Komputika: Jurnal Sistem Komputer
Volume 12, Nomor 2, Oktober 2023, hlm. 177 - 186
Terakreditasi Peringkat 3, SK No. 148/M/KPT/2020
DOI: **10.34010/komputika.v12i2.9460**

ISSN: 2252-9039 (print)
ISSN: 2655-3198 (online)

Uji Kernel SVM dalam Analisis Sentimen Terhadap Layanan Telkomsel di Media Sosial Twitter

Pangestu Fremmuzar[1], Anna Baita[2*]

[1,2)]Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta
Jl. Padjajaran, Ring Road Utara, Kel. Condongcatur, Kec. Depok, Kab. Sleman, Prop. Daerah Istimewa Yogyakarta 55283

**email: anna@amikom.ac.id*

(Naskah masuk: *15 Maret 2023*; diterima untuk diterbitkan: *19 Juli 2023*)

ABSTRAK – Telkomsel merupakan salah satu penyedia layanan internet di Indonesia yang diluncurkan pada tahun 1995. Sebagai penjedia jasa layanan internet dengan pengguna terbanyak, Telkomsel menjadi pusat perhatian para pengguna internet di Indonesia. Hal ini mengundang opini dan sudut pandang pengguna terhadap Telkomsel yang biasa disebut dengan sentimen. Salah satu media yang biasa digunakan untuk mengutarakan sebuah opini dan sudut pandang adalah Twitter. Twitter merupakan platform media sosial yang sering menjadi tempat untuk berbagi dan menyebarkan berita, diskusi ide, dan opini pengguna Twitter. Dalam penelitian ini algoritma yang digunakan adalah *Support Vector Machine*. Dalam *Support Vector Machine,* terdapat kernel trick yang akan digunakan untuk mengetahui performa kernel dan menganalisis sentimen. Sentimen yang dianalisis berjumlah 537 cuitan yang dikumpulkan dengan cara scraping. Cuitan yang terkumpul akan melewati tahap *preprocessing,* yaitu *cleaning*, *casefolding*, *tokenizing*, *normalization*, *stemming*, *stopword removal*, dan *detokenize*. Sebuah sentimen diklasifikasi menjadi 2 label, yaitu positif dan negatif. Berdasarkan hasil pengujian, kernel sigmoid memiliki performa terbaik dengan nilai *accuracy* sebesar 0.950, *precision* sebesar 0.945, *recall* sebesar 0.860, *f1-score* sebesar 0.896 dan sentimen condong ke arah negatif.

Kata Kunci – telkomsel, SVM, *preprocessing*, sentimen, kernel.

SVM Kernel Test in Sentiment Analysis of Telkomsel Services on Twitter Social Media

ABSTRACT – *Telkomsel is an internet service provider in Indonesia which was launched in 1995. As an internet service provider with the most users, Telkomsel has become the center of attention of internet users in Indonesia. This invites user opinions and perspectives on Telkomsel, which is commonly referred to as sentiment. One of the media commonly used to express an opinion and point of view is Twitter. Twitter is a social media platform that is often a place for sharing and spreading news, discussing ideas, and opinions of Twitter users. In this study the algorithm used is the Support Vector Machine. In the Support Vector Machine, there is a kernel trick that will be used to determine kernel performance and analyze sentiment. The sentiments analyzed amounted to 537 tweets collected by scraping. The collected tweets will go through the preprocessing stage, namely cleaning, case folding, tokenizing, normalization, stemming, stopword removal, and detokenizing. A sentiment is classified into 2 labels, namely positive and negative. Based on the test results, the sigmoid kernel has the best performance with an accuracy value of 0.950, a precision of 0.945, a recall of 0.860, an f1-score of 0.896 and sentiment tends towards negative.*

Keywords – telkomsel, SVM, *preprocessing*, *sentiment*, kernel.

1. PENDAHULUAN

Interconnection-networking (Internet) adalah sistem jaringan komputer di seluruh dunia yang menghubungkan satu sama lain dalam segala hal dunia menggunakan standar *internet protocol suite* [1]. Di Indonesia terdapat berbagai macam dan jenis layanan internet yang bisa digunakan oleh masyarakat umum. Salah satu penyedia layanan internet adalah Telkomsel. Telkomsel merupakan penyedia jasa layanan internet yang ada di Indonesia sejak tahun 1995 [2]. Pada tahun 2020, Telkomsel menjadi penyedia jasa layanan internet yang paling banyak digunakan oleh masyarakat [3]. Sebagai penjedia jasa layanan internet dengan pengguna terbanyak, Telkomsel menjadi pusat perhatian para pengguna internet di Indonesia. Hal ini mengundang opini dan sudut pandang pengguna terhadap Telkomsel yang biasa disebut dengan sentimen.

Sentimen adalah pernyataan subyektif yang mencerminkan persepsi seseorang tentang peristiwa atau objek tertentu [4]. Dalam hal pengembangan bisnis, sebuah sentimen dapat analisis untuk mengetahui sudut pandang konsumen apakah konsumen merasa puas atau tidak terhadap produk yang dikeluarkan dan memberi kesimpulan bagaimana pengembangan produk untuk kedepannya [5]. Analisis sebuah sentimen bisa dilakukan dengan menggunakan pendekatan machine learning. Machine learning merupakan salah satu aplikasi artificial intelligence (AI) yang fokus dalam mengembangkan sistem agar dapat belajar sendiri tanpa pemrograman ulang. Machine learning membutuhkan data (data training) sebagai proses pembelajaran sebelum hasilnya muncul [6].

