

ALGORITMA *SUPPORT VECTOR MACHINE* (SVM) UNTUK PENERAPAN ANALISIS SENTIMEN TERHADAP APLIKASI SAKTI.LINK

Muhammad Rizki Fakhri^{1*}, Haris Munandar², Imelda³, Siswanto⁴, Joko Christian Chandra⁵

^{1,2,3,4,5} Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, DKI Jakarta, Indonesia

Email: ^{1*}1811501897@student.budiluhur.ac.id, ²haris.munandar@budiluhur.ac.id, ³imelda@budiluhur.ac.id,
⁴siswanto@budiluhur.ac.id, ⁵joko.christian@budiluhur.ac.id

Abstrak- SAKTI.Link merupakan aplikasi penunjang sistem perkoperasian yang mudah digunakan oleh seluruh anggota koperasi yang terhubung di SiCUNDO-SAKTI Core Banking. SAKTI.Link memberikan kemudahan untuk para anggota koperasi melakukan aktivitas perkoperasian cukup dengan satu aplikasi dan semudah menggerakkan jari. Meskipun demikian banyak masyarakat yang memberikan opini baik itu pro atau kontra terhadap aplikasi SAKTI.Link. Banyaknya opini pengguna terhadap aplikasi SAKTI.Link menyebabkan pengembang aplikasi merasa kesulitan untuk memisahkan opini yang akan digunakan untuk perbaikan aplikasi. Penelitian ini bertujuan untuk membantu pengembang aplikasi SAKTI.Link dalam mengetahui respon dari pengguna, pada penelitian ini akan dilakukan analisis sentimen terhadap aplikasi SAKTI.Link. Beberapa pengguna memberikan ulasan dalam bentuk kuesioner yang dibagikan kepada pengguna aplikasi SAKTI.Link. Komentar kemudian akan dikelompokkan menjadi tiga kategori, yaitu komentar positif, negatif, dan netral. Sistem dibuat dengan bahasa pemrograman python dalam bentuk website yang dapat melakukan analisis terhadap aplikasi SAKTI.Link menggunakan metode Support Vector Machine (SVM). Selanjutnya, sistem dilakukan pengujian metode menggunakan confusion matrix dan mendapatkan nilai accuracy sebesar 94,9%, precision sebesar 90,07%, recall sebesar 94,9% dan F1-score sebesar 92,42%.

Kata Kunci: Analisis Sentimen, Aplikasi SAKTI.Link, Kuesioner, Support Vector Machine (SVM)

SUPPORT VECTOR MACHINE (SVM) ALGORITHM FOR SENTIMENT ANALYSIS APPLICATION ON SAKTI.LINK APPLICATION

Abstract- *SAKTI.Link is a cooperative system supporting application that is easy to use by all cooperative members connected to SiCUNDO-SAKTI Core Banking. SAKTI.Link makes it easy for cooperative members to carry out cooperative activities with just one application and it's as easy as moving a finger. Even so, many people give opinions, both the pros and cons of the SAKTI.Link application. The large number of user opinions on the SAKTI.Link application causes application developers to find it difficult to separate opinions that will be used to improve applications. This research aims to assist the SAKTI.Link application developer in knowing the response from users, in this study a sentiment analysis will be carried out towards the SAKTI.Link application. Some users provide reviews in the form of questionnaires which are distributed to users of the SAKTI.Link application. Comments will then be grouped into three categories, namely positive, negative, and neutral comments. The system is made with the Python programming language in the form of a website that can analyze applications. SAKTI.Link uses the Support Vector Machine (SVM) method. Furthermore, the system tested the method using the confusion matrix and obtained an accuracy value of 94.9%, precision of 90.07%, recall of 94.9% and F1-score of 92.42%.*

Keywords: *Sentiment Analysis, SAKTI.Link Application, Questionnaire, Support Vector Machine (SVM)*

1. PENDAHULUAN

Seiring dengan perkembangan teknologi, pengelolaan koperasi juga semakin berkembang, pengelolaan dengan memanfaatkan teknologi memberikan manfaat dapat dengan mudah diakses kapanpun dan dimanapun [1]. Terdapat beberapa aplikasi yang dapat digunakan dalam melakukan pengelolaan koperasi salah satunya yaitu SAKTI.Link. SAKTI.Link merupakan aplikasi penunjang sistem perkoperasian yang mudah digunakan oleh seluruh anggota koperasi yang terhubung di SiCUNDO-SAKTI Core Banking [2]. SAKTI.Link memberikan kemudahan untuk para anggota koperasi melakukan aktivitas perkoperasian cukup dengan satu aplikasi dan semudah menggerakkan jari.

Meskipun demikian banyak masyarakat yang memberikan opini baik itu pro atau kontra terhadap aplikasi SAKTI.Link. Salah satu diantara opini tersebut. Aplikasi ini telah di download 10.000 pengguna PlayStore, selain Playstore aplikasi ini juga tersedia di App Store. Dengan banyaknya opini terhadap aplikasi SAKTI.Link yang telah banyak digunakan pengguna menyebabkan pengembang aplikasi merasa kesulitan untuk memisahkan opini yang akan digunakan untuk perbaikan aplikasi.

