**JURNAL SISTEM INFORMASI TGD**
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*

# Perbandingan Metode Canny, Sobel, Dan Laplacian of Gaussian Dalam Mendeteksi Tepi Citra Objek Bergerak

**Muhammad Syafwanalif Moelya[1], Puji Sari Ramadhan[2], Mhd. Gilang Suryanata[3]**

[1,2,3] Sistem Informasi, STMIK Triguna Dharma
Email: [1]nalifmoel@gmail.com, [2]pujisariramadhan@gmail.com, [3]suryanatagilang@gmail.com
Email Penulis Korespondensi: nalifmoel@gmail.com

**Abstrak**
Pengolahan citra adalah bagian penting yang mendasari berbagai aplikasi nyata seperti pengenalan pola, pengindraan jarak jauh melalui satelit, pesawat udara, dan *Computer Vision*. Salah satu bagian dari pengolahan citra yang paling awal dan paling banyak diteliti yaitu penentuan tepian atau deteksi tepi. Deteksi tepi pada citra adalah proses yang menghasilkan tepi-tepi dari objek-objek citra yang dilakukan untuk menemukan garis pemisah dari suatu objek pada citra. Terdapat banyak metode yang bisa digunakan untuk melakukan pendeteksian tepi, antara lain metode *Canny*, *Sobel*, dan *Laplacian of Gaussian*. Penelitian ini akan membandingkan penggunaan metode *Canny*, *Sobel*, dan *Laplacian of Gaussian* untuk mendeteksi tepi citra bergerak berbentuk video singkat. Ketiga metode tersebut layak untuk dibandingkan karena masing-masing memiliki kelebihan dan kekurangan dalam pendeteksian tepi citra. Penelitian ini juga digunakan untuk mengetahui metode mana yang paling optimal dalam melakukan pendektesian tepi citra objek bergerak. Objek yang digunakan dalam penelitian ini adalah video bergerak, yang nantinya akan dilakukan pengukuran tingkat pengenalan hasil deteksi tepi dari setiap metode yang akan digunakan. Hasil deteksi tepi algoritma *canny* memiliki hasil deteksi yang paling optimal, dengan hasil tepi yang tebal, detail, dan minim *noise*. Hasil deteksi tepi algoritma *Sobel* menghasilkan tepi yang halus dan putus-putus, serta terdapat banyak *noise*. Hasil deteksi tepi algoritma *Laplacian of Gaussian* memiliki hasil deteksi yang sangat detail dengan garis yang tebal tetapi terdapat beberapa garis yang putus dan terdapat lumayan banyak derau.
**Kata Kunci:** Pengolahan Citra, Deteksi Tepi, *Canny*, *Sobel*, *Laplacian of Gaussian*, Citra Bergerak

***Abstract***
*Image processing is an important part that underlies many real-world applications such as pattern recognition, remote sensing via satellites, aircraft, and Computer Vision. One of the earliest and most researched parts of image processing is edge determination or edge detection. Edge detection in images is a process that produces edges of image objects to find the dividing line of an object in the image. There are many methods that can be used to perform edge detection, including the Canny, Sobel, and Laplacian of Gaussian methods. This research will compare the use of Canny, Sobel, and Laplacian of Gaussian methods to detect the edges of moving images in the form of short videos. The three methods are worth comparing because each has advantages and disadvantages in image edge detection. This research is also used to find out which method is the most optimal in detecting the edges of moving object images. The object used in this research is a moving video, which will later be measured the recognition rate of the edge detection results of each method to be used. The canny algorithm edge detection results have the most optimal detection results, with thick, detailed, and minimal noise edges. The edge detection results of the Sobel algorithm produce smooth and dashed edges, and there is a lot of noise. The Laplacian of Gaussian algorithm edge detection results have very detailed detection results with thick lines but there are some broken lines and there is quite a lot of noise.*
*Keywords: Image Processing, Edge Detection, Canny, Sobel, Laplacian of Gaussian, Moving Image*

## 1. PENDAHULUAN

Salah satu *technology growth* yang berdampak besar pada bidang sistem informasi komputer yaitu pengolahan citra (*Image Processing*). Pengolahan citra melakukan manipulasi citra menjadi citra yang memiliki kualitas lebih baik agar mudah diinterpretasikan oleh manusia atau mesin (komputer) [1].

Deteksi tepi (*Edge Detection*) pada suatu citra dilakukan untuk menemukan garis pemisah dari suatu objek pada citra. Umumnya deteksi tepi adalah proses awal segmentasi citra mengarah dalam identifikasi objek-objek yang terdapat pada citra. Segmentasi citra bagian dari proses pengolahan citra yang merupakan cara untuk membagi suatu citra ke dalam kelompok region yang bertujuan mengisolasi suatu objek pada citra. Objek yang telah tersegmentasi dapat dilakukan proses ekstraksi ciri citra, dimana langkah yang bertujuan untuk membedakan antara objek satu dengan yang lainnya [2].

Penelitian ini akan membandingkan penggunaan metode *Canny*, *Sobel*, dan *Laplacian of Gaussian* untuk mendeteksi tepi citra bergerak (citra *real-time*) berbentuk video singkat. *Canny* adalah metode yang digunakan dengan pendekatan konvolusi fungsi gambar dengan operator *gaussian* dan turunannya. *Sobel* adalah metode *Edge Detection* yang termasuk dalam *Gradient Edge Detector* [3]. *Laplacian of Gaussian* adalah salah satu operator deteksi tepi yang dikembangkan dari turunan kedua. *Laplacian of Gaussian* terbentuk dari proses *Gaussian* yang diikuti operasi *Laplace*. Fungsi *Gaussian* akan mengurangi derau sedangkan *Laplacian Mask* meminimalisasi kemungkinan kesalahan deteksi tepi [4].

