

# Analisa dan Perbandingan Algoritma Otsu Thresholding dengan Algoritma Region Growing Pada Segmentasi Citra Digital

Debi Razabni Erika Medinah, Sinar Sinurat

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email : <sup>1</sup>debirazabni1712@gmail.com, <sup>2</sup>sinarsssinar@gmail.com

Submitted: 09/07/2020; Accepted: 01/11/2020; Published: 27/11/2020

**Abstrak**—Segmentasi merupakan proses yang penting dalam pengolahan citra yang bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Segmentasi citra adalah pemisahan objek yang satu dengan objek yang lain dalam suatu citra atau antara objek dengan latar yang terdapat dalam sebuah citra. Dengan proses segmentasi tersebut, masing-masing objek pada citra dapat diambil secara individu sehingga dapat digunakan sebagai input. Tujuan segmentasi citra yaitu memisahkan wilayah objek dengan wilayah lain berdasarkan kesamaan atribut citra agar mudah di analisis dalam rangka mengenali objek untuk suatu keperluan. Jenis operasi ini berkaitan erat dengan pengenalan pola. Secara umum proses segmentasi dibagi menjadi tiga bagian berdasarkan klasifikasi, berdasarkan tepi dan berdasarkan daerah. Proses segmentasi pada citra digital dilakukan dengan deteksi tepi untuk menyediakan jalur target di sepanjang tepi (edge) objek citra. Algoritma yang digunakan yaitu algoritma otsu thresholding dan algoritma region growing dimana algoritma otsu thresholding dapat membagi histogram citra keabuan ke dalam dua daerah yang berbeda secara otomatis. Sedangkan algoritma region growing merupakan sebuah prosedur dimana sekumpulan piksel atau subregion hingga region yang lebih besar berdasarkan kriteria yang sudah didefinisikan.

**Kata Kunci:** Segmentasi, Citra, Algoritma Otsu Thresholding, Algoritma Region Growing

**Abstract**—Segmentation is an important process in image processing which aims to split an image into several segments with certain criteria. Image segmentation is the separation of one object with other objects in an image or between objects with a background contained in an image. With this segmentation process, each object in the image can be taken individually so that it can be used as input. The purpose of image segmentation is to separate the object area from other regions based on similarity of image attributes so that it is easy to analyze in order to recognize objects for a purpose. This type of operation is closely related to pattern recognition. In general, the segmentation process is divided into three parts based on classification, by edge and by region. The segmentation process in digital images is done with edge detection to provide a target path along the edge (edge) of the image object. The algorithm used is the otsu thresholding algorithm and the region growing algorithm where the otsu thresholding algorithm can divide the gray image histogram into two different regions automatically. While the region growing algorithm is a procedure in which a set of pixels or subregions to a larger region is based on defined criteria.

**Keywords:** Segmentation, Image, Otsu Thresholding Algorithm, Region Growing Algorithm

## 1. PENDAHULUAN

Pemanfaatan pengolahan citra dalam era yang hampir serba digital ini sangat dibutuhkan. Analisa komposisi suatu citra dapat berdampak terhadap perubahan kehidupan. Pengolahan citra adalah suatu metode atau teknik yang dapat digunakan untuk memproses citra atau gambar dengan jalan memanipulasinya menjadi data gambar yang diinginkan untuk mendapatkan informasi tertentu.

Segmentasi merupakan proses yang penting dalam pengolahan citra yang bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Segmentasi citra adalah pemisahan objek yang satu dengan objek yang lain dalam suatu citra atau antara objek dengan latar yang terdapat dalam sebuah citra. Dengan proses segmentasi tersebut, masing-masing objek pada citra dapat diambil secara individu sehingga dapat digunakan sebagai input. Jenis operasi ini berkaitan erat dengan pengenalan pola.

Proses segmentasi pada citra digital dilakukan dengan deteksi tepi untuk menyediakan jalur target di sepanjang tepi (edge) objek citra. Edge pada gambar adalah area pada tepi objek image dengan intensitas contrast yang kuat, sehingga lompatan intensitas dari satu pixel kepixel lainnya. Edge mendeteksi gambar dengan mengurangi secara signifikan jumlah data dan menyaring informasi yang tidak berguna filtering, dan juga menjaga property penting yang diperlukan pada gambar dalam hal ini yang mendukung penampakan tepi-tepi gambar. Deteksi tepi (Edge Detection) pada suatu citra suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah untuk menandai bagian yang menjadi detail citra dan untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra. Suatu titik (x,y) dikatakan sebagai tepi (edge) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya.

