

Penerapan Image Processing dalam Sorting Ukuran Telur

Haning Nanda Hapsari / 13519042
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): haningnandahapsari@gmail.com

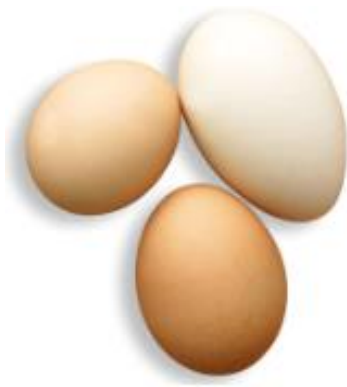
Abstract—Perkembangan teknologi, salah satunya bidang pengolahan citra sangatlah pesat. Banyak metode-metode pengolahan citra yang dapat dimanfaatkan manusia. Dengan menggunakan pengolahan citra, beberapa permasalahan sehari-hari dapat terbantu. Salah satu hal yang dapat terselesaikan dengan pengolahan citra yaitu pengukuran. Kini, selain dilakukan secara manual menggunakan alat ukur, pengukuran dapat dilakukan dengan estimasi menggunakan metode pengolahan citra. Meskipun demikian, hasil pengukuran menggunakan metode pengolahan citra masih kurang akurat sehingga masih perlu dikembangkan lebih lanjut.

Kata kunci—*pengolahan citra; pengukuran;*

I. PENDAHULUAN

Telur merupakan salah satu sumber protein yang sangat digemari oleh masyarakat. Harganya yang murah, tetapi memiliki kandungan gizi yang tinggi merupakan salah satu penyebabnya. Telur yang dijual dipasaran telah diklasifikasi berdasarkan beberapa kriteria, salah satunya yaitu ukuran telur.

Dalam melakukan klasifikasi telur, penjual menggunakan bobot telur sebagai dasar klasifikasi. Bobot telur berbanding lurus dengan ukuran telur. Telur yang berat cenderung memiliki ukuran yang besar dan telur yang ringan memiliki ukuran yang kecil.



Gambar 1.1 Ukuran Telur

Pada masa sekarang sudah terdapat berbagai macam teknologi yang mampu membantu manusia untuk melakukan segala sesuatu secara lebih praktis dan efisien. Perkembangan

teknologi ini memaksa manusia untuk melupakan metode lama yang menguras cukup banyak waktu. Salah satu metode yang dapat digunakan yaitu menggunakan citra sebagai objek.

Citra merupakan salah satu objek yang dapat memberikan banyak informasi secara tersirat. Terlebih lagi, teknologi yang semakin maju menyebabkan informasi dari citra dapat diolah menjadi suatu informasi yang sangat berguna untuk menyelesaikan berbagai permasalahan dalam kehidupan sehari-hari. Pengolahan gambar ini biasa disebut dengan pengolahan citra atau *image processing*.

Pada *image processing* dapat dilakukan berbagai metode seperti *image enhancement*, *image restoration*, *image compression*, dan *image representation and modelling*. Tidak hanya melakukan manipulasi citra saja, metode-metode ini dapat menghasilkan informasi-informasi yang dapat memecahkan persoalan-persoalan sehari-hari, salah satunya persoalan klasifikasi telur seperti yang digambarkan di atas.

Makalah ini berfokus pada metode pada *image processing* yang akan digunakan untuk melakukan pengukuran telur berdasarkan gambar. Sistem akan mendapatkan input berupa beberapa objek telur yang mana akan diproses sehingga menghasilkan ukuran masing-masing telur.

II. LANDASAN TEORI

A. Citra

Citra atau gambar merupakan sinyal dwimatra (dua dimensi) yang bersifat *continue* yang dapat diamati oleh sistem visual manusia. Dapat dikatakan bahwa citra merupakan fungsi dwimatra yang menyatakan intensitas cahaya pada bidang dwimatra. Citra mengandung banyak informasi meskipun bersifat tersirat atau tidak eksplisit.

