

# Evaluasi Algoritme Bresenham dan Digital Differential Analyzer (DDA) untuk Pengontrolan Koordinasi Gerakan Dua Motor Stepper pada Robot untuk Simulator Penusukan Jarum Medis

<sup>1</sup>Helma Devina<sup>\*</sup>), <sup>1</sup>Jessica, <sup>1</sup>Hesty Susanti, <sup>1</sup>Suprijanto,&<sup>1</sup>Eko Mursito Budi

<sup>1</sup>Instrumentasi dan Kontrol, Fakultas Teknologi Industri, Institut Teknologi Bandung

[helma.devina@gmail.com](mailto:helma.devina@gmail.com)<sup>\*</sup>)

## Abstrak

Citra USG merupakan salah satu kebutuhan pendukung yang krusial dalam menuntun penusukan jarum medis ke dalam tubuh pasien. Konsistensi citra visibilitas jarum tersebut ditentukan oleh dua faktor, yaitu teknis pencitraan yang digunakan (spesifikasi transduser USG dan karakteristik akustik jarum) dan posisi jarum (kedalaman dan sudut penusukan) terhadap muka gelombang bidang yang dipancarkan pada bidang pandang tertentu. Sebelum diaplikasikan ke tubuh pasien, diperlukan suatu informasi berupa peta visibilitas jarum yang dicitrakan USG dengan mensimulasikan teknik penusukan secara *in-plane* menggunakan sistem robot 3 DOF. Hal penting dari robot 3 DOF yang digunakan adalah koordinasi gerakan dua motor stepper yang menjamin gerakan penusukan jarum berada pada trajektori berupa garis lurus untuk berbagai kombinasi kedalaman dan sudut penusukan. Oleh karena itu, pada makalah ini akan dilakukan evaluasi algoritme Bresenham dan Digital Differential Analyzer (DDA) dalam mengatur pemberian jumlah pulsa masukan pada kedua motor stepper sehingga diperoleh gerakan yang diinginkan. Pengujian dilakukan dengan menggunakan jarum spinal 22G pada gerakan penusukan dengan rentang sudut penusukan 0<sup>o</sup> - 90<sup>o</sup> dengan perubahan sudut 22,5<sup>o</sup> dan kedalaman penusukan 10 - 50 mm dengan perubahan kedalaman 10 mm. Berdasarkan nilai RMSE, standar deviasi, linearitas, dan R<sup>2</sup>, algoritme Bresenham lebih baik dibandingkan dengan algoritme DDA.

*Kata Kunci: jarum medis, USG, motor stepper, robot 3 DOF*

## 1 Pendahuluan

USG sebagai umpan balik citra dalam prosedur intervensional invasif minimal merupakan metode yang lebih banyak digunakan dibandingkan dengan metode pencitraan lainnya, seperti X-Ray, CT-Scan, dan MRI. Hal ini dikarenakan USG tidak memiliki risiko radiasi atau ionisasi, ukuran perangkat yang relatif lebih kecil dan murah, dan resolusi temporal yang tinggi [1]. Jika dibandingkan dengan metode konvensional tanpa umpan balik, metode ini mampu meningkatkan kecepatan prosedur dan mengurangi risiko komplikasi akibat kesalahan lokasi penusukan. Selain itu, metode ini juga dapat mengurangi dosis obat anestesi secara signifikan sehingga mengurangi risiko keracunan anestesi lokal sistemik [2].

