PERBANDINGAN METODE-METODE *EDGE DETECTION* UNTUK PROSES SEGMENTASI CITRA DIGITAL

Mahmud Yunus
Program Studi Teknik Informatika
STMIK PPKIA Pradnya Paramita Malang
Jl. L.A Sucipto No. 249-A Malang
e-mail: myoenoes@pradnya.ac.id

ABSTRACT

Edge detection (Edge detection) is executed to detect the operating margins (edges) that restricts two homogeneous image regions with different brightness levels (Pitas, 1993). A digital image edge detection is the process of looking for differences in the intensity of the states boundaries of an object (sub-images) in the entire digital image in question. Purpose of edge detection is to improve the appearance of the boundary line of an area or object in the image. The process of image edge detection is done by looking for the locations of the intensity of the pixels that discontinue with the intensity of the pixels adjacent (neighboring / neighborhood).

There are several methods such as object edge detection operators are known, such as (1) Sobel, (2) Prewitt, (3) Robert, (4) Laplacian of Gaussian, (5) and Canny (6) Kiresh. Each operator has its own advantages and disadvantages in performing edge detection. By doing a combination or in the development of these operators allow to gain a better edge detection (obviously).

One of the contributions of this study for science and technology is the creation of a software application that can be used as a simulation tool using edge detection operators in digital image and its various combinations in order to gain a better edge detection (obviously).

***Keywords**: Edge detection, detection operator*

LATAR BELAKANG

Deteksi tepi *(Edge detection)* adalah operasi yang dijalankan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda (Pitas, 1993). Deteksi tepi sebuah citra digital merupakan proses untuk mencari perbedaan intensitas yang menyatakan batas-batas suatu objek (sub-citra) dalam keseluruhan citra digital yang dimaksud. Tujuan pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Proses deteksi tepi citra dilakukan dengan mencari lokasi-lokasi intensitas pixel-pixel yang *discontinue* dengan intensitas pixel-pixel yang berdekatan (bertetanggaan/*neighborhood*). Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra, bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya.

Pada proses pengolahan citra digital seperti segmentasi dan analisis citra digital, peranan metode-metode pendeteksian tepi citra sangat berperan penting terhadap keakuratan hasil segmentasi dan analisis citra digital. Penelitian tentang "Identifikasi Sel Kanker Prostat Menggunakan Metode Segmentasi Berdasarkan

Ukuran Objek Pada Citra", melakukan urutan proses prapengolahan citra digital dimulai dari akuisisi data citra, deteksi tepi, segmentasi dan pengambangan (*thresholding*), hingga citra siap dianalisis (Witeti, 2004). Hasil dari pendeteksian tepi pada sebuah citra digital sangatlah berpengaruh terhadap proses pengolahan citra lebih lanjut seperti segmentasi dan analisis citra. Semakin jelas hasil dari pendeteksian tepi objek-objek (sub-citra) maka semakin baik pula hasil segementasi dan analisis citra yang akan dilakukan.

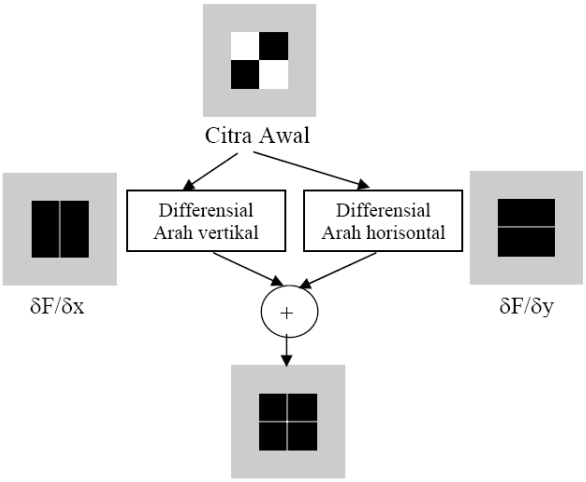
Terdapat beberapa metode berupa operator pendeteksian tepi objek yang dikenal, seperti (1) Sobel; (2) Prewitt; (3) Robert; (4) Laplacian of Gaussian; (5) Canny dan (6) Kiresh. Masing-masing operator tersebut memiliki kelebihan dan kekurangan dalam melakukan deteksi tepi. Terkadang pada sebuah citra digital akan diperoleh hasil pendeteksian tepi yang lebih jelas bila menggunakan operator Sobel, namun pada citra digital lainnya lebih baik menggunakan operator Prewitt atau yang lainnya. Kemungkinan melakukan kombinasi atau pengembangan pada operator-operator tersebut masih memungkinkan sebagai usaha untuk memperoleh hasil deteksi tepi yang lebih baik (jelas).

Ide untuk dapat menghasilkan deteksi tepi yang lebih baik dengan cara mengkombinasikan atau mengembangkan operator-operator deteksi tepi yang ada dapat diimplementasikan dalam bentuk perangkat lunak pengolahan citra digital dengan bahasa pemrograman Delphi. Berdasarkan hal tersebut penelitian ini difokuskan pada pembuatan perangkat lunak yang dapat melakukan simulasi pendeteksian tepi citra digital dengan menggunakan beberapa operator dan kombinasinya guna memperoleh hasil deteksi tepi yang lebih baik. Pada perangkat lunak yang dikembangkan ini juga mencakup praproses pengolahan citra digital seperti peningkatan kualitas (kontras & kecerahan) citra dan restorasi (penghalusan & penajaman) citra.

