

# **TUGAS GRAFIKA KOMPUTER**

## **“Membuat Garis”**

*Dibuat guna memenuhi tugas Mata Kuliah Grafika Komputer*



### **Dibuat Oleh:**

Nama : Ninda Sifa Riadi

Npm : 2413025008

Kelas : B

Tanggal : 15 Februari 2025

**PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI JURUSAN**

**PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN**

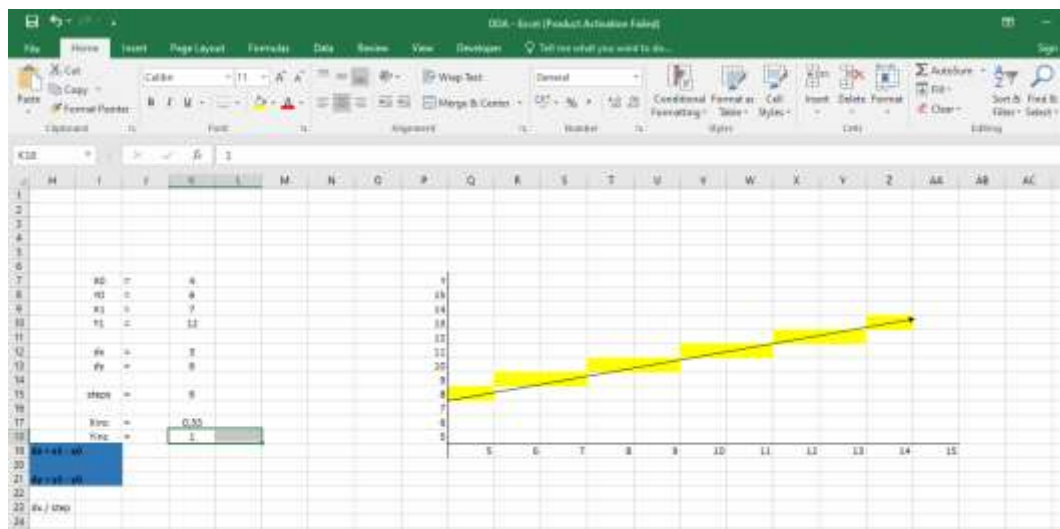
**UNIVERSITAS LAMPUNG**

**2025**

# 1. Membuat Algoritma Dda dan Bresenham

## DDA (Digital Differential Analyzer)

K18												
A	B	C	D	E	F	G	H	I	J	K	L	M
2	Nama : Ninda Sifa Riadi											
3	Kelas : B											
4	NPM : 2413025008											
5												
6			I		X	Y		(X,Y)				
7			0		4	6			X0 =	4		
8			1		5,33	7			Y0 =	6		
9			2		7	8			X1 =	7		
10			3		7	9			Y1 =	12		
11			4		7,33	10						
12			5		8	11			dx =	3		
13			6		8,33	12			dy =	6		
14			7		9	13						
15			8		9,33	14			steps =	6		
16			9		10	15						
17			Steps						Xinc =	0,33		
18									Yinc =	1		
19			step = jika $dx > dy$ = dx		x = x + xinc		dx = x1 - x0					
20			step = jika $dy > dx$ = dy		y = y + yinc		dy = y1 - y0					
21												
22												
23								dx / step				



DDA adalah sebuah algoritma yang digunakan untuk menggambar garis dan kurva pada layar. Algoritma ini bekerja dengan menghitung perbedaan antara koordinat x dan y dari dua titik yang akan dihubungkan oleh garis, Pada algoritma DDA, koordinat

x dan y dapat memiliki nilai desimal (misalnya 0,5 atau 3,7). karena algoritma DDA menggunakan perhitungan desimal untuk menghitung nilai x dan y untuk setiap titik pada garis. Jaadi pada algoritma DDA, ada komanya antara nilai x dan y.

#### Langkah-langkah

1. Tentukan dua titik yang akan dihubungkan oleh garis, yaitu  $(x_1, y_1)$  dan  $(x_2, y_2)$ .
2. Hitung perbedaan antara koordinat x dan y dari dua titik tersebut, yaitu  $dx = x_2 - x_1$  dan  $dy = y_2 - y_1$ .
3. Tentukan jumlah langkah yang diperlukan untuk menggambar garis, yaitu  $step = \max(|dx|, |dy|)$ .
4. Hitung inkrement x dan y untuk setiap langkah, yaitu  $xIncrement = dx / step$  dan  $yIncrement = dy / step$ .
5. Mulai dari titik awal  $(x_1, y_1)$ , gambar garis dengan menghitung nilai x dan y untuk setiap langkah menggunakan inkrement x dan y.

