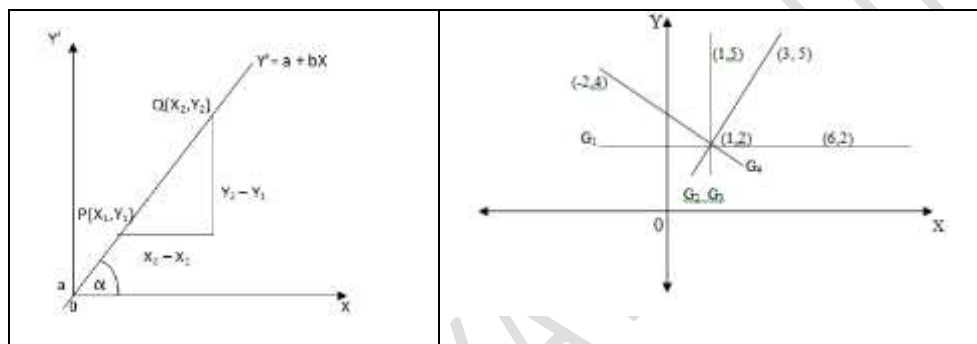


Algoritma garis adalah algoritma untuk menentukan lokasi piksel yang paling dekat dengan garis sebenarnya. Grafik garis lurus dibentuk dari fungsi atau persamaan linier. Daerah asal, daerah kawan dan daerah hasil dari fungsi linier adalah himpunan bilangan riil dan fungsi ini merupakan fungsi 1-satu-satu pada dengan fungsi inversnya. Fungsi linier dibentuk dari persamaan berikut ini :

$$y - y_1 = m(x - x_1) \quad (2.1)$$

Secara umum, jika titik P dengan koordinat  $(x_1, y_1)$  dan titik Q dengan koordinat  $(x_2, y_2)$  dua titik merupakan titik yang dilalui oleh sebuah garis lurus dengan kemiringan garis lurus ( $m$ ) merupakan ukuran kecuraman suatu garis maka semakin besar nilai  $m$  maka semakin curam garis tersebut.



Gambar 2.1. Kemiringan Garis

Nilai kemiringan garis tersebut dapat dihitung sebagai berikut :

$$m = \frac{\text{kenaikan}}{\text{lariah}} = \frac{y_2 - y_1}{x_2 - x_1} \quad (2.2)$$

Ada beberapa jenis kemiringan yang dapat terbentuk :

- Jika nilai  $m = 0$  maka akan terbentuk garis mendatar;
- Jika  $m$  tidak terdefinisi (karena dibagi 0) maka akan terbentuk garis tegak lurus;
- Jika nilai  $m$  bernilai positif maka akan terbentuk garis yang naik ke kanan;
- Jika nilai  $m$  bernilai negatif maka terbentuk garis yang jatuh ke kanan.

berdasarkan persamaan (2.2) diperoleh :

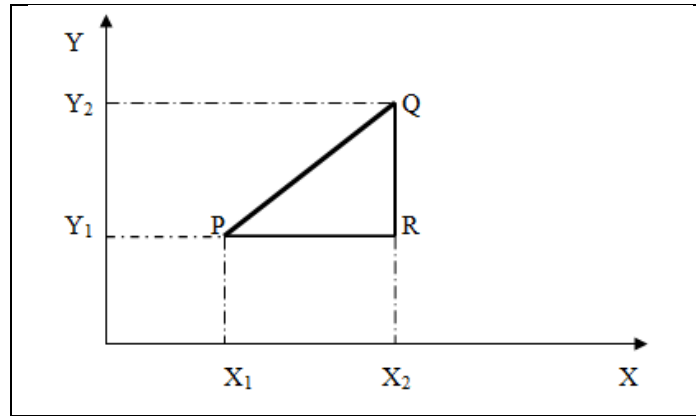
$$m(G_1) = \frac{2 - 2}{6 - 1} = \frac{0}{5} = 0$$

$$m(G_2) = \frac{5 - 2}{3 - 1} = \frac{3}{2}$$

$$m(G_3) = \frac{5 - 2}{1 - 1} = \frac{3}{0} = \text{tidak terdefinisi}$$

$$m(G_4) = \frac{4 - 2}{-2 - 1} = -\frac{4}{3}$$

Misalkan  $P(x_1, y_1)$  dan  $Q(x_2, y_2)$  dua buah titik pada bidang sistem koordinat,



**Gambar 2.2. Jarak Antara Dua Titik**

Untuk mencari jarak antara P dan Q kita bisa menggunakan teorema Pythagoras.

$$|PQ|^2 = |PR|^2 + |QR|^2$$

$$|PQ|^2 = |X_2 - X_1|^2 + |Y_2 - Y_1|^2$$

$$|PQ| = \sqrt{|X_2 - X_1|^2 + |Y_2 - Y_1|^2}$$

Misalkan terdapat dua titik P(6,7) dan Q(10,10), Jarak antara kedua titik ini adalah :

$$D(x,y) = \sqrt{(10 - 6)^2 + (10 - 7)^2} = \sqrt{25} = 5$$

Ada dua cara menentukan persamaan garis lurus, yaitu :

- berdasarkan 2 titik atau lebih yang dilalui oleh grafik; dan
- berdasarkan nilai koefisien arah / kemiringan garis dan sebuah titik yang dilalui grafik.

