

KLASIFIKASI KUALITAS APEL DENGAN PERBANDINGAN MENGUNAKAN ALGORITMA ANN DAN SVM PADA *DATASET APPLE QUALITY*

Naswa Azahra

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
E-mail: naswa.535220252@stu.untar.ac.id

ABSTRAK

Kualitas apel adalah faktor penting dalam industri pertanian dan distribusi makanan, mempengaruhi data saing produk dan kepuasan konsumen. Metode klasifikasi seperti *Artificial Neural Network* (ANN) dan *Support Vector Machine* (SVM) digunakan untuk memprediksi kualitas apel secara otomatis, meningkatkan efisiensi penilaian kualitas. Penelitian ini membandingkan performa ANN dan SVM dalam klasifikasi kualitas apel menggunakan dataset yang mencakup atribut fisik dan kategori kualitas. ANN menggunakan *feedforward neural network* dengan *backpropagation*, sementara SVM menggunakan kernel *Linear*, *Radial Basis Function* (RBF) dan Polinomial dengan optimasi *hyperparameter*. Dengan menguji kedua metode pada dataset yang sama, penelitian ini memberikan gambaran tentang efisiensi dan kinerja masing-masing metode, serta memberikan gambaran dasar untuk pemilihan metode klasifikasi yang sesuai dengan dataset apel. Hasilnya dapat membantu petani dan produsen dalam memprediksi kualitas apel dengan akurat, meminimalkan kerugian dan meningkatkan nilai tambah produk.

Kata kunci: Klasifikasi Apel, ANN, SVM, Kualitas Apel, Pertanian

ABSTRACT

Apple quality is an important factor in the agricultural and food distribution industries, influencing product sales data and consumer satisfaction. Classification methods such as *Artificial Neural Network* (ANN) and *Support Vector Machine* (SVM) are used to predict apple quality automatically, increasing the efficiency of quality assessment. This research compares the performance of ANN and SVM in apple quality classification using a dataset that includes physical attributes and quality categories. ANN uses a *feedforward neural network* with *backpropagation*, while SVM uses *Linear*, *Radial Basis Function* (RBF) and *Polynomial* kernels with *hyperparameter* optimization. By testing both methods on the same dataset, this research provides an overview of the efficiency and performance of each method, as well as providing a basic overview for selecting a classification method that is suitable for the apple dataset. The results can help farmers and producers accurately predict apple quality, minimize losses and increase product added value.

Keywords: Apple Classification, ANN, SVM, Apple Quality, Agriculture

1. PENDAHULUAN

Apel merupakan salah satu jenis buah yang sangat digemari banyak kalangan usia, yang memiliki kandungan seperti tinggi serat, vitamin c, dan berbagai macam antioksidan [1] [2]. Dalam industri pertanian dan distribusi makanan, penilaian kualitas produk menjadi kunci untuk memastikan standar yang tinggi dan kepuasan konsumen. Salah satu produk yang sering dinilai kualitasnya adalah apel. Kualitas apel merupakan faktor yang penting dalam memperkuat daya saing produk pertanian. Klasifikasi kualitas apel dapat dilakukan dengan menggunakan berbagai macam metode, salah satunya adalah dengan menggunakan metode klasifikasi *Artificial Neural Network* (ANN) dan *Support Vector Machine* (SVM) [3]. Dalam meningkatkan efisiensi penilaian kualitas apel, metode *machine learning* populer karena dapat menghasilkan model prediktif untuk mengklasifikasikan kualitas secara otomatis. Ini memungkinkan pembelajaran pola kompleks dari data fisik apel yang meningkatkan akurasi evaluasi. Dalam penelitian ini, akan dilakukan

perbandingan antara hasil klasifikasi menggunakan klasifikasi *Artificial Neural Network* (ANN) dan *Support Vector Machine* (SVM) [4].

Tujuan dari penelitian ini adalah untuk mengetahui performansi dari kedua metode algoritma tersebut dalam melakukan klasifikasi kualitas apel dan melakukan perbandingan antara kedua algoritma tersebut dalam melakukan klasifikasi kualitas apel. Data yang digunakan untuk penelitian ini berasal dari dataset yang disediakan oleh Nidula Elgiryewithana. Dataset ini mencakup beberapa atribut fisik apel, seperti ukuran, berat, manis, renyah, segar, matang, asam, dan kualitas. Kategori kualitas terdiri dari *Good* dan *Bad*. Dengan menggunakan dataset ini, penelitian bertujuan untuk mengembangkan model klasifikasi yang dapat dengan akurat memprediksi kualitas apel [5].

