

Klasifikasi Objek Terdeformasi Berdasarkan Nilai Deviasi Menggunakan Metode Kontinuitas Kontur

Setiawan Hadi¹, Akmal¹, Susi Herlina²

¹⁾ Fakultas MIPA, Program Studi Teknik Informatika, Universitas Padjadjaran
Jl. Raya Bandung Sumedang KM 21 Jatinangor Sumedang 45363
Email: setiawanhadi@unpad.ac.id, akmal@unpad.ac.id

²⁾ Fakultas MIPA, Program Studi Matematika, Universitas Padjadjaran
Jl. Raya Bandung Sumedang KM 21 Jatinangor Sumedang 45363
Email: susiherlina@gmail.com

ABSTRAK

Pendeteksian berbagai bentuk objek merupakan topik ilmu visi komputer yang dapat dimanfaatkan sebagai dasar untuk aplikasi komputer cerdas. Proses, yang mudah dilakukan oleh manusia ini, dapat menjadi rumit apabila dilakukan oleh mesin (komputer) karena sifat diskrit dan alami dari objek dalam format citra digital. Pada literatur dapat dijumpai banyak algoritma yang mendiskusikan proses identifikasi berbagai bentuk objek, tetapi algoritma-algoritma tersebut kurang efisien karena membutuhkan waktu pemrosesan yang tidak sedikit dan data harus memiliki tingkat resolusi yang tinggi. Pada makalah ini akan ditunjukkan sebuah pendekatan matematis untuk pengenalan bentuk objek yang sangat efisien dan cepat, khususnya untuk pendeteksian objek grafis yang terdeformasi. Pengembangan metode didasarkan pada sifat kontinuitas kontur dimana akan diukur penyimpangan atau deviasi dengan menggunakan sifat-sifat unik dari objek primitif yang bersangkutan. Pengujian telah dilakukan menggunakan 60 buah objek dan hasilnya menunjukkan bahwa metode yang dikembangkan mampu mendeteksi objek-objek geometris seperti lingkaran, elips, DES (*Deformed Elliptic Shapes*), persegi, dan garis dengan tingkat keakurasian sebesar 85%.

Kata kunci: pengenalan objek, klasifikasi objek, metode kontinuitas kontur, pengolahan citra

ABSTRACT

*Detection of objects is a popular and interesting computer vision topic that can be used as a fundamental concept of a smart computer applications. The process, which is easily performed by humans, can be complicated if performed by machine (computer) due to the nature of the discrete and natural objects in digital image format. In the literature, there are many algorithms that can be found that discussing the process of identification of the object shape, but the algorithms are less efficient because it requires large amount of computer resources and processing time. It also need high level of resolution data. This paper demonstrated a mathematical approach for the elementary objects detection that are very efficient and fast, especially for the detection of a deformed graphic object. Development of a method is conducted based on the continuity properties of the contour where irregularities or deviations will be measured using the unique properties of relevant primitive. Tests have been carried out using 60 image objects and the result shows that the performance of the developed method is successfully detecting geometric objects such as circles, ellipses, DES (*Deformed Elliptic Shapes*), a square, and a line, with a 85% accuracy rate.*

Keywords: object recognition, object classification, contour continuity, image processing

1. Pendahuluan

Deteksi berbagai bentuk objek telah berkembang dan menarik perhatian para peneliti hingga saat ini (Parekh dkk [3]). Pendeteksian objek yang biasanya berfokus pada pendeteksian wajah berkembang kepada pendeteksian objek-objek lain dengan memanfaatkan konsep pengenalan pola. Deteksi berbagai objek ini merupakan bagian dari ilmu visi komputer/visi mesin yang juga semakin hari kemampuannya semakin mendekati kepada kemampuan manusia. Pendeteksian objek dimulai dari pendeteksian berbagai bentuk geometri primitif seperti garis, segitiga, segiempat, lingkaran, elips dan bentuk yang lainnya. Kemampuan ini dapat dimanfaatkan dalam berbagai bidang diantaranya dalam lingkungan industri dimana aplikasi pengenalan bentuk objek secara otomatis diperlukan dalam proses industri.

