

PENERAPAN ALGORITMA EFFICIENT RANDOMIZED UNTUK MENGHITUNG JUMLAH KOIN DAN BOLA

Yuliana Melita Pranoto¹, Endang Setyati²

¹Teknik Informatika, Sekolah Tinggi Teknik Surabaya
Jl. Ngagel Jaya Tengah 73-77 Surabaya
E-mail: ymp@stts.edu

²Teknik Informatika, Sekolah Tinggi Teknik Surabaya
Jl. Ngagel Jaya Tengah 73-77 Surabaya
E-mail: endang@stts.edu

ABSTRAK

Mendeteksi lingkaran dari citra digital penting dalam pengenalan bentuk. Efficient Randomized Algorithm berfungsi untuk mendeteksi lingkaran pada citra atau gambar. Bila metode Hough Transform menggunakan akumulator untuk menyimpan informasi parameter, maka Efficient Randomized tidak perlu menggunakan akumulator.

Konsep utama yang digunakan dalam Efficient Randomized Algorithm adalah dengan memilih secara acak empat piksel tepi dalam sebuah gambar dan menetapkan kriteria jarak untuk menentukan apakah ada kemungkinan lingkaran pada gambar tersebut. Setelah menemukan lingkaran, akan dilakukan proses pengumpulan bukti untuk menentukan apakah lingkaran yang ditemukan tersebut benar lingkaran atau tidak.

Hasil percobaan dengan menggunakan 100 gambar, menunjukkan bahwa Efficient Randomized Algorithm lebih cepat mendeteksi dan menggunakan memori lebih kecil daripada metode Hough Transform.

Kata kunci: Efficient Randomized, Hough Transform, akumulator, piksel tepi.

ABSTRACT

Detecting circles from a digital image is important in shape recognition. Efficient Randomized Algorithm is used to detect circles on images. While Hough Transform method uses accumulators to store the parameter information, Efficient Randomized does not use the same idea.

Main concepts of Efficient Randomized Algorithm is to randomly select four edge pixels in one image and then establish distance criteria to determine whether there are circles in the image. After finding the circle, this algorithm will gather evidences to determine whether the discovered circle are indeed circles or not.

Results of experiments using 100 images showing Efficient Randomized Algorithm detects cricles faster and uses less memory than Hough Transform method.

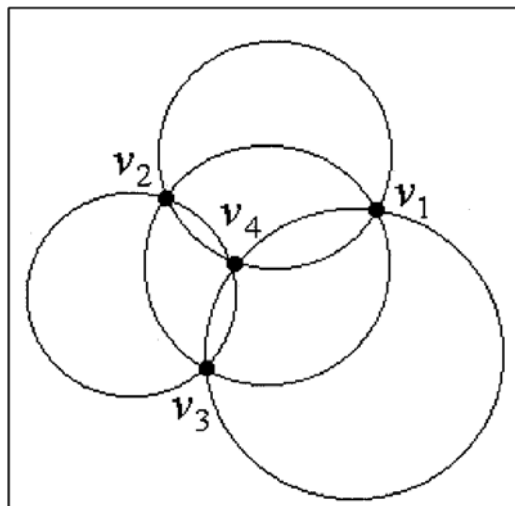
Keywords: digital image, Efficient Randomized, Hough Transform, accumulator, edge pixels.

1. PENDAHULUAN

Mendeteksi lingkaran dari citra digital sangat penting dalam pengenalan bentuk atau objek. Hough Transform terkenal dengan metode untuk deteksi bentuk lingkaran. Sebagai contoh (x, y) adalah sebuah piksel tepi pada suatu lingkaran dengan pusat koordinat (a, b) dan jari-jari r , lingkaran dapat dinyatakan seperti pada persamaan 1.

$$(x - a)^2 + (y - b)^2 = r^2 \dots\dots\dots(1)$$

Proses Efficient Randomized disini dilakukan dengan memilih empat piksel tepi dalam gambar atau citra secara acak. Seperti ditunjukkan dalam gambar 1, empat piksel tepi umumnya dapat menentukan empat lingkaran. Jika keempat piksel tepi yang dipilih secara acak berasal dari lingkaran yang sama, ada kemungkinan lingkaran ini adalah sebuah lingkaran yang nyata. Dari sini kemudian dikumpulkan bukti untuk lebih memastikan apakah mungkin lingkaran yang ditemukan tersebut merupakan lingkaran yang nyata atau tidak.