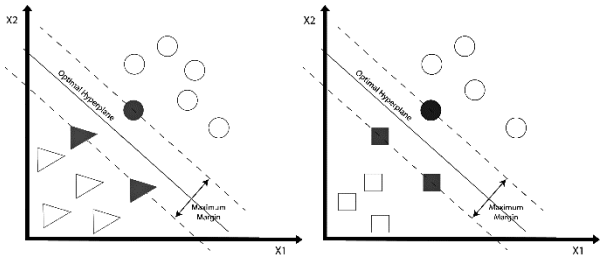
Algoritma yang umum biasanya digunakan dalam *machine learning* adalah *Naive Bayes* (NB), *Support Vector Machine* (SVM) dan *Maximum Entropy* (EM) [7]. Penelitian Pooja dan Bhalla R [8] juga mengatakan algoritma yang biasanya digunakan dalam klasifikasi dan regresi adalah *Support Vector Machine* (SVM), *Naïve Bayes* (NB), *Decision Tree* (DT), *K- Nearest Neighbors* (KNN), *Random Forest* (RF), *Artifcial Neural Network* (ANN), dan *Linear Regressions* (LR). Wijaya K dan Karyawatia A [9] melakukan penelitian tentang pengaruh dari penggunakan kernel pada algortima SVM terhadap sentimen yang akan dianalisis. Kernel yang digunakan pada penelitian Wijaya K dan Karyawatia A adalah kernel linear, kernel polynomial, dan kernel rbf. Dari 3 jenis kernel yang digunakan, kernel linear mendapatkan hasil yang paling bagus dari kernel yang lain. Hasil penelitian lain seperti pada penelitian Syahputra H [10] tentang sentimen komunitas pada *online store* dianalisis menggunakan algoritma SVM dengan kernel linear, rbf, dan sigmoid. Dalam penelitian Syahputra H, kernel yang menghasilkan nilai akurasi terbaik adalah kernel sigmoid dengan nilai akurasi sebesar 82 persen.

Dalam proses klasifikasi menggunakan *Support Vector Machine,* terdapat kernel trick yang bisa digunakan dalam mengklasifikasi sebuah sentimen untuk mendapatkan hasil yang lebih optimal [11]. Dalam penelitian Aulia T dan kawan-kawan, terdapat empat jenis kernel yang bisa digunakan dalam algoritma *Support Vector Machine* yaitu kernel linear, kernel rbf, kernel polynomial, dan kernel sigmoid [12]. Oleh karena itu, algoritma *Support Vector Machine* digunakan pada penelitian ini untuk menguji performa kernel yang ada pada *Support Vector Machine* dalam mengklasifikasi dan menganalisis sentimen terhadap Telkomsel.

2. METODE DAN BAHAN

Support Vector Machine

Algoritma *Support Vector Machine* termasuk dalam metode *supervised learning*, sehingga *dataset* yang akan dianalisis perlu diberikan label terlebih dahulu [13]. Mekanisme kerjanya adalah dengan menemukan *hyperplane* yang sesuai untuk mengklasifikasikan sampel data yang sudah dikumpulkan [14]. Algoritma *Support Vector Machine* akan mencari *hyperplane* terbaik dengan jarak *margin* yang maksimal seperti ilustrasi pada gambar 1.



Gambar 1. Ilustrasi *Hyperplane* dan *Margin* Maksimal

Dalam *Support Vector Machine* terdapat beberapa jenis kernel yang bisa digunakan seperti SVM Linear Kernel, SVM RBF Kernel, SVM Polynomial Kernel dan SVM Sigmoid Kernel [15].

Kernel Linear

Saat menganalisis dengan *kernel linear*, parameter yang dapat dioptimalkan adalah C atau Cost. Optimalisasi parameter C biasanya dilakukan dengan trial and error [15]. Persamaan kernel linear adalah sebagai berikut [13].

$$\mathrm{K}(xi, x) = X_i^T X \quad (1)$$

Kernel RBF

Saat menganalisis dengan kernel *RBF*, parameter yang dapat dioptimalkan adalah Cost (C) dan Gamma (γ). Sama seperti kernel linear, optimasi parameter biasanya dilakukan dengan cara *trial* dan *error* [15]. Persamaan Kernel RBF adalah sebagai berikut [13].

$$K(xi, x) = exp(-\gamma|X_i^T X|^2) \quad (2)$$

Kernel Polynomial

Saat menganalisis dengan *kernel polynomial*, parameter yang dapat dioptimasi adalah *Cost* (C) dan *Degree* (d). Sama halnya dengan kernel *linear* dan *RBF*, optimasi parameter biasanya dilakukan dengan trial dan errror [15]. Persamaan *Kernel Polynomial* adalah sebagai berikut [13].

$$K(xi, x) = (\gamma. X_i^T X + r)^p \quad (3)$$

Kernel Sigmoid

Saat menganalisis dengan *kernel sigmoid*, parameter yang dapat dioptimasi adalah *Cost* (C) dan *Gamma* (γ). Nilai parameter γ dan c harus dipilih dengan hati-hati untuk menghindarinya kesalahan dalam hasil yang didapatkan [16]. Persamaan *kernel sigmoid* adalah sebagai berikut [13].

$$K(xi, x) = tanh(\gamma. X_i^T X + r) \quad (4)$$

Scraping

Pada penelitian ini terdapat beberapa tahapan proses yang dilakukan, yaitu *scraping, dataset preparation, labeling, preprocessing,* TF-IDF, *splitting data,* dan SVM *implementation*. Pada tahap ini, data akan diambil dengan cara scraping. Proses scraping dilakukan secara otomatis menggunakan bahasa pemrograman python. Data yang diambil adalah cuitan twitter berbahasa Indonesia. Keyword yang digunakan dalam proses scraping data adalah "Telkomsel" dan didapatkan data mentah sebanyak 3500 cuitan twitter.

Dataset Preparation

Pada tahap ini, dataset akan diolah agar menjadi data yang layak dan siap pakai untuk melakukan sebuah analisis sentimen. Proses yang dilakukan yaitu menghapus data yang duplikat, menghapus user yang duplikat, menghapus cuitan dari Telkomsel, dan menghapus data yang kosong.

Labeling

Pada tahap ini, dataset akan diberikan label positif, netral, dan negatif. Proses pemberian label pada dataset dilakukan secara manual dengan membaca cuitan pada dataset satu persatu. Setelah semua data diberi label, selanjutnya data yang berlabel netral akan dihapus. Dalam penelitian ini, sentimen yang dianalisis hanya sentimen yang berlabel positif atau negatif.

