T - 9

Pengubahan Data Image Ikan Air Tawar ke Data Vektor menggunakan *Edge Detection* Metode *Canny*

Fety Fatimah [1] , Gibtha Fitri Laxmi [2] , Puspa Eosina [3]
Universitas Ibn Khaldun, Bogor
fety.fatimah@uika-bogor.ac.id,

Abstrak—Keberadaan jenis ikan air tawar yang tersebar di seluruh Indonesia serta semakin bertambahnya spesies ikan menyebabkan proses identifikasi semakin sulit dilakukan. Proses identifikasi selama ini dilakukan secara manual dengan melihat ciri-ciri ikan pada buku atau memerlukan praduga manusia yang bisa membedakannya. Hal yang dibutuhkan untuk mempercepat proses identifikasi adalah menggunakan teknologi. Dengan melakukan pengubahan data image ikan air tawar ke dalam data vektor dengan menggunakan edge detection metode *Canny*. Data yang digunakan berupa 10 jenis ikan air tawar. Prosesnya dimulai dengan menginput citra digital ikan, selanjutnya dikonversi ke citra grayscale. Data citra grayscale tersebut kemudian dilakukan proses scalling dengan ukuran 200x300 piksel. Lalu cek antarwarna di proses *Thresholding* untuk mengubah nilai k*pixel* menjadi 2 macam yaitu nilai 0 dan 1. Jika berwarna hitam atau kurang dari nilai *Threshold* maka citra di kodekan 0, sebaliknya jika berwarna putih dan lebih dari nilai *Threshold* maka di kodekan 1. Selanjutnya data citra diekstraksi dengan edge detection untuk memperoleh data vektor ciri. Tujuan dari penelitian ini untuk menghasilkan data hasil identifikasi citra ikan air tawar dalam bentuk data vektor. Metode yang digunakan untuk mengidentifikasi citra ikan adalah metode Canny pada edge detection. Untuk selanjutnya data vektor ini dapat digunakan untuk model klasifikasi untuk memudahkan pengenalan jenis ikan air tawar berdasarkan bentuknya.

Kata kunci : Ikan air tawar, Edge Detection, Canny

I. PENDAHULUAN

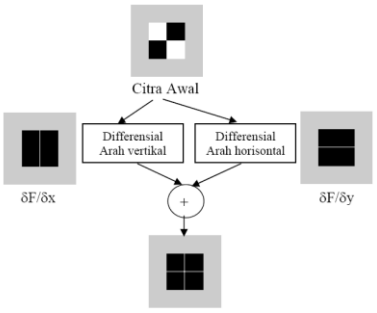
Indonesia sebagai negara tropis memiliki keanekaragaman hayati yang sangat tinggi. Hal tersebut dapat dilihat dari beragamnya jenis flora maupun fauna di Indonesia dan masuknya Indonesia ke dalam sepuluh negara yang kekayaan keanekaragaman hayatinya tertinggi atau dikenal dengan *megadiversity country*. Sekitar 16% dari spesies ikan dunia hidup di Indonesia. Dilaporkan di perairan Indonesia terdapat lebih dari 7000 spesies ikan, dimana 2000 spesies diantaranya adalah ikan air tawar. Dari 7000 spesies ikan tersebut, baru tercatat 40 spesies yang sudah berhasil dibudidayakan yaitu : 27 spesies ikan air tawar, 10 spesies ikan laut dan 3 spesies ikan air payau atau *diadromus* [1]. Keberadaan jenis ikan air tawar yang tersebar di seluruh Indonesia serta semakin bertambahnya spesies ikan menyebabkan proses identifikasi semakin sulit dilakukan. Proses identifikasi selama ini dilakukan secara manual dengan melihat ciri-ciri ikan pada buku atau memerlukan praduga manusia yang bisa membedakannya. Proses identifikasi dengan cara manual dan praduga membutuhkan waktu yang cukup lama serta memungkinkan terjadinya kesalahan. Salah satu upaya yang dibutuhkan untuk membantu proses identifikasi adalah menggunakan teknologi. *Content Based Image Retrieval* (CBIR) merupakan suatu teknik temu kembali citra yang mempunyai kemiripan karakteristik atau *content* serta informasi yang terkandung di dalam citra seperti warna, bentuk, dan tekstur. Ciri bentuk merupakan tahapan pertama yang dilakukan untuk menemukan informasi objek yang ada di dalam citra. Salah satu operasi yang dapat digunakan dalam pengolahan citra untuk ciri bentuk adalah deteksi tepi (*Edge Detection*). Tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang cepat/tiba-tiba (besar) dalam jarak yang singkat. Sedangkan deteksi tepi (E*dge 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 [2].