Ada banyak metode untuk pendeteksian tepi ini, tetapi metode untuk melokasikan edge ini merupakan karakteristik dari kategori “gradient filter” dan termasuk untuk pencarian segmentasi citra. Lokasi pixel dinyatakan atau di deklarasikan jika nilai dari gradient melewati ambang batasan (x,y). Teknik differensial yang dikembangkan, yaitu differensial pada arah horizontal dan differensial pada arah vertikal, dengan ditambahkan proses konversi biner setelah dilakukan differensial. Teknik konversi biner yang disarankan adalah konversi biner dengan meratakan distribusi warna hitam dan putih. Metode yang paling sederhana untuk proses segmentasi yaitu dengan image thresholding atau pengambangan citra. Pada penelitian ini digunakan algoritma otsu thresholding di mana algoritma thresholding dapat membagi histogram citra keabuan ke dalam dua daerah

yang berbeda secara otomatis sedangkan metode region growing adalah segmentasi berdasarkan analisa wilayah (region-based approach). Pada pendekatan ini segmentasi citra sangat dipengaruhi oleh kriteria yang diterapkan untuk menilai kesamaan sebuah wilayah. Untuk itu nilai ambang batas (threshold value) untuk memberikan keputusan pada algoritma untuk memasukkan sebuah piksel ke dalam sebuah wilayah menjadi sangat penting.

Algoritma Otsu thresholding merupakan salah satu metode untuk segmentasi citra digital dengan menggunakan nilai ambang secara otomatis, yakni mengubah citra digital warna abu-abu menjadi hitam putih berdasarkan perbandingan nilai ambang dengan nilai warna piksel citra digital. Metode Otsu thresholding diperkenalkan pertama kali oleh Nobuyuki Otsu, dalam jurnal ilmiahnya yang berjudul "A Threshold Selection Method from Grayscale Histogram" pada tahun 1979 [2].

Algoritma Region Growing merupakan sebuah prosedur dimana sekumpulan piksel atau subregion hingga region yang lebih besar berdasarkan kriteria yang sudah didefinisikan pendekatan dasar, misalkan sekumpulan titik awal dari sini region tumbuh dengan menambahkan pada masing-masing piksel [1].

## 2. METODOLOGI PENELITIAN

### 2.1 Segmentasi Citra

Segmentasi citra akan membagi-bagi suatu citra menjadi daerah-daerah atau obyek-obyek yang dimilikinya. Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek-objek yang terkandung di dalam citra atau membagi citra kedalam beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut". Dalam konteks citra digital daerah hasil segmentasi tersebut merupakan kelompok piksel yang bertetangga atau berhubungan.

### 2.2 Pengolahan Citra

Pengolahan citra merupakan proses untuk menghasilkan citra sesuai dengan keinginan atau kualitasnya menjadi lebih baik. Inputannya adalah citra dan keluarannya juga citra tapi dengan kualitas lebih baik dari pada citra masukan. Misal citra warnanya kurang tajam, kabur (blurring) dan mengandung noise (misal bintik-bintik putih) sehingga perlu ada pemrosesan untuk memperbaiki citra karena citra tersebut menjadi sulit diinterpretasikan karena informasi yang disampaikan menjadi berkurang.

### 2.3 Algoritma Otsu Thresholding

Algoritma Otsu Thresholding merupakan salah satu metode untuk segmentasi citra digital dengan menggunakan nilai ambang secara otomatis, yakni mengubah citra digital warna abu-abu menjadi hitam putih berdasarkan perbandingan nilai ambang dengan nilai warna piksel citra digital. Algoritma Otsu thresholding di perkenalkan pertama kali oleh Nobuyuki Otsu, dalam jurnal ilmiahnya yang berjudul "A Threshold Selection Method from Grayscale Histogram" pada tahun 1979. Untuk mendapatkan nilai threshold ada perhitungan yang harus dilakukan. Langkah awal yang harus dilakukan adalah membuat histogram. Dari histogram dapat diketahui jumlah piksel untuk setiap tingkat keabuan. Tingkat keabuan citra dinyatakan dengan  $i$  sampai dengan  $L$ . Level ke  $i$  dimulai dari 1, yaitu piksel 0. Untuk  $L$ , maksimal level adalah 256 dengan piksel bernilai 255. Nilai ambang yang akan dicari dari suatu citra grayscale dinyatakan dengan  $k$ . Nilai  $k$  berkisar antara 0 sampai dengan  $L-1$ , dengan nilai  $L=256$  (simbol histogram adalah  $P_i$ ). Jadi probabilitas setiap piksel pada level ke  $i$  dinyatakan dengan persamaan 1. [10]

$$p_i = \frac{n_i}{N} \quad (1)$$

Keterangan :