Opini tentang aplikasi SAKTI.link dapat digunakan sebagai sarana untuk memberikan penilaian, serta menjadi tolak ukur apakah aplikasi tersebut recommended atau tidak bagi para pengguna aplikasi SAKTI.Link. Namun hingga saat ini pihak SAKTI.Link belum melakukan tindakan apapun terkait dengan opini yang diberikan pengguna. Beberapa pengguna memberikan ulasan jika aplikasi SAKTI.Link masih jauh dari harapan pengguna, namun terdapat juga pengguna yang merasa bahwa aplikasi SAKTI.Link sudah cukup baik. Saat ini, pengguna dari aplikasi SAKTI.Link merasa kalau aplikasi masih terdapat kekurangan, hal ini dapat dilihat dari komentar pengguna tentang aplikasi. Pengguna aplikasi SAKTI.Link mengharapkan agar aplikasi kedepannya dapat lebih mudah untuk digunakan oleh pengguna serta perbaikan-perbaikan dari segi fitur aplikasi.

Dan untuk pengembang aplikasi, pengguna mengharapkan dapat mengembangkan lagi fitur yang ada. Permasalahannya adalah pengembang aplikasi kesulitan untuk memisahkan opini yang akan digunakan untuk perbaikan aplikasi. Pada penelitian ini peneliti bermaksud untuk melakukan analisis sentimen terhadap aplikasi SAKTI.Link dengan mengelompokkan respon positif, respon negatif dan netral dari pengguna. Analisis tersebut yang dapat digunakan oleh pengembang mengetahui opini dari pengguna dan memudahkan pengembang untuk segera dilakukan update pada aplikasi SAKTI.Link. Data opini atau komentar tersebut didapat dari hasil kuesioner yang dibagikan kepada pengguna aplikasi SAKTI.Link. Komentar kemudian akan dikelompokkan menjadi tiga kategori, yaitu komentar positif, negative, dan netral [3]. Berdasarkan permasalahan yang telah dijelaskan terkait opini atau komentar pengguna yang dapat bermanfaat apabila diolah kembali maka peneliti akan melakukan analisis sentimen terhadap aplikasi SAKTI.link.

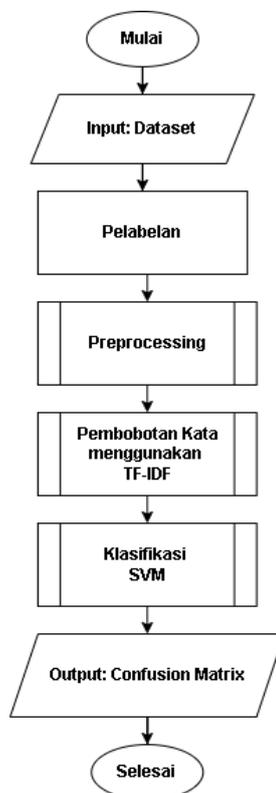
2. METODE PENELITIAN

2.1 Data Penelitian

Untuk menyelesaikan permasalahan diperlukan data yang akan diproses terlebih dahulu. Data yang akan digunakan pada penelitian ini diambil dari data opini atau komentar yang didapat dari hasil kuesioner yang dibagikan kepada 391 responden pengguna aplikasi SAKTI.Link dari bulan November 2022.

2.2 Penerapan Metode yang Digunakan

Penerapan metode yang diimplementasikan berawal dari data latih dan data uji yang dibagi menjadi 60% data latih dan 40 % data uji, dengan jumlah data latih 234, dan data uji 157. Alur proses dari deskripsi umum sistem akan ditunjukkan pada Gambar 1.



Gambar 1. Flowchart Sistem

2.2.1 Input Dataset

Langkah atau tahapan pertama dari proses analisis sentimen yaitu Input dataset. *Input dataset* adalah memasukkan dataset yang diambil dari hasil kuesioner berupa data opini atau komentar yang dibagikan kepada pengguna aplikasi SAKTI.Link. Perhitungan manual algoritma SVM pada penelitian ini menggunakan *dataset* hasil kuesioner yang dibagikan kepada 10 pengguna aplikasi SAKTI.Link.

2.2.2 Pelabelan

Langkah atau tahapan kedua dari proses analisis sentimen, yaitu pelabelan. Setelah berhasil mendapatkan dataset, maka tahapan selanjutnya yang dilakukan adalah proses pelabelan untuk membedakan komen yang mengandung nilai positif, netral dan negatif. Komen positif dilambangkan dengan angka 1, komen netral dilambangkan dengan angka 0 dan komen negatif dilambangkan dengan angka -1.

2.2.3 Preprocessing

Preprocessing merupakan tahapan untuk memproses data latih dan data uji [4]. Terdapat beberapa langkah dalam melakukan *preprocessing*, yaitu:

a. Input Dataset

Langkah atau tahapan pertama dari *preprocessing*, yaitu input dataset dengan memasukkan *dataset* yang diambil dari hasil kuesioner berupa data opini atau komentar yang dibagikan kepada 10 pengguna aplikasi SAKTI.Link. *Dataset* yang digunakan pada tahapan *preprocessing* telah melalui tahapan pelabelan data.

b. Case Folding

Langkah atau tahapan kedua dari *preprocessing* yaitu *case folding* yang merupakan proses membersihkan dokumen dan menyeleksi kata yang tidak diperlukan seperti html, *emoticon*, *hashtag*, *mention* dan URL seperti pada Tabel 1.

Tabel 1. Hasil *Case Folding*

<i>Comment</i>	<i>Case Folding</i>
aplikasinya sudah sangat bagus karena tidak menggunakan bilang koma (,) padahal kebanyakan usaha koperasi memiliki bunga dengan bilangan koma	aplikasinya sudah sangat bagus karena tidak menggunakan bilang koma padahal kebanyakan usaha koperasi memiliki bunga dengan bilangan koma

c. Tokenizing

Langkah atau tahapan ketiga dari *preprocessing* yaitu *tokenizing* yang merupakan proses seleksi pemotongan kata dalam kalimat. Diberikan pemisah seperti tanda koma (,), titik (.), dan tanda pemisah lainnya [5] seperti pada Tabel 2.

