

## Operasi Morfologi Dan Kode Rantai Untuk Menghitung Luas Area Basah Kertas Saring

Nafiatun Sholihah\*<sup>1</sup>, Agus Harjoko<sup>2</sup>

<sup>1</sup>Program Studi Ilmu Komputer, FMIPA UGM, Yogyakarta, Indonesia

<sup>2</sup>Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: \*<sup>1</sup>nafiatun@amikom.ac.id, <sup>2</sup>aharjoko@ugm.ac.id

### Abstrak

Perhitungan area basah yang dilakukan dengan bantuan kertas millimeter blok mempunyai kelemahan pada penyalinan tepian yang kurang tepat dan waktu perhitungan yang cukup lama. Masalah lain adalah konsistensi dan ketelitian yang dihasilkan tergantung pada faktor subyektif manusia dan kelelahan seseorang. Agar proses lebih cepat dan lebih konsisten, proses perhitungan menggunakan pengolahan citra sangat diperlukan. Prapemrosesan citra antara lain cropping, grayscalling, lowpass filter averaging, konversi ke citra biner berdasarkan otsu thresholding, dan komplemenkan citra agar piksel objek bernilai 1. Segmentasi dengan operasi morfologi, antara lain operasi opening untuk menghilangkan objek-objek kecil di sekeliling objek, operasi Filling Holes untuk mengisi lubang pada objek, operasi opening lagi untuk menghilangkan objek selain area basah. Proses penghitungan luas area basah menggunakan chain code. Berdasarkan hasil pengujian terhadap 81 citra, penggunaan operasi morfologi mampu menghasilkan segmentasi area basah yang mendekati area basah aslinya sebesar 96%. Nilai skala berpengaruh terhadap akurasi dan skala terbaik diperoleh dari penggunaan mistar. Penggunaan chain code mampu menghitung luas area basah pada kertas saring dengan rata-rata akurasi 95,73%, nilai tersebut lebih tinggi daripada penggunaan luas dengan menjumlahkan nilai piksel meskipun tidak signifikan. Rata-rata penghitungan area basah menggunakan sistem sekitar 0.8 detik atau 379 kali lebih cepat daripada menggunakan millimeter blok.

**Kata kunci**—Luas area basah, operasi morfologi, kode rantai

### Abstract

Calculation of wet areas carried out with the help of millimeter block paper has the disadvantage of copying the edges that are less precise and the calculation time is quite long. Another problem is the consistency and accuracy that is generated depends on the subjective factors of the person and one's fatigue. In order for the process to be faster and more consistent, the calculation process using image processing is very necessary. Image preprocessing includes cropping, grayscalling, lowpass filter averaging, conversion to binary image based on otsu thresholding, and complementing images to pixel objects of value 1. Segmentation with morphological operations, including opening operations to remove small objects around objects, Holes Filling operations to fill holes in objects, opening operations again to remove objects other than wet areas. The process of calculating wet areas uses chain code. Based on the results of testing of 81 images, the use of morphological operations is able to produce segmentation of wet areas that approach the original wet area. The scale value affects the accuracy and the best scale is obtained from the use of the ruler. The use of chain code is able to calculate the wet area on filter paper with an average accuracy of 95.73%, the value is higher than extensive use by summing the pixel value even though it is not significant. The average calculation of wet areas uses a system of about 0.8 seconds or 379 times faster than using millimeter block.

**Keywords**—Wet area, morphological operations, chain code

## 1. PENDAHULUAN

Luas area basah pada kertas saring merupakan bagian dalam menentukan nilai WHC. Luas area basah di sekeliling kertas saring beserta sampel daging ditandai dan setelah pengepresan selesai, dapat diukur (misalnya digambar pada kertas grafik) [1]. Pengukuran area basah menggunakan bantuan kertas milimeterblok (dinyatakan dalam  $\text{cm}^2$ ) [2]. Luas area yang dihitung dengan menggunakan metode balok atau *grid* (kotak) prinsipnya yaitu memperkirakan luas peta dengan membuat kotak atau balok yang kemudian dihitung luasnya berdasarkan perbandingan skala. Hasil penghitungan kedua metode tersebut tidak mutlak benar, hal ini karena ada wilayah pada area basah yang menjadi hilang atau bertambah. Sebagai contoh pada metode kotak jika wilayah pada area yang terpotong kotak bujur sangkar daerah yang ada kurang dari separuh maka daerah itu dihilangkan (dihitung 0), sedangkan jika daerahnya tergambar separuh atau lebih maka akan dihitung 1. Penghitungan dengan cara tersebut dapat menyebabkan luas area basah bisa menjadi lebih sempit atau justru lebih luas dari luas sebenarnya. Selain itu dapat pula terjadi ketidaktepatan dalam menyalin area basah pada plastik transparan, sehingga data luas yang diperoleh belum tentu konsisten keakuratannya.

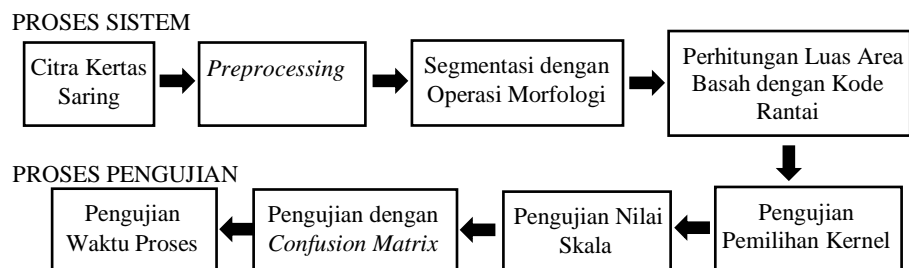
Pendeteksian objek area basah akan lebih konsisten dan seluruh area basah akan terhitung apabila prosesnya dilakukan dengan sistem pengolahan citra digital. Penelitian [3,4] menggunakan operasi morfologi yaitu *opening* untuk menghilangkan objek-objek kecil yang tidak akan diproses lebih lanjut dan *filling holes* untuk merekonstruksi citra seperti adanya lubang yang memutuskan suatu objek dalam citra. Penelitian [4,5] menggunakan kode rantai untuk menghitung luasan suatu objek. Kode rantai merupakan salah satu algoritma yang digunakan untuk memahami bentuk dan struktur suatu objek. Kode rantai banyak digunakan dalam penelusuran kontur karena memiliki informasi lebih dibandingkan dengan sebuah titik. Sistem penghitungan panjang dan luas menggunakan kode rantai digunakan untuk mengekstraksi fitur yang ada pada objek kemudian diolah menggunakan rumus sehingga hasil panjang dan luas dapat diketahui.

