

DETEKSI KESEGARAN KUBIS DENGAN METODE KNN BERBASIS FITUR GLCM DAN HSV



Di susun oleh :

Dina Amalia Zurli (2211510934)

Maela Alfa Fauza (2211510306)

Siti Nurhalizah (2211510132)

Kelompok Kelas : G3

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK INFORMASI
UNIVERSITAS BUDI LUHUR
JAKARTA
2024**

ABSTRAK

Pendeteksian kesegaran sayur kubis merupakan aspek penting dalam menjaga kualitas dan nilai jual produk pertanian. Penelitian ini mengembangkan sistem pendeteksian kesegaran kubis menggunakan metode K-Nearest Neighbors (KNN) berbasis ekstraksi fitur tekstur dan warna. Fitur tekstur diekstraksi menggunakan metode Gray Level Co-occurrence Matrix (GLCM), sedangkan fitur warna diperoleh melalui representasi model warna HSV (Hue, Saturation, Value). Dataset terdiri dari gambar kubis dengan berbagai tingkat kesegaran, yang diklasifikasikan ke dalam kategori segar dan tidak segar.

Hasil pengujian menunjukkan bahwa kombinasi fitur GLCM dan HSV memberikan akurasi yang signifikan dalam mendekripsi kesegaran kubis. Metode KNN yang digunakan mampu mengklasifikasikan data dengan tingkat akurasi mencapai 92.86%, tergantung pada parameter K dan konfigurasi fitur. Sistem ini diimplementasikan menggunakan Python dengan pustaka open-source, seperti OpenCV untuk pengolahan gambar dan Scikit-learn untuk algoritma klasifikasi.

Pengembangan sistem ini diharapkan dapat memberikan solusi praktis bagi petani, pedagang, dan konsumen dalam menilai kesegaran kubis secara cepat dan efisien, sehingga dapat meningkatkan manajemen distribusi dan mengurangi kerugian pascapanen.

Kata Kunci: Pendekripsi kesegaran, kubis, KNN, GLCM, HSV, pengolahan citra

ABSTRACT

Detecting the freshness of cabbage is an important aspect in maintaining the quality and selling value of agricultural products. This study develops a cabbage freshness detection system using the K-Nearest Neighbors (KNN) method based on texture and color feature extraction. Texture features are extracted using the Gray Level Co-occurrence Matrix (GLCM) method, while color features are obtained through the representation of the HSV (Hue, Saturation, Value) color model. The dataset consists of cabbage images with various levels of freshness, which are classified into fresh and non-fresh categories.

The test results show that the combination of GLCM and HSV features provides significant accuracy in detecting cabbage freshness. The KNN method used is able to classify data with an accuracy level reaching 92.86%, depending on the K parameter and feature configuration. This system is implemented using Python with open-source libraries, such as OpenCV for image processing and Scikit-learn for classification algorithms.

The development of this system is expected to provide a practical solution for farmers, traders, and consumers in assessing cabbage freshness quickly and efficiently, so as to improve distribution management and reduce post-harvest losses.

Keywords: Freshness detection, cabbage, KNN, GLCM, HSV, image processing

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sayur kubis merupakan salah satu komoditas pertanian yang banyak dikonsumsi masyarakat karena kandungan nutrisinya yang tinggi dan harga yang terjangkau. Namun, kualitas kesegaran kubis sangat memengaruhi nilai jualnya di pasar. Kesegaran kubis juga menjadi indikator utama dalam menentukan tingkat kesesuaian konsumsi dan daya tahan selama penyimpanan. Proses manual dalam mengevaluasi kesegaran kubis seringkali bersifat subjektif, memakan waktu, dan bergantung pada pengalaman individu, sehingga diperlukan suatu sistem otomatis yang akurat dan efisien. banyaknya sayur kol yang dipanen, membuat petani sedikit kerepotan dalam proses menyortir sayur kubis mana yang layak untuk dijual atau tidak. dan seiring berkembangnya teknologi maka ha ini menjadi sangat mungkin untuk membuat komputer melakukan pekerjaan yang dianggap biasa oleh manusia termasuk proses sortir kualitas sayur kol

Dalam penelitian ini, dikembangkan sistem pendekripsi kesegaran kubis menggunakan metode **K-Nearest Neighbors (K-NN)** dengan memanfaatkan fitur tekstur dan warna. Fitur tekstur diperoleh melalui metode **Gray Level Co-occurrence Matrix (GLCM)**, yang mampu merepresentasikan pola tekstur pada permukaan kubis secara kuantitatif. Di sisi lain, fitur warna diambil berdasarkan model warna **Hue, Saturation, Value (HSV)**, yang memiliki kemampuan untuk menggambarkan karakteristik warna kubis secara lebih mendalam dibandingkan model warna RGB.