Berdasarkan latar belakang tersebut, maka perlu dilakukan pengubahan data image ke data vektor berdasarkan bentuk menggunakan *Edge Detection* metode *Canny*. Batasan masalah dalam penelitian ini adalah sebagai berikut: (1) hanya mengidentifikasi jenis ikan berdasarkan bentuk, (2) hanya meneliti 10 jenis ikan, (3) gambar yang di teliti dengan format jpg dan berukuran 300 x 200 pixel, (4) dan *background* berwarna putih. Tujuan yang akan dicapai dalam penelitian ini adalah : (1) Merubah data image ikan air tawar menjadi data vektor. (2) Menganalisa data vektor menggunakan *edge detection* metode *canny* dengan cara mengukur kedekatan jarak data uji dan data latih. Adapun manfaat dari penelitian ini adalah : (1) Manfaat praktis dari penelitian ini yaitu implementasi teknik deteksi tepi untuk mengetahui bentuk dari ikan serta mengetahui kinerjanya sehingga bermanfaat bagi masyarakat untuk mengetahui jenis ikan yang mereka ingin ketahui berdasarkan bentuknya. (2) Manfaat teoritis dari penelitian ini adalah mengetahui *edge detection* yang sesuai untuk mengidentifikasi ikan sehingga bentuk ikan semakin jelas terlihat dan akan lebih mudah membedakannya.

II. METODE PENELITIAN

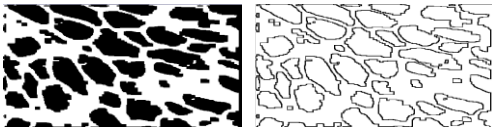
A. Edge Detection

Deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari objek-objek gambar. Suatu titik (*x,y*) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangga. Pada Gambar 1 dapat dilihat proses yang dilakukan untuk memperoleh tepi gambar dari suatu citra yang ada.



Gambar 1 Proses Deteksi tepi

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 piksel yang tinggi.



Gambar 2 Hasil Proses Deteksi Tepi *Canny*

Tujuan mendeteksi tepi untuk mengelompokkan objek-objek dalam citra dan digunakan untuk menganalisisi citra lebih lanjut. Deteksi tepi *canny* dikembangkan oleh John F. Canny tahun 1986 dengan menggunakan algoritma multi tahap untuk mendeteksi berbagai tepi gambar.

Kategori algoritma *Canny* yang dikembangkan oleh John F. Canny, antara lain [3] :

a. Mendeteksi dengan baik (Kriteria deteksi)
   Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan.
b. Melokalisasi dengan baik (kriteria lokalisasi)
   Menghasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli
c. Respon yang jelas (kriteria respon)
   Hanya ada satu respon untuk tiap tepi sehingga mudah dideteksi dan tidak menimbulkan kerancuan pada saat pengolahan citra

Canny dikenal juga sebagai operator deteksi tepi yang optimal. Seperti dijelaskan pada kategori Algortima Canny, algoritma ini memberikan tingkat kesalahan rendah, melokasikan titik tepi yang ditemukan dengan yang sesungguhnya sangat pendek, dan memberikan satu tanggapan untuk satu tepi[4].

Algoritma *Canny* berjalan dalam 5 langkah yang terpisah, yaitu [5] :

1) *Smoothing* : mengaburkan gambar untuk menghilangkan *noise*
2) *Finding gradien* : tepian harus ditandai pada gambar memiliki gradien yang besar
3) *Non-maksimum-suppresion* : hanya maxima lokal yang harus ditandai sebagai *edge*

4) *Double thresholding* : tepian yang berpotensi ditentukan oleh *thresholding*
5) *Edge Tracking by hysteresis* : tepian final ditentukan dengan menekan semua sisi yang tidak terhubung dengan tepian yang sangat kuat

B. Recall dan Precision

Dua parameter utama untuk mengevaluasi efektifitas temu kembali yang telah digunakan sejak lama adalah *recall* dan *precision*. *Recall* adalah perbandingan jumlah materi relevan yang ditemukembalikan terhadap jumlah materi yang relevan. Sementara itu *precision* adalah perbandingan jumlah materi relevan yang ditemukembalikan terhadap jumlah materi yang ditemukembalikan [6].