Dengan adanya jarum sebagai salah satu objek yang dicitrakan, fenomena fisis interaksi gelombang ultrasonik dengan jarum tidak sesederhana fenomena pemantulan biasa karena bentuk dan karakteristik jarum yang khas. Hal tersebut mengakibatkan visibilitas jarum menjadi tidak konsisten di sepanjang batangnya. Fenomena fisis perambatan dan interaksi gelombang (hamburan resonansi) pada jarum tersebut dipengaruhi oleh beberapa parameter, yaitu perambatan gelombang ultrasonik yang dipengaruhi oleh spesifikasi transduser USG dan sudut datang berkas gelombang ( $\alpha$ ), perbedaan karakteristik akustik antar medium, serta bentuk dan dimensi yang unik dari jarum itu sendiri [3]. Dalam aplikasi

klinis, pada citra USG 2 dimensi,  $\alpha$  direpresentasikan oleh posisi jarum yang merupakan kombinasi dari kedalaman dan sudut penusukan.

Untuk mempelajari dan memilih konfigurasi perangkat yang tepat sehingga citra jarum yang dihasilkan konsisten, diperlukan suatu peta visibilitas jarum yang menggambarkan pengaruh posisi penusukan dengan konsistensi citra yang dihasilkan. Peta visibilitas jarum tersebut akan dikembangkan menjadi sistem evaluasi performansi dari beberapa kelas perangkat ultrasonografi klinik sebelum diaplikasikan langsung ke tubuh pasien. Oleh karena itu, diperlukan simulasi penusukan jarum secara *in-plane* menggunakan sistem robot 3 DOF.

Sistem robot yang digunakan pada simulasi ini menggunakan dua motor stepper yang digerakan secara bergantian untuk menghasilkan gerakan penusukan. Hal penting dalam mengatur gerakan motor stepper tersebut adalah koordinasi gerakan yang mampu menjamin gerakan penusukan jarum berada pada trajektori berupa garis lurus sehingga getaran yang dihasilkan minimum dan homogenitas medium tetap terjaga. Oleh karena itu, pada makalah ini akan dilakukan evaluasi algoritme Bresenham dan *Digital Differential Analyzer* (DDA) dalam mengatur pemberian jumlah pulsa masukan pada kedua motor stepper sehingga diperoleh gerakan penusukan yang dibutuhkan.

Evaluasi kedua algoritme tersebut dilakukan dengan membandingkan nilai RMSE, linearitas, standar deviasi, dan standar kesalahan regresi dari setiap algoritme dalam melakukan skenario gerakan penusukan. Gerakan penusukan dilakukan pada rentang sudut penusukan 0 $^{\circ}$  - 90 $^{\circ}$  dengan perubahan sudut 22,5 $^{\circ}$  dan kedalaman penusukan 10 - 50 mm dengan perubahan kedalaman sebesar 10 mm. Pengujian dilakukan menggunakan jarum spinal 25G dengan panjang 9 cm dan terbuat dari *stainless steel* sebagai objek yang digerakan oleh sistem robot.

## 2 Metode

### 2.1 Algoritme Bresenham

Algoritme Bresenham adalah algoritme yang biasa digunakan untuk menggambar garis pada digital plotter. Algoritme ini berfungsi untuk menentukan titik - titik yang perlu dilalui untuk membuat garis lurus dari koordinat awal ke koordinat akhir. Pada algoritme ini dikenal istilah driving axis (DA) dan passive axis (PA). DA merupakan sumbu ordinat/absis yang berperan sebagai pengontrol dan nilainya selalu bertambah 1, sedangkan PA merupakan sumbu ordinat/absis yang bersifat pasif dan bernilai antara 0 atau 1, bergantung pada nilai variabel keputusan Bresenham ( $p_i$ ).

Algoritme ini dimulai dengan menentukan koordinat titik awal ( $x_1, y_1$ ) dan titik akhir ( $x_2, y_2$ ). Kemudian dilanjutkan dengan menentukan DA dan PA berdasarkan nilai kemiringan garis ( $m$ ) diantara kedua titik tersebut. Apabila nilai  $m$  lebih dari 1, maka sumbu ordinat akan berperan sebagai DA dan sumbu absis akan berperan sebagai PA. Sebaliknya, apabila nilai  $m$  kurang dari atau sama dengan 1, maka sumbu absis berperan sebagai DA dan sumbu ordinat berperan sebagai PA.