Salah satu kontribusi dari penelitian ini bagi ilmu pengetahuan dan teknologi adalah terciptanya sebuah aplikasi perangkat lunak yang dapat digunakan sebagai alat simulasi penggunaan operator-operator deteksi tepi citra digital beserta berbagai kombinasinya dengan tujuan untuk memperoleh hasil deteksi tepi yang lebih baik (jelas)

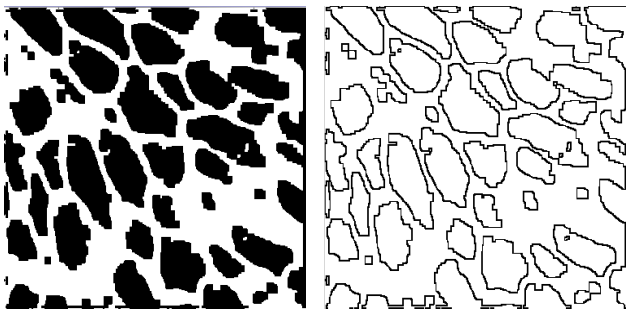
PRINSIP DETEKSI TEPI PADA CITRA DIGITAL

Tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang cepat/tiba-tiba (besar) dalam jarak yang singkat. Sedangkan deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah untuk (a) menandai bagian yang menjadi detail citra; dan (b) memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra. Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya. Pada gambar 1 berikut dapat dilihat proses yang dilakukan untuk memperoleh tepi gambar dari suatu citra yang ada.



Gambar 1 Proses Deteksi Tepi Citra Digital

Pada gambar 2 terlihat bahwa hasil deteksi tepi berupa tepi-tepi dari suatu gambar. Bila diperhatikan bahwa tepi suatu gambar terletak pada titik-titik yang memiliki perbedaan intensitas pixel yang tinggi.



Gambar 2 Hasil Deteksi Tepi Pada Suatu Citra Digital

Berdasarkan prinsip-prinsip filter pada citra maka tepi suatu gambar dapat diperoleh menggunakan High Pass Filter (HPF), yang mempunyai karakteristik:

$$\sum_{y}\sum_{x} H(x,y) = 0$$

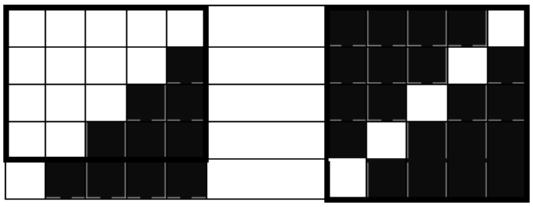
Misalkan terdapat fungsi suatu citra sebagai f (x, y) sebagai berikut;

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

Maka dengan menggunakan fungsi filter H (x,y) = [-1 1], akan diperoleh matrik citra sebagai berikut;

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

Sehingga bila digambarkan maka proses filter tersebut memiliki pixel-pixel masukan dan keluaran sebagai berikut;



Proses deteksi tepi (*edge detection*) sendiri dapat dikelompokkan berdasarkan operator atau metode yang digunakan dalam proses pendeteksian tepi suatu citra (*edge detection*) untuk memperoleh citra hasil.

Operator Deteksi Tepi Citra Digital

Proses pendeteksian tepi citra digital dapat dilakukan dengan teknik konvolusi menggunakan berbagai macam metode/operator. Operator deteksi tepi merupakan alat yang digunakan untuk memodifikasi nilai derajat keabuan sebuah titik berdasarkan derajat keabuan titik-titik yang ada disekitarnya (konvolusi/operasi ketetanggaan). Titik-titik yang dilibatkan dalam operasi ketetanggaan tersebut diberikan bobot yang

nilainya tergantung pada operasi yang akan dilakukan, sedangkan banyaknya titik yang dilibatkan biasanya 2x2, 3x3, 5x5, 7x7 dan seterusnya.

Operator yang dapat digunakan untuk deteksi tepi adalah operator (a) berbasis Gradient (turunan pertama) seperti Robert, Sobel, Prewitt dan (b) operator berbasis turunan kedua seperti *Laplacian* dan *Laplacian of Gaussian*. Metode yang banyak digunakan untuk proses deteksi tepi adalah metode Robert, Prewitt dan Sobel (Gonzalez dan Woods, 2002). Namun tidak menutup kemungkinan guna memperoleh hasil deteksi tepi citra yang lebih baik (jelas), operator-operator yang ada tersebut dikombinasikan dan dikembangkan dengan teknik-teknik tertentu.

Operator Gradient

Gradien adalah turunan pertama dari persamaan dua dimensi yang didefinisikan sebagai vektor berikut;

$$G[f(x,y)] = G\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Besar gradient dihitung dengan persamaan;

$$G[f(x,y)] = \sqrt{G_x^2 + G_y^2}$$

Pada prakteknya untuk pengolahan citra, besar gradient diperoleh dari persamaan berikut;

$$G[f(x,y)] = |G_x| + |G_y|$$

Arah dari vektor gradien dapat dihitung dengan persamaan;

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

dimana α diukur dari sumbu x sebagai garis acuan. Guna keperluan perhitungan pada citra digital, turunan persamaan;

$$G[f(x,y)] = G\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Lebih mudah dilakukan dengan pendekatan persaman difrerensial;

$$G_x \cong f(x+1,y) - f(x,y)$$
$$G_y \cong f(x,y) - f(x,y+1)$$

Metode Robert

Metode Robert adalah nama lain dari teknik differensial yang dikembangkan di atas, yaitu differensial pada arah horisontal dan differensial pada arah vertikal, dengan ditambahkan proses konversi biner setelah dilakukan differensial. Teknik konversi biner yang disarankan adalah konversi biner dengan meratakan distribusi warna hitam dan putih. Metode Robert ini juga disamakan dengan teknik DPCM (*Differential Pulse Code Modulation*).

Metode Robert dikenal juga dengan istilah operator Robert Cross (diagonal) yang menggunakan kernel ukuran 2x2 piksel, sehingga tepi yang dihasilkan berada pada tepi atas atau tepi bawah.

