

# PENGGUNAAN ALGORITMA TRANSFORMASI *HOUGH* UNTUK *PROTOTYPE* APLIKASI *MOBILE* PERHITUNGAN JUMLAH PIPA PADA BAGIAN GUDANG PT SPINDO Tbk

Moch. Hatta<sup>1</sup>, Moh. Mahmudi<sup>2</sup>

<sup>1</sup>Teknik Komputer, Fakultas Teknik

<sup>2</sup>Teknik Informatika, Fakultas Teknik

Universitas Maarif Hasyim Latif, Sidoarjo, Indonesia

e-mail: <sup>1</sup>moch.hatta@dosen.umaha.ac.id, <sup>2</sup>moh-mahmudi@student.umaha.ac.id

Diterima: 24 Oktober 2019. Disetujui : 26 Nopember 2019. Dipublikasikan : 31 Desember 2019



©2019 –TESJ Fakultas Teknik Universitas Maarif Hasyim Latif. Ini adalah artikel dengan akses terbuka di bawah lisensi CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>)

## ABSTRAK

Dalam perusahaan pipa baja, ujung dari pipa berbentuk lingkaran, untuk itu dibutuhkan suatu aplikasi agar dalam proses perhitungan jumlah pipa dapat lebih efektif. Dalam sistem ini dibutuhkan suatu citra *RGB* dari pengambilan ujung pipa yang terdapat objek lingkaran, kemudian diubah menjadi citra *grayscale* dan selanjutnya diikuti 4 proses lainnya yaitu : operasi morfologi dengan teknik dilasi agar objek lingkaran dapat terlihat lebih jelas, deteksi tepi menggunakan metode *canny* untuk memperbaiki detail dari citra yang blur serta mengurangi *noise*, selanjutnya deteksi lingkaran menggunakan algoritma transformasi *hough* agar objek lingkaran dalam citra dapat terdeteksi berdasarkan nilai minimal radius dan maksimal radius yang sudah diatur, dari deteksi tersebut menghasilkan sebuah *identifier* yang diinisialisasikan menjadi variabel untuk digunakan dalam proses perhitungan jumlah objek lingkaran pipa menggunakan *looping*. Hasil dari sistem ini berupa citra awal, citra hasil deteksi dan jumlah pipa yang terdeteksi. Nilai rata-rata dari deteksi objek lingkaran pipa tersebut adalah 87,36%.

**Kata kunci:** deteksi tepi, lingkaran, morfologi, pipa, transformasi *hough*

## PENDAHULUAN

Kemajuan teknologi yang pesat sekarang ini, dalam segala jenis pekerjaan yang masih manual dan kurang efisien harusnya sudah disetarakan dengan teknologi yang ada dengan sistem komputerisasi, harapannya agar segala jenis pekerjaan dapat lebih efektif. Perubahan lingkungan yang dinamis mendorong kompleksitas perusahaan untuk membuat suatu rancangan alat yang dapat menunjang pelayanan, agar perusahaan dapat tetap bersaing dan menyetarakan diri dengan revolusi teknologi dimasa kini.

Perhitungan jumlah pipa pada PT. SPINDO, Tbk masih dilakukan secara manual, khususnya para karyawan bagian gudang membutuhkan waktu yang cukup lama dalam proses perhitungan tersebut, padahal untuk melakukan perhitungan tersebut memerlukan kecepatan dan keakuratan agar tidak membuang-buang waktu.

Pada Gambar 1 terlihat petugas gudang dan petugas *security* sedang menghitung jumlah pipa secara manual pada sebuah truk peti kemas, adanya pihak *security* dalam proses menghitung

pipa tersebut adalah untuk memastikan bahwa jumlah pipa yang dikirim sesuai dengan yang dikeluarkan pihak karyawan bagian gudang dan sesuai dengan *work order* yang di pesan. Dalam sistem yang akan dibuat nanti adapun langkah penggunaan yang harus ditaati, agar penggunaan sistem dapat berjalan secara normal.



