Analisis Kompleksitas Algoritma Filter IRR Shen-Castan untuk Deteksi Tepi pada Citra Digital

Dewi Anggraini Puspa Hapsari[1], Widya Khafa Nofa[2], Sugeng Santoso*[3]
*Penulis Korespondensi
[1]Jurusan Manajemen, Fakultas Ekonomi, Universitas Gunadarma
[2]Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Universitas Gunadarma
[3]Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Raharja
E-mail: *[1]dewi.anggraini.puspa@gmail.com, [2]widyakhafa@gmail.com,
[3]**sugeng.santoso@raharja.info**

Abstrak

Penelitian ini bertujuan untuk mengukur tingkat kompleksitas sebuah algoritma filter IIR Shen-Castan untuk deteksi tepi pada citra digital berdasarkan kompleksitas waktu (time complexity) dan kompleksitas ruang (space complexity). Analisis ini sangat dibutuhkan pada saat akan melakukan sebuah optimasi-optimasi algoritma dan struktur data yang akan digunakan dalam sebuah sistem. Kompleksitas waktu (time complexity) menyatakan berapa lama suatu algoritma berjalan ketika runtime berdasarkan input yang diberikan. Sedangkan kompleksitas ruang (space complexity) menyatakan berapa banyak ruang dalam memori yang dibutuhkan suatu algoritma ketika beroperasi. Perhitungan time complexity dan space complexity dinotasikan dengan Big-O notation. Dari hasil perhitungan time complexity didapatkan jika algoritma filter IIR Shen-Castan mempunyai kompleksitas algoritma $O(n^2)$ dan membutuhkan space complexity $O(n^2)$ dimana algoritma tersebut akan membutuhkan setidaknya N unit ruang di memori dan bergantung pada panjang array-nya.

Kata Kunci*—Kompleksitas Algoritma, Filter IIR Shen-Castan, Deteksi Tepi, Citra Digital*

Abstract

This study aims to measurethe level of complexity of a Shen-Castan IIR filter algorithm for edge detection in digital images based on time complexity and space complexity. This analysis is very helpful when going to do an algorithm optimization and data structure that will be used in a system. Time complexity states how long an algorithm runs at runtime based on the input given. While space complexity states how much space in memory an algorithm needs when operating. The calculation of time complexity and space complexity is denoted by Big-O notation. From the results of time complexity calculations obtained if the Shen-Castan IIR filter algorithm has the complexity of the O (n) algorithm and requires space complexity O (n) where the algorithm will need at least N units of space in memory and depends on the length of the array.

Keywords*—Complexity Algorithm, Filter IIR Shen-Castan, Edge Detection, Digital Image*

1. PENDAHULUAN

Dalam sistem penglihatan manusia, tepi (*edge*) dan kontur (*contour*) dari sebuah objek memberi peranan penting dikarenakan keduanya dapat membantu sistem penglihatan dalam menggambarkan atau merekonstruksi objek. Pada bidang *computer vision*, pendeteksian tepi dan kontur merupakan salah satu pengolahan dasar dari citra dan sangat penting. Walau memiliki definisi yang berbeda namun dalam prakteknya kadang diartikan sama.

Madenda [5] mendefisikan tepi (*edge*) pada citra sebagai perbedaan intensitas atau warna antara satu piksel dengan piksel tetangga terdekatnya. Sedangkan Munir [6] mengartikan

tepi (*edge*) pada citra sebagai perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Sehingga dapat diartikan jika tepi (*edge*) pada citra adalah perubahan nilai intensitas derajat keabuan antara satu piksel dengan piksel tetangga terdekatnya yang sangat besar. Maka semkain tinggi tingkat perbedaan itu maka akan semakin tampak jelas tepi tersebut.

Untuk definisi kontur (*contour*), Madenda [5] mengartikan sebagai titik-titik tepi yang saling terhubung dan membentuk dan membentuk garis batas antara dua area yang berbeda. Dapat dikatakan jika kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel-piksel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi-tepi objek di dalam citra. Berdasarkan dari pengertian di atas maka sering dikatakan bahwa pendeteksian tepi objek sama dengan pendeteksian kontur objek pada citra.

