PENERAPAN SVM DALAM ANALISIS SENTIMEN PADA EDLINK MENGGUNAKAN PENGUJIAN *CONFUSION MATRIX*

Gistinia Rininda, Indyah Hartami Santi, Sabitul Kirom
Teknik Informatika, Universitas Islam Balitar
Universitas Islam Balitar; Jl. Majapahit No.2-4, Sananwetan, Kec. Sananwetan,
Kota Blitar, Jawa Timur, (0342) 813145
nindagistinia@gmail.com

ABSTRAK

Edlink adalah aplikasi yang membantu siswa dan dosen melakukan kegiatan pembelajaran secara daring. Namun, suatu aplikasi selalu memiliki kekurangan sehingga menimbulkan komentar dari pengguna aplikasi. Oleh karena itu penelitian ini bertujuan menganalisis sentimen komentar aplikasi Edlink untuk mempertimbangkan pemasalahan yang ada. Analisis sentimen merupakan proses menganalisis text untuk menentukan kecenderungan opini publik. Metode yang digunakan pada penelitian ini adalah *Support Vector Machine* (SVM). Langkah awal yang dilakukan yaitu mengumpulkan data ulasan pengguna aplikasi Edlink menggunakan teknik scrapping dan menghasilkan 86 data. Kemudian dilakukan proses preprocessing data yang meliputi *labelling*, *case folding*, *cleaning*, *tokenezing*, *stopwords removal*, dan *steeming* kemudian dilakukan pembobotan kata TF IDF dan diklasifikasikan menggunakan metode SVM, perhitungan performa menggunakan teknik *confusion matrix* menghasilkan nilai sentimen positif sebesar 24% dan pengujian *precision* sebesar 100% dan *recall* 25%, sedangkan kelas sentimen negatif sebesar 76% dan pengujian *precision* 81% dan *recall* 100% sehingga mendapatkan tingkat keakuratan sebesar 82%.

Kata kunci: *Analisis Sentimen, Support Vector Machine, Edlink, Preprocessing Data, Confusian Matrix*

1. PENDAHULUAN

Dunia pendidikan sekrang sangat dipengaruhi oleh kemajuan teknologi dan komunikas, termasuk munculnya E-Learning. E-Learning adalah pembelajaran yang dilakukan melalui komputer atau perangkat seluler yang terhubung ke internet dan memungkinkan interaksi antara pendidik dan siswa dalam lingkungan pembelajaran online [1]

Pada tahun 2020, Indonesia terkena dampak penyebaran virus Covid-19. Hal ini menimbulkan banyak dampak buruk, misalnya pada sektor sosial, pariwisata, ekonomi, dan pendidikan. Pada 24 Maret 2020, menteri pendidikan mengeluarkan pernyataan bahwa proses belajar mengajar dilakukan secara daring [2]. Secara tidak langsung mendorong masyarakat untuk menggunakan aplikasi pembelajaran elektronik yang tersedia di Google *Play Store*. Banyak platform e-learning yang tersedia di Google *Play Store*, termasuk Sevima Edlink. Aplikasi Sevima dirilis pada tahun 2016 dan memiliki sejumlah fitur berguna untuk memudahkan mahasiswa, termasuk peringatan untuk tenggat waktu yang akan datang dan fitur absensi. Bisa juga untuk presentasi langsung atau pengajuan KRS secara online. Tetapi setiap aplikasi selalu memiliki kelebihan dan kekurangannya sendiri. Di Universitas Islam Balitar memiliki kekurangan pada aplikasi Edlink seperti fitur penjadwalan dan pembagian dosen pada setiap mata kuliah. Data menunujukan bahwa notifikasi sering terlambat muncul, dengan tingkat akurasi 75%. Beragam tanggapan atau komentar pengguna aplikasi, seperti kepuasan atau kekecewaan.

Oleh karena itu untuk mempertimbangkan masalah yang ada akan membantu memahami respons pengguna aplikasi Edlink secara keseluruhan. Ini dapat dicapai dengan melihat pendapat publik tentang ulasan Edlink di Google *Play Store*. Hingga saat ini, aplikasi dengan jumlah *download* tertinggi dan ulasan bintang tertinggi di Play Store selalu dipilih sebagai aplikasi terbaik [3]. Dengan menggunakan analisis sentimen, ulasan positif dan negatif diklasifikasikan dengan metode klasifikasi *Support Vector Machine* (SVM). SVM menggunakan pembelajaran mesin (*supervised learning*) dan memprediksi kelas berdasarkan pola pada hasil proses pelatihan [4]. Dalam penelitian ini, SVM dipilih untuk digunakan di antara semua algoritma klasifikasi karena memiliki keunggulan dibandingkan dengan algoritma lainnya, meskipun data yang akan diolah masih menghasilkan validitas dan akurasi yang tinggi. Berdasarkan penelitian sebelumnya, analisis sentimen dilakukan berdasarkan aspek pelanggan layanan e-currency OVO, menghasilkan akurasi sebesar 98,7%[5] .

