

Klasifikasi Jenis Kismis (Kecimen dan Besni) Menggunakan Model Multi-Layer Perceptron (MLP)

Abdullah Mubarok Maspeke¹ Lukman Nur Rahman²

^{1,2} Teknik Informatika STMIK Tazkia

¹241552010001.maspeke@student.stmik.tazkia.ac.id

²241552010007.lukman@student.stmik.tazkia.ac.id

Abstrak

Kepuasan pelanggan dalam industri pangan sangat bergantung pada kualitas produk yang stabil. Namun, proses pensortiran kismis secara manual sering tidak konsisten karena pekerja bisa mengalami kelelahan. Untuk mengatasi hal ini, kami mengembangkan sistem berbasis kecerdasan buatan yang bisa membedakan jenis kismis Kecimen dan Besni lewat ciri-ciri fisiknya.

Dengan memanfaatkan data citra seperti luas, keliling, dan bentuk objek, kami menggunakan metode Multi-Layer Perceptron (MLP) untuk mengenali pola pada data tersebut. Hasilnya menunjukkan bahwa sistem ini punya kemampuan yang sangat baik dalam tugas klasifikasi kismis. Ini membuktikan bahwa machine learning bisa menjadi solusi ampuh untuk kontrol kualitas otomatis, membuat produksi lebih efisien dan kualitas barang lebih terjamin."

Kata Kunci: klasifikasi kismis, kontrol kualitas, machine learning, multi-layer perceptron, kecerdasan buatan.

Abstract

Customer satisfaction in the food industry relies heavily on consistent product quality. However, manual raisin sorting is often inconsistent due to worker fatigue. To address this, we developed an artificial intelligence-based system to distinguish between Kecimen and Besni raisins based on their physical characteristics.

By utilizing image data such as area, perimeter, and object shape, we employed the Multi-Layer Perceptron (MLP) method to recognize patterns within the data. The results show that this system demonstrates excellent performance in the raisin classification task. This proves that machine learning can be an effective solution for automated quality control, making production more efficient and ensuring product quality.

Keywords: raisin classification, quality control, machine learning, multi-layer perceptron, artificial intelligence

1.Pendahuluan

1.1 Latar Belakang

Dalam dunia industri pangan modern, konsistensi kualitas adalah hal yang menentukan kepercayaan konsumen terhadap sebuah merek. Standar kualitas yang tinggi menuntut setiap produk yang sampai ke tangan pelanggan harus seragam dan bebas cacat. Namun, kenyataannya di sesi produksi seringkali menunjukkan tantangan berat seperti proses penyortiran komoditas pertanian, seperti kismis, masih sangat bergantung pada inspeksi visual manual. Padahal, keterbatasan fisik manusia seperti kelelahan mata dan hilangnya konsentrasi akibat pekerjaan berulang sering menjadi penyebab utama inkonsistensi dalam pengendalian kualitas (Cinar, Koklu, & Tasdemir, 2020).

Artificial Intelligence (AI) hadir sebagai solusi yang baru. Penelitian ini mengusulkan penerapan metode Multi-Layer Perceptron (MLP), sebuah arsitektur Jaringan Syaraf Tiruan (Artificial Neural Network), untuk menangani tugas klasifikasi ini.

Penelitian sebelumnya yang dilakukan oleh Cinar et al. (2020) telah membuktikan bahwa pendekatan berbasis Machine Learning mampu memberikan hasil yang menjanjikan menggunakan Raisin Dataset. Melanjutkan fondasi tersebut, penelitian ini bertujuan untuk mengimplementasikan dan mengevaluasi performa model MLP dalam membedakan varietas Kecimen dan Besni.

Harapannya, sistem ini dapat menjadi bantuan bagi industri pertanian dalam mengubah proses penyortiran yang subjektif dan melelahkan menjadi sistem deteksi otomatis yang cepat, akurat, dan dapat diandalkan. Seluruh proses, dari data mentah hingga menjadi sebuah model yang berfungsi, akan kami coba sampaikan secara bertahap dalam laporan penelitian ini.

