Klasifikasi Kateogori Cerita Pendek Menggunakan *Support Vector Machine*

M Faisal Afandi[1], Ngurah Agus Sanjaya ER[2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Udayana, Bali
Jln. Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, 08261, Bali, Indonesia
[1]faisalafandi024@student.unud.ac.id
[2]agus_sanjaya@unud.ac.id

Abstract

Short stories are fascinating literary works to read because they present concise narratives that don't require readers to spend a lot of time to complete a story. Although the stories are short, determining the story category still requires careful reading to understand the content. However, it can become challenging when there is a large number of stories to be classified. Therefore, this research aims to develop a system that can automatically classify short story texts. The method used in this research is SVM (Support Vector Machine). The research is conducted to assist in automatically classifying short stories and create a system that bridges people to enjoying written works while enhancing literacy. The data used consists of short stories in the categories of romance, horror, and religion. The best-performing model is obtained through the training and validation process using new data. The results of testing the SVM method with a 70:30 data scenario, and hyperparameter C=10, gamma = 0.1 with kernel rbf or gamma = scale with kernel linear, yield an accuracy of 96% with a precision of 96.72%, recall of 96.36%, and an f1-score of 96.40%.

Keywords: *Cerita Pendek, Teks Klasifikasi, TF-IDF, Support Vector Machine*

1. Pendahuluan

Pada masa kini, persebaran informasi terus berkembang dengan pesat. Informasi yang ada terus bertambah dengan variatif yang sangat beragam [1]. Kemajuan teknologi menjadi penyebab persebaran informasi dapat dilakukan dengan mudah [2]. Salah satu informasi yang dapat terdampak yaitu persebaran karya tulis cerita pendek yang kini dapat diakses di media online dengan mudah. Cerita pendek umumnya dibedakan berdasarkan beberapa kategori, seperti cerita cinta, horor, fiksi, agama, dan sebagainya. Akan tetapi, dengan banyaknya informasi yang terus berkembang menyebabkan proses mengenali kategori cerita pendek menjadi lebih sulit jika dilakukan secara manual dengan membaca setiap isi dari cerita yang ingin dikategorikan. Oleh karena itu, dengan adanya proses klasifikasi kategori cerita pendek secara otomatis dengan metode tertentu akan sangat membantu dalam memilah dan mengenali cerita pendek berdasarkan kategorinya. Klasifikasi teks merupakan proses untuk mengklasifikasikan dokumen ke dalam kategori yang sudah ditentukan dengan tujuan membantu dalam mengorganisir informasi secara otomatis sehingga dapat dipahami oleh pengguna [3].

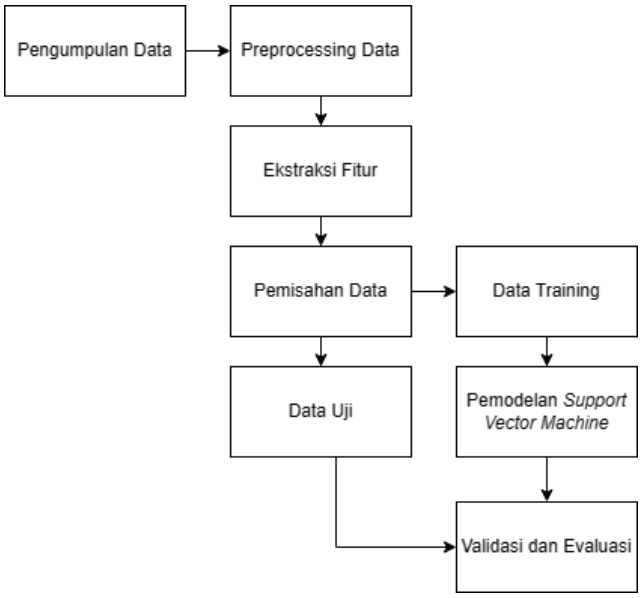
Penelitian mengenai klasifikasi sudah dilakukan oleh peneliti sebelumnya seperti penggunaan metode *Support Vector Machine* dalam mengklasifikasi jenis pantun menghasilkan akurasi 81,91% [3]. Kemudian penggunaan metode *Support Vector Machine* untuk klasifikasi berita menghasilkan akurasi 88% [2]. Penggunaan metode *Naïve Bayes* untuk klasifikasi cerita pendek berbahasa bali berdasarkan umur menghasilkan akurasi 72% [4]. Lalu komparasi algoritma klasifikasi *Support Vector Machine*, *Naïve Bayes Classification*, dan *K-Nearest Neighbor* pada analisis sentimen mendapatkan akurasi terbaik pada algoritma *Support Vector Machine* [5].

