

SEGMENTASI CITRA LAHAN BEKAS TAMBANG TERBUKA MENGUNAKAN METODE DETEKSI TEPI CANNY

Ayu Ambarwati¹, Rossi Passarella², Sutarno³

Abstract--Edge detection in the image of mining land become an important because this process is the first step improve the appearance of the boundary line of an object image. Canny edge detection method is popular and has the advantages of detecting and localizing well, as well as a clear response. Selection of parameters greatly affect the outcome Canny edge. Parameters include standard deviation Gaussian (σ), the size of the Gaussian filter matrix and a threshold value. Canny method has 5 process consists a smoothing, finding gradient, non-maxima suppression, double thresholding and edge tracking by hysteresis. In this study determined the size of the matrix of 5x5, 7x7 and 9x9 while the value of σ of 1, 2.5 and 5. Thresholds are adjusted using Otsu method. The program is created using programming language C#. The result showed that the greater size of the matrix and value of σ then produce blurred image. For the same matrix size, the greater the value of σ produce the smaller threshold and edge image obtained more detail.

Index Terms—Mining land, edge detection, Otsu, Gaussian Filtering, Canny.

I. PENDAHULUAN

Kegiatan eksploitasi pertambangan dengan pola lahan tambang terbuka tidak hanya memberikan keuntungan tetapi juga memberikan permasalahan seperti rusaknya ekosistem hutan berupa hilangnya fungsi hutan, meninggalkan lubang-lubang dan tanah kosong. Permasalahan ini haruslah diselesaikan dengan baik agar kerusakan alam akibat pasca pertambangan dapat dikembalikan fungsinya seperti semula. Salah satu kajian mengenai permasalahan diatas telah dilakukan oleh Rika (2013) pada penelitian [1] menggunakan pengolahan citra digital penginderaan jauh terhadap citra satelit dari suatu wilayah yang memiliki lahan bekas penambangan. Teknik penginderaan jauh merekam data dipermukaan bumi dan mengolah informasi yang diperoleh dari suatu objek atau daerah tersebut menggunakan sebuah alat tanpa kontak langsung dengan objek atau daerah yang dikaji[2]. Foto udara dikatakan sebagai produk dari teknik penginderaan jauh[3].

Metode untuk melakukan pemrosesan terhadap citra foto udara dengan pemanfaatan teknik pengolahan citra digital melalui deteksi tepi[3]. Deteksi tepi merupakan teknik memperoleh informasi tepi dari suatu gambar[3]. Deteksi tepi menjadi hal dianggap penting dan mendasar dalam pengolahan citra digital. Tepi menggambarkan batas-batas antara objek dan latar belakang yang membantu dalam segmentasi[4].

Canny merupakan algoritma deteksi tepi modern yang memberikan deteksi tepi baik, melokalisasi titik-titik tepi yang tepat dan respon yang jelas[5]. Pada penelitian [6] menggunakan *Canny* dalam mendeteksi sisi jalan pada foto udara. Selanjutnya penelitian [7] dan [8] menunjukkan hasil deteksi tepi *Canny* lebih halus, tipis dan nyata dibandingkan metode deteksi tepi yang lain.

Pemilihan parameter deteksi tepi *Canny* sangat mempengaruhi hasil tepian citra[5]. Parameter itu diantaranya nilai standar deviasi *gaussian* (σ) dan *threshold*. Tingkat kehalusan citra hasil tapis *Gaussian* dapat diatur dengan mengubah-ubah nilai σ . Sementara *threshold* diatur terlalu tinggi akan menghilangkan informasi penting pada gambar sedangkan jika terlalu rendah akan menyebabkan kesalahan mendeteksi (*false edge*). Salah satu metode yang dapat digunakan untuk menentukan *threshold* adalah metode *Otsu*[1].

Tujuan penelitian ini adalah mendeteksi tepi citra foto udara lahan bekas tambang menggunakan metode deteksi tepi *Canny*. Program dibuat menggunakan bahasa pemrograman C#.

II. TINJAUAN PUSTAKA

2.1 Perenggangan Kontras

Kontras menyatakan suatu sebaran gelap dan terang nilai keabuan pada citra[1][9]. Histogram citra dengan kontras rendah ditandai dengan sebaran nilai keabuannya mengelompok pada satu sisi gelap (histogram cenderung kiri) atau terang (histogram cenderung kanan). Citra dengan kontras baik menampilkan histogram yang sebaran nilai *pixel* yang seragam. *Pixel* pada citra masukan ditransformasikan dengan menggunakan persamaan 2.1.

$$f(x,y)' = \frac{f(x,y) - \min}{\max - \min} \quad (\text{pers.2.1})$$

dimana $f(x,y)'$ dan $f(x,y)$ berturut-turut nilai skala keabuan yang baru dan nilai skala keabuan citra semula pada koordinat (x,y) , kemudian \max dan \min merupakan nilai skala keabuan tertinggi dan terendah dari kelompok *pixel*.

