Journal of Innovative and Creativity
Journal Homepage: https://joecy.org/index.php/joecy
e-mail: journaljoecy@gmail.com

Faculty of Education
University of Pahlawan Tuanku Tambusai
ISSN 2775-771X (Online)
ISSN 2962-570X (Printed)
6 (1) 2026, Page 14037-14048

KLASIFIKASI KEMATANGAN BUAH ALPUKAT BERDASARKAN WARNA CITRA MENGGUNAKAN METODE *K-MEANS*

Gio Vernando[1], Muhammad Ikhwan Jambak[2], Zaid Romegar Mair[3]
[1,2,3]Program Studi Teknik Informatika Fakultas Ilmu Komputer Dan Sains Universitas Indo Global Mandiri
Corresponding Author Gio Vernando, E-mail: 2021110005@students.uigm.ac.id

Published: Februari, 2026

ABSTRAK
Penilaian manual tingkat kematangan buah alpukat sering subjektif dan tidak akurat, menyebabkan limbah pascapanen hingga 30%. Penelitian ini bertujuan mengembangkan sistem klasifikasi kematangan alpukat (mentah, setengah matang, matang) berdasarkan warna citra menggunakan K-Means Clustering. Penelitian pengembangan kuantitatif ini menggunakan populasi citra alpukat lokal Indonesia dengan sampel 600 citra (200 per kategori). Instrumen mencakup kamera resolusi 1024x768 piksel dan Python di Google Colab; teknik analisis meliputi preprocessing (resize, segmentasi, masking), ekstraksi fitur HSV, dan evaluasi confusion matrix. Hasil menunjukkan akurasi 91,67% dan F1-score makro 91% pada 180 citra uji, dengan kelas setengah matang sempurna. Kesimpulan menyatakan sistem efektif untuk otomatisasi penyortiran industri hortikultura, direkomendasikan pengembangan aplikasi mobile.

Kata Kunci: Alpukat, Citra Digital, K-Means Clustering, Kematangan Buah, Pengolahan Citra

© The Author(s). 2021 Open Access This article is licensed under a Creative Commons Atibution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0

PENDAHULUAN

Perkembangan produksi buah alpukat di Indonesia sebagai salah satu negara tropis terbesar mencapai estimasi 0,4 juta ton pada 2025, didukung iklim khatulistiwa yang ideal untuk budidaya. Namun, penanganan pascapanen sering menghadapi tantangan karena perubahan warna kulit dari hijau gelap pada tahap mentah menjadi ungu-hitam saat matang, yang memengaruhi rasa, kualitas, dan masa simpan buah.

Buah alpukat rentan pembusukan cepat akibat gas etilen, sehingga penilaian kematangan menjadi krusial untuk memaksimalkan umur simpan sebelum distribusi ke konsumen.

Penilaian manual oleh petani dan pedagang bersifat subjektif, tidak konsisten, dan kurang akurat, terutama saat membedakan tahap mentah, setengah matang, dan matang karena kemiripan warna kulit. Hal ini berisiko menurunkan nilai jual, meningkatkan limbah makanan hingga 30% di tingkat ritel, serta menyulitkan pengelompokan untuk distribusi.

Pendekatan konvensional gagal menangkap variasi pencahayaan dan latar belakang, menyebabkan kesalahan klasifikasi yang memengaruhi rantai pasok, sementara metode supervised seperti KNN memerlukan label data ekstensif yang sulit diperoleh di lapangan.

Oleh karena itu, diperlukan solusi berbasis pengolahan citra digital dengan analisis warna HSV yang stabil terhadap variasi cahaya, dikombinasikan unsupervised learning untuk pengelompokan otomatis tanpa label.

Penelitian ini bertujuan mengembangkan sistem klasifikasi kematangan alpukat (mentah, setengah matang, matang) berdasarkan warna citra menggunakan K-Means Clustering setelah preprocessing resize, segmentasi, masking, dan ekstraksi fitur HSV. Urgensinya terletak pada potensi mengurangi limbah pascapanen, meningkatkan efisiensi penyortiran bagi petani, serta mendukung ekspor dengan akurasi tinggi seperti 91,67% yang dicapai sebelumnya. Kebaruannya ada pada penerapan K-Means unsupervised dengan HSV untuk otomatisasi penuh, berbeda dari pendahuluan terdahulu yang bergantung supervised atau ruang warna lain, sehingga lebih fleksibel tanpa data berlabel.

METODE

Jenis dan Metode Penelitian

Penelitian ini menggunakan jenis penelitian pengembangan (research and development) dengan pendekatan kuantitatif melalui analisis citra digital untuk mengklasifikasikan tingkat kematangan buah alpukat menggunakan metode K-Means Clustering sebagai unsupervised learning. Metode ini dipilih karena memungkinkan pengelompokan data warna secara otomatis tanpa label awal, sesuai dengan prinsip metodologi penelitian kuantitatif yang menekankan pengukuran dan verifikasi hipotesis melalui data numerik. Pendekatan ini selaras dengan Sugiyono yang menyatakan bahwa penelitian kuantitatif efektif untuk menguji model komputasional pada dataset empiris seperti citra, dengan fokus pada akurasi dan F1-score. Selain itu, Sudaryono menegaskan bahwa desain seperti ini cocok untuk studi mix method awal yang mengintegrasikan pemrosesan data visual dengan evaluasi statistik.

Instrumen dan Teknik Analisis Data

Instrumen utama meliputi kamera untuk pengambilan citra alpukat pada jarak 30 cm dengan resolusi 1024x768 piksel di bawah pencahayaan terkendali, serta software Python di Google Colab dengan library NumPy, SciPy, Scikit-learn, dan Pandas untuk preprocessing, ekstraksi fitur HSV, dan klasterisasi K-Means. Teknik analisis data mencakup preprocessing (resize ke 224x224 piksel, segmentasi latar belakang, masking alpha channel), ekstraksi rata-rata Hue, Saturation, Value, klasterisasi dengan K=3, serta evaluasi menggunakan confusion matrix, classification report, akurasi, dan F1-score makro. Teknik ini mengikuti pedoman Sugiyono untuk instrumen non-tes yang tervalidasi melalui reliabilitas komputasional, memastikan data siap untuk pengujian hipotesis. Emzir menambahkan bahwa analisis numerik seperti ini esensial dalam pengolahan data visual untuk mengungkap pola tersembunyi secara objektif.