$P_i$  = Probabilitas piksel ke- $i$

$n_i$  = Jumlah piksel dengan tingkat keabuan  $i$

$N$  = Total jumlah piksel pada citra

### 2.4 Algoritma Region Growing

Algoritma region growing merupakan sebuah prosedur di mana sekumpulan piksel atau subregion hingga region yang lebih besar berdasarkan kriteria yang sudah didefinisikan pendekatan dasar, misalkan sekumpulan titik awal dari sini region tumbuh dengan menambahkan pada masing-masing piksel tetangga yang memiliki property yang sama dengan benih tersebut. Descriptor bisa mendapatkan hasil yang salah jika keterhubungan dan informasi piksel yang bersebelahan tidak digunakan dalam proses region growing [1].

Pendekatan segmentasi dengan algoritma region growing berdasarkan analisa delapan piksel berdampingan (8 connectivity) yaitu : Jika  $f(x,y)$  merupakan piksel masukan dari sebuah citra;  $S(x,y)$  adalah piksel permulaan (seed pixel) dan  $Q$  merupakan kriteria penilaian, maka proses region growing akan berjalan sebagai berikut: [11]

1. Piksel permulaan (seed pixel)  $S(x,y)$  temukan 8 piksel lain yang berdampingan.

2. Analisa ke delapan piksel berdampingan berdasarkan kriteria Q yang telah ditetapkan. Jika ada piksel berdampingan yang memenuhi kriteria Q, maka berikan label 1 pada piksel tersebut. Sebaliknya piksel yang tidak memenuhi kriteria diberi label 0.
3. Simpan piksel yang berlabel 1 pada citra keluaran sebagai sebuah wilayah.
4. Pada citra f(x,y), temukan piksel yang berlabel 0 dan lakukan analisa seperti pada poin 2 dan 3.
5. Lakukan proses tersebut sampai tidak ada lagi piksel berlabel 0 pada citra f(x,y).

Konversi gambar berwarna ke dalam hitam putih dapat dilakukan dengan persamaan 1. Dengan R adalah saluran merah. G saluran hijau dan B saluran biru.

$$\text{Grayscale Image} = 0.3R + 0.59G + 0.11B \quad (2)$$

Penentuan kriteria Q merupakan nilai batas maksimum antara seed pixel dengan piksel lain yang dianalisa. Nilai batas maksimum ini selanjutnya disebut dengan Treshold value (T). Nilai T ditentukan dengan persamaan 2.

$$T = k\sigma \quad (3)$$

Di mana k adalah nilai konstanta dan  $\sigma$  standar deviasi citra masukan. Standar deviasi digunakan untuk mengukur homogenitas piksel-piksel dalam citra. Semakin homogen nilai piksel-piksel pada sebuah citra, maka nilai  $\sigma$  menjadi lebih kecil sebaliknya menjadi semakin besar apabila nilai pikselnya semakin heterogen. Nilai  $\sigma$  dapat diperoleh dari persamaan 3.

$$\sigma = \sqrt{\frac{1}{rc} \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} (I_{ij} - \bar{I})^2} \quad (4)$$

Di mana r adalah jumlah baris piksel, c jumlah kolom piksel, i dan j merupakan posisi piksel dalam baris dan kolom (i,j), I intensitas piksel dan  $\bar{I}$  adalah rata-rata intensitas nilai piksel. Secara umum nilai konstanta k adalah 1 sehingga  $T = \sigma$ . Nilai T nantinya dapat dirubah dengan merubah nilai konstanta k. Setelah penentuan nilai T, selanjutnya ditentukan seed pixel pertama. Agar lebih sistematis dan seragam untuk semua citra masukan, seed pixel yang dipilih adalah piksel yang memiliki nilai yang sama yang terbanyak, sehingga dapat didefinisikan seperti persamaan 4. Di mana s1 adalah seed pixel pertama. In adalah piksel dengan nilai n.

