

PERANCANGAN PROTIPE PENGAMBAR POLA BATIK ROBOT KARTESIAN 2 DOF METODE PENGURUTAN DATA KOORDINAT JARAK EUCLIDEAN BERBASIS ARDUINO UNO

Solichul Huda^{*)}, Sumardi, Budi Setiyono

Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)E-mail: umyhugi94@gmail.com}

Abstrak

Teknologi yang terus maju mendorong manusia untuk meningkatkan produktivitas dengan mengubah sistem manual ke sistem otomatis. Modernisasi merambah hampir ke semua aspek tak terkecuali bidang kesenian. Batik adalah warisan leluhur nusantara yang telah diakui oleh UNESCO sebagai warisan budaya milik Indonesia namun industri batik di Indonesia belum begitu berkembang. Modernisasi industri di bidang seni batik diharapkan mampu mengoptimalkan produktivitas batik. Penelitian ini merancang algoritma dan program pada MATLAB yang digunakan untuk mengatur gerak prototipe pembatik robot kartesian dengan 2 derajat kebebasan (2 DOF). Data masukan berupa gambar yang kemudian diolah untuk didapatkan urutan koordinat tepinya. Urutan koordinat diproses dan dikirim pada plant dengan menggunakan software MATLAB berbasis Graphical User Interface (GUI). Robot Kartesian dengan 2 derajat kebebasan (DOF) dan sebuah pena tulis (end effector) yang dipasang pada ujung lengannya untuk menggambar pola batik. Sebuah servo dipasang untuk menggerakkan end effector, dan dua buah stepper yang terkopel dengan ulir digunakan untuk mengarahkan pena tulis pada koordinat X-Y yang telah tersusun. Pengujian dilakukan dengan mengirimkan data urutan koordinat pada plant atau dengan melihat urutan proses penggambaran pada grafik. Error yang didapatkan pada pengujian program dan algoritma adalah 2,32%

Kata Kunci : Batik, Matlab, Graphical User Interface (GUI)

Abstract

The development of technology pushes human to increase productivity by changing manual system to automatic system. Modernization affect almost all aspect especially in art. Batik is a ancestor legacy of indonesian people that recogized by UNESCO as culture legacy of Indonesia yet batik industries are not develop as much. Modernization in batik art is expected to optimize its productivity. This research design algorithm and the program in MATLAB that used for controlling movement protype of batik maker 2 dof cartesian robot. The input data is a picture that will be processed to obtain the order of coordinate edge. Then it procesed and sent to plantby using MATLAB based on Graphical User Interface (GUI). The 2 dof Cartesian robot and a pen (end effector) is used to draw a pattern. A servo motor is installed to activate end effector and two stepper motor coupled with thread mechanic that used to move it towards the X and Y axis. Testing is done by sending coordinat orders to the plant or by observing drawing process order. Obtained error from program and algorithm testing is 2,32%.

Keyword : Batik, Matlab, GUI

1. Pendahuluan

Indonesia merupakan negara yang kaya dengan karya seni dan budaya. Indonesia memiliki banyak keanekaragaman di dalam kesenian khususnya seni lukis dan batik. Batik adalah teknik perintang warna dengan menggunakan malam, yang telah ada sejak pertama kali diperkenalkan dengan nama batex oleh Chastelein, seorang anggota Raad van Indie (Dewan Hindia) pada tahun 1705 [1]. Hampir setiap daerah di Indonesia memiliki motif batik

yang khas sesuai karakteristik daerahnya masing-masing mulai dari batik Cirebon, Pekalongan, Solo, Yogyakarta sampai ke batik Ubud di daerah timur Indonesia. Oleh sebab itu bangsa dunia mengakui bahwa batik merupakan aset asli budaya Indonesia. Batik Indonesia adalah salah satu warisan budaya dunia (*World Heritage*) yang telah diakui dan disahkan oleh *United Nations Educational, Scientific, and Culture Organization* (UNESCO) pada tanggal 2 Oktober 2009 [2].