Terdapat dua macam citra yaitu citra diam dan citra bergerak. Citra diam dalam sebuah citra tunggal, sedangkan citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberikan kesan sebagai gambar yang bergerak.



Gambar 2.1 Citra Diam

Foto merupakan contoh citra diam dan video merupakan contoh citra bergerak.



Gambar 2.2 Kumpulan Frame Citra Bergerak

Dalam citra digital, berdasarkan jenis warnanya, citra dibedakan menjadi tiga jenis yaitu citra RGB, citra greyscale, dan citra biner. Citra RGB (*Red, Green, Blue*) merupakan citra yang nilai intensitas pixelnya tersusun oleh tiga kanal warna yaitu merah, hijau dan biru. Citra greyscale merupakan citra yang nilai intensitas pixelnya berdasarkan derajat keabuan, sedangkan citra biner yaitu citra yang hanya memiliki dua nilai intensitas yaitu hitam dan putih.

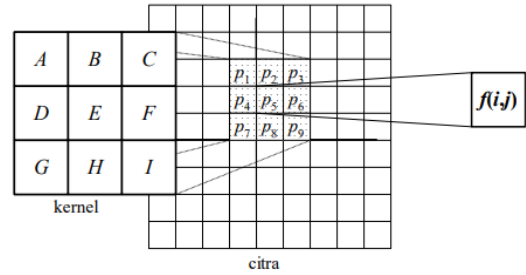


Gambar 2.3 Lena dalam Greyscale

B. Konvolusi Matriks

Teknik konvolusi merupakan salah satu teknik yang dapat digunakan dalam perbaikan kualitas citra. Konvolusi merupakan

penjumlahan dari perkalian setiap kernel dengan setiap titik dari citra masukan.



Gambar 2.4 Ilustrasi Konvolusi

Pada gambar 2.3 mengenai ilustrasi konvolusi, untuk mendapatkan titik $f(i,j)$ dilakukan penjumlahan dari perkalian titik kernel dengan titik matriks citra. Penjumlahan tersebut adalah sebagai berikut.

$$f(i, j) = (Ap_1 + Bp_2 + Cp_3 + Dp_4 + Ep_5 + Fp_6 + Gp_7 + Hp_8 + Ip_9)$$

C. Gaussian Filter Method

Gaussian Filter Method merupakan metode yang bertujuan untuk mengurangi *noise* dengan cara mengalikan matriks kernel dengan matriks citra asli. Perkalian matriks ini menggunakan metode konvolusi matriks. Ukuran penapis gaussian bergantung pada nilai σ . Berikut merupakan formula pembuatan matriks penapis gauss.

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp \frac{x^2+y^2}{2\sigma^2}$$

D. Edge Detection

Tepi merupakan keadaan yang terjadi apabila terdapat perubahan intensitas nilai keabuan yang sangat signifikan dalam jarak yang singkat. Adapun tujuan dalam pendeteksian tepi ini adalah sebagai berikut.

1. Meningkatkan penampakan garis batas suatu objek dalam citra
2. Mengekstraksi representasi gambar garis-garis dalam citra
3. Mengenali objek dalam citra (*image recognition*)

Terdapat beberapa operator yang digunakan untuk mendeteksi tepi. Salah satu operator tersebut adalah operator Canny. Operator ini sangatlah terkenal karena dapat menghasilkan tepi dengan ketebalan 1 pixel. Operator Canny sangat sering digunakan untuk melakukan pendeteksian tepi karena dapat mendeteksi semua tepi-tepi pada citra dengan baik.

E. Image Transformation

Image Transformation atau yang biasa disebut transformasi citra merupakan pengubahan suatu citra dari suatu kawasan ke kawasan yang lain. Transformasi citra dapat dilakukan dengan beberapa operator salah satunya yaitu operator geometri. Pada transformasi citra menggunakan operator geometri, koordinat

pixel berubah, namun intensitasnya tetap. Beberapa contoh operasi geometri, antara lain:

1. Translasi

Pada translasi, citra digeser berdasarkan parameter dalam arah x dan arah y. Berikut merupakan persamaan dari translasi.

$$B(x', y') = A(x + m, y + n)$$

Pada persamaan di atas, m merupakan pergeseran dalam arah x, dan n adalah pergeseran dalam arah y.

