

ANALISIS PERBANDINGAN ALGORITMA *K-NEAREST NEIGHBOR* (KNN) DAN *SUPPORT VECTOR MACHINE* (SVM) DALAM KLASIFIKASI KUALITAS SUSU

Shareen Stephanie

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
E-mail: shareen.535220112@stu.untar.ac.id

ABSTRAK

Analisis ini bertujuan untuk membandingkan kinerja dua algoritma klasifikasi, yaitu *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM) dalam konteks pengklasifikasian kualitas susu. Data untuk analisis ini diambil dari kumpulan dataset “Milk Quality” yang tersedia di platform *Kaggle*. Dengan menggunakan dataset tersebut, evaluasi dilakukan terhadap nilai *precision*, *recall*, *F1-Score*, dan akurasi dari kedua algoritma untuk mencari algoritma yang memiliki performa terbaik. Hasil menunjukkan bahwa algoritma KNN secara konsisten menunjukkan performa yang lebih unggul dibandingkan dengan SVM.

Kata kunci—Klasifikasi, Kualitas Susu, *K-Nearest Neighbor*, *Support Vector Machine*

ABSTRACT

This analysis aims to compare the performance of two classification algorithms, namely K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) in the context of milk quality classification. This analysis uses data from the “Milk Quality” dataset available on the Kaggle platform. To identify the best performing algorithm, the dataset was used to evaluate two algorithms’ precision, recall, F1-Score, and accuracy values. The KNN algorithm consistently outperforms SVM in terms of performance, as evidenced by the results.

Keywords—Classification, Milk Quality, *K-Nearest Neighbor*, *Support Vector Machine*

1. PENDAHULUAN

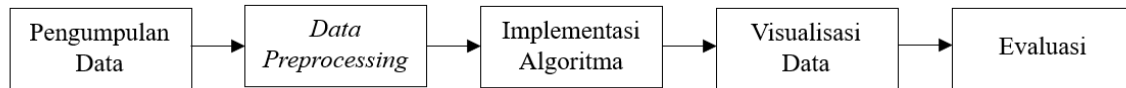
Susu merupakan salah satu opsi yang sering dipilih oleh manusia dalam memenuhi sumber nutrisi penting yang dibutuhkan oleh tubuh.[1] Dikenal dengan kandungan nutrisinya yang kaya, susu adalah sumber utama protein, kalsium, vitamin D, dan berbagai nutrisi esensial lainnya yang dibutuhkan untuk pertumbuhan, perkembangan, dan pemeliharaan kesehatan tubuh manusia. Susu sendiri dapat diperoleh dari beberapa sumber, baik hewani seperti sapi, kambing dan kerbau ataupun nabati seperti kedelai, oat dan almond.

Namun, di sisi lain, terdapat permasalahan yang timbul diluar dari manfaat dari susu tersebut, yaitu kualitas susu yang dikonsumsi. Kualitas susu dapat dipengaruhi oleh beberapa faktor, salah satunya adalah penyimpanan dari susu yang akan dikonsumsi.[2] Susu tidak dapat disimpan terlalu lama mengingat bakteri dan mikroorganisme yang ada pada susu juga membutuhkan nutrisi yang ada dalam susu untuk bakteri dan mikroorganisme tersebut berkembang.[3] Kontaminasi oleh bakteri dan mikroorganisme yang ada pada susu berlangsung dengan sangat cepat, sehingga akan mengurangi kualitas dari susu dan susu menjadi tidak layak untuk dikonsumsi karena dapat membahayakan tubuh manusia alih-alih menjadi sumber nutrisi bagi tubuh.[3]

Analisis ini bertujuan untuk mengevaluasi performa dua algoritma, yaitu *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM) dalam membedakan jenis kualitas susu dan mencari tahu algoritma mana yang memberikan kinerja terbaik dalam membedakan jenis kualitas susu. Dengan melakukan analisis ini, diharapkan dapat menjadi landasan untuk pengembangan sistem pengklasifikasi yang lebih baik dan akurat dalam memantau kualitas susu.

2. METODE PENELITIAN

Analisis ini menggunakan dua algoritma klasifikasi utama, yaitu *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM). Proses analisis ini dibagi menjadi lima tahapan, yaitu pengumpulan data, *data preprocessing*, implementasi algoritma klasifikasi, visualisasi data, dan evaluasi.



