

LAPORAN PRAKTIKUM GRAFIKA KOMPUTER

“Membuat Garis DDA dan Garis Bresenham”



Dosen Pengampu:

Febi Eka Febriansyah, M.T.

Wartariyus, S.Kom., M.T.I.

Putut Aji Nalendro, M.Pd.

Disusun Oleh:

Nama : Nabila Fatma Sari

NPM : 2453025003

Kelas : PTI 24 A

**PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
JURUSAN PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS LAMPUNG**

2025

Garis DDA (Digital Differential Analyzer)

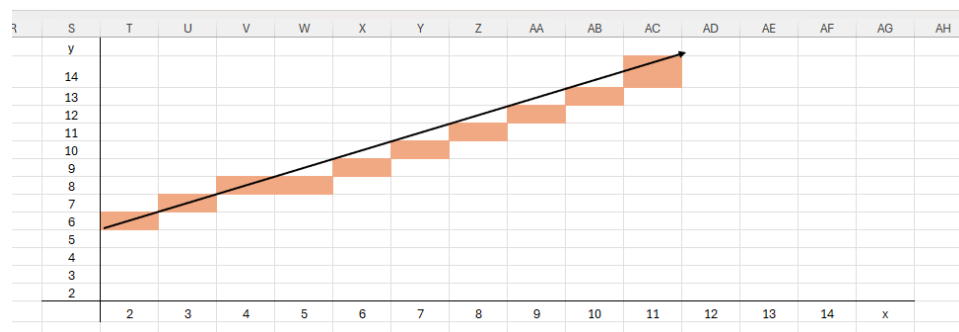
Digital Differential Analyzer (DDA) adalah algoritma pembentukan garis berdasarkan perhitungan dx maupun dy . Garis dibuat dengan menentukan dua endpoint, yaitu titik awal dan titik akhir. Setiap koordinat titik yang membentuk garis diperoleh dari perhitungan, kemudian dikonversikan menjadi nilai integer. Prinsip dari Algoritma Digital Differential Analyzer (DDA) adalah mengambil nilai integer terdekat dengan jalur garis berdasarkan atas sebuah titik yang telah ditentukan sebelumnya (titik awal garis).

Cara Kerja

1. Input titik awal (X_0, Y_0) dan titik akhir (X_1, Y_1)
2. Hitung selisih koordinat:
 - $dx = x_1 - x_0$
 - $dy = y_1 - y_0$
3. Tentukan jumlah langkah (steps) berdasarkan nilai maksimum antara nilai dx atau dy .
4. Hitung nilai increment untuk setiap langkah:
 - $X_{inc} = dx / \text{steps}$
 - $Y_{inc} = dy / \text{steps}$
5. Iterasi sebanyak steps kali:
 - Tambahkan X_{inc} ke x .
 - Tambahkan Y_{inc} ke y .
 - Bulatkan hasil ke pixel terdekat dan plot titik tersebut.
6. Ulangi langkah hingga mencapai titik akhir.

Berikut hasil Algoritma DDA dan Diagram Garis dalam table

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|----|-----|------------------------------|--------------------------|---|-------|--------------|--------------|---|---|---------|------|---|---|---|---|---|---|
| 1 | NPM | 2453025003 Nabila Fatma Sari | | | | | | | | | | | | | | | |
| 2 | | | i | | X | Y | (x,y) | | | X0 = | 2 | | | | | | |
| 3 | | | 0 | | 2 | 6 | 2,6 | | | Y0 = | 6 | | | | | | |
| 4 | | | 1 | | 3 | 6,83 | 3,7 | | | X1 = | 8 | | | | | | |
| 5 | | | 2 | | 4 | 7,66 | 4,8 | | | Y1 = | 11 | | | | | | |
| 6 | | | 3 | | 5 | 8,49 | 5,8 | | | | | | | | | | |
| 7 | | | 4 | | 6 | 9,32 | 6,9 | | | dx = | 6 | | | | | | |
| 8 | | | 5 | | 7 | 10,15 | 7,10 | | | dy = | 5 | | | | | | |
| 9 | | | 6 | | 8 | 10,98 | 8,11 | | | | | | | | | | |
| 10 | | | 7 | | 9 | 11,81 | 9,12 | | | steps = | 6 | | | | | | |
| 11 | | | 8 | | 10 | 12,64 | 10,13 | | | | | | | | | | |
| 12 | | | 9 | | 11 | 14 | 11,14 | | | Xinc = | 1 | | | | | | |
| 13 | | | steps | | | | | | | Yinc = | 0,83 | | | | | | |
| 14 | | | | | | | | | | | | | | | | | |
| 15 | | | step = jika dx > dy = dx | | | x = x + xinc | dx = x1 - x0 | | | | | | | | | | |
| 16 | | | step = jika dy > dx = dy | | | y = y + yinc | dy = y1 - y0 | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | |



Kode Program dan Penjelasan

1. Struktur Data Code

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Tugas Algoritma Garis DDA_Nabila</title>
7      <style>
8          body {
9              background-color: #800000;
10         }
11     </style>
12 </head>

```

- `<!DOCTYPE html>`, Menandakan bahwa dokumen ini adalah HTML5.
- `<html lang="en">`, Membuka elemen html, di mana atribut `lang="en"` menunjukkan bahwa bahasa utama halaman ini adalah bahasa Inggris.
- `<head>`, Bagian kepala dari dokumen HTML, tempat untuk metadata dan pengaturan tampilan halaman.

- `<meta charset="UTF-8">`, Menentukan bahwa halaman ini menggunakan encoding UTF-8, yang mendukung berbagai karakter dan simbol.
- `<meta name="viewport" content="width=device-width, initial scale=1.0">`, Memastikan bahwa tampilan halaman disesuaikan dengan lebar perangkat dan tidak diperbesar secara default.
- `<title>`, Tugas Algoritma Garis DDA_Nabila`</title>` Menentukan judul halaman yang akan ditampilkan di tab browser.

2. Elemen `<canvas>` dalam HTML

```

14 <body>
15   <canvas id="canvasku" width=" 370" height="370"></canvas>
16
17   <script>
18     let canvas = document.getElementById("canvasku");
19     let ctx = canvas.getContext("2d");
20
21     ctx.fillStyle = "#ffffff";
22     ctx.fillRect(0, 0, canvas.width, canvas.height);
23

```

- `<canvas>`, Elemen yang digunakan untuk menggambar grafik menggunakan JavaScript.
- `id="canvasku"`, Memberikan ID untuk mengakses elemen ini melalui JavaScript.
- `width="370" height="370"`, Menentukan ukuran kanvas 370x370 piksel.
- `document.getElementById("canvasku")`, Mengambil elemen `<canvas>` berdasarkan ID "canvasku".
- `canvas.getContext("2d")`, Mengakses konteks gambar 2D, yang memungkinkan kita menggambar bentuk, garis, atau warna pada kanvas.
- `ctx.fillStyle = "#ffffff"`; → Mengatur warna isi (fill color) menjadi putih (#ffffff).
- `ctx.fillRect(0, 0, canvas.width, canvas.height)`; → Menggambar sebuah persegi panjang yang menutupi seluruh area kanvas dengan warna yang telah ditentukan, yaitu putih.
 - (0,0), Titik awal menggambar (pojok kiri atas kanvas).
 - `canvas.width, canvas.height`, Ukuran persegi panjang yang sama dengan ukuran kanvas (370x370 piksel).

- dx, Selisih koordinat x antara titik awal dan titik akhir.
- dy, Selisih koordinat y antara titik awal dan titik akhir.
- Nilai dx dan dy menentukan seberapa besar perubahan posisi horizontal dan vertikal.
- Menentukan jumlah langkah (step) yang dibutuhkan untuk menggambar garis.
- Math.abs(dx) dan Math.abs(dy), Mengambil nilai absolut dari perubahan x dan y untuk memastikan hasilnya selalu positif.
- Math.max(), Memilih nilai terbesar antara dx dan dy, karena langkah yang dibutuhkan bergantung pada perubahan yang lebih besar.
- Xinc dan Yinc, Menentukan seberapa besar perubahan koordinat x dan y pada setiap langkah, dan nilai ini digunakan untuk memperbarui posisi titik yang digambar.
- Menginisialisasi koordinat awal (X0, Y0) sebelum mulai menggambar garis.

3. Perulangan Code Algoritma

```

36     for (let i = 0; i <= step; i++) {
37         ctx.fillStyle = "#000000";
38         ctx.fillRect(Math.round(x), Math.round(y), 2, 2);
39
40         x += xIncrement;
41         y += yIncrement;
42     }
43 }
44

```

- Loop berjalan sebanyak step untuk menggambar garis.
- Setiap iterasi:
 - Warna titik diatur ke hitam (#000000).
 - Titik (2x2 piksel) digambar di (x, y), dengan koordinat dibulatkan agar sesuai grid.
 - Koordinat diperbarui menggunakan xIncrement dan yIncrement untuk menuju titik akhir.
- Kode ini memastikan garis digambar dengan titik-titik kecil yang tersusun rapi.

4. Pemanggilan Fungsi Program

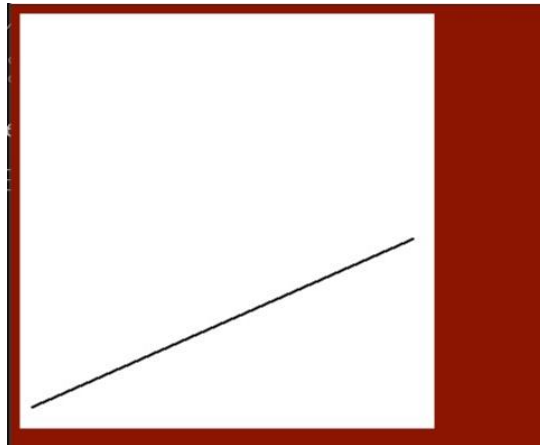
```

