ANTIVIRUS: Jurnal Ilmiah Teknik Informatika (p – ISSN: 1978 – 5232; e – ISSN: 2527 – 337X)
Vol. 15 No. 2 November 2021, pp. 262 – 276

SVM UNTUK *SENTIMENT ANALYSIS* CALON KEPALA DAERAH BERDASAR DATA KOMENTAR VIDEO DEBAT PILKADA DI YOUTUBE

Muhammad Harris Syafa'at[1)], Eka Rahayu Setyaningsih[2)], Yosi Kristian[3)]
[1,2,3)] *Teknologi Informasi Sekolah Tinggi Teknik Surabaya*
e-mail: muhammadharrissyafaat@gmail.com [1)], eka@stts.edu [2)], yosi@stts.edu [3)]

Abstrak : *YouTube adalah sosial media yang banyak digunakan orang untuk share video yang berisi bermacam jenis content. Pemakai tidak terdaftar bisa melihat video, sementara pemakai terdaftar bisa mengupload video serta memberi komentar dengan jumlah tidak terbatas. Biasanya beberapa video di YouTube yaitu clip musik (video klip), trailer film, video edukasi, video ulasan, dan video diskusi, debat atau dialog. Komentar dan opini pengguna di Youtube dapat dipakai sebagai indikator untuk melihat kecondongan pemakai pada pasangan calon kepala daerah yang berkaitan, hingga data komentar tersebut bisa menjadi sumber data opini dan sentimen masyarakat pada satu studi sosial. Pada suatu Tim Sukses pasangan calon Pemilihan kepala daerah, sentiment analysis dipakai sebagai alasan penetapan kebijakan serta taktik kampanye untuk menaikkan kepopuleran jago mereka serta untuk testing calon mereka di mata masyarakat apa bisa terima secara baik atau belum. Support Vector Machine (SVM) merupakah salah satu model sentiment analysis. SVM tergolong dalam kelompok algoritme dengan teknik supervised (diawasi). Dalam tiga kelompok kategorisasi yang dipakai dalam SVM akan mencari nilai maksimal hyperplane yang membagi ruangan pengetesan menjadi kelas-kelas yang terpisah satu yang lain. SVM termasuk algoritme komputasional yang memerlukan operasi yang besar lantaran menyertakan diskretisasi, normalisasi serta operasi titik produk yang berulangulang. Diharapkan dengan Support Vector Machine (SVM) ini bisa mengolah data komentar pada video debat calon pimpinan daerah yang berada di YouTube secara automatis serta seterusnya bisa mengklasifikasi sentimen analisis komentar masyarakat terhadapcalon kepala daerah yang akan dilaksanakan, hingga menjadi referensi untuk lebih lanjut buat yang berminat untuk mengembangkannya.*

Kata Kunci— sentiment analysis, media sosial, pilkada, Sentimen Analisis Tertarget, Youtube.

Abstract **:** *YouTube is a social media that is widely used by people to share videos that contain various types of content. Unregistered users can view videos, while registered users can upload videos and provide an unlimited number of comments. Mostly, videos on YouTube are music clips (video clips), movie trailers, educational videos, review videos, discussion videos, and debate or dialogue. Users' comments and opinions on YouTube can be used as an indicator to see their inclination to a particular regional head candidate; therefore, comments can be a source of data on public opinion and sentiment in a social study. Inside the candidate team for regional head elections, sentiment analysis is used as a rationale for determining policies and campaign tactics to increase the popularity of their candidate and to test whether the candidate is well accepted in the public eye. Support Vector Machine (SVM) is a sentiment analysis model. SVM belongs to the algorithm group with the supervised technique. The three groups of the categorization used in SVM will look for the maximum value of the hyperplane which divides the test room into separate classes. SVM is a computational algorithm that requires a large operation because it includes discretization, normalization, and repeated product point operations. It is expected that Support Vector Machine (SVM) can automatically process comment data on the debate video of regional head candidates posted on YouTube, and then continually classify sentiment analysis of people's comments on the regional head candidates. Additionally, this study is significant to become a further reference for those interested in developing SVM.*

Keywords— sentiment analysis, social media, regional head election, targeted sentiment analysis, YouTube..

I. PENDAHULUAN

PERKEMBANGAN internet yang begitu pesat saat ini telah membawa interaksi manusia yang intensif di dunia Internet ke era media sosial. Media sosial dapat didefinisikan sebagai kelompok dari aplikasi berbasis Internet yang berkumpul berdasarkan ideologi dan perkembangan teknologi Web 2.0 yang memperbolehkan adanya pembentukan dan pertukaran konten yang dibuat oleh pengguna. Salah satu media sosial yang popular saat ini adalah YouTube.

YouTube merupakan media sosial yang banyak dimanfaatkan orang untuk berbagi video yang memuat berbagai macam konten. Video dialog atau debat calon kepala daerah dan calon wakil kepala daerah

DOI: https://doi.org/10.35457/antivirus.v15i1.1539

banyak dilihat dan dikomentari pengguna di YouTube, dalam video tersebut masing masing pasangan calon akan menyampaikan visi misi, juga beradu argument terkait program kerja masing masing, pada video debat tersebut masing masing pasangan juga diuji dengan berbagai persoalan yang disodorkan para ahli, juga kritisi dari pasangan calon lainnya. Sehingga dari konten video tersebut, pengguna dapat menilai kualitas dari masing masing pasangan calon kepala daerah dan akhirnya dapat memberikan feedback melalui komentar bisa berupa opini yang menguatkan (positif) ataupun opini yang melemahkan (negatif) bahkan yang memilih bersikap netral. Komentar opini pengguna bisa digunakan sebagai indikator untuk melihat kecenderungan pengguna terhadap pasangan calon kepala daerah. Pada penelitian ini mengambil data komentar bukan dari *live streaming* komentar tetapi dari kolom komentar yang diisikan oleh para user terdaftar YouTube, dengan memakan waktu pengambilan data selama 2 hari dengan total data komentar sebanyak 1251 buah.

Dalam Penelitian ini studi kasus yang digunakan adalah sentiment analysis terhadap debat calon pemimpin daerah. Data komentar analisis sentiment berasal dari postingan video dari debat tersebut, yang memunculkan suatu komentar dari para pelihat video kemudian mengekstrasi komentar tersebut. Hasil dari ekstrasi komentar akan diuji dengan parameter yang telah ditetapkan yaitu 3 pengelompokan atribut positif, negatif, dan netral. Proses pengolahan data komentar mentah hingga didapatkan kumpulan informasi yang diinginkan melalui tahapan-tahapan yang terurut. Tahapan-tahapan tersebut antara lain pengolahan data, algoritma pembelajaran, dan metode pengujian [1][2][3].

