

**Analisis Sentimen Terhadap Anies Baswedan Menggunakan Metode Support Vector
Machine Studi Kasus Media Sosial Twitter**
*Sentiment Analysis of Anies Baswedan Using the Support Vector Machine Method Case Study of
Twitter Social Media*

Bagas Farasqa Nauval Akbar¹⁾, Deni Arifianto²⁾, Amalina Maryam Zakiyyah³⁾, Agus Milu Susetyo⁴⁾

¹Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember
email: bagasfarasqa33@gmail.com

²Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember
email: deniarifianto@unmuhjember.ac.id

³Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember
email: amalinamaryam@unmuhjember.ac.id

⁴Program Studi Pendidikan Bahasa dan Sastra Indonesia, Fakultas Keguruan dan Ilmu Pendidikan,
Universitas Muhammadiyah Jember
email: agusmilus@unmuhjember.ac.id

Abstrak

Anies Baswedan menjadi salah satu bakal calon presiden yang digulirkan media massa menjelang digelarnya Pemilihan Presiden 2024-2029 mendatang mulai menjadi topik pemberitaan media massa di Indonesia. Salah satu media sosial yang sering dipakai oleh publik yaitu Twitter. Banyaknya pengguna media sosial di Indonesia membuat media sosial terutama Twitter dapat digunakan secara efektif untuk melihat bagaimana pendapat dari masyarakat. Dari berbagai opini dan pendapat di twitter dibutuhkan sebuah teknik untuk membagi opini ke dalam kelas opini negatif, netral atau positif. Pada penelitian ini, digunakan Support Vector Machine Multiclass untuk proses klasifikasi dengan menggunakan metode One Against Rest dan Oversampling untuk mengatasi ketidakseimbangan kelas. Hasil dari penelitian ini mendapatkan kesimpulan bahwa masyarakat cenderung memiliki sentimen negatif terhadap Anies Baswedan. Hasil dari pengujian klasifikasi Support Vector Machine tanpa proses balancing data didapatkan nilai Accuracy sebesar 77%, Precision 76% dan Recall 72%, setelah proses balancing data didapatkan nilai Accuracy sebesar 97%, Precision 83% dan Recall 95%.

Keywords: Analisis sentimen, Twitter, SVM, Oversampling, One Against Rest.

Abstract

Anies Baswedan is one of the presidential candidates being rolled out by the mass media ahead of the upcoming 2024-2029 Presidential Election, starting to become the topic of mass media coverage in Indonesia. One of the social media that is often used by the public is Twitter. The large number of social media users in Indonesia means that social media, especially Twitter, can be used effectively to see what the public thinks. From the various opinions and opinions on Twitter, a technique is needed to divide opinions into negative, neutral or positive opinion classes. In this study, a Multiclass Support Vector Machine is used for the classification process using the One Against Rest and Oversampling method to overcome class imbalance. The results of this study conclude that people tend to have negative sentiments towards Anies Baswedan. The results of the Support Vector Machine classification test without the data balancing process obtained Accuracy values of 77%, Precision 76% and Recall 72%, after the data balancing process obtained Accuracy values of 97%, Precision 83% and Recall 95%.

Keywords: Sentiment analysis, Twitter, SVM, Oversampling, One Against Rest.

1. PENDAHULUAN

Anies Baswedan mampu menyita perhatian publik sepanjang tahun walau dia tergolong baru dalam dunia politik. Sepak terjangnya dalam kancah politik Indonesia tidak terhenti disitu saja, Anies Baswedan dikabarkan juga menjadi salah satu dari bakal calon presiden menjelang digelarnya Pemilihan Presiden tahun 2024-2029 mendatang, berita mengenai bakal calon presiden dan wakil presiden mulai menjadi topik pemberitaan media massa di Indonesia (Haris, Amalia & Hanafi, 2022). Hal ini tentu menarik perhatian masyarakat di media sosial, banyak spekulasi yang beredar dan menimbulkan pro dan kontra terutama oleh pengamat politik.

Berdasarkan fenomena di atas peneliti tertarik untuk mengkaji lebih dalam mengenai Anies Baswedan guna mendapatkan suatu data dan informasi yang dapat dijadikan referensi oleh tim sukses Anies Baswedan dalam melakukan komunikasi politik melalui opini dan pendapat masyarakat mengenai Anies Baswedan.