Gambar 1. Perhitungan Pipa Manual

## METODE PENELITIAN

Transformasi *Hough* merupakan suatu teknik untuk menentukan lokasi suatu bentuk dalam citra (*Hough, 1962*), P.V.C *Hough* adalah

yang mencetuskan pertama kali pada tahun 1962, oleh Rosenfeld (Rosenfeld, 1969) melihat potensi sebagai salah satu algoritma dalam pemrosesan citra, kemudian diimplementasikan oleh Duda (Duda & Hart, 1972) untuk mendeteksi garis dalam citra.

Implementasi Transformasi *Hough* menjelaskan sebuah pemetaan dari titik-titik gambar menuju ruang akumulator, pemetaan tersebut diperoleh dalam bentuk yang efisien secara matematis, berdasarkan fungsi yang menjelaskan kondisi dari target. pemetaan ini membutuhkan jauh lebih sedikit sumber perhitungan matematis dibandingkan dengan pencocokan pola. Bagaimanapun, Transformasi *Hough* masih membutuhkan penyimpanan signifikan dan perhitungan matematis tingkat tinggi. Masalah-masalah ini diselesaikan kemudian, karena mereka memfokuskan untuk pengembangan Transformasi *Hough* secara kontinu. Bagaimanapun, fakta bahwa Transformasi *Hough* ekuivalen dengan pencocokan pola telah menjadikannya sebagai salah satu dari teknik-teknik ekstraksi bentuk yang terpopuler yang ada.

**Desain Diagram Blok**

Diagram Blok pada Gambar 2 adalah diagram yang dibuat untuk mempetakan proses kerja sistem.



Gambar 2. Diagram Blok

Pada Gambar 2, citra RGB yang digunakan adalah hasil tangkapan dari kamera *smartphone* yang mempunyai resolusi 13 MP, serta pengaturan hasil citra 4:3, dengan jarak maksimal untuk pengambilan citra 75 cm. Nilai minimal radius dan nilai maksimal radius disini untuk penggunaan dalam menyeleksi objek lingkaran pipa didalam citra berdasarkan jari-jari dari tiap objek lingkaran.

**Proses**

Objek lingkaran pipa yang akan dideteksi dari citra RGB menggunakan Algoritma Transformasi *Hough* akan melewati proses-proses berikut :

- a. *Grayscale* ialah proses untuk mengubah citra RGB ke citra abu-abu.
- b. Morfologi sendiri bertujuan untuk memperbaiki hasil segmentasi, dalam proses morfologi ini, teknik dilasi yang digunakan agar citra dapat terlihat lebih jelas.
- c. Deteksi Tepi disini merupakan proses untuk menghasilkan tepi dari sebuah obyek citra, dalam proses deteksi tepi, Metode Canny disini digunakan penulis dikarenakan kemampuannya untuk mengurangi noise yang dapat dibilang paling efektif.
- d. Setelah dari proses-proses di atas, maka citra RGB sudah menjadi citra biner dan selanjutnya akan masuk proses deteksi objek lingkaran menggunakan algoritma Transformasi *Hough*, di sini objek lingkaran akan dideteksi menggunakan nilai minimal dan maksimal radius, perlu diketahui proses mendalam deteksi di sini yaitu mendeteksi objek lingkaran dari jari-jarinya menggunakan rumus fungsi dari matrik akumulator untuk melacak titik persimpangan pada ruang parameter lingkaran.
- e. Dari deteksi tersebut akan didapat *identifier* untuk proses menghitung jumlah lingkaran yang terdeteksi, adapun cara yang digunakan penulis adalah dengan teknik *looping*.