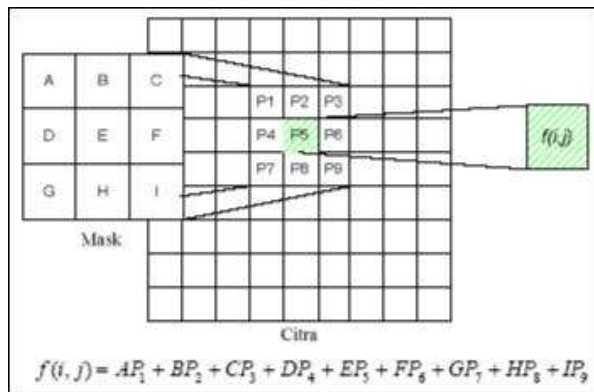
¹ Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya (e-mail: ayuambarwati29@gmail.com).

^{2,3} Dosen Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya (e-mail: ^{2,3} Passarella.rossi@gmail.com,

³ tarnoabufaruq@yahoo.com).

2.2 Konvolusi

Konvolusi merupakan proses memperoleh suatu *pixel* baru berdasarkan nilai *pixel* itu sendiri dan tetangganya, dengan melibatkan kernel[10]. Ilustrasi konvolusi dijelaskan pada gambar 2.1



Gambar 2.1. Contoh Konvolusi

Pada penelitian ditambahkan baris dan kolom pada bagian tepi dengan nilai 0 pada bagian tersebut sehingga dapat dilakukan proses konvolusi.

2.3 Thresholding

Thresholding mengubah citra keabuan menjadi citra biner bergantung pada nilai *threshold* (T) sehingga dapat diketahui daerah mana yang termasuk objek dan latar belakang[12]. Secara umum proses *thresholding* ditunjukkan pada persamaan 2.2.

$$1(,) \geq T$$

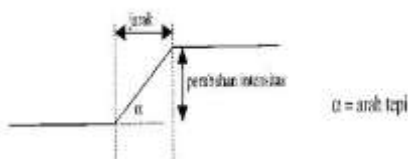
$$0(,) < T$$

(pers.2.2)

Persamaan 2.2 menjelaskan bahwa $g(x,y)$ adalah citra biner dari citra aras keabuan $f(x,y)$ dan T menyatakan nilai ambang.

2.4 Deteksi Tepi

Deteksi tepi digunakan untuk memperoleh tepi-tepi dari objek[10]. Tepi didefinisikan sebagai perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat seperti dilihat pada gambar 2.2. Tepi mengandung informasi penting berupa bentuk maupun ukuran objek. Suatu titik (x,y) pada citra dikatakan sebagai tepi ditandai tingginya perubahan nilai *pixel* sangat kontras terhadap *pixel* tetangganya[13].



Gambar 2.2. Model tepi satu dimensi

Besar gradient magnitude dihitung dengan persamaan 2.3.

$$G[f(x,y)] = \sqrt{G_x^2 + G_y^2}$$

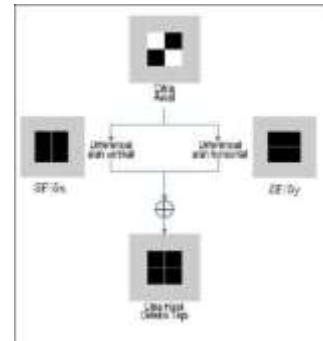
(pers.2.3)

Arah tepi dihitung dengan persamaan 2.4. dibawah ini :

$$\theta(x,y) = \arctan\left(\frac{G_y}{G_x}\right)$$

(pers.2.4)

Nilai gradient diperoleh dari gabungan hasil perhitungan arah vertikal dan horizontal dari dua buah kernel ditunjukkan pada gambar 2.3 dibawah ini.