Opini dan pendapat masyarakat dapat disampaikan dengan berbagai macam media. Menurut data dari agensi pemasaran We Are Social dan layanan konten HootSuite (Kemp, 2022), Pada bulan Januari tahun 2022 pengguna internet di Indonesia mencapai 204,7 juta. Salah satu media sosial yang sering dipakai oleh publik untuk beropini yaitu Twitter. Twitter menjadi salah satu cara masyarakat dalam merespon berbagai hal melalui media sosial, baik berupa respon positif maupun negatif (Pane & Ramdan, 2022). Oleh sebab itu, twitter memiliki kelebihan dibanding media sosial lain dan twitter dapat digunakan secara efektif untuk pengumpulan data opini dari masyarakat, namun dalam pengumpulan data setelah dilakukan preprocessing dapat terjadi imbalance data yang dapat mempengaruhi hasil yang diperoleh.

Berdasarkan permasalahan yang telah diuraikan di atas, penulis mengusulkan *Support Vector Machine multiclass* dengan metode *One Vs Rest* dengan kernel polynominal dan *oversampling* yang akan diterapkan untuk analisis sentimen terhadap tweet publik terhadap Anies. Pada penelitian ini sentimen

akan diklasifikasikan dalam 3 kelas yaitu positif, negatif dan netral.

2. TINJAUAN PUSTAKA

A. Sentiment Analysis

Sentiment analysis adalah kombinasi dari *data mining* dan *text mining*. *Sentiment analysis* merupakan metode untuk mengolah dan memahami sikap, emosi, pendapat, penilaian, pandangan dari sekumpulan teks (Nomleni, P, 2015).

B. Text mining

Text mining merupakan suatu langkah dari analisis teks yang dapat diartikan sebagai suatu proses yang dilakukan untuk mendapatkan wawasan dan informasi yang berkualitas tentang pengguna berinteraksi dengan koleksi dokumen dalam data tekstual (Pristiyanti, Fauzi & Muflikhah, 2018).

C. Twitter

Media sosial twitter merupakan layanan jejaring sosial *microblog* daring yang memungkinkan pengguna untuk berbagi *tweet* atau pesan teks dengan batas 280 karakter (Pane & Ramdan, 2022). Twitter dioperasikan oleh Twitter Inc, yang didirikan oleh Jack Dorsey pada tahun 2006. Fitur yang terdapat di Twitter salah satunya adalah Tweet atau kicauan. Berdasarkan kicauan yang ada di twitter setiap harinya, tweet tersebut dapat dijadikan sebagai sumber informasi sehingga dapat digunakan untuk data pada penelitian.

D. Text Preprocessing

Text Preprocessing memiliki tujuan memproses data untuk dibersihkan dari data yang tidak perlu. Beberapa tahap *text preprocessing* yang dilakukan di penelitian ini yaitu *cleansing*, *case folding*, *normalization*, *tokenizing*, *stopword*, dan terakhir dilakukan *stemming* (Hermawan & Ismiati, 2020).

• Cleansing

Cleansing merupakan proses untuk membersihkan data dari angka, simbol dan huruf yang tidak diperlukan pada sebuah dokumen.

• **Case folding**

Case folding memiliki tujuan menyamaratakan penggunaan huruf kapital dengan mengubah kata menjadi huruf kecil semua.

• **Normalisasi Slangword**

Normalisasi Slangword adalah tahapan yang dilakukan untuk mengubah kata-kata yang tidak baku dari data twitter kembali menjadi bahasa yang baku menurut KBBI (Riyadullah & Romadhony, 2021) dengan menggunakan kamus ‘bahasa slang’ dari Colloquial Indonesian Lexicon oleh Nikmatun Aliyah Salsabila.

• **Tokenizing**

Tokenizing adalah proses memisahkan kalimat berdasarkan tiap kata yang menyusun kalimat tersebut (Indraloka & Santosa, 2017).

• **Stopword Removal**

Stopword Removal merupakan proses penyaringan kata-kata yang sering ataupun jarang muncul yang disebut sebagai *stopword* (Saif, Fernandez, He & Alani, 2014).

• **Stemming**

Stemming ini bertujuan untuk mentransformasikan kata untuk menjadi kata dasarnya (*root word*) dengan cara menghilangkan imbuhan yang berada di awal kalimat dan di akhir kalimat dari data hasil *stopword* (Indraloka & Santosa, 2017).

• **Oversampling**

Oversampling bertujuan untuk meningkatkan sampel kelas minoritas sehingga berjumlah sama dengan kelas mayoritas dengan menduplikasi kelas minoritas secara acak (He, Wenyu & Shuai, 2018).

