

# Perbandingan Deteksi Tepi Citra Digital dengan Metode *Prewitt*, *Sobel* dan *Canny*

Sukatmi<sup>1, #</sup>

**Abstrak**— Deteksi tepi merupakan suatu proses yang bertujuan untuk mendeteksi garis tepi yang membatasi dua wilayah citra. Penulisan ini akan membahas tentang perbandingan deteksi tepi citra digital dengan metode *Prewitt*, *Sobel* dan *Canny* menggunakan pemrograman Matlab. Deteksi tepi dilakukan dengan mendeteksi 6 (enam) buah citra digital yaitu *hestain.png*, *onion.png*, *saturn.png*, *tape.png*, *tissue.png* dan *toysflash.png*. Hasil deteksi tepi ditunjukkan dengan banyaknya piksel putih yang perlihatkan dari proses deteksi tepi. Hasilnya menunjukkan bahwa untuk deteksi tepi citra digital metode *Canny* lebih baik dari metode *Sobel* dan metode *Sobel* lebih baik dari metode *Prewitt*.

**Kata kunci**— Deteksi tepi, *Prewitt*, *Sobel*, *Canny*, Matlab

**Abstract**— *Edge detection is a process that aims to detect line the edge that limits two areas image. Writing this will discuss about the comparison detection the edge of digital image with the methods prewitt, sobel and canny with programming Matlab. The detection of the edge of done with detect 6 (six) digital image namely hestain.png, onion.png, saturn.png, tape.png, tissue.png and toysflash.png. Sweep edge indicated by many white pixel show detection edge of the process. The results show that for the detection the image of digital Canny better method of the Sobel and methods Sobel better than Prewitt method.*

**Keywords**— *Edge detection, Prewitt, Sobel, Canny, Matlab*

## I. PENDAHULUAN

Pengolahan citra merupakan bagian penting yang mendasari berbagai aplikasi nyata seperti pengenalan pola, pengindraan jarak jauh melalui satelit atau pesawat udara dan *machine vision*. Pengolahan citra melakukan manipulasi citra menjadi citra yang memiliki kualitas lebih baik agar mudah diinterpretasikan oleh manusia atau mesin (*computer*).

Salah satu metode pengolahan citra yang biasa digunakan adalah segmentasi citra. Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek-objek yang terkandung di dalam citra atau membagi citra kedalam daerah dengan setiap objek atau daerah yang memiliki kemiripan atribut. Segmentasi biasa dilakukan pada langkah awal untuk melaksanakan klasifikasi objek. Pendeteksian tepi suatu citra adalah salah satu pendekatan pengolahan citra yang ada pada segmentasi citra.

Penulisan ini akan membandingkan hasil deteksi tepi dengan menggunakan metode *Prewitt*, *Sobel* dan *Canny* dengan pemrograman Matlab.

Artikel diterima 31 November 2016; direvisi 5 Januari 2017; disetujui 6 Januari 2017; dipublikasikan Februari 2017

<sup>1</sup>Program Studi Manajemen Informatika, AMIK DCC Bandar Lampung, Jl. Cut Nyak Dien No. 65 Durian Payung, Lampung 35119, Indonesia

<sup>#</sup>E-mail: sukatmi@dcc.ac.id

## II. TINJAUAN PUSTAKA

### A. Citra

Citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpan [1].

### B. Pengolahan Citra

Pengolahan citra melakukan manipulasi citra menjadi citra yang memiliki kualitas lebih baik agar mudah diinterpretasikan oleh manusia atau mesin (*computer*). Salah satu metode pengolahan citra yang biasa digunakan adalah segmentasi citra. Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek-objek yang terkandung di dalam citra atau membagi citra kedalam daerah dengan setiap objek atau daerah yang memiliki kemiripan atribut. Segmentasi biasa dilakukan pada langkah awal untuk melaksanakan klasifikasi objek. Pendeteksian tepi suatu citra adalah salah satu pendekatan pengolahan citra yang ada pada segmentasi citra.

