

# Klasifikasi Kematangan Melinjo (*Gnetum Gnemon Linn.*) Berdasarkan Citra HSV dengan *K-Nearest Neighbors*

Marella Azaria Putri\*, Jayanta, Neny Rosmawarni  
Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional  
Veteran Jakarta, Indonesia

\*E-mail koresponden: [marellaazariaputri@gmail.com](mailto:marellaazariaputri@gmail.com)

*Diserahkan* 15 Desember 2023; *Direview* 2 Januari 2024; *Dipublikasikan* 8 Januari 2024

## Abstrak

*Penyortiran buah melinjo yang selama ini dilakukan menjadi suatu masalah baru setelah buah melinjo berhasil dipanen. Hal ini disebabkan karena penentuan kematangan buah melinjo hanya dilakukan dengan cara melakukan pengamatan terhadap perubahan warna kulit buah melinjo secara manual dan menggunakan pandangan masyarakat sehari-hari. Sedangkan, masyarakat memiliki perbedaan pandangan dalam mengidentifikasi fase kematangan pada buah melinjo dan keakuratan dalam menghasilkan klasifikasi kematangan buah tidak bisa dianggap sah. Penelitian ini dapat membantu masyarakat dalam mengidentifikasi fase kematangan buah melinjo dalam bentuk citra digital menggunakan ekstraksi warna HSV (Hue, Saturation, Value) dengan algoritma KNN (k-Nearest Neighbor). Sebanyak 108 citra digunakan sebagai dataset yang terdiri dari tiga kelas yaitu 36 citra mentah, 36 citra setengah matang, dan 36 citra matang. Hasil penelitian yang ini menggunakan algoritma KNN, didapatkan akurasi dan classification report terbaik sebesar 100% dengan  $k = 2$  dan rata-rata akurasi yang didapatkan sebesar 97,56%.*

**Kata kunci:** Buah Melinjo; HSV; Kematangan Buah; Klasifikasi; KNN

## Abstract

*The sorting of melinjo fruit that has been carried out until now has become a new problem after the melinjo fruit has been successfully harvested. This is because determining the maturity of melinjo fruit is only performed by manually observing changes in the color of the melinjo fruit skin and using people's everyday views. Meanwhile, people have different views on identifying the maturity phase of melinjo fruit, and the accuracy in producing a classification of fruit maturity cannot be considered valid. This research can help the public identify the maturity phase of melinjo fruit in the form of digital images using Hue, Saturation, Value (HSV) color extraction with the K-Nearest Neighbor (KNN) algorithm. A total of 108 images were used as a dataset consisting of three classes, namely 36 raw images, 36 half-cooked images, and 36 mature images. The results of this research using the KNN algorithm, obtained the best accuracy and classification report of 100% using  $k = 2$  and the average accuracy obtained was 97.56%.*

**Keywords:** Classification; Fruit Maturity; HSV; KNN; Melinjo Fruit

## PENDAHULUAN

Melinjo (*Gnetum gnemon* Linn.) yang sering disebut Tangkil dalam Bahasa Sunda. Jenis tanaman yang asalnya dari Asia tropik, Melanesia, dan Pasifik Barat ini merupakan golongan berbiji terbuka (*Gymnospermae*). Selain sering disebut sebagai Tangkil, buah ini memiliki sebutan lain seperti *belinjo*, *mlinjo* (bahasa Jawa), *bago* (bahasa Melayu serta Tagalog), dan *Khalet* (Bahasa Kamboja). Tanaman melinjo sering dibudidayakan di pekarangan dan dimanfaatkan buahnya oleh masyarakat. Melinjo mampu tumbuh di wilayah yang memiliki iklim kering hingga tropis atau kondisi cuaca khusus. Melinjo mampu menyesuaikan diri dengan berbagai variasi suhu, sehingga di beberapa daerah, melinjo sangat mudah dijumpai. Namun, lingkungan dengan kadar garam yang tinggi seperti pantai, tidak cocok bagi pertumbuhan melinjo [1].

Terlepas dari pandangan buruk antara buah melinjo dan penyakit asam urat, banyak penelitian yang mengungkapkan salah satu tanaman yang dapat dimanfaatkan khasiatnya yakni tanaman melinjo. Bakteri seperti *Bacillus cereus* ATCC 10876, *Staphylococcus aureus* ATCC 25953, *Enterobacter aerogenes* ATCC 13048 mampu dibasmi dengan memanfaatkan ekstrak dari buah melinjo [2]. Buah melinjo berkhasiat sebagai antihiperlipidemia dengan mengurangi kadar glukosa dalam darah [3]. Biji melinjo terbukti memiliki senyawa antimikroba untuk menghambat pertumbuhan mikroorganisme makanan dan enterobakteri [4].

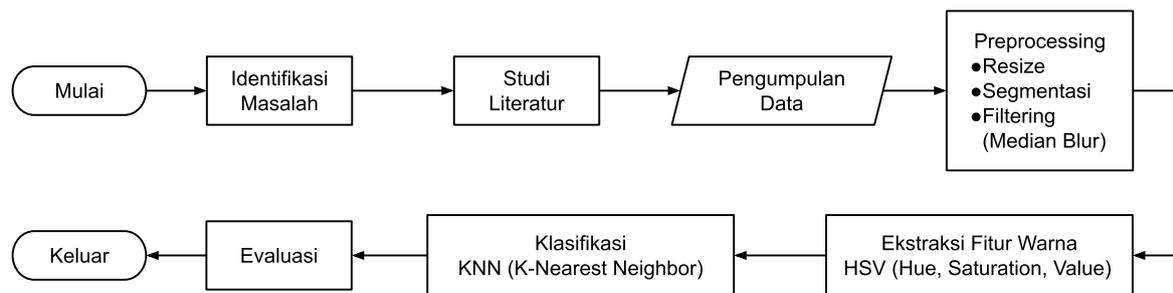
Fase kematangan pada buah melinjo ditandai dengan warna kulit buah hijau yang berarti mentah, kuning yang berarti setengah matang, dan merah yang berarti matang [5]. Penelitian klasifikasi kualitas buah melinjo menggunakan *k-Nearest Neighbor* (KNN) dengan ekstraksi RGB didapatkan akurasi sebesar 93% [6]. Penelitian tentang evaluasi terhadap kemutuan dari biji melinjo dengan memakai ekstraksi RGB dan HSV tanpa metode *machine learning* didapatkan akurasi 92,5%[7]. Penelitian klasifikasi kematangan buah melinjo menggunakan *masking*, *thresholding*, ekstraksi RGB dan HSV didapatkan akurasi sebesar 98% [8].

