

# Segmentasi Citra Digital Berbasis Clustering Menggunakan Deteksi Tepi Sobel

Torkis Nasution

Program Studi Teknik Informatika, STMIK-AMIK Riau

torkisnasution@stmik-amik-riau.ac.id

torkisnasution.com

## Abstrak

*Dalam proses analisis citra digital, segmentasi merupakan salah satu proses yang penting. Segmentasi digunakan untuk membagi citra ke dalam beberapa bagian atau region yang homogen berdasarkan kriteria kemiripan tertentu (region based segmentation). Salah satu metode yang dapat digunakan untuk melakukan segmentasi berbasis area (region based) adalah clustering. Dalam penelitian ini metoda clustering yang digunakan adalah deteksi tepi menggunakan metoda sobel. Metoda sobel, mendeteksi piksel dari sebuah gambar dengan menggunakan matrik sobel. Operasi matrik ini akan menghasilkan objek dalam citra digital yang sudah tersegmentasi. Pada proses pengujian akan dibuat aplikasi dengan bahasa pemrograman Microsoft Visual Basic. Langkah pertama dalam segmentasi objek pada ctra digital adalah meneteapkan citra warna, kemudian mengkonversi ke dalam citra biner, citra biner akan di filter dengan metoda low pass filter, selanjutnya citra akan di di deteksi tepi menggunakan metoda Sobel, objek yang di hasilkan sudah teridentifikasi melalui batas-batas tepi objek. Selanjutnya penajaman atas batas tersebut untuk menetapkan objek hasil segmentasi*

*Kata Kunci : deteksi tepi, sobel, citra biner, gray scale*

## Abstract

*In the process of digital image analysis, segmentation is one of the important processes. Segmentation is used to split the image into several parts or homogeneous regions based on its similarity (region based segmentation). One of the methods used to perform segmentation based area (region based) is the clustering. In this research, the clustering methods applied is edge detection uses Sobel Method. Sobel method detects pixels from an image using sobel Matrix. Matrix operation will result in a digital image of an object that is already segmented. In testing process, an application will be built with Microsoft Visual Basic. The first step in object segmentation of digital image is choosing color, then converting the image into binary, binary image will be filtered with low pass filter method, and next the image will be edge-detected by using Sobel method, objects that have been produced was already identified through the edges of an object, and then sharpening of the boundaries to define the result of the object segmentation.*

*Keywords : edge detection, sobel, binary image, gray scale*

## 1. Pendahuluan

Penggunaan citra digital sudah mulai menggeser penggunaan citra analog. Banyak

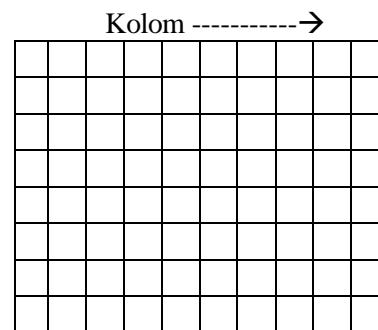
orang sudah mulai menyimpan gambar-gambar ke dalam bentuk digital. Hal ini disebabkan karena citra digital dapat disimpan dalam jangka waktu yang lama, tidak mudah rusak dan dapat diperbanyak dengan mudah sesuai dengan keinginan, cukup dengan meng-copy. Selain itu citra digital dapat dilihat dahulu sebelum di cetak dan dapat diubah, hal ini tidak dapat dilakukan pada citra analog.

Cukup banyak metoda dan algoritma yang tersedia untuk mengolah citra digital, antara lain thresholding, shapebased, region growing, dan clustering masing-masing metoda memiliki kelebihan dan kelemahan tergantung pada karakteristik dari citra yang akan di segmentasi. Satu diantara tujuan utama pengolah citra digital adalah untuk segmentasi objek, karena dengan segmentasi objek, pengolahan atau pemanipulasian dapat dilakukan lebih mudah dan lebih cepat. Segmentasi Objek adalah pemisahan objek yang satu dengan objek yang lain dalam suatu gambar. Secara umum, proses segmentasi dikenal sebagai *full segmentation* dan *partial segmentation*. *Full segmentation* adalah pemisahan suatu object secara individu dari *background* dan diberi ID (label) pada tiap-tiap segmen. *Partial segmentation* adalah pemisahan sejumlah data dari *background* dimana data yang disimpan adalah data yang dipisahkan saja untuk mempercepat proses. Mengenal objek secara otomatis dapat menggunakan deteksi tepi, dalam penelitian menggunakan metoda sobel. Keunggulan metoda ini, adalah kemampuannya dalam mempertajam tepi dari masing-masing objek, dengan demikian akan diperoleh hasil yang jelas.

Citra yang di segmentasi dengan metoda sobel adalah citra biner, hal ini akan lebih memperoleh hasil yang optimal. Untuk pengujian akan dilakukan pada objek yang mudah di kenali, yaitu benda-benda kubus, balok, bola dengan format yang berdiri sendiri, tidak ada overload masing-masing objek.