Proses pengenalan bentuk objek harus melalui tahap segmentasi dan evaluasi sehingga diperoleh hasil yang diinginkan dalam waktu yang singkat. Objek disegmentasi secara otomatis berdasarkan pengenalan *background* baik yang diketahui maupun tidak diketahui. Untuk *background* yang

diketahui, proses akan dilanjutkan dalam tahap evaluasi sedangkan jika background tidak diketahui maka dilakukan analisis bentuk objek tersebut (Gonzalez dkk, [1]).

Ada banyak aplikasi yang digunakan dalam menganalisis bentuk objek tetapi sedikitnya informasi mengenai struktur objek mengakibatkan aplikasi ini membutuhkan banyak waktu dan kerumitan dalam penyesuaian algoritma yang digunakan. Mengingat bahwa ketepatan dan kecepatan dari algoritma dalam analisis bentuk objek merupakan hal yang substansial, maka proses yang lambat dan rumit akan menjadi permasalahan, khususnya dalam aplikasi industri dimana, dibutuhkannya proses otomatisasi dan ketepatan dalam mendeteksi suatu objek secara *realtime*.

Pada makalah ini, pendeteksian objek akan difokuskan pada pengembangan metode menggunakan konsep matematika (Varberg, [5]) dalam hal kontinuitas kontur (*contour continuity*). Metode ini mampu mengidentifikasi berbagai objek secara otomatis dan langsung berdasarkan data atau informasi geometri yang dikandungnya. Proses selanjutnya adalah memeriksa nilai deviasi dan selanjutnya dilakukan evaluasi. Kontur diperoleh dengan estimator tepi menggunakan Metode Difference.

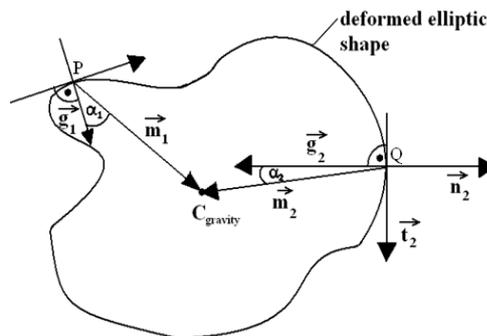
2. Metode Penelitian

2.1 Metode Kontinuitas Kontur

Kontur terdeformasi yang akan dideteksi merupakan kontur yang kontinu, tetapi citra digital berbentuk diskrit sehingga proses perhitungan berdasarkan pada setiap titik pada citra atau piksel. Semua sudut deviasi α_i antara vektor gradien \vec{g}_i dan semua vektor dari titik kontur ke pusat gravitasi \vec{m}_i dihitung dengan menggunakan persamaan sebagai berikut (Yilmaz, [6]):

$$D_\alpha = \frac{1}{N} \sum_{i=0}^{N-1} \cos^{-1}(\vec{g}_i \cdot \vec{m}_i)$$

Gambar 1 menunjukkan secara grafis bagaimana perhitungan D_α dari sebuah kontur terdeformasi berupa sebuah titik.



Gambar 1. Perhitungan D_α dari Sebuah Kontur Terdeformasi Sebuah Titik

Komputasi vektor \vec{m}_i dapat dirumuskan dengan persamaan sebagai berikut:

$$\vec{m}_i = \frac{\vec{m}_i}{|\vec{m}_i|} = \frac{(x_p - x)i + (y_p - y)j}{\sqrt{(x_p - x)^2 + (y_p - y)^2}}$$

Untuk setiap objek yang dideteksi, vektor gradien \vec{g}_i bergantung pada fungsi dari objek tersebut, yaitu:

1. Untuk lingkaran vektor gradien \vec{g}_i adalah sebagai berikut:

$$\vec{g}_i = \frac{\vec{g}_i}{|\vec{g}_i|} = \frac{(-2)(x - x_p)i + (-2)(y - y_p)j}{\sqrt{(-2(x - x_p))^2 + (-2(y - y_p))^2}}$$

