Pendeteksi Kesegaran Daging

Dosen pengampu : Irma Amelia Dewi, S.Kom., M.T.

Mata Kuliah : Pengolahan Citra Digital

Gamma Rizquha Wiradisastra (152021181)

Muhammad Rafi Yasir (152021191)

Muhammad Ghaza Azhar L (152021166)

EE

PRODI INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL BANDUNG

2023

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Warna adalah hasil persepsi cahaya pada spektrum yang terlihat oleh retina mata. Ruang warna merupakan model matematis abstrak yang memaparkan representasi suatu warna sebagai baris angka, umumnya terdiri dari tiga atau empat komponen warna. Contoh ruang warna yang umum digunakan termasuk RGB, CMY/CMYK, YIQ, YCbCr, HSI, HSL, HSV, dan CIELAB.

Model warna HSV (Hue, Saturation, Value) pertama kali diperkenalkan oleh A.R. Smith pada tahun 1978 dan didefinisikan sebagai warna dengan menggunakan konsep Hue, Saturation, dan Value. Keunggulan model HSV adalah kemampuannya untuk mempresentasikan warna dengan cara yang lebih sesuai dengan persepsi manusia terhadap warna. Sementara itu, model warna lain seperti RGB menggabungkan warna-warna primer untuk membentuk warna yang dihasilkan.

Daging sapi merupakan daging yang memiliki peranan penting dalam memenuhi kebutuhan gizi manusia, terutama protein. Kandungan gizi yang melimpah dalam daging sapi memberikan manfaat besar bagi pertumbuhan tubuh. Oleh karena itu, daging sapi menjadi pilihan favorit dalam berbagai hidangan, terutama di Indonesia.

1.2 Rumusan Masalah

1. Bagaimana cara mendeteksi kualitas daging sapi menggunakan citra?

2. Langkah apa saja yang harus dilakukan untuk mendeteksi kualitas daging sapi?

1.3 Tujuan

Tujuan dari laporan ini adalah membuat sebuah aplikasi yang dapat membedakan daging sapi yang memiliki kualitas bagus atau segar dengan daging sapi yang memiliki kualitas jelek atau busuk dengan menggunakan metode perbandingan HSV.

1.4 Ruang Lingkup

Pembagian ini terfokus pada :

1. Menggunakan Python sebagai bahasa pemrograman.
2. Menggunakan Metode HSV (warna merah) dalam project ini.
3. Mengambil range HSV untuk membedakan kualitas tiap daging.

BAB II
LANDASAN TEORI

2.1 Pengolahan Citra Digital

Pengolahan citra digital adalah bidang ilmu yang mempelajari pembentukan, pengolahan, dan analisis citra dengan tujuan menghasilkan informasi yang dapat dipahami oleh manusia. Dalam pengolahan citra digital, citra diproses dan diinterpretasikan secara digital menggunakan bantuan komputer.

2.2 Warna Pada Citra Digital

Warna merupakan spektrum tertentu yang terdapat dalam cahaya yang sempurna, yang dikenal sebagai cahaya putih. Identitas suatu warna ditentukan oleh panjang gelombang cahaya tersebut. Sebagai contoh, warna biru memiliki panjang gelombang sekitar 460 nanometer.

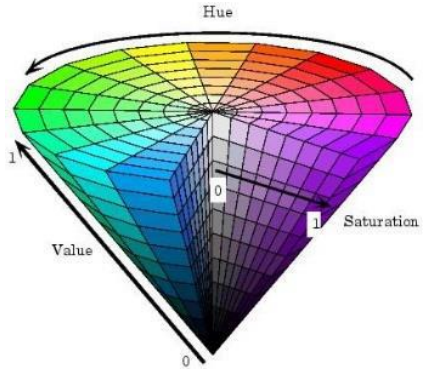
Dalam konteks peralatan optik, warna juga dapat merujuk pada interpretasi yang dibuat oleh otak berdasarkan campuran tiga warna primer cahaya, yaitu merah, hijau, dan biru, yang dikombinasikan dalam proporsi tertentu. Sebagai contoh, pencampuran 100% merah, 0% hijau, dan 100% biru akan menghasilkan interpretasi warna magenta.

2.3 RGB

Model warna RGB didasarkan pada pemahaman ilmu mata manusia tentang cahaya dan bagaimana informasinya diinterpretasikan oleh otak. Model warna RGB melibatkan proses pembentukan warna melalui kombinasi warna primer, yaitu R (red), G (green), dan B (blue). Warna primer ini dapat diklasifikasikan menjadi dua model, yaitu warna aditif dan warna subtraktif. Dalam konteks RGB, itu adalah model warna aditif yang menggambarkan warna dengan menggunakan "model cahaya".