Tabel 2. Hasil *Tokenizing*

<i>Comment</i>	<i>Tokenizing</i>
aplikasinya sudah sangat bagus karena tidak menggunakan bilang koma (,) padahal kebanyakan usaha koperasi memiliki bunga dengan bilangan koma	aplikasinya sudah sangat bagus karena tidak menggunakan bilang koma padahal kebanyakan usaha koperasi memiliki bunga dengan bilangan koma

d. Normalisasi

Langkah atau tahapan keempat dari *preprocessing* yaitu normalisasi yang merupakan tahap untuk mengubah kata-kata yang terdapat pada dataset, yang sebelumnya kata tersebut tidak baku menjadi kata dasar yang baku atau sesuai ejaan kata yang benar [6] seperti pada Tabel 3.

Tabel 3. Hasil Normalisasi

<i>Comment</i>	Normalisasi
aplikasinya sudah sangat bagus karena tidak menggunakan bilang koma (,) padahal kebanyakan usaha koperasi memiliki bunga dengan bilangan koma	aplikasinya sudah sangat bagus karena tidak menggunakan bilang koma padahal kebanyakan usaha koperasi memiliki bunga dengan bilangan koma

e. *Filtering/Stopwords.*

Langkah atau tahapan kelima dari *preprocessing*, yaitu *filtering/stopwords* yang merupakan tahap untuk menghilangkan kata yang kurang efektif [7] seperti pada Tabel 4.

Tabel 4. Hasil *Filtering/Stopwords*

<i>Comment</i>	<i>Filtering/Stopwords</i>
aplikasinya sudah sangat bagus karena tidak menggunakan bilang koma (,) padahal kebanyakan memiliki bunga dengan bilangan koma	aplikasi sudah sangat bagus guna bilang koma banyak usaha koperasi

f. *Stemming*

Langkah atau tahapan keenam dari *preprocessing* yaitu *stemming* yang merupakan tahap untuk menyaring kata yang terdapat kata sambung, kata ganti dan kata depan, sehingga menjadi kata dasar dengan menghilangkan awalan atau akhiran seperti pada Tabel 5.

Tabel 5. Hasil *Stemming*

<i>Comment</i>	<i>Stemming</i>
aplikasinya sudah sangat bagus karena tidak menggunakan bilang koma (,) padahal kebanyakan memiliki bunga dengan bilangan koma	aplikasi sudah sangat bagus guna bilang koma banyak usaha koperasi

2.2.4 Pembobotan Kata (TF-IDF)

Hasil dari proses *preprocessing* adalah data yang telah siap untuk diolah. Data tersebut akan dilakukan proses pembobotan. Alur pembobotan kata (TF-IDF) seperti berikut ini:

a. *Input Preprocessing*

Langkah atau tahapan pertama dari pembobotan kata TF-IDF yaitu *input preprocessing* dengan memasukkan dataset yang diambil dari hasil tahapan *preprocessing* seperti pada Tabel 6.

Tabel 6. *Input Preprocessing*

No	<i>Stemming</i>	<i>Class</i>
1	aplikasi sangat bagus guna bilang koma banyak usaha koperasi	1
2	aplikasi sangat bantu offline kembang simpan akun google data aman suatu saat ganti hp hangus	-1
3	aplikasi bagus hp reset data hilang pulih online backup akses anggota detail tabung orang	-1
4	bagus aplikasi bunga buat berapa pilih bunga turun	1
5	sangat bantu buku kegiatan koperasi kelola catat anggota upload google drive	0
6	aplikasi sangat bagus manfaat	1
7	aplikasi bagus tambah foto profil nasabah mudah kenal sebagai bukti	0
8	selama guna aplikasi bagi hitung sendiri catat administrasi	-1
9	usaha simpan pinjam kepada pemula sangat bagus mudah pahami aplikasi kembang aman data	1
10	aplikasi bagus kurang karena laporan harian atur tanggal jatuh tempo atur setor harian minggu bulan	-1

b. Deskripsi Nilai *Term*

Langkah atau tahapan kedua dari pembobotan kata TF-IDF yaitu deskripsi nilai *term*. Tahapan tersebut berguna untuk mendefinisikan *term* atau kata pada dokumen. Data *term* dan nilai *term* dapat dilihat pada Tabel 7.

Tabel 7. Data *Term*

Dokumen	<i>Term</i>
1	aplikasi
2	koperasi
3	data
4	google
5	anggota

Dokumen	Term
6	catat
7	administrasi
8	laporan
9	aman
10	usaha

Setelah menentukan nilai *term* dari setiap dokumen, maka tahapan selanjutnya adalah menghitung *Document Frequent* pada data *training* dan data *testing* pada Tabel 8 dan Tabel 9.