45 // Example: draw a line from (50, 350) ke (450, 150)
46 drawDDA(10, 350, 350, 200);
47 </script>
48 </body>
49 </html>

```

- drawDDA(10, 350, 350, 200);
 - (10, 350) Titik awal garis.
 - (350, 200) Titik akhir garis.
- Fungsi drawDDA() yang sudah dibuat sebelumnya akan menggambar garis berdasarkan koordinat ini.

Output yang dihasilkan



Garis Bresenham

Algoritma garis Bresenham adalah algoritma penggambaran garis yang menentukan titik-titik raster n -dimensi yang harus dipilih untuk membentuk perkiraan dekat terhadap garis lurus antara dua titik. Algoritma ini umumnya digunakan untuk menggambar garis primitif dalam gambar bitmap (misalnya pada layar komputer), karena algoritma ini hanya menggunakan penjumlahan, pengurangan, dan pergeseran bit integer, yang semuanya merupakan operasi yang sangat murah dalam arsitektur komputer yang umum secara historis. Algoritma ini adalah algoritma kesalahan inkremental, dan salah satu algoritma paling awal yang dikembangkan dalam bidang grafik komputer.

Cara Kerja

- Inisialisasi: Tentukan titik awal dan titik akhir garis serta hitung perbedaan koordinat dx dan dy.
- Tentukan arah pergerakan: Berdasarkan gradien garis, tentukan apakah garis lebih dominan dalam sumbu x (horizontal) atau y (vertikal).
- Hitung faktor keputusan: Gunakan rumus untuk menentukan piksel selanjutnya yang paling mendekati garis ideal.
- Iterasi melalui loop: Tambahkan nilai pada koordinat x atau y berdasarkan hasil perhitungan faktor keputusan, lalu gambar piksel.
- Ulangi hingga mencapai titik akhir: Proses ini dilakukan berulang kali hingga garis selesai tergambar di layar.

Berikut hasil Garis Bresenham dan Diagram Garis dalam table

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|----|---|---|---------------|---|------------------|----------------|---|---|----------|---|------|---|---|---|---|---|---|---|---|
| 2 | | | P | | X | Y | | | X0 | = | 2 | | | | | | | | |
| 3 | | | 4 | | 2 | 6 | | | Y0 | = | 6 | | | | | | | | |
| 4 | | | 2 | | 3 | 7 | | | X1 | = | 8 | | | | | | | | |
| 5 | | | 0 | | 4 | 8 | | | Y1 | = | 11 | | | | | | | | |
| 6 | | | -2 | | 5 | 9 | | | | | | | | | | | | | |
| 7 | | | 8 | | 6 | 10 | | | dx | = | 6 | | | | | | | | |
| 8 | | | 6 | | 7 | 11 | | | dy | = | 5 | | | | | | | | |
| 9 | | | 4 | | 8 | 12 | | | | | | | | | | | | | |
| 10 | | | 2 | | 9 | 13 | | | d1 | = | 10 | | | | | | | | |
| 11 | | | 0 | | 10 | 14 | | | d2 | = | 2 | | | | | | | | |
| 12 | | | -2 | | 11 | 15 | | | p | = | 4 | | | | | | | | |
| 13 | | | steps | | | | | | m | = | 0,83 | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | | |
| 15 | | | m=y1-y0/x1-x0 | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | | |
| 17 | | | garis0<m<1 | | P = d1-dx | d1=2*dy | | | dx=x1-x0 | | | | | | | | | | |
| 18 | | | | | jika p >=0 maka: | d2 = 2*(dx-dy) | | | dy=y1-y0 | | | | | | | | | | |
| 19 | | | | | p= p-d2 | | | | | | | | | | | | | | |
| 20 | | | | | y=y+1 | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | | | | | |
| 22 | | | | | jika p <0 maka: | | | | | | | | | | | | | | |
| 23 | | | | | p=p+d1 | | | | | | | | | | | | | | |
| 24 | | | | | y=y | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | | | | | | | |

