

Analisis Dan Perancangan Aplikasi Perbandingan Operator Edge Detection Pada Suatu Citra

Arief Mirmanda, Slamet Sugiarto

^{1,2}STMik Pontianak; Jl Merdeka Barat No. 372, (0561) 735555

³Jurusan Teknik Informatika, STMik Pontianak

e-mail: slamet.sugiarto@stmikpontianak.ac.id, ariefmirmanda.a.m@gmail.com

Abstrak

Deteksi tepi merupakan salah satu metode yang digunakan dalam pengolahan citra digital yang berfungsi untuk mengidentifikasi garis/tepi pada objek citra untuk menonjolkan informasi garis batas dari citra. Ada beberapa jenis operator edge detection yang dapat digunakan untuk mendeteksi garis tepi sebuah gambar. Penulis melakukan penelitian ini dengan menggunakan perbandingan operator Prewitt, Sobel, Robert, Isotropic, Laplace, Robinson dan Kirsch. Bentuk penelitian yang digunakan adalah studi literatur, dan metode penelitian yang digunakan adalah metode riset eksperimental. Sementara metode perancangan perangkat lunak yang digunakan ialah metode prototype, perancangan sistem ini menggunakan Flowchart sebagai alat bantu perancangan perangkat lunak. Hasil dari penelitian ini berupa perangkat lunak yang dapat melakukan pendeteksian garis tepi pada suatu citra digital. Metode pengujian perangkat lunak yang digunakan adalah metode Blackbox. Operator yang paling unggul dalam hal kecepatan proses adalah operator isotropik, karena operator isotropik mampu mendeteksi garis tepi pada citra digital yang beresolusi tinggi, sedang dan rendah dengan lebih cepat, sedangkan dari segi hasil deteksi tepi operator kirsch menjadi operator yang paling unggul, Karena deteksi tepi yang dihasilkan oleh operator kirsch lebih akurat, lebih detail dan lebih cerah, sehingga operator kirsch sangat cocok untuk melakukan pendeteksian terhadap citra digital dengan kualitas (resolusi) yang tinggi, sedang, dan rendah untuk mendapatkan hasil tepi yang optimal.

Kata kunci : Pengolahan Citra, Deteksi Tepi, Prototype, Flowchart

Abstract

Edge detection is one of the methods that are used in digital image processing that serves to identify lines / edges of the image object to highlight information lines of the image boundary. There are several kinds of edge detection operator that can be used to detect the outline of an image. The author conducted this study using a comparison operator Prewitt, Sobel, Robert, isotropic, Laplace, Robinson and Kirsch. Form of research is the study of literature, and the research method used is the method of experimental research. While the software design method used is prototype method, this system design using the Flowchart as software design tools. The results of the research IS software can perform edge detection lines on a digital image. Software testing method used is the Blackbox. THE most preeminent operator in terms of the speed of the process is the operator of the isotropic, because the operator isotropic able to detect the outline of the digital image in high resolution, medium and low more quickly, while in terms of the results of edge detection operator kirsch became the operator of the most superior, Because edge detection produced by the operator kirsch more accurate, more detailed and brighter, so that the operator kirsch is very suitable for the detection of digital image quality (resolution) of high, medium and low to get the optimal edge.

Kata kunci : Image Processing, Edge Detection, Prototype, Flowchart

1. PENDAHULUAN

Aplikasi pengolahan citra saat ini telah banyak digunakan hampir disemua bidang, antara lain adalah dibidang kedokteran, fotografi, militer dan geofisika. Bidang-bidang tersebut membutuhkan alat/kamera yang bisa digunakan untuk merakam keadaan yang diperlukan untuk kebutuhan analisis sehingga memungkinkan peneliti mendapatkan informasi yang diperlukan. Output alat-alat ini biasanya berupa citra. Citra inilah yang nantinya akan dianalisis untuk mendapatkan informasi yang berguna. Namun sayangnya, kebanyakan citra belum sesuai dengan hasil yang diharapkan. Hal ini dapat terjadi karena beberapa kemungkinan, misalnya adanya noise, adanya kabut yang menghalangi objek yang sedang di *capture*, lensa kamera kotor, dan lain-lain. Oleh sebab itu, proses pengolahan citra sangat diperlukan.

Edge detection / deteksi tepi merupakan salah satu metode yang digunakan dalam pengolahan citra digital yang berfungsi untuk mengidentifikasi garis/tepi pada objek citra untuk menonjolkan informasi garis batas dari citra. Deteksi tepi menjadi suatu yang penting karena manusia dalam mengenali objek suatu citra akan memperhatikan tepi dari objek tersebut, dengan adanya tepi dari objek, mata manusia akan mudah mengenali objek apa yang di tampilkan citra tersebut. Tujuan operasi pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Ada beberapa jenis metode *edge detection* yang dapat digunakan untuk mendeteksi garis tepi sebuah gambar, seperti operator *Krisch*, *Homogeneity*, *Horizontal*, *Vertical*, *Sobel*, *Isotropic*, *Prewitt*, *Robert*, *Laplace*, *Robinson*, *Krisch*, dan *Canny*. Masing-masing metode ini memiliki kelebihan dan kelemahan masing-masing.

