

**LAPORAN PRAKTIKUM
PEMGORAMAN OBJEK 2**

**MODUL 6
GRAPHICS 2D**

DISUSUN OLEH :

**Ramdani 2350081055
B**



**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN INFORMATIKA
UNIVERSITAS JENDERAL ACHMAD YANI
2025**

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	ii
BAB I. HASIL PRAKTIKUM	1
1.1. Program 6.1	1
1.2. Program 6.2	2
1.3. Program 6.3	3
1.4. Program 6.4	6
1.5. Program 6.5	7
1.6. Program 6.6	9
1.7. Program 6.7	11
1.8. Program 6.8	14
1.9. Program 6.9	15
BAB II. TUGAS PRAKTIKUM	24
2.1. Tugas 6.1	24
2.2. Tugas 6.2	26
2.3. Tugas Gabungan 6-1 dan 6-2	29
BAB III. TUGAS AKHIR	33
3.1. Tugas Akhir 1	33
3.2. Tugas Akhir 2	36
BAB IV. KESIMPULAN	39

DAFTAR GAMBAR

gambar I- 1 ColorJPanel.....	2
gambar I- 2 ShowColors.....	3
gambar I- 3 ShowColors2JFrame	5
gambar I- 4 ShowColors2JFrame	5
gambar I- 5 ShowColors2.....	6
gambar I- 6 ShowColors2.....	6
gambar I- 7 Program65	8
gambar I- 8 Program66	11
gambar I- 9 ShapesJPanel.....	14
gambar I- 10 Shapes.....	15
gambar I- 11 StrokeAndFill.....	21
gambar I- 12 StrokeAndFill.....	21
gambar I- 13 StrokeAndFill.....	22
gambar I- 14 StrokeAndFill.....	22
gambar I- 16 Tugas6_2_ButtonWarna.....	28
gambar I- 17 Tugas6_2_ButtonWarna.....	28
gambar I- 18 Tugas6_2_ButtonWarna.....	29
gambar I- 19 Tugas6_Gabungan6_1_6_2	32
gambar I- 20 Tugas Akhir 1 Blue	35
gambar I- 21 Tugas Akhhir 1 Yellow	35
gambar I- 22 StarCirclePattern	38

BAB I. HASIL PRAKTIKUM

1.1.Program 6.1

1.1.A. Source Code

```
import java.awt.Graphics;
import java.awt.Color;
import javax.swing.JPanel;

public class ColorJPanel extends JPanel {
    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(g);

        // Mengatur latar belakang panel menjadi putih
        this.setBackground(Color.WHITE);

        // Warna merah (RGB)
        g.setColor(new Color(255, 0, 0));
        g.fillRect(15, 25, 100, 20);
        g.drawString("Current RGB: " + g.getColor(), 130, 40);

        // Warna hijau kekuningan (float RGB)
        g.setColor(new Color(0.50f, 0.75f, 0.0f));
        g.fillRect(15, 50, 100, 20);
        g.drawString("Current RGB: " + g.getColor(), 130, 65);

        // Warna biru
        g.setColor(Color.BLUE);
        g.fillRect(15, 75, 100, 20);
        g.drawString("Current RGB: " + g.getColor(), 130, 90);

        // Warna magenta dan menampilkan nilai RGB-nya
        Color color = Color.MAGENTA;
        g.setColor(color);
```

```

        g.fillRect(15, 100, 100, 20);
        g.drawString("RGB values: " + color.getRed() + ", "
            + color.getGreen() + ", " + color.getBlue(), 130, 115);
    }
}

```

1.1.B. Hasil



gambar 1- 1 ColorJPanel

1.1.C. Analisa

Pada Latihan program 1 ini adalah ColorJPanel di atas merupakan implementasi dari komponen JPanel dalam Java Swing yang digunakan untuk mendemonstrasikan penggunaan berbagai warna menggunakan kelas Color. Di dalam metode paintComponent, latar belakang panel diatur menjadi putih, kemudian beberapa persegi panjang digambar menggunakan warna berbeda: merah dengan nilai RGB eksplisit, hijau kekuningan menggunakan nilai float, biru menggunakan konstanta bawaan Color.BLUE, dan magenta dengan detail nilai RGB ditampilkan secara eksplisit.

