Jurnal Komputer dan Informatika
Vol 20 No 1, April 2025: hlm 33 - 48

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

ANALISIS PERBANDINGAN ALGORITMA *SUPPORT VECTOR MACHINE* DAN *ARTIFICIAL NEURAL NETWORK* DALAM PREDIKSI DAN KLASIFIKASI KUALITAS PISANG

Hansen Pratama
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
Email: hansen.535220128@stu.untar.ac.id,

ABSTRAK

Penelitian untuk membandingkan kinerja Support *Vector Machine* (SVM) untuk klasifikasi kualitas pisang dan *Artificial Neural Network* (ANN) dalam memprediksi kualitas pisang. Metode penelitian mencakup penggunaan kernel RBF untuk SVM dan arsitektur MLP untuk ANN. Dataset pada kualitas pisang sebanyak 8000 sample dan 8 kolom yang digunakan sebagai dataset penelitian ini. SVM dan ANN dilatih dan dievaluasi menggunakan datasets yang sama untuk membandingkan keakuratannya. Hasil eksperimen rata rata menunjukkan bahwa SVM (kernel RBF) mendapatkan akurasi yang lebih tinggi, yaitu pada atribut *accuracy, precision, recall,* dan *f1-score* sebesar 0,981 dibandingkan dengan ANN (MLP) hanya mendapatkan 0,98. Namun, ANN menunjukkan kecenderungan untuk lebih fleksibel dalam menangani variasi data. SVM dengan kernel RBF lebih cocok untuk pemodelan kualitas pisang dengan dataset yang stabil, sementara ANN dengan MLP lebih sesuai untuk situasi dataset yang memiliki variasi lebih kompleks

Kata kunci: Prediksi, Klasifikasi, *Support Vector Machine, Artificial Neural Network,* kualitas pisang

ABSTRACT

This research to compare the performance of the Support Vector Machine (SVM) for banana quality classification and Artificial Neural Network (ANN) algorithms in predicting banana quality. The Research method involves using the RBF kernel for SVM and the MLP architecture for ANN. The dataset on banana quality consists of 8000 samples and 8 columns, used to for this research dataset. SVM and ANN are trained and evaluated using the same datasets to compare their accuracy. The average experimental results show that SVM (RBF Kernel) achieves higher accuracy, with attributes such as accuracy, precision, recall, and f1-score at 0,981 compared to ANN (MLP) which only achieves 0,98. However, ANN shows a tendency to be more flexible in handling data variations. SVM with the RBF kernel is more suitable for modeling banana quality with a stable dataset, while ANN with MLP is more appropriate for datasets with more complex variations

Keywords: *Predicition, Classification, Support Vector Machine, Artificial Neural Network, Banana Quality*

1. PENDAHULUAN

1.1 Latar Belakang Permasalahan

Buah pisang mengandung vitamin, diantaranya vitamin B6, vitamin C, Kalium, tiamin, riboflavin, dan niasin. Pisang merupakan salah satu buah yang diminati karena memiliki manfaat seperti mencegah sembelit, kanker usus, asam lambung, dan meningkatkan kekebalan tubuh. Serat daging buah pisang mengandung sebesar 2,6-gram serat di dalam 100 g buah pisang [1].

Kualitas pisang menjadi faktor penting yang menentukan nilai jual dan kepuasan konsumen. Namun, penilaian kualitas pisang secara manual seringkali tidak akurat, memakan waktu, dan ketidakefisienan. Oleh karena itu, diperlukan metode yang lebih objektif dan efisien untuk menganalisis kualitas pisang.

Algoritma *Support Vector Machine (SVM)* memiliki kemampuan untuk memisahkan data sehingga cocok untuk klasifikasi kualitas pisang, sedangkan *Artificial Neural Network (ANN)* memiliki kemampuan untuk mempelajari pola data yang kompleks, sehingga cocok untuk memprediksi kualitas pisang. Penggunaan kedua algoritma tersebut dapat memberikan hasil yang lebih akurat dan objektif dibandingkan dengan penilaian manual, lalu menggunakan algoritma ini digunakan untuk menganalisis kualitas pisang dalam jumlah besar dengan cepat dan mudah ke dalam kelas tertentu [7]

Penggunaan algoritma *SVM* dan *ANN* dalam menganalisis kualitas pada pisang memungkinkan proses yang cepat dan mudah, terutama dalam menangani jumlah pisang yang besar. Dengan menggunakan pendekatan ini, algoritma mana yang efektif dan efisien untuk industri makanan dapat dengan mudah memantau dan mengelola kualitas pisang secara efisien, mengurangi waktu dan biaya yang diperlukan untuk penilaian manual. Hal ini akan meningkatkan efisiensi produksi serta memastikan konsisten kualitas pada produk.

Dengan terus mengoptimalkan algoritma dan memanfaatkan kemajuan teknologi, dapat meningkatkan integrasi yang baik antara teknologi dan praktik industri yang sudah ada akan menjadi kunci keberhasilan dalam menerapkan sistem analisis kualitas pada pisang berbasis algoritma *SVM* dan *ANN*

1.2 Rumusan masalah

Penentuan kualitas pisang secara tradisional dilakukan secara manual oleh penilai terlatih. Hal ini memiliki beberapa kelemahan, seperti :

- Membutuhkan tenaga yang banyak, terutama untuk jumlah data pisang yang cukup besar
- Mengalami kesalahan, terutama jika dilakukan dengan terburu-buru atau kondisi pencahayaan yang kurang optimal

1.3 Tujuan dan Kegunaan

Tujuan menganalisis kualitas pada pisang menggunakan algoritma *SVM* dan *ANN* adalah untuk membantu petani, pedagang, dan konsumen dalam menentukan kualitas pada pisang dengan lebih akurat dan efisien

1.4 Manfaat Kegiatan

- Dapat memahami dan menerapkan algoritma *SVM* dan *ANN*
- Dapat memahami algoritma yang akurat dan efisien. Dengan membandingkan *SVM* dan *ANN*
- Dapat menambah wawasan baru tentang pemilihan algoritma yang optimal untuk data dengan jumlah nya cukup besar dan karakteristik tertentu