Perkembangan teknologi di dunia yang kian berkembang berdampak pada kemajuan di berbagai bidang tak terkecuali perkembangan di dalam dunia robotika dan industri. Dengan teknologi yang terus maju mendorong manusia untuk meningkatkan produktivitas khususnya para pelaku industri mulai mengganti sistem manual ke sistem otomatisasi [3]. Dunia industri erat kaitannya dengan dunia robotika hampir setiap industri menggunakan aplikasi robotik dalam menjalankan aktifitas produksinya. Diantara jenis robot yang banyak digunakan dalam produktivitas industri adalah robot kartesian. Robot kartesian di dalam perancangannya memiliki struktur yang terdiri dari tiga sumbu linier (prismatic). Masing-masing sumbu dapat bergerak ke arah sumbu X-Y-Z sehingga memungkinkan untuk mendapatkan hasil lebih akurat dalam menggambar pola batik [4].

Namun modernisasi industri seni di Indonesia belum begitu marak. Modernisasi dalam bidang kesenian khususnya seni rupa erat kaitannya dengan kualitas pengolahan citra yang memungkinkan aplikasi yang lebih luas di berbagai bidang teknologi tinggi [5]. *Graphical User Interface* (GUI) selain digunakan sebagai media interaksi antara pengguna dan *plant* juga memungkinkan di dalamnya untuk ditanamkan program pengolahan citra.

Pada penelitian sebelumnya telah dibahas mengenai sebuah robot yang dirancang untuk memotong styrofoam secara otomatis dengan dua derajat kebebasan. Pada penelitian tersebut *plant* berjalan berdasarkan algoritma kode rantai (*chain code*) program yang dirancang menggunakan perangkat lunak (*software*) *Borland Delphi 7*. Dengan mengembangkan konsep penelitian sebelumnya, pada penelitian ini akan dibahas mengenai perancangan algoritma yang membangun bahasa program dengan basis *Graphical User Interface* (GUI) *MATLAB* dan berfungsi menghasilkan *output* data koordinat terurut. *Output* tersebut akan diolah di dalam mikrokontroler *Arduino Uno* dan akan menggerakkan *plant* berupa *prototype* robot kartesian dalam menggambar pola batik. Diharapkan dari penelitian ini akan didapatkan algoritma program di dalam *MATLAB* yang memungkinkan *plant* menggambar motif dan pola batik lebih akurat.

2. Metode

2.1. Segmentasi Citra

Segmentasi merupakan tahapan pada proses analisis citra yang bertujuan untuk memperoleh informasi yang ada dalam citra tersebut dengan membagi ke dalam daerah - daerah terpisah dimana setiap daerah adalah homogen dan mengacu pada sebuah kriteria keseragaman yang jelas dan harus tepat agar informasi yang terkandung didalamnya dapat diterjemahkan dengan baik. Segmentasi harus diberhentikan apabila masing-masing obyek telah terisolasi atau terlihat dengan jelas. Tingkat keakurasiannya tergantung pada tingkat keberhasilan

prosedur analisis yang dilakukan dan diharapkan proses segmentasi memiliki tingkat keakuratan yang tinggi [6].

2.2. Deteksi Tepi Canny

Pendeteksian tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi di dalam citra. Tujuan pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Yang dimaksudkan dengan tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat.

Metode *canny* merupakan salah satu algoritma deteksi tepi. Deteksi tepi *canny* ditemukan oleh Marr dan Hildreth yang meneliti pemodelan persepsi visual manusia. Dalam memodelkan pendeteksian tepi, dia menggunakan ideal *step edge*, yang direpresentasikan dengan fungsi *Sign* satu dimensi. Pendekatan algoritma *canny* dilakukan dengan konvolusi fungsi *image* dengan operator *gaussian* dan turunan- turunannya [7]. Ada beberapa kriteria pendeteksian tepian paling optimum yang dapat dipenuhi oleh algoritma *Canny* [8]:

1. Mendeteksi dengan baik (kriteria deteksi) Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan.
2. Melokalisasi dengan baik (kriteria lokalisasi) Dengan *Canny* dimungkinkan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.
3. Respon yang jelas (kriteria respon) Hanya ada satu respon untuk tiap tepi. Sehingga mudah dideteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya.

Langkah-langkah dalam melakukan deteksi tepi *canny* [9]:

1. Menghilangkan derau yang ada pada citra dengan mengimplementasikan tapis *Gaussian*. Bila tidak dilakukan maka garis-garis halus juga akan dideteksi sebagai tepian. Salah satu contoh tapis *Gaussian* dengan $\sigma = 1.4$ dapat dilihat pada persamaan 1.

$$\frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (1)$$

2. Melakukan penguatan tepi dengan salah satu operator deteksi tepi seperti *Roberts*, *Prewitt*, atau *Sobel* [28] dengan melakukan pencarian secara horisontal (G_x) dan secara vertikal (G_y) melalui persamaan 2.