Gambar 1. Empat Lingkaran Yang Tepinya Saling Terhubung

2. METODE PENELITIAN

Metode Efficient Randomized terdiri tiga bagian. Bagian pertama menyajikan kriteria yang digunakan untuk menentukan jarak empat piksel tepi lingkaran yang dipilih apakah mungkin atau tidak. Bagian kedua menjelaskan cara untuk memeriksa apakah mungkin lingkaran tersebut adalah lingkaran sejati. Bagian ketiga berisi algoritma dari Efficient Randomized.

2.1 Kriteria Jarak Penentuan Kemungkinan Lingkaran

Persamaan lingkaran dapat ditulis seperti pada persamaan 2.

$$2xa + 2yb + d = x^2 + y^2 \dots\dots\dots(2)$$

Dimana :

$$d = r^2 - a^2 - b^2$$

r = radius

(a,b) = pusat lingkaran

(x,y) = piksel tepi

Variabel V menyimpan semua piksel tepi dari sebuah gambar. Misal tiga piksel tepi dalam gambar adalah $v_i = (x_i, y_i)$ dimana $i = 1, 2, 3$. Jika $v_1, v_2,$ dan v_3 merupakan

noncollinear, dapat ditentukan satu lingkaran (dinotasikan dengan C_{123}) dengan pusat (a_{123}, b_{123}) dengan radius r_{123} . Dari persamaan 2 dan fakta bahwa lingkaran tersebut melewati tiga piksel, diperoleh persamaan 3, 4, dan 5.

$$2x_1a_{123} + 2y_1b_{123} + d_{123} = x_1^2 + y_1^2 \dots\dots\dots(3)$$

$$2x_2a_{123} + 2y_2b_{123} + d_{123} = x_2^2 + y_2^2 \dots\dots\dots(4)$$

$$2x_3a_{123} + 2y_3b_{123} + d_{123} = x_3^2 + y_3^2 \dots\dots\dots(5)$$

Dimana:

$$d_{123}^2 = r_{123}^2 - a_{123}^2 - b_{123}^2.$$

Persamaan 6 adalah representasi persamaan 3, 4, dan 5 ke dalam bentuk matriks

$$\begin{pmatrix} 2x_1 & 2y_1 & 1 \\ 2x_2 & 2y_2 & 1 \\ 2x_3 & 2y_3 & 1 \end{pmatrix} \begin{pmatrix} a_{123} \\ b_{123} \\ d_{123} \end{pmatrix} = \begin{pmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ x_3^2 + y_3^2 \end{pmatrix} \dots\dots\dots(6)$$

Dengan menerapkan eliminasi Gauss, diperoleh persamaan 7.

$$\begin{pmatrix} 2x_1 & 2y_1 & 1 \\ 2(x_2 - x_1) & 2(y_2 - y_1) & 0 \\ 2(x_3 - x_1) & 2(y_3 - y_1) & 0 \end{pmatrix} \begin{pmatrix} a_{123} \\ b_{123} \\ d_{123} \end{pmatrix} = \begin{pmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 - (x_1^2 + y_1^2) \\ x_3^2 + y_3^2 - (x_1^2 + y_1^2) \end{pmatrix} \dots\dots\dots(7)$$

Karena v_1 , v_2 , dan v_3 merupakan noncollinear, diperoleh $(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \neq 0$. Dengan aturan Cramer, dapat diperoleh pusat (a_{123}, b_{123}) pada persamaan 8 dan 9.

$$a_{123} = \frac{\begin{vmatrix} x_2^2 + y_2^2 - (x_1^2 + y_1^2) & 2(y_2 - y_1) \\ x_3^2 + y_3^2 - (x_1^2 + y_1^2) & 2(y_3 - y_1) \end{vmatrix}}{4((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))} \dots\dots\dots(8)$$