1	0
0	-1

Gambar 3 Mask/Kernel Operator Robert Cross

Operator Robert mengambil arah diagonal untuk penentuan arah perhitungan nilai gradientnya,dimana perhitungn gradientnya adalah sebagai berikut;

$$G = |G_x| + |G_y|$$

$$G_x = f(x, y) - f(x+1, y+1)$$

$$G_y = f(x+1, y) - f(x, y+1)$$

Kernel yang digunakan dalam simulasi dapat dimodifikasi menjadi kernel 3x3 piksel dengan memberikan nilai 0 pada elemen kernel tambahan.

Robert Diagonal 1		
0	0	0
0	1	0
0	0	-1

Robert Diagonal 2		
0	0	0
0	0	1
0	-1	0

Gambar 4 Modifikasi Mask/Kernel Operator Robert Cross

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.

Metode atau operator Sobel merupakan operator yang menghindari adanya perhitungan gradient di titik interpolasi. Operator Sobel menggunakan kernel ukuran 3x3 piksel untuk perhitungan gradientnya, dengan pembobotan yang lebih besar pada piksel-piksel yang dekat dengan titik pusat.

Sobel Horisontal		
-1	0	1
-2	0	2
-1	0	1

Sobel Vertikal		
1	2	1
0	0	0
-1	-2	-1

Gambar 5 Mask/Kernel Operator Robert Cross

Operator Sobel melakukan deteksi tepi dengan memperhatikan tepi vertical dan horizontal. Gradient Magnitude dari operator Sobel adalah sebagai berikut;

Misalkan susunan piksel-piksel di sekitar piksel (x,y) adalah,

Piksel-Piksel di sekitar piksel(x,y)		
a_0	a_1	a_2
a_7	(x, y)	a_3
a_6	a_5	a_4

Gambar 6 Letak Piksel-Piksel Yang Dikonvolusi Dengan Kernel 3x3

maka besaran gradient yang dihitung menggunakan operator sobel adalah

$$G[f(x,y)] = \sqrt{(G_x^2 + G_y^2)}$$
$$G_x = (a_2 + c.a_3 + a_4) - (a_0 + c.a_7 + a_6)$$
$$G_y = (a_0 + c.a_1 + a_2) - (a_6 + c.a_5 + a_4)$$

c adalah konstanta bernilai 2.

Metode atau operator Prewitt menggunakan persamaan yang sama dengan operator Sobel, bedanya adalah pada nilai konstanta c yang digunakan bernilai 1, sehingga operator Prewitt dapat dikatakan sebagai operator yang tidak menekankan pembobotan pada piksel-piksel yang

lebih dekat dengan titik pusat kernel. Metode Prewitt menggunakan operator sebagai berikut;

Prewitt Horisontal		
-1	0	1
-1	0	1
-1	0	1

Prewitt Vertikal		
1	1	1
0	0	0
-1	-1	-1

Gambar 7 Mask/Kernel Operator Prewitt

Metode Isotropik

Metode atau operator Isotropik merupakan operator yang menggunakan pembobotan piksel-piksel yang lebih dekat dengan titik pusat kernel dengan nilai √2 (1.41421). Sehingga nilai pembobotan tersebut berada diantara pembobotan operator Sobel dan Prewitt, yaitu 2 > 1.41421 > 1.

Isotropik Horisontal		
-1	0	1
-√2	0	√2
-1	0	1

Isotropik Vertikal		
-1	-√2	-1
0	0	0
1	√2	1

Gambar 8 Mask/Kernel Operator Isotropik

Metode Kiresh

Operator Kiresh (Salem, Kalyankar dan Khamitkar: 2010) menggunakan kernel ukuran 3x3 piksel sebagai berikut;

Kiresh Horisontal		
0	1	1
-1	0	1
-1	-1	0

Kiresh Vertikal		
1	1	0
1	0	-1
0	-1	-1

Gambar 9 Mask/Kernel Operator Kiresh

Operator Turunan Kedua/Metode Laplacian

Metode atau operator Laplacian menggunakan turunan kedua dengan persamaan sebagai berikut;

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Laplacian 1		
0	-1	0
-1	4	-1
0	-1	0

Laplacian 2		
0	1	0
1	-4	1
0	1	0

Gambar 10 Mask/Kernel Operator Laplacian

Turunan kedua untuk arah x dan y, diperoleh dengan persamaan sebagai berikut;

$$\frac{\partial^2 f}{\partial x^2} = f(x-1, y) - 2f(x, y) + f(x+1, y)$$

Dan

$$\frac{\partial^2 f}{\partial y^2} = f(x, y-1) - 2f(x, y) + f(x, y+1)$$

Sehingga diperoleh persamaan sebagai berikut;

$$\nabla^2 f = 4f(x, y) - [f(x-1, y) + f(x+1, y) + f(x, y-1) + f(x, y+1)]$$

Metode Kombinasi Hasil Konvolusi Kernel

Metode Kombinasi Hasil Konvolusi Kernel merupakan cara yang mengkombinasikan hasil konvolusi 2 kernel pada masing-masing operator, seperti operator Laplacian1 dan Lapacian2, atau antara operator Sobel Horizontal dan Sobel Vertikal. Hasil konvolusi 2 kernel dapat dikombinasikan dengan cara mengambil nilai;

- Maksimal

$$K_0(x, y) = Max(|K_1(x, y)|, |K_2(x, y)|)$$

- Rerata

$$K_0(x,y) = \frac{|K_1(x,y)| + |K_2(x,y)|}{2}$$

- Rerata Geometri

$$K_0(x,y) = \sqrt{K_1(x,y)^2 + K_2(x,y)^2}$$

TUJUAN DAN MANFAAT PENELITIAN

Tujuan Penelitian

Penelitian ini dilaksanakan dengan tujuan untuk perbandingan metode-metode deteksi tepi (*edge detection*) objek untuk proses segmentasi citra digital serta merancang-bangun aplikasi pengolahan citra digital yang mampu melakukan pendeteksian tepi objek (sub-citra) untuk proses segmentasi citra digital dengan mengimplentasikan berbagai metode deteksi tepi dan kombinasinya guna menghasilkan pendeteksian tepi objek yang lebih jelas pada citra digital.