Penelitian kali ini akan mengusulkan dan membangun sebuah sistem yang dapat mengklasifikasikan teks cerita pendek berdasarkan kategori. Kategori cerita pendek yang akan digunakan adalah kategori cinta, horor, dan agama. Metode yang digunakan adalah *Support Vector Machine* dengan melihat dari penelitian sebelumnya bahwa algoritma ini mendapatkan hasil akurasi yang terbaik. Diharapkan dengan adanya sistem ini, akan memudahkan pengguna dalam mengakses, memahami, dan mengetahui karya tulis cerita pendek untuk menambah iterasi dan pengetahuan.

2. Metode Penelitian

2.1. Desain Penelitian

Desain penelitian yang dilakukan dapat dilihat pada Gambar 1, di bawah ini:



Gambar 1. Jnatia

Tahapan pertama pada penelitian ini yaitu melakukan pengumpulan data yang akan digunakan untuk penelitian berupa cerita pendek dengan 3 kategori, yaitu cerita cinta, horor, dan agama. Selanjutnya dilakukan *preprocessing* data yang akan digunakan. Kemudian dilakukan proses ekstraksi fitur dengan TF-IDF. Melakukan pemisahan data untuk memisahkan data yang akan digunakan sebagai pelatihan model dan pengujian. Selanjutnya melakukan pemodelan dengan metode *Support Vector Machine* yang kemudian akan dilakukan evaluasi pada model untuk menentukan klasifikasi cerita pendek.

2.2. Pengumpulan Data

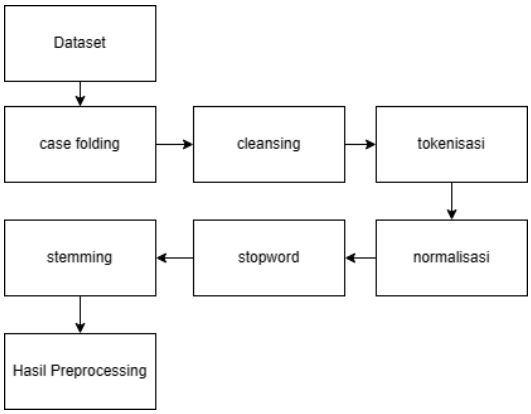
Dataset yang akan digunakan pada sistem ini, yaitu cerita pendek yang sudah dilabeli untuk masing-masing kategori cerita cinta, horor, dan agama. Data yang akan digunakan diambil dari situs cerpenmu.com yang diambil pada tanggal 10 Juni 2023. Data yang digunakan berjumlah 182 data, dengan 60 data cerita cinta dan 61 data untuk masing-masing cerita horor dan agama.

	judul	cerita	genre
0	Berdamai Dengan Luka	Sandra duduk di tepi pantai, merenung tentang ...	cinta
1	Ruteng; Kutemukan Rumah Istimewa	Seisi kota terlilit ikatan musim hujan yang ti...	cinta
2	Sepatu Bersih	Liburan sekolah hampir usai. Seragam seragam s...	cinta
3	Orang Yang Beruntung	Sinar matahari pagi yang menyinari belahan bum...	cinta
4	Senorita	I wish I could pretend I didn?t need ya? Kaki ...	cinta

Gambar 2. Detail *Dataset*

2.3. *Pre-processing* Data

Pada tahapan *preprocessing* data dilakukan untuk mempersiapkan data yang digunakan sebelum proses ekstraksi fitur dan pemodelan. Hal ini dilakukan agar data dapat diolah dengan lebih efisien dan menghasilkan hasil yang lebih akurat. Adapun alur dari tahapan *preprocessing* data dapat dilihat pada Gambar 3.