Perhitungan luas area basah akan lebih tepat karena area basah diambil citranya kemudian didigitalisasi sehingga didapatkan nilai-nilai pikselnya. Ukuran citra yang diambil menggunakan kamera apabila dihitung menggunakan metode balok akan didapatkan potongan yang lebih kecil daripada menggunakan milimeterblok. Milimeterblok hanya mampu menghitung dengan ketelitian panjang 100 kotak yang mewakili panjang kertas saring ukuran 10 cm. Menggunakan pengolahan citra dapat dihitung potongan citra berdasarkan panjang dan lebar pikselnya dengan tingkat ketelitian lebih dari 100 kotak. Menghitung luas dengan kode rantai juga akan meningkatkan nilai penghitungan karena dengan metode ini area diagonal dapat terhitung.

## 2. METODE PENELITIAN

### 2.1 Rancangan Penelitian

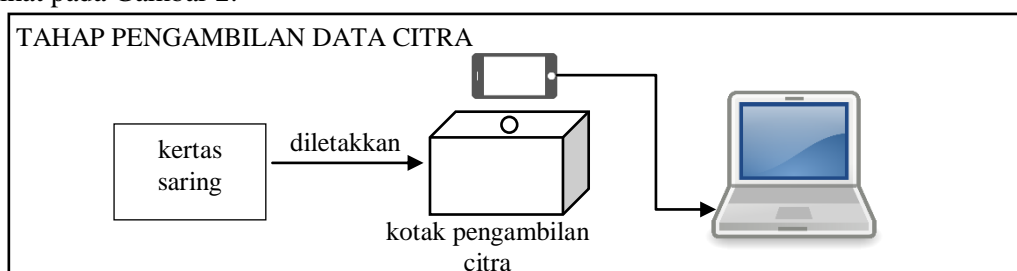
Penelitian ini terdiri dari proses sistem dan proses pengujian. Proses sistem terdiri dari beberapa tahapan yaitu input citra kertas saring, *preprocessing*, segmentasi dengan operasi morfologi, penghitungan luas area basah dan penghitungan nilai WHC. Proses pengujian yang dilakukan antara lain pengujian nilai skala, pengujian presisi dan akurasi dengan *confusion matrix*, serta pengujian waktu proses. Diagram alur penelitian dapat dilihat pada Gambar 1.



Gambar 1 Diagram blok alur penelitian

### 2. 1.1 Rancangan proses akuisisi citra

Penelitian dilakukan dengan sampel kertas saring sebanyak 27 buah yang diambil masing-masing sebanyak 3 kali sehingga total citra yang digunakan sebanyak 81 buah. Masukan sistem berupa data citra kertas saring dalam format JPEG (\*.jpeg atau \*.jpg). Proses pengambilan citra melalui kamera digital menggunakan *handphone* dengan resolusi 13 MP. Pengambilan citra dilakukan dengan memberikan cahaya lampu dengan objek berada pada posisi dan jarak yang sama, hal ini dilakukan dengan bantuan box yang dipersiapkan untuk proses pengambilan citra. Kotak pengambilan citra diberi lampu sebanyak 4 buah dan kamera diletakkan di tengah bagian atas kotak yang sudah dilubangi. Posisi kertas saring diletakkan ditengah-tengah kotak dengan lokasi yang sudah ditandai kemudian diambil citranya dan jarak antara kamera dengan kertas saring 16.5 cm. Proses pengambilan citra kertas saring dapat dilihat pada Gambar 2.

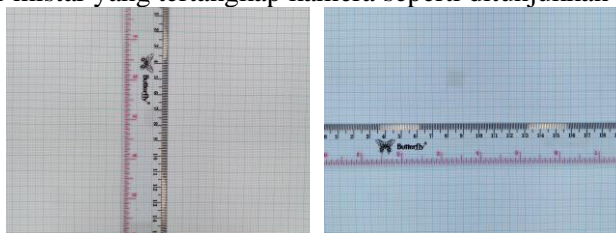


Gambar 2 Proses pengambilan data citra

### 2. 2 Rancangan proses kalibrasi

Kalibrasi digunakan untuk mendapatkan faktor pengali atau skala dari piksel citra agar dapat dikonversi panjangnya ke dalam centimeter. Untuk mendapatkan nilai akurasi dan presisi terbaik dalam menghitung luas objek ini dibutuhkan metode kalibrasi yang tepat. Beberapa penelitian yang menggunakan metode kalibrasi antara lain : kalibrasi panjang piksel citra untuk menghitung panjang pisang [7] dan kalibrasi alat ukur panjang beserta pengujiannya [8]. Kalibrasi ini dilakukan dengan beberapa cara, antara lain :

- 1) Kamera dalam posisi diam dan mistar diletakkan di depan kamera dengan jarak sama seperti saat pengambilan citra kertas saring. Ambil citra penggaris tersebut sehingga didapatkan panjang dan lebar mistar yang tertangkap kamera seperti ditunjukkan Gambar 3.



Gambar 3 Panjang dan lebar mistar tertangkap kamera

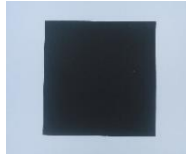
Nilai skala dihitung dengan persamaan (1).

$$s = \frac{LS}{LP} \quad (1)$$

Dengan LS yang merupakan luas sebenarnya. Luas sebenarnya dapat dihitung dengan menghitung panjang mistar yang tertangkap kamera dikalikan dengan lebar mistar yang tertangkap kamera. Sedangkan LP yang merupakan luas sistem dihitung dengan panjang piksel citra dikalikan dengan lebar piksel citra.

- 2) Menggunakan kalibrator ukuran panjang 1 cm dan lebar 1 cm. Kalibrator dibuat dengan kertas milimeter blok yang dipotong dengan bentuk persegi kemudian diberi warna hitam. Kertas warna hitam dipilih karena mewakili warna objek yang akan dihitung luasannya. Citra kertas kalibrator kemudian dihitung luasnya menggunakan sistem. Hasil dari

perhitungan sistem dalam satuan piksel. Skala dihitung dengan persamaan (1). Gambaran kertas kalibrator akan ditunjukkan pada Gambar 4.

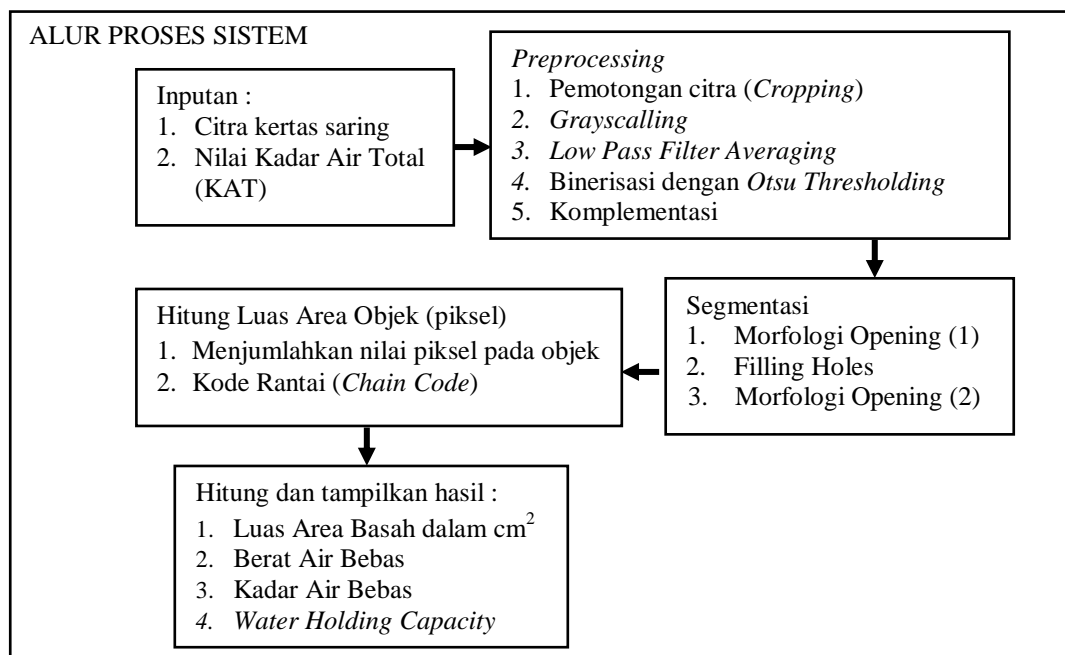


Gambar 4 Citra kertas kalibrator

- 3) Menggunakan cara pertama dengan ukuran panjang kalibrator sama. Sistem akan menghitung panjang dan lebar menggunakan fungsi *major axis length* dan *minor axis length*. Nilai panjang dan lebar digunakan untuk menghitung luas kalibrator kemudian digunakan untuk menghitung skala menggunakan persamaan (1).

### 2. 3 Implementasi perangkat lunak

Perangkat lunak pada penelitian ini dibuat pada Matlab R2017b. Dibutuhkan dua inputan untuk menghitung WHC, yaitu luas area basah untuk menghitung Kadar Air Bebas (KAB) dan Kadar Air Total (KAT). Inputan pertama didapatkan dari citra kertas saring yang diolah dengan pemrosesan citra sehingga diperoleh nilai luas area basahnya. Luas area objek hasil pemrosesan citra dapat dihitung luasnya dalam  $\text{cm}^2$  dengan menggunakan skala. Inputan kedua adalah nilai KAT hasil analisis proksimat yang tidak dilakukan oleh sistem, nilai KAT digunakan untuk menghitung nilai persentase WHC. Adapun alur pemrosesan sistem dapat dilihat pada Gambar 5.



Gambar 5 Diagram blok alur sistem

Sistem akan menampilkan 4 luaran hasil perhitungan, antara lain : Luas Area Basah, Berat Air Bebas, Kadar Air Bebas dan *Water Holding Capacity*. Luaran yang akan dianalisa lebih lanjut adalah Luas Area Basah, sedangkan luaran lainnya sebagai informasi tambahan hasil pengembangan dari Luas Area Basah yang dihasilkan.

#### 2. 3.1 Implementasi preprocessing

Sebelum citra kertas saring tersebut dapat diolah pada tahap selanjutnya maka harus dilakukan prapengolahan citra atau *preprocessing* terlebih dahulu. Proses prapengolahan terdiri

dari *grayscale*, *lowpass filter averaging*, *thresholding* atau binerisasi yang nilai threshold dihasilkan dengan Metode Otsu, kemudian terakhir adalah proses komplementasi.