Ketiga metode tersebut layak untuk dibandingkan karena masing-masing memiliki kelebihan dan kekurangan dalam pendeteksian tepi citra. Penelitian ini juga digunakan untuk mengetahui metode mana yang paling optimal dalam melakukan pendektesian tepi citra objek bergerak. Objek yang digunakan dalam penelitian ini adalah video bergerak, yang nantinya akan dilakukan pengukuran tingkat pengenalan hasil deteksi tepi dari setiap metode yang akan digunakan.

JURNAL SISTEM INFORMASI TGD
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*

## 2. METODOLOGI PENELITIAN

### 2.1 Pengolahan Citra

*Digital Image Processing* (Pengolahan Citra Digital) adalah bidang ilmu yang mempelajari teknik-teknik mengolah citra, citra yang dimaksud berupa citra gambar diam (foto) ataupun citra gambar bergerak (video). Pengolahan gambar/citra dilakukan secara digital menggunakan komputer, karena komputer tidak bisa langsung mengolah data gambar/citra maka citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit [5].

Pengolahan citra digital dilakukan agar mencapai hasil citra tertentu, seperti mengubah atau meningkatkan kualitas citra agar mudah melakukan pendeteksian pada objek yang menjadi *interest* melalui beberapa proses operasi berikut :

1. *Image Analysis* : Dalambahasa disebut pengorakan citra, proses mengekstrasi ciri-ciri tertentu yang membantu dalam identifikasi objek.
2. *Image Enhancement* : Perbaikan atau modifikasi citra dilakukan untuk meningkatkan kualitas citra
3. *Image Reduction* : Mengurangi ukuran citra atau mereduksi citra dengan tetap mempertahankan ukuran dan kualitas citra.
4. *Image Restoration* : Proses pemulihan atau perbaikan citra ke kondisi semula.
5. *Image Reconstruction* : Teknik rekonstruksi gambar digunakan untuk membuat gambar 2-D dan 3-D dari kumpulan proyeksi 1-D.
6. *Image Segmentation* : Memecah citra menjadi beberapa *segment* dengan kriteria tertentu

### 2.2 Deteksi Tepi

Deteksi tepi (*edge detection*) adalah suatu proses yang dilakukan pada citra untuk mengidentifikasi garis batas (*boundary*) atau tepi-tepi dari suatu objek yang terdapat dalam citra . Deteksi tepi menghasilkan tepi-tepi dari objek dengan membatasi dua wilayah citra yang memiliki intensitas kecerahan yang berbeda [4]. Pertemuan antara bagian objek dan bagian latar belakang disebut tepi objek, tepi sebuah objek berguna untuk memisahkan objek-objek yang saling bersinggungan agak tidak dianggap sebagai satu kesatuan objek yang besar dan tetap dapat dianalisis secara individu.

Dalam proses pendeteksian tepi, algoritma yang umumnya dikembangkan untuk pendeteksian tepi objek melalui langkah-langkah berikut ini [6] :

a. Pengaburan, operasi pengaburan digunakan untuk meningkatkan kinerja dari sebuah algoritma pendeteksian tepi agar yang berkaitan erat dengan *noise*. Semakin kuat operasi pengaburan maka akan semakin lemah pula kekuatan pengaruh *noise*, dan juga semakin lemah keberadaan suatu tepi. Dengan kata lain, usaha mengurangi *noise* dapat berdampak pada hilangnya informasi tepi objek di beberapa lokasi apalagi tepi objek yang keberadaannya memang lemah. Operasi ini akan sangat membantu
b. Penguatan, operasi penguatan dilakukan nilai intensitas piksel-piksel yang terletak di sekitar tepi objek menguat dan piksel-piksel pada area lainnya melemah terlepas rendah atau tingginya intensitas piksel sebelumnya. Melalui operasi penguatan intensitas, operasi pelacakan atau pendeteksian akan jauh lebih mudah dilaksanakan
c. Pelacakan atau pendeteksian, melalui proses *thresholding* atau pengelompokan piksel-piksel adalah salah satu cara untuk menentukan titik-titik mana saja yang merupakan tepi suatu objek. Proses *thresholding* dapat difungsikan untuk pendeteksian tepi berdasarkan kriteria yang digunakan.

#### 2.2.1 Operator Canny

Operator *Canny* termasuk ke dalam salah satu algoritma deteksi tepi modern, pendeteksian tepi dengan metode *Canny* dilakukan dengan cara membaca setiap piksel pada citra dari piksel paling kiri atas (timur utara) dan bergerak ke piksel paling kanan bawah (barat selatan) [7]. Desain operator *Canny* bertujuan untuk menghasilkan citra tepian yang optimal dengan tingkat *error* yang rendah atau minimum [8]. Keunggulan operator deteksi tepi *Canny* jika dibandingkan dengan yang lain :

a. Good Detection
b. Good Location
c. One Respon to Single Edge

Untuk penerapan operator *Canny* pada suatu citra yang akan dideteksi tepinya dilakukan melalui tahapan-tahapan berikut :

a. Konvolusi dengan *Gaussian Filter*
b. Menghitung *Gradient Magnitude*
c. Menentukan Arah Tepian
d. *Non-Maximum Suspression* (Penekanan Non-Maksimum)
e. *Hysteresis Thresholding*

#### 2.2.2 Operator Sobel

Operator *Sobel* adalah cara untuk menghindari *gradien* yang dihitung pada titik interpolasi dari piksel-piksel yang terlibat dengan cara melakukan konvolusi pada citra digital. Operator *Sobel* dikenal memiliki keunggulan untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi, operator ini juga merupakan pengembangan dari operator *Roberts* dengan diberi HPF (*High Pass Filter*) dan diberi satu angka 0 sebagai penyangga [1]. Operator ini

JURNAL SISTEM INFORMASI TGD
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*

merupakan operator yang lebih sensitif terhadap tepian diagonal daripada tepian vertikal dan horizontal, jendela (*kernel*) yang digunakan oleh operator *Sobel* yaitu *kernel* berukuran 3 x 3 piksel [9].