Dengan melakukan klasifikasi kualitas apel menggunakan metode klasifikasi ANN, akan menggunakan algoritma *feedforward neural network* dengan satu *hidden layer*. Dan menggunakan algoritma *backpropagation* untuk melakukan *training neural network* [6]. Data akan dibagi menjadi dua bagian, yaitu data *training* dan data *testing*. Data *training* akan digunakan untuk melatih ANN, sedangkan data *testing* akan digunakan untuk menguji akurasi klasifikasi ANN. Sedangkan dalam melakukan klasifikasi kualitas apel menggunakan metode klasifikasi SVM, akan menggunakan kernel linear, Radial Basis *Function* (RBF), dan polinomial. Dan menggunakan optimasi *hyperparameter* pada SVM. Data akan dibagi menjadi dua bagian, yaitu data *training* dan data *testing*. Data *training* akan digunakan untuk melatih SVM, sedangkan data *testing* akan digunakan untuk menguji akurasi klasifikasi SVM [7].

Mengujinya dengan menggunakan dua metode klasifikasi, yaitu *Artificial Neural Network* (ANN) dan *Support Vector Machine* (SVM), akan menghasilkan gambaran yang lebih tentang efisiensi dan kinerja dari setiap metode. Selain itu, perbandingan antara kedua metode algoritma ini dapat menjadi dasar dalam pemilihan metode klasifikasi yang paling tepat untuk digunakan pada dataset *apple quality* [8]. Hal ini dapat membantu para petani dan produsen dalam memprediksi kualitas apel dengan cepat dan tepat, sehingga dapat meminimalkan kehilangan dan meningkatkan nilai tambah produk [9].

2. METODE PENELITIAN

Penelitian ini akan menggunakan dataset dari *platform* Kaggle, khususnya dataset Apple Quality yang dapat diakses melalui tautan berikut:

<https://www.kaggle.com/datasets/nelgiryewithana/apple-quality>.

Jumlah dataset *apple quality* ini terdapat 4000, yang terdiri dari 9 kolom yaitu: *A_id*, *Size*, *Weight*, *Sweetness*, *Crunchiness*, *Juiciness*, *Ripeness*, *Acidity* dan *Quality*. Penelitian ini menggunakan *platform* yang disediakan oleh Google webnya ialah *Google Colab*, yang menggunakan Bahasa pemrograman *Python* secara langsung dari *web* tanpa perlu menginstal program apapun di komputer.

Selanjutnya peneliti menggunakan dua metode klasifikasi, yaitu *Artificial Neural Network* (ANN) dan *Support Vector Machine* (SVM). Dalam studi ini, peneliti akan membandingkan efisiensi masing-masing metode klasifikasi dengan menggunakan dataset *apple quality*. Pada metode klasifikasi ANN, peneliti menggunakan algoritma *feedforward neural network* dengan satu *hidden layer*. Dan menggunakan algoritma *backpropagation* untuk melakukan *training neural network*. Dan pada metode klasifikasi SVM, peneliti menggunakan kernel Radial Basis *Function* (RBF) dan kernel *Linear*.

2.1 Tahap Penelitian

Tahap pertama adalah tahap *input* data, Dimana data mentah yang diperlukan untuk analisis klasifikasi apel dikumpulkan dari berbagai sumber.

Tahap kedua adalah tahap data *cleaning*, data *cleaning* merupakan pembersihan data yang dilakukan sebelum data *preprocessing*, yaitu tugasnya untuk mendeteksi dan memperbaiki kesalahan pada data. Pada tahap ini peneliti melakukan pengecekan data yaitu *missing value* [10].

Tahap ketiga adalah tahap data *preprocessing*, data *preprocessing* adalah memperoleh kumpulan data akhir yang dapat dianggap benar dan berguna untuk algoritma selanjutnya [11].