Tujuan dari pendeteksian tepi adalah untuk meningkatkan penampakan batas suatu area atau objek pada sebuah citra. Tepi dapat mencirikan batas-batas objek yang berguna untuk segmentasi dan identifikasi objek dalam citra. Pendeteksian tepi termasuk ke dalam komponen berfrekuensi tinggi maka dapat dilakukan dengan penapis lolos-tinggi (*high-pass filter*).

Untuk dapat mendeteksi tepi objek pada citra digital harus dilakukan penghilangan atau penghalusan dearau (*noise)*. *Noise* yang dimaksud adalah *noise* numerik yang hadir secara acak pada saat pengolahan citra sehingga dapat mengakibatkan nilai piksel mengalami perubahan intensitas warna secara acak pula. Secara visual, gangguan *noise* pada citra dapat berbentuk bintik-bintik dengan intensitas yang acak pada piksel-piksel yang saling bertetangga dan pada umumnya hadir pada area yang berintensitas rendah dan area yang memiliki warna seragam (*uniform*). Akibat dari gangguan *noise* ini menyebabkan tepian objek menjadi tidak jelas dan tidak beraturan sehingga dapat mengakibatkan pergeseran posisi tepi. Proses penghilangan atau penghalusan *noise* dikenal dengan pemfilteran.

Secara garis besar, pada pengolahan citra digital dikenal dua jenis filter yaitu FIR (*Finite Impulse Response*) dan IIR (*Infinite Impulse Response*). FIR adalah filter yang memiliki respon impuls terbatas yang artinya hanya memiliki bagian umpan maju (*feedfoward*) dan tidak memiliki sistem umpan balik (*feedback*). Sedangkan filter IIR merupakan sebuah filter numerik yang memiliki sistem *feedfoward* dan *feedback* bersamaan, yang berarti bahwa keluaran dari filter ini tidak hanya bergantung pada data sesaat dan data sebelumnya tetapi juga bergantung pada hasil perhitungan sebelumnya.

Vitiarisni dkk [9] melakukan penelitian yang bertujuan untuk mengetahui kinerja FIR dan IIR serta membandingkan secara kuantitatif dan kualitatif pada sinyal EKG. Hasilnya adalah kinerja filter IIR lebih baik jika dibandingkan dengan filter FIR. Sehingga pada penelitian ini digunakanlah analisis kompleksitas algoritma filter IIR.

Ada beberapa filter IIR yang telah dikembangkan dalam teori pengolahan citra digital berdasarkan masing-masing karakteristik yang dianalasis, salah satunya adalah filter Shen-Castan. Filter Shen-Castan bertujuan untuk mengoptimalkan pendeteksian tepi pada citra bernoise serta meminimalkan waktu komputasi dan biaya implementasinya dalam bentuk rangkaian elektronik.

Hal terpenting dalam bidang pemrograman salah satunya adalah algoritma. Algoritma berperan penting dikarenakan untuk membantu *programmer* membuat sebuah program yang efektif dan efisien. Selain itu dapat membantu dalam memahami konsep logika pemrograman. Sebuah algoritma dikatakan telah benar jika algoritma tersebut dapat memberikan keluaran yang benar pada saat menerima masukkan sesuai dari definisi algoritma tersebut. Selain memberikan hasil yang benar, efisiensi dari waktu eksekusi ataupun penggunaan memori dari algoritma adalah hal yang penting bagi sebuah algoritma. Kompleksitas algoritma digunakan untuk menjelaskan model pengukuran waktu dan ruang ini.

Kompleksitas waktu, T(n), merupakan jumlah operasi yang dilakukan untuk melaksanakan algoritma sebagai fungsi dari input n. Sedangkan kompleksitas ruang menyatakan berapa banyak ruang dalam memori yang dibutuhkan suatu algoritma ketika beroperasi. Pengukuran kompleksitas waktu dan kompleksitas ruang algoritma Filter IIR Shen-Castan untuk deteksi tepi pada citra digital menggunakan notasi Big-O. Notasi Big-O adalah