2. METODE PENELITIAN

2.1 Metode Penelitian

Metode penelitian adalah cara ilmiah yang digunakan untuk mendapatkan data secara sistematis dan terstruktur dengan tujuan untuk menjawab pertanyaan penelitian, mengembangkan pengetahuan dan memecahkan masalah. Metode penelitian yang tepat akan membantu peneliti dalam mengumpulkan data yang akurat, valid, dan reliabel, sehingga hasil penelitian dapat dipercaya dan dipertanggungjawabkan [2]. Metode yang digunakan untuk penelitian terdapat dua algoritma, yaitu algoritma *Support Vector Machine (SVM)* dengan kernel *Radial Basis Function*

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

(RBF) dan *Artificial Neural Network (ANN)* untuk melakukan analisis kualitas pada pisang secara objektif dan efisien. Penelitian melalui dua tahap antara lain data akan di latih atau *training* dan data juga akan melalui tahap pengujian atau *testing* [11].

2.2 Jenis Jenis *Machine Learning*

Jenis jenis machine learning merangkum beragam pendekatan untuk pemrosesan data dan pembelajaran mesin, termasuk *Supervised learning,* dimana model mempelajari pola dari data yang telah dilabeli dan *Unsupervised learning,* model mengidentifikasi pola tanpa bantuan label [18].

2.2.1 *Supervised learning*

Supervised learning adalah pendekatan machine learning yang ditentukan berdasarkan penggunaan dataset berlabel *(labeled dataset)* dan model akan diawasi dalam pembelajarannya. Lalu pendekatan *Supervised learning* mengetahui *output* yang dihasilkan. Dalam dataset ini terdapat sebuah "*label*", yaitu satu kolom yang menjadi target output model. Tujuan dari model ini adalah untuk memetakan input ke output yang diinginkan. *Supervised learning* terdiri dari beberapa teknik, yaitu:

a. Regresi
   Regresi adalah teknik *machine learning* yang digunakan untuk memprediksi nilai variabel kontinu (*output*) berdasarkan variabel *input*. Teknik regresi ini melakukan pembelajaran perlu adanya pengawasan. Dalam regresi, model dilatih pada kumpulan data yang berisi contoh nilai input dan output yang sesuai. Model kemudian dapat digunakan untuk memprediksi nilai output baru untuk input yang tidak terlihat sebelumnya. Tujuan regresi, yaitu dapat digunakan untuk memprediksi antara variabel *input* dan *output*, dan untuk membangun model yang dapat digunakan untuk simulasi atau pengambilan keputusan. Dalam regresi terdapat algoritma yang dapat digunakan, seperti
   - *Linear regression*
   - *Classification and Regression Tree (CART)*
   - *Random forest regression*

b. *Classification*
   Classification atau klasifikasi merupakan salah satu teknik *machine learning* yang paling umum digunakan, dimana berfokus pada pengelompokkan data berdasarkan kategori atau kelas yang telah ditentukan sebelumnya. Dalam klasifikasi terdapat algoritma yang sering digunakan [10], seperti
   - *Logistic regression*
   - *Decision tree*
   - *Random forest*
   - *Naive Bayes Classification*
   - *Nearest Neighbor*
   - *Support Vector Machine*

2.2.2 *Unsupervised learning*

Unsupervised learning merupakan pendekatan *machine learning* dengan pembelajaran tanpa diawasi untuk menganalisis dan mengelompokkan data yang tidak berlabel (*unlabelled data*). Tujuan dari model ini adalah untuk menemukan pola tersembunyi dalam data tanpa perlu campur tangan manusia.

Unsupervised learning terdiri dari beberapa teknik yaitu[12][13][14] :

a. *Clustering*
   Clustering adalah teknik *machine learning* yang digunakan untuk mengelompokkan objek data berdasarkan kesamaan. Teknik *clustering* ini melakukan pembelajaran tanpa adanya pengawasan. *Clustering* tidak memerlukan label data yang sudah ditentukan sebelumnya.
   Algoritma *clustering* secara otomatis menemukan pola dalam data dan mengelompokkan objek yang memiliki karakteristik serupa. Tujuan *clustering*, yaitu mengidentifikasi kelompok yang berbeda, mengelompokkan data menjadi segmen yang bermakna, dan dapat membantu mengidentifikasi *outlier* atau data yang tidak biasa dalam dataset.

Contoh algoritma yang sering digunakan, yaitu *K-Means Clustering*

b. Asosiasi
Asosiasi merupakan teknik *machine learning* yang digunakan untuk menerapkan aturan berbeda untuk menemukan hubungan antara variabel dalam sebuah dataset

c. *Dimensionality Reduction*
Dimensionality Reduction merupakan teknik machine learning yang digunakan untuk mengurangi jumlah variabel *data training* atau data latih. Data seperti ini lebih menantang jika dilakukan pemodelan, maka sering disebut kutukan dimensionalitas *(dimensionality curse)*

2.3 *Algoritma Support Vector Machine (SVM)*

SVM merupakan metode klasifikasi data dengan menggunakan metode *machine learning* dalam analisis data dan mengurutkannya dan teknik klasifikasi yang cukup terkenal serta banyak dipakai oleh para ilmuan. Algoritma ini pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai konsep unggulan dalam bidang *pattern recognition*. *SVM* mencoba untuk mencari *hyperplane* yang memiliki margin terbesar, yaitu jarak terdekat antara *hyperplane* dan titik-titik dari kedua kelas. Sehingga dapat dinotasikan persamaan *hyperplane* sebagai berikut [3] :

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

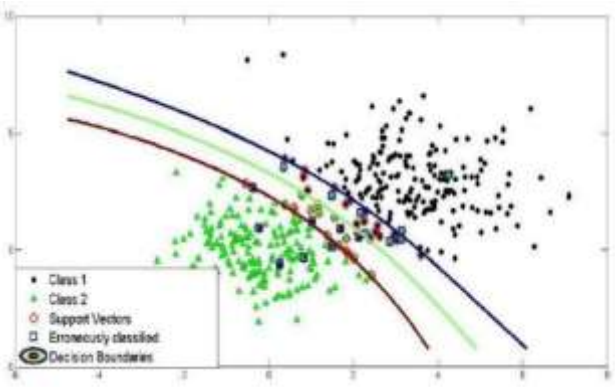
Keterangan :

