

IMPLEMENTASI ALGORITMA DETEKSI TEPI UNTUK MENENTUKAN KUALITAS SURFACE PADA MUTIARA LAUT DENGAN MENGGUNAKAN METODE SOBEL

(Study Kasus PT. AUTORE Pearl Farm Show Room)

Imam Munandar¹, Eko Budi Cahyono², Ilyas Nuryasin³

¹imammunandar_2607@yahoo.com, ²ekobudi@umm.ac.id, ³ilyas@umm.ac.id

Jurusan Teknik Informatika Universitas Muhammadiyah Malang

ABSTRACT

The quality of pearl is generally determined by perfection of shape and the smoothness of surface. The quality range is classified by counting the number of blemishes and smoothness of the pearls. This research aims to developing an application to detect the level of quality sea pearls by implamenting the image segmentation. Image segmentation has the meaning of dividing an image into homogeneous regions based on the similarity of certain criteria. There are some in the image segmentation process of detecting one of them is Edge Detection. Edge detection is performed to detect the operating margin (edges) that limit by two homogeneous image regions which have defferent levels of brightness. Sobel edge detection is the best edge detection methods compared with other edge detection methods such as Prewitt and Robert. Sobel perform 3x3 mask, wider than both methods. The method will then be used as a connected component labeling to distinguish group of pixels are connected as a spot or stain on pearls and labeling. The result of this research, Sobel method can produce a smoother edge and has a success rate by 75% of four times calibration testing with different qualities of sea pearl.

Keywords : Image Segmentation, Edge Detection, Sobel, Connected Components Labeling

I. Pendahuluan

1.1. Latar Belakang

Dengan kemajuan zaman dan teknologi, melakukan tahapan kualitas pada mutiara laut secara manual bisa dikatakan sangat kurang efisien mengingat perkembangan teknologi yang begitu pesat. Ada beberapa metode atau teknik yang dapat dijadikan sebagai acuan untuk mengidentifikasi tingkat kualitas dari mutiara laut tersebut.

Pengidentifikasian dapat dilakukan dengan menggunakan program simulasi komputer yang dapat melakukan simulasi pengolahan citra dengan cepat. Dengan menggunakan algoritma deteksi tepi dan kontur citra mutiara yang telah tercapture dapat diidentifikasi. Deteksi tepi merupakan satu tahapan pada pengolahan citra digital (*Image Processing*).

Pengolahan citra *digital* merupakan pengolahan dan *analisis* gambar dua dimensi menggunakan computer (Putra, 2010). Pengolahan citra bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau computer.

Deteksi tepi merupakan proses untuk menentukan titik-titik yang merupakan tepi objek (Hambali, 2011). Data yang digunakan dalam deteksi tepi berupa citra digital. Ada beberapa metode yang dapat digunakan untuk pendeteksian tepi, contohnya adalah metode Roberts dan Sobel. Kedua metode ini dalam perhitungannya berbasis

gradient. Metode Sobel lebih baik di bandingkan dengan metode Roberts, karena operator dan ukuran *kernel* yang digunakan berbeda. Metode Roberts menggunakan operator Roberts berukuran 2x2 piksel, sedangkan metode Sobel menggunakan operator Sobel berukuran 3x3 piksel. Metode Roberts lebih menekankan perhitungan gradient arah diagonal sedangkan Sobel lebih kearah vertikal dan horizontal.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut maka dapat diuraikan rumusan masalah sebagai berikut :

- 1) Bagaimana membangun system aplikasi untuk menentukan surface pada mutiara laut dari citra yang inputkan ?
- 2) Bagaimana mengimplementasikan algoritma deteksi tepi pada citra yang ditentukan.

1.3 Tujuan

Tujuan yang ini dicapai dalam penelitian dan pembuatan tugas akhir ini adalah bagaimana merancang dan membangun perangkat lunak untuk menentukan kualitas mutiara laut dengan mengikuti faktor penentu kualitas mutiara yaitu pada tahapan surface dengan menggunakan algoritma deteksi tepi atau kontur dan deteksi objek.

1.4 Batasan Masalah

Terdapat beberapa batasan masalah yang diangkat sebagai parameter pengerjaan tugas akhir ini diantaranya adalah sebagai berikut :

- 1) Mutiara yang dideteksi hanyalah mutiara laut.
- 2) Mengetahui grading surface dalam menentukan kualitas mutiara laut.
- 3) Inputan berupa citra mutiara laut yang sudah tercapture dan yang akan diidentifikasi.
- 4) Aplikasi dibuat menggunakan bahasa pemrograman java, berbasis aplikasi desktop

II. Dasar Teori

Citra digital merupakan citra yang disimpan dalam format digital (dalam bentuk *file*). “Pengolahan citra digital menurut (Putra, 2010) adalah proses yang bertujuan untuk memanipulasi dan menganalisis citra digital dengan bantuan komputer”.

2.1. Grayscale

(Wijaya & Tjiharjadi, 2009) menjelaskan Proses awal yang sering dilakukan pada image processing adalah mengubah citra berwarna menjadi citra gray-scale. Hal ini dilakukan untuk menyederhanakan model citra. Di dalam suatu gambar true color (RGB) terdapat tiga layer matriks, yaitu R-layer, G-layer, dan B-layer. Pada image processing dilakukan proses-proses terhadap ketiga layer tersebut, berarti dilakukan perhitungan yang sama pada setiap layer. Dengan demikian konsep grey-scale adalah mengubah tiga layer tersebut menjadi satu layer matriks grey-scale, yang menghasilkan satu citra grey-scale. Di dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan.

Secara umum untuk menghasilkan citra grey-scale, konversi dilakukan dengan mengambil rata-rata dari nilai R, G dan B sehingga menghasilkan nilai S sebagai nilai grey-scale. Dengan rumus sebagai berikut :

$$S = \frac{r + g + b}{3} \dots\dots\dots (1)$$

Pada penjelasan di atas pengubah citra berwarna menjadi citra grey-scale adalah mencari nilai rata-rata grey-scale dari setiap layer r, g, dan b.

2.2. Histogram

Histogram dalam pengolahan citra digital merupakan grafik yang menunjukkan distribusi dari intensitas citra (Anggraeni, 2007). Untuk meningkatkan kualitas citra salah satunya dapat dilakukan dengan ekualisasi histogram.