2. Untuk elips dan DES vektor gradien \vec{g}_i adalah sebagai berikut:

$$\vec{g}_i = \frac{\vec{g}_i}{|\vec{g}_i|} = \frac{(-2)\frac{(x-x_p)}{a^2}i + (-2)\frac{(y-y_p)}{b^2}j}{\sqrt{(-2\frac{(x-x_p)}{a^2})^2 + (-2\frac{(y-y_p)}{b^2})^2}}$$

3. Vektor gradien \vec{g}_i untuk persegi dengan titik piksel berdekatan yang memiliki tangen yang hampir sama nilai vektornya tetap sehingga membentuk sudut sama dengan nol (0). Sedangkan untuk titik piksel yang membentuk sudut pada keempat sisinya vektor gradien \vec{g}_i adalah sebagai berikut:

$$\vec{g}_i = \frac{\vec{g}_i}{|\vec{g}_i|} = \frac{-(x - x_p)i - (y - y_p)j}{\sqrt{(-(x - x_p))^2 + (-(y - y_p))^2}}$$

4. Untuk garis vektor gradien \vec{g}_i adalah sebagai berikut:

$$\vec{g}_i = \frac{\vec{g}_i}{|\vec{g}_i|} = 0$$

Adapun penjelasan parameter-parameter yang digunakan adalah sebagai berikut :

- D_α = deviasi/simpangan yang terjadi dari nilai rata-rata semua α_i ,
- N = banyaknya titik pada kontur,
- \vec{g}_i = vektor gradien,
- \vec{m}_i = vektor-vektor dari titik kontur ke pusat gravitasi,
- (x,y) = koordinat titik piksel dari kontur,
- (x_p,y_p) = koordinat titik pusat,
- a = jarak dari titik pusat ke titik x terbesar,
- b = jarak dari titik pusat ke titik y terbesar.

2.1 Proses pengolahan citra

Proses pengolahan citra yang digunakan adalah sebagai berikut:

1. Grayscale

Citra warna dengan tiga ruang dimensi (RGB) dapat disederhanakan menjadi satu dimensi yaitu *grayscale*. *Grayscale* adalah citra keabuan yang memiliki nilai antara 0 – 255. Nilai tersebut menunjukkan tingkat derajat keabuan atau kecerahan dari citra (0 = hitam/gelap dan 255 = putih/terang). Dengan mengacu pada standard NTSC, konversi citra RGB ke *grayscale* dapat dilakukan dengan perhitungan mencari nilai dari pembobotan antara ketiga indeks dalam citra RGB, yaitu:

$$Grayscale = 0,299R + 0,587G + 0,114B$$

Dari hasil konversi citra ke dalam RGB maka nilai dalam indeks grayscale diasumsikan mewakili nilai dan informasi yang terkandung dalam citra RGB [1].

2. Smoothing

Image smoothing merupakan teknik untuk mereduksi *noise* yang terdapat pada citra. Dalam pengertian *noise* sebagai suatu nilai yang berbeda dengan semua tetangganya maka dapat dikatakan *noise* merupakan nilai-nilai yang berada pada frekuensi tinggi. Smoothing merupakan fungsi dari proses konvolusi antara citra $f(x, y)$ dengan filter $h(x, y)$:

$$g(x, y) = f(x, y) * h(x, y)$$

Pada penelitian ini digunakan convolution mask 3×3 menggunakan teknik filter rata-rata (*mean filter*). Filter rata-rata merupakan filter H dalam bentuk matriks yang berukuran $m \times n$, dan nilainya sama untuk setiap elemen, dan karena bersifat *Low Pass Filter* (LPF) maka jumlah seluruh elemen adalah satu, dan dituliskan dengan:

$$H(i, j) = \frac{1}{m \times n}, 1 \leq i \leq m, 1 \leq j \leq n$$

Filter rata-rata berukuran 3×3 adalah:

$$H = \begin{matrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{matrix}$$

3. Binerisasi

Semua citra RGB yang telah dikonversi ke grayscale kemudian disederhanakan kembali menjadi citra biner dengan *thresholding*. *Thresholding* digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra sesuai dengan keinginan. Proses binerisasi ini pada dasarnya adalah proses pengubahan kuantisasi pada citra menjadi dua level (gelap dan terang). Untuk mengubah citra tersebut menjadi citra biner, maka kita memerlukan suatu nilai yang disebut *threshold*. Dalam penelitian ini menggunakan nilai *threshold*=118.

