

IMPLEMENTASI METODE CANNY DAN SOBEL UNTUK MENDETEKSI TEPI CITRA

Asmardi Zalukhu

Mahasiswa Teknik Informatika STMIK Budi Darma Medan
Jl. Sisingamangaraja No. 338 Simpang Limun Medan

ABSTRAK

Deteksi tepi merupakan langkah awal dari segmentasi citra untuk mendapatkan informasi dalam sebuah citra. Tepi berisi kumpulan dari titik yang mempunyai perbedaan tinggi dengan yang lainnya. Hasil dari deteksi tepi adalah garis batas dari tingkat kecerahan yang berbeda dari suatu objek yang berada dalam citra. Pembahasan yang dibahas pada penelitian ini adalah bagaimana menerapkan metode canny dan sobel untuk mendeteksi tepi citra. Deteksi tepi merupakan proses awal untuk memperoleh informasi dalam citra dan mencirikan batas-batas objek, selanjutnya tepi dapat digunakan dalam proses segmentasi dan identifikasi objek didalam citra dengan menggunakan metode matlab. Dari hasil analisa keluaran program maka dapat diketahui program meningkatkan dan mendeteksi tepi gambar akan lebih baik jika masukan gambar mempunyai banyak tekstur, hasil dari gambar keluaran ditentukan dengan faktor perkalian kernel (Marks). Dari hasil analisa keluaran program operator canny dan operator sobel sama-sama mendeteksi tepi citra dengan baik.

Kata Kunci : Metode canny, sobel, deteksi tepi citra

I. PENDAHULUAN

Perkembangan teknologi saat ini banyak data yang direpresentasikan dalam bentuk citra atau gambar untuk berbagai informasi kepada orang lain. Untuk mendapatkan informasi tertentu pengolahan citra sangat dibutuhkan untuk mendapatkan informasi yang diperlukan. Pengolahan citra adalah suatu metode atau teknik yang dapat digunakan untuk memproses citra atau gambar dengan jalan memanipulasinya menjadi data gambar yang diinginkan untuk mendapatkan informasi tertentu.

Segmentasi merupakan proses yang penting dalam pengolahan citra yang bertujuan untuk memecahkan suatu citra ke dalam beberapa segmentasi dengan suatu kriteria tertentu. Dengan proses segmentasi tersebut, masing-masing objek pada citra dapat diambil secara individu sehingga dapat digunakan sebagai *input*. Jenis operasi ini berkaitan erat dengan pengenalan pola.

Salah satu bagian dari segmentasi citra adalah deteksi tepi. Deteksi tepi merupakan langkah awal dari segmentasi citra untuk mendapatkan informasi dalam citra. Tepi berisi kumpulan dari titik yang mempunyai perbedaan tinggi dengan yang lainnya. Hasil dari deteksi tepi adalah garis batas dari tingkat kecerahan yang berbeda dari suatu objek yang berada dalam citra. Banyak metode yang digunakan untuk melakukan deteksi tepi citra, diantaranya di dalam penelitian ini penulis menggunakan metode *canny* dan *sobel* untuk melakukan segmentasi citra deteksi tepi, dengan mengimplementasikan proses deteksi tepi dengan menggunakan kedua metode tersebut.

Pembahasan yang dibahas pada penelitian ini adalah bagaimana menerapkan metode *canny* dan *sobel* untuk mendeteksi tepi citra dengan membangun sebuah aplikasi deteksi tepi citra menggunakan perangkat lunak *matlab* 6.5. Deteksi tepi merupakan proses awal untuk memperoleh informasi dalam citra dan mencirikan batas-batas objek, selanjutnya tepi dapat digunakan dalam

proses segmentasi dan identifikasi objek didalam citra. Pendeteksian tepi citra dipengaruhi oleh kualitas citra sebelumnya. Semakin jelas dan akurat tepi yang diperoleh sangat membantu identifikasi terhadap informasi yang dibawa citra.

Metode *Canny* merupakan deteksi tepi yang optimal. Permasalahan pada segmentasi citra digital adalah mengetahui nilai pada setiap pikselnya pada citra asli dan harus dirubah ke citra *grayscale* untuk mendapatkan hasil deteksi tepi suatu objek. Erick Wijaya (2012). "Analisis Intensitas Metode Pendeteksian Tepi *Sobel*" metode *sobel* merupakan pengembangan dari metode robert dengan menggunakan filter HDF (*High Pass Filter*) yang di beri satu angka n_l penyangga. Kelebihan dari metode ini adalah kemampuan untuk mengurangi *noise* sebelum melakukan perhitungan pendeteksian tepi. Penyebutan istilah metode *canny* dan metode *sobel* dapat juga digunakan dengan istilah penyebutan operator *canny* dan operator *sobel*.

