

## DETEKSI KANDIDAT LINGKARAN MENGGUNAKAN KODE RANTAI

Ahmad Fashiha Hastawan<sup>1)</sup>, Indah Soesanti<sup>2)</sup>, Risma Septiana<sup>3)</sup>

<sup>1), 2), 3)</sup> Jurusan Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada Yogyakarta  
Jl Grafika No.2 Yogyakarta 55281

Email : [fashiha.mti13@mail.ugm.ac.id](mailto:fashiha.mti13@mail.ugm.ac.id)<sup>1)</sup>, [indsanti@gmail.com](mailto:indsanti@gmail.com)<sup>2)</sup>, [risma.septiana.mti13@mail.ugm.ac.id](mailto:risma.septiana.mti13@mail.ugm.ac.id)<sup>3)</sup>

### Abstrak

Deteksi lingkaran merupakan salah satu proses pengenalan pola yang sering digunakan dalam bidang computer vision. Untuk meningkatkan efisiensi

hasil deteksi maka sangat penting dilakukan proses pencarian potensi objek yang diduga sebagai kandidat lingkaran terlebih dahulu sebelum melakukan proses deteksi lingkaran lebih dalam. Bertujuan untuk menyikapi hal tersebut, penelitian ini mengajukan metode pencarian kandidat lingkaran baru berbasis kode rantai untuk mencari lokasi lingkaran dalam suatu citra. Metode ini menggunakan teknik pemrosesan citra digital untuk mendapatkan citra biner dan menerapkan deteksi tepi operator Canny untuk proses ekstraksi kontur sehingga didapatkan kontur berukuran satu piksel. Setelah itu dilakukan proses penelusuran kode rantai dengan menggunakan delapan arah ketetanggaan. Terakhir merupakan proses pencarian delapan titik acuan yang merupakan dasar dari metode yang diajukan. Dalam metode ini adanya kedelapan titik acuan perpindahan arah kode rantai dianggap sebagai salah satu fitur dari objek yang diduga sebagai lingkaran. Kedelapan titik yang menunjukkan posisi perubahan arah kode rantai yang signifikan ini didapatkan melalui proses first order difference code dari kode rantai kontur yang didapatkan. Hasil uji coba menunjukkan algoritma deteksi kandidat lingkaran menggunakan kode rantai memberikan hasil yang bagus baik dari segi efisiensi dan akurasi hasil.

**Kata kunci:** Kandidat lingkaran, kode rantai, first order difference code

### 1. Pendahuluan

Pengembangan dari metode-metode deteksi lingkaran secara umum dapat diklasifikasikan menjadi dua yaitu metode deterministik dan metode non deterministik. Tujuan utama dari metode deterministik lebih mengutamakan keakuratan deteksinya, sedangkan metode non deterministik lebih cenderung mengutamakan efisiensi dalam hal komputasinya.

Pengembangan metode deteksi lingkaran berbasis Hough Transform (HT) masih terus berkembang. Chen, dkk[1] mengembangkan metode HT untuk mendeteksi pusat lingkaran. Namun, hasil metode ini masih tidak efisien dalam waktu deteksinya. Untuk mengatasi hal tersebut Jiang[2] memperkenalkan teknik Randomized Hough

Transform menggunakan novel probability sampling dan features points. Sun, dkk[3] mengimplementasikan metode RHT untuk deteksi lingkaran jamak secara real-time. Hasil uji coba dari metode ini cukup baik dalam mendeteksi pergerakan lingkaran secara real-time.