4. Edge detection

Deteksi tepi (*edge detection*) pada suatu citra adalah suatu proses convolution atau non-convolution yang menghasilkan tepi-tepi dari objek citra, tujuannya adalah untuk menandai bagian yang menjadi detail citra dan untuk memperbaiki detail dari citra yang kabur karena eror atau adanya efek dari proses akuisisi citra. Suatu titik (x, y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan intensitas yang tinggi dari tetangganya.

Ada berbagai metode yang digunakan dalam proses deteksi tepi pada suatu citra, antara lain Metode Prewitt, Metode Sobel, Metode Horizontal, Metode Vertikal, dan Metode Difference. Pada pendeteksian objek ini digunakan Metode Difference karena menghasilkan tepi-tepi yang kontinu. Metode Difference merupakan suatu *metode non-convolution* yang bekerja dengan mencari nilai beda terbesar antara suatu piksel dengan delapan piksel yang berdekatan (*eight neighbours*). Metode Difference menggunakan nilai *threshold* sebagai pembandingan dengan beda nilai piksel terbesar yang dihasilkan.

5. Penentuan titik pusat

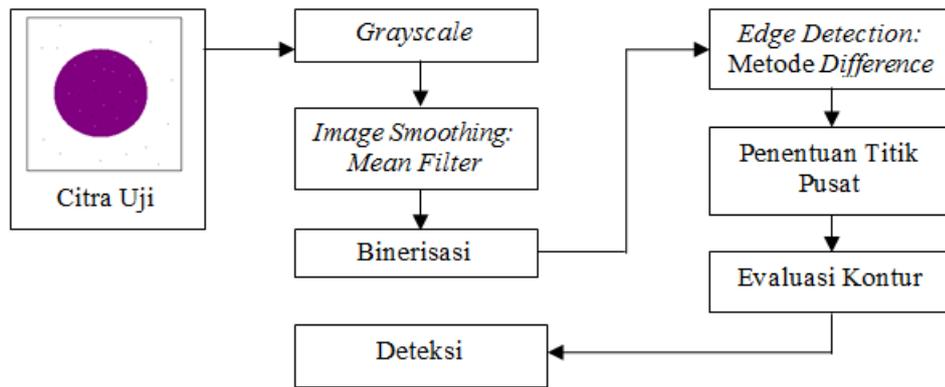
Perhitungan titik pusat berdasarkan pada mensuration. Mensuration merupakan suatu proses dimana pengukuran data dihitung untuk objek-objek citra (Parekh, [3]). Perhitungan yang paling sederhana adalah menghitung titik pusat dari objek citra berdasarkan koordinat titik-titik pikselnya dengan menggunakan persamaan berikut ini:

$$\bar{x} = \frac{1}{n} \sum_0^{n-1} x_i \quad \text{dan} \quad \bar{y} = \frac{1}{n} \sum_i^{n-1} y_i$$

dimana (x_i, y_i) adalah koordinat piksel i pada objek dan n = jumlah total piksel pada objek. Titik pusat (\bar{x}, \bar{y}) merupakan pusat gravitasi dari kontur yang akan digunakan dalam perhitungan evaluasi kontur.