II. LANDASAN TEORI

A. Citra

Citra merupakan istilah lain untuk gambar sebagai salah satu komponen multimedia memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Melalui sebuah citra kita dapat melihat informasi dari sebuah objek. Ada sebuah peribahasa yang berbunyi "sebuah gambar bermakna lebih dari seribu kata" (*a picture is more than a thousand words*) (Putra, Ardiansyah, 2007).

B. Operator *Canny*

Operator *canny* yang dikemukakan oleh Jhon Canny pada tahun 1986, terkenal sebagai operator deteksi tepi yang optimal, Abdul Kadir (2013:361). Salah satu algoritma deteksi tepi modern adalah deteksi tepi dengan menggunakan metode *Canny*.

C. Operator Sobel

Operator *Sobel* adalah satu cara untuk menghindari *gradien* yang dihitung pada titik interpolasi dari *pixel-pixel* yang terlibat dengan cara menghaluskan citra digital. Proses penghalusan yang digunakan merupakan proses konvolusi dari jendela yang ditetapkan terhadap citra yang dideteksi dengan menggunakan jendela 3x3 untuk perhitungan *gradien*, sehingga perkiraan *gradien* berada tepat ditengah jendela.

III. ANALISA DAN PEMBAHASAN

Sebelumnya perlu diketahui cara untuk mendeteksi tepi citra, tahapan yang digunakan untuk mendeteksi tepi citra menggunakan dua operator yaitu operator *canny* dan *sobel* yang mana tahap ekstraksi warna, tahap *cropping*, tahap deteksi tepi menggunakan metode *canny* dan metode *sobel*, tahap operasi dilandasi citra, tahap pengisian objek, tahap segmentasi citra dan tahap pembuatan garis tepi. Dalam masalah ini, telah disiapkan sebuah citra digital yang mempunyai citra digital yang mempunyai dimensi 426 x 426 dengan format *.Jpg

Analisa Metode *canny*

Analisa proses yang akan dilakukan disini adalah analisa tentang bagaimana proses deteksi tepi citra digital dengan operator *canny*. Terdapat lima langkah yang dilakukan untuk mengimplementasikan deteksi tepi *canny*, yaitu:

Langkah 1 :

Pertama dilakukan penapisan terhadap citra dengan tujuan untuk menghilangkan derau menggunakan *filter Gaussian* dengan kadar sederhana dengan ketentuan kadar yang digunakan berukuran jauh lebih kecil dari pada ukuran citra.

Langkah 2

Setelah penghalus gambar terhadap derau dilakukan, selanjutnya proses mendapatkan kekutan tepi (*edge strenght*) dengan menggunakan operator *Gaussian*. Gradien citra dapat dihitung dengan rumus:

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

Langkah 3

Menghitung arah tepi. Rumus yang digunakan adalah:

$$\theta = \tan^{-1}(G_x/G_y)$$

Langkah 4

Menghubungkan arah tepi dengan sebuah arah yang dapat dilacak citra.

Langkah 5

Proses *hysteresis*, proses ini menghilangkan garis-garis yang terputus-putus

Berdasarkan penjelasan langkah diatas, untuk mendeteksi tepi dengan metode *Canny*, kita akan menggunakan gradien $G(x,y)$ yang merupakan sebuah *vektor* yang terdiri dari dua unsur yaitu G_x dan G_y . Deteksi tepi dilakukan dengan cara membaca setiap *pixel* pada citra dengan cara

membaca dari *pixel* paling kiri atas (timur utara) dan bergerak ke *pixel* paling kanan bawah (barat selatan). Oleh karena itu, untuk membantu penelusuran tepi, gradien G_x dan G_y masing-masing dihitung dengan *matriks* operator *Canny Mask* 3x3.