Gambar 3. Alur *Pre-processing*

Dalam tahapan ini data melalui tahap *case folding* untuk merubah seluruh karakter menjadi *lowercase* atau huruf kecil. Selanjutnya dilakukan proses *cleansing* untuk menghapus karakter dan tanda baca yang tidak diperlukan pada data. Lalu dilakukan proses tokenisasi untuk memisahkan kalimat menjadi beberapa token untuk setiap kata. Setelah itu dilakukan proses normalisasi untuk mengubah kata-kata aneh ke dalam bentuk kata yang mendekati seperti ‘yg’ menjadi ‘yang’. Kemudian dilakukan proses stopword untuk menghilangkan kata yang tidak memiliki makna. Tahapan terakhir yaitu melakukan proses stemming untuk mengubah kata ke dalam bentuk kata dasar [4].

2.4. Ekstraksi Fitur

Ekstraksi fitur yang digunakan pada penelitian ini yaitu TF-IDF yang merupakan pengalian kedua algoritma yakni *Term Frequency* dan *Inverse Document Frequency*. TF (*Term Frequency*) merupakan banyanya jumlah term pada suatu data yang diperhitungkan berdasarkan tingkat kemunculannya dalam satu dokumen. Semakin besar nilai TF dari suatu dokumen maka semakin tinggi bobot dokumen nya. Sedangkan IDF (*Inverse Document Frequency*) merupakan pengurangan nilai dari TF sebelumnya, jika sebuah kata sering muncul dalam dokumen tersebut maka nilai TF-IDF akan semakin kecil. Semakin jarang suatu kata muncul maka nilai TF-IDF akan semakin besar. Hal ini karena semakin jarang sebuah kata dalam dokumen lain, menunjukan bahwa kata tersebut mempunyai pengaruh yang besar [2].

Nilai TF dapat dicari dengan menghitung frekuensi kemunculan *term* pada *dataset*. Kemudian untuk mencari nilai IDF dapat menggunakan formula persamaan 1:

$$IDF_j = log\ (\frac{n}{DF_j}) \quad (1)$$

n = jumlah total dokumen dalam data
j = kata dasar
DF_j = jumlah dokumen dimana kata j muncul

Setelah mendapatkan nilai TF dan IDF dapat menggunakan nilai-nilai tersebut untuk mencari nilai TF-IDF dengan formula persamaan 2:

$$TF - IDF = TF * IDF \quad (2)$$

2.5. Pemisahan Data

Tahapan selanjutnya yaitu memisahkan data yang akan digunakan sebagai pelatihan dan pengujian. Pada penelitian ini digunakan 3 skenario pembagian data, yaitu 80% data latih : 20% data uji, 70% data latih : 30% data uji, dan 60% data latih : 40% data uji dari total data 182 cerita pendek.

2.6. *Support Vector Machine*

Support Vector Machine (SVM) merupakan algoritma *machine learning* yang sering digunakan pada klasifikasi data berupa teks dengan mencari *hyperplane* beserta margin maksimal terbaik sebagai pemisah antar kelas. Pada model klasifikasi, SVM mempunyai konsep yang lebih kuat dan lebih jelas secara matematis. Pada representasi 2 dimensi fungsi yang dipakai untuk klasifikasi antar kelas disebut *line whereas*, sedangkan pada representasi 3 dimensi fungsi yang digunakan disebut *plane similarity* [6]. Dalam algoritma SVM, terdapat beberapa kernel yang dapat digunakan, diantaranya kernel Linear, RBF, Polynomial, dan Sigmoid. Kernel Linear akan baik digunakan ketika data yang digunakan terpisah secara linier, sedangkan Kernel Polynomial, Sigmoid dan RBF biasa digunakan saat data yang digunakan tidak dapat terpisah secara linier [7].

2.7. Evaluasi

Evaluasi kinerja pada model klasifikasi yang dilakukan dapat di ukur dengan menggunakan *confusion matrix*. *Confusion matrix* merupakan tabel yang merangkum kinerja dari model *machine learning* pada sejumlah data uji. Matriks ini memiliki dua baris dan dua kolom yang melaporkan jumlah *true*-positif, *false*-negatif, *false*-positif, dan *true*-negatif. Setiap baris dalam matriks mewakili kelas aktual, sementara setiap kolom mewakili kelas yang diprediksi. *Confusion matrix* memberikan hasil pengujian dengan 4 parameter, yaitu akurasi, *presisi, recall*, dan F1-*score* [6]. Akurasi yaitu mengukur peforma dokumen yang bernilai positif diantara seluruh dokumen. *Presisi* yaitu mengukur peforma dokumen yang bersifat relevan dan bernilai positif diantara seluruh dokumen yang bersifat relevan. *Recall* yaitu mengukur performa dokumen yang bersifat relevan dan bernilai positif diantara seluruh dokumen yang bernilai benar. F1-*score* yaitu mengukur rata-rata harmonic dari *presisi* dan *recall* [4].