Metode KNN dipilih karena kesederhanaannya serta kemampuannya dalam mengklasifikasikan data berdasarkan kemiripan dengan tetangga terdekat. Kombinasi fitur GLCM dan HSV diharapkan mampu meningkatkan akurasi klasifikasi dalam membedakan kubis segar dan tidak segar. Sistem ini diimplementasikan menggunakan bahasa pemrograman Python, yang memiliki berbagai pustaka pendukung seperti OpenCV untuk pengolahan citra dan Scikit-learn untuk algoritma klasifikasi.

Penelitian ini bertujuan untuk menghasilkan sistem otomatis yang dapat membantu petani, pedagang, dan konsumen dalam menilai kesegaran kubis dengan cepat dan tepat, sehingga dapat meminimalkan kerugian pascapanen dan meningkatkan efisiensi rantai distribusi.

1.2. Perumusan Masalah

Berdasarkan latar belakang masalah maka identifikasi penelitian ini adalah

:

1. Bagaimana pengaruh ekstraksi fitur *Hue, Saturation, Value (HSV)* dan *Gray Level Co-occurrence Matrix (GLCM)* yang digunakan dalam sistem ini?

2. Bagaimana cara mengimplementasikan metode *Gray Level Co-occurrence Matrix* (GLCM) dengan algoritma *K-Nearest Neighbors* (K-NN) untuk mendeteksi kualitas sayur kubis?

1.3. Tujuan Penelitian

Berdasarkan rumusan permasalahan di atas maka dapat diuraikan bahwa tujuan dari penelitian ini untuk mengetahui :

1. Untuk mengetahui pengaruh ekstraksi fitur *Hue, Saturation, Value* (HSV) dan *Gray Level Co-occurrence Matrix* (GLCM) yang digunakan dalam sistem ini?
2. Untuk mengetahui cara mengimplementasikan metode *Gray Level Co-occurrence Matrix* (GLCM) dengan algoritma *K-Nearest Neighbors* (K-NN) untuk mendeteksi kualitas sayur kubis?

BAB II LANDASAN TEORI

2.1. Kubis

Kubis (*Brassica oleracea* L.) adalah salah satu sayuran yang sangat diminati masyarakat untuk memenuhi kebutuhan gizi sehari-hari. Akibatnya, petani banyak membudidayakan kubis untuk memenuhi permintaan pasar. Selain memiliki kandungan nutrisi yang baik, harga kubis juga relatif terjangkau. Kandungan gizi kubis segar terdiri dari 92% air, 0,2 g lemak, 1,4 g protein, 40 mg kalsium, 70 mg fosfor, 0,1 mg vitamin B1, 50 mg vitamin C, dan 0,5 mg zat besi. Kandungan air yang tinggi membuat kubis rentan terhadap serangan bakteri patogen, yang dapat menyebabkan proses dekomposisi berlangsung dengan cepat. Contoh gambar kubis segar dan busuk dapat ditunjukkan pada gambar 2.1.



a) Gambar Kubis Busuk



b) Gambar Kubis Segar

Gambar 2.1 Kubis

2.2 Pengolahan Citra

Citra atau image adalah representasi spasial dari suatu objek yang sebenarnya yang biasanya ditulis dalam koordinat cartesian x-y, dan setiap koordinat yang biasanya ditulis dalam koordinat cartesian x-y, dan setiap koordinat merepresentasikan satu sinyal terkecil dari objek.

Piksel (0,0) terletak pada sudut kiri atas pada sebuah citra, indeks x bergerak ke kanan dan indeks y bergerak ke bawah. Konvensi ini dipakai merujuk pada cara penulisan larik yang digunakan dalam pemrograman komputer. Citra digital merupakan citra yang memiliki fungsi dua variabel $f(x,y)$, dimana x dan y adalah koordinat dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut.

Pengolahan suatu citra juga dapat diartikan sebagai segala bentuk operasi baik itu untuk memperbaiki, menganalisa, atau juga mengubah suatu gambar. Jenis citra ini memiliki spesifikasi yang kompleks jika dibandingkan dengan citra jenis citra grayscale, hasil yang diperoleh dari jenis citra ini sangat menyerupai dengan warna objek asli. Rentang warna yang sangat luas dalam pencampuran warna dapat ditemukan pada warna merah, hijau, dan biru. RGB juga dapat dikatakan sebagai warna primer (pokok). Intensitas pada suatu warna memiliki nilai tersendiri dengan maksimal 255 (8 bit).

