

## Penerapan Smallest Univalued Segment Assimilating Nucleus (SUSAN) Untuk Mendeteksi Tepi Citra Mammogram

Dania Silalahi<sup>1\*</sup>

<sup>1</sup>Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia  
Email: <sup>1\*</sup>daniasilalahi97@gmail.com

**Abstrak**—Perkembangan dan kemajuan teknologi informasi memberikan pengaruh besar dalam penelitian dibidang medis yaitu dengan menggunakan teknik pengolahan citra, saat ini penelitian dengan teknik pengolahan citra pada citra medis berkembang sangat pesat itu semua disebabkan karena munculnya berbagai macam metode. Penelitian mammogram ini menggunakan metode SUSAN yang merupakan metode untuk mendeteksi tepi dalam gambar digital, yaitu kurva kontinu di mana terdapat variasi kuat dan cepat dalam kecerahan gambar. Metode deteksi tepi SUSAN sederhana dan beroperasi dengan menggabungkan gambar dengan Laplacian dari fungsi Gaussian, atau, sebagai perkiraan cepat dengan perbedaan Gaussians. Mammogram atau kanker payudara merupakan salah satu penyakit yang timbul pada manusia yang pada umumnya terhadap perempuan. Terdapat beberapa kelemahan dalam menganalisis mammogram payudara guna mendeteksi keberadaan kanker payudara. Hal ini bisa diakibatkan oleh sel kanker yang tertutup oleh noise, kontras citra yang rendah dan faktor manusiawi lainnya.

**Kata Kunci:** SUSAN, Laplacian, Gaussian, Mammogram

**Abstract**—The development and advancement of information technology has a major influence in research in the medical field, namely by using image processing techniques, currently research with image processing techniques on medical images is growing very rapidly, it is all due to the emergence of various methods. This mammogram study uses the SUSAN method, which is a method for detecting edges in digital images, which are continuous curves where there are strong and rapid variations in image brightness. The SUSAN edge detection method is simple and operates by combining the image with the Laplacian of the Gaussian function, or, as a quick approximation with the Gaussian difference. Mammogram or breast cancer is one of the diseases that arise in humans, which generally affects women. There are several weaknesses in analyzing breast mammograms to detect the presence of breast cancer. This can be caused by cancer cells covered by noise, low image contrast and other human factors.

**Keywords:** SUSAN, Laplacian, Gaussian, Mammogram

### 1. PENDAHULUAN

Dewasa ini telah banyak sekali aplikasi-aplikasi yang memanfaatkan pengolahan citra. Hal ini ditunjukkan dengan semakin banyaknya metode-metode baru yang dikembangkan untuk mendukung berbagai macam aplikasi komputer yang digunakan pada berbagai bidang, salah satu contohnya yaitu dalam bidang biometrik. Bidang ini telah banyak diaplikasikan pada kehidupan sehari-hari. Teknik-teknik yang dipakai biasanya berfungsi untuk melakukan proses identifikasi dengan menggunakan karakteristik alami yang terdapat pada manusia, seperti contohnya wajah, sidik jari, iris dan retina mata, suara, dan tandatangan. Payudara merupakan salah satu fitur biologis yang sangat populer dipakai untuk autentikasi sistem. Karena payudara merupakan bagian dari tubuh yang tidak dapat dicuri ataupun diduplikasi tidak seperti halnya metode konvensional lain yang menggunakan *id card* atau *password* yang masih dapat disalahgunakan oleh pihak yang tidak berwenang.

Algoritma *Smallest Univalued Segment Assimilating Nucleus* (SUSAN) memiliki pendekatan berbeda dari metode pendeteksi tepi sebelumnya seperti Canny Edge Detector. Algoritma yang digunakan pada SUSAN tidak bergantung pada penghalusan gambar sebelum gambar tersebut diproses. SUSAN juga tidak memperhitungkan turunan dari ruang pixel pada gambar, melainkan menentukan tepi objek dengan membandingkan intensitas suatu pixel dengan pixel disekitarnya.[1]

Peningkatan kualitas citra merupakan salah satu proses awal dalam peningkatan mutu citra. Peningkatan mutu citra diperlukan karena seringkali citra yang di jadikan objek pembahasan mempunyai kualitas yang buruk, misalnya citra mengalami deraw, kabur. Metode *Smallest Univalued Segment Assimilating Nucleus* (SUSAN) dibuat dengan beberapa tahapan yaitu antara lain: *image enhancement* berupa proses perbaikan citra dengan meningkatkan kualitas citra baik kontras maupun kecerahan.[2]