Setelah menentukan DA dan PA, algoritme Bresenham akan melakukan beberapa perhitungan yang digunakan untuk mendapatkan nilai variabel keputusan Bresenham ( $p_i$ ). Berikut ini merupakan persamaan yang digunakan untuk menentukan nilai  $p_i$ .

$$d_1 = y - y_i \quad (1)$$

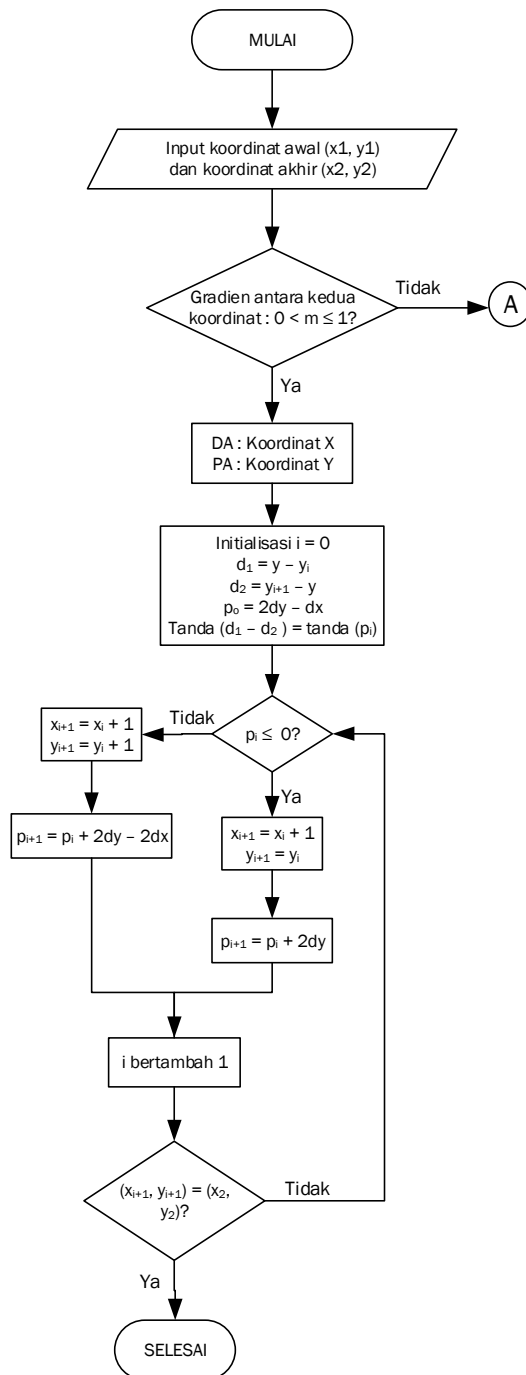
$$d_2 = y_{i+1} - y \quad (2)$$

$$p_o = 2dx - dy \quad (3)$$

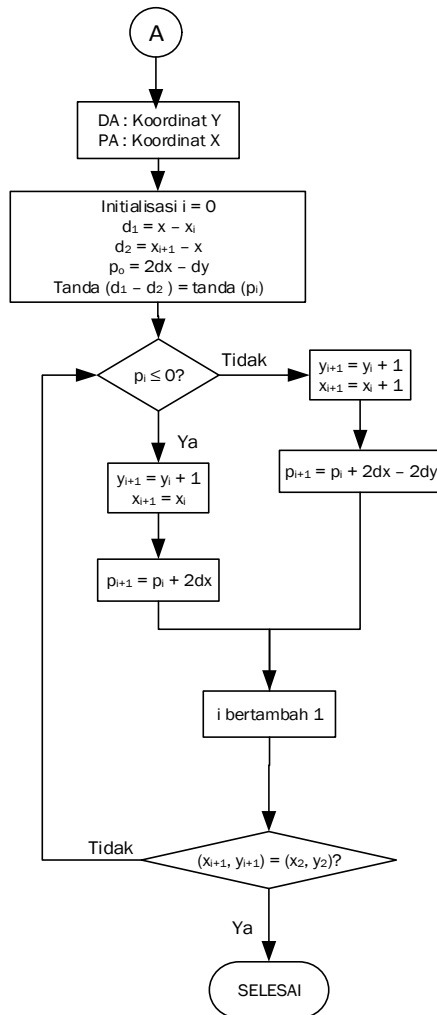
$$\text{tanda}(d_1 - d_2) = \text{tanda}(p_i) \quad (4)$$