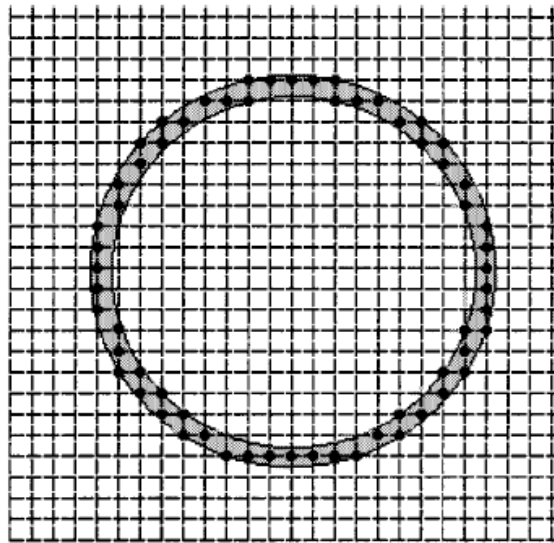
$$b_{123} = \frac{\begin{vmatrix} 2(x_2 - x_1) & x_2^2 + y_2^2 - (x_1^2 + y_1^2) \\ 2(x_3 - x_1) & x_3^2 + y_3^2 - (x_1^2 + y_1^2) \end{vmatrix}}{4((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))} \dots\dots\dots(9)$$

Setelah pusat (a_{123}, b_{123}) , radius dapat dihitung dengan persamaan 10.

$$r_{123} = \sqrt{(x_i - a_{123})^2 + (y_i - b_{123})^2} \dots\dots\dots(10)$$

Dimana :

$i = 1, 2$ dan 3 .



Gambar 2. Sebuah Lingkaran Digital

Persamaan $(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) = 0$ menunjukkan bahwa ketiga piksel segaris dan piksel tidak dapat membentuk sebuah lingkaran. Misalkan $v_4 = (x_4, y_4)$ adalah piksel tepi keempat, maka jarak antara v_4 dan lingkaran C_{123} dapat dihitung dengan persamaan 11.

$$d_{4 \rightarrow 123} = \left| \sqrt{(x_4 - a_{123})^2 + (y_4 - b_{123})^2} - r_{123} \right| \dots \dots \dots (11)$$

Bila v_4 terletak pada lingkaran C_{123} , maka nilai $d_{4 \rightarrow 123}$ pada Persamaan 11 adalah 0. Pada gambar digital, jarang terjadi piksel tepi yang terletak persis di tepi lingkaran. Oleh karena itu, tujuan dari deteksi lingkaran ini adalah untuk mendeteksi satu set tepi pixel yang terletak tidak persis tetapi kira-kira pada lingkaran digital (lihat gambar 2). Untuk kemudahan, kumpulan piksel tepi yang membentuk sebuah lingkaran digital juga disebut sebagai lingkaran. Kumpulan piksel tepi ini dikatakan co-circular.

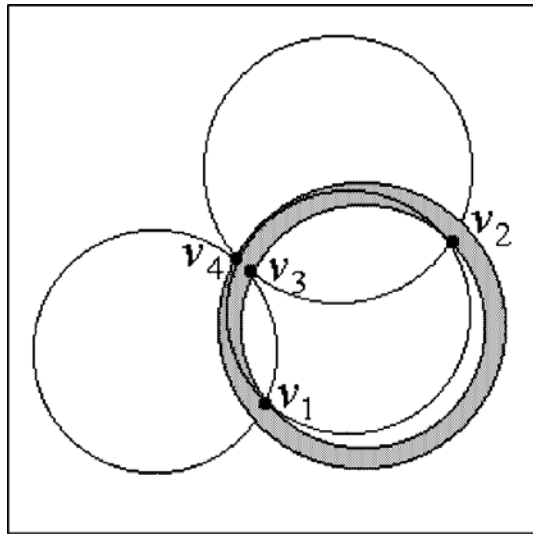
Perhatikan gambar 3, v_4 terletak pada lingkaran C_{123} , dan nilai $d_{4 \rightarrow 123}$ pada persamaan 11 sangat kecil. Oleh karena itu, persamaan 11 dapat digunakan untuk menentukan apakah v_4 terletak pada lingkaran C_{123} atau tidak.