3. Hasil dan Pembahasan

Pada penelitian ini, tahap awal adalah *preprocessing* data meliputi c*ase folding* untuk merubah seluruh karakter menjadi *lowercase*, *cleansing* untuk menghapus karakter dan tanda baca yang tidak diperlukan, tokenisasi untuk memisahkan kalimat menjadi beberapa token untuk setiap kata, normalisasi untuk mengubah kata-kata aneh ke dalam bentuk kata yang tepat, *stopword* untuk menghilangkan kata yang tidak memiliki makna, *stemming* untuk mengubah kata ke dalam bentuk kata dasar. Hasil *preprocessing* dapat dilihat pada gambar :

	cerita	genre	Case_Folding	Cleansing	Tokenize	Normalization	Stopword	Stemming
0	Sandra duduk di tepi pantai, merenung tentang ...	cinta	sandra duduk di tepi pantai, merenung tentang ...	sandra duduk di tepi pantai merenung tentang m...	[sandra, duduk, di, tepi, pantai, merenung, te...	sandra duduk di tepi pantai merenung tentang m...	sandra duduk tepi pantai merenung masa lalu su...	sandra duduk tepi pantai renung masa lalu suli...
1	Seisi kota terlilit ikatan musim hujan yang ti...	cinta	seisi kota terlilit ikatan musim hujan yang ti...	seisi kota terlilit ikatan musim hujan yang ti...	[seisi, kota, terlilit, ikatan, musim, hujan, ...	seisi kota terlilit ikatan musim hujan yang ti...	seisi kota terlilit ikatan musim hujan tinggi ...	isi kota lilit ikat musim hujan tinggi jalan r...
2	Liburan sekolah hampir usai. Seragam seragam s...	cinta	liburan sekolah hampir usai. seragam seragam s...	liburan sekolah hampir usai seragam seragam se...	[liburan, sekolah, hampir, usai, seragam, sera...	liburan sekolah hampir usai seragam seragam se...	liburan sekolah hampir usai seragam seragam se...	libur sekolah hampir usai seragam seragam seko...
3	Sinar matahari pagi yang menyinari belahan bum...	cinta	sinar matahari pagi yang menyinari belahan bum...	sinar matahari pagi yang menyinari belahan bum...	[sinar, matahari, pagi, yang, menyinari, belah...	sinar matahari pagi yang menyinari belahan bum...	sinar matahari pagi menyinari belahan bumi dit...	sinar matahari pagi sari bahan bumi tambah kic...
4	I wish I could pretend I didn?t need ya? Kaki ...	cinta	i wish i could pretend i didn?t need ya? kaki ...	wish could pretend didnt need ya kaki kita ...	[wish, could, pretend, didnt, need, ya, kaki, ...	wish could pretend didnt need ya kaki kita mel...	wish could pretend didnt need kaki melangkah l...	wish could pretend didnt need kaki lang lamban...
...	...	...	...	...	...	...	...	...
177	Di suatu malam di kota Jakarta, burhan baru pu...	Horror	di suatu malam di kota jakarta, burhan baru pu...	di suatu malam di kota jakarta burhan baru pul...	[di, suatu, malam, di, kota, jakarta, burhan, ...	di suatu malam di kota jakarta burhan baru pul...	suatu malam kota jakarta burhan baru pulang ke...	suatu malam kota jakarta burhan baru pulang ke...
178	Pada suatu hari ada seorang bernama Dian, ia a...	Horror	pada suatu hari ada seorang bernama dian, ia a...	pada suatu hari ada seorang bernama dian ia ad...	[pada, suatu, hari, ada, seorang, bernama, dia...	pada suatu hari ada seorang bernama dian ia ad...	suatu hari seorang bernama dian adalah seorang...	suatu hari orang nama dian adalah orang sopir ...
179	Waktu pulang kerja, Yadi melihat sebuah kaca m...	Horror	waktu pulang kerja, yadi melihat sebuah kaca m...	waktu pulang kerja yadi melihat sebuah kaca ma...	[waktu, pulang, kerja, yadi, melihat, sebuah, ...	waktu pulang kerja yadi melihat sebuah kaca ma...	waktu pulang kerja yadi melihat sebuah kaca ma...	waktu pulang kerja yad lihat buah kaca mata ge...
180	Pada suatu hari ada lima orang sahabat yaitu R...	Horror	pada suatu hari ada lima orang sahabat yaitu r...	pada suatu hari ada lima orang sahabat yaitu r...	[pada, suatu, hari, ada, lima, orang, sahabat,...	pada suatu hari ada lima orang sahabat yaitu r...	suatu hari lima orang sahabat raya jessie vega...	suatu hari lima orang sahabat raya jessie vega...
181	Pada suatu hari aku dan teman-temanku pergi ke...	Horror	pada suatu hari aku dan teman-temanku pergi ke...	pada suatu hari aku dan temantemanku pergi ke ...	[pada, suatu, hari, aku, dan, temantemanku, pe...	pada suatu hari aku dan temantemanku pergi ke ...	suatu hari aku temantemanku pergi gunung ungar...	suatu hari aku temantemanku pergi gunung ungar...