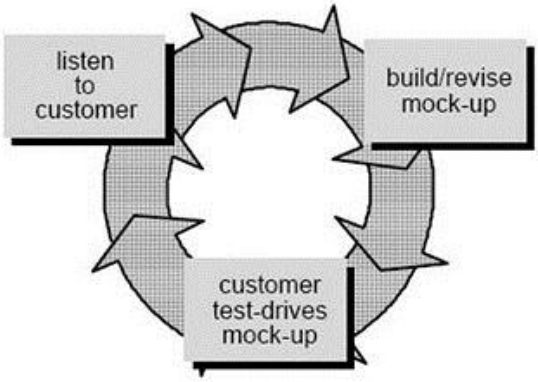
salah satu cara untuk mengkonversi keseluruhan langkah-langkah algoritma ke dalam bentuk aljabar yaitu dengan menghiraukan konstanta yang lebih kecil dan koefisien yang tidak berdampak besar terhadap keseluruhan kompleksitas permasalahan yang diselesaikan oleh algioritma tersebut.

Tujuan dari penelitian ini adalah untuk mengukur kompleksitas dari algoritma filter IIR Shen-Castan untuk deteksi tepi pada citra digital berdasarkan pada kompleksitas waktu dan kompleksitas ruang sehingga dapat dijadikan sebagai pertimbangan pemilihan algortima yang akan digunakan dalam melakukan operasi deteksi tepi pada citra digital.

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 (versi trial).



Gambar 1. Prototipe SDLC

3. HASIL DAN PEMBAHASAN

3.1. Konsep Filter IIR

IIR merupakan filter numerik atau filter digital yang memiliki sistem *feedfoward* dan *feedback* sekaligus. Hal ini berarti keluaran yang dihasilkan dari filter IIR tidak hanya tergantung pada data sesaat dan data selanjutnya tetapi juga bergantung pada hasil perhitungan sebelumnya.

Pada umumnya, formulasi dari filter IIR adalah sebagai berikut :

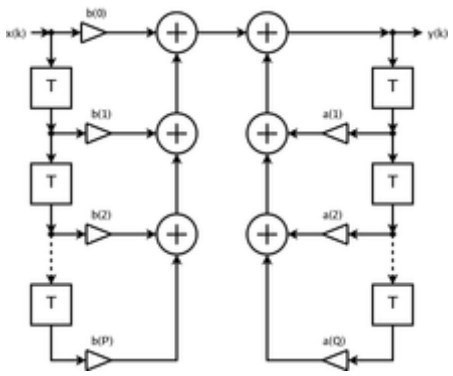
$$\sum_{l=0}^{L} b_0 y(n-1) = \sum_{k=0}^{K} a_k x(n-k) \qquad (1)$$

dimana nilai L menunjukkan orde rekursif dari filter IIR, x(n) adalah sinyal masukkan, y(n) adalah sinyal keluaran, a_1 adalah koefisien filter bagian *feedfoward* dan b1 adalah koefisien filter bagian *feedback*. Persamaan (1) dapat disederhanakan menjadi :

$$y(n) = \sum_{k=0}^{K} a_0 x(n-k) - \sum_{l=1}^{L} b_1 y(n-1) \qquad (2)$$

dengan asumsi $b_0 = 1$. Variabel x dan y adalah variable dalam sebuah fungsi pada koordinat x dan y sedangkan x(n) digunakan untuk menyatakan sinyal masukkan dan y(n) digunakan untuk menyatakan sinyal keluaran.

Kelebihan dari filter IIR dibandingkan dengan filter FIR terletak pada efisiensinya dalam implementasi. Pada saat diimplementasikan dalam sinyal prosesor, filter IIR menghasilkan jumlah kalkulasi yang lebih sedikit dibandingkan filter FIR sehingga memerlukan spesifikasi perangkat yang lebih rendah. Dibawah ini adalah salah satu contoh rangkaian dari filter IIR, yaitu :



Gambar 2. Contoh filter IIR

Ada beberapa filter IIR yang dikembangkan dalam teori pengolahan citra digital yaitu filter Shen-Castan, filter Canny-Deriche dan filter Bourenane.

