

## **IMPLEMENTASI ALGORITMA DDA PADA PEMROGRAMAN JAVA NETBEANS**

**Apriyanto<sup>1</sup>, Dewinta septiani<sup>2</sup>, Ahmad Ridho Mubarok<sup>3</sup>, Aditya Wahyudi<sup>4</sup>**

1,2,3Program Studi Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

Email: [lapriyy042702@gmail.com](mailto:lapriyy042702@gmail.com), [dewintaseptiani08@gmail.com](mailto:dewintaseptiani08@gmail.com), [adityawahyudi306@gmail.com](mailto:adityawahyudi306@gmail.com), [leverre08@gmail.com](mailto:leverre08@gmail.com)

Abstrak-Zaman sekarang merupakan zaman globalisasi dimana segala aspek kehidupan hampir seluruhnya telah didukung oleh bantuan teknologi yang ada. Dalam berbagai macam pengaplikasian sistem telah dibuat demi memudahkan pekerjaan sehari-hari manusia itu sendiri. Salah satunya ada pada dunia pendidikan. Dimana dunia pendidikan ini membuka celah kemajuan teknologi yang semakin pesat. Namun, tidak dapat dipungkiri pula bahwa terkadang dikalangan mahasiswa masih sulit untuk memahami terlebih menyelesaikan suatu permasalahan teknologi itu sendiri. Jenis pembelajaran yang berbeda akan memberikan pemahaman tersendiri bagi mahasiswa tersebut. Salah satunya dengan mengaplikasian pengcodingan menggunakan Java Netbeans 8.1, JDK 1.8 untuk membentuk suatu garis demi memahami setiap masalah pada Mata Kuliah Grafika Komputer.

**Kata Kunci:** Algoritma DDA, Garis,Grafika Komputer,Java Programming.

*Abstract- The current era is an era of globalization where almost all aspects of life are supported by the help of existing technology. In various applications, systems have been created to facilitate people's daily work. One of them is in the world of education. Where the world of education opens up gaps in increasingly rapid technological progress. However, it cannot be denied that sometimes among students it is still difficult to understand, let alone solve a technological problem itself. Different types of learning will provide students with their own understanding. One of them is by applying coding using Java Netbeans 8.1, JDK 1.8 to form a line to understand each problem in the Computer Graphics Course.*

**Keywords:** DDA algorithm, line, computer graphics, Java programming.

### **1. PENDAHULUAN**

Seiring dengan perkembangan ilmu serta teknologi yang semakin hari semakin berkembang segala aspek kehidupan masyarakat tidak dapat terlepas dari suatu teknologi yang ada saat ini. Dari teknologi yang ada tersebutlah segala informasi dapat tersampaikan dengan baik. penyampaian informasi tersebut dapat diberikan dalam bentuk gambar, tulisan serta suara.Pemberian informasi menggunakan gambar tentunya diperlukan suatu pembentuk dasar untuk menghasilkan gambar yang dapat dipahami sehingga menjadi suatu informasi, pembentukan dasar tersebut berasal dari garis. Namun untuk pemahaman pembentukan garis pada mata kuliah Grafika Komputer ini ada beberapa teori yang dapat dipelajari dan dipahami. Salah satu teori pembentukan garis adalah algoritma garis DDA. Grafika komputer sendiri merupakan suatu mata kuliah wajib yang ada di Jurusan Teknik Informatika. Jurusan yang kita ketahui bahwa pemahamannya lebih dapat mudah diterima menggunakan logika. Logika yang akan digunakan pada pembentukan garis ini adalah menggunakan bahasa Java. Dengan memperhatikan beberapa hal tersebut, penulis akan mencoba membuat sebuah media

### **2. METODE PENELITIAN**

#### **2.1 Algoritma 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).

## 2.2 Titik

Grafik komputer adalah bagian dari ilmu komputer yang berkaitan dengan pembuatan dan manipulasi gambar (visual) secara digital. Untuk itu grafik sangat erat kaitannya dengan titik dan garis sebagai dasar dari manipulasi gambar. Sebuah titik hanya dapat ditentukan oleh letaknya, tetapi tidak mempunyai ukuran (dikatakan tidak berdimensi). Sebuah titik digambarkan dengan memakai tanda noktah, kemudian dibubuh dengan nama titik itu. Nama sebuah titik biasanya menggunakan huruf kapital seperti A, B, C, P, Q atau R. Pada gambar diperlihatkan dua titik, yaitu titik A dan titik P.