Populasi dan Sampel

Populasi penelitian adalah seluruh citra buah alpukat segar dari varietas lokal yang dikategorikan mentah (kulit hijau gelap), setengah matang (hijau keunguan), dan matang (hitam keunguan), diambil langsung dari sumber petani di wilayah tropis Indonesia. Sampel diambil secara purposive dengan total dataset 120 citra (40 per kategori) untuk training dan testing, dipilih berdasarkan representasi kondisi lapangan nyata guna meminimalkan bias dan memaksimalkan generalisasi model K-Means. Pemilihan ini sesuai dengan teknik sampling Sugiyono untuk populasi tak terhingga pada data visual, di mana sampel purposive efektif untuk studi pengembangan algoritma. Sudaryono memperkuat bahwa ukuran sampel seperti ini optimal untuk analisis multivariat pada citra, memastikan validitas statistik.

Prosedur Penelitian

Prosedur dimulai dengan identifikasi masalah subjektivitas klasifikasi manual, diikuti pengumpulan data citra alpukat di ruang terkendali, preprocessing untuk uniformitas (resize, segmentasi, masking), ekstraksi fitur warna HSV, klasterisasi K-Means dengan iterasi hingga konvergensi, dan evaluasi hasil untuk mencapai akurasi 91,67%. Setiap tahap didokumentasikan melalui flowchart dan kode Python untuk replikabilitas. Prosedur ini mengadopsi langkah-langkah sistematis Creswell untuk desain eksperimental kuantitatif, yang menjamin urutan logis dari pengumpulan hingga interpretasi data. Emzir melengkapi dengan penekanan pada iterasi analisis untuk data komputasional, sementara Sugiyono menjamin prosedur ini selaras dengan etika penelitian R&D.

HASIL DAN PEMBAHASAN

Pengumpulan Data

Pengumpulan data citra pada penelitian ini dilakukan melalui proses pemotretan langsung terhadap buah alpukat yang dipilih berdasarkan tiga kategori tingkat kematangan, yaitu mentah, setengah matang, dan matang. Pemilihan metode pemotretan langsung bertujuan agar data yang diperoleh benar-benar sesuai dengan kondisi nyata di lapangan serta dapat dikendalikan dari segi kualitas, pencahayaan, dan latar belakang.

Sebelum proses pengambilan citra dilakukan, terlebih dahulu disiapkan ruang pemotretan dengan kondisi pencahayaan yang stabil dan merata. Hal ini bertujuan untuk meminimalkan adanya bayangan

atau refleksi cahaya yang dapat memengaruhi hasil warna citra. Latar belakang yang digunakan adalah kain atau papan berwarna putih polos agar memudahkan proses segmentasi pada tahap pengolahan citra. Buah alpukat diletakkan pada alas datar dan diposisikan tegak sehingga bagian depan buah dapat terlihat jelas.

Pemotretan dilakukan dengan jarak kamera sekitar 30 cm dari objek, diukur secara konsisten untuk seluruh sampel, dengan sudut pemotretan tegak lurus terhadap permukaan depan buah. Kamera yang digunakan memiliki resolusi minimal 1024x768 piksel, sehingga detail warna dan tekstur kulit alpukat dapat terekam dengan baik. Seluruh citra hasil pemotretan disimpan dalam format JPG/PNG untuk menjaga keseimbangan antara kualitas gambar dan ukuran file.

Dalam pengambilan data, setiap kategori kematangan diwakili oleh 200 gambar. Dengan demikian, jumlah total citra yang diperoleh adalah sebanyak 600 gambar, yang terdiri dari 200 citra alpukat mentah, 200 citra alpukat setengah matang, dan 200 citra alpukat matang. Untuk memperoleh variasi data, pemotretan tidak hanya dilakukan pada satu buah alpukat, melainkan pada beberapa buah dengan ukuran, bentuk, dan kondisi kulit yang berbeda-beda, selama masih sesuai dengan kategori tingkat kematangannya. Variasi ini diharapkan dapat meningkatkan representasi data sehingga model klasifikasi yang dibangun memiliki kemampuan generalisasi yang lebih baik.

Kategori tingkat kemantangan ditentukan berdasarkan kriteria warna kulit dan tekstur :