Pada penelitian ini digunakan algoritma klasifikasi *k-Nearest Neighbor* (KNN) dan metode ekstraksi fitur warna HSV (*Hue*, *Saturation*, *Value*) untuk mendapatkan ciri dari tingkat kematangan buah melinjo dan membantu proses klasifikasi dari tingkat kematangannya berdasarkan citra. Algoritma *k-Nearest Neighbor* adalah sebuah metode untuk melakukan klasifikasi terhadap objek yang berdasarkan dari data pembelajaran yang jaraknya paling dekat dengan objek tersebut [9][13].

Cara yang dapat dilakukan dalam mengidentifikasi fase kematangan buah melinjo di rumpun ilmu informatika yakni mengidentifikasi warna kematangan dari buah melinjo tersebut dengan menerapkan metode klasifikasi *machine learning*. Klasifikasi *k-Nearest Neighbor* (KNN) serta metode ekstraksi warna HSV akan dipadukan dalam penelitian kali ini guna memperoleh ciri dari fase kematangan buah melinjo dan membantu proses klasifikasi dari fase kematangannya berdasarkan citra dengan mengetahui akurasi optimal dalam mengklasifikasikan fase kematangan buah melinjo. Informasi hasil identifikasi dapat dikatakan bahwa penelitian ini memiliki manfaat yang berdampak bagi masyarakat luas karena dapat memberikan solusi dari permasalahan yang terjadi di masyarakat dalam mengidentifikasi fase kematangan buah. Keterbaruan pada penelitian ini ditunjukkan dari perpaduan KNN dan ekstraksi HSV serta dilakukannya *filtering* pada tahap *preprocessing*.

## METODE PENELITIAN

Metode penelitian dibuat dalam bentuk alur tahapan agar tiap proses dalam penelitian berjalan secara sistematis. Alur tahapan penelitian dapat dilihat pada Gambar 1.



**Gambar 1. Alur Metode Penelitian**

## Identifikasi Masalah

Metode penelitian diawali dengan mengidentifikasi masalah sesuai dengan bagian pendahuluan untuk mengetahui tujuan dan manfaat dari dilakukannya penelitian ini bagi masyarakat.

## Studi Literatur

Setelah itu, dilakukan tahapan studi literatur dengan menganalisis beberapa penelitian yang dinilai cukup mirip sehingga dapat dijadikan acuan dan diciptakannya keterbaharuan dalam mengerjakan penelitian hingga selesai. Penelitian yang dijadikan literatur diambil dari tahun 2018 hingga 2022.

## Pengumpulan Data

Data dalam penelitian ini adalah data citra buah melinjo yang akan digunakan untuk data latih dan data uji. Dalam penelitian ini, proses pengumpulan data yang digunakan mencakup: observasi dengan mengunjungi langsung lokasi penjualan buah melinjo yang ada di pasar tradisional Pasar Minggu dan mengambil beberapa dokumentasi citra yang berkaitan dengan hal penelitian yaitu buah melinjo dengan tiga kelas atau tingkat kematangan antara lain hijau yang artinya mentah, kuning atau oranye yang artinya setengah matang, dan merah yang artinya matang. Kamera yang digunakan untuk pengumpulan citra buah melinjo ialah *smartphone* vivo iQOO Z7 (64MP OIS).

Sebanyak 27 buah melinjo dikumpulkan untuk dijadikan *dataset* yang terdiri dengan 3 kategori masing-masing 9 melinjo mentah, 9 melinjo setengah matang, dan 9 melinjo matang. Citra diperoleh dari 4 sisi yang dimiliki tiap buah melinjo yang dapat dilihat pada Gambar 2. Sehingga, terdapat 108 citra yang dibagi sesuai dengan fase kematangannya yakni 36 melinjo mentah, 36 melinjo setengah matang, serta 36 melinjo matang berbentuk citra digital.

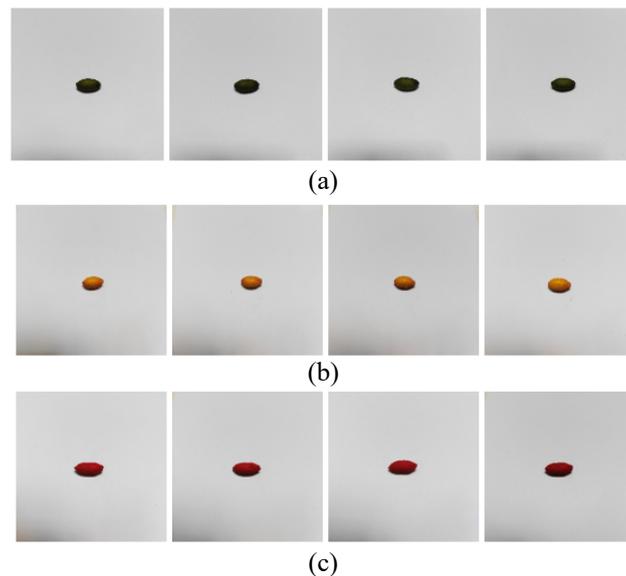
Pada kode program akan dibuat label numerik untuk 3 fase kematangan buah melinjo, yakni 0 untuk mentah, 1 untuk setengah matang, dan 2 untuk matang.

## Preprocessing

Dalam penelitian ini, proses *preprocessing* yang dilakukan mencakup *resize* citra dengan mengubah ukuran keseluruhan citra yang ada menjadi 300 x 300 *pixel*, segmentasi citra dengan memisahkan antara objek dengan latar belakangnya, dan *filtering* citra untuk mengurangi *noise* pada citra.