Tabel 8. TF-IDF Data *Training*

TF <i>Term</i>	IDF								DF
	D1(1)	D2(1)	D3(1)	D4(1)	D5(1)	D6(1)	D7(1)	D8(1)	
aplikasi	1	1	1	1	0	1	1	1	7
koperasi	1	0	0	0	1	0	0	0	2
data	0	1	1	0	0	0	0	0	2
google	0	1	0	0	1	0	0	0	2
anggota	0	0	1	0	1	0	0	0	2
catat	0	0	0	0	1	0	0	1	2
administrasi	0	0	0	0	0	0	0	1	1
laporan	0	0	0	0	0	0	0	0	0
pengamanan	0	0	0	0	0	0	0	0	0
usaha	1	0	0	0	0	0	0	0	1

Tabel 9. TF-IDF Data *Testing*

TF <i>Term</i>	IDF		DF
	D9(0)	D10(0)	
aplikasi	1	1	2
koperasi	0	0	0
data	1	0	1
google	0	0	0
anggota	0	0	0
catat	0	0	0
administrasi	0	0	0
laporan	0	1	1
pengamanan	1	0	1
usaha	1	0	1

Kesimpulan dari proses perhitungan *Document Frequent* adalah hasil perhitungan nilai DF data training pada term aplikasi yaitu 7. Hasil 7 artinya kata aplikasi muncul pada dokumen 1 hingga dokumen 8 sebanyak 7 kali muncul, begitu juga pada *term* yang lainnya.

c. Menghitung Nilai TF

Langkah atau tahapan ketiga dari pembobotan kata TF-IDF yaitu menghitung nilai TF (*Term Frequent*) yang diperoleh dari sebuah kata di dalam dokumen, serta menjadi kata yang sering dibicarakan dan kemudian dipilih berdasarkan dokumen yang digunakan [8]. Rumus untuk menghitung nilai TF dapat dilihat pada Persamaan 1. Berikut ini merupakan salah satu contoh perhitungan nilai TF *term* aplikasi pada *data training*.

$$TF_{t,d} = (1 + \log(1))$$

$$TF_{t,d} = 1 \tag{1}$$

d. Menghitung Nilai IDF

Langkah atau tahapan keempat dari pembobotan kata TF-IDF yaitu menghitung nilai IDF dengan menggunakan rumus yang ada pada Persamaan 2. Berikut ini merupakan salah satu contoh perhitungan nilai IDF term aplikasi pada *data training*.

$$IDF_t = \log \frac{8}{1 + DF_t}$$

$$IDF_t = 0 \quad (2)$$

e. Menghitung Nilai Bobot (Wtd)

Langkah atau tahapan kelima dari pembobotan kata TF-IDF yaitu menghitung nilai bobot (Wtd) dengan menggunakan Persamaan 3. Berikut ini merupakan salah satu contoh perhitungan nilai bobot (Wtd) term aplikasi pada *data training*.

$$W_{t,d} = 1 * 0$$

$$W_{t,d} = 0 \quad (3)$$

f. Output Normalisasi TF-IDF

Langkah atau tahapan keenam dari pembobotan kata TF-IDF yaitu menghitung normalisasi TF-IDF dengan menggunakan Persamaan 4.

$$V^{\rightarrow} = \frac{w^{\rightarrow}}{\sqrt{\sum_i w_{i,d}^2}} \quad (4)$$

2.2.5 Analisis Menggunakan SVM

Support Vector Machine dilakukan setelah melalui tahap *preprocessing* dan pembobotan nilai TF-IDF. Hasil atau keluaran dari metode ini, yaitu klasifikasi kelas positif, negatif dan netral yang diperoleh berdasarkan bobot pada setiap fitur dokumen teks.

a. Input Normalisasi TF-IDF

Langkah atau tahapan pertama dari klasifikasi *support vector machine* yaitu input normalisasi TF-IDF. Tabel hasil normalisasi *data training* dan *data testing* yang akan dimasukkan ke dalam tahapan klasifikasi *support vector machine*.

b. Transpose Matrix

Langkah atau tahapan kedua dari klasifikasi *support vector machine* yaitu *transpose matrix*. *Trasnpose matrix* didefinisikan sebagai sebuah matriks yang didapatkan dengan cara menukar unsur-unsur baris menjadi unsur-unsur kolom dan sebaliknya.

c. Menghitung Kernel

Langkah selanjutnya yaitu menghitung kernel. pada penelitian ini kernel yang cocok untuk digunakan sebagai modeling pada SVM yaitu kernel *polynomial* karena proses normalisasi data di tahapan *preprocessing*.

d. Menghitung Matrix Hessian

Langkah keempat yaitu menghitung *matrix hessian*, berikut ini merupakan salah satu contoh perhitungan *matrix hessian* dokumen 1 pada fitur 1 data training.

$$D_{ij} = (1 * 1) * (2 + 5^2)$$

$$D_{ij} = 27$$

e. Menentukan Nilai Error

Langkah atau tahapan kelima dari klasifikasi *support vector machine* yaitu menghitung nilai error. Pada langkah ini nilai alfa 0,5 artinya 50% dari 100 tingkat kesalahan yang ditetapkan 0,5 atau 50%. Apabila nilai alfa semakin besar artinya tingkat kesalahan pelatihan data akan semakin menghasilkan model algoritma yang buruk dan apabila nilai alfa semakin kecil artinya tingkat kesalahan pelatihan data akan semakin menghasilkan model algoritma yang baik.

f. Menghitung Lagrange Multiplier

Langkah atau tahapan keenam dari klasifikasi *support vector machine* yaitu menghitung nilai alfa. Perhitungan nilai alfa membutuhkan nilai alfa awal, *cLR*, *gamma*, dan *complexity*.

g. Menghitung Nilai Bias

Langkah atau tahapan ketujuh dari klasifikasi *support vector machine* yaitu menghitung nilai bias. Berikut ini merupakan salah satu contoh perhitungan nilai bias data ke-1 pada data *training*.

$$bias_i = -\frac{1}{2} * (-0,495677094 * 0,003703704 * 17,33126203)$$

$$bias_i = 0,015908721$$

h. Klasifikasi SVM

Langkah selanjutnya yaitu klasifikasi *support vector machine*. Berikut ini merupakan hasil perhitungan klasifikasi SVM pada data testing pada Tabel 10.