#### ➤ Kelebihan

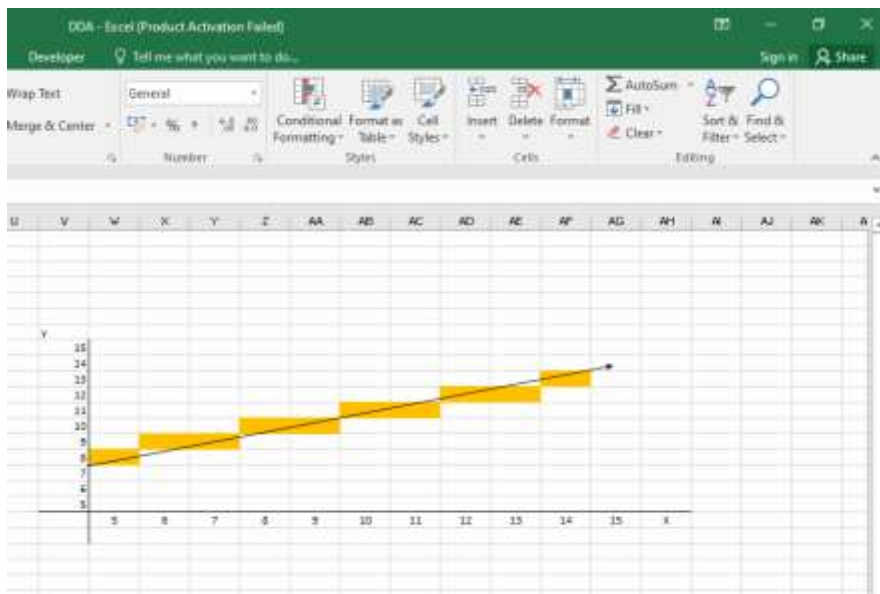
- Mudah diimplementasikan
- Cepat dalam menggambar garis
- Dapat digunakan untuk menggambar garis dan kurva yang kompleks

#### ➤ Kekurangan

- Tidak dapat menggambar garis yang sangat curam
- Dapat menghasilkan garis yang tidak halus jika inkrement x dan y terlalu besar

## Bresenham

	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1			Nama : Ninda Sifa Riadi											
2			Kelas : B											
3			NPM : 2413025008											
4														
5														
6					P		X	Y			X0	=	5	
7					1		5	8			Y0	=	8	
8					-7		6	9			X1	=	14	
9					3		7	9			X2	=	13	
10					-5		8	10						
11					5		9	10			dx	=	9	
12					-3		10	11			dy	=	5	
13					7		11	11						
14					-1		12	12			d1	=	10	
15					9		13	12			d2	=	8	
16					1		14	13			p	=	1	
17											m	=	0,5	
18														
19														
20					$m = y1 - y0 / x1 - x0$									
21					garis 0 < m < 1									
22							P = d1 - dx				d1 = 2 * dy		dx = x1 - x0	
23							jika p > 0 maka:				d2 = 2 * (dx - dy)		dy = y1 - y0	
24							p = p - d2							
25							y = y + 1							
26														
27							jika p < 0 maka:							
28							p = p + d1							
29							y = y							
30														
31														



Bresenham adalah sebuah algoritma yang digunakan untuk menggambar garis pada layar dengan menggunakan teknik interpolasi. Algoritma ini bekerja dengan menghitung nilai x dan y untuk setiap titik pada garis. Pada algoritma Bresenham, koordinat x dan y hanya dapat memiliki nilai integer (misalnya 0 atau 3). algoritma

Bresenham menggunakan perhitungan integer untuk menghitung nilai x dan y untuk setiap titik pada garis. Karena itu, pada algoritma Bresenham, tidak ada komanya antara nilai x dan y.

#### Langkah-langkah

1. Tentukan dua titik yang akan dihubungkan oleh garis, yaitu  $(x_1, y_1)$  dan  $(x_2, y_2)$ .
2. Hitung perbedaan antara koordinat x dan y dari dua titik tersebut, yaitu  $dx = x_2 - x_1$  dan  $dy = y_2 - y_1$ .
3. Tentukan jumlah langkah yang diperlukan untuk menggambar garis, yaitu  $step = \max(|dx|, |dy|)$ .
4. Hitung inkrement x dan y untuk setiap langkah, yaitu  $xIncrement = dx / step$  dan  $yIncrement = dy / step$ .
5. Mulai dari titik awal  $(x_1, y_1)$ , gambar garis dengan menghitung nilai x dan y untuk setiap langkah menggunakan inkrement x dan y.
6. Pada setiap langkah, periksa apakah nilai x dan y telah melebihi batas layar. Jika ya, maka hentikan proses penggambaran.

#### ➤ Kelebihan

- Dapat menggambar garis yang sangat curam
- Menghasilkan garis yang halus
- Efisien dalam penggunaan memori

#### ➤ Kekurangan

- Lebih kompleks daripada algoritma DDA
- Memerlukan lebih banyak perhitungan daripada algoritma DDA

The screenshot shows a web browser window with a canvas-based drawing application. The canvas displays a blue square with a black border and the text "TUGAS ALGORITMA GURU DSA NINDA SIFA KHALI" in the center. The browser's developer tools are open, showing the HTML and JavaScript code. The HTML includes a canvas element with id "canvasku" and a script tag. The JavaScript code defines a drawDGA function that draws a square and text on the canvas.

