

ANALISIS IDENTIFIKASI POLA WARNA IKAN KOI MENGGUNAKAN METODE SOBEL EDGE DETECTION DALAM KARAKTERISTIK CITRA SHARPENING

Milfa Yetri^{#1} Yusnidah^{*2} Mukhlis Ramadhan^{*3}

[#]Program Studi Sistem Informasi, STMIKTriguna Dharma

^{*}Akademi Maritim Indonesia Medan

Jl. A. H. Nasution No. 73 F Medan Johor

¹*airputih.girl@gmail.com*

Abstrak

Deteksi tepi merupakan suatu proses pengolahan citra yang bertujuan untuk membedakan antara objek dengan latar belakang yang ada pada sebuah citra, sehingga proses ini dapat diimplementasikan untuk kepentingan tertentu, seperti pengenalan pola, deteksi wajah pada kamera dan sebagainya. Beberapa metode pendeteksian tepi memiliki kemampuan yang berbeda-beda dalam proses pengolahan citra untuk mendapatkan pola garis tepi. Salah satu metode yang digunakan pada penelitian ini adalah metode Sobel dengan objek penelitian menggunakan citra ikan koi. Kemampuan metode ini dibuktikan dengan menganalisis pola warna ikan yang terdapat pada objek citra yang telah mengalami proses karakteristik citra Shapening (penajaman citra). Hasilnya metode Sobel mendapatkan pola garis tepi warna ikan koi lebih baik dan akurat.

Kata Kunci : Pengolahan Citra, Deteksi Tepi, Pola Warna Ikan.

Abstract

Edge detection image processing is a process that aims to differentiate between objects with no background in an image, so that this process can be implemented for specific interests, such as pattern recognition, face detection on the camera and so on. Several edge detection methods have different capabilities in image processing to get a line pattern edge. One of the methods used in this study is the object of research Sobel method using image koi fish. The ability of this method is proved by analyzing the color patterns of fish contained in the image of the object that has mengalamii process Shapening image characteristics (image enhancement). The result Sobel method to get the outline of the pattern color of koi fish better and accurate.

Keywords : Image Processing, Edge Detection, Fish Color Pattern

PENDAHULUAN

Ikan koi merupakan salah satu primadona di kelas ikan hias. Bentuknya yang mirip ikan mas dengan warna dan jenis sisik

yang beragam menambah daya tarik ikan ini. Tidak hanya warna dan pola sisik yang

menarik, ikan koi ini juga terkenal dengan ukuran tubuhnya yang mampu mencapai ukuran jumbo.

Nishikigoi adalah nama jepangnya untuk Koi, yang mulai dikembangkan di Jepang 200 tahun yang lalu. Awalnya petani beras di Jepang melihat bahwa beberapa ikan di sawahnya memiliki warna yang cemerlang, lalu diambil dan dipeliharanya.

Pada tahun 1914, ketika Niigata Koi memamerkan dalam suatu expo, mulailah terjadi demam memelihara koi di seluruh Jepang. Hobi memelihara Koi tersebar keseluruh penjuru dunia setelah kantong plastik dan sarana transport sudah memadai untuk pengiriman ikan dengan selamat.

Memelihara Koi adalah hobi yang menyenangkan dan diyakini dapat mengurangi tingkat stress. Koi adalah ikan yang pintar dan bisa diajarkan untuk makan dari tangan anda. Namun kadang seperti ikan rakus yang akan memakan apa saja yang anda lemparkan ke kolam.

Koi juga bisa mendengar dan akan merespon suara-suara, terutama suara dari pemiliknya.

Di Jepang, ikan ini dinamakan Nishikigoi (*Cyprinus carpadie*). Artinya, ikan berwarna warni. Goi sendiri artinya ikan karper. Koi sendiri berasal dari bahasa Cina. Kata koi berasal dari bahasa Jepang yang berarti ikan karper. Lebih spesifik lagi ia merujuk pada nishikigoi yang kurang lebih bermakna ikan karper yang bersulam emas atau perak. Di Jepang, koi menjadi semacam simbol cinta dan persahabatan. Ini karena koi merupakan homofon untuk kata lain yang juga bermakna kasih sayang atau cinta.

Koi dikembangkan dari ikan karper biasa. Pada tahun 1820-an, di Jepang mulai muncul usaha mengembangkan ikan karper untuk warnanya. Ini bermula di kota Ojiya di prefektur Niigata yang berada di wilayah timur laut Pulau Honshu.

LANDASAN TEORI

1. Pengertian Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan.

Suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitud f di titik koordinat x,y dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar X, foto yang tercetak di kertas foto, lukisan, pemandangan alam, hasil CT scan, gambar-gambar yang terekam pada pita kaset, dan lain sebagainya. Citra analog tidak dapat dipresentasikan dalam komputer sehingga tidak bisa diproses secara langsung. Oleh sebab itu, agar citra ini dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu.

Defenisi Citra Digital

Secara umum, pengolahan citra digital menunjukkan pada pemrosesan gambar dua dimensi menggunakan komputer. Dalam konteks yang lebih luas pengolahan citra digital mengacu pada pemrosesan setiap data dua dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu.

Pembentukan citra digital (diskrit) melalui beberapa tahapan yaitu akuisisi citra, sampling, dan kuantisasi.

Akuisisi citra

Proses akuisisi citra adalah pemetaan suatu pandangan (*scene*) menjadi citra kontinu yang menggunakan sensor.