2.2 Skema Deteksi Kontur

Berikut ini skema deteksi kontur dari sebuah citra uji dengan objek berbentuk lingkaran yang berisi *noise* yang dapat dilihat pada gambar di bawah ini:



Gambar 2. Skema Deteksi Kontur

2.3 Deteksi dengan Metode Kontinuitas Kontur

Deteksi objek didasarkan atas perolehan nilai penyimpangan atau deviasi yang menunjukkan bentuk-*bentuk* yang mengalami deformasi. Hasil-hasil perhitungan deviasi memperlihatkan kemungkinan untuk membedakan kontur-kontur yang ada yang merepresentasikan lingkaran, elips, DES (*Deformed Elliptic Shapes*), persegi, dan garis. Tabel 1 menunjukkan klasifikasi bentuk dari kontur yang dideteksi berdasarkan deviasinya.

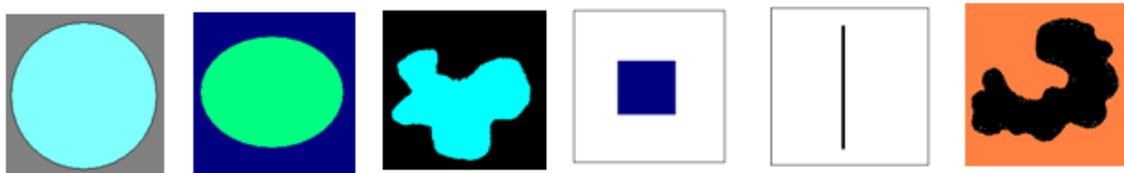
Tabel 1. Klasifikasi Bentuk Kontur Berdasarkan Deviasinya (Parekh, [3])

Kelas Objek	Bentuk	Range Nilai Deviasi
Lingkaran		00.00-01.00
Elips		27.00-35.00
DES		35.10-45.00
Persegi		20.00-25.00
Garis		89.00-90.00

3. Hasil dan Pembahasan

3.1 Data yang Digunakan

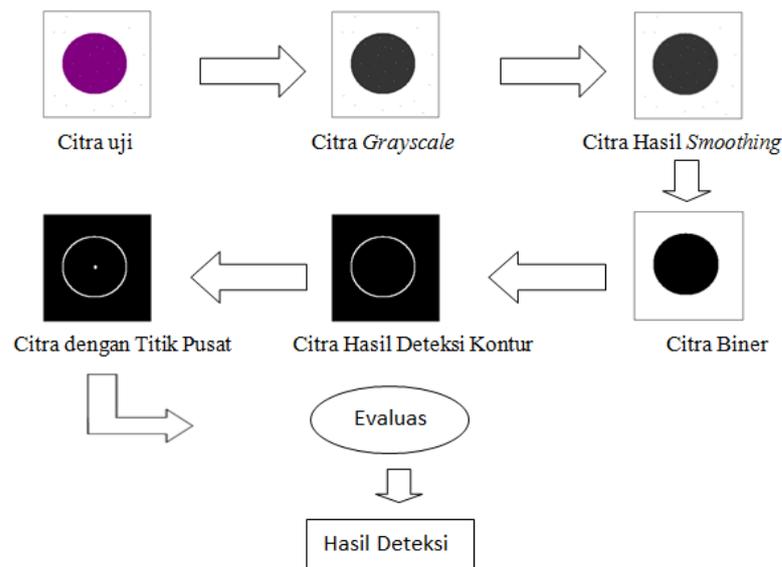
Citra uji yang digunakan pada proses deteksi ini menggunakan citra yang dibuat secara manual dengan resolusi 200×200 piksel berformat BMP sebanyak 60 buah. Citra uji terdiri dari berbagai macam bentuk yang diklasifikasikan ke dalam bentuk-bentuk yang sudah dikenal seperti lingkaran, elips, DES, persegi, garis, dan bentuk di luar klasifikasi (objek lain).3 menunjukkan sebagian citra uji yang digunakan.