Citra dipotong sehingga berukuran 2300x2300 piksel. Citra RGB kemudian dikonversi ke *grayscale*. Proses konversi menggunakan menggunakan metode *Luminance* yang dinyatakan dalam persamaan (2).

$$f(x,y) = 0.299 R + 0.587 G + 0.144 B \quad (2)$$

Citra *grayscale* kemudian diproses dengan *lowpass filter averaging*. Tujuan dilakukan *lowpass filter* adalah untuk menghaluskan (*smoothing*) dan mengurangi *noise* yang ada pada citra [9]. Ukuran kernel yang diujicobakan pada proses *filtering* ini adalah 3x3, 5x5 dan 7x7. Kernel pada proses *averaging* dapat dinyatakan dalam persamaan (3) dengan M merupakan jumlah piksel-piksel pada jendela NxN

$$h(x,y) = \frac{1}{M} \sum_{(k,l) \in N} f(k,l) \quad (3)$$

Selanjutnya, binerisasi citra digunakan untuk mengubah citra aras keabuan hasil *filtering* menjadi citra hitam putih (biner). Proses pengambangan dinyatakan dalam persamaan (4).

$$g(x,y) = \begin{cases} 0, & \text{jika } f(x,y) < T \\ 1, & \text{jika } f(x,y) \geq T \end{cases} \quad (4)$$

Metode Otsu digunakan dalam penentuan nilai batas ambang. Objek pada citra biner masih berwarna hitam yang mempunyai nilai piksel 0, sedangkan dalam melakukan proses morfologi dibutuhkan objek dengan nilai piksel 1 (warna putih) sehingga citra biner harus dikomplemenkan. Proses komplementasi citra biner menggunakan persamaan (5).

$$y = 1 - x \quad (5)$$

### 2. 3.2 Implementasi Segmentasi

Proses segmentasi dibutuhkan karena didalam citra masih terdapat objek-objek lain yang tidak diharapkan. Operasi-operasi morfologi seperti *opening* dan proses pengisian lubang (*filling holes*) digunakan dalam proses segmentasi ini.

Operasi *opening* pada dasarnya adalah operasi erosi diikuti dilasi dengan elemen penstruktur / kernel yang sama. Erosi digunakan untuk menghilangkan objek kecil dan dilasi digunakan untuk mengembalikan bentuk objek setelah dilakukan pengikisan. Proses *opening* yang pertama bertujuan untuk menghilangkan detail kecil dan terang, sedangkan fitur yang lebih besar dan terang relatif tidak berubah. Kernel yang akan diujicobakan adalah kernel dengan bentuk 'disk' berukuran jari-jari 4, 8 dan 12 atau berukuran matriks 7x7, 15x15 dan 23x23.

Operasi selanjutnya adalah penutupan lubang (*filling holes*) yang bertujuan untuk mengisi lubang di dalam objek atau menjadikan 0 di setiap tempat kecuali yang berada dalam border objek.

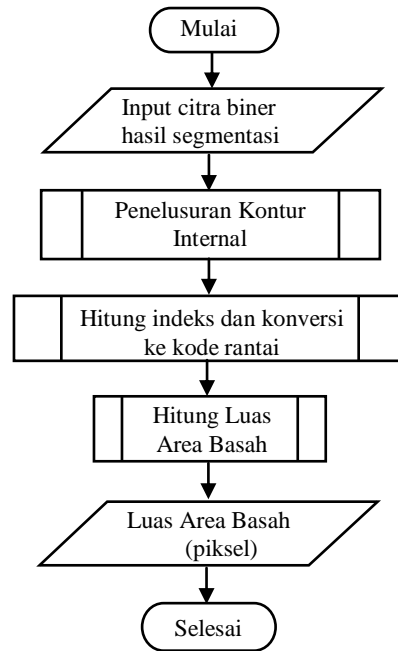
Operasi *opening* kedua digunakan untuk menghilangkan objek selain area basah, operasi ini akan menggunakan operasi 8 ketetanggaan dan menjadikan 0 pada objek yang nilainya dibawah batas ambang yang akan diujicoba dengan mengambil nilai batas atas dari luas objek selain area basah.

### 2. 3.3 Implementasi Penghitungan Luas

Luas area pada umumnya dihitung dengan menjumlahkan seluruh piksel bernilai 1 (objek berwarna putih) pada citra biner yang diperoleh. Penghitungan luas ini sesuai dengan persamaan (6).

$$Luas = \sum_{i=1}^n \sum_{j=1}^m f(i,j) \quad (6)$$

Cara lain dalam menghitung luas area objek yaitu dengan kode rantai (*chain code*). Kode rantai tidak hanya menghitung objek berdasarkan nilai pikselnya saja, namun kode-kode ganjil yang memiliki arah diagonal juga ditambahkan nilainya pada perhitungan luas. Proses menghitung luas dengan kode rantai digambarkan pada Gambar 6.



Gambar 6 Alur proses menghitung luas dengan kode rantai

Citra biner hasil segmentasi menjadi inputan untuk proses menghitung luas. Selanjutnya, dilakukan penelusuran kontur untuk mendapatkan koordinat-koordinat dari tepi objek yang dapat diperoleh menggunakan algoritma pelacakan kontur Moore. Hasil dari algoritma ini adalah koordinat tepian yang merupakan kontur internal objek. Koordinat kontur digunakan untuk menentukan indeks. Pembuatan indeks dihitung dengan persamaan (7), nilai indeks selanjutnya dikonversi menjadi kode rantai seperti yang ditunjukkan Tabel 1. Perhitungan luas area berdasarkan kode rantai secara umum dapat dinyatakan dalam Tabel 2.

$$indeks = 3 \Delta y + \Delta x + 5 \quad (7)$$

Tabel 1 Indeks dan kode rantai dua piksel yang bertetangga

$\Delta x$	$\Delta y$	Indeks	Kode Rantai
0	+1	8	6
0	-1	2	2
-1	+1	7	5
-1	-1	1	3
+1	+1	9	7
+1	-1	3	1
-1	0	4	4

Tabel 2 Luas berdasarkan kode rantai

Kode	Area
0	Area = Area + Y
1	Area = Area + (Y + 0.5)
2	Area = Area + 0
3	Area = Area - (Y + 0.5)
4	Area = Area - Y
5	Area = Area - (Y - 0.5)
6	Area = Area + 0
7	Area = Area + (Y - 0.5)

### 3. HASIL DAN PEMBAHASAN



#### 3.1 Pengujian pemilihan kernel

##### 3.1.1 Pengujian kernel pada lowpass filter averaging

Penelitian [5] tidak menggunakan proses perbaikan citra pada *preprocessing*-nya. Proses perbaikan citra pada penelitian ini menggunakan *lowpass filter averaging* yang

merupakan proses penghalusan (*smoothing*). Tabel 3 menunjukkan setelah proses binerisasi, perbedaan terlihat jelas antara citra yang dilewatkan *filter* dan tidak dilewatkan *filter*.