1.2 Rumusan Masalah

- Bagaimana cara mengimplementasikan model Multi-Layer Perceptron (MLP) untuk mengklasifikasikan jenis kismis (Kecimen vs. Besni) menggunakan 7 fitur morfologis yang tersedia?
- Bagaimana performa model MLP ini dibandingkan dengan hasil akurasi yang dilaporkan dalam penelitian terdahulu (Cinar et al., 2020)?

1.3 Tujuan Penelitian

- Membangun model klasifikasi biner menggunakan Multi-Layer Perceptron (MLP) untuk membedakan antara kismis jenis Kecimen dan Besni.
- Mengevaluasi performa model menggunakan metrik standar.

2. Metodologi

2.1. Sumber dan Modifikasi Data

Sumber penelitian ini menggunakan data sekunder yang tersedia untuk publik yang bersumber dari "Raisin Dataset" di UCI Machine Learning Repository. Dataset ini berisi data pengukuran fisik dari butiran kismis, termasuk luas area, keliling, panjang sumbu utama, dan kelengkungan. Dengan tujuan agar dapat diproses oleh komputer, kami melakukan sedikit modifikasi pada data asli. Data aslinya memuat kolom "Class" yang berisi teks "Kecimen" dan "Besni". Karena komputer tidak dapat membaca teks secara langsung untuk dihitung, kami mengubah kolom ini menjadi angka. Ini penting supaya mesin bisa membedakan mana kismis jenis A dan jenis B tanpa kebingungan.

Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perimeter	Class
87524	442,2460114	253,291155	0,819738392	90546	0,758650579	1184,04	Kecimen
75166	406,690687	243,0324363	0,801805234	78789	0,68412957	1121,786	Kecimen
90856	442,2670483	266,3283177	0,798353619	93717	0,637612812	1208,575	Kecimen
45928	286,5405586	208,7600423	0,684989217	47336	0,699599385	844,162	Kecimen
79408	352,1907699	290,8275329	0,56401133	81463	0,792771926	1073,251	Kecimen
49242	318,125407	200,12212	0,777351277	51368	0,658456354	881,836	Kecimen
42492	310,1460715	176,1314494	0,823098681	43904	0,665893562	823,796	Kecimen
60952	332,4554716	235,429835	0,706057518	62329	0,74359819	933,366	Kecimen
42256	323,1896072	172,5759261	0,845498789	44743	0,698030924	849,728	Kecimen
64380	366,9648423	227,7716147	0,784055626	66125	0,664375716	981,544	Kecimen
80437	449,4545811	232,3255064	0,856042518	84460	0,674235757	1176,305	Kecimen
43725	301,3222176	186,9506295	0,784258452	45021	0,697068248	818,873	Kecimen
43441	276,6108288	201,8131355	0,683882337	45133	0,690855598	803,748	Kecimen

Gambar 1: Potongan Dataset " Raisin Dataset "

Sumber data ini memberikan total 900 butir kismis sebagai sampel. Jumlah ini termasuk cukup banyak, tapi jika dipikirkan kembali mungkin masih kurang banyak untuk mewakili jutaan kismis di dunia nyata. Ada juga kekhawatiran jika data ini diambil dari satu panen saja, sehingga mungkin hasilnya berbeda jika dipakai untuk kismis dari kebun lain. Data yang dipakai hanya 7 fitur fisik (bentuk dan ukuran) mungkin ada faktor lain seperti warna atau tekstur kulit yang juga berpengaruh ke jenisnya tapi tidak ada di data ini. Hal-hal seperti kualitas kamera saat memfoto kismisnya juga dapat berpengaruh tidak menutup kemungkinan ada bias di datanya. Karena itu, penelitian dimasa depan sebaiknya menambahkan variasi data dari berbagai sumber agar modelnya semakin baik.

2.2. Persiapan Data

Sebelum data diproses oleh model *computer* data tersebut perlu dibersihkan dan disiapkan terlebih dahulu. Dalam kode program kami, kami melakukan beberapa hal:

- Mengubah Label Jadi Angka (Encoding)
Komputer lebih memahami angka. Sehingga, label Besni dan Kecimen diubah menjadi angka 0 dan 1 dengan alat bernama LabelEncoder. Ini memudahkan komputer menghitung probabilitasnya nanti.
- Membuat Standar Skala (Scaling)
Data kismis ini mempunyai rentang angka yang jauh. Misalnya dalam kolom Area angkanya dapat puluhan ribu, tapi dalam kolom Eccentricity angkanya berupa nol koma sekian. Jika langsung dimasukan komputer akan mengira Area itu jauh lebih penting karena angkanya besar. Maka dari itu semua data numerik disamakan skalanya dengan StandardScaler.