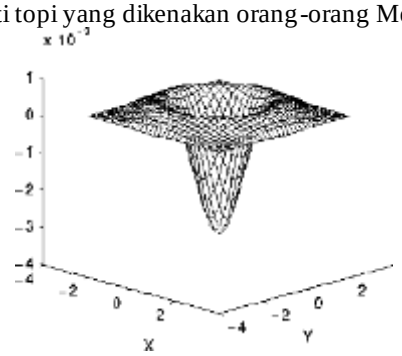
Penerapan algoritma *Sobel* dilakukan melalui tahapan-tahapan berikut :

a. Meninjau Pengaturan Piksel
b. Menghitung *Gradient Magnitude*
c. Menentukan Arah Tepian
d. *Hysteresis Thresholding*

### 2.2.3 Operator Laplacian of Gaussian

*Laplacian of Gaussian* merupakan operator deteksi tepi hasil pengembangan dari operator turunan kedua. Operator ini merupakan hasil kombinasi dari dua operator yaitu operator *Gaussian* dan operator *Laplacian*, operator *Laplacian*/*Laplace* disebut juga sebagai operator turunan kedua. Penggunaan fungsi operator *Gaussian* dimaksudkan untuk memuluskan (konvolusi) citra yang berdampak terhadap pengurangan derau (*noise*) pada citra. Sedangkan untuk *Laplacian mask* berguna untuk meminimalisir kemungkinan kesalahan pendeteksian tepi karena operator *Laplace* memiliki *zero-crossing* yang dapat mendefinisikan lokasi tepi lebih akurat khususnya pada tepi yang curam [10][2][5].

Metode *Laplacian of Gaussian* dilakukan dengan konvolusi lalu menentukan bagian tepi citra dengan metode turunan orde kedua, operator *Laplacian* menghasilkan kepekaan terhadap derau (*noise*) di tiap piksel. Fungsi operator *Laplacian of Gaussian* yang diperoleh melalui konvolusi disebut juga filter topi Meksiko (*the Mexican Hat Filter*) karena bentuk kurvanya mirip seperti topi yang dikenakan orang-orang Meksiko [5].




Gambar 1. *the Mexican Hat Filter*

Berikut ini tahapan-tahapan penerapannya pada pendeteksian tepi citra :

a. Konvolusi dengan *Gaussian Filter*
b. Konvolusi dengan Operator *Laplacian*
c. Menghitung dengan Operator *Laplacian of Gaussian*
d. *Zero-Crossing*

### 2.3 Objek Bergerak

Objek memiliki arti sesuatu yang memiliki wujud, ukuran, warna, dimensi dan ciri bentuk tertentu. Objek mempunyai bentuk dua dimensi dan tiga dimensi, warna objek terdiri dari RGB (*Red*, *Green*, *Blue*) dengan tekstur yang bervariasi. Objek adalah hal yang bisa dijadikan sasaran untuk diteliti, diidentifikasi, dan diperhatikan. Mengidentifikasi objek yang bergerak dalam video merupakan salah satu bagian dari pengolahan citra digital yaitu bagian *Image Processing*. Di ilmu sains fisika, gerak pada dasarnya adalah suatu bentuk perpindahan yang di alami suatu benda sehingga benda tersebut berubah tempat dan arah dari titik awalnya. Sebuah benda dikatakan bergerak jika benda itu berpindah kedudukan terhadap benda lainnya baik perubahan kedudukan yang menjauhi maupun yang mendekati.

Objek bergerak adalah objek yang dapat melakukan perpindahan dari keadaan semula ke keadaan terakhir terhadap suatu acuan tertentu. Oleh karena itu, suatu objek yang bergerak dicirikan oleh kedudukannya dan arah gerakannya. Kemudian objek bergerak tersebut diabadikan menggunakan lensa kamera ke dalam bentuk video dengan format digital *mp4* atau *avi* yang menjadikannya bisa diproses oleh komputer.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Deskripsi Data Penelitian

Data video dalam penelitian ini memiliki durasi yang bervariasi dan tingkat detail yang beragam dari objek yang ada di dalam video. *File* video dikumpulkan dengan format *mp4* atau *avi*, sumber pengumpulan *file* video didapatkan melalui internet dan *Library Windows*. Dari pengumpulan data yang dilakukan diperoleh data video sebagai berikut :

Tabel 1. Data Video

JURNAL SISTEM INFORMASI TGD
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*

| No | Video | Format | Durasi | Size | Frame Rate |
|---|---|---|---|---|---|
| 1 | **walking.mp4**  | mp4 | 15 detik | *Frame* : 640 x 325<br>*File* : 1,72 MB | 25 *frames/ second* |
| 2 | **hewan.avi**  | avi | 9 detik | *Frame* : 720 x 480<br>*File* : 32,9 MB | 29 *frames/ second* |
| 3 | **shibuyacrossing.mp4**  | mp4 | 10 detik | *Frame* : 1280 x 720<br>*File* : 5,81 MB | 25 *frames/ second* |