Tahap keempat adalah tahap data *training* dan data *testing*, data *training* digunakan untuk melatih model *machine learning* dengan menerapkannya pada sejumlah besar contoh data yang telah diketahui hasilnya, sehingga dapat mempelajari pola-pola yang ada dalam data dan membuat prediksi yang akurat. Data *testing* digunakan untuk mengevaluasi kinerja model dan dapat memprediksi hasil yang benar untuk data baru yang tidak pernah dilihat sebelumnya [12]. Pada pengujian ANN dan SVM ini peneliti menggunakan data *test size* dengan ukuran data *testing* 20% dan data *training* 80%, dan data *testing* 30% dan data *training* 70%.

Tahap kelima adalah hasil klasifikasi, hasil ini menggunakan evaluasi *metrics* dengan model perhitungan *accuracy*, *precision*, *recall* dan *f1-score*. *Precision* menunjukkan bagaimana kecocokan antara bagian data yang diambil dengan informasi yang dibutuhkan. *Recall* menunjukkan bagaimana tingkat keberhasilan sistem dalam menemukan kembali informasi. *Accuracy* menunjukkan bagaimana kedekatan nilai yang didapat terhadap nilai sebenarnya. *F1-score* menunjukkan bagaimana menghitung perbandingan rata-rata antara *Precision* dan *Recall* [13].

2.1.1 Diagram Alir



Gambar 1 Diagram Alir

Terdapat 5 tahapan pada penelitian ini, tahap pertama mengumpulkan data dari *platform* Kaggle dengan judul dataset *Apple Quality*. Tahap kedua adalah data *cleaning*, digunakan untuk membersihkan data sebelum ke tahap *preprocessing*. Tahap ketiga adalah data *preprocessing*, tahap ini dataset sudah dianggap benar. Tahap keempat adalah data *training* dan data *testing*, digunakan untuk melatih dan menguji dataset untuk mengevaluasi algoritma yang digunakan. Tahap kelima adalah hasil klasifikasi, hasil ini memberikan gambaran tentang kinerja model yang telah dikembangkan evaluasi *metrics* seperti *precision*, *recall*, *f1-score* dan akurasi untuk menilai seberapa baik model dapat membedakan antara kelas yang berbeda.

2.2 Rumus Matematika

2.2.1 Support Vector Machine (SVM)

Pemahaman sederhana konsep SVM ini digambarkan sebagai usaha mencari *hyperplane* terbaik [14]. Dalam SVM, fungsi keputusan digunakan untuk memisahkan kelas data. Untuk kernel *linear*, fungsi keputusan merupakan hasil dari produk titik antara *vector* bobot dan *vector fitur*, ditambah dengan bias. Sementara itu, dalam kernel polinomial, fungsi keputusan menggabungkan *vector* bobot dengan transformasi polinomial dari *vector* fitur input, ditambah dengan bias. Sedangkan kernel RBF, fungsi keputusan merupakan hasil dari jumlah bobot dikali dengan ekspresi dari kernel RBF, yang mengukur kemiripan antara *vector fitur input* dan titik-titik data yang mendukung [15]. Dengan pemilihan kernel sesuai, SVM mampu menghasilkan model klasifikasi yang akurat dan kuat untuk berbagai jenis data [16]. Berikut adalah rumus kernel:

- Kernel Linear

$$f(x) = w^T \cdot x + b \quad (1)$$

Keterangan:

w : *vector* bobot

x : *vector input*

b : bias

- Kernel Polinomial

$$f(x) = \sum_{i=1}^n \alpha_i y_i (x_i \cdot x + c)^d + b \quad (2)$$

Keterangan:

α_i : koefisien *Lagrange*

y_i : label kelas

x_i : *vector fitur* data

c : konstanta

d : derajat polinomial

b : bias

- Kernel RBF

$$f(x) = \sum_{i=1}^n \alpha_i y_i \exp(-\gamma \|x - x_i\|^2) + b \quad (3)$$

Keterangan:

α_i : koefisien *Lagrange*

y_i : label kelas

x_i : *vector fitur* data

γ : *parameter* skala

b : bias

2.2.2 Artificial Neural Network (ANN)

ANN melibatkan perhitungan berulang dari aktivasi *neuron* disetiap *layer*. Untuk setiap *neuron* dalam jaringan, nilai keluarannya dihitung dengan menggunakan fungsi aktivasi, yang dapat berupa fungsi *sigmoid* untuk lapisan tersembunyi dan fungsi *softmax* untuk lapisan *output* dalam kasus klasifikasi multikelas. Rumus matematika ini juga melibatkan penyesuaian bobot dan bias disetiap iterasi melalui proses pembelajaran menggunakan algoritma seperti *backpropagation*, dimana

gradien dari fungsi kerugian terhadap bobot dan bias digunakan untuk memperbarui *parameter* jaringan [17]. Secara keseluruhan, rumus matematika untuk ANN mencerminkan komputasi *paralel* yang kompleks dari setiap *layer neuron*, yang dijalankan berulang kali selama proses pelatihan untuk mempelajari pola-pola yang kompleks dalam data [18]. Berikut adalah rumus ANN:

- *Feedforward*

$$z = w \cdot x + b \quad (4)$$

$$a = \sigma(z) \quad (5)$$

Keterangan:

z: *input* dari fungsi aktivasi

w: *vector* bobot

x: *vector input*

b: bias

σ : fungsi aktivasi

a: *output* dari *neuron*

- *Backpropagation*

$$\delta = \frac{\partial L}{\partial z} \quad (6)$$

$$\frac{\partial L}{\partial w} = \delta \cdot x \quad (7)$$

$$\frac{\partial L}{\partial b} = \delta \quad (8)$$

Keterangan:

δ : *gradien* fungsi terhadap *input neuron*

L: fungsi kerugian

$\frac{\partial L}{\partial w}$: *gradien* fungsi kerugian bobot

$\frac{\partial L}{\partial b}$

$\frac{\partial L}{\partial b}$: *gradien* fungsi kerugian bias

$\frac{\partial L}{\partial b}$

3. HASIL DAN PEMBAHASAN

Sebelum dianalisis, data diperiksa untuk diuji apakah ada *missing value* atau tidak. Jika sudah tidak ada yang *missing value*, maka lanjut ke tahap selanjutnya.

```
[6] df.isna().sum().sort_values(ascending=False)
A_id      0
Size      0
Weight    0
Sweetness 0
Crunchiness 0
Juiciness 0
Ripeness  0
Acidity   0
Quality   0
dtype: int64
```

Gambar 2 Memeriksa *missing value*

Selanjutnya menulis fungsi untuk membagi data menjadi data latih dan data uji. Variabel *x* dan *y* menyimpan data *input* dan *output*, sedangkan *test_size* menentukan proporsi data uji. fungsi

train_test_split akan mengacak data sebelum membagikannya, dapat dilihat dari nilai *random_state* yang diberikan [19].

```
[9] from sklearn.model_selection import train_test_split
     x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.30, random_state = 215)
```

Gambar 3 Menulis fungsi

Setelah menulis fungsi, kode *python* menampilkan hasil laporan klasifikasi untuk data uji. Laporan tersebut membandingkan label sebenarnya *y_test* dengan label prediksi *y_predict_ann* yang dihasilkan oleh model ANN. Untuk melihat hasilnya dengan cara menulis fungsi *classification_report* *sckit-learn*. Fungsi ini digunakan untuk mengevaluasi performa model klasifikasi dan menghitung berbagai *metrics* seperti *precision*, *recall*, *f1-score* dan akurasi untuk setiap kelas dalam kumpulan data.

```
print(classification_report(y_test, y_predict_ann))
```

Gambar 4 Kode *python* untuk menghasilkan laporan klasifikasi

	precision	recall	f1-score	support
bad	0.83	0.57	0.67	605
good	0.67	0.88	0.76	595
accuracy			0.72	1200
macro avg	0.75	0.72	0.72	1200
weighted avg	0.75	0.72	0.72	1200

Gambar 5 Hasil laporan klasifikasi ANN

Selanjutnya untuk metode algoritma SVM setelah menulis fungsi seperti contoh ANN diatas, melakukan prediksi pada data pengujian menggunakan *svc.fit()* dan *predict()* kelas SVM, dan menghitung skor akurasi menggunakan fungsi *accuracy_score()* dari *scikit-learn*. Performa ini konsisten dengan keluaran model yang sama pada data uji berbeda.

```
svc.fit(x_train, y_train)
y_pred=svc.predict(x_test)
print('Akurasi model dengan default hyperparameters: {0:0.4f}'.format(accuracy_score(y_test, y_pred)))
```