*Mammogram* atau kanker payudara merupakan salah satu penyakit yang timbul pada manusia yang pada umumnya terhadap perempuan. Terdapat beberapa kelemahan dalam menganalisis *mammogram* payudara guna mendeteksi keberadaan kanker payudara. Hal ini bisa diakibatkan oleh sel kanker yang tertutup oleh *noise*, kontras citra yang rendah dan faktor manusiawi lainnya seperti : kelelahan, *mood*, dan lainnya. Untuk meminimalisir hal tersebut dibutuhkan suatu metode yang dapat membantu dokter dalam menganalisis citra *mammogram* payudara. Pada penelitian ini, dilakukan suatu proses yang bertujuan untuk meningkatkan kualitas *mammogram* agar lebih memudahkan dokter dalam mendiagnosis kelainan pada payudara [3]

*Mammogram* memiliki manfaat sangat besar untuk mendeteksi adanya kanker payudara, *cancer.gov* menyebutkan bahwa ada beberapa kekurangan yang dimiliki oleh *screening mammogram* Hasil negatif yang salah. Ini terjadi ketika hasil *screening mammogram* menunjukkan kondisi normal meskipun sebenarnya ada kanker payudara. Secara keseluruhan, *mammogram* tidak mendeteksi 20% kanker payudara yang ada pada saat *screening* [4]. Ini lebih sering terjadi pada wanita dengan usia muda. Hal inilah yang menyebabkan pengolahan citra digital memiliki kegunaan yang sangat luas.

Berdasarkan permasalahan diatas, diperlukan solusi untuk memperbaiki proses citra, salah satu alternatif yang dapat digunakan adalah metode *smallest univalued segment assimilating nucleus* (SUSAN) yang dapat membantu dan mempermudah dalam pendeteksian tepi citra mammogram.

## 2. METODOLOGI PENELITIAN

### 2.1 Citra Digital

Citra digital merupakan citra yang dinyatakan dalam kumpulan data digital dan dapat diproses oleh komputer. Akuisisi citra digital dengan menggunakan berbagai peranti digital sebagai contoh, gambar awan diperoleh melalui kamera digital, citra artikel koran diperoleh melalui alat pemindai (*scanner*). Citra di dalam komputer disusun atas jumlah sejumlah *pixels*, sebuah *pixels* dapat dibayangkan sebagai sebuah titik yang dinyatakan dengan bentuk  $(y, x)$  dengan  $y$  menyatakan baris dan  $x$  menyatakan kolom. Umumnya, koordinat pojok kiri atas dinyatakan dengan  $(0,0)$  dan dengan demikian, jika suatu citra berukuran  $M$  baris dan  $N$  kolom atau biasa dinyatakan sebagai  $M \times N$ , koordinat *pixels* terbawah dan terkanan berada di koordinat  $(M-1, N-1)$ . [6]

### 2.2 Deteksi Tepi

Deteksi tepi mencakup berbagai metode matematis yang bertujuan untuk mengidentifikasi poin dalam gambar digital di mana kecerahan gambar berubah secara tajam atau, lebih formal, memiliki penghentian. Titik di mana perubahan kecerahan gambar secara tajam biasanya diatur menjadi satu set segmen garis melengkung yang disebut tepinya. Masalah yang sama menemukan penghentian dalam satu-dimensi sinyal dikenal sebagai langkah Deteksi dan masalah menemukan penghentian sinyal dari waktu yang dikenal sebagai deteksi perubahan. Deteksi tepi adalah alat mendasar dalam pemrosesan gambar, visi mesin dan visi komputer, khususnya di bidang deteksi fitur dan ekstraksi fitur[13].

### 2.3 Smallest Univalued Segment Assimilating Nucleus (SUSAN)

Algoritma *Smallest Univalued Segment Assimilating Nucleus* (SUSAN) memiliki pendekatan berbeda dari metode pendeteksi tepi sebelumnya seperti Canny Edge Detector. Algoritma yang digunakan pada SUSAN tidak bergantung pada penghalusan gambar sebelum gambar tersebut diproses. SUSAN juga tidak memperhitungkan turunan dari ruang pixel pada gambar, melainkan menentukan tepi objek dengan membandingkan intensitas suatu pixel dengan pixel disekitarnya[1].

Perbandingan nilai intensitas pixel akan dibatasi dalam sebuah mask berbentuk lingkaran. Intensitas pixel yang dibedakan adalah pixel yang berada di tengah mask dengan seluruh pixel disekitarnya. Berikut ini adalah persamaan yang digunakan dalam metode SUSAN.

$$c(\bar{r}, \bar{r}_0) = \begin{cases} 1 & \text{if } |I(\bar{r}) - I(\bar{r}_0)| \leq t \\ 0 & \text{if } |I(\bar{r}) - I(\bar{r}_0)| > t \end{cases} \dots\dots\dots(1)$$

## 3. HASIL DAN PEMBAHASAN

### 3.1 Analisa Masalah

*Mammogram* atau kanker payudara merupakan salah satu penyakit yang timbul pada manusia yang pada umumnya terhadap perempuan. Adapun gejala penyebab kanker payudara yaitu adanya benjolan di payudara atau penebalan jaringan yang terasa berbeda dari jaringan di sekitarnya, perubahan pada bentuk dan ukuran payudara, kulit payudara memerah, pengelupasan kulit areola dan kulit payudara, nyeri dan pembengkakan pada payudara, darah ke luar dari puting payudara, benjolan atau pembengkakan di bawah ketiak, puting tertarik masuk ke dalam.