Algoritma pembelajaran yang dipilih dalam penelitian ini adalah *algoritma Support Vector Machine* (SVM). SVM termasuk dalam kategori algoritma dengan teknik *supervised* (diawasi) [1]. Dalam tiga kelompok klasifikasi yang digunakan dalam SVM akan mencari nilai maksimum *hyperplane* yang membagi ruang pengujian menjadi kelas-kelas yang terpisah satu dengan yang lain. SVM termasuk algoritma komputasional yang membutuhkan operasi yang besar karena melibatkan diskretisasi, normalisasi dan operasi titik produk yang berulang-ulang [7].

II. TINJAUAN PUSTAKA

A. Targeting

Pada proses *Targeting* komentar video debat, untuk masing masing calon kepala daerah diterapkan variabel-variabel tertentu yang diset diawal sebagai penanda bagi masing masing calon saat sebuah komentar ditujukan padanya. Variabel-variabel per calon tersebut bisa bervariasi, dapat berupa nama asli, nama potongan atau julukan calon yang umum digunakan masyarakat untuk menyebut calon-calon tersebut. Seperti contoh pada Pak Susilo Bambang Yudhoyono biasa dipanggil juga sebagai SBY atau pak Beye dan lainnya, atau Pak Presiden Jokowi yang biasa dipanggil juga sebagai JKW, Jokowi, Wiwi dan lainya.

Praktek untuk menentukan suatu komentar yang tertarget maka dibutuhkan proses klasifikasi komentar sesuai dengan *variable* calon kepala daerah, dalam hal ini akan digunakan *Support Vector Machine* dengan memanfaatkan *library* pada bahasa pemrograman PHP yaitu SVM. PHP SVM adalah *library* PHP yang gratis dan sangat efisien untuk mining dan analisis data [6].

B. Support Vector Machine (SVM)

SVM mengklasifikasi data dalam dua kelas yang berbeda, dengan cara membuat *decision boundary* atau umum disebut juga *hyperplane* Konsep dasar SVM adalah mencari *hyperplane* yang memaksimalkan *margin* [5].

Berikut model linier secara umum yang dipakai dalam SVM untuk menghasilkan *hyperplane*

$$y = sign(w^T x + b) \quad (1)$$

Hyperplane yang dihasilkan membagi data menjadi dua kelas, yaitu kelas positif dan kelas negatif yang dimodelkan sebagai berikut:

$$w^T x_i + b \geq 1, \text{ untuk } y^1 \text{ bernilai } 1 \quad (2)$$

$$w^T x_i + b \geq -1, \text{ untuk } y^1 \text{ bernilai } -1 \quad (3)$$

DOI: https://doi.org/10.35457/antivirus.v15i1.1539

dapat diformulasikan sebagai berikut :

$$y_i = (w^T x_i + b) \geq 1 \quad (4)$$

Hyperplane optimal yang dihasilkan merupakan *hyperplane* yang memaksimalkan jarak minimum antara kedua *hyperplane* diatas. Sehingga didapat persamaan sebagai berikut:

$$\frac{min}{wb} \frac{||w||2}{2} \quad (5)$$

Dengan syarat :

$y_1(w^T x_{i+} b) \geq 1$

Dari persamaan 5 selanjutnya ditambahkan variabel slack $\sum i \geq 0$ sehingga menjadi:

$$\frac{min}{w.b} \left(\frac{1}{2} |[w]|^2 + b \sum_{i=1}^{n} \xi_1\right) \quad (6)$$

Dengan syarat :

$y_i (w^T x_i + b) \geq 1 \xi_i$

$\xi_I \geq 0$

Variabel C merupakan *Tradeoff* dari margin dan variabel *slack*. Dalam bentuk dual, persamaan akan berbentuk sebagai berikut:

$$\max L = \sum_{i=1}^{N} \alpha_1 - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \gamma_i \gamma_j x_i^T x_j \quad (7)$$

Dengan syarat :

$0 \leq \alpha_i \leq C, i = 1, \ldots, N$

$\sum_{i=1}^{N} \alpha_1 y_i = 0$

Setiap perkalian x_i^T x_j akan diubah menjadi $K(x_i, x_j)$ yang memiliki bentuk sebagai berikut:

$$\frac{\max L}{\alpha} = \sum_{i=1}^{N} \alpha_1 - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \alpha_i \alpha_j K(x_i, x_j) \quad (8)$$

