

# Klasifikasi Kematangan Buah Apel dengan Ekstraksi Fitur Haralick dan KNN

I Kadek Bagus Deva Diga Dana Putra<sup>a1</sup>, I Ketut Gede Suhartana<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali Indonesia  
<sup>1</sup>bgsdeva999@gmail.com  
<sup>2</sup>ikg.suhartana@unud.ac.id

## Abstract

*This research aims to classify the ripeness level of apple fruits based on texture features using the Haralick method and color features using histograms. A dataset of 76 apple fruit images was collected. In the preprocessing stage, the apple images were converted to grayscale, followed by the application of a median filter to remove salt and pepper noise, and histogram equalization to enhance image contrast. Texture features were extracted using the Haralick method to obtain contrast, correlation, energy, homogeneity, and entropy features. Color features were extracted using histograms to obtain mean, standard deviation, skewness, and kurtosis. A K-Nearest Neighbor (KNN) model with  $k = 6$  was used for classification. The evaluation results showed an accuracy of 89.47%, precision of 93.75%, recall of 93.75%, and F1-score of 93.75%. This research indicates that texture and color features can effectively classify the ripeness level of apple fruits. Future research can explore more diverse datasets and parameter adjustments to further improve model performance.*

**Keywords:** apple fruit, ripeness classification, texture features, color features.

## 1. Pendahuluan

Tingkat kematangan buah apel merupakan salah satu faktor penting yang mempengaruhi kualitas dan rasa buah apel. Tingkat kematangan buah apel dapat ditentukan berdasarkan warna dan tekstur kulitnya. Buah apel yang masih mentah biasanya memiliki warna hijau atau kuning dan tekstur kulit yang keras. Buah apel yang sudah matang biasanya memiliki warna merah atau oranye dan tekstur kulit yang lembut [1].

Salah satu penelitian yang dilakukan oleh [2], yaitu klasifikasi tingkat kematangan apel manalagi merupakan varietas apel yang populer di Indonesia. Salah satu keunikan apel manalagi adalah warna kulitnya ketika mentah dan matang cukup sulit dibedakan karena perbedaannya tidak terlalu signifikan. Apel yang masih mentah berwarna hijau, sedangkan apel yang sudah matang berwarna hijau kekuningan. Hal ini membuat orang awam cukup sulit untuk membedakannya [2].

Mendeteksi tingkat kematangan buah apel secara manual dapat menyebabkan kesalahan dan ketidakkonsistenan karena bergantung pada persepsi manusia. Oleh karena itu, diperlukan suatu sistem yang dapat mengklasifikasikan tingkat kematangan buah apel secara otomatis dan akurat berdasarkan citra digital buah apel. Sistem ini dapat membantu petani, pedagang, atau konsumen dalam menentukan dan memilih buah apel yang berkualitas dan sehat.

Salah satu cara untuk mengklasifikasikan tingkat kematangan buah apel berdasarkan citra digital adalah dengan menggunakan metode ekstraksi fitur haralick dan metode klasifikasi KNN. Metode ekstraksi fitur haralick adalah metode yang menghitung fitur tekstur dari citra berdasarkan matriks ko-kemunculan abu-abu (GLCM). Fitur tekstur dapat merepresentasikan karakteristik permukaan citra, seperti kehalusan, kekasaran, atau kehomogenan [3]. Metode klasifikasi KNN adalah metode yang mengklasifikasikan citra berdasarkan jarak terdekat antara citra uji dan citra latih. Metode KNN dapat menyesuaikan diri dengan data yang tidak linear dan memiliki kompleksitas

perhitungan yang rendah [4].

Penelitian ini bertujuan untuk mengimplementasikan dan menguji akurasi, presisi, recall, dan F1-score dari metode ekstraksi fitur haralick dan metode klasifikasi KNN untuk mengklasifikasikan tingkat kematangan buah apel berdasarkan citra digital buah apel. Penelitian ini diharapkan dapat memberikan kontribusi bagi pengembangan sistem pengenalan pola citra, khususnya dalam bidang pertanian.