Preprocessing

Preprocessing meliputi proses mempersiapkan data untuk digunakan dalam *text mining*. Sehingga dataset menjadi bersih dan siap untuk digunakan [17]. Tahap *preprocessing* yang dilakukan yaitu:

Cleaning

Cleaning merupakan proses pengurangan noise pada teks dengan cara menghilangkan beberapa karakter tertentu seperti tanda baca, angka, simbol, emoticon dan link url yang menuju ke sebuah website. Sebuah data akan dianalisis kualitasnya dengan cara mengedit, memodifikasi, atau menghapus kata yang tidak lengkap, tidak akurat, dan data yang dianggap tidak perlu dalam database untuk menghasilkan data yang berkualitas tinggi [18].

Casefolding

Case folding adalah salah satu tahap penting dalam *text preprocessing* untuk melakukan *text mining*. Semua huruf akan diubah menjadi huruf kecil untuk mencegah kasus yang sensitif. Hal ini dapat meningkatkan kinerja algoritma dalam mengklasifikasi tanpa mempertimbangkan teks yang tidak konsisten [19].

Tokenizing

Tokenizing dapat didefinisikan sebagai pemecahan teks menjadi bagian praktis yang biasa disebut dengan token, seperti kata, frasa, simbol atau unit lain agar pengerjaan teks lebih efektif [19].

Normalization

Normalization merupakan proses mengubah kata yang ada di dataset menjadi bentuk baku yang lebih formal dan disesuaikan dengan KBBI [20]. Contoh nya seperti merubah kata 'kepake' menjadi 'terpakai'. Proses ini diperlukan karena data cuitan di sosial media Twitter banyak menggunakan bahasa tidak baku.

Stemming

Stemming merupakan proses merubah kata-kata yang ditemukan dalam teks menjadi bentuk dasarnya dengan menghilangkan prefiks (memotong awalan kata) dan sufiks (memotong akhiran kata). Misalnya, kata "memakan", "dimakan", "termakan" akan ditransformasi menjadi kata "makan" [19].

Stopword Removal

Stopword removal atau biasa juga disebut dengan filtering adalah tahap pemilihan kata yang penting dari hasil tokenizing [18]. Stopword Removal akan menghilangkan kata yang tidak memiliki arti dan pengaruh dalam menganalisis sebuah data. Kata-kata yang tidak lebih dari 3 huruf tidak memiliki arti yang berarti. Misalnya "di", "ke", "oh" dan lain-lain. Jadi, menghapus kata-kata ini mengurangi perhitungan ekstra yang tidak perlu [21].

Detokenize

Detokenize merupakan proses menggabungkan sebuah kata yang sudah dipecah menjadi sebuah kalimat. Proses *detokenize* akan menggabungkan

sebuah kata yang dipisah dengan tanda "," dan menghilangkan tanda "," dari teks.

TF-IDF

TF-IDF(*Term Frequency-Inverse Document Frequency*) merupakan proses pemberian bobot pada suatu kata, semakin banyak suatu kata muncul dalam data, maka semakin tinggi nilai bobot kata tersebut. Metode TF-IDF dapat menangkap kata-kata yang sering muncul dengan cara menghitung frekuensi kata dan menghindari kata tidak penting yang ada di dataset [22]. Pembobotan diperoleh berdasarkan jumlah kemunculan term dalam kalimat (TF) dan berdasarkan jumlah kemunculan term dalam seluruh kalimat dokumen (IDF). Bobot suatu term akan semakin tinggi jika term tersebut sering muncul dalam dokumen dan semakin rendah jika term tersebut muncul dalam beberapa dokumen[23]. Pada tahap ini, kata disajikan dalam format vektor dan TF-IDF. Dengan menggunakan metode TF-IDF dalam proses pembobotan dapat menghasilkan vektor dengan beberapa suku, dimana setiap kata dapat dikenali, yang dihitung sebagai satu fitur [24]. Untuk melakukan proses TF-IDF dapat menggunakan persamaan berikut:

$$\text{tf} = 0{,}5 + 0{,}5\frac{tf}{\max(tf)} \quad (5)$$

$$\text{idf} = \ln\frac{n}{df} + 1 \quad (6)$$

$$\text{TF-IDF} = \text{tf} \times \text{idf} \quad (7)$$

Splitting Data

Sebelum melakukan analisis, *dataset* akan dibagi menjadi *data training* dan *data testing*. *Data training* bertujuan untuk melatih algoritma untuk mempelajari sebuah dataset sehingga akan menghasilkan sebuah model. Model ini nantinya akan digunakan dalam melakukan proses pengujian terhadap *data testing*.

SVM Implementation

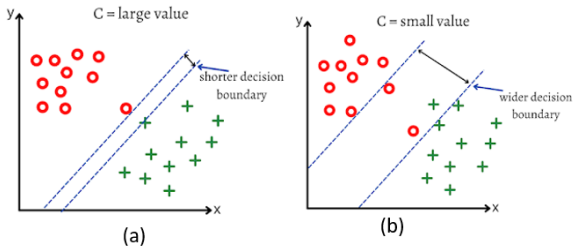
Pembuatan Model

Pembuatan model akan dilakukan menggunakan *data training* yang sudah melalui tahap *preprocessing* dan perhitungan TF-IDF. Dalam proses pembuatan model, parameter yang digunakan akan berbeda sesuai dengan jenis kernel yang digunakan

Hyperparameter tuning SVM

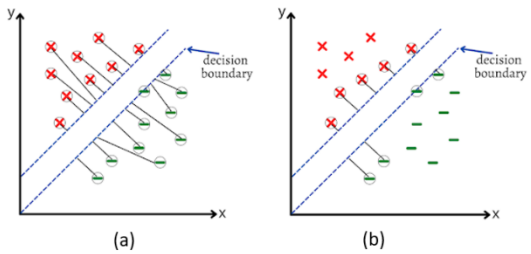
Terdapat beberapa parameter yang akan diujikan dalam penelitian ini, antara lain Cost (C), gamma(γ), serta degree(d)

a. Cost. Parameter cost terdapat di semua kernel SVM. nilai cost yang besar akan memperkecil jarak decision boundary, sehingga sifat generalisasi dari classifier akan menghilang. Pengaruh nilai cost terhadap klasifikasi diperlihatkan gambar 2 berikut ini.