Gambar 1 Alur Tahapan Analisis

2.1 Pengumpulan Data

Data yang digunakan dalam analisis ini merupakan *dataset* yang bersumber dari *Kaggle.com* ("*Milk Quality*"). *Dataset* tersebut memiliki 1059 baris (*entries*) dan 8 kolom (*variable*) yang berisi pH, *Temprature*, *Taste*, *Odor*, *Fat*, *Turbidity*, *Colour*, dan *Grade*.

	pH	Temprature	Taste	Odor	Fat	Turbidity	Colour	Grade
0	6.6	35	1	0	1	0	254	high
1	6.6	36	0	1	0	1	253	high
2	8.5	70	1	1	1	1	246	low
3	9.5	34	1	1	0	1	255	low
4	6.6	37	0	0	0	0	255	medium
...	...	...	...	...	...	...	...	...
1054	6.7	45	1	1	0	0	247	medium
1055	6.7	38	1	0	1	0	255	high
1056	3.0	40	1	1	1	1	255	low
1057	6.8	43	1	0	1	0	250	high
1058	8.6	55	0	1	1	1	255	low

1059 rows × 8 columns

Gambar 2 *Dataset "Milk Quality"*

2.2 Data Preprocessing

Pada tahap awal bagian ini, *dataset* dilakukan pemeriksaan apakah terdapat nilai yang hilang (*missing values*) di dalamnya. Langkah ini penting untuk memastikan kualitas data sebelum dilanjutkan ke tahapan berikutnya. *Missing values* dapat dilihat melalui kolom "*Non-Null Count*". Jika setiap kolom memiliki jumlah *non-null* yang sama dengan jumlah total *entries dataset*, maka di dalam *dataset* tersebut tidak terdapat *missing values*. [4][5]

