

---

# Analisa Perbandingan Deteksi Tepi Pada Citra Digital Mengalami Blur Menggunakan *Shen-Castan* Dan *Canny-Deriche*

Dewi Anggraini Puspa Hapsari\*<sup>1</sup>, Widya Khafa Nofa<sup>2</sup>

<sup>1</sup>Program Studi Manajemen Universitas Gunadarma

<sup>2</sup>Program Studi Sistem Informasi Universitas Gunadarma

E-mail: <sup>1</sup>\*dewi.anggraini.puspa@gmail.com, <sup>2</sup>widyakhafa@gmail.com

## Abstrak

Algoritma pada Filter Infinite Impulse Response (FIIR) terdiri dari dua buah yaitu Algoritma Shen-Castan dan Algoritma Canny-Deriche. Dua algoritma ini menggunakan cara yang berbeda untuk melakukan deteksi tepi. Tujuan dari penelitian ini adalah membandingkan algoritma Canny-Deriche dengan algoritma Shen-Castan pada kondisi citra uji yang mengalami gangguan noise dan juga mengalami gangguan efek blur. Hasil pengujian adalah algoritma Canny-Deriche cukup mampu dengan jelas mendeteksi tepi citra walau citra uji mengalami gangguan efek buram dibandingkan dengan algoritma Shen-Castan dimana intensitas tepi objek lebih tinggi dibandingkan dengan intensitas noise yang terfilter.

**Kata Kunci**— Deteksi Tepi, Shen-Castan, Canny-Deriche, Citra Buram

## Abstract

*The algorithm in the Infinite Impulse Response (FIIR) filter consists of two, namely the Shen-Castan Algorithm and the Canny-Deriche Algorithm. These two algorithms use different ways to do edge detection. The purpose of this study is to compare the Canny-Deriche algorithm with the Shen-Castan algorithm under test image conditions that experience noise disturbances and also experience blur effect interference. The test results are the Canny-Deriche algorithm is quite capable of clearly detecting the edges of the image even though the test image has a blurred interference effect compared to the Shen-Castan algorithm where the edge intensity of the object is higher than the intensity of the filtered noise.*

**Keywords**— Edge detection, Shen-Castan, Canny-Deriche, Blurred Image

## 1. PENDAHULUAN

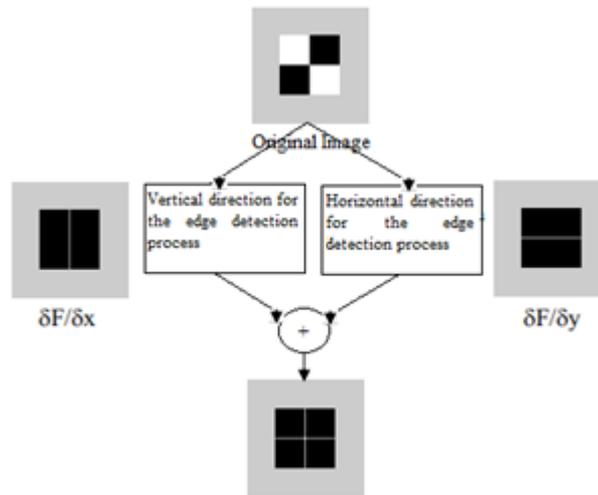
Citra biner adalah citra dengan setiap piksel hanya dinyatakan dengan sebuah nilai dari dua buah kemungkinan (yaitu nilai 0 dan 1). Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Citra jenis ini banyak dipakai dalam pemrosesan citra, misalnya untuk kepentingan memperoleh tepi bentuk suatu objek [1]

Tepi (*edge*) dan kontur (*contour*) merupakan hal penting dalam membantu membantu sistem penglihatan dalam menggambarkan atau merekonstruksi objek. Tepi dalam suatu citra didefinisikan sebagai perbedaan intensitas atau warna antara satu piksel dengan piksel tetangga terdekatnya [6].

Untuk dapat mendeteksi tepi objek pada citra digital harus dilakukan penghilangan atau penghalusan *noise*. *Noise* pada pengolahan citra adalah nilai numerik yang dapat muncul secara acak dan dapat mengakibatkan nilai piksel-piksel dalam sebuah citra mengalami perubahan intensitas warna secara acak. Secara visual, gangguan noise pada citra akan tampak seperti bitnik-bintik dengan intensitas warna yang acak pada piksel-piksel yang saling bertetangga.