$$S1 = [\sum I_n] = \max \quad (5)$$

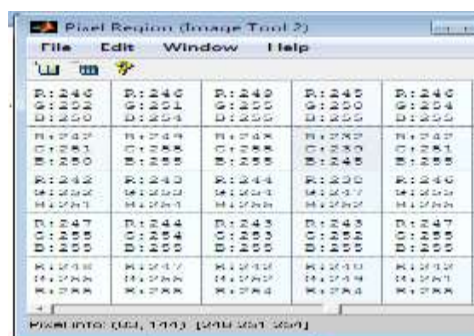
Setelah piksel permulaan ditentukan, maka algoritma akan melakukan proses iterasi. Apabila sebuah piksel memenuhi persamaan 5, maka piksel tersebut akan dikelompokkan ke dalam satu kelompok atau area yang sama.

$$(|I_{ij} - \bar{R}| \leq T) = \text{TRUE} \quad (6)$$

### 3. HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai analisis segmentasi citra digital dengan menggunakan algoritma otsu thresholding dan algoritma region growing. Sistem identifikasi kuku dibagi menjadi 2 tahap yaitu pengambilan data kuku dan pengenalan bentuk kuku. Cara kerja sistem ini dibagi menjadi 3 tahapan, yakni tahapan pencapturan citra kuku tangan, segmentasi citra kuku, perbandingan citra kuku antara algoritma otsu thresholding dan algoritma region growing runtuk mendapatkan berapa jumlah sampel kuku yang baik menggunakan otsu thresholding dan region growing.

Segmentasi yaitu mereduksi citra menjadi objek atau region, misalnya memisahkan objek-objek yang berbeda dengan mengekstraksi batas-batas objek (boundary). Segmentasi merupakan suatu proses untuk mendapatkan area atau obyek yang diinginkan pada suatu citra dengan memisahkan antara area atau objek dari latar belakangnya. Proses segmentasi memisahkan objek dari latar belakangnya pada tepi yang terhubung, jika tepi tidak terhubung maka objek tidak tersegmentasi. Citra hasil segmentasi berbentuk citra biner terbagi atas dua bagian, yaitu bagian hitam dan bagian putih, angka 0 menyatakan warna latar belakang (background) dan angka 1 menyatakan warna tinta atau objek atau dalam bentuk angka 0 untuk hitam dan angka 1 untuk warna putih.



Gambar 1. RGB Citra Awal

### 3.1 Penerapan *Algoritma Region Growing*

Pada proses ini, pertama-tama, citra inputan (citra RGB) akan dikonversi ke dalam citra gray, citra gray sendiri merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian RED=GREEN=BLUE. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing r, g dan b menjadi citra grayscale dengan nilai s, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai r, g dan b. Berikut ini nilai matriks R, G dan B dari citra kuku.

Pada Citra warna bisa diubah menjadi citra grayscale dengan cara menghitung rata-rata elemen warna Red, Green, Blue. Secara matematis rumus perhitungannya adalah sebagai berikut.

$$f_o(X, Y) = \frac{f_i^R(X, Y) + f_i^G(X, Y) + f_i^B(X, Y)}{3}$$

Pada gambar citra asli di atas dapat di lihat nilai pixel R,G,B seperti pada gambar 3.2. Dari hasil pixel R,G,B akan diubah menjadi pixel grayscale Perhitungan fungsi negasi dilakukan seperti berikut. Setiap titik yang terletak di posisi (X,Y), nilai-nilai komponen Red, Green dan Blue ditambahkan, kemudian hasilnya dibagi 3. Berikut ini contoh perhitungannya:

- |   |   |
|---|---|
| 1. $f_o = \frac{246+252+250}{3} = 249$  | 14. $f_o = \frac{238+247+252}{3} = 246$ |
| 2. $f_o = \frac{246+251+254}{3} = 250$  | 15. $f_o = \frac{246+255+255}{3} = 252$ |
| 3. $f_o = \frac{249+255+255}{3} = 253$  | 16. $f_o = \frac{247+255+255}{3} = 252$ |
| 4. $f_o = \frac{245+250+255}{3} = 250$  | 17. $f_o = \frac{244+254+255}{3} = 251$ |
| 5. $f_o = \frac{246+254+255}{3} = 252$  | 18. $f_o = \frac{243+253+255}{3} = 250$ |
| 6. $f_o = \frac{242+251+250}{3} = 248$  | 19. $f_o = \frac{243+252+255}{3} = 250$ |
| 7. $f_o = \frac{247+255+255}{3} = 253$  | 20. $f_o = \frac{247+255+255}{3} = 252$ |
| 8. $f_o = \frac{248+255+255}{3} = 253$  | 21. $f_o = \frac{248+255+255}{3} = 253$ |
| 9. $f_o = \frac{232+239+245}{3} = 239$  | 22. $f_o = \frac{247+255+255}{3} = 252$ |
| 10. $f_o = \frac{242+251+255}{3} = 249$ | 23. $f_o = \frac{242+252+254}{3} = 249$ |
| 11. $f_o = \frac{242+252+251}{3} = 248$ | 24. $f_o = \frac{240+249+254}{3} = 248$ |
| 12. $f_o = \frac{243+253+254}{3} = 250$ | 25. $f_o = \frac{242+251+255}{3} = 249$ |
| 13. $f_o = \frac{244+244+255}{3} = 251$ |   |

Maka hasil yang didapat dari pixel warna ke grayscale seperti pada tabel 3.1 berikut.