**Output**

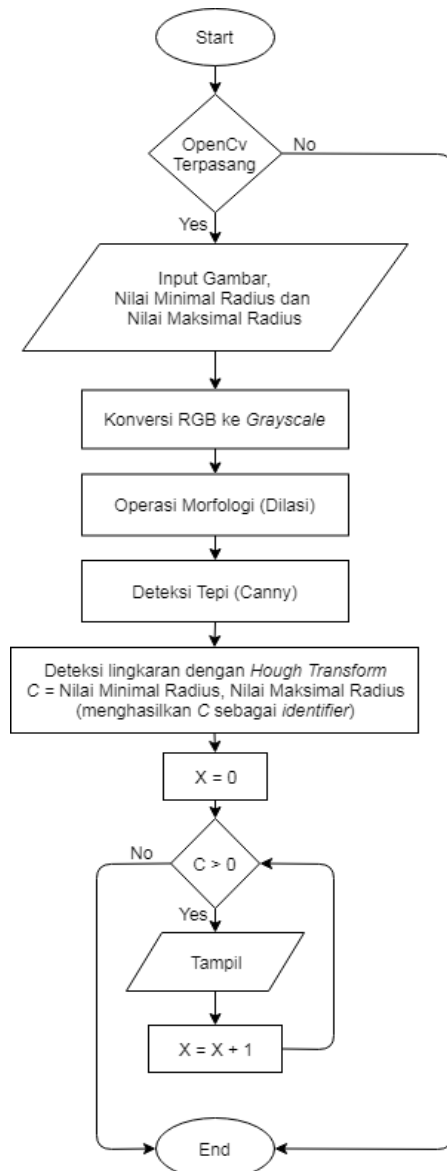
Hasil yang dikeluarkan dari sistem ini yaitu berupa citra RGB asli, citra hasil deteksi algoritma Transformasi *Hough* dan jumlah dari objek lingkaran yang terdeteksi.

**Desain Flowchart System**

Desain *flowchart system* pada Gambar 3 adalah diagram alir untuk sistem ini. Adapun Interaksi *User* dengan sistem dalam Gambar 3 diterangkan sebagai berikut :

- a. Diawali pada halaman utama yang akan disuguhkan kepada *user*.
- b. Setelah itu sistem akan memeriksa *library* OpenCV apakah sudah terpasang atau belum, kalau belum terpasang maka sistem akan keluar.
- c. Apabila OpenCV sudah terpasang, *user* dapat memasukan gambar, nilai minimal dan maksimal radius untuk deteksi objek kedalam sistem, setelah itu sistem akan memprosesnya.
- d. Sistem akan mengkonversi citra, dimana konversi citra bertujuan mengubah jenis citra dari jenis satu ke jenis lainnya, disini citra RGB yang dimasukan *User* akan dikonversi menjadi citra abu-abu (*grayscale*).
- e. Setelah menjadi citra abu-abu, selanjutnya akan masuk Operasi Morfologi dengan teknik Dilasi, agar objek didalam citra terlihat lebih jelas.
- f. Berikutnya memasuki proses deteksi tepi menggunakan metode canny, tujuan dari deteksi tepi ini adalah agar objek didalam citra dapat

dikenali. Kelebihan dari metode Canny adalah kemampuannya untuk mengurangi *noise* untuk hasil lebih jelas. *Output* dari proses deteksi tepi ini berupa citra biner.



Gambar 3. Flowchart System

- g. Setelah menjadi citra biner, selanjutnya masuk proses untuk deteksi lingkaran dengan algoritma Transformasi Hough, disini objek lingkaran pipa akan dideteksi dengan nilai minimal radius dan nilai maksimal radius, hasil dari proses deteksi tersebut akan di inisialisasi kedalam variabel *C* sebagai *identifier* untuk menghitung jumlah objek yang terdeteksi.
- h. Diproses selanjutnya, akan dilakukan pengecekan terhadap objek yang terdeteksi variabel *C* sebagai *identifire* dikondisikan  $C > 0$ , jika *Yes* maka akan menampilkan objek lingkaran pipa yang terdeteksi, jika *No* maka akan selesai.

- i. Dalam proses counter  $X = X + 1$  disini, nilai *X* yang awalnya sudah diinisialisasi 0 akan ditambah 1 dan akan dicek dengan kondisi sebelumnya  $X > 0$ , jika *Yes* maka akan ditampilkan dan diulang lagi sampai kondisi menyatakan *No* baru proses akan selesai.

### HASIL DAN PEMBAHASAN

Prototype Aplikasi Perhitungan Jumlah Pipa diimplementasikan pada *Smartphone* Android, dengan menggunakan *Tools* Android Studio versi 3.3 dan *Library* dari OpenCV versi 3.4.1. Berdasarkan perancangan sistem yang telah dibuat.