Untuk mempermudah, lingkaran C_{ijk} yang dilewati oleh v_i, v_j, v_k dengan titik pusat $(a_{ijk}; b_{ijk})$ dan jari-jari r_{ijk} . Jarak antara v_1 dan lingkaran C_{ijk} dinyatakan sebagai $d_{1 \rightarrow ijk}$. Pusat lingkaran, radius, dan jarak dapat dihitung dengan persamaan 8 - 11. Sebagai contoh, persamaan 11 dapat seperti pada persamaan 12.

Langkah berikutnya adalah untuk memutuskan ada lingkaran yang ditentukan oleh tiga atau empat piksel tepi dan apakah keempat piksel tepi tersebut terletak pada lingkaran.

Diberikan empat piksel tepi $v_i, i = 1, 2, 3, 4$ (ada 4 lingkaran), yaitu $C_{123}, C_{124}, C_{134},$ dan C_{234} , dengan empat jarak, yaitu, $d_{4 \rightarrow 123}; d_{3 \rightarrow 124}; d_{2 \rightarrow 134},$ dan $d_{1 \rightarrow 234}$. Ketika ditemukan satu jarak yang lebih kecil dari batas/threshold (T_d) yang diberikan, misal $T_d = 0.5$ atau 1, dapat diambil kesimpulan bahwa keempat piksel tepi ini co-circular. Misalnya, jika $d_{4 \rightarrow 123}$ adalah jarak pertama yang memenuhi syarat $d_{4 \rightarrow 123} \leq T_d$, dapat disimpulkan bahwa keempat piksel tepi adalah co-circular dan lingkaran C_{123} mungkin benar sebuah lingkaran. Bila C_{ijk} mungkin adalah sebuah lingkaran (disebut sebagai

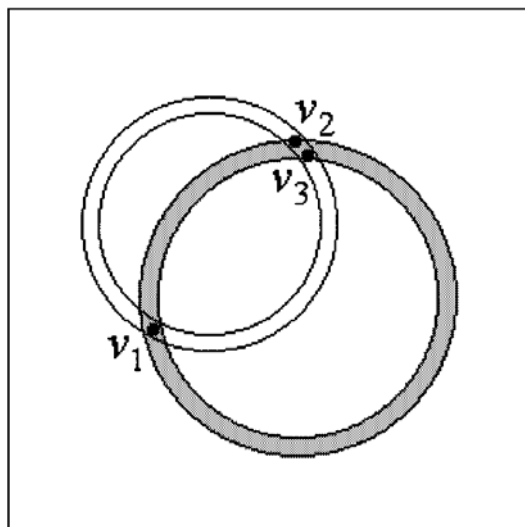
possible circle), maka dikatakan ketiga piksel tepi v_i , v_j , dan v_k menjadi anggota dari *possible circle*.



Gambar 3. Contoh Empat Piksel Dalam Lingkaran Digital

$$d_{l \rightarrow ijk} = \left| \sqrt{(x_l - a_{ijk})^2 + (y_l - b_{ijk})^2} - r_{ijk} \right| \dots\dots\dots(12)$$

Ada beberapa kasus yang tidak diinginkan muncul misalnya ketika dua dari tiga piksel anggota *possible circle* letaknya terlalu dekat, mungkin saja *possible circle* tersebut bukan lingkaran benar. Seperti ditunjukkan pada gambar 4, v_1 , v_2 , dan v_3 terletak pada lingkaran yang benar (lingkaran yang lebih besar), tetapi lingkaran yang lebih kecil yang ditentukan oleh v_1 , v_2 , dan v_3 berbeda dari lingkaran yang benar. Kasus ini terjadi ketika letak v_2 dan v_3 terlalu dekat. Untuk menghindari hal ini, jarak antara dua piksel anggota harus lebih besar dari batas yang diberikan (T_a). Hal ini berarti ketiga piksel anggota punya alasan yang kuat untuk menjadi anggota dari *possible circle*.