Pengembangan metode deteksi lingkaran non HT banyak didasarkan pada karakteristik geometri objek. Shang, dkk[4] memperkenalkan deteksi lingkaran dengan menggunakan metode Right Triangles Inscribed in a Circle (RTIC). Metode ini menggunakan algoritma Right Triangles Search untuk proses pencarian kandidat lingkarannya. Metode ini memiliki hasil yang cukup akurat meskipun keakuratannya masih bergantung pada penentuan waktu pencarian Right Triangle. Metode deteksi lingkaran berdasarkan pelacakan tepi dan karakteristik geometri objek dilakukan oleh Chen dan Dong[5] dan Jiang, dkk[6]. Metode tersebut menerapkan pengecekan titik sampling terlebih dahulu untuk menentukan objek yang diduga sebagai lingkaran. Metode deteksi lingkaran yang dianggap lebih baik dari metode RHT yaitu Randomized Circle Detection (RCD) dikembangkan oleh [7][8][9]. Berbeda dengan RHT, RCD melakukan deteksi kandidat lingkaran terlebih dahulu dengan menerapkan algoritma verifikasi hipotesis yang dapat memangkas ketidakefektifan dalam proses deteksi. Cuevas dkk[10][11] memperkenalkan algoritma deteksi lingkaran menggunakan electro-magnetism optimization yang cocok untuk objek nyata dan juga bentuk pendekatan metode deteksi lingkaran lainnya berbasis Learning Otomata (LA). Kedua metode ini memberikan hasil yang baik namun memiliki algoritma yang rumit. Jia dkk [12] memperkenalkan metode deteksi lingkaran efektif non-HT untuk mendeteksi titik tengah dan jari-jari lingkaran berdasarkan Feature Circle Energy (FCE). Penelitian deteksi lingkaran yang didasarkan pada parallel operator dan perpotongan dua garis dilakukan oleh [13][14]. Kedua metode ini menggunakan perhitungan gradien kemiringan setiap titik sampling dalam menentukan pusat lingkarannya. Deteksi lingkaran dengan menggunakan penelusuran garis dan histogram diperkenalkan oleh Chen, dkk[15]. Algoritma deteksi lingkaran dengan menggunakan local voting dilakukan oleh Yang, dkk[16]. Metode deteksi lingkaran dengan melakukan pencarian kandidat lingkaran terlebih dahulu berdasarkan keterhubungan garis lengkung diperkenalkan oleh Akinlar dan Topal [17]. Metode ini juga menerapkan kontrol terhadap kesalahan deteksi yang disebut algoritma EDCircles. Algoritma deteksi lingkaran berdasar transformasi

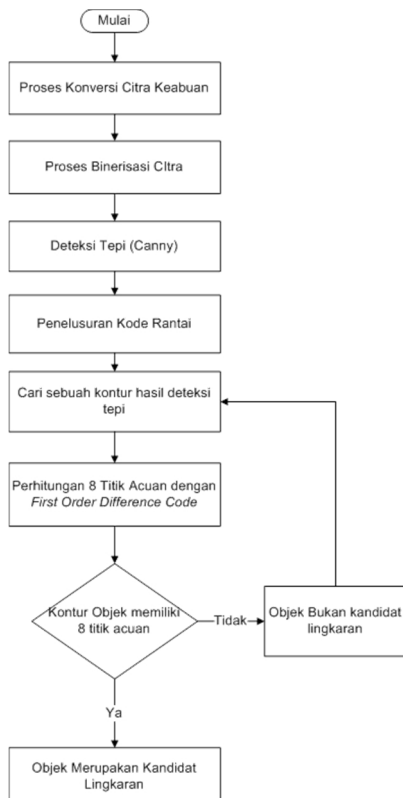
Radon diperkenalkan oleh Okman dan Akar[18]. Penerapan deteksi lingkaran menggunakan metode RANSAC dan *Laser Range Finder* untuk *positioning system* dilakukan oleh Huang, dkk[19]. Penggunaan kode rantai untuk deteksi lingkaran diperkenalkan oleh Fu dan Han[20]. Metode ini tidak melakukan operasi pencarian kandidat deteksi lingkaran terlebih dahulu sehingga dapat menyebabkan ketidakefektifan deteksi karena semua kontur dihitung. Hasil pengujian masih kurang presisi karena semua kode rantai dilibatkan secara mentah-mentah sehingga dapat menimbulkan ketidakpresisian dalam penentuan titik tengah.

Dibandingkan dengan metode-metode yang telah digunakan pada penelitian terdahulu, metode yang kami ajukan tidak melibatkan seluruh anggota kode rantai dalam penentuan titik tengah sehingga mengurangi potensi kesalahan. Selain itu metode ini mampu mendeteksi lebih dini objek yang diamati apabila bukan merupakan lingkaran, sehingga dapat meningkatkan efisiensi dalam proses deteksi.