Kemudian, tanda dari  $p_i$  akan menentukan koordinat PA selanjutnya. Apabila nilai  $p_i$  kurang dari atau sama dengan nol, maka koordinat PA selanjutnya ( $y_{i+1}$ ) sama dengan koordinat PA saat itu ( $y_i$ ) ditambah dengan 1. Namun, jika nilai  $p_i$  lebih dari nol, maka koordinat PA selanjutnya ( $y_{i+1}$ ) sama dengan koordinat PA saat itu ( $y_i$ ). Proses perhitungan koordinat ini akan terus diulang hingga titik – titik yang dihitung ( $x_{i+1}, y_{i+1}$ ) sudah mencapai titik akhir ( $x_2, y_2$ ) [4].

Secara sederhana, alur penggunaan algoritme Bresenham dapat dilihat pada Gambar 1 dan Gambar 2 berikut ini.



Gambar 1 diagram alir algoritme Bresenham (1)



Gambar 2 diagram alir algoritme Bresenham (2)

## 2.2 Algoritme DDA

Selain algoritme Bresenham, algoritme DDA juga biasa digunakan untuk membuat garis lurus pada *digital plotter*. Berbeda dengan Bresenham, algoritme ini tidak memiliki DA atau PA dan penambahan koordinat pada setiap titik yang dilalui selalu sama. Cara kerja algoritme ini diawali dengan menentukan titik awal koordinat  $(x_1, y_1)$  dan titik akhir  $(x_2, y_2)$ . Selanjutnya, dihitung nilai  $dx$  dan  $dy$  sebagai berikut,

$$dx = x_2 - x_1 \quad (5)$$

$$dy = y_2 - y_1 \quad (6)$$

Apabila nilai  $|dx| < |dy|$ , maka jumlah langkah yang harus ditempuh sama dengan nilai  $|dy|$ , sebaliknya, ketika nilai  $|dx| > |dy|$ , maka jumlah langkahnya sama dengan nilai  $|dx|$ .

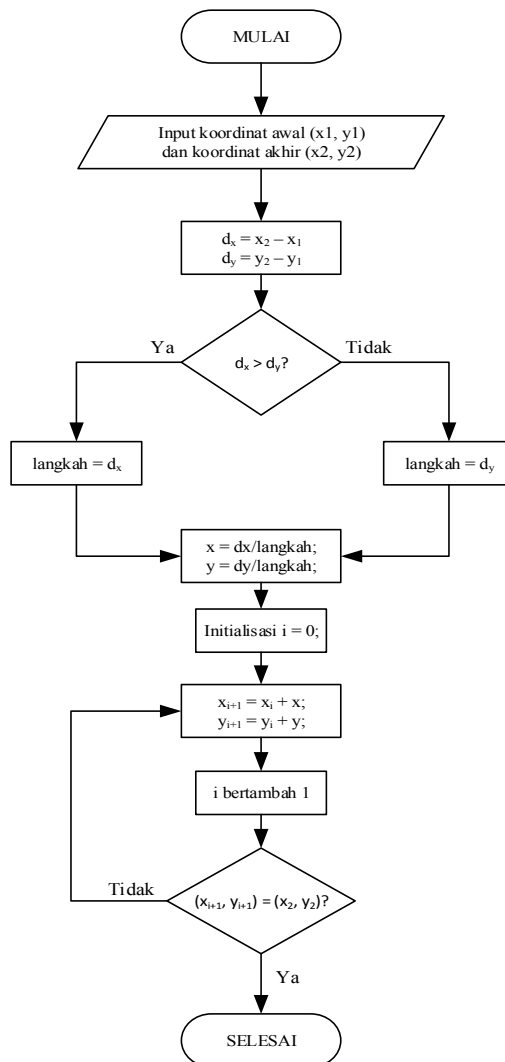
Selanjutnya, dilakukan perhitungan penambahan absis ( $x$ ) dan ordinat ( $y$ ) dengan persamaan sebagai berikut,