2.4 HSV

Model HSV didefinisikan sebagai warna dengan menggunakan konsep Hue, Saturation, dan Value. Keunggulan model HSV adalah kemampuannya untuk mempresentasikan warna dengan cara yang lebih sesuai dengan persepsi manusia terhadap warna.



2.5 OpenCV

OpenCV adalah sebuah library(perpustakaan) yang digunakan untuk mengolah gambar dan video. Kata open pada OpenCV dimaksudkan open source gratis dan bisa di download dimana saja. Sementara CV pada kata OpenCV adalah kependekan dari Computer Vision, ini berarti menggunakan komputer yang diambil untuk mengolah image (citra/gambar) yang ditangkap oleh alat perekam seperti kamera atau webcam yang dikonversi dari analog ke digital lalu diolah dalam komputer.

2.6 Python

Python adalah bahasa pemrograman tingkat tinggi yang dikenal karena sintaksis yang sederhana, mudah dipelajari, dan mudah dibaca. Python mendukung pemrograman berorientasi objek, pemrograman fungsional, dan memiliki berbagai modul dan pustaka yang luas untuk berbagai keperluan pengembangan. Python digunakan secara luas dalam berbagai bidang seperti pengembangan perangkat lunak, analisis data, kecerdasan buatan, pengembangan web, dan banyak lagi.

BAB III

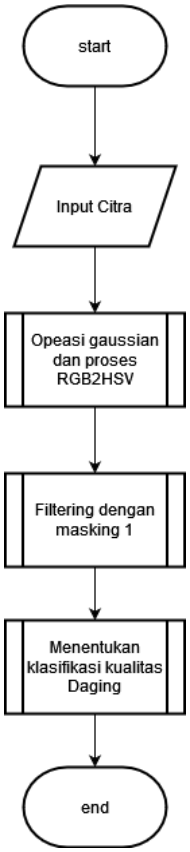
METODE PENELITIAN

3.1 Deskripsi Sistem

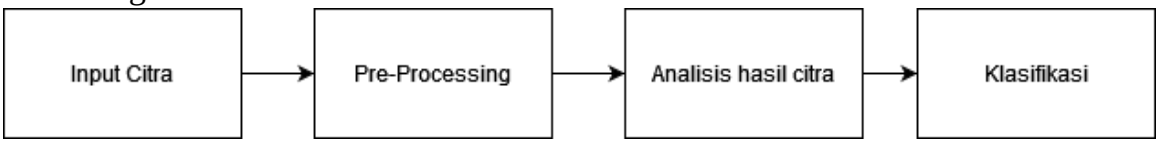
Sistem yang kami kembangkan yaitu sebuah aplikasi berbasis GUI yang memiliki sistem kerja menginputkan citra, lalu citra yang diinput akan diubah lagi menggunakan algoritma gaussian untuk mendapatkan citra yang lebih jelas untuk mendeteksi kualitas daging dengan metode HSV.

3.2 Alur Proses

Flowchart :



Blok Diagram :



3.3 Parameter Yang Akan Diteliti

Pada penelitian ini, objek yang akan diteliti adalah warna dan nilai pixel pada citra yang akan dibandingkan sesuai klasifikasi yang telah dibuat dan melihat keakuratan warnanya. Hasil dari penelitian menentukan kualitas daging mana yang masih segar dan mana yang sudah busuk.

- Masking Daging Normal

  lower : [0, 35, 45]

  Upper : [179, 229, 240]

- Masking Daging Berkualitas Baik

  lower : [0, 106, 154]

  Upper : [10, 255, 246]

- Masking Daging Berkualitas Buruk

  lower : [0, 0, 0]

  Upper : [10, 110, 134]

3.4 Operasi Pre-Processing

Data preprocessing adalah proses yang mengubah data mentah ke dalam bentuk yang lebih mudah dipahami. Proses ini penting dilakukan karena data mentah sering kali tidak memiliki format yang teratur dan untuk mempermudah analisis data.

3.4.1 Operasi Gaussian Blur

- Tujuan : Digunakan untuk proses penghalusan citra, pengaburan, menghiilangkan detail, menghilangkan noise.