### C. Deteksi Tepi

Deteksi tepi (*edge detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari objek-objek gambar. Suatu titik ( $x,y$ ) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangga. Pendeteksian tepi citra berfungsi untuk memperoleh tepi objek. Deteksi tepi memanfaatkan perubahan nilai intensitas yang drastis pada batas dua area. Jika suatu citra jelas dan tajam maka untuk menentukan letak tepi suatu citra akan lebih mudah, namun jika suatu citra tidak jelas dan mendapatkan gangguan seperti adanya *noise* maka akan timbul kesulitan dalam menentukan letak tepi suatu citra. Macam-macam metode deteksi tepi antara lain:

1) *Metode Prewitt*: metode *Prewitt* merupakan pengembangan metode Robert dengan menggunakan filter HPF (*High Pass Filter*) yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF (*High Pass Filter*) [2].

2) *Metode Sobel*: metode *Sobel* merupakan pengembangan metode Robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode *Sobel* ini adalah kemampuan untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi [3]. Biasanya operator *Sobel* menempatkan penekanan atau pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat jendela, sehingga pengaruh piksel-piksel tetangga akan

berbeda sesuai dengan letaknya terhadap titik di mana gradien dihitung. Dari susunan nilai-nilai pembobotan pada jendela juga terlihat bahwa perhitungan terhadap gradien juga merupakan gabungan dari posisi mendatar dan posisi vertikal.

3) *Metode Canny*: Perancangan sebuah prosedur dengan menerapkan langkah-langkah metode Canny edge detection akan menghasilkan sebuah tampilan gambar yang berbeda dengan menampilkan efek relief di dalamnya. Efek relief adalah seperti sebuah tampilan batu kasar yang diukir, yaitu garis-garis kasar yang membentuk sebuah penggambaran objek di dalamnya. Efek relief terbentuk dari bayangan terang dan gelap. Kedua bayangan ini terjadi akibat adanya sorotan sinar mengenai gambar dari arah tertentu. Kelebihan dari metode Canny ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi sehingga tepi-tepi yang dihasilkan lebih banyak. Deteksi Tepi Canny dapat mendeteksi tepian yang sebenarnya dengan tingkat error yang minimum dengan kata lain, operator Canny didesain agar menghasilkan citra tepian yang optimal [4].

### III. METODE PENELITIAN

Metode penelitian yang digunakan adalah penelitian pustaka, yaitu mempelajari konsep-konsep dan teori-teori tentang pengolahan citra digital yang dapat digunakan untuk melakukan pendeteksian tepi citra digital yaitu metode Prewitt, Sobel dan Canny, kemudian membuat program untuk deteksi tepi citra digital dengan menggunakan program Matlab [5].  
*Perangkat Lunak*

### IV. HASIL DAN PEMBAHASAN

Untuk mengetahui tingkat perbedaan deteksi tepi citra dengan menggunakan metode Prewitt, Sobel dan Canny akan dilakukan pembuktian melalui pembuatan program dengan menggunakan Matlab. Akan dilakukan deteksi tepi dengan menggunakan ketiga metode tersebut terhadap 6 (enam buah) gambar (Gbr. 1-Gbr. 6).

#### A. Deteksi Tepi Gambar Hestain.png

```
i= imread('hestain.png');
imshow(i);
I=rgb2gray(i);
BW1 = edge (I, 'canny');
BW2 = edge(I, 'sobel');
BW3 = edge(I, 'prewitt');
subplot(1,4,1);imshow(i);title('Original');
subplot(1,4,2);imshow(BW1);title('Canny');
subplot(1,4,3);imshow(BW2);title('Sobel');
subplot(1,4,4);imshow(BW3);title('Prewitt');
%Hitung Jumlah Pixel Putih Metode Canny
[m,n,o]= size(BW1);
count = 0;
for i = 1 : m;
    for j = 1 : n;
        if BW1(i,j) == 1;
            count = count + 1;
        else,
            end
        end
    end
end
hestain_canny = count;
```

#### B. Deteksi Tepi Gambar Onion.png

```
i= imread('onion.png');imshow(i);
I=rgb2gray(i);
```

```
BW1 = edge (I, 'canny');
BW2 = edge(I, 'sobel');
BW3 = edge(I, 'prewitt');
subplot(1,4,1);imshow(i);title('Original');
subplot(1,4,2);imshow(BW1);title('Canny');
subplot(1,4,3);imshow(BW2);title('Sobel');
subplot(1,4,4);imshow(BW3);title('Prewitt');
%Hitung Jumlah Pixel Putih Metode Canny
[m,n,o]= size(BW1);
count = 0;
for i = 1 : m;
    for j = 1 : n;
        if BW1(i,j) == 1;
            count = count + 1;
        else,
            end
        end
    end
end
onion_canny = count;
```