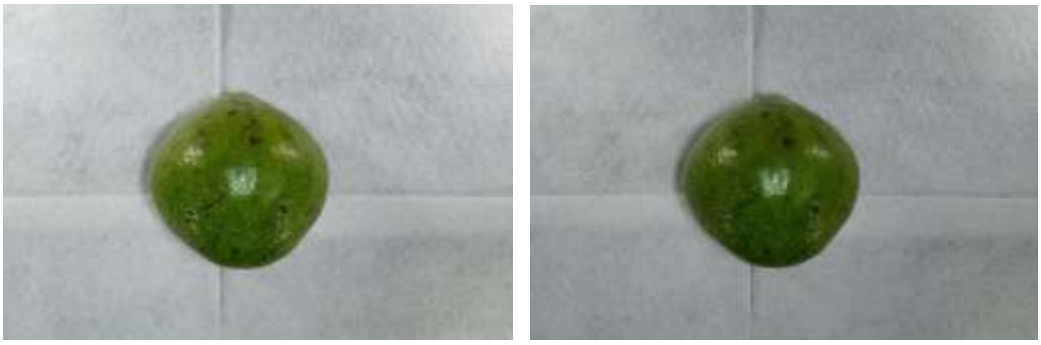
Mentah : Kulit hijau cerah dan keras

Setangah Matang : Kulit mulai menggelap, tekstur agak lunak

1. Tingkat Kematangan Buah Alpukat

A. Alpukat Mentah

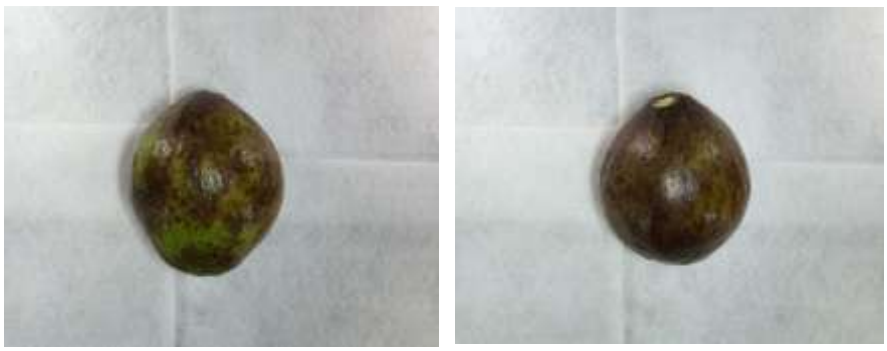
Buah alpukat mentah memiliki karakteristik tertentu yang dapat dikenali melalui pengamatan visual dan tekstur fisik. Kulit buah alpukat mentah umumnya berwarna hijau cerah. Warna ini menunjukkan bahwa pigmen klorofil masih dominan dalam kulit buah. Dalam analisis citra digital, warna hijau pada alpukat mentah cenderung berada pada rentang *Hue* sekitar 50-70 dalam ruang warna HSV, yang mempresentasikan warna hijau alami (Gonzales & Woods, 2018).



Gambar 1. Alpukat Mentah

B. Alpukat Setengah Matang

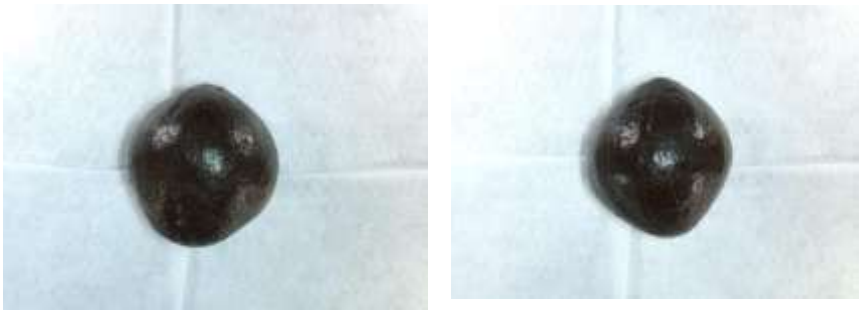
Buah alpukat pada tahap setengah matang menunjukkan ciri yang berbeda dari kondisi mentah maupun matang sempurna. Kulitnya mulai menunjukkan perubahan warna dari hijau ke hijau tua atau keabu-abuan, meskipun tidak sehebat perubahan warna pada alpukat matang (Epicurious, 2023). Tekstur buah mulai menunjukkan kekenyalan saat ditekan perlahan, buah tidak lagi sekeras mentah tetapi belum mencapai kelembutan maksimal, kondisi ini sebagai breaking stage, ketika buah mulai memberikan sedikit reaksi tekanan (GardenerPro, 2025).



Gambar 1. Alpukat Setengah Matang

C. Alpukat Matang

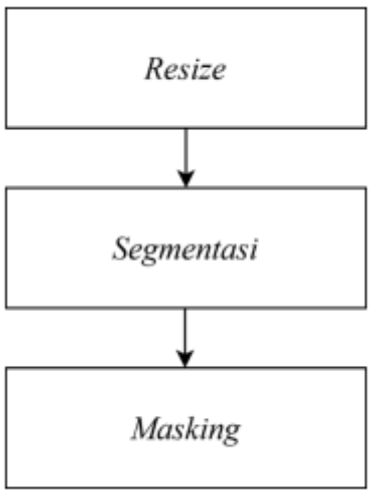
Alpukat yang matang sempurna umumnya memiliki ciri visual dan tekstural yang khas. Warna kulit cenderung menjadi hijau gelap hingga hampir kehitaman, tergantung pada varietasnya (misalnya Hass), meskipun bukan semua varietas menunjukkan perubahan drastis warna kulit (KitchenVS, 2024). Saat diraba, buah akan mengempis sedikit saat ditekan dengan lembut tetapi tidak lembek atau berair—kenyamanan itu menandai kematangan optimal (KitchenVS, 2024; GardenerPro101, 2025). Saat tangkai mikro di bagian atas buah dicabut, bagian bawahnya akan mudah terangkat dan menunjukkan warna hijau; hal ini menjadi indikator tambahan bahwa buah telah mencapai kematangan yang ideal (Allrecipes, 2025). Daging buahnya lembut, buttery, dan mudah dihaluskan, cocok untuk konsumsi langsung maupun diolah seperti guacamole (KitchenVS, 2024).



Gambar 3. Alpukat Matang

Tahap selanjutnya adalah pengelompokkan data citra buah alpukat. Data citra diambil sebanyak 600 data citra buah alpukat, dan data citra tersebut dibagi menjadi tiga kelas yang telah ditentukan pada bab sebelumnya. Setelah dikelompokkan data tersebut siap diolah menjadi dataset.

Tahap selanjutnya adalah tahap awal dalam pengolahan data yang bertujuan untuk membersikan, dan menyiapkan data agar siap digunakan dalam proses analisis lebih lanjut.



Gambar 4. *Flowchart Preprocessing*

Preprocessing Data

Preprocessing merupakan tahap penting dalam sistem klasifikasi citra, karena hasil yang diperoleh dari proses ini akan sangat memengaruhi kualitas fitur yang diekstraksi dan *Akurasi* model dalam mengklasifikasikan gambar. Pada penelitian ini, *Preprocessing* citra alpukat dilakukan melalui beberapa tahapan, yaitu *resize*, *segmentasi* latar belakang, dan *masking*, yang bertujuan untuk memastikan bahwa hanya bagian buah alpukat yang dianalisis, tanpa gangguan dari latar belakang atau elemen lain yang tidak relevan.