Beberapa manfaat yang bisa diperoleh dari deteksi tepi adalah untuk menentukan stadium kanker, mendeteksi tepi citra USG janin, dan mendeteksi karies pada gigi, sehingga bentuk citra yang dihasilkan dapat terlihat lebih jelas, seperti penelitian yang telah dilakukan. Memiliki hasil bahwa operator *canny* dapat bekerja lebih baik dalam mendeteksi tumor. Deteksi tepi juga dapat digunakan untuk aplikasi pengenalan plat kendaraan, aplikasi pengenalan sidik jari, dan untuk membedakan uang asli dengan uang palsu [1]. Ada beberapa operator pendeteksian tepi dalam pengolahan citra, perlu pengetahuan yang tepat untuk menentukan operator yang sesuai dengan kondisi citra yang akan dideteksi. Dengan demikian tepi yang dihasilkan merupakan tepi yang optimal dan tepat. Apabila penggunaan operator deteksi tepi tidak tepat, akan menghasilkan pendeteksian yang gagal.

Berdasarkan keadaan inilah maka diperlukan pengetahuan atau informasi mengenai kinerja dari setiap operator, sehingga kesalahan atau ketidak sesuaian dalam pemilihan operator untuk pendeteksian tepi citra dapat dihindari.

Menurut penelitian sebelumnya, memberikan hasil deteksi tepi yang optimal dan mampu memberikan sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan [2]. Penelitian lainnya juga memiliki hasil bahwa penggunaan deteksi tepi *Canny* untuk segmentasi citra atau untuk *feature extraction* dapat meningkatkan akurasi dalam mengenali sebuah pola [3]. Berbeda lagi pada penelitian ini, hasil penelitian ini belum dapat menentukan tingkat hubungan keefektifan kinerja diantara keempat metode tersebut, namun telah dapat menyusun peringkat kinerjanya [4].

Bersumber dari penelitian inilah, penulis mencoba mengembangkan perangkat lunak pendeteksi tepi dengan melakukan penambahan operator-operator deteksi tepi seperti *Prewitt*, *Robert*, *Sobel*, *Isotropic*, *Laplace*, *Robinson* dan *Kirsch* dalam bentuk pengujian perbandingan terhadap metode-metode *edge detection*, sehingga dapat memberikan informasi yang akurat mengenai kinerja dari masing-masing operator untuk dikembangkan lebih lanjut. Perancangan perangkat lunak pendeteksi tepi pada pengolahan citra digital ini, difokuskan untuk mendapatkan operator manakah yang mampu memberikan kualitas detail tepi yang baik dan tanpa menghilangkan informasi utama dari citra dengan waktu seefisien mungkin.

2. METODE PENELITIAN

Bentuk penelitian yang dilakukan oleh penulis dalam penelitian ini adalah studi literatur. Dimana penelitian yang berkait erat dengan permasalahan yang ingin dipecahkan serta menentukan masalah yang ingin dipecahkan dengan melakukan eksperimen atau percobaan. Untuk menunjang penelitian, penulis melakukan pematatan teori dengan mencari informasi dan referensi yang diperlukan, dari buku-buku serta artiket-artikel yang berkaitan dengan penelitian.

Metode penelitian yang digunakan penulis adalah metode riset eksperimental, yaitu dengan melakukan percobaan pada objek dengan menggunakan teknik yang sudah ditentukan secara langsung, untuk mendapatkan hasil yang dicapai [5].

Metode pengumpulan data merupakan bagian yang terpenting dari sebuah penelitian ini. Ketersediaan data akan sangat menentukan dalam proses pengolahan data dan analisa selanjutnya. Karena dengan adanya pengumpulan data yang tepat maka diharapkan jawaban dari perumusan masalah tidak biasa. Data yang dikumpulkan sesuai dengan tujuan dari penelitian. Sumber data dari penelitian ini merupakan data primer dan data sekunder.

Data primer yang berkaitan langsung dengan data yang diperoleh dari observasi yaitu dengan mempelajari hasil dari program-program deteksi tepi yang sudah jadi. Sedangkan data sekunder berkaitan dengan semua hasil pengumpulan data yang mendukung data yang diperoleh dari studi dokumentasi dimana data diperoleh dari buku dan *internet* mengenai artiket-artikel, jurnal, dan adanya hasil dari penelitian sebelumnya yang dapat digunakan sebagai bahan perbandingan dengan penelitian yang dilakukan.

Metode analisis dan perancangan yang digunakan penulis untuk mengembangkan sistem adalah metode *prototype*. Metode *prototype* ini melakukan pendekatan secara sistematis dan urut, mulai dari level identifikasi kebutuhan sistem, kemudian tahap mengembangkan *prototype*, menguji kelayakan *prototype*, memprogram sistem berdasarkan kebutuhan, pengujian sistem, menentukan kelayakan sistem, dan penggunaan sistem [6].

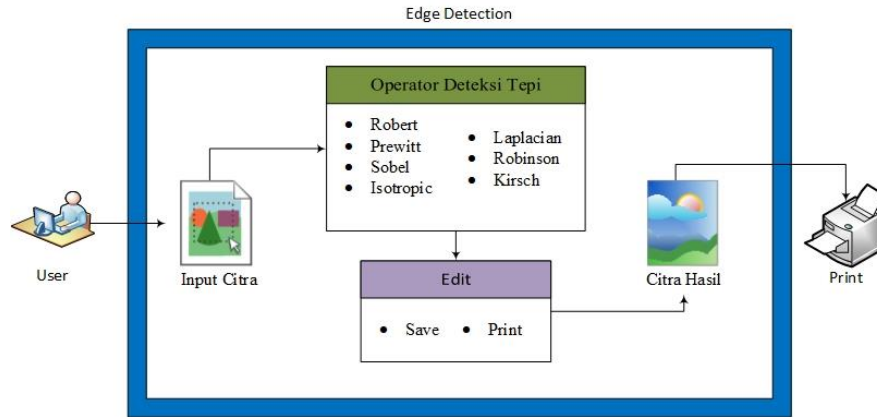
Pada penelitian ini penulis tidak menggunakan sesi wawancara kepada pengguna dikarenakan pada kasus ini perangkat lunak tidak bertujuan untuk kepuasan pengguna, melainkan bertujuan untuk menemukan operator yang lebih unggul dalam hal deteksi tepi, yang nantinya akan dikembangkan ke tahap selanjutnya.