$$x = \frac{dx}{langkah} \quad (7)$$

$$y = \frac{dy}{langkah} \quad (8)$$

sehingga koordinat titik berikutnya ( $x_{i+1}, y_{i+1}$ ) akan bernilai sama dengan koordinat titik saat itu ( $x_i, y_i$ ) ditambah  $x$  dan  $y$ . Apabila nilai  $x$  dan  $y$  bukan merupakan bilangan bulat, maka akan dilakukan pembulatan. Proses perhitungan koordinat ini akan terus diulang hingga titik – titik yang dihitung ( $x_{i+1}, y_{i+1}$ ) sudah mencapai titik akhir ( $x_2, y_2$ ) [5].

Berikut ini merupakan diagram alir penggunaan algoritme DDA.



Gambar 3 diagram alir algoritme DDA

Kedua algoritme di atas digunakan untuk menggerakkan motor stepper pada sistem robot 3 DOF sehingga diperlukan beberapa penyesuaian. Motor stepper merupakan aktuator yang mengubah energi elektromagnetik menjadi energi mekanik. Motor ini bergerak secara inkremental dengan mengubah pulsa masukan menjadi langkah yang memiliki penambahan sudut tetap [6].

Pulsa masukan yang diberikan bersifat biner sehingga hasil perhitungan dari kedua algoritme harus merupakan bilangan bulat. Hal ini sesuai untuk algoritme Bresenham dikarenakan keluaran dari DA pasti bernilai 1 dan PA bernilai antara 0 atau 1. Akan tetapi, untuk algoritme DDA perlu dilakukan pembulatan karena keluarannya bergantung pada hasil bagi dari  $dx$  atau  $dy$  dengan langkah.

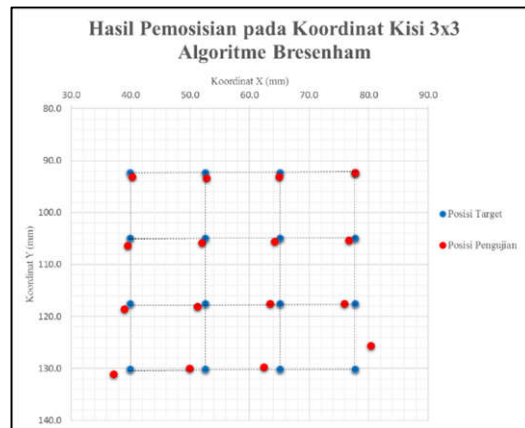
Pada umumnya, penggunaan algoritme Bresenham dan DDA menggunakan koordinat absolut sedangkan pada sistem robot 3 DOF ini digunakan koordinat inkremental. Oleh karena itu, informasi yang akan diambil dari kedua algoritme ini adalah jumlah langkahnya ( $x_{i+1} - x_i$  dan  $y_{i+1} - y_i$ ), bukan koordinat selanjutnya ( $x_{i+1}$  dan  $y_{i+1}$ ).