## 2. Pembahasan

### I. Algoritme yang Diajukan

Secara umum ide dasar algoritme yang diajukan adalah dengan melakukan pra proses untuk mendapatkan titik tepi objek dengan menerapkan metode deteksi tepi canny. Setelah itu dilakukan penelusuran kode rantai terhadap tepi objek yang didapatkan. Setelah itu dilakukan proses pencarian 8 titik acuan untuk menentukan objek sebagai kandidat lingkaran. Gambar 1 menunjukkan diagram alir dari algoritme deteksi kandidat lingkaran.



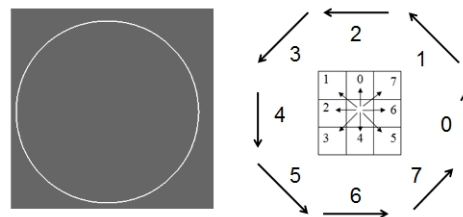
**Gambar 1.** Diagram Alir Metode

### A. Proses Pelacakan Tepi

Pelacakan tepi dilakukan dengan menggunakan kode rantai delapan arah ketetanggaan. Sebelum dilakukan pelacakan tepi dilakukan terlebih dahulu proses konversi citra ke aras keabuan dan konversi citra biner kemudian dilanjutkan proses deteksi tepi menggunakan operator *Canny*. Hasil deteksi tepi operator *Canny* ini menghasilkan objek dengan lebar satu piksel yang berwarna putih dan menghasilkan warna hitam untuk bagian yang dianggap bukan tepi sehingga memudahkan dalam pencarian titik permulaan pencarian kontur. Proses pelacakan tepi dilakukan dengan melakukan pencarian kontur yang ada di dalam dalam citra. Proses pencarian kontur ini dimulai dari suatu titik permulaan. Titik permulaan ini dilakukan dengan cara melakukan pencarian piksel yang memiliki intensitas piksel berwarna putih dari baris dan kolom citra. Kontur yang dicari kemudian disimpan dalam vektor array  $Vektor = \{Vektor1, Vektor2, Vektor3, \dots, Vn\}$ , dimana  $n$  merupakan jumlah keseluruhan kontur yang ada di dalam sebuah citra. Dalam vektor tersebut di dalamnya berisi titik koordinat  $(x,y)$ .

### B. Penentuan Kandidat Lingkaran

Proses penentuan kandidat lingkaran didasarkan pada konsep dasar perpindahan kode rantai pada lingkaran yang mana sebuah lingkaran seharusnya memiliki delapan perpindahan arah kode rantai yang urut. Dari kedelapan posisi perpindahan ini maka akan didapatkan delapan titik acuan. Proses ini dilakukan dengan menerapkan metode pencarian sudut berdasarkan first order difference code yang diperkenalkan oleh Nain, dkk[21]. Berikut gambar ilustrasi penentuan titik acuan:



**Gambar 2.** Konsep penentuan titik acuan lingkaran

Setelah proses pelacakan tepi, maka didapatkan beberapa vektor kontur ( $V_n$ ). Setiap vektor yang didapat kemudian dilakukan perhitungan mencari kode rantai yang kemudian dihitung dengan menggunakan algoritma *Difference Chain Code as Curvature*[21] untuk mencari 8 titik acuan. Titik acuan tersebut merupakan posisi koordinat dari perpindahan arah kode rantai. Langkah yang dilakukan adalah sebagai berikut :

1. Menghitung nilai slope; hasil perhitungan disimpan ditandai sebagai *Boundary Chain (BC)*. BC ini merupakan kumpulan kode rantai dengan hubungan delapan ketetanggaan. BC menunjukkan arah perpindahan antara piksel yang sedang diamati.