#### Implementasi Program

Implementasi program yaitu menguji coba prototype aplikasi yang telah dibuat. Secara garis besar Aplikasi yang berjalan di *smartphone* Android. Adapun implementasinya sebagai berikut:

- a. Pertama *Smartphone* harus di Install Aplikasi *OpenCVManager\_3.4.3.apk* terlebih dahulu sebagai *Library*, seperti yang sudah terpasang pada Gambar 4.



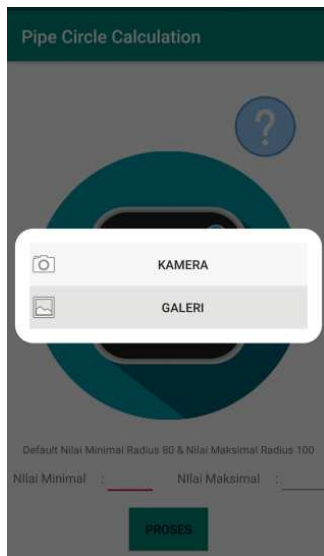
Gambar 4. Tampilan Library OpenCV

- b. Setelah itu *Smartphone* juga harus di Install Prototype Aplikasi *PipeCircleCalculation.apk*. Prototype aplikasi yang sudah berjalan terlihat pada Gambar 5.



Gambar 5. Tampilan Menu Utama

- c. Jika prototype aplikasi sudah dalam posisi yang terlihat pada Gambar 5 maka pengguna dapat memilih *icon* kamera yang nantinya akan menampilkan menu kamera dan galeri seperti Gambar 6.



Gambar 6. Tampilan Menu untuk Upload

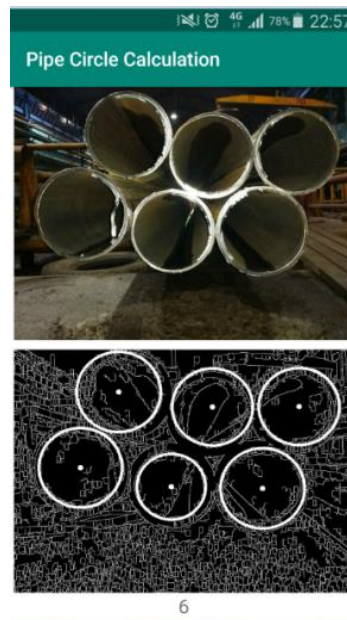
- d. Setelah menu tersebut tampil, pilih salah satu pilihan untuk memasukan gambar yang akan di unggah, dan setelah gambar yang berhasil diunggah seperti contoh pada Gambar 7.



Gambar 7. Tampilan Gambar ter Upload

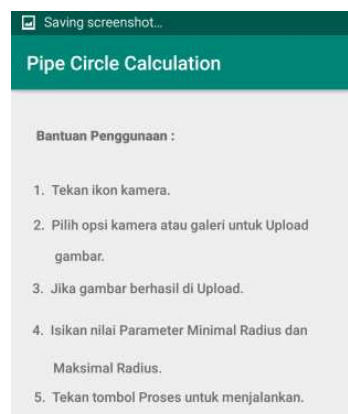
- e. Selanjutnya masukan nilai minimal radius dan maksimal radius, untuk nilai *default* atau nilai optimal yaitu 80px untuk minimal radius dan 100px untuk maksimal radius, setelah itu tekan tombol Proses untuk menjalankan prototype aplikasi melakukan proses deteksi jumlah lingkaran dari gambar tersebut, setelah itu tunggu sampai aplikasi selesai menjalankan

proses dan mengeluarkan hasil seperti pada Gambar 8.



Gambar 8. Tampilan Hasil

- f. Pada Gambar 8 menampilkan hasil akhir, dimana pada tampilan bagian atas masih menampilkan gambar awal sebelum diproses dan bagian bawahnya menampilkan gambar hasil proses deteksi lingkaran, sedangkan bagian bawah sendiri menunjukkan angka dari hasil jumlah lingkaran yang terdeteksi. Jika masih bingung mengenai penggunaan prototype aplikasi tersebut, lihat pada tampilan Gambar 5 pilih *Icon* tanda tanya untuk bantuan penggunaan aplikasi dan akan tampil seperti pada Gambar 9.