#### C. Deteksi Tepi Gambar Saturn.png

```
i= imread('saturn.png');imshow(i);
I=rgb2gray(i);
BW1 = edge (I, 'canny');
BW2 = edge(I, 'sobel');
BW3 = edge(I, 'prewitt');
subplot(1,4,1);imshow(i);title('Original');
subplot(1,4,2);imshow(BW1);title('Canny');
subplot(1,4,3);imshow(BW2);title('Sobel');
subplot(1,4,4);imshow(BW3);title('Prewitt');
%Hitung Jumlah Pixel Putih Metode Canny
[m,n,o]= size(BW1);
count = 0;
for i = 1 : m;
    for j = 1 : n;
        if BW1(i,j) == 1;
            count = count + 1;
        else,
            end
        end
    end
end
saturn_canny = count;
```

#### D. Deteksi Tepi Gambar Tape.png

```
i= imread('tape.png');imshow(i);
I=rgb2gray(i);
BW1 = edge (I, 'canny');
BW2 = edge(I, 'sobel');
BW3 = edge(I, 'prewitt');
subplot(1,4,1);imshow(i);title('Original');
subplot(1,4,2);imshow(BW1);title('Canny');
subplot(1,4,3);imshow(BW2);title('Sobel');
subplot(1,4,4);imshow(BW3);title('Prewitt');
% Hitung Jumlah Pixel Putih Metode sobel
[m,n,o]= size(BW2);
count = 0;
for i = 1 : m;
    for j = 1 : n;
        if BW2(i,j) == 1;
            count = count + 1;
        else,
            end
        end
    end
end
tape_sobel = count;
```

#### E. Deteksi Tepi Gambar Tissue.png

```
i= imread('tissue.png');
imshow(i);
```

```

I=rgb2gray(i);
BW1 = edge (I,'canny');
BW2 = edge (I,'sobel');
BW3 = edge (I,'prewitt');
subplot(1,4,1);imshow(i);title('Original');
subplot(1,4,2);imshow(BW1);title('Canny');
subplot(1,4,3);imshow(BW2);title('Sobel');
subplot(1,4,4);imshow(BW3);title('Prewitt')
);
% Hitung Jumlah Pixel Putih Metode Prewitt
[m,n,o]= size(BW3);
count = 0;
for i = 1 : m;
    for j = 1 : n;
        if BW3(i,j) == 1;
            count = count + 1;
        else,
            end
        end
    end
end

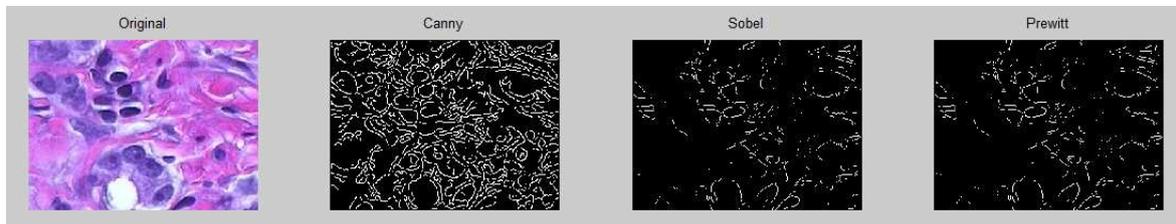
```