## 2. Citra Digital

Citra digital merupakan citra yang berbentuk array dua dimensi yang terdiri dari blok-blok kecil yang disebut piksel. Piksel merupakan elemen penyusun warna terkecil yang menyusun suatu citra. Citra dibentuk dari kota-kotak persegi yang teratur sehingga jarak horizontal dan vertikal antara piksel adalah sama pada seluruh bagian citra. Setiap piksel diwakili oleh bilangan bulat (integer) untuk menunjukkan lokasinya dalam bidang citra. Sebuah bilangan bulat juga digunakan untuk menunjukkan cahaya atau keadaan terang atau gelap piksel tersebut.



Gambar 1. Deskripsi piksel

di representasikan ke dalam matrik berukuran  $n \times m$  menjadi :

$$i = \begin{bmatrix} i(1,1) & i(1,2) & i(1,M) \\ i(2,1) & i(2,2) & i(2,M) \\ i(N,1) & i(N,2) & i(N,M) \end{bmatrix}$$

Untuk menunjukkan lokasi piksel, koordinat (x,y) berfungsi untuk menunjukkan posisi sudut kiri atas pada citra, indeks x bergerak ke kanan dan indeks y bergerak ke bawah. Koordinat (x-1,y-1) digunakan untuk menunjukkan posisi kanan bawah dalam citra berukuran  $m \times n$  piksel. Untuk menunjukkan tingkat pencahayaan suatu piksel, digunakan bilangan bulat yang besarnya 8 bit (1 byte) untuk setiap piksel, dengan lebar selang antara 0 – 255, di mana 0 untuk warna hitam, 255 untuk warna putih dan tingkat keabuan ditandai dengan nilai 0

-255. Berdasarkan sifat dari nilai terkuantisasinya, citra digital dapat diklasifikasikan sebagai berikut :

1. Citra biner, yaitu citra digital yang tiap pikselnya hanya memiliki dua kemungkinan nilai, yaitu 0 dan 1.
2. Citra crayscale, yaitu citra yang setiap piksel merupakan sampel tunggal, yaitu informasi intensitas. Citra jenis ini terbentuk hanya dari warna abu-abu pada tingkatan yang berbeda-beda, mulai dari warna hitam pada tingkat intensitas terendah hingga warna putih pada tingkat intensitas tertinggi. Citra ini disebut juga citra hitam putih atau citra monokromatik.
3. Citra warna, adalah citra digital yang setiap pikselnya mengandung informasi warna.. informasi warna ini biasanya dibentuk dari paling sedikit 3 sampel (saluran warna). Saluran warna yang umum dipakai dalam komputer adalah Red-Green-Blue (RGB), tetapi dalam konteks lain sering juga digunakan warna lain seperti Cyan-Magenta-Yellow-Black

### 2.1. Jenis Citra Digital

Jenis citra digital dapat dibedakan menjadi dua tipe yaitu :

- a. Tipe raster, merupakan suatu struktur dat yang berbentuk dalam persegi-persegi kecil (piksel), yang memiliki warna tersendiri. Raster dapat disimpan dalam suatu tipe data dengan berbagai macam format (PSD, JPG, GIF, TIF, BMP). Kelebihan dari tipe raster adalah dapat menampilkan suatu citra secara lebih realistik yang membutuhkan variasi warna yang kompleks. Dan kekurangannya adalah apabila ukurannya diperbesar maka ketajamannya akan hilang dan akan terlihat balok-balok piksel
- b. Tipe vector, adalah citra yang mengandung vektor atau titik-titik yang telah disusun dan terhubung untuk membentuk suatu objek. Kelebihannya adalah dapat dibentuk secara matematis, sehingga dapat diperbesar atau diperkecil tanpa mengurangi kualitas citra.

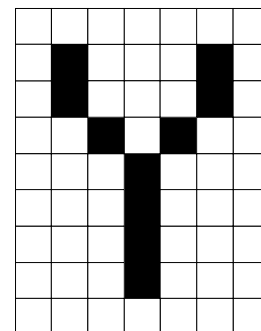
Dan kekurangannya adalah tidak dapat menampilkan citra secara realistis

### 2.2. Citra Biner

Citra Biner adalah citra yang hanya mempunyai dua nilai derajat keabuan : hitam dan putih. Citra biner banyak digunakan pada citra logo instansi (yang hanya memiliki warna hitam dan putih), citra kode batang (*bar code*), citra hasil pemindaian (*scanner*). Tujuan penggunaan citra biner untuk keperluan *pattern recognition* yang sederhana, pengenalan angka, pengenalan huruf, pengenalan objek. Untuk memperoleh hasil yang optimal di sarankan untuk mengubah citra gray scale menjadi citra biner, bukan dari citra warna menjadi citra biner. Hal ini di dasarkan pada proses penetapan derajat keabuan yang mengambil nilai tengah. Untuk citra dengan derajat keabuan 256, maka nilai tengah nya adalah 128 sehingga untuk mengubah menjadi citra biner dapat di tuliskan :