**Tabel 1.** Hasil Citra Grayscale

249	250	253	250	252
248	253	253	239	249
248	250	251	246	252
252	251	250	250	252
253	252	249	248	249

Matriks bobot 5x5 merupakan matriks simeteris yang akan dilakukan perkalian terhadap matriks A yaitu setiap pixel (posisi tengah) yang ada pada citra beserta 24 pixel tetangga yang mengelilingi. Matriks akan di normalisasi dengan cara membagi setiap isi matriks dengan jumlah seluruh isi matriks yaitu 4090.

Untuk melakukan segmentasi dengan algoritma region growing, digunakan gradien G(x,y) yang merupakan sebuah vektor yang terdiri dari dua unsur yaitu Gx dan Gy. Deteksi tepi dilakukan dengan cara membaca setiap pixel pada citra dengan cara membaca dari pixel paling kiri atas (timur utara) dan bergerak ke pixel paling kanan bawah (barat selatan). Oleh karena itu, untuk membantu penelusuran tepi, gradien Gx dan Gy masing-masing dihitung dengan matriks Mask 3x3. Operator ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Nilai C konstanta bernilai dua, sehingga terbentuk matriks operator sobel seperti tabel 2. di bawah ini:

**Tabel 2.** Matrix 3x3

$S_x =$	-1	0	1	$S_y =$	1	2	1
	-2	0	2		0	0	0
	-1	0	1		-1	-2	-1

Sehingga besar gradient dapat di hitung dengan menggunakan persamaan:

$$S_x = (p_1 + cp_2 + p_3) - (p_7 + cp_6 + p_5)$$

$$S_y = (p_1 + cp_8 + p_7) - (p_3 + cp_4 + p_5)$$

$$|S| = |S_x| + |S_y|$$

Berikut adalah hasil perhitungan perkalian matrix penggunaan operator sobel untuk mengetahui segmentasi sebuah gambar citra dengan matrix 5x5. Konvolusi pertama dilakukan terhadap pixel yang bernilai 1 (titik pusat mask).

**Tabel 3.** Citra Grayscale Matrix 5x5

249	250	253	240	252
248	253	253	239	249
248	250	251	246	252
252	251	250	250	252
253	252	249	248	249