#### F. Deteksi Tepi Gambar Toyflash.png

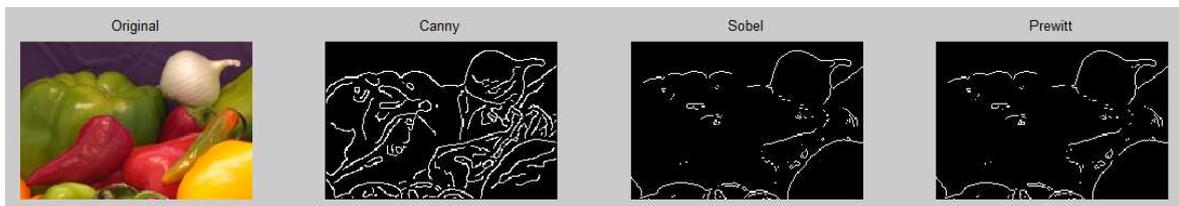
```

i= imread('toysflash.png');
imshow(i);
I=rgb2gray(i);
BW1 = edge (I,'canny');
BW2 = edge (I,'sobel');
BW3 = edge (I,'prewitt');
subplot(1,4,1);imshow(i);title('Original');
subplot(1,4,2);imshow(BW1);title('Canny');
subplot(1,4,3);imshow(BW2);title('Sobel');
subplot(1,4,4);imshow(BW3);title('Prewitt');
%Hitung Jumlah Pixel Putih Metode Canny
[m,n,o]= size(BW1);
count = 0;
for i = 1 : m;
    for j = 1 : n;
        if BW1(i,j) == 1;
            count = count + 1;
        else,
            end
        end
    end
end
toysflash_canny = count;

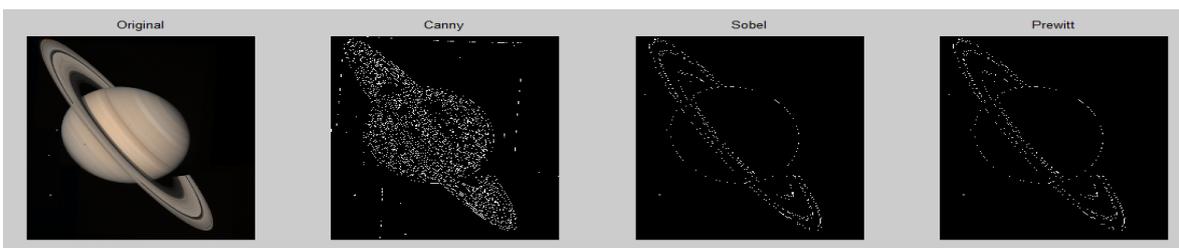
```