Contoh data sebagai berikut :

$$X = 1 \ 3 \ 2 \ 5 \ 3 \ 0 \ 2 \ 1 \ 2 \ 4 \ 2 \ 3$$

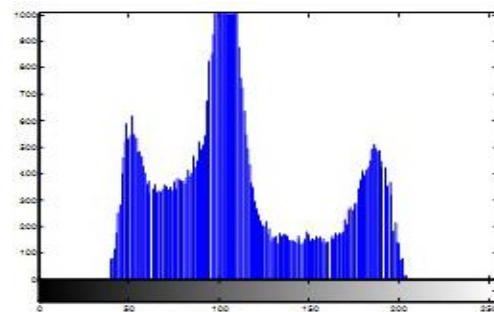


(a)

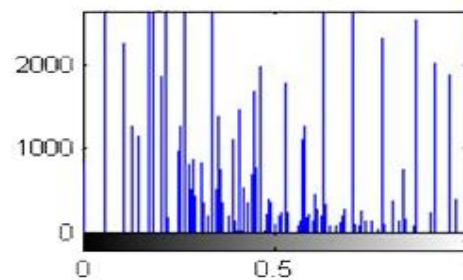
(b)

Gambar 2.1 (a) Citra Beras, (b) Citra Pohon

Maka histogram yang akan muncul pada setiap nilai yaitu : nilai 0 muncul 1 kali, nilai 1 muncul 2 kali, nilai 2 muncul 4 kali, nilai 3 muncul 3 kali, nilai 4 muncul 1 kali, dan nilai 5 muncul 1 kali. Karena citra mempunyai derajat keabuan 256 yaitu (0-255) maka histogram menyatakan jumlah kemunculan setiap nilai 0-255. Berikut adalah beberapa contoh citra dan histogramnya.



(a)



(b)

Gambar 2.2 (a). Histogram Citra Beras, (b) Histogram Citra Pohon

2.3. Thresholding

Thresholding adalah proses untuk mengelompokkan semua piksel pada citra dengan nilai tertentu menjadi dua bagian dengan nilai *grey level* yang telah ditentukan. Pembuatan citra biner adalah salah satu bentuk *Thresholding* dengan nilai 0 dan 1, yaitu melakukan perubahan semua nilai piksel yang lebih kecil dari nilai ambang menjadi 0 (Wijaya & Tjiharjadi, 2009).

Thresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Dengan menggunakan *thresholding* maka derajat keabuan akan bisa diubah sesuai dengan keinginan, sebagai contoh diinginkan menggunakan derajat keabuan 16, maka tinggal membagi nilai derajat keabuan dengan 16. Proses *Thresholding* ini pada dasarnya adalah sebagai proses pengubah kuantitas pada citra, sehingga untuk melakukan *thresholding* dengan derajat keabuan dapat digunakan rumus :

$$X = \frac{w}{b} \dots\dots\dots (2)$$

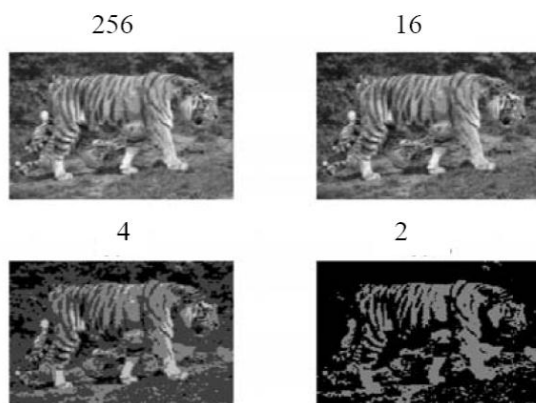
Keterangan :

X : nilai derajat keabuan setelah *Thresholding*

w : nilai derajat keabuan sebelum *Thresholding*

b : jumlah derajat keabuan yang diinginkan

Pada *Thresholding* yang tinggi tidak tampak perbedaan karena keterbatasan mata, tetapi untuk *Thresholding* tingkat rendah seperti 2, 4, 8, 16 akan tampak perbedaannya.



Gambar 2.3 Grey-scale dengan beberapa tingkat *Thresholding*

2.4. Segmentasi Citra

Segmentasi citra merupakan bagian dari proses pengolahan citra. Proses segmentasi citra ini lebih banyak merupakan suatu proses pra pengolahan pada system pengenalan objek dalam citra.

(Hardiyanto, Purwananto, Kom, Kom, & Soelaiman, 2012) menyatakan bahwa segmentasi adalah proses membagi citra digital menjadi beberapa daerah atau kelompok, dimana masing masing daerah terdiri dari sekumpulan piksel. Segmentasi merupakan suatu bagian yang sangat penting dalam analisis citra secara otomatis, sebab pada prosedur ini obyek yang diinginkan akan disadap untuk proses selanjutnya, misalnya : pada pengenalan pola.

Secara umum algoritma segmentasi citra didasarkan pada satu dari dua property nilai intensitas yaitu mendeteksi diskontinuitas atau mendeteksi kesamaan. Diskontinuitas memiliki pendekatan memecah atau memilih citra berdasarkan perubahan intensitas yang tiba-tiba atau cukup besar. Proses segmentasi berdasarkan mendeteksi diskontinuitas antara lain : Deteksi Titik, Deteksi Garis, dan Deteksi Tepi. Sementara itu untuk similaritas berdasarkan pada memecah citra ke dalam wilayah yang sama menurut beberapa kriteria yang telah di tentukan, antara lain seperti proses : *thresholding*, *region growing*, dan *region splitting* dan *merging*.

2.5. Deteksi Tepi Sobel

sobel merupakan salah satu pengembangan dari teknik *edge detection* sebelumnya (Metode Robert) dengan menggunakan HPF (*High Pass Filter*) yang diberi satu angka nol penyangga (Yunus, n.d.), juga pengembangan dari operator Prewit. Algoritma ini termasuk algoritma pemrograman yang berfungsi sebagai *filter image*. *Filter* ini mendeteksi keseluruhan edge yang ada. Dalam prosesnya filter ini menggunakan sebuah operator, yang dinamakan operator sobel. Operator sobel menggunakan matriks NxN dengan berordo 3 x 3, 5 x 5, 7 x 7, dan sebagainya. Matriks seperti ini digunakan untuk mempermudah untuk mendapatkan piksel tengah sehingga menjadi titik tengah matrik (a_{ij}). Piksel tengah ini merupakan piksel yang akan diperiksa. Cara pemanfaatan matrik ini sama seperti pemakaian sebuah grid, yaitu dengan cara memasukkan piksel-piksel disekitar yang sedang diperiksa (piksel tengah) ke dalam matrik. Cara yang demikian disebut *spatial filtering*.

a_0	a_1	a_2
a_7	a_{ij}	a_3
a_6	a_5	a_4

Gambar 2.4 Matriks 3 x 3 pada area image

Didefinisikan G_y sebagai arah penelusuran secara vertikal

$$G_y = (a_0 - a_6) + (a_2 - a_4) + 2(a_1 - a_5) \dots\dots\dots (3)$$

$$G_y = (a_0 + 2a_1 + a_2) - (a_4 + 2a_5 + a_6)$$

Dan G_x sebagai arah penelusuran secara horizontal

$$G_x = (a_0 - a_2) + (a_6 - a_4) + 2(a_7 - a_3) \dots\dots\dots (4)$$

$$G_x = (a_0 + 2a_7 + a_6) - (a_2 + a_4 + 2a_3)$$

Definisi menggunakan nilai mutlak diberikan

$$G = |G_x| + |G_y| \dots\dots\dots (5)$$

Dengan membandingkan area diatas dari persamaan (3) bisa dilihat bahwa G_y adalah berbeda antara baris pertama dan baris ketiga, dimana elemen terdekat a_{ij} yaitu (a_1 dan a_5) lebih besar dua kali dibandingkan dengan nilai disekelilingnya (hal ini berdasarkan intusi wilayah/ area0 juga pada persamaan (4), G_x adalah berbeda antara kolom a_3 dan a_7 . G_x adalah arah dari x dan G_y merupakan arah dari y. persamaan dari (3) dan (4) dapat diimplementasikan dari operasi sobel didapat nilai hasil persamaan (5).