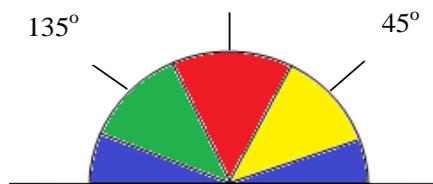
$$|G| = |G_x| + |G_y| \quad (2)$$

3. Menentukan arah tepian yang ditemukan dengan menggunakan persamaan 3.

$$\theta = \arctan(G_y, G_x) \quad (3)$$

Selanjutnya kedalam 4 warna sehingga garis dengan arah berbeda dan memiliki warna yang berbeda. Pembagiannya adalah 0 – 22,5 dan 157,5 – 180 derajat berwarna kuning, 22,5 – 67,5 berwarna hijau, dan derajat 67,5 – 157,5 berwarna merah. Pembagian arah berdasarkan arah tepian dapat dilihat pada Gambar 1.

- Setelah arah tepi diperoleh, penghilangan non-maksimum dilaksanakan. Penghilangan non-maksimum dilakukan di sepanjang tepi pada arah tepi dan menghilangkan piksel-piksel (piksel diatur menjadi 0) yang tidak dianggap sebagai tepi. Dengan cara seperti itu, diperoleh tepi yang tipis.



Gambar 1. Area untuk mengonversi arah tepi

- Langkah terakhir berupa proses yang disebut *hysteresis*. Proses ini menghilangkan garis-garis yang seperti terputus-putus pada tepi objek. Caranya adalah dengan menggunakan dua ambang T_1 dan T_2 . Lalu, semua piksel citra yang bernilai lebih besar daripada T_1 dianggap sebagai piksel tepi. Selanjutnya, semua piksel yang terhubung dengan piksel tersebut dan memiliki nilai lebih besar dari T_2 juga dianggap sebagai piksel tepi.

2.3. Jarak Euclid

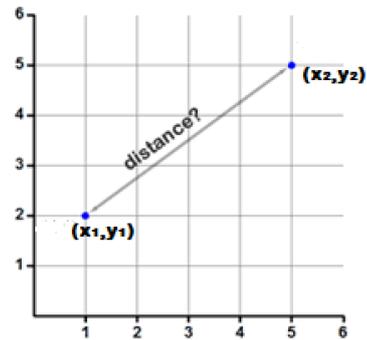
Jarak *euclid* merupakan tipe pengukuran jarak dalam analisis *cluster* yang paling umum digunakan untuk mengukur jarak dari obyek data ke pusat *cluster*. Jarak *euclid* merupakan jarak geometris antardua obyek data. Semakin dekat jarak maka semakin mirip suatu obyek data tersebut.

Dalam bentuk umum jarak *euclid* (d) dapat diperoleh dengan:

$$x = (x_1, x_2, \dots, x_n) \quad (4)$$

$$y = (y_1, y_2, \dots, y_n) \quad (5)$$

Pada dua dimensi, misalkan ada 2 titik koordinat (x_1, x_2) dan (y_1, y_2) seperti pada gambar berikut



Gambar 2. Koordinat kartesian

Maka jarak *euclid* (d) diantara dua titik tersebut adalah

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (6)$$

Atau bisa dibentuk juga persamaan bahwa [10]

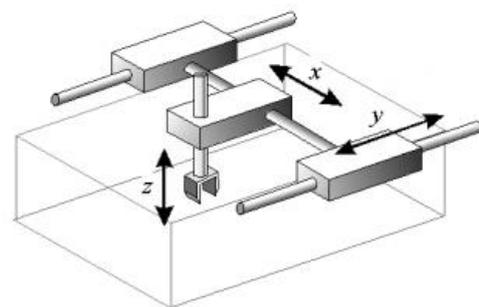
$$d_e = \sum_{k=1}^m \sqrt{\sum_{k=1}^m (fd_{i,k} - k_j)^2} \quad (7)$$

Keterangan :

- d_e = jarak euclidean
- fd_i = data buah training
- k_j = data buah uji
- m = jumlah data pelatihan

2.4. Robot Kartesian

Manipulator dengan konfigurasi kartesian ditunjukkan pada Gambar 3. Konfigurasi ini secara relatif adalah yang paling kokoh untuk tugas mengangkat beban yang berat. Struktur ini banyak dipakai secara permanen pada instalasi pabrik, baik untuk mengangkat dan memindah barang produksi maupun untuk mengangkat peralatan-peralatan berat pabrik ketika melakukan kegiatan instalasi. crane di galangan kapal juga banyak mengadopsi struktur ini.



Gambar 3. Model kerangka robot kartesian

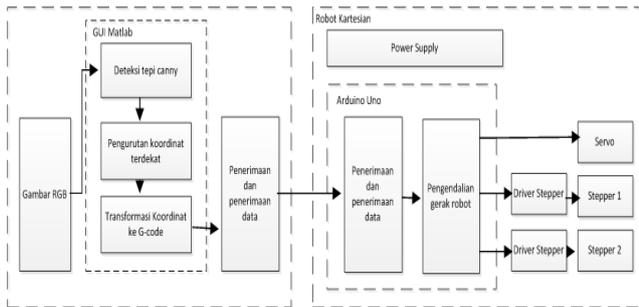
Pada aplikasi yang sesungguhnya, biasanya struktur penyangga, badan dan lengan dibuat sedemikian rupa hingga tumpuan beban merata pada struktur. Misalnya, penyanggah dipasang dari ujung ke ujung. Mekanik pengangkat di badan menggunakan sistem rantai dan

sprocket atau sistem belt. Pergerakan lengan dapat menggunakan sistem seperti rel di kiri-kanan lengan [11].

2.5. Perancangan Sistem

2.5.1. Gambaran umum system

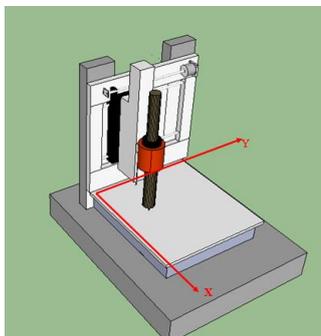
Konsep dasar perancangan penelitian ini dapat dijelaskan secara singkat pada gambar dibawah ini :



Gambar 4. Konsep dasar perancangan

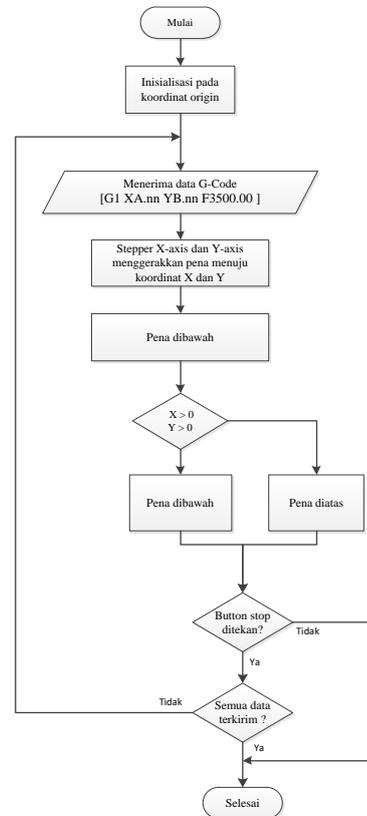
Kerja sistem dimulai dari input awal yaitu gambar RGB. Kemudian gambar tersebut diolah pada beberapa pemrosesan gambar diantaranya proses pemilihan gambar (uploading), perubahan ukuran gambar (resizing), pendeteksian tepi canny, kemudian pemrosesan pada program transformasi koordinat hingga didapatkan perintah-perintah yang akan diubah kedalam bentuk G-code untuk kemudian dikirim ke plant dengan menggunakan komunikasi serial. Prototipe pembuat pola batik dibangun untuk melakukan pembuktian kebenaran program yang telah dirancang tersebut. Adapun Arduino Uno digunakan untuk mengolah dan menterjemahkan kode-kode perintah dari perangkat lunak PC yang masih berupa data serial.

2.5.2. Perancangan Hardware



Gambar 5. Arah gerak sumbu prototipe

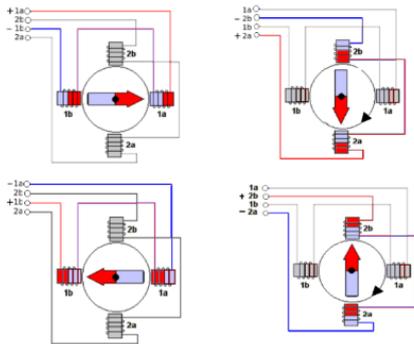
Adapun penjelasan gerak sistem ditunjukkan dengan flowchart berikut :



Gambar 6. Flowchart gerak sistem secara umum

Dari flowchart diatas dapat dijelaskan bahwa gerak robot dimulai dari inisialisasi sistem. Kemudian Arduino Uno akan mengeksekusi data G-code menjadi gerakan sehingga mampu menggerakkan pena ke koordinat tujuan. Inputan sistem pada Arduino Uno adalah data G-code, masing-masing nilai X dan Y pada G-code yang akan diolah dan dihitung untuk menggerakkan stepper X-axis dan stepper-Y axis. Apabila koordinat X dan Y memiliki nilai atau salah satunya memiliki nilai, maka ini adalah koordinat tujuan dan pena turun sedangkan apabila X dan Y semuanya bernilai 0 maka dengan data tersebut memberikan perintah pena terangkat.

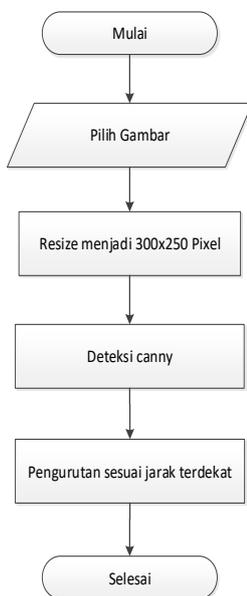
Motor stepper yang digunakan pada sumbu X dan Y adalah motor stepper bipolar sedangkan metode geraknya adalah menggunakan teknik full drive step. Dengan menggunakan metode ini maka hanya 1 coil yang diaktifkan pada setiap waktu. Dan sequence dari stepper motor dapat diketahui dengan melakukan pengecekan common grounds, yaitu dengan menggunakan multimeter mode buzzer hingga menemukan pasangan kabel yang short (berupa coil).



Gambar 7. Common ground pada stepper

2.5.3. Pengurutan Koordinat pada GUI Matlab

Perancangan ini dibuat dengan tujuan untuk mendapatkan urutan koordinat yang akurat pada tepian pola suatu gambar sehingga dengan urutan koordinat tersebut, *plant* dapat menggambar pola gambar sesuai urutan koordinat yang telah dibuat. Pemrograman ini dibangun di dalam aplikasi MATLAB dengan menggunakan prinsip jarak *euclidean* (*euclidean distance*). Adapun alur pengolahan gambar menjadi koordinat terurut pada kerja GUI MATLAB dapat dipahami dengan melihat diagram alir berikut ini :



Gambar 8. Flowchart algoritma transformasi koordinat

Pada gambar diagram alir diatas dapat dilihat skema kerja sistem dari awal sampai akhir proses. Skema diatas dapat dirinci menjadi beberapa sub-sub bab pembahasan diantaranya proses pendeteksian tepi, proses pengurutan koordinat, dan proses menerjemahkan urutan koordinat kedalam kode *G-code*.

3. Hasil dan Analisis

Pengujian yang dilakukan yaitu, menguji program pengurutan data, dan menguji program luaran matlab kepada prototype dengan data inputan awal bangun dasar persegi segitiga dan lingkaran serta pola motif batik bunga

3.1. Pengujian Program Pengurutan Data

Tabel 1. data koordinat titik awal dan titik akhir masing-masing label

Label	Titik awal		Titik akhir	
	X (A1)	Y(B1)	X(A2)	Y(B2)
1	26	13	26	14
2	183	14	182	15
3	214	43	213	44
4	193	78	192	79
5	195	119	195	120
6	108	156	107	157
7	16	166	15	167
8	177	169	176	170
9	86	202	86	203

Sehingga dengan data tersebut akan didapatkan urutan dalam bentuk tabel sebagai berikut :

Tabel 2. Urutan label berdasarkan jarak terdekat

Urutan	Label
1	1
2	7
3	9
4	6
5	8
6	5
7	4
8	3
9	2

3.2. Pengujian Transformasi Koordinat G-Code

Pengujian ini adalah untuk mengetahui keberhasilan program matlab yang telah dibuat dalam mentransformasikan koordinat ke dalam bentuk kode-kode *G-code*. Program yang telah dimasukkan kedalam *Arduino Uno* adalah program yang akan membaca kode *G-Code* maka data proses yang akan dikirimkan ke *Arduino Uno* harus dalam bentuk kode *G-Code*.

Tabel 3. Transformasi koordinat asal menjadi G-code

Data ke	Koordinat asal		Transformasi koordinat ke		Keterangan
	X	Y	G-Code dengan faktor pembagi X= 10.5, Y=7		
1	26	13	G1 X2.48 Y1.86 F3500.00		Sesuai
2	27	13	G1 X2.57 Y1.86 F3500.00		Sesuai
3	28	13	G1 X2.67 Y1.86 F3500.00		Sesuai
4	29	13	G1 X2.76 Y1.86 F3500.00		Sesuai
5	30	13	G1 X2.95 Y1.86 F3500.00		Sesuai
6	31	13	G1 X3.05 Y1.86 F3500.00		Sesuai

Tabel 3. Tabel lanjutan

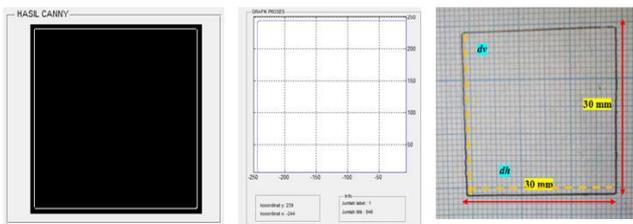
Data ke	Koordinat asal		Transformasi koordinat ke	Keterangan
	X	Y	G-Code dengan faktor pembagi X= 10.5, Y=7	
7	32	13	G1 X3.14 Y1.86 F3500.00	Sesuai
8	33	13	G1 X3.24 Y1.86 F3500.00	Sesuai
9	34	13	G1 X3.33 Y1.86 F3500.00	Sesuai
10	35	13	G1 X3.43 Y1.86 F3500.00	Sesuai

Dari tabel diatas adalah data yang diperoleh dari *running* program dalam matlab. Dapat disimpulkan bahwa program yang dibuat dalam matlab untuk mengubah data koordinat menjadi kode-kode G-code berhasil.

3.3. Pengujian Bangun Dasar Sederhana

Berikut ini adalah gambar bangun dasar sederhana yang diujikan

3.3.1. Persegi

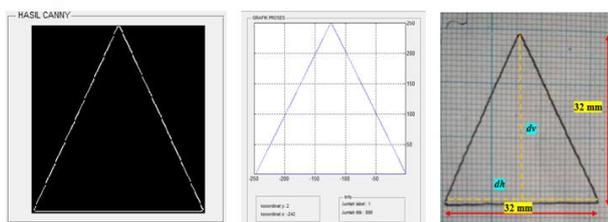


Gambar 9. Pengujian bangun persegi

Tabel 4. Pengujian bangun persegi

Diameter	Koordinat awal	Koordinat akhir	Diameter gambar digital (Satuan Panjang)	Diameter gambar plant	Ket
dv	(6, 243)	(6,6)	237	30 mm	Sesuai
dh	(6,243)	(243, 243)	237	30 mm	Sesuai

3.3.2. Segitiga

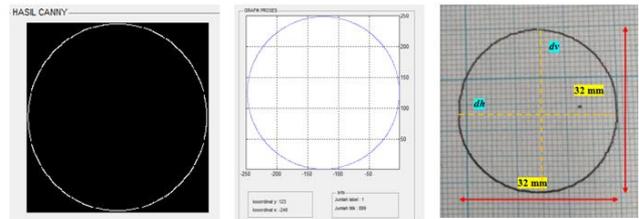


Gambar 10. Pengujian bangun segitiga

Tabel 5. Pengujian bangun segitiga

Diameter	Koordinat awal	Koordinat akhir	Diameter gambar digital (Satuan Panjang)	Diameter gambar plant	Ket
dv	(126,248)	(126,2)	246	32 mm	Sesuai
dh	(3,248)	(249, 248)	246	32 mm	Sesuai

3.3.3. Lingkaran

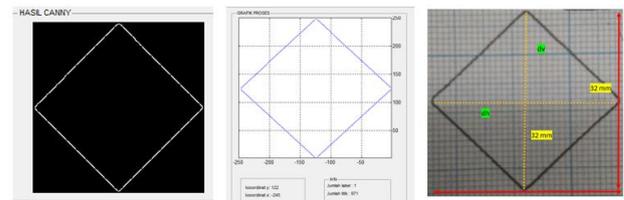


Gambar 11. Pola dasar lingkaran

Tabel 6. Pengujian bangun lingkaran

Diameter	Koordinat awal	Koordinat akhir	Diameter gambar digital (Satuan Panjang)	Diameter gambar plant	Ket
dv	(2, 125)	(248,125)	246	32 mm	Sesuai
dh	(124,3)	(124, 249)	246	32 mm	Sesuai

3.3.4. Belah Ketupat



Gambar 12. Pengujian bangun persegi

Tabel 7. Pengujian bangun belah ketupat

Diameter	Koordinat awal	Koordinat akhir	Diameter gambar digital (Satuan Panjang)	Diameter gambar plant	Ket
Dv	(126,249)	(126,3)	246	32 mm	Sesuai
Dh	(2,126)	(248,126)	246	32 mm	Sesuai

Dari pengujian bangun dasar sederhana diatas yang meliputi pengujian bangun persegi, lingkaran, segitiga, belah ketupat didapatkan hasil dimesni gambar *plant* yang cukup simetri yaitu dengan nilai dimensi maksimal 32 mm x 32 mm, sedangkan ukuran ideal dimensi apabila menggunakan faktor pembagi yang telah ditentukan

adalah 33 mm x 33 mm. Dengan data tersebut, dapat dihitung *error* pada sistem robot adalah

$$Error (\%) = \left| \frac{nilai_aktual - nilai_ideal}{nilai_ideal} \right| \times 100 \%$$

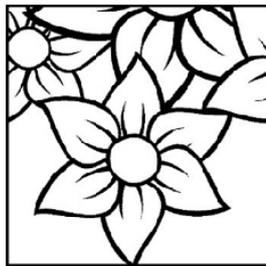
$$Error (\%) = \left| \frac{32 - 33}{33} \right| \times 100 \%$$

$$Error (\%) = 3,03 \%$$

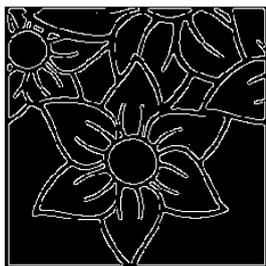
Maka dapat disimpulkan *error* dimensi sistem adalah 3,03 %

3.3.5. Pola Batik Motif Bunga

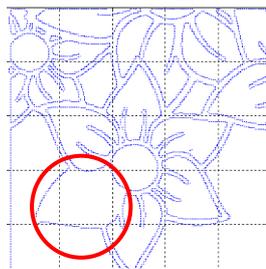
Pengujian pola batik dilakukan dengan mengupload pola batik yang berkekestensi file .JPG utuk kemudain diproses sebagaimana yang prinsip kerja program yang telah dibangun.



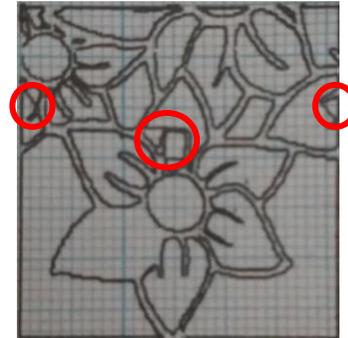
Gambar 13. Gambar asli yang sudah dirubah ukurannya menjadi 250 x 250 pixel.



Gambar 14. Gambar deteksi tepi hasil dari pengolahan gambar asli



Gambar 15. Hasil pemrosesan algoritma pengurutan koordinat dengan masukan gambar *cann*



Gambar 16. Hasil penggambaran oleh *plant*

Dengan menganalisis Gambar 13 , Gambar 14, Gambar 15, Gambar 16 dapat disimpulkan bahwa deteksi tepi yang merupakan algoritma bawaan Matlab mampu mengubah gambar RGB menjadi gambar tepian sebagaimana tertera pada Gambar 14. Dan gambar *canny* tersebut merupakan inputan untuk algoritma pengurutan koordinat menggunakan metode *euclidean*. Pada gambar terdapat bagian yang tidak terdeteksi sebagai label dan juga ada garis yang terputus-putus. Hal ini dikarenakan kekurangan kondisi dalam pembuatan program sehingga dalam kondisi tertentu algoritma program yang dibuat tidak mampu mendeteksi label. Pada kondisi ideal, gambar akan sama hasilnya dengan Gambar 15. Namun pada Gambar 16. terdapat beberapa garis lurus yang bukan merupakan label juga ikut tergambar. Hal ini dikarenakan kurangnya kondisi program mikrokontroler dan pemasangan mekanik yang kurang stabil.