2. Rotasi

Pada rotasi, citra diputar berdasarkan sudut yang diinginkan. Persamaan rotasi adalah sebagai berikut.

$$A(x, y) = B(x \cos(\theta) - y \sin(\theta), x \sin(\theta) + y \cos(\theta))$$

3. Dilatasi

Dilatasi merupakan perubahan ukuran citra menggunakan skala tertentu. Rumus penyekalaan citra dinyatakan sebagai:

$$B(x', y') = B(S_x \cdot x, S_y \cdot y) = A(x, y)$$

dengan S_x dan S_y sebagai faktor penyekalaan.

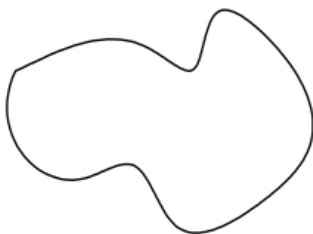
4. Refleksi

Refleksi merupakan salah satu operasi yang sangat familiar. Refleksi atau yang biasa disebut pencerminan terdiri menjadi 2 macam yaitu horizontal dan vertikal. Pencerminan horizontal merupakan pencerminan citra terhadap sumbu x dan pencerminan horizontal merupakan pencerminan citra terhadap sumbu y.

F. Kontur

Kontur merupakan rangkaian pixel-pixel tepi yang membentuk batas daerah (*region boundary*). Batas daerah berguna untuk mendeskripsikan bentuk objek dalam tahap analisis citra, seperti mengenali objek. Kontur ditimbulkan oleh perubahan intensitas pada pixel-pixel yang bertetangga.

Kontur dapat terbuka maupun tertutup. Kontur tertutup berkorespondensi dengan batas yang mengelilingi suatu daerah.



Gambar 2.5 Kontur Tertutup

Sedangkan kontur terbuka dapat berupa fragmen garis atau bagian dari batas daerah yang tidak membentuk sirkuit.



Gambar 2.6 Kontur Terbuka

Representasi kontur dapat berupa senarai tepi (*edge list*) atau kurva. Senarai tepi merupakan himpunan terurut pixel-pixel tepi, sedangkan jika dalam bentuk kurva representasi tersebut digambarkan dalam bentuk persamaan.

G. Euclidean Distance

Euclidean distance merupakan suatu fungsi heuristic untuk menghitung jarak antara dua titik menggunakan akar dari penjumlahan selisih koordinat kedua titik.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

H. OpenCV

OpenCV (*Open Source Computer Vision Library*) merupakan suatu library yang dikembangkan oleh inter untuk menyederhanakan pemrograman terkait citra digital. OpenCV ditujukan dalam bahasa pemrograman C/C++ yang kini telah dikembangkan dalam Bahasa python, java, dan matlab.

OpenCV memiliki banyak fitur yang dapat dimanfaatkan. Berikut merupakan fitur utama dari OpenCV.

1. Image dan video I/O

Pada fitur ini, pengguna dapat membaca data gambar maupun video. Selain itu, pengguna juga dapat membuat file gambar maupun video.

2. General computer vision

Pada fitur ini dapat dilakukan berbagai standar algoritma *computer vision* seperti *edge detection*, proyeksi elips, *image pyramid*, transformasi, dan lain-lain.

3. Modul computer vision

OpenCV juga menyediakan berbagai modul seperti paket modul *machine learning*.

4. Sampling gambar dan transformasi

Pada OpenCV terdapat *interface* untuk substraksi subregion gambar, random sampling, rotation, dan lain-lain yang bertujuan untuk menciptakan dan menganalisa gambar biner.