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1059 entries, 0 to 1058
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0    pH          1059 non-null  float64
 1  Temprature  1059 non-null  int64  
 2    Taste       1059 non-null  int64  
 3    Odor         1059 non-null  int64  
 4    Fat          1059 non-null  int64  
 5  Turbidity   1059 non-null  int64  
 6  Colour      1059 non-null  int64  
 7    Grade       1059 non-null  object 
dtypes: float64(1), int64(6), object(1)
memory usage: 66.3+ KB
  
```

Gambar 3 Pemeriksaan *Missing Values*

Setelah tahap penanganan *missing values*, tahap selanjutnya adalah dengan melakukan *encoding label*. Dengan menggunakan *encoding label*, nilai-nilai yang ada di kolom yang bertipe *object* dikodekan menjadi bilangan bulat.[5] Hal ini dimaksudkan agar data dapat digunakan dalam proses algoritma yang akan digunakan. Pada analisis ini, kolom yang dilakukan *encoding label* adalah kolom *Grade*.

	pH	Temperature	Taste	Odor	Fat	Turbidity	Colour	Grade
0	6.6	35	1	0	1	0	254	0
1	6.6	36	0	1	0	1	253	0
2	8.5	70	1	1	1	1	246	1
3	9.5	34	1	1	0	1	255	1
4	6.6	37	0	0	0	0	255	2
...	...	...	...	...	...	...	...	...
1054	6.7	45	1	1	0	0	247	2
1055	6.7	38	1	0	1	0	255	0
1056	3.0	40	1	1	1	1	255	1
1057	6.8	43	1	0	1	0	250	0
1058	8.6	55	0	1	1	1	255	1

1059 rows × 8 columns

Gambar 4 Dataset Setelah Dilakukan *Encoding Label*

Tahap selanjutnya dari bagian ini adalah standarisasi data. Standarisasi data sangat penting dalam metode klasifikasi khususnya saat digunakan dalam algoritma KNN dan algoritma SVM. Hal ini dikarenakan dengan melakukan standarisasi data, kedua algoritma tersebut dipastikan dahulu supaya dapat bekerja secara optimal.[6]

Kemudian tahap terakhir dari bagian *data preprocessing* adalah pemisahan data. *Dataset* dibagi menjadi data uji dan data latih. Data latih akan digunakan untuk melatih model dan data uji akan digunakan untuk menguji kinerja model yang dihasilkan.[7][8][9]

2.3 Algoritma KNN

Algoritma *K-Nearest Neighbor* (KNN) merupakan sebuah algoritma klasifikasi yang paling sederhana dibandingkan dengan algoritma *supervised learning* lainnya. KNN bekerja dengan cara menggunakan kemiripan nilai data-data yang ada dalam jarak *k* sebagai dasar untuk klasifikasi.[10][11]

$$d_1 = \sqrt{\sum_{i=1}^p (x_1 - x_2)^2} \quad (1)$$

Persamaan (1) di atas merupakan representasi matematis yang digunakan untuk menghitung jarak antara dua titik pada algoritma KNN dengan menggunakan metode *euclidean*. Pada persamaan di atas digunakan variabel *d* sebagai jarak, variabel *i* sebagai variabel data, variabel *p* sebagai dimensi data, variabel *x₁* sebagai sampel data dan variabel *x₂* sebagai data uji.[11] Persamaan ini menentukan seberapa dekat atau seberapa jauh dua titik berada dalam ruang fitur dengan mengukur panjang garis lurus antara keduanya.[11][12] Dengan menggunakan metode *euclidean*, persamaan tersebut memperhitungkan perbedaan antara nilai-nilai fitur dari kedua titik, sehingga menciptakan dasar untuk menentukan kemiripan antara keduanya.[13]

2.4 Algoritma SVM

Algoritma *Support Vector Machine* (SVM) merupakan salah satu algoritma klasifikasi yang bekerja dengan cara mencari *hyperplane* atau garis pemisah data yang optimal dalam ruang fitur untuk memisahkan kelas-kelas data yang berbeda.[14][15] SVM terdiri atas dua jenis, yaitu SVM *Linear* dan SVM *Non-linear*. [16] Analisis ini menggunakan SVM jenis *linear* untuk mengklasifikasikan data kualitas susu.

$$f(x) = w \cdot x + b \quad \text{atau} \quad f(x) = \sum_{i=1}^m \alpha_i y_i K(x, x_i) + b \quad (2)$$

$$K(x, x_i) = x x_i^T \quad (3)$$