Menurut penelitian [6] Analisis Sentimen pada Review Aplikasi Grab di Google Play Store Menggunakan *Support Vector Machine* menemukan nilai akurasi sebesar 85,54% dengan nilai akurasi NBC sebesar 80%. Di sisi lain, penelitian [7] yang memeriksa ulasan aplikasi kredivo dengan membandingkan dua metode SVM, yaitu NBC dan SVM, menemukan bahwa SVM adalah metode yang paling efisien untuk pengklasifikasian dengan dataset ulasan [6].

Penelitian dilakukan untuk mengetahui sentimen pengguna aplikasi Edlink, yang dapat dijadikan tolak ukur penilaian bagi PT Senyra Vidya Utama (SEVIMA) terhadap aplikasi.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Pada penelitian yang dilakukan oleh Rizky Wahyudi (2021) yang bertujuan untuk menganalisis sentimen review pengguna aplikasi Grab pada Google Play Store menggunakan *Support Vector Machine* (SVM) menghasilkan akurasi sebesar 85,54%. Metode SVM dalam penelitian berfungsi agar hasil klasifikasi teks ulasan menjadi lebih baik dan akurat [6].

Penelitian yang dilakukan oleh Rian Tineges (2020). Studi ini menggunakan metode SVM untuk menganalisis oponi pengguna layanan Indihome di Twitter untuk menentukan akurasi yang dihasilkan SVM melalui analisis sentimen. Data yang digunakan untuk studi ini berasal dari pengguna Twitter yang menyebutkan akun Indihome (@Indihome) [8].

Penelitian yang dilakukan oleh Fajar Romadoni (2020) menggunakan analisis sentimen yang digunakan sebagai slusi untuk mengekstrak opini pelanggan terhadap layanan *e-money* OVO melalui teknik text mining menggunakan algortima *Support Vector Machine* (SVM) menjadi positif dan negatif. Data yang diambil berjumlah 3853 data dengan kata kunci @ovo_id. Kernel yang digunakan yaitu *linear*, *polynomial*, rbf, dan *sigmoid*. Hasil akurasi terbaik diperoleh oleh kernel *linier* dengan rasio data 90:10 yang mencapai nilai akurasi sebesar 98.7% [5]

2.2. Edlink

Aplikasi Sevima Edlink merupakan aplikasi berbasis android yang diciptakan untuk menghubungkan antara Dosen dan Mahasiswa dalam dunia pendidikan guna menjada proses pembeljaran tetap teratur dan menghemat waktu [9]. Aplikasi Edlink memiliki fitur seperti membuat forum diskusi, mengirim pesan pribadi, membuat post info untuk memberi tahu anggota forum tentang hal-hal lain, membuat post acara untuk mengatur acara, dan membuat post survei untuk membuat forum. [10].

2.3. Analisis Sentimen

Analisis sentimen atau *opinion mining* bertujuan untuk menganalisis, memahami, mengolah, dan mengekstrak data teks berupa opini tentang entitas seperti produk, layanan, organisasi, orang, dan topik tertentu. Analisis ini digunakan untuk mengekstrak beberapa informasi dari kumpulan data yang ada [11].

2.4. *Preprocessing*

Karena data mentah seringkali memiliki kalimat dan format yang tidak konsisten, maka dari itu *preprocessing* sangat penting dilakukan untuk membersihkan, memodifikasi sumber data, baik berupa karakter non-abjad maupun kata-kata yang tidak perlu, serta memilih fitur dari kumpulan data mentah. [12].

2.5. TF IDF

Term Frequency Inverse Document Frequency (TF IDF) adalah metode pembobotan hubungan suata kata (*term*) dengan suatu dokumen. Bobot kata memberikan nilai pada setiap kata yang telah melalui proses *preprocessing* [13]. Rumus TF dapat didefinisikan dengan persamaan rumus 1.

$$Tf_{t,d} = \frac{Jumlah\ Frekuensi\ Kata\ Terpilih}{Jumlah\ TF} \text{ .....(1)}$$

Dokumen term dapat didistribusikan secara acak menggunakan Inverse *Document Frequency* (IDF). Rumus IDF dapat didefinisikan dengan persamaan rumus 2

$$(idf_t) = \log \frac{D}{DF} \text{ ....(2)}$$

Keterangan :

Idf	: *Inverse Document Frequency*
D	: jumlah dokumen
T	: kata ke t dari kata kunci
DF	: banyaknya dokumen yang memiliki kata t

Berdasarkan persamaan rumus diatas dapat ditentukan nilia pembobotan dengan persamaan rumus 3

$$W_{t,d} = tf_{t,d} * idf_t \text{ ....(3)}$$

2.6. *Support Vector Machine* (SVM)

Support Vector Machine (SVM) membandingkan parameter standar dari set nilai diskrit yang disebut kandidat set, dan kemudian memilih yang memiliki klasifikasi terbaik. Dalam pengajaran yang diawasi, SVM termasuk kedalam *supervised learning* [14].