Tabel 3 Perbedaan hasil binerisasi citra tanpa LPF dan citra dengan LPF kernel 3x3

	
Hasil Binerisasi citra tanpa LPF memiliki banyak <i>noise</i>	Hasil Binerisasi citra dengan LPF memiliki sedikit <i>noise</i>

Tabel 3 menunjukkan bahwa citra hasil *filtering* lebih halus. Citra menjadi lebih halus karena variansi nilai piksel dari piksel berikutnya berkurang. Hasil pengurangan *noise* pada proses *smoothing* sudah sangat baik, namun terlihat *noise* masih terdeteksi dengan adanya titik putih di dalam objek dan titik hitam di luar objek.

Pengujian kernel filter digunakan untuk mencari kernel yang menghasilkan citra konsisten baik dengan hasil objek area basah terlihat jelas. Pengujian menggunakan 3 kernel ujicoba yaitu ukuran 3x3, 5x5 dan 7x7. Citra yang dinilai hasilnya adalah citra *grayscale* hasil *filtering* yang sudah dikonversi menjadi biner. Hasil pengujian ditunjukkan pada Tabel 4.

Tabel 4 Perbedaan hasil binerisasi citra LPF dengan berbagai kernel

Nama Citra	Kernel 3x3	Kernel 5x5	Kernel 7x7
P1U3_1	√	√	√
P3U1_2	√	√	×
P5U1_1	√	√	×
P6U3_3	√	×	×
P9U2_3	√	√	√

Keterangan : (√) Objek area basah terlihat jelas  
(×) Objek area basah tidak terlihat jelas


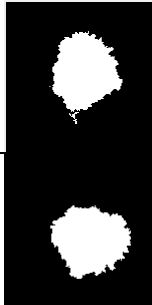


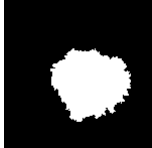
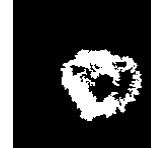
Tabel 4 menunjukkan bahwa kernel ukuran 3x3 mampu menghasilkan citra biner dengan objek area basah terlihat jelas secara konsisten untuk semua citra uji. Kernel ukuran 5x5 dan 7x7 tidak secara konsisten mampu menghasilkan citra yang diharapkan setelah proses *Lowpass Filter* dilakukan.

Pengujian dengan berbagai kernel secara garis besar dapat diketahui bahwa jika semakin besar ukuran kernel maka jendela konvolusi semakin besar. Hal tersebut mengakibatkan derajat pemfilteran semakin besar pada hasil prosesnya. Akibat penghapusan *noise* yang disebut penghalusan atau pengaburan citra dengan filter yang besar juga menghilangkan informasi yang dibutuhkan pada citra.

### 3.1.2 Pengujian kernel pada operasi morfologi opening

Pengujian kernel kedua pada proses morfologi opening menggunakan 3 nilai jari-jari (Radius=R) dalam pemilihan kernel, yaitu 4, 8, dan 12. Penggunaan kernel atau elemen penstruktur sangat berpengaruh pada hasil pengolahan pada tahap ini. Tabel 5 menunjukkan perbedaan hasil dari penggunaan ukuran kernel bentuk disk yang berbeda nilai radiusnya.

Tabel 5 Perbedaan hasil proses morfologi dengan berbagai ukuran kernel


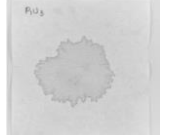
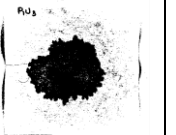
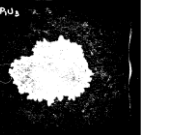
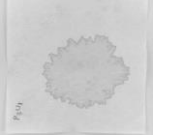
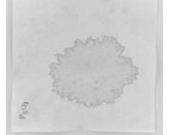
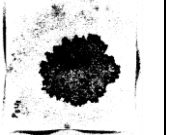
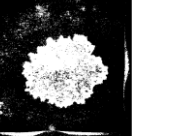
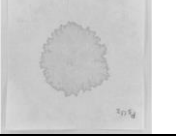


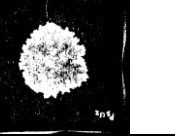
Nama File	R = 4 (7x7)	R = 8 (15x15)	R = 12 (23x23)
P6U3_3			
P6U3_2			

Tabel 5 menunjukkan bahwa kernel ukuran 23x23 atau radius 12 mampu menghasilkan objek yang sempurna untuk sampel uji P6U3\_3, namun jika kernel tersebut digunakan untuk citra lain misalnya sampel uji P6U3\_2 tidak mampu menghasilkan objek yang sempurna setelah proses *filling holes* dan *opening* kedua. Sehingga diputuskan untuk menggunakan kernel ukuran 15x15 dengan radius 8 yang mampu mengurangi objek berlebih pada sampel uji P6U3\_3 namun tidak merusak kesempurnaan objek dari sampel uji P6U3\_2.

### 3.2 Hasil implementasi

Pengujian sistem dilakukan pada citra sebanyak 81 buah. Implementasi pada tahap ini menggunakan kernel yang sudah dipilih dari hasil pengujian kernel pada sub bab 3.1 yaitu kernel *filtering* menggunakan ukuran kernel 3x3 dan kernel pada operasi morfologi menggunakan kernel radius 8 dengan ukuran 15x15. Tabel 6 menunjukkan hasil implementasi *preprocessing* pada beberapa citra.