## 2. Metode Penelitian

### 2.1. Pengumpulan Data

Data yang digunakan pada penelitian ini adalah data sekunder yang diambil dari situs Kaggle[5]. Data ini berisi 240 data latih dan 60 data uji citra buah-buahan yang terdiri dari buah apel, pisang, dan jeruk. Namun, akan diambil data buah apelnnya saja untuk penelitian ini yang terdiri dari 76 data latih dan 19 data uji. Data ini memiliki resolusi 100x100 dan berformat jpg.

### 2.2. Pra-pemrosesan data

Pra-pemrosesan data dilakukan untuk meningkatkan kualitas citra dan mengurangi noise atau gangguan yang ada pada citra. Tahapan pra-pemrosesan data yang dilakukan adalah sebagai berikut:

- a. Konversi citra RGB menjadi citra abu-abu (grayscale) dengan rumus:

$$I = 0.299R + 0.587G + 0.114B \quad (1)$$

- b. Penerapan filter median untuk menghilangkan noise salt and pepper dengan ukuran kernel 3x3 piksel.
- c. Penerapan histogram equalization untuk meningkatkan kontras citra dengan menggunakan fungsi bawaan dari OpenCV

### 2.3. Ekstraksi Fitur

Ekstraksi fitur dilakukan untuk mendapatkan fitur-fitur yang merepresentasikan karakteristik citra, seperti warna, tekstur, bentuk, dll. Fitur-fitur tersebut akan digunakan sebagai input untuk proses klasifikasi. Tahapan ekstraksi fitur yang dilakukan adalah sebagai berikut:

- a. Ekstraksi fitur tekstur menggunakan metode Haralick. Metode ini menghitung fitur tekstur berdasarkan matriks ko-kemunculan abu-abu (GLCM) yang mengukur hubungan spasial antara piksel-piksel pada citra. Fitur tekstur yang dihasilkan oleh metode ini adalah:
  1. Contrast: mengukur variasi intensitas antara piksel-piksel yang berdekatan.
  2. Correlation: mengukur ketergantungan linear antara piksel-piksel yang berdekatan.
  3. Energy: mengukur keseragaman distribusi intensitas pada citra.
  4. Homogeneity: mengukur kemiripan antara piksel-piksel yang berdekatan.
  5. Entropy: mengukur ketidakteraturan atau kompleksitas pada citra.
- b. Ekstraksi fitur warna menggunakan metode Color Histogram. Metode ini menghitung frekuensi kemunculan setiap nilai intensitas pada citra. Fitur warna yang dihasilkan oleh metode ini adalah:
  1. Mean: rata-rata nilai intensitas pada citra.
  2. Standard deviation: simpangan baku nilai intensitas pada citra.
  3. Skewness: ukuran kemiringan distribusi nilai intensitas pada citra.
  4. Kurtosis: ukuran keruncingan distribusi nilai intensitas pada citra.

## 2.4. Klasifikasi

Klasifikasi dilakukan untuk membedakan tingkat kematangan buah apel berdasarkan fitur-fitur yang telah diekstraksi. Metode klasifikasi yang digunakan adalah metode KNN. Metode ini mengklasifikasikan sebuah citra uji berdasarkan jarak terdekat dengan k citra latih yang memiliki label kelas yang sama. Jarak antara dua citra dihitung menggunakan rumus jarak Euclidean:

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2)$$

dimana d adalah jarak, n adalah jumlah fitur,  $x_i$  adalah nilai fitur ke-i dari citra uji, dan  $y_i$  adalah nilai fitur ke-i dari citra latih.

## 2.5. Evaluasi

Evaluasi dilakukan untuk mengukur kinerja dari metode klasifikasi yang digunakan. Metrik evaluasi yang digunakan adalah akurasi, presisi, recall, dan F1-score. Rumus-rumus untuk menghitung metrik evaluasi tersebut adalah sebagai berikut:

- a. Akurasi: rasio antara jumlah citra yang diklasifikasikan dengan benar dengan jumlah total citra.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

- b. Presisi: rasio antara jumlah citra positif yang diklasifikasikan dengan benar dengan jumlah total citra positif yang diprediksi.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (4)$$

- c. Recall: rasio antara jumlah citra positif yang diklasifikasikan dengan benar dengan jumlah total citra positif yang sebenarnya.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

- d. F1-score: rata-rata harmonik dari presisi dan recall.