Gambar 4. Hasil *Preprocessing*

Hasil dari *preprocessing* kemudian diubah ke vector menggunakan metode ekstraksi fitur TF-IDF *Term Frequency Inverse Document Frequency*. Berikut hasil ekstraksi fitur.

```
0.0000 0.0274 0.0000 0.0000 0.000(
0.0000 0.0632 0.0000 0.0000 0.000(
0.0000 0.0000 0.0000 0.0000 0.000(
0.0000 0.0000 0.0000 0.0000 0.000(
0.0000 0.0000 0.0000 0.0000 0.000(
0.0000 0.0000 0.0000 0.0000 0.000(
0.0000 0.0000 0.0000 0.0000 0.000(
0.0000 0.0000 0.0000 0.0000 0.000(
0.0000 0.0000 0.0000 0.0000 0.000(
```

Gambar 5. Hasil TF-IDF

Dataset dibagi menjadi 2 skenario, yaitu 80% data latih: 20% data uji, dan 70% data latih : 30% data uji. Kemudian, melatih model menggunakan algoritma SVM. Setelah pelatihan dilakukan didapatkan nilai akurasi terbesar pada skenario 70%:30% sebesar 96% berdasarkan *hyperplane* terbaik berdasarkan *Grid Search*, yaitu nilai C: 10, gamma: 0.1, dan kernel: rbf. Sedangkan pada skenario 80%:20% mendapatkan akurasi yang tidak berbeda jauh, yaitu 95%. Sedangkan dengan skenario 60%:40% mendapatkan akurasi sebesar 90%.

Tabel 1. Hasil Klasifikasi *Hyperplane* Berdasarkan *Grid Search*

Skenario	Akurasi
60:40	90%
70:30	96%
80:20	95%

	precision	recall	f1-score	support
Agama	1.00	0.95	0.97	19
Horror	1.00	0.94	0.97	18
cinta	0.90	1.00	0.95	18
accuracy			0.96	55
macro avg	0.97	0.96	0.96	55
weighted avg	0.97	0.96	0.96	55

Gambar 6. Hasil Akurasi Skenario 70:30

Lalu dilakukan eksperimen melakukan komparasi perbedaan kernel SVM dan pengaruhnya terhadap nilai akurasi dan penggunaan gamma *scale*. Hasil akurasi dari masing masing kernel dapat dilihat pada tabel berikut.

Tabel 2. Hasil Klasifikasi Tiap Kernel

Skenario		Akurasi
rbf	60:40	85%
	70:30	91%
	80:20	92%
linear	60:40	90%
	70:30	96%
	80:20	95%
poly	60:40	75%
	70:30	87%
	80:20	86%
sigmoid	60:40	88%
	70:30	93%
	80:20	95%

	precision	recall	f1-score	support
Agama	1.00	0.95	0.97	19
Horror	1.00	0.94	0.97	18
cinta	0.90	1.00	0.95	18
accuracy			0.96	55
macro avg	0.97	0.96	0.96	55
weighted avg	0.97	0.96	0.96	55

Gambar 7. Hasil Klasifikasi Kernel Linear dengan skenario 70:30

Berdasarkan pengujian skenario dan parameter kernel pada tabel diatas. Dihasilkan akurasi tertinggi pada parameter C yang bernilai 10, gamma = *scale*, menggunakan kernel linear, dan skenario data latih dan uji 70%:30% yaitu dengan nilai akurasi 96% sebanding dengan penggunaan kernel rbf dengan gamma = 0.1.