Persamaan (2) merupakan rumus dari algoritma SVM, dimana $f(x)$ merupakan fungsi keputusan klasifikasi, w merupakan vektor bobot yang menggambarkan arah *hyperplane*, b merupakan bias yang menentukan posisi *hyperplane* terhadap asal koordinat, m merupakan jumlah *support vector data* yang memiliki $\alpha_i > 0$, α_i merupakan nilai bobot setiap titik data, dan $K(x, x_i)$ merupakan fungsi kernel.[14][17][18] Algoritma SVM memiliki beberapa fungsi kernel, salah satunya yang digunakan pada analisis ini yaitu fungsi kernel linear.[14] Dengan demikian, untuk menghitung nilai $K(x, x_i)$ seperti yang dijelaskan dalam persamaan (2), dapat dilakukan dengan menggunakan persamaan (3).[19][20]

2.5 Visualisasi Data

Pada analisis ini, hasil implementasi kedua algoritma, *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM), disajikan secara rinci dalam bentuk tabel yang memudahkan untuk membandingkan kinerja kedua algoritma tersebut.

2.6 Evaluasi

Setelah mengimplementasikan kedua algoritma di atas, langkah terakhir dari metode analisis kualitas susu ini adalah dengan mengevaluasi hasil dari kedua algoritma tersebut. Evaluasi dilakukan dengan membandingkan nilai dari *precision*, *recall*, *F1-score*, dan akurasi dari setiap skenario yang dijalankan menggunakan kedua algoritma. Kemudian, dilakukan pencarian untuk menemukan nilai terbaik di antara semua skenario yang telah dievaluasi.

3. HASIL DAN PEMBAHASAN

Pada Gambar 2, dapat dilihat bahwa di dalam *dataset* tersebut terdapat beberapa parameter dengan karakteristik yang berbeda. Parameter *Taste*, *Odor*, *Fat*, dan *Turbidity* memiliki rentang nilai antara 0 dan 1. Nilai-nilai ini memiliki interpretasi khusus, yaitu jika parameter memenuhi kondisi optimal, nilainya akan 1, dan jika tidak, nilainya akan 0. Sementara itu, parameter pH dan *Temperature* memiliki nilai aktual yang mencerminkan pengukuran aktual. Selanjutnya, nilai untuk parameter *Grade* terdapat 3 kategori, yaitu *Low*, *Medium*, dan *High*. Dalam proses klasifikasi, model memerlukan data numerik, oleh karena itu dilakukan proses *data preprocessing*. Dalam proses ini, nilai-nilai parameter *Grade* diubah menjadi bilangan bulat menggunakan teknik *encoding label*, seperti yang terlihat pada Gambar 4. Dalam representasi ini, angka 0 menunjukkan kategori *High*, angka 1 menunjukkan kategori *Low*, dan angka 2 menunjukkan kategori *Medium*.

3.1 Hasil Implementasi Algoritma KNN

Implementasi algoritma KNN dilakukan melalui serangkaian skenario yang berbeda, termasuk variasi nilai n yaitu 3, 5, dan 7, serta berbagai jenis data uji dan data latih yang digunakan untuk evaluasi dan pengujian.

3.1.1 Hasil Implementasi Algoritma KNN dengan k bernilai 3

Tabel 1 Hasil Algoritma KNN dengan k Bernilai 3 Menggunakan 60% Data Latih dan 40% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.99	1.00	1.00	1.00
2	0.99	0.99	0.99	0.99
3	0.98	0.99	0.98	0.98
4	0.99	0.99	0.99	0.99
5	1.00	1.00	1.00	1.00
Rata-rata	0.99	0.99	0.99	0.99

Tabel 2 Hasil Algoritma KNN dengan k Bernilai 3 Menggunakan 70% Data Latih dan 30% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.99	0.99	0.99	0.99
2	0.99	0.99	0.99	0.99
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	0.99	0.99	0.99	0.99
Rata-rata	0.99	0.99	0.99	0.99

Tabel 3 Hasil Algoritma KNN dengan k Bernilai 3 Menggunakan 80% Data Latih dan 20% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.99	1.00	0.99	1.00
2	1.00	1.00	1.00	1.00
3	0.99	0.99	0.99	0.99
4	0.99	0.99	0.99	0.99
5	1.00	1.00	1.00	1.00
Rata-rata	0.99	0.99	0.99	0.99

Tabel 4 Hasil Algoritma KNN dengan k Bernilai 3 Menggunakan 90% Data Latih dan 10% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.99	0.99	0.99	0.99
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	0.99	0.99	0.99	0.99
Rata-rata	0.99	0.99	0.99	0.99

Tabel 5 Rata-Rata Seluruh Hasil Implementasi Algoritma KNN dengan k Bernilai 3

Data Latih	Data Uji	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
60%	40%	0.99	0.99	0.99	0.99
70%	30%	0.99	0.99	0.99	0.99
80%	20%	0.99	0.99	0.99	0.99