1.2.Program 6.2

1.2.A. Source Code

```

import javax.swing.JFrame;

public class ShowColors {
    public static void main(String[] args) {
        // Membuat frame dengan judul
    }
}

```

```

JFrame frame = new JFrame("Using colors");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Menambahkan panel kustom ke frame
ColorJPanel colorJPanel = new ColorJPanel();
frame.add(colorJPanel);

// Mengatur ukuran dan menampilkan frame
frame.setSize(400, 180);
frame.setVisible(true);
}
}

```

1.2.B. Hasil



gambar I- 2 ShowColors

1.2.C. Analisa

Program ShowColors di atas adalah aplikasi Java Swing sederhana yang menampilkan antarmuka grafis untuk mendemonstrasikan penggunaan warna. Program ini membuat sebuah JFrame dengan judul "Using colors" dan menambahkan panel kustom ColorJPanel, yang sebelumnya telah dirancang untuk menggambar beberapa kotak berwarna dan menampilkan nilai RGB-nya.

1.3.Program 6.3

1.3.A. Source Code

```
import java.awt.BorderLayout;
```

```

import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JColorChooser;
import javax.swing.JPanel;

public class ShowColors2JFrame extends JFrame {
    private JButton changeColorJButton;
    private Color color = Color.LIGHT_GRAY;
    private JPanel colorJPanel;

    public ShowColors2JFrame() {
        super("Using JColorChooser");

        colorJPanel = new JPanel();
        colorJPanel.setBackground(color);

        changeColorJButton = new JButton("Change Color");
        changeColorJButton.addActionListener(
            new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent event) {
                    color = JColorChooser.showDialog(
                        ShowColors2JFrame.this,
                        "Choose a color", color
                    );

                    if (color == null) {
                        color = Color.LIGHT_GRAY;
                    }
                }
            }
        );
    }
}

```

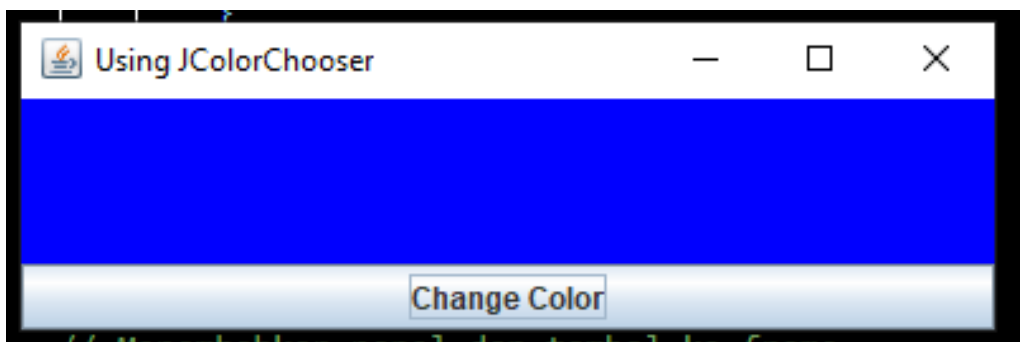
```

        colorJPanel.setBackground(color);
    }
}
);

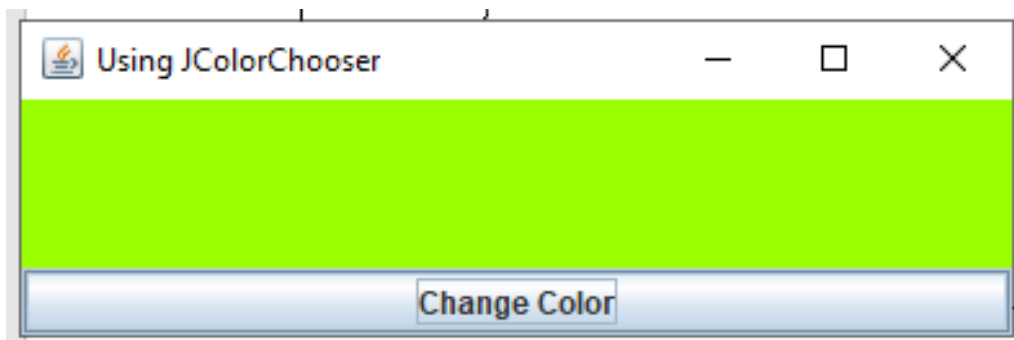
add(colorJPanel, BorderLayout.CENTER);
add(changeColorJButton, BorderLayout.SOUTH);
setSize(400, 130);
setVisible(true);
}
}