Tabel 6 Contoh beberapa citra hasil *preprocessing*

Nama File	Hasil Grayscale	Hasil LPF Averaging	Hasil Binerisasi	Hasil Komplementasi
P1U3_1				
P3U1_2				
P9U2_3				

Proses kedua adalah segmentasi. Segmentasi bertujuan untuk mendapatkan objek tunggal berupa area basah tanpa adanya *noise*. Citra tanpa *noise* adalah citra hasil pemisahan yang telah mengalami penghapusan objek yang terindikasi bukan objek area basah (*noise*), sehingga citra tersebut hanya menyisakan area basah yang diharapkan.

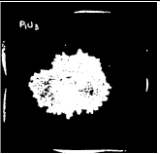
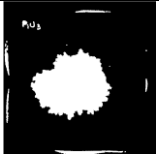
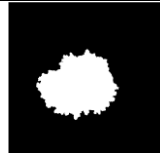
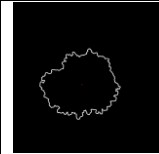
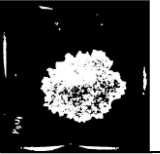


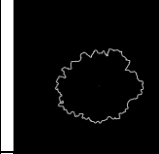
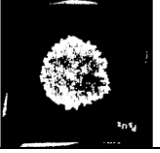

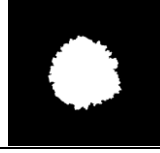
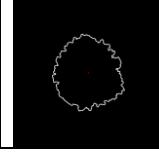
Penelitian sebelumnya yang menggunakan operasi morfologi erosi dan dilasi dengan nilai *structuring element* (SE) berbeda-beda untuk tiap citra tergantung dari berapa banyak objek yang diinginkan. Penelitian ini menggunakan satu nilai SE sesuai hasil pengujian kernel pada sub bab 3.1.2 yang sudah dibahas sebelumnya. Operasi morfologi yang digunakan pertama kali adalah operasi *opening* yang merupakan operasi erosi dilanjutkan dengan dilasi dengan



menggunakan kernel yang sama. Operasi *opening* yang pertama digunakan untuk menghilangkan objek-objek yang lebih kecil dari ukuran elemen penstruktur atau kernel yang ditetapkan.


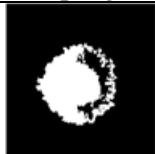

Proses kedua adalah *filling holes* untuk membuat lubang di dalam objek. Terakhir, proses *opening* kedua digunakan untuk menghapus objek selain area basah, hal ini dilakukan dengan menghitung luas area pada masing-masing objek kemudian ditentukan nilai *threshold* dimana objek-objek yang berada dibawah nilai tersebut kemudian dihapus atau diset menjadi 0. Tampilan hasil dari proses segmentasi ditunjukkan oleh Tabel 7.

Tabel 7 Contoh beberapa citra hasil segmentasi

Nama File	Hasil Opening (1)	Hasil <i>Filling Holes</i>	Hasil Opening (2)	Tepi Objek
P1U3_1				
P3U1_2				
P9U2_3				

Tabel 7 menunjukkan bahwa objek area basah yang diharapkan sudah bersih dari *noise* dan hanya ada objek tunggal yaitu area basah. Penggunaan operasi morfologi *opening* dan *filling holes* pada tahap segmentasi ini dapat menghasilkan citra deteksi tepi dengan bentuk objek area basah mendekati aslinya pada 78 dari 81 buah citra uji, atau sebanyak 96%. Meskipun 78 citra uji tersebut sudah bersih dari *noise*, ada 3 buah citra uji yang masih terdapat objek yang proses pengolahannya belum sempurna seperti ditunjukkan oleh Tabel 8.

Tabel 8 Citra dengan hasil pengolahan kurang sempurna

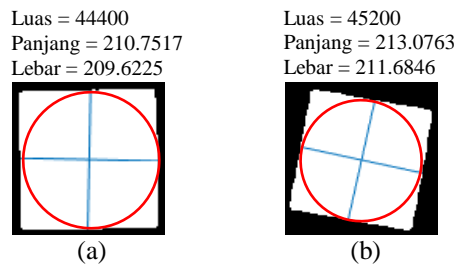
		
P6U3_3 <i>Over counted</i>	P8U1_3 <i>Not Counted</i>	P9U1_1 <i>Not Counted</i>

Kesalahan pemisahan objek dapat menyebabkan *over counted* pada proses perhitungan luas area seperti yang ditunjukkan Tabel 8 untuk citra P6U3\_3. Pemilihan kernel yang tidak tepat dapat membuat objek yang harusnya terhubung menjadi terpisah, sehingga saat proses *filling holes* terdapat area yang tidak bernilai 1 atau berwarna putih dan mengakibatkan area tersebut tidak terhitung (*not counted*) saat proses penghitungan luas.

### 3.3 Hasil pengujian nilai skala

Pengujian nilai skala ini bertujuan untuk membandingkan nilai piksel yang tertangkap kamera dengan skala mistar yang digunakan sebagai skala standar. Skala yang sudah di bahas pada tahap kalibrasi terdiri dari 3 jenis yaitu skala standar, skala 1D dan skala 2D. Nilai Skala standar yang didapatkan dari mistar seperti pada Gambar 3. Skala 1D merupakan nilai skala

yang didapatkan dari penghitungan luas dari kalibrator. Adapun skala 2D adalah nilai skala yang didapatkan dari perkalian panjang kalibrator dan lebar kalibrator. Hasil panjang dan lebar kalibrator apabila divisualisasikan dengan sistem dapat dilihat pada Gambar 7.