Gambar 3. Contoh Citra Uji yang Digunakan (Kiri-Kanan: Lingkaran, Elips, DES, Persegi, Garis dan Objek Lain)

3.2 Proses Deteksi

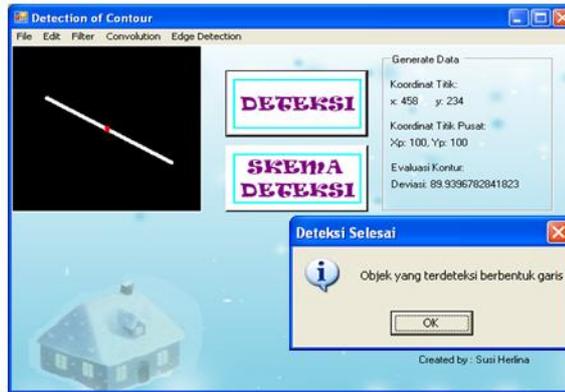
Gambar di bawah ini mengilustrasikan proses deteksi yang dilakukan.



Gambar 4. Skema Proses Deteksi Kontur

3.3 Proses Pengujian

Berdasarkan metode yang dikembangkan dan juga skema seperti yang digambarkan pada Gambar 4, dibuatlah sebuah program komputer (Sharp, [4]) yang digunakan sebagai *Testbed Program*, untuk menguji metode yang dikembangkan menggunakan data uji seperti yang dijelaskan pada bagian sebelumnya. Gambar 5 menunjukkan contoh *screen-shot* program yang dikembangkan dan ilustrasi numeris proses pengujian yang dilakukan. Tabel 2 menunjukkan ilustrasi numeris perhitungan pendeteksian garis. Dari Tabel 2 terlihat bahwa nilai D_α yang diperoleh adalah 90, yang berarti proses pendeteksian akan memberikan jawaban bahwa objek yang dideteksi adalah garis, karena nilai intervalnya ada di dalam interval garis (Lihat kriteria pada Tabel 1).



Gambar 5. Screen-shot Testbed Program untuk Pengujian

Tabel 2. Ilustrasi Numeris Perhitungan Pendeteksian Garis

No.	(x, y)	(x_p, y_p)	$\vec{m}_i = \langle m_x, m_y \rangle$	$\vec{g}_i = \langle g_x, g_y \rangle$	α
1.	(39,65)	(100,100)	$\langle 0.88, 0.49 \rangle$	$\langle 0, 0 \rangle$	90
2.	(51,71)	(100,100)	$\langle 0.86, -0.51 \rangle$	$\langle 0, 0 \rangle$	90
3.	(70,83)	(100,100)	$\langle 0.87, 0.49 \rangle$	$\langle 0, 0 \rangle$	90
4.	(95,97)	(100,100)	$\langle 0.86, 0.51 \rangle$	$\langle 0, 0 \rangle$	90
5.	(110,107)	(100,100)	$\langle -0.82, -0.57 \rangle$	$\langle 0, 0 \rangle$	90
6.	(123,115)	(100,100)	$\langle -0.84, -0.55 \rangle$	$\langle 0, 0 \rangle$	90
7.	(143,125)	(100,100)	$\langle -0.86, -0.50 \rangle$	$\langle 0, 0 \rangle$	90
No.	(x, y)	(x_p, y_p)	$\vec{m}_i = \langle m_x, m_y \rangle$	$\vec{g}_i = \langle g_x, g_y \rangle$	α
8.	(167,100)	(100,100)	$\langle -0.85, -0.52 \rangle$	$\langle 0, 0 \rangle$	90
$Jumlah \alpha = \sum_{i=0}^7 \cos^{-1}(\vec{g}_i \cdot \vec{m}_i)$					720
$D_\alpha = \frac{1}{8} \sum_{i=0}^7 \cos^{-1}(\vec{g}_i \cdot \vec{m}_i)$					90

3.4 Kriteria Akurasi Pendeteksian

Setiap citra hasil deteksi dikategorikan berdasarkan tingkat keakurasian deteksi yang diperoleh melalui perhitungan deviasi dari citra uji dalam software sesuai dengan bentuk objek berdasarkan penglihatan manusia dengan kriteria sebagai berikut (Myler, [2]):