Gambar 2.3. Proses deteksi tepi pada operator gradient

2.4.1 Sobel

Sobel menggunakan dua buah kernel horizontal dan vertikal berukuran 3x3 sehingga perkiraan gradien berada tepat ditengah jendela[8][7]. Keuntungan operator ini memberikan hasil deteksi tepi yang baik dan mudah dalam implementasi[8]. Operator *Sobel* sensitif terhadap arah tepi diagonal daripada tepi vertikal dan horizontal[10].

$$\begin{bmatrix} 0 & 1 & 2 \\ 7 & 0 & 3 \\ 6 & 5 & 4 \end{bmatrix}$$

Gambar 2.4. Kernel ukuran 3x3

Didefinisikan G_x dan G_y pada persamaan 2.5:

$$\begin{aligned} G_x &= (a_2 - a_0) + (a_4 - a_6) + c(a_3 - a_7) \\ G_x &= (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \\ G_y &= (a_0 - a_6) + (a_2 - a_4) + c(a_1 - a_5) \\ G_y &= (a_0 + ca_1 + a_2) - (a_4 + ca_5 + a_6) \end{aligned}$$

(pers.2.5)

c adalah konstanta yang bernilai 2.

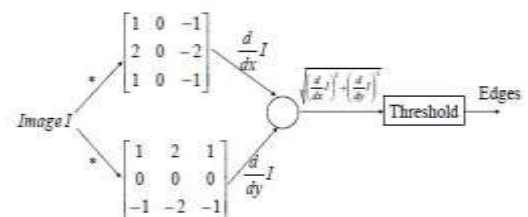
-1	0	1
-2	0	2
-1	0	1

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

(a) G_x

(b) G_y

Gambar 2.5. Kernel Sobel



Gambar 2.6. Deteksi Tepi Sobel

2.4.2 Canny

Canny dikembangkan oleh John Canny pada tahun 1986, menggunakan algoritma multi-tahap dalam mendeteksi tepi citra[14]. Terdapat lima tahap dalam mendeteksi tepi Canny sebagai:

1. Mengimplementasikan tapis Gaussian untuk mereduksi noise dan meningkatkan kualitas detail citra. Proses ini menghasilkan citra tampak sedikit lebih buram yang. Filter Gaussian 2D dinyatakan persamaan 2.6.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (\text{pers.2.6})$$

dimana $G(x,y)$ elemen matriks gaussian pada posisi (x,y) , $\pi=22/7$, $e = 2.71828182846$, σ ialah nilai standar deviasi. Berdasarkan rumus diatas, salah satu contoh filter Gaussian dengan $\sigma = 2,5$ dan ukuran kernel 5x5 ditunjukkan gambar 2.7.

$\frac{1}{2\pi(2.5)^2} e^{-\frac{(-2)^2+(-2)^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{(-2)^2+(-1)^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{(-2)^2+0^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{(-2)^2+1^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{(-2)^2+2^2}{2(2.5)^2}}$
$\frac{1}{2\pi(2.5)^2} e^{-\frac{(-1)^2+(-2)^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{(-1)^2+(-1)^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{(-1)^2+0^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{(-1)^2+1^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{(-1)^2+2^2}{2(2.5)^2}}$
$\frac{1}{2\pi(2.5)^2} e^{-\frac{0^2+(-2)^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{0^2+(-1)^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{0^2+0^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{0^2+1^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{0^2+2^2}{2(2.5)^2}}$
$\frac{1}{2\pi(2.5)^2} e^{-\frac{1^2+(-2)^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{1^2+(-1)^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{1^2+0^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{1^2+1^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{1^2+2^2}{2(2.5)^2}}$
$\frac{1}{2\pi(2.5)^2} e^{-\frac{2^2+(-2)^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{2^2+(-1)^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{2^2+0^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{2^2+1^2}{2(2.5)^2}}$	$\frac{1}{2\pi(2.5)^2} e^{-\frac{2^2+2^2}{2(2.5)^2}}$

Gambar 2.7. Filter Gaussian $\sigma = 2,5$ dan matriks 5x5

Semua posisi diisi sesuai dengan titik koordinat x,y sehingga menghasilkan nilai bobot dibawah ini :

0,01340	0,01710	0,01850	0,01710	0,01340
0,01710	0,02170	0,02350	0,02170	0,01710
0,01850	0,02350	0,02550	0,02350	0,01850
0,01710	0,02170	0,02350	0,02170	0,01710
0,01340	0,01710	0,01850	0,01710	0,01340

Gambar 2.8. Bobot nilai filter Gaussian Gaussian $\sigma = 2,5$ dan matriks 5x5

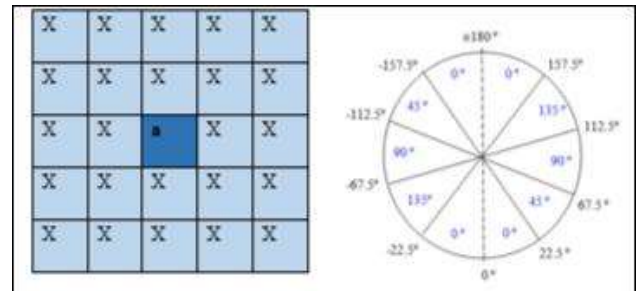
Setelah didapat nilai pada masing-masing posisi koordinat x,y selanjutnya dilakukan normalisasi dengan mengalikan setiap nilai pada tersebut terhadap faktor pengali. Faktor pengali dibuat sama yang dihitung dengan persamaan 2.7 :

$$\frac{1}{\sum_{x,y} G(x,y)} \quad (\text{pers.2.7})$$

Dari nilai-nilai diatas, didapat jumlah keseluruhan 1,086 dan faktor pengali 0,9208103 dan kemudian mengalikannya dengan nilai-nilai disetiap koordinat x,y.

0,02847	0,03633	0,03930	0,03633	0,02847
0,03633	0,04610	0,04993	0,04610	0,03633
0,03930	0,04993	0,05417	0,04993	0,03930
0,03633	0,04610	0,04993	0,04610	0,03633
0,02847	0,03633	0,03930	0,03633	0,02847

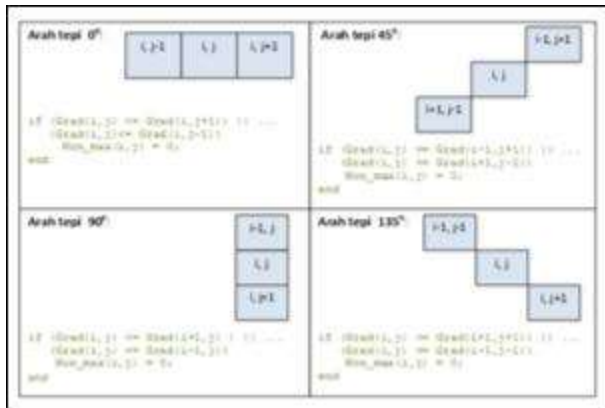
2. Melakukan perhitungan gradient magnitudo dan arah tepi. Perhitungan gradient magnitudo menggunakan kernel Sobel. Sedangkan perhitungan arah tepi menggunakan persamaan 2.4. Selanjutnya melakukan orientasi tepi dengan menghubungkan arah tepi ke arah yang dapat ditelusuri dalam citra ditunjukkan pada gambar 2.10.



Gambar 2.9. Hasil perhitungan normalisasi bobot nilai filter Gaussian $\sigma = 2,5$ dan matriks 5x5

Gambar 2.10. Pembagian sudut gradient penentu arah

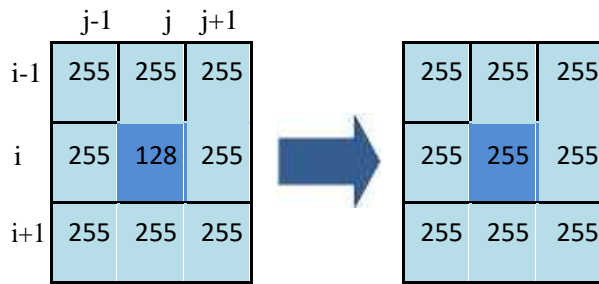
3. Memperkecil garis tepi yang muncul dengan menerapkan *nonmaximum suppression* sehingga menghasilkan garis tepian yang lebih ramping. *Nonmaximum Suppression* dilaksanakan dengan memperhatikan dua titik tetangga yang terletak pada arah tepi ditunjukkan pada gambar 2.11.



Gambar 2.11. *Nonmaximum Suppression*

4. Menerapkan dua buah *threshold* (*double thresholding*). Sederhananya *double thresholding* bertujuan klasifikasi dua buah nilai *High-threshold* (T_2) dan *Low-threshold* (T_1), dengan $T_2 \approx 2T_1$ [15]. Jika nilai *pixel* lebih besar atau sama dengan T_2 maka diatur nilai 255, jika nilai *pixel* kurang dari atau sama dengan T_1 maka diatur menjadi 0. *Pixel* diantara T_1 dan T_2 disebut kandidat *pixel* tepi maka sementara diberi nilai 128.

5. *Edge tracking by hysteresis* bertujuan memperoleh tepian final dengan menekan semua sisi yang tidak terhubung pada tepian yang sangat kuat[14]. Nilai 128 selanjutnya dilakukan pengecekan pada *pixel* dari 8 arah tetangganya, sehingga *pixel* hanya bernilai 0 atau 255. Perubahan nilai 128 menjadi nilai 255 apabila semua kondisi terpenuhi yakni jika salah satu atau semua *pixel* pada 8 arah tetangganya bernilai 255 ditunjukkan pada gambar 2.12.