Data pelatihan (training) dan data tes adalah bagian dari proses klasifikasi SVM. Data pelatihan (training) dilakukan dengan memasukan data pelatihan secara manual ke dalam database, kemudian dilakukan proses preprocessing. Selanjutnya, ekstrasi dilakukan menggunakan TF-IDF yaitu menghitung kemunculan *term* yang kemudian diubah ke dalam format *Support Vector Machine* (SVM), dan setiap kata atau term diberi label positif atau negatif. Kemudian dilakukan kernelisasi menggunakan fungsi RBF. Berikut persamaan rumus fungsi RBF.

$$K\ (x, xi) = \exp(-\frac{||x_1 - x_2||^2}{2\sigma^2}) \text{...(4)}$$

Berikut tahapan *Support Vector Machine* adalah sebagai berikut [15]

a. Melakukan inisialisasi parameter *alfa* (α), *gamma* (γ), *complexity* (C), λ dan *parameter epsilon* (ε)

b. Melakukan perhitungan matriks *Hessian*

$$D_{ij} = y_i y_j\ (K(x_i, x_j) + \lambda^2) \text{ ... (5)}$$

Keterangan :

x_i	= data ke i
x_j	= data ke j
y_i	= kelas data ke i
y_j	= kelas data ke j

c. Menghitung nilai dari $Ei, \delta ai$, dan ai

1. $E_i = \sum_{j=1}^{n} a_i\ D_{ij}$ ... (6)

Keterangan :

E_i	= *error rate*
a_i	= nilai alfa ke j
D_{ij}	= matriks hessian

2. $\delta ai = \min(\max[\gamma(1 - E_i), a_i], C - a_i$ …(7)

Keterangan :
a_i = nilai alfa ke j
γ = nilai *gamma*
E_i = nilai *error rate*
C = nilai kompleksitas

3. $ai = ai + \delta ai$ …(8)

Keterangan :
ai = *alfa* ke i
δai = *delta* alfa ke i

d. Mencari nilai b (bias)

$b = -\frac{1}{2}$ [$\sum_{i=1}^{m} a_i\, y_i K(x_i x^+)$ + [$\sum_{i=1}^{m} a_i\, y_i K(x_i x^-)$] … (9)

Keterangan :
a_i = alfa ke i
y_i = kelas data latih ke i
m = jumlah data
$K(x_i x^+)$, $K(x_i x^+)$ = fungsi kernel yang digunakan

e. Menghitung fungsi keputusan klasifikasi SVM *sign* (f(x))

$f(x) = \sum_{i=1}^{m} a_i y_i K(x_i x) + b$ … (10)

2.7. *Confusion Matrix*

Metode yang dikenal sebagai Confusion Matrix digunakan untuk menghitung kinerja atau tingkat kebenaran akurasi proses klasifikasi. Keakuratan hasil diukur dengan nilai *recall*, *precission*, *accuracy*. Di mana *recall* (*True Positive Rate*) adalah rasio identifikasi benar positif dibandingkan keseluruhan data yang benar positif, *precission* (*Positive Predictive Value*) adalah rasio identifikasi benar positif terhadap semua hasil identifikasi positif, dan akurasi adalah rasio identifikasi benar positif untuk semua data. [16]. Tabel *Confusion Matrix* ditunjukan pada tabel 1.

Tabel 1 Confusion Matrix

		Kelas prediksi	
		Positif	Negatif
Aktual	Positif	TP	FP
	Negatif	FN	FP

TP (*True Positive*) : data positif yang diprediksi benar
FN (*False Negative*) : data negatif yang diprediksi benar
FP (*False Positive*) : data negatif namun diprediksi sebagai data positif
FN (*False Negative*) : data positif namun diprediksi data negatif

2.8. Python

Python adalah bahasa pemrograman yang dapat melakukan *multitasking* dan pemrograman berorientasi objek dengan semantik dinamis. Python memiliki perpustakaan lengkap yang memungkinkan pemrogram membuat aplikasi menggunakan *source code* yang tampak sederhana [17].

3. METODE PENELITIAN

3.1. Teknik Pengumpulan Data

3.1.1. Observasi

Observasi merupakan teknik pengumpulan data dengan cara mengamati objek penelitian yaitu dengan mengamati review pengguna Edlink di website Google *Playstore* untuk mengetahui jumlah dan jenis komentar yang diberikan oleh pengguna aplikasi Edlink yang dilakukan setiap bulannya, yaitu bulan Januari hingga Maret.

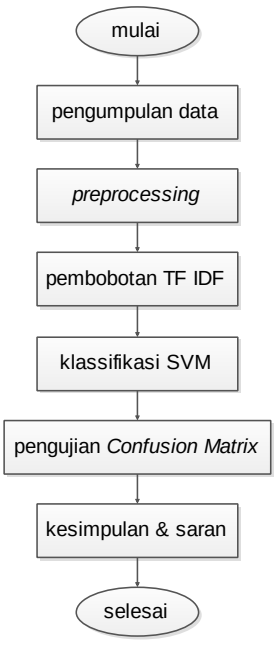
3.1.2. *Scrapping*

Data yang tersedia di website dapat diekstraksi dan disimpan ke dalam database atau file sistem dengan teknik scrapping web [18]. Data dikumpulkan dari ulasan yang diberikan oleh pengguna aplikasi Edlink di Google *Play Store*. Teknik *Scrapping* dilakukan dengan rentan waktu bulan Januari sampai Maret 2023.

