

# ANALISIS DETEKSI TEPI CANNY PADA CITRA DENGAN GAUSSIAN FILTERING DAN BILATERAL FILTERING

**Safriadi<sup>1)</sup>, Aulia Essra<sup>2)</sup>, Rahmadani<sup>3)</sup>**

*Magister Teknik Informatika, Universitas Sumatera Utara<sup>1,2)</sup>*

*Universitas Pembangunan Panca Budi<sup>3)</sup>*

*Jl. Universitas No.24A Kampus USU, Medan, 20155, Indonesia*

*E-mail : safriadi48@gmail.com<sup>1)</sup>*

---

## **ABSTRACT**

*Edge detection is one of the most important things in image processing. Canny edge detection operator is one of the most excellent edge detection. There are four basic stages in canny edge detection is filtering, determine the value of the gradient, pressure gradient value and connect each edge of the image. In this research the process of filtering using Gaussian filtering and Bilateral filtering. The imagery used in this study is an image with the file extension \*.bmp. in this research testing is done with an image resolution of different. The results of the testing that was done that with bilateral filtering can produce a more optimal image information than with Gaussian filtering.*

**Keyword :** *canny edge detection, gaussian filtering, bilateral filtering*

## **ABSTRAK**

*Deteksi tepi merupakan salah satu hal yang paling penting dalam pengolahan citra. Deteksi tepi canny merupakan salah satu operator deteksi tepi yang paling baik. Ada empat tahapan dasar pada deteksi tepi canny yaitu filtering, menentukan nilai gradient, penekanan nilai gradient dan menghubungkan setiap tepi pada citra. Pada penelitian ini proses filtering menggunakan filtering Gaussian dan Bilateral filtering. Adapun citra yang digunakan pada penelitian ini adalah citra dengan ekstensi file \*.bmp. pada penelitian ini pengujian dilakukan dengan resolusi gambar yang berbeda – beda. Hasil dari pada pengujian yang telah dilakukan bahwa dengan bilateral filtering dapat menghasilkan informasi citra yang lebih optimal dari pada dengan Gaussian filtering.*

**Kata kunci:** *deteksi tepi canny, Gaussian filtering, Bilateral filtering*

## PENDAHULUAN

Deteksi tepi merupakan salah satu bagian yang paling penting dalam pengolahan citra<sup>[1]</sup>. Tujuan dari deteksi tepi adalah untuk menyederhanakan sebuah gambar dan dikenali dengan mendeteksi garis-garis yang membentuk sebuah objek yang ada pada citra. Salah satu operator untuk deteksi tepi pada citra yang baik adalah operator *Canny*<sup>[2]</sup>.

Pada dasarnya deteksi tepi *Canny* mempunyai 4 tahapan dasar yaitu: 1) *low-pass filtering* dengan *Gaussian Filtering*; 2) menentukan nilai *gradient* sebuah citra; 3) penekanan *non-maxima* pada nilai *gradient*; 4) memeriksa dan menghubungkan setiap tepi pada citra. Dari keempat tahapan deteksi tepi *Canny*, tahapan filtering merupakan tahapan yang penting untuk mendeteksi tepi citra dengan *Canny*<sup>[3][4]</sup>.

*Gaussian Filtering* adalah salah satu teknik untuk mengurangi noise yang umum digunakan. *Bilateral Filtering* adalah Bilateral Filter merupakan salah satu metode restorasi citra khususnya penapis derau pada citra digital. Pada metode ini, nilai piksel citra hasil diperoleh dari rata - rata pembobotan piksel – piksel tetangga melalui proses konvolusi<sup>[1]</sup>.

## METODE PENELITIAN

Pada metode penelitian dijelaskan rancangan penelitian yang dilakukan berdasarkan tahapan – tahapan yang digunakan untuk proses deteksi tepi. Adapun teknik yang digunakan untuk mengurangi noise adalah dengan *bilateral filtering* dan *Gaussian filtering*.

### Filtering

Tahapan pertama yang dilakukan pada citra sebelum deteksi tepi *canny* adalah *filtering* untuk mengurangi noise. Proses filtering dapat mencegah munculnya deteksi tepi palsu yang

disebabkan dari noise. Untuk mengurangi noise digunakan metode *bilateral filtering* dan *Gaussian filtering* sehingga menghasilkan *bilateral filtering image* dan *Gaussian Filtering Image*.

