

Perbandingan Metode Naïve Bayes dan SVM dalam Analisis Sentimen Netizen Twitter Terhadap Isu Kemenkeu


A. Anugrah Aqsa^{a,1,*}, Irawati^{a,2}, dan Lukman Syafie^{b,3}

^a Program Studi Teknik Informatika, Universitas Muslim Indonesia, Jl. Urip Sumoharjo KM.05, Makassar dan 90231, Indonesia

¹ andianugrahaqsa@gmail.com; ² irawati.irawati@umi.ac.id; ³ lukman.syafie@umi.ac.id;
*corresponding author

INFORMASI ARTIKEL	ABSTRAK
<p>Diterima : 22 – 07 – 2023 Direvisi : 30 – 07 – 2023 Diterbitkan : 04 – 12 – 2023</p> <hr/> <p>Kata Kunci: Analisis Sentimen Kemenkeu Naïve Bayes SVM Twitter</p>	<p>Pada awal bulan Maret 2023 Menkopolhukan, Bapak Mahfud MD menyampaikan bahwasanya adanya dugaan transaksi yang mencurigakan yang terjadi di Kemenkeu berdasarkan dari laporan temuan PPATK kepada Mahfud MD, sontak hal tersebut menjadi sorotan di berbagai media sosial salah satunya Twitter, beragam <i>tweet</i> yang dilontarkan oleh netizen di Twitter, ada yang memberikan <i>tweet</i> positif, negatif, dan juga netral. Pada penelitian ini bertujuan untuk membandingkan metode Naïve Bayes dan SVM dalam analisis sentimen netizen Twitter terhadap isu Kemenkeu. Penelitian ini menunjukkan bahwa sentimen netizen didominasi dengan sentimen negatif kemudian diikuti sentimen positif, dan terakhir sentimen netral. Hasil pengujian klasifikasi terhadap kedua metode tersebut didapatkan dari membagi secara acak dataset menjadi dua bagian yaitu data latih dan data uji dengan rasio 70:30. Setelah dilakukan pengujian ditemukan bahwa Naïve Bayes mendapatkan nilai akurasi sebesar 71,7%, presisi sebesar 55,2%, <i>recall</i> sebesar 45,3%, dan <i>f1-score</i> sebesar 44,8%, sedangkan pada SVM mendapatkan nilai akurasi sebesar 74%, presisi sebesar 87,8%, <i>recall</i> sebesar 49,1%, dan <i>f1-score</i> sebesar 49,8%.</p>

This is an open access article under the [CC-BY-SA](#) license



I. Pendahuluan

Analisis sentimen merupakan sebuah pengamatan yang dikelola berdasarkan pendapat, sentimen, evaluasi, sikap, dan perasaan orang lain, terutama berdasarkan apa yang mereka tulis. Pendapat lain menyebutkan bahwa analisis sentimen dilakukan untuk mengetahui apa yang dipikirkan orang lain berdasarkan informasi, seperti opini. Dari kedua pendapat tersebut, dapat dikatakan bahwa analisis sentimen dilakukan atas pendapat tertulis. Orang sering kali mengungkapkan dan menuliskan pendapatnya di media sosial karena perkembangan era digital yang membuat kita tidak bisa lepas dari media sosial [1].

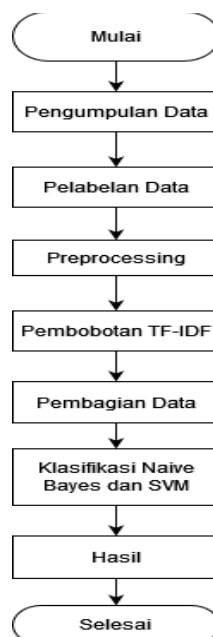
Pada penelitian sebelumnya melakukan analisis sentimen opini masyarakat terhadap vaksinasi booster Covid-19 dengan perbandingan metode Naive Bayes, *Decision Tree* dan SVM, Hasil pada penelitian ini menunjukkan skor AUC terbesar jatuh kepada model SVM(75.40%), namun untuk presisi yang lebih akurat jatuh kepada model Naive Bayes (83.81%). Selain itu, terdapat *confusion matrix* yang menunjukkan bahwa uji coba model Naive Bayes yang dilakukan berjalan dengan baik [1]. Penelitian terkait lainnya yaitu melakukan analisis sentimen masyarakat terhadap kinerja Dewan Perwakilan Rakyat (DPR) yang diungkapkan melalui media sosial twitter dengan menggunakan metode Naive Bayes Classifier. Hasil dari penelitian ini didapatkan bahwa DPR mendapatkan 95 *tweet* positif dengan polaritas 0.75 atau 75% sentimen positif, 693 *tweet* netral dengan polaritas 0.79 atau 79% sentimen netral dan 758 *tweet* negatif dengan polaritas 0.82 atau 82% sentimen negatif dengan accuracy score 0.8 atau 80% berdasarkan data testing sebanyak 20% [2]. Adapun penelitian terkait lainnya yaitu perbandingan akurasi dan waktu proses algoritma K-NN dan SVM dalam analisis sentimen Twitter yang dilakukan oleh Muhammad Rangga Aziz Nasution dan Mardhiya Hayaty. Adapun hasil pada perhitungan akurasi menunjukkan bahwa metode *Support Vector Machine* lebih unggul dengan nilai 89,70% tanpa K-Fold Cross Validation dan 88,76% dengan K-Fold Cross Validation. Sedangkan pada perhitungan waktu proses metode K-Nearest Neighbor lebih unggul dengan waktu proses 0.0160s tanpa K-Fold Cross Validation dan 0.1505s dengan K-Fold Cross Validation [3]. Twitter sebagai salah satu platform yang banyak digunakan oleh masyarakat Indonesia. *We Are Social* mencatat pada tahun 2022, pengguna Twitter di Indonesia mencapai 18,45 juta. Hal tersebut membuat Indonesia berada pada urutan kelima dengan pengguna Twitter terbanyak di dunia [4]. Seperti media sosial lainnya, Twitter merupakan platform media