- Alur Proses :

- Perhitungan Manual :
  - Citra awal : [50, 100, 150]

    [200, 250, 100]

    [150, 100, 50]
  - Kernel Gaussian 3x3 : [1, 2, 1]

    [2, 4, 2]

    [1, 2, 1]
  - Citra awal kali kernel gaussian : (1 * 50) + (2 * 100) + (1 * 150) + (2 * 200) + (4 * 250) + (2 * 100) + (1 * 150) + (2 * 100) + (1 * 50) = 1800
  - Ulangi langkah perkalian untuk piksel lainnya dalam gambar

- Hasil Output gaussian :

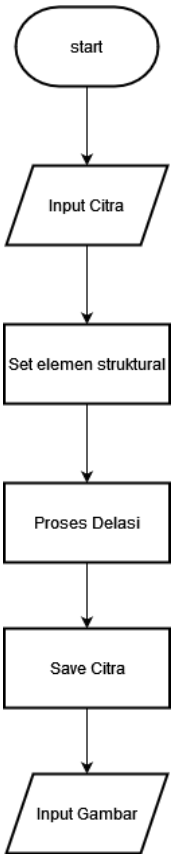
  [87, 138, 135]

  [163, 200, 188]

  [87, 113, 100]

3.4.2 Operasi Dilasi

- Tujuan : digunakan untuk memperluas atau memperbesar wilayah objek pada gambar. Fungsi operasi dilasi adalah menggabungkan piksel objek dengan piksel tetangga yang memiliki nilai piksel yang sama atau mirip.
- Alur Proses :



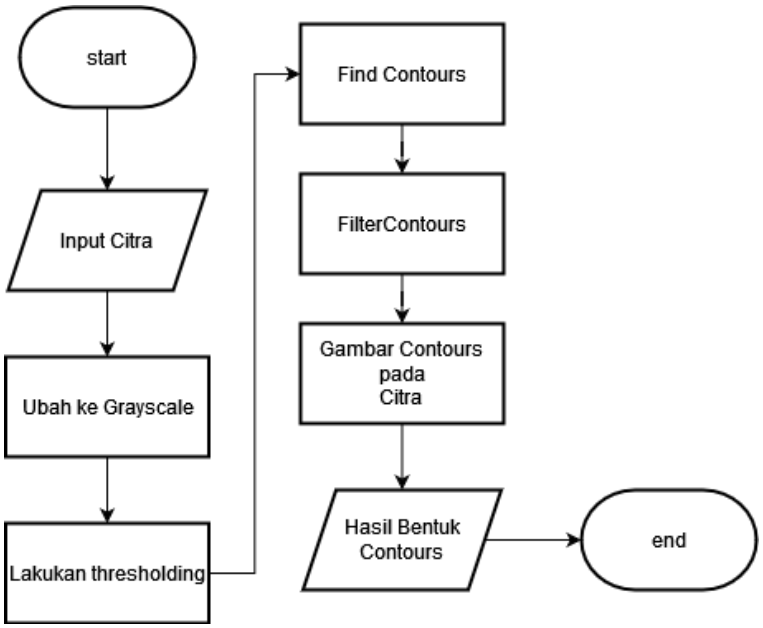
- Perhitungan Manual :
  - Citra awal : [0, 0, 0, 0, 0]
    [0, 0, 1, 0, 0]
    [0, 1, 1, 1, 0]
    [0, 0, 1, 0, 0]
    [0, 0, 0, 0, 0]

  - Element Struktural 3x3 : [1, 1, 1]
    [1, 1, 1]
    [1, 1, 1]

  Langkah-langkah perhitungan:
  - Pilih piksel pertama pada gambar awal (nilai 0).
  - Letakkan elemen struktural di atas piksel pertama pada posisi tertentu.
  - Periksa elemen struktural dengan piksel-piksel yang ada di bawahnya pada gambar awal. Jika ada piksel dengan nilai 1 yang berada di bawah elemen struktural, tandai piksel tersebut sebagai piksel terdilasi.
  - Pindah ke piksel berikutnya pada gambar awal.
  - Ulangi langkah 2 hingga 4 untuk setiap piksel pada gambar awal.
  - Hasil Output Dilasi:
    [0, 0, 1, 0, 0]
    [0, 1, 1, 1, 0]
    [0, 1, 1, 1, 0]
    [0, 1, 1, 1, 0]
    [0, 0, 1, 0, 0]

3.4.3 Operasi Contours

- Tujuan : Untuk mengidentifikasi dan mengekstraksi batas-batas atau bentuk objek yang ada dalam gambar. Operasi ini berfokus pada deteksi dan analisis fitur-fitur kontur seperti bentuk, ukuran, posisi, dan hubungan spasial antara objek dalam gambar.
- Alur Proses :



- Perhitungan Manual :
  - Terdapat gambar biner 8x8 dengan piksel objek (putih) dan piksel latar belakang (hitam). :

    [0 0 0 0 0 0 0 0]
    [0 1 1 0 0 0 0 0]
    [0 1 1 1 1 0 0 0]
    [0 0 0 1 0 0 0 0]
    [0 0 1 1 1 0 0 0]
    [0 0 0 0 0 0 0 0]
    [0 0 0 0 0 0 0 0]
    [0 0 0 0 0 0 0 0]