### 3 Hasil dan Diskusi

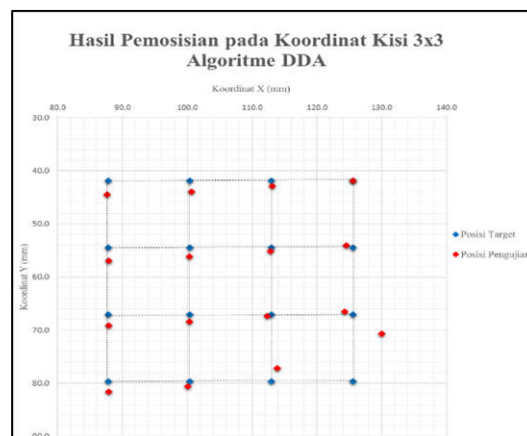
Setelah diimplementasikan pada robot 3 DOF yang ada, kedua algoritme tersebut diuji kinerjanya terhadap 4 parameter, yaitu akurasi, presisi, linearitas, dan getaran. Pengujian diawali dengan merekam gerakan robot ketika melakukan gerakan menggunakan algoritme DDA dan Bresenham. Hasil rekaman selanjutnyadikonversikan menjadi kumpulan gambar menggunakan perangkat lunak 'Free Video to JPG Converter'. Kumpulan gambar yang diperoleh diolah menggunakan perangkat lunak MATLAB untuk mencari trajektori pergerakan ujung jarum. Trajektori pergerakan tersebut kemudian dijadikan basis data dalam uji parameter.

#### 3.1 Uji Akurasi

Uji akurasi dilakukan dengan membandingkan posisi ujung jarum yang diperoleh dengan posisi target. Adapun hasil *plot* posisi ujung jarum hasil pengukuran dengan posisi target dapat dilihat pada gambar berikut ini,



Gambar 4 hasil gerakan pemosisian jarum menggunakan algoritme Bresenham



Gambar 5 hasil gerakan pemosisian jarum menggunakan algoritme DDA



Berdasarkan data posisi ujung jarum yang diperoleh, dilakukan uji RMSE dengan menggunakan persamaan sebagai berikut,

$$RMSE = \sqrt{\frac{\sum(\text{hasil pengukuran} - \text{nilai target})^2}{\text{jumlah data}}} \quad (9)$$

Berdasarkan rumus (9) diperoleh nilai RMSE menggunakan algoritme Bresenham sebesar 1,5mm sedangkan nilai RMSE menggunakan algoritme DDA sebesar 1,97mm. Maka dari itu, algoritme Bresenham memiliki tingkat akurasi yang lebih baik dari DDA.

### 3.2 Uji Presisi

Uji presisi dilakukan dengan menghitung standar deviasi dari titik-titik target pada trajektori ujung jarum. Adapun standar deviasi tersebut diperoleh dengan menggunakan persamaan sebagai berikut,

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}} \quad (10)$$

dimana,

s = standar deviasi

$x_i$  = hasil pengukuran ke - i

$\bar{x}$  = rata-rata hasil pengukuran

n = jumlah data

Dengan menggunakan persamaan (8), dilakukan perhitungan standar deviasi pada koordinat X dan koordinat Y untuk masing-masing gerakan penusukan dan penarikan jarum. Perhitungan uji presisi pada sudut penusukan 45<sup>o</sup> menggunakan kedua algoritme dapat dilihat pada Tabel 1 dan Tabel 2.

**Tabel 1 Standar deviasi penusukan jarum pada sudut 45<sup>o</sup> dengan algoritme Bresenham**

Gerakan	Target Kedalaman (cm)	Standar Deviasi Koordinat X ( mm)	Standar Deviasi Koordinat Y ( mm)
Penusukan	1	0,2	0,2
	2	0,3	0,2
	3	0,3	0,3
	4	0,3	0,3
	5	0,3	0,3
Penarikan	4	0,2	0,3
	3	0,3	0,2
	2	0,3	0,2
	1	0,8	0,2