```

1.3.B. Hasil



gambar I- 3 ShowColors2JFrame



gambar I- 4 ShowColors2JFrame

1.3.C. Analisa

Program ShowColors2JFrame merupakan aplikasi Java GUI yang mendemonstrasikan penggunaan komponen JColorChooser untuk memilih warna secara interaktif. Program ini membuat jendela (JFrame) berjudul "Using JColorChooser" yang berisi sebuah panel (JPanel) dengan latar belakang warna awal

abu-abu muda dan sebuah tombol "Change Color" di bagian bawah

1.4.Program 6.4

1.4.A. Source Code

```
import javax.swing.JFrame;

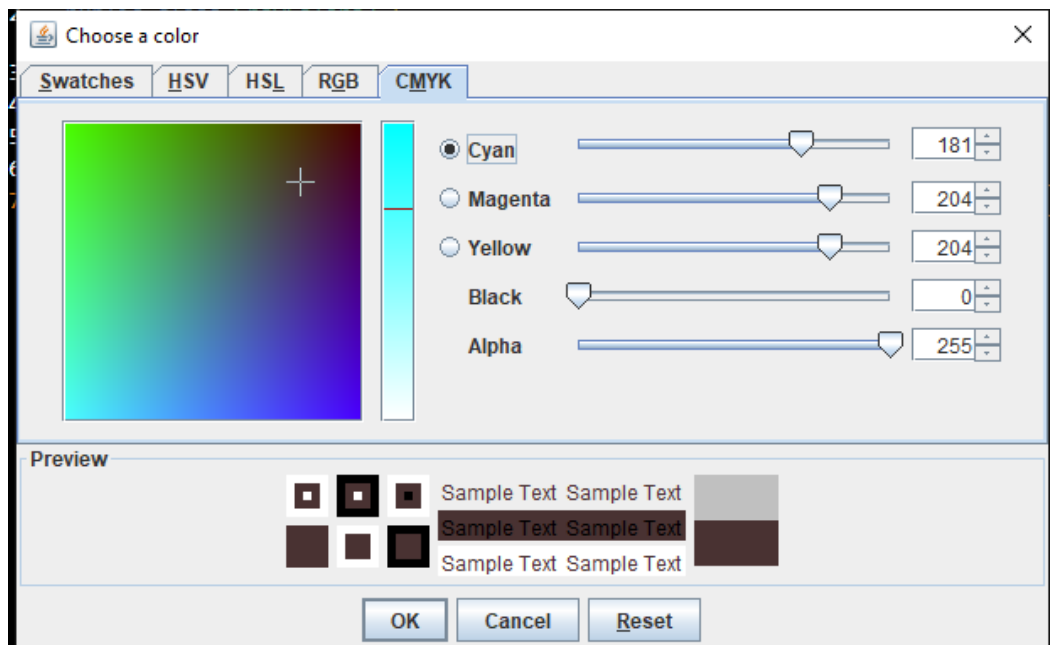
public class ShowColors2 {

    public static void main( String args[] ) {

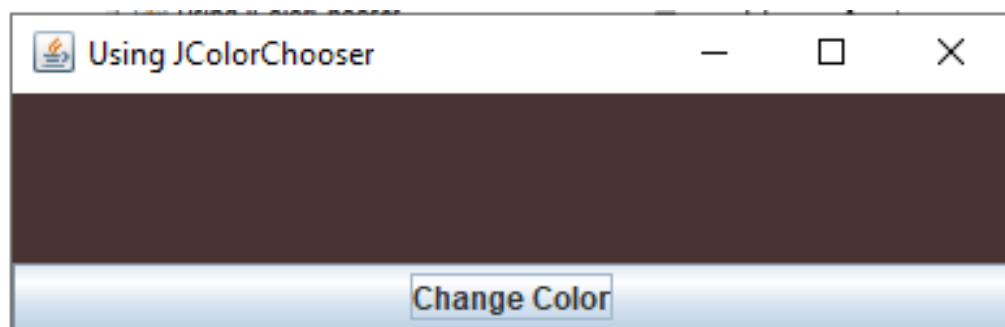
        ShowColors2JFrame application = new ShowColors2JFrame();
        application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

}
```