4. Kesimpulan

Berdasarkan hasil evaluasi, didapatkan bahwa algoritma *Support Vector Machine* menghasilkan performa paling baik dengan penggunaan kernel linear dengan gamma = *scale* dibanding kernel lainnya dengan nilai C=10 dan perbandingan data 70:30 menghasilkan akurasi yang sama sebesar 96% dengan *presisi* sebesar 96.72%, *recall* sebesar 96.36%, dan *f1-score* sebesar 96.40%. Penggunaan kernel rbf dengan gamma = 0.1 juga menghasilkan nilai akurasi yang sama yaitu 96%. Berdasarkan performa hasil klasifikasi tersebut, dapat disimpulkan bahwa algoritma *Support Vector Machine* dapat mengklasifikasikan data teks cerita pendek berdasarkan kategorinya dengan sangat baik.

Untuk penelitian yang akan datang terdapat beberapa hal yang akan dilakukan, yaitu menerapkan seleksi fitur untuk mempercepat dan meningkatkan akurasi mengingat banyaknya kata pada cerita pendek yang menyebabkan proses menjadi lama. Kemudian menambahkan kategori yang akan diklasifikasikan beserta *dataset* yang akan digunakan untuk dapat menghasilkan tingkat akurasi yang lebih akurat.

Daftar Pustaka

[1] K. I. Gunawan dan J. Santoso, "Multilabel Text Classification Menggunakan SVM dan Doc2Vec Classification pada Dokumen Berita Bahasa Indonesia," *Journal of Information System, Graphics, Hospitality and Technology,* vol. III, no. 01, pp. 29-38, 2021.

[2] R. Nanda, E. Haerani, S. K. Gusti dan S. Ramadhani, "Klasifikasi Berita Menggunakan Metode Support Vector Machine," *Jurnal Nasional Komputasi dan Teknologi Informasi,* vol. V, no. 02, pp. 269-278, 2022.

[3] H. N. Irmanda dan R. Astriratma, "Klasifikasi Jenis Pantun dengan Metode Support Vector Machines (SVM)," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi),* vol. IV, no. 5, pp. 915-922, 2021.

[4] L. Ristiari, A. E. Karyawati, I. P. G. H. Suputra, A. Muliantara, I. D. M. B. A. Darmawan dan I. M. Widiartha, "Klasifikasi Cerita Pendek Berbahasa Bali Berdasarkan Umur Pembaca dengan Metode Naive Bayes," *Jurnal Elektronik Ilmu Komputer Udayana,* vol. X, no. 4, pp. 363-370, 2022.

[5] J. Ipmawati, K. dan E. T. Luthfi, "Komparasi Teknik Klasifikasi Teks Mining Pada Analisis Sentimen," *Indonesian Journal on Networking and Security,* vol. VI, no. 1, pp. 28-36, 2017.

[6] R. W. Pratiwi, S. F. H, D. D. I. Af'idah, Q. R. A dan A. G. F, "Analisis Sentimen Pada Review Skincare Female Daily Menggunakan Metode Support Vector Machine (SVM)," *Journal of Informatics, Information System, Software Engineering and Applications,* vol. I, no. 1, pp. 40-46, 2021.

[7] V. Setiawan dan I. K. G. Suhartana, "Implementasi Algoritma Support Vector Machine dalam Deteksi Depresi Pada Twitter," *JNATIA,* vol. I, no. 1, pp. 285-290, 2022.

[8] C. Chairunnisa, I. Ernawati dan M. M. Santoni, "Klasifikasi Sentimen Ulasan Pengguna Aplikasi PeduliLindungi di Google Play Menggunakan Algoritma Support Vector Machine dengan Seleksi Fitur Chi-Square," *JURNAL INFORMATIK,* vol. 18, no. 1, pp. 69-79, 2022.

[9] P. T. Rahayu, Daryanto dan Q. A'yun, "Perbandingan Algoritma K-Nearest Neighbor Dan Gaussian Naïve Bayes Pada Klsifikai Penyakit Diabetes Melitus," *Jurnal Smart Teknologi,* vol. III, no. 4, p. 366 – 373, 2022.

Halaman ini sengaja dibiarkan kosong