## Resize

Langkah pertama yang dilakukan adalah dengan melakukan *resize* citra dengan mengubah ukuran keseluruhan citra yang sebelumnya diambil dengan ukuran 1: 1 menjadi 300 x 300 *pixel* guna mempercepat proses komputasi yang dilakukan oleh program dan menciptakan hasil akurasi dan model yang tepat.



Gambar 2 Sisi Citra Melinjo Berdasarkan Kategori Mentah (a), Setengah Matang (b) dan Matang (c)

### *Segmentasi*

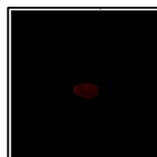
Keseluruhan dari citra buah melinjo akan disegmentasi guna memisahkan buah melinjo dengan kertas HVS yang menjadi latar belakang dari buah melinjo tersebut. Tujuannya agar memudahkan proses ekstraksi HSV agar proses identifikasi hanya berfokus pada warna yang dimiliki oleh buah melinjo sebagai objek utamanya.



Gambar 3 Contoh Citra Buah Melinjo Hasil Segmentasi

### *Filtering menggunakan Median Blur*

Langkah ketiga yaitu menggunakan *Median Blur* guna mengurangi *noise* pada citra dengan menggunakan *library* OpenCV dengan *kernel* berukuran  $5 \times 5$ , sehingga citra akan tampak jelas dan memudahkan proses selanjutnya.



Gambar 4 Citra Buah Melinjo Hasil *Filtering*

### **Ekstraksi Fitur Warna HSV**

Citra buah melinjo yang sudah dilakukan preprocessing memiliki format warna RGB (*Red, Green, Blue*). Dimana citra RGB memiliki ruang warna yang terlalu besar sehingga sulit untuk disegmentasi. Sehingga dalam penelitian ini, citra buah melinjo yang sudah dilakukan *preprocessing* tersebut akan dikonversi dari ruang warna RGB ke ruang warna HSV (*Hue, Saturation, Value*) guna memisahkan objek dengan latar belakangnya. Pada tahap ini, data citra buah melinjo akan diolah agar dapat mengeluarkan ciri dari citra tersebut. Dalam proses segmentasi citra, *Hue* digunakan untuk merepresentasikan nilai warna, membedakan tiap

warna, dan mengidentifikasi *greenness* (kehijauan), dan *redness* (kemerahan) dari cahaya (Persamaan 3). *Saturation* pada citra menunjukkan intensitas warnanya dimana perhitungannya menggunakan Persamaan 1. Terakhir ialah *value* pada citra mewakili fase kecerahannya (Persamaan 1) [11].

$$Value = \max (Red, Green, Blue) \quad (1)$$

$$Saturation = \begin{cases} 0, & \text{if } Value = 0 \\ 1 - \frac{\min(Red, Green, Blue)}{v}, & \text{if } Value > 0 \end{cases} \quad (2)$$

$$Hue = \begin{cases} 0, & \text{if } Saturation = 0 \\ 60^\circ * \frac{(Green - Blue)}{Saturation * Value}, & \text{if } Value = Red \\ 60^\circ * \left[ 2 + \frac{Blue - Red}{Saturation * Value} \right], & \text{if } Value = Green \\ 60^\circ * \left[ 4 + \frac{Blue - Red}{Saturation * Value} \right], & \text{if } Value = Blue \end{cases} \quad (3)$$

### Klasifikasi KNN (*K-Nearest Neighbor*)

Model dibangun dengan algoritma KNN menggunakan *k-Neighbors Classifier* yang terdapat pada *library Sklearn*. Prosesnya melibatkan beberapa langkah, yakni menentukan nilai parameter *k* untuk membuat model KNN yang dilatih menggunakan *data training* dan label kelasnya, setelah model KNN dilatih, prediksi akan dilakukan terhadap *data testing*. Hasil akan berisi prediksi kelas dari *data testing* berdasarkan model KNN yang telah dilatih sebelumnya untuk menentukan besaran akurasi dan *error* secara manual dalam mengolah tiap citra dari buah melinjo. Kelas yang paling banyak muncul dalam algoritma ini akan menjadi kelas hasil dari klasifikasi [12].

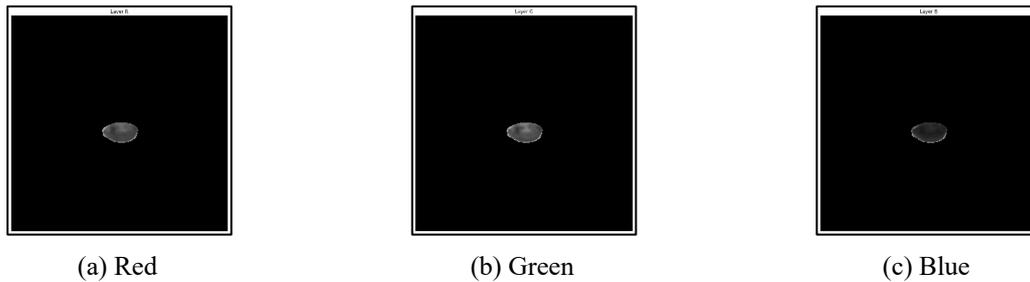
### Evaluasi Model

Pada tahapan ini citra yang sudah di ekstraksi cirinya akan dilakukan pengujian klasifikasi menggunakan KNN dengan menggunakan *library Sklearn*. Lalu, diuji dengan mengimport fungsi *accuracy score* guna melihat tingkat atau hasil akurasi dalam pengujian dan mengimport fungsi *classification report* untuk mengevaluasi model menggunakan metrik lain seperti *precision*, *recall*, dan *F1-score*. Pengujian dilakukan guna menentukan besaran akurasi dan *error* secara manual dalam mengolah tiap citra dari buah melinjo. Dimana kelas label 0 untuk mentah, 1 untuk setengah matang, dan 2 untuk matang menggunakan Persamaan 4.

$$Akurasi = \frac{\sum Data Akurat}{n} \times 100\% \quad (4)$$

## HASIL DAN PEMBAHASAN

Penelitian ini menggunakan citra buah melinjo yang telah dikumpulkan sebelumnya yang kemudian dilakukan *preprocessing* dengan mentransformasi dari yang sebelumnya merupakan citra dengan ruang warna RGB menjadi citra dengan ruang warna HSV (*Hue*, *Saturation*, *Value*) guna mengeluarkan ciri dari citra tersebut. Tahapan ini dilakukan dengan menangkap citra Mentah 26.jpg sebagai contoh yang sudah dilakukan *preprocessing*. Dari citra tersebut akan dipisahkan nilai dari lapisan *Red (R)*, *Green*, dan *Blue* dan menampilkan citra dari masing-masing lapisan warna tersebut terlihat pada Gambar 5.