Tabel 1. Matrix 3x3 operator *canny*

C_x	1	0	-1	C_y	-1	-1	-1
	1	0	-1		0	0	0
	1	0	-1		1	1	1

Analisa Metode *Sobel*

Analisa proses yang akan dilakukan disini adalah analisa tentang bagaimana proses deteksi tepi citra digital dengan metode *sobel*. Untuk mendeteksi tepi dengan metode *Sobel*, kita akan menggunakan gradien $G(x,y)$ sama halnya dengan *canny*, yang merupakan sebuah *vektor* yang terdiri dari dua unsur yaitu G_x dan G_y . Deteksi tepi dilakukan dengan cara membaca setiap *pixel* pada citra dengan cara membaca dari *pixel* paling kiri atas (timur utara) dan bergerak ke *pixel* paling kanan bawah (barat selatan). Oleh karena itu, untuk membantu penelusuran tepi, gradien G_x dan G_y masing-masing dihitung dengan *matrix* metode *sobel Mask* 3x3. 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. Nilai C konstanta bernilai dua, sehingga terbentuk *matriks* operator *sobel* seperti tabel di bawah ini :

Tabel 2. Matrix 3x3 Operator *Sobel*

$S_x =$	-1	0	1	$S_y =$	1	2	1
	-2	0	2		0	0	0
	-1	0	1		-1	-2	-1

dan

Desain logik pada penelitian ini menggunakan *tool* alat bantu perancangan yaitu *Unified Modelling Language (UML)*. Perancangan logik pada penelitian terdiri dari rancangan *Use Case* untuk segmentasi citra ke *grayscale*, rancangan diagram *use case* operator *canny* dan terakhir rancangan diagram *use case* operator *sobel*. Desain fisik menggambarkan tentang desain *Form* Utama, *Form* operator *canny* dan *Form* operator *sobel*.

Diagram Use Case *Grayscale*

Diagram *Use Case* Proses Segmentasi Citra ke *Grayscale*, proses yang dilakukan pertama-tama yaitu membuka gambar yang akan di segmentasikan lalu masukan objek citra tersebut kedalam program, setelah itu citra awal dirubah kedalam bentuk *grayscale* untuk mempermudah dalam segmentasi citra, setelah masuk pada proses segmetasi jika citra biner atau citra yang digunakan terhubung maka gambar tersebut bisa menjadi citra *grayscale* dengan pembagian *pixel* yang baik dan benar sehingga

menghasilkan gambar, jika tidak terhubung dengan bilangan biner 1 maka proses segmentasi dibatalkan. Lebih jelas dapat dilihat gambar *use case* di bawah ini :

Gambar 1: *Diagram Use Case* Proses Segmentasi Citra ke *Grayscale*

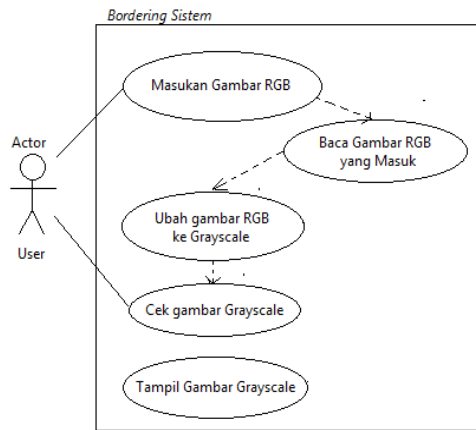
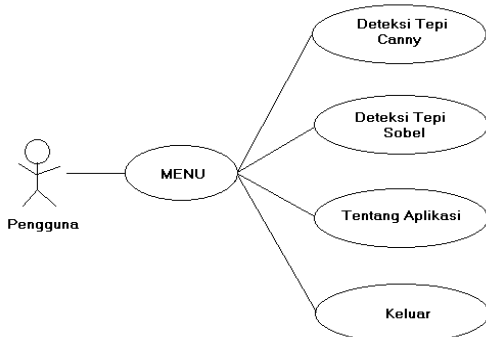


Diagram Use Case Menu Utama

Diagram use case menu utama menguraikan tentang uraian menu utama. Adapun langkah-langkah menu utama dapat dilihat pada gambar diagram *use case* berikut :