E. Tf-Idf

Pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) merupakan suatu metode untuk menghitung bobot atau nilai dari setiap kata yang telah di ekstrak (Akbari, Novianty & Setianingsih, 2017). Menurut (Santoso, 2021) rumus perhitungan TF-IDF adalah

$$W_{t,d} = W_{tf,t} \times idf_t$$

Keterangan:

$W_{t,d}$ = Nilai dari Tf-Idf

W_{tft} = nilai suatu kata pada dokumen

idf_t = nilai inverse document frequency

F. Support Vector Machine

Algoritma *Support Vector Machine* memiliki konsep untuk pengklasifikasian data, yaitu menentukan *Hyperplane* terbaik. Dimana *hyperplane* terbaik didapatkan dengan memaksimalkan nilai margin. Jarak antara titik terdekat dengan *hyperplane* disebut sebagai *margin*. *Support Vector* merupakan himpunan data yang letaknya didalam *margin* (Pratiwi, 2020). Tahapan dalam metode SVM menurut (Tineges, 2020):

1. Menginisialisasi terhadap a_i , C , γ , dan ϵ .
2. hitung nilai matriks Hessian

$$D_{ij} = y_i y_j K(x_i, x_j) + \lambda^2$$

3. menghitung Nilai dari *error rate*, *delta alfa*, dan *alfa* dengan persamaan sebagai berikut

$$E_i = \sum_{j=1}^i a_j D_{ij}$$

Keterangan:

E_i = nilai *error rate*

a_j = nilai *alfa* ke-j

D_{ij} = nilai matriks Hessian

$\delta a_i = \min(\max[\gamma(1 - E_i), a_i], C - a_i)$

Keterangan:

a_i = nilai *alfa* ke i

γ = nilai gamma

E_i = *error rate*

C = Variabel slack

$a_i = a_i + \delta a_i$

Keterangan:

a_i = *alfa* ke i

δa_i = *delta alfa* ke i

4. Mencari nilai bias melalui persamaan berikut

$$b = -\frac{1}{2} \left(\sum_{i=1}^n a_i y_i K(x_i, x^-) + \sum_{i=1}^n a_i y_i K(x_i, x^+) \right)$$

Keterangan:

$K(x_i, x^-)$ = fungsi kernel negatif.

$K(x_i, x^+)$ = fungsi kernel positif.

5. Menguji dokumen menggunakan persamaan

$$f(x) = \sum_{i=1}^m a_i y_i K(x_i, x) + b$$

G. K-Fold Cross Validation

K-fold Cross Validation merupakan salah satu metode untuk mengukur suatu kinerja model algoritma dengan menghitung varian kinerja dari K model. Kemudian nilai rata-rata kinerja model adalah sebuah perkiraan kinerja model terhadap data uji (Putra, 2020).

H. Confusion Matrix

Confusion matrix adalah cara untuk memvisualisasikan kinerja sebuah model machine learning untuk tugas klasifikasi. *Confusion matrix* menunjukkan kombinasi jumlah data yang diprediksi dan jumlah data yang sebenarnya (Nugroho, 2019).

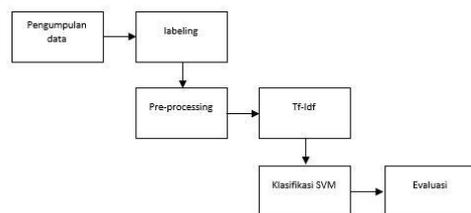
I. Python

Python adalah bahasa pemrograman yang dibuat oleh Guido Van Rossum. Menurut Guido Van Rossum, *Python* adalah bahasa pemrograman tingkat tinggi dan di desain untuk membaca kode dan sintaks yang bertujuan untuk memudahkan *programmer* untuk mengekspresikan konsep hanya dalam beberapa baris kode (Hardiansyah & Farizy, 2018). *Python* memiliki beberapa keunggulan seperti efisien, *reliability*, *interoperabilitas*, dan multifungsi juga dukungan komunitas yang memadai (Wahyono, 2018).

J. Google Colaboratory

Google Colaboratory, atau singkatnya disebut “Colab”, merupakan produk dari Google Research. Colab memungkinkan siapa saja menulis dan mengeksekusi kode *python* arbitrer melalui browser, dan sangat cocok untuk *machine learning*, analisis data, serta pendidikan (Carneiro, 2018).

3. METODE PENELITIAN



Gambar 1. Metode Penelitian

Sumber: Gambar penelitian sendiri

A. Pengumpulan Data

Pengumpulan data dilakukan dengan *crawling* dibantu dengan bahasa pemrograman *python* dengan memasukkan *Application Programming Interface* (API) Key Twitter lalu memasukkan kata kunci pencarian yaitu Anies Baswedan pada bulan november 2022 dan diperoleh data sebanyak 7524 data. Data yang diperoleh dari *crawling* disimpan dalam format *Comma Separated Value* (CSV).

B. Pelabelan

Pelabelan dilakukan dengan melabeli data secara manual oleh Bagas Farasqa Nauval Akbar lalu data hasil labeling dilakukan validasi oleh dosen Agus Milu Susetyo, S.Pd. M.Pd. Pemberian label pada setiap data berupa label negatif yang mengandung unsur penghinaan ataupun berkata kasar dengan menggunakan angka -1, netral yang berisi informasi dengan angka 0, dan label positif yang mengandung pujian dan dukungan dengan angka 1.