Gambar 5 Nilai Setiap Lapisan (a) R = 0,4058 (b) G= 0.3863, dan (c) B= 0.2077

Proses ekstraksi fitur ialah mentransformasi masing-masing lapisan nilai RGB menjadi nilai HSV menggunakan Persamaan 1 untuk nilai *Value*, Persamaan 2 untuk nilai *Saturation*, dan Persamaan 3 untuk nilai *Hue* dengan hasil perhitungan seperti berikut :

$$Red = 0,4058; Green = 0,3863; Blue = 0,2077$$

$$Value = 0,4058 ; (Red)$$

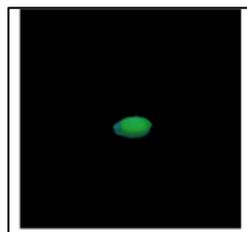
$$Saturation = 1 - \frac{0,2077}{0,4058} = 0,4882$$

$$Hue = \frac{60}{360} * \frac{(0,3863 - 0,2077)}{0,4882 * 0,4058} = 54,0938$$

Keseluruhan nilai yang telah diekstraksi dapat dilihat pada Tabel 1 dengan hasil citra ekstraksi pada Gambar 6.

Tabel 1. Luaran Citra Mentah 26.jpg Setelah Diekstraksi

	<i>Red</i>	<i>Green</i>	<i>Blue</i>	<i>Hue</i>	<i>Saturation</i>	<i>Value</i>
Mentah 26.jpg	0.405867	0.386365	0.207767	54.093300	0.488090	0.405867



Gambar 6 Citra Buah Melinjo Hasil Ekstraksi HSV

Setelah tahapan preprocessing seluruh citra RGB-HSV akan terbagi sebesar 70% atau sebanyak 75 citra sebagai data latih serta 30% atau sebanyak 33 citra sebagai data uji menggunakan *train test split* yang terdapat pada *library Sklearn*. Sehingga didapatkan matriks data input yang telah di *split* pada Tabel 2 sebagai hasil luaran dari *data training* yang sudah dilakukan ekstraksi ke ruang warna HSV dan Tabel 3 sebagai luaran dari *data testing* yang sudah dilakukan ekstraksi ke ruang warna HSV.

Hasil ekstraksi citra yang merupakan matriks input memiliki nilai penciri yang membedakan masing-masing kelas yang telah ditentukan. Ciri ini akan digunakan untuk membuat model klasifikasi dengan KNN dengan masing-masing nilai *k* sebanyak 5 bilangan prima yang telah ditentukan yaitu 2, 3, 5, 7 dan 11. Model akan dilakukan evaluasi menggunakan kode program *library Sklearn* dan mengimport fungsi *accuracy score* yang sesuai dengan Persamaan 4 untuk melihat tingkat atau hasil akurasi dalam pengujian dan mengimport fungsi *classification report*

untuk memeriksa hasil akurasi lebih detail.

**Tabel 2. Luaran Data Training Setelah Diekstraksi**

Data	Red	Green	Blue	Hue	Saturation	Value	Label
1	0.525166	0.362844	0.111988	36.428186	0.786755	0.525166	1
2	0.667697	0.186351	0.145950	4.646071	0.781412	0.667697	2
3	0.414890	0.376764	0.208344	48.924785	0.497832	0.414890	0
4	0.380167	0.350161	0.269670	43.706471	0.290654	0.380167	0
5	0.659077	0.175196	0.165725	1.151842	0.748549	0.659077	2
...	...	...	...	...	...	...	...
71	0.391731	0.442250	0.166017	70.973131	0.624608	0.442250	0
72	0.400996	0.325842	0.273160	24.726449	0.318796	0.400996	1
73	0.406814	0.383221	0.209963	52.809070	0.483882	0.406814	0
74	0.532513	0.346481	0.121005	32.875619	0.772765	0.532513	1
75	0.405394	0.337756	0.256849	32.679636	0.366420	0.405394	1

**Tabel 3. Luaran Data Testing Setelah Diekstraksi**

Data	Red	Green	Blue	Hue	Saturation	Value	Label
1	0.578632	0.299022	0.122344	23.232472	0.788561	0.578632	1
2	0.662638	0.192364	0.144996	5.490332	0.781182	0.662638	2
3	0.459288	0.350400	0.190310	35.710663	0.585640	0.459288	1
4	0.400495	0.388897	0.210606	56.335369	0.474134	0.400495	0
5	0.603066	0.208782	0.188150	2.983540	0.688009	0.603066	2
...	...	...	...	...	...	...	...
29	0.391235	0.386848	0.221915	58.445158	0.432782	0.391235	0
30	0.392284	0.432912	0.174803	69.444373	0.596215	0.432912	0
31	0.400502	0.384870	0.214627	54.954296	0.464104	0.400502	0
32	0.413293	0.389177	0.197529	53.293778	0.522059	0.413293	0
33	0.539887	0.347142	0.112970	32.911177	0.790752	0.539887	1