Jika  $x < 128$  maka  $x = 0$ , jika  $x \geq 128$  maka  $x = 255$ . Hasil dari konversi, pada citra biner akan memiliki karakteristik sebagai berikut :

1. Citra biner hanya mempunyai dua nilai derajat keabuan, yaitu hitam dan putih.
2. *Pixel-pixel* objek bernilai 1 dan *pixel-pixel* latar belakang bernilai 0.
3. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam.
4. Jadi pada citra biner, latar belakang berwarna putih, sedangkan objek berwarna hitam.



(a) Citra digital

1	1	1	1	1	1	1
1	0	1	1	1	0	1
1	0	1	1	1	0	1
1	1	0	1	0	1	1
1	1	1	0	1	1	1
1	1	1	0	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

(b) bentuk biner

Gambar 2. Deskripsi citra digital

Citra biner ukuran 9x7 piksel dan representasinya dalam data digital

- citra hasil scan text/buku
- citra hasil deteksi tepi
- citra hasil *threshold*

Citra biner memiliki keunggulan pada Kebutuhan memori kecil karena nilai derajat keabuan hanya mempunyai representasi 1 bit. Waktu pemrosesan lebih cepat dibandingkan dengan citra hitam putih. Konversi citra gray scale menjadi citra biner dilakukan untuk tujuan :

- Untuk mengidentifikasi keberadaan objek.
- Untuk lebih memfokuskan pada analisis bentuk morfologi.
- Untuk menampilkan citra pada piranti keluaran yang hanya mempunyai resolusi intensitas satu bit.
- Untuk menampilkan citra pada piranti keluaran yang hanya mempunyai resolusi intensitas satu bit.
- Mengkonversi citra yg telah ditingkatkan kualitas tepinya (*edge enhancement*) ke penggambaran garis-garis tepi.

Proses yang dilakukan untuk konversi citra gray scale menjadi citra biner adalah :

- Konversi dari citra hitam putih ke citra biner dilakukan dengan operasi pengambangan (*thresholding*).
- Operasi pengambangan mengelompokkan nilai derajat keabuan setiap pixel ke dalam 2 kelas, hitam dan putih.

- Pendekatan yg digunakan dalam operasi pengambangan adalah pengambangan secara global dan pengambangan secara lokal.

Proses pengambangan dapat melalui dua metoda yaitu : Pengambangan secara global (*global image thresholding*), dengan tahapan yang dilakukan adalah :

- setiap *pixel* di dalam citra dipetakan ke dua nilai, 1 atau 0 dengan fungsi pengambangan :

$$f_B(i, j) = \begin{cases} 1, & f_g(i, j) \leq T \\ 0, & \text{lainnya} \end{cases}$$

Dengan :  $f_g(i, j)$  = Citra hitam putih

$f_b(i, j)$  = Citra biner

$T$  = nilai ambang yg

dispesifikasi

- Dengan operasi pengambangan tsb, objek dibuat berwarna gelap (1 atau hitam), sedangkan latar belakang berwarna terang (0 atau putih).
- Jika nilai intensitas objek diketahui dalam selang  $[T_1, T_2]$ , maka kita dapat menggunakan fungsi pengambangan :

$$f_B(i, j) = \begin{cases} 1, & T_1 \leq f_g(i, j) \leq T_2 \\ 0, & \text{lainnya} \end{cases}$$

Pengambangan secara lokal adaptif (*locally adaptive image thresholding*), tahapan yang dilakukan adalah :

- Pengambangan secara lokal dilakukan terhadap daerah-daerah di dalam citra, dengan memecah citra menjadi bagian-bagian kecil, kemudian proses pengambangan dilakukan secara lokal.
- Dengan pengambangan secara lokal adaptif, secara subjektif citra biner yang dihasilkan terlihat lebih menyenangkan.

Proses pengambangan menghasilkan citra biner. Seringkali citra biner yang dihasilkan mengandung beberapa daerah yang dianggap sebagai gangguan. Biasanya daerah gangguan itu berukuran kecil. Penapis luas dapat digunakan untuk menghilangkan daerah gangguan tsb. Dengan cara menyatakan daerah di luar objek

dengan 0. Citra Biner umumnya dikodekan dengan metode *Run-length encoding* (RLE), yaitu panjang *run*, dimulai panjang *run* 1. Contoh : Misalkan citra binernya adalah sebagai berikut :

```
1 1 1 0 0 0 1 1 0 0 0 1 1 1 1 0 1 1 0 1 1 1
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
```

Hasil pengkodean adalah :

```
3, 3, 2, 3, 4, 1, 2, 1, 3
0, 4, 13, 1, 4
3, 13, 6
```

### 2.3. Citra Gray Scale

Dalam komputasi, suatu citra digital grayscale atau greyscale adalah suatu citra dimana nilai dari setiap pixel merupakan sample tunggal. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra grayscale berbeda dengan citra "hitam-putih", dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu "hitam" dan "putih" saja. Pada citra grayscale warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra grayscale seringkali merupakan perhitungan dari intensitas cahaya pada setiap pixel pada spektrum elektromagnetik single band.