$f(x)$ = mewakili nilai prediksi untuk variabel target, diberikan sekumpulan fitur input x
W = vektor bobot, setiap bobot sesuai dengan kepentingan fitur yang sesuai dalam mempengaruhi nilai prediksi
T = menunjukkan operasi transpos, yang mengubah vektor menjadi padanan barisnya
x = vektor fitur input, mewakili nilai variabel independen untuk titik data tertentu
b = bias, nilai konstan yang ditambahkan ke jumlah tertimbang dari fitur fitur

SVM dibedakan menjadi dua, yaitu *SVM* terstrukur *(linear)* dan *SVM* tidak terstruktur (*non-linear)*. *SVM* tidak terstruktur memerlukan pemetaan dengan dimensi lebih tinggi, sedangkan *SVM linear* dapat dipisahkan secara linear. Terdapat beberapa jenis kernel yang digunakan dalam *SVM* untuk menangani data yang tidak linear secara efisien [8].

Kernel

Kernel adalah fungsi yang digunakan untuk menghitung titik antara pasangan datadalam ruangan fitur yang mungkin memiliki dimensi yang lebih tinggi [9] daripada dimensi data asli dan menjadikan data nonlinear terpisah secara linear [4]. Beberapa kernel umum yang digunakan dalam *SVM* yaitu kernel *Linear*, kernel *Polynomial*, dan kernel *Radial Basis Function (RBF).* Kernel RBF adalah jenis *kernel* untuk menangani data yang tidak terstruktur atau *non-linear*. Sebagai contoh klasifikasi pada gambar dibawah ini menggunakan *kernel* RBF. Pemisah di ruang fitur ketika diproyeksikan kembali ke dimensi asli menjadi fungsi *non-linear* dari ruang data awal



Gambar 1 *Kernel RBF*

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

Fungsi *kernel* ini adalah untuk mengukur jarak atau kesamaan antara dua titik dalam ruang fitur [7]. Fungsi *kernel* RBF dapat dinotasikan sebagai berikut :

$$K(\text{□□}) = \text{□□□}\left(\frac{-||\text{□} - \text{□}||^2}{2.\sigma^2}\right) \quad (2)$$

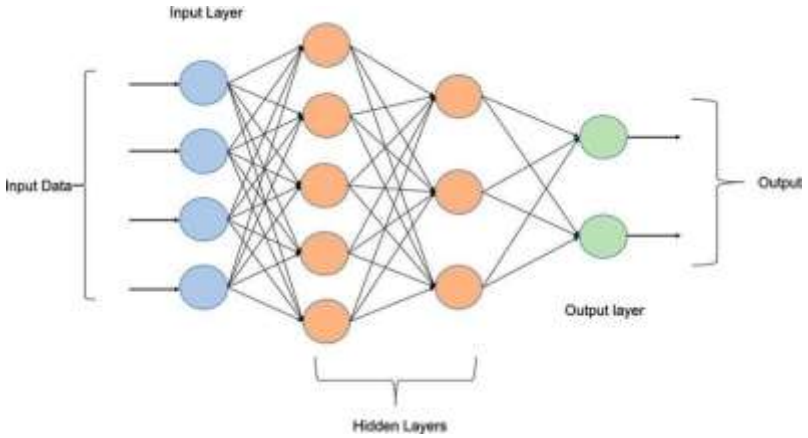
Keterangan :
x = vektor data pertama
y = vektor data kedua
σ = gamma, yang mengontrol seberapa cepat nilai *kernel* menurun dengan bertambahnya jarak antara dua titik

$||\text{□} - \text{□}||^2$ = menghitung jarak euclidean kuadrat antara vektor x dan y
Exp (□ = menghitung fungsi eksponensial dari x

2.4 *Algoritma Artificial Neural Network (ANN)*

ANN merupakan sebuah sistem yang cerdas dengan teknik pengolahan informasi yang terinspirasi oleh cara kerja sistem saraf biologis, pada sel otak manusia untuk memproses informasi [5]. Algoritma dikenalkan oleh David Rumelahrt, Geoffrey Hinton dan Ronald Wiliams pada papernya di tahun 1986 [15]. *ANN* dapat direpresentasikan menjadi 3 bagian yang terdiri dari *layer* masuk atau *input*, *layer* keluar atau *output* dan layer yang tersembunyi atau *hidden layer* [16]. Kemudian hasil dari masing masing *layer* tersembunyi disebut *activation* atau *node value*. Fungsi pada aktivasi atau *activation* yang berfungsi sebagai sinyal untuk menentukan output ke neuron lainnya [17], tidak semua memiliki *hidden layer* tetapi bisa juga untuk memperbaiki penimbang pada lapisan tersembunyi atau *hidden layer* [19].

Layer input berfungsi untuk memproses satu elemen dari data masukan (misalnya, nilai piksel dan gambar), selanjutnya layer tersembunyi atau *hidden layer* berfungsi untuk melakukan pemrosesan dan transformasi data yang diterima dari lapisannya dan ukuran lapisan tersembunyi menentukan kompleksitas dan kapasitas belajar dari *ANN*. Salah satu arsitektur ANN yang paling banyak diketahui adalah *Multilayer Perceptron (MLP). Multilayer Perceptron (MLP)* merupakan arsitektur ANN yang paling dasar dan umum digunakan. Arsitektur nya terdiri dari beberapa lapisan neuron yang terhubung satu sama lain dengan cara *feed forward*. Berikut ini adalah penggambaran dari *MLP*