Namun sering terjadi banyak yang tidak sadar penderita kanker payudara bahwa dirinya menderita penyakit kanker payudara. Kendala yang terjadi untuk mengetahui penyakit kanker payudara setelah di CT-SCAN adalah kurangnya sumber informasi secara detail keberadaan dan letak sel kanker tersebut. Terdapat beberapa kelemahan dalam menganalisis *mammogram* payudara guna mendeteksi keberadaan kanker payudara. Hal ini bisa diakibatkan oleh sel kanker yang tertutup oleh *noise*, kontras citra yang rendah dan faktor manusiawi lainnya seperti kelelahan, *mood*, dan lainnya.

Untuk meminimalisir hal tersebut dibutuhkan suatu metode yang dapat membantu dokter dalam menganalisis citra *mammogram* dengan menggunakan algoritma SUSAN untuk mendeteksi tepi citra *mammogram*. Proses deteksi tepi dengan menggunakan metode SUSAN dilakukan dengan terlebih dahulu menentukan kernel filter gaussian selanjutnya melakukan proses konvolusi dan kemudian Penentuan *Zero Kraus* untuk proses Analisa dengan algoritma SUSAN.

#### 3.1.1 Penerapan Metode SUSAN

Metode SUSAN atau yang biasa disebut Canny edge detector ditemukan oleh Sam yang meneliti pemodelan persepsi visual manusia. Dalam memodelkan pendeteksi tepi, dia menggunakan ideal step edge, yang direpresentasikan dengan fungsi sign satu dimensi. Pendekatan algoritma canny dilakukan dengan konvolusi fungsi image dengan operator gaussian dan turunan-turunannya. Metode SUSAN ini memiliki banyak kesamaan dengan metode canny, tetapi ada sedikit

perbedaan antara kedua metode ini yaitu jika metode SUSAN menggunakan turunan pertama untuk melakukan deteksi tepi, sedangkan metode canny menggunakan turunan kedua untuk melakukan proses deteksi tepi. Nilai R, G dan B citra sampel dikonversi menjadi nilai RGB seperti pada tabel 1. berikut ini:

Tabel 1. Nilai RGB Citra Sampel

RGB						
21	17	19	27	52	47	17
20	20	19	37	58	40	16
16	22	18	48	64	32	18
12	24	20	59	64	22	24
14	23	31	64	58	21	23
21	20	47	66	51	25	18
27	15	62	69	43	30	13

Langkah-langkah perhitungan analisa contoh kasus dengan metode SUSAN:

1. Pembentukan kernel Filter Gaussian

- a. Untuk mempermudah perhitungan maka dibuat tabel matriks 5 x 5 seperti tabel 3.2 sebagai hasil matriks *Filter Gaussian* nantinya.

Tabel 2. Matriks *Filter Gaussian*

(x,y)	-2	-1	0	1	2
-2	?	?	?	?	?
-1	?	?	?	?	?
0	?	?	?	?	?
1	?	?	?	?	?
2	?	?	?	?	?

- b. Proses membuat nilai elemen pada matriks *Filter Gaussian*

Untuk proses pengurangan *noise* pada gambar menggunakan *Filter Gaussian*, dapat dinyatakan pada persamaan di bawah. Persamaan inilah yang dipakai sebagai dasar untuk menentukan nilai-nilai setiap elemen dalam *Filter Gaussian* yang akan dibuat. penulis memberikan standard deviasi ( $\sigma$ ) dengan besaran nilai 2 (dua). Jadi untuk memberikan nilai elemen pada masing-masing koordinat matriks *filter g(x,y)*. maka, masing-masing koordinat dimasukkan kedalam rumus, adapun rumus dan perhitungannya dapat dilihat dibawah ini :

$$g(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Dimana :

$g$  = gaussian

$x$  = nilai koordinat  $x$

$y$  = nilai koordinat  $y$

$e$  = adalah konstanta *euler* (2.718281828)

$\sigma$  = Standard deviasi (2)

maka :