3.1.3. Jenis Data

Jenis data yang digunakan adalah data sekunder, yaitu data yang diperoleh dari sumber atau objek penelitian lain. Data yang digunakan berasal dari *review* pengguna aplikasi Edlink yang ditemukan di website Google *Play Store*. Ini dilakukan melalui teknik scraping menggunakan bahasa pemrograman Python.

3.1.4. Tahap Penelitian



Gambar 1. Alur Tahap Penelitian

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengumpulan Data

Hasil dari pengumpulan data yang dilakukan pada bulan Januari sampai bulan Maret 2023 menggunakan teknik *scrapping* berjumlah 86 dimana pada bulan Januari diperoleh 36 ulasan, Februari 27 ulasan, dan Maret 24 ulasan. Gambar 2 merupakan *source code* untuk teknik scrapping dan tabel 2 adalah hasil dari proses *scrapping*.

```
[ ] from google_play_scraper import Sort, reviews

    result, continuation_token = reviews(
        'id.co.sevima.edlink',
        lang='id', # defaults to 'en'
        country='id', # defaults to 'us'
        sort=Sort.NEWEST, # defaults to Sort.MOST_RELEVANT you can use Sort.NEWEST to get newst reviews
        count=86, # defaults to 100
        filter_score_with=None
    )
```

Gambar 2. Source Code Scrapping

Tabel 2. Hasil Proses Scrapping

No.	*userName*	*at*	*content*
1	Luqia Cia	30/03/2023 01:00	Tolong di perbaiki lagi,karna setiap ingin masuk pasti lelet terus menerus,apalagi di waktu penting seperti UTS dak UAS
2	Rosyid Munawar Wicaksono	27/03/2023 11:29	Segera di atasi, edlink error sekarang. Please ada tugas dari dosen
3	Randa Setiawan	22/03/2023 12:46	Bikin ribet
…	…	…	…
…	…	…	…
85	Melisa Audina	09/01/2023 01:39	Sangat membantu
86	KABAR BAIK CHANNEL	07/01/2023 23:37	Luar biasa
87	Kholifa Tussalamah	02/01/2023 12:51	Kok data diweb siak dan diaplikasi beda yah seperti nilai dan tagihan, terus buka aplikasi nya lemot banget padahal jaringan stabil

4.2. Preprocessing

Preprocessing merupakan tahap awal yang dilakukan untuk mempersiapkan teks agar memudahkan dalam mengolah data berdasarkan data set yang digunakan. Alur *preprocessing* ditunjukan pada gambar 3.

Gambar 3. Alur Preprocessing

4.2.1. Labelling

Labelling dilakukan pemberian kategori positif dan negatif pada ulasan.

Tabel 3. Contoh Labelling

Ulasan	Label
Aplikasi ini sangat berguna dan bermanfaat bagi para pelajar khususnya mahasiswa, sebagai tempat pengumpulan tugas dan mendapatkan materi , dan juga lebih simpel dan tidak membuang² waktu dalam pengumpulan. - Universitas Islam Blitar	Positif
Sangat mempersulit mahasiswa	Negatif
Sangat membantu	Positif

4.2.2. Case Folding

Case folding merupakan tahap pemrosessan text dimana menyamaratakan ulasan menjadi huruf kecil (*lowercase*). Gambar 4 merupakan *source code* untuk proses *case folding* dan tabel 4 adalah hasil dari proses *case folding.*

```
[ ]  #Case Folding

     import re
     def clean_lower(content):
       content = content.lower()
       return content
     data ['content'] = data['content'].apply(clean_lower)
     data.to_csv('casefolding.csv', index=False)
     data.head()
```

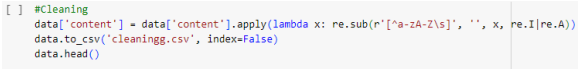
Gambar 4. Source Code Case Folding

Tabel 4. Contoh Case Folding

Ulasan	*Case Folding*
Aplikasi ini sangat berguna dan bermanfaat bagi para pelajar khususnya mahasiswa, sebagai tempat pengumpulan tugas dan mendapatkan materi , dan juga lebih simpel dan tidak membuang² waktu dalam pengumpulan. - Universitas Islam Blitar	aplikasi ini sangat berguna dan bermanfaat bagi para pelajar khususnya mahasiswa, sebagai tempat pengumpulan tugas dan mendapatkan materi , dan juga lebih simpel dan tidak membuang² waktu dalam pengumpulan. - universitas islam blitar
Sangat mempersulit mahasiswa	sangat mempersulit mahasiswa
Sangat membantu	sangat membantu

4.2.3. Cleaning

Cleaning merupakan proses membersihkan ulasan dari karakter atau tanda baca dan symbol. Gambar 5 merupakan *source code* untuk proses *cleaning* dan tabel 5 adalah hasil dari proses *cleaning.*



```
[ ] #Cleaning
    data['content'] = data['content'].apply(lambda x: re.sub(r'[^a-zA-Z\s]', '', x, re.I|re.A))
    data.to_csv('cleaningg.csv', index=False)
    data.head()
```