Teknik *spatial filtering* menggunakan sebuah matrik tambahan yang di sebut *mask*. Ukuran matrik *mask* sama besar dengan matrik piksel yaitu $N \times N$. Didalam *mask* ini intinya disimpan jenis operasi yang akan dilakukan terhadap matrik piksel, akan tetapi tidak semua filter *spatial filtering* menggunakan *mask* untuk menyimpan operasinya.

Sobel operator diterapkan dalam dua buah *mask*, yaitu sebagai berikut :

-1	0	1
-2	0	2
-1	0	1

Mask (a) Vertikal

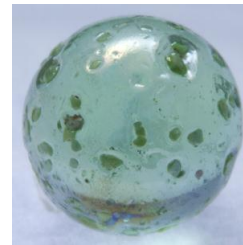
-1	-2	-1
0	0	0
1	2	1

Mask (b)
Horizontal

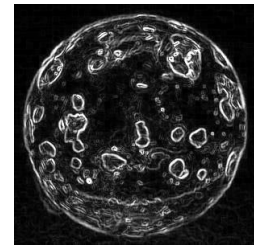
Gambar 2.5 Mask Sobel 3x3

Mask pertama yaitu mask (a) digunakan untuk menghitung siselish titik pada sisi vertikal sehingga dihasilkan titik penelusuran arah vertikal. Mask kedua yaitu mask (b) digunakan untuk menghitung selisih titik pada sisi horizontal sehingga dihasilkan titik hasil penelusuran arah horizontal.

Bila *edge* yang ditemukan merupakan sekumpulan piksel signifikan yang membentuk objek image, maka warna piksel tersebut akan di pertegas kembali, artinya piksel ini akan di perbesar intensitasnya sehingga warna edge ini akan tampak jelas. Keadaan edge yang demikian nantinya akan memperlihatkan suatu objek dalam image.



(a)



(b)

Gambar 2.6 (a) Citra Awal, (b) Citra Sobel

Deteksi Obyek

Deteksi Obyek merupakan langkah dalam mendeteksi kumpulan titik-titik piksel yang memiliki warna berbeda (lebih terang atau lebih gelap) dari latar belakang dan menyatukan dalam suatu region (Putra, 2010).

Dalam deteksi obyek metode yang akan digunakan adalah dengan menggunakan metode deteksi blob. Algoritma yang digunakan dalam deteksi obyek ini adalah algoritma Penandaan komponen terhubung (*Connected Components Labeling*)

Penandaan komponen terhubung merupakan langkah memeriksa suatu citra dan mengelompokkan setiap pixel ke dalam suatu komponen terhubung menurut aturan keterhubungan (4, 8 atau *m-connectivity*) (Putra, 2010). Komponen yang tidak saling terhubung (*disjoin*) pada suatu citra akan diberikan tanda (*label*) berbeda.

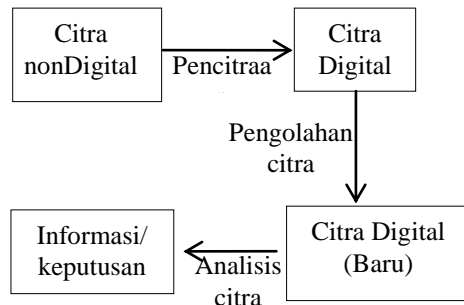
Penandaan komponen terhubung dengan memeriksa piksel per piksel dari suatu citra yaitu dari kiri ke kanan dan dari atas ke bawah untuk mengidentifikasi area piksel terhubung yaitu suatu area dari pixel berbatasan yang memiliki nilai intensitas yang sama atau nilai intensitas berada dalam suatu himpunan V (pada citra biner $V = \{1\}$, pada citra keabuan V disesuaikan dengan kebutuhan). Penandaan komponen terhubung dapat dilakukan pada citra biner maupun citra keabuan.

III. Analisa dan Perancangan Sistem

Program pendeteksian mutiara laut akan di bagi menjadi dua proses yaitu deteksi tepi dan deteksi objek.

3.1. Analisa Citra

Analisis citra merupakan proses menganalisa citra sehingga menghasilkan informasi untuk menetapkan keputusan.



Gambar 3.1 Proses Analisis Citra

3.2. Analisa Deteksi Tepi Sobel

Sobel *edge detection* merupakan metode deteksi tepi yang termasuk dalam *Gradient edge detector*. Pendeteksian batas (*edge*) dari sebuah gambar dilakukan dengan melakukan operasi matriks sobel terhadap gambar yang akan dilakukan pendeteksian gambar. Matriks sobel merupakan matriks yang berukuran 3x3 yang telah dijelaskan pada gambar 2.5 dengan koefisien yang telah ditentukan. Berikut akan dijelaskan perhitungan matriks sobel dengan contoh citra berupa matriks.

3	4	5	2	7
2	1	6	4	2
3	5	7	2	4
4	2	5	7	1
2	5	1	6	9

Gambar 3.2 Citra Awal

Proses Konvolusi Citra antara matriks citra awal dengan matriks sobel 3x3 (Gambar 2.5).

$$S_x = 3x(-1) + 2x(-1) + 3x(-1) + 2x(1) + 6x(2) + 7x(1) = 11$$

$$S_y = 3x(1) + 4x(2) + 2x(1) + 3x(-1) + 5x(-2) + 7x(-1) = -7$$

$$\begin{aligned} \text{Maka } M &= |S_x| + |S_y| \\ &= 11 + (-7) = 18 \end{aligned}$$

*	*	*	*	*
*	18	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*

Gambar 3.3 Citra Hasil Konvolusi

Hasil akhir filter sobel adalah ditemukannya beberapa piksel dengan intensitas

yang lebih besar atau tajam dan juga ukuran tepi objek yang jauh lebih besar dari ukuran sebelumnya. Keadaan ini dikarenakan titik-titik yang lebih dekat dengan titik tengah (terperiksa) diberi nilai yang lebih dominan dalam perhitungan. Pada gambar 3.3, terbukti pada awal intensitas piksel image mempunyai range intensitas (nomor warna) antara 0 sampai 4, setelah dilakukan proses filtering maka terjadi pergeseran intensitas antara 2 sampai 20. Bila piksel-piksel ini terseleksi dengan menggunakan ketentuan seperti operasi *Thresholding*, maka setiap piksel kemungkinan hanya mempunyai dua warna dominan yaitu warna hitam dan warna putih. Warna hitam diibaratkan sebagai background permukaan image dan warna putih memunculkan piksel-piksel signifikan tersebut.