$$recall = \frac{\text{jumlah citra relevan yang terambil}}{\text{jumlah citra relevan dlm basis data}}$$

$$precision = \frac{\text{jumlah citra relevan yang terambil}}{\text{jumlah seluruh citra yang terambil}}$$

III. HASIL DAN PEMBAHASAN

A. Data Citra Ikan

Akuisisi citra ikan air tawar dilakukan dengan pemotretan sepuluh jenis ikan air tawar yang didapat dengan pembelian ikan. Pemotretan dilakukan menggunakan kamera digital Canon 60D + Lensa L series 16-35 f/2.8 mark ii. Total citra ikan yang digunakan adalah 300 yang terdiri atas 10 jenis ikan, depan dan belakang (masing-masing kelas 30 citra). Citra ikan air tawar berformat JPG dan berukuran 300 × 200 piksel.

B. Ekstraksi Bentuk dengan Edge Detection

Ekstraksi bentuk pada citra ikan air tawar hanya dilakukan pada piksel yang menyusun citra tersebut. Citra dikonversi ke *mode* warna *grayscale*. Selanjutnya membentuk matrik ciri citra dari titik-titik tepi citra menggunakan metode *Canny.*

C. Tahapan Algoritma Canny

1) Smoothing

Citra yang akan diproses seringkali mengandung derau. Untuk itu perlu dibersihkan terlebih dahulu dengan menggunakan filter. Metode yang digunakan melalui filter Filter Gausian tergolong sebagai filter lolos rendah (hasil filter hanya terjadi sedikit penghalusan). Untuk mencegah *noise* dideteksi sebagai tepian, maka kita harus memperhalus gambar menggunakan gaussian filter. Inti dari *Gaussian filter* adalah standar deviasi dengan $\sigma = 1.4$ [7]

2) Finding Gradien

Menemukan titik tepi pada gambar grascale dengan nilai intensitas yang paling besar. Gradien gambar yang telah diperhalus. Dengan memperkirakan gradien arah x dan y. Seperti pada persamaan dengan menggunakan kernel Sobel.

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

Contoh citra :

65	50	55
78	68	60
60	60	62

f (2,2) = 68

maka nilai piksel hasil konvolusi berupa :

Horisontal = $K_1(x, y)$ = | (-1*62) + (0*60) + (-1*60) + (-2*60) + (0*68) + (2*78) + (-1*55) + (0*50) + (1*65) | = | 40 | = 40

Vertikal = $K_2(x, y)$ = | (1*62) + (2*60) + (1*60) + (0*60) + (0*68) + (0*78)

$$+ (-1*55) + (-2*50) + (-1*65) | = |22| = 22$$

Mengkonversi tepian pada gambar hasil magnitude gradien sehingga menghasilkan tepian yang tajam (Arah Tepian). Dilakukan dengan mempertahankan semua maxima lokal dalam gambar gradien dan menghapus segala sesuatu yang lain.

$$K_0(x,y) = (| K_1(x,y)| + | K_2(x,y) |) = | 40 | + | 22 | = | 62 | = 62$$

Gambar citra dengan gradien yang besar sering menunjukkan tepian yang cukup jelas. Namun, tepian biasanya luas sehingga tidak dapat menunjukkan letak persis tepiannya [7]. Untuk menentukan tepian yang sebenarnya, arah tepian harus ditentukan dan disimpan dengan menggunakan persamaan berikut:

$$\theta = \arctan\left(\left|\frac{G_x}{G_y}\right|\right)$$

3) *Non-maksimum-suppresion*

Setelah arah tepi diperoleh, perlu menghubungkan antara arah tepi dengan sebuah arah yang dapat dilacak dari citra. Tujuan dari langkah ini adalah untuk mengubah tepi "kabur" pada citra magnitudo gradien ke tepi "tajam". Pada dasarnya hal ini dilakukan dengan melestarikan semua maxima lokal pada citra gradien, dan menghapus segala sesuatu yang lain. Semua arah tepi yang berkisar antara 0 dan 22,5 atau 157,5 dan 180 derajat dirubah menjadi 0 derajat. Kisaran antara 22,5 dan 67,5 menjadi 45 derajat. Kisaran antara 67,5 dan 112,5 diubah menjadi 90 derajat. Dan kisaran antara 112,5 dan 157,5 menjadi 135 derajat [8]. Algoritma ini untuk setiap pixel dalam gradien image:

1. Putar arah gradien θ ke arah 45° terdekat, kemudian hubungkan dengan ketetanggan piksel (8 titik tetangga yang terhubung dengannya).
2. Bandingkan nilai piksel tepian saat ini dengan nilai piksel tepian dalam arah positif dan negatif gradien. Jika arah gradien adalah utara (θ =90 ◦), bandingkan dengan piksel ke utara dan selatan.
3. Jika nilai piksel tepian saat ini adalah yang terbesar, maka simpan nilai tepian tersebut, namun jika bukan, hapus nilai tersebut.