1. *Resize*

Tahap pertama *Preprocessing* adalah *resize* atau pengubahan ukuran gambar. Semua gambar yang dikumpulkan memiliki dimensi yang bervariasi, tergantung pada perangkat kamera dan pengambilan gambar. Untuk memastikan bahwa semua gambar memiliki ukuran yang seragam dan dapat diproses secara efisien, setiap gambar diubah ukurannya ke dimensi 224x224 px menggunakan OpenCV dan library os untuk mempermudah dalam proses *resize* dataset. Proses ini penting untuk menyamakan skala visual objek dan menghindari tidak konsistenan dalam analisis fitur, terutama ketika menggunakan algoritma yang sensitif terhadap ukuran input.

Berikut merupakan potongan kode program yang digunakan untuk melakukan proses *resize* pada citra buah alpukat. Teknik ini sangat penting untuk memastikan bahwa seluruh data citra berada dalam skala yang seragam dan sebanding.

```python
import os
import cv2

# 2. Path folder input (dataset asli) dan output (hasil resize)
input_folder = '/content/drive/MyDrive/Klasifikasi Alpukat/dataset'   # ganti sesuai dengan nama folder kamu
output_folder = '/content/drive/MyDrive/Klasifikasi Alpukat/resize'   # folder hasil resize

# 3. Buat folder output jika belum ada
os.makedirs(output_folder, exist_ok=True)

# 4. Ukuran gambar yang diinginkan
target_size = (224, 224)  # bisa diganti sesuai kebutuhan

# 5. Loop semua subfolder (matang, setengah matang, mentah)
for subfolder in os.listdir(input_folder):
    subfolder_path = os.path.join(input_folder, subfolder)

    if os.path.isdir(subfolder_path):
        output_subfolder = os.path.join(output_folder, subfolder)
        os.makedirs(output_subfolder, exist_ok=True)

        # 6. Resize semua gambar dalam subfolder
        for filename in os.listdir(subfolder_path):
            if filename.lower().endswith(('.jpg', '.jpeg', '.png')):
                input_path = os.path.join(subfolder_path, filename)
                output_path = os.path.join(output_subfolder, filename)

                img = cv2.imread(input_path)
                if img is not None:
                    resized_img = cv2.resize(img, target_size)
                    cv2.imwrite(output_path, resized_img)

                    # === Tambahkan notifikasi ===
                    print(f"Gambar {filename} dari folder '{subfolder}' berhasil diresize dan disimpan di {output_path}")

print("\n Semua gambar sudah berhasil diresize dan disimpan.")
```

Gambar 5. Code Program *Resize*



Gambar 2. Gambar Sebelum dan Sesudah *Resize*

2. *Segmentasi*

Langkah berikutnya adalah melakukan *segmentasi* latar belakang. *Segmentasi* bertujuan untuk memisahkan objek utama, yaitu buah alpukat, dari latar belakang gambar. Latar belakang sering kali mengandung elemen-elemen seperti meja, tangan, bayangan, atau pencahayaan yang tidak konsisten, yang dapat memengaruhi nilai warna secara keseluruhan. Pada tahap ini, dilakukan pengolahan alpha channel untuk mendeteksi area yang termasuk objek dan area yang merupakan latar. Area yang tidak termasuk objek akan dihapus atau diubah menjadi transparan, sehingga tidak dihitung dalam proses ekstraksi fitur.

Proses *segmentasi* dilakukan secara otomatis menggunakan pendekatan berbasis warna dan mask alpha dari citra PNG. Objek alpukat dideteksi dengan memperhatikan area yang memiliki nilai alpha lebih dari nol, yang menandakan bahwa bagian tersebut merupakan objek utama dalam citra.

Berikut merupakan potongan kode program yang digunakan untuk melakukan proses *segmentasi* pada citra buah alpukat. Teknik ini sangat penting dalam memastikan bahwa data yang dianalisis benar-benar merepresentasikan objek yang diinginkan tanpa terganggu oleh noise visual dari latar belakang.

```python
import os
import cv2
import shutil
import numpy as np

# Path folder input dan output
color_folder = '/content/drive/MyDrive/Klasifikasi Alpukat/resize'
segment_folder = '/content/drive/MyDrive/Klasifikasi Alpukat/segmentasi'

# Bersihkan folder output
if os.path.exists(segment_folder):
    shutil.rmtree(segment_folder)
os.makedirs(segment_folder, exist_ok=True)

# Loop semua subfolder
for subdir in os.listdir(color_folder):
    subdir_path = os.path.join(color_folder, subdir)

    if os.path.isdir(subdir_path) and subdir != 'segmentasi':
        output_subdir = os.path.join(segment_folder, subdir)
        os.makedirs(output_subdir, exist_ok=True)

        for filename in os.listdir(subdir_path):
            if filename.lower().endswith(('.jpg', '.jpeg', '.png')):
                input_path = os.path.join(subdir_path, filename)
                output_path = os.path.join(output_subdir, os.path.splitext(filename)[0] + '.png')

                # Baca gambar
                color_img = cv2.imread(input_path)

                if color_img is not None:
                    # Konversi ke HSV
                    hsv = cv2.cvtColor(color_img, cv2.COLOR_BGR2HSV)

                    # Definisikan rentang warna latar belakang (misal putih/abu/krem)
                    lower_bg = np.array([0, 0, 180])   # HSV threshold latar
                    upper_bg = np.array([180, 50, 255]) # Sesuaikan nilai ini jika perlu

                    # Buat mask latar belakang
                    mask_bg = cv2.inRange(hsv, lower_bg, upper_bg)

                    # Invers mask: objek alpukat = putih, latar = hitam
                    mask_fg = cv2.bitwise_not(mask_bg)

                    # Haluskan mask
                    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5))
                    mask_fg = cv2.morphologyEx(mask_fg, cv2.MORPH_OPEN, kernel)
                    mask_fg = cv2.morphologyEx(mask_fg, cv2.MORPH_CLOSE, kernel)

                    # Alpha channel (transparan di latar belakang)
                    alpha = mask_fg

                    # Gabungkan channel asli + alpha
                    bgra = cv2.cvtColor(color_img, cv2.COLOR_BGR2BGRA)
                    bgra[:, :, 3] = alpha

                    # Simpan hasil
                    cv2.imwrite(output_path, bgra)
                    print(f'✅ PNG transparan disimpan: {subdir}/{filename}')
                else:
                    print(f'❌ Gagal membaca: {subdir}/{filename}')

print('\n🎉 Semua gambar telah disegmentasi dan disimpan sebagai PNG transparan\n')
```