$$G(0,0) = e^{-\frac{0^2+0^2}{2.2^2}} = 1$$

$$G(1,0) = e^{-\frac{1^2+0^2}{2.2^2}} = 0,778800783$$

$$G(0,1) = e^{-\frac{0^2+1^2}{2.2^2}} = 0,778800783$$

$$G(-1,0) = e^{-\frac{-1^2+0^2}{2.2^2}} = 0,778800783$$

$$G(0,-1) = e^{-\frac{0^2+(-1)^2}{2.2^2}} = 0,778800783$$

$$G(1,1) = e^{-\frac{1^2+1^2}{2.2^2}} = 0,60653066$$

$$G(1,-1) = e^{-\frac{1^2+(-1)^2}{2.2^2}} = 0,60653066$$

$$G(-1,1) = e^{-\frac{-1^2+1^2}{2.2^2}} = 0,60653066$$

$$G(-1,-1) = e^{-\frac{-1^2+(-1)^2}{2.2^2}} = 0,60653066$$

$$G(2,1) = e^{-\frac{2^2+1^2}{2.2^2}} = 0,286504797$$

$$G(1,2) = e^{-\frac{1^2+2^2}{2.2^2}} = 0,286504797$$

$$G(-2,1) = e^{-\frac{-2^2+1^2}{2.2^2}} = 0,286504797$$

$$G(1,-2) = e^{-\frac{1^2 + (-2)^2}{2 \cdot 2^2}} = 0,286504797$$

$$G(-1,-2) = e^{-\frac{(-1)^2 + (-2)^2}{2 \cdot 2^2}} = 0,286504797$$

$$G(-2,-1) = e^{-\frac{(-2)^2 + (-1)^2}{2 \cdot 2^2}} = 0,286504797$$

$$G(2,0) = e^{-\frac{2^2 + 0^2}{2 \cdot 2^2}} = 0,3678794$$

$$G(0,2) = e^{-\frac{0^2 + 2^2}{2 \cdot 2^2}} = 0,3678794$$

$$G(0,-2) = e^{-\frac{0^2 + (-2)^2}{2 \cdot 2^2}} = 0,3678794$$

$$G(-2,0) = e^{-\frac{(-2)^2 + 0^2}{2 \cdot 2^2}} = 0,3678794$$

$$G(2,2) = e^{-\frac{2^2 + 2^2}{2 \cdot 2^2}} = 0,135335283$$

$$G(-2,-2) = e^{-\frac{(-2)^2 + (-2)^2}{2 \cdot 2^2}} = 0,135335283$$

$$G(-2,2) = e^{-\frac{(-2)^2 + 2^2}{2 \cdot 2^2}} = 0,135335283$$

$$G(2,-2) = e^{-\frac{2^2 + (-2)^2}{2 \cdot 2^2}} = 0,135335283$$

c. Hasil elemen matriks *Filter Gaussian*

Dari hasil perhitungan di atas, maka diperoleh hasil matriks elemen sebagai berikut :

Tabel 3. Elemen Matriks *Filter Gaussian* Setelah Di isi

(x,y)	-2	-1	0	1	2
-2	0,135335283	0,286504797	0,3678794	0,286504797	0,135335283
-1	0,286504797	0,60653066	0,778800783	0,60653066	0,286504797
0	0,367879441	0,778800783	1	0,778800783	0,367879441
1	0,286504797	0,60653066	0,778800783	0,60653066	0,286504797
2	0,135335283	0,286504797	0,3678794	0,286504797	0,135335283

d. Membuat nilai normalisasi matriks *Filter Gaussian*

Normalisasi adalah proses membulatkan masing-masing nilai matriks *filter* yang telah didapat, selain memudahkan perhitungan komputer, nilai-nilai pembobotan harus dibuat bulat sebab nilai intensitas *pixel* pada citra digital bernilai bulat, maka nilai *Filter Gaussian* juga harus dibuat kedalam bilangan bulat. oleh karena itu untuk membuat nilai normalisasi dari *filter* maka nilai 1 sebagai pembagi dibagi dengan nilai *filter* terkecil. hal ini bertujuan agar jumlah nilai-nilai pembobot sama dengan satu. Kemudian nilai normalisasi dikalikan dengan semua nilai *filter* yang belum dinormalisasi.

$$c = \left\lfloor \frac{1}{\min(g(x,y))} \right\rfloor$$

e. Hasil normalisasi matriks *Filter Gaussian*

Jika diperhatikan, nilai matriks *filter* pada tabel 3. untuk nilai terkecil dalam matriks  $g(x,y) = 0,135335283$ , maka akan diperoleh nilai normalisasi ( $c$ ) sebagai berikut :

$$c = \frac{1}{0,135335283} = 7,389056096 = 7 \text{ (setelah dibulatkan keatas)}$$

Jadi nilai normalisasi yang didapat dari rumus diatas setelah proses pembulatan adalah 7 (tujuh).

f. Proses perkalian matriks kernel dengan nilai normalisasi

Mengalikan matriks dengan nilai normalisasi bertujuan untuk menyederhanakan angka yang digunakan dalam matriks *Filter Gaussian* dengan mengalikan masing-masing nilai *filter*  $g(x,y)$  dengan nilai normalisasi yang sudah didapat, sehingga menghasilkan matriks *Filter Gaussian* baru. Berdasarkan persamaan *filter gaussian* dengan menggunakan nilai  $c = 7$ , maka :

$$g(x,y) = c \cdot e^{-\frac{x^2 + y^2}{2 \cdot \sigma^2}}$$

g. Hasil proses perkalian matriks kernel dengan nilai normalisasi

Dengan menggunakan persamaan 5.3 di atas, sehingga diperoleh hasil proses perkalian matriks kernel dengan nilai normalisasi :