90%	10%	0.99	0.99	0.99	0.99
Rata-rata		0.99	0.99	0.99	0.99

3.1.2 Hasil Implementasi Algoritma KNN dengan k bernilai 5

Tabel 6 Hasil Algoritma KNN dengan k Bernilai 5 Menggunakan 60% Data Latih dan 40% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.98	0.98	0.98	0.98
2	1.00	0.99	1.00	1.00
3	0.99	0.99	0.99	0.99
4	0.98	0.98	0.98	0.98
5	0.98	0.98	0.98	0.98
Rata-rata	0.99	0.98	0.99	0.99

Tabel 7 Hasil Algoritma KNN dengan k Bernilai 5 Menggunakan 70% Data Latih dan 30% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.97	0.98	0.97	0.97
2	1.00	1.00	1.00	1.00
3	0.99	0.99	0.99	0.99
4	0.99	0.99	0.99	0.99
5	0.99	1.00	0.99	0.99
Rata-rata	0.99	0.99	0.99	0.99

Tabel 8 Hasil Algoritma KNN dengan k Bernilai 5 Menggunakan 80% Data Latih dan 20% Data Uji

Eksperimen ke	Precision	Recall	F1-Score	Akurasi
1	0.98	0.98	0.98	0.98
2	0.99	1.00	1.00	1.00
3	0.98	0.99	0.99	0.99
4	1.00	1.00	1.00	1.00
5	0.99	0.99	0.99	0.99
Rata-rata	0.99	0.99	0.99	0.99

Tabel 9 Hasil Algoritma KNN dengan k Bernilai 5 Menggunakan 90% Data Latih dan 10% Data Uji

Eksperimen ke	Precision	Recall	F1-Score	Akurasi
1	1.00	1.00	1.00	1.00
2	0.98	0.98	0.98	0.98
3	1.00	1.00	1.00	1.00
4	0.99	0.99	0.99	0.99
5	0.97	0.97	0.97	0.97
Rata-rata	0.99	0.99	0.99	0.99

Tabel 10 Rata-Rata Seluruh Hasil Implementasi Algoritma KNN dengan k Bernilai 5

Data Latih	Data Uji	Precision	Recall	F1-Score	Akurasi
60%	40%	0.99	0.98	0.99	0.99
70%	30%	0.99	0.99	0.99	0.99
80%	20%	0.99	0.99	0.99	0.99
90%	10%	0.99	0.99	0.99	0.99
Rata-rata		0.99	0.99	0.99	0.99

3.1.3 Hasil Implementasi Algoritma KNN dengan k bernilai 7

Tabel 11 Hasil Algoritma KNN dengan k Bernilai 7 Menggunakan 60% Data Latih dan 40% Data Uji

Eksperimen ke	Precision	Recall	F1-Score	Akurasi
1	0.99	0.99	0.99	0.98
2	0.98	0.98	0.98	0.98
3	0.99	0.99	0.99	0.99
4	0.97	0.97	0.97	0.97
5	0.97	0.97	0.97	0.97
Rata-rata	0.98	0.98	0.98	0.98

Tabel 12 Hasil Algoritma KNN dengan k Bernilai 7 Menggunakan 70% Data Latih dan 30% Data Uji

Eksperimen ke	Precision	Recall	F1-Score	Akurasi
1	0.98	0.97	0.97	0.97
2	0.99	0.99	0.99	0.99
3	0.98	0.98	0.98	0.98
4	0.98	0.98	0.98	0.97
5	0.99	0.99	0.99	0.99
Rata-rata	0.98	0.98	0.98	0.98

Tabel 13 Hasil Algoritma KNN dengan k Bernilai 7 Menggunakan 80% Data Latih dan 20% Data Uji

Eksperimen ke	Precision	Recall	F1-Score	Akurasi
1	0.98	0.97	0.97	0.97
2	0.99	0.99	0.99	0.99
3	0.98	0.98	0.98	0.98
4	0.98	0.98	0.98	0.97
5	0.99	0.99	0.99	0.99
Rata-rata	0.98	0.98	0.98	0.98

Tabel 14 Hasil Algoritma KNN dengan k Bernilai 7 Menggunakan 90% Data Latih dan 10% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
3	0.98	0.99	0.98	0.98
4	1.00	1.00	1.00	1.00
5	0.99	0.99	0.99	0.99
Rata-rata	0.99	0.99	0.99	0.99

Tabel 15 Rata-Rata Seluruh Hasil Implementasi Algoritma KNN dengan k Bernilai 7

Data Latih	Data Uji	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
60%	40%	0.98	0.98	0.98	0.98
70%	30%	0.98	0.98	0.98	0.98
80%	20%	0.98	0.98	0.98	0.98
90%	10%	0.99	0.99	0.99	0.99
Rata-rata		0.98	0.98	0.98	0.98

3.2 Hasil Implementasi Algoritma SVM

Implementasi algoritma SVM dilakukan melalui serangkaian skenario yang berbeda, termasuk variasi nilai C yaitu 0.1, 1, dan 10, serta berbagai jenis data uji dan data latih yang digunakan untuk evaluasi dan pengujian.

3.2.1 Hasil Implementasi Algoritma SVM dengan C bernilai 0.1

Tabel 16 Hasil Algoritma SVM dengan C Bernilai 0.1 Menggunakan 60% Data Latih dan 40% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.77	0.76	0.76	0.77
2	0.85	0.86	0.85	0.85
3	0.81	0.82	0.80	0.82
4	0.81	0.82	0.81	0.82
5	0.74	0.73	0.73	0.75
Rata-rata	0.80	0.80	0.79	0.80

Tabel 17 Hasil Algoritma SVM dengan C Bernilai 0.1 Menggunakan 70% Data Latih dan 30% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.75	0.74	0.73	0.74
2	0.76	0.78	0.76	0.77
3	0.74	0.73	0.73	0.75
4	0.80	0.80	0.79	0.80
5	0.85	0.85	0.85	0.86
Rata-rata	0.78	0.78	0.77	0.78