Sampling adalah transformasi citra kontinu menjadi citra digital dengan cara membagi citra analog (kontinu) menjadi M kolom dan N baris sehingga menjadi citra diskrit. Semakin besar nilai M dan N, semakin halus citra digital yang dihasilkan dan artinya resolusi citra semakin tinggi. Persilangan antara baris dan kolom tertentu disebut dengan piksel.

Proses kuantisasi adalah proses perubahan nilai amplitudo kontinu menjadi nilai baru yang berupa nilai diskrit. Nilai amplitudo yang dikuantisasi adalah nilai-nilai pada koordinat diskrit hasil proses.

2. Jenis-Jenis Citra

a. Citra Biner

Citra Biner adalah citra digital yang hanya dapat memiliki dua kemungkinan nilai pixel yaitu hitam putih. Citra biner juga disebut dengan citra *B & W* (*black and white*) atau citra monokrom. Hanya dibutuhkan nilai 1 bit untuk mewakili nilai setiap pixel dari data biner. Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengambangan, morfologi, ataupun dithering.

b. Citra Grayscale

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian *RED = GREEN = BLUE* Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan di sini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih.

Citra grayscale berikut memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan).

c. Citra Warna

Setiap pixel dari citra warna 24 bit diwakili 24 bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat dilihat penglihatan manusia. penglihatan manusia dipercaya hanya dapat membedakan hingga 10 juta warna saja. Setiap poin informasi pixel (RGB) disimpan ke dalam 1 byte data. 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai hijau pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah.

3. Deteksi Tepi (*Edge Detection*)

Edge Detection merupakan salah satu proses yang fundamental dalam Pengolahan Citra yang bertujuan mengidentifikasi titik-titik pada citra digital dimana tingkat kecerahan (*brightness*) berubah drastis atau terjadi diskontinuitas. *Edge Detection* bertujuan membentuk sejumlah kurva yang saling terhubung yang mengindikasikan batas-batas objek, tanda-tanda permukaan, serta kurva-kurva yang mengindikasikan diskontinuitas pada orientasi permukaan. Penerapan *Edge Detection* secara signifikan dapat mengurangi jumlah data yang diproses dan menyaring informasi yang mungkin dianggap kurang relevan, sambil menjaga sifat struktural penting dari suatu gambar. *Edge Detection* banyak digunakan dalam aplikasi.

Metode pendeteksian tepi Sobel adalah metode pendeteksian tepi terbaik dibandingkan dgn metode pendeteksian tepi lainnya seperti metode Prewitt dan metode Robert. Jika pada metode Robert atau Prewitt hanya menggunakan jendela 2x2 pada maskingnya, maka pada metode Sobel menggunakan jendela 3x3 pada maskingnya sehingga hasil yang di dapat menjadi lebih

sempurna dari metode-metode pendeteksian tepi yang lainnya.

Bagaimana jika masking pada sobel ini

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

kita kalikan dengan suatu nilai (0-255) sehingga kita dapat mengubah kekuatan intensitas pendeteksian tepi pada metode sobel agar mendapatkan hasil yang terbaik dari metode pendeteksian tepi sobel dengan melihat nilai-nilai yang didapat.

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } H = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Perhatikanlah bahwa operator sobel menempatkan penekanan atau pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat jendela. Dengan demikian pengaruh piksel-piksel tetangga akan berbeda sesuai dengan letaknya terhadap titik di mana gradien dihitung. Gradien adalah hasil pengukuran perubahan dalam sebuah fungsi intensitas, dan sebuah citra dapat dipandang sebagai kumpulan beberapa fungsi intensitas kontinyu dari citra. Dari susunan nilai-nilai pembobotan pada jendela juga terlihat bahwa perhitungan terhadap gradien juga merupakan gabungan dari posisi horisontal dan vertikal.

4. Penajaman Citra (Sharpening)

Prinsip penajaman citra adalah menjumlahkan citra asli dengan citra hasil dari operasi deteksi tepi. Operator deteksi tepi yang digunakan adalah operator Laplacian. Dengan cara ini bagian tepi objek akan tampak berbeda dengan latar belakangnya sehingga citra yang dihasilkan terkesan lebih tajam. Banyaknya penambahan komponen citra tepi diatur dengan suatu nilai yang disebut sebagai derajat ketajaman α . Besar kecilnya tingkat ketajaman bisa disesuaikan dengan keinginan

kita dengan mengatur nilai α . Secara matematis operasi ini ditulis sebagai berikut.

$$h(x, y) = f(x, y) + \alpha \cdot \nabla^2 f(x, y)$$

Di mana $\nabla^2 f(x, y)$ adalah Laplacian dari fungsi $f(x, y)$ yang mempunyai rumus berikut.

Berdasarkan persamaan ini maka filter untuk operator penajaman citra di titik (x, y) dapat dibuat sebagai berikut.