1. Nilai 16 pada citra hasil konvolusi pertama didapatkan dengan perhitungan :  
 $S_x = (249)(-1) + (248)(-2) + (248)(-1) + (253)(1) + (253)(2) + (251)(1) = 17$   
 $S_y = (249)(1) + (250)(2) + (253)(1) + (248)(-1) + (250)(-2) + (251)(-1) = 3$
2. Konvolusi kedua dilakukan terhadap pixel yang bernilai 10 (titik pusat mask)  
 $S_x = (250)(-1) + (253)(-2) + (250)(-1) + (240)(1) + (239)(2) + (246)(1) = 42$   
 $S_y = (250)(1) + (253)(2) + (240)(1) + (250)(-1) + (251)(-2) + (246)(-1) = 2$
3. Konvolusi ketiga dilakukan terhadap pixel yang bernilai 10 (titik pusat mask)  
 $S_x = (253)(-1) + (253)(-2) + (251)(-1) + (252)(1) + (249)(2) + (252)(1) = 8$   
 $S_y = (253)(1) + (240)(2) + (252)(1) + (251)(-1) + (246)(-2) + (252)(-1) = 10$
4. Konvolusi keempat dilakukan terhadap pixel yang bernilai 18 (titik pusat mask)  
 $S_x = (248)(-1) + (248)(-2) + (252)(-1) + (253)(1) + (251)(2) + (250)(1) = 9$   
 $S_y = (248)(1) + (253)(2) + (253)(1) + (252)(-1) + (251)(-2) + (250)(-1) = 3$
5. Konvolusi kelima dilakukan terhadap pixel yang bernilai 12 (titik pusatmask)  
 $S_x = (253)(-1) + (250)(-2) + (251)(-1) + (239)(1) + (246)(2) + (250)(1) = 23$   
 $S_y = (253)(1) + (253)(2) + (239)(1) + (251)(-1) + (250)(-2) + (250)(-1) = 3$
6. Konvolusi keenam dilakukan terhadap pixel yang bernilai 4 (titik pusat mask)  
 $S_x = (253)(-1) + (251)(-2) + (250)(-1) + (249)(1) + (252)(2) + (252)(1) = 0$   
 $S_y = (253)(1) + (239)(2) + (249)(1) + (250)(-1) + (250)(-2) + (252)(-1) = 22$
7. Konvolusi ketujuh dilakukan terhadap pixel yang bernilai 24 (titik pusat mask)  
 $S_x = (248)(-1) + (252)(-2) + (253)(-1) + (251)(1) + (250)(2) + (249)(1) = 5$   
 $S_y = (248)(1) + (250)(2) + (251)(1) + (253)(-1) + (252)(-2) + (249)(-1) = 7$
8. Konvolusi kedelapan dilakukan terhadap pixel yang bernilai 12 (titik pusat mask)  
 $S_x = (250)(-1) + (251)(-2) + (252)(-1) + (246)(1) + (250)(2) + (248)(1) = 10$   
 $S_y = (250)(1) + (251)(2) + (246)(1) + (252)(-1) + (249)(-2) + (248)(-1) = 0$
9. Konvolusi kesembilan dilakukan terhadap pixel yang bernilai 4 (titik pusat mask)  
 $S_x = (251)(-1) + (250)(-2) + (249)(-1) + (252)(1) + (252)(2) + (249)(1) = 5$   
 $S_y = (251)(1) + (246)(2) + (252)(1) + (249)(-1) + (248)(-2) + (249)(-1) = 1$

**Tabel 4.** Hasil Konvolusi Matrix 3x3

20	44	18
12	26	22
12	10	6

Maka hasil yang didapat dari perhitungan matrix untuk mengetahui segmentasi pada gambar menggunakan algoritma region growing, hasil yang didapat pada perkalian matrix 3x3 dengan citra grayscale matrix 5x5 pada tabel 4.

### 3.2 Penerapan Algoritma Otsu Thresholding

Algoritma otsu bertujuan untuk membagi histogram citra keabuan ke dalam dua daerah yang berbeda secara otomatis untuk memasukkan nilai ambang. Pendekatan yang dilakukan oleh metode otsu yaitu dengan analisis diskriminan yang menentukan suatu variabel sehingga dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Adapun tahapan yang dilakukan metode otsu thresholding antara lain :

1. Tentukan nilai matriks R (Red), G (Green), dan B (Blue) dari citra masukan. Setelah nilai matrik R, G dan B ditentukan maka lakukan perhitungan grayscale, maka hasil yang didapat dari pixel warna ke grayscale seperti pada tabel dibawah ini.

**Tabel 5.** Hasil Citra Grayscale

249	250	253	250	252
248	253	253	239	249
248	250	251	246	252
252	251	250	250	252

253	252	249	248	249
-----	-----	-----	-----	-----

- Setelah proses grayscale selesai dilanjutkan dengan proses top-bottom hat. Proses top-bottom hat digunakan untuk mempertajam kontras citra antara bagian dan latar belakang.
- Citra hasil grayscale dan struktur elemen 5 x 5 berbentuk square (persegi) menjadi input untuk proses top-bottom hat.

**Tabel 5.** Struktur Elemen 5x5 Berbentuk Square

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Tanda yang berwarna abu-abu menunjukkan pusat struktur elemen.

- Lakukan proses di lasi

Berikut perhitungan proses di lasi dilakukan dengan cara :

Struktur elemen 5x5 dengan posisi tengah diletakkan diatas nilai pertama citra hasil grayscale, masing-masing nilai struktur elemen dikalikan dengan nilai citra grayscale pada titik yang sama.

$$G(0,0) = \{(249).(1), (250).(1), (253).(1), (248).(1), (253).(1), (253).(1), (248).(1), (250).(1), (251).(1), (0).(1), (0).(1), (0).(1), (0).(1), (0).(1), (0).(1), (0).(1), (0).(1), (0).(1), (0).(1), (0).(1), (0).(1), (0).(1)\}$$

$$G(0,0) = \{(249), (250), (253), (248), (253), (253), (248), (250), (251), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0)\}$$