2.3 *K-Nearest Neighbors (K-NN)*

K-Nearest Neighbor (KNN) merupakan salah satu algoritma klasifikasi Supervised learning, yaitu suatu objek yang dekat satu sama lain dan memiliki karakter yang mirip. KNN bertujuan untuk mengklasifikasi objek baru berdasarkan atribut atau ciri dari sampel data latih yang sudah ada pada sistem. Dengan berdasarkan jarak ketetanggaan terdekat dari data uji ke data latih. Langkah-langkah perhitungan algoritma KNN :

- a. Menentukan nilai K atau tetangga terdekat dari data latih terhadap data uji
- b. Menghitung jarak dengan Euclidean distance masing-masing objek terhadap data latih yang diberikan.
- c. Mengurutkan objek-objek tersebut dari data yang terkecil ke terbesar.
- d. Mengelompokkan data sejumlah K yang telah ditentukan sebelumnya.
- e. Memilih kategori yang paling banyak muncul.

$$d = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2}$$

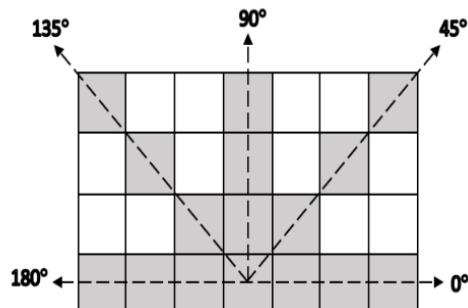
setelah mendapatkan hasil klasifikasi, maka langkah selanjutnya yaitu perhitungan akurasi. Untuk perhitungan akurasi dapat dihitung dengan

menggunakan persamaan berikut. yang dimana x adalah jumlah data yang bernilai benar, dan y adalah jumlah keseluruhan data.

$$Akurasi = \frac{x}{y} \times 100\%$$

2.4 Pengolahan Citra

GLCM (*Gray-Level Co-occurrence Matrix*) adalah metode ekstraksi fitur tekstur yang diperkenalkan oleh Haralick pada tahun 1973. Metode ini menghasilkan 28 fitur yang merepresentasikan pola spasial. Proses GLCM dimulai dengan membentuk matriks co-occurrence, yang menggambarkan hubungan spasial antara piksel referensi dengan tetangganya berdasarkan sudut dan jarak $\backslash(d\backslash)$, di mana jarak $\backslash(d\backslash)$ biasanya bernilai 1 dan diwakili oleh sudut 0° . Dengan memanfaatkan satu piksel dalam lima arah sudut ($0^\circ, 45^\circ, 90^\circ, 135^\circ$, dan 180°), berbagai fitur seperti disimilitas, energi, kontras, homogenitas, dan korelasi dapat diekstraksi dari matriks tersebut (Lamasigi dkk., 2022). Gambar 2.1 menunjukkan arah sudut yang digunakan pada GLCM.



Gambar 2. 2 Arah Sudut GLCM (Lamasigi dkk., 2022)

Dalam penelitian ini ada 5 fitur ekstraksi ciri yang digunakan, diantaranya adalah :

a. Kontras

Kontras (Contrast) dalam GLCM merupakan statistik yang digunakan untuk mengukur tingkat perbedaan intensitas antara pasangan piksel dengan nilai tertinggi (terang) dan terendah (gelap) yang terdapat dalam matriks GLCM. Nilai kontras ini mencerminkan tingkat variasi lokal dalam citra. Kontras dapat dirumuskan sebagai berikut:

$$kontras = \sum_{i,j} |i-j|^2 p(i,j)$$

Dimana:

- $\langle i \rangle$ = tingkat keabuan pada baris ke- $\langle i \rangle$
- $\langle j \rangle$ = tingkat keabuan pada kolom ke- $\langle j \rangle$
- $\langle p(i,j) \rangle$ = probabilitas kemunculan pasangan tingkat keabuan pada baris ke- $\langle i \rangle$ dan kolom ke- $\langle j \rangle$ dalam jarak dan arah tertentu.

b. Korelasi

Korelasi (*Correlation*) dalam GLCM adalah metrik yang digunakan untuk mengukur tingkat kedekatan atau hubungan linear antara pasangan piksel dalam matriks GLCM. Nilai korelasi memberikan gambaran tentang sejauh mana perubahan intensitas pada satu piksel berkaitan dengan perubahan intensitas pada piksel pasangannya. Korelasi dapat dirumuskan sebagai berikut:

$$\text{korelasi} = \sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)p(i,j)}{\sigma_i\sigma_j}$$

Dimana:

- μ_i = rata-rata tingkat keabuan baris ke - i μ_j = rata-rata tingkat keabuan kolom ke - j σ_i = standar deviasi tingkat keabuan baris ke - i.
 σ_j = standar deviasi tingkat keabuan kolom ke - j.

c. Energi

Energi (*Energy*) dalam GLCM adalah metrik yang mengukur kekuatan total atau tingkat keteraturan pasangan piksel dalam matriks GLCM pada tingkat keabuan tertentu. Energi mencerminkan keseragaman tekstur, di mana nilai yang lebih tinggi menunjukkan tekstur yang lebih homogen. Energi dapat dirumuskan sebagai berikut:

$$\text{energi} = \sum_{i,j} p(i,j)^2$$

Dimana:

- i = tingkat keabuan baris ke - i j = tingkat keabuan kolom ke - j $p(i,j)$ = probabilitas kemunculan tingkat keabuan baris ke - i dan kolom ke - j dalam jarak dan arah yang ditentukan.

d. Homogenitas

Homogenitas (*Homogeneity*) dalam GLCM adalah metrik yang mengukur tingkat keseragaman atau kehalusan variasi intensitas keabuan pada

matriks GLCM. Nilai homogenitas yang tinggi menunjukkan bahwa pasangan piksel memiliki nilai intensitas yang serupa. Homogenitas dapat dirumuskan sebagai berikut:

$$homogenitas = \sum_{i,j} \frac{p(i,j)}{1 + |i - j|} \quad (2.4)$$

Dimana:

i = tingkat keabuan baris ke - i j = tingkat keabuan kolom ke - j $p(i,j)$ = probabilitas kemunculan tingkat keabuan baris ke - i dan kolom ke - j dalam jarak dan arah yang ditentukan.

e. Disimilitas

Disimilitas (*Dissimilarity*) dalam GLCM adalah metrik yang mengukur tingkat ketidakmiripan suatu tekstur. Nilai disimilitas akan tinggi jika perbedaan intensitas antar pasangan piksel besar, yang mencerminkan tekstur acak, dan akan rendah jika intensitasnya seragam. Disimilitas dapat dirumuskan sebagai berikut:

$$Dissimilarity = \sum_i \sum_j |i-j|.p(i,j) \quad (2.5)$$

Dimana:

$P(i,j)P(i,j)$ adalah nilai elemen pada posisi (i, j) dalam GLCM.

ii dan jj adalah intensitas tingkat keabuan (gray levels) dari dua piksel yang dibandingkan.

$|i-j||i-j|$ adalah perbedaan absolut antara tingkat keabuan dari pasangan piksel.

2.5 Ruang Warna HSV

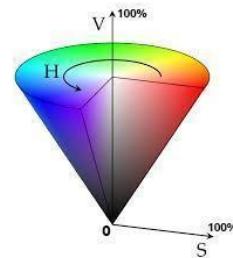
Ruang warna HSV memisahkan warna menjadi tiga komponen utama: Hue (H), Saturation (S), dan Value (V). Komponen-komponen ini membuat HSV lebih intuitif dan praktis dalam mendeskripsikan warna dibandingkan dengan ruang warna RGB.

Hue (H) mewakili jenis warna, seperti merah, biru, atau hijau, dengan rentang nilai 0° hingga 360° .

Saturation (S) menunjukkan intensitas atau kejemuhan warna, dari warna pucat (0%) hingga warna penuh (100%).

Value (V) menggambarkan kecerahan warna, dari gelap (0%) hingga terang (100%).

Pendekatan ini mempermudah analisis dan manipulasi warna dalam berbagai aplikasi, seperti pengolahan citra dan pengenalan pola.



Gambar 2.3 Ilustrasi Ruang Warna HSV

Dari Gambar 2.3 Dapat diketahui bahwa *Hue* mencerminkan posisi warna dasar seperti merah, ungu, kuning, dan lainnya. *Saturation* mengindikasikan tingkat kemurnian atau intensitas warna. Sementara itu, *Value* menunjukkan tingkat kecerahan warna dengan nilai yang berkisar dari 0 hingga 100%. Jika nilai *value* adalah 0, warnanya akan menjadi , sedangkan semakin tinggi nilai *Value*, semakin cerah warna tersebut (Himmah dkk., 2020). Konversi dari ruang warna RGB ke HSV dapat ditentukan dengan persamaan berikut:

$$V = \max (R, G, B) \quad (2.6)$$

$$S = \begin{cases} 0, & \text{Jika } V = 0 \\ 1 - \frac{\min(R, G, B)}{V}, & \text{Jika } V > 0 \end{cases} \quad (2.7)$$

$$H = \begin{cases} 0, & \text{Jika } S = 0 \\ \frac{60 * (G - B)}{S * V}, & \text{Jika } V = R \\ 60 * \left[2 + \frac{B - R}{S * V} \right], & \text{Jika } V = G \\ 60 * \left[4 + \frac{R - G}{S * V} \right], & \text{Jika } V = B \end{cases} \quad (2.8)$$

$$H = H + 360 \quad \text{Jika } H < 0 \quad (2.9)$$

Dimana:

R = Nilai Red

G = Nilai Green

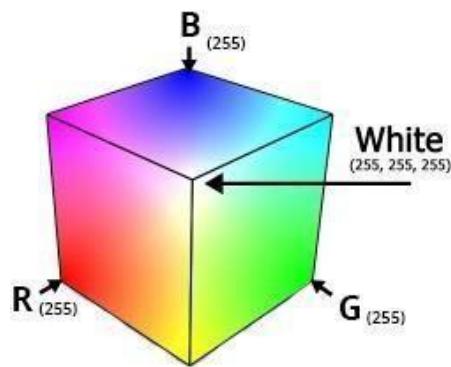
B = Nilai Blue

V = Nilai Perhitungan Value

S = Nilai Perhitungan Saturation
H = Nilai Perhitungan Hue

2.6 Ruang Warna RGB

Ruang warna RGB adalah model warna yang menggunakan kombinasi tiga komponen utama: merah (Red), hijau (Green), dan biru (Blue). Model ini umum digunakan dalam representasi warna digital, seperti pada layar komputer, televisi, dan perangkat elektronik lainnya.