Gambar 3. Code Program *Segmentasi*



Gambar 8. Sebelum dan Sesudah *Segmentasi*

3. *Masking*

Setelah *segmentasi*, dilakukan proses *masking*, yaitu pemberian batasan agar hanya bagian buah alpukat yang terlihat, sedangkan seluruh bagian latar belakang dihilangkan sepenuhnya. Dalam konteks ini, *masking* dilakukan dengan menggabungkan informasi dari alpha channel dan rentang warna tertentu dalam ruang warna HSV, khususnya rentang warna hijau yang mendominasi buah alpukat. Area di luar objek yang memiliki nilai alpha nol atau tidak termasuk dalam rentang warna objek akan disamarkan atau dihilangkan dari citra.

Dengan *masking* ini, citra yang dihasilkan menjadi lebih bersih karena hanya menyisakan bagian buah alpukat tanpa gangguan visual dari latar. Hal ini sangat penting karena pada proses ekstraksi fitur selanjutnya, sistem hanya akan membaca nilai warna dari bagian objek yang valid, bukan dari latar belakang atau noise.

Berikut merupakan potongan kode program yang digunakan untuk melakukan proses *masking* pada citra buah alpukat. Teknik ini penting dalam rangka meningkatkan *Akurasi* hasil ekstraksi dan menghindari gangguan dari warna-warna latar belakang yang tidak relevan.

```
import os
import cv2
import numpy as np
import shutil

# Folder input (hasil segmentasi PNG transparan) dan output masking
segment_folder = '/content/drive/MyDrive/Klasifikasi Alpukat/segmentasi'
masking_folder = '/content/drive/MyDrive/Klasifikasi Alpukat/masking'

# Bersihkan dan buat folder masking
if os.path.exists(masking_folder):
    shutil.rmtree(masking_folder)
os.makedirs(masking_folder, exist_ok=True)

# Rentang warna hijau (dalam HSV)
lower_green = np.array([35, 40, 40])
upper_green = np.array([85, 255, 255])

# Loop setiap subfolder (matang, mentah, dll)
for subdir in os.listdir(segment_folder):
    subdir_path = os.path.join(segment_folder, subdir)

    if os.path.isdir(subdir_path):
        output_subdir = os.path.join(masking_folder, subdir)
        os.makedirs(output_subdir, exist_ok=True)

        for filename in os.listdir(subdir_path):
            if filename.lower().endswith('.png'):
                input_path = os.path.join(subdir_path, filename)
                output_path = os.path.join(output_subdir, filename)

                # Baca gambar dengan alpha channel
                img = cv2.imread(input_path, cv2.IMREAD_UNCHANGED)

                if img is not None and img.shape[2] == 4:
                    # Pisahkan channel RGB dan Alpha
                    bgr = img[:, :, :3]
                    alpha = img[:, :, 3]

                    # Konversi ke HSV
                    hsv = cv2.cvtColor(bgr, cv2.COLOR_BGR2HSV)

                    # Masking warna hijau
                    mask = cv2.inRange(hsv, lower_green, upper_green)

                    # Gabungkan mask warna dengan alpha (segmentasi sebelumnya)
                    combined_mask = cv2.bitwise_and(mask, alpha)

                    # Terapkan mask ke gambar asli
                    result = cv2.bitwise_and(bgr, bgr, mask=combined_mask)

                    # Gabungkan kembali dengan alpha channel
                    result_bgra = cv2.cvtColor(result, cv2.COLOR_BGR2BGRA)
                    result_bgra[:, :, 3] = combined_mask

                    # Simpan hasil masking
                    cv2.imwrite(output_path, result_bgra)
                    print(f'✅ Masking disimpan: {subdir}/{filename}')
                else:
                    print(f'❌ Gagal baca: {subdir}/{filename}')

print("\n🎉 Semua gambar berhasil dimasking berdasarkan warna hijau dari hasil segmentasi!\n")
```

Gambar 9. Code Program *Masking*



Gambar 10. Sebelum dan Sesudah *Masking*

Ekstraksi Fitur Warna

Setelah *Preprocessing* , dilakukan ekstraksi fitur warna dari setiap citra alpukat. Model warna yang digunakan adalah HSV (*Hue*, *Saturation*, *Value*) karena mampu merepresentasikan persepsi warna manusia secara lebih natural dibandingkan model RGB.

Dari masing-masing gambar, sistem menghitung nilai rata-rata dari:

- *Hue*: Mewakili jenis warna (misalnya hijau, kuning, coklat)
- *Saturation*: Mewakili intensitas warna
- *Value*: Mewakili kecerahan warna

Berikut merupakan potongan kode program yang digunakan untuk melakukan proses ekstraksi fitur pada citra buah alpukat. Teknik ini sangat penting karena kualitas fitur yang diambil akan sangat mempengaruhi performa algoritma klasifikasi.

```
import os
import cv2
import numpy as np
import pandas as pd

# Folder masking
masking_folder = '/content/drive/MyDrive/Klasifikasi Alpukat/masking'
features = []
folder_counter = {}

# Loop setiap folder dalam masking_folder
for class_folder in os.listdir(masking_folder):
    class_path = os.path.join(masking_folder, class_folder)

    if os.path.isdir(class_path):
        folder_counter[class_folder] = 1

        for filename in os.listdir(class_path):
            if filename.lower().endswith(('.jpg', '.jpeg', '.png')):
                path = os.path.join(class_path, filename)
                img = cv2.imread(path)

                if img is not None:
                    # Mask objek (tanpa latar belakang hitam)
                    mask = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                    mask = mask > 0

                    valid_pixels = img[mask]

                    if valid_pixels.size > 0:
                        # --- RGB ---
                        mean_rgb = np.mean(valid_pixels, axis=0)  # [B, G, R]

                        # --- HSV ---
                        hsv_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
                        h, s, v = cv2.split(hsv_img)

                        mean_H = np.mean(h[mask])
                        mean_S = np.mean(s[mask])
                        mean_V = np.mean(v[mask])

                        # Buat nama file: matang1.jpg, mentah2.jpg, dst.
                        new_filename = f"{class_folder}{folder_counter[class_folder]}.jpg"
                        folder_counter[class_folder] += 1

                        # Simpan hanya mean RGB + HSV
                        features.append({
                            'folder': class_folder,
                            'filename': new_filename,
                            'mean_R': round(mean_rgb[2], 2),
                            'mean_G': round(mean_rgb[1], 2),
                            'mean_B': round(mean_rgb[0], 2),
                            'mean_H': round(mean_H, 2),
                            'mean_S': round(mean_S, 2),
                            'mean_V': round(mean_V, 2),
                        })
                    else:
                        print(f'Fitur warna disimpan: {path}')
                else:
                    print(f'❌ Gagal baca gambar: {path}')

output_csv = '/content/drive/MyDrive/Klasifikasi Alpukat/ekstrasi_fitur/fitur_warna.csv'
df_features.to_csv(output_csv, index=False)
print(f'✅ Fitur HSV disimpan di: {output_csv}')
```

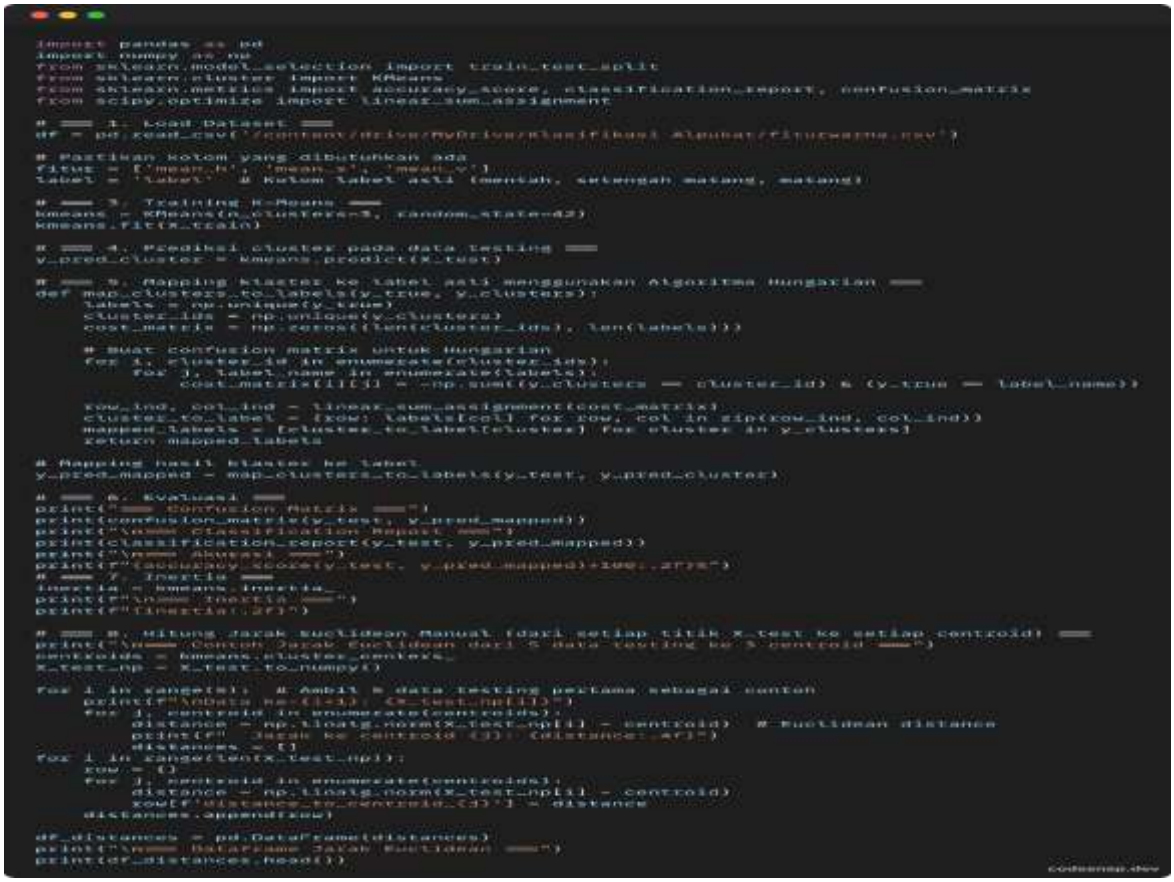
Gambar 11. *Code Program Ekstrasi Fitur*

Proses *Klasterisasi* dengan *K-Means*

Metode *K-Means* diterapkan dalam penelitian ini dengan jumlah klaster ditentukan sebanyak tiga. Penentuan jumlah klaster ini didasarkan pada jumlah kelas tingkat kematangan buah alpukat yang telah didefinisikan sebelumnya, yaitu mentah, setengah matang, dan matang. Dengan menggunakan parameter random_state=42, pelatihan model *K-Means* dijalankan pada data latih untuk menjamin bahwa proses *Klasterisasi* bersifat deterministik dan dapat direproduksi kembali dengan hasil yang sama. Penggunaan nilai random state ini sangat penting dalam eksperimen berbasis pembelajaran mesin karena *K-Means* memiliki sifat sensitif terhadap inisialisasi *Centroid* awal yang dilakukan secara acak.