Dengan syarat,

$0 \leq \alpha_i \leq C, i = 1, \ldots, N$

$\sum_{i=1}^{N} \alpha_i \gamma_i = 0$

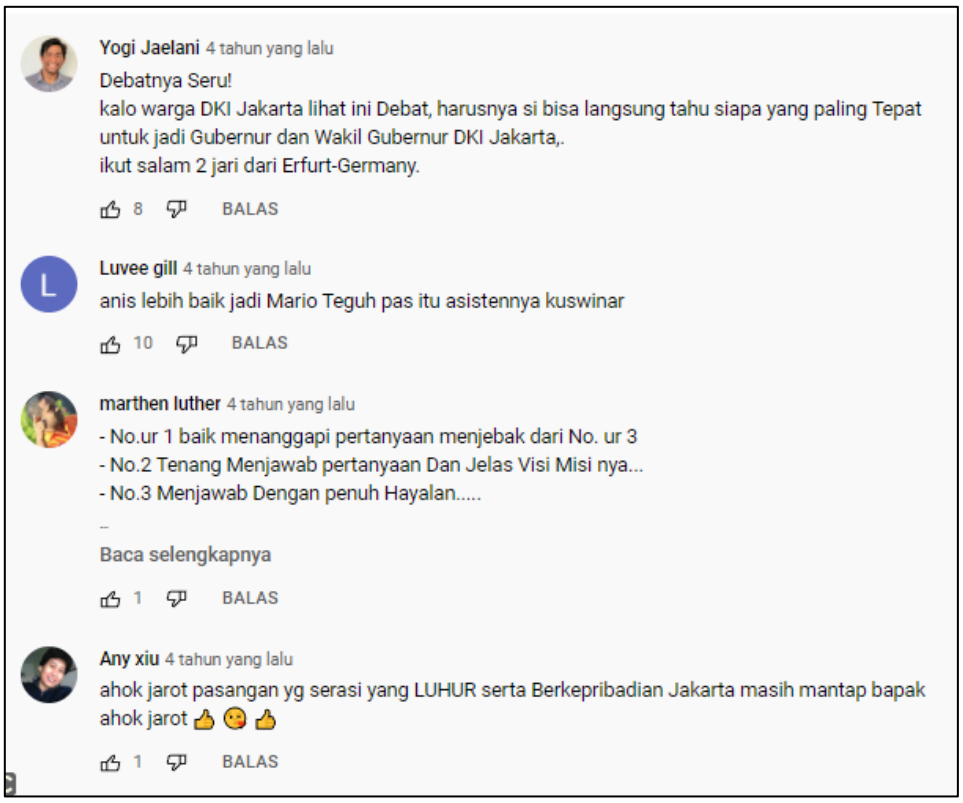
III. PERANCANGAN

A. Data Observasi dan Motivasi

1) Menentukan Sumber Data (Data Input)

Dalam penelitian ini data diambil dari opini publik pada data komentar video debat calon kepala daerah di YouTube, yaitu dari video debat 2017 DKI Jakarta terdapat 3 tahap debat yaitu debat Pilkada tahap 1, tahap 2 dan tahap final (tahap 3). Dari masing masing tahap debat Pilkada akan dipilih 1 Video yang diambil dari video yang diupload oleh stasiun TV pelaksana debat publik terkait, jadi total ada 3 video, dari 3 tahap debat Pilkada masing masing debat calon. Dari masing masing video akan diambil data komentar 1251 terdiri dari 32545 kata.

DOI: https://doi.org/10.35457/antivirus.v15i1.1539

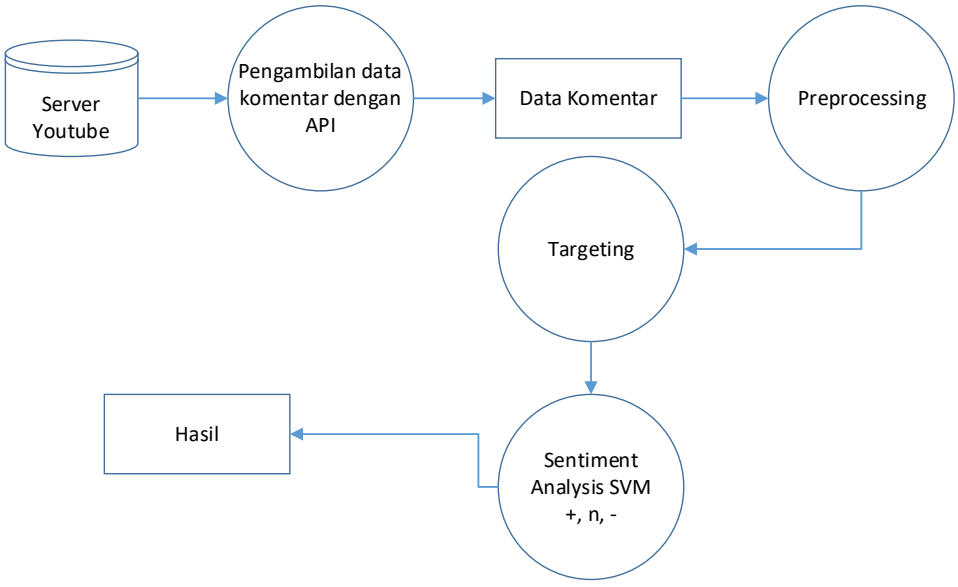


Gambar 1. Contoh Data Komentar Video di Youtube

2) Data Output

Data output yang diharapkan adalah berupa hasil Pilkada yaitu data sentiment masyarakat terhadap Calon kepala daerah yang memiliki tingkat terpilih paling tinggi. Data ini berupa sentiment *positif, negative* atau netral yang tertarget untuk masing masing calon.

B. Architecture



Gambar 2. Blok Diagram *System*

Pada Blok Diagram *System* diatas terdapat beberapa tahap sebagai berikut:

1) Tahap Persiapan data

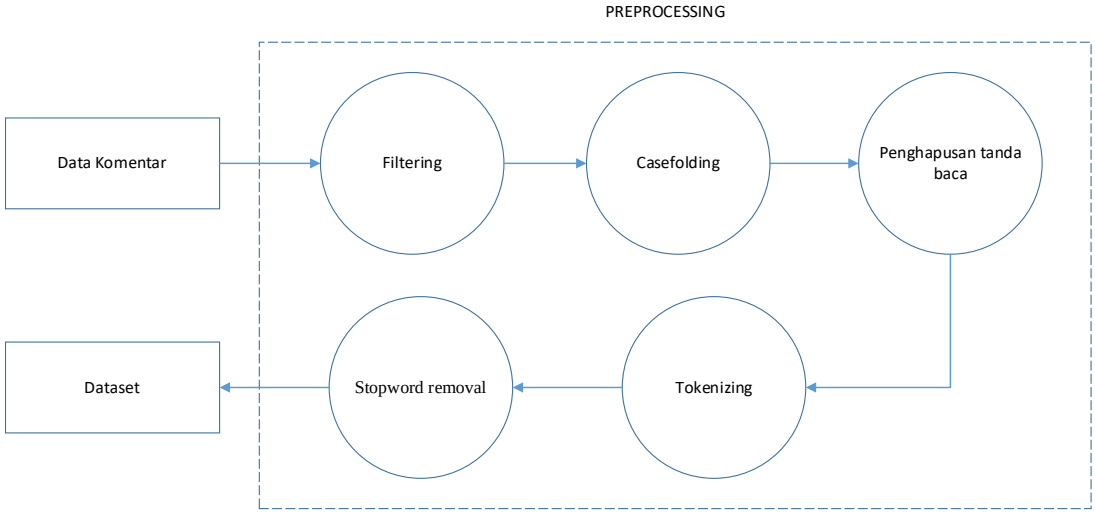
Pada tahapan ini dimulai dengan mempersiapkan data opini publik. Data diambil dari komentar komentar pada video debat Pilkada 2017 DKI Jakarta dengan menggunakan API YouTube. Hasil

DOI: https://doi.org/10.35457/antivirus.v15i1.1539

dari tahap ini adalah data komentar, yaitu data mentah dari hasil pengambilan data langsung melalui API, yang selanjutnya akan dilakukan preprocessing.