Setelah arah tepi diperoleh, penghilangan non maksimum dilaksanakan. Penghilangan non maksimum dilakukan sepanjang tepi pada arah tepi dan menghilangkan piksel (piksel diatur menjadi 0) yang tidak dianggap sebagai tepi [9].

4) *Double thresholding*

Metode Canny menggunakan sistem *thresholding* ganda dimana tepian dengan nilai yang lebih besar dari *threshold* atas ditandai sebagai titik kuat, tepian dengan nilai yang lebih kecil dari *threshold* bawah akan dihapus, dan tepian dengan nilai piksel antara *threshold* atas dengan *threshold* bawah akan ditandai sebagai tepian yang lemah. Tepian yang kuatlah yang akan menghasilkan tepian yang asli pada gambar. Tepian yang lemah dapat terjadi karena memang merupakan tepian yang nyata atau *noise* / variasi warna. [10]

5) *Edge Tracking by hysteresis*

Proses ini menghilangkan garis-garis yang terputus-putus pada tepi objek. Caranya dengan menggunakan dua ambang T_1 dan T_2. Lalu semua piksel citra yang bernilai lebih besar daripada T_1, dianggap sebagai piksel tepi. Dan nilai yang kurang dari T1 akan dihilangkan. Sementara nilai yang berada diantaranya akan melihat kecenderungan pada hubungan ketetanggaan. [11].

D. Proses pelaksanaan penelitian

1) Akuisisi Citra

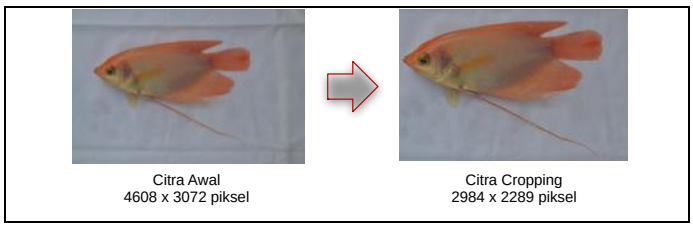
Akuisisi citra merupakan proses pengambilan citra dengan menggunakan sebuah alat bantu pengambil gambar. Kamera digital merek Canon dengan resolusi 16 *megapixel* digunakan untuk pengambilan citra. Citra yang diperoleh berupa citra warna RGB dengan ukuran 4608 x 3072 pixel seperti contoh pada Gambar 3

2) Pembagian Citra

Jumlah citra yang digunakan dalam penelitian ini adalah 300 citra ikan air tawar. Selanjutnya citra ikan ikan air tawar tersebut dibagi menjadi dua bagian yaitu 240 citra ikan air tawar digunakan sebagai citra latih dan 60 citra ikan air tawar sebagai citra kueri yaitu citra uji yang tidak melalui pelatihan.

3) Praproses Citra

Sebelum masuk ke dalam tahap ekstraksi fitur citra dilakukan praproses terlebih dahulu dimana citra ikan merupakan citra RGB dengan latar belakang putih. Selanjutnya akan dilakukan proses *cropping* dan *scalling*. *Cropping* merupakan proses memotong citra dan mengambil bagian dari citra yang dibutuhkan. Citra ikan air tawar yang diambil merupakan citra umum, artinya tidak semua bagian dibutuhkan dalam sistem yang akan dirancang, sehingga perlu proses *cropping*. Proses ini menggunakan *software* Photoshop CS6. Berikut Gambar 3 merupakan hasil *Cropping*.