Gambar 2. (a). klasifikasi dengan cost rendah (b) klasifikasi dengan cost tinggi

b. Gamma. Gamma menentukan seberapa besar pengaruh yang dimiliki oleh sebuah proses training. Jika nilai gamma tinggi, maka titik-titik yang berada disekitar garis hyperplane akan dipertimbangkan dalam perhitungan. Sebaliknya jika nilai gamma rendah maka titik yang berada jauh dari garis hyperplane akan dipertimbangkan dalam perhitungan. Ilustrasi penggunaan gamma dalam klasifikasi diperlihatkan oleh gambar 3 berikut ini.



Gambar 3. (a). klasifikasi dengan gamma rendah (b) klasifikasi dengan gamma tinggi

c. Degree. Parameter degree hanya terdapat pada kernel polynomial. Semakin tinggi nilai degree maka hyperplane akan semakin melengkung.

1. Pengujian Model
   Setelah mendapatkan model, selanjutnya data testing akan diuji menggunakan model yang sudah didapatkan. Hasil pengujian akan dipilih berapa nilai parameter yang paling optimal dengan cara membandingkan nilai akurasi yang didapatkan dari setiap nilai parameter yang digunakan.
2. Menghitung Performa
   Langkah selanjutnya adalah menghitung performa dari setiap kernel menggunakan *confusion matrix*. Dengan *confusion matrix* kita dapat mengetahui berapa nilai *accuracy*, *precision*, *recall*, dan *f1-score*.

3. HASIL DAN PEMBAHASAN

Dataset pada penelitian ini diperoleh dengan menggunakan teknik *scraping* menggunakan *keyword* "Telkomsel" yang diambil pada bulan Desember 2022. Proses scraping ini mendapatkan data mentah sebesar 3500 cuitan. Dataset tersebut kemudian dilakukan dataset preparation, yakni menghapus tweet yang duplikat, menghapus user yang duplikat, menghapus cuitan dari Telkomsel, dan menghapus data yang kosong. Hasil pengolahannya ditunjukkan oleh tabel 1.

Tabel 1. Penghapusan data

Jenis Data	Jumlah Data
Duplicate tweet	101
Duplicate User	671
Tweet from @Telkomsel	1.231
Data Kosong	10

Berdasarkan tabel 1, dataset yang dihapus ada sebanyak 2013 data. Data akhir menjadi 1.487 data.

Data Labelling

Proses pelabelan data pada penelitian ini dilakukan secara manual. Contoh pelabelan ditunjukkan oleh tabel 2.

Tabel 2. Pelabelan

	Cuitan Twitter	Label
1	@wintierngels @Telkomsel Di sini sih alhamdulillah kak paketan Telkomsel mayan murah, aku biasa langganan paket internet sakti ga sampe 100k sebulan.	Positif
2	Hei @Telkomsel kenapa fitur daily check in nya ditiadakan? Padahal kepake banget tuh bonusnya kalo lagi ngirit/bokek yg mau beli paket data ðŸ¥²	Negatif

Pada penelitian ini hanya menggunakan data tweet yang bernilai positif dan negatif. Oleh karena itu, data tweet yang memiliki sentiment netral dihapus. Tweet netral yang dihapus sejumlah 950 tweet. Total data bersih yang dipergunakan dalam penelitian ini ditunjukkan oleh tabel 3.

Tabel 3. Data dan Label

Label	Jumlah Data
Positive	85
Negative	452

Dataset yang digunakan dalam penelitian ini tidak seimbang, sehingga metric yang digunakan untuk menghitung performa algoritma adalah f1-score.

Data Preprocessing

a. Hasil Cleaning

Hasil cleaning ditunjukkan oleh tabel 4.

Tabel 4. Hasil Cleaning

Cuitan Twitter	Hasil Cleaning
@wintierngels @Telkomsel Di sini sih alhamdulillah kak paketan Telkomsel mayan murah, aku biasa langganan paket internet sakti ga sampe 100k sebulan.	Di sini sih alhamdulillah kak paketan Telkomsel mayan murah aku biasa langganan paket internet sakti ga sampe k sebulan

b. Hasil Case folding

Proses case folding dilakukan dengan mengubah teks menjadi lowercase. Hasil casefolding perlihatkan oleh table 5.

Tabel 5. Hasil Case Folding

Cuitan Twitter	Hasil Case folding
Di sini sih alhamdulillah kak paketan Telkomsel mayan murah aku biasa langganan paket internet sakti ga sampe k sebulan	di sini sih alhamdulillah kak paketan telkomsel mayan murah aku biasa langganan paket internet sakti ga sampe k sebulan

c. Hasil Tokenizing

Pada proses tokenizing, setiap spasi dianggap sebagai pemisah dari sebuah token. Pengubahan kalimat menjadi token ditunjukkan oleh table 6.

Tabel 6. Hasil Tokenizing

Cuitan Twitter	Hasil Tokenizing
di sini sih alhamdulillah kak paketan telkomsel mayan murah aku biasa langganan paket internet sakti ga sampe k sebulan	['di', 'sini', 'sih', 'alhamdulillah', 'kak', 'paketan', 'telkomsel', 'mayan', 'murah', 'aku', 'biasa', 'langganan', 'paket', 'internet', 'sakti', 'ga', 'sampe', 'k', 'sebulan']

d. Hasil Normalization

kata-kata yang tidak baku akan dikonversi menjadi kata-kata yang baku seperti diperlihatkan oleh table 7.

Tabel 7. Hasil Normalization

Cuitan Twitter	Hasil Normalization
['di', 'sini', 'sih', 'alhamdulillah', 'kak', 'paketan', 'telkomsel', 'mayan', 'murah', 'aku', 'biasa', 'langganan', 'paket', 'internet', 'sakti', 'ga', 'sampe', 'k', 'sebulan']	['di', 'sini', 'sih', 'alhamdulillah', 'kak', 'paketan', 'telkomsel', 'lumayan', 'murah', 'saya', 'biasa', 'langganan', 'paket', 'internet', 'sakti', 'tidak', 'sampai', 'k', 'sebulan']

e. Hasil Stemming

Proses stemming dalam penelitian ini menggunakan algoritma nazief & adriani yang terdapat dalam library sastrawi. hasil stemming ditunjukkan oleh table 8.

Tabel 8. Hasil Stemming

Cuitan Twitter	Hasil Tokenizing
['di', 'sini', 'sih', 'alhamdulillah', 'kak', 'paketan', 'telkomsel', 'lumayan', 'murah', 'saya', 'biasa', 'langganan', 'paket', 'internet', 'sakti', 'tidak', 'sampai', 'k', 'sebulan']	['di', 'sini', 'sih', 'alhamdulillah', 'kak', 'paket', 'telkomsel', 'lumayan', 'murah', 'saya', 'biasa', 'langgan', 'paket', 'internet', 'sakti', 'tidak', 'sampai', 'k', 'bulan']

f. Hasil Stopword Removal

Pada tahap stopword removal term yang terdiri dari 1 karakter huruf juga akan di hapus, seperti diperlihatkan oleh tabel 9.

Tabel 9. Hasil Stopword Removal

Cuitan Twitter	Hasil Tokenizing
['di', 'sini', 'sih', 'alhamdulillah', 'kak', 'paket', 'telkomsel', 'lumayan', 'murah', 'saya', 'biasa', 'langgan', 'paket', 'internet', 'sakti', 'tidak', 'sampai', 'k', 'bulan']	['sini', 'sih', 'alhamdulillah', 'kak', 'paket', 'telkomsel', 'lumayan', 'murah', 'biasa', 'langgan', 'paket', 'internet', 'sakti', 'bulan']

g. Hasil Detokenize

Sebelum Ekstraksi fitur, maka semua token dalam kalimat diubah Kembali menjadi kalimat, seperti diperlihatkan oleh table 10.

Tabel 10. Hasil Stopword Removal

Cuitan Twitter	Hasil Tokenizing
['sini', 'sih', 'alhamdulillah', 'kak', 'paket', 'telkomsel', 'mayan', 'murah', 'biasa', 'langgan', 'paket', 'internet', 'sakti', 'bulan']	sini sih alhamdulillah kak paket telkomsel mayan murah biasa langgan paket internet sakti bulan

Feature Extraction Using TF-IDF

Hasil dari proses ekstraksi fitur menggunakan TF-IDF diperlihatkan oleh gambar 4.

Implementasi SVM

Proses implementasi SVM pada penelitian ini akan menggunakan pembagian *data testing* dan *data training* yang mendapatkan nilai *f1-score* paling optimal. Nilai *f1-score* dipilih karena jumlah sentimen positif dan sentimen negatif tidak seimbang. Pembagian jumlah data testing dan data training yang digunakan dalam penelitian akan ditentukan melalui perbandingan hasil *f1-score* yang didapatkan. Pembagian jumlah *data testing* dan *data training* yang digunakan bisa dilihat pada tabel 11.

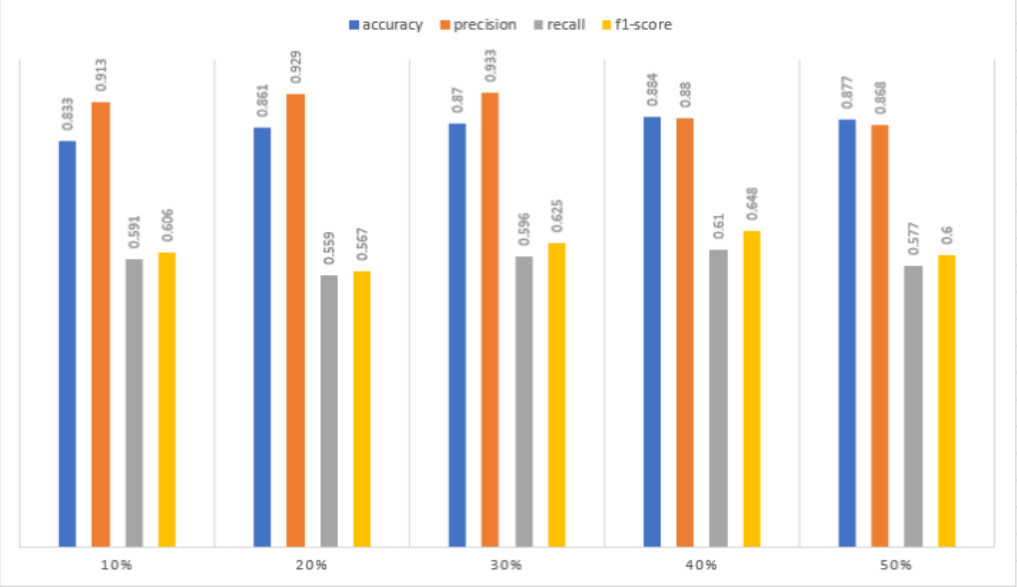
Tabel 11. Pembagian Jumlah Dataset

Perbandingan	Jumlah Split
A	10% testing, 90% training
B	20% testing, 80% training
C	30% testing, 70% training
D	40% testing, 60% training
E	50% testing, 50% training

Dalam mencari dan melakukan perbandingan

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	...	D528	D529	D530	D531	D532	D533	D534	D535	D536	D537
abyss	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
access	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
acer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
adek	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
airplane	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
you	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.529816	0.0	0.0
youtuban	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
youtube	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
zaidan	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
zoom	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0

Gambar 4. Hasil TF-IDF



Gambar 5. Perbandingan Performa Pembagian Dataset

nilai *f1-score*, algoritma SVM akan diimplementasi dengan kernel dan nilai parameter *default*. Hal ini dilakukan dengan tujuan mencari pembagian jumlah *data testing* dan *data training* yang paling optimal. Hasil implementasi SVM *default* dapat dilihat pada gambar 5. Berdasarkan gambar 5 perbandingan yang memiliki nilai *f1-score* paling tinggi adalah adalah perbandingan dengan pembagian *dataset* 40% sebagai *data testing* dan 60% sebagai *data training* dengan nilai *f1-score* 0.648. Perbandingan ini akan digunakan dalam implementasi *kernel trick* algoritma *Support Vector Machine* untuk mendapatkan nilai *f1-score* yang lebih tinggi dengan mengoptimalkan nilai parameter pada masing-masing kernel.