Tabel 18 Hasil Algoritma SVM dengan C Bernilai 0.1 Menggunakan 80% Data Latih dan 20% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.84	0.85	0.85	0.85
2	0.71	0.71	0.71	0.72
3	0.87	0.88	0.86	0.87
4	0.82	0.84	0.82	0.83
5	0.76	0.75	0.75	0.77
Rata-rata	0.80	0.81	0.80	0.81

Tabel 19 Hasil Algoritma SVM dengan C Bernilai 0.1 Menggunakan 90% Data Latih dan 10% Data Uji

Eksperimen ke	Precision	Recall	F1-Score	Akurasi
1	0.85	0.85	0.84	0.84
2	0.77	0.78	0.77	0.79
3	0.87	0.88	0.86	0.87
4	0.85	0.86	0.84	0.84
5	0.86	0.88	0.86	0.86
Rata-rata	0.84	0.85	0.83	0.84

Tabel 20 Rata-Rata Seluruh Hasil Implementasi Algoritma SVM dengan C Bernilai 0.1

Data Latih	Data Uji	Precision	Recall	F1-Score	Akurasi
60%	40%	0.80	0.80	0.79	0.80
70%	30%	0.78	0.78	0.77	0.78
80%	20%	0.80	0.81	0.80	0.81
90%	10%	0.84	0.85	0.83	0.84
Rata-rata		0.81	0.81	0.80	0.81

3.2.2 Hasil Implementasi Algoritma SVM dengan C bernilai 1

Tabel 21 Hasil Algoritma SVM dengan C Bernilai 1 Menggunakan 60% Data Latih dan 40% Data Uji

Eksperimen ke	Precision	Recall	F1-Score	Akurasi
1	0.85	0.85	0.85	0.85
2	0.84	0.86	0.84	0.84
3	0.85	0.85	0.85	0.85
4	0.86	0.87	0.85	0.86
5	0.85	0.87	0.85	0.85
Rata-rata	0.85	0.86	0.85	0.85

Tabel 22 Hasil Algoritma SVM dengan C Bernilai 1 Menggunakan 70% Data Latih dan 30% Data Uji

Eksperimen ke	Precision	Recall	F1-Score	Akurasi
1	0.84	0.86	0.84	0.85
2	0.84	0.86	0.85	0.85
3	0.84	0.84	0.83	0.83
4	0.82	0.82	0.82	0.83
5	0.85	0.86	0.85	0.86
Rata-rata	0.84	0.85	0.84	0.84

Tabel 23 Hasil Algoritma SVM dengan C Bernilai 1 Menggunakan 80% Data Latih dan 20% Data Uji

Eksperimen ke	Precision	Recall	F1-Score	Akurasi
1	0.83	0.83	0.83	0.83
2	0.87	0.90	0.88	0.89
3	0.82	0.81	0.81	0.81
4	0.86	0.85	0.85	0.85
5	0.82	0.82	0.81	0.82
Rata-rata	0.84	0.84	0.84	0.84

Tabel 24 Hasil Algoritma SVM dengan C Bernilai 1 Menggunakan 90% Data Latih dan 10% Data Uji

Eksperimen ke	Precision	Recall	F1-Score	Akurasi
1	0.87	0.87	0.87	0.87
2	0.85	0.86	0.85	0.86
3	0.83	0.82	0.83	0.83
4	0.84	0.84	0.83	0.83
5	0.88	0.88	0.88	0.89
Rata-rata	0.85	0.85	0.85	0.86

Tabel 25 Rata-Rata Seluruh Hasil Implementasi Algoritma SVM dengan C Bernilai 1

Data Latih	Data Uji	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
60%	40%	0.85	0.86	0.85	0.85
70%	30%	0.84	0.85	0.84	0.84
80%	20%	0.84	0.84	0.84	0.84
90%	10%	0.85	0.85	0.85	0.86
Rata-rata		0.85	0.85	0.85	0.85

3.2.3 Hasil Implementasi Algoritma SVM dengan C bernilai 10

Tabel 26 Hasil Algoritma SVM dengan C Bernilai 10 Menggunakan 60% Data Latih dan 40% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.89	0.90	0.89	0.89
2	0.87	0.87	0.87	0.87
3	0.87	0.87	0.87	0.88
4	0.88	0.88	0.88	0.88
5	0.88	0.89	0.88	0.89
Rata-rata	0.88	0.88	0.88	0.88

Tabel 27 Hasil Algoritma SVM dengan C Bernilai 10 Menggunakan 70% Data Latih dan 30% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.89	0.90	0.89	0.89
2	0.88	0.89	0.89	0.89
3	0.90	0.91	0.90	0.91
4	0.86	0.88	0.87	0.87
5	0.91	0.92	0.91	0.91
Rata-rata	0.89	0.90	0.89	0.89

Tabel 28 Hasil Algoritma SVM dengan C Bernilai 10 Menggunakan 80% Data Latih dan 20% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.88	0.88	0.88	0.88
2	0.88	0.90	0.89	0.90
3	0.91	0.93	0.92	0.92
4	0.91	0.90	0.91	0.90
5	0.91	0.93	0.92	0.92
Rata-rata	0.90	0.91	0.90	0.90

Tabel 29 Hasil Algoritma SVM dengan C Bernilai 10 Menggunakan 90% Data Latih dan 10% Data Uji

Eksperimen ke	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
1	0.91	0.92	0.91	0.92
2	0.89	0.89	0.89	0.89
3	0.90	0.91	0.90	0.91
4	0.92	0.92	0.92	0.92
5	0.91	0.91	0.91	0.91
Rata-rata	0.91	0.91	0.91	0.91

Tabel 30 Rata-Rata Seluruh Hasil Implementasi Algoritma SVM dengan C Bernilai 10

Data Latih	Data Uji	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi
60%	40%	0.88	0.88	0.88	0.88
70%	30%	0.89	0.90	0.89	0.89
80%	20%	0.90	0.91	0.90	0.90
90%	10%	0.91	0.91	0.91	0.91
Rata-rata		0.90	0.90	0.90	0.90