Gambar 7 menunjukkan dengan nilai  $k=2$  didapatkan nilai akurasi sebesar 100% yang menjadi akurasi paling tinggi pada penelitian ini. Gambar 8 menunjukkan nilai *precision*, *recall*, dan *F1-score* sebesar 100% pada seluruh fase kematangan buah melinjo yang menyatakan tidak ditemukan citra yang kelas prediksinya tidak sesuai dengan kelas pengujianya. Luaran pengujian dapat dilihat pada Tabel 4.

```
[42] knn = KNeighborsClassifier(n_neighbors=2)
      knn.fit(X_train, y_train) #model KNN dilatih dengan menggunakan data pelatihan (X_train,
      y_train)
      y_predict = knn.predict(X_test) #setelah model KNN dilatih, melakukan prediksi terhadap
      data pengujian (X_test)

[43] print("Akurasi = ", float(accuracy_score(y_test, y_predict))*100, end=""); print("%")
      #menghitung akurasi dgn membandingkan label sebenarnya dari data pengujian (y_test) den
      label prediksi (y_predict)
      Akurasi = 100.0%
```

**Gambar 7 Akurasi  $k=2$**

```
[31] labels = ['Mentah', 'Setengah Matang', 'Matang']
      print(classification_report(y_test, y_predict2, target_names = labels))

              precision    recall  f1-score   support

 Mentah         1.00         1.00         1.00         11
Setengah Matang 1.00         1.00         1.00         11
 Matang         1.00         1.00         1.00         11

 accuracy              1.00         33
 macro avg             1.00         1.00         1.00         33
 weighted avg         1.00         1.00         1.00         33
```

**Gambar 8 Classification Report  $k=2$**

Gambar 9 menunjukkan dengan nilai  $k=3$  didapatkan nilai akurasi sebesar 96,96%. Gambar 10 menunjukkan nilai *precision* sebesar 100%, *recall* sebesar 91%, dan *F1-Score* sebesar 95% pada fase kematangan mentah. Pada fase setengah matang didapatkan nilai *precision* sebesar 92%, *recall* sebesar 100%, dan *F1-Score* sebesar 96%. Pada fase matang didapatkan nilai *precision*, *recall*, dan *F1-score* sebesar 100%.

Pengujian yang dilakukan dengan  $k=3$  sesuai pada Tabel 5 ditemukan 1 citra yang kelas prediksinya tidak sesuai dengan kelas pengujianya.

```
[44] knn = KNeighborsClassifier(n_neighbors=3)
      knn.fit(X_train, y_train) #model KNN dilatih dengan menggunakan data pelatihan (X_train,
      y_train)
      y_predict = knn.predict(X_test) #setelah model KNN dilatih, melakukan prediksi terhadap
      data pengujian (X_test)

[45] print("Akurasi = ", float(accuracy_score(y_test, y_predict))*100, end=""); print("%")
      #menghitung akurasi dgn membandingkan label sebenarnya dari data pengujian (y_test) den
      gkan label prediksi (y_predict)
      Akurasi = 96.969696969697%
```

Gambar 9 Akurasi k=3

```
[32] labels = ['Mentah', 'Setengah Matang', 'Matang']
      print(classification_report(y_test, y_predict3, target_names = labels))
```

	precision	recall	f1-score	support
Mentah	1.00	0.91	0.95	11
Setengah Matang	0.92	1.00	0.96	11
Matang	1.00	1.00	1.00	11
accuracy			0.97	33
macro avg	0.97	0.97	0.97	33
weighted avg	0.97	0.97	0.97	33

Gambar 10 Classification Report k=3

Gambar 11 menunjukkan dengan nilai k=5 didapatkan nilai akurasi sebesar 96,96%. Gambar 12 menunjukkan nilai precision sebesar 100%, recall sebesar 91%, dan F1-Score sebesar 95% pada fase kematangan mentah. Pada fase setengah matang didapatkan nilai precision sebesar 92%, recall sebesar 100%, dan F1-Score sebesar 96%. Pada fase matang didapatkan nilai precision, recall, dan F1-score sebesar 100%.

Tabel 4. Luaran Pengujian KNN dengan k=2

Data	Kelas Asli	Kelas Prediksi	Kesimpulan
1	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
2	2 (Matang)	2 (Matang)	Tepat
3	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
4	0 (Mentah)	0 (Mentah)	Tepat
5	2 (Matang)	2 (Matang)	Tepat
6	2 (Matang)	2 (Matang)	Tepat
7	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
9	0 (Mentah)	0 (Mentah)	Tepat
10	0 (Mentah)	0 (Mentah)	Tepat
11	0 (Mentah)	0 (Mentah)	Tepat
12	2 (Matang)	2 (Matang)	Tepat
13	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
14	2 (Matang)	2 (Matang)	Tepat
16	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
17	2 (Matang)	2 (Matang)	Tepat
18	2 (Matang)	2 (Matang)	Tepat
19	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
20	2 (Matang)	2 (Matang)	Tepat
21	2 (Matang)	2 (Matang)	Tepat
22	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
23	0 (Mentah)	0 (Mentah)	Tepat
24	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
25	0 (Mentah)	0 (Mentah)	Tepat
26	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
27	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
28	0 (Mentah)	0 (Mentah)	Tepat
29	2 (Matang)	2 (Matang)	Tepat
30	2 (Matang)	2 (Matang)	Tepat
31	0 (Mentah)	0 (Mentah)	Tepat
32	0 (Mentah)	0 (Mentah)	Tepat
33	0 (Mentah)	0 (Mentah)	Tepat