- Cari nilai maksimum dari perkalian

$$G(0,0) = \text{Max} \{f(0,0), f(0,0+1), f(0,0+2), f(0+1,0), f(0+1,0+1), f(0,+1,0+2), f(0+2,0), f(0+2,0+1), f(0+2,0+2), f(0,0-1), f(0+1,0-1), f(0+2,0-1), f(0,0-2), f(0+1,0-2), f(0+2,0-2), f(0-1,0), f(0-1,0+1), f(0-2,0), f(0-2,0+1), f(0-2,0+2), f(0-2,0+1), f(0-2,0+2), f(0-2,0-1), f(0-2,0-2)\}$$

$$G(0,0) = \text{Max} \{f(0,0), f(0,1), f(0,2), f(1,0), f(1,1), f(1,2), f(2,0), f(2,1), f(2,2), f(0,-1), f(1,-1), f(2,-1), f(0,-2), f(1,-2), f(2,-2), f(-1,0), f(-1,1), f(-1,2), f(-1,-1), f(-1,-2), f(-2,0), f(-2,1), f(-2,2), f(-2,-1), f(-2,-2)\}$$

$$G(0,0) = \text{Max} \{(249), (250), (253), (248), (253), (253), (248), (250), (251), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0)\}$$

$$G(0,0) = 248$$

- Cari nilai minimum dari perkalian

$$G(0,0) = \text{Min} \{f(0,0), f(0,0+1), f(0,0+2), f(0+1,0), f(0+1,0+1), f(0,+1,0+2), f(0+2,0), f(0+2,0+1), f(0+2,0+2), f(0,0-1), f(0+1,0-1), f(0+2,0-1), f(0,0-2), f(0+1,0-2), f(0+2,0-2), f(0-1,0), f(0-1,0+1), f(0-2,0), f(0-2,0+1), f(0-2,0+2), f(0-2,0+1), f(0-2,0+2), f(0-2,0-1), f(0-2,0-2)\}$$

$$G(0,0) = \text{Min} \{f(0,0), f(0,1), f(0,2), f(1,0), f(1,1), f(1,2), f(2,0), f(2,1), f(2,2), f(0,-1), f(1,-1), f(2,-1), f(0,-2), f(1,-2), f(2,-2), f(-1,0), f(-1,1), f(-1,2), f(-1,-1), f(-1,-2), f(-2,0), f(-2,1), f(-2,2), f(-2,-1), f(-2,-2)\}$$

$$G(0,0) = \text{Min} \{(249), (250), (253), (248), (253), (253), (248), (250), (251), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0), (0)\}$$

$$G(0,0) = 250$$

- Hitung nilai probabilitas piksel dengan menggunakan persamaan 1.

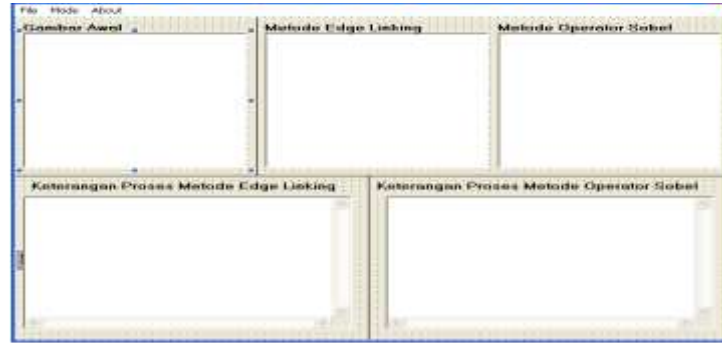
$$P_i = \frac{n_i}{N}$$

$$P_{252} = \frac{250}{25} = 10$$

### 3.3 Implementasi Program

Setelah melakukan implementasi terhadap rancangan yang diperoleh sebelumnya, diperoleh hasil berupa sebuah perangkat lunak yang dapat melakukan segmentasi terhadap citra digital dalam format BMP dan JPG dengan algoritma otsu dan region growing. Adapun hasil yang diperoleh tersebut berupa tampilan hasil dan hasil pengujian sistem.

Tampilan Form Utama merupakan form yang ditampilkan pertama kali pada saat perangkat lunak dijalankan. Form Utama ini berisi tampilan menu dan picture box yang akan menampilkan gambar awal yang dipilih pengguna. Adapun tampilan Form Utama seperti ditunjukkan pada Gambar 2.