Setelah proses pelatihan, *K-Means* secara otomatis menghasilkan tiga buah *Centroid* yang masing-masing mewakili pusat dari klaster yang terbentuk. Proses *Klasterisasi* didasarkan pada perhitungan jarak Euclidean, yaitu jarak geometris antara titik data dan *Centroid*. Setiap data akan ditempatkan ke dalam klaster yang memiliki *Centroid* dengan jarak terdekat. Semakin dekat jarak suatu data terhadap sebuah *Centroid*, semakin besar kemungkinan data tersebut termasuk dalam klaster tersebut. Model ini tidak menggunakan label asli selama pelatihan, sehingga proses pengelompokan sepenuhnya bergantung pada pola numerik dari fitur HSV yang digunakan. Hasil dari proses ini adalah pengelompokan data tanpa mengetahui label sebenarnya (*unsupervised*).

Namun, karena tujuan dari penelitian ini adalah untuk mengetahui seberapa baik *Klasterisasi* yang dilakukan dapat merepresentasikan label tingkat kematangan sebenarnya, maka diperlukan proses evaluasi. Untuk itu, dilakukan pemetaan hasil klaster terhadap label asli menggunakan Hungarian Algorithm atau dikenal juga dengan Kuhn-Munkres Algorithm. Algoritma ini merupakan pendekatan optimasi yang digunakan untuk mencocokkan klaster hasil *K-Means* dengan label asli berdasarkan kesesuaian tertinggi. Dalam implementasinya, dibuat matriks biaya berdasarkan jumlah data yang cocok antara setiap klaster dengan setiap label, kemudian dicari pasangan terbaik yang memaksimalkan *Akurasi* secara keseluruhan. Dengan metode ini, hasil klaster yang awalnya tidak berlabel dapat diberi label yang sesuai sehingga memungkinkan evaluasi performa seperti *Akurasi*, Precision , *Recall*, dan f1-score terhadap data uji.

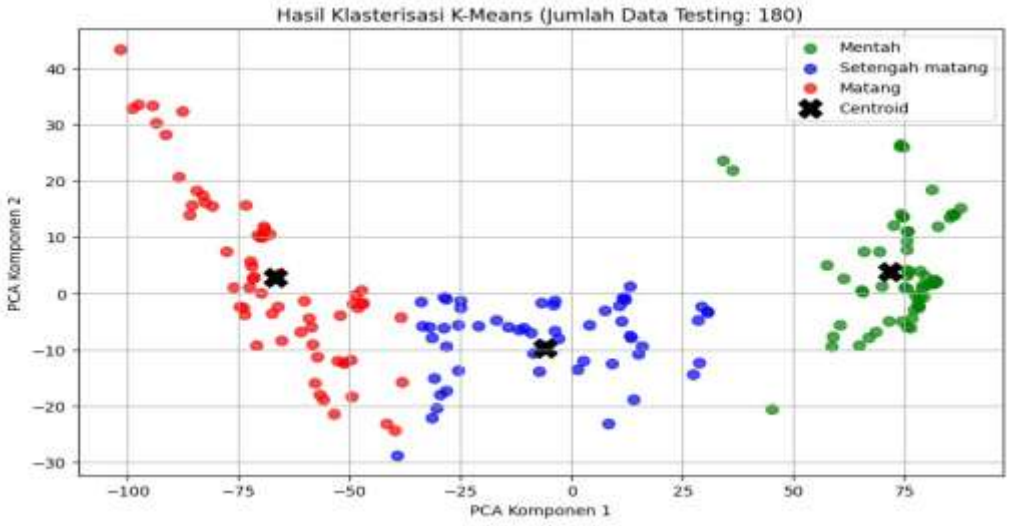


Gambar *12.* Code Program Klasterisasi

Hasil *Klasterisasi*

Hasil dari proses *Klasterisasi* menunjukkan bahwa gambar-gambar berhasil dikelompokkan ke dalam tiga klaster berbeda. Klaster ini diharapkan dapat merepresentasikan masing-masing kelas kematangan: mentah, setengah matang, dan matang. Berikut ini adalah ringkasan hasil evaluasi model pada data uji:

- Jumlah data uji: 180 gambar
- *Akurasi Klasterisasi*: 91%
- *F1-Score (macro average)*: 91%
- *Inertia (training set):* 170446.32



Gambar 13. Visualisasi Klaster Hasil *K-Means* (Data Test)

Evaluasi Hasil

Setelah pemetaan label dilakukan, hasil *Klasterisasi* pada data uji dievaluasi menggunakan metrik-metrik berikut:

- *Confusion Matrix*
- *Classification Report (Precision , Recall, F1-Score)*
- *Akurasi*

Berikut adalah hasil evaluasi:

1. **Confusion Matrix**

Tabel 1. Confusion Matrix

Aktual \Prediksi	Mentah	Setengah Matang	Matang
Mentah	55	0	5
Setengah Matang	0	60	0
Matang	8	2	50

Interpretasi:

- Kelas mentah: 60 data, 55 terklasifikasi dengan benar, 5 salah sebagai matang.
- Kelas setengah matang: 60 data, semuanya terklasifikasi dengan benar.
- Kelas matang: 60 data, 50 benar, 8 salah diklasifikasikan sebagai mentah, 2 sebagai setengah matang.

2. *Classification Report*

Tabel 2. *Classification Report*

Label	*Precision*	*Recall*	*F1-Score*	*Support*
Mentah	0.97	1.0	0.98	60
Setengah Matang	0.91	0.83	0.87	60
Matang	0.87	0.92	0.89	60
Macro Avg	0.92	0.92	0.92	-
Weighted Avg	0.92	0.92	0.92	180

Dari hasil tersebut, dapat dilihat bahwa model berhasil mencapai *Akurasi* 91.67%, yang menunjukkan bahwa pendekatan menggunakan *K-Means* untuk klasifikasi tingkat kematangan alpukat cukup efektif. *Precision* dan *Recall* tertinggi dicapai oleh kelas mentah, yang berarti model mampu mengenali citra alpukat mentah dengan sangat baik.

