LAPORAN TUGAS KELOMPOK

PENGUJIAN PEMROGRAMAN GRAFIKA KOMPUTER

Disusun sebagai salah satu tugas

mata kuliah Grafika Komputer



Angga Kresnabayu – 140810160001

Fauzi Faruq Nabbani – 140810160007

Muhammad Jordiansyah – 140810160040

Adryan Luthfi Faiz – 140810160049

Muhammad Islam Taufikurahman – 140810160062

Patricia Joanne – 140810160065

Dikumpulkan tanggal

30 September 2018

PROGRAM STUDI S-1 TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

2018

**BAB I**

**PENDAHULUAN**

* 1. Latar Belakang

Grafika Komputer adalah proses untuk menciptakan suatu gambar berdasarkan deskripsi obyek maupun latar belakang yang terkandung pada gambar tersebut.  Grafika komputer merupakan teknik untuk membuat gambar obyek sesuai dengan obyek tersebut di alam nyata. Grafika komputer bertujuan menghasilkan gambar/citra dengan primitif-primitif geometri seperti garis, lingkaran, dsb.  Primitif-primitif geometri tersebut memerlukan data deskriptif untuk melukis elemen-elemen gambar seperti koordinat titik, panjang garis, jari-jari lingkaran, tebal garis, warna, dan lain-lain.

* 1. Rumusan Masalah

Rumusan masalah dari tugas grafika computer ini adalah sebagai berikut:

1. Bagaimana proses instalasi IDE Netbeans pada sistem operasi Windows?

2. Bagaimana cara membuat *project* pada IDE Netbeans?

3. Bagaimana cara membuat objek berupa garis dan bentuk pada IDE Netbeans?

* 1. Tujuan

Berdasarkan permasalahan yang ada, maka laporan ini dibuat dengan tujuan sebagai berikut:

1. Menginstal IDE Netbeans dan menggunakannya tanpa ada *error*.
2. Membuat *project* pada IDE Netbeans dan membuat kodingan tanpa ada masalah.
3. Membuat objek berupa garis dan bentuk yang dilengkapi dengan warna baik *border* maupun dalam bentuknya.

**BAB II**

**PEMBAHASAN**

2.1. Kajian Keilmuan

Membuat program untuk menggambar bentuk di Java sama seperti pada bahasa lainnya. Hal yang perlu kita ketahui adalah *class* apa yang perlu dipanggil dan cara mengkoding bentuk yang diinginkan. Jika kita sudah tahu bagaimana cara memanggil *class* dan cara mengkoding bentuk yang diinginkan, kita dapat menjadikannya template untuk kita gunakan dalam program lain. Kelebihan dari bahasa Java adalah penggunaannya cukup mudah digunakan untuk mengkoding bentuk karena adanya dukungan GUI.