2) *Preprocessing*

Untuk tahap Preprocessing lebih jelasnya dapat dilihat pada blok diagram berikut ini :



Gambar 3. Preprocessing

Keterangan:

a. *Filtering*
   Proses *filtering* digunakan untuk penanganan komentar yang sambung menyambung alias ada *user* yang menulis komentar lebih dari 1 kali, maka pada proses *filtering user* yang menulis komentar lebih dari 1 kali hanya akan diambil 1 komentar user tersebut, dipilih berdasarkan komentar dengan kata paling banyak dan paling terakhir, dan otomatis komentar lain pada *user* tersebut dibuang.
b. *Case Folding*
   Pada proses ini semua kata akan ditransformasi menjadi huruf kecil.
c. *Penghapusan Tanda Baca*
   Pada proses ini, menghapus tanda baca seperti [!”#$%&’()*+,-./:;<=>?@[\]^_`{|}~].
d. *Tokenizing*
   Pada proses ini, data akan dipisah berdasarkan spasi dari setiap kata.
e. *Stopword removal*
   Pada proses ini, kata-kata yang tidak baku akan dihilangkan. Menggunakan metode *wordlist*, ada database kata-kata yang baku, dan jika hasil tokenisasi ada yang termasuk kata baku dalam database tersebut, maka hasil tokenisasi akan disimpan. Jika tidak termasuk maka akan dibuang. Landasan penyusunan database *Wordlist* adalah dari database Kamus Besar Bahasa Indonesia (KBBI) dan dapat di update secara berkala sesuai KBBI yang berlaku.

3) *Targeted dan Sentimen Analisis*

Pada proses *Targeted* dan *Sentimen Analysis* pada penelitian ini memiliki alur proses bisnis yang sama juga dengan menggunakan SVM, perbedaan mendasar terletak pada dataset (*training* dan *testing*), jika *targeted* acuan datasetnya adalah kata kata kunci dari nama nama tokoh yang dijadikan target, dan untuk *sentiment analysis* acuannya adalah data set komentar yang sudah dilabelkan berdasar 3 data uji yaitu : *positif*, *negative* dan *netral*.

Praktek untuk menentukan *sentiment* suatu komentar yang tertarget maka dibutuhkan proses klasifikasi komentar sesuai dengan data training calon kepala daerah untuk *targeted* dan komentar yang terlabelkan *positif*, *negatif* dan *netral* untuk *sentiment analisis*, dalam hal ini akan digunakan *Support Vector Machines* dengan memanfaatkan *Library* pada bahasa pemrograman *Python* yaitu

DOI: https://doi.org/10.35457/antivirus.v15i1.1539

Scikit-Learn. Scikit-Learn adalah *Library Python* yang gratis dan efisien untuk mining dan analisis data.

IV. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan tema dataset Debat Pilkada 2017 DKI Jakarta, sehingga kata kunci yang dipakai harus yang ada hubungannya dengan Debat Pilkada 2017 di DKI Jakarta. Data Komentar hasil *crawling* data dengan API YouTube adalah 1251 terdiri dari 32545 kata, yang sudah mengalami proses *filtering*, yaitu hanya mengambil salah satu data komentar user yang sama walau berkomentar lebih dari satu kali dengan kriteria komentar yang memiliki kosa kata paling banyak dan paling baru. Tahap demi tahap proses *pre-processing* pada penelitian ini sebagai berikut :

A. Pre-Processing

1) Case Folding

Case folding adalah salah satu bentuk *text preprocessing* yang paling sederhana dan efektif meskipun sering diabaikan. Tujuan dari *case folding* untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampai 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap *delimiter*. Pada tahap ini tidak menggunakan *external library* apapun, kita bisa memanfaatkan fungsi yang ada di PHP. Hasil algoritme 4.1 Case Folding sebagai berikut:

```
01 : Masukkan kalimat.
02 : Rubah tiap kata menjadi huruf kecil semua.
03 : Lakukan hingga seluruh kata menjadi huruf kecil seluruhnya
```

Berikut adalah contoh data yang belum memasuki tahapan *case folding* :

Tabel 1. Data sebelum *case folding*

Index	Komentar	*Targeted*
15811	Awal mula sblum Anies mnjdi gubernur se-Indonesia	Anies Baswedan
15796	Bpk anies ngomong nya enak bgt Kelihatan cerdas nya	Anies Baswedan
15780	Apa yg dipikirkan Ahok,,maka Anis tinggi ikut sama semua	Ahok
8616	Waduh puti kalah telak	Puti Sukarno
8914	Ya jelas emil .. bicara fakta.. jadi sokong khofifah & EMIL	Emil Dardak

Untuk melakukan *case folding* pada bahasa pemrograman php, maka *script* yang digunakan adalah *strtolower($string).* Seperti pada contoh berikut :

```
#1. casefolding
$set['casefolding'] = strtolower($komentar);
```

Berikut adalah hasil dari tahapan *case folding* pada dataset :

Tabel 2. Hasil tahapan *case folding*

Index	Komentar	*Targeted*
15811	awal mula sblum anies mnjdi gubernur se-indonesia	Anies Baswedan

DOI: https://doi.org/10.35457/antivirus.v15i1.1539

Index	Komentar	*Targeted*
15796	bpk anies ngomong nya enak bgt kelihatan cerdas nya	Anies Baswedan
15780	apa yg dipikirkan ahok,,maka anis tinggi ikut sama semua	Ahok
8616	waduh puti kalah telak	Puti Sukarno
8914	ya jelas emil .. bicara fakta.. jadi sokong khofifah & emil	Emil Dardak

2) Penghapusan tanda baca

Sama halnya dengan angka, tanda baca dalam kalimat tidak memiliki pengaruh pada *text preprocessing*. Hasil algoritme 4.2 Penghapusan tanda baca sebagai berikut:

```
01 : Masukkan kalimat.
02 : Deteksi kalimat mengandung tanda baca
03 : Hapus tanda baca
04 : Lakukan hingga seluruh kalimat terhapus tanda bacanya
```

Menghapus tanda baca seperti [!”#$%&’()*+,-./:;<=>?@[\]^_`{|}~] dapat dilakukan di php seperti dibawah ini :

```
$str = preg_replace('/[^0-9a-zA-Z]/', ' ', $str);
$str = preg_replace('/\s+/', ' ', $str);
```