Gambar 4. Kasus dimana letak v_2 dan v_3 terlalu dekat

2.2 Penentuan Lingkaran yang Benar

Langkah selanjutnya adalah mengumpulkan bukti-bukti untuk memastikan bahwa *possible circle* adalah lingkaran yang benar.

Pada awalnya, counter (C) diinisialisasi dengan nilai 0, variabel ini digunakan untuk menghitung berapa banyak piksel tepi yang terletak pada *possible circle*. Untuk setiap piksel tepi v_1 jarak $d_{1 \rightarrow ijk}$ dapat dihitung dengan persamaan 12. Jika $d_{1 \rightarrow ijk}$ tidak lebih besar dari treshold jarak (T_d), maka counter C ditambah dengan satu dan keluarkan v_1 dari V, sebaliknya lanjutkan proses ke piksel tepi berikutnya.

Proses tersebut diatas dilakukan hingga semua piksel tepi dalam V telah diperiksa. Dalam proses pengumpulan bukti, variabel n_p menunjukkan jumlah piksel tepi pada *possible circle*. Hasil akhirnya adalah nilai C sama dengan n_p . Jika nilai n_p lebih besar daripada nilai global treshold (T_g), dapat dikatakan bahwa *possible circle* ini adalah benar sebuah lingkaran (lingkaran sejati). Sebaliknya, *possible circle* bukan merupakan lingkaran dan nilai n_p kembali pada himpunan V. Perlu diperhatikan bahwa ketika sebuah lingkaran sejati terdeteksi, maka piksel tepi yang berada pada lingkaran diambil keluar dari himpunan piksel tepi (V) saat ini. Hal ini akan menyebabkan deteksi lingkaran berikutnya lebih cepat.

Teknik dengan menggunakan threshold T_g memiliki masalah dalam hal normalisasi. Lingkaran dengan jari-jari yang berbeda memiliki keliling yang berbeda. Oleh karena itu, menggunakan nilai global yang besar pada threshold T_g tidak adil bagi lingkaran dengan jari-jari yang kecil. Untuk mengatasi masalah normalisasi, maka ketika ada n_p piksel tepi terletak pada *possible circle* C_{ijk} dan nilai rasio n_p melebihi nilai $2\pi r_{ijk}$ dan threshold T_r , maka dapat dikatakan bahwa *possible circle* ini benar merupakan lingkaran. Sebaliknya, *possible circle* bukan merupakan lingkaran dan nilai n_p kembali pada himpunan V.

2.3 Algoritma Efficient Randomized

Berikut ini adalah algoritma dari Efficient Randomizes.

1. Diawali dengan menyimpan semua piksel tepi yang ada $v_i = (x_i, y_i)$ ke dalam himpunan V dan menginisialisasi counter kegagalan f menjadi 0. Ada lima macam treshold yang digunakan, yaitu T_f , T_{min} , T_a , T_d , dan T_r . T_f menunjukkan jumlah kegagalan yang dapat ditoleransi. Jika jumlah piksel dalam V kurang dari T_{min} , maka proses deteksi lingkaran akan dihentikan. Jarak antara dua piksel tepi dari sebuah lingkaran harus lebih besar dari T_a . T_d adalah treshold jarak. T_r adalah treshold rasio. $|V|$ menunjukkan jumlah piksel tepi didalam V.
2. Jika $f = T_f$ atau $|V| < T_{min}$, maka hentikan proses. Sebaliknya, ambil empat piksel v_i , $i = 1, 2, 3, 4$ secara acak dari V. Ketika v_i telah ditentukan, ubah isi $V = V - \{v_i\}$.
3. Dari keempat piksel tepi, cari *possible circle* sedemikian hingga jarak antara dua dari tiga piksel anggota lebih besar dari T_a , dan jarak antara keempat piksel dengan lingkaran lebih besar dari T_d , lanjutkan ke langkah 4. Sebaliknya, kembalikan v_i , $i = 1, 2, 3, 4$ pada himpunan V; lakukan $f = f + 1$; kembali ke Langkah 2.
4. Asumsikan C_{ijk} adalah *possible circle*. Inisialisasi nilai $C = 0$. Untuk setiap v_1 pada V, periksa apakah nilai $d_{1 \rightarrow ijk}$ tidak lebih besar dari treshold jarak T_d . Jika ya, maka lakukan $C = C + 1$ dan keluarkan v_k dari V. Setelah memeriksa semua piksel tepi dalam V, asumsikan $C = n_p$.
5. Jika $n_p \geq 2\pi r_{ijk} T_r$, lanjutkan ke Langkah 6. Sebaliknya, anggap *possible circle* ini bukan sebagai lingkaran, kembalikan n_p pada himpunan V, lakukan $f = f + 1$; kembali ke langkah 2.