0	$-\alpha$	0
$-\alpha$	$1 + 4\alpha$	$-\alpha$
0	$-\alpha$	0

5. Format File Citra

Format file citra standar yang digunakan saat ini terdiri dari beberapa jenis. Format-format ini digunakan dalam menyimpan citra dalam sebuah file. Setiap format memiliki karakteristik masing-masing. Berikut adalah penjelasan beberapa format umum digunakan saat ini.

a. Bitmap (.bmp)

Format .bmp adalah format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai pixel.

b. Tagged Image Format (.tif, .tiff)

Format .tif merupakan format penyimpanan citra yang dapat digunakan untuk menyimpan citra bitmap sehingga citra dengan warna palet terkompresi. Format ini dapat digunakan untuk menyimpan citra yang tidak terkompresi dan juga citra terkompresi.

c. Portable Network Graphics (.png)

Format .png adalah format penyimpanan citra terkompresi. Format ini dapat digunakan pada citra grayscale, citra dengan

palet warna, dan juga citra fullcolor. Format .png juga mampu menyimpan informasi hingga kanal alpha dengan penyimpanan sebesar 1 hingga 16 bit per kanal.

d. JPEG (.jpg)

Format .jpg adalah format yang sangat umum digunakan saat ini khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG.

e. Graphics Interchange Format (.gif)

Format ini dapat digunakan pada citra warna dengan palet 8 bit. Penggunaan umumnya pada aplikasi web. Kualitas yang rendah menyebabkan format ini tidak terlalu populer di kalangan peneliti pengolahan citra digital.

f. RGB (.rgb)

Format ini merupakan format penyimpanan citra yang dibuat oleh silicon graphics untuk menyimpan citra berwarna.

PEMBAHASAN

1. Analisis Permasalahan

Adapun proses tahapan yang dilakukan dalam pendeteksian tepi pada citra bitmap adalah sebagai berikut:

- a. Tempatkan kernel pada sudut kiri atas.
- b. Geser kernel satu pixel ke kanan, kemudian hitung nilai pixel pada posisi (1, 1) dari kernel. Dan lakukan proses konvolusi dengan mengalikan matriks citra dengan matriks konvolusi. Lakukan proses ini sampai posisi (3,3) dari kernel berada pada posisi akhir (kanan) citra.
- c. Selanjutnya, geser kernel satu pixel ke bawah, lalu mulai lagi melakukan konvolusi dari sisi kiri citra. Setiap kali konvolusi, geser kernel satu pixel ke kanan seperti pada langkah ke dua. Lakukan proses ini sampai posisi (0, 3) dari kernel berada pada posisi akhir (bawah) citra.

Akuisisi citra adalah tahap awal untuk mendapatkan citra digital. Tujuan akuisisi citra

adalah untuk menentukan data yang diperlukan dan memilih metode perekaman citra digital (Darma Putra, 2010: 28). Tahap ini dimulai dari objek yang akan diambil gambarnya persiapan alat-alat, sampai pada pencitraan. Pencitraan adalah kegiatan transformasi dari citra tampak (foto, gambar, lukisan, patung, pemandangan, dan lain-lain) menjadi citra digital. Beberapa alat yang dapat digunakan untuk pencitraan adalah:

- a. Video kamera
- b. Kamera digital
- c. Kamera konvensional dan converter analog to digital
- d. Scanner
- e. Photo sinar-x /sinar infra merah

Setiap piksel mewakili tidak hanya satu titik dalam sebuah citra melainkan sebuah bagian berupa kotak yang merupakan bagian terkecil (sel). Nilai dari sebuah piksel haruslah dapat menunjukkan nilai rata-rata yang sama untuk seluruh bagian dari sel tersebut.

Pada citra 3D satuan atau bagian terkecilnya bukan lagi sebuah pixel melainkan sebuah voxel. Voxel adalah singkatan dari volume element. Posisi dalam voxel ditentukan dengan tiga buah variabel yaitu k yang menyatakan kedalaman (depth), m menyatakan posisi baris, dan n yang menyatakan posisi kolom. penggambarannya dapat dilakukan dengan sumbu kartesian. Berikut ini adalah penggambaran voxel.

Resolusi pixel merupakan perhitungan jumlah pixel dalam sebuah citra digital. Sebuah citra dengan tinggi N pixel dan lebar M pixel berarti memiliki resolusi sebesar M x N. Resolusi pixel akan memberikan dua buah angka integer yang secara berurutan akan mewakili jumlah pixel lebar dan jumlah pixel tinggi dari citra tersebut.

Pengertian lainnya dari resolusi pixel adalah merupakan hasil perkalian jumlah pixel lebar dan tingginya dan kemudian dibagi dengan 1 juta. Jenis resolusi pixel seperti ini sering kali dijumpai dalam kamera digital.

Suatu citra yang memiliki lebar 2048 pixel dan tinggi 1536 pixel maka akan memiliki total pixel sebanyak $2048 \times 1536 = 3.145.728$ pixel atau 3,1 mega pixel.

Perhitungan lainnya menyatakan dalam satuan pixel per inci. Satuan ini menyatakan banyaknya pixel yang ada sepanjang 1 inci baris dalam citra

Sebuah citra digital membedakan intensitas ke dalam beberapa spektrum. Citra multi spektrum akan memberikan spektrum atau panjang gelombang yang lebih baik yang akan digunakan untuk menampilkan warna.

Resolusi Temporal

Resolusi temporal berkaitan dengan video. Suatu video merupakan kumpulan frame statis yang berupa citra yang berurutan dan ditampilkan secara cepat. Resolusi temporal memberikan jumlah frame yang dapat ditampilkan setiap detik dengan satuan frame per second (fps).

Resolusi Radiometrik

Resolusi ini memberikan nilai atau tingkat kehalusan citra yang dapat ditampilkan dan biasanya ditampilkan dalam satuan bit contoh citra 8 bit dan citra 256 bit. Semakin tinggi resolusi radiometrik ini maka semakin baik perbedaan intensitas yang ditampilkan.