C. Text Preprocessing

Setelah proses pengumpulan data selesai dilakukan, karena dataset tersebut termasuk dalam data yang tidak terstruktur. Maka perlu terlebih dahulu dilakukan proses text preprocessing pada dataset. Berikut adalah alur dari text preprocessing antara lain, *cleansing*, *case folding*, *normalization*, *tokenizing*, *stopword removal* dan *stemming*. Setelah data bersih kemudian dilakukan *oversampling* data awal dengan jumlah 7524 data dengan kelas sentimen -1 berjumlah 3284 data, kelas 0 dengan 1779 data dan kelas 1 berjumlah 2461 data. Setelah *oversampling* jumlah data kelas -1, 0 dan 1 masing-masing berjumlah 3284 data sehingga keseluruhan data berjumlah 9852 data.

D. Pembobotan Tf-Idf

Setelah *preprocessing* selesai tahap selanjutnya adalah melakukan pembobotan TF-IDF terhadap data yang sudah bersih untuk mengubah data dari berupa kata menjadi data numerik yang mempunyai nilai dengan menggunakan *library TfIdfVectorizer*. Hasil pembobotan TF-IDF dapat dilihat pada Gambar 2.

```

iterasi 1
(0, 3373) 0.28775473639035576
(0, 5433) 0.31987357195583327
(0, 547) 0.0545283362100942
(0, 238) 0.051333221543346255
(0, 5876) 0.23686917560783782
(0, 4069) 0.2747568229495034
(0, 5350) 0.3542070434768742
(0, 2535) 0.21960722441541544
(0, 6167) 0.2776609069522088
(0, 823) 0.23991553084571382
(0, 3174) 0.5078813918729127
(0, 1895) 0.33374067835595117
    
```

Gambar 2. Hasil Tf-Idf fold ke-6 iterasi ke-1
 Sumber: Hasil penelitian sendiri

E. Klasifikasi Support Vector Machine

Setelah nilai TfIdf setiap dokumen ditemukan selanjutnya dilakukan Proses klasifikasi dengan algoritma *Support Vector Machine*. Hasil klasifikasi dengan *Support Vector Machine* dapat dilihat pada Gambar 3.

```

iterasi 1
0 True
1 True
2 True
3 False
4 True
...
1249 True
1250 True
1251 True
1252 True
1253 True
    
```

Gambar 3. Hasil prediksi pada *fold* ke-6 iterasi ke-1
 Sumber: Hasil penelitian sendiri

F. K-Fold Cross Validation

K-fold cross validation dilakukan untuk menemukan model terbaik. Hasil pengujian *k-fold cross validation* dilakukan sebanyak 15 percobaan dengan 4 skenario seperti *2 fold*, *3 fold*, *4 fold*, dan *6 fold* pada data sebelum *resampling* dan sesudah *resampling*. Hasil pembagian data dengan *K-fold* dapat dilihat pada Gambar 4.

```

iterasi 1
training: [1254 1255 1256 ... 7521 7522 7523] having: 6270
test: [ 0 1 2 ... 1251 1252 1253] having: 1254
    
```

Gambar 4. Hasil split data dengan *6-fold* iterasi ke-1
 Sumber: Hasil penelitian sendiri

G. Confusion Matrix

Confusion matrix menunjukkan kombinasi jumlah data yang diprediksi dan jumlah data yang sebenarnya. Ada empat istilah untuk mempresentasikan hasil proses klasifikasi, yaitu *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)* dan *False Negative (FN)*. Nilai tersebut dapat digunakan untuk mengukur

performa model seperti *accuracy*, *precision* dan *recall*.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

Hasil perhitungan *Accuracy*, *Precision* dan *Recall* dapat dilihat pada Gambar 5.

```

{'fit_time': array([5.93104362, 6.01644635, 6.5317111, 5.69541168, 5.71128821,
 6.3182714 ]), 'score_time': array([0.61872315, 0.57920098, 0.68232679, 0.61173463, 0.57834983,
 0.5845685 ]), 'test_acc': array([0.72966507, 0.74082935, 0.77591707, 0.76395534, 0.68508542,
 0.68740832]), 'train_acc': array([0.98708134, 0.97783094, 0.97974483, 0.98229665, 0.97751196,
 0.97591707]), 'test_prec_macro': array([0.74381342, 0.73132488, 0.74469208, 0.70151618, 0.70897751,
 0.71734752]), 'train_prec_macro': array([0.98688319, 0.97717196, 0.97983116, 0.98169136, 0.97785571,
 0.97596718]), 'test_rec_macro': array([0.71074728, 0.67630566, 0.72982362, 0.72662493, 0.6566739,
 0.65900408]), 'train_rec_macro': array([0.98508429, 0.9756404, 0.97767459, 0.979608, 0.97445617,
 0.97288383])}
    
```