6. *Possible circle* C_{ijk} telah dikenali sebagai lingkaran sejati. Inisialisasi $f = 0$ dan kembali ke langkah 2.

3. HASIL DAN KESIMPULAN

Pada awalnya, dilakukan tahap preprocessing untuk mempersiapkan gambar/citra yang akan dideteksi. Preprocessing meliputi proses Gray Scaling (untuk mendapatkan gambar gray scale), Sobel & Thinning (untuk memperoleh piksel tepi dari gambar), Hysteresis (mengubah gambar gray scale menjadi gambar biner). Hasil akhir dari preprocessing ini adalah gambar biner yang terdiri dari kumpulan piksel tepi.

Uji coba dilakukan dengan membandingkan efisiensi waktu, memori yang digunakan, serta hasil yang diperoleh dengan menggunakan Efficient Randomized dan Hough Transform. Uji coba dilakukan terhadap 100 gambar yang memiliki objek lingkaran. Detail uji coba untuk 10 percobaan dapat dilihat pada tabel 1. Pada tabel ini dicatat jumlah lingkaran seharusnya pada gambar, jumlah lingkaran yang terdeteksi, waktu yang dibutuhkan untuk mendeteksi lingkaran, serta memori yang digunakan.

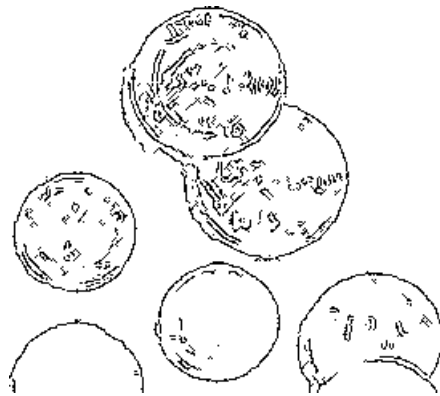
Tabel 1. Hasil Percobaan

No	Gambar / Jumlah Lingkaran	Hough Transform			Efficient Randomized		
		Deteksi	Waktu	Memory	Deteksi	Waktu	Memory
1	Taplak Meja / 3	3	4.125 s	24,808 K	3	0,234 s	11,852 K
2	Koin-1 / 7	3	6.015 s	30,192 K	4	0.296 s	5,064 K
3	Koin-2/ 7	3	5.891 s	30,384 K	10	0.265 s	5,216 K
4	Bola / 7	3	6.999 s	49,316K	9	0.2970 s	9,064 K
5	Alat Tulis / 3	1	11.781 s	54,872 K	2	0.219 s	9,432 K
6	Biskuit / 5	1	8.765 s	53,524 K	6	0.156 s	9,408 K
7	Mata / 3	3	5.985 s	31,104 K	3	0.274 s	5,456 K
8	Jam Dinding / 7	5	24.765 s	49,916 K	3	0.687 s	4.576 K
9	Icon / 7	1	15.000 s	23,496 K	4	0.500 s	8,240 K
10	Ban Mobil / 3	1	0.785 s	7,976 K	1	0.078 s	7,016 K

Hasil uji coba untuk percobaan 3 dapat dilihat pada gambar 5, 6, dan 7. Gambar 5 menunjukkan input pada percobaan 3, hasil preprocessing ditunjukkan oleh gambar 6. Gambar 7 menunjukkan hasil dari deteksi lingkaran, dimana objek yang dianggap sebagai lingkaran digambar lagi dengan menggunakan warna merah.