Gambar 3 Citra Hasil *Cropping*

Proses *cropping* citra perlu dilakukan untuk menyeragamkan skala dimensi citra serta menghilangkan bagian yang tidak perlu. Setelah proses *cropping* citra hasil *cropping* dilakukan proses *scaling* untuk menyeragamkan ukuran citra. Hal ini dimaksudkan agar ukuran citra yang akan diolah dalam sistem mempunyai ukuran yang sama. Proses komputasi dimensi matriks yang besar membutuhkan waktu proses yang lama sehingga diperlukan reduksi dimensi citra untuk meminimalisir waktu proses. Ukuran citra dalam penelitian ini diubah menjadi 300 x 200 *pixel*, dengan proses *scalling* diformulasikan sebagai berikut :

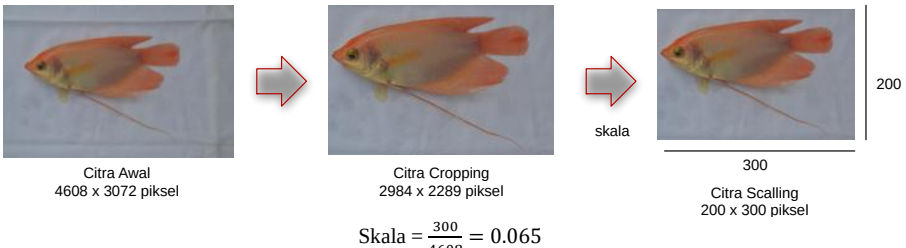
Skala = $\frac{max}{limit}$

$New_width = width * skala$

$New_height = height * skala$

Nilai *width* dan *height* merupakan panjang dan tinggi dari citra, sedangkan *limit* merupakan panjang maksimal dari sebuah citra dimana jika *width* lebih besar dari *height* maka yang diambil nilai *width*, begitu juga sebaliknya jika *height* yang lebih besar, maka yang diambil adalah nilai *height*. Nilai max adalah batas maksimal ukuran citra yang akan di *scalling*, dimana max ditentukan secara manual.

Proses *scalling* dilakukan untuk mengubah resolusi atau ukuran horizontal dan vertikal suatu citra dimana ukuran maksimal untuk horizontal dan vertikal secara manual ditentukan sebesar 300 pixel. Ilustrasi *scalling* dapat dilihat pada Gambar 4



Skala = $\frac{300}{4608} = 0.065$

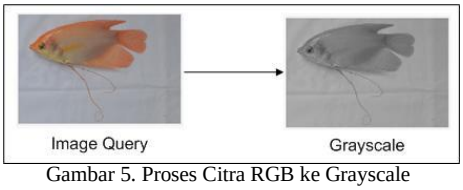
$new\ height = 4608 \times 0.065 = 300$

$new\ width = 3072 \times 0.065 = 200$

Gambar 4 Proses *Scalling.*

4) Mengubah Citra RGB Menjadi *Grayscale*

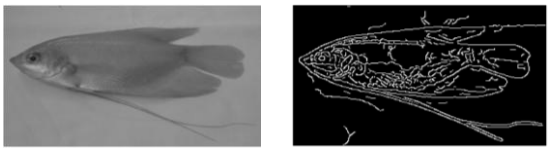
Citra yang diinputkan dalam sitem awalnya adalah citra tipe RGB. Citra RGB akan membentuk vektor 3 lapis sehingga akan sulit untuk dilakukan proses selanjutnya. Citra perlu diubah tipenya menjadi *grayscale* untuk mempermudah proses pengolahan citra pada proses selanjutnya. Citra *grayscale* dapat dilihat pada Gambar 5.



Gambar 5. Proses Citra RGB ke Grayscale

5) Deteksi Tepi *Canny*

Proses deteksi tepi ini adalah salah satu cara untuk mengenali bentuk yang dimiliki oleh setiap ikan. Oleh karena itu, setelah melalui proses *grayscale* langkah selanjutnya adalah deteksi tepi *(Edge Detection)*. Proses deteksi tepi ini diolah dengan deteksi tepi *Canny* sehingga tepi citra ikan akan tampak lebih jelas. Contoh hasil deteksi tepi *Canny* dapat dilihat pada Gambar 6.

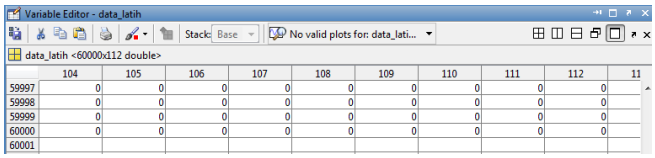


Gambar 6. Hasil deteksi tepi *Canny*

Pada Gambar 7 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. Citra hasil *Canny* selanjutnya akan di proses *Thresholding* untuk mengubah nilai *pixel* menjadi 2 macam yaitu nilai 0 dan 1. Seringkali dikenal juga dengan proses binerisasi. Jika berwarna hitam atau kurang dari nilai *Threshold* maka citra di kodekan 0, sebaliknya jika berwarna putih dan lebih dari nilai *Threshold* maka di kodekan 1. cara yang jelas untuk mengekstrak obyek dari *background* adalah dengan memilih *Threshold T* yang membagi mode-mode ini. Sembarang titik (x,y) untuk dimana $f\ x,y \geq T$ disebut *object point*, sedangkan yang lain disebut *background point*. Citra yang di-*threshold* $g(x,y)$ didefinisikan sebagai:

$$g(x,y) = \begin{cases} 1 jika\ f(x,y) & \geq T \\ 0\ jika\ f(x,y) & < T \end{cases}$$