Tabel 5. Luaran Pengujian KNN dengan k=3

Data	Kelas Asli	Kelas Prediksi	Kesimpulan
1	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
2	2 (Matang)	2 (Matang)	Tepat
3	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
4	0 (Mentah)	0 (Mentah)	Tepat
5	2 (Matang)	2 (Matang)	Tepat
6	2 (Matang)	2 (Matang)	Tepat
7	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
9	0 (Mentah)	1 (Setengah Matang)	Tidak Tepat
10	0 (Mentah)	0 (Mentah)	Tepat
11	0 (Mentah)	0 (Mentah)	Tepat
12	2 (Matang)	2 (Matang)	Tepat
13	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
14	2 (Matang)	2 (Matang)	Tepat
16	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
17	2 (Matang)	2 (Matang)	Tepat
18	2 (Matang)	2 (Matang)	Tepat
19	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
20	2 (Matang)	2 (Matang)	Tepat
21	2 (Matang)	2 (Matang)	Tepat
22	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
23	0 (Mentah)	0 (Mentah)	Tepat
24	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
25	0 (Mentah)	0 (Mentah)	Tepat
26	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
27	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
28	0 (Mentah)	0 (Mentah)	Tepat
29	2 (Matang)	2 (Matang)	Tepat
30	2 (Matang)	2 (Matang)	Tepat
31	0 (Mentah)	0 (Mentah)	Tepat
32	0 (Mentah)	0 (Mentah)	Tepat
33	0 (Mentah)	0 (Mentah)	Tepat

```
[46] knn = KNeighborsClassifier(n_neighbors=5)
      knn.fit(X_train, y_train) #model KNN dilatih dengan menggunakan data pelatihan (X_train,
      y_train)
      y_predict = knn.predict(X_test) #setelah model KNN dilatih, melakukan prediksi terhadap
      data pengujian (X_test)

[47] print("Akurasi = ", float(accuracy_score(y_test, y_predict)*100, end=""); print("%")
      #menghitung akurasi dgn membandingkan label sebenarnya dari data pengujian (y_test) den
      label prediksi (y_predict)
      Akurasi = 96.96969696969697%
```

Gambar 11 Akurasi k=5

```
[33] labels = ['Mentah', 'Setengah Matang', 'Matang']
      print(classification_report(y_test, y_predict5, target_names = labels))
```

	precision	recall	f1-score	support
Mentah	1.00	0.91	0.95	11
Setengah Matang	0.92	1.00	0.96	11
Matang	1.00	1.00	1.00	11
accuracy			0.97	33
macro avg	0.97	0.97	0.97	33
weighted avg	0.97	0.97	0.97	33

Gambar 12 Classification Report k=5

Pengujian yang dilakukan dengan k=5 sesuai pada Tabel 6 ditemukan 1 citra yang kelas prediksinya tidak sesuai dengan kelas pengujiannya.

Tabel 6. Luaran Pengujian KNN dengan k=5

Data	Kelas Asli	Kelas Prediksi	Kesimpulan
1	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
2	2 (Matang)	2 (Matang)	Tepat
3	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
4	0 (Mentah)	0 (Mentah)	Tepat
5	2 (Matang)	2 (Matang)	Tepat
6	2 (Matang)	2 (Matang)	Tepat
7	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
9	0 (Mentah)	1 (Setengah Matang)	Tidak Tepat
10	0 (Mentah)	0 (Mentah)	Tepat
11	0 (Mentah)	0 (Mentah)	Tepat
12	2 (Matang)	2 (Matang)	Tepat
13	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
14	2 (Matang)	2 (Matang)	Tepat
16	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
17	2 (Matang)	2 (Matang)	Tepat
18	2 (Matang)	2 (Matang)	Tepat
19	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
20	2 (Matang)	2 (Matang)	Tepat
21	2 (Matang)	2 (Matang)	Tepat
22	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
23	0 (Mentah)	0 (Mentah)	Tepat
24	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
25	0 (Mentah)	0 (Mentah)	Tepat
26	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
27	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
28	0 (Mentah)	0 (Mentah)	Tepat
29	2 (Matang)	2 (Matang)	Tepat
30	2 (Matang)	2 (Matang)	Tepat
31	0 (Mentah)	0 (Mentah)	Tepat
32	0 (Mentah)	0 (Mentah)	Tepat
33	0 (Mentah)	0 (Mentah)	Tepat

Tabel 7. Luaran Pengujian KNN dengan K = 7

Data	Kelas Asli	Kelas Prediksi	Kesimpulan
1	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
2	2 (Matang)	2 (Matang)	Tepat
3	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
4	0 (Mentah)	0 (Mentah)	Tepat
5	2 (Matang)	2 (Matang)	Tepat
6	2 (Matang)	2 (Matang)	Tepat
7	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
9	0 (Mentah)	1 (Setengah Matang)	Tidak Tepat
10	0 (Mentah)	0 (Mentah)	Tepat
11	0 (Mentah)	0 (Mentah)	Tepat
12	2 (Matang)	2 (Matang)	Tepat
13	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
14	2 (Matang)	2 (Matang)	Tepat
16	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
17	2 (Matang)	2 (Matang)	Tepat
18	2 (Matang)	2 (Matang)	Tepat
19	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
20	2 (Matang)	2 (Matang)	Tepat
21	2 (Matang)	2 (Matang)	Tepat
22	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
23	0 (Mentah)	0 (Mentah)	Tepat
24	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
25	0 (Mentah)	0 (Mentah)	Tepat
26	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
27	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
28	0 (Mentah)	0 (Mentah)	Tepat
29	2 (Matang)	2 (Matang)	Tepat
30	2 (Matang)	2 (Matang)	Tepat
31	0 (Mentah)	0 (Mentah)	Tepat
32	0 (Mentah)	0 (Mentah)	Tepat
33	0 (Mentah)	0 (Mentah)	Tepat

Gambar 13 menunjukkan dengan nilai k=7 didapatkan nilai akurasi sebesar 96,96%. Gambar 14 menunjukkan nilai precision sebesar 100%, recall sebesar 91%, dan F1-Score sebesar 95% pada fase kematangan mentah. Pada fase setengah matang didapatkan nilai precision sebesar 92%, recall sebesar 100%, dan F1-Score sebesar 96%. Pada fase matang didapatkan nilai precision, recall, dan F1-score sebesar 100%.

```

[48] knn = KNeighborsClassifier(n_neighbors=7)
      knn.fit(X_train, y_train) #model KNN dilatih dengan menggunakan data pelatihan (X_train,
      y_train)
      y_predict = knn.predict(X_test) #setelah model KNN dilatih, melakukan prediksi terhadap
      data pengujian (X_test)

[49] print("Akurasi = ", float(accuracy_score(y_test, y_predict))*100, end=""); print("%")
      #menghitung akurasi dgn membandingkan label sebenarnya dari data pengujian (y_test) den
      label sebenarnya (y_train)
      Akurasi = 96.96969696969697%

```

Gambar 13 Akurasi  $k=7$ 

```

[34] labels = ['Mentah', 'Setengah Matang', 'Matang']
      print(classification_report(y_test, y_predict7, target_names = labels))

```

	precision	recall	f1-score	support
Mentah	1.00	0.91	0.95	11
Setengah Matang	0.92	1.00	0.96	11
Matang	1.00	1.00	1.00	11
accuracy			0.97	33
macro avg	0.97	0.97	0.97	33
weighted avg	0.97	0.97	0.97	33

Gambar 14 Classification Report  $k=7$ 

Pengujian yang dilakukan dengan  $k=7$  terlihat pada Tabel 7 ditemukan 1 citra yang kelas prediksinya tidak sesuai dengan kelas asli.