Gambar 2 *Multilayer Perceptron*

MLP memiliki neuron dan lapisan yang dihubungkan oleh bobot dan sangat baik dalam mendekati fungsi dalam ruang berdimensi tinggi [6]. Semakin banyak jumlah *hidden layer* maka memerlukan komputasi waktu lebih lama [20] contoh pada gambar 2 *hidden layer* berada di posisi tengah, dan sebaiknya disesuaikan dengan kompleksitas permasalahan. Untuk menghitung output

setiap neuron di *hidden layer* dapat dinotasikan sebagai berikut ini

$$\square_{\square} = \sum_{\square 1}^{\square} \square_{\square\square}\square_{\square} + \square \qquad \square_{\square} = \sum_{\square 1}^{\square} \square_{\square\square}\square_{\square} + \square \tag{3}$$

Keterangan :

$\square_{\square}$ = matrik input neuron

$\square_{\square}$ = matrik output neuron

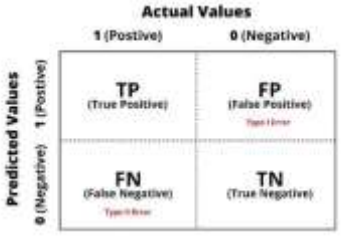
$\square_{\square\square}$ dan $\square_{\square\square}$ = bobot penghubung antara neuron pada dua layer berbeda

b = bias

n dan m = jumlah neuron

2.6 *Confusion Matrix*

Confusion Matrix adalah suatu metode untuk melakukan pengukuran performa dari model klasifikasi yang digunakan dan dapat disebut juga sebagai evaluasi model. Tujuan dari *Confusion Matrix* adalah untuk melihat kelebihan dan kekurangan dari model yang dibuat, sehingga terdapat perbaikan dan memberikan hasil yang lebih baik.



Gambar 3 *Confusion Matrix*

Confusion Matrix terdapat empat istilah yaitu *True* Positif *(TP), True* Negatif *(TN), False* Positif *(FP), dan False* Negatif *(FN).* Lalu dalam *confusion matrix* terdapat nilai, yaitu *accuracy, precision, recall,* dan *f1-score*. Cara untuk mendapatkan nilai tersebut, sebagai berikut :

- *Accuracy*
  Accuracy atau akurasi adalah presentasi dari total data yang diidentifikasikan dan dinilai

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{4}$$

- *Precision*
  Precision atau presisi adalah data yang diambil berdasarkan informasi yang kurang

$$Precision = \frac{TP}{(TP + FP)} \tag{5}$$

- *Recall*
  Recall atau sensitivitas adalah data penghapusan yang berhasil diambil dari data yang relevan dengan kueri

$$Recall = \frac{TP}{(TP + FN)} \tag{6}$$

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

- *F1-score*

F1-score adalah evaluasi yang mencerminkan keseimbangan antara presisi atau *precision* dan sensitivitas atau *recall*

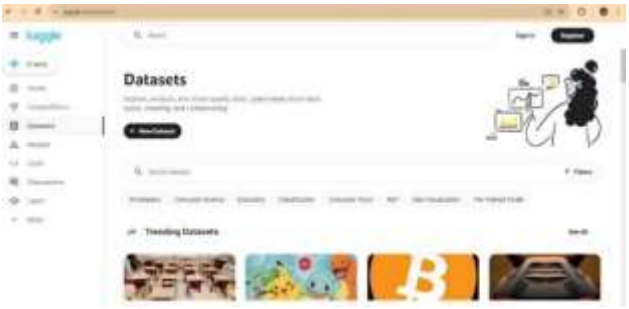
$$\text{F1-score} = \frac{(2 * \frac{TP}{(TP + FN)} * \frac{TP}{(TP + FP)})}{(\frac{TP}{(TP + FN)} + \frac{TP}{(TP + FP)})} \quad (7)$$

3. HASIL DAN PEMBAHASAN

3.1 Pembahasan

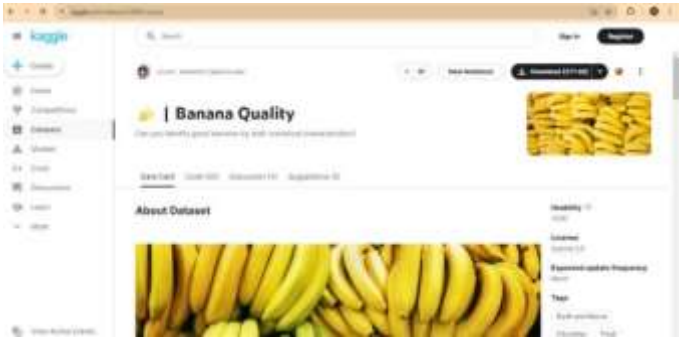
3.1.1 Pengumpulan data

Peneliti mendapatkan dataset yang bersumber dari website kaggle.com. Berikut tampilan website Kaggle



Gambar 4 Tampilan website kaggle

Setelah sudah mengunjungi website kaggle, lalu pilih dataset yang digunakan untuk penelitian, pada penelitian menggunakan dataset dengan nama *Banana Quality*. Dataset tersebut menjelaskan tentang kualitas pada pisang yang layak untuk dikonsumsi. Dataset tersebut dapat disimpan dengan cara klik *Download* pada halaman dataset lalu lakukan ekstrak pada folder. Berikut tampilan halaman pada dataset *Banana Quality*



Gambar 5 Tampilan halaman datasets

3.1.2 Fungsi Atribut

Penelitian ini menggunakan dataset tentang kualitas pada pisang terdapat 8000 sampel dan 9 kolom. Lalu pada 8 kolom tersebut, yaitu *Size*, *Weight*, *Sweetness*, *Softness*, *HarvestTime*, *Ripeness*, *Acidity*. Atribut *Size* atau ukuran mengacu pada dimensi fisik pisang, biasanya diukur berdasarkan panjang atau lingkar, pisang yang lebih besar umumnya dianggap lebih diinginkan.

Setelah itu terdapat atribut *Weight* atau berat dianggap sebagai indikator kematangan dan kualitas keseluruhan. Selanjutnya pada atribut *Sweetness* atau kemanisan, atribut termasuk penting karena menentukan rasa dan daya tariknya bagi konsumen. Kemanisan pisang meningkat seiring dengan kematangan.