### 3.1.1 Penerapan Metode Pendeteksian Tepi Citra

a. Ekstrasi *Frame*

Untuk dapat melakukan proses deteksi tepi citra pada video, harus dilakukan proses ekstrasi *frame* terlebih dahulu agar algoritma dapat melakukan proses deteksi tepi pada setiap *frame* atau satu per satu. Berikut merupakan perhitungan untuk mendapatkan jumlah *frame* dalam satu video :

$$Jumlah\ Frame = Durasi\ x\ Frame\ Rate \quad (1)$$

b. Konversi RGB ke *Grayscale*

Bertujuan untuk mengurangi informasi yang dibutuhkan dalam memproses setiap elemen citra, karena warna abu-abu adalah satu warna dalam komponen warna Merah, Hijau dan Biru yang memiliki intensitas yang sama dalam ruang RGB sehingga hanya perlu untuk menentukan satu nilai intensitas untuk setiap elemen citra.

Untuk mencari nilai kanal warna RGB sesuai dengan koordinat yang ingin dijadikan sampel, digunakan beberapa baris perintah berikut :

$$variabel - name = imread('file - name.file - format');$$
$$variabel - name(:,:,:) \quad (2)$$

Untuk menghitung nilai konversi *frame* ke *grayscale* peneliti menggunakan sampel kanal warna RGB yang berasal dari beberapa koordinat x dan y.

Tabel 2. Sampel Kanal Warna RGB pada Titik Koordinat x dan y

| Titik Koordinat | | Kanal Warna | | |
|---|---|---|---|---|
| **x** | **y** | **R** | **G** | **B** |
| 1 | 5 | 226 | 251 | 255 |
| 50 | 100 | 46 | 65 | 98 |
| 100 | 200 | 160 | 187 | 217 |
| 150 | 300 | 206 | 237 | 255 |
| 200 | 400 | 224 | 231 | 249 |
| 250 | 500 | 41 | 52 | 80 |
| 300 | 600 | 101 | 107 | 131 |
| 350 | 700 | 78 | 84 | 108 |
| 400 | 800 | 17 | 21 | 30 |
| 450 | 900 | 23 | 28 | 58 |

Dilakukan konversi *RGB* ke *Grayscale* dengan rumus konversi rata-rata :

$$f_0(x,y) = \frac{fi^R(x,y) + fi^G(x,y) + fi^B(x,y)}{3} \quad (3)$$

Berikut adalah perhitungannya :

$$f_1(1,5) = \frac{fi^R(x,y) + fi^G(x,y) + fi^B(x,y)}{3} = \frac{226 + 251 + 255}{3} = 244$$
$$f_2(50,100) = \frac{46 + 65 + 98}{3} = 69{,}6 \approx 70$$
$$f_3(100,200) = \frac{160 + 187 + 217}{3} = 188$$

**JURNAL SISTEM INFORMASI TGD**
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*

$f_4(150,300) = \frac{206+237+255}{3} = 232{,}6 \approx 233$
$f_5(200,400) = \frac{224+231+249}{3} = 234{,}6 \approx 235$
$f_6(250,500) = \frac{41+52+80}{3} = 57{,}6 \approx 58$
$f_7(300,600) = \frac{101+107+131}{3} = 113$
$f_8(350,700) = \frac{78+84+108}{3} = 90$
$f_9(400,800) = \frac{17+21+30}{3} = 22{,}6 \approx 23$
$f_{10}(450,900) = \frac{23+28+58}{3} = 36{,}3 \approx 36$

Berdasarkan perhitungan konversi rata-rata *RGB* ke *Grayscale* terhadap beberapa sampel titik koordinat x dan y, maka :

Tabel 3. Nilai *Grayscale* Hasil Konversi dari *Frame RGB*

| Titik Koordinat | | Nilai *Grayscale* |
|---|---|---|
| x | y | |
| 1 | 5 | 244 |
| 50 | 100 | 70 |
| 100 | 200 | 188 |
| 150 | 300 | 233 |
| 200 | 400 | 235 |
| 250 | 500 | 58 |
| 300 | 600 | 113 |
| 350 | 700 | 90 |
| 400 | 800 | 23 |
| 450 | 900 | 36 |

c. Penerapan Operator Deteksi Tepi

Baris perintah yang digunakan dapat langsung memproses tiap *frame* yang awalnya *RGB* kemudian dikonversi ke *grayscale* dan langsung dilakukan pendeteksian tepi pada setiap *frame*.

```
4 -   n=1;
5 -   while hasFrame(v)
6 -   img=readFrame(v);
7
8 -     a=rgb2gray(img);
9 -     edg=edge(a,'sobel');
10
11 -  imwrite(edg, strcat('shibuyacrossing_sobel/image_', num2str(n),'.png'));
12 -  n=n+1;
13 -  end
```

Gambar 2. Baris Perintah Pendeteksian Tepi

1. Operator Canny

- Konvolusi dengan *Gaussian Filter*

  Penyaringan citra dengan *Gaussian Filter* yang bertujuan untuk mengeliminasi dearu (*noise*). Salah satu contoh kernel *Gaussian Filter* 5x5 dengan $\sigma = 1{,}4$ [11].

$$\frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

- Menghitung *Gradient Magnitude*

$$G = |G_x| + |G_y| \quad (4)$$

  Untuk membantu menelusuri tepian, gradien *Gx* dan *Gy* dihitung menggunakan metode matriks 3x3 *Canny* Operator [12].

$$Cx = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad Cy = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Menentukan Arah Tepian

$$\theta = tan^{-1}\left(\frac{G_x}{G_y}\right) \quad (5)$$

- *Non-Maximum Suspression* (Penekanan Non-Maksimum)

**JURNAL SISTEM INFORMASI TGD**
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*

Meminimalkan garis/arah tepi yang muncul atau dapat dilacak yang menghasilkan garis tepian yang lebih ramping.