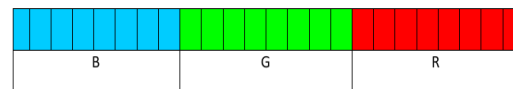
Citra grayscale disimpan dalam format 8 bit untuk setiap sample pixel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak. Pada aplikasi lain seperti pada aplikasi medical imaging dan remote sensing biasa juga digunakan format 10,12 maupun 16 bit.

### 3. Citra Warna

Suatu citra biasanya mengacu ke citra RGB. Sebenarnya bagaimana citra disimpan dan

dimanipulasi dalam komputer diturunkan dari teknologi televisi, yang pertama kali mengaplikasikannya untuk tampilan grafis komputer. Jika dilihat dengan kaca pembesar, tampilan monitor komputer akan terdiri dari sejumlah triplet titik warna merah (RED), hijau (GREEN) dan biru (BLUE). Tergantung pada pabrik monitornya untuk menentukan apakah titik tersebut merupakan titik bulat atau kotak kecil, tetapi akan selaluterdiri dari 3 triplet red, green dan blue.

Citra dalam komputer tidak lebih dari sekumpulan sejumlah triplet dimana setiap triplet terdiri atas variasi tingkat keterangan (brightness) dari elemen red, green dan blue. Representasinya dalam citra, triplet akan terdiri dari 3 angka yang mengatur intensitas dari Red (R), Green (G) dan Blue (Blue) dari suatu triplet. Setiap triplet akan merepresentasikan 1 pixel (picture element). Suatu triplet dengan nilai 67, 228 dan 180 berarti akan mengeset nilai R ke nilai 67, G ke nilai 228 dan B ke nilai 180. Angka-angka RGB ini yang seringkali disebut dengan color values. Pada format .bmp citra setiap pixel pada citra direpresentasikan dengan dengan 24 bit, 8 bit untuk R, 8 bit untuk G dan 8 bit untuk B, dengan pengaturan seperti pada gambar 1.



Gambar 3. Gambar true color

### 4. Segmentasi Citra Digital

Proses awal yang dilakukan dalam menganalisis objek di dalam citra biner adalah segmentasi objek. Proses segmentasi bertujuan mengelompokkan *pixel-pixel* objek menjadi wilayah (*region*) yang merepresentasikan objek. Pendekatan yang digunakan, adalah segmentasi berdasarkan batas wilayah (tepi dari objek). *Pixel-pixel* tepi ditelusuri sehingga rangkaian *pixel* yang menjadi batas (boundary) antara objek dengan latar belakang dapat diketahui secara keseluruhan, dengan cara kerja :

- a. Pada citra biner, batas antara objek dengan latar belakang terlihat jelas. *Pixel* objek berwarna hitam sedangkan *pixel* latar belakang berwarna putih. Pertemuan antara pixel hitam dan putih dimodelkan sebagai segmen garis. Penelusuran batas wilayah dianggap sebagai pembuatan rangkaian keputusan untuk bergerak lurus, belok kiri atau belok kanan. Metode pendeteksian batas wilayah yang lain adalah pendeteksian secara topologi.
- b. Pada metode topologi, setiap kelompok 4 *pixel* bertetangga diperiksa dan bila kelompok tersebut sama dengan salah satu bentuk pada gambar berikut, maka pada titik tengah dari kelompok *pixel* tersebut terdapat tepi.
- c. Wilayah (*region*) di dalam citra biner dapat direpresentasikan dalam beberapa cara. Salah satu cara yang populer adalah representasi wilayah dengan **pohon empatan** (*quadtree*).
- d. Setiap simpul di dalam pohon empatan merupakan salah satu dari tiga kategori : putih, hitam, dan abu-abu.
- e. Pohon empatan diperoleh dengan membagi citra secara rekursif. Wilayah di dalam citra di bagi menjadi empat upa wilayah yang berukuran sama.
- f. Untuk setiap upa wilayah, bila *pixel-pixel* di dalam wilayah tersebut semuanya hitam atau semuanya putih, maka proses pembagian dihentikan.
- g. Sebaliknya, bila *pixel-pixel* di dalam upa wilayah mengandung baik pixel hitam maupun pixel putih (kategori abu-abu), maka upa wilayah tersebut dibagi lagi menjadi empat bagian.
- h. Demikian seterusnya sampai diperoleh upa wilayah yang semua *pixel*-nya hitam atau semua *pixel*-nya putih.
- i. Proses pembagian tersebut digambarkan dengan pohon empatan.
- j. Dinamakan pohon empatan karena setiap simpul mempunyai tepat empat anak.
- k. Gambar berikut memperlihatkan contoh representasi wilayah dengan pohon empatan.