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

## 3. Hasil dan Diskusi

### 3.1. Persiapan Data

Pada dataset penelitian ini, penulis melakukan labeling secara manual, tetapi mengikuti literatur yang ada [2]. Menurut penelitian [2], apel yang tidak matang adalah apel yang masih berwarna kehijauan (tidak merah pekat). Oleh karena itu, penulis melakukan labeling manual dengan melihat warna apel. Warna apel yang kehijauan dilabeli dengan tidak matang (ditulis 0) dan warna apel yang sudah merah dilabeli dengan matang (ditulis 1).



**Gambar 1.** Contoh Apel Matang Pada Dataset



**Gambar 2.** Contoh Apel Yang Tidak Matang Pada Dataset

### 3.2. Prapemrosesan

Sebelum melakukan ekstraksi fitur dengan metode Haralick, diperlukan menghilangkan informasi warna pada dataset gambar terlebih dahulu. Hal tersebut karena metode Haralick adalah metode yang mengoperasikan tekstur pada gambar sehingga informasi warna tidak terlalu diperlukan pada pengolahan.

```
1 # list untuk menyimpan hasil prapemrosesan dari setiap gambar
2 preprocessed_images_test = []
3
4 # Loop melalui 76 gambar dalam dataset Anda
5 for i in range(1, 76):
6     # Load image
7     try:
8         image_path = f"C:/Users/hp/OneDrive/Documents/Belajar Pemrograman/SNATIA/dataset/test_116/test/apple_{i+76}.jpg"
9         image = io.imread(image_path)
10
11         # Periksa jika citra memiliki 4 channel, lalu konversi menjadi RGB
12         if image.shape[2] == 4:
13             image = color.rgb2rgb(image)
14
15         # Konversi citra RGB menjadi citra abu-abu (grayscale)
16         gray_image = color.rgb2gray(image)
17
18         # Penerapan filter median
19         median_filtered = median(gray_image, selem=np.ones((3, 3)))
20
21         # Konversi gambar menjadi tipe data ubyte (0-255)
22         median_filtered = img_as_ubyte(median_filtered)
23
24         # Penerapan histogram equalization menggunakan OpenCV
25         equalized_image = cv2.equalizeHist(median_filtered)
26
27         # Tambahkan hasil prapemrosesan ke dalam list
28         preprocessed_images_test.append(equalized_image)
29     except:
30         print('sath happen')
```

**Gambar 3.** Kode Untuk Melakukan Penghilangan Informasi Warna Pada Data Test



**Gambar 4.** Contoh Salah Satu Gambar Yang Sudah Dihilangkan Informasi Warnanya

### 3.3. Ekstraksi Fitur

#### a. Ekstraksi Fitur Haralick

Setelah dataset dihilangkan informasinya, sekarang bisa dilakukan pengekstraksian fitur dengan metode Haralick dan akan diekstrak fitur berupa contrast, correlation, energy, homogeneity, dan entropy.

```
1 # List untuk menyimpan fitur-fitur yang diekstraksi
2 features_haralick_test = []
3
4 # Loop melalui gambar-gambar hasil prapemrosesan
5 for image in preprocessed_images_test:
6     # Hitung matriks GLCM dengan jarak 1 dan sudut 0 derajat
7     glcm = greycomatrix(image, distances=[1], angles=[0], levels=256, symmetric=True, normed=True)
8
9     # Ekstraksi fitur contrast
10    contrast = greycoprops(glcm, 'contrast')[0, 0]
11    features_haralick_test.append(contrast)
12
13    # Ekstraksi fitur correlation
14    correlation = greycoprops(glcm, 'correlation')[0, 0]
15    features_haralick_test.append(correlation)
16
17    # Ekstraksi fitur energy
18    energy = greycoprops(glcm, 'energy')[0, 0]
19    features_haralick_test.append(energy)
20
21    # Ekstraksi fitur homogeneity
22    homogeneity = greycoprops(glcm, 'homogeneity')[0, 0]
23    features_haralick_test.append(homogeneity)
24
25    # Ekstraksi fitur entropy
26    entropy = -np.sum(glcm * np.log2(glcm + 1e-10))
27    features_haralick_test.append(entropy)
```

**Gambar 5.** Kode Melakukan Ekstraksi Fitur Haralick Pada Data Test

Digunakan jarak 1 karena mempertimbangkan piksel yang berdekatan secara langsung. Ini mengakibatkan matriks GLCM mencerminkan hubungan spasial antara piksel-piksel yang saling berdekatan. Digunakan sudut 0 derajat karena mempertimbangkan pasangan piksel yang berada dalam satu arah horizontal (piksel di sebelah kanan).