## 2.3 Garis

Sebuah garis (dimaksudkan adalah garis lurus) dapat diperpanjang sekehendak kita. Namun mengingat terbatasnya bidang tempat gambar, sebuah garis hanya dilukiskan sebagian saja. Bagian dari garis ini disebut wakil garis. Garis hanya mempunyai ukuran panjang, tetapi tidak mempunyai ukuran lebar. Nama dari sebuah garis dapat ditentukan dengan menyebutkan nama wakil garis itu dengan memakai huruf kecil g, h, k, atau menyebutkan nama segmen garis dari titik pangkal ke titik ujung. Pada gambar diperlihatkan dua buah garis, yaitu garis g dan segmen garis AB.

## 2.4 Piksel

Piksel adalah unsur gambar atau representasi sebuah titik terkecil dalam sebuah gambar grafis yang dihitung per inci. Piksel sendiri berasal dari akronim bahasa Inggris Picture Element yang disingkat menjadi Pixel. Pada ujung tertinggi skala resolusi, mesin cetak gambar berwarna dapat menghasilkan hasil cetak yang memiliki lebih dari 2.500 titik per inci dengan pilihan 16 juta warna lebih untuk setiap inci, dalam istilah komputer berarti gambar seluas satu inci persegi yang bisa ditampilkan pada tingkat resolusi tersebut sepadan dengan 150 juta bit informasi.

## 3. ANALISA DAN PEMBAHASAN

Implementasi Algoritma DDA pada Pemrograman Java Netbeans adalah seperti yang dijabarkan di bawah ini, Java Netbeans yang digunakan ialah versi 8.1 dengan JDK 1.8.

Langkah pembuatan garis menggunakan Algoritma DDA yaitu dengan cara:

1. Tentukan 2 buah titik.
2. Tentukan yang menjadi titik awal ( $X_0, Y_0$ ) dan titik akhir ( $X, Y$  ).
3. Hitung  $Dx$  dan  $Dy$  dimana  $Dx = X_1 - X_0$  dan  $Dy = Y_1 - Y_0$
4. Bandingkan  $Abs(Dx)$  dan  $Abs(Dy)$  Jika  $Abs(Dx) > Abs(Dy)$  maka  $Steps = Abs(Dx)$  bila tidak  $Steps = Abs(Dy)$
5. Hitung penambahan koordinat piksel, yaitu:  $X\_increment = Dx/steps$ , dan  $Y\_increment = Dy/steps$ .
6. Koordinat selanjutnya, yaitu  $X+X\_increment$  dan  $Y+Y\_increment$

7. Posisi piksel ditentukan dengan pembulatan nilai koordinat tersebut.
8. Ulangi langkah 6 dan 7 untuk posisi selanjutnya sampai  $X = X_1$ ,  $Y = Y_1$

### 3.1 Class garis.java

```
package DDA;
import java.awt.Graphics;
public class Garis {
    public void Garisku(Graphics g, int x0, int y0, int x1, int y1) {
        int dx, dy, steps;
        int x_tambah, y_tambah, x, y;
        dx=x1-x0;
        dy=y1-y0;
        if(Math.abs(dx)> Math.abs(dy))
            steps = Math.abs(dx);
        else{
            steps = Math.abs(dy);
            if(x_tambah=dx/steps);
            y_tambah=dy/steps;
            x=x0;
            y=y0;
            g.fillRect(x, y, 1, 1);
            for(int k=1; k<steps; k++){
                x+=x_tambah;
                y+=y_tambah;
                g.fillRect(x, y, 1, 1);
            }
        }
    }
}
```

Gambar 1. Class Garis.java

### 3.2 Main Class PanggilGaris.java

```
package DDA;
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class PanggilGaris extends JPanel {
    public void paintComponent(Graphics g) {
        Garis baru = new Garis();
        g.setColor(Color.BLACK);
        baru.Garisku(g, 50, 50, 70, 100);
    }
    public static void main(String[] args) {
        Garis baru=new Garis();
        JFrame frame = new JFrame("Gambarkan Garis");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.add(new PanggilGaris());
        frame.setSize(400,400);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }
}
```

Gambar 2. Main Class PanggilGaris.java

Pada kedua program tersebut dimasukan langkah-langkah untuk membuat garis, berikut identifikasi langkah beserta kode program yang sesuai dengan langkah-langkah tersebut :

- 1) Tentukan 2 buah titik.

Terdapat 2 buah titik yaitu x dan y

- 2) Tentukan yang menjadi titik awal ( $X_0, Y_0$ ) dan titik akhir ( $X_1, Y_1$ ).