5

1. `<!DOCTYPE html>`

deklarasi dokumen HTML5.

2. `<title>Tugas Algoritma Garis DDA_Ninda Sifa Riadi</title>`

elemen yang menentukan judul dokumen.

3. `<style>body { background-color: rgb(206, 132, 175); }</style>`

Elemen CSS gunanya menentukan bahwa warna latar belakang dokumen harus rgb(206, 132, 175). Sesuai dengan warna yang ingin dipakai atau yang ingin dipilih.

4. `<body>`

Badan dari isi seluruh elemen

5. `<canvas id="canskaku" width="500" height="500"></canvas>`

Ini adalah elemen yang membuatkan sebuah canvas dengan ukuran 500x500 piksel.

Atribut id menentukan bahwa canvas ini memiliki ID dengan nama canskaku.

6. `<script>...</script>`

Berisikan kode JavaScript.

7. `let canvas = document.getElementById("canskaku");`

Kode JavaScript yang membuatkan sebuah variabel canvas yang memanggil canvas dengan ID canskaku.

8. `let ctx = canvas.getContext("2d");`

Membuatkan sebuah variabel ctx yang mereferensikan konteks 2D dari canvas.

8. `ctx.fillStyle = "#ffffff";`

Kode JavaScript yang menentukan bahwa warna pengisi dari konteks 2D harus #ffffff (putih).

```
10. ctx.fillRect(0, 0, 500, 500);
```

Ukuran 500x500 piksel di posisi (0, 0) dengan warna putih.

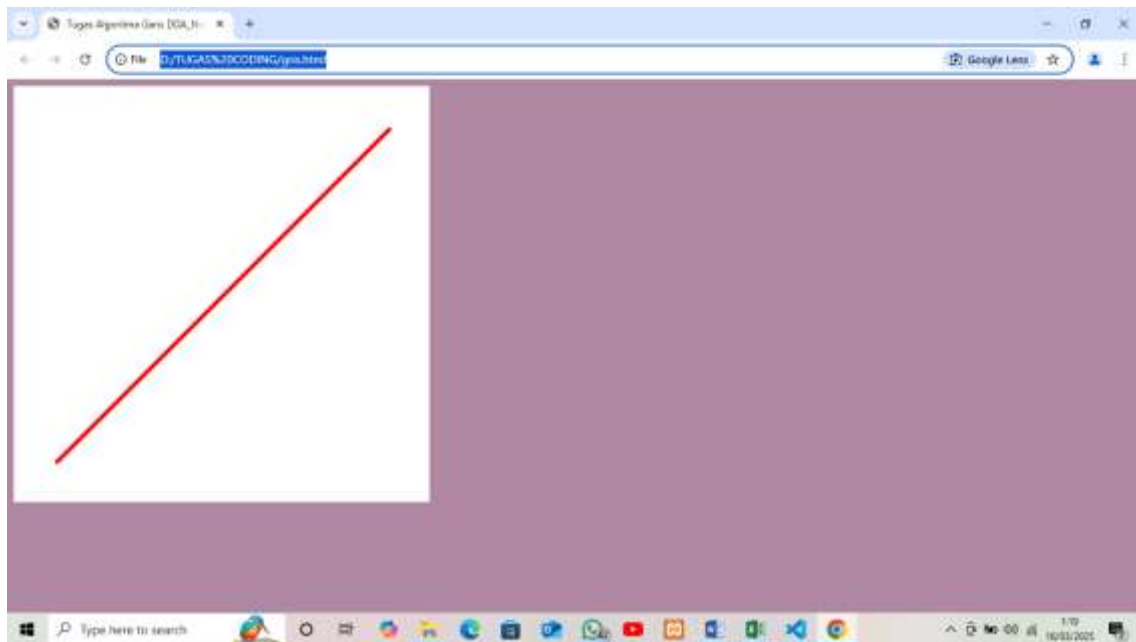
```
11. function drawDDA(x1, y1, x2, y2) { ... }
```

Sebuah fungsi drawDDA yang mengambil empat parameter: x1, y1, x2, dan y2.

```
16. drawDDA(450, 50, 50, 450);*
```

Yang memanggil fungsi drawDDA dengan parameter x1=450, y1=50, x2=50, dan y2=450.

## Hasil





### Kesimpulan

Pada algoritma DDA, koordinat x dan y dapat memiliki nilai desimal (misalnya 0,5 atau 3,7). Algoritma DDA menggunakan perhitungan desimal untuk menghitung nilai x dan y untuk setiap titik pada garis. algoritma DDA, ada komanya antara nilai x dan y.

Pada algoritma Bresenham, koordinat x dan y hanya dapat memiliki nilai integer (misalnya 0 atau 3). Bresenham menggunakan perhitungan integer untuk menghitung nilai x dan y untuk setiap titik pada garis. Bresenham, tidak ada komanya antara nilai x dan y. DDA menggunakan nilai desimal dan memiliki komanya antara nilai x dan y. Bresenham menggunakan nilai integer dan tidak memiliki komanya antara nilai x dan y.