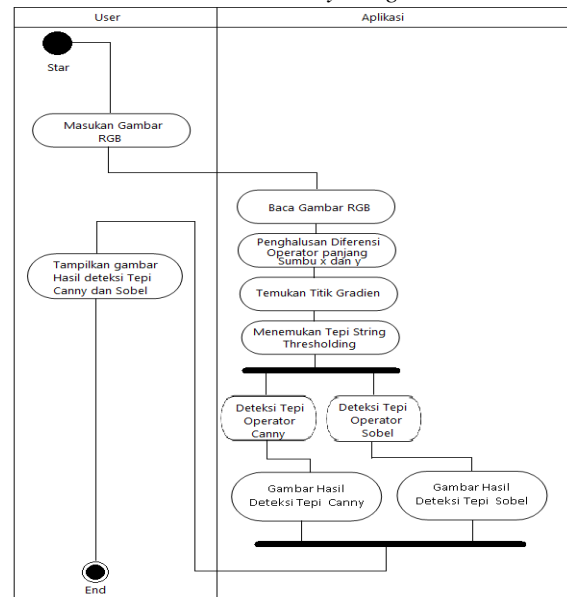
Gambar 2: *Diagram use case* Deteksi Tepi Operator Canny



Activity Diagram

Activity Diagram memperlihatkan aliaran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram aktifitas menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana berakhir.

Gambar 3: *Activity Diagram*



Pengujian sistem dilakukan setelah perancangan sistem selesai dilakukan. Pengujian sistem yang dilakukan dapat dilihat seperti berikut ini:

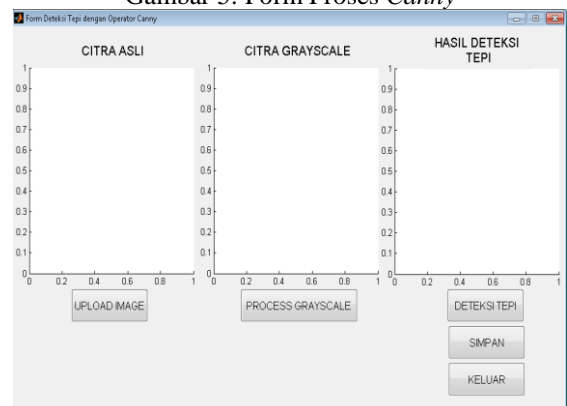
1. Form Utama

Gambar 4: *Form Utama*

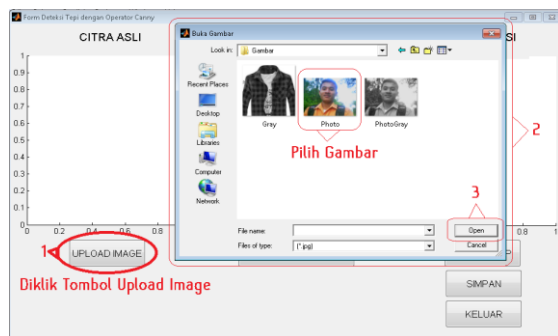


2. Form Canny

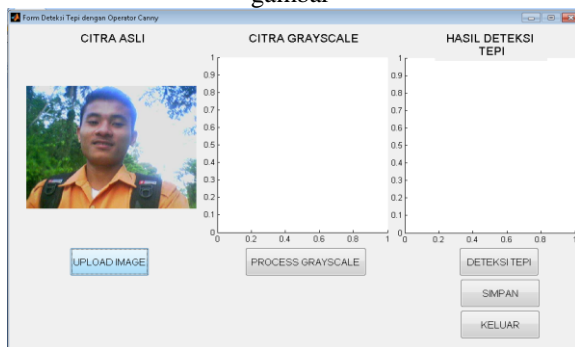
Gambar 5: *Form Proses Canny*



Gambar 6 : Tampilan form setelah klik tombol
“UPLOAD IMAGE”



Gambar 7: Tampilan form *canny* Setelah pemilihan gambar



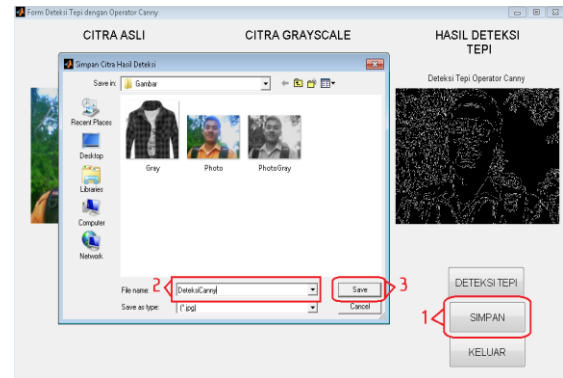
Gambar 8: Tampilan form setelah klik tombol