## 5. Segmentasi Citra Digital

Deteksi tepi Sobel merupakan metode edge detection yang termasuk dalam gradient edge detector. Piksel dari sebuah gambar yang akan dilakukan pendeteksian batas (edge) akan menjadi sebuah edge jika piksel tersebut melewati batas (threshold) tertentu. Sehingga apabila threshold telah ditetapkan maka nilai dari setiap piksel dapat di tentukan apakah piksel tersebut merupakan batas edge atau bukan. Pendeteksian batas (edge) dari sebuah gambar dilakukan dengan melakukan operasi matriks sobel terhadap gambar yang akan dilakukan pendeteksian gambar. Matrik sobel merupakan matrik yang berukuran 3 x 3 dengan koefisien yang telah ditentukan. Berikut ini disajikan gambar dari matriks sobel.

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

**G<sub>x</sub>**

+1	+2	+1
0	0	0
-1	-2	-21

**G<sub>y</sub>**

**Gambar 4. Matrik Sobel**

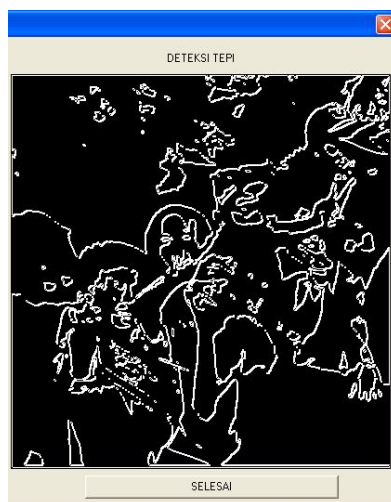
pada gambar dapat dilihat sepasang matrik sobel yang digunakan untuk mendeteksi batas dari sebuah gambar. Untuk melakukan deteksi tepi batas gambar (edge), dilakukan dengan melkakukan perkalian terhadap matrik secara terpisah. Matrik G<sub>x</sub>, seperti terlihat pada gambar, berfungsi untuk memperkirakan gradient pada arah sumbu x, sedangkan matrik G<sub>y</sub> berfungsi untuk memperkiarakan gradient pada arah sumbu y. berikut ini disajikan gambar dari hasil pengolahan dengan edge detection dengan menggunakan metode sobel.



(a) Citra True Color



(b) Citra Gray Scale



(c) Citra Deteksi Tepi

setelah melakukan operasi perkalian matrik sobel diatas dengan gambar, maka akan dapat dihasilkan gambar Citra Deteksi Tepi.

Peninjauan pengaturan pixel di sekitar pixelnya (x,y) adalah :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x, y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Operator sobel adalah magnitudo dari gradien yang dihitung dengan :

$$M = \sqrt{s_x^2 + s_y^2}$$

Dimana dalam hal ini turunan parsial dihitung dengan :

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$s_y = (a_0 + ca_1 + a_{22}) - (a_6 + ca_5 + a_4)$$

Dengan konstanta  $c = 2$ . Dalam bentuk mask,  $s_y$  dan  $s_x$  dapat dinyatakan sebagai :

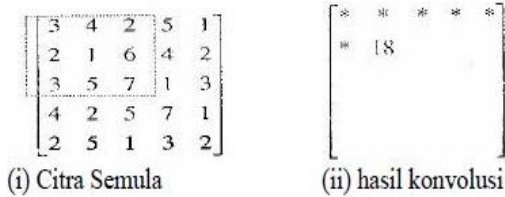
$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Arah tepi dihitung dengan persamaan :

$$\alpha(x,y) = \tan^{-1} \left( \frac{s_y}{s_x} \right)$$

Berikut adalah contoh penggunaan operator sobel . Konvolusi pertama dilakukan terhadap pixel yang bernilai 1 (titik pusat mask) :

Gambar 5. Hasil deteksi tepi dengan metode sobel



Nilai 18 pada citra hasil konvolusi didapatkan dengan perhitungan :

$$\begin{aligned} \bar{S}_x &= (3)(-1) + (2)(-2) + (3)(-1) + (2)(1) + (6)(2) + (7)(1) = 11 \\ \bar{S}_y &= (3)(1) + (4)(2) + (2)(1) + (3)(-1) + (5)(-2) + (7)(-1) = -7 \end{aligned}$$

## 6. Pengujian

### 6.1. Konversi Citra Bitmap ke Gray Scale

Proses konversi citra Bitmap ke dalam grayscale akan menggunakan citra warna dapat diubah menjadi citra keabuan dengan cara menghitung rata-rata setiap elemen warna, yaitu *Red*, *Green*, dan *Blue*. Secara matematis, perhitungan untuk konversi citra warna menjadi citra keabuan dapat dirumuskan sebagai berikut:

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

- $f_o(x,y)$  = Derajat keabuan *o* di titik (x,y)
- $f_i^R(x,y)$  = Nilai komponen warna merah pada derajat keabuan *i* di titik (x,y)
- $f_i^G(x,y)$  = Nilai komponen warna hijau pada derajat keabuan *i* di titik (x,y)
- $f_i^B(x,y)$  = Nilai komponen warna biru pada derajat keabuan *i* di titik (x,y)

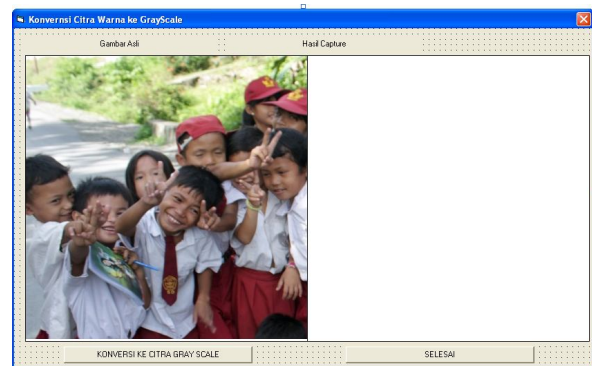
Berikut implementasi perubahan citra warna ke citra gray scale.

1. Buat form baru dengan ketentuan sebagai berikut :

Objek	Property	Nilai
Form	Name Caption	CitraBiner Konversi Ke Citra Biner
Picture1	Picture Appereance	Nama file gambar Flat

Picture2	Appereance	Flat
Command1	Caption	Konversi Ke Citra Biner
Command2	Caption	Selesai

2. Setelah dilakukan penyesuaian terlihat seperti berikut :



Gambar 6. Desain tampilan konversi dari citra warna ke citra grayscale

3. Klik Command1, isikan program berikut:

```
Private Sub Command1_Click()
For i = 1 To Picture1.Width Step 15
For j = 1 To Picture1.Height Step 15
Warna = Picture1.Point(i, j)
r = Warna And RGB(255, 0, 0)
g = Int((Warna And RGB(0, 255, 0)) / 256)
b = Int(Int((Warna And RGB(0, 0, 255)) / 256) / 256)
x = (r + g + b) / 3
Picture2.PSet (i, j), RGB(x, x, x)
Next j
Next i
End Sub
```

4. Pada Command2, isikanlah program berikut :

```
Private Sub Command2_Click()
End
End Sub
```

5. Pada Picture2, event MouseDown, ketik source code berikut :



```
Private Sub Picture2_MouseDown(Button
As Integer, Shift As Integer, X As Single, Y
As Single)
Dim Psn As Byte
w = Picture1.Point(X, Y)
r = w And RGB(255, 0, 0)
g = Int((w And RGB(0, 255, 0)) / 256)
b = Int(Int((w And RGB(0, 0, 255)) / 256) /
256)
txtr = r
txtg = g
txtb = b
Psn = MsgBox("Warna Merah adalah " &
txtr & vbCrLf & "Warna Kuning adalah "
& txtg & vbCrLf & "Warna Biru adalah "
& txtb, vbInformation + vbOKOnly,
"Hasil")
End Sub
```

6. Jalankanlah program diatas sehingga terlihat hasil sebagai berikut :



Gambar 7. Tampilan konversi dari citra warna ke citra grayscale

### 6.2. Konversi Citra Warna ke Citra Biner

Citra Biner (hitam putih) merupakan citra yang banyak dimanfaatkan untuk keperluan pattern recognition yang sederhana seperti mengenal angka atau pengenalan huruf. Untuk mengubah suatu citra gray scale menjadi citra biner, sebetulnya prosesnya sama dengan Thresholding yaitu mengubah kuantisasi citra. Untuk Citra dengan derajat keabuan 256 maka tengahnya adalah 128 sehingga mengubah menjadi citra biner dapat dituliskan:

Jika  $x < 128$  maka  $x = 0$ , jika tidak maka  $x = 255$

Untuk membuat program Konversi ke citra biner, langkahnya adalah sebagai berikut :

1. Buat form baru dengan ketentuan sebagai berikut :

Objek	Property	Nilai
Form	Name Caption	CitraBiner Konversi Ke Citra Biner
Picture1	Picture Appereance	Nama file gambar Flat
Picture2	Appereance	Flat
Command1	Caption	Konversi Ke Citra Biner
Command2	Caption	Selesai