Slope dari piksel i yang sedang diamati adalah sebagai berikut :

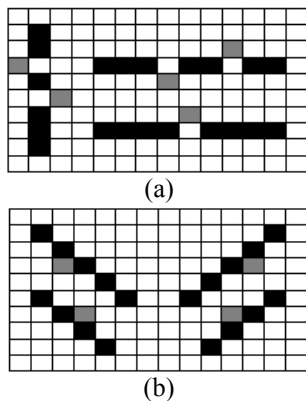
$$\phi_i = \tan^{-1} \left( \frac{\phi_i}{4} BC_i \right) \dots\dots(1)$$

2. Penghalusan BC

Pada tahap penulusuran *Boundary Chain* ini memiliki beberapa kemungkinan kesalahan akibat hasil dari proses deteksi tepi diantaranya yaitu :

- a. Kesalahan pertama yang terjadi adalah terdapat piksel yang berubah arah pada sekumpulan piksel dalam satu garis lurus, dimana hanya sebuah piksel yang berubah diantara garis lurus ini.
- b. Kesalahan kedua yang terjadi adalah terdapat piksel tambahan terhadap sekumpulan piksel pada arah diagonal terhadap piksel acuan, dimana piksel acuan sudah memiliki hubungan ketetangaan terhadap salah satu arah pada posisi garis lurus.

Ilustrasi dari kedua bentuk kesalahan yang terjadi dapat dilihat pada Gambar 3 berikut :



Gambar 3. Ilustrasi kesalahan penelusuran BC : (a) kesalahan pertama, (b) kesalahan kedua

Untuk mengatasi permasalahan yang terjadi maka dilakukan tahap kedua yaitu *Smoothing Boundary Chain (Smoothing BC)*. Dalam proses *Smoothing BC* dilakukan dua proses meliputi penggantian kode serta penyesuaian piksel tunggal pada sekumpulan kode dominan dengan ketentuan pada Tabel 1.

Tabel 1. Ketentuan penggantian kode BC

No	BC sebelum proses smoothing	BC setelah proses smoothing
1	(0, 7, 1) / (0, 1, 7)	(0, 0, 0)
2	(2, 3, 1) / (2, 1, 3)	(2, 2, 2)
3	(4, 5, 3) / (4, 3, 5)	(4, 4, 4)
4	(6, 5, 7) / (6, 7, 5)	(6, 6, 6)
5	(1, 2, 0) / (1, 0, 2)	(1, 1)
6	(3, 4, 2) / (3, 2, 4)	(3, 3)
7	(5, 6, 4) / (5, 4, 6)	(5, 5)
8	(7, 0, 6) / (7, 6, 0)	(7, 7)

Sedangkan untuk ketentuan penyesuaian piksel tunggal pada sekumpulan kode dominan adalah sebagai berikut.

- a) Cek dua piksel sebelumnya dan dua piksel selanjutnya pada posisi BC. Jika  $BC_{i \pm n(n=1,2)}$  adalah sama dan  $BC_i \neq BC_{i \pm n}$  maka  $BC_i \neq BC_{i+1}$
- b) Cek dua piksel sebelumnya dan piksel setelahnya atau dua piksel setelahnya dan satu piksel sebelumnya pada posisi BC. Jika  $(BC_{i-n(n=1,2)} = BC_{i+1} \text{ AND } \neq BC_i)$  atau  $(BC_{i+n(n=1,2)} = BC_{i-1} \text{ AND } \neq BC_i)$  maka  $BC_i \neq BC_{i \pm n}$  maka  $BC_i = BC_{i+1}$  atau  $(BC_i = BC_{i-1})$
- c) Cek dua kode BC sebelumnya dan kode kedua BC setelahnya pada posisi piksel tersebut atau cek dua kode BC setelahnya dan kode kedua sebelumnya pada posisi piksel tersebut. Jika  $(BC_{i-n(n=1,2)} = BC_{i+2} \text{ AND } BC_i \neq BC_{i+2})$  atau  $(BC_{i+n(n=1,2)} = BC_{i-2} \text{ AND } BC_i \neq BC_{i-2})$  maka  $BC_i \neq BC_{i \pm n}$  maka  $BC_i = BC_{i+2}$  atau  $(BC_i = BC_{i-2})$

- 3. Menghindari kesalahan perpindahan arah  
 Hasil dari *Smoothing Boundary Chain (Smoothing BC)* ini dimodifikasi dan dilakukan pembentukan kode baru kemudian hasilnya disebut *Modified Chain (MC)*. Modifikasi dilakukan dengan cara memeriksa kode BC sebelumnya dan kode BC setelahnya pada posisi BC. Apabila  $BC_{i+1} = BC_{i-1}$  dan  $BC_i \neq BC_{i \pm 1}$  maka  $BC_i = BC_{i-1}$  atau  $BC_i = BC_{i+1}$ .