2.3. Cara Kerja Model

Algoritma utama yang kami gunakan adalah *Multi-Layer Perceptron* (MLP). Multi Layer Perceptron (MLP) adalah sebuah imitasi dari jaringan saraf (Dahman, 2021). Di model kami terdapat lapisan input (data masuk), dua lapisan tersembunyi (*hidden layer*) yang isinya masing-masing 128 dan 64 neuron. Setiap data kismis yang masuk akan dipindahkan antar neuron ini. Setiap neuron akan memberi bobot (nilai penting) ke data itu.

Sama halnya dengan belajar, model ini diajari dengan melihat data latih. Dia akan menebak, "Ini Besni!". Jika salah, dia akan dikoreksi dan dia akan mundur untuk memperbaiki bobot-bobot di sarafnya atau belajar dari error. Proses ini diulang sampe 2000 kali sesuai kodingan *max_iter* sampai dia mampu membedkan pola bentuk kismisnya.

2.4. Pengujian Model

Teknik yang kami terapkan untuk mengetahui kemampuan model kami termasuk umum dalam *machine learning*. Kami membagi seluruh data yang kami miliki (900 data kismis) menjadi dua bagian:

1. **Data Latihan (80%)**

Data ini diberikan kepada model untuk belajar. Model akan mempelajari semua pola dari data ini.

2. **Data Ujian (20%)**

Data ini dirahasiakan dari model selama proses belajar. Setelah model selesai

belajar, kami mengujinya dengan data ini untuk melihat seberapa baik performanya pada data yang belum pernah ia lihat sebelumnya.

Keberhasilan model diukur dengan metrik akurasi khusus, yaitu persentase tebakan benar yang berhasil dibuat oleh model pada data ujian.

2.5. Tools

Pembuatan model ini memakai beberapa hal yang penting diantaranya:

1. Python

Kami menggunakan Python untuk menulis semua perintah yang harus dijalankan oleh komputer. Bahasa ini dipilih karena populer dan punya banyak fitur siap pakai yang mempercepat pekerjaan. Selain itu bahsa ini sudah kami pelajari di kelas sehingga lebih familiar.

2. Kode Editor (VS Code & Jupyter)

Sebagian besar proses kami kerjakan di VS Code. Ini adalah sebuah lembar kerja digital di mana kami bisa menulis perintah, menjalankan perintah itu, dan langsung melihat hasilnya, baik berupa teks maupun gambar, di satu tempat. Ini juga sangat disarankan oleh dosen kami.

3. Alat Bantu Tambahan (Library Python)

Kami juga memakai beberapa alat bantu tambahan untuk tugas-tugas spesifik agar tidak menulis dari awal:

- scikit-learn:
Ini adalah alat utama untuk membuat program kecerdasan buatannya. Di sinilah semua proses belajarnya si komputer terjadi.
- pandas:
Alat ini dipakai untuk mengelola data. Tugasnya adalah membaca dan merapikan tabel data agar siap untuk dianalisis.
- numpy:
Berfungsi sebagai kalkulator canggih untuk menangani semua proses hitung-hitungan yang berat.
- matplotlib & seaborn:
Ini adalah alat untuk menggambar. Kami menggunakannya untuk mengubah data angka menjadi grafik yang lebih mudah dilihat dan dipahami.

3. Hasil Dan Pembahasan

Pada bab ini, kami akan memberikan pembahasan hasil dari eksperimen klasifikasi kismis Kecimen dan Besni. Analisis difokuskan pada kinerja model Multi-Layer Perceptron (MLP) Clasifier yang telah dilatih menggunakan data fitur morfologis kismis. Sebelum mendapatkan model terbaik, kami melakukan beberapa tahap pengujian untuk mencari pola yang cocok untuk arsitektur jaringan sarafnya.

3.1. Eksperimen Optimasi Mencari Arsitektur Terbaik

Sebelum menetapkan arsitektur yang baik, penelitian ini melalui proses trial and error (penyetelan hyperparameter). Tujuannya adalah untuk menemukan konfigurasi yang optimal dan stabil dalam memisahkan kedua jenis kismis. Kami menguji tiga konfigurasi utama dengan pemahaman bahwa perubahan kecil pada arsitektur atau parameter dapat menyebabkan fluktuasi kinerja yang signifikan. Hasil evaluasi menunjukkan perbedaan performa yang signifikan antara ketiga percobaan:

- Percobaan pertama, percobaan dengan “hidden_layer_sizes=(200, 100, 50),” dan “max_iter=5000,” meskipun terhenti di iterasi ke 300 hasil percobaan ini memberikan Akurasi Model MLP: 82.22%.

```

39 mlp = MLPClassifier(
40     hidden_layer_sizes=(200, 100, 50),
41     max_iter=5000,
42     activation='relu',
43     solver='adam',
44     alpha=0.0001,
45     random_state=42,
46     tol=1e-4,
47 )

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Memulai Pelatihan Model MLP...

Iteration 1, loss = 0.63573066
 Iteration 2, loss = 0.53019201
 Iteration 3, loss = 0.45030201
 Iteration 4, loss = 0.38671539
 Iteration 5, loss = 0.34902242
 Iteration 6, loss = 0.33614027
 Iteration 7, loss = 0.33554255
 Iteration 8, loss = 0.33406769
 Iteration 9, loss = 0.33102409
 Iteration 10, loss = 0.32713835
 Iteration 11, loss = 0.32317894
 Iteration 12, loss = 0.32153644
 Iteration 13, loss = 0.31934951
 Iteration 14, loss = 0.31744632
 Iteration 15, loss = 0.31590316
 Iteration 16, loss = 0.31433314
 Iteration 17, loss = 0.31262901
 Iteration 18, loss = 0.31069228

Gambar 2: Potongan Hasil/output Percobaan 1

```

Iteration 300, loss = 0.12684900
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Pelatihan Selesai. :)

~~~~~
>\< Akurasi Model MLP: 82.22% >\<
~~~~~

Laporan Klasifikasi
      precision    recall   f1-score   support
      Besni       0.82      0.83      0.82       90
      Kecimen     0.83      0.81      0.82       90

      accuracy          0.82      0.82      0.82      180
      macro avg       0.82      0.82      0.82      180
      weighted avg    0.82      0.82      0.82      180

Membuat Visualisasi Grafik...

```

Gambar 3: Potongan Hasil/output Percobaan 1

- Percobaan kedua, percobaan dengan “hidden_layer_sizes=(100, 50),” dan “max_iter=1000,” meskipun terhenti di iterasi ke 676 hasil percobaan ini memberikan Akurasi Model MLP: 87.22%. Meningkat dibandingkan percobaan pertama.

```

38
39 mlp = MLPClassifier(
40     hidden_layer_sizes=(100, 50),
41     max_iter=1000,
42     activation='relu',
43     solver='adam',
44     alpha=0.0001,
45     random_state=42,
46     tol=1e-4,
47     verbose=True
48 )
49
50 print("Memulai Pelatihan Model MLP...")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\Heaven Path\Tazkia\tugas tazkia\pak hendri\Raisin_Dataset\kelompok_1_machine-learning> & C:/Users/lulocal/Microsoft/WindowsApps/python3.11.exe "d:/Heaven Path/Tazkia/tugas tazkia/pak hendri/Raisin_Dataset/keIne-learning/mlp.py"
Iteration 675, loss = 0.15686872
Iteration 676, loss = 0.15667082
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Pelatihan Selesai. :)

~~~~~
>\< Akurasi Model MLP: 87.22% >\<
~~~~~
```

Laporan Klasifikasi				
	precision	recall	f1-score	support
Besni	0.95	0.79	0.86	90
Kecimen	0.82	0.96	0.88	90
accuracy			0.87	180
macro avg	0.88	0.87	0.87	180
weighted avg	0.88	0.87	0.87	180

Gambar 4: Potongan Hasil/output Percobaan 2

- Percobaan ketiga/final, percobaan dengan “hidden_layer_sizes=(128, 64),” dan “max_iter=500,” meskipun terhenti di iterasi ke 471 hasil percobaan ini memberikan Akurasi Model MLP: 87.78%. Meningkat dibandingkan percobaan pertama dan kedua.