2.2. Proses Instalasi Program

2.2.1. Instalasi Netbeans

1. Sebelum menggunakan Netbeans, *install* Java terlebih dahulu dengan mengunduhnya di <https://java.com/en/download/>

2. Klik 2x pada *installer* Java tersebut, klik *next* terus hingga instalasi dimulai.

3. Cari “Edit the system environment variables” dan pilih Environment Variables

4. Pada “System Variables”, klik Path dan klik Edit…

5. Klik New dan tambahkan lokasi penginstalan Java tadi, contohnya C:\Program Files\Java\jdk1.8.0\_111\bin

6. Klik OK maka Java sudah siap dipakai

7. Setelah itu, buka situs <https://netbeans.org/downloads>

8. Pilih Netbeans dengan Java SE dan unduh *software* tersebut

9. Klik 2x pada *installer* yang telah diunduh

10. Klik *next* terus hingga instalasi dimulai

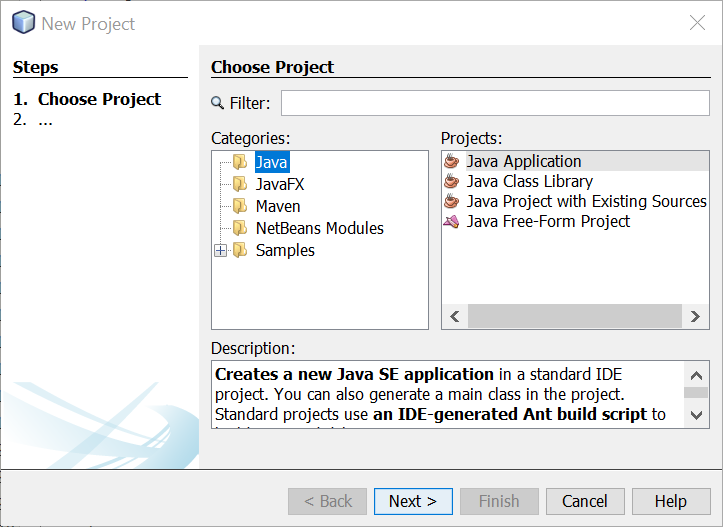
11. Tunggu instalasi selesai dan Netbeans siap dipakai

2.2.2. Cara Membuat *Project* pada Netbeans

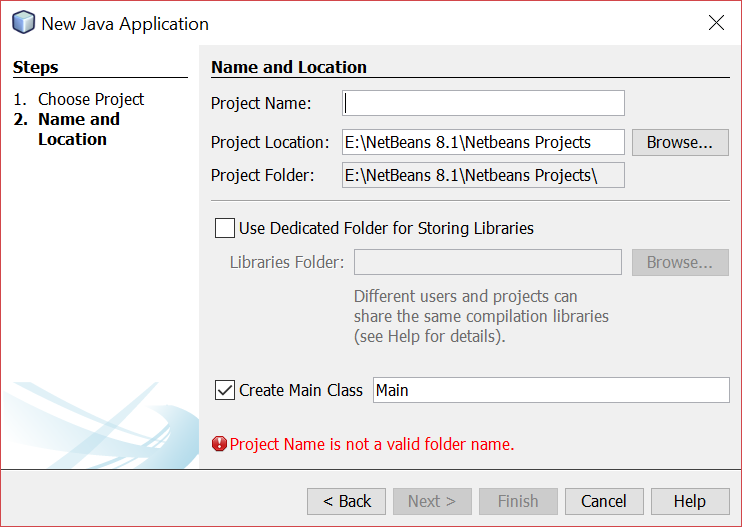
1. Buka aplikasi Netbeans

2. Klik File > New Project

3. Pilih Java dan Java Application



4. Beri nama *project* yang akan kita buat lalu klik Finish



2.3. Hasil

Pada dasarnya program menggambar bentuk yang akan kita buat tersusun sebagai berikut:

package drawingcanvas;

import java.awt.\*;

import javax.swing.\*;

public class DrawingCanvas extends Canvas {

    //Constructor

    public DrawingCanvas(){

    }

    //Masukkan fungsi menggambar shape di sini

    @Override

    public void paint (Graphics graphics){

    }

    public static void main(String[] args) {

        //Membuat canvas baru

        DrawingCanvas canvas = new DrawingCanvas();

        //Membuat JFrame

        JFrame frame = new JFrame();

        //Mengatur besar frame

        frame.setSize(500, 500);

        //Buat agar tombol x bisa dipakai untuk menutup frame

        frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        //Menggambar ke canvas

        frame.getContentPane().add(canvas);

        //Menampilkan frame

        frame.setVisible(true);

    }

}

Setelah diisi maka jadinya seperti ini:

package drawingcanvas;

import java.awt.\*;

import javax.swing.\*;

public class DrawingCanvas extends Canvas {

    //Constructor

    Polygon p1, p2;

public DrawingCanvas(){

         //Menggambar segitiga

p1 = new Polygon();

p1.addPoint(50, 500);

p1.addPoint(100, 350);

p1.addPoint(150, 500);

//Menggambar segilima

p2 = new Polygon();

p2.addPoint(225, 500);

p2.addPoint(200, 450);

p2.addPoint(250, 400);

p2.addPoint(300, 450);

p2.addPoint(275, 500);

    }

    //Masukkan fungsi menggambar shape di sini

    @Override

    public void paint (Graphics graphics){

         //Menggambar garis

graphics.setColor(Color.WHITE);

graphics.drawLine(250, 120, 400, 380);

//Menggambar persegi panjang

graphics.setColor(Color.BLUE);

graphics.fillRect(100, 100, 100, 200); //FILL

graphics.setColor(Color.RED);

graphics.drawRect(100, 100, 100, 200); //BORDER

//Menggambar lingkaran

graphics.setColor(Color.RED);

graphics.fillOval(380, 100, 150, 150); //FILL

graphics.setColor(Color.YELLOW);

graphics.drawOval(380, 100, 150, 150); //BORDER

//Draw a Polygon

graphics.setColor(Color.BLACK);

graphics.drawPolygon(p1);

graphics.setColor(Color.GREEN);

graphics.fillPolygon(p1);