Tabel 10. Hasil Klasifikasi Data *Testing*

Dokumen ke-	Class Aktual	Hasil Klasifikasi
9	1	-1
10	-1	-1

i. Optimasi *One Againts-All*

Langkah atau tahapan kesembilan dari klasifikasi *support vector machine* optimisasi *one againts-all*. Metode *one againts-all* ini membandingkan satu kelas dengan semua kelas lainnya, sehingga metode ini juga dikenal sebagai *one againts-therest*, yang mana untuk mengklasifikasikan data kedalam k-kelas, harus membangun sejumlah k model SVM biner. Untuk k model SVM biner yang telah dibangun selanjutnya SVM biner ke-i dilatih menggunakan keseluruhan data training (latih), guna menjawab apakah sebuah data diklasifikasikan sebagai kelas ke-i atau tidak [9]. Berikut merupakan hasil penjumlahan nilai alfa dan nilai bias setiap *class* data *training* Tabel 11 dan data *testing* Tabel 12.

Tabel 11. Penjumlahan Nilai Alfa dan Nilai Bias setiap *Class* Data *Training*

	Nilai Alfa	Nilai Bias
Kelas 1	-1,473454872	0,034015717
Kelas 0	-0,496296296	0,001953018
Kelas -1	-1,424228155	-0,013910216

Tabel 12. Hasil Keputusan Data *Testing*

Dokumen ke-	Class	Keputusan Class Positif	Keputusan Class Netral	Keputusan Class Negatif	Nilai Max	Hasil Keputusan
9	1	0,763140764	0,247540492	0,690855461	0,763140764	Class Positif
10	-1	0,734813849	0,237999281	0,663474921	0,734813849	Class Positif

2.3 Rancangan Pengujian

Tahap selanjutnya adalah pengujian sistem, untuk menguji tingkat akurasi yang dihasilkan dari metode SVM dilakukan dengan menggunakan metode *confusion matrix* [10].

TP (True Positive) : Jumlah prediksi benar bahwa yang diprediksi hasilnya *positive*

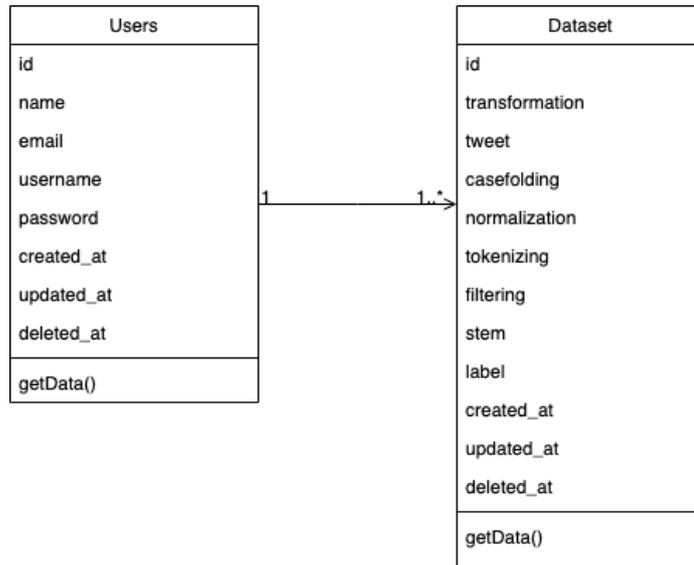
TN (True Negative) : Jumlah prediksi salah bahwa hasil prediksi bernilai *negative*.

FN (False Negative) : Jumlah prediksi salah, harusnya nilai positif diprediksi *negative*.

FP (False Positive) : Jumlah prediksi salah, harusnya nilai *negative* diprediksi positif.

2.4 Rancangan Basis Data

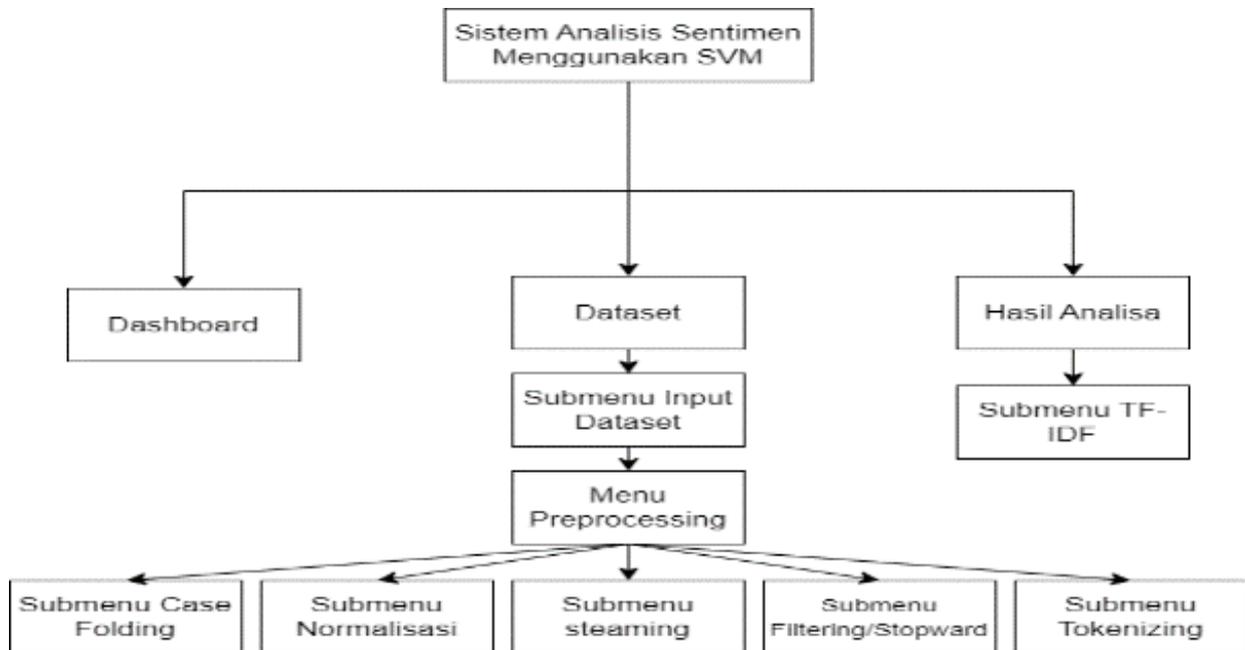
Berikut ini pada Gambar 2 merupakan rancangan class diagram yang akan digunakan dalam pembangunan basis data.