Gambar 7 Hasil kalibrator

Berdasarkan Gambar 7 diketahui bahwa sumbu *majoraxis* dan *minoraxis* membentuk sumbu horizontal dan vertikal. Panjang dan lebar citra dihitung menggunakan bantuan *majoraxis* dan *minoraxis*. Nilai skala dihitung menggunakan panjang dan lebar hasil perhitungan pada Gambar 7 (a). Tabel 9 menunjukkan penentuan nilai skala yang digunakan.

Tabel 9 Penentuan nilai skala

Nama	Ukuran sebenarnya	Ukuran di dalam citra	Skala
Skala Standar	Panjang = 19.6 cm	Panjang = 4160 piksel	2.205E-05
	Lebar = 14.6 cm	Lebar = 3120 piksel	
Skala 1D	Luas = 1 cm <sup>2</sup>	Luas = 44400 piksel	2.252E-05
Skala 2D	Panjang = 1 cm	Panjang = 210.7517 piksel	2.264E-05
	Lebar = 1 cm	Lebar = 209.6225 piksel	

Berdasarkan Tabel 9 diketahui bahwa ketiga nilai skala hampir sama. Nilai skala 1D dan 2D sangat dekat. Namun, skala 1D lebih dekat dengan skala standar. Ketiga nilai skala diujicoba untuk menghitung luas area basah dalam satuan cm<sup>2</sup>. Tabel 10 menunjukkan beberapa hasil penggunaan nilai skala pada data luas yang dihasilkan sistem. Luas secara umum berdasarkan nilai piksel ditunjukkan sebagai L1 dan luas berdasarkan *chain code* ditunjukkan sebagai L2.

Tabel 10 Hasil penggunaan nilai skala

Nama File	L1 (piksel)	L2 (piksel)	Luas MmBlok	Skala Standar		Skala 1D		Skala 2D	
				L1 (cm <sup>2</sup> )	L2 (cm <sup>2</sup> )	L1 (cm <sup>2</sup> )	L2 (cm <sup>2</sup> )	L1 (cm <sup>2</sup> )	L2 (cm <sup>2</sup> )
P3U3_1	876768	874373	20.49	19.33	19.28	19.75	19.69	19.85	19.79
P3U3_2	870594	868223.5	20.49	19.19	19.14	19.61	19.55	19.71	19.65
P3U3_3	872540	870144	20.49	19.24	19.18	19.65	19.60	19.75	19.70
P4U1_1	903243	900728	19.69	19.91	19.86	20.34	20.29	20.45	20.39
P4U1_2	975387	972617	19.69	21.50	21.44	21.97	21.91	22.08	22.02
P4U1_3	968898	966199	19.69	21.36	21.30	21.82	21.76	21.93	21.87

### 3.4 Hasil pengujian dengan *confusion matrix*

Proses pengujian akurasi digunakan untuk mengukur kinerja algoritma pada program yang telah dibuat untuk menghitung luasan suatu objek yang diukur menggunakan *confusion matrix*. Akurasi bukanlah ukuran yang cocok untuk beberapa kasus yang melibatkan data tidak seimbang, karena mungkin mencapai akurasi yang sangat tinggi, tetapi tidak mengidentifikasi gangguan [10]. Tabel 11 menunjukkan contoh penghitungan nilai akurasi.

Tabel 11 Perhitungan nilai akurasi

No	Nama File	Luas Sistem	Luas Manual	Correctly Counted (TP)	Over Counted (FP)	Not Counted (FN)	Presisi $\left(\frac{TP}{FP + TP}\right)$	Recall $\left(\frac{TP}{FN + TP}\right)$	Akurasi $\left(\frac{2 \cdot \text{Presisi} \cdot \text{Recall}}{\text{Presisi} + \text{Recall}} \cdot 100\%\right)$
1	P3U3_1	19.33	20.49	19.33	0	1.16	1	0.94	97.09
2	P3U3_2	19.19	20.49	19.19	0	1.30	1	0.94	96.74
3	P3U3_3	19.24	20.49	19.24	0	1.25	1	0.94	96.85
4	P4U1_1	19.91	19.69	19.69	0.22	0	0.99	1	99.43
5	P4U1_2	21.50	19.69	19.69	1.81	0	0.92	1	95.59
6	P4U1_3	21.36	19.69	19.69	1.67	0	0.92	1	95.93

Luas sistem dalam Tabel 11 adalah luas sistem dihitung dengan panjang piksel citra dikalikan dengan lebar piksel citra, luas hasil penghitungan sistem kemudian dikalikan dengan skala sehingga didapatkan nilai luas dalam satuan  $\text{cm}^2$  seperti yang ditunjukkan Tabel 10. Semua nilai luas dengan berbagai skala diujicoba dan dihitung nilai akurasinya. Tabel 12 menunjukkan perbandingan hasil akurasi secara keseluruhan.

Tabel 12 Perbandingan hasil akurasi

	Skala Standar		Skala 1D		Skala 2D	
	L1	L2	L1	L2	L1	L2
Akurasi Tertinggi	99.43	99.57	99.26	99.13	99.51	99.38
Akurasi Terendah	90.86	91.07	89.81	90.02	89.56	89.77
Rata-rata Akurasi	95.72	95.73	95.40	95.43	95.32	95.36

Tabel 10 menunjukkan bahwa penggunaan skala 1D dan 2D menunjukkan hasil sedikit lebih rendah. Namun, perbedaan nilai skala tidak berpengaruh terhadap akurasi secara signifikan. Nilai skala standar mampu memaksimalkan hasil akurasi dengan pencapaian rata-rata akurasi 95.73%. Hal ini dapat membuktikan bahwa penggunaan mistar sebagai skala standar lebih baik untuk diimplementasikan dalam program. Nilai yang didapat dari skala mistar juga tidak berubah-ubah seperti halnya penggunaan nilai luas ataupun *majoraxislength* pada kalibrator. Luas berdasarkan *Chain Code* (L2) mempunyai akurasi selalu lebih tinggi daripada luas dengan menjumlahkan nilai piksel (L1) walaupun tidak signifikan.