```

36     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
37 )
38
39 mlp = MLPClassifier(
40     hidden_layer_sizes=(128, 64),
41     max_iter=500,
42     activation='relu',
43     solver='adam',
44     alpha=0.0001,
45     random_state=42,
46     tol=1e-4,

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\Heaven Path\Tazkia\tugas tazkia\pak hendri\Raisin_Dataset\kelompok_1_machine-learning> & C:/U
3.11.exe "d:/Heaven Path/Tazkia/tugas tazkia/pak hendri/Raisin_Dataset/kelompok_1_machine-learning/
Iteration 398, loss = 0.19995386
Iteration 399, loss = 0.19917471
Iteration 400, loss = 0.19924135
Iteration 401, loss = 0.20239829
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Pelatihan Selesai. :)

~~~~~
>\< Akurasi Model MLP: 87.78% >\<
~~~~~
```

Laporan Klasifikasi				
	precision	recall	f1-score	support
Besni	0.96	0.79	0.87	90
Kecimen	0.82	0.97	0.89	90
accuracy			0.88	180
macro avg	0.89	0.88	0.88	180
weighted avg	0.89	0.88	0.88	180

Gambar 5: Potongan Hasil/output Percobaan 3/final

3.2. Analisis Kinerja Model

Demi melihat dan mendapatkan perilaku model terbaik pada setiap jenis kismis (Besni dan Kecimen). Percobaan ketiga dengan arsitektur (128, 64) dan max_iter=500 ditetapkan sebagai model final karena menghasilkan Akurasi tertinggi 87.78%.

Akurasi dan Laporan Klasifikasi Percobaan ketiga. Akurasi Model Final: 87.78%. Akurasi ini menunjukkan bahwa model berhasil mengklasifikasikan 158 kismis dengan benar dari total 180 sampel uji sekitar ($158/180 = 0.8778\$$).

```

PS D:\Heaven Path\Tazkia\tugas tazkia\pak hendri\Raisin_Dataset\kelompok_1_machine-learning> & C:/U
3.11.exe "d:/Heaven Path/Tazkia/tugas tazkia/pak hendri/Raisin_Dataset/kelompok_1_machine-learning/
Iteration 398, loss = 0.19995386
Iteration 399, loss = 0.19917471
Iteration 400, loss = 0.19924135
Iteration 401, loss = 0.20239829
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
Pelatihan Selesai. :)