sosial yang memungkinkan penggunaannya untuk berinteraksi secara personal ataupun terbuka. Berbagai topik ramai diperbincangkan di Twitter mulai dari politik, ekonomi, sosial, budaya, dan hukum [5]. Pada awal bulan Maret 2023 Menkopolhukan, Bapak Mahfud MD menyampaikan bahwasanya adanya dugaan transaksi yang mencurigakan yang terjadi di Kemenkeu berdasarkan dari laporan temuan PPATK kepada Mahfud MD, sontak hal tersebut menjadi sorotan di Twitter, bagaimana tidak, pasalnya Kementerian Keuangan merupakan instansi pemerintah yang memiliki peranan yang sangat penting dalam mengelolah keuangan negara dan dengan peranan tersebut terdengar kabar bahwa adanya dugaan kejanggalan transaksi yang dilakukan oleh oknum-oknum di dalam instansi tersebut, hal itu tentu saja menimbulkan bermacam *tweet* yang diposting oleh netizen di Twitter, ada yang memberikan *tweet* negatif, positif dan juga netral. Berdasarkan hal tersebut penulis tertarik untuk melakukan penelitian untuk menganalisis sentimen netizen Twitter terkait isu tersebut dengan menggunakan metode klasifikasi Naive Bayes dan *Support Vector Machine* yang nantinya hasil kedua metode tersebut akan dibandingkan untuk menentukan metode yang mana yang lebih baik dalam melakukan analisis sentimen terhadap isu Kemenkeu tersebut.

II. Metode

A. Tahapan Penelitian

Tahapan penelitian ini berisikan tahapan-tahapan penelitian yang akan dilakukan, tahapan penelitian ini berdasarkan tahapan metode penelitian yang tercantum pada gambar di bawah:



Gambar 1. Tahapan penelitian

Penjelasan:

- 1) Tahap pengumpulan data
 - a) Studi Pustaka
Mencari literatur yang berkaitan dengan penelitian yang dilakukan.
 - b) *Crawling* Data
Crawling data adalah metode pengumpulan data dengan cara mengambil data dari media sosial Twitter. Pada penelitian ini menggunakan *library* sncscrape untuk melakukan *crawling* data pada data *tweet* Twitter.
- 2) Tahap pelabelan data
Pelabelan data dilakukan secara manual dengan membagi data tersebut menjadi 3 kelas, yaitu positif, negatif, dan netral.
- 3) Tahap *Preprocessing*
Adapun tahapan pada tahap *preprocessing* di antaranya:
 - a) *Cleansing*
Pada proses ini, semua karakter didalam posts yang bukan alfabet dihapus sehingga dapat mengurangi karakter yang tidak dikehendaki dan tidak memiliki arti dalam analisis sentimen. Karakter tersebut seperti angka, #, @, emoji, hingga *link* dari sebuah situs web yang ada didalam sebuah *posts* [6].
 - b) *Case Folding*

Case folding yang dilakukan untuk menyeragamkan bentuk huruf kedalam bentuk *lowercase* [7].

c) *Tokenizing*

Tokenizing dilakukan untuk memotong kalimat berdasarkan kata yang menyusunnya dengan menghilangkan delimiter yang dimiliki seperti *whitespace*, *enter*, tabulasi, spasi dan sebagainya sehingga menghasilkan kata yang berdiri sendiri [7].

d) *Normalization*

Normalisasi merupakan tahapan yang dilakukan untuk mengubah bentuk kata pada teks kedalam bentuk yang lebih baku [7].

e) *Stopword Removal*

Stopword removal merupakan tahap penghilangan kata yang tidak diperlukan karena dianggap tidak mempengaruhi makna dan informasi yang terkandung dalam teks [7].

f) *Stemming*

Stemming merupakan proses mengubah kata yang berimbuhan menjadi kata dasar [6].

4) Tahap Pembobotan *TF-IDF*

Tahap di mana melakukan pembobotan kata dengan menggunakan *TF-IDF*.