Gambar 5. Hasil *Accuracy*, *Precision*, *Recall* pada *fold* ke-6 data sebelum *Oversampling*
 Sumber: Hasil penelitian sendiri

4. HASIL DAN PEMBAHASAN

Pembagian data latih dan data uji dilakukan menggunakan *k-fold* dengan *2-fold*, *3-fold*, *4-fold* dan *6-fold* pada data sebelum dan sesudah *oversampling* dimana tabel kuning merupakan data testing dan tabel putih merupakan data training. Pembagian *k-fold* dapat dilihat pada tabel berikut

Tahap pertama pembagian data menjadi 2 bagian 1 data training dan 1 data testing pada data sebelum *oversampling*. Hasil pembagian ditunjukkan Tabel 1.

Tabel 1. Hasil *2-fold* data sebelum *oversampling*

K=2	DATA	
Iterasi 1	3762	3762
Iterasi 2	3762	3762

Sumber: Hasil penelitian sendiri

Pembagian data menjadi 3 bagian 2 data training dan 1 data testing pada data sebelum *oversampling*. Hasil pembagian ditunjukkan Tabel 2.

Tabel 2. Hasil *3-fold* data sebelum *oversampling*

K=3	DATA		
Iterasi 1	2508	2508	2508
Iterasi 2	2508	2508	2508
Iterasi 3	2508	2508	2508

Sumber: Hasil penelitian sendiri

Pembagian data menjadi 4 bagian 3 data training dan 1 data testing pada data sebelum *oversampling*. Hasil pembagian ditunjukkan Tabel 3.

Tabel 3. Hasil 4-fold data sebelum *oversampling*

K=4	DATA			
Iterasi 1	1881	1881	1881	1881
Iterasi 2	1881	1881	1881	1881
Iterasi 3	1881	1881	1881	1881
Iterasi 4	1881	1881	1881	1881

Sumber: Hasil penelitian sendiri

Pembagian data menjadi 6 bagian 5 data training dan 1 data testing pada data sebelum *oversampling*. Hasil pembagian ditunjukkan Tabel 4.

Tabel 4. Hasil 6-fold data sebelum *oversampling*

K=6	DATA					
Iterasi 1	1254	1254	1254	1254	1254	1254
Iterasi 2	1254	1254	1254	1254	1254	1254
Iterasi 3	1254	1254	1254	1254	1254	1254
Iterasi 4	1254	1254	1254	1254	1254	1254
Iterasi 5	1254	1254	1254	1254	1254	1254
Iterasi 6	1254	1254	1254	1254	1254	1254

Sumber: Hasil penelitian sendiri

Tahap kedua pembagian data menjadi 2 bagian 1 data training dan 1 data testing pada data sesudah *oversampling*. Hasil pembagian ditunjukkan Tabel 5.

Tabel 5. Hasil 2-fold data sesudah *oversampling*

K=2	DATA	
Iterasi 1	4926	4926
Iterasi 2	4926	4926

Sumber: Hasil penelitian sendiri

Pembagian data menjadi 3 bagian 2 data training dan 1 data testing pada data sesudah *oversampling*. Hasil pembagian ditunjukkan Tabel 6.

Tabel 6. Hasil 3-fold data sesudah *oversampling*

K=3	DATA		
Iterasi 1	3284	3284	3284
Iterasi 2	3284	3284	3284
Iterasi 3	3284	3284	3284

Sumber: Hasil penelitian sendiri

Pembagian data menjadi 4 bagian 3 data training dan 1 data testing pada data sesudah *oversampling*. Hasil pembagian ditunjukkan Tabel 7.

Tabel 7. Hasil 4-fold data sesudah *oversampling*

K=4	DATA			
Iterasi 1	2463	2463	2463	2463
Iterasi 2	2463	2463	2463	2463
Iterasi 3	2463	2463	2463	2463
Iterasi 4	2463	2463	2463	2463

Sumber: Hasil penelitian sendiri

Pembagian data menjadi 6 bagian 5 data training dan 1 data testing pada data sesudah *oversampling*. Hasil pembagian ditunjukkan Tabel 8.

Tabel 8. Hasil 6-fold data sesudah *oversampling*

K=6	DATA					
Iterasi 1	1642	1642	1642	1642	1642	1642
Iterasi 2	1642	1642	1642	1642	1642	1642
Iterasi 3	1642	1642	1642	1642	1642	1642
Iterasi 4	1642	1642	1642	1642	1642	1642
Iterasi 5	1642	1642	1642	1642	1642	1642
Iterasi 6	1642	1642	1642	1642	1642	1642

Sumber: Hasil penelitian sendiri

Nilai dari accuracy data sebelum dilakukan *oversampling* dengan *2-fold*, *3-fold*, *4-fold* dan *6-fold* dapat dilihat pada Tabel 9.

Tabel 9. Accuracy pada data sebelum *oversampling*

Fold	Iterasi	Accuracy
2	Iterasi 1	0,716
	Iterasi 2	0,648
3	Iterasi 1	0,719
	Iterasi 2	0,761
	Iterasi 3	0,656
4	Iterasi 1	0,737
	Iterasi 2	0,752
	Iterasi 3	0,748
	Iterasi 4	0,664
6	Iterasi 1	0,729
	Iterasi 2	0,74
	Iterasi 3	0,775
	Iterasi 4	0,763
	Iterasi 5	0,685
	Iterasi 6	0,687

Sumber: Hasil penelitian sendiri

Nilai *accuracy* tertinggi pada data sebelum *oversampling* didapatkan pada *6-fold* iterasi 3 dengan nilai 0,775 atau 77%.

Nilai dari *accuracy* data sesudah dilakukan *oversampling* dengan *2-fold*, *3-fold*, *4-fold* dan *6-fold* dapat dilihat pada Tabel 10.

Tabel 10. *Accuracy* pada data sesudah *oversampling*

Fold	Iterasi	Accuracy
2	Iterasi 1	0,688
	Iterasi 2	0,699
3	Iterasi 1	0,764
	Iterasi 2	0,79
	Iterasi 3	0,853
4	Iterasi 1	0,777
	Iterasi 2	0,839
	Iterasi 3	0,796
	Iterasi 4	0,948
6	Iterasi 1	0,799
	Iterasi 2	0,814
	Iterasi 3	0,839
	Iterasi 4	0,79
	Iterasi 5	0,848
	Iterasi 6	0,977

Sumber: Hasil penelitian sendiri

Nilai *accuracy* tertinggi pada data sesudah *oversampling* didapatkan pada *6-fold* iterasi 6 dengan nilai 0,977 atau 97%.

Nilai dari *precision* data sebelum *oversampling* pada *2-fold* dapat dilihat pada Tabel 11.

Tabel 11. Nilai *precision fold* ke 2 data sebelum *oversampling*

kelas	pre	pre
-1	0,873862	0,618337
0	0,676724	0,833333
1	0,587769	0,66433

Sumber: Hasil penelitian sendiri

Nilai dari *precision* data sebelum *oversampling* pada *3-fold* dapat dilihat pada Tabel 12.

Tabel 12. Nilai *precision fold* ke 3 data sebelum *oversampling*

kelas	pre	pre	pre
-1	0,775566	0,862097	0,677661
0	0,699387	0,733516	0,807512
1	0,667595	0,640943	0,593132

Sumber: Hasil penelitian sendiri

Nilai dari *precision* data sebelum *oversampling* pada *4-fold* dapat dilihat pada Tabel 13.

Tabel 13. Nilai *precision fold* ke 4 data sebelum *oversampling*

kelas	pre	pre	pre	pre
-1	0,799266	0,891253	0,799092	0,680829
0	0,737226	0,726592	0,748201	0,84456
1	0,673418	0,609375	0,685596	0,594744

Sumber: Hasil penelitian sendiri

Nilai dari *precision* data sebelum *oversampling* pada *6-fold* dapat dilihat pada Tabel 14.

Tabel 14. Nilai *precision fold* ke 6 data sebelum *oversampling*

kelas	pre	pre	pre	pre	pre	pre
-1	0,7387	0,84303	0,90639	0,80172	0,74221	0,72771
0	0,7990	0,70491	0,72580	0,76439	0,79878	0,82352
1	0,6935	0,64601	0,60187	0,71842	0,58593	0,60079

Sumber: Hasil penelitian sendiri

Nilai *precision* tertinggi pada data sebelum *oversampling* didapatkan pada *6-fold* iterasi 4 dengan *precision* kelas -1 sebesar 0,801, kelas 0 sebesar 0,764 dan kelas 1 sebesar 0,718 dengan nilai rata-rata 76%.

Nilai dari *precision* data sesudah *oversampling* pada *2-fold* dapat dilihat pada Tabel 15.

Tabel 15. Nilai *precision fold* ke 2 data sesudah *oversampling*

kelas	pre	pre
-1	0,986891	0,550708
0	0,591698	0,964372
1	0,617201	0,663158

Sumber: Hasil penelitian sendiri

Nilai dari *precision* data sesudah *oversampling* pada *3-fold* dapat dilihat pada Tabel 16.