Deteksi Tepi (*Edge Detection*) bertujuan untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Karena tepi termasuk ke dalam komponen berfrekuensi tinggi, maka

pendeteksian dilakukan dengan pelapis lolos tinggi. pendeteksian tepi merupakan langkah utama. Berikut adalah gambar proses dari deteksi objek :



**Gambar 1.** Proses dari deteksi objek [2]

Penghilangan atau penghalusan *noise* dikenal dengan pemfilteran. Salah satu jenis filter yaitu IIR (*Infinite Impulse Response*). Filter IIR merupakan filter numerik atau digital yang memiliki sistem feedforward dan feedback sekaligus. Ada beberapa filter IIR yang telah dikembangkan yaitu algoritma Shen-Castan dan algoritma Canny-Deriche.

Intan Dwi Kurniawati, Apriani Kusumawardhani [5] dalam penelitiannya menunjukkan jika Deteksi tepi menggunakan algoritma Canny-Deriche dapat diaplikasikan untuk pengenalan wajah dapat mendeteksi tepi dengan jelas.

Gibtha Fitri Laxmi, Puspa Eosina dan Fety Fatimah [4] dalam penelitiannya menunjukkan bahwa dengan deteksi tepi menggunakan algoritma Canny-Deriche dapat memperjelas konten dari objek citra ikan yang akan diidentifikasi dengan citra uji tanpa gangguan *noise*.

Arief Yudianto, Murinto dan Supomo [2] dalam penelitiannya menunjukkan bahwa hasil dari deteksi tepi menggunakan metode Canny menghasilkan titik tepian yang bersambung cukup jelas sehingga mempermudah dalam uji mutu daun tebakau dengan citra uji ada yang mengalami gangguan *noise*.

Olivier Laligant, Nathalie Reynaud dan Frédéric Truchetet [7] dalam penelitiannya melakukan perbandingan implementasi algoritma Shen-Castan dan algoritma Canny-Deriche pada citra digital.

Penelitian ini melakukan proses pendeteksi tepi pada citra uji yang tidak mengalami *noise* atau juga pada citra uji yang ber*noise*. Akan tetapi ada kondisi dimana citra uji selain mengalami gangguan *noise* juga mengalami gangguan efek buram. Maka tujuan dari penelitian ini adalah membandingkan algoritma Canny-Deriche dengan algoritma Shen-Castan pada kondisi citra uji yang mengalami gangguan *noise* dan juga mengalami gangguan efek blur.

## 2. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah :

1. Studi litelatur, yaitu suatu metode untuk mendapatkan informasi dan melakukan pengumpulan data dengan membaca dan mempelajari berbagai litelatur-litelatur antara lain bersumber dari buku, jurnal, modul, refrensi internet, dan lain-lain yang mana sumber-sumber tersebut berhubungan dengan masalah yang diangkat sehingga dapat membantu dalam meyelesaikan permasalahan yang ada.

2. Metode pengembangan perangkat lunak menggunakan metode SDLC (*Software Development Life Cycle*) dengan model *Waterfall* yaitu: analisa, rancangan, pengkodean, dan implementasi. Perangkat lunak yang dikembangkan adalah MATLAB.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Konsep Algoritma Shen-Castan

Dikembangkan oleh Jun Shen dan Serge Castan dan dipublikasikan pada tahun 1992 untuk tujuan optimalisasi pendeteksian tepi citra yang memiliki noise dan meminimalisasi waktu komputasi serta biaya implementasinya pada rangkaian elektronika.

Filter Shen-Castan terdiri dari satu pasang dan merupakan fungsi 1-Dimensi (1-D). Fungsi matematisnya diberikan pada persamaan (3) dan (4) berikut :

$$f(x) = Ce^{-\alpha|x|} \quad (3)$$

$$\text{jika } x < 0 \quad (4)$$

$$h(x) = \begin{cases} K \cdot e^{-\alpha|x|} \\ -K \cdot e^{-\alpha|x|} \end{cases} \quad \text{jika } x \geq 0$$

dimana :  $C = \frac{1-e^{-\alpha}}{1+e^{-\alpha}}$  dan  $K = 1-e^{-\alpha}$  merupakan konstanta normalisasi dari fungsinya masing-masing.