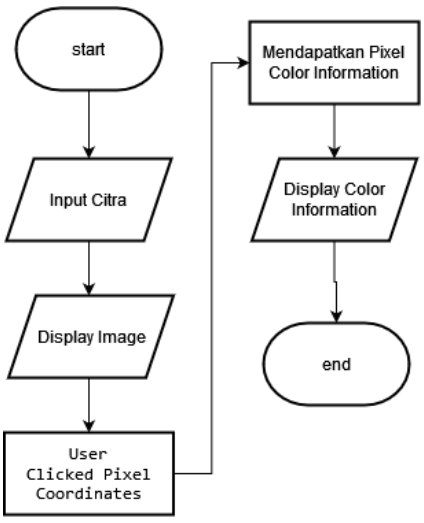
Langkah perhitungan manual contours:

- Tandai piksel (1,1) sebagai titik kontur karena setidaknya satu tetangga piksel (0,1) adalah piksel latar belakang.
- Tandai piksel (1,2), (2,1), (2,2), (2,3), (2,4), (3,3), (4,2), (4,3), (4,4), (5,3) sebagai titik kontur karena setidaknya satu tetangga piksel adalah piksel latar belakang.
- Tambahkan koordinat piksel-piksel yang ditandai ke dalam daftar titik kontur.
- Hasil Output Contours: [(1,1), (1,2), (2,1), (2,2), (2,3), (2,4), (3,3), (4,2), (4,3), (4,4), (5,3)]

3.5 Operasi Ekstraksi Fitur

3.5.1 Color Picker

- Tujuan : Alat atau fitur yang memungkinkan pengguna untuk memilih warna secara interaktif dari suatu gambar atau tampilan layar.
- Alur Proses :



- Perhitungan Manual : Pengguna mengklik pada titik tertentu pada gambar.
  - Koordinat piksel yang diklik oleh pengguna (misalnya, (x, y)) diambil.
  - Dalam gambar, nilai intensitas merah (R), hijau (G), dan biru (B) pada titik tersebut diambil.

- Normalisasi nilai R, G, dan B ke dalam rentang 0-1 jika perlu.
- Jika ada penggunaan sistem warna lain (seperti HSV atau Lab), konversi dari RGB ke sistem warna tersebut dilakukan.
- Nilai warna RGB (atau sistem warna lain jika digunakan) dari titik yang diklik dihasilkan.
- Nilai warna ini dapat ditampilkan kepada pengguna atau digunakan dalam proses selanjutnya, seperti mengubah warna objek lain atau menyimpan ke dalam palet warna.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Kode Program

```
import sys
import cv2
import numpy as np
from PyQt5 import QtCore, QtWidgets
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUi
from matplotlib import pyplot as plt


class showImage(QMainWindow):
  def __init__(self):
    super(showImage, self).__init__()
    loadUi('GUI.ui', self)
    self.Image = None


    self.actionOpen.triggered.connect(self.loadImage)
    self.btnCheck.clicked.connect(self.detect)
    self.btncolorpick.clicked.connect(self.colorPickPic)


  def loadImage(self):
    image, filter = QFileDialog.getOpenFileName(self, 'Open File', 'C:\\User\\', "Image Files(*.jpg)")
    self.Image = cv2.imread(image, 1)
    self.displayImage(1)


  def detect(self):
    # array untuk masking 1 berdasarkan nilai hsv
    lower = np.array([0, 35, 45])
    upper = np.array([179, 229, 240])
```

```
# array threshold nilai hsv untuk daging kualitas yang bagus
lower_normal = np.array([0, 106, 154])
upper_normal = np.array([10, 255, 246])

# array threshold nilai hsv untuk kualitas daging yang jelek
lower_jelek = np.array([0, 0, 0])
upper_jelek = np.array([10, 110, 134])

# Read Image
img = self.Image
self.imgawal = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Konvolusi dengan kernel gaussian
img = cv2.GaussianBlur(img, (5, 5), 0)
self.imggauss = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
result = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

# Filtering dengan masking 1
mask = cv2.inRange(result, lower, upper)
kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (9, 9))
result = cv2.dilate(mask, kernel)

# membuat contours
contours, hierarchy = cv2.findContours(result.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
result = cv2.cvtColor(result, cv2.COLOR_GRAY2BGR)
self.imgmask_background = result

# menentukan kualitas masing-masing daging yang terdeteksi
if len(contours) != 0:
    for (i, c) in enumerate(contours):
        area = cv2.contourArea(c)
        print("area : ", area)
        if area > 1000:
```

```
cv2.drawContours(img, c, -1, (255, 255, 0), 8)  # menggambar contours hasil mask
x, y, w, h = cv2.boundingRect(c)
cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 5)
croppedk = img[y:y + h, x:x + w]

hsv = cv2.cvtColor(croppedk, cv2.COLOR_BGR2HSV)

# deteksi kualitas daging
# warna normal
normalmask = cv2.inRange(hsv, lower_normal, upper_normal)

cv2.imshow('mask warna normal', normalmask)
cnt_n = 0
for n in normalmask:
    cnt_n = cnt_n + list(n).count(255)
print("nilai mask normal : ", cnt_n)

# warna jelek
badmask = cv2.inRange(hsv, lower_jelek, upper_jelek)

cv2.imshow('mask warna busuk', badmask)
cnt_j = 0
for j in badmask:
    cnt_j = cnt_j + list(j).count(255)
print("nilai mask jelek : ", cnt_j)

# Penamaan klasifikasi
# segar
if cnt_n > cnt_j:
    cv2.putText(img, 'Daging masih segar', (x+10, y+10), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 0, 0), 2)
    print("Terdeteksi : daging bagus")
# busuk
if cnt_j > cnt_n:
    cv2.putText(img, 'Daging sudah busuk', (x+10, y+10), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
```