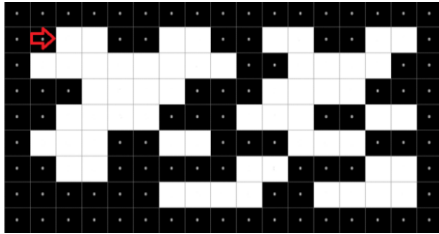
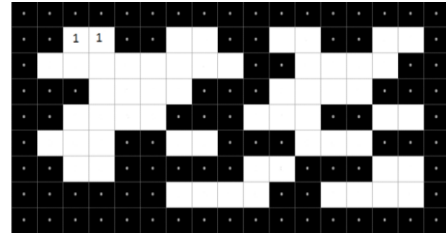
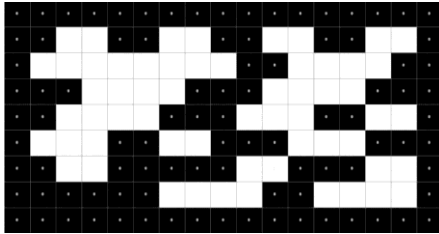
Edge atau garis di tepi objek terlihat lebih terang dari sebelumnya. Warna grayscale merupakan perpaduan warna dari dua warna dominan yaitu perpaduan antara warna maksimum dan warna minimum. Perpaduan warna yang dimaksud disebut sebagai warna medium atau setengah terang atau warna menengah. Bila image menggunakan perpaduan antara warna hitam dan putih, maka warna yang demikian dikenal dengan sebutan warna medium grey atau *greyscale*. Dengan demikian setiap piksel yang dihasilkan akan disesuaikan dengan set warna medium. Set warna ini dimulai dari warna hitam sebagai warna minimum dan naik secara perlahan-lahan menjadi lebih terang dari sebelumnya sampai pada warna maksimal yaitu berwarna.

3.3. Analisa Deteksi Objek

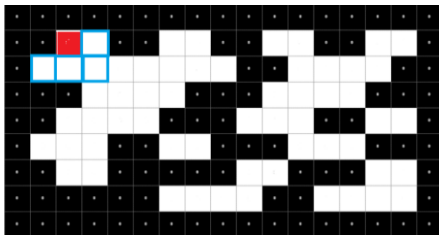
Suatu piksel atau kumpulan piksel yang berhubungan dengan piksel lainnya disebut dengan komponen terhubung (*connected component*), untuk membedakan kelompok piksel yang terhubung dilakukan pemberian label secara unik (Budisanjaya & Kunci, 2013). Proses ekstraksi komponen terhubung menghasilkan objek baru dimana kelompok piksel tersebut terhubung dengan diberikan nilai interger secara berurutan, misalnya latar belakang (*background*) memiliki nilai 0, piksel objek pertama diberikan nilai 1, piksel objek berikutnya yang berbeda dari piksel tetangganya di beri nilai 2 dan seterusnya.

Algoritma Penandaan Komponen Terhubung

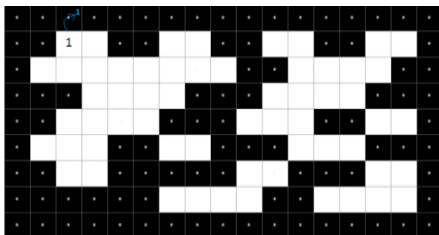
- Periksa (scan) citra dengan bergerak sepanjang baris sampai menemukan piksel p (nilai p berada dalam himpunan V).



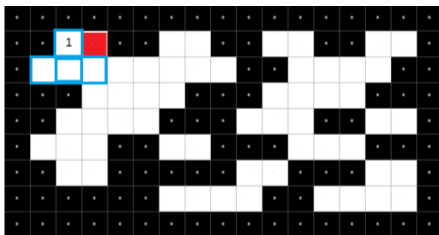
- Bila p sudah ditemukan maka periksa nilai piksel tetangga dari p , yaitu piksel di atas dan di kiri dari p .



- Bila kedua piksel tetangga bernilai 0 maka berilah tanda (label) baru pada p .

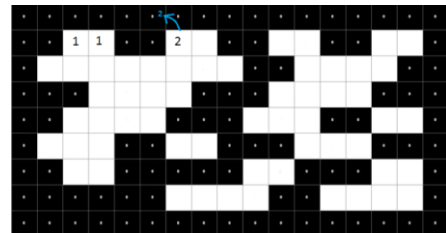
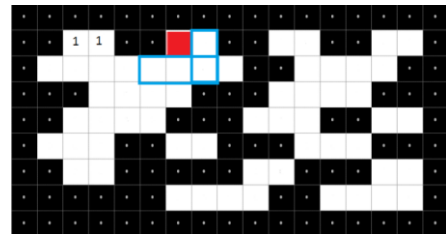


- Jika hanya satu saja dari piksel tetangga tersebut bernilai 1 maka berilah tanda dari piksel tetangga tersebut pada p .

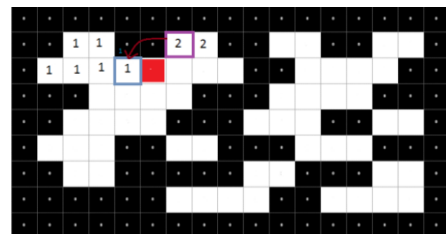
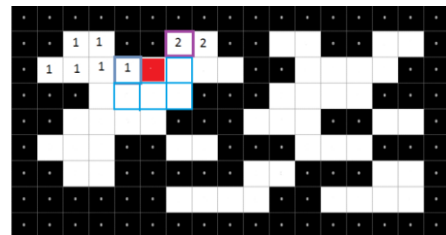


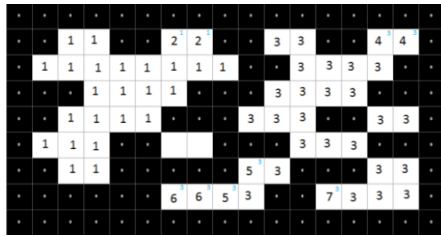
- Bila kedua piksel tetangga bernilai 1 dan memiliki tanda sama maka berilah tanda dari piksel tetangga tersebut pada p .