Fungsi  $f(x)$  berperan sebagai filter *lowpass* yang ditujukan untuk memfilter *noise* dan fungsi  $h(x)$  merupakan filter *bandpass* yang berperan sebagai pendeteksi tepi citra. Parameter  $\alpha$  adalah parameter *noise* yang nilainya dapat diatur sesuai dengan seberapa besar tingkat kekasaran *noise* yang ingin difilter.

Filter  $f(x)$  dan  $h(x)$  dapat dijadikan bentuk filter IIR dengan bantuan transformasi Z, yang nantinya akan diperoleh sinyal keluaran  $y(n)$  sebagai respon impuls dari filter numerik  $f(x)$  terhadap sinyal masukan  $x(n)$  seperti persamaan berikut :

$$\begin{aligned} y^+(n) &= a_1 \cdot x(n) - b_1 \cdot y^+(n-1) && \text{untuk } n = 1, \dots, N \\ y^-(n) &= a_1 \cdot x(n+1) - b_1 \cdot y^-(n+1) && \text{untuk } n = N, \dots, 1 \\ y(n) &= y^+(n) + y^-(n) && (5) \end{aligned}$$

dengan koefisien filter  $a_1 = C = \frac{1-e^{-\alpha}}{1+e^{-\alpha}}$  dan  $b_1 = -e^{-\alpha}$ .

Adapun pseudocode dari Algoritma Shen-Castan adalah :

```

for i=1:N                                %fiter noise arah horizontal
    yp(i,1)=a1*I(1,i);
    yn(i,M)=a1*I(i,M);
    ym(i,M-1)=a1*I(i,M-1)-b1*yn(i,M);
    for j=2:M-1
        yp(i,j)=a1*I(i,j)-b1*yp(i,j);
        yn(i,M-j)=a1*I(i,(M-j)+1)-b1*yn(i,(M-j)+1);
    end
    yp(i,M)=a1*I(i,M)-b1*yp(i,M-1);
end
for i=1:N
    for j=1:M
        Yx(i,j)=yp(i,j)+yn(i,j);
    end
end
    
```

### 3.2. Konsep Algoritma Canny-Deriche

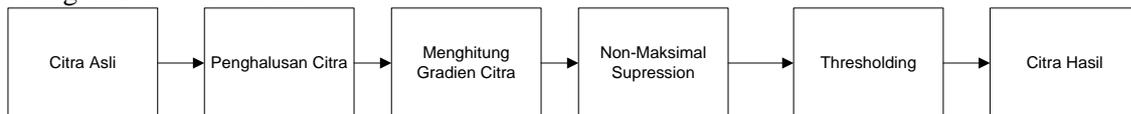
Berdasarkan dari ketiga kombinasi kriteria, maka algoritma Canny-Deriche secara matematis dapat menurunkan dua filter yaitu filter  $h(x)$  sebagai penghalus noise (smoothing filter) dan filter  $f(x)$  sebagai pendeteksi tepi (derivative filter). Kedua filter tersebut dapat ditunjukkan pada persamaan berikut [4] :

$$h(x) = K(\alpha|x|+1)e^{(-\alpha|x|)} \quad (1)$$

$$f(x) = -Cxe^{(-\alpha|x|)} \quad (2)$$

Parameter  $\alpha$  sebagai parameter penentu seberapa halus noise akan difilter.  $K$  dan  $C$  adalah konstanta normalisasi.

Urutan proses pendeteksian tepi menggunakan algoritma Canny-Deriche akan ditunjukkan oleh gambar di bawah ini :



**Gambar 2.** Urutan proses algoritma Canny-Deriche [1]

Langkah awal pada algoritma Canny adalah penghalusan citra yang didapat dari operasi konvolusi, yang merupakan hasil operasi perkalian antara matriks kernel Gaussian dengan matriks citra asli. Persamaan dari matriks kernel Gaussian adalah :

$$G(i,j) = c \cdot e^{-\left( \frac{[(i-u)]^2 + [(j-v)]^2}{[2\delta]^2} \right)} \quad (3)$$

dimana  $G(i, j)$  adalah elemen matriks kernel Gaussian pada posisi  $(i,j)$ ,  $C$  dan  $\sigma$  adalah konstanta dan  $(u, v)$  adalah indeks tengah dari matriks kernel Gaussian. Langkah berikutnya adalah menghitung gradien citra dengan menggunakan operator yang paling mendekati definisi dari sebuah edge, misalnya seperti Prewitt atau Sobel.