III. IMPLEMENTASI DAN ANALISIS

Penyelesaian masalah pengukuran telur pada makalah ini dilakukan dengan membangun sebuah program menggunakan Bahasa python. Program akan menerima input berupa citra

berisi beberapa telur yang kemudian akan diolah dan ditampilkan hasil berupa pengukuran telur. Tahapan pembuatan program ini akan dijelaskan lebih rinci pada subbab berikut.

A. Persiapan Citra yang Akan Dideteksi

Citra yang akan dideteksi akan melalui beberapa tahapan yaitu perubahan citra menjadi greyscale, melakukan *blurring* menggunakan gaussian filter, dan menghilangkan background dari citra. Adapun source code dari program adalah sebagai berikut.

```
# Mengubah citra berwarna ke Grayscale
gray = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)
# Melakukan blurring menggunakan filter
gauss
gray = cv2.GaussianBlur(gray, (3, 3), 0)

# Menghilangkan background
result = image.copy()
result[result<210] = (255, 255, 255)
```

Pada kode di atas, pertama citra diubah menjadi greyscale yang kemudian dilakukan *blurring* menggunakan penapis gauss dengan ukuran 3x3 dan dilanjutkan dengan menghapus background dengan mengubah keabuan warna dengan *threshold* 210, jika dibawah 210 menjadi berwarna putih. *Threshold* disesuaikan dengan *background* citra.



Gambar 3.1 Citra Persiapan

B. Pendeteksian Tepi

Setelah citra yang ingin dideteksi difilter, citra akan dilakukan pendeteksian tepi menggunakan operator Canny. Berikut merupakan kode program pendeteksian tepi.

```
# Edge Detection Menggunakan Operator
Canny
edged = cv2.Canny(result, 50, 100)
# Dilatation Image
edged = cv2.dilate(edged, None,
iterations=1)
# Pengikisan pixel tepi untuk mengetahui
batas
edged = cv2.erode(edged, None,
iterations=1)
```

Pada program di atas, setelah melakukan pendeteksian tepi dilakukan dilatasi dan pengikisan citra untuk mengetahui batas-batas citra.



Gambar 3.2 Tepi Citra

C. Deteksi Kontur

Setelah pendeteksian tepi kemudian dilakukan pencarian kontur dari citra. Setelah itu, dilakukan penentuan besar pixel per *metric* yang digunakan untuk menentukan ukuran asli dari gambar. Penentuan ukuran matriks disesuaikan dengan jarak kamera dengan objek. Berikut merupakan kode yang digunakan untuk mencari kontur dan menentukan skalanya.

```
# Mencari kontur
cnts = cv2.findContours(edged.copy(),
cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)

(cnts, _) = contours.sort_contours(cnts)
pixelsPerMetric = 25
```

D. Perhitungan Jarak

Titik-titik yang dihasilkan dalam pencarian kontur ini kemudian akan dihitung titik tengah dan jarak antara titik tengah tersebut akan dihitung menggunakan dengan menggunakan metode *Euclidean distance*.

```
# Fungsi untuk mencari titik tengah
def midpoint(a, b):
    return ((a[0] + b[0]) * 0.5,
(a[1] + b[1]) * 0.5)

# Mencari midpoint titik kiri-atas
dengan kanan-atas, serta kiri-bawah
dengan kanan-bawah
(tl, tr, br, bl) = box
(tltrX, tltrY) = midpoint(tl, tr)
(blbrX, blbrY) = midpoint(bl, br)

# Mencari midpoint titik kiri-atas
dengan kiri-bawah, serta kanan-atas
dengan kanan-bawah
(tlblX, tlblY) = midpoint(tl, bl)
(trbrX, trbrY) = midpoint(tr, br)

# Mencari jarak menggunakan euclidean
distance
```

```

dA = dist.euclidean((tltrX, tltrY),
(blbrX, blbrY))
dB = dist.euclidean((tlblX, tlblY),
(trbrX, trbrY))

```

```

# Menghitung ukuran objek
dimA = dA / pixelsPerMetric
dimB = dB / pixelsPerMetric