Gambar 2. Class Diagram

2.5 Rancangan Menu

Rancangan menu pada penelitian ini direpresentasikan kedalam struktur navigasi website. Struktur navigasi website menjelaskan bagaimana berbagai halaman di situs Anda diatur dan dihubungkan satu sama lain. Struktur navigasi website dapat mengatur halaman mana yang harus di klik terlebih dulu oleh pengguna, lalu ke halaman mana yang dituju selanjutnya. Struktur navigasi website yang bagus adalah struktur yang ramah terhadap pengguna dan tidak membingungkan. Pada penelitian ini, rancangan menu navigasi dapat dilihat pada gambar 3.



Gambar 3. Rancangan Menu

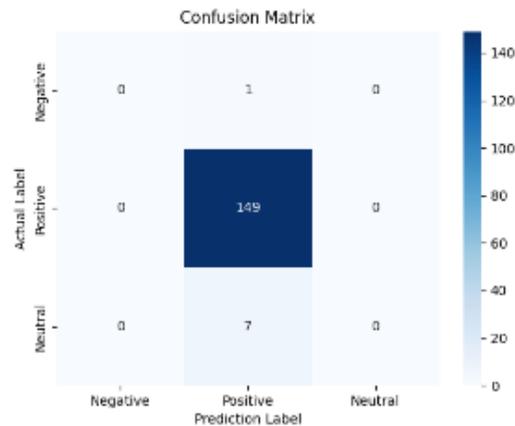
3. HASIL DAN PEMBAHASAN

3.1 Pengujian

Gambar 4 merupakan hasil *confusion matrix* dari metode SVM berupa grafik *heatmap*, serta rumus yang digunakan untuk menghitung *confusion matrix* dapat dilihat pada Tabel 13 merupakan model dari *confusion matrix*.

Tabel 13. Model *Confusion Matrix*

		<i>Confusion Matrix</i>		
Actual label	Negatif	cell 1	cell 2	cell 3
	Positif	cell 4	cell 5	cell 6
	Netral	cell 7	cell 8	cell 9
		Negatif	Positif	Netral
		<i>Prediction Label</i>		



Accuracy = 94.9 %
Precision = 90.07 %
Recall = 94.9 %
F-Score = 92.42 %

Gambar 4. Hasil *Confusion Matrix Program*

Tabel 14 merupakan penjabaran dari hasil *confusion matrix* yang ada pada Gambar 4.

Tabel 14. Hasil *Confusion Matrix*

		<i>Confusion Matrix</i>		
Actual label	Negatif	0	1	0
	Positif	0	149	0
	Netral	0	7	0
		Negatif	Positif	Netral
		<i>Prediction Label</i>		

Tabel 15 merupakan penjumlahan *class* dari hasil *confusion matrix*

	Negatif	Positif	Netral	Total
Negatif	0	1	0	1
Positif	0	149	0	149
Netral	0	7	0	7
Total	0	157	0	157

Jadi hasil perhitungan *confusion matrix* seluruh *class* sebagai berikut:

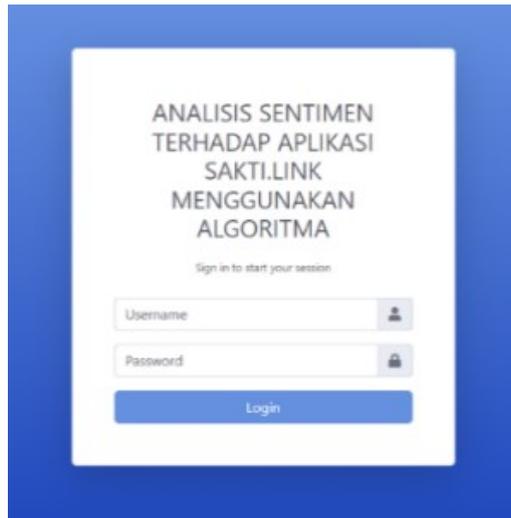
Akurasi = 94,9%
Presisi = 90,07%
Recall = 94,9%
F1 Score = 92,42%

3.2 Tampilan Layar Aplikasi

Sistem analisis sentimen terhadap aplikasi SAKTI.Link ini dibuat dengan *platform website* yang dapat digunakan oleh aktor admin. Ketika masuk ke dalam sistem maka aktor harus melakukan login terlebih dahulu. Berikut merupakan penjelasan hasil interface pada website sistem algoritma SVM untuk penerapan analisis sentimen terhadap aplikasi SAKTI.Link.