Langkah ketiga adalah membangun potensi gradien di suatu piksel dari kandidat edge jika piksel tersebut bukan merupakan maksimal lokal pada arah edge di posisi piksel tersebut untuk mengubah tepi yang blur pada citra gradien menjadi tepi yang tajam. Pada langkah ini membagi garis-garis yang ada menjadi 4 warna terpisah dengan sudut masing-masing, lalu memperkecil masing-masing garis tepi agar menjadi tipis (non maximum suppression). Langkah terakhir adalah melakukan proses binerisasi berdasarkan nilai low dan high threshold yang diberikan.

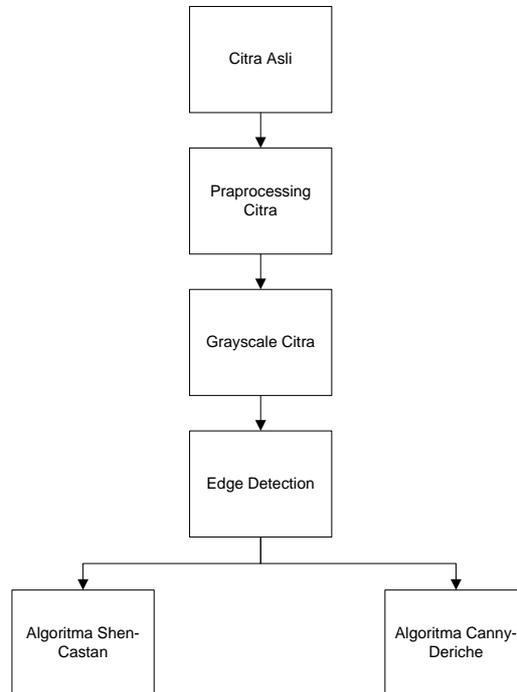
Adapun pseudocode dari algoritma Canny-Deriche adalah :

```

for i=1:N           %fiter noise arah horizontal
    yp(i,1)=a0*I(1,i);
    yp(i,2)= a0*I(i,2)+a1*I(i,1)-b1*yp(i,1);
    yn(i,M)= 0;
    ym(i,M-1)=a1*I(i,M-1)-b1*yn(i,M);
    ym(i,M-2)=a2*I(i,M)+a3*I(i,M)-b1*yn(i,M-1)-
b2*yn(i,M);
    for j=3:M-2
        yp(i,j)=a0*I(i,j)-b1*yp(i,j-1)-b2*yp(i,j-2);
        yn(i,M-j)=a2*I(i,(M-j)+1)+a3*I(i,(M,j)+2)-
b1*yn(i,(M-j)+1)-b2*yn(i,(M-j)+2);
    end
    yp(i,M-1)=a0*I(i,M-1)+a1*I(i,M-2)-b1*yp(i,M-2) -
b2*yp(i,M-3);
    yp(i,M)=a0*I(i,M)+a1*I(i,M-1)-b1*yp(i,M-1)-
b2*yp(i,M-2);
end
    
```

### 3.3. Implementasi

Penelitian ini memiliki tahapan-tahapan sebagai berikut :



**Gambar 3.** Tahapan Penelitian

Citra asli yang digunakan berbentuk jpeg. Adapun citra asli yang digunakan untuk pengujian adalah sebagai berikut :

**Tabel 1.** Citra Asli

Citra Asli	Keterangan Citra
	Nama : Citra_Butterfly.jpg Size : 9.96 KB
	Nama : Citra_Lena.jpg Size : 7,28 KB

Dari citra asli tersebut maka dilakukan proses mengaburkan citra dengan rata-rata piksel dengan delapan piksel tetangganya. Hal ini dimaksudkan untuk memberikan gangguan efek blur (buram) pada citra asli. Proses ini menggunakan perangkat lunak MATLAB (versi trial). Berikut adalah kode program untuk proses tersebut [4] :

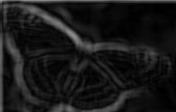
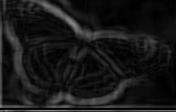
```

    rgbImage = imread('Citra_Lena.jpg'); % Sample image.
    windowWidth = 13; % Whatever you want. More blur for larger
    numbers.
    kernel = ones(windowWidth) / windowWidth ^ 2;
    blurredImage = imfilter(rgbImage, kernel); % Blur the image.
    imshow(blurredImage); % Display it.
    
```

Dari kode program di atas maka akan dihasilkan citra asli yang mengalami gangguan efek blur (buram).