3.2. Algoritma Filter 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 :

$$f(x) = Ce^{-\alpha|x|} \qquad \text{jika } x < 0 \qquad (3)$$

$$h(x) = \begin{Bmatrix} K \cdot e^{-\alpha|x|} \\ -K \cdot e^{-\alpha|x|} \end{Bmatrix} \qquad \text{jika } x \geq 0 \qquad (4)$$

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 α 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 masukkan x(n) seperti persamaan berikut :

$$y^{+}(n) = a_1 \cdot x(n) - b_1 \cdot y^{+}(n-1) \qquad \text{untuk } n = 1, \ldots, N$$

$$y^{-}(n) = a_1 \cdot x(n+1) - b_1 \cdot y^{-}(n+1) \qquad \text{untuk } n = N, \ldots, 1$$

$$y(n) = y^{+}(n) + y^{-}(n) \qquad (5)$$

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

Kemudian untuk sinyal keluaran y(n) sebagai respon impuls dari filter numerik h(x) terhadap sinyal masukkan x(n) dapat dinyatakan dengan persamaan berikut :

$$y^{+}(n) = c_1 \cdot x(n) - b_1 \cdot y^{+}(n-1) \quad \text{untuk n = 1, …, N}$$

$$y^{-}(n) = c_1 \cdot x(n+1) - b_1 \cdot y^{-}(n+1) \quad \text{untuk n = N, …, 1}$$

$$y(n) = y^{+}(n) + y^{-}(n) \quad (6)$$

dengan koefisien filter $c_1 = K = 1\text{-}e^{-\alpha}$ dan $b1 = 1\text{-}e^{-\alpha}$.

Tanda "+" pada $y^{+}(n)$ menyatakan bahwa proses perhitungan dilakukan dari kiri ke kanan untuk arah horizontal dan dari atas ke bawah untuk arah vertikal. Sebaliknya tanda "-" pada $y^{-}(n)$ menyatakan bahwa proses perhitungan dilakukan dari kanan ke kiri untuk arah horizontal dan dari bawah ke atas untuk arah vertikal.

Roushdy [8] melakukan penelitian deteksi tepi dengan menggunakan algoritma filter Shen-Castan dengan hasil seperti berikut :

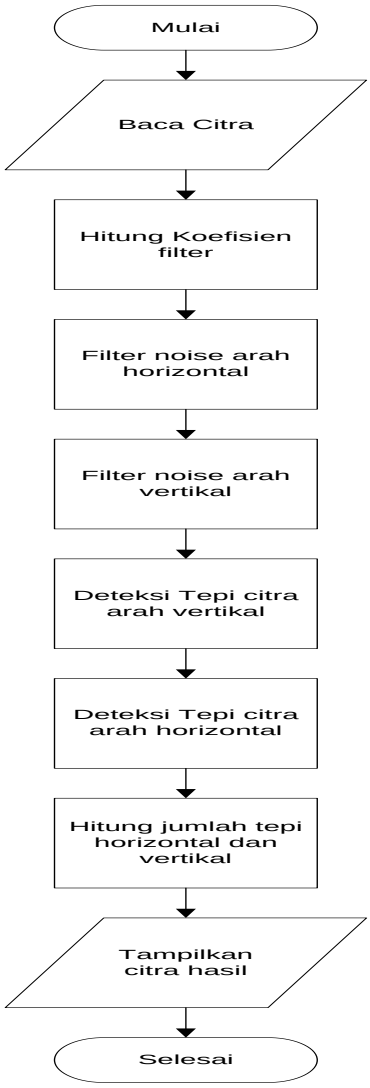
(a) Citra Asli

(b) Tepi Citra Shen-Castan

Gambar 3. Tepi citra hasil deteksi tepi filter Shen-Castan

3.3. Pseudocode dari Algoritma filter Shen-Castan

Citra merupakan fungsi 2-D (dua dimensi) maka operasi pemfilteran *noise* dan pendeteksian tepi dilakukan secara terpisah dan dengan arah sumbu kordinat yang berbeda. Flowchart untuk algoritma filter Shen-Castan adalah sebagai berikut :



Gambar 2. Flowchart Filter Shen-Castan

Pendeteksian tepi citra pada arah horizontal dilakukan melalui proses pemfilteran *noise* pada arah vertikal terhadap citra asli terlebih dahulu kemudian dilanjutkan dengan pendeteksian tepi pada arah horizontal terhadap hasil citra pemfilteran vertikal tersebut.

Begitu pun sebaliknya, pendeteksian tepi citra arah vertikal dilakukan melalui proses pemfilteran noise pada arah horizontal terhadap citra asli terlebih dahulu kemudian dilanjutkan dengan pendeteksian tepi arah vertikal terhadap hasil citra pemfilteran horizontal itu. Hasil akhir tepi citra adalah jumlah harga mutlak dari tepi horizontal dan harga mutlak dari tepi vertikal.

Pendeteksian tepi citra dengan arah sumbu vertikal terdiri dari dua buah langkah kerja yaitu dilakukan pemrosesan filter *noise* dengan arah sumbu horizontal terlebih dahulu baru kemudian dilanjutkan dengan pendeteksian tepi dengan arah sumbu vertikal Berikut adalah pseudocode dari algoritma filter Shen-Castan untuk filter *noise* dengan arah sumbu horizontal :

```
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
```

Setelah dilakukan pemisahan *noise* pada citra mengikuti sumbu horizontal, maka dilakukan langkah kedua yaitu pendeteksian tepi citra dengan arah vertikal. Di bawah ini adalah pseudocode dari algortima filter Shen-Castan untuk pendeteksian tepi citra dengan arah vertikal:

```
for j=1:M         %Deteksi tepi citra arah vertikal
   yp(1,j)=c1*yx(1,j);
   yn(N,j)=-c1*yx(N,j);
   yn(N-1,j)=-c1*yx(N-1,j)-b1*yn(N,j);
   for i=2:N-1
      yp(i,j)=c1*yx(i-1,j)-b1*yp(i-1,j);
      yn(N-i,j)=-c1*yx((N-i)+1,j)-b1*yn((N-i)+1,j);
   end
   yp(N,j)=c1*yx(N,j)-b1*yp(N-1,j);
end
for i=1:N
   for j=1:M
      Yx(i,j)=abs(yp(i,j)+yn(i,j));
   end
end
```

Untuk proses berikutnya yaitu pendeteksian tepi citra dengan arah sumbu horizontal. Pendeteksian tepi citra dengan sumbu horizontal terdiri dari dua buah langkah kerja juga yaitu dilakukan pemrosesan filter *noise* dengan arah sumbu vertikal terlebih dahulu baru kemudian dilanjutkan dengan pendeteksian tepi dengan arah sumbu horizontal Berikut adalah pseudocode dari algoritma filter Shen-Castan untuk filter *noise* dengan arah sumbu vertikal :

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

Seperti pada proses pertama sebelumnya maka setelah dilakukan pemisahan *noise* pada citra mengikuti sumbu vertikal, maka dilakukan langkah kedua yaitu pendeteksian tepi citra dengan arah horizontal. Di bawah ini adalah pseudocode dari algortima filter Shen-Castan untuk pendeteksian tepi citra dengan arah horizontal :

```
for i=1:N          %Deteksi tepi citra arah horizontal
   yp(i,1)=c1*yy(i,1);
   yn(i,M)=-c1*yy(i,M);
   yn(i,M-1)=-c1*yy(i,M-1)-b1*yn(i,M);
   for j=2:M-1
      yp(i,j)=c1*yy(i,j-1)-b1*yp(i,j-1);
      yn(i,M-j)=-c1*yy(i,(M-j)+1)-b1*yn(i,(M-j)+1);
   end
   yp(i,M)=c1*yy(i,M)-b1*yp(i,M-1);
end
for i=1:N
   for j=1:M
      Yx(i,j)=abs(yp(i,j)+yn(i,j));
   end
end
```

Dalam menentukan kompleksitas waktu sebuah algoritma diperlukan ukuran masukan n dan *running time* dari algoritma tersebut. *Running time* pada sebuah algoritma pada umumnya akan meningkat seiring dengan bertambahnya ukuran n. *Running time* algoritma pada masukan n tertentu merupakan jumlah operasi atau langkah yang dieksekusi.