Implementasi Kernel Linear

Pengujian algoritma *Support Vector Machine* dengan kernel *linear* menggunakan nilai parameter Cost (C) yang berbeda-beda dengan tujuan mencari nilai *f1-score* yang paling optimal. Nilai parameter yang diujikan yaitu 0.5, 1, 2, 3, dan 4. Hasil pengujian nilai parameter dengan kernel linear dapat dilihat pada tabel 12.

Tabel 12. Hasil Pengujian Parameter Kernel Linear

C	Accuracy	Precision	Recall	F1-score
0.5	0.884	0.880	0.610	0.648
1	0.888	0.833	0.653	0.696
2	**0.907**	**0.888**	**0.704**	**0.757**
3	0.902	0.881	0.688	0.739
4	0.907	0.915	0.691	0.747

Berdasarkan tabel 12 dari semua nilai parameter yang diujikan, nilai *f1-score* terbaik adalah 0.757 dengan nilai parameter C = 2. Setelah dilakukan percobaan dengan C < 2 nilai f1-score yang didapatkan lebih kecil dari C = 2. Begitu juga dengan nilai parameter C > 2 tidak mendapatkan nilai *f1-score* yang lebih baik dari C = 2

Implementasi Kernel RBF

Pengujian algoritma *Support Vector Machine* dengan kernel RBF menggunakan nilai parameter Cost (C) dan Gamma (γ) yang berbeda-beda dengan tujuan mencari nilai *f1-score* yang paling optimal. Nilai parameter yang diujikan yaitu 0.5, 1, 2, 3, dan 4. Hasil pengujian nilai parameter dengan kernel RBF dapat dilihat pada tabel 13.

Tabel 13. Hasil Pengujian Parameter Kernel RBF

C	γ	Accuracy	Precision	Recall	F1-score
1	0.5	0.884	0.880	0.610	0.648
1	1	0.884	0.880	0.610	0.648
2	0.5	0.893	0.896	0.642	0.690
2	1	0.898	0.947	0.645	0.697
2	2	0.879	0.938	0.581	0.606
3	0.5	0.893	0.865	0.656	0.703
3	1	0.898	0.947	0.645	0.697
3	2	0.879	0.938	0.581	0.606
4	**0.5**	**0.898**	**0.903**	**0.659**	**0.710**
4	1	0.898	0.947	0.645	0.697
4	2	0.879	0.938	0.581	0.606

Pada tabel 13 nilai parameter yang diperlihatkan hanya parameter yang menghasilkan nilai f1-score lebih dari 0.6. Dari 25 kali percobaan, hanya 11 parameter yang menghasilkan nilai f1-score di atas 0.6. Dari semua nilai parameter yang diujikan, nilai f1-score terbaik pada kernel RBF adalah 0.710 dengan nilai parameter C = 4 dan parameter γ = 0.5.

Implementasi Kernel Polynomial

Pengujian algoritma Support Vector Machine dengan kernel polynomial menggunakan nilai parameter Cost (C) dan Degree (d) yang berbeda-beda dengan tujuan mencari nilai *f1-score* yang paling

optimal. Nilai yang diujikan pada parameter C yaitu 0.5, 1, 2, 3, 4 dan nilai yang diujikan pada parameter d yaitu 1, 2, 3, 4, 5. Hasil pengujian nilai parameter dengan kernel polynomial dapat dilihat pada tabel 14.

Tabel 14. Hasil Pengujian Parameter Kernel Polynomial

C	d	Accuracy	Precision	Recall	F1-score
0.5	1	0.884	0.880	0.610	0.648
1	1	0.888	0.833	0.653	0.696
1	2	0.884	0.940	0.597	0.630
2	**1**	**0.907**	**0.888**	**0.704**	**0.757**
2	2	0.888	0.942	0.613	0.654
3	1	0.902	0.881	0.688	0.739
3	2	0.888	0.942	0.613	0.654
4	1	0.907	0.915	0.691	0.747
4	2	0.888	0.942	0.613	0.654

Pada tabel 14 nilai parameter yang diperlihatkan hanya parameter yang menghasilkan nilai *f1-score* lebih dari 0.6. Dari 25 kali percobaan, hanya 9 parameter yang menghasilkan nilai *f1-score* di atas 0.6. Dari semua nilai parameter yang diujikan, nilai *f1-score* terbaik pada kernel polynomial adalah 0.757 dengan nilai parameter C = 2 dan parameter d = 1.

Implementasi Kernel Sigmoid

Pengujian algoritma Support Vector Machine dengan kernel Sigmoid menggunakan nilai parameter Cost (C) dan Gamma (γ) yang berbeda-beda dengan tujuan mencari nilai akurasi yang paling optimal. Nilai parameter yang diujikan yaitu 0.5, 1, 2, 3, 4. Hasil pengujian nilai parameter dengan kernel sigmoid dapat dilihat pada tabel 15.