**Tabel 2 Standar deviasi penusukan jarum pada sudut 458 dengan algoritme DDA**

Gerakan	Target Kedalaman (cm)	Standar Deviasi Koordinat X (mm)	Standar Deviasi Koordinat Y ( mm)
Penusukan	1	0,2	0,2
	2	0,1	0,1
	3	0,2	0,1
	4	0,2	0,1
	5	0,2	0,1
Penarikan	4	0,2	0,2
	3	0,1	0,1
	2	0,2	0,1
	1	0,1	0,1

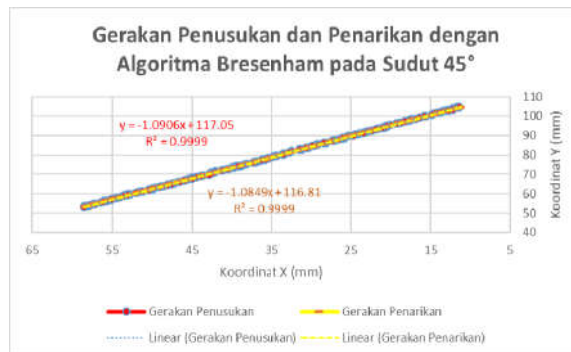
Berdasarkan kedua tabel diatas, diperoleh rentang nilai standar deviasi pada algoritme Bresenham adalah 0,2 – 0,8mm dan algoritme DDA adalah 0,1 – 0,2mm. Semakin kecil rentang standar deviasi yang diperoleh, maka semakin presisi gerakan yang dihasilkan. Maka dari itu, algoritme yang lebih presisi untuk menggerakkan robot 3 DOF adalah algoritme DDA.

### 3.3 Uji Linearitas

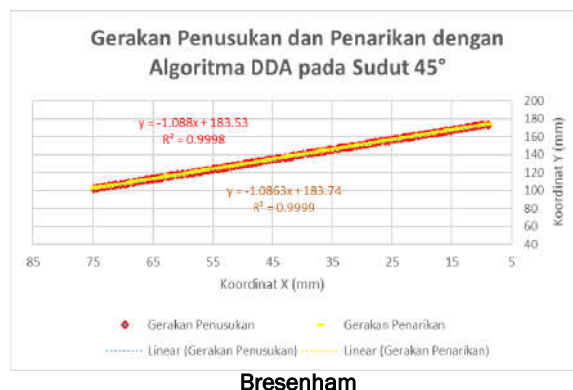
Uji linearitas dilakukan dengan membuat *plot* trajektori dari ujung jarum untuk kemudian dihitung nilai linearitasnya menggunakan nilai koefisien determinasi,  $R^2$ . Nilai koefisien determinasi ini diperoleh dengan menggunakan persamaan sebagai berikut,

$$R^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} \tag{11}$$

Adapun hasil trajektori pergerakan ujung jarum pada sudut penusukan 458 dapat dilihat pada Gambar 6 dan Gambar 7.



Gambar 6 trajektori gerakan penusukan dan penarikan jarum pada sudut 45<sup>o</sup> dengan algoritme



Bresenham

Gambar 7 trajektori gerakan penusukan dan penarikan jarum pada sudut 45<sup>o</sup> dengan algoritme DDA

Berdasarkan kedua gambar diatas, diperoleh  $R^2$  untuk kedua algoritme tidak memiliki perbedaan yang signifikan, yaitu sekitar 0,9999. Hal ini menunjukkan kedua algoritme memiliki nilai linearitas yang baik dikarenakan nilainya mendekati 1.