Gambar 6 Model *hyperparameters*

Setelah menulis model *hyperparameters*, selanjutnya menulis fungsi kernel RBF, kernel *Linear* dan kernel Polinomial dengan hasil akurasi berbeda beda setiap fungsi kernel.

```
svc=SVC(C=1.0)
svc.fit(x_train, y_train)
y_pred=svc.predict(x_test)
print('Akurasi model dengan rbf kernel and C=1.0 : {0:0.4f}'.format(accuracy_score(y_test, y_pred)))
```

Gambar 7 Kernel RBF

```
linear_svc=SVC(kernel='linear', C=1.0)
linear_svc.fit(x_train, y_train)
y_pred_test=linear_svc.predict(x_test)
print('Model accuracy score with linear kernel and C=1.0 : {0:0.4f}'.format(accuracy_score(y_test, y_pred_test)))
```

Gambar 8 Kernel Linear

```
poly_svc=SVC(kernel='poly', C=1.0)
poly_svc.fit(x_train, y_train)
y_pred_test=poly_svc.predict(x_test)
print('Model accuracy score with polynomial kernel and C=1.0 : {0:0.4f}'.format(accuracy_score(y_test, y_pred_test)))
```

Gambar 9 Kernel Polinomial

Kode python menampilkan fungsi *classification_report* dan *confusion_matrix* dari modul *sklearn.metrics* untuk mengevaluasi performa pengklasifikasi machine learning. Fungsi *classification_report* menghasilkan performa model klasifikasi, termasuk *precision*, *recall*, *F1-score*, dan akurasi di semua kelas [20]. Fungsi *confusion_matrix* menghasilkan matriks yang merangkum jumlah prediksi positif benar, negatif benar, positif palsu, dan negatif palsu yang dibuat oleh pengklasifikasi. Variabel *y_test* dan *y_pred_test* kemungkinan masing-masing mewakili label benar dan prediksi untuk kumpulan data pengujian.

```
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

print(classification_report(y_test, y_pred_test))

cm = confusion_matrix(y_test, y_pred_test)
cm
```

Gambar 10 Kode Python untuk menghasilkan laporan klasifikasi

	precision	recall	f1-score	support
bad	0.00	0.00	0.00	610
good	0.49	1.00	0.66	590
accuracy			0.49	1200
macro avg	0.25	0.50	0.33	1200
weighted avg	0.24	0.49	0.32	1200

Gambar 11 Hasil laporan klasifikasi SVM

Hasil dari menguji dataset dengan eksperimen 10 kali menggunakan 20% data uji dan 80% data latih dan 30% data uji dan 70% data latih menggunakan algoritma ANN dan SVM. Untuk SVM menggunakan fungsi kernel *linear*, RBF dan polinomial.

Tabel 1 ANN 20% data uji dan 80% data latih

Eksperimen	Precision	Recall	F1-Score	Akurasi
1 (534)	0.74	0.71	0.70	0.71
2 (324)	0.73	0.73	0.72	0.73
3 (253)	0.74	0.72	0.71	0.72
4 (142)	0.73	0.70	0.69	0.71
5 (352)	0.73	0.69	0.69	0.70
6 (425)	0.74	0.73	0.73	0.73
7 (124)	0.68	0.65	0.63	0.66
8 (523)	0.73	0.73	0.73	0.73
9 (215)	0.78	0.77	0.77	0.77
10 (154)	0.76	0.76	0.76	0.76
Rata-rata	0.73	0.71	0.71	0.72

Tabel 2 ANN 30% data uji dan 70% data latih

Eksperimen	Precision	Recall	F1-Score	Akurasi
1 (534)	0.70	0.62	0.58	0.63
2 (324)	0.74	0.73	0.73	0.73
3 (253)	0.73	0.69	0.68	0.69
4 (142)	0.75	0.75	0.75	0.75
5 (352)	0.75	0.74	0.74	0.74
6 (425)	0.73	0.72	0.72	0.72
7 (124)	0.74	0.68	0.68	0.68
8 (523)	0.73	0.71	0.70	0.71
9 (215)	0.74	0.72	0.72	0.72
10 (154)	0.75	0.72	0.72	0.72
Rata-rata	0.73	0.70	0.70	0.70

Tabel 3 SVM 20% data uji dan 80% data latih (Polinomial)