2. Algoritma Sistem

a. Proses Pendeteksian Tepi Dengan Penajaman Citra (*Sharpening*)

Berikut ini diberikan contoh kasus pendeteksian tepi dengan penajaman citra terhadap citra bitmap. Contoh sebuah citra awal $f(x, y)$ dengan ukuran 5 x 5 piksel yang diambil dari sebuah citra digital dengan nilai intensitas warna sebagai berikut:



Gambar 1. Contoh Citra Digital yang diproses

Y \ X	1	2	3	4	5
1	8	5	2	7	5
2	5	10	8	4	4
3	9	8	4	8	11
4	2	8	3	6	6
5	3	5	9	10	6

Dengan sebuah kernel $k(x, y)$ berukuran 3 x 3 dengan nilai sebagai berikut:

0	-1	0
-1	5	-1
0	-1	0

Maka proses pendeteksian tepi dengan penajaman citra (*sharpening*) yang terjadi adalah:

$$\begin{aligned}
 1. \text{ Nilai intensitas warna citra awal } f(1,1) - f(3,3) \text{ di kali dengan kernel } k(1,1) - k(3,3) \\
 f(x, y) * k(x, y) &= (0 \times 8) + (-1 \times 5) + (0 \times 2) \\
 &\quad + (-1 \times 5) + (5 \times 10) + (-1 \times 8) + (0 \times 9) + (-1 \times 8) + (0 \times 4) \\
 &= 0 + (-5) + 0 + (-5) + 50 + (-8) + 0 + (-8) + 0 \\
 &= 24
 \end{aligned}$$

$$\begin{aligned}
 2. \text{ Nilai intensitas warna citra awal } f(2,1) - f(4,3) \text{ di kali dengan kernel } k(1,1) - k(3,3) \\
 f(x, y) * k(x, y) &= (0 \times 5) + (-1 \times 2) + (0 \times 7) \\
 &\quad + (-1 \times 10) + (5 \times 8) + (-1 \times 4) + (0 \times 8) + (-1 \times 4) + (0 \times 8) \\
 &= 0 + (-2) + 0 + (-10) + 40 + (-4) + 0 + (-4) + 0 \\
 &= 20
 \end{aligned}$$

$$\begin{aligned}
 3. \text{ Nilai intensitas warna citra awal } f(3,1) - f(5,3) \text{ di kali dengan kernel } k(1,1) - k(3,3) \\
 f(x, y) * k(x, y) &= (0 \times 2) + (-1 \times 7) + (0 \times 5) \\
 &\quad + (-1 \times 8) + (5 \times 4) + (-1 \times 4) + (0 \times 4) + (-1 \times 8) + (0 \times 11) \\
 &= 0 + (-7) + 0 + (-8) + 20 + (-4) + 0 + (-8) + 0 \\
 &= |-7| = 7
 \end{aligned}$$

$$\begin{aligned}
 4. \text{ Nilai intensitas warna citra awal } f(1,2) - f(3,4) \text{ di kali dengan kernel } k(1,1) - k(3,3) \\
 f(x, y) * k(x, y) &= (0 \times 5) + (-1 \times 10) + (0 \times 8) \\
 &\quad + (-1 \times 9) + (5 \times 8) + (-
 \end{aligned}$$

$$\begin{aligned}
 &1 \times 4) + (0 \times 2) + (-1 \times 8) + (0 \times 3) \\
 &= 0 + (-10) + 0 + (-9) + 40 + (-4) + 0 + (-8) + 0 \\
 &= 9
 \end{aligned}$$

$$\begin{aligned}
 5. \text{ Nilai intensitas warna citra awal } f(2,2) - f(4,4) \text{ di kali dengan kernel } k(1,1) - k(3,3) \\
 f(x, y) * k(x, y) &= (0 \times 10) + (-1 \times 8) + (0 \times 4) + (-1 \times 8) + (5 \times 4) + (-1 \times 8) + (0 \times 8) + (-1 \times 3) + (0 \times 6) \\
 &= 0 + (-8) + 0 + (-8) + 20 + (-8) + 0 + (-3) + 0 \\
 &= |-7| = 7
 \end{aligned}$$

$$\begin{aligned}
 6. \text{ Nilai intensitas warna citra awal } f(3,2) - f(5,4) \text{ di kali dengan kernel } k(1,1) - k(3,3) \\
 f(x, y) * k(x, y) &= (0 \times 8) + (-1 \times 4) + (0 \times 4) \\
 &\quad + (-1 \times 4) + (5 \times 8) + (-1 \times 11) + (0 \times 3) + (-1 \times 6) + (0 \times 6) \\
 &= 0 + (-4) + 0 + (-4) + 40 + (-11) + 0 + (-6) + 0 \\
 &= 15
 \end{aligned}$$

Proses ini sampai posisi (0, 3) dari kernel berada pada posisi akhir (bawah) citra,.

b. Proses Pendeteksian Tepi Dengan Algoritma Sobel Edge Detection

Proses selanjutnya pendeteksian tepi menggunakan *Sobel Edge Detection*, dimana operator sobel yang digunakan yaitu:

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Maka proses pendeteksian dengan operator sobel adalah sebagai berikut:

1. a. Nilai intensitas warna citra awal $f(1,1) - f(3,3)$ di kali dengan operator $G_x(1,1) - G_x(3,3)$:

$$S_x = | (-1*8) + (-2*5) + (-1*2) + (1*9) + (2*9) + (1*7) | = 14$$

- b. Nilai intensitas warna citra awal $f(1,1) - f(3,3)$ di kali dengan operator $G_y(1,1) - G_y(3,3)$:

$$S_y = | (1*2) + (2*20) + (1*7) + (-1*8) + (-2*5) + (-1*9) | = 22$$

$$|S| = \sqrt{14^2 + 22^2} = 26$$

2. Nilai intensitas warna citra awal $f(2,1) - f(4,3)$ di kali dengan operator $G_x(1,1) - G_x(3,3)$:

$$S_x = | (-1*5) + (-2*2) + (-1*7) + (1*9) + (2*7) + (1*15) | = 22$$

- b. Nilai intensitas warna citra awal $f(2,1) - f(4,3)$ di kali dengan operator $G_x(1,1) - G_x(3,3)$:

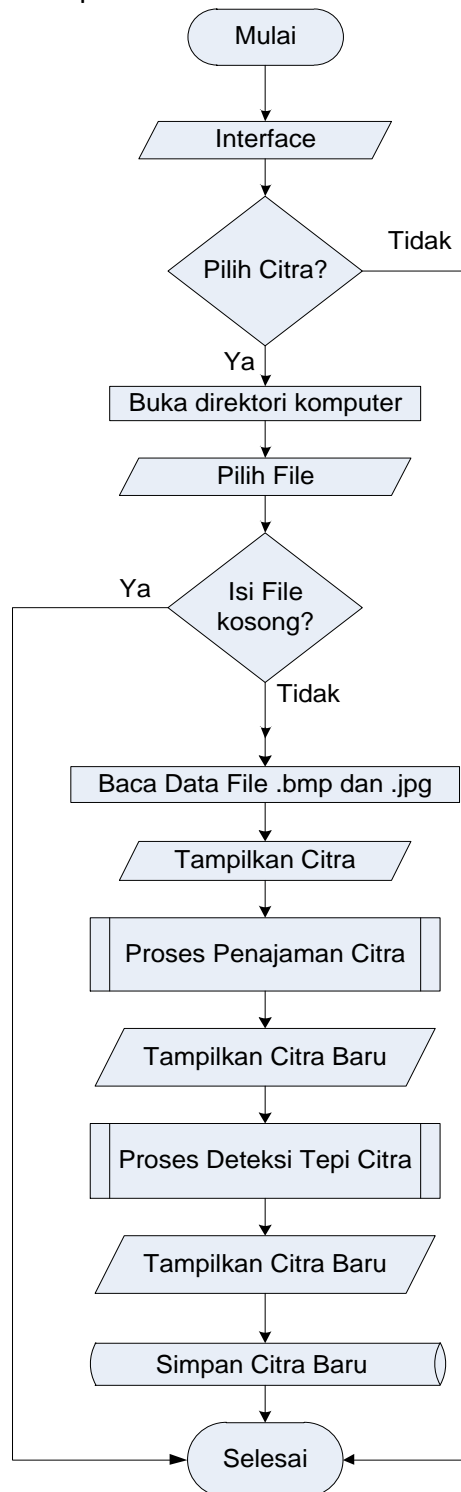
$$S_y = | (1*7) + (2*7) + (1*15) + (-1*5) + (-2*24) + (-1*9) | = |-26| = 26$$

$$|S| = \sqrt{22^2 + 26^2} = 34$$

Dan proses terus dilakukan hingga akhir citra sama seperti proses yang dilakukan sebelumnya. Sehingga Hasil akhir nilai intensitas warna akhir citra bitmap sebagai hasil proses pendeteksian tepi dengan penajaman citra (*sharpening*) dan menggunakan algoritma *sobel edge detection* nilai intensitas warna hasil proses pendeteksian memiliki nilai intensitas warna yg tinggi daripada nilai intensitas awal citra.

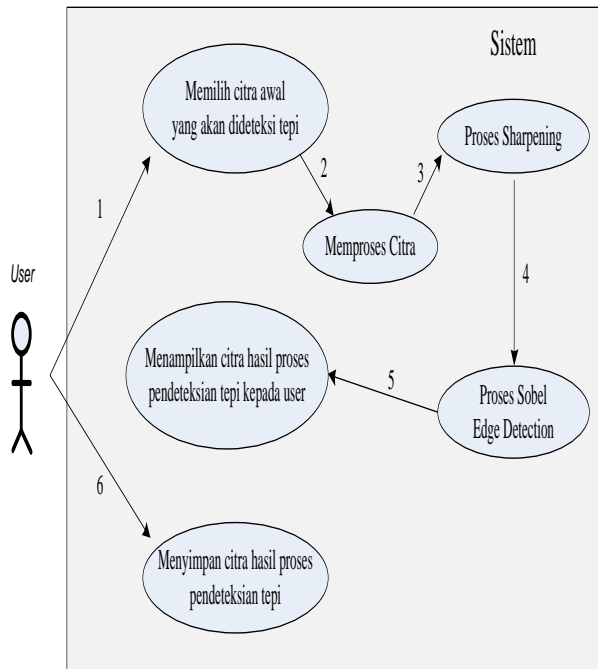
3. Flowchart Program

Flowchart program adalah prosedur sesungguhnya yang dilakukan oleh suatu program dan diterangkan lebih rinci. Flowchart ini menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Berikut ini adalah merupakan gambar flowchart proses pendeteksian tepi terhadap citra bitmap.