3. HASIL DAN PEMBAHASAN

Pada tahap ini, penulis mendefinisikan fungsi-fungsi yang akan dipakai dalam pembuatan aplikasi. Penelitian bertujuan untuk merancang sebuah aplikasi yang dapat melakukan pendeteksian terhadap garis tepi sebuah citra digital dengan menggunakan metode Edge Detection *Prewit, Robert, Sobel, Isotropic, Laplace, Robinson dan Kirsch*. Adapun langkah-langkah yang dilakukan dalam menyelesaikan masalah ini adalah sebagai berikut.

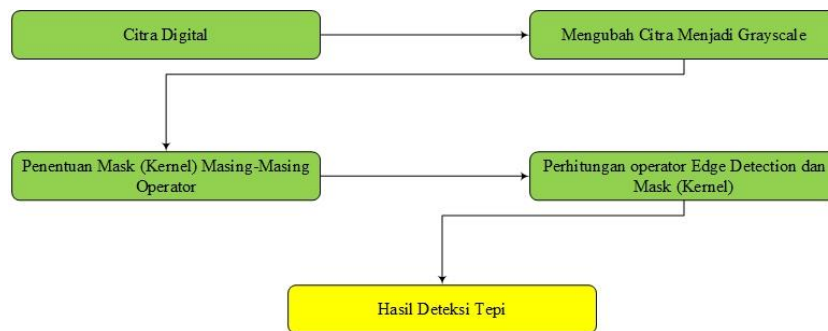
Mengumpulkan teori dan contoh-contoh kasus yang berhubungan dengan masalah pendeteksian garis tepi, citra digital, metode Edge Detection *Prewit, Robert, Sobel, Isotropic, Laplace, Robinson dan Kirsch*. Teori-teori ini dikumpulkan dari beberapa sumber seperti buku-buku di perpustakaan, artiket-artikel di internet serta referensi dari paper yang berhubungan dengan masalah yang dihadapi. Selain mengumpulkan teori-teori, juga dikumpulkan contoh-contoh kasus dalam bentuk jurnal penelitian sebagai referensi dalam memecahkan masalah pendeteksian garis tepi khususnya yang menggunakan metode Edge Detection *Prewit, Robert, Sobel, Isotropic, Laplace, Robinson dan Kirsch*.

Arsitektur perangkat lunak merupakan suatu pernyataan yang menggambarkan komponen perangkat lunak serta hubungan antara komponen tersebut. Agar perangkat lunak yang dibuat lebih mudah dipahami, berikut ini gambaran dari arsitektur perancangan perangkat lunak perbandingan operator *edge detection* pada suatu citra.



Gambar 1. Arsitektur Perangkat Lunak *Edge Detection*

Setelah mengetahui arsitektur dari perangkat lunak yang dirancang, dibutuhkan juga sebuah alur kerja sistem untuk menggambarkan secara jelas bagaimana cara kerja dari perangkat lunak tersebut. Alur kerja sistem ini terdiri dari citra digital, perhitungan dan pemetaan pixel pada suatu citra, penentuan *mask (kernel)* masing-masing operator, pembagian nilai pixel berdasarkan nilai RGB dan mendeteksi tepi setiap nilai RGB masing-masing pixel untuk menghasilkan garis tepi.



Gambar 2. Proses Deteksi Tepi Suatu Citra

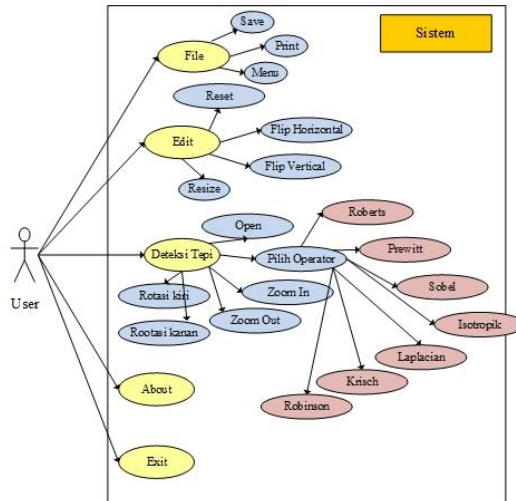
Dalam perancangan perangkat lunak *image processing* ini, diambil tiga perancangan *Unified Modeling Language (UML)*, yaitu *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram*.

Use case diagram berisi gambaran fungsionalitas yang diharapkan dari sebuah sistem dengan penekanan pada apa yang dilakukan oleh sistem. *Use case diagram* terdapat satu pihak yang berhubungan dengan *use case*. *Use case* adalah suatu *set scenario* yang dikumpulkan bersama-sama oleh hasil dari pengguna yang biasa terjadi. Diagram *use case* menggambarkan fungsi-fungsi dari sebuah sistem menggunakan aktor dan *use case*. *Use case* merupakan pelayan atau fungsi yang dimiliki oleh sistem untuk penggunaannya. Tujuan dari pembuatan *use case* adalah:

- a. Untuk memecah-mecah permintaan dari pengguna menjadi beberapa bagian yang memiliki kesatuan arti.
- b. Sebagai dasar dalam perencanaan konstruksi.
- c. Sebagai basis untuk mencoba sistem.