Gambar 5. Source Code Cleaning

Tabel 5. Contoh Cleaning

Case Folding	*Cleaning*
aplikasi ini sangat berguna dan bermanfaat bagi para pelajar khususnya mahasiswa, sebagai tempat pengumpulan tugas dan mendapatkan materi ,	aplikasi ini sangat berguna dan bermanfaat bagi para pelajar khususnya mahasiswa sebagai tempat pengumpulan tugas dan mendapatkan materi dan

Case Folding	*Cleaning*
dan juga lebih simpel dan tidak membuang² waktu dalam pengumpulan. - universitas islam blitar	juga lebih simpel dan tidak membuangÂ² waktu dalam pengumpulan universitas islam blitar
lumayan	lumayan
sangat membantu	sangat membantu

4.2.4. Tokenezing

Tokenzing dilakukan untuk memisahkan kata dalam suatu kalimat pada ulasan. Gambar 6 merupakan *source code* untuk proses *tokenezing* dan tabel 6 adalah hasil dari proses *tokenezing.*

```
[ ]  #Tokenezing
     def token(comments) :
         nstr = comments.split(' ')
         dat = []
         a = -1
         for hu in nstr:
             a = a+1
         if hu == ' ' :
             dat.append(a)
         p = 0
         b = 0
         for q in dat:
             b = q - p
             del nstr[b]
             p = p+1
         return nstr
     data['content'] = data['content'].apply(token)
     data.head()
     data.to_csv('tokonezing.csv', index=False)
```

Gambar 6. Source Code Tokenezing

Tabel 6. Contoh Tokenezing

Cleaning	*Tokenezing*
aplikasi ini sangat berguna dan bermanfaat bagi para pelajar khususnya mahasiswa sebagai tempat pengumpulan tugas dan mendapatkan materi dan juga lebih simpel dan tidak membuangÂ² waktu dalam pengumpulan universitas islam blitar	['aplikasi', 'ini', 'sangat', 'berguna', 'dan', 'bermanfaat', 'bagi', 'para', 'pelajar', 'khususnya', 'mahasiswa', 'sebagai', 'tempat', 'pengumpulan', 'tugas', 'dan', 'mendapatkan', 'materi', '', 'dan', 'juga', 'lebih', 'simpel', 'dan', 'tidak', 'membuangÂ²', 'waktu', 'dalam', 'pengumpulan', '', 'universitas', 'islam', 'blitar']
lumayan	['lumayan']
sangat membantu	['sangat', 'membantu']

4.2.5. Stopword Removal

Stopword removal merupakan roses memilih kata-kata dari ulasan untuk mengambil kata-kata yang penting atau menggambarkan komentar. Gambar 7 merupakan *source code* untuk proses *stopword removal* dan tabel 7 adalah hasil dari proses *stopword removal.*

```
[ ]  #Stopword Removal

     import nltk
     nltk.download('stopwords')
     from nltk.corpus import stopwords

     def stopword_removal(comments):
       filtering = stopwords.words('indonesian')
       x = []
       data = []
       def myFunc (x):
         if x in filtering:
           return False
         else :
           return True
       fit = filter(myFunc, comments)
       for x in fit:
           data.append(x)
       return data
     data['content'] = data['content'].apply(stopword_removal)
     data.head()
     data.to_csv('stopword.csv', index=False)
```

Gambar 7. Source Code Stopword Removal

Tabel 7. Contoh Stopword Removal

Tokenezing	*Stopwords Removal*
['aplikasi', 'ini', 'sangat', 'berguna', 'dan', 'bermanfaat', 'bagi', 'para', 'pelajar', 'khususnya', 'mahasiswa', 'sebagai', 'tempat', 'pengumpulan', 'tugas', 'dan', 'mendapatkan', 'materi', '', 'dan', 'juga', 'lebih', 'simpel', 'dan', 'tidak', 'membuangÂ²', 'waktu', 'dalam', 'pengumpulan', '', 'universitas', 'islam', 'blitar']	['aplikasi', 'berguna', 'bermanfaat', 'pelajar', 'mahasiswa', 'pengumpulan', 'tugas', 'materi', '', 'simpel', 'membuangÂ²', 'pengumpulan', '', 'universitas', 'islam', 'blitar']
['lumayan']	['lumayan']
['sangat', 'membantu']	['membantu']

4.2.6. Stemming

Stemming merupakan proses menghilangkan infleksi kata atau mengubah kata kebentuk dasarnya. Gambar 8 merupakan *source code* untuk proses *stemming* dan tabel 8 adalah hasil dari proses *stemming.*

```
[ ]  #Stemming

     from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

     def stemming (comments):
       factory = StemmerFactory()
       stemmer = factory.create_stemmer()
       do = []
       for w in comments:
         dt = stemmer.stem(w)
         do.append(dt)
       d_clean=[]
       d_clean=" ".join(do)
       print(d_clean)
       return d_clean
     data['content'] = data['content'].apply(stemming)
     data.to_csv('data_bakuok.csv', index=False)
```