### Grayscale

Proses yang terjadi selanjutnya setelah citra dihilangkan noise adalah mengubah pixel – pixel citra RGB menjadi citra *grayscale*. Citra grayscale digunakan untuk menyederhanakan model citra. Citra berwarna terdiri dari tiga layer matrik yaitu R- layer, G-layer dan B-layer. Sehingga untuk melakukan proses-proses selanjutnya tetap dengan memperhatikan tiga layer tersebut<sup>[5]</sup>. Bila setiap proses perhitungan dilakukan menggunakan tiga layer, berarti dilakukan tiga perhitungan yang sama. Konsep tersebut diubah dengan mengubah tiga layer menjadi satu layer matrik grayscale dan hasilnya adalah citra grayscale. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan. 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<sup>[6]</sup>.

### Mencari Arah Dan Jarak Gradient

Deteksi tepi *Canny* mencari tepi dimana intensitas derajat keabuan citra berubah secara drastis. Lokasi tersebut didapatkan dengan menentukan gradien dari citra. Gradien pada tiap pixel dari citra yang telah dihaluskan ditentukan dengan operator Sobel. Langkah pertama adalah untuk memperkirakan gradien sesuai jumlah operator arah dengan mengaplikasikan kernel<sup>[7]</sup>.

### Partisii Citra

Citra dipartisi menjadi beberapa blok dimana masing-masing blok mempunyai ukuran B x B pixel<sup>[7]</sup>. Tujuan dilakukan partisi citra menjadi beberapa blok adalah untuk melakukan

segmen beberapa objek. Ukuran tiap blok dapat disesuaikan.

### Operasi Blok Hasil Partisi Citra

Pada proses ini dilakukan operasi pada tiap pixel dalam blok untuk semua blok. Sehingga setiap blok melakukan operasi yang sama<sup>[2]</sup>. Operasi yang dimaksud adalah mencari titik-titik tetangga dari koordinat pixel (x,y).

### Penentuan Target Setiap Kandidat

Setelah titik-titik tetangga dari semua koordinat dalam blok didapatkan, langkah berikutnya adalah mencari perbedaan nilai gray level untuk setiap *d*. Dimana *d* adalah arah, *r* dan *l* adalah tetangga suatu koordinat dan *mn* adalah lokasi suatu blok. Terdapat empat nilai selisih gray level sesuai dengan *d* atau arah yang digunakan<sup>[2]</sup>.

### Dual Threshold

Pada tahapan terakhir ini sebelum menghubungkan setiap tepi pada citra, maka terlebih dahulu untuk meentukan nilai *dual-threshold*. Dual threshold ini akan digunakan sebagai parameter untuk menghubungkan setiap tepi pada citra<sup>[2]</sup>.

### MSE (Mean Square Error)

MSE adalah rata-rata kuadrat nilai kesalahan antara citra asli dengan citra hasil pengolahan<sup>[8]</sup>. Yang mana secara matematis dapat dirumuskan sebagai berikut:

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x,y) - g(x,y))^2 \dots (1)$$

### PSNR (Peak Signal to Noise Ratio)

PSNR merupakan nilai perbandingan antara harga maksimum warna pada citra hasil *filtering* dengan kuantitas gangguan (*noise*), yang dinyatakan dalam satuan desibel (dB), *noise* yang dimaksud adalah akar rata-rata kuadrat nilai kesalahan ( $\sqrt{MSE}$ )<sup>[8]</sup>. Secara matematis, nilai PSNR dapat dirumuskan sebagai berikut :

$$PSNR = 20 \log_{10} \left( \frac{255}{\sqrt{MSE}} \right) \dots \dots (2)$$

## HASIL DAN PEMBAHASAN

Berdasarkan hasil rancangan penelitian yang telah dibahas pada metode penelitian, untuk menampilkan dan membahas hasil rancangan, maka dilakukan pengujian. Pengujian dilakukan berdasarkan data gambar yang telah ditentukan sebelumnya pada rancangan penelitian. Berdasarkan tujuan yang ingin dicapai pada penelitian ini hasil pengujian tersebut dianalisis berdasarkan MSE dan PSNR.