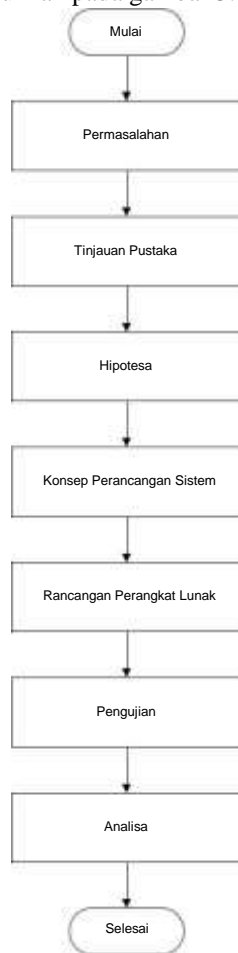


Gambar 2.12. Pengujian mengubah nilai 128 menjadi 255

III. METODOLOGI

1.1 Kerangka Kerja

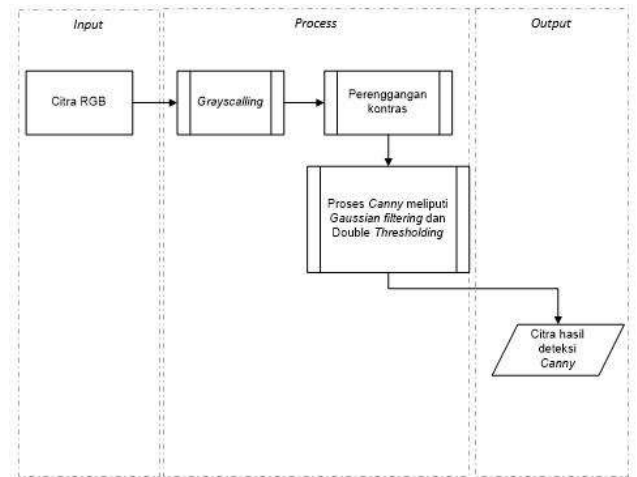
Metodologi yang digunakan melalui beberapa tahapan kerangka kerja supaya penelitian ini lebih terstruktur dan teratur seperti ditunjukkan pada gambar 3.1.



Gambar 3.1. Kerangka Kerja Penelitian

1.2 Kerangka Kerja

Konsep perancangan merupakan desain atau alur dari sistem yang akan dibuat yang ditunjukkan pada gambar 3.2.



Gambar 3.2. Bagan umum sistem

1.3 Rancangan Perangkat Lunak

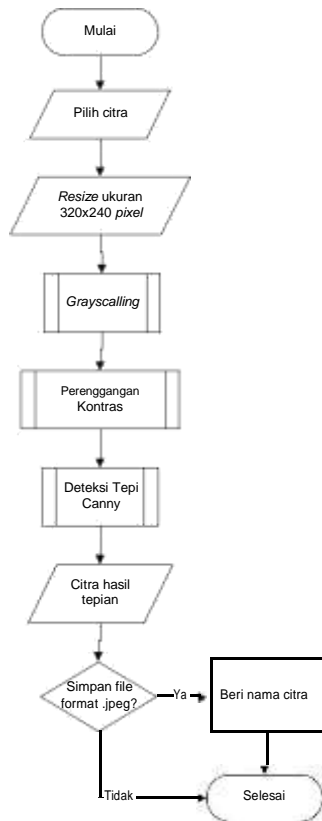
Setiap awal data berupa masukan citra RGB kemudian di-*resize* menjadi ukuran 320x240 *pixel*. Dilanjutkan proses *grayscale* kemudian proses perbaikan citra berupa perengangan kontras. Setelah citra diperbaiki kontrasnya dilakukan proses deteksi tepi *Canny*. Hal yang pertama yang dilakukan yaitu proses *gaussian filtering*. Tingkat kehalusan citra hasil tapis *Gaussian* dapat diatur dengan mengubah-ubah nilai σ . Penelitian ini ditetapkan ukuran matriks 5x5, 7x7 dan 9x9 sementara nilai σ yang digunakan 1, 2.5 dan 5. Selain kedua parameter yang telah disebutkan, *threshold* juga mempengaruhi hasil deteksi tepi *Canny* di proses *double thresholding*. Pada penelitian ini menggunakan metode *Otsu* untuk mendapatkan nilai ambang (*threshold*).