Gambar 8. Source Code Stemming

Tabel 8. Contoh Stemming

Stopwords Removal	*Stemming*
['aplikasi', 'berguna', 'bermanfaat', 'pelajar', 'mahasiswa', 'pengumpulan', 'tugas', 'materi', '', 'simpel', 'membuangÂ²',	[‘aplikasi’, ‘guna’, ‘manfaat’, ‘ajar’, ‘mahasiswa’, ‘kumpul’, ‘tugas’, ‘materi’, ‘simpel’, ‘buang’,

Stopwords Removal	***Stemming***
'pengumpulan', '', 'universitas', 'islam', 'blitar']	‘kumpul’, ‘universitas’, ‘islam’, ‘blitar’]
['lumayan']	[‘lumayan’]
['membantu']	[‘bantu’]

4.3. Pembobotan TF IDF

Pembobotan TF IDF dilakukan setelah melalui tahap *preprocessing*. Pembobotan TF IDF dilakukan untuk mengubah teks menjadi angka yang diperoleh dari kemunculan term dalam suatu ulasan dan jumlah kemunculan dokumen. Untuk melakukan perhitungan TF IDF digunakan bahasa pemrograman python menggunakan *library sklearn* seperti pada gambar 9.

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer= TfidfVectorizer()
X = vectorizer.fit_transform(df['content'])
vector=pd.DataFrame(X.todense().T,index=vectorizer.get_feature_names_out(),
                    columns=[f'D{i+1}' for i in range(len(df['content']))])
vector.tail()
```

Gambar 9. Source Code Pembobotan TF IDF

Tabel 9. Hasil Pemboobotan TF IDF

index	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	…	…	D82	D83
absen	0	0	0	0	0	0	0	0	0	0	…	…	0	0
absensi	0	0	0	0	0	0	0	0	0	0	…	…	0	0
adlink	0	0	0	0	0	0	0,36	0	0	0	…	…	0	0
ado	0	0	0	0	0	0	0	0	0	0	…	…	0	0
ahlak	0	0	0	0	0	0	0	0	0	0	…	…	0	0
aja	0	0	0	0	0	0	0	0	0	0	…	…	0	0
ajar	0	0	0	0	0	0	0	0	0	0	…	…	0	0
akibat	0	0	0	0	0	0	0	0	0	0	…	…	0	0
akses	0	0	0	0	0	0	0	0	0	0	…	…	0	0
akun	0	0	0	0	0	0	0	0	0	0	…	…	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
uts	0,37	0	0	0	0	0	0	0	0	0	...	...	0	0
va	0	0	0	0	0	0	0	0	0	0	...	...	0	0
variasi	0	0	0	0	0	0	0	0	0	0	...	...	0	0
verifikasi	0	0	0	0	0	0	0	0	0	0	...	...	0	0
warna	0	0	0	0	0	0	0	0	0	0	...	...	0	0
web	0	0	0	0	0	0	0	0	0	0	...	...	0	0
wifi	0	0	0	0	0	0	0	0	0	0	...	...	0	0
ya	0	0	0	0	0	0	0	0	0	0,22	...	...	0	0
yah	0	0	0	0	0	0	0	0	0	0	...	...	0	0,29
yg	0	0	0	0	0	0	0	0	0	0	...	...	0	0

Tabel 9 merupakan hasil pembobotan TF IDF pada *tools* python yang diambil 10 data teratas dan 10 data terakhir. Dari 86 data ulasan yang digunakan, setelah melalui proses pembobotan TF IDF terjadi perubahan jumlah data yaitu menjadi 336 data, karena TF IDF merupakan pembobotan setiap kata atau term.

4.4. Klasifikasi *Support Vector Machine* (SVM)

Klasifikasi SVM dilakukan setelah preprocessing dan pembobotan TF IDF selesai. Klasifikasi dilakukan dengan membagi data set menjadi data training dan data testing dengan rasio 80:20. Pembagian data *training* akan memiliki 80% dari total data set yang dapat memberikan lebih banyak sampel kepada model untuk belajar pola yang ada dalam data sedangkan data *testing* memiliki 20% dari total data set untuk menguji performa model pada data. Gambar 10 merupakan *soruce code* pada python yang digunakan untuk membagi data set menggunakan pustaka *sckit-learn* (*sklearn*) pada python.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20,random_state=109)
print("Train: ",X_train.shape,y_train.shape,"Test: ",(X_test.shape,y_test.shape))
```

Gambar 10. Source Code Pembagian Data

```
Train:  (66, 336) (66,) Test:  ((17, 336), (17,))
```

Gambar 11. Hasil Pembagian Dataset

Pada gambar 11 menghasilkan pembagian data *training* sebanyak 66 data dan 17 data *testing* dari keseluruhan 336 data.