- public void Garisku(Graphics g, int x0, int y0, int x1, int y1){
- baru.Garisku(g, 50, 50, 70, 100);
- artinya titik awal ( $x_0, x_1$ ) titik akhir( $y_1, y_0$ )  $X_0=50, Y_0=50, X_1=70, Y_1=100$

- 3.) Hitung Dx dan Dy dimana  $Dx = X_1 - X_0$  dan  $Dy$

- a.  $= Y_1 - Y_0$
- b.  $dx = x_1 - x_0; dy = y_1 - y_0;$

- c.  $dx = 70 - 50 = 20; dy = 100 - 50 = 50;$

- 4.) Bandingkan Abs(Dx) dan Abs(Dy) Jika Abs(Dx)

Abs(Dy) maka Steps = Abs(Dx) bila tidak Steps = Abs(Dy)  
(Math.abs(dx) > Math.abs(dy))  
{

Jika disubstitusikan :

If (20 > 50){  
Steps = Math.abs(dx);  
}

Else{ steps = Math.abs(dy)}

Karena abs(dy) > abs(dx) maka steps  
= abs(dy) = 50

- 5.) Hitung penambahan koordinat pixel, yaitu:  $X\_increment = dx/steps$ , dan  $Y\_increment = dy/steps$ .

```
x_tambah = dx/steps;    y_tambah = dy/steps;  
x = x0;  
y = y0;
```

Jika disubstitusikan :

x\_tambah = 20/50 = 0,4; y\_tambah = 50/50 = 1;

6.) Koordinat selanjutnya, yaitu  $X+X\_increment$   $Y+Y\_increment$

$x+=x\_tambah;$   $y+=y\_tambah;$

7.) Posisi pixel ditentukan dengan pembulatan nilai koordinat tersebut.

8.) Ulangi langkah 6 dan 7 untuk posisi selanjutnya sampai  $X = X_1$ ,  $Y = Y_1$ .

Pada Main Class PanggilGaris.java ditujuan untuk memanggil hasil dari perhitungan pada Class Garis.java sehingga dapat dilihat hasilnya pada Gambar 3.

### 3.3 Output



Gambar 3. Garis Hasil Program

Gambar diatas merupakan hasil dari perhitungan / implementasi persamaan algoritma DDA yang dijalankan menggunakan Program Java Netbeans.

### 3.4 Perhitungan Secara Manual

Berdasarkan perhitungan manual seperti di bawah ini, maka akan dihasilkan Tabel 1 yaitu tabel pembentukan garis output pada Gambar 3.

Perhitungan Manual :

1. Tentukan 2 buah titik.

Diketahui titik  $x,y$

2. Tentukan yang menjadi titik awal ( $X_0, Y_0$ ) dan

titik akhir (X1,Y1).

Titik awal = A(50,50)

Titik Akhir = B(70,100)

## 4. KESIMPULAN

Berdasarkan penjabaran di atas dapat di tarik kesimpulan bahwa Algoritma Digital Differential Analyzer (DDA) dapat diimplementasikan untuk pembuatan garis menggunakan Bahasa Pemrograman Java dan tools NetBeans, kode pemrograman yang ada di dalam program tersebut sesuai dengan urutan langkah-langkah pembuatan garis Digital Differential Analyzer (DDA).

## DAFTAR PUSTAKA

- Pujiyanta, Ardi, Ir. MT., 2011, Grafika Komputer, Universitas Ahmad Dahlan, Yogyakarta.
- Sutopo, Aries Hadi. Pengantar Grafika komputer. Gava Media, Yogyakarta. 2002
- Agustama, M., & Handayaningsih, S. (2014). Media Pembelajaran Algoritma Garis dan Lingkaran Berbasis Multimedia. Jurnal Sarjana Teknik Informatika, Volume 2 Nomor 1.
- [http://eprints.dinus.ac.id/6327/1/02\\_AlgoritmaGaris.pdf](http://eprints.dinus.ac.id/6327/1/02_AlgoritmaGaris.pdf)
- Diunduh tanggal 16 Desember 2023 pukul 19.00 WIB <https://id.wikipedia.org/wiki/Piksel>.
- Diakses 16 Desember 2023 pukul 19.00 WIB <http://solidadventure07.blogspot.com/2016/02/algoritma-pembentuk-garis-pada-graphic.html?m=1>
- Diakses 16 Desember 2023 pukul 22.00 WIB <http://nurmalitaca.blogspot.co.id/2017/11/hallo-perkenalkan-aku-nurmaalita-cindy.html?m=1> Diakses tanggal 16 Desember 2023 pukul 22.00 WIB.