a. Halaman *Login*

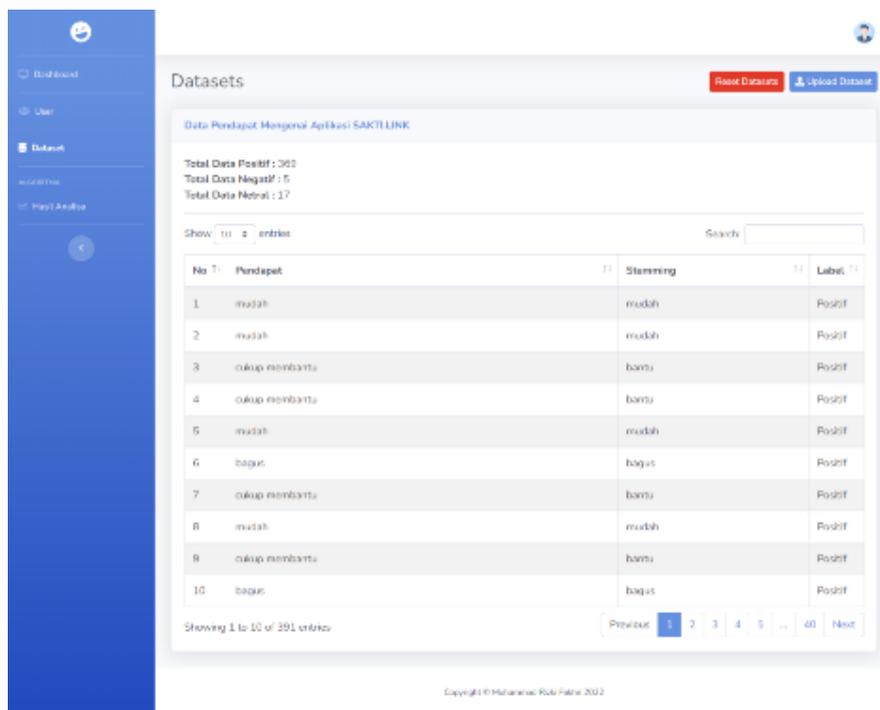
Halaman *login* merupakan halaman yang digunakan admin untuk melakukan *login* dengan mengisi *username* dan *password*. Implementasi halaman *login* dapat dilihat pada Gambar 5.



Gambar 5. Halaman *Login*

b. Halaman *Dataset*

Halaman *dataset* merupakan halaman yang menampilkan informasi *dataset* yang digunakan seperti pendapat, *stemming*, label dan jumlah data positif, jumlah data negatif dan jumlah data netral. Implementasi halaman *dataset* dapat dilihat pada Gambar 6.



Gambar 6. Halaman *Dataset*

c. Halaman Hasil Analisa

Halaman hasil analisa digunakan untuk melakukan analisa terhadap sentimen pengguna aplikasi SAKTI.Link menggunakan metode SVM. Implementasi halaman hasil analisa dapat dilihat pada Gambar 7 sampai dengan Gambar 8. Pada Gambar 7 menampilkan informasi data train yang digunakan untuk melakukan analisa metode SVM yang terdiri dari *stemmed* pendapat, label, jumlah data positif, jumlah data negatif dan jumlah data netral.

Hasil Analisa

ANALISIS SENTIMEN TERHADAP APLIKASI SAKTI.LINK MENGGUNAKAN ALGORITMA

DATA TRAIN

Total Data Positif : 220
Total Data Negatif : 4
Total Data Netral : 10

Show 10 entries Search:

No	Stemmed Tweet	Label
1	aplikasi bagus simple	Positif
2	mudah	Positif
3	mudah paham bantu	Positif
4	bagus menu	Positif
5	pakai aplikasi bantu aplikasi offline coba kembang simpan kati akun google agat data aman ganti hp hpnya mati data hangus	Positif
6	mudah	Positif
7	mudah	Positif
8	bantu	Positif
9	mudah	Positif
10	bagus kolom edit	Positif

Showing 1 to 10 of 234 entries

Previous 1 2 3 4 5 ... 24 Next

Gambar 7. Halaman Hasil Analisa – Data Train

Pada Gambar 8 menampilkan informasi *data test* yang digunakan untuk melakukan analisa metode SVM yang terdiri dari *stemmed* pendapat, label, jumlah data positif, jumlah data negatif dan jumlah data netral.

DATA TEST

Total Data Positive : 149
Total Data Negative : 1
Total Data Netral : 7

Show 10 entries Search:

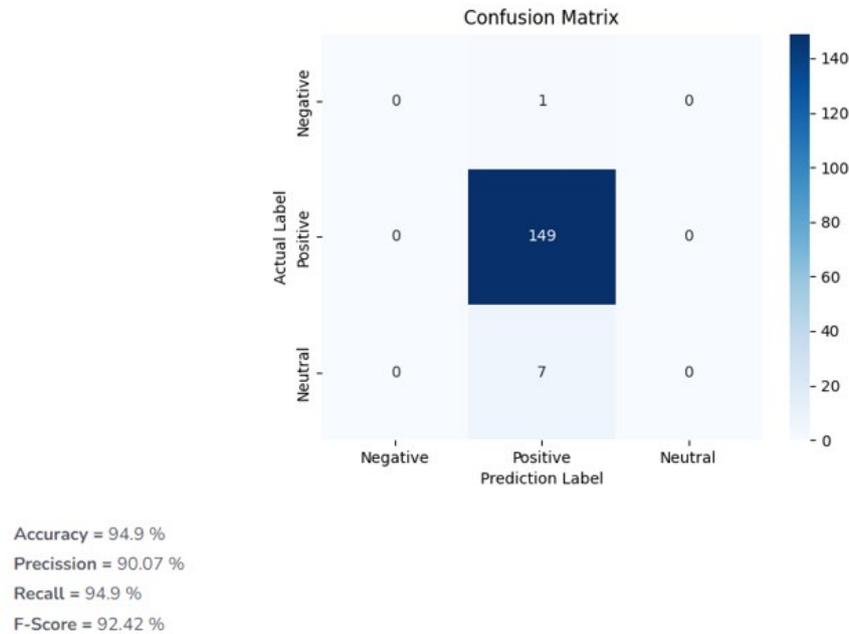
No	Stemmed Tweet	Label
1	bagus	Positif
2	bantu	Positif
3	aplikasi mudah bantu anggota koperasi	Positif
4	bantu	Positif
5	aplikasi bagus bantu anggota koperasi mudah praktis harga produk digital kompetitif	Positif
6	aplikasi bagus sayang print transaksi	Positif
7	cocok finansial kontrol uang	Positif
8	bugnya error beli telkomset pending hari	Negatif
9	mudah	Positif
10	aplikasi bagus foto profile nasabah mudah nali bukti	Positif