Pada percobaan penggambaran beberapa beberapa pola batik yang lain dengan jumlah label dan titik yang banyak akan menghasilkan *error* berupa penggambaran beberapa garis lurus saat perpindahan dikarenakan pena tidak terangkat pada saat perpindahan titik satu ke titik yang lainnya. Terutama apabila gambar asli yang dimasukkan ke program adalah gambar yang memiliki nilai *pixel* kecil akan menghasilkan *error* berupa penggambaran beberapa garis lurus yang lebih banyak. Adapun untuk percobaan gambar pola batik bunga diatas, ada 1 dari 43 label yang tidak tergambar. Oleh sebab itu *error* saat penggambaran label dapat dinyatakan sebagai berikut :

$$Error (\%) = \left| \frac{nilai_aktual - nilai_ideal}{nilai_ideal} \right| \times 100 \%$$

$$Error (\%) = \left| \frac{42 - 43}{43} \right| \times 100 \%$$

$$Error (\%) = 2,32 \%$$

Maka dapat disimpulkan *error* program adalah 2,32%

4. Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

Algoritma yang telah dirancang berhasil disimulasikan baik pada grafik GUI maupun pada *plant* robot kartesian. *Prototype* robot kartesian pembuat pola batik berbasis *Graphical User Interface* telah berhasil dirancang dan dapat dijalankan dengan durasi maksimal selama 7 menit. *Graphical User Interface* sebagai *interface* robot kartesian telah berhasil dirancang dan semua fasilitas berfungsi dengan baik. *Interface* dirancang dengan kecepatan transfer data 9600 bit per detik. *Prototype* robot kartesian berhasil membuat pola dasar bangun persegi, lingkaran, dan segitiga dan pola motif batik bunga dengan *error* program sebesar 2,32%.

Referensi

- [1] Hokky Situngkir dan Rolan Dahlan. (2008). Fisika Batik. Jakarta : Gramedia Pustaka Utama.
- [2] UNESCO, "Evaluation of the nominations for inscription on the Representative List of the Intangible Cultural Heritage of Humanity," 2009.
- [3] D. Z. Aridanana, E. Sunarno, dan M. Safrodin, "Rancang Bangun Konveyor Untuk Sistem Sortir Berdasarkan Berat Barang," Laporan Tugas Akhir, Politeknik Elektronika Negeri Surabaya, Surabaya, 2010.
- [4] Pitowarno, Endra, (2006). Robotika: Desain, Kontrol, Dan Kecerdasan Buatan. Buku Teks. Yogyakarta: ANDI. 1-3
- [5] V. Kale dan V. Kulkarni, "Object Sorting System Using Robotic Arm," Int. J. Adv. Res. Electr. Instrum. Eng., vol. 2, no. 7, pp. 3400–3407, 2013.
- [6] Darma Putra,"Pengolahan Citra Digital,"Yogyakarta: Andi Offset,2010.
- [7] Zhulian Lesmana," Analisis Suatu Objek Wajah Dengan Svp Menggunakan Program Scilab" Universitas Gunadarma, 2011.
- [8] Rinaldi Munir,"Pengolahan Citra Digital dengan Pendekatan Algoritmik," Informatika: Bandung,2004.
- [9] Johnson, A. R dan Dean W.Wichem. 2002. Applied Multivariate Stastitical Analysis. Edisi ke-5. New York: Prentice-Hall, Inc.
- [10] Johnson, A. R dan Dean W.Wichem. 2002. Applied Multivariate Stastitical Analysis. Edisi ke-5. New York: Prentice-Hall, Inc.
- [11] Rahdiyanta Dwi' "MODUL MESIN BUBUT CNC ", Fakultas Teknik Universitas Negeri Yogyakarta. 2009.