//Draw a Polygon

graphics.setColor(Color.MAGENTA);

graphics.drawPolygon(p2);

graphics.setColor(Color.ORANGE);

graphics.fillPolygon(p2);

    }

    public static void main(String[] args) {

        //Membuat canvas baru

        DrawingCanvas canvas = new DrawingCanvas();

        //Membuat JFrame

        JFrame frame = new JFrame();

        //Mengatur besar frame

        frame.setSize(500, 500);

        //Buat agar tombol x bisa dipakai untuk menutup frame

        frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        //Menggambar ke canvas

        frame.getContentPane().add(canvas);

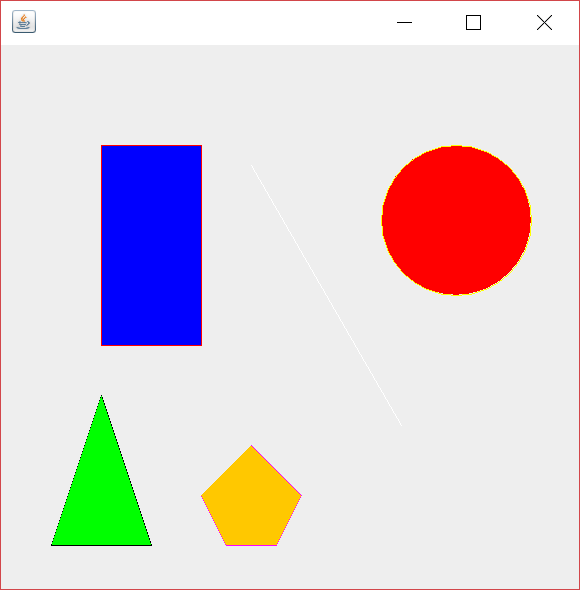
        //Menampilkan frame

        frame.setVisible(true);

    }

}

Screenshot setelah di*run*:



Posisinya memang tidak rapi pada *screenshot* tersebut, namun hal tersebut bisa diperbaiki dengan mengatur titik-titiknya. Begitu pula dengan warnanya, cukup me*replace* warnanya saja. Tidak cuma warna dalam bentuk, tetapi kita juga bisa mengubah warna *border* dari bentuk tersebut.

**BAB III**

**PENUTUP**

3.1. Kesimpulan

Pada prinsipnya program menggambar bentuk menggunakan Bahasa Java tidak jauh berbeda dengan bahasa lainnya. Bahasa Java bahkan memiliki kelebihan yang tidak semua bahasa miliki yaitu dukungan GUI. Dengan GUI, hasil pemrograman dapat tervisualisasikan dengan baik.

3.2. Saran

Sejauh ini program yang kelompok kami buat masih jauh dari sempurna dikarenakan keterbatasan kemampuan dari kami dan kurangnya sumber referensi untuk meneliti lebih dalam.

Program yang telah dibuat sebenarnya masih bisa dikembangkan lagi lebih dalam. Jika pada program yang telah dibuat hanya dapat memunculkan semua bentuk, maka ke depannya kami bisa membuat program yang hanya memunculkan bentuk-bentuk yang ingin ditampilkan saja. Selain itu, dapat ditambahkan juga fitur untuk memilih warna dan posisi sendiri lewat hasil *run* tanpa mengubah isi kodingan.

**DAFTAR PUSTAKA**

Ferguson, Ian. 2002. *Computer Graphics via Java*. Durham: Ab-libris Ltd.

<https://hubpages.com/technology/Programming-Basic-2D-Shapes-in-Java-Using-Graphics-and-JFrame-Classes>

<https://www.temukanpengertian.com/2013/08/pengertian-grafika-komputer.html>

<https://www.youtube.com/watch?v=e5IxPM0J2yA>

<https://www.youtube.com/watch?v=fzFmfbXghj4>

<https://www.youtube.com/watch?v=y7YbQrrcpAQ>

<https://www.youtube.com/watch?v=YsN9g14Rlw8>

<https://www.youtube.com/watch?v=ZHShjRdg5wM>