1.4.B. Hasil



gambar I- 5 ShowColors2



gambar I- 6 ShowColors2

1.4.C. Analisa

Program ShowColors2 berfungsi sebagai kelas utama (main class) yang menjalankan aplikasi GUI ShowColors2JFrame, di mana pengguna dapat memilih warna melalui komponen JColorChooser.

1.5. Program 6.5

1.5.A. Source Code

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

public class Program65 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Program 6-5: Gambar Bentuk Terisi
Warna");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);

        // Tambahkan panel custom
        frame.add(new GambarPanel());

        frame.setLocationRelativeTo(null); // posisi tengah layar
        frame.setVisible(true);
    }
}

// Kelas panel khusus untuk menggambar bentuk
class GambarPanel extends JPanel {
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);

        // Polygon
```

```

int[] xPoints = {50, 100, 150, 100};
int[] yPoints = {50, 20, 50, 100};
g.setColor(Color.MAGENTA);
g.fillPolygon(xPoints, yPoints, xPoints.length);

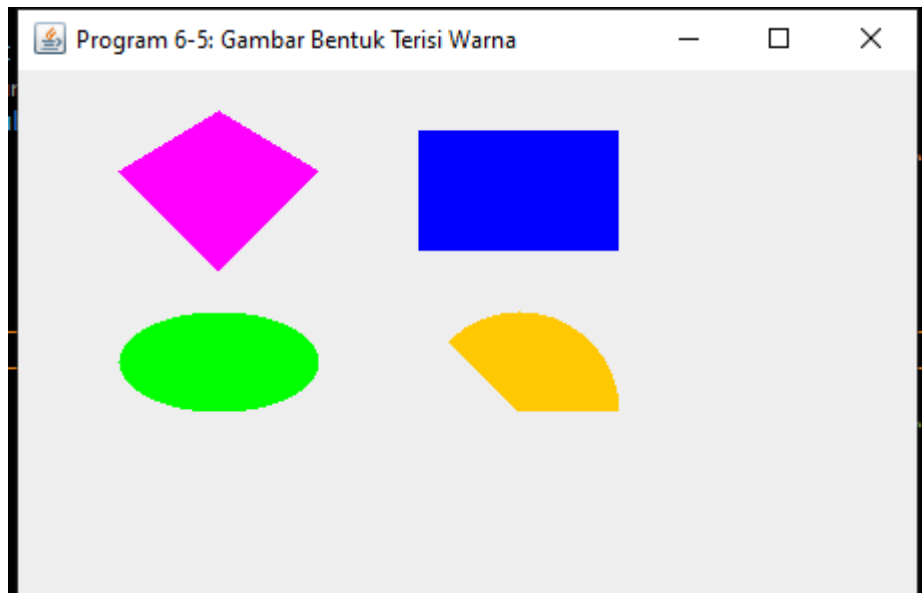
// Rectangle
g.setColor(Color.BLUE);
g.fillRect(200, 30, 100, 60);

// Oval
g.setColor(Color.GREEN);
g.fillOval(50, 120, 100, 50);

// Arc
g.setColor(Color.ORANGE);
g.fillArc(200, 120, 100, 100, 0, 135); // seperti pie chart
}
}

```

1.5.B. Hasil



gambar I- 7 Program65

1.5.C. Analisa

Program Program65 adalah aplikasi Java Swing yang menggambar berbagai bentuk

2D berwarna di dalam sebuah panel khusus (JPanel). Aplikasi ini membuat jendela (JFrame) dengan ukuran 400x300 piksel dan menempatkan panel GambarPanel sebagai kontennya. Di dalam GambarPanel, metode paintComponent digunakan untuk menggambar empat jenis bentuk: sebuah polygon berbentuk wajik ungu, sebuah persegi panjang biru, sebuah oval hijau, dan sebuah arc oranye menyerupai irisan pie.

1.6.Program 6.6

1.6.A. Source Code

```
import javax.swing.*;
import java.awt.*;
import java.awt.geom.Line2D;

public class Program66 extends JPanel {

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);

        Graphics2D g2 = (Graphics2D) g;

        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                             RenderingHints.VALUE_ANTIALIAS_ON);

        g2.setFont(new Font("SansSerif", Font.BOLD, 16));
        g2.setColor(Color.DARK_GRAY);
        g2.drawString("Program 6.6 - Shape Drawer", 50, 20);

        //draw Rectangle.2D
        g2.setColor(new Color(70, 130, 180));
        g2.fillRect(50, 30, 100, 50);
        g2.setColor(Color.BLACK);
        g2.drawRect(50, 30, 100, 50);
```

```

        //draw RoundedRectangle2D.Double
        g2.setColor(new Color(244, 164, 96));
        g2.fillRoundRect(50, 90, 100, 50, 30, 30);
        g2.setColor(Color.BLACK);
        g2.drawRoundRect(50, 90, 100, 50, 30, 30);

        //draw Arc2D.Double
        g2.setColor(new Color(60, 179, 113));
        g2.fillArc(50, 150, 100, 50, 90, 180);
        g2.setColor(Color.BLACK);
        g2.drawArc(50, 150, 100, 50, 90, 180);

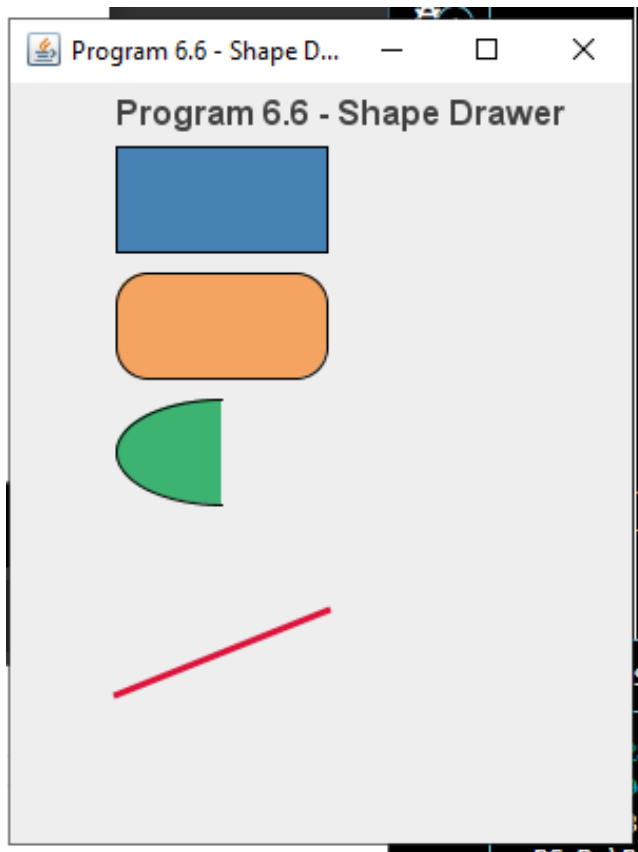
        //draw Line2D.Double
        g2.setColor(new Color(220, 20, 60));
        g2.setStroke(new BasicStroke(3));
        Line2D.Double line = new Line2D.Double(50, 290, 150, 250);
        g2.draw(line);
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Program 6.6 - Shape Drawer");
        Program66 panel = new Program66();