Eksperimen	Precision	Recall	F1-Score	Akurasi
1 (534)	0.24	0.50	0.33	0.48
2 (324)	0.24	0.50	0.32	0.48
3 (253)	0.25	0.50	0.33	0.50
4 (142)	0.23	0.50	0.32	0.47
5 (352)	0.24	0.50	0.33	0.48
6 (425)	0.23	0.50	0.32	0.47
7 (124)	0.24	0.50	0.32	0.48
8 (523)	0.23	0.50	0.32	0.47
9 (215)	0.49	0.50	0.45	0.49
10 (154)	0.52	0.51	0.46	0.51
Rata-rata	0.29	0.50	0.35	0.48

Tabel 4 SVM 20% data uji dan 80% data latih (RBF)

Eksperimen	Precision	Recall	F1-Score	Akurasi
1 (534)	0.75	0.75	0.75	0.48
2 (324)	0.48	0.48	0.48	0.48
3 (253)	0.48	0.48	0.48	0.49
4 (142)	0.49	0.49	0.49	0.46
5 (352)	0.52	0.52	0.51	0.51
6 (425)	0.50	0.50	0.50	0.46
7 (124)	0.50	0.50	0.50	0.48
8 (523)	0.47	0.47	0.47	0.46
9 (215)	0.49	0.49	0.49	0.47
10 (154)	0.48	0.48	0.48	0.49
Rata-rata	0.51	0.51	0.51	0.47

Tabel 5 SVM 20% data uji dan 80% data latih (Linear)

Eksperimen	Precision	Recall	F1-Score	Akurasi
1 (534)	0.77	0.76	0.76	0.76
2 (324)	0.74	0.74	0.74	0.74
3 (253)	0.76	0.76	0.76	0.76
4 (142)	0.76	0.76	0.75	0.75
5 (352)	0.73	0.73	0.73	0.73
6 (425)	0.72	0.72	0.72	0.72
7 (124)	0.74	0.74	0.74	0.73
8 (523)	0.73	0.73	0.73	0.72
9 (215)	0.78	0.78	0.78	0.77
10 (154)	0.77	0.76	0.76	0.76
Rata-rata	0.75	0.74	0.74	0.74

Tabel 6 SVM 30% data uji dan 70% data latih (Polinomial)

Eksperimen	Precision	Recall	F1-Score	Akurasi
1 (534)	0.25	0.50	0.33	0.49
2 (324)	0.25	0.50	0.33	0.49
3 (253)	0.25	0.50	0.33	0.49
4 (142)	0.24	0.50	0.33	0.48
5 (352)	0.25	0.50	0.33	0.49
6 (425)	0.24	0.50	0.32	0.47
7 (124)	0.25	0.50	0.33	0.49
8 (523)	0.24	0.50	0.33	0.48
9 (215)	0.25	0.50	0.33	0.49
10 (154)	0.25	0.50	0.33	0.49
Rata-rata	0.24	0.50	0.32	0.48

Tabel 7 SVM 30% data uji dan 70% data latih (RBF)

Eksperimen	Precision	Recall	F1-Score	Akurasi
1 (534)	0.53	0.53	0.53	0.49
2 (324)	0.49	0.49	0.49	0.49
3 (253)	0.49	0.49	0.49	0.49
4 (142)	0.49	0.49	0.49	0.48
5 (352)	0.51	0.51	0.50	0.49
6 (425)	0.50	0.50	0.50	0.47
7 (124)	0.51	0.51	0.51	0.49
8 (523)	0.50	0.50	0.50	0.48
9 (215)	0.50	0.50	0.50	0.49
10 (154)	0.77	0.77	0.76	0.48
Rata-rata	0.52	0.52	0.52	0.48

Tabel 8 SVM 30% data uji dan 70% data latih (Linear)

Eksperimen	Precision	Recall	F1-Score	Akurasi
1 (534)	0.75	0.75	0.75	0.74
2 (324)	0.74	0.74	0.74	0.74
3 (253)	0.76	0.76	0.76	0.75
4 (142)	0.75	0.75	0.75	0.74
5 (352)	0.74	0.74	0.74	0.73
6 (425)	0.73	0.73	0.73	0.72
7 (124)	0.74	0.74	0.74	0.74
8 (523)	0.74	0.74	0.74	0.74
9 (215)	0.77	0.77	0.77	0.77
10 (154)	0.77	0.77	0.76	0.76
Rata-rata	0.74	0.74	0.74	0.74

Dari hasil rata-rata diatas, akan di rangkum menjadi satu agar dapat mempermudah untuk dibaca dan dimengerti.