Setelah pembagian data set langkah berikutnya adalah membuat model menggunakan *Support Vector Machine* (SVM). Pembuatan permodelan menggunakan *library sckit-lear machine learning* pada python. Berikut adalah *source code* yang digunakan untuk membuat model *Support Vector Machine* (SVM).

```
[ ]  from sklearn.svm import SVC
     clf = SVC(C=1.0, kernel='rbf')
     clf.fit(X_train, y_train)
```

Gambar 12. Pembuatan Model SVM

4.5. *Confusion Matrix*

Data yang digunakan untuk pengujian *confusion matrix* adalah data *testing* pada gambar 11 yang berjumlah 17 data dan menghasilkan matrix 2x2.

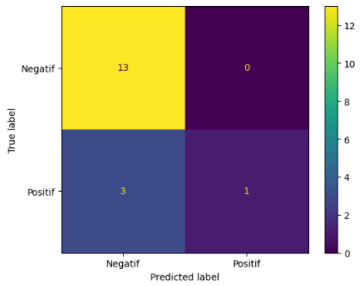
Gambar 13 merupakan *source code* untuk pengujian *confusion matrix* dan gambar 14 adalah hasil dari *confusion mtarix*

```
[ ]  #melakukan prediksi terhadap data X-Test
     y_pred = clf.predict(X_test)

     #melakukan evaluasi menggunakan cunfosion matrix
     from sklearn.metrics import classification_report, confusion_matrix
     print("Confusion Matrix SVM")
     print(confusion_matrix(y_test,y_pred))
     print(classification_report(y_test,y_pred))
     from sklearn.metrics import accuracy_score
     acu_dt=accuracy_score(y_test, y_pred)

     #hasil akurasi menggunakan SVM
     print('AKURASI SVM: %.3f' % acu_dt)
```

Gambar 13. Source Code Confusion Matrix



Gambar 14. Confusion Matrix SVM

Berdasarkan gambar 14 dapat diuraikan bahwa pengujian *confusion matrix* mendapatkan nilai TP (*True Positif*) 1, FP (*False Positif*) 0, FN (*False Negative*) 3, dan TN (*True Negative*) 13. Pada penelitian ini untuk mengukur *performance metrics* yang digunakan adalah *accuracy*, *precission*, dan *recall*. Gambar 15 merupakan hasil dari perhitungan *performance metrics*.

	precision	recall	f1-score	support
Negatif	0.81	1.00	0.90	13
Positif	1.00	0.25	0.40	4
accuracy			0.82	17
macro avg	0.91	0.62	0.65	17
weighted avg	0.86	0.82	0.78	17

AKURASI SVM: 0.824

Gambar 15. Hasil Performance Confusion Matrix

Berdasarkan gambar 15 hasil pengujian data ulasan menggunakan *confusion matrix* pada sentimen negatif menghasilkan nilai presentase *precission* 81%, dan *recall* 100%. Sedangkan sentimen positif menghasilkan nilai presentase *precission* 100% dan *recall* 25% dengan tingkat keakuratan pengujian klasifikasi Algoritma *Support Vector Machine* (SVM) sebesar 82,4%.

5. KESIMPULAN DAN SARAN

Hasil analisis sentimen menggunakan algoritma *Support Vector Machine* (SVM) meliputi proses pengambilan data menggunakan teknik *scrapping*, preprocessing, pembobotan TF IDF, klasifikasi SVM, dan pengujian *confusion matrix*. Berdsarkan hasil pengujian menggunakan *confusion matrix* menghasilkan sentimen negatif sebesar 76% dengan keakuratan hasil pegujian akurasi sebesar 82% dan pengujian *precission* sebesar 81%, serta *recall* sebesar 100%. Sedangkan sentimen positif menghasilkan nilai sentimen sebesar 24% dan nilai evaluasi *precission* sebanyak 100%, serta *recall* sebanyak 25%. Sehingga dapat disimpulkan bahwa opini dari masyarakat terhadap kekurangan aplikasi Edlink masih cukup tinggi. Untuk meningkatkan hasil penelitian agar lebih maksimal perlu menambahkan pendekatan dengan pendektesian emoticon dengan metode *Lexicon Based Approach* agar menghasilkan data yang lebih akurat. Penelitian selanjutnya disaranakn untuk menggunakan dua atau lebih metode algoritma lainnya untuk membandingkan hasil akurasi yang akurat.

DAFTAR PUSTAKA

[1] S. Silahuddin, "Penerapan E-Learning dalam Inovasi Pendidikan," *CIRCUIT J. Ilm. Pendidik. Tek. Elektro*, vol. 1, no. 1, hal. 48–59, 2015, doi: 10.22373/crc.v1i1.310.

[2] A. H. Mulyadi dan S. Lestari, "Terbit online pada laman web jurnal: https://ejurnalunsam.id/index.php/jicom/ Analisis Sentimen Terhadap Sekolah Saat Covid-19 Pada Twitter Menggunakan Metode Lexicon Based," vol. 03, no. 01, hal. 17–23, 2022, [Daring]. Tersedia pada: https://ejurnalunsam.id/index.php/jicom/

[3] N. Yolanda, I. H. Santi, D. Fanny, dan Permadi, "Analisis Sentimen Popularitas Aplikasi Modle Dan Edmodo Menggunakan Algoritma Support Vector Machine," vol. 3, no. 1, hal. 48–59, 2022.