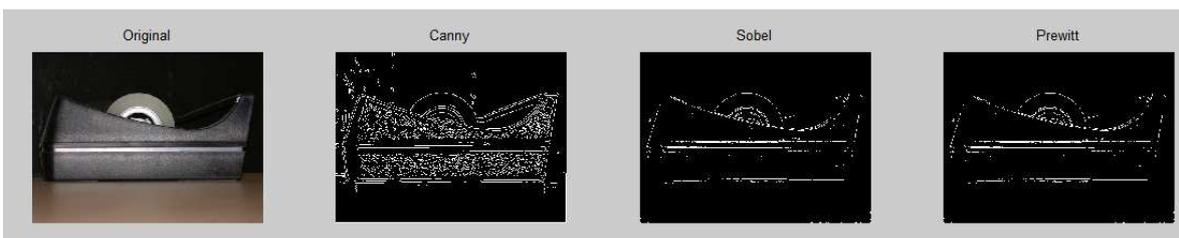
Gbr. 1 Deteksi Tepi Gambar Hestain.png



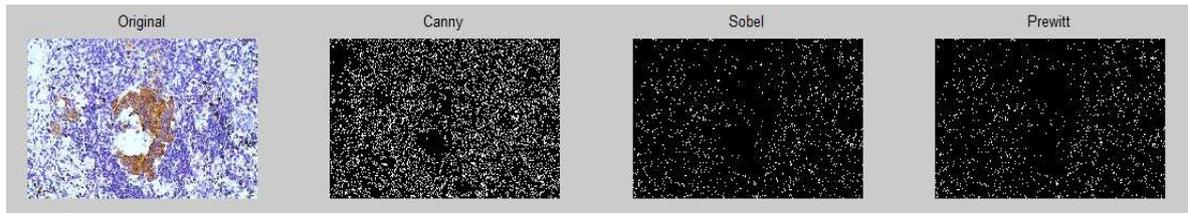
Gbr. 2 Deteksi Tepi Gambar Onion.png



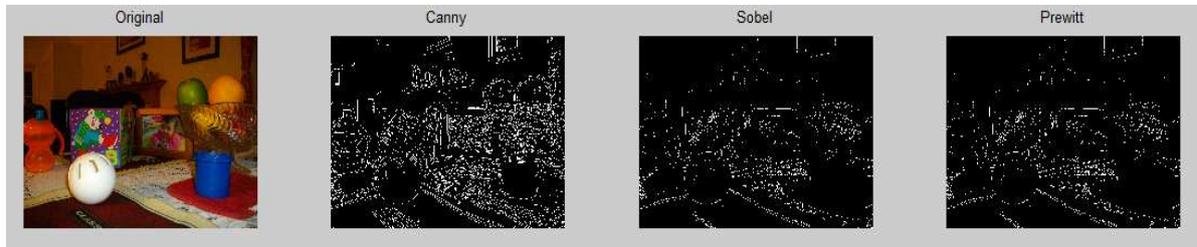
Gbr. 3 Deteksi Tepi Gambar Saturn.png



Gbr. 4 Deteksi Tepi Gambar Tape.png



Gbr. 5 Deteksi Tepi Gambar Tissue.png



Gbr. 6 Deteksi Tepi Gambar Toyflash.png

Jumlah piksel warna putih merupakan piksel yang didapatkan dari hasil deteksi tepi. Semakin banyak piksel warna putih maka semakin banyak pula tepi yang diperoleh. Oleh karena itu, jumlah warna putih yang didapat dari perhitungan masing-masing metode deteksi tepi untuk semua citra yang diuji. Untuk menghitung jumlah piksel warna putih dengan menggunakan fungsi *nnz* (*number of nonzero entries*). Tabel 1 menyajikan jumlah piksel warna putih untuk semua citra sesuai dengan metode deteksi tepi masing-masing.

Berdasarkan Tabel 1 dapat dilihat bahwa hasil dari deteksi tepi memperlihatkan bahwa jumlah piksel berwarna putih dengan menggunakan metode Canny lebih besar dari metode Sobel dan metode Sobel lebih lebih besar dari metode Prewitt.

#### V. KESIMPULAN

Berdasarkan hasil deteksi tepi terhadap 6 (enam) buah citra digital dapat disimpulkan bahwa deteksi tepi dengan menggunakan metode Canny lebih baik dari metode Sobel, dan metode Sobel lebih baik dari metode Prewitt.

#### REFERENSI

- [1] T. Sutoyo, *Teknik Pengolahan Citra Digital*, Yogyakarta: Andi, 2009.
- [2] Septian Dwi Cahyo. *Analisis Perbandingan Tepi Pada Delphi*. Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma.
- [3] Amelia, L., & Marwati, R. *Perbandingan Metode Roberts dan Sobel dalam Mendeteksi Tepi suatu Citra Digital*, Eureka Matika: Jurnal Online Matematika S1, 2013.
- [4] Febriani & ETP Lussiana. *Analisis Penelusuran Tepi Citra Menggunakan Detektor Tepi Sobel Dan Canny*. Proceeding Komputer dan Sistem Intelijen: KOMMIT, 2008.
- [5] Eko Prasetyo, *Pengolahan Citra digital dan Aplikasi Menggunakan Matlab*, Yogyakarta: Andi Publisher, 2013.

TABEL I  
DAFTAR JUMLAH PIKSEL WARNA PUTIH PADA MASING-MASING CITRA

Nama Citra	Jumlah Piksel Warna Putih			Kesimpulan
	Prewitt	Sobel	Canny	
Hestain.png	2056	2074	9945	Canny lebih besar dari Sobel dan Sobel lebih besar dari Prewitt
Onion.png	800	810	2780	Canny lebih besar dari Sobel dan Sobel lebih besar dari Prewitt
Saturn.png	13806	13984	78961	Canny lebih besar dari Sobel dan Sobel lebih besar dari Prewitt
Tape.png	4454	4487	16049	Canny lebih besar dari Sobel dan Sobel lebih besar dari Prewitt
Tissue.png	16607	17128	71776	Canny lebih besar dari Sobel dan Sobel lebih besar dari Prewitt
Toysflash.png	18433	18322	52222	Canny lebih besar dari Prewitt dan Prewitt lebih besar dari Sobel