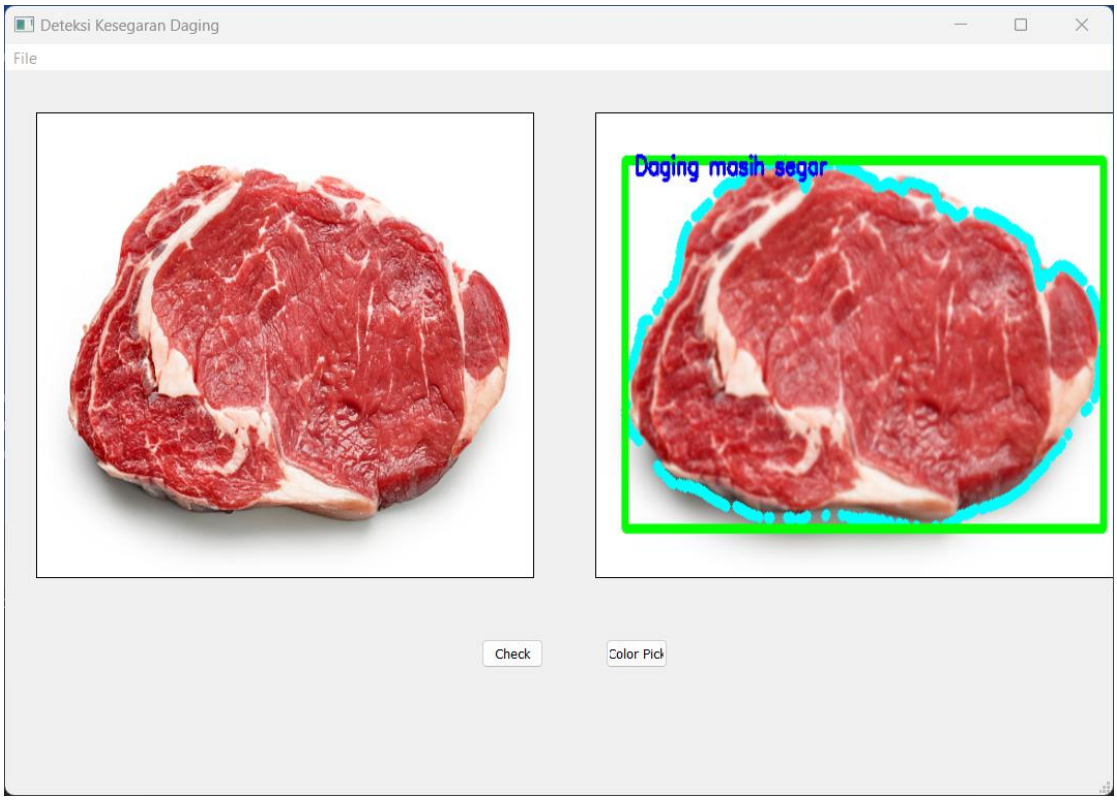
```
                print("Terdeteksi : daging busuk")
            print("------------------------------")
        else:
            print("Tidak ada objek yang terdeteksi")

        # Tampilkan hasil
        self.Image = img
        self.imgakhir = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.displayImage(2)
        img2 = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
        # Tampilkan histogram dari hasil
        h, s, v = img2[:, :, 0], img2[:, :, 1], img2[:, :, 2]
        hist_h = cv2.calcHist([h], [0], None, [256], [0, 256])
        hist_s = cv2.calcHist([s], [0], None, [256], [0, 256])
        hist_v = cv2.calcHist([v], [0], None, [256], [0, 256])
        plt.plot(hist_h, color='r', label="h")
        plt.plot(hist_s, color='g', label="s")
        plt.plot(hist_v, color='b', label="v")
        plt.legend()
        plt.show()
        cv2.waitKey(0)
        cv2.destroyAllWindows()

    def displayImage(self, windows=1):
        qformat = QImage.Format_Indexed8

        if len(self.Image.shape) == 3:
            if (self.Image.shape[2]) == 4:
                qformat = QImage.Format_RGBA8888

            else:
                qformat = QImage.Format_RGB888
        img = QImage(self.Image, self.Image.shape[1], self.Image.shape[0],
                     self.Image.strides[0], qformat)
```

```
        img = img.rgbSwapped()
        if windows == 1:
            self.labelCitra.setPixmap(QPixmap.fromImage(img))

            self.labelCitra.setAlignment(QtCore.Qt.AlignHCenter | QtCore.Qt.AlignVCenter)

            self.labelCitra.setScaledContents(True)
        if windows == 2:
            self.labelCitra_2.setPixmap(QPixmap.fromImage(img))

            self.labelCitra_2.setAlignment(QtCore.Qt.AlignHCenter | QtCore.Qt.AlignVCenter)

            self.labelCitra_2.setScaledContents(True)

    def colorPickPic(self):
        def nothing(x):
            pass

        img = self.Image
        cv2.namedWindow("Trackbar")
        cv2.createTrackbar("Lower H", "Trackbar", 0, 179, nothing)
        cv2.createTrackbar("Lower S", "Trackbar", 0, 255, nothing)
        cv2.createTrackbar("Lower V", "Trackbar", 0, 255, nothing)
        cv2.createTrackbar("Upper H", "Trackbar", 179, 179, nothing)
        cv2.createTrackbar("Upper S", "Trackbar", 255, 255, nothing)
        cv2.createTrackbar("Upper V", "Trackbar", 255, 255, nothing)

        while True:
            hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
            LH = cv2.getTrackbarPos("Lower H", "Trackbar")
            LS = cv2.getTrackbarPos("Lower S", "Trackbar")
            LV = cv2.getTrackbarPos("Lower V", "Trackbar")
            UH = cv2.getTrackbarPos("Upper H", "Trackbar")
```