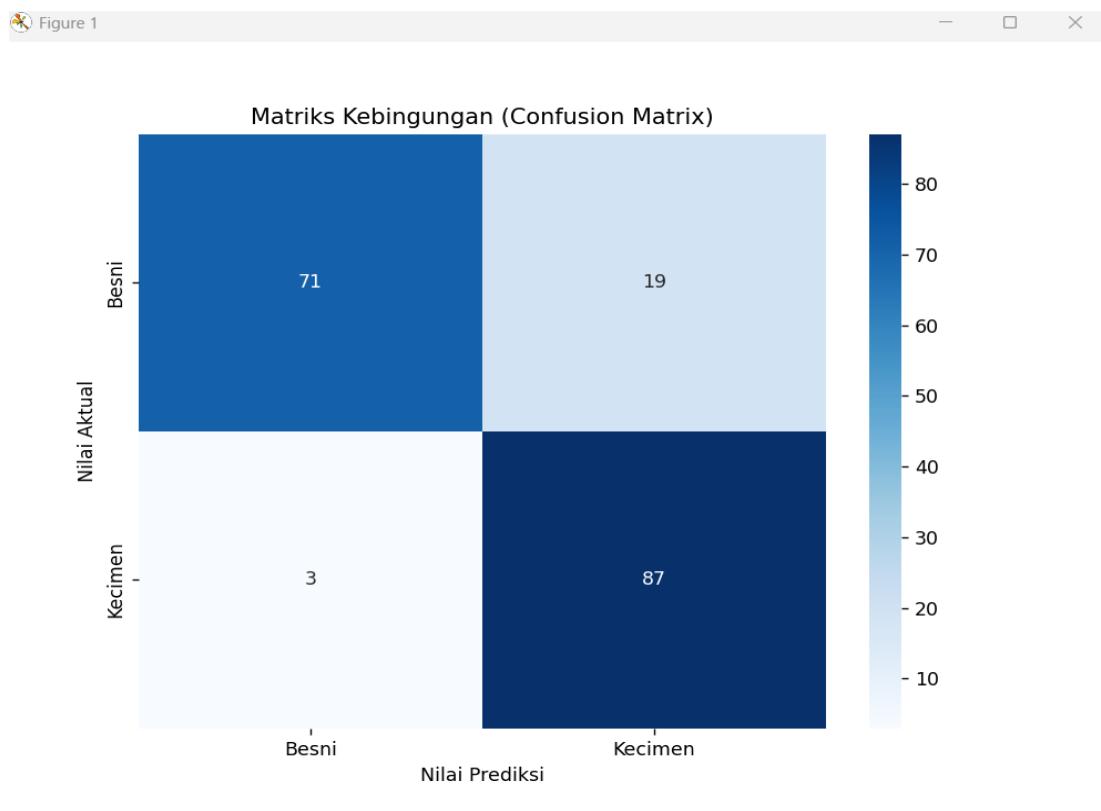
~~~~~
>\< Akurasi Model MLP: 87.78% >\<
~~~~~
```

Laporan Klasifikasi				
	precision	recall	f1-score	support
Besni	0.96	0.79	0.87	90
Kecimen	0.82	0.97	0.89	90
accuracy			0.88	180
macro avg	0.89	0.88	0.88	180
weighted avg	0.89	0.88	0.88	180

Gambar 6: Potongan Hasil/output Percobaan ketiga dengan arsitektur (128, 64) dan max_iter=500

3.3 Analisis Kesalahan (Confusion Matrix)

Berdasarkan Classification Report, kita dapat menyusun Matriks Kebingungan (Confusion Matrix) untuk menjelaskan sumber kesalahan model.



Gambar 7: Hasil/output Grafik model final dengan arsitektur (128, 64) dan max_iter=500

Matriks Kebingungan (Confusion Matrix) untuk hasil akurasi 87.78% dapat dihitung sebagai berikut:

- False Positives (FP = 19): Sebanyak 19 kismis Besni salah diklasifikasikan sebagai Kecimen.
- False Negatives (FN = 3): Sebanyak 3 kismis Kecimen salah diklasifikasikan sebagai Besni.

Tingginya angka *False Positives* (19) merupakan penyebab utama mengapa Akurasi tidak lebih dari 87.78%. Ini juga menunjukkan bahwa beberapa kismis Besni memiliki kemiripan fitur yang signifikan dengan Kecimen yang dapat mengecoh model.

Masalah Utama dalam model ini adaahaa ia cenderung lebih sering melakukan kesalahan Tipe I (False Positive), di mana 19 butir kismis Besni salah diklasifikasikan sebagai Kecimen. Ini terlihat dari nilai Recall Besni yang rendah (0.79) dan Precision Kecimen yang lebih rendah (0.82) dibandingkan Bedni.

3.3. Analisis Proses Pelatihan (Loss Curve dan Konvergensi)

Proses pelatihan pada model final/percobaan 3 menunjukkan bagaimana *loss* (kerugian/kesalahan) menurun seiring berjalannya iterasi. Berikut poin pentingnya:

- Iterasi Terbaik dalam model ini terhenti secara otomatis pada Iterasi ke-401 dari max_iter 500.
- Penghentian dalam model ini berhenti karena *Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.* Artinya Kerugian pelatihan tidak membaik lebih dari toleransi 0.0001 selama 10 epoch berturut-turut.
- Penghentian ini adalah fungsi Early Stopping yang efektif. Itu menandakan bahwa pada Iterasi ke-401, model sudah mencapai titik elajar maksimalnya. Melanjutkan iterasi tidak akan secara signifikan meningkatkan akurasi dan berisiko menyebabkan *Overfitting*.
- Grafik gamabr di atas menjelaskan bahwa proses pelatihan berjalan baik, namun menunjukkan juga bahwa batas kemampuan model dengan konfigurasi (128, 64) dan iterasi 500 pada *dataset* saat ini adalah 87.78%.

Hasil penelitian ini menunjukkan bahwa model *Multi-Layer Perceptron (MLP)*dengan akurasi 87.78% karena menggunakan model dengan konfigurasi (128, 64) dan iterasi 500 memiliki potensi sebagai teknologi pendukung/ bantuan bagi industri pertanian dalam mengubah proses penyortiran yang subjektif dan melelahkan menjadi sistem deteksi otomatis yang cepat, akurat, dan dapat diandalkan. Namun, analisis kinerja mengungkap akurasi hanya 87.78% yang masih memiliki ruang pengembangan kedepannya. Komposisi ygang cocok dapat membawa model ini menuju akurasi yang lebih besar lagi.