Berikut adalah hasil dari penghapusan tanda baca pada teks yang akan diproses :

Tabel 3. Hasil dari penghapusan tanda baca

Index	Komentar	*Targeted*
15811	awal mula sblum anies mnjdi gubernur se indonesia	Anies Baswedan
15796	bpk anies ngomong nya enak bgt kelihatan cerdas nya	Anies Baswedan
15780	apa yg dipikirkan ahok maka anis tinggi ikut sama semua	Ahok
8616	waduh puti kalah telak	Puti Sukarno
8914	ya jelas emil bicara fakta jadi sokong khofifah emil	Emil Dardak

3) Tokenizing

Tokenizing adalah proses pemisahan teks menjadi potongan-potongan yang disebut sebagai token untuk kemudian di analisa. Kata, angka, simbol, tanda baca dan entitas penting lainnya dapat dianggap sebagai token. Didalam NLP, token diartikan sebagai “kata” meskipun *tokenize* juga dapat dilakukan pada paragraf maupun kalimat. Hasil algoritme 4.3 *Tokenizing* sebagai berikut:

```
01 : Masukkan kalimat hasil dari proses Remove Punctuation.
02 : Membentuk sebuah array dari kalimat dengan menggunakan
     fungsi explode pada php menggunakan pemisah spasi.
03 : Membentuk json dari hasil proses explode.
```

```
$set['tokenizing'] = build(function() use ($set){
    $str = $set['casefolding'];
    $arr = explode(' ', $str);
    $arr = array_filter($arr);
```


DOI: https://doi.org/10.35457/antivirus.v15i1.1539

```
        return json_encode($arr);
    });
```

Tabel 4. Hasil dari tokenisasi

Index	Komentar	*Targeted*
15811	["awal","mula","sblum","anies","mnjdi","gubernur","se","indonesia"]	Anies Baswedan
15796	["bpk","anies","ngomong","nya","enak","bgt","kelihatan","cerdas","nya"]	Anies Baswedan
15780	["apa","yg","dipikirkan","ahok","maka","anis","tinggi","ikut","sama","semua"]	Ahok
8616	["waduh","puti","kalah","telak"]	Puti Sukarno
8914	["ya","jelas","emil","bicara","fakta","jadi","sokong","khofifah","emil"]	Emil Dardak

4) Stopword Removal

Stopword adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh *stopword* dalam bahasa Indonesia adalah "yang", "dan", "di", "dari", dll. Makna di balik penggunaan *stopword* yaitu dengan menghapus kata-kata yang memiliki informasi rendah dari sebuah teks, kita dapat fokus pada kata-kata penting sebagai gantinya. Hasil algoritme 4.4 *Stopword Removal* sebagai berikut:

```
01 : Masukkan data hasil tokenizing.
02 : Ubah data menjadi array menggunakan fitur json decode.
03 : Mengambil data wordlist dari database dalam bentuk array.
04 : Hapus kata yang tidak ditemukan, dengan membandingkan data hasil
     tokenizing dan data wordlist menggunakan fungsi array intersect.
05 : Ubah data array hasil perbandingan menjadi teks kembali.
```

Berikut adalah *script* untuk melakukan penghapusan *stopwords* pada bahasa php menggunakan librari nltk dengan konfigurasi *stopword* menggunakan Bahasa Indonesia :

```
# stopword
$set['stopword'] = build(function() use ($set, $wordlist)
{
 $tokenizing = json_decode($set['tokenizing'], true);
 $str = array_intersect($tokenizing, $wordlist);
 return implode(' ', $str);
});
```

Berikut adalah hasil dari *stopword removal* :

Tabel 5. Hasil *Stopword Removal*.

Index	Komentar	*Targeted*
15811	awal mula gubernur se indonesia	Anies Baswedan
15796	nya enak kelihatan cerdas nya	Anies Baswedan
15780	apa dipikirkan maka anis tinggi ikut sama semua	Ahok
8616	waduh puti kalah telak	Puti Sukarno
8914	ya jelas bicara fakta jadi sokong	Emil Dardak

DOI: https://doi.org/10.35457/antivirus.v15i1.1539

B. Proses Targeted dan Analisis Sentiment

Setelah tahap *pre-processing* kemudian adalah Klasifikasi SVM, ada 2 proses yaitu *Targeted* dan Sentiment Analisis. Yang pertama, untuk menentukan untuk siapa komentar tersebut di tujukan, meliputi label data naman ama tokoh peserta Pilkada. Yang kedua, Sentiment Analisis menentukan label data uji positif, negatif dan netral terhadap tiap tiap komentar. Terdapat beberapa *Library Python* yang digunakan. Gambar 4 menunjukkan *Library Python* yang digunakan pada penelitian ini:

```
import pandas as pd
import time
import db

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import svm
from sklearn.metrics import classification_report

```

Gambar 4. *Library Python*

Library Pandas berfungsi mengambil data train dari database MySQL. *Library* DB digunakan untuk mengkoneksikan *Python* dengan *database*. Library sklearn.feature_extraction.text Tf-Idf Vectorizer berfungsi untuk *feature extraction*. *Library* time berfungsi menghitung lama waktu masa *training* dan lama waktu *testing*. *Library* sklearn.svm. SVC berfungsi pada proses *classification* dengan SVM.

```
        def __init__(self, minDf, maxDf, trainData, testData):
                # Create feature vectors
                vectorizer = TfidfVectorizer(min_df = minDf,
                                             max_df = maxDf,
                                             sublinear_tf = True,
                                             use_idf = True)
                train_vectors = vectorizer.fit_transform(trainData['Content'])
                test_vectors = vectorizer.transform(testData['Content'])
```