$$G(0,0) = 7 \times 1 = 7$$

$$G(1,0) = 7 \times 0,778800783 = 6$$

$$G(0,1) = 7 \times 0,778800783 = 6$$

$$G(-1,0) = 7 \times 0,778800783 = 6$$

$$G(0,-1) = 7 \times 0,778800783 = 6$$

$$G(1,1) = 7 \times 0,60653066 = 4$$

$$G(1,-1) = 7 \times 0,60653066 = 4$$

$$G(-1,1) = 7 \times 0,60653066 = 4$$

$$G(-1,-1) = 7 \times 0,60653066 = 4$$

- $G(2,1) = 7 \times 0,286504797 = 2$
- $G(1,2) = 7 \times 0,286504797 = 2$
- $G(-2,1) = 7 \times 0,286504797 = 2$
- $G(1,-2) = 7 \times 0,286504797 = 2$
- $G(-1,-2) = 7 \times 0,286504797 = 2$
- $G(-2,-1) = 7 \times 0,286504797 = 2$
- $G(2,0) = 7 \times 0,3678794 = 3$
- $G(0,2) = 7 \times 0,3678794 = 3$
- $G(0,-2) = 7 \times 0,3678794 = 3$
- $G(-2,0) = 7 \times 0,3678794 = 3$
- $G(2,2) = 7 \times 0,135335283 = 1$
- $G(-2,-2) = 7 \times 0,135335283 = 1$
- $G(-2,2) = 7 \times 0,135335283 = 1$
- $G(2,-2) = 7 \times 0,135335283 = 1$

Maka matrik *Filter Gaussian* baru yang telah dinormalisasi seperti pada tabel 4.3 berikut ini.

Tabel 4. Matriks *Filter Gaussian* Setelah Dinormalisasikan

(x,y)	-2	-1	0	1	2
-2	1	2	3	2	1
-1	2	4	6	4	2
0	3	6	7	6	3
1	2	4	6	4	2
2	1	2	3	2	1

Setelah mendapatkan nilai elemen pada matrik *filter*, selanjutnya adalah mencari jumlah semua elemen nilai pembobot  $g(x,y)$  pada matrik *filter* sebagai pembagi matrik *filter* yang sudah didapat, tujuannya agar nilai intensitas citra tetap seperti semula dan menjaga agar nilai baru tidak berada di luar batas nilai *grayscale* artinya nilai *pixel* citra hasil perhitungan harus dinormalkan. Berdasarkan matrik tersebut jumlah semua elemen nilai pembobot pada *filter* adalah = 79, jadi *filter* hasil rancangannya adalah :

$$G_{(x,y)} = \frac{1}{79} \begin{matrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 7 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{matrix}$$

2. Proses Konvolusi

Konvolusi sering kali dilibatkan dalam operasi ketetanggaan *pixel*. Konvolusi pada citra sering disebut konvolusi 2 dimensi. Konvolusi 2 dimensi didefinisikan sebagai proses untuk memperoleh suatu *pixel* berdasarkan nilai *pixel* itu sendiri dan tetangganya, dengan melibatkan suatu matriks yaitu kernel yang mempresentasikan pembobotan. Penjelasan rumus yang digunakan dalam konvolusi *Filter Gaussian* pada persamaan 2.7 dibawah :

$$h(x, y) = f(x, y) * g(x, y) = \sum_{k=2}^M \sum_{l=2}^N f(k, l). g(x - k, y - l)$$

Dimana :

$h(x,y)$ : gambar *output*

$f(x,y)$  : adalah gambar *input*

$g(x,y)$  : adalah *Filter Gaussian*

Jadi secara umum rumus di atas adalah jumlah dari perkalian antara *pixel* citra dengan *Filter Gaussian* dan hasilnya dibagi dengan jumlah atau *Sum* dari matriks *filter* seperti berikut ini:

21	17	19	27	52	47	17	* $\frac{1}{79}$	1	2	3	2	1
20	20	19	37	58	40	16		2	4	6	4	2
16	22	18	48	64	32	18		3	6	7	6	3
12	24	20	59	64	22	24		2	4	6	4	2
14	23	31	64	58	21	23		1	2	3	2	1
21	20	47	66	51	25	18						
27	15	62	69	43	30	13						

$$G_{(x,y)} = (21*1)+(17*2)+(19*3)+(27*2)+(52*1)+(20*2)+(20*4)+(19*6)+(37*4)+(58*2)+(16*3)+(22*6)+(18*7)+(48*6)+(64*4)+(12*2)+(24*4)+(20*6)+(59*4)+(64*2)+(14*1)+(23*2)+(31*3)+(64*2)+(58*1)/79$$

$$G_{(x,y)} = 31$$

Sehingga diperoleh nilai piksel yang baru seperti berikut ini:

21	17	19	27	52	47	17
20	20	19	37	58	40	16
16	22	31			32	18
12	24				22	24
14	23				21	23
21	20	47	66	51	25	18
27	15	62	69	43	30	13