4. Use Case Diagram Proses Penajaman dan Pendeteksian Tepi Citra

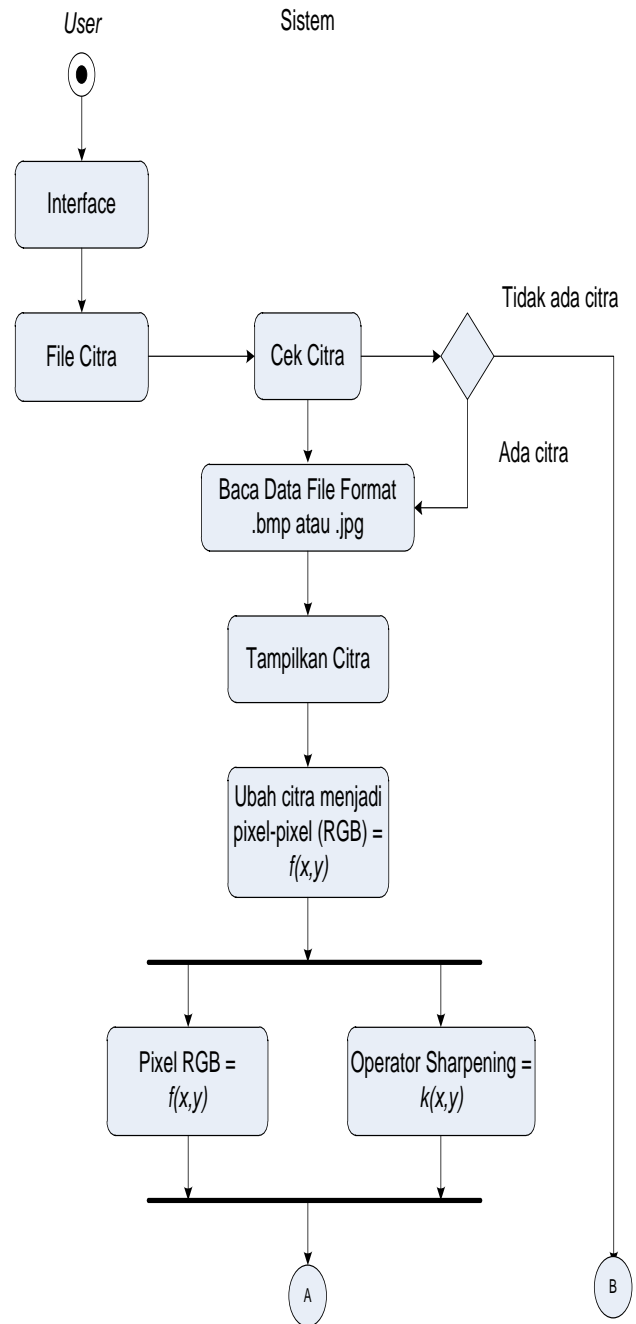
Diagram use case digunakan untuk memberikan gambaran kebutuhan perangkat lunak secara visual, tahapan kinerja dari sistem dalam berinteraksi terhadap pengguna. Berikut ini merupakan use case diagram yang akan dirancang.