        frame.add(panel);
        frame.setSize(300, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }
}

```

1.6.B. Hasil



gambar 1- 8 Program66

1.6.C. Analisa

Program Program66 adalah aplikasi Java Swing yang menampilkan beberapa bentuk geometris 2D berwarna dengan gaya yang lebih halus dan profesional menggunakan Graphics2D. Dengan mengaktifkan anti-aliasing, program memastikan tepian bentuk-bentuk tersebut terlihat halus. Aplikasi ini menggambar judul teks, kemudian empat bentuk: persegi panjang berwarna biru keabu-abuan, persegi panjang dengan sudut membulat berwarna oranye, arc berwarna hijau (setengah oval melengkung), dan sebuah garis miring merah dengan ketebalan 3 piksel.

1.7.Program 6.7

1.7.A. Source Code

```
import java.awt.Color;
import java.awt.Graphics;
import java.awt.BasicStroke;
import java.awt.GradientPaint;
import java.awt.TexturePaint;
```

```

import java.awt.Rectangle;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import java.awt.geom.Rectangle2D;
import java.awt.geom.RoundRectangle2D;
import java.awt.geom.Arc2D;
import java.awt.geom.Line2D;
import java.awt.image.BufferedImage;
import javax.swing.JPanel;

public class ShapesJPanel extends JPanel {
    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D) g;

        // Gradien warna biru ke kuning pada ellipse
        g2d.setPaint(new GradientPaint(5, 30, Color.BLUE, 35, 100,
Color.YELLOW, true));
        g2d.fill(new Ellipse2D.Double(5, 30, 65, 100));

        // Persegi panjang merah dengan garis tebal
        g2d.setPaint(Color.RED);
        g2d.setStroke(new BasicStroke(10.0f));
        g2d.draw(new Rectangle2D.Double(80, 30, 65, 100));

        // Gambar tekstur berbasis BufferedImage
        BufferedImage buffImage = new BufferedImage(10, 10,
BufferedImage.TYPE_INT_RGB);
        Graphics2D gg = buffImage.createGraphics();
        gg.setColor(Color.YELLOW);
        gg.fillRect(0, 0, 10, 10);
        gg.setColor(Color.BLACK);

```

```

        gg.drawRect(1, 1, 6, 6);
        gg.setColor(Color.BLUE);
        gg.fillRect(1, 1, 3, 3);
        gg.setColor(Color.RED);
        gg.fillRect(4, 4, 3, 3);

        // RoundedRectangle dengan tekstur
        g2d.setPaint(new TexturePaint(buffImage, new Rectangle(10, 10)));
        g2d.fill(new RoundedRectangle2D.Double(155, 30, 75, 100, 50, 50));

        // Arc putih dengan stroke tebal
        g2d.setPaint(Color.WHITE);
        g2d.setStroke(new BasicStroke(6.0f));
        g2d.draw(new Arc2D.Double(240, 30, 75, 100, 0, 270, Arc2D.PIE));

        // Garis hijau polos
        g2d.setPaint(Color.GREEN);
        g2d.setStroke(new BasicStroke(1.0f));
        g2d.draw(new Line2D.Double(395, 30, 320, 150));

        // Garis putus-putus kuning
        float[] dashes = {10};
        g2d.setPaint(Color.YELLOW);
        g2d.setStroke(new BasicStroke(4, BasicStroke.CAP_ROUND,
        BasicStroke.JOIN_ROUND, 10, dashes, 0));
        g2d.draw(new Line2D.Double(320, 30, 395, 150));
    }
}

```