Untuk menghindari kesalahan perpindahan arah agar tidak terjadi kesalahan dalam penentuan titik acuan maka dilakukan pemeriksaan terhadap kode MC. Pemeriksaan kode MC ini membutuhkan nilai ambang (*l*) untuk menentukan apakah kumpulan kode yang diperiksa merupakan perpindahan yang dimaksud atau tidak. Berikut rumus untuk menentukan nilai ambangnya.

$$l = (\text{Boundarylenh})^{\frac{1}{4}} \dots\dots(2)$$

Dalam pemeriksaan kode MC, untuk piksel posisi *l*, jika  $MC_{i+1} = MC_{i-1}$ , ini menunjukkan perpindahan pertama dari suatu kode MC. Dalam perubahan arah  $MC_i = MC_{i-n(n=1,2,3,\dots,l)}$ , apabila panjang dari kode tersebut  $< l$  maka kumpulan dari kode tersebut tidak dianggap sebagai perubahan arah. Sebagai contoh apabila  $l=4$ , dan misalkan kode rantai adalah 2222222333222222224443333333 dan perpindahan kode MC berada pada posisi 1,8,11,21,24. Kode MC pada posisi 8 s/d 11 dianggap tidak ada perpindahan dan dianggap sebagai satu kesatuan karena  $l=4$  sedangkan 8 s/d 11 hanya  $l=11$

- 4. Menghitung kurva menggunakan *First Order Difference Codes*

*Difference Codes* ini diperoleh dari hasil *Modified Chain (MC)*. Tahap ini merupakan tahap yang dilakukan untuk menghindari kesalahan titik acuan serta untuk menentukan letak titik acuan yang benar. Hasil dari proses ini disebut DC. Untuk menghitung DC digunakan rumus sebagai berikut :

$$DC_i = (MC_{i+1} - MC_i + 8) \bmod 8 \quad \dots\dots(3)$$

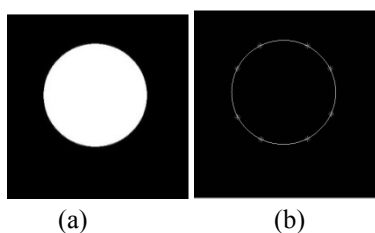
5. Menentukan 8 titik acuan

Dari perhitungan DC ini maka akan diperoleh posisi-posisi perpindahan kode yang memiliki nilai dan terletak diantara kode bernilai nol. Apabila perpindahan  $P_i - P_{i-1} > 1$ , maka titik perpindahan pada posisi  $P_i$  dan  $P_{i-1}$  merupakan calon titik acuan. Sebagai contoh apabila kode MC adalah 11111222222244444444, maka  $DC = 000100000002000000$ . DC yang bernilai bukan nol  $P_i$  adalah 5, 13. Perubahan yang terjadi antara DC yang bernilai bukan nol adalah  $5 > 1$ , maka calon titik acuan berada pada posisi 5 dan 13. Setelah semua calon titik acuan diperoleh kemudian dilakukan pemeriksaan hubungan perubahan arah kode rantai. Apabila dari hasil pemeriksaan hubungan arah tersebut didapatkan adanya delapan kali perubahan arah yang urut, maka objek tersebut ditentukan sebagai kandidat lingkaran.

II. Hasil Pengujian dan Analisis

Hasil pengujian menggunakan citra sintesis dengan berbagai macam ukuran dan variasi macam-macam objek di dalam citra tersebut. Citra sintetis diperoleh dari citra hasil buatan perangkat lunak Paint dan bertipe citra \*.png. Perangkat keras yang digunakan dalam pengujian adalah intel i-4200U CPU @ 1.6GHz(4 CPUs), RAM 4 GB.