Tabel 9 Rata-rata ANN

Data Latih	Data Uji	Nilai Rata-rata			
		Precision	Recall	F1-Score	Akurasi
80%	20%	0.73	0.71	0.71	0.72
70%	30%	0.73	0.70	0.70	0.70

Tabel 10 Rata-rata SVM Polinomial

Data Latih	Data Uji	Nilai Rata-rata			
		Precision	Recall	F1-Score	Akurasi
80%	20%	0.29	0.50	0.35	0.48
70%	30%	0.24	0.50	0.32	0.48

Tabel 11 Rata-rata SVM RBF

Data Latih	Data Uji	Nilai Rata-rata			
		Precision	Recall	F1-Score	Akurasi
80%	20%	0.51	0.51	0.51	0.47
70%	30%	0.52	0.52	0.52	0.48

Tabel 12 Rata-rata SVM Linear

Data Latih	Data Uji	Nilai Rata-rata			
		Precision	Recall	F1-Score	Akurasi
80%	20%	0.75	0.74	0.74	0.74
70%	30%	0.74	0.74	0.74	0.74

Dari rata-rata diatas, dapat disimpulkan bahwa rata-rata evaluasi terbaik diperoleh dari SVM dengan kernel *Linear*, yang memiliki rata-rata 0.74 baik untuk data latih 80% dan data uji 20%, maupun untuk data latih 70% dan data uji 30%. Oleh karena itu, SVM dengan kernel *Linear* memberikan kinerja terbaik berdasarkan rata-rata evaluasi yang diberikan.

4. KESIMPULAN

SVM dengan kernel *Linear* menunjukkan kinerja terbaik berdasarkan rata-rata evaluasi yang diberikan. Dengan rata-rata evaluasi yang sama tinggi 0.74 baik untuk data latih 80% dan data uji 20%, maupun untuk data latih 70% dan data uji 30%, SVM dengan kernel *Linear* dapat dianggap sebagai pilihan yang konsisten dalam memprediksi kualitas apel.

Kelebihan dari SVM dengan kernel *Linear*:

1. Konsistensi dalam kinerja: SVM dengan kernel *Linear* memberikan hasil yang stabil dan konsisten pada dua skenario pemisahan data yang berbeda.
2. Model SVM dengan kernel *Linear* lebih mudah diinterpretasikan karena batas keputusan yang dihasilkan berupa *hyperplane linear*.
3. Cocok untuk data dengan dimensi tinggi: SVM dengan kernel *Linear* sering kali lebih baik dalam menangani data dengan dimensi tinggi karena dapat memisahkan kelas yang kompleks dalam ruang fitur yang lebih tinggi.

Namun, SVM dengan kernel *Linear* juga memiliki beberapa kelemahan:

1. Kurang mampu menangani data *non-linear*: SVM dengan kernel *Linear* cenderung kurang efektif dalam memisahkan kelas dalam data yang tidak dapat dipisahkan secara *linier*.
2. Sensitif terhadap data yang tidak seimbang: SVM dengan kernel *Linear* dapat memberikan hasil yang buruk jika kelas dalam data tidak seimbang.

Untuk pengembangan selanjutnya, beberapa langkah yang mungkin dilakukan adalah:

1. Peningkatan pengelolaan data tidak seimbang: Menggunakan Teknik *oversampling* atau *undersampling* untuk menangani ketidakseimbangan kelas dalam data.
2. Penyesuaian kernel: Eksplorasi penggunaan kernel *non-linear* yang dapat meningkatkan kemampuan SVM dalam menangani data yang tidak linier.
3. Seleksi fitur: Melakukan seleksi fitur untuk memperbaiki kinerja model dengan menghilangkan fitur-fitur yang tidak informatif.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada nim 535220108 dan 535220149 yang telah memberikan dukungan dan masukan yang berharga dalam pembuatan artikel ilmiah ini. Kontribusi dari pihak-pihak tersebut sangat membantu dalam meningkatkan kualitas artikel ilmiah ini.