Lalu pada atribut *softness* atau kelembutan, biasanya lebih lembut akan mudah dihaluskan sedangkan pisang yang kurang matang memiliki tekstur yang lebih keras. Selanjutnya pada atribut *HarvestTime* atau waktu panen, berfungsi untuk mengetahui jika pisang yang dipanen lebih awal mungkin kurang matang dan memiliki tingkat kemanisan yang lebih rendah, sementara pisang yang terlambat biasanya lebih matang dan lebih manis. Lalu pada atribut *Ripeness* atau kematangan berfungsi untuk menentukan kelayakan makan dan kualitas keseluruhan pisang. Pisang matang memiliki kulit kuning dengan bintik bintik coklat, tekstur yang lembut, dan tingkat kemanisan yang tinggi. Dan pada atribut terakhir, yaitu atribut *Acidity* atau keasaman, pada atribut ini mengacu nilai asam dari rasa pisang. Pisang dengan tingkat keasaman yang lebih tinggi mungkin kurang disukai untuk beberapa konsumen

3.2 Pengolahan data

Pada penelitian ini menghasilkan 4 pengamatan dan masing masing pengamatan melakukan 5 kali eksperimen untuk mendapatkan hasil rata rata dengan cara menjumlahkan semua hasil dari 5 kali eksperimen lalu dibagi dengan 5. Pengamatan tersebut dilakukan pada dua algoritma yaitu *SVM* dengan kernel *RBF* dan *ANN* dengan arsitektur *MLP* untuk membandingkan algoritma mana yang sesuai dengan datasets *Banana Quality*, lalu hasil pengamatan dikemas dalam tabel supaya lebih mudah untuk dibaca

3.2.1 Langkah langkah proses data

Langkah pertama, peneliti mengimpor *library drive* dari *google colab*. Lalu menghubungkan *google drive (gdrive)* secara di dalam notebook.

```
[1] from google.colab import drive
    drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Gambar 5 Struktur kode untuk menghubungkan *gdrive* dalam notebook

Langkah selanjutnya mengimport library yang diperlukan untuk melakukan penelitian pada datasets.

```
[2] import numpy as np
    from sklearn import datasets
    import matplotlib.pyplot as plt
    import pandas as pd
    import seaborn as sns
    from sklearn.neural_network import MLPClassifier
    from sklearn.datasets import make_classification
    from sklearn.metrics import classification_report
    from sklearn.metrics import confusion_matrix
    from sklearn.model_selection import train_test_split
    from numpy.random import rand
    from numpy.random import randint
    import random
    from sklearn import svm
    %matplotlib inline
```

Gambar 6 *Import library*

Langkah selanjutnya membaca data dengan format CSV file dengan menggunakan *library* dari *pandas* lalu diinisialisasikan sebagai *pd.*

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

```
import pandas as pd
df = pd.read_csv('/content/drive/My Drive/Colab Notebooks/Dataset/banana_quality.csv')
df
```

	Size	Weight	Sweetness	Softness	HarvestTime	Ripeness	Acidity	Quality
0	-1.924968	0.468078	3.077832	-1.472177	0.294799	2.435570	0.271290	Good
1	-2.409751	0.486870	0.346921	-2.495099	-0.892213	2.067549	0.307325	Good
2	-0.357607	1.483176	1.568452	-2.645145	-0.647267	3.090643	1.427322	Good
3	-0.868524	1.566201	1.889605	-1.273761	-1.006278	1.873001	0.477862	Good
4	0.651825	1.319199	-0.022459	-1.209709	-1.430692	1.078345	2.812442	Good
...	...	...	...	...	...	...	...	...
7995	-6.414403	0.723565	1.134953	2.952763	0.297928	-0.156946	2.398091	Bad
7996	0.851143	-2.217875	-2.812175	0.489249	-1.323410	-2.316883	2.113136	Bad
7997	1.422722	-1.907665	-2.532364	0.964976	-0.562375	-1.834766	0.697361	Bad
7998	-2.131904	-2.742600	-1.008029	2.126946	-0.802632	-3.580266	0.423569	Bad
7999	-2.660879	-2.044666	0.159026	1.499706	-1.581856	-1.605859	1.435644	Bad

Gambar 7 Membaca Data Frame

Langkah selanjutnya untuk mengetahui banyak nya sampel dan kolom pada datasets

```
[4] #Mengetahui ukuran data terdapat 8000 sample dan 8 kolom
    df.shape

    (8000, 8)
```

Gambar 8 Mengetahui banyak sampel dan kolom

Langkah selanjutnya dilakukan pengecekan *missing value* di setiap atribut, jika nilai nya sama dengan 0 atau nol maka tidak terdapat *missing value* sedangkan nilai sama dengan 1 maka terdapat *missing value.* Sebagai berikut.

```
#Menghitung jumlah missing values setiap kolom
df.isna().sum().sort_values(ascending=False)

Size          0
Weight        0
Sweetness     0
Softness      0
HarvestTime   0
Ripeness      0
Acidity       0
Quality       0
```

Gambar 9 Menghitung jumlah missing *values*

Langkah selanjutnya untuk mengetahui informasi setiap kolom dan penggunaan memori datasets.

```
[6] #Mengetahui info data setiap kolom
    df.info()

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 8000 entries, 0 to 7999
    Data columns (total 8 columns):
     #   Column       Non-Null Count  Dtype
    ---  ------       --------------  -----
     0   Size         8000 non-null   float64
     1   Weight       8000 non-null   float64
     2   Sweetness    8000 non-null   float64
     3   Softness     8000 non-null   float64
     4   HarvestTime  8000 non-null   float64
     5   Ripeness     8000 non-null   float64
     6   Acidity      8000 non-null   float64
     7   Quality      8000 non-null   object
    dtypes: float64(7), object(1)
    memory usage: 500.1+ KB
```

Gambar 10 Mengetahui informasi di setiap kolom

Langkah selanjutnya menampilkan deskripsi di setiap kolom data numerik dengan cara sebagai berikut.