### 3.5 Hasil pengujian waktu proses

Proses pengujian waktu proses ini dilakukan dengan membandingkan hasil penghitungan area basah antara penggunaan millimeterblok dan sistem. Pengujian waktu proses menghitung area basah menggunakan millimeterblok dibutuhkan waktu antara 4-7 menit dan rata-rata sekitar 5 menit. Adapun pengujian sistem untuk menghitung luas hingga didapatkan nilai WHC membutuhkan waktu antara 0.3-2.4 detik dan rata-rata 0.8 detik. Dibandingkan dengan menggunakan millimeterblok, rata-rata penghitungan area basah menggunakan sistem 379 kali lebih cepat. Hal ini dapat membantu para peneliti dalam bidang pangan atau peternakan dalam menghitung luas area basah kertas saring dengan lebih cepat.

## 4. KESIMPULAN

Penggunaan operasi morfologi mampu menghasilkan segmentasi dengan bentuk objek area basah yang mendekati aslinya sebesar 96%. Menghitung luas dengan *chain code* menunjukkan hasil akurasi lebih tinggi daripada penggunaan luas dengan menjumlahkan nilai piksel walaupun tidak signifikan. Penggunaan *chain code* sudah tepat untuk menghitung luas objek yang menyerupai pulau dan memiliki kontur berliku-liku. Metode yang diusulkan berhasil melakukan proses penghitungan luas area dengan akurasi tertinggi sebesar 99.57% dan terendah sebesar 91.07%. Rata-rata akurasi sistem secara keseluruhan dari 81 citra adalah sebesar 95.73%. Nilai rata-rata penghitungan area basah menggunakan sistem sekitar 0.8 detik, sedangkan rata-rata waktu proses menghitung dengan millimeterblok selama 5 menit. Hal ini berarti bahwa penghitungan dengan sistem 379 kali lebih cepat.

## 5. SARAN

Penelitian selanjutnya dapat menggunakan citra kertas saring yang lebih beragam dan perlu diujicoba dengan metode lain untuk mengatasi citra dengan objek berdempet yang ekstrim.

## DAFTAR PUSTAKA

- [1] Sundari, "Nanoenkapsulasi ekstrak kunyit dengan kitosan dan sodium-tripolifosfat sebagai aditif pakan dalam upaya perbaikan pencernaan, kinerja dan kualitas daging ayam broiler," Proquest Dissertation Universitas Gadjah Mada, 2014. Available : <http://etd.repository.ugm.ac.id>.
- [2] R. E. Dewayani and H. N. O. Sjojfan, "Pengaruh penggunaan onggok dan ampas tahu terfermentasi mix culture aspergillus niger dan rhizopus oligosporus sebagai pengganti jagung dalam pakan terhadap kualitas fisik daging ayam pedaging," *J. Ilmu dan Teknol. Has. Ternak*, vol. 10, no. 1, pp. 9–17, 2015 [Online]. Available : <https://jitek.ub.ac.id/index.php/jitek/article/view/237>. [Accessed: 16-Mar-2018]
- [3] A. Susanto, "Penerapan operasi morfologi matematika citra digital untuk ekstraksi area plat nomor kendaraan bermotor," *Pseudocode*, vol. VI, no. 1, pp. 49–57, 2019 [Online]. Available : <https://ejournal.unib.ac.id/index.php/pseudocode/article/view/7265>. [Accessed: 16-Mar-2019]
- [4] Arini, F. Fahrianto, A. Agusta, and A. T. Muharam, "Pendeteksian posisi plat nomor mobil menggunakan metode morfologi dengan operasi dilasi, filling holes, dan opening," *Tek. Inform.*, vol. 8, no. 1, pp. 10–15, 2015 [Online]. Available : <http://repository.uinjkt.ac.id/dspace/bitstream/123456789/31590/1/Arini%20dkk.pdf>. [Accessed: 16-Mar-2018]
- [5] A. Rizkinawati, I. Santoso, and A. A. Zahra, "Aplikasi metode kode rantai untuk menghitung panjang perbatasan daerah dan luas pulau di indonesia," *TRANSMISI*, vol. 17, no. 2, 2015 [Online]. Available : <https://ejournal.undip.ac.id/index.php/transmisi/article/view/8574>. [Accessed: 16-Mar-2018]
- [6] S. A. Syakri, Mulyadi, and Z. K. Simbolon, "Identifikasi tingkat kebulatan buah pepaya berdasarkan luas objek dengan pengolahan citra," *Infomedia*, vol. 2, no. Desember, p. 2, 2017 [Online]. Available : <http://ejournal.pnl.ac.id/index.php/infomedia/article/view/517/533>. [Accessed: 16-Mar-2018]
- [7] Y. P. Wiharja and A. Harjoko, "Pemrosesan citra digital untuk klasifikasi mutu buah pisang menggunakan jaringan saraf tiruan," *IJEIS*, vol. 4, no. 1, pp. 57–68, 2014 [Online]. Available : <https://jurnal.ugm.ac.id/ijeis/article/view/4222>. [Accessed: 16-Mar-2018]
- [8] A. Wicaksono and I. D. W. Susanto, "Sistem otomasi penggerak kamera dengan motor step sebagai alat bantu kalibrasi alat ukur panjang," *J.Oto.Ktrl.Inst.*, vol. 6, no. 2, pp. 105–120, 2014 [Online]. Available : <https://ejournal.unib.ac.id/index.php/pseudocode/article/view/7265/3632>. [Accessed: 16-Mar-2018]
- [9] P. Hidayatullah, *Pengolahan citra digital teori dan aplikasi nyata*. Bandung: Informatika, 2017.
- [10] A. Luque, A. Carrasco, A. Martin, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognit.*, vol. 91, pp. 216–231, 2019 [Online]. Available : <https://www.sciencedirect.com/science/article/pii/S0031320319300950>. [Accessed: 1-Agust-2019]