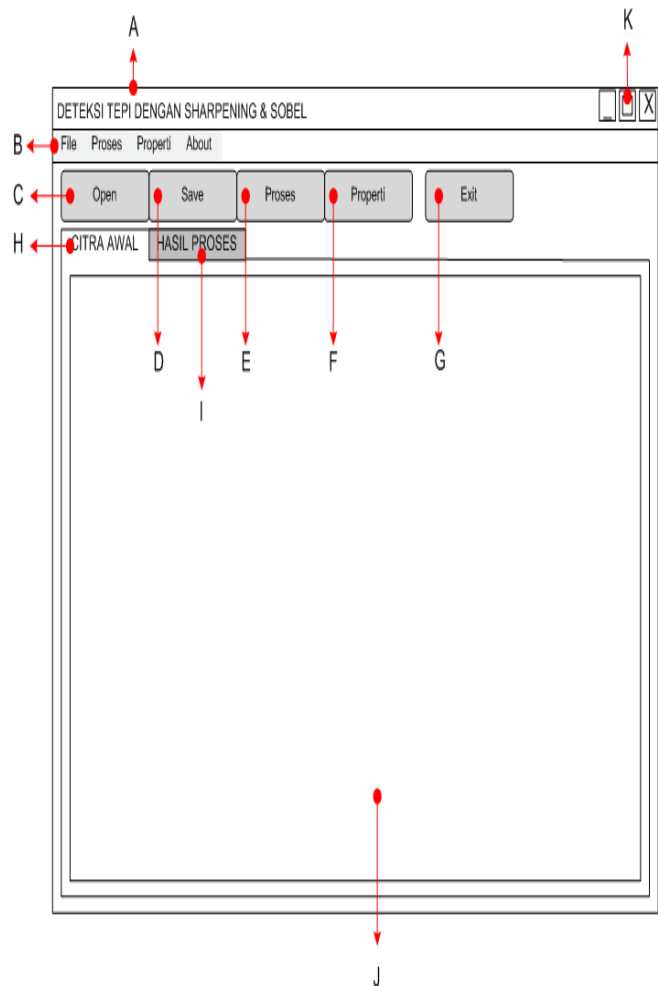
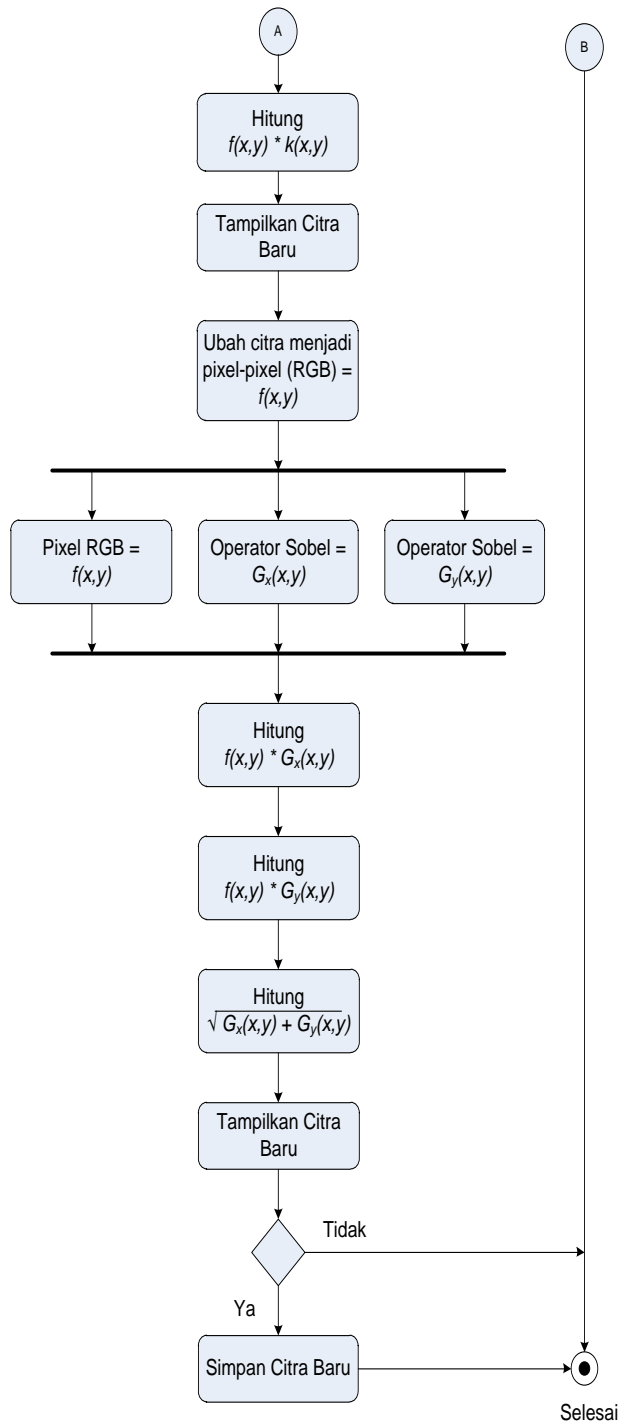
Gambar 2. Use Case Proses Penajaman dan Pendeteksian Tepi Citra

User merupakan pengguna yang akan memberikan perintah terhadap sistem pendeteksian tepi. Langkah awal user memberikan perintah terhadap sistem dalam memilih citra awal yang akan dideteksi tepi. Kemudian use case berikutnya merupakan langkah memproses citra yang telah dipilih yang dilakukan oleh sistem. Proses yang dilakukan terdiri dari dua tahapan yaitu diawali dengan proses *Sharpening* dan proses *Sobel Edge Detection*. Hasil dari proses akan ditampilkan oleh sistem kepada user. Citra hasil dari proses yang ditampilkan oleh sistem dapat disimpan oleh user.

5. Activity Diagram Proses Penajaman dan Pendeteksian Tepi Citra



dalamnya terdapat *form* lainnya yang memberikan akses untuk masuk ke dalam *form* lainnya. Berikut ini merupakan gambar



dari *form* menu utama yang akan dirancang:

Gambar 3. Struktur Rancangan *Form* Menu Utama

Berikut ini merupakan penjelasan dari rancangan *Form* Menu Utama yang dapat dilihat pada tabel berikut ini:

Penjelasan Rancangan *Form* Menu Utama

- TitleBar: Berisikan judul program pada *window*
- MenuBar: Bagian ini menampilkan Menu yang terdapat pada antarmuka yang terdiri dari menu *File*, *Proses*, *Properti*, dan *About*. Pada menu *File* terdapat

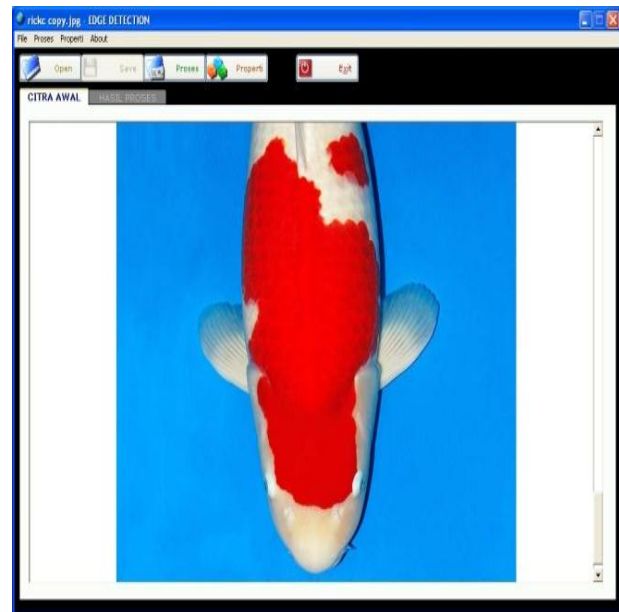
6. Rancangan Menu Utama

Pada suatu aplikasi *form* menu utama dikenal sebagai sebuah *window* utama yang disebut juga sebagai *window* induk yang di

pilihan untuk mengambil file gambar yang akan diproses, menyimpan hasil proses dan pilihan untuk keluar dari program. Menu Proses digunakan untuk memulai proses pendeteksian tepi. Menu Properti untuk melihat resolusi dan ukuran gambar yang dibuka. Menu About untuk melihat sekilas keterangan dari sistem antarmuka pada *form about*.