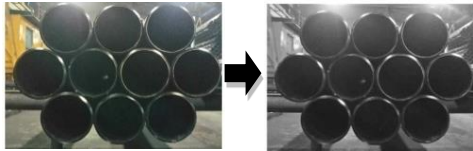
Gambar 9. Tampilan Bantuan

### Implementasi Fungsi

Dalam tahap ini akan dilakukan 4 proses, yaitu *grayscale*, morfologi, deteksi tepi, dan Transformasi Hough.

#### a. *Grayscale*

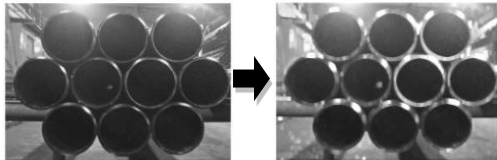
Proses ini merupakan proses konversi citra RGB menjadi citra *grayscale*. Hasilnya dapat dilihat pada Gambar 10.



Gambar 10. Konversi citra RGB menjadi grayscale

**b. Morfologi**

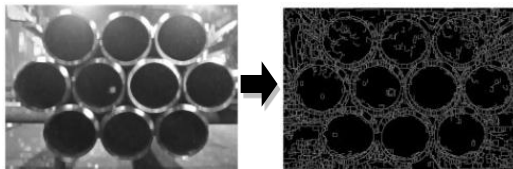
Dalam proses ini dilakukan teknik dilasi untuk memperbesar segmen objek dari citra grayscale. Hasilnya dapat dilihat pada Gambar 11.



Gambar 11. Konversi citra grayscale menjadi citra morfologi

**c. Deteksi Tepi**

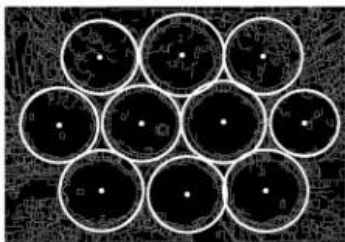
Proses ini dilakukan pendeteksian tepi pada citra hasil morfologi dengan metode Canny. Hasilnya dapat dilihat pada Gambar 12.



Gambar 12. Konversi citra hasil morfologi menjadi citra deteksi tepi

**d. Transformasi Hough**

Proses ini untuk mendeteksi objek lingkaran dalam citra. Hasilnya dapat dilihat pada Gambar 13.



Gambar 13. Citra hasil deteksi Transformasi Hough

**Identifikasi Hasil Pengujian**

Pengujian di lapangan dilakukan menggunakan nilai minimal radius 80 px dan nilai maksimal radius 100 px, objek lingkaran penuh yang terdapat dalam citra RGB dapat terdeteksi dengan cukup baik oleh algoritma Transformasi Hough, pengujian dilakukan dengan ukuran pipa mulai dari 3", 4", 6", 7", dan 8", masing-masing pengujian dilakukan lebih dari 2 kali sampai 9 kali pengambilan citra.

Akurasi dalam pengujian ini diambil rata-rata dari pengujian semua sampel di atas dengan persamaan:

$$Akurasi = \frac{terdeteksi}{jumlah\ objek} \times 100\% \quad (1)$$

Dari persamaan 1 dapat dihitung persentase keberhasilan tiap ukuran pipa, setelah itu dapat dihitung total rata-rata persentase keberhasilan yang dapat dilihat pada Tabel 1.

Tabel 1. Rata-rata persentase hasil 5 pengujian

Ukuran	Akurasi
3"	47,82%
4"	100%
6"	98,88%
7"	92,30%
8"	97,82%
<b>Total rata-rata persentase Keberhasilan</b>	<b>87,36%</b>

**PENUTUP**

Pada uji coba prototype aplikasi perhitungan jumlah pipa di lapangan dengan total uji coba sampel meliputi 5 ukuran pipa, nilai minimal radius default yakni 80 px dan nilai maksimal radius 100 px, dengan citra yang berbeda dan uji test berbeda pula tiap ukuran, yang meliputi ukuran 3" mendapatkan akurasi 47,82% dari 2 kali uji test, ukuran 4" akurasi 100% dari 8 kali uji test, ukuran 6" 98,88% dari 9 uji test, ukuran 7" mendapat akurasi 92,30% dari 3 uji coba, dan ukuran 8" mendapatkan akurasi 97,82% dari 8 kali uji sampel.