Manfaat Penelitian

Hasil-hasil dari penelitian ini diharapkan dapat memberikan manfaat:

a. Pada bidang ilmu pengolahan citra digital dapat digunakan sebagai acuan dalam menentukan metode-metode deteksi tepi yang paling cocok untuk proses segmentasi.
b. Pada pendidikan dapat digunakan sebagai bahan referensi dalam pengembangan IPTEK, khususnya pada pegembangan metode pendeteksian tepi citra dan pengembangan keilmuan pengolahan citra digital.

METODE PENELITIAN

Hasil akhir dari penelitian ini adalah terciptanya suatu produk berupa software aplikasi berbasis komputer yang dapat digunakan untuk melakukan simulasi pendeteksian tepi citra digital dengan menggunakan berbagai metode atau operator seperti Robert, Prewitt, Sobel, Kiresh, Isotropik dan Laplacian. Selain itu, software yang dibangun juga dapat mensimulasikan berbagai kombinasi dari berbagai operator deteksi tepi tersebut guna memperoleh hasil deteksi tepi yang lebih jelas (baik). Hasil pendeteksian tepi citra digital yang jelas dapat digunakan untuk proses segmentasi citra digital. Pendekatan yang sesuai dengan tuntutan hasil akhir tersebut, adalah model pendekatan penelitian dan pengembangan (*research and development*).

Strategi Penelitian

Guna tercapainya tujuan dari penelitian ini, maka dirancang suatu strategi penelitian yang meliputi baberapa tahapan, yaitu;

a. Mengumpulkan data dan teori tentang pengaksesan dan pengolahan citra digital secara umum serta teknik-teknik pendeteksian tepi citra dengan berbagai metode deteksi tepi (*edge detection*) citra.
b. Melakukan analisa dan pemecahan masalah terhadap data yang terkumpul. Merancang algoritma pembacaan data citra digital dengan format BMP (*Bitmap Image*) dan algortima pendeteksian tepi citra dengan metode Robert, Prewitt, Sobel, Kireh, Isotropik dan Laplacian serta dengan berbagai kombinasinya.
c. Merancang desain antar muka (*interface*), sebagai media interaksi antara user dengan

sistem.

d. Melakukan pengkodean program untuk membangun mesin pembaca data dan pendeteksi tepi citra digital yang menggunakan metode atau operator deteksi tepi citra Robert, Prewitt, Sobel, Kireh, Isotropik dan Laplacian serta dengan berbagai kombinasinya

e. Implementasi dan pengujian terhadap aplikasi sistem yang dibangun.

Data, Analisis dan Pemecahan Masalah

Data dan teori yang dikumpulkan dan digunakan dalam penelitian ini adalah data dan teori-teori mengenai pendeteksian tepi citra digital dengan metode atau operator Robert, Prewitt, Sobel, Kiresh, Isotropik dan Laplacian serta dengan berbagai kombinasinya. Penggunaan berbagai operator deteksi tepi tersebut dalam penelitian ini adalah bertujuan untuk mensimulasikan dan membandingkan hasil-hasil pendeteksian tepi citra dengan berbagai operator guna menentukan hasil deteksi tepi citra yang paling baik (jelas). Hasil perbandingan metode-metode pedeteksian tepi citra yang menentukan metode apa dan dengan kombinasi yang bagaimana untuk menghasilkan tepi citra yang lebih baik (jelas), dapat digunakan sebagai dasar dalam proses segmentasi citra sebagai proses lanjutan.

Proses deteksi tepi bertujuan untuk meningkatkan penampakan garis pada citra; jadi prosesnya mempunyai sifat diferensiasi atau memperkuat komponen frekuensi tinggi. Beberapa metode deteksi tepi yang ada adalah;

1. Deteksi tepi dengan nilai ambang, yaitu proses penentuan garis batas suatu wilayah yang homogen atau deteksi tepi, dapat dilakukan dengan menggunakan teknik ambang (*thresholding*). Pada penentuan proses batas wilayah ini diperlukan satu nilai ambang untuk membedakan warna hitam dan warna putih;
2. Deteksi tepi dengan gradien pertama. Mutu kontras gambar yang kurang baik bisa mempunyai efek yang bersifat proses pemerataan atau integrasi, karena itu dalam proses penajaman atau peningkatan kontrasnya digunakan upaya yang bersifat proses diferensiasi. Proses diferensiasi ini merupakan bentuk turunan yang biasanya diterapkan dalam bentuk operator gradien. Untuk citra yang kontinu bentuk gradiennya adalah:

$$\nabla f = \sqrt{\left(\frac{df}{dx}\right)^2 + \left(\frac{df}{dy}\right)^2}$$

Teknik deteksi tepi juga sering dilakukan dengan menggabungkan proses gradien digital dengan teknik nilai ambang. Rumusan gradien yang digunakan dalam bentuk perumusan gradien yang lebih kompleks yang kemudian dikenal sebagai detektor Sobel, yang dapat diterangkan dengan gambaran sebagai berikut:

a	b	c
d	[e]	f
g	h	i

Gambar 11 Detektor Sobel

Dengan a, b, c, d, f, g, h, i merupakan nilai-nilai intensitas piksel tetangga dan nilai e adalah nilai piksel yang akan digantikan

setelah penerapan detektor Sobel.