Gambar 5. Potongan Kode untuk *Feature Extraction Tf-Idf*

Gambar 5 adalah potongan kode memakai *library TfidfVectorizer* untuk mengubah *data train* dan *data test* kebentuk *matrix tf-idf*. Bentuk transformasi disimpan pada variabel *train_vectors* untuk data train dan variabel *test_vectors* untuk *data test*. *Library TfidfVectorizer* mamakai parameter. Parameter *min_df* memiliki nilai ambang bawah (*cut-off*) *df*. Parameter *max_df* memiliki nilai ambang batas (*threshold*) *df*. Parameter *sublinear_tf* berarti bahwa *tf_scling* diaplikasikan dengan cara *sublinear*. Parameter *use_idf* menunjukkan akan dilakukan pembobotan ulang pada *Idf*.

```
                # Perform classification with SVM, kernel=linear
                classifier_linear = svm.SVC(kernel='rbf')
                t0 = time.time()
                classifier_linear.fit(train_vectors, trainData['Label'])
                t1 = time.time()
                prediction_linear = classifier_linear.predict(test_vectors)
                t2 = time.time()
                time_linear_train = t1-t0
                time_linear_predict = t2-t1

                self.classifier_linear = classifier_linear
                self.vectorizer = vectorizer

sentimentTrain, sentimentTesting = db.getSentimentDataSet()
aspectTrain, aspectTesting = db.getAspectDataSet()
targetTrain, targetTesting = db.getTargetDataSet()
```

Gambar 6. Potongan Kode untuk Klasifikasi dengan SVM

DOI: https://doi.org/10.35457/antivirus.v15i1.1539

Gambar 6 berisi potongan kode memakai *Library* SVM untuk proses *classification*. Pada baris awal ditunjukkan bahwa klasifikasi memakai SVM memakai *linear kernel* untuk proses *classification*. Tahap *classification* diimplementasikan di *train_vectors* (hasil *feature extraction*) pada label tiap dataset. Tahap *prediction* diproses berlandaskan hasil *classification* pada *test_vectors* (hasil *feature extraction*). Variabel time_linear_train dimanfaatkan guna menghitung lama waktu proses *training*. Variabel time_linear_predict dimanfaatkan guna menghitung lama waktu proses *testing*. Kemudian hasil hitung waktu dimunculkan bersamaan dengan *classification report*.

1) Targeted

Pada tahap targeted, dilakukan proses klasifikasi untuk menentukan target pada setiap komentar yang ada berdasarkan kata kunci nama nama tokoh calon Pilkada, berikut adalah alurnya hasil algoritme 4.5 Targeted:

```
01  Menginisialisasi class svm run.
02  Mengambil daftar kata kunci targeting dari database.
03  Melakukan Training data.
04  Looping data komentar.
05  Melakukan klasifikasi data dari daftar kata kunci.
```

```
#Targeted Analysis
tA = svmRun(minDf=1, maxDf=2, trainData=targetTrain, testData=targetTesting)

def isEmpty(s):
        if(len(str(s).strip()) > 0):
                return False
        else:
                return True

lanjutkan = True
while lanjutkan:
        # komentar mentah
        komentar = db.getKomentarMentah()

        for v in komentar:
                string = v[1]

                #targeted klasifikasi
                targetedTransform = tA.vectorizer.transform([string])
                [targetedClassify] = tA.classifier_linear.predict(targetedTransform)
                #jika tidak dikenali maka diberi nilai 0
                if(isEmpty(targetedTransform)):
                        targetedClassify = 0

                #jika tidak memiliki status targeted
                if(v[2]==0):
                        targetedClassify = 0
                        pass

                db.setSentiment(kId=v[0], sentiment=sentimentClassify,
                                target_id=targetedClassify, aspected_id=aspectedClassify)

        length = len(komentar)

        if(length > 1):
                print(f'{length} komentar telah diproses.')
                lanjutkan = True
        else:
                lanjutkan = False

        pass

print('Semua komentar telah diproses')

# mematikan koneksi mysql
db.conn.close()
```

Gambar 7. Potongan Kode Prediksi untuk *Targeted*

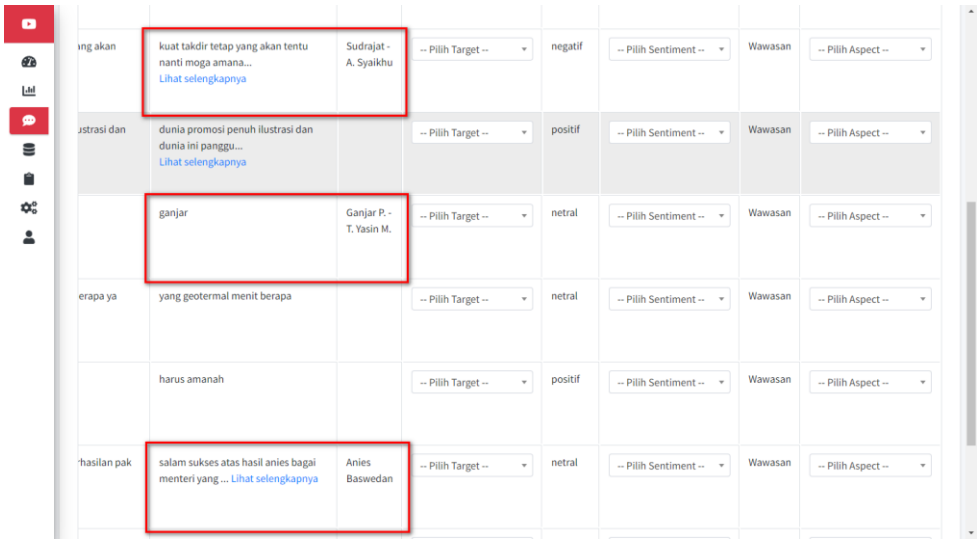
DOI: https://doi.org/10.35457/antivirus.v15i1.1539

Gambar 7 menampilkan potongan kode pada proses prediksi untuk suatu *targeted*. Kalimat yang disimpan di variabel *review* ditransformasikan memakai *library TfidfVectorizer* yang kemudian disimpan menjadi variabel review_vector. Variabel review_vector kemudian akan dilakukan proses prediksi memakai *library* SVM. Hasil proses prediksi sentimen berbentuk label sesuai dengan label nama nama tokoh per kata kunci *targeted*. Berikut hasil dari proses *targeted* pada Tabel 4:

Tabel 4. Hasil Targeted.