Showing 1 to 10 of 157 entries

Previous 1 2 3 4 5 ... 16 Next

Gambar 8. Hasil Analisa – Data Test

Pada Gambar 9 menampilkan informasi hasil prediksi metode SVM berupa grafik *heatmap* yang menampilkan nilai *accuracy*, *precision*, *recall* dan *f1-score*. Nilai tersebut didapatkan dari hasil analisis terhadap sentimen pengguna aplikasi SAKTI.Link.



Gambar 9. Halaman Analisa – Hasil *Confusion Matrix*

4. KESIMPULAN

Dari hasil penelitian yang telah dilakukan terkait analisis sentimen terhadap aplikasi SAKTI.Link menggunakan algoritma *Support Vector Machine* (SVM), maka didapatkan kesimpulan bahwa dalam mengetahui kecenderungan opini, baik yang pro, kontra maupun netral terhadap aplikasi SAKTI.Link, yaitu dengan melakukan analisis pada sentimen pengguna aplikasi SAKTI.Link. Kemudian, dilakukan pelabelan data untuk menentukan sentimen positif, sentimen negatif dan sentiment netral. Selanjutnya, data yang telah diberi label dilakukan preprocessing data yang nantinya akan digunakan untuk proses klasifikasi menggunakan metode SVM, sehingga dihasilkan opini terhadap aplikasi SAKTI.Link.

Hasil analisis sentimen terhadap tanggapan pengguna terkait aplikasi SAKTI.Link menggunakan metode *Support Vector Machine* (SVM), yaitu berupa sentimen positif, sentimen negatif, dan sentimen netral terhadap data aktual dan data prediksi. Selanjutnya dilakukan pengujian metode SVM menggunakan *confusion matrix* dan didapatkan nilai *accuracy* sebesar 94,9%, *precision* sebesar 90,07%, *recall* sebesar 94,9% dan *F1-score* sebesar 92,42%.

DAFTAR PUSTAKA

- [1] M. Demartini, S. Evans and F. Tonelli, “Digitalization Technologies for Industrial Sustainability,” *Procedia manufacturing*, vol. 33, pp. 264-271, 2019.
- [2] S. “SAKTI.Link,” SAKTI.Link, 2020. [Online]. Available: <https://sakti.link/>. [Accessed 2 Februari 2023].
- [3] R. D. Irfan, A. and S. Hedyati, “Aplikasi Pengukur Tingkat Sentimen Pelanggan Berdasarkan Komplain Pelanggan PLN Menggunakan Algoritma K-Nearest Neighbor,” *Jurnal Inovtek Polbeng*, vol. 5, pp. 332-336, 2020.
- [4] I. A. Nikmatun and I. Waspada, “Implementasi Data Mining untuk Klasifikasi Masa Studi Mahasiswa Menggunakan Algoritma K-Nearest Neighbor,” *Jurnal Simetris*, vol. 10, pp. 421-432, 2019.
- [5] A. R. Isnain, A. I. Sakti, D. Alita and N. S. Marga, “Sentimen Analisis Publik Terhadap Kebijakan Lockdown Pemerintah Jakarta Menggunakan Algoritma Svm,” *JDMSI*, vol. 2, no. 1, pp. 31-37, 2021.

- [6] S. D. Pramukti, A. Nugroho and A. S. Sunge, “Analisis Sentimen Masyarakat Dengan Metode Naïve Bayes dan Particle Swarm Optimization,” *Techno.COM*, vol. 21, no. 1, pp. 61-74, 2022.
- [7] L. Ardiani, H. Sujaini and T. , “Implementasi Sentiment Analysis Tanggapan Masyarakat Terhadap Pembangunan di Kota Pontianak,” *Jurnal Sistem dan Teknologi Informasi*, vol. 8, pp. 183-190, 2020.
- [8] A. Alwasi'a, “Analisis Sentimen pada review Aplikasi Berita Online Menggunakan Metode Maximum Entropy (Studi Kasus: Review Detikcom pada Google Play 2019),” 2020.
- [9] R. D. Merianti, “Implementasi Metode Support Vector Machine Dan Random Forest Pada Data Ekspresi Gen Microarray (Studi Kasus : Novel Mutation Target Distinct Subgrups of Medulloblastoma in America),” Universitas Islam Indonesia, Yogyakarta, 2020.
- [10] U. D. Nuswantoro, “Pertemuan Ke 8,” 2021. [Online]. Available: https://repository.dinus.ac.id/docs/ajar/pert_ke_7_sistem_cerdas.pdf.