3. Deteksi tepi dengan gradien arah. Proses deteksi ini dapat digolongkan pada proses penapisan tidak linear. Proses penapisan tidak linear disini dilakukan dengan menggunakan tapis linear yang cara operasinya tidak terhadap setiap titik tetapi terhadap suatu sumbu tertentu, jadi mempunyai arah dalam operasinya. Cara penapisan tidak linear juga dapat dilakukan berdasarkan estimasi suatu nilai statistik pada sekelompok piksel.
4. Deteksi tepi dengan cara geser dan selisih citra. Melalui proses deteksi tepi dengan cara geser dan selisih citra akan diperoleh hasil citra dengan tepi yang lebih jelas. Proses penampilan tepi ini dapat dilakukan menurut arah yang diingini, bisa hanya dari tepi arah vertikalnya saja atau tepi arah horisontalnya saja, atau bahkan seluruh tepi dalam kedua arah tersebut.
5. Deteksi tepi dengan gradien kedua. Deteksi tepi dengan gradien kedua disebut juga detektor Laplacian. Bentuk turunan kedua untuk citra yang kontinu adalah sebagai berikut:

$$\nabla^2 f = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}$$

Dengan demikian operator Laplace akan mempunyai bentuk diskret sebagai berikut:

$$L(i,j) = G(i-1,j) + G(i+1,j) + G(i,j-1) + G(i,j+1) + 4G(i,j)$$

Dari bentuk persamaan diskret citra Laplace di atas dilihat bahwa proses deteksi tepi dengan operator. Laplace dapat dilakukan dengan kernel:

Laplacian 1		
0	-1	0
-1	4	-1
0	-1	0

Laplacian 2		
0	1	0
1	-4	1
0	1	0

Gambar 12 Mask/Kernel Operator Laplacian

Masing-masing operator tersebut memiliki kelebihan dan kekurangan dalam melakukan deteksi tepi. Terkadang pada sebuah citra digital akan diperoleh hasil pendeteksian tepi yang lebih jelas bila menggunakan operator Sobel, namun pada citra digital lainnya lebih baik menggunakan operator Prewitt atau yang lainnya. Kemungkinan melakukan kombinasi atau pengembangan pada operator-operator tersebut masih memungkinkan sebagai usaha untuk memperoleh hasil deteksi tepi yang lebih baik (jelas).

Ide untuk dapat menghasilkan deteksi tepi yang lebih baik dengan cara mengkombinasikan atau mengembangkan operator-operator deteksi tepi yang ada dapat diimplementasikan dalam bentuk perangkat lunak pengolahan citra digital dengan bahasa pemrograman Delphi. Berdasarkan hal tersebut penelitian ini difokuskan pada pembuatan perangkat lunak yang dapat melakukan simulasi pendeteksian tepi citra digital dengan menggunakan beberapa operator dan kombinasinya guna memperoleh hasil deteksi tepi yang lebih baik.

Metode Pembangunan Sistem Pendeteksian Tepi Citra Digital

Secara garis besar sistem pendeteksian tepi

citra digital dengan berbagai metode atau operator dapat dilakukan melalui langkah-langkah sebagai berikut:

1. Pemilihan file citra digital yang akan diproses dan pengenalan/penentuan format pixel citra
   pf1bit Citra Biner
   pf8bit Citra Grayscale
   pf24bit Citra True Color
2. Inisialisasi matrik data Kernel atau Mask untuk operator Robert, Prewitt, Sobel, Isotropik, Kiresh atau Laplacian;
3. Siapkan variabel untuk matrik data citra digital dan matrik data tiap elemen pixel.
4. Tentukan matrik data Kernel atau Mask sesuai pilihan (Robert, Prewitt, Sobel, Isotropik,Kiresh atau Laplacian)
5. Tentukan lebar (*width*) dan tinggi (*height*) citra digital sesuai dengan file citra yang akan diproses
6. Uji, apakah format pixel citra (*pixel format*) adalah citra Biner atau *Grayscale*? jika ya maka baca data citra dan tulis dalam matrik 2 dimensi (*width x height*). Lakukan proses konvolusi pada matrik data citra dengan matrik Kernel atau Mask 3 x 3 untuk operator yang sesuai pilihan (Robert, Prewitt, Sobel, Isotropik,Kiresh atau Laplacian) beserta kombinasinya dan Lakukan penulisan ulang matrik data citra dengan matrik data hasil konvolusi;
7. Jika format pixel citra (*pixel format*) adalah citra *True Color* (24 bit), maka baca data citra untuk lapisan warna Merah (*Red*), Biru (*Blue*) dan Hijau (*Green*) serta untuk masing-masng lapisan tersebut tulis dalam matrik 2 dimensi (*width x height)*. Lakukan proses konvolusi pada masing-masing matrik data lapisan citra (*Red, Blue* dan *Green*) dengan matrik Kernel atau Mask 3 x 3 untuk operator yang sesuai pilihan (Robert, Prewitt, Sobel, Isotropik,Kiresh atau Laplacian) beserta kombinasinya. Lakukan penulisan ulang matrik data citra dengan matrik data hasil konvolusi;
8. Proses selesai

Dikarenakan setiap metode pendeteksian tepi citra memiliki matrik kernel atau mask yang berbeda, maka proses pendeteksian tepi citra digital pada masing-masing metode atau operator juga berbeda.