### Pengujian Deteksi Tepi Canny Dengan Gaussian Filtering

Sebelum citra diproses untuk mendeteksi tepi, terlebih dahulu citra melewati tahapan *filtering*, proses *filtering* ini bertujuan untuk mengurangi *noise* pada citra. Proses pengujian ini dilakukan terhadap 10 sampel citra. Hasil pengujian dari 10 sampel citra dengan proses *filtering Gaussian* ini disajikan dalam tabel 1.

**Tabel 1.** MSE Proses Gaussian Filtering

Gambar	Pixel	Size	MSE
<i>Lena.bmp</i>	512 x 512	197 KB	308.56
<i>Cameraman.bmp</i>	512 x 512	262 KB	308.73
<i>Fruit.bmp</i>	512 x 512	65 KB	306.49
<i>House.bmp</i>	512 x 512	525 KB	312.68
<i>Jetplane.bmp</i>	512 x 512	525 KB	312.68
<i>Lake.bmp</i>	512 x 512	525 KB	312.68
<i>Livingroom.bmp</i>	512 x 512	262 KB	308.73
<i>Mandrill.bmp</i>	512 x 512	787 KB	315.19
<i>Peppers.bmp</i>	512 x 512	526 KB	312.68
<i>Pirate.bmp</i>	512 x 512	262 KB	308.73
<b>Rata-Rata</b>			<b>310.715</b>

Berdasarkan nilai MSE pada tabel 1, kemudian mengukur nilai perbandingan antara kualitas citra asli sebelum diproses dengan *filtering Gaussian* dan citra setelah diproses dengan *filtering Gaussian*. Hasil perbandingan kualitas citra disajikan dalam tabel 2

**Tabel 2.** PSNR Proses Gaussian Filtering

Gambar	Pixel	Size	PSNR
<i>Lena.bmp</i>	512 x 512	197 KB	65.155
<i>Cameraman.bmp</i>	512 x 512	262 KB	65.155
<i>Fruit.bmp</i>	512 x 512	65 KB	69.402
<i>House.bmp</i>	512 x 512	525 KB	65.215
<i>Jetplane.bmp</i>	512 x 512	525 KB	65.215
<i>Lake.bmp</i>	512 x 512	525 KB	65.215
<i>Livingroom.bmp</i>	512 x 512	262 KB	65.155
<i>Mandrill.bmp</i>	512 x 512	787 KB	62.216
<i>Peppers.bmp</i>	512 x 512	526 KB	65.215
<i>Pirate.bmp</i>	512 x 512	262 KB	65.155
<b>Rata-Rata</b>			<b>65.3098</b>

### Pengujian Deteksi Tepi Canny Dengan Bilateral Filtering

Sebelum citra diproses untuk mendeteksi tepi, terlebih dahulu citra melewati tahapan filtering, proses *filtering* ini bertujuan untuk mengurangi *noise* pada citra. Proses pengujian ini dilakukan terhadap 10 sampel citra. Hasil pengujian dari 10 sampel citra dengan proses *Bilateral Filtering* ini disajikan dalam tabel 3.

**Tabel 3.** MSE Proses Gaussian Filtering

Gambar	Pixel	Size	MSE
<i>Lena.bmp</i>	512 x 512	197 KB	193.138
<i>Cameraman.bmp</i>	512 x 512	262 KB	193.138
<i>Fruit.bmp</i>	512 x 512	65 KB	196.854
<i>House.bmp</i>	512 x 512	525 KB	227.45

<i>Jetplane.bmp</i>	512 x 512	525 KB	227.45
<i>Lake.bmp</i>	512 x 512	525 KB	227.45
<i>Livingroom.bmp</i>	512 x 512	262 KB	193.138
<i>Mandrill.bmp</i>	512 x 512	787 KB	196.854
<i>Peppers.bmp</i>	512 x 512	526 KB	227.45
<i>Pirate.bmp</i>	512 x 512	262 KB	193.138
<b>Rata-Rata</b>			<b>207.606</b>