[4] A. Erfina, E. S. Basryah, A. Saepulrohman, dan D. Lestari, "Analisis Sentimen Aplikasi Pembelajaran Online Di Play Store Pada Masa Pandemi Covid-19 Menggunakan Algoritma Support Vector Machine (Svm)," *Semin. Nas. Inform.*, vol. 1, no. 1, hal. 145–152, 2020, [Daring]. Tersedia pada: http://jurnal.upnyk.ac.id/index.php/semnasif/article/view/4094

[5] F. Romadoni, Y. Umaidah, dan B. N. Sari, "Text Mining Untuk Analisis Sentimen Pelanggan Terhadap Layanan Uang Elektronik Menggunakan Algoritma Support Vector Machine," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 2, hal. 247–253, 2020, doi: 10.32736/sisfokom.v9i2.903.

[6] R. Wahyudi dan G. Kusumawardana, "Analisis Sentimen pada Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *J. Inform.*, vol. 8, no. 2, hal. 200–207, 2021, doi: 10.31294/ji.v8i2.9681.

[7] A. Muhammadin dan I. A. Sobari, "Analisis Sentimen Pada Ulasan Aplikasi Kredivo Dengan Algoritma Svm Dan Nbc," *Reputasi J. Rekayasa Perangkat Lunak*, vol. 2, no. 2, hal. 85–91, 2021, doi: 10.31294/reputasi.v2i2.785.

[8] R. Tineges, A. Triayudi, dan I. D. Sholihati, "Analisis Sentimen Terhadap Layanan Indihome

Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM)," *J. Media Inform. Budidarma*, vol. 4, no. 3, hal. 650, 2020, doi: 10.30865/mib.v4i3.2181.

[9] R. A. Purba *dkk.*, *Model dan Aplikasi Pembelajaran: Inovasi Pembelajaran Di Situasi Tidak Normal*. Yayasan Kita Menulis, 2020. [Daring]. Tersedia pada: https://www.google.co.id/books/edition/Model_dan_Aplikasi_Pembelajaran_Inovasi/c8R6EAAAQBAJ?hl=id&gbpv=0

[10] M. Tubagus, *Model Pembelajaran Terbuka Jarak Jauh*. Nas Media Pustaka, 2021. [Daring]. Tersedia pada: https://www.google.co.id/books/edition/Model_Pembelajaran_Terbuka_Jarak_Jauh/LXEyEAAAQBAJ?hl=id&gbpv=0

[11] H. K. Mangat, "Analisis sentimen pada tweet dengan tagar #bpjsrasarentenir menggunakan metode support vectore machine (svm) skripsi," 2021.

[12] I. Werdiningsih, D. C. R. Novitasari, dan D. Z. Haq, *Pengelolaan Data Mining dengan Pemrograman Matlab*. Airlangga University Press, 2022. [Daring]. Tersedia pada: https://www.google.co.id/books/edition/Pengelolaan_Data_Mining_dengan_Pemrogram/CgOdEAAAQBAJ?hl=id&gbpv=0

[13] I. Sari, *TEXT MINING: Praktek Klasifikasi dan Pemodelan Topik dengan Phyton*. uwais inspirasi indonesia, 2023. [Daring]. Tersedia pada: https://www.google.co.id/books/edition/TEXT_MINING_Praktek_Klasifikasi_dan_Pemo/wA6mEAAAQBAJ?hl=id&gbpv=1

[14] S. R. Hakim, M. A. Rizki, N. I. Zekha F, N. Fitri, Y. R. A, dan R. Nooraeni, "Analisis Sentimen Pengguna Instagram Terhadap Kebijakan Kemdikbud Mengenai Bantuan Kuota Internet Dengan Metode Support Vector Machine (Svm)," *J. MSA (Mat. dan Stat. serta Apl.)*, vol. 8, no. 2, hal. 15, 2020, doi: 10.24252/msa.v8i2.16795.

[15] R. A. Wijayanti, M. T. Furqon, dan S. Adinugroho, "Penerapan Algoritme Support Vector Machine Terhadap Klasifikasi Tingkat Risiko Pasien Gagal Ginjal," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 10, hal. 3500–3507, 2018, [Daring]. Tersedia pada: http://j-ptiik.ub.ac.id

[16] Y. Julianto, D. H. Setiabudi, dan S. Rostianingsih, "Analisis Sentimen Ulasan Restoran Menggunakan Metode Support Vector Machine," *J. Infra*, vol. 10, no. 1, 2022.

[17] L. Perkovic, *Introduction to Computing Using Python An Application Development Focus*. Wiley, 2015. [Daring]. Tersedia pada: https://www.google.co.id/books/edition/Introduction_to_Computing_Using_Python/rE9FEAAAQBAJ?hl=id&gbpv=0

[18] K. D. Setiawan, N. Made, I. Marini, dan D. P. Githa, "Analisis Respon Pelanggan Terhadap Suatu Produk Dengan Metode Support Vector Machine (SVM)," vol. 3, no. 2, 2022.