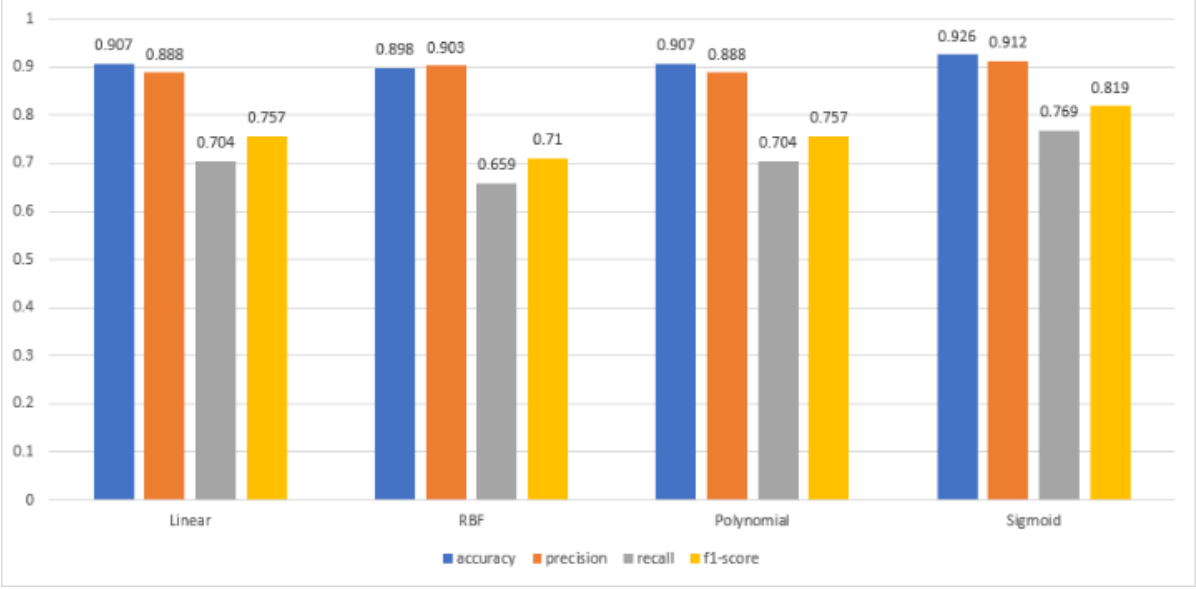
Tabel 15. Hasil Pengujian Parameter Kernel Sigmoid

C	γ	Accuracy	Precision	Recall	F1-score
1	3	0.916	0.901	0.737	0.789
1	4	0.912	0.876	0.734	0.781
2	1	0.916	0.883	0.750	0.796
2	2	0.907	0.842	0.744	0.781
2	3	0.907	0.842	0.744	0.781
2	4	0.902	0.825	0.742	0.774
3	1	0.916	0.901	0.737	0.789
3	**2**	**0.926**	**0.912**	**0.769**	**0.819**
3	3	0.912	0.862	0.747	0.789
3	4	0.902	0.807	0.782	0.794
4	0.5	0.907	0.888	0.704	0.757
4	1	0.907	0.869	0.718	0.765
4	2	0.907	0.833	0.758	0.788
4	3	0.907	0.826	0.771	0.795
4	4	0.893	0.790	0.750	0.767

Pada tabel 15 nilai parameter yang diperlihatkan hanya parameter yang menghasilkan nilai *f1-score* lebih dari 0.75. Dari 25 kali percobaan, ada 15 parameter yang menghasilkan nilai f1-score di atas 0.75. Dari semua nilai parameter yang diujikan, nilai *f1-score* terbaik pada kernel sigmoid adalah 0.819 dengan nilai parameter C = 3 dan parameter γ = 2.

Perbandingan Hasil Kernel

Dari semua kernel yang diujikan, kernel sigmoid mendapatkan nilai *f1-score* tertinggi dari kernel yang lain. Perbandingan hasil *f1-score* dari tiap kernel dapat dilihat pada gambar 6.



Gambar 6. Perbandingan Performa Kernel

Berdasarkan gambar 6, kernel sigmoid mendapatkan nilai *f1-score* paling tinggi diantara kernel SVM yang lain, yaitu sebesar 0.819 dengan nilai parameter C = 3 dan parameter γ = 2.

Nilai perbandingan F1-Score beserta parameter optimal dari masing-masing kernel ditunjukkan pada tabel berikut ini

Tabel 16. Perbandingan F1-Score

Kernel	C	γ	Degree	F1-score
Linear	2	-	-	0.757
RBF	4	0.5	-	0.710
Polynomial	2	-	1	0.757
Sigmoid	3	2	-	0.819

Hasil Akhir

Berdasarkan hasil uji tersebut diatas, kernel yang memiliki performa terbaik adalah kernel Sigmoid. Dari hasil klasifikasi menggunakan kernel sigmoid, didapatkanlah kumpulan kata dari masing-masing sentimen. Kata yang sering muncul dalam sentimen negatif dan positif bisa dilihat pada gambar 7 dan gambar 8.

Gambar 7. Sentimen Negatif Yang Sering Muncul

Berdasarkan gambar 7, kumpulan kata tersebut bisa diurutkan menjadi lima kata yang paling sering digunakan, seperti pada tabel 17.

Tabel 17. Kata Yang Sering Muncul Pada Sentimen Negatif

No	Kata	Jumlah Muncul
1	sinyal	135
2	jaring	72
3	jelek	59
4	lelet	56
5	paket	54

Gambar 8. Sentimen Positif Yang Sering Muncul

Berdasarkan gambar 8, kumpulan kata tersebut bisa diurutkan menjadi lima kata yang paling sering digunakan, seperti pada tabel 18.

Tabel 18. Kata Yang Sering Muncul Pada Sentimen Positif

No	Kata	Jumlah Muncul
1	murah	27
2	paket	18
3	harga	13
4	bagus	10
5	kuota	9

4. KESIMPULAN

Dari hasil pengujian dan analisis, diketahui bahwa algoritma *Support Vector Machine* dengan kernel sigmoid mendapatkan performa yang lebih baik dari kernel yang lain. Kernel sigmoid dengan parameter C = 3 dan parameter γ = 2 menghasilkan nilai *accuracy* sebesar 0.926, *precision* sebesar 0.912, *recall* sebesar 0.769, dan *f1-score* sebesar 0.819. Berdasarkan hasil klasifikasi menggunakan kernel sigmoid, kata dari sentimen negatif yang sering muncul adalah "sinyal" dan kata yang sering muncul pada sentimen positif adalah "murah".

DAFTAR PUSTAKA

[1] A. G. Gani, "Sejarah dan Perkembangan Internet di Indonesia," *Jurnal Mitra Manajemen*, vol. 5, no. 2, 2020.