- Bila kedua piksel tetangga bernilai 1 dan memiliki tanda berbeda maka berilah tanda dari salah satu piksel tetangga tersebut pada p dan buat catatan bahwa kedua tanda yang berbeda tersebut adalah ekuivalen.



- Bila kedua piksel tetangga masing-masing bernilai 1 dan 2 maka ditetapkan piksel p menjadi label terkecil dari piksel tetangga.





Gambar 3.4 Hasil Akhir Penandaan komponen Terhubung

3.4. Analisa Mutiara

Pada mutiara yang akan dianalisa ada beberapa kriteria yang harus diperhatikan. (Autore, 2010) menjelaskan bebapa kriteria grade mutiara yang dilihat dari surface spotnya. Diantaranya :

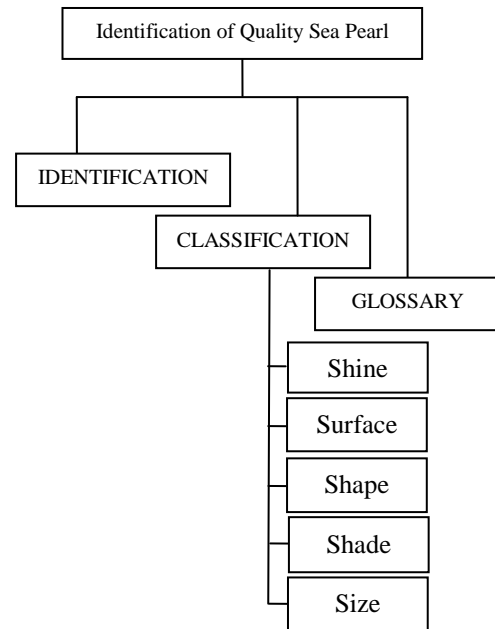
- Grade A : Permukaan mutiara bebas dari cacat atau tidak ada noda yang menempel di permukaan mutiara.
- Grade B1 : Terlihat satu hingga empat noda di permukaan mutiara.
- Grade B2 : Terlihat 30% noda/ cacat pada Permukaan Mutiara.
- Grade C1 : Terlihat 50% noda/ cacat pada Permukaan Mutiara.
- Grade C2 : noda hampir terlihat di seluruh permukaan mutiara.
- Grade D : Tingkat paling rendah, noda hampir terlihat di seluruh permukaan mutiara dan memiliki mantel yang tipis.

3.5. Perancangan Sistem

Pada perancangan system ini akan dibahas mengenai alur kerja pembuatan aplikasi desktop *Identification of Quality Sea Pearl* dengan menggunakan metode sobel sebagai deteksi tepi atau kontur dan metode Penandaan komponen terhubung (*Connected Components Labeling*) sebagai pendeteksi objek. Untuk lebih jelasnya, seluruh proses kerja sistem akan digambarkan menggunakan *flowchart* mulai dari inputan data yang dibutuhkan sampai tahap akhir.

3.5.1. Skema Aplikasi

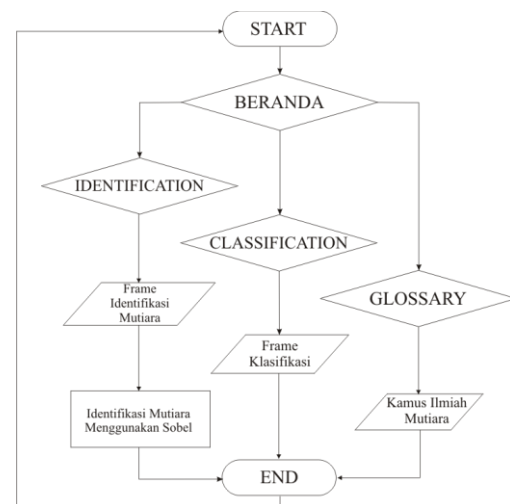
Pada skema aplikasi *Identification of Quality Sea Pearl* ada beberapa fitur yang disediakan (a). fitur identification yaitu fitur yang digunakan untuk mendeteksi grade dari citra mutiara yang diinputkan. (b). fitur classification, fitur ini berisikan informasi seputar klasifikasi mutiara, diantaranya Surface, Shape, Shade, Shine dan Size. (c). fitur glossary berisi tentang bahasa ilmiah seputar mutiara yang sering digunakan.



Gambar 3.5 Skema Aplikasi

3.5.2. Perancangan Flowchart

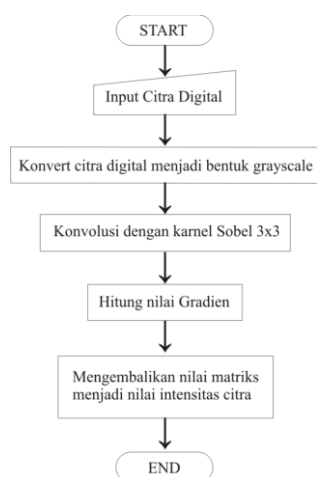
Pada Gambar 3.6 flowchart alur system dari *Identification of Quality Sea Pearl* ini dimulai dari halaman awal (branda) dan diteruskan ke beberapa pilihan yang disediakan diantaranya Identification, Classification, dan Glossary. Ketika User memilih Identification maka system akan masuk ke halaman identification yang berisikan proses untuk mendeteksi mutiara yang telah tercapture. Untuk menambah wawasan seputar mutiara laut user dapat memasuki halamana classification dan glossary yang berisikan informasi seputar mutiara laut.



Gambar 3.6 Flowchart Alur Sistem

• Deteksi Sobel

pada gambar 3.7 menjelaskan alur dari deteksi sobel yang dimulai dari menginputkan citra yang akan dideteksi dan selanjutnya citra yang masih berupa citra RGB akan di ubah ke citra *greyscale* dengan cara yang sudah di jelaskan pada poin 2.1 persamaan (1) yaitu dengan cara mencari nilai rata-rata *greyscale* dari setiap layer R, G dan B, selanjutnya citra hasil *greyscale* akan di konvolusikan dengan Mask sobel 3 x 3 seperti yang sudah digambarkan pada poin 2 gambar 2.5. setelah proses konvolusi selesai dan mendapatkan gradient dari proses deteksi sobel langkah selanjutnya yaitu mengembalikan nilai matriks menjadi nilai intensitas citra.