Pixel yang diberi nilai 1 berkaitan dengan obyek sedangkan *pixel* yang diberi nilai 0 berkaitan dengan *background.* Hasil dari deteksi tepi dan *Threshold* ini akan di cocokan dengan citra latih yang ada dalam sistem. Vektor citra latih dapat dilihat pada Gambar 7.



	104	105	106	107	108	109	110	111	112	11
59997	0	0	0	0	0	0	0	0	0	
59998	0	0	0	0	0	0	0	0	0	
59999	0	0	0	0	0	0	0	0	0	
60000	0	0	0	0	0	0	0	0	0	
60001										

Gambar 7. Vektor Citra Latih

Data Citra yang dilatih berbentuk vektor dengan ukuran 60000x112, vektor ini diperoleh dari proses *reshape* yaitu merubah matriks data citra kedalam bentuk vektor. Karena jumlah citra latih sebanyak 112 maka matriks yang terbentuk dalam data latih sistem adalah matriks berukuran 60000x112 yaitu gabungan dari vektor-vektor citra latih sistem yang disimpan dalam satu database yang dinamakan data_latih.mat.

IV. SIMPULAN

Pengubahan data citra digital ke bentuk vektor untuk mengenali 10 jenis ikan air tawar dengan menggunakan *edge dection* metode *canny*. Metode *Canny* merupakan hasil pengembangan *edge detection* yang telah ada sebelumnya. Metode *edge detection canny* menghasilkan gambar tepian yang baik dan mengurangi kemungkinan kesalahan mendeteksi. Deteksi tepi *Canny* didapatkan dari hasil pemisahan bayangan terang dan gelap. Kelebihan dari metode *Canny* ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi sehingga tepi-tepi yang dihasilkan lebih banyak. Kemanfaatan penelitian ini mempermudah proses identifikasi. Manfaat penelitian ini untuk selanjutnya digunakan dalam pengklasifikasian.

Daftar Pustaka

[1] Santoso L. "Biologi reproduksi ikan belida (chitala lopis) di sungai tulang bawang, lampung", Berkala Perikanan Terubuk, Vol. 37. No.1. 2009

[2] Yunus M. "Perbandingan metode-metode edge detection untuk proses segmentasi citra digital", Program Studi Teknik Informatika, STMIK PPKIA Pradnya Paramita Malang, Jurnal Teknologi Informasi Vol. 3 No. 2. 2008.

[3] Canny, J. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and MachineIntelligence, 8, 679-698. 1986

[4] Biman Debbarma and Dibyendu Ghoshal, A modified canny edge detection algorithm with variable sigma, International journal of engineering trends in electrical and electronics (IGETEE), Vol. 1,Issue 2, March 2013

[5] C Danil. Edge Detection dengan Algoritma Canny. STMIK IBBI. 2001.

[6] Ginting, Elias Dianta. *Deteksi tepi menggunakan metode canny dengan matlab untuk membedakan uang asli dan uang palsu.* Skripsi Program Studi Teknik Informatika. 2014.

[7] Oskoei, Mohammadreza Asghari, and Huosheng Hu. A survey on edge detection methods. University of Essex, UK. 2010.

[8] Kadir Abdul. Adhi Susanto. Teori dan Aplikasi Pengolahan Citra. Penerbit Andi Yogyakarta. 2013

[9] He, Wenhao, and Kui Yuan. An improved Canny edge detector and its realization on FPGA. Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on. IEEE, 2008.

[10] Mandloi, Gaurav. A Survey on Feature Extraction Techniques for Color Images. International Journal of Computer Science and Information Technologies 5.3. 2014

[11] Dhankhar, Poonam, and Neha Sahu. A review and research of edge detection techniques for image segmentation. International Journal of Computer Science and Mobile Computing (IJCSMC) 2.7 .2013