Gambar 5. Gambar input percobaan 3

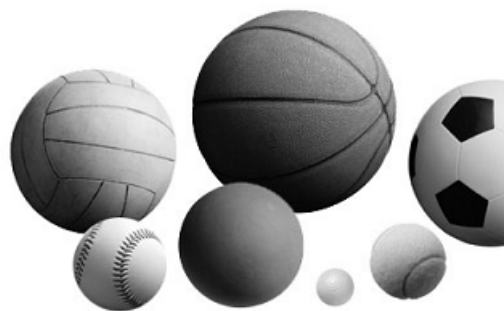


Gambar 6. Hasil preprocessing gambar 5



Gambar 7. Hasil Efficient Randomized gambar 6

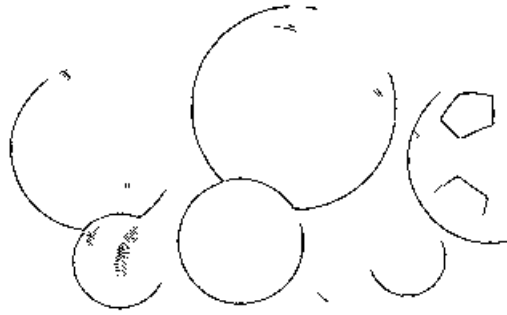
Hasil uji coba untuk percobaan 4 dapat dilihat pada gambar 8, 9, dan 10. Gambar 8 menunjukkan input pada percobaan 4, hasil preprocessing ditunjukkan oleh gambar 9. Gambar 10 menunjukkan hasil dari deteksi lingkaran, dimana objek yang dianggap sebagai lingkaran digambar lagi dengan menggunakan warna merah.



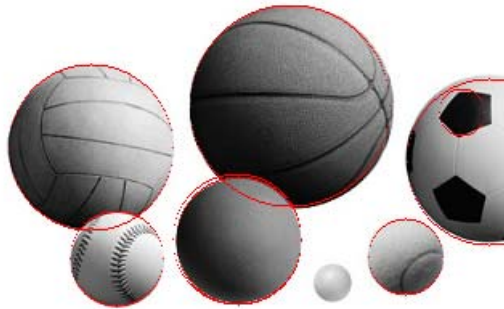
Gambar 8. Gambar input percobaan 4

Berdasarkan uji coba yang telah dilakukan, diambil kesimpulan sebagai berikut:

1. Efficient Randomized tidak membutuhkan akumulator untuk menyimpan parameter seperti pada metode Hough Transform, sehingga memori yang digunakan lebih kecil.
2. Efficient Randomized membutuhkan waktu lebih cepat untuk mendeteksi lingkaran dibandingkan dengan Hough Transform.



Gambar 9. Hasil preprocessing gambar 8



Gambar 10. Hasil Efficient Randomized gambar 9

4. DAFTAR PUSTAKA

- D. Ioannou, W. Huda, and A. F. Laine. Circle recognition through a 2D Hough transform and radius histogramming. *Image Vision Comput.*, 17. 1999.
- E. R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*. Academic Press, London. 1990.
- H. K. Yuen, J. Princen, J. Illingworth, and J. Kittler. Comparative study of Hough transform methods for circle finding. *Image Vision Comput.*, 8. 1990
- R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison Wesley, New York. 1992.
- Teh-Chuan Chen and Kuo-Liang Chung. An Efficient Randomized Algorithm For Detecting Circles. Department of Information Management, Institute of Computer Science and Information Engineering, National Taiwan University of Science and Technology. 2001.