Hasil dari pengujian metode yang digunakan terhadap citra objek lingkaran tunggal dapat dilihat pada Gambar 4. Citra asli dapat dilihat pada Gambar 4(a) yang menunjukkan objek lingkaran tunggal tanpa ada objek lain di dalam citra tersebut. Pada Gambar 4(b) dapat dilihat, objek lingkaran dibubuhkan penanda pada objek yang menunjukkan bahwa metode yang digunakan mampu mendeteksi bahwa objek di dalam citra tersebut merupakan kandidat lingkaran.

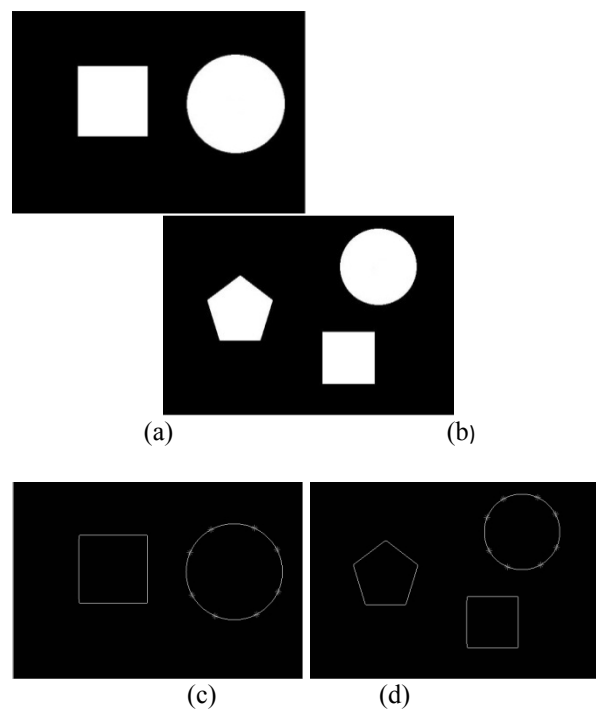


Gambar 4. Deteksi kandidat lingkaran objek tunggal

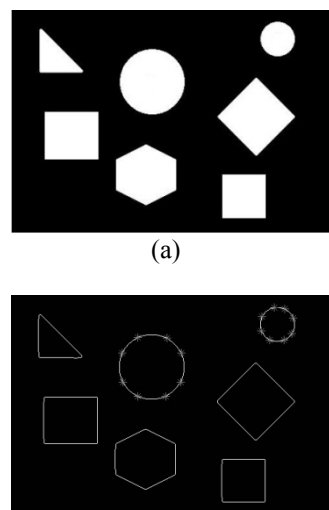
Gambar 5 menunjukkan hasil pengujian terhadap citra dengan objek jamak dimana dalam sebuah citra terdapat

lebih dari satu objek. Gambar 5 (a) (b) menunjukkan citra asli sebagai masukan system dan Gambar 5 (c) (d) merupakan hasil dari deteksi kandidat lingkaran. Dari kedua hasil pengujian menunjukkan metode yang digunakan mampu mendeteksi objek yang diduga sebagai lingkaran.

Pengujian terhadap citra yang memiliki objek jamak sekaligus memiliki objek lingkaran lebih dari satu ditunjukkan pada Gambar 6. Pada Gambar 6 (a) merupakan citra asli, dan Gambar 6 (b) merupakan hasil deteksi kandidatnya. Dari citra hasil pengujian dapat kita amati bahwa metode yang digunakan juga mampu mendeteksi kandidat lingkaran dengan baik meskipun objek di dalam citra bermacam-macam dan kandidat lingkarannya lebih dari satu.