3.4 Perbandingan Kinerja dengan Studi Acuan/ Paper Referensi

Untuk mengevaluasi penelitian ini maka perbandingan dilakukan dengan hasil model yang dilaporkan dalam studi acuan/ paper referensi yang menggunakan *dataset* yang sama. Perbandingan ini memberikan dua temuan penting:

1. MLP Lebih Unggul
Model MLP yang dikembangkan dalam penelitian ini memberikan akurasi 87.78% berhasil mengungguli model MLP yang dilaporkan dalam studi acuan/ paper referensi yang memberikan akurasi 86.33%. Peningkatan kinerja ini menunjukkan bahwa penggunaan fungsi aktivasi ReLU dan algoritma *solver* Adam pada arsitektur yang lebih ramping (128, 64) lebih efektif dalam mempelajari pola fitur morfologis kismis. Ini juga menunjukkan pentingnya komposisi dari sebuah model.
2. Melampaui Metode Terbaik dalam Studi Acuan
Akurasi 87.78% dalam model yang dikembangkan ini juga melampaui akurasi tertinggi

yang dicapai oleh metode lain dalam studi acuan, yaitu akurasi model SVM dengan 86.44%. Hal ini memvalidasi bahwa setelah proses *tuning* yang tepat, model *Multi-Layer Perceptron* adalah metode klasifikasi yang paling efektif untuk *dataset* kismis ini.

3.4. kode python

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

file_id = "Raisin.csv"
try:
    df = pd.read_csv(file_id)
except FileNotFoundError:
    print(f"Error: File {file_id} not found.")
    exit()

print("--- Data Awal ---")
print(df.head(min(901, len(df))))
print("\n")

X = df.drop('Class', axis=1)
y = df['Class']

le = LabelEncoder()
y_encoded = le.fit_transform(y)
print("Label Encoding Kelas:")
for i, name in enumerate(le.classes_):
    print(f" {name}: {i}")
print("\n")

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y_encoded, test_size=0.2, random_state=42, stratify=y_encoded
)
```

```
mlp = MLPClassifier(  
    hidden_layer_sizes=(128, 64),  
    max_iter=500,  
    activation='relu',  
    solver='adam',  
    alpha=0.0001,  
    random_state=42,  
    tol=1e-4,  
    verbose=True  
)  
  
print("Memulai Pelatihan Model MLP...")  
mlp.fit(X_train, y_train)  
print("Pelatihan Selesai. :)\n")  
  
y_pred = mlp.predict(X_test)  
  
accuracy = accuracy_score(y_test, y_pred) * 100  
  
print("~" * 40)  
print(F">\\< Akurasi Model MLP: {accuracy:.2f} % >\\<")  
print("~" * 40 + "\n")  
  
print(" Laporan Klasifikasi ")  
print(classification_report(y_test, y_pred, target_names=le.classes_))  
  
  
print("Membuat Visualisasi Grafik...")  
cm = confusion_matrix(y_test, y_pred)  
class_names = le.classes_  
  
plt.figure(figsize=(8, 6))  
sns.heatmap(  
    cm,  
    annot=True,  
    fmt='d',  
    cmap='Blues',  
    xticklabels=class_names,  
    yticklabels=class_names  
)  
plt.title('Matriks Kebingungan (Confusion Matrix)')  
plt.ylabel('Nilai Aktual')  
plt.xlabel('Nilai Prediksi')
```

```
plt.show()
```