Tabel 16. Nilai *precision fold* ke 3 data sesudah *oversampling*

kelas	pre	pre	pre
-1	0,965139	0,938343	0,655709
0	0,735772	0,799547	0,977241
1	0,648508	0,672781	0,795886

Sumber: Hasil penelitian sendiri

Nilai dari *precision* data sesudah *oversampling* pada *4-fold* dapat dilihat pada Tabel 17.

Tabel 17. Nilai *precision fold* ke 4 data sesudah *oversampling*

kelas	pre	pre	pre	pre
-1	0,93617	0,951315	0,918342	0,518868
0	0,770522	0,807971	0,844646	0,995836
1	0,68	0,729638	0,669323	0,922489

Sumber: Hasil penelitian sendiri

Nilai dari *precision* data sesudah *oversampling* pada *6-fold* dapat dilihat pada Tabel 18.

Tabel 18. Nilai *precision fold* ke 6 data sesudah *oversampling*

kelas	pre	pre	pre	pre	pre	pre
-1	0,8809	0,9606	0,9204	0,8957	0,8406	0
0	0,7936	0,7900	0,8221	0,8521	0,9751	0,9840
1	0,7456	0,6661	0,7585	0,6729	0,6039	0,9841

Sumber: Hasil penelitian sendiri

Nilai *precision* tertinggi pada data sesudah *oversampling* didapatkan pada *6-fold* iterasi 3 dengan *precision* kelas -1 sebesar 0,920, kelas 0 sebesar 0,822 dan kelas 1 sebesar 0,758 dengan nilai rata-rata 83%.

Nilai dari *recall* pada data sebelum *oversampling* pada *2-fold* dapat dilihat pada Tabel 19.

Tabel 19. Nilai *recall fold* ke 2 data sebelum *oversampling*

kelas	rec	rec
-1	0,775087	0,935484
0	0,4	0,231388
1	0,835076	0,622332

Sumber: Hasil penelitian sendiri

Nilai dari *recall* pada data sebelum *oversampling* pada *3-fold* dapat dilihat pada Tabel 20.

Tabel 20. Nilai *recall fold* ke 3 data sebelum *oversampling*

kelas	rec	rec	rec
-1	0,814639	0,865672	0,881092
0	0,424581	0,49262	0,245714
1	0,782372	0,786842	0,7289

Sumber: Hasil penelitian sendiri

Nilai dari *recall* pada data sebelum *oversampling* pada *4-fold* dapat dilihat pada Tabel 21.

Tabel 21. Nilai *recall fold* ke 4 data sebelum *oversampling*

kelas	rec	rec	rec	rec
-1	0,819322	0,804696	0,865929	0,891452
0	0,505	0,503896	0,484848	0,288496
1	0,777778	0,837209	0,774648	0,74266

Sumber: Hasil penelitian sendiri

Nilai dari *recall* pada data sebelum *oversampling* pada *6-fold* dapat dilihat pada Tabel 22.

Tabel 22. Nilai *recall fold* ke 6 data sebelum *oversampling*

kelas	rec	rec	rec	rec	rec	rec
-1	0,8026	0,7953	0,8519	0,8874	0,8125	0,9016
0	0,5485	0,3944	0,5443	0,4965	0,3922	0,3060
1	0,7809	0,839	0,793	0,7958	0,7653	0,7717

Sumber: Hasil penelitian sendiri

Nilai *recall* tertinggi pada data sebelum *oversampling* didapatkan pada *6-fold* iterasi 3 dengan *recall* kelas -1 sebesar 0,851, kelas 0 sebesar 0,544 dan kelas 1 sebesar 0,793 dengan nilai rata-rata 72%.

Nilai dari *recall* pada data sesudah *oversampling* pada *2-fold* dapat dilihat pada Tabel 23.

Tabel 23. Nilai *recall fold* ke 2 data sesudah *oversampling*

kelas	rec	rec
-1	0,47307	0,88447
0	0,909524	0,533124
1	0,840413	0,80868

Sumber: Hasil penelitian sendiri

Nilai dari *recall* pada data sesudah *oversampling* pada *3-fold* dapat dilihat pada Tabel 24.

Tabel 24. Nilai *recall fold* ke 3 data sesudah *oversampling*

kelas	rec	rec	rec
-1	0,649464	0,706821	0,898104
0	0,811659	0,831567	0,802378
1	0,890472	0,86755	0,911232

Sumber: Hasil penelitian sendiri

Nilai dari *recall* pada data sesudah *oversampling* pada *4-fold* dapat dilihat pada Tabel 25.

Tabel 25. Nilai *recall fold* ke 4 data sesudah *oversampling*

kelas	rec	rec	rec	rec
-1	0,678881	0,820319	0,732465	0,948276
0	0,789675	0,8463	0,806916	0,931818
1	0,884828	0,865772	0,871595	0,976879

Sumber: Hasil penelitian sendiri

Nilai dari *recall* pada data sesudah *oversampling* pada *6-fold* dapat dilihat pada Tabel 26.