	Contrast	Correlation	Energy	Homogeneity	Entropy
0	413.874867	0.961636	0.021384	0.279138	12.216877
1	20.967311	0.998223	0.135205	0.562925	9.079305
2	56.970746	0.996978	0.462538	0.749005	5.908961
3	336.483045	0.980058	0.499982	0.877345	3.893027
4	28.213824	0.998236	0.340881	0.773430	6.578070
5	222.498216	0.988583	0.490225	0.739776	6.005709

Gambar 6. Hasil Ekstraksi Fitur Haralick Pada 6 Gambar Data Test

### b. Ekstraksi Fitur Color Histogram

Pada tahap ini penulis melakukan ekstraksi fitur warna dengan metode Color Histogram. Setelah fitur warna diekstraksi, fitur akan digabungkan antara fitur Haralick dan fitur warna dari Color Histogram yang nantinya akan dilakukan proses pemodelan dengan KNN.

```

1 # List untuk menyimpan fitur-fitur yang diekstraksi
2 features_color_test = []
3
4 # Loop melalui gambar-gambar dalam dataset Anda
5 for i in range(1, 20):
6     try:
7         # Load image
8         image_path = f"C:/Users/hp/OneDrive/Documents/Belajar Pemrograman/SNATIA/dataset/test_zip/test/apple_{i+76}.jpg"
9         image = cv2.imread(image_path)
10
11        # Ekstraksi histogram warna
12        hist = cv2.calcHist([image], [0, 1, 2], None, [8, 8, 8], [0, 256, 0, 256, 0, 256])
13        hist = cv2.normalize(hist, hist).flatten()
14
15        # Ekstraksi fitur mean
16        mean = np.mean(hist)
17        features_color_test.append(mean)
18
19        # Ekstraksi fitur standar deviasi
20        std = np.std(hist)
21        features_color_test.append(std)
22
23        # Ekstraksi fitur skewness
24        skewness = moment(hist, moment=3)
25        features_color_test.append(skewness)
26
27        # Ekstraksi fitur kurtosis
28        kurtosis = moment(hist, moment=4)
29        features_color_test.append(kurtosis)
30    except:
31        print("ada error di nomor ",i)
    
```

Gambar 7. Kode Melakukan Ekstraksi Fitur Warna Data Test Dengan Color Histogram

	Mean	Standart	Skewness	Kurtosis
0	0.015879	0.041243	0.000290	0.000069
1	0.005670	0.043829	0.001061	0.000730
2	0.003889	0.044023	0.001568	0.001405
3	0.002635	0.044116	0.001908	0.001892
4	0.004438	0.043971	0.001637	0.001509
5	0.003311	0.044070	0.001831	0.001782

Gambar 8. Hasil Ekstraksi Fitur Warna Dengan Color Histogram Pada 6 Gambar Data Test

```
1 X_train = pd.concat([df_haralick_train, df_color_train], axis=1)
2 X_test = pd.concat([df_haralick_test, df_color_test], axis=1)
```

**Gambar 9.** Kode Untuk Melakukan Penggabungan Fitur Pada Data Test Dan Train

### 3.4. Pemodelan

Setelah fitur dari setiap data testing dan data training sudah digabungkan. Sekarang data siap untuk dimodelkan. Pemodelan akan dilakukan dengan metode KNN dengan k adalah 6. Pemilihan k bernilai 6 karena itu adalah hasil yang paling optimal pada kasus penelitian ini. Didapatkan hasil akurasi sebesar 89,47%, presisi 93,75%, recall 93,75%, dan F1-score sebesar 93,75%.

```
1 # Membuat model KNN
2 knn = KNeighborsClassifier(n_neighbors=6)
3
4 # Melatih model KNN
5 knn.fit(X_train, y_train)
6
7 # Memprediksi kelas untuk data uji
8 y_pred = knn.predict(X_test)
9
10 # Menghitung akurasi
11 accuracy = accuracy_score(y_test, y_pred)
12
13 # Menghitung presisi
14 precision = precision_score(y_test, y_pred)
15
16 # Menghitung recall
17 recall = recall_score(y_test, y_pred)
18
19 # Menghitung F1-score
20 f1 = f1_score(y_test, y_pred)
```