Gambar 3.7 Flowchart Deteksi Sobel

• Penandaan Komponen Terhubung (Connected Components Labeling)



Gambar 3.8 Flowchart Connected Components Labeling

Flowchart pada gambar 3.8 ini menjelaskan tentang proses terjadinya deteksi objek pada citra biner. citra biner dipakai dalam proses deteksi ini dikarenakan ada beberapa alasan yaitu untuk mengidentifikasi keberadaan objek, yang direpresentasikan sebagai daerah (region) di dalam citra. Setelah citra biner diinputkan langkah selanjutnya yaitu mengidentifikasi piksel citra apakah itu piksel background yang di tandai dengan angka 0 atau bukan. Jika piksel yang diidentifikasi bukan piksel background maka proses selanjutnya yaitu mengecek piksel tetangga jika piksel tetangga belum diberi label maka piksel yang diidentifikasi akan diberi label baru sedangkan jika piksel tetangga sudah diberi label maka proses selanjutnya yaitu mengecek label induk yang berada dalam daftar jika label induk belum ada maka akan ditambah ke daftar baru dan proses akhir yaitu mencetak hasil label.

IV. Implementasi dan Pengujian Sistem

Implementasi system pada bagian ini mengacu pada analisis dan perancangan system, sehingga pada bagian ini akan dijelaskan implementasi pada proses tersebut. Secara umum program ini merupakan aplikasi desktop untuk menganalisa tingkat surface mutiara laut. Aplikasi ini berisi fitur Detection untuk mendeteksi kualitas mutiara yang telah dicapture, fitur selanjutnya yaitu Classification, fitur ini memberikan informasi dan beberapa contoh citra mutiara berdasarkan surface, shine, shade, shape dan size, dan fitur yang terakhir yaitu Glossary, fitur ini berisikan informasi kata-kata ilmiah yang digunakan dalam budidaya mutiara laut.

4.1. Implementasi Kode Program Sobel

```

// --- kernel 3x3 Sobel Gx ---
float[][] kernelGx = {{+1,0,-1},
                      {+2,0,-2},
                      {+1,0,-1}};

// --- kernel 3x3 Sobel y ---
float[][] kernelGy = {{+1,+2,+1},
                      {0,0,0},
                      {-1,-2,-1}};

//---- mengisi Gx kernel ----
for (int i=0; i < kernelSize; i++) {
    for (int j=0; j < kernelSize; j++) {
        value=coeffNormIn * kernelGx[i][j] / (kernelSize*kernelSize);
        opencv_core_cvtSet2D(matrix2D, i, j, opencv_core_cvScalarAll(value));
    } // fin j
} //fin i
opencv_imgproc_cvFilter2D(Trans16S3C, Trans16S3C1, matrix2D, opencv_core_cvPoint (-1,-1));

//---- kernel mengisi Gy ----
for (int i=0; i < kernelSize; i++) {
    for (int j=0; j < kernelSize; j++) {
        value=coeffNormIn * kernelGy[i][j] / (kernelSize*kernelSize);
        opencv_core_cvtSet2D(matrix2D, i, j, opencv_core_cvScalarAll(value));
    } // fin j
} //fin i
opencv_imgproc_cvFilter2D(Trans16S3C, Trans16S3C2, matrix2D, opencv_core_cvPoint (-1,-1));
  
```

Gambar 4.1 Perhitungan Deteksi Tepi Pada Library OpenCV

Gambar di atas menjelaskan kode deteksi tepi atau kontur yang berada pada library OpenCv yang berisi perhitungan kernel sobel.

Setelah mengimplementasikan kode karnel sobel diatas selanjutnya yaitu mengisi kernel-kernel sobel dengan mengkonvolusi matriks citra dengan kernel sobel. untuk pemanggilan library sobel pada java di gambarkan pada gambar 4.1

```
//--- Operasi Pada Gambar ---
opencv.sobel();
opencv.gray();
image(opencv.getBuffer(), 0, 0);
```

Gambar 4.2 Kode Pemanggilan Deteksi Tepi/Kontur Pada Library OpenCV

4.2. Implementasi Kode Program Blob

Pada gambar di bawah ini menjelaskan kode pemanggilan deteksi objek yang ada pada library openCV.

```
blobsArray = opencv.blobs(opencv.area()/2500, opencv.area()/2, 1000, true, 5000, true);
Detections.grade = blobsArray.length;
println("jumlah = "+blobsArray.length);
```

Gambar 4.3 Kode pemanggilan Deteksi Blobs Pada Library OpenCV

Fungsi library blob :

```
blobsArray = blobs(minArea, maxArea, maxBlob, findHoles, maxVertices, debug);
```

Fungsi parameter :

- minArea : daerah minimal Blob
- maxArea: maksimum daerah Blobs
- maxBlob: jumlah maksimum Blob yang dihitung
- findHoles: menentukan jika diperhitungkan atau tidak di dalamnya - benar = Cari internal dan eksternal kontur
- maxVertices: maksimum jumlah vertex
- Debug: untuk menampilkan rincian item terdeteksi

4.3. Implementasi User Interface

Implementasi user interface pada aplikasi Identification of Quality Sea Pearl terdiri dari beberapa menu diantaranya tampilan awal (Home), tampilan menu Detection, tampilan menu Classification, tampilan menu Glossary.

Tampilan Menu Detection

Tampilan menu *Detection* merupakan halaman yang ditampilkan jika user ingin mengidentifikasi grade dari mutiara laut. Adapun

tampilan menu Detection seperti yang ditunjukkan pada Gambar 4.5.



Gambar 4.5 Tampilan Menu Detection

Pada halaman menu *Detection* ini ada beberapa button yang disediakan di antara : (1). Button Browse, button ini digunakan untuk mencari citra mutiara laut yang telah dicapture. (2). Button Clear, button ini digunakan untuk menghapus citra yang telah ditampilkan dan ingin menampilkan citra baru. (3). Button Back, digunakan untuk kembali ke menu *Beranda*. (4). Button Start, button ini digunakan untuk mulai memproses/ mendeteksi grade dari mutiara yang telah dipilih dan memunculkan hasilnya pada bagian Grade.

4.4. Pengujian Sistem Sobel

Pada pengujian deteksi tepi/ kontur ini pengujian dibagi dalam beberapa tahap yaitu sebagai berikut :

1. Menetapkan Parameter Pengujian

Parameter yang digunakan dalam pengujian system ini sebagai berikut:

a. Kestabilan system

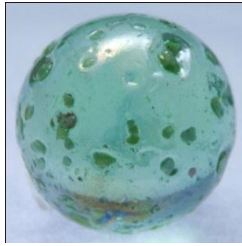
Parameter ini digunakan untuk menguji apakah system masih mengalami *error* pada saat dieksekusi atau pada saat melakukan proses pendeteksian garis tepi dan kontur.

b. Ketepatan hasil

Parameter ini digunakan untuk menguji apakah system telah dapat bekerja seperti apa yang diharapkan dalam perancangan.

2. Menyiapkan Perangkat Pengujian

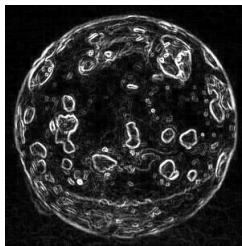
Dalam tahap ini, telah disiapkan citra hasil dari pengcapture-an dan menghasilkan citra digital yang mempunyai dimensi 300 x 300 dengan format JPG. Adapun tampilan dari salah satu citra digital yang dijadikan perangkat pengujian ini seperti terlihat pada gambar 4.16



Gambar 4.8 Citra Digital Pengujian dengan Format JPG

3. Pengujian Segmentasi Metode Sobel

Dalam tahap ini, dilakukan pengujian terhadap system dalam melakukan segmentasi citra digital dengan metode sobel terhadap citra digital yang telah diinputkan. Pada tahap ini, akan diuji parameter kestabilan system, serta ketepatan hasil yang diharapkan.

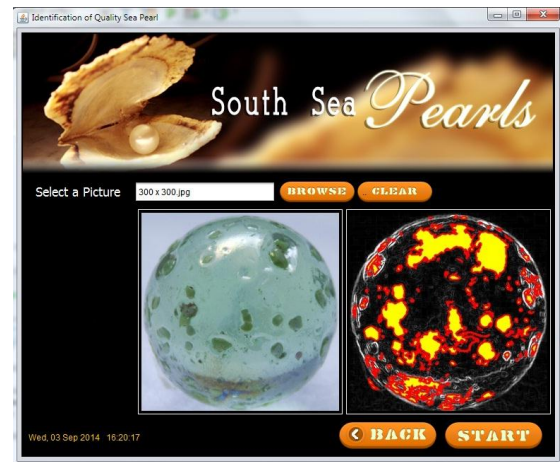


Gambar 4.9 Citra Hasil Pengujian

Pada pengujian ini terlihat system telah dapat melakukan segmentasi terhadap citra digital dengan menggunakan metode *Sobel*. Berdasarkan tampilan pada Gambar 4.9, garis tepi citra dan kontur citra digital yang disegmentasi telah dapat dilakukan dan sesuai dengan yang diharapkan, sehingga diambil kesimpulan bahwa pada pengujian sobel ini tidak memiliki masalah dalam melakukan segmentasi.

4.5. Pengujian Deteksi Objek

Dalam tahap ini dilakukan perhitungan terhadap noda yang ada pada permukaan mutiara laut yang sudah dilakukan proses deteksi tepi/ deteksi kontur sebelumnya. Citra yang akan digunakan pada tahap ini adalah citra hasil dari segmentasi sobel dan selanjutnya akan dilakukan proses perhitungan deteksi objek berupa noda yang terlihat pada permukaan mutiara. Adapun tampilan dari hasil pendeteksian objek seperti terlihat pada Gambar 4.17



Gambar 4.10 Hasil Deteksi Objek



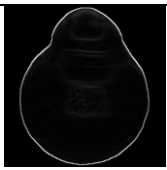
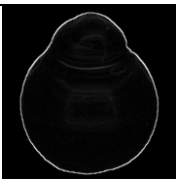

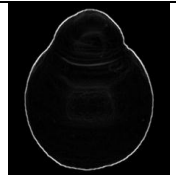
Pada pengujian deteksi objek di atas terlihat bahwa ada beberapa objek kecil yang tidak terdeteksi dengan sempurna, jadi kesimpulan yang dapat diambil yaitu pada proses deteksi objek system sudah dapat melakukan pendeteksian objek sebesar 75%.

4.6. Pengujian Perhitungan Grade untuk Kualitas Mutiara Laut

Pada tahap ini akan dilakukan perhitungan terhadap noda-noda yang ada pada mutiara laut. Perhitungan ini dilakukan untuk mengetahui tingkat kualitas mutiara pada mutiara laut tersebut.

Grade A	: Jika permukaan mutiara tidak terdeteksi adanya objek/ noda.
Grade B1	: Jika permukaan mutiara terdeteksi adanya 1 sampai 4 objek/ noda.
Grade B2	: Jika terdeteksi adanya 5 sampai 30% objek/ noda pada permukaan mutiara.
Grade C1	: Jika terdeteksi adanya 30% sampai 50% objek/ noda pada permukaan mutiara.
Grade C2	: Jika terdeteksi adanya 50% sampai 70% objek/ noda pada permukaan mutiara.
Grade D	: Jika terdeteksi adanya objek/ noda lebih dari 70% pada permukaan mutiara.



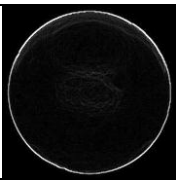
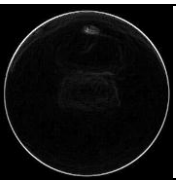
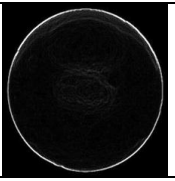
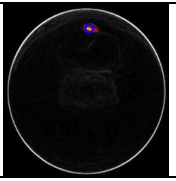
Untuk menghitung persentase digunakan cara menghitung luas lingkaran dari mutiara yang tercapture dengan menggunakan jari-jari (r) = 3 dan $\pi = 3.14$ **Error! Reference source not found.** *Luas Lingkaran = πr^2* **Error! Reference source not found.** Setelah didapatkan hasil, maka dikali dengan persen yang ditentukan yaitu untuk B2 = 30%, C1 = 50%, C2 = 70%.

Pengujian 1	
Citra Awal Front	Citra Awal Back
	
Citra Hasil Sobel Front	Citra Hasil Sobel Back
	
Citra Hasil Blob Front	Citra Hasil Blob Back
	

Tabel 4.1 Hasil Pengujian Citra Mutiara dengan Kualitas A

Keterangan Pengujian 1					
Citra	Grade PT.AU TORE	Jumlah Noda/Spot	Jumlah Noda Terdeteksi	Jumlah Noda Tidak Terdeteksi	Hasil Grade System
Front	A	0	0	0	A
Back		0	0	0	



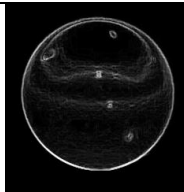
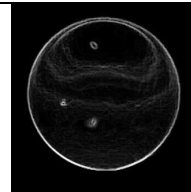
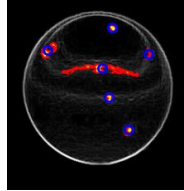
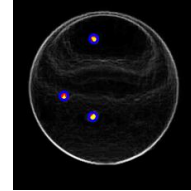
Tabel 4.2 Keterangan Hasil Pengujian

Pengujian 2	
Citra Awal Front	Citra Awal Back
	
Citra Hasil Sobel Front	Citra Hasil Sobel Back
	
Citra Hasil Blob Front	Citra Hasil Blob Back
	

Tabel 4.3 Hasil Pengujian Citra Mutiara dengan Kualitas B1

Keterangan Pengujian 2					
Citra	Grade PT.AU TORE	Jumlah Noda/Spot	Jumlah Noda Terdeteksi	Jumlah Noda Tidak Terdeteksi	Hasil Grade System
Front	B1	0	0	0	B1
Back		1	1	0	