- *Hysteresis Thresholding*
  Untuk mengeliminasi tepi-tepi yang rusak.
  $T_{min}\,(Threshold\ Nilai\ Rendah)$ & $T_{max}\,(Threshold\ Nilai\ Tinggi)$

$$g(x,y) = \begin{cases} 1, jika\ f(x,y) \geq 128 \\ 0, jika\ f(x,y) < 128 \end{cases} \quad (6)$$

2. Operator Sobel
   - Meninjau Pengaturan Piksel
     Tinjau pengaturan piksel di sekitar pixel (x,y) yang digunakan Operator *Sobel* [13].

$$\begin{bmatrix} a0 & a1 & a2 \\ a7 & (x,y) & a3 \\ a6 & a5 & a4 \end{bmatrix}$$

   - Menghitung *Gradient Magnitude*
     Untuk menghitung *Gradient Magnitude* dari Operator *Sobel* digunakan rumus [14]:

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \quad (7)$$
$$S_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \quad (8)$$
$$M = \sqrt{Sx^2 + Sy^2} \text{ atau } M = |S_x| + |S_y| \quad (9)$$

   - Menentukan Arah Tepian
   - *Hysteresis Thresholding*
3. Operator Laplacian of Gaussian
   - Konvolusi dengan *Gaussian Filter*
   - Konvolusi dengan Operator *Laplacian*
     Melakukan proses konvolusi menggunakan operator konvolusi *Laplacian* dengan kernel :

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ dan } \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \text{ atau } \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

   - Menghitung dengan Operator *Laplacian of Gaussian*
     Menghitung dengan operator *Laplacian of Gaussian* dengan rumus :

$$LoG = \frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (10)$$

   - *Zero-Crossing*
     Ada fungsi turunan kedua, yang dimana memang nilainya 0. Tetapi ada perubahan nilai yang sangat tajam dari positif ke negatif, itulah yang disebut dengan *Zero-Crossing*.

d. Penyatuan *Frame* Menjadi Video
Setelah seluruh *Frame* dari hasil ekstrasi telah melalui proses pendeteksian tepi berdasarkan operator, maka untuk dapat menampilkan hasil deteksi tepi dilakukan penyatuan atau pembuatan video terhadap seluruh *frame* hasil deteksi tepi.

```
22
23 -    video = VideoWriter('shibuyacrossing_sobel.avi' );
24
25 -    open( video );
26 -  for i=1:n
27 -      img=imread(strcat('shibuyacrossing_sobel/image_',num2str(i),'.png'));
28 -      img=im2double(img);
29
30 -      writeVideo( video, img );
31 -  end
32 -    close(video);
```

Gambar 3. Perintah Menyatukan Frame Menjadi Video

e. Menampilkan Video Hasil Pendeteksian Tepi Citra Objek Bergerak
Dari keseluruhan proses pada kerangka kerja yang telah dijelaskan menunjukkan bahwa dengan menggunakan metode *Canny*, *Sobel*, dan *Laplacian of Gaussian*, tepi objek yang bergerak dapat terdeteksi.

JURNAL SISTEM INFORMASI TGD
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*

Gambar 4. Sampel Hasil Deteksi Tepi Citra Objek Bergerak

Untuk dapat menampilkan video hasil deteksi tepi pada *movie player* digunakan baris perintah berikut :

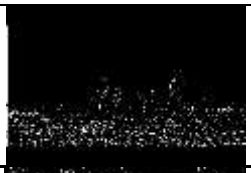
```
34
35 -    implay('video');
36
```

Gambar 5. Perintah Menampilkan Hasil Pada Movie Player

f. Membandingkan Hasi Deteksi Operator Pendeteksian Tepi

Dapat disimpulkan bahwa operator *Canny* merupakan operator deteksi tepi dengan hasil pendeteksian yang paling optimal dengan keakuratan yang hampir serupa dengan objek asli. Untuk operator *Sobel*, mempunyai kekurangan pada hasil garis tepi objek yang dideteksi halus dan putus-putus, sedangkan untuk operator *Laplacian of Gaussian* memiliki kekurangan pada pendeteksian *background objects* yang ikut terdeteksi juga banyaknya *noise* pada hasil pendeteksian. Maka operator *Canny* adalah operator deteksi tepi yang paling optimal dan lebih unggul dibandingkan dengan operator *Sobel* dan *Laplacian of Gaussian*.

Tabel 4. Perbandingan Hasil Pendeteksian Tepi Tiap Operator

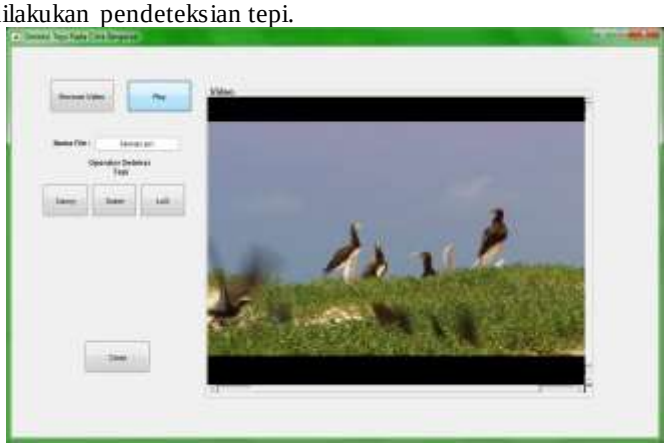
| **Citra Awal** | ***Canny*** | ***Sobel*** | ***Laplacian of Gaussian*** |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 3.2 Implementasi

Hasil dari penelitian yang telah dilakukan yaitu perbandingan metode Canny, Sobel, dan Laplacian of Gaussian dalam mendeteksi tepi citra objek bergerak. Didapatkan perbandingan hasil pendeteksian tepi objek citra bergerak yang digunakan untuk menentukan operator/metode pendeteksian tepi yang paling optimal.

a. Tampilan Antarmuka Program

*Figure* Deteksi Tepi Pada Citra Objek Bergerak disediakan untuk melakukan proses terhadap citra bergerak (video) yang akan dilakukan pendeteksian tepi.