Algoritma garis adalah menggambarkan garis pada komputer berdasarkan dua buah titik yaitu titik awal  $(x_1, y_1)$  dan  $(x_2, y_2)$ .

Kriteria algoritma garis yang baik :

- Garis harus lurus
- Kerapatan titik-titik konstan
- Kerapatan titik tidak tergantung pada kemiringan garis
- Waktu proses algoritma harus cepat;

Kuadran garis berdasarkan arah garis,

Kuadran	Kriteria	Arah Garis	Contoh
I	$(x_1 < x_2)$ dan $(y_1 < y_2)$		$(1,1) - (4,5)$ $(-3,2) - (-1,4)$
II	$(x_1 > x_2)$ dan $(y_1 < y_2)$		$(4,2) - (3,4)$ $(-3,-3) - (-6,-1)$
III	$(x_1 > x_2)$ dan $(y_1 > y_2)$		$(6,-2) - (4,-5)$ $(9,5) - (1,2)$
IV	$(x_1 < x_2)$ dan $(y_1 > y_2)$		$(3,9) - (6,2)$ $(-2,1) - (4,-5)$

### 3.1. ALGORITMA DDA (*Digital Differential Analyzer*)

DDA adalah salah satu algoritma pembentukan garis berdasarkan pada dua buah titik yang diketahui sebagai titik awal  $(x_1, y_1)$  dan titik akhir  $(x_2, y_2)$  sebuah garis. Langkah-langkah pembentukan garis menurut algoritma DDA yaitu

- 1) Tentukan dua titik yang akan dihubungkan sebagai titik awal  $(x_1, y_1)$  dan titik akhir  $(x_2, y_2)$ ;
- 2) Hitung nilai  $x$  dan  $y$  dengan persamaan berikut :
$$dx = x_2 - x_1$$
$$dy = y_2 - y_1$$
- 3) Tentukan nilai beda yaitu nilai penambahan nilai  $x$  dan nilai  $y$ , dengan menggunakan persamaan :
$$beda = \begin{cases} |dx|, & |dy| < |dx| \\ |dy|, & \text{selain itu} \end{cases}$$
- 4) Tentukan koordinat berikutnya dengan persamaan
$$X_+ = \frac{dx}{beda}$$
$$Y_+ = \frac{dy}{beda}$$
- 5) Tentukan koordinat titik selanjutnya yaitu  $(x_1 + X_+, y_1 + Y_+)$ ,  $(x_1 + X_+ + X_+, y_1 + Y_+ + Y_+)$ , dan seterusnya;
- 6) Ulangi langkah 5 sampai ke koordinat akhir  $(x_2, y_2)$ ;
- 7) Gambarkan titik titik dari koordinat awal sampai koordinat akhir;

#### Contoh kasus :

Gambarkan garis dari titik  $(-3, 2)$  sampai  $(4, 6)$  dengan algoritma DDA!

Penyelesaian :

- 1) Tentukan dua titik yang akan dihubungkan sebagai titik awal  $(-3, 2)$  dan titik akhir  $(4, 6)$ 
$$x_1 = -3, y_1 = 2, x_2 = 4, y_2 = 6;$$
- 2) Hitung nilai  $x$  dan  $y$  dengan persamaan berikut :
$$dx = 4 - (-3) = 7$$
$$dy = 6 - 2 = 4$$
- 3) Tentukan nilai beda yaitu nilai penambahan nilai  $x$  dan nilai  $y$ ,  $|4| < |7|$  sehingga  $beda = 7$ ;
- 4) Tentukan koordinat berikutnya dengan persamaan
$$X_+ = \frac{7}{7} = 1$$
$$Y_+ = \frac{4}{7} = 0,57$$

5) Koordinat titik selanjutnya :

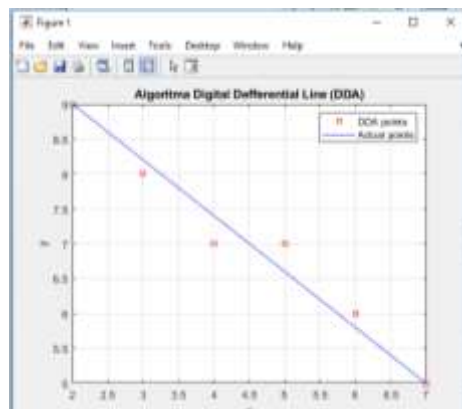
$(-3+1, 2+0.57) = (-2, 2.57)$  koordinat selanjutnya  $(-2+1, 2.57+0.57) = (-1, 3.14)$  koordinat selanjutnya  $(-1+1, 3.14+0.57) = (0, 3.71)$  dan seterusnya.