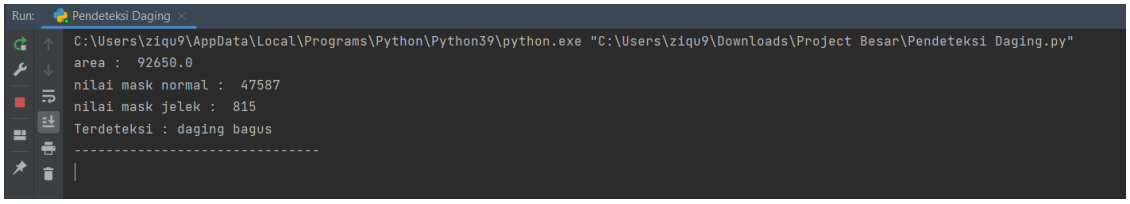
```
        US = cv2.getTrackbarPos("Upper S", "Trackbar")
        UV = cv2.getTrackbarPos("Upper V", "Trackbar")
        lower_color = np.array([LH, LS, LV])
        upper_color = np.array([UH, US, UV])
        mask = cv2.inRange(hsv, lower_color, upper_color)
        result = cv2.bitwise_and(img, img, mask=mask)
        cv2.imshow("Frame", img)
        cv2.imshow("Mask", mask)
        cv2.imshow("Hasil", result)
        key = cv2.waitKey(1)
        if key == 27:  # tekan esp untuk stop
            break
    cv2.destroyAllWindows()


app = QtWidgets.QApplication(sys.argv)
window = showImage()
window.setWindowTitle('Deteksi Kesegaran Daging')
window.show()
sys.exit(app.exec_())
```