Dan selanjutnya adalah melakukan proses pendeteksian tepi dengan menggunakan algoritma Shen-Castan dan algoritma Canny-Deriche sehingga didapatkan hasil deteksi tepi citra adalah sebagai berikut :

**Tabel 2.** Hasil deteksi tepi

Citra Modifikasi	Nilai $\alpha$	Algoritma	Hasil Deteksi Tepi		
	0,75	Shen-Castan			
		Canny-Deriche			
	1	Shen-Castan			
		Canny-Deriche			
	1,15	Shen-Castan			
		Canny-Deriche			
	0,75	Shen-Castan			
		Canny-Deriche			

	1	Shen-Castan	
		Canny-Deriche	
	1,15	Shen-Castan	
		Canny-Deriche	

Dari tabel 2 di atas maka hasil yang didapat adalah algoritma Canny-Deriche cukup mampu dengan jelas mendeteksi tepi citra walau citra uji mengalami gangguan efek buram dibandingkan dengan algoritma Shen-Castan dimana intensitas tepi objek lebih tinggi dibandingkan dengan intensitas noise yang terfilter.

#### 4. KESIMPULAN

Kesimpulan yang dapat di ambil berdasarkan hasil pengujian adalah algoritma Canny-Deriche cukup mampu dengan jelas mendeteksi tepi citra walau citra uji mengalami gangguan efek buram dibandingkan dengan algoritma Shen-Castan dimana intensitas tepi objek lebih tinggi dibandingkan dengan intensitas noise yang terfilter.

#### 5. SARAN

Adapun saran yang penulis berikan untuk pengembangan penelitian ini selanjutnya adalah penelitian ini tidak menentukan seberapa besar gangguan dari efek blur karena modifikasi untuk citra asli ditentukan secara acak sehingga tidak diketahui secara pasti toleransi algoritma Shen-Castan atau algoritma Canny-Deriche dapat mendeteksi tepi citra. Perlu penelitian lebih lanjut mengenai besarnya SNR dari kedua algoritma dalam mendeteksi tepi citra.

#### DAFTAR PUSTAKA

- [1.] Abdul Kadir, Adhi Susanto, *Pengolahan Citra Teori dan Aplikasi*, Yogyakarta, 2012, p. 31.
- [2.] Arief Yudianto, Murinto dan Soepomo, “Implementasi Metode Canny untuk Deteksi Tepi Mutu Daun Tembakau”, *Jurnal Sarjana Teknik Informatika*, Volume 2 Nomor 3, Oktober 2014, e-ISSN: 2338-5197
- [3.] Dewi Anggraini Puspa Hapsari, Widya Khafa Nova, Sugeng Santoso, “Analisis Kompleksitas Algoritma Filter IRR Shen-Castan untuk Deteksi Tepi pada Citra Digital”, *CCIT Journal Vol 12 No 2 2019 Hal 218-228*, ISSN : 1978 -8282, Online ISSN: 2655-4275
- [4.] Gibtha Fitri Laxmi, Puspa Eosina dan Fety Fatimah, “Analisis Perbandingan Metode Prewitt dan Canny untuk Identifikasi Ikan Air Tawar”, *Prosiding SINTAK 2017*, ISBN: 978-602-8557-20-7
- [5.] Intan Dwi Kurniawati, Apriani Kusumawardhani, “Implementasi Algoritma Canny dalam Pengenalan Wajah menggunakan Antarmuka GUI Matlab”, *Image Processing, IET*, December 2017
- [6.] Madenda, Sarifuddin, *Pengolahan Citra & Video Digital : Teori, Aplikasi, Dan Pemrograman Menggunakan MATLAB*, Erlangga, Jakarta, 2015.
- [7.] Olivier Laligant, Nathalie Reynaud dan Frédéric Truchetet, “Generalization of Shen-Castan and Canny-Deriche filters”, <http://proceedings.spiedigitallibrary.org/> on 05/18/2015 Terms of Use: <http://spiedl.org/terms>
- [8.] Prasetyo, Eko, *Pengolahan Citra Digital dan Aplikasinya menggunakan MATLAB*, Andi, Yogyakarta, 2011.