Gambar 2. 4 Ilustrasi Ruang Warna RGB

Dari Gambar 2.4 dapat diketahui bahwa ruang warna RGB menggunakan skala tertentu yang dimulai dari 0 hingga 255. Setiap warna piksel dalam gambar ditentukan oleh kombinasi intensitas Merah (R), Hijau (G), dan Biru (B) yang berada pada lokasi piksel tertentu (Himmah dkk., 2020).

Perhitungan ruang warna RGB dapat ditentukan dengan persamaan berikut:

$$R = \frac{R}{(R+G+B)} \quad (2.10)$$

$$G = \frac{G}{(R+G+B)} \quad (2.11)$$

$$B = \frac{B}{(R+G+B)} \quad (2.12)$$

2.7 Confusion Matrix

Confusion matrix adalah alat yang sering digunakan untuk mengevaluasi performa sebuah model klasifikasi. Alat ini memberikan gambaran detail mengenai prediksi yang benar dan yang salah yang dibuat oleh model. Melalui confusion matrix, kita dapat melihat jumlah prediksi yang tepat serta jumlah kesalahan yang terjadi, baik itu prediksi positif yang salah (*false positives*) atau prediksi negatif yang salah (*false negatives*). Informasi ini sangat berguna untuk menilai efektivitas model dan mengidentifikasi area yang perlu perbaikan.

Selain itu, confusion matrix juga memungkinkan kita untuk menghitung beberapa metrik penting, seperti akurasi, presisi, *recall* (sensitivitas), dan specificity, yang memberikan wawasan lebih mendalam mengenai kinerja model dalam klasifikasi data. Ini membantu dalam memahami seberapa baik model bekerja secara keseluruhan.

$$Recall = \frac{TP}{TP+FN} \quad (2.13)$$

$$Spesificity = \frac{TN}{TN+FP} \quad (2.14)$$

$$Precision = \frac{TP}{TP+FP} \quad (2.15)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.16)$$

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <small>Type I Error</small>
	0 (Negative)	FN (False Negative) <small>Type II Error</small>	TN (True Negative)

Gambar 2.5 *Confusion Matrix*

2.8 Segmentasi dengan Metode *Thresholding*

Menurut Heryanto dkk. (2020), metode ini menggunakan nilai ambang T untuk menentukan apakah sebuah piksel akan diubah menjadi hitam atau putih. Nilai ambang tersebut dihitung menggunakan persamaan:

$$F(x,y) = 255, \text{ jika } f(x,y) \geq T \quad (2.18)$$

$$F(x,y) = 0, \text{ jika } f(x,y) < T \quad (2.19)$$

2.9 Metode *Grayscale*

Gambar *grayscale* hanya memiliki warna tingkat keabuan dan lebih sering digunakan daripada gambar berwarna karena menggunakan lebih sedikit informasi per piksel daripada gambar berwarna. Ini adalah gambar digital dengan satu kanal per-piksel yang dapat diwakili dengan 8 bit, yang mengubah gambar RGB menjadi gambar *grayscale*. Pada gambar *grayscale*, warna abu-abu merah, hijau, dan biru memiliki intensitas yang sama (Nasional dkk., 2024).

BAB III HASIL DAN PEMBAHASAN

Sistem yang dirancang dalam penelitian ini berbasis metode *K-Nearest Neighbors* (KNN) yang dikombinasikan dengan ekstraksi fitur GLCM (*Gray Level Co-occurrence Matrix*) untuk tekstur gambar dan model warna HSV (*Hue, Saturation, Value*) untuk menganalisis kesegaran sayur kubis. Sistem ini menggunakan pemrograman Python dan pustaka open-source seperti OpenCV untuk pengolahan citra dan Scikit-learn untuk algoritma klasifikasi.

Secara lebih spesifik, sistem ini berbasis pada teknik pengolahan citra dan pembelajaran mesin (machine learning), dengan tujuan mendeteksi kesegaran kubis secara otomatis menggunakan pengolahan fitur visual seperti tekstur dan warna.