```

import javax.swing.JFrame;

public class Shapes {
    public static void main(String[] args) {

```

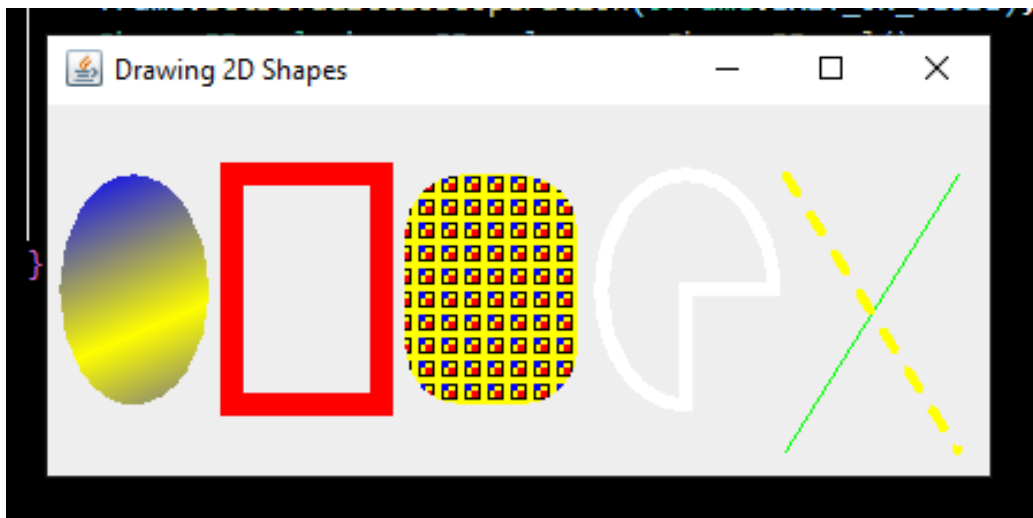


```

JFrame frame = new JFrame("Drawing 2D Shapes");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ShapesJPanel shapesJPanel = new ShapesJPanel();
frame.add(shapesJPanel);
frame.setSize(425, 200);
frame.setLocationRelativeTo(null);
frame.setVisible(true);
}
}

```

1.7.B. Hasil



gambar I- 9 ShapesJPanel

1.7.C. Analisa

Program ke 7 ini tidak bisa dijalankan tanpa ada nya kelas Main, karena program ini hanya untuk Custom JPanel, bukan aplikasi java yang bisa dijalankan secara langsung tanpa adanya kelas Main.

1.8.Program 6.8

1.8.A. Source Code

```

import javax.swing.JFrame;