5) Tahap Pembagian Data

Pembagian data dilakukan dengan membagi data menjadi 2 yaitu data latih dan data uji dengan rasio 70:30

6) Tahap klasifikasi

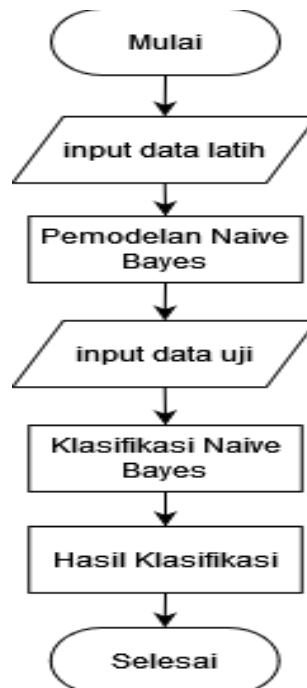
Tahap klasifikasi adalah tahap dilakukannya proses pemodelan menggunakan data latih serta melakukan klasifikasi terhadap data uji berdasarkan model yang telah dihasilkan.

7) Hasil

Pada tahap ini telah diketahui perbandingan performa dari metode Naïve Bayes dan *SVM* dalam analisis sentimen netizen Twitter terhadap isu Kemenkeu dengan menggunakan *confusion matrix*.

B. Desain Penelitian

1) *Flowchart* Naïve Bayes

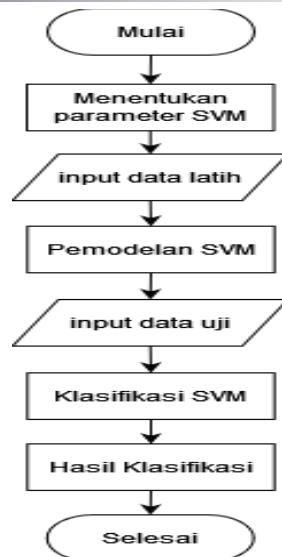


Gambar 2. *Flowchart* Naïve Bayes

Adapun keterangan dari gambar 2 diatas adalah sebagai berikut:

- a) Mulai
- b) Memasukkan data latih
- c) Melakukan pemodelan terhadap data *training*
- d) Memasukkan data uji untuk dilakukan pengujian klasifikasi
- e) Melakukan klasifikasi terhadap data uji berdasarkan hasil *training* sebelumnya
- f) Mendapatkan hasil klasifikasi
- g) Selesai

2) *Flowchart* *SVM*



Gambar 3. Flowchart SVM

Adapun keterangan dari gambar 3 diatas adalah sebagai berikut:

- a) Mulai
- b) Menentukan parameter
- c) Memasukkan data latih
- d) Melakukan pemodelan terhadap data *training*
- e) Memasukkan data uji untuk dilakukan pengujian klasifikasi
- f) Melakukan klasifikasi terhadap data uji berdasarkan hasil *training* sebelumnya
- g) Mendapatkan hasil klasifikasi
- h) Selesai

C. Tinjauan Pustaka

1) Kementerian Keuangan (Kemenkeu)

Kementerian Keuangan merupakan instansi pemerintah yang memiliki peranan vital dalam suatu negara. Peranan vital Kementerian Keuangan adalah mengelola keuangan negara dan membantu pimpinan negara di bidang keuangan dan kekayaan negara. Oleh karena itu, Kementerian Keuangan dikatakan sebagai penjaga keuangan negara [8].

2) *TF-IDF*

TF-IDF merupakan metode perhitungan bobot setiap kata yang paling umum digunakan. Metode ini merupakan algoritma yang melakukan penggabungan dua metode yaitu konsep frekuensi kemunculan *term* dalam sebuah dokumen dan inverse frekuensi dokumen yang mengandung kata tersebut, sehingga proposisi jumlah dokumen yang ditemukan kembali dan dianggap relevan dengan query akan meningkat [9].

3) Naïve Bayes

Naïve Bayes yaitu salah satu metode *machine learning* yang memanfaatkan perhitungan probabilitas dan *statistic* yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, cara kerja Naive Bayes yaitu memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya [8]. NBC bisa disebut sebagai Multinomial Naïve Bayes merupakan model penyederhanaan Metode Bayes yang cocok dalam klasifikasi teks atau dokumen [10].

4) *Support Vector Machine (SVM)*

Support Vector Machine merupakan sistem pembelajaran yang menggunakan hipotesis berupa fungsi-fungsi linear dalam sebuah fitur yang berdimensi tinggi dan dilatih dengan menggunakan algoritma pembelajaran yang didasarkan dengan teori optimasi. *Support Vector Machine* diperkenalkan pertama kali oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep konsep unggulan dalam bidang *pattern recognition*. Tingkat akurasi pada model yang akan dihasilkan oleh proses peralihan pada *SVM* sangat bergantung pada fungsi kernel dan parameter yang digunakan. Berdasarkan dengan karakteristiknya metode *SVM* dibagi menjadi dua yaitu linear dan non linear, *SVM* linear merupakan data yang dipisahkan secara linear yaitu memisahkan dua kelas pada *hyperplane* dengan *soft margin*. Sedangkan non linear yaitu merupakan fungsi dari kernel *trick* terhadap ruang yang berdimensi tinggi [11].

5) *Confusion Matrix*

Confusion matrix adalah matriks untuk perhitungan hasil klasifikasi. *Confusion matrix* berisi hasil klasifikasi yang dapat digunakan untuk mengukur performa model klasifikasi. Melalui *confusion matrix*, akurasi, presisi, *recall*, dan *f1-score* dapat diketahui [12].