Fitur GLCM:

- Contrast menunjukkan variasi intensitas pada tekstur kubis, dengan nilai rata-rata sebesar 50.02. Kubis segar cenderung memiliki pola tekstur yang lebih teratur, ditunjukkan oleh nilai Contrast yang lebih rendah.
- Homogeneity memiliki rata-rata 0.197, yang mengindikasikan tingkat homogenitas tekstur pada kubis. Kubis yang segar umumnya memiliki nilai Homogeneity lebih tinggi karena pola tekturnya lebih seragam.
- Correlation rata-rata 0.966, menunjukkan bahwa tekstur kubis memiliki hubungan kuat antar piksel, yang mendukung penilaian kualitas visual.

Fitur HSV:

- Hue (H_mean) memiliki rata-rata 31.82, menunjukkan dominasi warna hijau. Kubis segar memiliki nilai Hue yang lebih mencerminkan warna hijau cerah.
- Saturation (S_mean) dengan rata-rata 16.48, menunjukkan tingkat kejernihan warna. Saturation yang lebih tinggi biasanya ditemukan pada kubis yang lebih segar.
- Value (V_mean) rata-rata 174.51, yang menunjukkan tingkat kecerahan gambar kubis. Kubis segar cenderung memiliki nilai Value lebih tinggi karena permukaannya yang cerah dan tidak kusam.

Di bawah ini adalah contoh input data citra yang telah dilakukan resizing 224 x 224 Piksel :



1. Pengujian *Confusion Matrix*

Hasil dari pengujian akurasi sistem, yang dilakukan dengan *dataset* pengujian yang berisi 15 citra ikan kembung, dengan 50 citra ikan kembung segar dan 50 citra ikan kembung busuk. Hasil *confusion matrix* dapat ditunjukkan pada Gambar 4.14.

	Segar (Pred)	Busuk (Pred)
Segar (True)	7	1
Busuk (True)	0	6

Gambar 4.14

Dari hasil *confusion matrix* pada Gambar 4.14 didapatkan informasi data aktual dan data prediksi. Nilai untuk kelas ikan segar dan ikan busuk dihasilkan dalam pengujian *dataset* untuk model yang telah dilakukan. Berikut adalah perhitungan *Accuracy*, *Recall*, *Precision* dan *F1-Score* pada model yang didapat.

	precision	recall	f1-score	support
Segar	1,00	0,88	0,93	8
Busuk	0,86	1,00	0,92	6
accuracy	0,93	0,93	0,93	0,93
macro avg	0,93	0,94	0,93	14
weighted avg	0,94	0,93	0,93	14

2. Pengumpulan Data

Pengumpulan data dilakukan dengan mengambil citra sayur kubis sebanyak 31 data kemudian di klasifikasai menjadi 2 kelas, yaitu sayur kubis dengan kualitas segar (15 data) dan tidak segar (16 Data).

3. Preprocessing

Preprocessing merupakan proses pengolahan citra dari citra asli menjadi bentuk citra baru seperti pengubahan citra warna ke citra keabuan lalu ke citra hitam putih lalu ke citra HSV.



4. Resized

Resizing merupakan proses mengubah ukuran baik memperbesar atau memperkecil resolusi citra. Matriks yang dihasilkan dari citra seperti ini akan berukuran besar dan membutuhkan waktu yang lama dalam proses komputasi. Sehingga ukuran citra dalam penelitian ini diubah yang sebelumnya 3024 x 3024 piksel menjadi 224 x 224 piksel.

```
64  def main():
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83      # Tentukan ukuran gambar yang akan digunakan untuk resize
84      target_size = (224, 224) # Ukuran target untuk semua gambar
85
86
87
88
89
90
91
92
93
94      # Resize gambar asli dan crop untuk diproses
95      h, w, _ = image.shape
96      cropped_image = image[h//4:3*h//4, w//4:3*w//4] # Crop tengah 50%
97      resized_image = cv2.resize(image, target_size) # Resize ke 224x224
98      gray_image = cv2.cvtColor(resized_image, cv2.COLOR_BGR2GRAY) # Grayscale
99
100
101
102
103      # Resize cropped image dan grayscale image agar ukurannya konsisten
104      cropped_image_resized = cv2.resize(cropped_image, target_size)
105      gray_image_resized = cv2.resize(gray_image, target_size)
```

5. Ekstraksi Citra

Ekstraksi citra dilakukan untuk mendapatkan intensitas warna *Red* (R), *Green* (G) dan *Blue* (B) pada citra sayuran, yang selanjutnya akan diubah kedalam warna abu-abu (*Grayscale*). Pada tahap ini berfungsi untuk melakukan penajaman objek tulisan pada citra agar memudahkan sistem untuk digunakan pada prosesproses berikutnya. Proses awal, citra berwarna RGB (Red, Green, Blue) diubah menjadi format citra keabuan (*Grayscale*). untuk Rumus Grayscale :