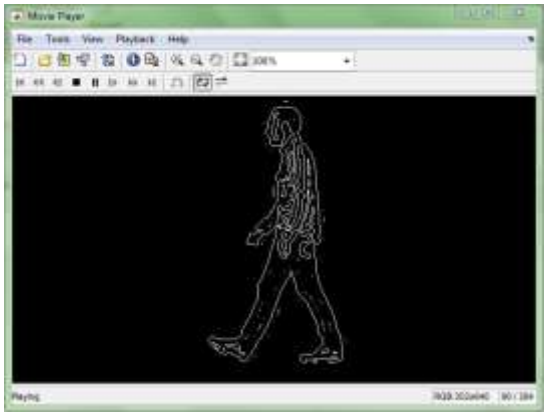


Gambar 6. Tampilan Antarmuka Program

b. Tampilan Movie Player Hasil Pendeteksian Tepi

Hasil keluaran dari *figure* di atas adalah *movie player* yang menampilkan video hasil pendeteksian tepi citra objek bergerak.

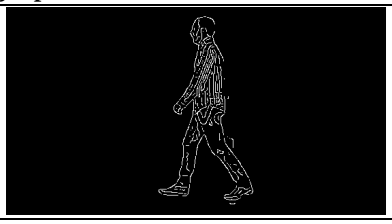
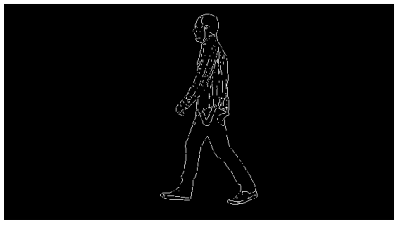
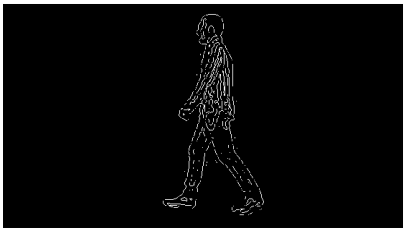
**JURNAL SISTEM INFORMASI TGD**
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*



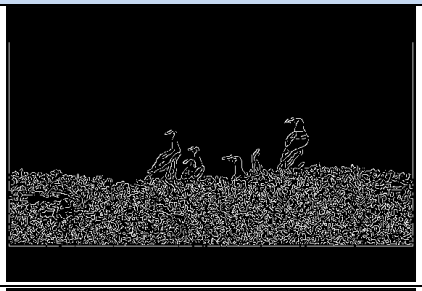
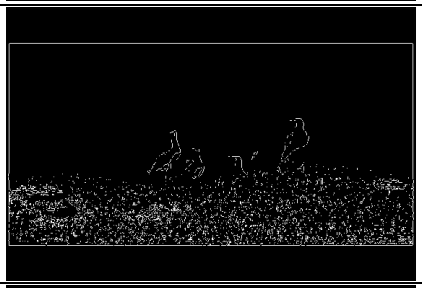
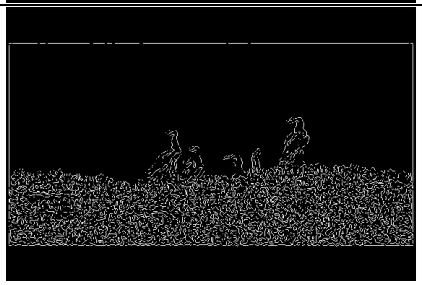
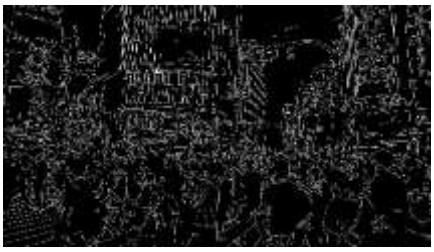
Gambar 7. Tampilan Movie Player Hasil Pendeteksian Tepi

c. *Black Box Testing*
Hasil yang diperoleh dari pengujian digunakan untuk membandingkan seluruh video hasil deteksi tepi operator *Canny*, *Sobel*, dan *Laplacian of Gaussian* yang telah dibuat apakah dapat mendeteksi tepi citra objek bergerak yang melintas dalam video dengan baik atau tidak.