Algoritma dan Kode Program Pembacaan Data Citra Digital BMP

Algoritma pembacaan data citra adalah sebagai berikut;

1. Uji, apakah proses membuka file citra sukses dilakukan ? jika ya tampilkan citra pada form frCanvasCitra
2. Lakukan pembacaan informasi mengenai file citra yang akan diproses, yaitu mengenai nama file citra, ukuran dimensinya (*width x height*), format pixel citra (*Pixel Format*)
   pf1bit Citra Biner
   pf8bit Citra Grayscale
   pf24bit Citra True Color
3. Proses selesai

Kode program selengkapnya adalah sebagai berikut;

```
procedure TfrMainMenu.Open1Click(Sender: TObject);
begin
```

```
    if OpenPictureDialog1.Execute then begin
        Application.CreateForm(TfrCanvasCitra,
frCanvasCitra);
        with frCanvasCitra do begin
        Caption:=OpenPictureDialog1.FileName;
            FormStyle := fsMDIChild;
            Show;
        end;
    end;
end;


procedure TfrCanvasCitra.FormActivate(Sender:
TObject);
begin
    Image1.Picture.LoadFromFile(Caption);
    Image2.Picture.LoadFromFile(Caption);
    ImageSize:=  IntToStr(Image1.Picture.Width) + '
x ' +
    IntToStr(Image1.Picture.Width)+' Pixel';
    case Image1.Picture.Bitmap.PixelFormat of
        pf1bit      : PixelFormat := 'Biner';
        pf8bit      : PixelFormat := 'Grayscale';
        pf24bit : PixelFormat := 'True Color';
    end;
end;
```

Algoritma Pendeteksian Tepi Citra

1. Siapkan matrik data Kernel atau Mask untuk operator Robert, Prewitt, Sobel, Isotropik, Kiresh atau Laplacian;

   Misalkan;

   ```
   PrewitMask1: Mask3x3          = ((-
   1.00000,    0.00000,    1.00000),  (-1.00000,
   0.00000,    1.00000),  (  -1.00000,    0.00000,
   1.00000));
    PrewitMask2: Mask3x3 =  ((1.00000,
   1.00000,  1.00000), ( 0.00000,  0.00000,  0.00000),
   (-1.00000, -1.00000, -1.00000));
   ```

2. Siapkan variabel untuk matrik data citra digital dan matrik data tiap elemen pixel

   ```
   PC: PByteArray;
   Ki, Bi, Gi, Ri, Ko, Bo, Go, Ro: Array of Array
   of Byte;
   ```

3. Uji, apakah data citra sudah ada? Jika ya maka ke langkah 4, Jika tidak tentukan terlebih dahulu file citra digital yang akan diproses.

4. Tentukan matrik data Kernel atau Mask sesuai pilihan (Robert, Prewitt, Sobel, Isotropik,Kiresh atau Laplacian)

   Misalkan;

   ```
   Mask1 := PrewittMask1;
   Mask2 := PrewittMask2;
   ```

5. Tentukan lebar (*width*) dan tinggi (*height*) citra digital sesuai dengan file citra yang akan diproses

   ```
   w := Image2.Picture.Bitmap.Width;
   h := Image2.Picture.Bitmap.Height;
   ```

6. Uji, apakah format pixel citra (*pixel format*) adalah citra Biner atau *Grayscale*? jika ya maka baca data citra dan tulis dalam matrik 2 dimensi (*width x height*),

   ```
   case Image2.Picture.Bitmap.PixelFormat of
           pf1bit,
           pf8bit  :   begin
               SetLength(Ki, w, h);
               SetLength(Ko, w, h);
               for i := 0 to H - 1 do begin
       PC:=Image2.Picture.Bitmap.ScanLine[i];
       for j := 0 to W - 1 do Ki[j, i] := PC[j];
       end;
   ```

   Lakukan proses konvolusi pada matrik data citra dengan matrik Kernel atau Mask 3 x 3 untuk operator yang sesuai pilihan (Robert, Prewitt, Sobel, Isotropik,Kiresh atau Laplacian) beserta kombinasinya;

```
for i := 1 to W - 2 do begin
for j := 1 to H - 2 do begin
Jumlah1 := 0;
Jumlah2 := 0;
for u := -1 to 1 do begin
for v := -1 to 1 do begin
   Jumlah1 := Jumlah1 + Mask1[u, v] * Ki[i-u, j-v];
   Jumlah2 := Jumlah2 + Mask2[u, v] * Ki[i-u, j-v];
end;
end;

case suiComboBox1.ItemIndex of {Pilihan
Kombinasi}
     0 : Ko[i, j] := Round(Abs(Jumlah1));
     1 : Ko[i, j] := Round(Abs(Jumlah2));
     2 : if (Jumlah1 > Jumlah2) then
        Ko[i, j] := Round(Abs(Jumlah1))
```

```
    else Ko[i, j] := Round(Abs(Jumlah2));
  3 : Ko[i, j] := Round(Sqrt(Jumlah1 * Jumlah1 +
Jumlah2 * Jumlah2));
end;
end;
end;
```

Lakukan penulisan ulang matrik data citra dengan matrik data hasil konvolusi;

```
for i := 0 to H - 1 do begin
PC := Image2.Picture.Bitmap.ScanLine[i];
for j := 0 to W - 1 do PC[j] := Ko[j, i];
end;
Ki := Nil;   Ko := Nil;
```

7. Jika format pixel citra (*pixel format*) adalah citra *True Color* (24 bit), maka baca data citra untuk lapisan warna Merah (*Red*), Biru (*Blue*) dan Hijau (*Green*) serta untuk masing-masng lapisan tersebut tulis dalam matrik 2 dimensi (*width x height)*

```
pf24bit :  begin
SetLength(Bi, w, h);   SetLength(Gi,   w,   h);
      SetLength(Ri, w, h);
SetLength(Bo, w, h);   SetLength(Go,   w,   h);
      SetLength(Ro, w, h);
for i := 0 to H - 1 do begin
PC := Image2.Picture.Bitmap.ScanLine[i];
for j := 0 to W - 1 do begin
Bi[j, i] := PC[j*3+0];   Gi[j, i] := PC[j*3+1];Ri[j,
   i] := PC[j*3+2];
Bo[j, i] := PC[j*3+0]; Go[j, i] := PC[j*3+1];Ro[j, i]
:= PC[j*3+2];
end;
end;
```