[2] T. S. Siregar and A. Lubis, "Faktor – Faktor Yang Mempengaruhi Kepuasan Pelanggan Kartu Telkomsel Di Kecamatan Medan Baru," *Jurnal Ilmiah Manajemen dan Bisnis (JIMBI)*, vol. 3, no. 1, pp. 92–105, May 2022, doi: 10.31289/jimbi.v3i1.988.

[3] S. Nurajizah, N. A. Ambarwati, and S. Muryani, "Sistem Pendukung Keputusan Pemilihan Internet Service Provider Terbaik Dengan Metode Analytical Hierarchy Process," *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, vol. 6, no. 3, pp. 231–238, Aug. 2020, doi: 10.33330/jurteksi.v6i3.632.

[4] M. R. Firdaus, F. M. Rizki, F. M. Gaus, and I. K. Susanto, "Analisis Sentimen Dan Topic Modelling Dalam Aplikasi Ruangguru," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 4, no. 1, pp. 66–76, 2020.

[5] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artif Intell Rev*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.

[6] C. Chazar and B. Erawan, "Machine Learning Diagnosis Kanker Payudara Menggunakan Algoritma Support Vector Machine,"

INFORMASI (Jurnal Informatika Dan Sistem Informasi), vol. 12, no. 1, pp. 67–80, 2020.

[7] A. R. Isnain, A. I. Sakti, D. Alita, and N. S. Marga, "Sentimen Analisis Publik Terhadap Kebijakan Lockdown Pemerintah Jakarta Menggunakan Algoritma SVM," *Jurnal Data Mining Dan Sistem Informasi*, vol. 2, no. 1, pp. 31–37, 2021.

[8] Pooja and R. Bhalla, "A Review Paper on the Role of Sentiment Analysis in Quality Education," *SN Comput Sci*, vol. 3, no. 6, p. 469, Sep. 2022, doi: 10.1007/s42979-022-01366-9.

[9] K. D. Y. Wijaya and A. E. Karyawati, "The Effects of Different Kernels in SVM Sentiment Analysis on Mass Social Distancing," *Jurnal Elektronik Ilmu Komputer Udayana p-ISSN*, vol. 2301, p. 5373, 2020.

[10] H. Syahputra, "Sentiment Analysis of Community Opinion on Online Store in Indonesia on Twitter using Support Vector Machine Algorithm (SVM)," *J Phys Conf Ser*, vol. 1819, no. 1, p. 012030, Mar. 2021, doi: 10.1088/1742-6596/1819/1/012030.

[11] E. Utami and A. C. Khotimah, "Comparison Naïve Bayes Classifier, K-Nearest Neighbor and Support Vector Machine In The Classification Of Individual On Twitter Account," *Jurnal Teknik Informatika (JUTIF)*, vol. 3, no. 3, pp. 673–680, 2022.

[12] T. M. P. Aulia, N. Arifin, and R. Mayasari, "Perbandingan Kernel Support Vector Machine (SVM) Dalam Penerapan Analisis Sentimen Vaksinisasi Covid-19," *SINTECH (Science and Information Technology) Journal*, vol. 4, no. 2, pp. 139–145, 2021.

[13] M. Rahardi, A. Aminuddin, F. F. Abdulloh, and R. A. Nugroho, "Sentiment Analysis of Covid-19 Vaccination using Support Vector Machine in Indonesia," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, 2022, doi: 10.14569/IJACSA.2022.0130665.

[14] B. Gaye, D. Zhang, and A. Wulamu, "Improvement of Support Vector Machine Algorithm in Big Data Background," *Math Probl Eng*, vol. 2021, pp. 1–9, Jun. 2021, doi: 10.1155/2021/5594899.

[15] H. al Azies, D. Trishnanti, and E. M. P. Hermanto, "Comparison of Kernel Support Vector Machine (SVM) in Classification of Human Development Index (HDI)," *IPTEK Journal of Proceedings Series*, no. 6, pp. 53–57, 2019.

[16] I. S. Al-Mejibli, J. K. Alwan, and H. A. Dhafar, "The effect of gamma value on support vector machine performance with different kernels," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, p. 5497, Oct. 2020, doi: 10.11591/ijece.v10i5.pp5497-5506.

[17] R. Kurniawan and A. Apriliani, "Analisis Sentimen Masyarakat Terhadap Virus Corona Berdasarkan Opini Dari Twitter Berbasis Web Scraper," *Jurnal INSTEK (Informatika Sains dan Teknologi)*, vol. 5, no. 1, p. 67, Apr. 2020, doi: 10.24252/instek.v5i1.13686.

[18] 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, 2021.

[19] M. Işik and H. Dağ, "The impact of text preprocessing on the prediction of review ratings," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 28, no. 3, pp. 1405–1421, May 2020, doi: 10.3906/elk-1907-46.

[20] S. Fransiska, R. Rianto, and A. I. Gufroni, "Sentiment Analysis Provider by. U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method," *Scientific Journal of Informatics*, vol. 7, no. 2, pp. 203–212, 2020.

[21] A. Kulkarni and S. Mhaske, "Tweet Sentiment Analysis and Study and Comparison of Various Approaches and Classification Algorithms Used," *IRJET*, 2020.

[22] T. Wang, K. Lu, K. P. Chow, and Q. Zhu, "COVID-19 sensing: negative sentiment analysis on social media in China via BERT model," *Ieee Access*, vol. 8, pp. 138162–138169, 2020.

[23] A. Z. Z. Abidin and A. Sukmadinata, "Sistem Deteksi Kerusakan pada Sistem Operasi Menggunakan Metode TF-IDF dan Cosine Similarity," *Jurnal Ilmiah Informatika*, vol. 8, no. 02, pp. 107–112, 2020.

[24] D. Darwis, E. S. Pratiwi, and A. F. O. Pasaribu, "Penerapan Algoritma Svm Untuk Analisis Sentimen Pada Data Twitter Komisi Pemberantasan Korupsi Republik Indonesia," *Edutic - Scientific Journal of Informatics Education*, vol. 7, no. 1, Nov. 2020, doi: 10.21107/edutic.v7i1.8779.