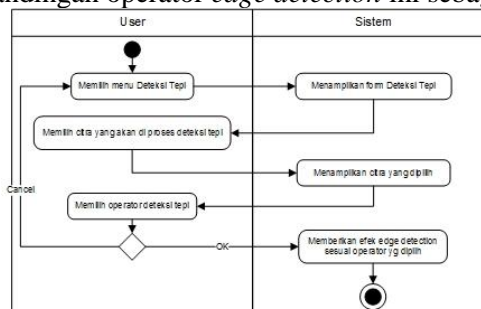
Analisis Dan Perancangan Aplikasi Perbandingan Operator Edge Detection Pada Suatu Citra

Penggunaan *use case diagram* disini untuk menjelaskan perancangan aplikasi manipulasi citra dengan metode konvolusi. Dalam model *use case*, aktor merupakan satu-satunya kesatuan eksternal yang berinteraksi dengan sistem. Adapun aktor yang berperan dapat dilihat pada gambar berikut:



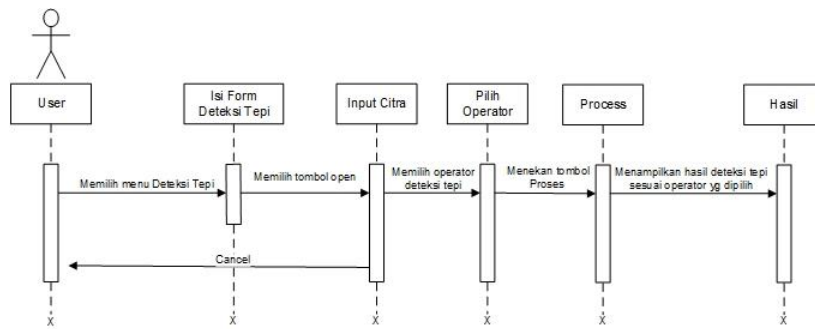
Gambar 3. Diagram Use Case User

Activity Diagram adalah teknik untuk menggambarkan logika *procedural*, proses bisnis dan aliran kerja dalam banyak kasus dalam sistem yang sedang dirancang dan bagaimana masing-masing berawal dan bagaimana sistem itu berakhir. *Activity Diagram* merupakan *state diagram* khusus dimana sebagian besar *state* adalah *action* dan sebagian transisi di *trigger* oleh selesainya *state* sebelumnya. Sebuah aktivitas dapat dijalankan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Gambaran *activity diagram* pada perangkat lunak perbandingan operator *edge detection* ini sebagai berikut:



Gambar 4. Activity Diagram Edge Detection

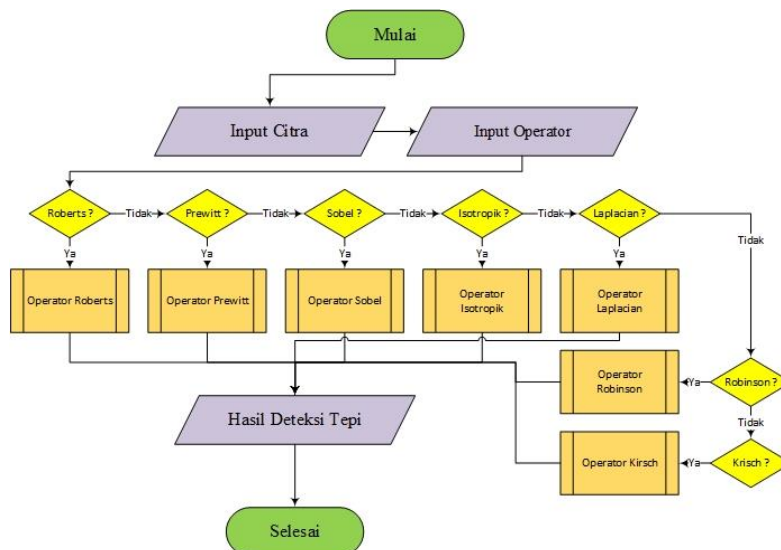
Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. *Sequence Diagram* digunakan untuk memberikan gambaran detail dari setiap *use case diagram* yang dibuat sebelumnya. Setiap objek yang terlihat dalam sebuah *use case* digambarkan dengan garis putus-putus vertikal, kemudian *message* yang dikirim oleh objek digambarkan dengan garis horizontal. Gambaran *Sequence Diagram* pada perangkat lunak perbandingan operator *edge detection* ini sebagai berikut:



Gambar 5. Sequence Diagram Edge Detection

Kemudian langkah selanjutnya adalah merancang *flowchart*, *flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan poses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah, setelah *flowchart* selesai disusun, selanjutnya pemrogram (*programmer*) menterjemahkannya ke bentuk program dengan bahasa pemrograman.