Tabel 5. *Black Box Testing*

| No | Operator Deteksi Tepi | *Test Case* | Hasil Pengujian | Keterangan |
|---|---|---|---|---|
| | **File : "*walking.mp4*"** | | | |
| 1 | *Canny* |  | Operator *Canny* berhasil mendeteksi tepi objek yang bergerak pada video *walking.mp4* dengan hasil yang detail dengan garis tepi yang tebal | Valid |
| | *Sobel* |  | Operator *Sobel* berhasil mendeteksi tepi objek yang bergerak pada video *walking.mp4* dengan hasil yang cukup detail dan minim derau yang mengindikasikan operator ini bekerja dengan sangat baik untuk mendeteksi tepi objek yang tidak rumit dan mendetail | Valid |
| | *Laplacian of Gaussian* |  | Operator *Laplacian of Gaussian* berhasil mendeteksi tepi objek yang bergerak pada video *walking.mp4* dengan hasil yang baik dan detail meskipun beberapa elemen *background* yang serharusnya tidak terdeteksi malah ikut terdeteksi yang menandakan bahwa operator ini bekerja untuk mendeteksi tepi dengan sangat mendetail dan rapi | Valid |
| | | | | |
| | **File : "*hewan.avi*"** | | | |

**JURNAL SISTEM INFORMASI TGD**
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*

| No | Operator Deteksi Tepi | *Test Case* | Hasil Pengujian | Keterangan |
|---|---|---|---|---|
| 2 | *Canny* |  | Operator *Canny* berhasil mendeteksi tepi objek yang bergerak pada video *hewan.avi* dengan hasil yang jelas dan garis tepi yang tebal | Valid |
| | *Sobel* |  | Operator *Sobel* berhasil mendeteksi tepi objek yang bergerak pada video *hewan.avi* meskipun dengan hasil tepi yang putus-putus | Valid |
| | *Laplacian of Gaussian* |  | Operator *Laplacian of Gaussian* berhasil mendeteksi tepi objek yang bergerak pada video *hewan.avi* dengan hasil beberapa tepi yang kurang detail meskipun tepi yang terdeteksi cukup tebal, juga lebih halus dari operator *Sobel*. Terdapat derau (*noise*) | Valid |
| | | | | |
| | **File : "*shibuyacrossing.mp4*"** | | | |
| 3 | *Canny* |  | Operator *Canny* berhasil mendeteksi tepi objek yang bergerak pada video *shibuyacrossing.mp4* dengan hasil yang cukup detail dan garis tepi yang tebal | Valid |
| | *Sobel* |  | Operator *Sobel* berhasil mendeteksi tepi objek yang bergerak pada video *shibuyacrossing.mp4* meskipun dengan hasil tepi yang putus-putus, tidak detail dan terdapat derau | Valid |

JURNAL SISTEM INFORMASI TGD
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*

| No | Operator Deteksi Tepi | *Test Case* | Hasil Pengujian | Keterangan |
|---|---|---|---|---|
| | *Laplacian of Gaussian* |  | Operator *Laplacian of Gaussian* berhasil mendeteksi tepi objek yang bergerak pada video *shibuyacrossing*.mp4. Dengan hasil tepi yang kurang detail meskipun tepi yang terdeteksi cukup tebal, jauh lebih detail dibandingkan operator *Sobel*. Bagian *background* dari objek berhasil terdeteksi tetapi mengandung sedikit derau (*noise*) | Valid |

d. Tabel Analisa Pengujian

Dari tabel analisa dengan menggunakan teknik *Black Box Testing* pada operator *Canny*, *Sobel*, dan *Laplacian of Gaussian*, dapat dibuat perbandingan antara ketiga algoritma tersebut dengan tiga video bergerak yang dilakukan pendeteksian tepi bahwa video dengan algoritma *Canny* dapat menghasilkan deteksi tepi yang paling optimal dibandingkan kedua algoritma lainnya, tepi yang dihasilkan lebih halus dan hampir serupa dengan aslinya dibandingkan dengan algoritma *Sobel* dan *Laplacian of Gaussian* [3].

Tabel 6. Tabel Analisa Pengujian

| Kriteria Analisa | Algoritma Deteksi Tepi *Canny* | Algoritma Deteksi Tepi *Sobel* | Algoritma Deteksi Tepi *Laplacian of Gaussian* |
|---|---|---|---|
| *Edge Detection* (Deteksi Tepi) | • Berhasil mendeteksi tepi objek bergerak dengan optimal, menghasilkan tepi objek yang tebal dan cukup detail<br>• Hampir tidak ada *noise* yang ikut terdeteksi<br>• Hampir keseluruhan bagian objek terdeteksi, hanya detail yang kecil dan rumit yang tidak terdeteksi | • Berhasil mendeteksi tepi objek bergerak tetapi dengan hasil garis tepi yang halus dan putus-putus, banyak objek yang tidak terdeteksi<br>• Terdapat banyak *noise* yang ikut terdeteksi<br>• Banyak bagian objek yang menghilang dan tidak terdeteksi, ditambah banyak tepi yang putus-putus<br>• Bekerja lebih baik untuk mendeteksi objek bergerak yang simpel dan tunggal | • Berhasil mendeteksi tepi objek bergerak dengan hasil garis tebal meskipun putus-putus, objek yang dapat terdeteksi juga sangat mendetail<br>• Terdapat banyak *noise* yang ikut terdeteksi<br>• Hampir keseluruhan bagian objek terdeteksi bahkan detail yang cukup rumit dan curam dapat terdeteksi<br>• Dapat mendeteksi dengan sangat detail hingga *background* yang tidak seharusnya terdeteksi malah ikut terdeteksi |

## 4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan terhadap hasil pendeteksian tepi objek bergerak, diketahui bahwa perbandingan keakuratan pendeteksian tepi antara algoritma *Canny*, *Sobel*, dan *Laplacian of Gaussian* berhasil menghasilkan tepi yang sesuai dengan objek bergerak yang dideteksi. Hasil deteksi tepi algoritma *Canny* memiliki hasil deteksi yang paling optimal daripada kedua algoritma lainnya, dengan hasil tepi yang tebal, detail, dan minim *noise*. Hasil deteksi tepi algoritma *Sobel* menghasilkan tepi yang halus dan putus-putus, serta terdapat banyak *noise*. Tetapi untuk melakukan pendeteksian tepi terhadap objek yang simpel, algoritma *Sobel* memiliki hasil yang hampir mirip dengan *Canny*. Hasil deteksi tepi algoritma *Laplacian of Gaussian* memiliki hasil deteksi yang sangat detail dengan garis yang tebal tetapi terdapat beberapa garis yang putus dan terdapat lumayan banyak derau. Dari ketiga algoritma pendeteksian tepi yang digunakan dalam penelitian ini, algoritma *Canny* memiliki tingkat akurasi yang paling optimal karena tepi yang dihasilkan tebal dan minim *noise*.