Gambar 1. Tampilan Form Utama

Setelah mendapatkan hasil tampilan perangkat lunak, tahap selanjutnya dilakukan pengujian terhadap sistem tersebut. Adapun metode pengujian sistem yang dilakukan adalah metode statis (static technique) dimana pengujian dibagi dalam beberapa tahapan, sebagai berikut :

1. Menetapkan Parameter Pengujian

Adapun parameter pengujian yang digunakan dalam pengujian sistem ini adalah sebagai berikut :

- a. Kestabilan Sistem Parameter ini digunakan untuk menguji apakah sistem masih mengalami error pada saat dieksekusi atau pada saat melakukan proses pendeteksian garis tepi.
- b. Ketepatan Hasil Parameter ini digunakan untuk menguji apakah sistem telah dapat bekerja seperti apa yang diharapkan dalam perancangan.

2. Menyiapkan Perangkat Pengujian

Dalam tahap ini, telah disiapkan sebuah citra digital yang mempunyai dimensi 2048 x 1536 dengan format JPG, citra digital yang mempunyai dimensi 640 x 480 dengan format JPG, dan citra digital dimensi 640 x 480 dengan format BMP . Adapun tampilan dari salah satu citra digital yang dijadikan perangkat pengujian ini seperti terlihat pada Gambar 3.



Gambar 3. Citra Digital Pengujian Dengan Format BMP

3. Melakukan Pengujian Segmentasi

Dalam tahap ini, dilakukan pengujian terhadap kemampuan sistem dalam melakukan segmentasi citra digital dalam mode Edge Linking terhadap citra digital yang disiapkan. Pada tahap ini, akan diuji parameter kestabilan sistem, serta ketepatan hasil yang diharapkan. Adapun hasil pengujian yang diperoleh seperti terlihat pada Gambar 4.



Gambar 4. Pengujian Segmentasi

#### **4. KESIMPULAN**

Setelah melakukan penelitian, maka dapat ditarik kesimpulan bahwa algoritma otsu thresholding dapat mendeteksi suatu citra digital walaupun memiliki tingkat noise yang tinggi, sedangkan operator region growing tidak mampu mengolah citra digital yang memiliki tingkat noise yang tinggi. Bahkan, noise yang ada pada citra didefinisikan sebagai tepi. Tepi yang dihasilkan oleh algoritma otsu thresholding berupa tepi tunggal, sehingga kualitas tepi yang dihasilkan lebih akurat dibandingkan kualitas tepi yang dihasilkan operator region growing yang berupa tepi ganda dan tepi tunggal.

#### **REFERENCES**

- [1] Hermawati, Fajar Astuti. 2013. "Pengolahan Citra Digital Konsep dan Teori". Penerbit Andi, Yogyakarta.
- [2] Syafii, Slamet Imam, dkk, 2015, "Segmentasi Obyek Pada Citra Digital Menggunakan Metode Otsu Thresholding", Jurnal Informatika, Vol. 13, No.1, ISSN : 1411-0105.
- [3] Kadir, Abdul dan Adhi Susanto. 2013. "Teori dan Aplikasi pengolahan citra". Penerbit Andi, Yogyakarta.
- [4] <http://informatika.web.id/segmentasi-deteksi-warna.html>, diakses tanggal 7 Mei 2018
- [5] <http://informatika.web.id/pengertian-citra-digital.html>, diakses tanggal 7 Mei 2018
- [6] <http://informatika.web.id/pengertian-citra-digital.html>, diakses tanggal 7 Mei 2018
- [7] Wijaya, Marvin ch.2007. "Pengolahan Citra Digital Matlab", Penerbit Andi, Yogyakarta.
- [8] [http://id.wikipedia.org/wiki/Pengolahan\\_citra](http://id.wikipedia.org/wiki/Pengolahan_citra), diakses tanggal 7 Mei 2018. [9] [http://id.wikipedia.org/wiki/Pengolahan\\_citra](http://id.wikipedia.org/wiki/Pengolahan_citra), diakses tanggal 7 Mei 2018.
- [9] Syafii, Imam, Slamet, dkk, "Segmentasi Obyek Pada Citra Digital Menggunakan Metode Otsu Thresholding", Jurnal Informatika Vol.13. No.1. Mei 2015, ISSN 1411-0105.
- [10] Mulka, "Implementasi Algoritma Region Growing Untuk Segmentasi Retakan Bidang Batuan", Jurnal TECHSI Penelitian Teknik Informatika.
- [11] A.S. Rosa dan Shalahuddin. M, 2013, "Rekayasa Perangkat Lunak Terstruktur", Andi, Yogyakarta.
- [12] Aditya, Arif Primananda, 2013, "Dasar-Dasar Pemrograman Databaseekstop Dengan Visual Basic.Net2008", PT. Elex Media Komputindo, Jakarta.