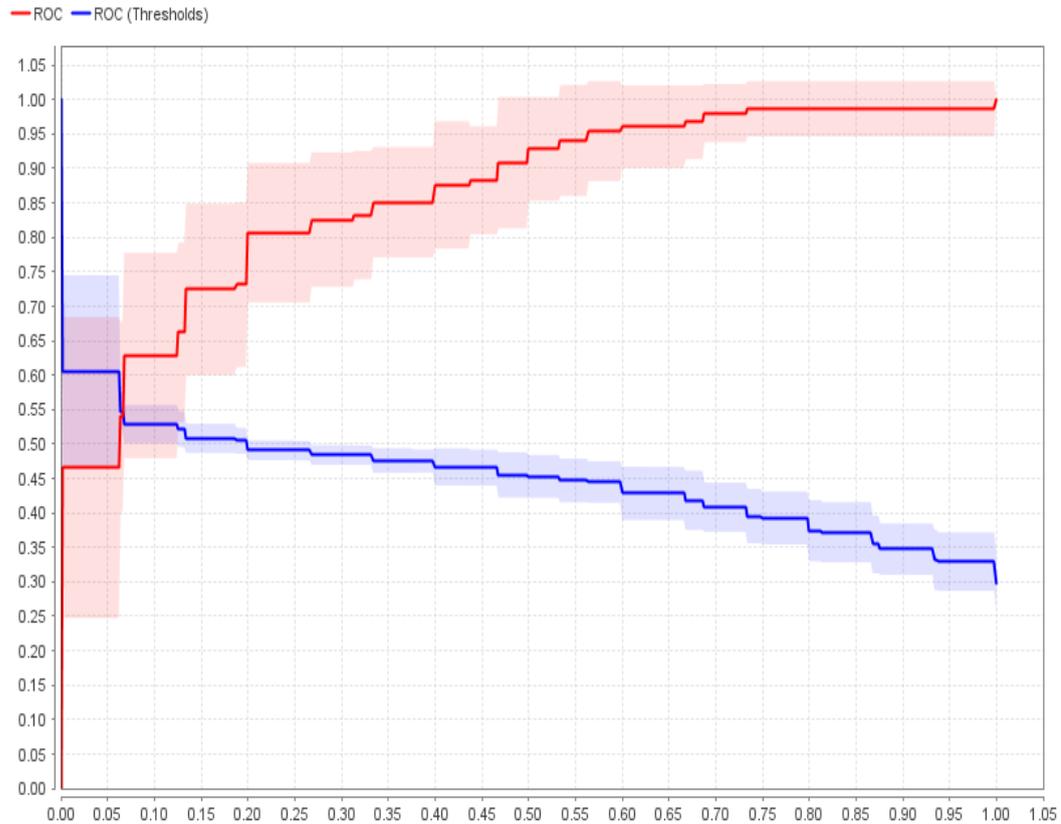
b. Model Support Vector Machine

Hasil pengujian menggunakan metode *Support Vector Machine* menghasilkan *AUC* sebesar 0.863. Akurasi yang didapatkan adalah 77.28%, *precision* pada prediksi GNPF-Ulama sebesar 73.56% dan *recall* sebesar 84.21%, *precision* pada prediksi NU sebesar 82.09% dan *recall* sebesar 70.51% .

Tabel 9. Confussion Matrix Support Vector Machine

	true GNPF-U	true NU	class precision
pred. GNPF-U	128	46	73,56%
pred NU	24	110	82,09%
class recall	84,21%	70,51%	

AUC: 0.863 +/- 0.059 (micro average: 0.863) (positive class: NU)



Gambar 15. Kurva ROC *Support Vector Machine*

Perbandingan Hasil *Naive Bayes* dan *Support Vector Machine*

Berdasarkan hasil pengujian di atas, didapat perbandingan akurasi dan AUC sebagai berikut:

Tabel 10. Komparasi Hasil Pengujian

Algoritma	Accuracy	AUC
<i>Naive Bayes</i>	76.92%,	0.748
<i>Support Vector Machine</i>	77.28%	0.863

Pengujian menggunakan metode *Support Vector Machine* mendapatkan akurasi yang lebih baik dibandingkan metode *Naive Bayes* dengan hasil 77.28% berbanding dengan 76.92%. Nilai AUC pada metode *Support Vector Machine* juga mendapat hasil lebih baik dibanding *Naive Bayes* dengan nilai 0.863 berbanding dengan 0.748, dari nilai tersebut maka performa pada metode *Support*

Vector Machine sudah termasuk ke dalam *good classification*, adapun performa pada metode *naive bayes* termasuk ke dalam *fair classification* (Gorunescu, 2011).

Dari pengujian di atas juga didapatkan hasil *recall* (sensitifitas) pada masing-masing label. *Recall* pada prediksi GNPF-Ulama menjawab pertanyaan “Berapa persen *tweet* yang diprediksi dari kelompok GNPF-Ulama dibandingkan dengan keseluruhan *tweet* dari GNPF-Ulama?”, *recall* pada prediksi NU menjawab pertanyaan “Berapa persen *tweet* yang diprediksi dari kelompok NU dibandingkan dengan keseluruhan *tweet* dari NU?”. Hasilnya adalah *recall* pada prediksi GNPF-Ulama lebih baik dengan nilai 82.24% menggunakan *Naive Bayes* dan 84.21% menggunakan *Support Vector Machine* dibandingkan dengan 71.79% dan 70.51% pada prediksi NU.