Akurasi didapat dengan menghitung rasio dari nilai prediksi benar terhadap data. Artinya nilai akurasi berguna untuk mengukur tingkat akuratnya model klasifikasi dalam memprediksi dengan benar. Adapun persamaan dari akurasi dapat dilihat pada rumus persamaan (1)[13].

$$Akurasi = \frac{TP+TNet+TNeg}{Total\ data\ latih} \quad (1)$$

Presisi untuk kelas c merupakan perbandingan dari jumlah dokumen yang diklasifikasikan benar pada kelas c dengan jumlah dokumen yang diklasifikasikan sebagai kelas c. Precision pada suatu kelas c dapat dihitung dengan menggunakan persamaan (2)[14].

$$Presisi_c = \frac{TP(kelas\ c)}{Perdasi(kelas\ c)} \quad (2)$$

Recall untuk kelas c merupakan perbandingan dari jumlah dokumen yang diklasifikasikan benar pada kelas c dengan jumlah seluruh dokumen yang sebenarnya berada pada kelas c. Perhitungan *recall* pada suatu kelas c dapat dilakukan dengan persamaan (3)[14].

$$Recall = \frac{TP}{Total(kelas\ c)} \quad (3)$$

F1-score didapat dengan menghitung rata-rata presisi dan recall yang dibobokan. Artinya *f1-score* berguna untuk mengukur seberapa baiknya presisi dan *recall* pada model klasifikasi. Adapun persamaan dari *f1-score* dapat dilihat pada rumus persamaan (4)[13].

$$F1 - Score = \frac{2 \times Presisi \times Recall}{Presisi + Recall} \quad (4)$$

D. Metode Penelitian

1) Metode pengumpulan data

a) Studi Pustaka

Mencari literatur yang berkaitan dengan penelitian yang dilakukan.

b) *Crawling* Data

Crawling data adalah metode pengumpulan data dengan cara mengambil data dari media sosial Twitter. Pada penelitian ini menggunakan *library* snsrape untuk melakukan *crawling* data pada data *tweet* Twitter.

2) Metode klasifikasi

Metode klasifikasi menggunakan algoritma Naïve Bayes dan *Support Vector Machine*

3) Lokasi penelitian

Adapun lokasi penelitian yakni di Fakultas Ilmu Komputer UMI Makassar, Sulawesi Selatan. Penelitian dilakukan selama kurang lebih 4 bulan terhitung dari bulan Maret – Juni 2023.

4) Instrumen Penelitian

a) Perangkat Keras (*Hardware*)

Perangkat keras atau *hardware* yang digunakan adalah laptop Asus dengan spesifikasi sebagai berikut:

- *Processor AMD Quad Core R5-2500U*
- *RAM 8 GB*
- *SSD 256 GB.*

b) Perangkat Lunak (*Software*)

- Windows 10 Home 64 Bit sebagai sistem operasi.
- Python sebagai bahasa pemrograman yang digunakan dalam penelitian ini.
- VS Code sebagai *text editor*
- Jupyter *extention* untuk menampilkan susunan kode seperti pada tampilan Jupyter Notebook
- Google Chrome sebagai web browser
- Microsoft Word untuk menulis penelitian.

III. Hasil dan Pembahasan

A. Hasil Penelitian

1) Pengumpulan data

Data *tweet* yang diambil berjumlah 1000 data yang diambil dari *keyword* “Kemenkeu PPATK” dari tanggal 8 maret 2023 sampai tanggal 7 april 2023 dengan menggunakan *library* *snsrape*.

2) Pelabelan data

Pelabelan data dilakukan secara manual dengan memberikan 3 label yaitu positif, netral, dan negatif.

3) *Preprocessing*

Berikut adalah tahapan *preprocessing* pada penelitian ini:

a) *Cleansing*

Pada tahap ini melakukan pembersihan pada data *tweet* dengan menghilangkan karakter seperti angka, tanda baca, @, #, hingga *link*.

Tabel 1. *Cleansing*

Sebelum <i>cleansing</i>	Sesudah <i>cleansing</i>
@KemenkeuRI @PPATK Yok bisa yok. Kemenkeu terus mengambil langkah tegas dan bekerja sama dengan seluruh pihak untuk menegakkan integritas di lingkungan Kemenkeu.	Yok bisa yok Kemenkeu terus mengambil langkah tegas dan bekerja sama dengan seluruh pihak untuk menegakkan integritas di lingkungan Kemenkeu

b) *Case folding*

Pada tahap ini mengubah format data *tweet* ke dalam bentuk *lowercase*

Tabel 2. *Case Folding*

Sebelum <i>case folding</i>	Sesudah <i>case folding</i>
Yok bisa yok Kemenkeu terus mengambil langkah tegas dan bekerja sama dengan seluruh pihak untuk menegakkan integritas di lingkungan Kemenkeu	yok bisa yok kemenkeu terus mengambil langkah tegas dan bekerja sama dengan seluruh pihak untuk menegakkan integritas di lingkungan kemenkeu

c) *Tokenizing*

Pada tahap ini mengubah data *tweet* yang semulanya berbentuk kalimat dipisah-pisah menjadi perkata dengan menggunakan fungsi *WordPunctTokenizer* dari *library* *nlTK*.