Tulis listing program ini menggunakan software MATLAB.

```
clc
clear all
x0=input('Enter x0=');
y0=input('Enter y0=');
x1=input('Enter x1=');
y1=input('Enter y1=');
dx=x1-x0;
dy=y1-y0;
if (abs(dx)>abs(dy))
    steps=abs(dx);
else
    steps=abs(dy);
end
xinc=(dx/steps);
yinc=(dy/steps);

x(1)=x0; y(1)=y0;
for i=1:steps
    x(i+1)=x(i)+ xinc;
    y(i+1)=y(i)+ yinc;
end
plot(round(x),round(y),'rs');
grid on, hold on
plot(x,y,'b');
legend('DDA points','Actual points');
xlabel('x');
ylabel('y');
title('Algoritma Digital Defferential Line (DDA)')
```

Akan menghasilkan :



Gambar 3.1. Garis Hasil DDA dengan Matlab

### 3.2. ALGORITMA GARIS BRESSENHEN

Prosedur untuk menggambar kembali garis dengan membulatkan nilai x atau y ke bilangan integer membutuhkan waktu serta variabel x,y dan m merupakan bilangan real karena kemiringan merupakan nilai pecahan. Bresenham mengembangkan algoritma klasik yang lebih menarik karena hanya menggunakan perhitungan matematika dengan bilangan integer. Dengan demikian tidak perlu membulatkan nilai posisi setiap pixel setiap waktu. Algoritma garis Bresenham disebut juga midpoint line algorithm adalah algoritma konversi penambahan nilai integer yang juga dapat diadaptasi untuk menggambar sebuah lingkaran.

Langkah-langkah untuk membentuk garis menurut algoritma ini adalah :

- 1) Tentukan dua titik yang akan dihubungkan dalam pembentukan garis.
- 2) Tentukan salah satu titik di sebelah kiri sebagai titik awal ( $x_0, y_0$ ) dan titik akhir ( $x_1, y_1$ )
- 3) Hitung  $x, y, 2x$  dan  $2y-2x$
- 4) Hitung parameter  $p_0 = 2y-x$
- 5) Untuk setiap  $x_k$  sepanjang jalur garis, dimulai dengan  $k=0$  bila  $p_k < 0$  maka titik selanjutnya ( $x_k + 1, y_k$ ) dan  $p_k + 1 = p_k + 2y$ , bila tidak maka titik selanjutnya adalah ( $x_k + 1, y_k+1$ ) dan  $p_{k+1}=p_k+2y-2x$
- 6) Ulangi langkah nomor 5 untuk menentukan posisi pixel selanjutnya sampai  $x = x_1$  dan  $y = y_1$

Tulis listing program ini menggunakan software MATLAB

```
clc
clear all
point = input('Input Koordinat[ x0 y0 x1 y1]: ');
x0 = point(1);
y0 = point(2);
x1 = point(3);
y1 = point(4);

if (abs(point(4)-point(2)) > abs(point(3)-point(1)))
    x0 = point(2); y0 = point(1); x1 = point(4); y1=point(3);
    token =1;
else
    x0 = point(1); y0 = point(2); x1 = point(3); y1=point(4);
    token = 0;
end
if(x0 >x1)
    temp1 = x0; x0 = x1; x1 = temp1;
    temp2 = y0; y0 = y1; y1 = temp2;
end
dx = abs(x1 - x0) ;
dy = abs(y1 - y0);
sx = sign(x1 - x0);
sy = sign(y1 - y0);
clf, subplot(2,1,1) ,hold on , grid on ,axis([x0-1 x1+1 y0-1 y1+1]);
title('Bresenham Line Generation Algorithm: Point form')
x = x0; y = y0;
param = 2*dy - dx ;
for i = 0:dx-1
    x_coord(i+1) = x;
    y_coord(i+1) = y;
    if (token ==0)
        plot(x,y, 'r*');
    else
        plot(y,x, 'r*');
```

```

end
param = param + 2*dy;
if (param > 0)
    y = y + 1*sy;
    param = param - 2*(dx );
end
x = x + 1*sx;
end
subplot(2,1,2)
if (token == 0)
    plot(x_coord,y_coord,'r-','LineWidth',2);
else
    plot(y_coord,x_coord,'r-','LineWidth',2);
end
grid on
axis([x0-1 x1+1 y0-1 y1+1]);
title('Bresenham Line Generation Algorithm: Line fragment form')

```

Menghasilkan :