Berdasarkan pseudocode dari filter *noise* dengan arah sumbu horizontal didapatkan *running time* sebagai berikut :

Tabel 1. Perhitungan *running time* untuk filter *noise* dengan arah sumbu horizontal

No	**Pseudocode**	**Nilai**	**Waktu**
1	for i=1:N	c_1	n
2	yp(i,1)=a1*I(1,i);	c_2	1
3	yn(i,M)=a1*I(i,M);	c_3	1
4	ym(i,M-1)=a1*I(i,M-1)-b1*yn(i,M);	c_4	1
5	for j=2:M-1	c_5	n
6	yp(i,j)=a1*I(i,j)-b1*yp(i,j);	c_6	1
7	yn(i,M-j)=a1*I(I,(M-j)+1)-b1*yn(i,(M-j)+1);	c_7	1
8	End	c_8	0
Lanjutan Tabel 2….			
No	**Pseudocode**	**Nilai**	**Waktu**
9	yp(i,M)=a1*I(i,M)-b1*yp(i,M-1);	c_9	1
10	End	c_{10}	0
11	for i=1:N	c_{11}	n
12	for j=1:M	c_{12}	n
13	Yx(i,j)=yp(i,j)+yn(i,j);	c_{13}	1
14	end	c_{14}	0
15	end	c_{15}	0

Sedangkan berdasarkan pseudocode untuk deteksi tepi citra dengan arah sumbu vertikal didapatkan *running time* sebagai berikut :

Tabel 2. Perhitungan running time untuk deteksi tepi citra dengan arah sumbu vertical

No	Pseudocode	Nilai	Waktu
1	for j=1:M	c_1	n
2	yp(1,j)=c1*yx(1,j);	c_2	1
3	yn(N,j)=-c1*yx(N,j);	c_3	1
4	yn(N-1,j)=-c1*yx(N-1,j)-b1*yn(N,j);	c_4	1
5	for i=2:N-1	c_5	n
6	yp(i,j)=c1*yx(i-1,j)-b1*yp(i-1,j);	c_6	1
7	yn(N-i,j)=-c1*yx((N-i)+1,j)-b1*yn((N-i)+1,j);	c_7	1
8	End	c_8	0
9	yp(N,j)=c1*yx(N,j)-b1*yp(N-1,j);	c_9	1
10	End	c_{10}	0
11	for i=1:N	c_{11}	n
12	for j=1:M	c_{12}	n
13	yx(i,j)=abs(yp(i,j)+yn(i,j));	c_{13}	1
14	End	c_{14}	0
15	end	c_{15}	0

Berdasarkan pseudocode dari filter *noise* dengan arah sumbu vertikal didapatkan *running time* sebagai berikut :

Tabel 3. Perhitungan running time untuk filter *noise* dengan arah sumbu vertical

No	Pseudocode	Nilai	Waktu
1	for j=1:M	c_1	n
2	yp(1,j)=a1*I(1,j);	c_2	1
3	yn(N,j)=a1*I(N,j);	c_3	1
4	yn(N-1,j)=a1*I(N-1,j)-b1*yn(N,j);	c_4	1
5	for i=2:N-1	c_5	n
6	yp(i,j)=a1*I(i,j)-b1*yp(i-1,j);	c_6	1
7	yn(N-i,j)=a1*I((N-1)+1,j)-b1*yn((N-i)+1,j);	c_7	1
8	end	c_8	0
9	yp(N,j)=a1*I(N,j)-b1*yp(N-1,j);	c_9	1
10	end	c_{10}	0
11	for i=1:N	c_{11}	n
12	for j=1:M	c_{12}	n
13	yy(i,j)=abs(yp(i,j)+yn(i,j);	c_{13}	1
14	end	c_{14}	0
15	end	c_{15}	0

Dan berdasarkan pseudocode untuk deteksi tepi citra dengan arah sumbu horizontal didapatkan *running time* sebagai berikut :

Tabel 4. Perhitungan running time untuk deteksi tepi citra dengan arah sumbu horizontal

No	Pseudocode	Nilai	Waktu
1	for i=1:N	c_1	n
2	yp(i,1)=c1*yy(i,1);	c_2	1
3	yn(i,M)=-c1*yy(i,M);	c_3	1
4	yn(i,M-1)=-c1*yy(i,M-1)-b1*yn(i,M);	c_4	1

5	for j=2:M-1	c_5	n
6	yp(i,j)=c1*yy(i,j-1)-b1*yp(i,j-1);	c_6	1
7	yn(i,M-j)=-c1*yy(i,(M-j)+1)-b1*yn(i,(M-j)+1);	c_7	1
8	end	c_8	0
9	yp(i,M)=c1*yy(i,M)-b1*yp(i,M-1);	c_9	1
10	end	c_{10}	0
11	for i=1:N	c_{11}	n
12	for j=1:M	c_{12}	n
13	Yx(i,j)=abs(yp(i,j)+yn(i,j));	c_{13}	1
14	end	c_{14}	0
15	end	c_{15}	0