Tabel 3. *Tokenizing*

Sebelum <i>tokenizing</i>	Sesudah <i>tokenizing</i>
yok bisa yok kemenkeu terus mengambil langkah tegas dan bekerja sama dengan seluruh pihak untuk menegakkan integritas di lingkungan kemenkeu	[yok, bisa, yok, kemenkeu, terus, mengambil, langkah, tegas, dan, bekerja, sama, dengan, seluruh, pihak, untuk, menegakkan, integritas, di, lingkungan, kemenkeu]

d) *Normalization*

Pada tahap ini dilakukan normalisasi terhadap kata yang tidak baku menjadi kata yang lebih baku, pada tahap ini menggunakan kamus normalisasi.

Tabel 4. *Normalization*

Sebelum <i>normalization</i>	Sesudah <i>normalization</i>
[yok, bisa, yok, kemenkeu, terus, mengambil, langkah, tegas, dan, bekerja, sama, dengan, seluruh, pihak, untuk, menegakkan, integritas, di, lingkungan, kemenkeu]	[ayo, bisa, ayo, kemenkeu, terus, mengambil, langkah, tegas, dan, bekerja, sama, dengan, seluruh, pihak, untuk, menegakkan, integritas, di, lingkungan, kemenkeu]

e) *Stopword Removal*

Tahap ini bertujuan untuk menghilangkan kata yang dianggap kurang penting, tahap ini menggunakan kamus *stopword* dari *library* *nlTK*.

Tabel 5. *Stopword removal*

Sebelum <i>stopword removal</i>	Sesudah <i>stopword removal</i>
[ayo, bisa, ayo, kemenkeu, terus, mengambil, langkah, tegas, dan, bekerja, sama, dengan, seluruh, pihak, untuk, menegakkan, integritas, di, lingkungan, kemenkeu]	[ayo, ayo, kemenkeu, mengambil, langkah, menegakkan, integritas, lingkungan, kemenkeu]

f) *Stemming*

Tahap ini bertujuan untuk mengubah setiap kata menjadi kata dasar dari kata yang memiliki imbuhan, pada tahap ini menggunakan *library* Sastrawi.

Tabel 6. *Stemming*

Sebelum <i>stemming</i>	Sesudah <i>stemming</i>
[ayo, ayo, kemenkeu, mengambil, langkah, menegakkan, integritas, lingkungan, kemenkeu]	ayo ayo kemenkeu ambil langkah tegak integritas lingkung kemenkeu

4) Pembobotan *TF-IDF*

Setelah melakukan *preprocessing* data selanjutnya melakukan pembobotan *TF-IDF*. Tahap ini menggunakan fungsi *TfidfVectorizer* dari *library* sklearn untuk melakukan pembobotan *TF-IDF*.

5) Pembagian data

Data yang telah dilakukan pembobotan *TF-IDF* kemudian dibagi menjadi 2 dataset yaitu data latih dan data uji dengan rasio 70:30. Pembagian data dilakukan secara acak menggunakan fungsi *train_test_split* dari *library* sklearn.

6) Klasifikasi

Proses klasifikasi dilakukan dengan melakukan *training* atau pemodelan terhadap data latih yang telah dibagi sebelumnya kemudian dilakukan pengujian model terhadap data latih. Adapun untuk penilaian hasil klasifikasi ditampilkan dalam bentuk *confusion matrix* 3 X 3.

Berikut hasil klasifikasi dari setiap metode:

a) Naïve Bayes

Proses klasifikasi menggunakan fungsi *MultinomialNB* dari *library* sklearn untuk melakukan proses *training* dan *testing* pada dataset yang ada. Berikut kode program yang digunakan:



```

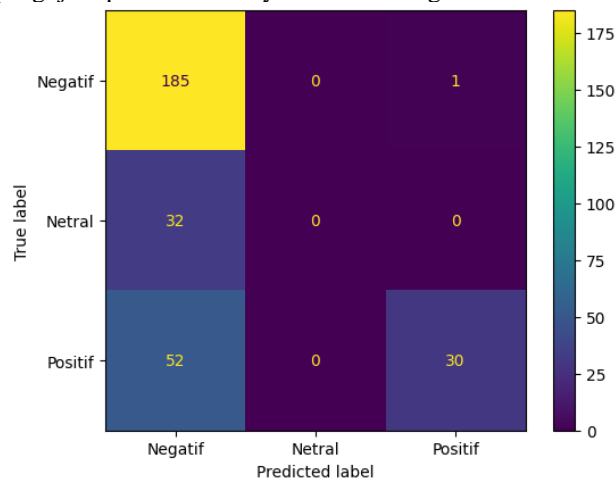
1 from sklearn.naive_bayes import MultinomialNB
2
3 nbc = MultinomialNB()
4 nbc.fit(x_train, y_train)
5 hasil_nbc = nbc.predict(x_test)
6
7 cm_display = ConfusionMatrixDisplay(
8     confusion_matrix = confusion_matrix(y_test, hasil_nbc),
9     display_labels = ["Negatif", "Netral", "Positif"]
10 )
11
12 cm_display.plot()
13 plt.show()
14
15 print(classification_report(y_test, hasil_nbc, digits=3, zero_division=0))

```

Gambar 4. Kode program klasifikasi Naïve Bayes

Gambar di atas merupakan potongan kode program untuk melakukan klasifikasi Naïve Bayes dengan menggunakan *library* sklearn serta menampilkan *confusion matrix* dan hasil penilaian dari proses klasifikasi Naïve Bayes.