Tabel 8. Luaran Pengujian KNN dengan  $K = 11$ 

Data	Kelas Asli	Kelas Prediksi	Kesimpulan
1	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
2	2 (Matang)	2 (Matang)	Tepat
3	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
4	0 (Mentah)	0 (Mentah)	Tepat
5	2 (Matang)	2 (Matang)	Tepat
6	2 (Matang)	2 (Matang)	Tepat
7	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
9	0 (Mentah)	1 (Setengah Matang)	Tidak Tepat
10	0 (Mentah)	0 (Mentah)	Tepat
11	0 (Mentah)	0 (Mentah)	Tepat
12	2 (Matang)	2 (Matang)	Tepat
13	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
14	2 (Matang)	2 (Matang)	Tepat
16	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
17	2 (Matang)	2 (Matang)	Tepat
18	2 (Matang)	2 (Matang)	Tepat
19	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
20	2 (Matang)	2 (Matang)	Tepat
21	2 (Matang)	2 (Matang)	Tepat
22	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
23	0 (Mentah)	0 (Mentah)	Tepat
24	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
25	0 (Mentah)	0 (Mentah)	Tepat
26	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
27	1 (Setengah Matang)	1 (Setengah Matang)	Tepat
28	0 (Mentah)	0 (Mentah)	Tepat
29	2 (Matang)	2 (Matang)	Tepat
30	2 (Matang)	2 (Matang)	Tepat
31	0 (Mentah)	0 (Mentah)	Tepat
32	0 (Mentah)	0 (Mentah)	Tepat
33	0 (Mentah)	0 (Mentah)	Tepat

Gambar 15 menunjukkan dengan nilai  $k=11$  didapatkan nilai akurasi sebesar 96,96%. Gambar 16 menunjukkan nilai *precision* sebesar 100%, *recall* sebesar 91%, dan *F1-Score* sebesar 95% pada fase kematangan mentah. Pada fase setengah matang didapatkan nilai *precision* sebesar 92%, *recall* sebesar 100%, dan *F1-score* sebesar 96%. Pada fase matang didapatkan nilai *precision*, *recall*, dan *F1-score* sebesar 100%.

Pengujian yang dilakukan dengan  $k=11$  sesuai pada Tabel 8 ditemukan 1 citra yang kelas prediksinya tidak sesuai dengan kelas pengujiannya atau kelas asli.

```
[50] knn = KNeighborsClassifier(n_neighbors=11)
      knn.fit(X_train, y_train) #model KNN dilatih dengan menggunakan data pelatihan (X_train,
      y_train)
      y_predict = knn.predict(X_test) #setelah model KNN dilatih, melakukan prediksi terhadap
      data pengujian (X_test)

[51] print("Akurasi = ", float(accuracy_score(y_test, y_predict))*100, end=""); print("%")
      #menghitung akurasi dgn membandingkan label sebenarnya dari data pengujian (y_test) den
      gkan label prediksi (y_predict)
      Akurasi = 96.969696969697%
```

Gambar 13 Akurasi  $k=11$ 

```
[35] labels = ['Mentah', 'Setengah Matang', 'Matang']
      print(classification_report(y_test, y_predict11, target_names = labels))

              precision    recall  f1-score   support

 Mentah         1.00         0.91         0.95         11
Setengah Matang  0.92         1.00         0.96         11
 Matang         1.00         1.00         1.00         11

 accuracy         0.97         0.97         0.97         33
 macro avg        0.97         0.97         0.97         33
 weighted avg     0.97         0.97         0.97         33
```

Gambar 14 Classification Report  $k=11$ 

Tabel 9 menunjukkan rata-rata akurasi yang didapatkan pada penelitian ini sebesar 97,56% sehingga penerapan metode klasifikasi *machine learning* KNN dan metode ekstraksi HSV yang dipadukan berhasil dalam mengklasifikasikan 3 fase kematangan pada buah melinjo. Selain itu, *classification report* menghasilkan 3 metrik untuk mengevaluasi model pada setiap fase kematangan. Dimana, model KNN dengan nilai  $k=2$  mendapatkan nilai 100% pada 3 metrik yakni *precision*, *recall*, dan *F1-score*. Sedangkan untuk model KNN dengan nilai  $k$  lainnya didapatkan nilai *precision* sebesar 100%, *recall* sebesar 91%, dan *F1-Score* sebesar 95% pada fase kematangan mentah. Pada fase setengah matang didapatkan nilai *precision* sebesar 92%, *recall* sebesar 100%, dan *F1-score* sebesar 96%. Pada fase matang didapatkan nilai *precision*, *recall*, dan *F1-score* sebesar 100%. Berdasarkan hasil luaran dari masing-masing parameter  $k$  diketahui bahwa identifikasi melinjo dengan tingkat kematangan mentah yang terjadi kesalahan identifikasi, akan tetapi kesalahan hanya terjadi pada data citra ke 9 yang memiliki tingkat kematangan mentah dengan penciri mirip dengan kematangan setengah matang.