Index	Komentar	Targeted
15811	awal mula anies gubernur indonesia	Anies Baswedan
15796	anies enak lihat cerdas	Anies Baswedan
15780	apa pikir ahok maka anis tinggi ikut sama semua	Ahok
8616	waduh puti kalah telak	Puti Sukarno
8914	jelas emil bicara fakta jadi sokong emil	Emil Dardak

Berikut hasil proses *targeted* pada tampilan web:



Gambar 8. Hasil proses targeted

2) Sentiment

Lanjut pada tahap Sentiment, dilakukan proses klasifikasi untuk menentukan sentiment pada setiap komentar yang ada berdasar pada acuan data training komentar yang sudah di labelkan positif, netral dan negatif, berikut adalah alurnya Algoritme 4.6. Sentiment

```
01  Menginisialisasi class svm run.
02  Mengambil daftar sentiment targeting dari database.
03  Melakukan Training data.
04  Looping data komentar.
05  Melakukan klasifikasi data dari daftar kata kunci.
```


DOI: https://doi.org/10.35457/antivirus.v15i1.1539

```
38
39 #Sentiment Analysis
40 sA = svmRun(minDf=5, maxDf=0.8, trainData=sentimentTrain, testData=sentimentTesting)
41
45 def isEmpty(s):
46         if(len(str(s).strip()) > 0):
47                 return False
48         else:
49                 return True
50
51 lanjutkan = True
52 while lanjutkan:
53         # komentar mentah
54         komentar = db.getKomentarMentah()
55
56         for v in komentar:
57                 string = v[1]
58
59                 #sentiment klasifikasi
60                 sentimentTransform = sA.vectorizer.transform([string])
61                 [sentimentClassify] = sA.classifier_linear.predict(sentimentTransform)
62
74
75                 db.setSentiment(kId=v[0], sentiment=sentimentClassify,
76                                 target_id=targetedClassify, aspected_id=aspectedClassify)
77
78         length = len(komentar)
79
80         if(length > 1):
81                 print(f'{length} komentar telah diproses.')
82                 lanjutkan = True
83         else:
84                 lanjutkan = False
85
86         pass
87
88 print('Semua komentar telah diproses')
89
90 # mematikan koneksi mysql
91 db.conn.close()
```

Gambar 9. Potongan Kode Prediksi untuk *Sentiment*

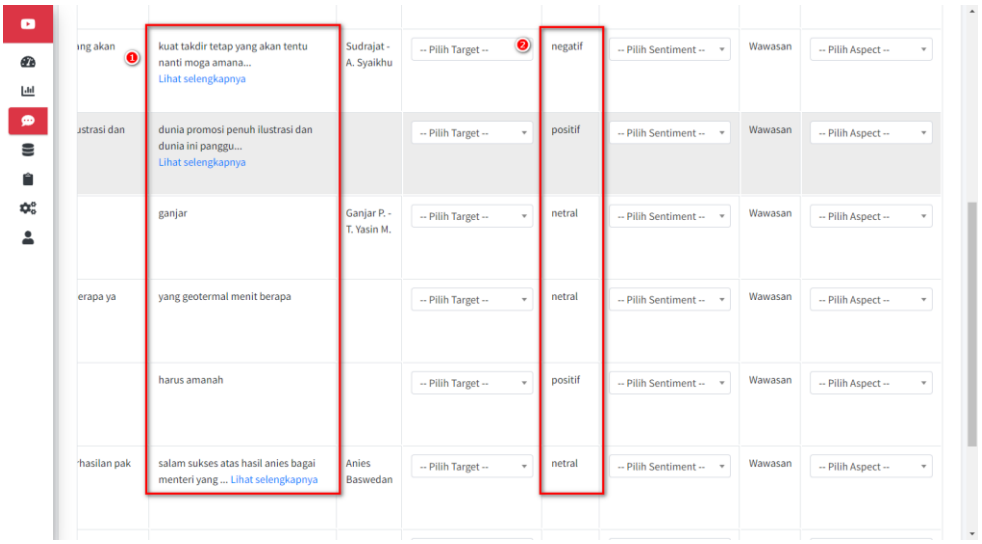
Gambar 9 merupakan potongan kode untuk proses prediksi untuk *Sentiment*. Kalimat yang disimpan di variabel *review* ditransformasikan memakai *library TfidfVectorizer* yang kemudian disimpan dalam bentuk variabel review_vector. Variabel review_vector kemudian dilakukan proses prediksi memakai *library* SVM. Hasil prediksi sentimen dalam bentuk label pos pada sentimen positif, label net pada sentiment netral dan label neg pada sentimen negatif. Berikut adalah hasil dari proses *sentiment*:

Tabel 4. Hasil Sentiment.

Index	Komentar	Sentiment
8996	cara logika lebih banyak prestasi bisa lihat trek dia profesional bukan lahir dari politis jadi wajar kalau ganjar pandai kuat berat untuk	Positif
9189	ahok kenal cerdas nyata anis	Netral
14492	bohong demi bohong bukti baru anies hanya debat kusir lihat kampanye hebat bukti nol	Negatif
9376	kang emil mantap	Netral

Berikut hasil dari proses *sentiment* pada halaman web:

DOI: https://doi.org/10.35457/antivirus.v15i1.1539



Gambar 10. Hasil proses sentiment

Pada *sentiment analysis* ini pelabelan tiap data berbeda karena setelah proses *crawling* data dipilih tidak dengan cara manual, sehingga data hasil dari *crawling* langsung dapat diproses didalam sistem, dan sistem melabelkan data komentar kedalam 3 kategori yaitu positif, negatif, dan netral. Hasil ujicoba bisa dilihat pada Tabel 6, 7 dan 8 berikut :

Tabel 6. Persentase (%) pada SVM *Sentiment Analysis*

No.	*Keyword*	Positif (%)	Negatif (%)	Netral (%)
1	Anies Baswedan	99.10(%)	0.4(%)	0.5(%)
2	Basuki T. Purnama	98.15(%)	0.85(%)	1(%)
3	Agus Harimurti Yudhoyono	95.80(%)	4(%)	0.2(%)

Tabel 7. Jumlah Data Hasil dari Sentimen Analisis dengan SVM

No	*Keyword*	Jumlah Data Uji Positif	Jumlah Data Uji Negatif	Jumlah Data Uji Netral	Jumlah Data Hasil Uji Positif	Jumlah Data Hasil Uji Negatif	Jumlah Data Hasil Uji Netral
1	Anies Baswedan	600	313	87	651	298	51
2	Basuki T. Purnama	600	224	116	547	330	63
3	Agus Harimurti Yudhoyono	600	41	19	475	124	61

Tabel 8. Akurasi dan Waktu Pemrosesan SVM dalam Menganalisis Sentimen

No	*Keyword*	Akurasi (%)	Waktu Proses (Detik)
1	Anies Baswedan	66.77(%)	107.80 detik
2	Basuki T. Purnama	66.10(%)	121.75 detik
3	Agus Harimurti Yudhoyono	66.90(%)	86.42 detik