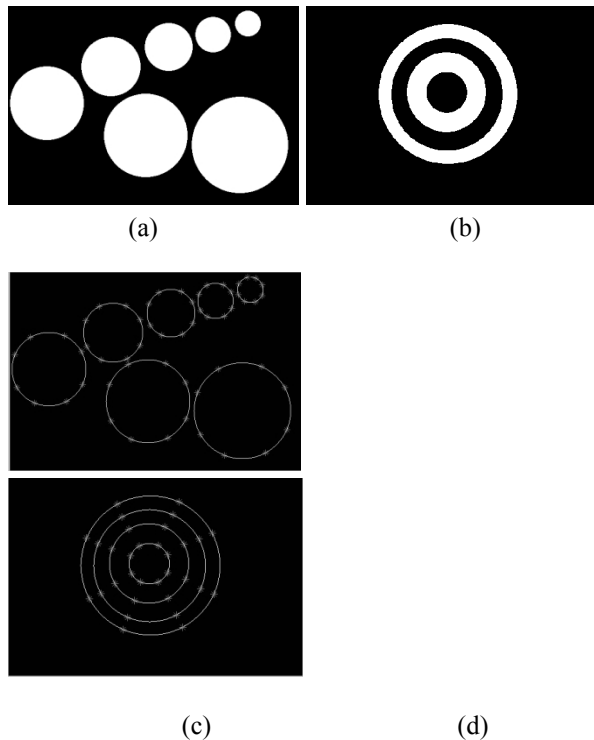


Gambar 5. Deteksi kandidat lingkaran objek jamak



(b)  
**Gambar 6.** Deteksi kandidat lingkaran jamak

Pengujian terhadap citra yang memiliki objek lingkaran jamak dapat dilihat pada Gambar 7. Pada Gambar 7 (a)(b) merupakan citra asli, dan Gambar 7 (c) (d) merupakan hasil deteksi kandidatnya. Dari citra hasil pengujian dapat kita amati bahwa metode yang digunakan mampu mendeteksi semua kandidat lingkaran dalam berbagai ukuran lingkaran yang berbeda.



(c) (d)  
**Gambar 7.** Deteksi kandidat lingkaran jamak

Perbandingan waktu eksekusi dari masing-masing pengujian dapat dilihat pada Tabel 2 berikut :

**Tabel 2.** Perbandingan waktu eksekusi pengujian

No	Gambar	Waktu (s)	Jumlah Objek
1	Gambar 4b	0.41726	1
2	Gambar 5c	0.59369	2
3	Gambar 5d	0.67644	3
4	Gambar 6b	0.81247	7
5	Gambar 7c	1.06430	7
6	Gambar 7d	0.75219	4

### 3. Kesimpulan

Hampir semua proses deteksi lingkaran membutuhkan proses penentuan kandidat lingkaran terlebih dahulu untuk meringankan proses komputasi dan menghindari ketidakefektifan dalam pendeteksian. Apabila citra yang diujikan bukan memiliki objek jamak di dalamnya, maka lokasi kandidat lingkaran akan sangat sulit dilakukan dan prosesnya akan sangat rumit. Deteksi

kandidat- kandidat lingkaran dengan menggunakan kode rantai mampu mengatasi permasalahan tersebut. Selain itu, algoritme sederhana yang diterapkan dapat menyelesaikan permasalahan kompleksitas algoritme penentuan kandidat lingkaran pada metode sebelumnya.. Keunggulan lain dari metode ini adalah mendeteksi kandidat lingkaran dalam suatu citra secara cepat dan akurat.

Algoritme yang diajukan , saat ini hanya dapat digunakan untuk mendeteksi kandidat lingkaran. Sehingga belum dapat digunakan untuk memprediksi suatu bentuk lingkaran. Untuk pengembangan pada penelitian selanjutnya, diharapkan pengenalan lingkaran dapat dilakukan dengan cara memprediksi suatu bentuk walaupun bentuk lingkaran dalam keadaan tidak utuh.