```
[7] #Menampilkan statistik untuk setiap kolom data numerik
    df.describe()
```

	Size	Weight	Sweetness	Softness	HarvestTime	Ripeness	Acidity
count	8000.000000	8000.000000	8000.000000	8000.000000	8000.000000	8000.000000	8000.000000
mean	-0.747802	-0.761019	-0.770224	-0.014441	-0.751288	0.781098	0.008725
std	2.136023	2.015934	1.948455	2.065216	1.996661	2.114289	2.293467
min	-7.998074	-8.283002	-6.434022	-6.959320	-7.570008	-7.423155	-8.226977
25%	-2.277651	-2.223574	-2.107329	-1.590458	-2.120659	-0.574226	-1.629450
50%	-0.897514	-0.868659	-1.020673	0.202644	-0.934192	0.964952	0.098735
75%	0.654216	0.775491	0.311048	1.547120	0.507326	2.261650	1.682063
max	7.970800	5.679692	7.539374	8.241555	6.293280	7.249034	7.411633

Gambar 11 Menampilkan deskripsi pada kolom data numerik

Langkah selanjutnya untuk menyimpan fitur dan target kelas lalu melakukan eksplor pada variabel yang bertipe *int64*, yaitu *Quality* dan menampilkan jumlah dari banyaknya *Good* dan *Bad.* Jadi yang mendapat kualitas baik sebanyak 4006 dan mendapatkan kualitas yang buruk sebanyak 3994.

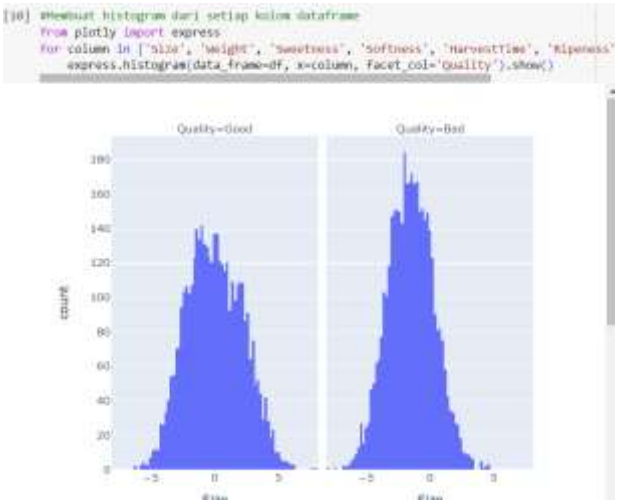
```
[8] #Menentukan kelas dan fitur
    #Variabel X menyimpan fitur dan variabel Y untuk menyimpan target kelas
    X = df.drop('Quality', axis = 1)
    Y = df['Quality']
```

```
#Explore variabel
df['Quality'].value_counts()
```

```
Quality
Good    4006
Bad     3994
Name: count, dtype: int64
```

Gambar 12 Menampilkan jumlah *Good* dan *Bad*

Langkah selanjutnya untuk menampilkan visualisasi histogram dari setiap kolom dengan cara *import module express* dari *library plotly*. Histogram ini terdiri *count* atau jumlah dan *quality* atau kualitas pada setiap kolom. Sebagai berikut :



Gambar 13 Menampilkan *Histogram*

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

Langkah selanjutnya mengubah label menjadi kode numerik dengan menggunakan *method replace()* untuk mengganti nilai label "*Good*" menjadi kode numerik "1" dan label "*Bad*" menjadi kode numerik "0"

```
[11] df['Quality'] = df['Quality'].replace({'Good': 1, 'Bad': 0})
     df['Quality'].value_counts()

     Quality
     1    4006
     0    3994
     Name: count, dtype: int64
```

Gambar 14 Mengubah label menjadi kode numerik

Langkah selanjutnya masukkan nilai data uji dan data latih dengan struktur kode sebagai berikut

```
[12] # Data latih 70% dan data uji 30%

     X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.3, random_state = 30)
```

Gambar 15 Contoh nilai data latih dan data uji

Langkah selanjutnya membuat struktur kode sesuai algoritma yang dipakai, pada penelitian ini menggunakan dua algoritma, yaitu algoritma *SVM* dengan kernel *RBF* dan algoritma *ANN* dengan arsitektur *MLP* untuk membandingkan algoritma mana yang akurat dan efisien dalam penelitian ini

```
#import svc classifier
from sklearn.svm import SVC

#import metrics
from sklearn.metrics import accuracy_score

#instansiasi classifier dengan default hyperparameters
svc=SVC()

#melatihan model dengan data latih
svc.fit(X_train, y_train)

#melatihan model dengan data uji
y_pred=svc.predict(X_test)

print('Akurasi model dengan default hyperparameter: {0:0.4f}'.format(accuracy_score(y_test, y_pred)))
```

```
Akurasi model dengan default hyperparameter: 0.9654
```

```
[14] #rbf
     #menjalankan SVM with rbf kernel and c=100
     rbf_svc=SVC(kernel='rbf', C=100.0)

     #melatihan model dengan data latih dan data uji
     rbf_svc.fit(X_train, y_train)
     y_pred=rbf_svc.predict(X_test)

     print('Akurasi model dengan rbf kernel dan C=100.0: {0:0.4f}'.format(accuracy_score(y_test, y_pred)))
```

```
Akurasi model dengan rbf kernel dan C=100.0: 0.9833
```

Gambar 16.1 Struktur kode algoritma *SVM (RBF)*

```
#mendefinisikan model ANN
clf = MLPClassifier (random_state= 30, max_iter=100).fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:686: ConvergenceWarning:

Stochastic Optimizer: Maximum iterations (100) reached and the optimization hasn't converged yet.
```

```
#melatih model ANN
clf.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:686: ConvergenceWarning:

Stochastic Optimizer: Maximum iterations (100) reached and the optimization hasn't converged yet.
```

```
MLPClassifier
MLPClassifier(max_iter=100, random_state=30)
```

```
[15] #melakukan prediksi kelas pada data uji
     y_predict_ann = clf.predict(X_test)
     y_predict_ann
```