Gambar 3.3. Tampilan form main program

1.3.1 Pengolahan Citra Digital

Pengolahan citra digital menjadi proses utama dalam penelitian ini seperti ditunjukkan gambar 3.4.



Gambar 3.4. Diagram alir aplikasi deteksi tepi

1.3.2 Mendapatkan Citra Digital

Adapun citra yang diolah adalah dalam penelitian ini data citra yang diperoleh dari sumber internet berjumlah 2 buah citra digital lahan bekas tambang berformat JPEG dengan ukuran yang sudah di *resize* menjadi ukuran 320x240 *pixel* dapat dilihat pada gambar 3.5.



Gambar 3.5. Citra Uji

1.3.3 Grayscale

Selanjutnya dilakukan proses *grayscale* untuk mempermudah proses pengolahan yang ditunjukkan pada *pseudocode* gambar 3.6. berikut :

```

Tetapan :
red, green, blue, grayscale : integer;
citra : bitmap;

Deskripsi :
for y ← 0 to citra.height do
  for x ← 0 to citra.width do
    red = citra.GetPixel(x,y).R;
    green = citra.GetPixel(x,y).G;
    blue = citra.GetPixel(x,y).B;
    grayscale = red*0,299 + green*0,587 + blue*0,114;
  end for
end for

```

Gambar 3.6 Pseudocode Grayscale

1.3.4 Perengangan Kontras

Tujuan dari proses ini supaya objek didalam citra lebih kontras dengan latar belakang citra yang ditunjukkan pada *pseudocode* gambar 3.7.

```

Tetapan :
citra : bitmap;
nilaiPixel, jumlah, accum, minI, maxI : int;
int[] tampung = new int[256];
double blackPointPercent = 0.1;
double whitePointPercent = 0.1;
blackPixel, whitePixel : double;
jumlah = citra.width * citra.height;
blackPixel = jumlah * blackPointPercent;
whitePixel = jumlah * whitePointPercent;

Deskripsi :
for y ← 0 to citra.height do
  for x ← 0 to citra.width do
    nilaiPixel = citra.getPixel(x,y).R;
    tampung[nilaiPixel]++;
  end for
end for

minI = 0;
accum = 0;

while(minI < 255) do
  accum += tampung[minI];
  if (accum > blackPixel) break;
  minI++;
endwhile

maxI = 0;
accum = 0;

while(maxI > 0) do
  accum += tampung[maxI];
  if (accum > whitePixel) break;
  maxI--;
endwhile

double spread = 255 / (maxI - minI);

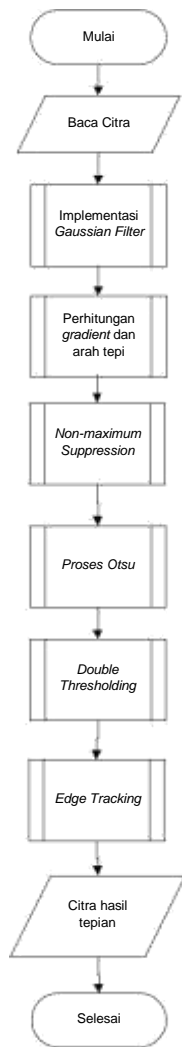
for y ← 0 to citra.height do
  for x ← 0 to citra.width do
    nilaiPixel = citra.getPixel(x,y).R;
    double value = ((nilaiPixel - minI) * spread);
    byte val = (math.Min(Math.Max(value, 0), 255));
  end for
end for

```

Gambar 3.7. Pseudocode Perengangan Kontras

1.3.5 Deteksi Tepi Canny

Canny umumnya memiliki 6 tahapan dalam mendeteksi tepi yang ditunjukkan pada gambar 3.8.

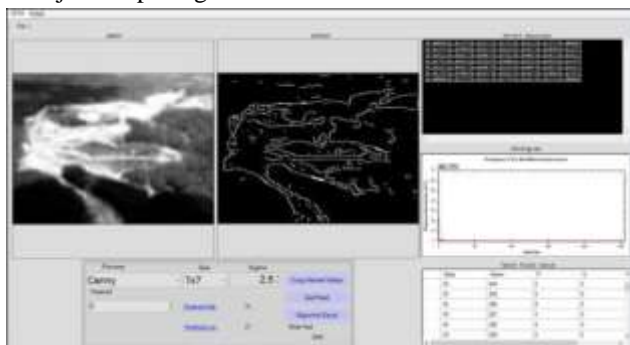


Gambar 3.8. Flowchart Deteksi Tepi Canny

IV. HASIL PENELITIAN

4.1 Pengujian Program

Pada pengujian perangkat lunak terdiri dari proses *grayscale*, perenggangan kontras dan deteksi tepi Canny sehingga diperoleh citra deteksi tepi Canny ditunjukkan pada gambar 4.1. berikut :