2. Setelah dilakukan penyesuaian terlihat seperti berikut :



Gambar 8. Tampilan konversi dari citra warna ke citra biner

3. Klik Command1, isikan program berikut:

```
Private Sub Command1_Click()
Dim i As Integer, j As Integer
Dim X As Double, r As Double, g As
Double, b As Double
For i = 1 To Picture1.Width Step 15
For j = 1 To Picture1.Height Step 15
Warna = Picture1.Point(i, j)
r = Warna And RGB(255, 0, 0)
g = Int((Warna And RGB(0, 255, 0)) /
256)
b = Int(Int((Warna And RGB(0, 0,
255)) / 256) / 256)
```

```

X = (r + g + b) / 3
If X < 128 Then X = 0 Else X = 255
Picture2.PSet (i, j), RGB(X, X, X)
Next j
Next i
End Sub
    
```

4. Pada Command2, isikanlah program berikut :  
 Private Sub Command2\_Click()  
 End  
 End Sub

5. Pada Picture2, event MouseDown, ketik source code berikut :

```

Private Sub Picture2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
Dim Psn As Byte
w = Picture1.Point(X, Y)
r = w And RGB(255, 0, 0)
g = Int((w And RGB(0, 255, 0)) / 256)
b = Int(Int((w And RGB(0, 0, 255)) / 256) / 256)
txtr = r
txtg = g
txtb = b
Psn = MsgBox("Warna Merah adalah " & txtr & vbCrLf & "Warna Kuning adalah " & txtg & vbCrLf & "Warna Biru adalah " & txtb, vbInformation + vbOKOnly, "Hasil")
End Sub
    
```

6. Jalankanlah program diatas sehingga terlihat hasil sebagai berikut :



**Gambar 9. Tampilan hasil konversi dari citra warna ke citra biner**

Untuk menguji nilai pada hasil konversi, dapat dilakukan dengan mengklik pada sembarang tempat, hasil :



Penjelasan

Hasil dari program diatas tampak kurang sempurna. Hal ini disebabkan distribusi nilai derajat keabuan tidak simetri antara yang dibawah 128 dan yang diatas 128. Jika ingin mendapatkan citra biner yang lebih baik maka harus dihitung dulu nilai rata-rata derajat keabuan dan kemudian ditentukan Thresholdingnya.

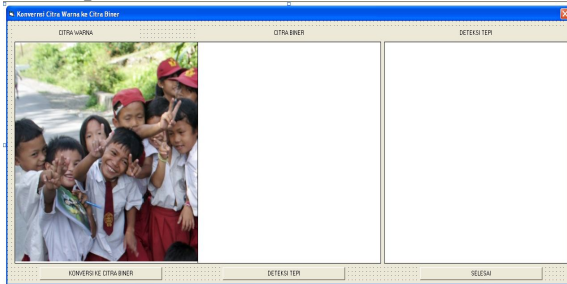
**6.3. Deteksi Tepi Sobel**

Algoritma deteksi tepi dengan metoda Sobel dapat di buat ke dalam bahasa pemrograman, pembuktian ini dilakukan dengan Microsoft Visual Basic.

1. Buat form baru dengan ketentuan sebagai berikut :

Objek	Property	Nilai
Form	Name Caption	CitraBiner Konversi Ke Citra Biner
Picture1	Picture Apperance	Nama file gambar Flat
Picture2	Apperance	Flat
Picture3	Apperance	Flat
Command1	Caption	Konversi Ke Citra Biner
Command2	Caption	Selesai
Command3	Caption	

2. Setelah dilakukan penyesuaian terlihat seperti berikut :



**Gambar 10. Tampilan konversi dari citra warna ke citra biner**

3. Bukalah source code, pada objek form, dengan event load, ketik source code berikut :

```
Private Sub Form_Load()
h1(1, 1) = -1: h1(1, 2) = 0: h1(1, 3) = 1
h1(2, 1) = -2: h1(2, 2) = 0: h1(2, 3) = 2
h1(3, 1) = -1: h1(3, 2) = 0: h1(3, 3) = 1
For i = 1 To 3
  For j = 1 To 3
    h2(i, j) = h1(j, i)
  Next j
Next i
End Sub
```

4. Pada general, ketik source code berikut :  
Dim h1(3, 3), h2(3, 3) As Single

5. Klik Command1, isikan program berikut:

```
Private Sub Command1_Click()
Dim i As Integer, j As Integer
Dim X As Double, r As Double, g As Double, b As Double
For i = 1 To Picture1.Width Step 15
  For j = 1 To Picture1.Height Step 15
    Warna = Picture1.Point(i, j)
    r = Warna And RGB(255, 0, 0)
    g = Int((Warna And RGB(0, 255, 0)) / 256)
    b = Int(Int((Warna And RGB(0, 0, 255)) / 256) / 256)
    X = (r + g + b) / 3
    If X < 128 Then X = 0 Else X = 255
    Picture2.PSet (i, j), RGB(X, X, X)
  Next j
Next i
```

```
Next i
End Sub
```