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. Form dan modul yang sudah didefinisikan sebelumnya beserta komponennya disatukan untuk membentuk suatu program utuh. Hubungan antar modul dengan *form* juga didefinisikan. Pembuatan *flowchart* (bagan alir) ini untuk mempermudah dalam perancangan perangkat lunak perbandingan operator *edge detection* pada citra digital. Berikut ini *Flowchart* pemilihan operator deteksi tepi :



Gambar 6. Flowchart Operator Deteksi Tepi

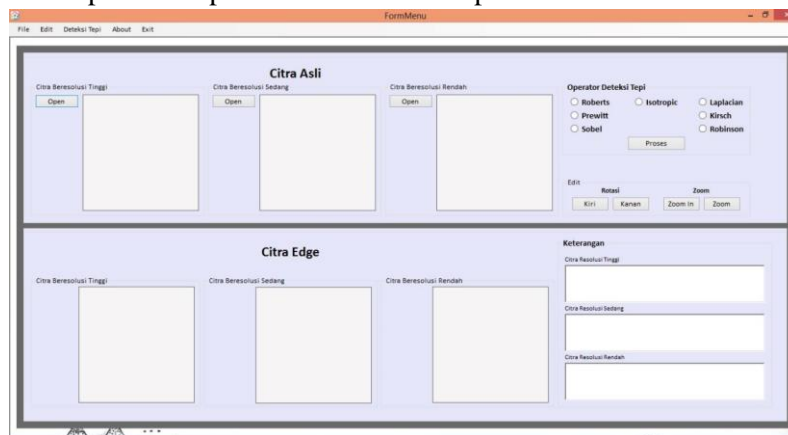
Berikut ini adalah algoritma dari *flowchart* proses pemilihan operator deteksi tepi :

- a. Mulai
- b. Memasukkan gambar yang akan di proses Deteksi Tepi
- c. Pilih operator deteksi tepi pada menu pilihan operator yang terdapat pada perangkat lunak
- d. Jika operator *Roberts* yang dipilih, maka perangkat lunak akan menjalankan proses operator *Robert*.

- e. Jika operator *Prewitt* yang dipilih pada menu pilihan operator, maka perangkat lunak akan menjalankan proses operator *Prewitt*.
- f. Jika operator *Sobel* yang dipilih pada menu pilihan operator, maka perangkat lunak akan menjalankan proses operator *Sobel*.
- g. Jika operator *Isotropik* yang dipilih pada menu pilihan operator, maka perangkat lunak akan menjalankan proses operator *Isotropik*.
- h. Jika operator *Laplacian* yang dipilih pada menu pilihan operator, maka perangkat lunak akan menjalankan proses operator *Laplacian*.
- i. Jika operator *Krisch* yang dipilih pada menu pilihan operator, maka perangkat lunak akan menjalankan proses operator *Krisch*.
- j. Jika operator *Robinson* yang dipilih pada menu pilihan operator, maka perangkat lunak akan menjalankan proses operator *Robinson*.
- k. Setelah perhitungan operator yang dipilih selesai, maka perangkat lunak akan menampilkan hasil deteksi tepi sesuai dengan operator yang dipilih.

Penulis membangun perangkat lunak *edge detection* pada objek citra digital ini dengan menggunakan Visual Studio .NET 2010 sebagai compiler. Instrumen penelitian ini berupa form-form perangkat lunak dan *coding* dari perangkat lunak tersebut.

Berikut merupakan tampilan Form Deteksi Tepi:



Gambar 7. Form Deteksi Tepi

Setelah modul dirancang ke dalam program tersebut, berikutnya melakukan *testing* pada form yang membuat modul tersebut. Setelah setiap modul dan form terbentuk dan diuji, semua modul dan form tersebut kemudian disatukan dan dilakukan pengujian kembali akan integritasnya, termasuk didalamnya pengujian validitas input tiap form.

Pengujian terhadap suatu perangkat lunak bertujuan untuk melihat apakah perangkat lunak tersebut berfungsi sebagai aplikasi pengolahan citra yang dapat mengenal suatu citra dengan metode yang diinginkan. Pada pengujian perangkat lunak pengolahan citra ini, digunakan metode pengujian *blackbox*. Tipe *file* yang akan diuji adalah jpg (jpeg). Dalam tahap ini, telah disiapkan sebuah 3 buah citra digital yang mempunyai 3 kualitas berbeda (tinggi, sedang dan rendah). Adapun tampilan dari citra digital yang akan dijadikan perangkat pengujian ssebagai berikut:

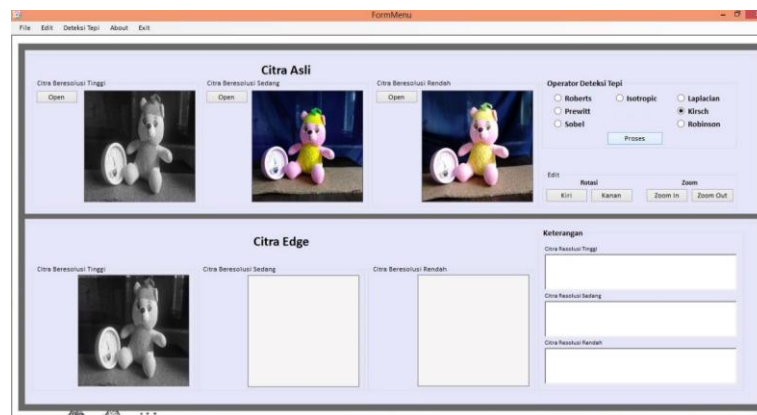


Gambar 8. Citra Beresolusi Rendah Gambar 9. Citra Beresolusi Sedang



Gambar 10. Citra Beresolusi Tinggi

Selanjutnya dilakukan pengujian terhadap kemampuan sistem dalam melakukan konversi citra asli ke citra *grayscale*. Pada tahap ini, akan diuji parameter kestabilan sistem, serta kestabilan hasil yang diharapkan. Adapun hasil pengujian yang diperoleh sebagai berikut:



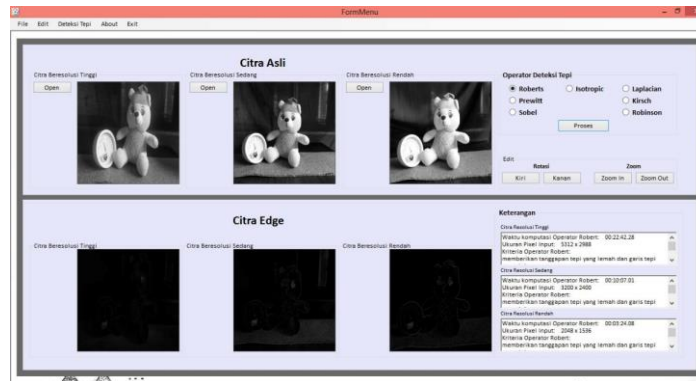
Gambar 11. Hasil Pengujian Grayscale

Pada pengujian ini, terlihat system telah dapat melakukan konversi citra digital. Berdasarkan hasil tampilan tersebut konversi citra telah dapat dilakukan dengan sempurna, sehingga diambil kesimpulan bahwa perangkat lunak ini tidak memiliki masalah dalam melakukan konversi citra asli ke *grayscale*.

Selanjutnya dilakukan pengujian terhadap kemampuan sistem dalam melakukan deteksi tepi citra dengan 3 resolusi berbeda yang sudah disiapkan. Pada tahap ini, akan diuji parameter kestabilan sistem, serta kestabilan hasil yang diharapkan. Adapun hasil pengujian yang diperoleh sebagai berikut:

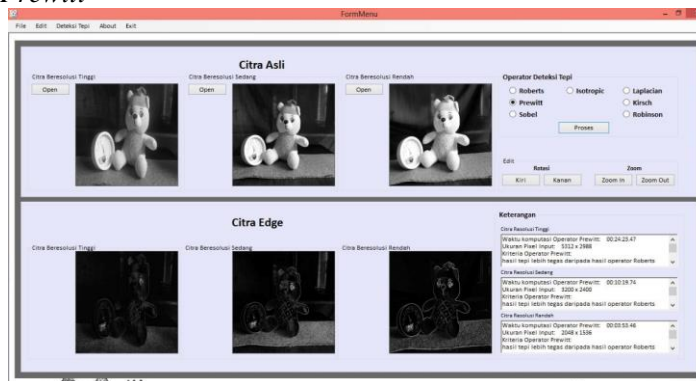
Operator *Roberts*

Analisis Dan Perancangan Aplikasi Perbandingan Operator Edge Detection Pada Suatu Citra



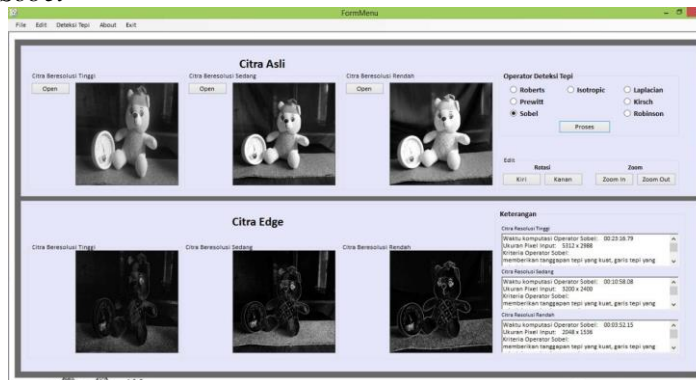
Gambar 12. Hasil Pengujian Operator *Roberts*

Operator *Prewitt*



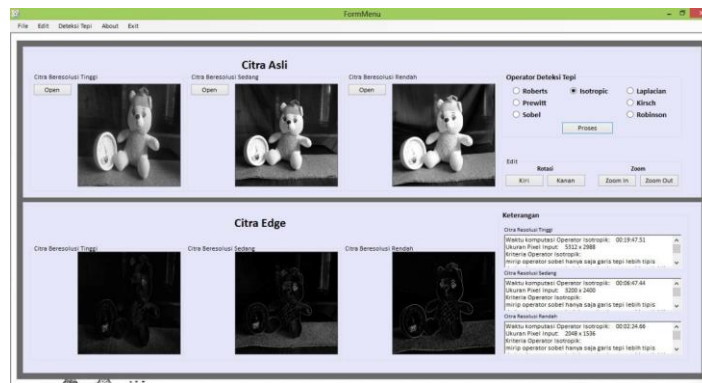
Gambar 13. Hasil Pengujian Operator *Prewitt*

Operator *Sobel*



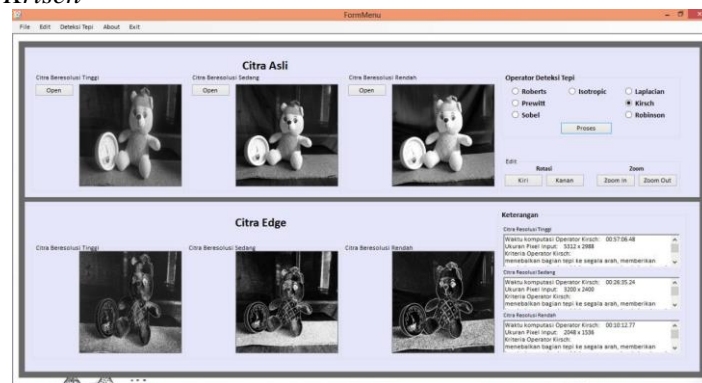
Gambar 14. Hasil Pengujian Operator *Sobel*