Tabel 4.4 Keterangan Hasil Pengujian 2

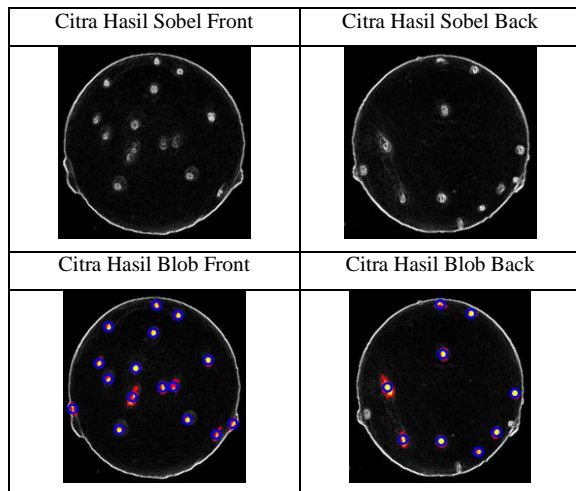
Pengujian 3	
Citra Awal Front	Citra Awal Back
	
Citra Hasil Sobel Front	Citra Hasil Sobel Back
	
Citra Hasil Blob Front	Citra Hasil Blob Back
	

Tabel 4.5 Hasil Pengujian Citra Mutiara dengan Kualitas C1

Keterangan Pengujian 3					
Citra	Grade PT.AUT ORE	Jumlah Noda/Spot	Jumlah Noda Terdeteksi	Jumlah Noda Tidak Terdeteksi	Hasil Grade System
Front	C1	6	6	0	C1
Back		3	3	0	

Tabel 4.6 Keterangan Hasil Pengujian 3

Pengujian 4	
Citra Awal Front	Citra Awal Back
	



Tabel 4.7 Hasil Pengujian Citra Mutiara dengan Kualitas D

Keterangan Pengujian 4					
Citra	Grade PT.AU TORE	Jumlah Noda/ Spot	Jumlah Noda Terdeteksi	Jumlah Noda Tidak Terdeteksi	Hasil Grade System
Front	D	19	16	3	D
Back		15	9	6	

Tabel 4.8 Keterangan Hasil Pengujian 4

Dari hasil pengujian di atas, system telah mampu mendeteksi tepi atau kontur dari citra mutiara laut yang telah tercapture dan system juga mampu mendeteksi dan menghitung jumlah spot/ noda yang ada pada mutiara laut tersebut akan tetapi ada beberapa spot/ noda yang belum mampu terdeteksi oleh system.

V. Kesimpulan dan Saran

5.1. Kesimpulan

Setelah menyelesaikan perancangan system, implementasi dan melakukan analisa system terhadap hasil yang didapat dari penelitian yang dilakukan terdapat beberapa kesimpulan sebagai berikut :

1. Deteksi tepi menggunakan operator sobel menghasilkan titik tepi yang halus sehingga menghasilkan output yang diinginkan.
2. Dengan memanfaatkan metode *connected component labeling (Blobs)* Spot/ Noda pada mutiara dapat terdeteksi sehingga memudahkan untuk mengkategorikan kualitas surface mutiara tersebut.
3. Pendeteksian noda pada mutiara laut dengan menggunakan metode deteksi tepi atau kontur dan deteksi objek yang disini menggunakan penandaan komponen terhubung (*Connected Components Labeling*) telah mencapai tingkat keberhasilan sebesar 75%, dengan melakukan pengujian sebanyak 4 kali dengan kualitas

surface mutiara yang berbeda-beda. Adapun kegagalan dalam segmentasi citra yaitu pada pendeteksi objek yang kurang sempurna, dikarenakan peralatan yang kurang mendukung/ memadai, dan system deteksi objek yang tidak dapat mendeteksi noda yang sangat kecil.

5.2. Saran

Adapun saran untuk pengembangan selanjutnya agar dapat meningkatkan aplikasi ini adalah sebagai berikut :

1. Memahami grading pada mutiara sehingga mampu membuat system yang lebih akurat.
2. Kualitas mutiara yang dideteksi tidak hanya pada surface tetapi mampu untuk mendeteksi tingkat kualitas yang lainnya seperti shine, shape dan size.
3. Memilih algoritma yang tepat untuk setiap kriteria kualitas yang akan dideteksi seperti menggunakan algoritma histogram untuk mendeteksi kualitas shine pada mutiara laut, menggunakan pengenalan pola untuk mendeteksi kualitas shape pada mutiara laut.
4. Menggunakan peralatan yang tepat dalam mendeteksi mutiara laut seperti menggunakan lensa makro agar spot/ noda yang kecil dapat terlihat jelas, menggunakan kamera x-ray yang dapat mendeteksi bintik depan dan belakang dengan melakukan sekali pengcapture-an.

Daftar Pustaka

- Anggraeni, K. (2007). Histogram Citra. *Ilmu Komputer*, 7.
- Autore. (2010). *Pearl-Culture*. Retrieved from www.pearlautore.com.au
- Budisanjaya, I. P., & Kunci, K. (2013). Perangkat Lunak Pengolahan Citra Untuk Segmentasi dan Cropping Daun Sawi Hijau. *Prosiding Conference on Smart-Green Technology in Electrical and Information Systems*, (November), 14–15.
- Hambali, Y. A. (2011). Aplikasi Area Process Berbasis C# menggunakan Visual Studio. *Ilmu Komputer*, 14.
- Hardiyanto, I., Purwananto, Y., Kom, S., Kom, M., & Soelaiman, R. (2012). Implementasi Segmentasi Citra dengan Menggunakan Metode Generalized Fuzzy C- Means Clustering Algorithm with Improved Fuzzy Partitions. *Jurnal Teknik Pomits*, 1(1), 1–5.
- Putra, D. (2010). *Pengolahan Citra Digital*. (Westriningsih, Ed.) (p. 420). Yogyakarta: C.V ANDI OFFSET. Retrieved from http://books.google.co.id/books?id=NectMutqXJAC&printsec=frontcover&hl=id&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

- Wijaya, M. C., & Tjiharjadi, S. (2009). Mencari Nilai Threshold yang Tepat Untuk Perancangan Pendeteksi Kanker Trofoblas. *Seminar Nasional Aplikasi Teknologi Informasi 2009*, (SNATI), 3–8.
- Yunus, M. (n.d.). Perbandingan Metode Edge Detection Untuk Proses Segmentasi Citra Digital. *Jurnal Teknologi Informasi*, 3(2), 146–160.