Lakukan proses konvolusi pada masing-masing matrik data lapisan citra (*Red, Blue* dan *Green*) dengan matrik Kernel atau Mask 3 x 3 untuk operator yang sesuai pilihan (Robert, Prewitt, Sobel, Isotropik,Kiresh atau Laplacian) beserta kombinasinya;

```
for i := 1 to W - 2 do begin
for j := 1 to H - 2 do begin
Jumlah1 := 0; Jumlah2 := 0;
for u := -1 to 1 do begin
for v := -1 to 1 do begin
Jumlah1 := Jumlah1 + Mask1[u, v] * Bi[i-u, j-v];
Jumlah2 := Jumlah2 + Mask2[u, v] * Bi[i-u, j-v];
end;
end;

case suiComboBox1.ItemIndex of {Pilihan
Kombinasi}
 0 : Bo[i, j] := Round(Abs(Jumlah1));
1 : Bo[i, j] := Round(Abs(Jumlah2));
2 : if (Jumlah1 > Jumlah2) then
Bo[i, j] := Round(Abs(Jumlah1))
else Bo[i, j] := Round(Abs(Jumlah2));
3 : Bo[i, j] := Round(Sqrt(Jumlah1 * Jumlah1 +
Jumlah2 * Jumlah2));
end;
Jumlah1 := 0; Jumlah2 := 0;
for u := -1 to 1 do begin
for v := -1 to 1 do begin
Jumlah1 := Jumlah1 + Mask1[u, v] * Gi[i-u, j-v];
Jumlah2 := Jumlah2 + Mask2[u, v] * Gi[i-u, j-v];
end;
end;

case suiComboBox1.ItemIndex of {Pilihan
Kombinasi}
0 : Go[i, j] := Round(Abs(Jumlah1));
1 : Go[i, j] := Round(Abs(Jumlah2));
2 : if (Jumlah1 > Jumlah2) then
Go[i, j] := Round(Abs(Jumlah1))
else Go[i, j] := Round(Abs(Jumlah2));
3 : Go[i, j] := Round(Sqrt(Jumlah1 * Jumlah1 +
Jumlah2 * Jumlah2));
end;
Jumlah1 := 0; Jumlah2 := 0;
for u := -1 to 1 do begin
for v := -1 to 1 do begin
Jumlah1 := Jumlah1 + Mask1[u, v] * Ri[i-u, j-v];
Jumlah2 := Jumlah2 + Mask2[u, v] * Ri[i-u, j-v];
end;
end;

case suiComboBox1.ItemIndex of {Pilihan
Kombinasi}
0 : Ro[i, j] := Round(Abs(Jumlah1));
1 : Ro[i, j] := Round(Abs(Jumlah2));
2 : if (Jumlah1 > Jumlah2) then
Ro[i, j] := Round(Abs(Jumlah1))
else Ro[i, j] := Round(Abs(Jumlah2));
3 : Ro[i, j] := Round(Sqrt(Jumlah1 * Jumlah1 +
Jumlah2 * Jumlah2));
end;
if suiCheckBox1.Checked then begin
Bo[i, j] := 255 - Bo[i, j];
Go[i, j] := 255 - Go[i, j];
Ro[i, j] := 255 - Ro[i, j];
end;
end;
end;
```

Lakukan penulisan ulang matrik data citra dengan matrik data hasil konvolusi;

```
for i := 0 to H - 1 do begin
   PC := Image2.Picture.Bitmap.ScanLine[i];
   for j := 0 to W - 1 do begin
        PC[j*3+0] := Bo[j, i];   PC[j*3+1]   :=
Go[j, i];   PC[j*3+2] := Ro[j, i];
   end;
end;
Bi := Nil; Gi := Nil; Ri := Nil; Bo := Nil; Go :=
Nil;     Ro := Nil;
```

8. Proses selesai

HASIL DAN PEMBAHASAN

Hasil Studi Literatur

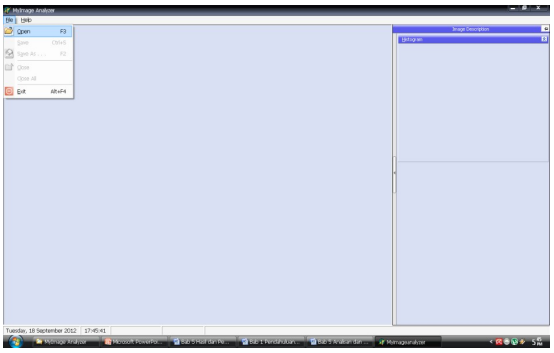
Penelitian tersebut menghasilkan:

A. Landasan teori yang paling tepat untuk membangun aplikasi simulasi pendeteksian tepi citra digital dengan menggunakan beberapa operator dan kombinasinya guna memperoleh hasil deteksi tepi yang lebih baik berupa :
   a. Pengolahan citra digital secara umum mengenai peningkatan kualitas citra, penajaman (*sharpening*) dan penghalusan (*smoothing*), proses invers dan konversi format citra.
   b. Metode-metode atau operator pendeteksian tepi citra dan berbagai kemungkinan kombinasinya
   c. Bahasa pemrograman Delphi khususnya mengenai teknik-teknik atau algoritma pengolahan citra digital

B. Memperoleh software pendukung yang tepat untuk membangun dan implementasi sistem ini meliputi:
   a. Algoritma dan struktur data
   b. Sistem operasi yang dipilih adalah Windows XP Profesional
   c. Bahasa pemrograman yang digunakan adalah Borland Delphi 6