Gambar 9: Tampilan form setelah di klik tombol
“DETEKSI TEPI”



Gambar 10: Tampilan form setelah di klik tombol
“SIMPAN”

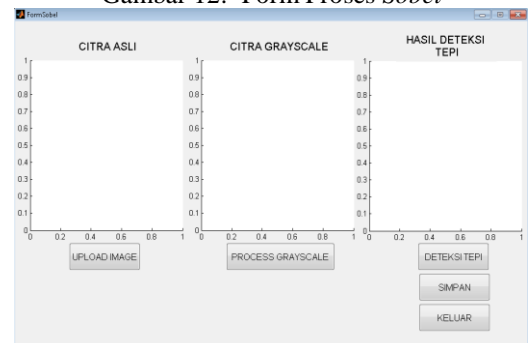


Gambar 11: Tampilan gambar hasil deteksi tepi
citra dengan metode *canny*



3. Form Sobel

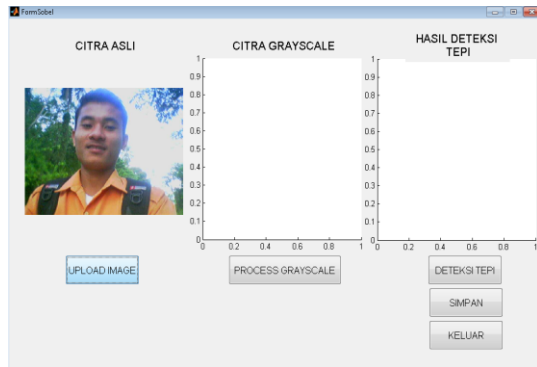
Gambar 12: Form Proses Sobel



Gambar 13: Tampilan form setelah klik tombol
“UPLOADIMAGE”



Gambar 14: Tampilan form *sobel* Setelah pemilihan
gambar



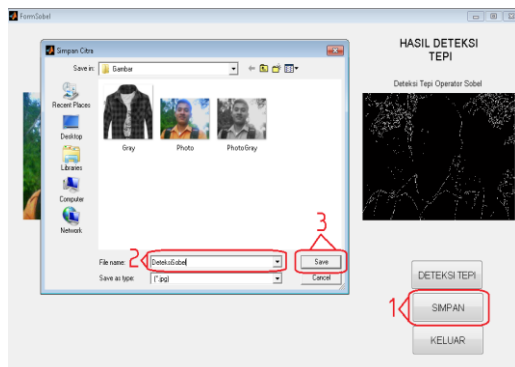
Gambar 15: Tampilan form setelah klik tombol “PROCESS GRAYSCALE”



Gambar 16: Tampilan form setelah diklik tombol “DETEKSI TEPI”



Gambar 17: Tampilan form setelah di klik tombol “SIMPAN”



Gambar 18. Tampilan gambar hasil deteksi tepi citra dengan metode *sobel*



Selanjutnya tombol proses “KELUAR” digunakan untuk keluar dari aplikasi deteksi tepi citra dengan metode *sobel*.

V. KESIMPULAN

Dari hasil analisa keluaran program maka dapat ditarik kesimpulan antara lain :

1. Program meningkatkan dan mendeteksi tepi gambar akan lebih baik jika masukan gambar mempunyai banyak tekstur.
2. Hasil dari gambar keluaran ditentukan dengan faktor perkalian kernel (*Mask*).
3. Dari hasil analisa keluaran program oprator *canny* dan operator *sobel* sama-sama mendeteksi tepi citra dengan baik.

VI. DAFTAR PUSTAKA

1. Erick Wijaya, *Analisis Intensitas Metode Pendeteksian Tepi Sobol*, 2012
2. Putra, *Defenisi Dari Pada Citra Dan Pergrtian Citra Analog*, Ardiansyah, 2007
3. Gonzales, Rafael C, Richard E.Woods, *Representasi Citra Digital*, 2005
4. Bayes, Gregory *Pengolahan Citra Dan Pengolahan Pola*, 1994
5. Adi Nugroho, *unified Modeling Language (UML)*, 2010
6. Wahana Komputer, *Matlab Pada Destop*, Andi, 2010
7. http://en.wikipedia.org/wiki/Canny_Operator
8. <http://www.academia.edu/5295802/> mengenal-
_Use_case_diagram