Operator *Isotropic*



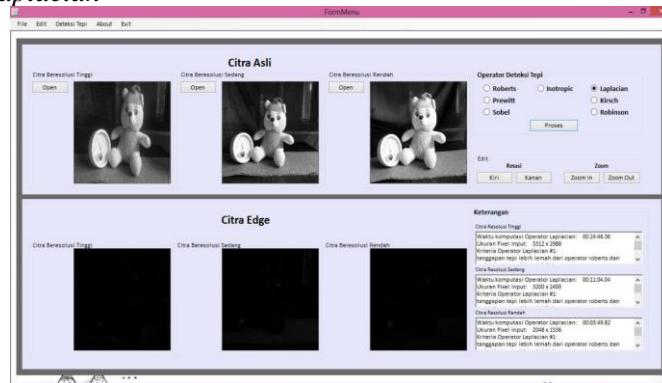
Gambar 15. Hasil Pengujian Operator *Isotropic*

Operator *Krisch*



Gambar 16. Hasil Pengujian Operator *Krisch*

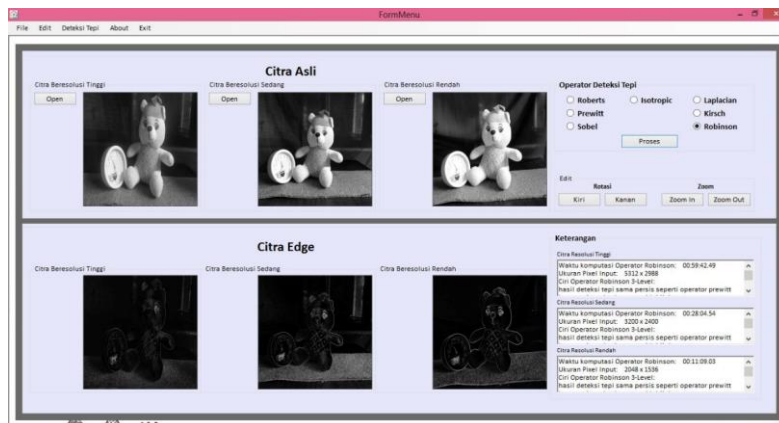
Operator *Laplacian*



Gambar 17. Hasil Pengujian Operator *Laplacian*

Operator *Robinson*

Analisis Dan Perancangan Aplikasi Perbandingan Operator Edge Detection Pada Suatu Citra



Gambar 18. Hasil Pengujian Operator *Robinson*

Dari hasil pengujian tersebut terlihat system telah dapat melakukan deteksi tepi menggunakan operator *Robert*, *Prewitt*, *Sobel*, *Isotropic*, *Laplacian*, *Krisch* dan *Robinson* terhadap citra digital. Berdasarkan hasil tampilan tersebut deteksi tepi terhadap citra telah dapat dilakukan dengan sempurna, sehingga dapat diambil kesimpulan bahwa proses deteksi tepi tidak memiliki masalah dalam melakukan deteksi tepi citra digital.

Penelitian ini menggunakan dua parameter, yaitu kecepatan proses dan perbandingan hasil deteksi tepi pada suatu citra. Dengan menggunakan parameter ini, dapat dilihat perbandingan dalam proses waktu yang diperlukan untuk eksekusi citra. Kecepatan proses dapat dihitung dengan:

$$\text{ExecutionTime} = \text{selesai} - \text{mulai}$$

Dengan *executionTime* merupakan waktu yang diperlukan untuk melakukan setiap proses deteksi tepi pada suatu citra, selesai merupakan waktu berakhir dan mulai merupakan waktu mulai dalam proses deteksi tepi. Dari hasil pengujian yang dilakukan pada tiga jenis citra dengan resolusi yang berbeda-beda yang telah disiapkan untuk setiap operator deteksi tepi, hasil perbandingannya, yaitu sebagai berikut :

Tabel 1. Pengujian Perbandingan Kecepatan Proses Resolusi Tinggi

Resolusi Citra	Operator Deteksi Tepi	Kecepatan Proses (Menit)
5312 x 2988	Isotropik	19:47
	Roberts	22:42
	Sobel	23:16
	Prewitt	24:23
	Laplacian#2	24:45
	Laplacian#1	24:48
	Krisch	57:06
	Robinson 5 Lv	57:30
Robinson 3 Lv	59:42	

Tabel 2. Pengujian Perbandingan Kecepatan Proses Resolusi Sedang

Resolusi Citra	Operator Deteksi Tepi	Kecepatan Proses (Menit)
3200 x 2400	Isotropik	06:47
	Roberts	10:07
	Prewitt	10:19
	Laplacian#2	10:30
	Sobel	10:58
	Laplacian#1	11:04
	Robinson 5 Lv	24:53
	Kirsch	26:35
	Robinson 3 Lv	28:04

Tabel 3. Pengujian Perbandingan Kecepatan Proses Resolusi Rendah

Resolusi Citra	Operator Deteksi Tepi	Kecepatan Proses (Menit)
2048 x 1536	Isotropik	02:24
	Roberts	03:24
	Laplacian#1	03:49
	Sobel	03:52
	Prewitt	03:53
	Laplacian#2	03:57
	Kirsch	10:12
	Robinson 5 Lv	10:18
	Robinson 3 Lv	11:09

Dari hasil pengujian kecepatan proses deteksi tepi diatas dapat disimpulkan bahwa operator Isotropik adalah operator paling cepat dalam melakukan komputasi deteksi tepi dengan gambar beresolusi tinggi, sedang dan rendah. Besarnya resolusi citra mempengaruhi proses deteksi tepi, ini dikarenakan proses deteksi tepi merupakan pengolahan citra yang mengambil nilai setiap piksel pada citra dan secara otomatis besarnya resolusi citra sangat mempengaruhi proses waktu eksekusi citra.