3.3 Ringkuman Hasil Rata-Rata Implementasi Algoritma KNN dan SVM

Tabel 31 Hasil Rata-Rata Seluruh Implementasi Algoritma KNN

<i>k</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Akurasi</i>
3	0.99	0.99	0.99	0.99
5	0.99	0.99	0.99	0.99
7	0.98	0.98	0.98	0.98

Tabel 32 Hasil Rata-Rata Seluruh Implementasi Algoritma SVM

<i>C</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Akurasi</i>
0.1	0.81	0.81	0.80	0.81
1	0.85	0.85	0.85	0.85
10	0.90	0.90	0.90	0.90

3.4 Hasil Analisis

Berdasarkan Tabel 31 dan 32, dapat dilihat bahwa algoritma *K-Nearest Neighbors* (KNN) menunjukkan kinerja yang sangat baik dibandingkan algoritma *Support Vector Machine* (SVM). Hal ini terlihat dari nilai *precision*, *recall*, *F1-Score*, dan akurasi dari algoritma KNN yang konsisten tinggi untuk berbagai nilai *k* yang digunakan, yakni 3, 5, dan 7. Pada algoritma KNN dengan *k*=3, nilai *precision*, *recall*, *F1-score*, dan akurasinya berturut-turut adalah 0.99, 0.99, 0.99, 0.99. Pada algoritma KNN dengan *k*=5, nilai *precision*, *recall*, *F1-score*, dan akurasinya juga berturut-turut adalah 0.99, 0.99, 0.99, 0.99. Dan pada algoritma KNN dengan *k*=7, nilai *precision*, *recall*, *F1-score*, dan akurasinya berturut-turut adalah 0.98, 0.98, 0.98, 0.98.

Di sisi lain, algoritma SVM juga menunjukkan kinerja yang memuaskan, dengan peningkatan performa yang signifikan seiring dengan peningkatan nilai parameter *C*. algoritma SVM dengan *C*=10 memiliki nilai *precision*, *recall*, *F1-Score*, dan akurasi yang lebih tinggi dibandingkan dengan SVM dengan *C* yang lebih rendah. Hal tersebut terlihat pada saat algoritma SVM dengan *C*=0.1, nilai *precision*, *recall*, *F1-Score*, dan akurasinya berturut-turut adalah 0.81, 0.81, 0.80, 0.81. Pada algoritma SVM dengan *C*=1, nilai *precision*, *recall*, *F1-Score*, dan akurasinya berturut-turut adalah 0.85, 0.85, 0.85, 0.85. Sedangkan pada algoritma SVM dengan *C*=10, nilai *precision*, *recall*, *F1-Score*, dan akurasinya berturut-turut adalah 0.90, 0.90, 0.90, 0.90.

4. KESIMPULAN

Kesimpulan dari hasil pengujian dan analisis yang dilakukan adalah sebagai berikut:

1. Algoritma *K-Nearest Neighbor* (KNN) dan algoritma *Support Vector Machine* (SVM) dapat diimplementasikan dengan baik, langkah pertama yaitu pengumpulan *dataset* “*Milk Quality*” dengan 8 variabel yang meliputi pH, *Temprature*, *Taste*, *Odor*, *Fat*, *Turbidity*, *Colour*, dan *Grade*. Kemudian data tersebut dilakukan *data preprocessing* dengan berbagai tahapan, yaitu pemeriksaan *missing values*, melakukan *encoding label*, melakukan standarisasi data, dan pembagian data latih dan data uji. Setelah proses *data preprocessing*, dilakukan implementasi kedua algoritma klasifikasi tersebut dan dilanjutkan dengan memvisualisasikan hasil eksperimen ke dalam tabel dan mengevaluasinya dengan membandingkan nilai *precision*, *recall*, *F1-score*, dan akurasi pada setiap skenario.
2. Algoritma KNN memiliki performa yang lebih baik dibandingkan dengan algoritma SVM pada *dataset* “*Milk Quality*”. Hal tersebut terlihat pada algoritma KNN secara konsisten menunjukkan nilai *precision*, *recall*, *F1-score*, dan akurasi yang lebih tinggi daripada algoritma SVM.

Saran penulis untuk pengembangan analisis perbandingan algoritma *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM) dalam klasifikasi kualitas susu kedepannya adalah

melakukan lebih banyak eksperimen pada kedua algoritma tersebut dengan variasi skenario dan nilai parameter yang lebih luas. Dengan melibatkan lebih banyak eksperimen, akan memungkinkan analisis yang lebih mendalam tentang implementasi kedua algoritma klasifikasi tersebut dalam proses pengklasifikasian kualitas susu.

DAFTAR PUSTAKA

- [1] F. Adiba, A. A. N. Risal and M. Tahir, "Implementasi Algoritma Backpropagation untuk Klasifikasi Kualitas Susu Sapi," *Jurnal MediaTIK: Jurnal Media Pendidikan Teknik Informatika dan Komputer* Vol.6 No.2, pp. 42-45, Mei 2023.