Adapun hasil pengujian pada Naïve Bayes adalah sebagai berikut:



Gambar 5. *Confusion matrix* pengujian Naïve Bayes

Adapun perhitungan nilai akurasi, presisi, *recall*, dan *f1-score* berdasarkan gambar 5 sebagai berikut dengan menggunakan persamaan (1), (2), (3), dan (4):

- Akurasi

$$Akurasi = \frac{30+0+185}{300} = 71,7\%$$
- Presisi

$$Presisi_{positif} = \frac{30}{31} = 96,8\%$$

$$Presisi_{netral} = \frac{0}{0} = 0\%$$

$$Presisi_{negatif} = \frac{185}{269} = 68,8\%$$

$$Rata - rata = \frac{96,8\%+0\%+68,8\%}{3} = 55,2\%$$
- Recall

$$Recall_{positif} = \frac{30}{82} = 36,6\%$$

$$Recall_{netral} = \frac{0}{32} = 0\%$$

$$Recall_{negatif} = \frac{185}{186} = 99,5\%$$

$$Rata - rata = \frac{36,6\%+0\%+99,5\%}{3} = 45,3\%$$
- F1-score

$$F1 - score_{positif} = \frac{2 \times 96,8\% \times 36,6\%}{96,8\% + 36,6\%} = 53,1\%$$

$$F1 - score_{netral} = \frac{2 \times 0\% \times 0\%}{0\% + 0\%} = 0\%$$

$$F1 - score_{negatif} = \frac{2 \times 68,8\% \times 99,5\%}{68,8\% + 99,5\%} = 81,3\%$$

$$Rata - rata = \frac{53,1\%+0\%+81,3\%}{3} = 44,8\%$$

Kesimpulan dari hasil perhitungan nilai akurasi, presisi, *recall*, dan *f1-score* di atas pada hasil klasifikasi Naïve Bayes, untuk lebih jelasnya dapat dilihat pada table berikut:

Tabel 7. Hasil *perhitungan* nilai akurasi, presisi, *recall*, dan *f1-score* Naïve Bayes

Kelas	Presisi	Recall	F1-Score	Akurasi
Positif	96,8%	36,6%	53,1%	71,7%
Netral	0%	0%	0%	
Negatif	68,8%	99,5%	81,3%	
Rata-rata	55,2%	45,3%	44,8%	

b) Support Vector Machine

Proses awal yaitu mencari parameter terbaik yang akan digunakan pada saat melakukan klasifikasi, dengan menggunakan fungsi GridSearchCV dari *library* sklearn ditemukan

parameter terbaik adalah $C = 1$, $\gamma = 1$, $\text{coef} = 0$, dan $\text{kernel} = \text{sigmoid}$. Berikut potongan kode program untuk mencari parameter terbaik dengan fungsi GridSearchCV:

```

1 from sklearn.svm import SVC
2 from sklearn.model_selection import GridSearchCV
3
4 parameters={
5     'kernel': ['rbf', 'poly', 'linear', 'sigmoid'],
6     'C': [0.5, 1, 10, 100],
7     'gamma': [1, 0.1, 0.01, 0.001],
8     'coef0': [0, 1, 10, 0.1]
9 }
10
11 grid_search = GridSearchCV(estimator=SVC(),
12                             param_grid=parameters,
13                             n_jobs=6)
14
15 grid_search.fit(feature_transformed, y)
16
17 print(f'Best Score: {grid_search.best_params_}')

```

Gambar 6. Kode program pencarian parameter terbaik untuk klasifikasi SVM

Gambar di atas merupakan potongan kode program untuk mencari parameter terbaik dengan menggunakan fungsi GridSearchCV dari *library* sklearn yang hasilnya akan digunakan dalam melakukan klasifikasi SVM.

Selanjutnya menggunakan fungsi SVC dari *library* sklearn untuk melakukan klasifikasi terhadap dataset dengan menggunakan parameter yang didapatkan dari hasil pencarian parameter terbaik. Berikut kode program yang digunakan:

```

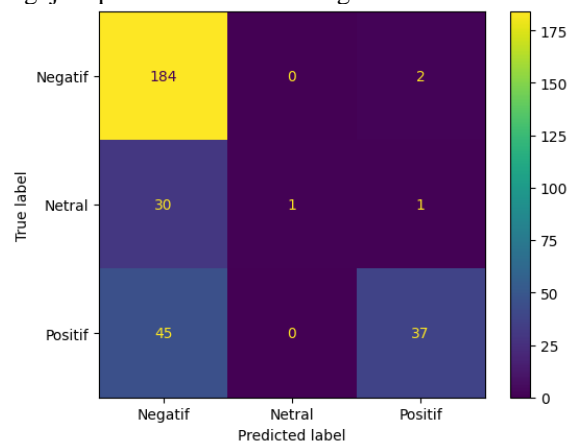
1 from sklearn.svm import SVC
2
3 svm = SVC(kernel="sigmoid", C=1, gamma=1, coef0=0)
4
5 svm.fit(x_train, y_train)
6 hasil_svm = svm.predict(x_test)
7
8 import matplotlib.pyplot as plt
9 cm_display = ConfusionMatrixDisplay(
10     confusion_matrix = confusion_matrix(y_test, hasil_svm),
11     display_labels = ["Negatif", "Netral", "Positif"]
12 )
13
14 cm_display.plot()
15 plt.show()
16
17 print(classification_report(y_test, hasil_svm, digits=3, zero_division=0))