Berdasarkan nilai MSE pada tabel 3, kemudian mengukur nilai perbandingan antara kualitas citra asli sebelum diproses dengan *Bilateral Filtering* dan citra setelah diproses dengan *Bilateral Filtering*. Hasil perbandingan kualitas citra disajikan dalam tabel 4

**Tabel 4.** PSNR Proses Bilateral Filtering

Gambar	Pixel	Size	PSNR
<i>Lena.bmp</i>	512 x 512	197 KB	68.048
<i>Cameraman.bmp</i>	512 x 512	262 KB	68.048
<i>Fruit.bmp</i>	512 x 512	65 KB	70.035
<i>House.bmp</i>	512 x 512	525 KB	68.426
<i>Jetplane.bmp</i>	512 x 512	525 KB	68.426
<i>Lake.bmp</i>	512 x 512	525 KB	68.426
<i>Livingroom.bmp</i>	512 x 512	262 KB	68.048
<i>Mandrill.bmp</i>	512 x 512	787 KB	65.155
<i>Peppers.bmp</i>	512 x 512	526 KB	68.426
<i>Pirate.bmp</i>	512 x 512	262 KB	68.048
<b>Rata-Rata</b>			<b>68.1086</b>

### Perbandingan Nilai MSE Proses Bilateral Filtering Dan Gaussian Filtering

Proses pengujian *filtering* dilakukan untuk mengurangi *noise*. Proses filtering ini dilakukan untuk menganalisis perbandingan kualitas citra asli sebelum di *filtering* dan citra setelah dilakukan proses *filtering*. Adapun hasil dari pada perbandingan

kualitas citra dari kedua *filter* yang digunakan yaitu, *Gaussian Filtering* dan *Bilateral Filtering*, disajikan kedalam tabel 5 dan 6

**Tabel 5.** Nilai MSE Proses *Bilateral Filtering* Dan *Gaussian Filtering*

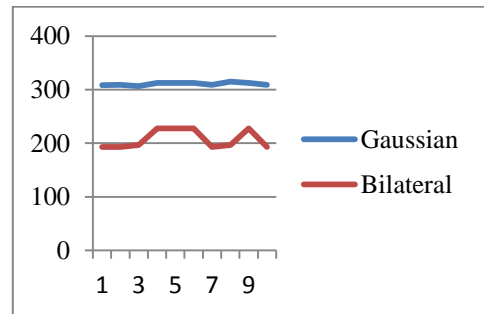
Gambar	Pixel	Size	MSE	
			Bilateral	Gaussian
<i>Lena.bmp</i>	512 x 512	197 KB	308.56	193.138
<i>Cameraman.bmp</i>	512 x 512	262 KB	308.73	193.138
<i>Fruit.bmp</i>	512 x 512	65 KB	306.49	196.854
<i>House.bmp</i>	512 x 512	525 KB	312.68	227.45
<i>Jetplane.bmp</i>	512 x 512	525 KB	312.68	227.45
<i>Lake.bmp</i>	512 x 512	525 KB	312.68	227.45
<i>Livingroom.bmp</i>	512 x 512	262 KB	308.73	193.138
<i>Mandrill.bmp</i>	512 x 512	787 KB	315.19	196.854
<i>Peppers.bmp</i>	512 x 512	526 KB	312.68	227.45
<i>Pirate.bmp</i>	512 x 512	262 KB	308.73	193.138
<b>Rata-Rata</b>			<b>310.715</b>	<b>207.606</b>

**Tabel 6.** Nilai PSNR Proses *Bilateral Filtering* Dan *Gaussian Filtering*

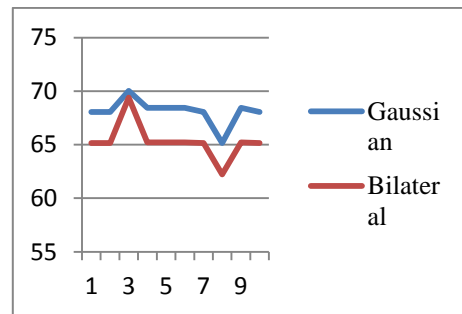
Gambar	Pixel	Size	PSNR	
			Bilateral	Gaussian
<i>Lena.bmp</i>	512 x 512	197 KB	68.048	65.155
<i>Cameraman.bmp</i>	512 x 512	262 KB	68.048	65.155
<i>Fruit.bmp</i>	512 x 512	65 KB	70.035	69.402
<i>House.bmp</i>	512 x 512	525 KB	68.426	65.215
<i>Jetplane.bmp</i>	512 x 512	525 KB	68.426	65.215
<i>Lake.bmp</i>	512 x 512	525 KB	68.426	65.215
<i>Livingroom.bmp</i>	512 x 512	262 KB	68.048	65.155