Dari jumlah hasil kali nilai dengan waktu maka di peroleh perhitungan running time dari algoritma filter IIR Shen-Castan untuk deteksi tepi citra adalah sebagai berikut :

$$T(n) = \frac{n^2 + n}{2} = O(n^2) \quad (7)$$

Dari hasil perhitungan *time complexity* didapatkan jika algoritma filter IIR Shen-Castan mempunyai kompleksitas algoritma $O(n^2)$. Dengan interval nilai α = {0.5, 0,75, 1.0, 1.25, 1.5, 1.75, 2.0}, algoritma Shen-Castan membutuhkan *space complexity* $O(n^2)$.

4. KESIMPULAN

Algoritma untuk deteksi tepi pada citra digital cukup beragam dan tentunya dalam proses ada metode yang tercepat dan memerlukan ruang penyimpanan yang sedikit. Algoritma Shen-Castan yang dapat menghasilkan filter yang lebih baik dari filter turunan pertama Gaussian dan LoG ternyata memerlukan *time complexity* sebesar $O(n^2)$ dan membutuhkan *space complexity* $O(n^2)$ dimana algoritma tersebut akan membutuhkan setidaknya N unit ruang di memori dan bergantung pada panjang array-nya.

5. SARAN

Adapun saran yang penulis berikan untuk pengembangan penelitian ini selanjutnya adalah membandingkan kompleksitas waktu dan kompleksitas ruang algoritma Shen-Castan ini dengan algoritma filter lainnya yang sejenis sehingga dapat memberikan referensi pada peneliti yang ingin melakukan metode teknik deteksi tepi pada citra digital.

DAFTAR PUSTAKA

[1.] Bahig, Hazem M., 2019, Complexity Analysis and Performance of Double Hashing Sort Algorithm, *Journal of the Egyptian Mathematical Society*, https://doi.org/10.1186/s42787-019-000402

[2.] Bourennane, E., Gouton, P., Paindavoine, M. dan Truchettet, F., 2002. Generalization of Canny-Deriche Filter for Detection Edge of Noisy Expoinential Edge, *Proceedings of SPIE - The International Society for Optical Enginering · October 1998*

[3.] Hasan, Mohd Ashri Abu, Ibrahim, Asmah, Khalid, Noor Elaiza Abdul dan Noor, Noorhayati Mohamed. Evaluation of Sobel, Canny, Shen & Castan Using Sample Line Histogram Method, DOI: 10.1109/ITSIM.2008.4632072 · Source: IEEE Xplore

[4.] Lidyawati, Lita, Arsyad Ramadhan Darlis dan Alfin Fernando Tamba, 2015, Implementasi Filter Infinite Impulse Response (IIR) dengan Respon Butterworth dan

Chebyshev menggunakan DSK TMS320C6713, Jurnal Elektro Telekomunikasi Juli 2015, DOI: 10.25124/jett.v2i1.97

[5.] Madenda, Sarifuddin, 2015, *Pengolahan Citra & Video Digital : Teori, Aplikasi, Dan Pemrograman Menggunakan MATLAB*, Erlangga, Jakarta.

[6.] Munir, Rinaldi, 2004, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Informatika, Bandung.

[7.] Prasetyo, Eko, 2011, *Pengolahan Citra Digital dan Aplikasinya menggunakan MATLAB*, Andi, Yogyakarta.

[8.] Roushdy, Mohamed, 2006, Comparative Study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter, *GVIP Journal, Volume 6, Issue 4, December, 2006*

[9.] Vitriasni, Elishabet Yori, Arif Surtono dan Gurum Ahmad P, 2013, Perbandingan Kinerja Filter Digital IIR dan FIR untuk Mereduksi Derau Interferensi Jaringan Listrik 60 Hz pada Sinyal EKG, Jurnal Teori dan Aplikasi Fisika, Vol 01, No. 01, Januari 2013.

[10.] Zhou, Yuren, He, Jun dan Nie, Qing, 2009, A Comparative Runtime Analysis of Heuristic Algorithms for Satisfiability Problems, *doi:10.1016/j.artint.2008.11.002*