Pengujian hasil deteksi tepi dari masing-masing operator dilakukan dengan gambar dari hasil kamera *handphone* yang berbeda untuk mewakili kualitas resolusi tinggi, sedang dan rendah, untuk citra beresolusi tinggi menggunakan kamera Samsung Note 4 dengan resolusi gambar 5312 x 2988, untuk citra beresolusi sedang menggunakan kamera Xiaomi Redmi 2 dengan resolusi gambar 3200 x 2400 sedangkan untuk citra beresolusi rendah menggunakan kamera Sony Xperia E1 dengan resolusi gambar 2048 x 1536.

Hasil setelah melakukan pengujian bahwa operator isotropik adalah operator yang paling unggul dalam hal kecepatan proses dibandingkan dengan operator roberts, prewitt, sobel, laplacian, kirsch dan robinson saat mendeteksi citra digital beresolusi tinggi, sedang dan rendah.

Sedangkan operator kirsch menjadi operator yang paling unggul jika dilihat dari segi hasil deteksi tepi dibandingkan dengan operator roberts, prewitt, sobel, isotropiks, laplacian dan robinson. Hasil garis tepi yang dihasilkan oleh operator kirsch lebih tebal, lebih akurat, lebih detail dan lebih cerah, sehingga operator kirsch sangat cocok untuk melakukan pendeteksian terhadap citra digital dengan kualitas (resolusi) yang tinggi, sedang, dan rendah untuk mendapatkan hasil tepi yang optimal.

4. KESIMPULAN

Besarnya resolusi citra mempengaruhi waktu proses deteksi tepi, ini dikarenakan proses deteksi tepi merupakan pengolahan citra yang mengambil nilai setiap piksel pada citra dan secara otomatis besar dan kecilnya resolusi citra sangat mempengaruhi waktu proses eksekusi citra. Besarnya resolusi citra juga mempengaruhi tebal dan tipisnya garis tepi, semakin besar resolusi citra maka semakin tipis juga garis tepi yang dihasilkan. Ini dikarenakan semakin besar resolusi sebuah gambar maka jumlah piksel gambar tersebut semakin banyak mengakibatkan garis tepi terlihat samar bahkan tidak terlihat.

Berdasarkan ukuran gambar dengan range lebih dari 2000 piksel maka, Operator yang paling unggul dalam hal kecepatan proses adalah operator isotropik, karena operator isotropik mampu mendeteksi garis tepi pada citra digital yang beresolusi tinggi, sedang dan rendah dengan lebih cepat dibandingkan operator lain, sedangkan dari hasil deteksi tepi, operator kirsch menjadi operator yang paling unggul dibandingkan dengan operator lainnya. Karena deteksi tepi yang dihasilkan oleh operator kirsch lebih akurat, lebih detail, lebih tebal dan lebih cerah, sehingga operator kirsch sangat cocok untuk melakukan pendeteksian terhadap citra digital dengan kualitas (resolusi) yang tinggi, sedang, dan rendah untuk mendapatkan hasil garis tepi yang optimal.

5. SARAN

Perangkat lunak ini dapat dikembangkan dengan menambah format citra digital yang dapat dideteksi seperti gambar bergerak (GIF) atau video. Perangkat lunak ini dapat dikembangkan dengan menambahkan operator edge detection lain untuk mendapatkan hasil perbandingan yang lebih baik. Perangkat lunak ini memerlukan waktu yang cukup lama dalam mendeteksi garis tepi pada citra digital. Penulis berharap perangkat lunak ini dapat dikembangkan agar proses deteksi tepi bisa lebih cepat. Operator yang paling unggul dapat dikembangkan dalam ilmu pengolahan citra dan dibidang-bidang seperti photographer, kedokteran dll.

DAFTAR PUSTAKA

- [1] Suryaningsih, Fitri., komparasi algoritma deteksi tepi (edge detection) untuk segmentasi citra tumor hepar, Nomor 01, Volume (06): 178-315 Juni 2012.
- [2] Winarno, Edy, Aplikasi Deteksi Tepi pada Realtime Video menggunakan Algoritma Canny Detection, *Jurnal Teknologi Informasi Dinamik*, Nomor 1, Volume (16): 44-49, Januari 2011.
- [3] Yodha, Johannes Widagdho, dan Achmad Wahid Kurniawan, perbandingan penggunaan deteksi tepi dengan metode laplace, sobel dan prewit dan canny pada pengenalan pola, Nomor 3, Volume (13): 189-197, Agustus 2014.
- [4] Lusiana, Veronica, Deteksi Tepi pada Citra Digital menggunakan Metode Kirsch dan Robinson, *Jurnal Teknologi Informasi Dinamik*, Nomor.2, Volume (18): 182-189, Juli 2013.
- [5] Sommerville, I., 2011. *Software Engineering*. Ninth Edition, Addison-Wesley.
- [6] Pressman, Roger S, 2002, *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*, CN Harnaningrum, Andi Offset, Yogyakarta.