$$(0.2989*R)+(0.5870*G)+(0.1140*B)$$

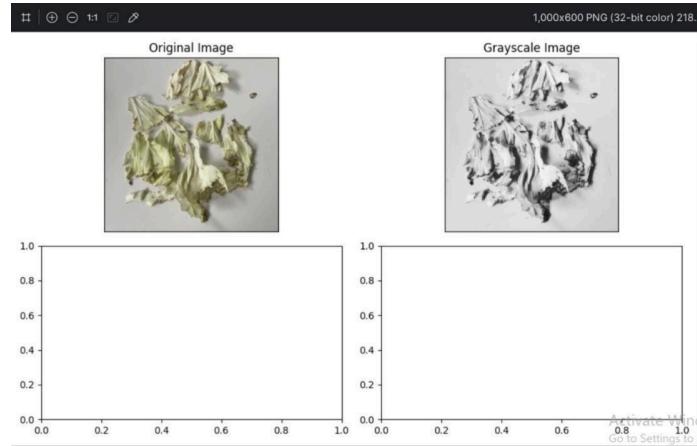
6. Konversi Citra RGB ke Citra Grayscale

Pada proses ini dilakukan konversi citra warna sayur kubis Red Green Blue (RGB) menjadi citra Grayscale dengan tujuan memudahkan pengolahan citra dalam proses ekstraksi data.

```

13
14 # 1. Preprocessing Data: Ekstraksi Fitur
15 def extract_glc_m_features(image):
16     """Ekstrak fitur GLCM dari gambar grayscale"""
17     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
18     glcm = graycomatrix(gray, distances=[1], angles=[0], levels=256, symmetric=True, normed=True) # Perbaikan nama fungsi
19     contrast = graycoprops(glcm, 'contrast')[0, 0]
20     dissimilarity = graycoprops(glcm, 'dissimilarity')[0, 0]
21     homogeneity = graycoprops(glcm, 'homogeneity')[0, 0]
22     energy = graycoprops(glcm, 'energy')[0, 0]
23     correlation = graycoprops(glcm, 'correlation')[0, 0]
24
25     return [contrast, dissimilarity, homogeneity, energy, correlation]

```



7. Ekstraksi Fitur RGB dan HSV

Pada proses fitur warna HSV, citra RGB di konversi menjadi citra HSV dengan tujuan mendapatkan nilai dari Hue, Saturation, dan Value.

```

25
26 def extract_color_features(image):
27     """Ekstrak fitur RGB dan HSV"""
28     rgb_mean = np.mean(image, axis=(0, 1)) # Rata-rata RGB
29     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
30     hsv_mean = np.mean(hsv, axis=(0, 1)) # Rata-rata HSV
31     return list(rgb_mean) + list(hsv_mean)
32
33 def extract_features(image):
34     """Gabungkan fitur GLCM dan warna"""
35     glcm_features = extract_glc_m_features(image)
36     color_features = extract_color_features(image)
37     return glcm_features + color_features
38

```



segar : original > biner > hsv



busuk : original > biner > hsv

8. Fitur GLCM

Pada proses ini dilakukan ekstraksi tekstur menggunakan fitur *Gray Level CoOccurrence Matrix* (GLCM) dengan tujuan untuk mendapatkan nilai dari empat fitur diantaranya terdiri dari contrast, correlation, energy,dan homogeneity. Dimana contrastuntuk mengukur jarak diagonal rata-rata matriks, correlation untuk mengukur ketergantungan linier tingkat keabuan antara piksel tetangga, energy karakteristik untuk melihat keseragaman tekstur, dan homogeneity untuk mengukur kedekatan distribusi elemen GLCM ke diagonal utama

```

14
15 # 1. Preprocessing Data: Ekstraksi Fitur
16 def extract_glcn_features(image):
17     """Ekstrak fitur GLCM dari gambar grayscale"""
18     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
19     glcm = graycomatrix(gray, distances=[1], angles=[0], levels=256, symmetric=True, normed=True)
20     contrast = graycoprops(glcm, 'contrast')[0, 0]
21     dissimilarity = graycoprops(glcm, 'dissimilarity')[0, 0]
22     homogeneity = graycoprops(glcm, 'homogeneity')[0, 0]
23     energy = graycoprops(glcm, 'energy')[0, 0]
24     correlation = graycoprops(glcm, 'correlation')[0, 0]
25     return [contrast, dissimilarity, homogeneity, energy, correlation]
26