Gambar 4.1. Tampilan awal program

4.2 Hasil Pengujian

Berikut hasil pengujian dari proses deteksi tepi :

4.2.1 Grayscale dan Perenggangan Kontras

Dari proses yang menggunakan *pseudocode* pada gambar 3.6 dan 3.7 didapat hasil citra proses *grayscale* dan perenggangan kontras kedua citra yang ditunjukkan pada tabel 4.1.

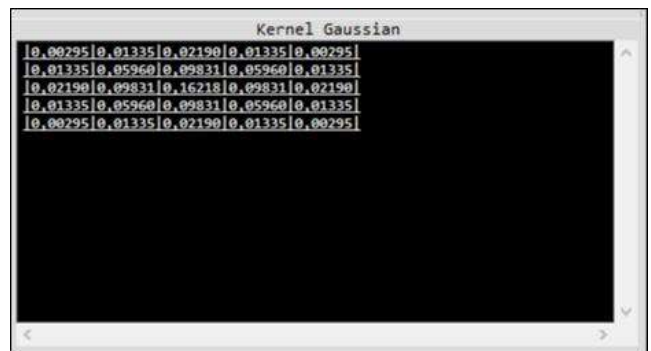
Citra	Grayscale	Perenggangan Kontras
tambang1.jpg		
tambang2.jpg		

Tabel 4.1 Hasil Pengujian Grayscale dan perenggangan kontras

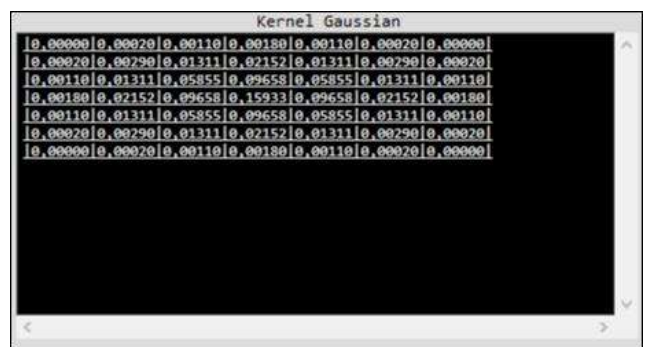
4.2.2 Deteksi Tepi Canny

4.2.2.1 Implementasi Gaussian Filtering

Pada tahap ini terdapat dua proses yaitu perhitungan nilai bobot kernel berdasarkan ukuran kernel & nilai standar deviasi *gaussian* (σ) seperti ditunjukkan pada gambar 4.3 dan 4.4 kemudian dilanjutkan proses konvolusi antara citra perenggangan kontras dengan nilai bobot kernel yang dihitung sebelumnya yang ditunjukkan pada tabel 4.2 dan 4.3.



Gambar 4.3. Nilai bobot kernel pada matriks 5x5 $\sigma = 1$



Gambar 4.4. Nilai bobot kernel pada matriks 7x7 $\sigma = 1$

Nilai σ	Matriks 5x5	Matriks 7x7	Matriks 9x9
1			
2,5			
5			

Tabel 4.2 Hasil Pengujian *Gaussian Filtering* citra tambang1.jpg

Nilai σ	Matriks 5x5	Matriks 7x7	Matriks 9x9
1			
2,5			
5			

Tabel 4.3 Hasil Pengujian *Gaussian Filtering* citra tambang2.jpg

4.2.2.2 Edge Tracking

Setelah proses *gaussian filtering*, dilanjutkan berturut-turut proses perhitungan gradient dan arah tepi, *nonmaximum supression*, proses *Otsu*. Dari proses *Otsu* didapat masing-masing *threshold* untuk digunakan dalam proses *double thresholding*. Proses terakhir yakni *edge tracking* ditunjukkan pada tabel 4.4 dan tabel 4.5.