### Daftar Pustaka

- [1] X. Chen, L. Lu, and Y. Gao, "A new concentric circle detection method based on Hough transform," 2012, pp. 753–758.
- [2] L. Jiang, "Efficient randomized Hough transform for circle detection using novel probability sampling and feature points," *Optik*, vol. 123, no. 20, pp. 1834–1840, 2012.
- [3] H. Sun, Y. Mao, N. Yang, and D. Zhu, "A real-time and robust multi-circle detection method based on randomized Hough transform," 2012, pp. 175–180.
- [4] F. Shang, J. Liu, X. Zhang, and D. Tian, "An improved circle detection method based on right triangles inscribed in a circle," 2009, vol. 6, pp. 382–387.
- [5] A. Chen and G. Dong, "Efficient method for rapidly detecting circles based on edge-tracking," 2009, vol. 1, pp. 402–405.
- [6] Y. Jiang, X. Fu, and H. Gao, "A new circular region detection algorithm based on the geometric characteristics," *J. Softw.*, vol. 8, no. 11, pp. 2899–2907, 2013.
- [7] L.-Q. Jia, C.-Z. Peng, H.-M. Liu, and Z.-H. Wang, "A fast randomized circle detection algorithm," 2011, vol. 2, pp. 820–823.
- [8] Y.-H. Huang, K.-L. Chung, W.-N. Yang, and S.-H. Chiu, "Efficient symmetry-based screening strategy to speed up randomized circle-detection," *Pattern Recognit. Lett.*, vol. 33, no. 16, pp. 2071–2076, 2012.
- [9] K.-L. Chung, Y.-H. Huang, S.-M. Shen, A. S. Krylov, D. V. Yurin, and E. V. Semeikina, "Efficient sampling strategy and refinement strategy for randomized circle detection," *Pattern Recognit.*, vol. 45, no. 1, pp. 252–263, 2012.
- [10] E. Cuevas, D. Oliva, D. Zaldivar, M. Pérez-Cisneros, and H. Sossa, "Circle detection using electro-magnetism optimization," *Inf. Sci.*, vol. 182, no. 1, pp. 40–55, 2012.
- [11] E. Cuevas, F. Wario, D. Zaldivar, and M. Pérez-Cisneros, "Circle detection on images using learning automata," *IET Comput. Vis.*, vol. 6, no. 2, pp. 121–132, 2012.
- [12] L.-Q. Jia, H.-M. Liu, Z.-H. Wang, and H. Chen, "An effective non-HT circle detection for centers and radii," 2011, vol. 2, pp. 814–818.
- [13] L.-Q. Jia and C.-Z. Peng, "A new circle detection method based on parallel operator," 2012, vol. 3, pp. 1085–1090.
- [14] L. Luo, D. Xu, Z. Zhang, J. Zhang, and W. Qu, "A fast and robust circle detection method using perpendicular bisector of chords," 2013, pp. 2856–2860.
- [15] M. Chen, F. Zhang, Z. Du, and R. Liu, "Circle detection using scan lines and histograms," *Opt. Rev.*, vol. 20, no. 6, pp. 484–490, 2013.
- [16] H. Yang, J. Luo, Z. Shen, and W. Wu, "A local voting and refinement method for circle detection," *Optik*, vol. 125, no. 3, pp. 1234–1239, 2014.



- [17] C. Akinlar and C. Topal, "EDCircles: A real-time circle detector with a false detection control," *Pattern Recognit.*, vol. 46, no. 3, pp. 725–740, 2013.
- [18] O. E. Okman and G. B. Akar, "A circle detection approach based on Radon Transform," 2013, pp. 2119–2123.
- [19] X. Huang, T. Sasaki, H. Hashimoto, and F. Inoue, "Circle detection and fitting based positioning system using laser range finder," 2010, pp. 442–447.
- [20] Z. Fu and Y. Han, "A circle detection algorithm based on mathematical morphology and chain code," 2012, pp. 253–256.
- [21] N. Nain, V. Laxmi, B. Bhadviya, and A. Gopal, "Corner detection using difference chain code as curvature," 2007, pp. 821–825.

### **Biodata Penulis**

**Ahmad Fashiha Hastawan**, memperoleh gelar Sarjana Teknik (S.T), Jurusan Teknik Elektro Universitas Diponegoro (UNDIP) Semarang, lulus tahun 2013. Saat ini sedang menyelesaikan study Magister Engineering (M.Eng) di jurusan Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada Yogyakarta.

**Indah Soesanti**, Saat ini menjadi Dosen di Jurusan Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada Yogyakarta.

**Risma Septiana**, memperoleh gelar Sarjana Teknik (S.T), Jurusan Teknik Elektro Universitas Diponegoro (UNDIP) Semarang, lulus tahun 2012. Saat ini sedang menyelesaikan study Magister Engineering (M.Eng) di jurusan Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada Yogyakarta