C. Pembahasan

Dari hasil ujicoba pada Tabel 6 dan 7 dapat dipahami bahwa SVM dapat memproses data s*entiment analysis* dengan baik dengan pola SVM yang mengacu pada ketersediaan *support vector* dalam membentuk *hyperlane*. Data komentar pada Video Debat Pilkada di YouTube dalam penelitian ini lebih cenderung terdistribusi linier yang mana SVM punya kelebihan dalam menganalisis data terdistribusi linier. SVM punya rata-rata akurasi yang cukup baik. Banyaknya data tes tidak begitu punya pengaruh

DOI: https://doi.org/10.35457/antivirus.v15i1.1539

pada hasil generalisasinya. Tetapi, hasilnya bisa lebih baik jika data trainingnya lebih banyak atau sama dengan jumlah dari data tesnya. Dari hasil ujicoba komputer diperoleh kesimpulan bahwa SVM dapat mengklasifikasi data tes dengan kemampuan yang cukup baik.

Seluruh matriks data diuji menggunakan fungsi kernel linear, formula SVM mentransformasikan data pada dimensi ruang fitur dengan mamakai fungsi kernel. Kernel liner adalah kernel paling sederhana sehingga memiliki performa yang lebih baik, dan kebetulan data komentar di YouTube di penelitian ini lebih cenderung terdistribusi linier, maka sangat sesuai sekali dengan fungsi kernel linier itu sendiri. Proses pengujian memiliki tujuan untuk membangun model serta mengukur tingkat akurasi SVM dalam mengklasifikasi data testing.

Training SVM butuh parameter yang sesuai dengan kernelnya. Setiap proses *training* SVM yang memakai fungsi kernel dibutuhkan parameter paling baik untuk memperoleh tingkat akurasi yang paling baik jika telah mencapai rataan nilai tertinggi. Tetapi, pada penelitian ini diperoleh rata-rata nilai yang hampir sama pada setiap iterasi pemodelan maka pengambilan parameter diproses pada nilai akurasi tertinggi yang pertama.

Jumlah Data Komentar yang dipakai untuk dianalisis pada penelitian ini adalah 1251 data komentar dalam bentuk sentimen acak. SVM memiliki tingkat akurasi cukup baik untuk menganalisa teks, sebab pada penelitian ini SVM memakai kernel linear yaitu kernel paling sederhana dibanding dengan semua fungsi kernel lain. Kernel ini mempunyai kelebihan dalam analisis teks, akan tetapi SVM kurang maksimal apabila digunakan pada pemrosesan data dalam skala besar. Dan pada SVM perlu ditentukan nilai parameter k (jumlah variable terdekat), training yang dilakukan berdasarkan jenis jarak digunakan serta atribut apa yang akan difungsikan untuk memperoleh hasil paling baik, dan biaya komputasi yang cukup tinggi diperlukan untuk menghitung jarak dari masing masing *query instance* pada semua *sample training*, akan tetapi SVM mempunyai kelebihan pada data testing yang mempunyai banyak *noise* dan tetapi lebih efektif jika digunakan pada jumlah data testing yang banyak.

V. KESIMPULAN

Support Vector Machine (SVM) dapat diterapkan pada proses analisis sentimen Komentar pada Video Debat Pilkada di YouTube. Sebelum diproses pada analisis, data komentar akan melewati berbagai tahapan yaitu *preprocessing* yang selanjutnya data komentar juga klasifikasi *sentiment analisis* memakai label *positif, negatif* dan *netral*, dengan acuan *data training* dan tentukan target komentar tersebut untuk calon dengan nama siapa, kemudian dihitung jumlah *sentiment user* untuk masing masing calon kepala daerah. Sehingga dapat diketahui kecenderungan *sentiment* masyarakat terhadap para calon kepala daerah tersebut, sehingga bisa jadi bahan evaluasi dan masukan untuk para calon dan tim sukses untuk memperbaiki kualitas dari masing masing calon kepala daerah serta memetakan kekuatannya.

DAFTAR PUSTAKA

[1] R. Feldman, “Techniques and applications for sentiment analysis,” Commun. ACM, vol. 56, no. 4, pp. 82–89, Apr. 2013.
[2] E. Haddi, X. Liu, and Y. Shi, “The role of text pre-processing in sentiment analysis,” Procedia Computer Science, vol. 17, no. 0, pp. 26 – 32, 2013, first International Conference on Information Technology and Quantitative Management.
[3] H. Tang, S. Tan, and X. Cheng, “A survey on sentiment detection of reviews,” Expert Systems with Applications, vol. 36, no. 7, pp. 10760 – 10773, 2009.
[4] M. Thelwall, K. Buckley, and G. Paltoglou, “Sentiment in twitter events,” J. Am. Soc. Inf. Sci. Technol., vol. 62, no. 2, pp. 406–418, Feb. 2011.
[5] D V Nagarjuna Devi, Chinta Kishore Kumar, Siriki Prasad, “A Feature Based Approach for Sentiment Analysis by Using Support Vector Machine”, 2016.
[6] Upma Kumari, Arvind K Sharma Dr., Dinesh Soni, “Sentiment Analysis of Smart Phone Product Review using SVM Classiification Technique”, 2017.

DOI: https://doi.org/10.35457/antivirus.v15i1.1539

[7] Nurulhuda Zainuddin, Ali Selamat, "Sentiment Analysis Using Support Vector Machine", 2014.
[8] Rofiqoh, U., Perdana, R. S., dan Fauzi, M. A. 2017. Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexicon Based Features. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 1, No. 12, Hal: 1725-1732.
[8] Wahid, D. H. dan Azhari. 2016. Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity. Indonesian Journal of Computing and Cybernetics Systems (IJCCS), Vol. 10 , No. 2, Hal: 207-218.
[9] Suyanto. 2019. Data Mining: untuk Klasifikasi dan Klasterisasi Data Edisi Revisi. Bandung: Informatika.
[10] Lu,Shuxia., Jin, Zhao., 2017, Improved Stochastic Gradient Descent Algorithm for SVM, International Journal of Recent Engineering Science(IJRES). Vol 4, Issue 4, Hal. 28 – 31.

DOI: https://doi.org/10.35457/antivirus.v15i1.1539