Nilai σ	Matriks 5x5	Matriks 7x7	Matriks 9x9
1			
2,5			
5			

Tabel 4.4 Hasil Pengujian *Edge Tracking* citra tambang1.jpg

Nilai σ	Matriks 5x5	Matriks 7x7	Matriks 9x9
1			
2,5			
5			

Tabel 4.5 Hasil Pengujian *Edge Tracking* citra tambang2.jpg

V. PENUTUP

Setelah dilakukan pengujian dan pembahasan dapat disimpulkan sebagai berikut:

1. Telah dirancang program aplikasi yang mengimplementasikan algoritma deteksi tepi *Canny* terhadap citra lahan bekas tambang terbuka serta menerapkan metode *Otsu* untuk mendapatkan nilai ambang.
2. Berdasarkan hasil percobaan menggunakan variasi ukuran matriks dan nilai σ . Semakin besar ukuran matriks dan nilai σ maka menghasilkan citra yang lebih buram pada proses *Gaussian Filtering*.
3. *Threshold* diperoleh dari perhitungan *Otsu*. Variasi ukuran matriks dan nilai σ akan menghasilkan variasi nilai *threshold*. Nilai *threshold* yang diperoleh merupakan nilai ambang yang optimal.
4. Pada ukuran matriks yang sama, semakin besar nilai σ yang digunakan maka menghasilkan *threshold* semakin kecil dan hasil tepi yang diperoleh semakin detail.

Saran untuk penelitian selanjutnya :

1. Penelitian bisa dilakukan pada objek lain seperti pada citra *MRI* gigi manusia.
2. Perlu adanya parameter tambahan seperti jumlah *pixel* putih untuk menunjukkan keakuratan hasil tepi yang diperoleh.

DAFTAR PUSTAKA

- [1] R. F. Gusa, "Pengolahan Citra Digital untuk Menghitung Luas Daerah Bekas Penambangan Timah," *J. Nas. Tek. Elektro Univ. Bangka Belitung*, vol. 2, no. 2, pp. 27–34, 2013.
- [2] L. Pambudi, A. Hidayatno, and R. I. Rizal, "Identifikasi Luas Bencana Tsunami dengan Menggunakan Segmentasi Citra," *J. Nas. Tek. Elektro Univ. Diponegoro*, pp. 1–9, 2011.
- [3] F. Liantoni and R. I. Perwira, "Optimasi Algoritma Semut untuk Deteksi Tepi pada Foto Udara," *J. Nas. Tek. Inform. Inst. Teknol. Surabaya*, vol. 9, pp. 63–69, 2016.

- [4] M. Avlash and L. Kaur, "Performances Analysis of Different Edge Detection Methods on Road Images," *Int. J. Adv. Res. Eng. Appl. Sci.*, vol. 2, no. 6, pp. 27–38, 2013.
- [5] N. Sa'adah and A. Purqon, "Variasi Filter pada Deteksi Tepi Metode Canny untuk Mendeteksi Kanker Payudara," *Pros. SKF Inst. Teknol. Bandung*, pp. 530–536, 2015.
- [6] M. Iqbal, "Analisis Threshold untuk Mengekstraksi Jalan Menggunakan Canny dari Foto Udara," *J. Nas. Ilmu Komput. Univ. Sumatera Utara*, 2010.
- [7] Z. Othman, M. Rafiq, and A. Kadir, "Comparison of Canny and Sobel Edge Detection in MRI Images," pp. 133–136.
- [8] S. B. Kutty, S. Saaidin, P. Nor, A. Megat, and S. A. Hassan, "Evaluation of Canny and Sobel Operator for Logo Edge Detection," vol. 2, no. Istmet, pp. 153–156, 2014.
- [9] Murinto, "Perbaikan Kualitas Citra Menggunakan Histogram Linear Contrast Stretching pada Citra Skala Keabuan," *Conf. Pap. Ahmad Dahlan Univ. August 2005*, 2016.
- [10] A. Kadir and A. Susanto, *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta, Indonesian: Andi, 2013.
- [11] E. Prasetyo, *Pengolahan Citra Digital dan Aplikasinya Menggunakan MatLab*. Yogyakarta, Indonesian: Andi, 2011.
- [12] V. Sucharita, S. Jyothi, and D. M Mamatha, "Evaluation of the Digital images of Penaeid Prawns Species Using Canny Edge Detection and Otsu Thresholding Segmentation," *Int. J. Emerg. Technol. Comput. Appl. Sci. (IJETCAS)*, pp. 117–121, 2013.
- [13] M. Yunus, "Perbandingan Metode-metode Edge Detection untuk Proses Segmentasi Citra Digital," *J. Teknol. Inf.*, vol. 3, no. 2, pp. 146–160.
- [14] C. Danil, "Edge Detection dengan Algoritma Canny," pp. 197–203.
- [15] M. Fang, G. Yue, and Q. Yu, "The Study on An Application of Otsu Method in Canny Operator," *Int. Symp. Inf. ...*, vol. 2, no. 4, pp. 109–112, 2009.