Tabel 26. Nilai *recall fold* ke 6 data sesudah *oversampling*

kelas	rec	rec	rec	rec	rec	rec
-1	0,7524	0,7743	0,8328	0,7227	0,8792	0
0	0,7731	0,7900	0,8372	0,7884	0,8289	0,9768
1	0,8602	0,9039	0,8495	0,8746	0,8683	0,9781

Sumber: Hasil penelitian sendiri

Nilai *recall* tertinggi pada data sesudah *oversampling* didapatkan pada *4-fold* iterasi 4 dengan *recall* kelas -1 sebesar 0,948, kelas 0 sebesar 0,931 dan kelas 1 sebesar 0,976 dengan nilai rata-rata 95%.

5. KESIMPULAN

A. Kesimpulan

Hasil dari analisis sentimen dengan menggunakan *Support Vector Machine* terhadap Anies Baswedan pada media sosial twitter tanpa *balancing data*, didapatkan *accuracy* sebesar 77%, *precision* 76%, dan *recall* sebesar 72% dan hasil pengujian menggunakan *oversampling* dapat meningkatkan nilai dari *accuracy*, *precision*, dan *recall* menjadi *accuracy* 97%, *precision* menjadi 83%, dan *recall* menjadi 95%.

B. Saran

Untuk penelitian selanjutnya sebaiknya memperbanyak kosa kata dari kamus slang yang digunakan untuk normalisasi pada *preprocessing* data. Dapat pula menggunakan kernel lain yang pada algoritma *Support Vector Machine* dan metode yang lain seperti *One Versus One* untuk membandingkan akurasi dengan model SVM yang telah dibangun pada penelitian ini. Selain itu penelitian selanjutnya

dapat menggunakan algoritma klasifikasi lain sehingga dapat membandingkan hasil uji untuk mencari algoritma terbaik.

6. DAFTAR PUSTAKA

- Akbari, M. I. H. A. D., Novianty, A., & Setianingsih, C. (2017). Analisis Sentimen Menggunakan Metode Learning Vector Quantization. *eProceedings of Engineering*, 4(2).
- Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of google colab as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677-61685.
- Hardiansyah, & Salman Farizy. (2020). Perancangan Sistem Informasi Persediaan Barang Dengan Menggunakan Python 3.6.3 Dan My Sql Untuk Umkm. *Jurnal Ilmu Komputer*, 3(3), 1.
- Haris, A., Amalia, A., & Hanafi, K. (2022). Citra Politik Anies Baswedan Di Media Massa. *CommuniVerse: Jurnal Ilmu Komunikasi*, 7(2), 15-24.
- He, H., Zhang, W., & Zhang, S. (2018). A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications*, 98, 105–117. <https://doi.org/10.1016/j.eswa.2018.01.012>
- Hermawan, L., & Ismiati, M. B. (2020). Pembelajaran text preprocessing berbasis simulator untuk mata kuliah information retrieval. *Jurnal Transformatika*, 17(2), 188-199.
- Kemp, S. (2022). Digital 2022: Another Year Of Bumper Growth.
- Nomleni, P. (2015). Sentiment Analysis Menggunakan Support Vector Machine (Svm) (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).

- Nugroho, K. S. (2019). Confusion Matrix untuk Evaluasi Model pada Supervised Learning.
- Pane, S. F., & Ramdan, J. (2022). Pemodelan Machine Learning: Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM Menggunakan Data Twitter. *Jurnal Sistem Cerdas*, 5(1), 12-20.
- Pratiwi, K. S. (2020). Support Vector Machine Clasiffication with Python
- Pristiyanti, R. I., Fauzi, M. A., & Muflikhah, L. (2018). Sentimen Analisis Peringkasan Review Film Menggunakan Metode Information Gain dan K-Nearest Neighbor. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*
- Riyaddulloh, R., & Romadhony, A. (2021). Normalisasi Teks Bahasa Indonesia Berbasis Kamus Slang Studi Kasus: Tweet Produk Gadget Pada Twitter. *eProceedings of Engineering*, 8(4).
- Santoso, G. T. (2021). Analisis Sentimen Pada Tweet Dengan Tagar# bpjsrasarentenir Menggunakan Metode Support Vectore Machine (SVM) (Doctoral dissertation, Universitas Islam Riau).
- Sholihati, I. D., Triayudi, A., & Tineges, R. (2020). Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter. *Jurnal Media Informatika Budidarma*, 650-658.
- Wahyono, T. (2018). *Fundamental of Python for Machine Learning: Dasar-Dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan*. Yogyakarta: Gava Media.