- a. *True Positive*, jika objek dalam citra uji berbentuk lingkaran/elips/DES/persegi/garis, saat dideteksi, terdeteksi sesuai dengan bentuk objek dalam citra uji tersebut.
- b. *True Negative*, jika objek dalam citra uji dengan bentuk di luar klasifikasi, saat dideteksi, terdeteksi tidak sesuai dengan bentuk di luar klasifikasi objek dalam citra uji tersebut.
- c. *False Negative* jika objek dalam citra uji yang berbentuk lingkaran/elips/DES/persegi, saat dideteksi terdeteksi tidak sesuai dengan bentuk objek dalam citra uji tersebut. Kriteria ini disebut Error Tipe 1.
- d. *False Positive*, jika objek dalam citra uji dengan bentuk di luar klasifikasi, saat dideteksi terdeteksi sesuai dengan bentuk di luar klasifikasi objek dalam citra uji. Kriteria ini disebut Error Tipe 2.

Tabel 3 menunjukkan tingkat keakurasian keseluruhan dan Tabel 3 menunjukkan rician akurasi dari setiap objek yang dideteksi.

Tabel 3. Tingkat Keakurasian Deteksi Kontur Secara Keseluruhan

Jumlah	<i>True Positive + True Negative</i>	<i>False Negative + False Positive</i>	Tingkat Akurasi
60	51	9	85%

Tingkat akurasi ditentukan menggunakan formulasi

$$akurasi = \frac{(True\ Positives + True\ Negatives)}{Jumlah\ citra\ uji} \times 100\%$$

Tabel 4. Tingkat Keakurasian Deteksi Kontur Berdasarkan Objek yang Dideteksi

Objek	Jumlah	<i>True-positive</i>	<i>True Negative</i>	<i>False-negative</i>	<i>False-Positive</i>	Akurasi
Lingkaran	12	10	0	2	0	83.33%
Elips	12	10	0	2	0	83.33%
DES	12	11	0	1	0	91.67%
Persegi	12	10	0	2	0	83.33%
Garis	10	8	0	1	1	80.00%
Bentuk lain	2	0	2	0	0	100.00%
Jumlah	60	49	2	8	1	85%

4. Kesimpulan

Sebuah penelitian yang berkaitan dengan konsep citra digital dengan tujuan mengembangkan metode pendeteksian dan pengklasifikasian objek digital telah dilakukan dengan baik. Metode yang dikembangkan adalah metode kontinuitas kontur melalui estimasi klasifikasi objek menggunakan tingkat deviasi. Objek digital dalam pengujian berupa lingkaran, elips, DES, persegi, dan garis dan banyaknya objek yang digunakan adalah 60 buah. Tingkat akurasi yang dihasilkan melalui pengujian yang dilakukan adalah sebesar 85%.

Untuk pengembangan selanjutnya dapat dilakukan penambahan jenis/klasifikasi objek dan variasi dari objek yang ada sehingga dapat meningkatkan kualitas metode pengujian yang dikembangkan. Selain itu dapat dikembangkan metode lain dalam hal pendeteksian tepi sehingga proses deteksi tepi dapat dilakukan lebih efisien, lebih cepat dan lebih akurat.

Daftar Pustaka

- Gonzalez, Rafael C., and Woods, Richard E. 2007. *Digital Image Processing (3rd Ed.)*. Prentice Hall.
- Myler, Harley R., and Weeks, Arthur R. 2012. *Computer Imaging Recipes in C*. Prentice Hall.
- Parekh, Himani S., Thakore, Darshak G., and Jaliya, Udesang K.. 2014 *A Survey on Object Detection and Tracking Methods*. Int. J. of Innovative Research in Computer and Technology Eng., Vol. 2, No. 2.
- Sharp, John. 2013. *Microsoft Visual C# 2013 Step by Step (Step by Step Developer)*. Microsoft Press.
- Varberg, Dale, Purcell, Edwin, and Rigdon, Steve. 2006. *Calculus (9th Ed.)*. Pearson.

6. Yilmaz, Alper, Javed, Omar, Shah, and Mubarak Shah. 2006. *Object Tracking: A Survey*, ACM Computing Surveys. Vol. 38, No. 4, Article 13.