DAFTAR PUSTAKA

- [1] C. Suryanti and M. G. Rohman, "Klasifikasi Kualitas Buah Apel Berdasarkan Warna dan Bentuk Menggunakan Metode KNN," *Generation Journal*, vol. 8, no. 1, pp. 2549-2233, 2023.
- [2] S. Ulum, R. F. Alifa, P. Rizkika and C. Rozikin, "Perbandingan Performa Algoritma KNN dan SVM dalam Klasifikasi Kelayakan Air Minum," *Generation Journal*, vol. 7, no. 2, pp. 2549-2233, 2023.
- [3] R. A. Safitri, S. Nurdiani, D. Riana and S. Hadianti, "Klasifikasi Jenis Buah Apel Menggunakan Metode Orde 1 dengan Algoritma Multi Support-Vector Machines," *Paradigma*, vol. 21, no. 2, pp. 2579-3500, 2019.
- [4] S. K. Rajput and R. K. Agrawal, " A comparative study of machine learning techniques for apple fruit quality classification.," *International Journal of Agricultural and Biological Engineering*, vol. 14, no. 1, pp. 123-135, 2021.

- [5] N. Elgiriye withana and E. Nidula, "Performance Evaluation of Classification Algorithms for Predicting Apple Quality," *Journal of Artificial Intelligence Research*, vol. 12, no. 4, pp. 345-356, 2020.
- [6] Y. Li, Y. Liu and X. Han, "Apple quality identification and classification by image processing based on convolutional neural networks," *Scientific Reports*, vol. 11, no. 16618, 2021.
- [7] G. Aurelien, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media, 2019.
- [8] A. D. P. Wicaksono and H. Tantyoko, "Hybrid Algoritma Vgg16-Net dengan Support Vector Machine untuk Klasifikasi Jenis Buah dan Sayuran," *Jurnal Teknologi Informasi dan Multimedia*, vol. 5, no. 2, pp. 56-65, 2023.
- [9] I. D. Apostolopoulos , M. Tzani and S. I. Aznaouridis, "A General Machine Learning Model for Assessing Fruit Quality Using Deep Image Features," *MDPI*, vol. 4, no. 4, pp. 812-830, 2023.
- [10] F. I. Ihab and C. Xu, Data Cleaning, Morgan & Claypool, 2019.
- [11] S. Garcia, S. Ramirez-Gallego, J. Luengo, J. M. Benitez and F. Herrea, "Big Data Analytics," *Big data preprocessing: methods and prospects*, vol. 1, no. 9, 2016.
- [12] T. Hastie, J. Friedman and R. Tibshirani, The Elements of Statistical Learning, Springer Link, 2001.
- [13] A. Faradibah, D. Widyawati, A. U. T. Syahar, S. R. Jabir and P. L. L. Belluano, "Comparison Analysis of Random Forest Classifier, Support Vector Machine, and Artificial Neural Network Performance in Multiclass Brain Tumor Classification," *Indonesian Journal of Data and Science*, vol. 4, no. 2, 2023.
- [14] A. A. Kasim and M. Sudarsono, "Algoritma Support Vector Machine (SVM) untuk," *Seminar Nasional APTIKOM (SEMNASTIK)*, vol. 1, no. 2922-8929, p. 569, 2019.
- [15] R. Munawarah, O. Soesanto and M. R. Faisal, "PENERAPAN METODE SUPPORT VECTOR MACHINE PADA DIAGNOSA HEPATITIS," *Jurnal Ilmiah KLIK Kumpulan Jurnal Ilmu Komputer*, vol. 3, no. 1, 2016.
- [16] M. Jordan, P. J. Kleinberg and S. Bernhard, Pattern Recognition and Machine Learning Christopher M. Bishop, New York: Springer Science, 2006.
- [17] E. Saraswati, Y. Umaidah and A. Voutama, "Penerapan Algoritma Artificial Neural Network untuk Klasifikasi Opini Publik Terhadap Covid-19," *Generation Journal*, vol. 5, no. 2, 2021.
- [18] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.
- [19] S. Raschka and V. Mirjalili, Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, Packt Publishing, 2019.
- [20] S. S. Tripathy and B. Behera, "PERFORMANCE EVALUATION OF MACHINE LEARNING," *Journal of Biomechanical Science and Engineering*, p. 621, 2023.