```

9. Klasifikasi KNN

Pada proses klasifikasi ini, yang pertama dilakukan adalah menentukan nilai K terlebih dahulu. Pada penelitian ini, nilai K yang digunakan yaitu K=3. Setelah menentukan nilai K, kemudian menghitung jarak terdekat antara data latih dan data uji menggunakan persamaan *Euclidean Distance*. Jarak diartikan sebagai selisih atau perbedaan paling kecil dari tetangga terdekat, sehingga data uji dapat diklasifikasikan berdasarkan ciri dari tetangga terdekatnya.

```

70
71     # Train KNN
72     knn = KNeighborsClassifier(n_neighbors=3) # K=3
73     knn.fit(X_train, y_train)
74
75     # Load data testing
76     test_features, test_labels = load_images(test_folder)
77
78     if not test_features:
79         print("Tidak ada gambar pada folder test!")
80         exit()
81
82     X_test = np.array(test_features)
83     if X_test.ndim == 1:
84         X_test = X_test.reshape(1, -1) # Mengubah menjadi 2D jika perlu
85     X_test = scaler.transform(X_test)
86     y_test = np.array(test_labels)
87
88     # Cek jika data pengujian valid
89     if len(y_test) == 0:
90         print("Tidak ada label pada data pengujian!")
91         exit()
92
93     # Prediksi
94     y_pred = knn.predict(X_test)
95

```

Berdasarkan data yang telah diurutkan, kemudian ditentukan jarak terdekat sampai urutan K, yaitu 3 data dengan nilai terkecil. Kemudian dapat ditentukan kelas pada data uji kelas X kedalam mayoritas data latih, berdasarkan nilai K=3 yang digunakan. Sehingga dapat dikatakan bahwa kelas data uji termasuk kedalam mayoritas kelas segar.

Filename	True Label	Predicted Label
segara4.jpg	Segar	Busuk
Segara5.jpg	Segar	Segar
Segara7.jpg	Segar	Segar
Segara6.jpg	Segar	Segar
segara2.jpg	Segar	Segar
segara3.jpg	Segar	Segar
segara1.jpg	Segar	Segar
busuka1.jpg	Busuk	Busuk
busuka2.jpg	Busuk	Busuk
busuka3.jpg	Busuk	Busuk
Busuka6.jpg	Busuk	Busuk
Busuka4.jpg	Busuk	Busuk
Busuka5.jpg	Busuk	Busuk
Segara8.jpg	Segar	Segar

BAB IV **PENUTUP**

1. Kesimpulan

Berdasarkan hasil dari pengujian yang dilakukan dengan klasifikasi kesegaran sayur kubis berdasarkan warna dan tekstur menggunakan algoritma KNN dapat diambil beberapa kesimpulan :

1. Sistem Pengenalan sayur kol berdasarkan warna memiliki tingkat akurasi Sebesar 92.86% dalam mengenali 17 data gambar sayur kol
2. Penelitian menggunakan 31 total sampel data yang digunakan, 17 data untuk proses data training yang akan menjadi vektor acuan dalam proses pembelajaran kedekatan jarak tetangga untuk vektor gambar pengujian pada algoritma KNN dan 14 data untuk proses pengujian jenis sayur 1. Nilai k yang digunakan dalam algoritma KNN sebanyak 3 jarak ketetanggaan.
3. Dari hasil pengujian terhadap pengklasifikasian sayur kubis menggunakan metode K-Nearest Neighbor diperoleh akurasi sebesar 92,86% pada K=3. Hal tersebut menunjukkan bahwa nilai akurasi akan bergantung pada nilai variasi K yang digunakan.
4. Metode KNN dalam mengklasifikasi kesegaran sayur kubis berdasarkan ekstraksi fitur warna HSV dan fitur tekstur GLCM berhasil diimplementasikan pada klasifikasi kesegaran sayur kubis dengan kelas segar dan tidak segar

DAFTAR PUSTAKA

- [1] Wiastopo, Januwa Putra, “*Deteksi Kesegaran Ikan Kembung dengan Metode KNN Berdasarkan Fitur GLCM dan RGB-HSV*,” Jakarta, 2024.
- [2] M. A. A. Kurniawan, , E. Ermatita, and N. Falih, “*Pemanfaatan Pengolahan Citra dan Klasifikasi KNearrest Neighbor pada Citra Telur Ayam*,” *Inform. J. Ilmu Komput.*, vol. 16, no. 3, p. 164, 2020, doi: 10.52958/iftk.v16i3.2131.
- [3] A. Salsabila, R. Yunita, and C. Rozikin, “*Identifikasi Citra Jenis Bunga menggunakan Algoritma KNN dengan Ekstrasi Warna HSV dan Tekstur GLCM*,” *Technomedia J.*, vol. 6, no. 1, pp. 124–137, 2021, doi: 10.33050/tmj.v6i1.1667.
- [4] J. A. Widians, H. S. Pakpahan, E. Budiman, H. Haviluddin, and M. Soleha, “*Klasifikasi Jenis Bawang Menggunakan Metode K-Nearest Neighbor Berdasarkan Ekstraksi Fitur Bentuk dan Tekstur*,” *J. Rekayasa Teknol. Inf.*, vol. 3, no. 2, p. 139, 2019, doi: 10.30872/jurti.v3i2.3213.
- [5] Wiastopo, Januwa Putra, “*Deteksi Kesegaran Ikan Kembung dengan Metode KNN Berdasarkan Fitur GLCM dan RGB-HSV*,” Jakarta, 2024.