4. Penutup

4.1 Kesimpulan

Berdasarkan hasil eksperimen dan analisis kinerja model klasifikasi kismis Kecimen dan Besni menggunakan Multi-Layer Perceptron (MLP) diperoleh beberapa kesimpulan utama:

- Akurasi Klasifikasi Tercapai
Model MLP dengan arsitektur optimal (128, 64) berhasil mengklasifikasikan varietas kismis pada data uji dengan akurasi tertinggi sebesar 87.78%. Angka ini menunjukkan bahwa model ini yang dirancang ini mampu membedakan kedua jenis kismis tersebut dengan tingkat kebenaran yang baik.
- Optimalisasi Arsitektur Berhasil
Proses trial and error (Percobaan 1, 2, dan 3) membuktikan bahwa penyesuaian hyperparameter dan arsitektur jaringan saraf (dari 82.22% ke 87.78%) sangat penting dalam meningkatkan performa model. Model mencapai titik pada Iterasi ke-401 yang menunjukkan proses pelatihan berjalan efektif.
- Identifikasi Sumber Kesalahan
Analisis kinerja menemukan adanya kelemahan signifikan pada kemampuan model dalam mendeteksi kismis Besni dengan Recall hanya 0.79. Mayoritas kesalahan prediksi 19 kasus adalah False Positives Dimana kismis Besni salah dikira Kecimen). Hal ini mengindikasikan bahwa fitur dari beberapa kismis Besni memiliki kemiripan yang tinggi dengan kismis Kecimen, sehingga mengecoh model.
- Model MLP yang Lebih Unggul Dibandingkan dengan Studi Acuan/ Paper Referensi
Model MLP yang dikembangkan dalam penelitian ini memberikan akurasi 87.78% berhasil mengungguli model MLP yang dilaporkan dalam studi acuan/ paper referensi yang memberikan akurasi 86.33%.
- Melampaui Metode Terbaik dalam Studi Acuan/ paper referensi
Akurasi 87.78% dalam model yang dikembangkan ini juga melampaui akurasi tertinggi yang dicapai oleh metode lain dalam studi acuan, yaitu akurasi model SVM dengan 86.44%.

4.2 Saran

Meskipun model telah mencapai akurasi 87.78% terdapat ruang untuk perbaikan guna mencapai tingkat akurasi yang lebih tinggi dan stabilitas yang lebih baik, terutama untuk aplikasi quality control di tingkatan industri. Sebagaimana model ini dapat mengungguli model terbaik dalam studi acuan/ paper referensi. Saran untuk penelitian selanjutnya adalah Pengujian Arsitektur

Lanjutan dan Penggunaan Model Alternatif seperti *Support Vector Machine* (SVM) atau *Random Forest*.

Ucapan Terima Kasih

Dengan penuh rasa terima kasih dan rasa hormat, kami menyampaikan apresiasi kepada Bapak Hendri Karisma, S.Kom., M.T. atas segala pengajaran, bimbingan, motivasi, serta saran yang telah membantu dalam penyusunan penelitian ini dan selama pembelajaran. Penulis juga menyampaikan rasa terima kasih dan rasa hormatnya kepada pihak penyedia dataset melalui platform archive.ics.uci.edu yang telah memfasilitasi tersedianya data publik sebagai dasar penelitian ini. Penulis juga menyampaikan rasa terima kasih ke semua pihak yang telah menyediakan artikel online yang sangat membantu dalam penyusunan penelitian ini.

Daftar Pustaka

Cinar, I., Koklu, M., & Tasdemir, S. (2020). Classification of raisin grains using machine vision and artificial intelligence methods. *Gazi Journal of Engineering Sciences*, 6(3), 200–209. <https://dergipark.org.tr/tr/download/article-file/1227592>

Cinar, I., Koklu, M., & Tasdemir, S. (2020). Raisin [Data set]. UCI Machine Learning Repository. <https://archive.ics.uci.edu/dataset/850/raisin>

Dahman W., D. (2021). Multi Layer Perceptron. Medium. <https://medium.com/sysinfo/multi-layer-perceptron-6ccaae0dcc8>

Mubarok Maspeke, A., & Nur Rahman, L. (2024). Proposal: Klasifikasi jenis kismis (Kecimen dan Besni) menggunakan model Multi-Layer Perceptron (MLP). Sekolah Tinggi Manajemen Informatika dan Komputer Tazkia. https://docs.google.com/document/d/17sDLy8k7twdD-8YO3RJXZJlxI9H0_cclU2xIHtMmCec/edit?tab=t.0#heading=h.2otvilyiemv0