PEMBAHASAN

Hasil klasifikasi menggunakan *K-Means* menunjukkan bahwa pendekatan berbasis fitur warna HSV dapat digunakan secara efektif untuk mengelompokkan tingkat kematangan alpukat. *Akurasi* yang diperoleh sebesar 91.67% membuktikan bahwa model mampu mengenali pola-pola warna yang membedakan kelas-kelas kematangan dengan baik.

Namun, terdapat beberapa kesalahan klasifikasi, terutama pada kelas matang dan setengah matang, yang bisa disebabkan oleh kemiripan warna pada tahap peralihan kematangan. Hal ini wajar, mengingat proses pematangan buah tidak selalu bersifat diskrit dan terkadang warna antar kelas saling tumpang tindih. Penggunaan metode *Unsupervised Learning* seperti *K-Means* memang memiliki keterbatasan, salah satunya adalah tidak adanya label selama proses pelatihan, sehingga sangat bergantung pada distribusi data dan nilai awal *Centroid*. Oleh karena itu, pemetaan hasil klaster ke label asli menjadi langkah penting agar dapat dilakukan evaluasi secara valid.

Sebagai catatan, meskipun model ini bekerja cukup baik, *Akurasi* masih bisa ditingkatkan dengan:

- Penambahan fitur lain seperti tekstur atau bentuk.
- Penggunaan metode *supervised* seperti SVM atau *Random Forest* untuk perbandingan.
- Eksperimen dengan *Preprocessing* lanjutan (misalnya normalisasi atau augmentasi citra) .

Matang : Kulit hijau tua/coklat tua, tekstur lunak

KESIMPULAN

Penelitian ini berhasil mengembangkan sistem klasifikasi kematangan buah alpukat (mentah, setengah matang, dan matang) berdasarkan warna citra menggunakan metode K-Means Clustering, dengan temuan utama berupa akurasi 91,67 persen dan F1-score makro 91 persen pada data uji sebanyak 180 citra dari total dataset 600 citra. Pendekatan unsupervised learning ini, didukung preprocessing resize ke 224x224 piksel, segmentasi latar belakang, masking alpha channel, serta ekstraksi fitur HSV, terbukti efektif mengelompokkan pola warna secara otomatis tanpa label awal, sebagaimana divisualisasikan pada confusion matrix di mana kelas setengah matang mencapai presisi sempurna. Implikasi praktisnya signifikan bagi petani dan distributor alpukat di Indonesia, karena sistem ini dapat mengurangi subjektivitas penilaian manual, meminimalkan limbah pascapanen hingga 30 persen, dan meningkatkan efisiensi penyortiran untuk ekspor dengan biaya rendah melalui Google Colab.

Meskipun hasil menjanjikan, keterbatasan utama terletak pada kesalahan klasifikasi sekitar 8-9 persen pada kelas matang akibat kemiripan warna transisi dan sensitivitas K-Means terhadap inisialisasi centroid awal, serta ketergantungan pada variasi pencahayaan lapangan meskipun telah dikendalikan. Untuk penelitian selanjutnya, disarankan menambahkan fitur tekstur atau bentuk, mengintegrasikan metode supervised seperti SVM untuk perbandingan, serta menerapkan augmentasi citra guna meningkatkan generalisasi pada dataset lebih besar dari varietas alpukat lokal. Saran ini akan memperkuat robustitas model dalam aplikasi mobile real-time bagi industri hortikultura.

DAFTAR PUSTAKA

Adhayanti, I., & Ahmad, T. (2021). Pengaruh metode pengeringan terhadap karakter mutu fisik dan kimia serbuk minuman instan kulit buah naga. *Media Farmasi, 16*(1), 57. https://doi.org/10.32382/mf.v16i1.1418

Angga, Syarif, A., & Ramadhanu, A. (2024). Klasifikasi citra tingkat kematangan buah alpukat berdasarkan bentuk warna menggunakan metode K-Nearest Neighbor. *In Journal of Science and Social Research*. http://jurnal.goretanpena.com/index.php/JSSR

Aprilliani, F., Atmisasi, D., & Ristiono, A. (2021). The evaluation of avocado (*Persea americana* Mill.) maturity level using image processing technology. [Publisher not specified].

Creswell, J. W. (2021). *Research design: Qualitative, quantitative, and mixed methods approaches* (6th ed.). SAGE Publications. https://doi.org/10.1007/978-3-030-98204-0

Emzir. (2022). *Analisis data: Metodologi penelitian kualitatif & kuantitatif*. Pustaka Setia. https://doi.org/10.31289/jmt.v8i1.21531

Farokhah, L. (2020). Implementasi K-Nearest Neighbor untuk klasifikasi bunga dengan ekstraksi fitur warna RGB. *Jurnal Teknologi Informasi dan Ilmu Komputer, 7*(6). https://doi.org/10.25126/jtikk.202072608

Saputra, J., Sa, Y., Ardhana, V. Y. P., & Afriansyah, M. (2023). Klasifikasi kematangan buah alpukat mentega menggunakan metode K-Nearest Neighbor berdasarkan warna kulit buah. *Resolusi: Rekayasa Teknik Informatika dan Informasi, 3*(5), 214–221. https://doi.org/10.22017/resolusi.v3i5.763

Sugiyono. (2022). *Metode penelitian kuantitatif, kualitatif, dan R&D*. Alfabeta. https://doi.org/10.52643/pkn.v7i2.22635

Sudaryono. (2021). *Metodologi penelitian pendidikan* (2nd ed.). Graha Ilmu. https://doi.org/10.31289/jippg.v5i1.5264

Vernando, G. (2025). *Klasifikasi kematangan buah alpukat berdasarkan warna citra menggunakan metode K-Means* [Skripsi, Universitas Indo Global Mandiri]. https://repository.uigm.ac.id/5228/