### 3.4 Uji Getaran

Uji getaran dilakukan dengan menghitung nilai *standard error of the estimate* (S). Nilai ini menunjukkan besarnya kesalahan dari setiap titik yang dilalui oleh ujung jarum. Semakin besar nilai S, maka semakin banyak titik-titik yang berada di luar trajektori sehingga getaran yang dihasilkan akan semakin besar. Berikut ini merupakan persamaan yang digunakan untuk menghitung S.

$$S = \sqrt{\frac{\sum_{i=1}^N (X_i - \bar{X})^2}{N}} \quad (12)$$

dimana,

S = standar kesalahan regresi

$X_i$  = data ke - i

$\bar{X}$  = nilai rata - rata data

N = jumlah data

Untuk melakukan perhitungan S, diperlukan persamaan garis regresi dari trajektori gerakan yang dihasilkan. Berdasarkan Gambar 6 dan Gambar 7, diperoleh persamaan garis regresi algoritme Bresenham dan DDA sebagai berikut,

$$y_{penusukan,Bresenham} = -1,0906x + 183,53 \quad (13)$$

$$y_{penarikan,Bresenham} = -1,0849x + 116,81 \quad (14)$$

$$y_{penusukan,DDA} = -1,088x + 183,53 \quad (15)$$

$$y_{penarikan,DDA} = -1,0863x + 183,74 \quad (16)$$

Dengan menggunakan data lokasi ujung jarum pada trajektori gerakan dan persamaan (13) hingga (16) diperoleh nilai S gerakan penusukan dan penarikan jarum menggunakan algoritme Bresenham sebesar 0,15 dan 0,14mm, sedangkan nilai S gerakan penusukan maupun penarikan menggunakan algoritme DDA sebesar 0,25mm. Berdasarkan nilai S tersebut, algoritme Bresenham memiliki nilai S yang lebih kecil sehingga getaran yang dihasilkan pun lebih kecil. Maka dari itu, ditinjau dari segi getaran, algoritme Bresenham lebih baik digunakan.

#### 4 Kesimpulan

Berdasarkan pengujian yang dilakukan, diperoleh hasil uji algoritme Bresenham dan DDA seperti pada Tabel 3 berikut ini,

**Tabel 3 Hasil uji 4 parameter dengan algoritme DDA dan Bresenham**

Parameter Pengujian	Algoritme DDA	Algoritme Bresenham
Akurasi (RMSE)	1,97 mm	1,5 mm
Presisi (s)	0,1 - 0,2 mm	0,2 - 0,8 mm
Linearitas (R <sup>2</sup> )	0,9999	0,9999
Getaran (S)	Penusukan dan Penarikan: 0,25 mm	Penusukan: 0,15mm Penarikan: 0,14mm

Hasil uji keempat parameter di atas menunjukkan bahwa pada parameter akurasi dan getaran, algoritme Bresenham lebih baik daripada DDA. Sedangkan pada parameter presisi, algoritme DDA memiliki rentang presisi yang lebih baik, dan pada parameter

linearitas, kedua algoritme memiliki kinerja yang sama baiknya. Maka dari itu, dapat disimpulkan bahwa algoritme yang lebih tepat digunakan untuk mengontrol koordinasi gerakan dua motor stepper pada robot untuk simulator penusukan jarum medis adalah Bresenham.

## 5 Daftar Pustaka

- [1] K. Nichols, L. Wright, T. Spencer and W. Culp, 2003, "Changes in Ultrasonographic Echogenicity and Visibility of Needles with Changes in Angles of Insonation," *J Vasc Interv Radiol*, vol. 14, pp. 1553-1557.
- [2] P. Marhofer, M. Greher and S. Kapral, 2005, "Ultrasound Guidance in Regional Anaesthesia," *Br J Anaesth*, vol. 94, pp. 7-1.
- [3] H. Susanti, Suprijanto and D. Kurniadi, 2018, "Two-Dimensional Mapping of Needle Visibility with Linear and Curved Array for Ultrasound-Guided Interventional Procedure," *AIP Conference Proceedings*.
- [4] E. Claridge, "Derivation Of The Bresenham's Line Algorithm," Birmingham.
- [5] E. Angel and D. Shreiner, *Interactive Computer Graphics*, ARM, Inc, 2015.
- [6] C. W. d. Silva ,2007, *Sensors and Actuators: Control System Instrumentation*, New Jersey: Prentice Hall, pp. 405-411.