Pergerakan objek dan spesifikasi video seperti resolusi, *size*, *format*, variasi objek, detail objek, dan *frame rate* berpengaruh besar terhadap hasil deteksi, dengan penjelasan tersebut maka dibutuhkan algoritma pendeteksian tepi yang tepat.

**JURNAL SISTEM INFORMASI TGD**
**Volume 3, Nomor 4, Juli 2024, Hal 450-460**
P-ISSN : 2828-1004 ; E-ISSN : 2828-2566
*https://ojs.trigunadharma.ac.id/index.php/jsi*

## UCAPAN TERIMAKASIH

Saya ucapakan terima kasih kepada Bapak Puji Sari Ramadhan, S.Kom., M.Kom. selaku Pembimbing I dan Bapak Mhd. Gilang Suryanata, S.Kom., M.Kom. selaku Pembimbing II yang telah memberikan bimbingan dan usaha yang terbaik dalam penelitian ini. Dan juga saya berterima kasih kepada kedua orang tua, saudara, dan teman-teman yang selalu ada untuk membantu saya dalam menyelesaikan penelitian ini. Terima kasih kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini. Besar harapan saya jurnal ini dapat bermanfaat bagi para pembacanya.

## DAFTAR PUSTAKA

[1] S. Sukatmi, "Perbandingan Deteksi Tepi Citra Digital dengan Menggunakan Metode Prewitt, Sobel dan Canny," *KOPERTIP J. Ilm. Manaj. Inform. dan Komput.*, vol. 1, no. 1, pp. 1–4, 2017, doi: 10.32485/kopertip.v1i1.3.

[2] S. Aripin, L. Sarumaha, and M. N. Sinaga, "Implementasi Metode Laplacian of Gaussian Dalam Deteksi Tepi Citra Gigi Berlubang," *Sainteks*, pp. 393–396, 2020.

[3] W. Supriyatin, "Perbandingan Metode Sobel, Prewitt, Robert dan Canny pada Deteksi Tepi Objek Bergerak," *Ilk. J. Ilm.*, vol. 12, no. 2, pp. 112–120, 2020, doi: 10.33096/ilkom.v12i2.541.112-120.

[4] B. Sinaga, J. Manurung, M. H. Silalahi, and S. Ramen, "Deteksi Tepi Citra Dengan Metode Laplacian of Gaussian Dan Metode Canny," vol. 5, no. September, pp. 1066–1084, 2021.

[5] J. Nababan, "Pendeteksian Tepi Citra untuk Menentukan Kualitas Beras Berdasarkan Jenis Menggunakan Metode Laplacian of Gaussian," vol. 1, no. 1, pp. 13–19, 2020.

[6] U. Ahmad, *PENGOLAHAN CITRA DIGITAL DAN TEKNIK PEMROGRAMANNYA*, Edisi Pert. Yogyakarta: GRAHA ILMU, 2017.

[7] A. Zalukhu, "Implementasi Metode Canny Dan Sobel Untuk Mendeteksi Tepi Citra," *J. Ris. Komput.*, vol. 3, no. 6, pp. 25–29, 2016.

[8] M. M. Sobel, R. Canny, P. Teguh, K. Putra, N. Kadek, and A. Wirdiani, "Pengolahan Citra Digital Deteksi Tepi Untuk Membandingkan Metode Sobel, Robert dan Canny," *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, vol. 2, no. 2, pp. 253–261, 2016.

[9] J. W. Yodha and A. W. Kurniawan, "Perbandingan Penggunaan Deteksi Tepi Dengan Metode Laplace , Sobel Dan Prewit Dan Canny Pada Pengenalan Pola," *Techno.COM*, vol. 13, no. 3, pp. 189–197, 2014.

[10] E. D. Prasetyo, "Deteksi Tepi Menggunakan Metode Laplacian of Gaussian Pada Citra Bola Futsal," vol. 1, no. 6, pp. 309–316, 2020.

[11] G. Gunawan *et al.*, "Mobile Application Detection of Road Damage using Canny Algorithm," *J. Phys. Conf. Ser.*, vol. 1019, no. 1, 2018, doi: 10.1088/1742-6596/1019/1/012035.

[12] S. B. Utomo, J. F. Irawan, and R. R. Alinra, "Early warning flood detector adopting camera by Sobel Canny edge detection algorithm method," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 3, pp. 1796–1802, 2021, doi: 10.11591/ijeecs.v22.i3.pp1796-1802.

[13] L. Multimedia, "PENGOLAHAN CITRA DIGITAL Disusun Oleh D r . Ir . H . Sumijan , M . Sc PENGOLAHAN CITRA Universitas Putra Indonesia " YPTK " Padang."

[14] V. H. Saputra, D. E. Herwindiati, and T. Sutrisno, "Car shape clustering using sobel edge detection with divisive average linkage and single linkage algorithm (case: Bus, sedan, city car, mpv, and truck)," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1007, no. 1, 2020, doi: 10.1088/1757-899X/1007/1/012136.