```
array(['Good', 'Good', 'Bad', ..., 'Good', 'Good', 'Good'], dtype='<U4')
```

Gambar 16.2 Struktur kode algoritma *ANN (MLP)*

Langkah selanjutnya membuat laporan dan *confusion matrix*, sebagai berikut

```
[15] print(classification_report(y_test, y_pred))

     cm = confusion_matrix(y_test, y_pred)
     print('Confusion matrix\n\n', cm)

                  precision    recall  f1-score   support

             Bad       0.98      0.99      0.98      1210
            Good       0.99      0.98      0.98      1190

        accuracy                           0.98      2400
       macro avg       0.98      0.98      0.98      2400
    weighted avg       0.98      0.98      0.98      2400

    Confusion matrix

     [[1193   17]
     [  23 1167]]
```

Gambar 16.3 *Confusion matrix* dengan kernel *RBF*

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

cm = confusion_matrix(y_test, y_predict_ann)
print('Confusion matrix\n\n', cm)

              precision    recall  f1-score   support

         Bad       0.98      0.98      0.98      1210
        Good       0.98      0.98      0.98      1190

    accuracy                           0.98      2400
   macro avg       0.98      0.98      0.98      2400
weighted avg       0.98      0.98      0.98      2400

Confusion matrix

 [[1191   19]
 [  23 1167]]
```

Gambar 16.4 *Confusion matrix* dengan arsitekur *MLP*

Langkah selanjut nya membuat matriks korelasi yang berfungsi untuk merangkum kekuatan dan arah hubungan antara variable numerik dalam Kumpulan data

```
# correlation matrix
plt.figure(figsize=(15,8))
plt.title('correlation matrix')
sns.heatmap(df.corr(),annot=True)
```

Gambar 17.1 Struktur kode untuk menampilkan visualisasi matriks korelasi

Gambar 17.2 Tampilan visualisasi matriks korelas

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

3.3 Hasil

3.3.1 Hasil Algoritma SVM

Pengamatan 1 : Data latih 90% dan data uji 10 %

Tabel 1 Hasil evaluasi pengamatan 1 dengan kernel RBF

Kernel *RBF*				
Eksperimen	***accuracy***	***precision***	***recall***	***f1-score***
1	0,99	0,99	0,99	0,99
2	0,98	0,98	0,98	0,98
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	0,984	0,984	0,984	0,984

Pengamatan 2: Data latih 80% dan data uji 20 %

Tabel 2 Hasil evaluasi pengamatan 2 dengan kernel RBF

Kernel *RBF*				
Eksperimen	***accuracy***	***precision***	***recall***	***f1-score***
1	0,99	0,99	0,99	0,99
2	0,98	0,98	0,98	0,98
3	0,98	0,98	0,98	0,98
4	0,98	0,98	0,98	0,98
5	0,98	0,98	0,98	0,98
Rata rata	0,982	0,982	0,982	0,982

Pengamatan 3: Data latih 70% dan data uji 30 %

Tabel 3 Hasil evaluasi pengamatan 3 dengan kernel RBF

Kernel *RBF*				
Eksperimen	***accuracy***	***precision***	***recall***	***f1-score***
1	0,98	0,98	0,98	0,98
2	0,98	0,98	0,98	0,98
3	0,98	0,98	0,98	0,98
4	0,97	0,97	0,97	0,97
5	0,98	0,98	0,98	0,98
Rata rata	0,978	0,978	0,978	0,978

Pengamatan 4: Data latih 60% dan data uji 40 %

Tabel 4 Hasil evaluasi pengamatan 4 dengan kernel RBF

Kernel *RBF*				
Eksperimen	***accuracy***	***precision***	***recall***	***f1-score***
1	0,98	0,98	0,98	0,98
2	0,98	0,98	0,98	0,98
3	0,98	0,98	0,98	0,98
4	0,98	0,98	0,98	0,98
5	0,98	0,98	0,98	0,98
Rata rata	0,98	0,98	0,98	0,98

Tabel 5 Hasil Rata Rata SVM dengan kernel RBF

Kernel RBF					
Data latih	**Data uji**	***Accuracy***	***Presion***	***recall***	***Fl-score***
90%	10%	0,984	0,984	0,984	0,984
80%	20%	0,982	0,982	0,982	0,982
70%	30%	0,978	0,978	0,978	0,978
60%	40%	0,98	0,98	0,98	0,98
Rata rata		0,981	0,981	0,981	0,981

3.3.2 Hasil Algoritma ANN

Pengamatan 1: Data latih 90% dan data uji 10 %

Tabel 6 Hasil evaluasi pengamatan 1 dengan arsitektur MLP

Arsitektur MLP				
Eksperimen	***Accuracy***	***Precision***	***Recall***	***F1-score***
1	0,99	0,99	0,99	0,99
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	0,98	0,98	0,98	0,98
Rata rata	0,982	0,982	0,982	0,982

Pengamatan 2: Data latih 80% dan data uji 20 %

Tabel 7 Hasil evaluasi pengamatan 2 dengan arsitektur MLP

Arsitektur *MLP*				
Eksperimen	***Accuracy***	***Precision***	***Recall***	***F1-score***
1	0,98	0,98	0,99	0,98
2	0,98	0,98	0,98	0,98
3	0,98	0,98	0,98	0,98
4	0,97	0,97	0,97	0,97
5	0,98	0,98	0,98	0,98
Rata rata	0,978	0,978	0,98	0,978

Pengamatan 3: Data latih 70% dan data uji 30 %

Tabel 8 Hasil evaluasi pengamatan 3 dengan arsitektur MLP

Arsitektur *MLP*				
Eksperimen	***accuracy***	***precision***	***recall***	***f1-score***
1	0,98	0,98	0,98	0,98
2	0,98	0,98	0,98	0,98
3	0,98	0,98	0,98	0,98
4	0,97	0,97	0,97	0,97
5	0,98	0,98	0,98	0,98
Rata rata	0,978	0,978	0,978	0,978