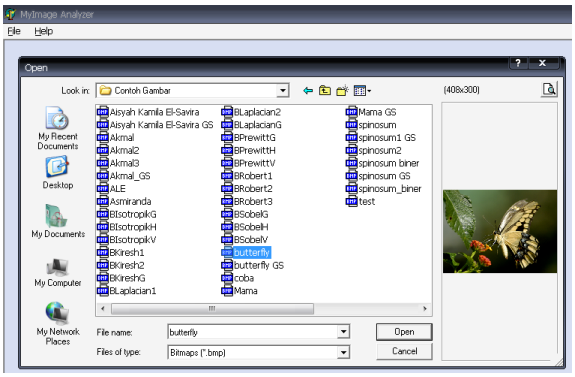
Pengujian Sistem Deteksi Tepi

Berikut ini adalah tampilan halaman utama yang berisi menu pilihan proses dan bagian *Image Description*:



Gambar 13 Tampilan Halaman Utama

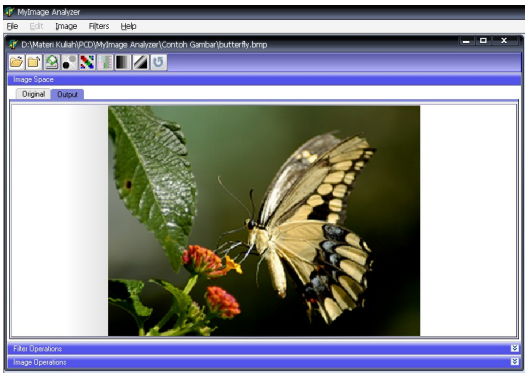
Berikut ini adalah tampilan untuk memilih gambar atau citra digial yang akan dilakukan pendeteksian tepinya:



Gambar 14 Kotak Dialog Pemilihan File Citra Yang Akan Diproses

Pada proses pengujian yang dilakukan, citra yang dipilih adalah citra BUTTERFLY.BMP dengan format file Bitmap Image (BMP) dan resolusi 408 x 408 piksel 24 bit (*true color*).

Tampilan di layar setelah pemilihan file citra adalah sebagai berikut;



Gambar 15 Tampilan Setelah Pemilihan File Citra

Guna memperoleh hasil deteksi tepi yang optimal, hasil konvolusi dengan menggunakan berbagai macam operator deteksi tepi, diterapkan pula proses Invers matrik data citra untuk memperoleh matrik data citra negatif dengan persamaan $P_o(x, y) = 255 – P_i(x, y)$.

Rangkuman Hasil Pengujian Deteksi Tepi

Gambar-gambar berikut ini merupakan rangkuman hasil-hasil pengujian deteksi tepi citra digital dengan menggunakan berbagai operator dan kombinasinya;

Gambar 16 Citra-Citra Hasil Deteksi Tepi Dengan Berbagai Operator

Berdasarkan citra-citra digital hasil pengujian deteksi tepi tersebut, secara umum hasil deteksi tepi dengan Rerata Geometri pada setiap operator memiliki hasil yang lebih baik (lebih jelas) daripada operator aslinya. Operator berbasis gradient seperti Robert, Prewitt, Sobel, Isotrophic dan Kiresh menghasilkan deteksi tepi yang lebih baik daripada operator berbasis turunan kedua (Laplacian). Operator Prewitt dan Isotropik menghasilkan deteksi tepi yang paling jelas diantara operator berbasis gradient lainnya.

KESIMPULAN DAN SARAN

Kesimpulan

Setelah dilakukan pengujian dan analisa terhadap hasil simulasi penggunaan berbagai kombinasi operator pentedeksi tepi citra untuk menghasilkan tepi objek yang lebih jelas, maka dapat disimpulkan sebagai berikut;

a. Secara umum hasil deteksi tepi dengan Rerata Geometri pada setiap operator memiliki hasil yang lebih baik (lebih jelas) daripada operator aslinya
b. Operator berbasis gradient menghasilkan deteksi tepi yang lebih baik daripada operator berbasis turunan kedua (Laplacian)
c. Operator Prewitt dan Isotropik menghasilkan deteksi tepi yang paling jelas diantara operator berbasis gradient lainnya.

Saran

Pada penelitian selanjutnya, dapat dilakukan pengembangan pada metode-metode segmentasi untuk menghasilkan bagian-bagian citra (sub-citra) yang terlabeli, sehingga dapat membedakan antara sub-citra yang satu dengan yang lainnya. Proses segmentasi citra dapat dilakukan pada citra yang telah mengalami praproses deteksi tepi dengan berbagai kombinasi deteksi citra yang dihasilkan oleh perangkat lunak yang telah dibuat ini.

DAFTAR PUSTAKA

Adipranta,R. 2005, Penelitian: *Perancangan dan Pembuatan Aplikasi Segmentasi Gambar Dengan Menggunakan Metode Morphological Watershed*.

Borg, Walter R. dan Meredith Damien Gall. 1979. *Educational Research, An Introduction*. third edition. New York: Longman.

Gonzales, Rafael C., Woods Richard E. 2002. *Digital Image Processing*.

Pitas, I., 1993. *Digital Image Processing Algorithms, Prentice Hall*. Singapore.

Salem Saleh Al-amri, Dr. N.V. Kalyankar and Dr. Khamitkar S.D. 2010.*Image Segmentation By Using Edge Detection*. (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 03, 2010, 804-807.

Witeti. 2004. *Identifikasi Sel Kanker Prostat Menggunakan Metode Segmentasi Berdasar Ukuran Objek Pada Citra*. Teknik Elektro Universitas Diponegoro. Semarang.