```

Gambar 7. Kode program klasifikasi SVM

Gambar di atas merupakan potongan kode program untuk melakukan klasifikasi SVM dengan menggunakan *library* sklearn serta menampilkan *confusion matrix* dan hasil penilaian dari klasifikasi SVM.

Adapun hasil pengujian pada SVM adalah sebagai berikut:



Gambar 8. Confusion matrix pengujian SVM

Adapun perhitungan nilai akurasi, presisi, *recall*, dan *f1-score* berdasarkan gambar 8 sebagai berikut dengan menggunakan persamaan (1), (2), (3), dan (4):

- Akurasi

$$Akurasi = \frac{37+1+184}{300} = 74\%$$
- Presisi

$$Presisi_{positif} = \frac{37}{40} = 92,5\%$$

$$Presisi_{netral} = \frac{1}{1} = 100\%$$

$$Presisi_{negatif} = \frac{184}{259} = 71\%$$

$$Rata - rata = \frac{92,5\%+100\%+71\%}{3} = 87,8\%$$
- Recall

$$Recall_{positif} = \frac{37}{82} = 45,1\%$$

$$Recall_{netral} = \frac{1}{32} = 3,1\%$$

$$Recall_{negatif} = \frac{184}{186} = 98,9\%$$

$$Rata - rata = \frac{45,1\%+3,1\%+98,9\%}{3} = 49,1\%$$
- F1-score

$$F1 - score_{positif} = \frac{2 \times 92,5\% \times 45,1\%}{92,5\% + 45,1\%} = 60,7\%$$

$$F1 - score_{netral} = \frac{2 \times 100\% \times 3,1\%}{100\% + 3,1\%} = 6,1\%$$

$$F1 - score_{negatif} = \frac{2 \times 71\% \times 98,9\%}{71\% + 98,9\%} = 82,7\%$$

$$Rata - rata = \frac{60,7\%+6,1\%+82,7\%}{3} = 49,1\%$$

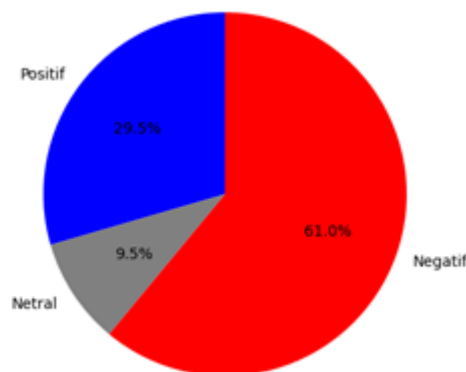
Kesimpulan dari hasil perhitungan nilai akurasi, presisi, *recall*, dan *f1-score* di atas pada hasil klasifikasi SVM, untuk lebih jelasnya dapat dilihat pada table berikut:

Tabel 8. Hasil perhitungan nilai akurasi, presisi, *recall*, dan *f1-score* SVM

Kelas	Presisi	Recall	F1-Score	Akurasi
Positif	92,5%	45,1%	60,7%	74%
Netral	100%	3,1%	6,1%	
Negatif	71%	98,9%	82,7%	
Rata-rata	87,8%	49,1%	49,8%	

B. Pembahasan

Perbandingan sentimen dapat dilihat pada gambar di bawah ini:



Gambar 9. Persentase sentimen

Pada gambar di atas menunjukkan bahwa sentimen netizen twitter terhadap isu Kemenkeu didominasi dengan sentimen negatif dengan jumlah 610 *tweet*, kemudian diikuti sentimen positif dengan jumlah 295 *tweet*, dan terakhir sentimen netral dengan jumlah 95 *tweet*.

Berdasarkan hasil pengujian klasifikasi dengan *confusion matrix* didapatkan hasil performa dari masing-masing metode yang diuraikan pada Tabel 9.

Tabel 9. Hasil pengujian

Nilai	Metode	
	Naïve Bayes	SVM
Akurasi	71,7%	74%
Presisi	55,2%	87,8%
Recall	45,3%	49,1%
F1-Score	44,8%	49,8%

Berdasarkan tabel 9 dapat dilihat bahwa Naïve Bayes memiliki nilai akurasi sebesar 71,7%, presisi sebesar 55,2%, *recall* sebesar 45,3% dan *f1-score* sebesar 44,8%, sedangkan SVM memiliki nilai nilai akurasi sebesar 74%, presisi sebesar 87,8%, *recall* sebesar 49,1% dan *f1-score* sebesar 49,8%, dari hasil tersebut diketahui bahwa nilai akurasi, presisi, *recall*, dan *f1-score* SVM lebih tinggi dibandingkan Naïve Bayes.

IV. Kesimpulan dan saran

Berdasarkan hasil penelitian diperoleh kesimpulan bahwa penelitian ini menunjukkan bahwa sentimen negatif yang paling banyak kemudian diikuti sentimen positif dan yang paling sedikit adalah sentimen netral. Pada penelitian ini diketahui bahwa nilai akurasi, presisi, *recall*, *f1-score* pada SVM lebih tinggi dibandingkan dengan Naïve Bayes. Ini menandakan bahwa performa SVM lebih baik dibandingkan Naïve Bayes dalam melakukan analisis sentimen netizen twitter terhadap isu Kemekeu.

Pada penelitian selanjutnya dapat menggunakan metode klasifikasi yang lain seperti *Decision Tree*, *KNN*, dan yang lainnya. Diharapkan pada penelitian selanjutnya dapat lebih memperbanyak kosa kata pada kamus normalisasi.

Daftar Pustaka

- [1] R. T. Aldisa and P. Maulana, "Analisis Sentimen Opini Masyarakat Terhadap Vaksinasi Booster COVID-19 Dengan Perbandingan Metode Naive Bayes, Decision Tree dan SVM," *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 1, pp. 106–109, Jun. 2022, doi: 10.47065/bits.v4i1.1581.
- [2] D. Duei Putri, G. F. Nama, and W. E. Sulistiono, "Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 10, no. 1, Jan. 2022, doi: 10.23960/jitet.v10i1.2262.
- [3] M. Rangga, A. Nasution, and M. Hayaty, "Perbandingan Akurasi dan Waktu Proses Algoritma K-NN dan SVM dalam Analisis Sentimen Twitter," *Jurnal Informatika*, vol. 6, no. 2, pp. 212–218, 2019, [Online]. Available: <http://ejournal.bsi.ac.id/ejournal/index.php/ji>
- [4] P. Zalfa Salsabila, H. Pratikto, and A. Ririn Aristawati, "Kecemasan pada pengguna media sosial Twitter: Benarkah menyebabkan self-injury?," *INNER: Journal of Psychological Research*, vol. 3, no. 1, pp. 117–127, 2023.
- [5] P. Arsi and R. Waluyo, "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM)," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 8, no. 1, pp. 147–156, 2021, doi: 10.25126/jtiik.202183944.
- [6] H. Tuhuteru and U. Kristen Indonesia Maluku Jl Ot Pattimaipauw, "Analisis Sentimen Masyarakat Terhadap Pembatasan Sosial Berksala Besar Menggunakan Algoritma Support Vector Machine," *Information System Development (ISD)*, vol. 5, no. 2, pp. 7–13, 2020.
- [7] N. N. I. W. A. Astuti, I. M. A. D. Suarjaya, and I. M. S. Raharja, "Analisis Sentimen Masyarakat Terhadap Pelaksanaan Layanan Kesehatan Selama Masa Pandemi Di Indonesia Menggunakan Teknologi Big Data," *JITTER- Jurnal Ilmiah Teknologi dan Komputer*, vol. 3, no. 2, 2022.
- [8] D. Darwis, N. Siskawati, and Z. Abidin, "Penerapan Algoritma Naive Bayes untuk Analisis Sentimen Review Data Twitter BMKG Nasional," *Jurnal Tekno Kompak*, vol. 15, no. 1, pp. 131–145.
- [9] H. Sari, G. Leonarde Ginting, and T. Zebua, "Penerapan Algoritma Text Mining dan TF-IDF Untuk Pengelompokan Topik Skripsi Pada Aplikasi Repository STMIK Budi Darma," *TIN: Terapan Informatika Nusantara*, vol. 2, no. 7, pp. 414–432, 2021, [Online]. Available: <https://ejournal.seminar-id.com/index.php/tin>
- [10] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 5, no. 2, pp. 697–711, 2021.

- [11] A. Rahman Isnain, A. Indra Sakti, D. Alita, and N. Satya Marga, "Sentimen Analisis Publik Terhadap Kebijakan Lockdown Pemerintah Jakarta Menggunakan Algoritma SVM," *JDMSI*, vol. 2, no. 1, pp. 31–37, 2021, [Online]. Available: <https://t.co/NfhmfMjtXw>
- [12] R. R. Adhitya, Wina Witanti, and Rezki Yuniarti, "Perbandingan Metode Cart Dan Naïve Bayes Untuk Klasifikasi Customer Churn," *Infotech Journal*, vol. 9, no. 2, pp. 307–318, Jul. 2023, doi: 10.31949/infotech.v9i2.5641.
- [13] W. Winata, A. Zaidiah, and N. Chamidah, "Analisis Sentimen Pada Ulasan Produk Masker Di Marketplace Shopee Menggunakan Support Vector Machine Dan Seleksi Fitur Chi Square," *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, pp. 491–500, 2022.
- [14] A. Sabrani, I. W. Gede Putu Wirarama Wedashwara, and F. Bimantoro, "Metode Multinomial Naïve Bayes Untuk Klasifikasi Artikel Online Tentang Gempa Di Indonesia (Multinomial Naïve Bayes Method For Classification Of Online Article About Earthquake In Indonesia)," *JTIKA*, vol. 2, no. 1, 2020, [Online]. Available: <http://jtika.if.unram.ac.id/index.php/JTIKA/>