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

Pengamatan 4: Data latih 60% dan data uji 40 %

Tabel 9 Hasil evaluasi pengamatan 4 dengan arsitektur MLP

Arsitektur *MLP*				
Eksperimen	***Accuracy***	***Precision***	***Recall***	***F1-score***
1	0,98	0,98	0,98	0,98
2	0,98	0,98	0,98	0,98
3	0,98	0,98	0,98	0,98
4	0,97	0,97	0,97	0,97
5	0,98	0,98	0,98	0,98
Rata rata	0,978	0,978	0,978	0,978

Tabel 10 Hasil Rata Rata *ANN* dengan Arsitektur *MLP*

Arsitektur *MLP*					
Data latih	**Data uji**	***Accuracy***	***Presion***	***Recall***	***Fl-score***
90%	10%	0,982	0,982	0,982	0,982
80%	20%	0,982	0,982	0,982	0,982
70%	30%	0,978	0,978	0,978	0,978
60%	40%	0,978	0,978	0,978	0,978
Rata rata		0,98	0,98	0,98	0,98

Tabel 11 Hasil Rata rata kedua algoritma

Algoritma	Rata rata			
	Precision	*Recall*	*F1-Score*	*Accuracy*
ANN (MLP)	0,98	0,98	0,98	0,98
SVM (RBF)	0,981	0,981	0,981	0,981

4. KESIMPULAN

Kesimpulan yang bisa didapatkan dari penelitian ini sebagai berikut

- Pada algoritma *SVM* dengan kernel *RBF* yang memiliki akurasi tertinggi terdapat di pengamatan 1 dan akurasi terendah pada pengamatan 3 dan 4. Sedangkan pada algoritma *ANN* dengan arsitektur *MLP* yang memiliki akurasi tertinggi terdapat di penganatan 1,2,3 dan akurasi terendah pada pengamatan 4
- Dari penelitian yang dilakukan dengan pengamatan sebanyak empat kali dan eksperimen yang dilakukan sebanyak lima kali serta data latih dan data uji yang bervariasi, bahwa *kernel RBF* konsisten yang lebih tinggi dari pada arsitektur *MLP*. Nilai rata rata yang diperoleh *kernel RBF* sebesar 0,981 lalu untuk arsitekur *MLP* memperoleh nilai rata rata sebesar 0,98.

Saran untuk mengembangkan analisis perbandingan Algoritma *Support Vector Machine (SVM)* dengan *kernel RBF* dan *Artificial Neural Network (ANN*) dengan arsitektur *MLP* adalah melakukan analisis lebih lanjut dengan mempertimbangkan faktor faktor lain untuk menentukan algoritma yang optimal secara absolut dan melibatkan lebih banyak lagi untuk eksperimen kedepannya

DAFTAR PUSTAKA

[1] Ashari, S. (2006). “Analisis Kandungan Gizi dan Energi Buah Pisang.” Jurnal Gizi dan Pangan, 12(2), 78-85.

[2] Sugiyono, Sugiono. (2015). Metode Penelitian Kuantitatif, Kualitatif, dan Kombinasi. Bandung:Alfabeta.

[3] Santosa, B. 2007. Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis. Graha Ilmu : Yogyakarta.

[4] Awad, M., & Khanna, R. (2015). *Efficient learning machines*: *Theories, concepts, and application for engineers and system designers. In Efficient learning Machines: Theories, Concepts, and Applications for Engineers and System Designers* (Issue May 2016). https://doi.org/10.1007/978-1-4302-5990-9

[5] Nielsen, M. (2015). *Neural Networks and Deep Learning*

[6] Du, R., Liu, W., & Jia, X. (2002). *Multi-layer perceptrons for pattern recognition. IEEE Transactions on Neural Network*, 13(2), 347-361

[7] Sihombing, P.R. dan Hendarsin, O.P. (2017). Perbandingan Metode *Artificial Neural Network (ANN)* dan *Support Vector Machine (SVM)* untuk Klasifikasi Kinerja Perusahaan Daerah Air Minum (PDAM) di Indonesia. Jurnal Ilmu Komputer VOL XIII No. 1: p-ISSN: 1979-5661 e-ISSN: 2622-321X.

[8] Siagin, R. Y., 2011. *Klasifikasi Parket Kayu Jati Menggunakan Metode Support Vector Machine (SVM)*. Jawa Barat, Skripsi, Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas

[9] Rachman, 2012. *Perbandingan Klasifikasi Tingkat Keganasan Breast Cancer dengan Menggunakan Regresi Logistik Ordinal dan Support Vector Machine. s.I., s.n.*

[10] Weiss, S.M., & Davidson, D. (2010). *Algorihtms and techniques for empirical knowledge discovery*. *Springer Science & Business Media*

[11] Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). *Extreme learning machine: theory and applications. Neurocomputing*, 70(1-3), 489-501

[12] Brownlee, J. (2016). Master Machine Learning Algorithms: discover how they work and implement them from scratch. Jason Brownlee.

[13] Wu, G., Kim, M., Wang, Q., Gao, Y., Liao, S., & Shen, D. (2013). *Unsupervised deep feature learning for deformable registration of MR brain images. International Conference on Medical Image Computing and Computer-Assisted Intervention*, 649-656

[14] Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*.Malaysia;Pearson Education Limited,.

[15] M. A. Nielsen, "*Neural Networks and Deep Learning." Determination Press*, 2015.

[16] A. *Intelligence, "Fundamentals of Neural Networks Artificial Intelligence Fundamentals of Neural Networks Artificial Intelligence*," pp. 6-102, 2010.

[17] Puspitaningrum, D. 2006. *Pengantar Jaringan Saraf Tiruan*. Andi Offset, Yogyakarta. [18] Fausset, L.V. 1994. *Fundamentals of Neural Networks: Architectures, Algorithms, andApplications*. Prentice Hall, New Jersey

[19] Purnomo, M.H. dan Kurniawan A. 2006. *Supervised Neural Network*. Graha Ilmu, Surabaya

[20] Alexander Amini, *Intro to Deep Learning*, MIT 6.S191, 2021