**Gambar 10.** Kode Melakukan Pemodelan dengan KNN

```
Akurasi: 0.8947368421052632
Presisi: 0.9375
Recall: 0.9375
F1-score: 0.9375
```

**Gambar 11.** Hasil Akurasi, Presisi, Recall, Dan F1-Score

## 4. Kesimpulan

Dari penelitian ini, dapat diambil kesimpulan sebagai berikut:

- Akurasi: Model KNN dengan menggunakan fitur-fitur Haralick dan histogram warna menghasilkan tingkat akurasi sebesar 89,47%. Hal ini menunjukkan bahwa model dapat melakukan klasifikasi dengan akurasi yang cukup baik.
- Presisi: Presisi sebesar 93,75% menunjukkan bahwa model KNN memiliki kemampuan untuk mengidentifikasi dengan baik buah apel yang matang dari dataset yang digunakan.
- Recall: Recall sebesar 93,75% menunjukkan bahwa model memiliki kemampuan untuk secara efektif mengenali dan menemukan buah apel yang matang dalam dataset.

- d. F1-score: F1-score sebesar 93,75% merupakan ukuran gabungan antara presisi dan recall. Skor ini mengindikasikan bahwa model KNN memiliki keseimbangan yang baik antara kemampuan untuk mengenali dan mengklasifikasikan dengan benar buah apel yang matang.

Dalam penelitian ini, model KNN dengan  $k = 6$  digunakan untuk klasifikasi. Hasil metrik yang diperoleh menunjukkan bahwa model ini berhasil dalam mengklasifikasikan kematangan buah apel berdasarkan fitur tekstur dan fitur warna yang diekstraksi. Namun, perlu diingat bahwa kesimpulan ini berdasarkan pada dataset yang digunakan dalam penelitian ini, yaitu dataset 76 gambar. Untuk menggeneralisasikan kesimpulan ini, penelitian lebih lanjut dengan menggunakan dataset yang lebih besar dan beragam dapat dilakukan. Selain itu, juga perlu mempertimbangkan parameter-parameter lain seperti pemilihan fitur, pemrosesan citra, dan pengaturan parameter pada model KNN untuk memperbaiki dan meningkatkan performa model lebih lanjut

#### Daftar Pustaka

- [1] Hanif, A. I., Razka, H., & Wiratomo, D. S. (2021). Klasifikasi tingkat kematangan buah apel berdasarkan fitur warna menggunakan algoritma K-Nearest Neighbor dan ekstraksi warna HSV. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer* E-ISSN: 2548-964X Vol. 5 No. 6 Mei 2021 Hlm: 2188-2199.
- [2] Ciputra, A., De Rosal, I. M., Setiadi, I. M., Rachmawanto, E. H., & Susanto, A. (2018). Klasifikasi tingkat kematangan buah apel manalagi dengan algoritma naive bayes dan ekstraksi fitur citra digital. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika (JITEKI)*, 4(1), 1-10.
- [3] Wijaya, N., & Ridwan, A. (2019). Klasifikasi jenis buah apel dengan metode K-nearest neighbors dengan ekstraksi fitur HSV dan LBP. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 8(1), 1-10.
- [4] Yusuf, T., Prahudaya, A., & Harjoko, A. (2017). Metode klasifikasi mutu jambu biji menggunakan KNN berdasar fitur warna dan tekstur. *Jurnal Ilmiah Teknologi Informasi Asia*, 11(2), 125-136.
- [5] Kinaci, M. B. (2018). Fruit images for object detection. Kaggle.com. Retrieved from <https://www.kaggle.com/mbkinaci/fruit-images-for-object-detection>