```

E. Klasifikasi Telur

Setelah dilakukan pengukuran telur, akan dilakukan klasifikasi telur. Contoh klasifikasi yang dilakukan yaitu dengan mengalikan ukuran telur dan memasukkan hasil perkalian tersebut kedalam kelompok tertentu. Berikut merupakan contoh kode pengklasifikasian telur.

```

# Melakukan klasifikasi telur
kelas = ''
if (dimA*dimB>6):
    kelas = 'Besar'
elif (4<dimA*dimB<6):
    kelas = 'Sedang'
else:
    kelas = 'Kecil'

```

F. Penampilan Hasil

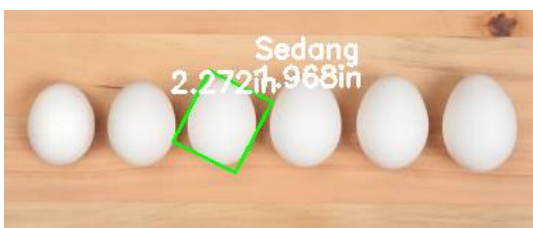
Hasil dari perhitungan diatas adalah sebagai berikut.



Gambar 3.3 Hasil Telur 1



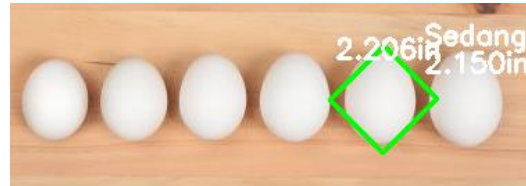
Gambar 3.4 Hasil Telur 2



Gambar 3.5 Hasil Telur 3



Gambar 3.6 Hasil Telur 4



Gambar 3.7 Hasil Telur 5

Berdasarkan hasil di atas, dapat dilihat jika pengukuran telur berhasil dilakukan. Namun kontur yang dibentuk kurang presisi.

IV. KESIMPULAN

Teknik pengolahan citra dapat dimanfaatkan untuk mengatasi banyak hal dalam kehidupan sehari-hari. Salah satu teknik yang bermanfaat adalah *edge detection* atau pendeteksi tepi. Dengan pendeteksi tepi ini dapat dilakukan berbagai macam manfaat lainnya seperti pengukuran objek seperti yang dijelaskan pada makalah ini untuk objek khusus berupa telur, mendeteksi keretakan pada ember, dan lain sebagainya.

Pada hasil percobaan implementasi pendeteksi ukuran telur, implementasi telah berhasil dilakukan. Namun, pada implementasi program dilakukan input konstanta pixel secara manual sehingga hasil pengukuran kurang akurat.

LINK VIDEO YOUTUBE

<https://youtu.be/DJli2DD0a1U>

REFERENSI

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/01-Pengantar-Pengolahan-Citra-Bag1-2022.pdf>. Diakses pada 16 Desember 2022.
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/02-Pengantar-Pengolahan-Citra-Bag2-2022.pdf>. Diakses pada 16 Desember 2022.
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2020-2021/07-Konvolusi.pdf>. Diakses pada 16 Desember 2022.
- [4] https://lmsspada.kemdikbud.go.id/pluginfile.php/61155/mod_resource/content/1/PCD-3.pdf. Diakses pada 16 Desember 2022.
- [5] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/10-Image-Enhancement-Bagian3-2022.pdf>. Diakses pada 16 Desember 2022.
- [6] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/18-Pendeteksian-Tepi-Bagian1-2022.pdf>. Diakses pada 16 Desember 2022.
- [7] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/19-Pendeteksian-Tepi-Bagian2-2022.pdf>. Diakses pada 16 Desember 2022.

- [8] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2019-2020/14-Kontur.pdf>. Diakses pada 16 Desember 2022.
- [9] https://elibrary.unikom.ac.id/id/eprint/2678/8/12.%2010113247_HENNI%20YULIANTI%20S_BAB%202.pdf. Diakses pada 16 Desember 2022.

Bandung, 19 Desember 2022



Haning Nanda Hapsari / 132519042

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.