Tabel 9 Hasil Keseluruhan Akurasi

K	Akurasi (%)
2	100%
3	96,96%
5	96,96%
7	96,96%
11	96,96%
<b>Rata-Rata</b>	<b>97,56%</b>

## KESIMPULAN

Penelitian ini berhasil mengetahui tingkat akurasi optimal dalam mengklasifikasikan fase kematangan buah melinjo dengan memanfaatkan metode klasifikasi *machine learning k-Nearest Neighbor* yang dipadukan dengan metode ekstraksi warna HSV (*Hue, Saturation, Value*) yang diterapkan pada citra digital. Dimana algoritma KNN dapat mengklasifikasikan 3 fase kematangan pada buah melinjo melalui pengujian dengan *data testing* dengan baik karena memiliki rata-rata akurasi sebesar 97,56%. Penelitian ini menghasilkan informasi penting bahwa jarak  $k=2$  dapat mengidentifikasi HSV yang dimiliki tiap citra buah melinjo dengan tingkat akurasi dan *classification report* tertinggi yakni sebesar 100% hal ini disebabkan parameter  $k=2$  memiliki pembeda kelas yang diasumsikan hanya dibedakan ya atau tidak saja jadi tidak membedakan dengan detail antara data yang satu dengan yang lainnya sehingga ketika parameter  $k$  dinaikan maka terjadi penurunan nilai akurasi, *precision*, *recall*, dan *F1-score*.

## DAFTAR PUSTAKA

1. H. I. Manner and C. R. Elevitch, *Species Profiles for Pacific Island Agroforestry* [www.traditionaltree.org](http://www.traditionaltree.org) In *BrIef*. 2006. [Online]. Available: [www.traditionaltree.org](http://www.traditionaltree.org)

2. A. JAN NEXSON PARHUSIP and A. BOING SITANGGANG, “Antimicrobial Activity of Melinjo Seed and Peel Extract (*Gnetum gnemon*) Against Selected Pathogenic Bacteria,” *Microbiol Indones*, vol. 5, no. 2, pp. 103–112, Jun. 2011, doi: 10.5454/mi.5.3.2.
3. C. W. Dwi Feri Ira, C. Ikhda NHS, F. Farmasi, and I. Ilmu Kesehatan Bakti Wiyata Kediri, “Efek Farmakologi Infusa Biji Melinjo (*Gnetum gnemon* L.) Sebagai Antihiperlikemia pada Mencit (*Mus musculus*) yang Diinduksi Dextrosa Monohidrat 40%,” *JOURNAL OF PHARMACEUTICAL SCIENCE AND PHARMACY PRACTICE*, vol. 2, no. 1, 2015, doi: 10.33508/jfst.v2i1.700.
4. E. Kato, Y. Tokunaga, and F. Sakan, “Stilbenoids Isolated from the Seeds of Melinjo (*Gnetum gnemon* L.) and Their Biological Activity,” *J Agric Food Chem*, vol. 57, no. 6, pp. 2544–2549, Feb. 2009, doi: 10.1021/jf803077p.
5. K. Khaira Ummah and S. Susanti, “Distribusi Anatomis Metabolit Sekunder dan Aktivitas Antioksidan Biji Melinjo (*Gnetum gnemon* L.) pada Tiga Fase Kematangan,” *J Biologi Indones*, vol. 18, no. 2, pp. 213–218, 2022, doi: 10.47349/jbi/18022022/213.
6. C. Jatmoko, D. Sinaga, H. Lestiawan, and C. A. Sari, “Klasifikasi Kualitas Buah Melinjo Menggunakan K-NN,” *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB)*, pp. 198–202, 2022.
7. S. Widodo and M. Kalili, “EVALUASI MUTU BIJI MELINJO (*Gnetum gnemon* L.) MENGGUNAKAN PENGOLAHAN CITRA DIGITAL,” *Jurnal Teknik Pertanian Lampung*, vol. 7, pp. 106–114, 2018, doi: 10.23960/jtep-Lv7i2.106-114.
8. D. Iskandar, S. Tinggi Ilmu Komputer Cipta Karya Informatika, and E. Jakarta, “CLASSIFICATION OF MELINJO FRUIT LEVELS USING SKIN COLOR DETECTION WITH RGB AND HSV,” *Journal of Applied Engineering and Technological Science*, vol. 4, no. 1, pp. 123–130, 2022, doi: 10.37385/jaets.v4i1.958.
9. Asahar Johar, Delfi Yanosma, and Kurnia Anggriani, “IMPLEMENTASI METODE K-NEAREST NEIGHBOR (KNN) DAN SIMPLE ADDITIVE WEIGHTING (SAW) DALAM PENGAMBILAN KEPUTUSAN SELEKSI PENERIMAAN ANGGOTA PASKIBRAKA,” *Jurnal Pseudocode*, vol. III, pp. 98–112, 2016.
10. D. Imantata Muhammad, Ermatita, and N. Falih, “Penggunaan K-Nearest Neighbor (KNN) untuk Mengklasifikasi Citra Belimbing Berdasarkan Fitur Warna,” *JURNAL INFORMATIK*, vol. 17, no. 1, pp. 9–16, 2021.
11. M. Aryo Aji Kurniawan, “PEMANFAATAN PENGOLAHAN CITRA DAN KLASIFIKASI K-NEAREST NEIGHBOR PADA CITRA TELUR AYAM,” Universitas Pembangunan Nasional Veteran Jakarta, Jakarta, 2020. [Online]. Available: <http://repository.upnvj.ac.id/id/eprint/8181>
12. R. G. D. Prasetyo, “PENGUNAAN K-NEAREST NEIGHBOR (KNN) UNTUK MENGLASIFIKASI CITRA AYAM BERFORMALIN DAN TIDAK BERFORMALIN BERDASARKAN WARNA KULIT,” Universitas Pembangunan Nasional Veteran Jakarta, Jakarta, 2019. [Online]. Available: <https://repository.upnvj.ac.id/607/>
13. Saputra, S., Yudhana, A., & Umar, R. (2022). Identifikasi Kesegaran Ikan Menggunakan Algoritma KNN Berbasis Citra Digital. *Krea-TIF: Jurnal Teknik Informatika*, 10(1), 1–9. <https://doi.org/10.32832/krea-tif.v10i1.6845>