4.2 Hasil Pengujian

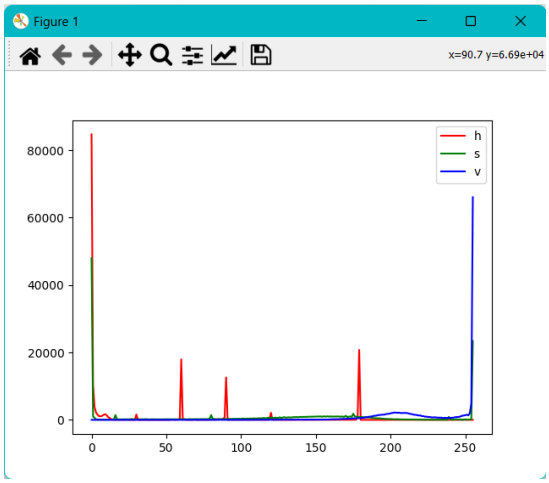
4.2.1 Deteksi Daging Segar



gambar hasil pengujian terhadap daging segar

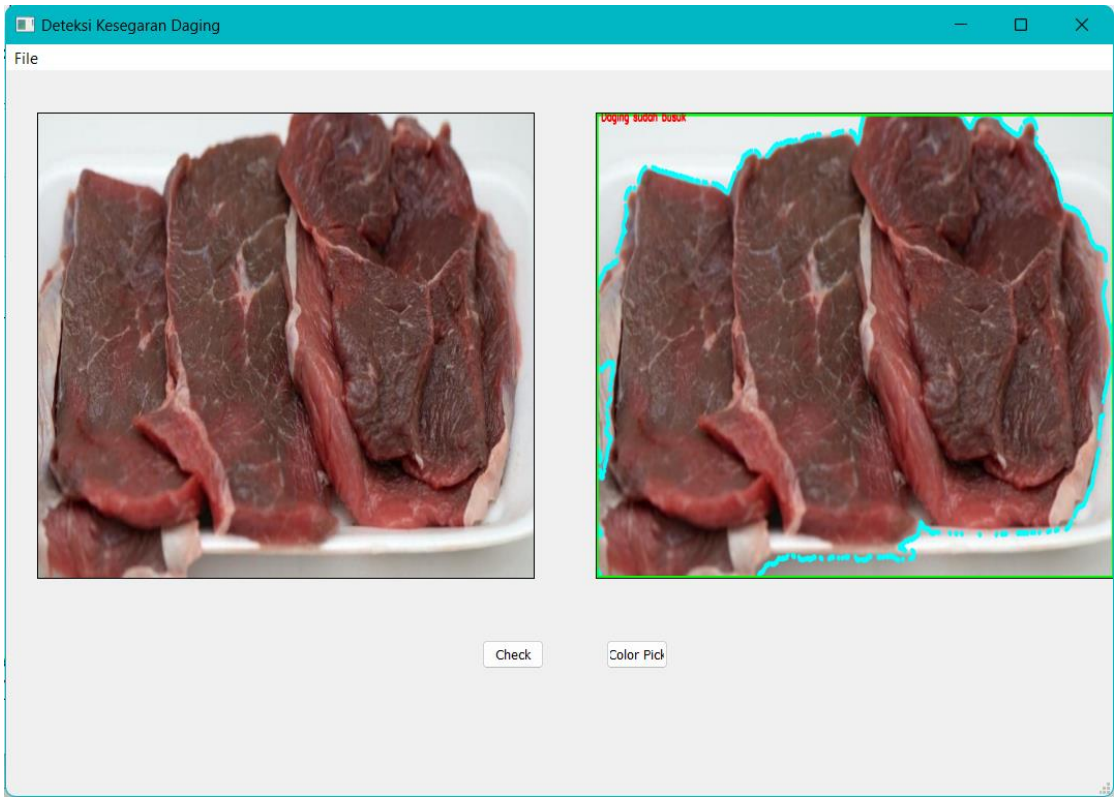


gambar hasil output pada console dari pengujian terhadap daging segar

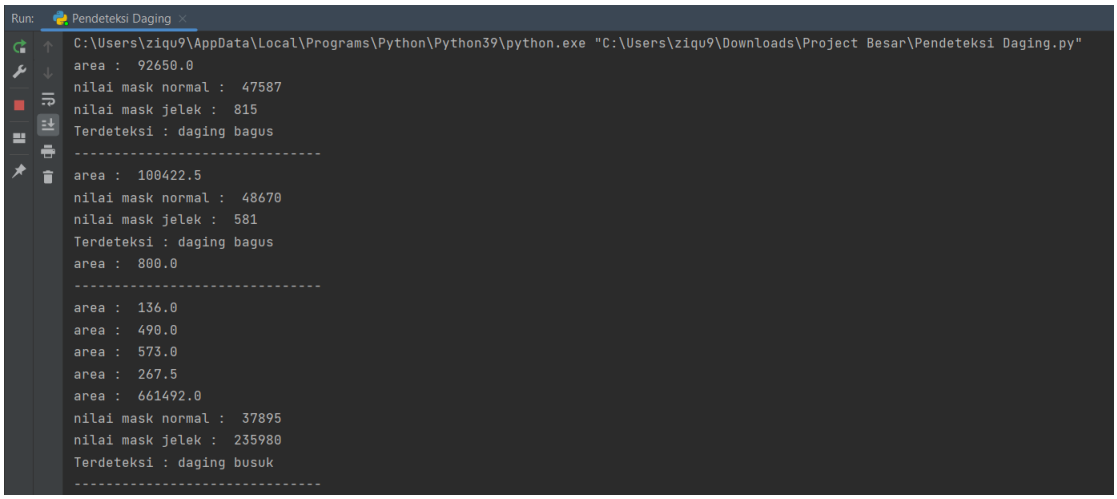


gambar histogram dari citra daging segar

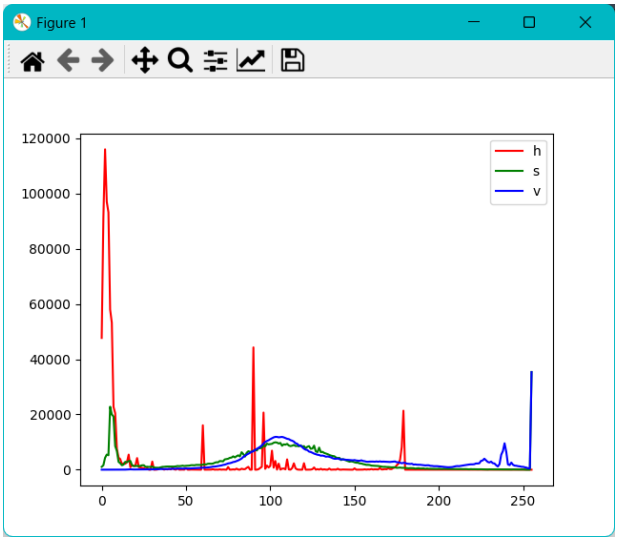
4.2.2 Deteksi Daging Busuk



gambar hasil pengujian terhadap daging busuk



gambar hasil output pada console dari pengujian terhadap daging segar



gambar histogram dari citra daging busuk

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian, didapatkan kesimpulan sebagai berikut:

1. Deteksi kualitas daging sapi dapat dilakukan dengan cara membandingkan nilai HSV yang diambil dari citra tersebut dengan *threshold* HSV yang sudah ditentukan.

2. Langkah yang dilakukan untuk mendeteksi kualitas daging sapi adalah dengan memasukkan citra input ke dalam program, lalu dilakukan penghalusan citra menggunakan operator gaussian, kemudian ubah format warna citra dari RGB ke HSV, kemudian program akan membandingkan nilai HSV citra dengan *threshold* HSV yang sudah ditentukan untuk diklasifikasikan, dan kemudian citra hasil ditampilkan dengan label yang menyatakan bahwa daging tersebut segar atau busuk.