6. Pada Command2, isikanlah program berikut :

```
Private Sub Command2_Click()
Dim X(500, 500) As Integer
n1 = 0
For i = 1 To Picture2.ScaleWidth Step 15
  Picture1.ScaleWidth Step 15
  n1 = n1 + 1
  n2 = 0
  For j = 1 To Picture2.ScaleHeight Step 15
    Picture1.ScaleHeight Step 15
    Warna = Picture2.Point(i, j)
    r = Warna And RGB(255, 0, 0)
    g = Int((Warna And RGB(0, 255, 0)) / 256)
    b = Int(Int((Warna And RGB(0, 0, 255)) / 256) / 256)
    n2 = n2 + 1
    X(n1, n2) = Int((r + g + b) / 3)
    Picture3.PSet (i, j), RGB(X(n1, n2), X(n1, n2), X(n1, n2))
  Next j
Next i
```

'Proses filter prewitt dengan konvolusi

```
For i = 1 To n1
  For j = 1 To n2
    z1 = 0
    z2 = 0
    For u1 = -1 To 1
      For u2 = -1 To 1
        z1 = z1 + h1(u1 + 2, u2 + 2) * X(i + u1, j + u2)
        z2 = z2 + h2(u1 + 2, u2 + 2) * X(i + u1, j + u2)
      Next u2
    Next u1
    z = Int(Abs(z1 + z2))
    If z > 255 Then z = 255
    Picture3.PSet ((i - 1) * 15 + 1, (j - 1) * 15 + 1), RGB(z, z, z)
  Next j
Next i
End Sub
```

7. Pada Picture2, event MouseDown, ketik source code berikut :

```
Private Sub Picture2_MouseDown(Button
As Integer, Shift As Integer, X As Single, Y
As Single)
Dim Psn As Byte
w = Picture2.Point(X, Y)
r = w And RGB(255, 0, 0)
g = Int((w And RGB(0, 255, 0)) / 256)
b = Int(Int((w And RGB(0, 0, 255)) / 256) /
256)
txtr = r
txtg = g
txtb = b
Psn = MsgBox("Warna Merah adalah " &
txtr & vbCrLf & "Warna Kuning adalah "
& txtg & vbCrLf & "Warna Biru adalah "
& txtb, vbInformation + vbOKOnly,
"Hasil")
End Sub
```

8. Pada Picture3, event MouseDown, ketik source code berikut :

```
Private Sub Picture3_MouseDown(Button
As Integer, Shift As Integer, X As Single, Y
As Single)
Dim Psn As Byte
w = Picture3.Point(X, Y)
r = w And RGB(255, 0, 0)
g = Int((w And RGB(0, 255, 0)) / 256)
b = Int(Int((w And RGB(0, 0, 255)) / 256) /
256)
txtr = r
txtg = g
txtb = b
Psn = MsgBox("Warna Merah adalah " &
txtr & vbCrLf & "Warna Kuning adalah "
& txtg & vbCrLf & "Warna Biru adalah "
& txtb, vbInformation + vbOKOnly,
"Hasil")
End Sub
```

9. Jalankanlah program diatas sehingga terlihat hasil sebagai berikut :



Gambar 11. Citra hasil deteksi tepi

10. Periksalah nilai RGB masing-masing picture, dengan cara klik pada sembarang tempat, tentu posisi klik pada layar akan memperoleh hasil yang berbeda satu sama lainnya.

- a. Citra Warna, diperoleh :



- b. Citra Gray Scale, diperoleh



## 7. Kesimpulan

Setelah dilakukan pengujian program dalam bentuk training atas data, maka dapat di tarik kesimpulan sebagai berikut :

1. Sumber citra digital akan mempengaruhi hasil pada deteksi sobel, citra digital harus bersumber dari citra biner.
2. Penelitian ini masih dapat di kembangkan kedalam penerapan K-Mean sebagai algoritma segmentasi citra digital.
3. Kerumitan objek yang di segmentasi sangat menentukan terhadap hasil akhir yang diperoleh, semakin sederhana objek tersebut maka akan semakin mudah untuk dikenali, dan objek yang rumit akan semakin susah di kenali.

## Referensi

- [1] Wan-Ting Lin, Chuen-Hong Lin, Tsung-Ho Wu, Yung-Kuan Chan, *Image Segmentation Using the K-means Algorithm for Texture Features*, World Academy of Science, Engineering and Technology, Volume 41, 2010, pp 612-615
- [2] Pushpa. R. Suri, Mahak, *Image Segmentation With Modified K-Means Clustering Method*, International Journal of Recent Technology and Engineering, ISSN 2277-3878, Volumen-1, Issue-2, June 2012
- [3] Albar, Ismail, Fibriyanti, *Identifikasi dengan Menggunakan Algoritma K-Means pada Plat Kenderaan*, Poli Rekayasa ISSN: 1858-3709, Volume 6, Nomor 1, Oktober 2010
- [4] Identifikasi dengan Menggunakan Algoritma K-Means pada Plat Kenderaan
- [5] Yiu-Ming Cheung, *k-Means: A new generalized k-means clustering algorithm*, Pattern Recognition Letters, Elsevier, 2003