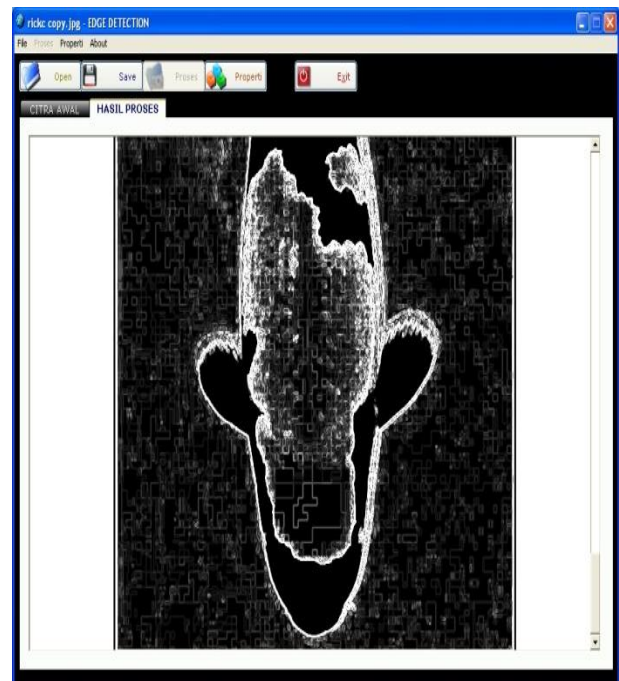
- c. Command Button (*Open*): Berfungsi sebagai tombol untuk mengambil file gambar yang akan diproses.
- d. Command Button (*Save*) Berfungsi sebagai tombol untuk menyimpan gambar hasil proses pendeteksian tepi.
- e. Command Button (*Proses*) Berfungsi sebagai tombol untuk memulai proses pendeteksian tepi.
- f. Command Button (*Properti*) Berfungsi sebagai tombol untuk melihat properti gambar yang dipilih seperti resolusi dan ukuran gambar.
- g. Command Button (*Exit*) Berfungsi sebagai tombol untuk keluar/menutup program.
- h. TabImage (*Citra Awal*): Berfungsi untuk menampung lokasi gambar dari citra awal yang akan ditampilkan pada PictureBox.
- i. TabImage (*Hasil Proses*): Berfungsi untuk menampung lokasi gambar dari hasil proses pendeteksian tepiyang akan ditampilkan pada j. PictureBox:
Untuk menampilkan citra yang telah dipilih pada tab image (*Citra Awal*) dan yang telah diproses pada tab image (*Hasil Proses*).
- k. CommandButton: Tombol untuk meminimize form utama,
- l. Tombol untuk keluar: menutup program.

7. Tampilan Utama Citra Bitmap



Gambar 4. Contoh Citra Bitmap Sebelum Diproses

8. Tampilan Hasil Proses Identifikasi Citra Bitmap



Gambar 5. Contoh Hasil Proses Indentifikasi Citra Bitmap

SIMPULAN

Dari hasil analisis dan implementasi yang diperoleh, maka dapat diambil simpulan sebagai berikut:

1. Proses identifikasi edge dilakukan dengan proses penajaman citra (*sharpenning*) kemudian diproses dengan Algoritma Sobel Edge Detection. Sistem pendeteksian yang dilakukan baik berupa gambar objek benda, buah, dan juga wajah manusia.
2. Saat pengujian sistem hanya terdapat dua jenis format citra yang dapat diproses pada pendeteksi tepi yaitu format citra bmp dan jpeg.

DAFTAR PUSTAKA

Ario Suryokusumo, 2001, *Microsoft Visual Basic 6.0*, PT. Elex Media Komputindo.

Darma Putra, 2009, *Pengolahan Citra Digital*, Jakarta.

<http://te.unib.ac.id/lecturer/indraagustian/2013/06/defnisi-citra/>

Djoko Pramono, 2002, *Mudah menguasai*

Visual Basic 6, PT. Elex Media Komputindo.

Vincent Suhartono, 2002, *Teori Pengolahan Citra Digital*, PT. Elex Media Komputindo.

http://t3.gstatic.com/upload/files/4596_Pertemuan_8.pdf (diakses tanggal 10 Mei 2013).

<http://202.91.15.14/upload/files/>

<http://Wikipedia.org>, Wikipedia (diakses tanggal 18 Mei 2013).

<http://images.simplyos.multiply.com/attachment/0/SPs5-AoKCmwAAC0hQzY1/PCD%20-%20Topik4%2020Image%20Restoration01.pdf?nmid=005128> (diakses tanggal 10 Mei 2013).

<http://www.cs.su.ac.th//hornil.com/517483/ppt/Chapter%205.ppt> diakses tanggal 20 Mei 2013).

T. Sutoyo, Edy Mulyanto. TT. *Teori Pengolahan Citra Digital*. Yogyakarta: ANDI.