3. Berdasarkan pengujian menggunakan dataset sebanyak 10 citra, didapatkan 9 dari 10 citra yang dideteksi sesuai dengan klasifikasi, sehingga dapat disimpulkan aplikasi ini memiliki tingkat akurasi 90%.

DAFTAR PUSTAKA

Rismawan Fajril Falah1, Oky Dwi Nurhayati2, Kurniawan Teguh Martono (2016, April). *Aplikasi Pendeteksi Kualitas Daging Menggunakan Segmentasi Region of Interest Berbasis Mobile* .From https://jtsiskom.undip.ac.id/article/view/12720

Courtney Taylor (2019, 28 April). What Is a Histogram?. From https://www-thoughtco-com.translate .goog/what-is-a-histogram 3126359?_x_tr_sl=en&_x_tr_ tl=id&_x_ tr_hl=id&_x_tr_pto=tc

Jati Sasongko (2011). *Deteksi dan Klasifikasi Citra Berdasarkan Warna Kulit Menggunakan HSV* .From https://www.neliti.com/id/publications/243286/deteksi-dan-klasifikasi-citra-berdasarkan -warna-kulit-menggunakan-hsv

Bamai Uma (2023, 6 Febuari). *Pengertian Pengolahan Citra Digital*. From https://bamai.uma.ac.id/ 2023/02/06/mari-simak-pengertian-pengolahan-citra-digital/

Jeffrey Ansen (2020). *Apa itu Model Warna RGB? Nih, Penjelasan Lengkapnya!*. From https://solusiprinting.com/apa-itu-model-warna-rgb-nih-penjelasan-lengkapnya/

Srimulia (2022, 31 Agustus). *Mengenal OpenCV Dalam Python: Pengertian , Sejarah, Dukungan pada OS, Fitur-fitur.* From https://idmetafora.com/news/read/1177/Mengenal-OpenCV-Dalam -Python-Pengertian-Sejarah-Dukungan-pada-OS-Fitur-fitur.html