Selanjutnya geser kernel satu piksel ke kanan dan lakukan konvolusi seperti berikut:

21	17	19	27	52	47	17				1	2	3	2	1
20	20	19	37	58	40	16				2	4	6	4	2
16	22	18	48	64	32	18				3	6	7	6	3
12	24	20	59	64	22	24				2	4	6	4	2
14	23	31	64	58	21	23				1	2	3	2	1
21	20	47	66	51	25	18								
27	15	62	69	43	30	13								

\*  $\frac{1}{79}$

$$G_{(x,y)} = (17*1)+(19*2)+(27*3)+(52*2)+(47*1)+(20*2)+(19*4)+(37*6)+(58*4)+(40*2)+(22*3)+(18*6)+(48*7)+(64*6)+(32*4)+(24*2)+(20*4)+(59*6)+(64*4)+(22*2)+(23*1)+(31*2)+(64*3)+(58*2)+(21*1)/79$$

$$G_{(x,y)} = 41$$

Sehingga diperoleh nilai piksel yang baru seperti berikut ini:

21	17	19	27	52	47	17
20	20	19	37	58	40	16
16	22	31	41		32	18
12	24				22	24
14	23				21	23
21	20	47	66	51	25	18
27	15	62	69	43	30	13

Geser lagi kernel satu piksel ke kanan dan lakukan konvolusi seperti berikut:

21	17	19	27	52	47	17				1	2	3	2	1
20	20	19	37	58	40	16				2	4	6	4	2
16	22	18	48	64	32	18				3	6	7	6	3
12	24	20	59	64	22	24				2	4	6	4	2
14	23	31	64	58	21	23				1	2	3	2	1
21	20	47	66	51	25	18								
27	15	62	69	43	30	13								

\*  $\frac{1}{79}$

$$G_{(x,y)} = (19*1)+(27*2)+(52*3)+(47*2)+(17*1)+(19*2)+(37*4)+(58*6)+(40*4)+(16*2)+(18*3)+(48*6)+(64*7)+(32*6)+(18*4)+(20*2)+(59*4)+(64*6)+(22*4)+(24*2)+(23*1)+(64*2)+(58*3)+(21*2)+(23*1)/79$$

$$G_{(x,y)} = 42$$

Sehingga diperoleh nilai piksel yang baru seperti berikut ini:

21	17	19	27	52	47	17
20	20	19	37	58	40	16
16	22	31	41	42	32	18
12	24				22	24
14	23				21	23
21	20	47	66	51	25	18
27	15	62	69	43	30	13

Selanjutnya geser kernel satu piksel ke bawah dan lakukan konvolusi seperti proses diatas. Sehingga diperoleh nilai piksel yang baru seperti berikut ini:

21	17	19	27	52	47	17
20	20	19	37	58	40	16
16	22	31	41	42	32	18
12	24	34	42	43	22	24

14	23	38	45	43	21	23
21	20	47	66	51	25	18
27	15	62	69	43	30	13

3. Zero Crossing SUSAN

Setelah nilai akhir konvolusi maka perlu mengetahui arah tepi yaitu menghubungkan antara arah tepi dengan sebuah arah yang dapat dilacak dari citra dengan menggunakan zero crossing SUSAN dengan persamaan:

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| \leq t \\ 0 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| > t \end{cases}$$

Jika diasumsikan nilai *Thresholding* (*t*) sebesar = 30 maka diperoleh syarat untuk Zero Kraus SUSAN sebagai berikut :

- a. Jika nilai piksel  $\leq 30$  maka diganti nilainya menjadi 1
  - b. jika nilai piksel  $> 30$  maka diganti nilainya menjadi 0
- Sehingga diperoleh tabel hasil Zero Crossing seperti berikut:

Tabel 5. Hasil Zero Crossing

1	1	1	1	0	0	1
1	1	1	0	0	0	1
1	1	0	0	0	0	1
1	1	0	0	0	1	1
1	1	0	0	0	1	1
1	1	0	0	0	1	1
1	1	0	0	0	1	1

3.2. Implementasi

Untuk bisa menjalankan aplikasi ini dibutuhkan beberapa requirement berupa perangkat keras maupun perangkat lunak agar sistem dapat berjalan dengan lancar. Adapun kebutuhan sistem yang harus dipenuhi adalah sebagai berikut:

Perangkat keras yang dibutuhkan laptop atau PC yang memiliki spesifikasi:

- 1. Processor Minimal Pentium IV
- 2. Memory 128 MB
- 3. Hardisk Minimal 80 GB
- 4. Monitor
- 5. Keyboard dan mouse sebagai sarana piranti input program

Perangkat lunak yang dibutuhkan untuk dapat menjalankan sistem tersebut:

- 1. Sistem Operasi Windows 7
- 2. Matlab 6.1

Tampilan program adalah tampilan atau interface dari program yang akan telah dibangun serta proses penggunaan program tersebut. Selanjutnya implementasi program dilakukan untuk mengetahui sejauh mana sistem informasi yang dirancang dapat mengatasi masalah serta untuk mengetahui hubungan antar komponen sistem. Tampilan input adalah tampilan yang dapat dilihat oleh pengguna saat pengguna akan melakukan *input* data

1. GUI SUSAN

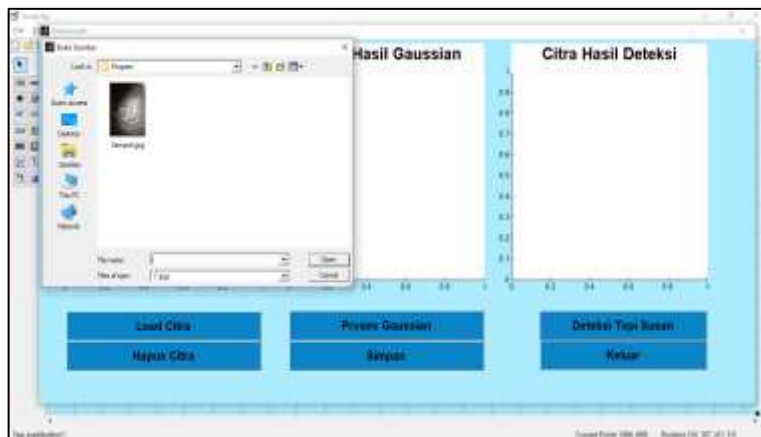
GUI Susan merupakan *interface* dari program yang telah dirancang untuk proses penerapan metode SUSAN seperti gambar 1. berikut ini:



Gambar 1. GUI SUSAN

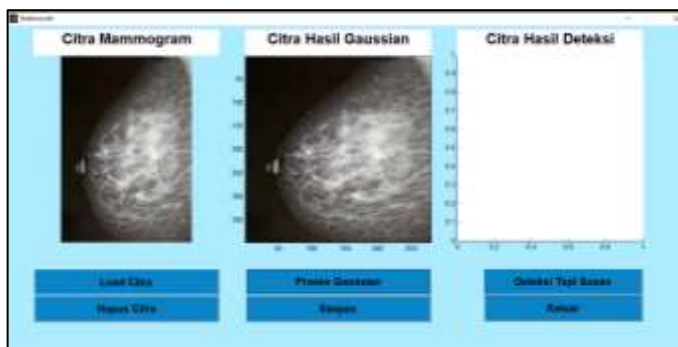
Tampilan *Output* merupakan tampilan proses pengujian program yang telah dirancang dengan menerapkan metode SUSAN pada citra *mammogram*.

Adapun proses pengujian pada pilihan Load Citra adalah seperti gambar 2. di bawah ini:



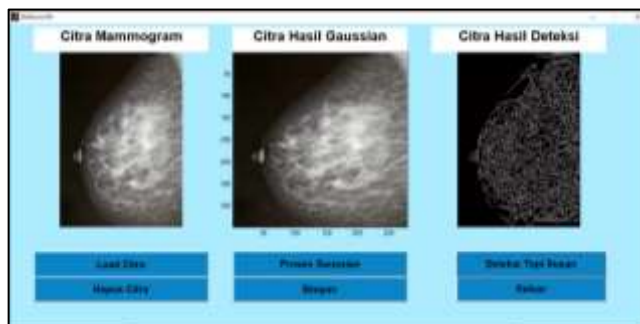
Gambar 2. Tampilan Load Citra

Adapun proses pengujian pada pilihan Proses Gaussian adalah seperti gambar 2. di bawah ini:



Gambar 3. Tampilan Proses Gaussian

Adapun proses pengujian pada pilihan Deteksi SUSAN adalah seperti gambar 4.4 di bawah ini:



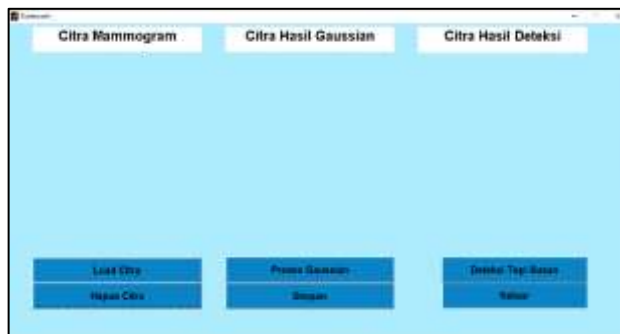
Gambar 4. Tampilan Proses Gaussian

Adapun proses pengujian pada pilihan Deteksi SUSAN adalah seperti gambar 4.5 di bawah ini:



Gambar 5. Tampilan Proses Gaussian

Adapun proses pengujian pada pilihan Deteksi SUSAN adalah seperti gambar 4.6 di bawah ini:



**Gambar 6.** Tampilan Proses Gaussian

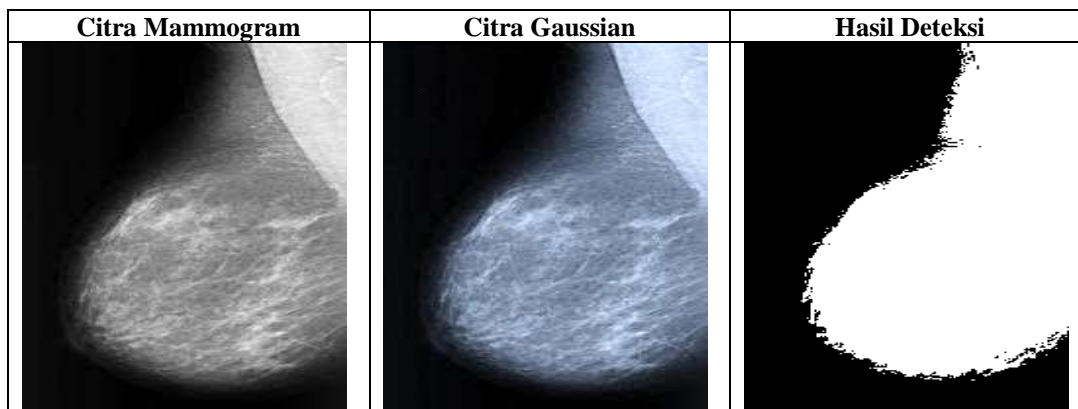
Adapun proses pengujian pada pilihan Deteksi SUSAN adalah seperti gambar 4.7 di bawah ini:



**Gambar 7.** Tampilan pilihan Keluar

Berikut merupakan tabel hasil pengujian citra mammogram dengan menerapkan metode SUSAN seperti berikut ini:

Citra Mammogram	Citra Gaussian	Hasil Deteksi



Dasi hasil pengujian di atas maka dapat diambil kesimpulan bahwa proses deteksi tepi dengan menerapkan metode SUSAN dapat mendeteksi area tumor pada citra *mammogram* tersebut.

#### 4. KESIMPULAN

Berdasarkan analisa dan pembahasan yang dilakukan, maka dapat disimpulkan Prosedur deteksi tepi citra mammogram dilakukan mengubah citra grayscale menjadi citra biner. Kemudian dilanjutkan dengan membentuk matrix filter dengan *filter gaussian*. Selanjutnya dilakukan konvolusi dengan matriks kernel yang telah ditentukan terlebih dahulu. Setelah citra hasil konvolusi diperoleh selanjutnya adalah menentukan *zero crossing* berdasarkan nilai thresholding sehingga diperoleh *zero cross* SUSAN sebagai citra hasil deteksi tepi. Penerapan Algoritma Smallest Univalue Segment Assmilating Nucleus (SUSAN) untuk deteksi tepi citra mammogram dilakukan berdasarkan nilai *zero cross* SUSAN. Nilai *zero cross* SUSAN bergantung pada nilai *Thresholding*. Semakin tinggi nilai *Thresholding* maka akan semakin jelas terlihat objek yang terkandung di dalam citra.

#### REFERENCES

- [1] Smith&Brady, "Algoritma Smallest Univalue Segment Assmilating Nucleus (SUSAN)," *Algorith. Smallest Univalue Segm. Assmilating Nucl.*, 1995.
- [2] E. Marr, D. ; Hildreth, "Teori Deteksi Tepi," *Prosiding Royal Society of London.* .
- [3] Hanifah Rahmi Fajrin, "PERBANDINGAN METODE UNTUK PERBAIKAN KUALITAS CITRA MAMMOGRAM," *Mammogram*, 2016.
- [4] J. S. Muspika Helja, . Nurhasanah, "Analisis Fraktal Citra Mammogram Berbasis Tekstur Sebagai Pendukung Diagnosis Kanker Payudara," 2013.
- [5] H. Hanifah, *Implementasi Kebijakan dan Politik*. 2002.
- [6] A. Kadir, *Dasar Pengolahan citra dengan Delphi*, I. Yogyakarta: Penerbit Andi, 2013.
- [7] T. Sutoyo, *Teori Pengolahan Citra Digital*. Malang, 2009.
- [8] D. Putra, "Pengolahan Citra Digital," 2010.
- [9] D. Putra, "Dasar Pengolahan Citra," 2010.
- [10] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.
- [11] "Pemindai Gambar Image Scanner," 2016.
- [12] A. M. Rahmat, "Restorasi Citra Dengan Menggunakan Metode Iteratif Lanczos Hybrid Regularization."
- [13] Yeni and Herdiyeni, "Deteksi Tepi (Edge Detection)," 2009.
- [14] S. Laguni, "Citra Mammogram," 2016.
- [15] Jogyanto, *Analisis dan Design*. Yogyakarta: Penerbit Andi, 2005.
- [16] "Picture MATLAB,Definisi MATLAB."
- [17] Masfaqih, *Pengertian Matlab dan Fungsi Bagian-bagiannya*. 2015.