- [2] H. RESNAWATI, "KUALITAS SUSU PADA BERBAGAI PENGOLAHAN," *Semiloka Nasional Prospek Industri Sapi Perah Menuju Perdagangan Bebas*, pp. 497-502, 2020.
- [3] N. Sholikah, A. A. Mufid., A. S. Bachrul., T. R. Hidayat and Y. Yoga, "PENGOLAHAN SUSU SAPI MENJADI SUSU PASTEURISASI UNTUK MENINGKATKAN NILAI SUSU DAN DAYA JUAL," *JURNAL PEMBELAJARAN PEMBERDAYAAN MASYARAKAT* Vol. 2 No. 1, pp. 75-79, 2021.
- [4] A. T. Akbar, N. Yudistira and A. Ridok, "IDENTIFIKASI GAGAL GINJAL KRONIS DENGAN MENGIMPLEMENTASIKAN METODE SUPPORT VECTOR MACHINE BESERTA K-NEAREST NEIGHBOUR (SVM-KNN)," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)* Vol. 10, No. 2, pp. 301-308, April 2023.
- [5] S. S. Berutu, H. Budiati, J. and F. Gulo, "*Data preprocessing approach for machine learning-based sentiment classification*," *JURNAL INFOTEL*, pp. 317-325, 2023.
- [6] M. R. Firmansyah and Y. P. Astuti, "Stroke Classification Comparison with KNN through Standardization and Normalization Techniques," *Advance Sustainable Science, Engineering and Technology (ASSET)* Vol. 6, No.1, pp. 02401012-01 ~ 02401012-08, January 2024.
- [7] B. K. W., H. Prasetyo, G. A. L., B. A. Wibisono and D. S. Prasvita, "Analisis Perbandingan Algoritma SVM dan CNN untuk Klasifikasi Buah," September 2021.
- [8] R. A. Pradana, I. Cholissodin and N. Hidayat, "Klasifikasi Mutu Susu Sapi menggunakan Metode Modified K-Nearest Neighbor (MKNN)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* Vol. 6, No. 3, pp. 1360-1367, Maret 2022.
- [9] N. Wakhidah and S. N. Rochmah, "Klasifikasi kualitas mutu susu pasteurisasi menggunakan metode klasifikasi K-Nearest Neighbor," *AITI: Jurnal Teknologi Informasi*, Volume 21 No. 1, pp. 58-71, Maret 2024.
- [10] E. Y. Boateng, J. Otoo and D. A. Abaye, "*Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review*," *Journal of Data Analysis and Information Processing*, pp. 341-357, 2020.
- [11] S. R. Cholil, T. Handayani, R. Prathivi and T. Ardianita, "Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa," *IJCIT (Indonesian Journal on Computer and Information Technology)*, pp. 118-127, 2021.
- [12] A. S. Naufal, A. and W. Astuti, "Analisis Perbandingan Klasifikasi Support Vector Machine (SVM) dan K-Nearest Neighbors (KNN) untuk Deteksi Kanker dengan Data Microarray," *JURIKOM (Jurnal Riset Komputer)*, pp. 162-168, Februari 2020.
- [13] S. Zhang, "Challenges in KNN Classification," *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, VOL. 34, NO. 10, Oktober 2022.
- [14] L. A. Demidova, "*Two-Stage Hybrid Data Classifiers Based on SVM and kNN Algorithms*," 2021.
- [15] R. Nanda, E. Haerani, S. K. Gusti and S. Ramadhani, "Klasifikasi Berita Menggunakan Metode Support Vector Machine," *Jurnal Nasional Komputasi dan Teknologi Informasi* Vol. 5 No. 2, pp. 269-278, April 2022.
- [16] S. Ghosh, A. Dasgupta and A. Swetapadma, "A Study on Support Vector Machine based Linear and Non-Linear Pattern Classification," *International Conference on Intelligent Sustainable Systems*, pp. 24-28, 2019.
- [17] R. A. Wati, H. Irsyad and M. E. A. Rivan, "Klasifikasi Pneumonia Menggunakan Metode Support Vector Machine," *Jurnal Algoritme* Vol. 1, No. 1, pp. 21-32, Oktober 2020.
- [18] Z. Soumaya, B. D. Taoufiq, N. Benayad, K. Yunus and. A. Abdelkrim, "*The detection of Parkinson disease using the genetic algorithm and SVM classifier*," 2021.

- [19] N. G. Ramadhan and A. Khoirunnisa, "Klasifikasi Data Malaria Menggunakan Metode Support Vector Machine," JURNAL MEDIA INFORMATIKA BUDIDARMA, Vol. 5 No. 4, pp. 1580-1584, Oktober 2021.
- [20] J. Cervantes, F. G. Lamont, L. R. Mazahua and A. Lopez, "*A comprehensive survey on support vector machine classification: Applications, challenges and trends*," 2019.