Berdasarkan hasil pembahasan dan uji coba sistem dapat disimpulkan bahwa sistem dapat mendeteksi lingkaran pipa dengan menggunakan Algoritma Transformasi Hough dengan baik, sehingga metode ini layak untuk dikembangkan lebih lanjut. Penggunaan smartphone untuk menangkap gambar sangat berpengaruh terhadap hasil yang akan diperoleh dengan menggunakan Algoritma Transformasi Hough. Prototype Aplikasi Perhitungan Jumlah Pipa menggunakan Algoritma Transformasi Hough dapat digunakan pada saat pipa akan dimuat ke dalam bak truk ataupun peti kemas dengan langkah mengambil gambar per packing.

Penggunaan Algoritma Transformasi Hough sangat dipengaruhi oleh citra hasil tangkapan. Oleh karena itu dibutuhkan hasil citra dengan latar belakang seragam, agar hasil deteksi objek lingkaran dapat lebih jelas. Perlu diadakan prapemrosesan citra karena penangkapan citra menggunakan media kamera smartphone sangat rentan terhadap perbedaan kualitas, terlebih dalam intensitas cahaya yang kurang ataupun tidak merata yang mengakibatkan pipa membayang juga objek yang tidak presisi yang berakibat tidak mendapatkan pencahayaan. Adapun syarat lain agar objek lingkaran terdeteksi yaitu parameter

*mindist* yang merupakan jarak antar titik pusat objek yang terdeteksi yang juga berpengaruh ketika pengambilan citra melebihi sudut  $30^{\circ}$  terhadap hasil objek yang nantinya tidak berbentuk bulat dan yang perlu diperhatikan disini, *mindist* bisa dibuat kondisional sesuai dengan kondisi lapangan dalam penelitian, semisal objek lingkaran yang dideteksi tidak terlalu banyak alangkah lebih baik nilai *mindist* disini dapat diatur lebih besar agar optimal untuk deteksi.

#### DAFTAR PUSTAKA

- Doavers Development Team. (2018). Macam-macam Metode Deteksi Tepi (Edge Detection Methods). Retrieved July 17, 2019, from <https://www.doavers.com/blog/macam-macam-metode-deteksi-tepi-edge-detection-methods>
- Duda, R. O., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), 11-15.
- Gonzales, R., & Woods, R. (1992). *Digital image processing*. USA: Addison-Wesley Publishing Company.
- Hough, P. V. C. (1962, December 18). *Method and means for recognizing complex patterns*. Google Patents.
- Munir, R. (2004). *Pengolahan citra digital dengan pendekatan algoritmik*. Bandung: Informatika.
- Pernanda, A. Y. (2017). Operasi Morfologi pada Pengolahan Citra | DevTriK. Retrieved July 17, 2019, from <https://devtrik.com/opencv/operasi-morfologi-pada-pengolahan-citra/>
- Prasetyo, E. (2011). *Pengolahan citra digital dan aplikasinya menggunakan MATLAB*. Yogyakarta: Penerbit ANDI.
- Rinanto, L., Sugiharto, A., & Indriyati, I. (2014). Aplikasi Pendeteksi Objek Lingkaran pada Citra dengan Transformasi Hough. *Journal of Informatics and Technology*, 2(4), 1-9.
- Rosenfeld, A. (1969). Picture processing by computer. *ACM Computing Surveys (CSUR)*, 1(3), 147-176.
- Santoso, A., Uliontang, U., Arif, I., & Hatta, M. (2017). Deteksi Objek Senjata Tajam Pada Citra X-ray Dengan Metode Pengukuran Dimensi Citra. *Teknika: Engineering and Sains Journal*, 1(1), 1-10.
- Yagi, M. (2012). Pengolahan Citra (Tugas 4) | mochamadyagi. Retrieved July 17, 2019, from <https://mochamadyagi.wordpress.com/2012/04/17/pengolahan-citra-tugas-4/>