<i>Mandrill.bmp</i>	512 x 512	787 KB	65.155	62.216
<i>Peppers.bmp</i>	512 x 512	526 KB	68.426	65.215
<i>Pirate.bmp</i>	512 x 512	262 KB	68.048	65.155
<b>Rata-Rata</b>			<b>68.1086</b>	<b>65.3098</b>

Berikut ini grafik perbandingan kualitas citra asli dengan citra setelah diproses dengan *Gaussian filtering* dan *Bilateral Filtering* ditunjukkan pada gambar 1 dan 2



**Gambar 1.** Nilai MSE Proses *Bilateral Filtering* Dan *Gaussian Filtering*



**Gambar 2.** Nilai PSNR Proses *Bilateral Filtering* Dan *Gaussian Filtering*

Dari tabel 5 menunjukkan bahwa rata-rata hasil pengujian *filtering* dengan *Gaussian filtering* menghasilkan nilai MSE dari citra sebesar 310.715 db dan dengan *Bilateral filtering* menghasilkan nilai MSE sebesar 207.606 db. Dari tabel 6 menunjukkan

juga bahwa hasil perbandingan citra asli dengan *Bilateral filtering* ini lebih baik dari pada *Gaussian filtering*, hal ini ditandai dengan nilai rata-rata PSNR dari *Bilateral filtering* hanya 68.1086 db sedangkan PSNR dari *Gaussian filtering* mencapai 65.3098 db.

## KESIMPULAN

Berdasarkan hasil dan pembahasan pengujian deteksi tepi *canny* dengan *Gaussian Filtering* dan *filtering bilateral* dapat disimpulkan bahwa:

1. Untuk mengurangi noise pada deteksi tepi *canny*, metode *Bilateral filtering* lebih baik dari pada metode *Gaussian filtering* hal ini ditandai dengan nilai rata-rata PSNR pada *Gaussian filtering* sebesar 65.3098 db dan *Bilateral filtering* sebesar 68.1086 db
2. Dengan optimalnya *Dual-Threshold* pada deteksi tepi *Canny*, informasi dari citra yang akan didapatkan lebih optimal.
3. *Bilateral filtering* dan *Gaussian Filtering* hanya digunakan untuk penghalusan citra dari noise yang ada pada citra sebelum diproses untuk mendeteksi tepi dengan *Canny*

## DAFTAR PUSTAKA

- [1]Gao, J, & Liu, N. 2012. An improved adaptive threshold canny edge detection algorithm. In *Computer*
- [2]Xiandong, L., Yu, Y., Liu, B., & Li, Z. 2013. Bowstring-based dual-threshold computation method for adaptive Canny edge detector. In *Image and Vision Computing New Zealand (IVCNZ)*: 13-18
- [3]Yuan, K.H., Wei, G., Zhang, Y. D., & Wu, L. N. 2010. An adaptive threshold for the Canny Operator of edge detection. In *Image*

*Analysis and Signal Processing (IASP)*: 371-374

- [4]Canny, J. 1986. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions.* 679-698.
- [5]Saravanan, C. 2010. Color image to grayscale image conversion. In *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on.* IEEE 2:196-199.
- [6]Russo, F. 2010. New method for performance evaluation of grayscale image denoising filters. *IEEE Signal Processing Letters, 17*(5), 417-420.
- [7]HAN, H. Y., & Han, X. 2012. Application of Morphology and Otsu Method in The Canny Edge Detection Operator. *Microelectronics & Computer, 2*, 034.
- [8]Kaushik, P., & Sharma, Y. 2012. Comparison of different image enhancement techniques based upon PSNR & MSE. *International Journal of Applied Engineering Research, 7*(11), 2010-2014.