Sehingga berdasarkan hasil di atas, dapat disimpulkan bahwa baik metode *Naive Bayes* ataupun *Support Vector Machine* bisa mengklasifikasikan data *tweet* dari kelompok NU dan GNPF-Ulama dengan cukup baik. Namun model pengujian menggunakan metode *Support Vector Machine* memiliki kinerja yang lebih baik dibandingkan model yang menggunakan metode *Naive Bayes*. Dan kinerja kedua metode dalam memprediksi *tweet* dari kelompok GNPF-Ulama lebih baik daripada prediksi *tweet* dari kelompok NU.

PENUTUP

Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai klasifikasi *tweet Influencer* NU dan GNPF-Ulama di media sosial *Twitter*, dengan dataset yang terdiri dari 308 *tweet* setelah melalui rangkaian proses *preprocessing*, metode klasifikasi *naive bayes* mendapatkan akurasi sebesar 76.92% dan AUC sebesar 0.748 (*fair classification*). Sedangkan metode klasifikasi *support vector machine* mendapatkan akurasi sebesar 77.28% dan AUC sebesar 0.863 (*good classification*). Sehingga bisa disimpulkan bahwa model yang menggunakan metode *support vector machine* memiliki kinerja yang lebih baik pada penelitian ini dibandingkan dengan metode *naive bayes*.

Pada penelitian ini juga, berdasarkan nilai *recall* dari masing-masing label, *tweet* dari kelompok GNPf-Ulama bisa diprediksi lebih baik dibandingkan dengan *tweet* dari kelompok NU, dengan nilai 82.24% berbanding dengan 71.79% pada metode *naive bayes* dan 84.21% berbanding dengan 70.51% pada metode *support vector machine*.

Saran

Untuk penelitian selanjutnya, penulis menyarankan untuk menggunakan sampel dataset yang lebih besar agar bisa lebih mewakili populasi dari objek penelitian. Selain itu, masih terdapat metode klasifikasi lain seperti *decision tree* dan *k-nearest neighbor* yang juga bisa digunakan untuk perbandingan lebih lanjut. Berbagai variasi seleksi fitur seperti *Term Frequency*, *TF-IDF*, dan *Term Occurrences* juga bisa digunakan pada penelitian selanjutnya.

DAFTAR PUSTAKA

- Baydhowi, Muhamad, Widya Apriliah, and Ilham Kurniawan. 2019. "Klasifikasi Tweet Berdasarkan Keterkaitan Tweet Terhadap Topik Tertentu Pada Twitter Menggunakan Naïve Bayes." *Information System for Educators and Professionals*, 4(1): 95–103.
- Farih, Amin. 2016. "Nahdlatul Ulama (NU) Dan Kontribusinya Dalam Memperjuangkan Kemerdekaan Dan Mempertahankan Negara Kesatuan Republik Indonesia (NKRI)." *Walisono*, 24(2): 251–84.
- Gorunescu, Florin. 2011. *Data Mining : Concepts, Models and Techniques*. eds. Janusz Kacprzyk and Lakhmi C. Jain. Berlin: Springer.
- Ibrohim, Muhammad Okky, and Indra Budi. 2019. "Multi-Label Hate Speech and Abusive Language Detection in Indonesian Twitter." In *3rd Workshop on Abusive Language Online*, Florence: ALW3, 46–57. <https://www.aclweb.org/anthology/W19-3506.pdf>.
- Jayanto, Dian Dwi. 2019. "Mempertimbangkan Fenomena Populisme Islam Di Indonesia Dalam Perspektif Pertarungan Diskursif: Kontestasi Wacana Politik Antara Gerakan Nasional Pengawal Fatwa Ulama (GNPF-Ulama) Dan Nahdlatul Ulama (NU)." *Jurnal Filsafat* 29(1): 1–25. <https://journal.ugm.ac.id/wisdom/article/view/41131>.
- Jollyta, Deny, William Ramdhan, and Muhammad Zarlis. 2020. *Konsep Data Mining Dan Penerapan*. Sleman: Deepublish Publisher.
- Oswin Rahadiyan Hartono. 2016. "Indonesian Stoplist." [Online]

<https://www.kaggle.com/oswinrh/indonesian-stoplist> [29 November 2020].

Purnamasari, Ni Made Gita Dwi, M Ali Fauzi, Indriarti, and Liana Shinta Dewi. 2018. "Identifikasi Tweet Cyberbullying Pada Aplikasi Twitter Menggunakan Metode Support Vector Machine (SVM) Dan Information Gain (IG) Sebagai Seleksi Fitur." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(11): 5326–32.

Suyanto. 2019. *Data Mining : Untuk Klasifikasi Dan Klasterisasi Data*. Bandung: Informatika.

Tala, F Z. 2003. *Universiteit van Amsterdam The Netherlands "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia."* [Online]. <https://eprints.illc.uva.nl/740/1/MoL-2003-02.text.pdf>.