public class Shapes {
    public static void main(String[] args) {

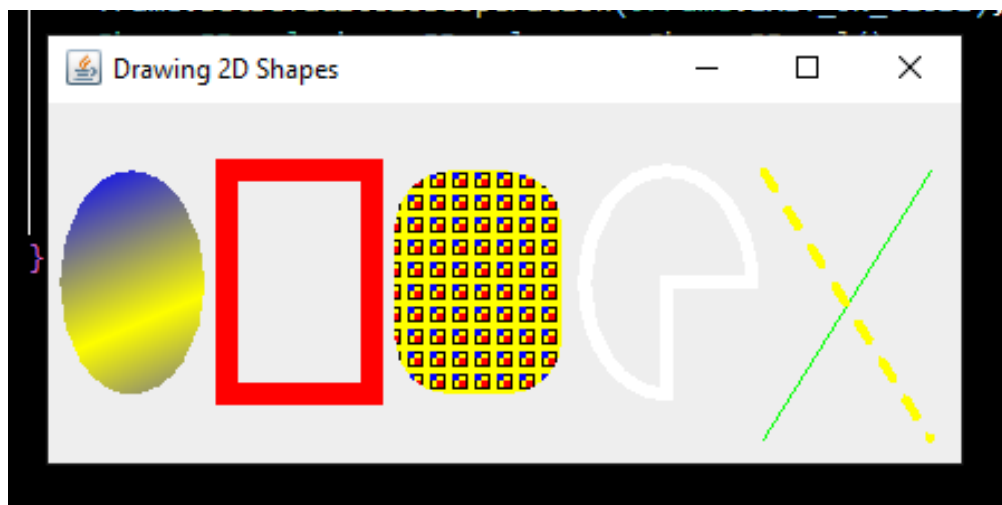
```

```

JFrame frame = new JFrame("Drawing 2D Shapes");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ShapesJPanel shapesJPanel = new ShapesJPanel();
frame.add(shapesJPanel);
frame.setSize(425, 200);
frame.setLocationRelativeTo(null);
frame.setVisible(true);
    }
}

```

1.8.B. Hasil



gambar I- 10 Shapes

1.8.C. Analisa

Program ke 8 ini berfungsi sebagai program utama untuk menampilkan window GUI menggunakan JFrame. Di dalam metode main, sebuah JFrame dibuat dengan judul "Drawing 2D Shapes", kemudian disetel agar aplikasi berhenti saat jendela ditutup (EXIT_ON_CLOSE). Selanjutnya, objek ShapesJPanel yang berisi metode paintComponent untuk menggambar bentuk-bentuk 2D ditambahkan ke frame tersebut. Ukuran frame disetel menjadi 425x200 piksel, dan posisi frame diatur agar muncul di tengah layar dengan setLocationRelativeTo(null).

1.9. Program 6.9

1.9.A. Source Code

```

import java.awt.*;
import java.awt.event.*;

```

```

import java.awt.geom.*;
import java.awt.image.BufferedImage;
import javax.swing.*.*;
import java.awt.font.GlyphVector;

public class StrokeAndFill extends JFrame implements ItemListener {
    private JLabel primLabel, lineLabel, paintLabel, strokeLabel;
    private ShapePanel display;
    private static JComboBox<String> primitive, line, paint, stroke;
    public static boolean noD = false;

    public StrokeAndFill() {
        super("Stroke and Fill Example");
        initUI();
    }

    private void initUI() {
        setLayout(new BorderLayout());

        JPanel controlPanel = new JPanel(new GridLayout(2, 4, 10, 5));
        Font newFont = new Font("SansSerif", Font.BOLD, 14);

        primLabel = createLabel("Primitive", newFont);
        controlPanel.add(primLabel);

        lineLabel = createLabel("Lines", newFont);
        controlPanel.add(lineLabel);

        paintLabel = createLabel("Paints", newFont);
        controlPanel.add(paintLabel);

        strokeLabel = createLabel("Rendering", newFont);
        controlPanel.add(strokeLabel);

        primitive = createComboBox(new String[]{"Rectangle", "Ellipse", "Text"},
newFont);

```

```

        controlPanel.add(primitive);

        line = createComboBox(new String[]{"Thin", "Thick", "Dashed"}, newFont);
        controlPanel.add(line);

        paint = createComboBox(new String[]{"Red", "Gradient", "Texture"}, newFont);
        controlPanel.add(paint);

        stroke = createComboBox(new String[]{"Stroke", "Fill", "Stroke & Fill"},
newFont);
        controlPanel.add(stroke);

        add(controlPanel, BorderLayout.NORTH);

        display = new ShapePanel();
        display.setBackground(Color.WHITE);
        add(display, BorderLayout.CENTER);

        display.renderShape();
    }

    private JLabel createLabel(String text, Font font) {
        JLabel label = new JLabel(text, JLabel.CENTER);
        label.setFont(font);
        return label;
    }

    private JComboBox<String> createComboBox(String[] items, Font font) {
        JComboBox<String> combo = new JComboBox<>(items);
        combo.setFont(font);
        combo.addItemListener(this);
        return combo;
    }

    @Override
    public void itemStateChanged(ItemEvent e) {

```

```

        if (e.getStateChange() == ItemEvent.SELECTED) {
            display.renderShape();
        }
    }

    public static void main(String[] args) {
        if (args.length > 0 && args[0].equals("-no2d")) {
            noD = true;
        }

        SwingUtilities.invokeLater() -> {
            StrokeAndFill frame = new StrokeAndFill();
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setSize(700, 450);
            frame.setLocationRelativeTo(null);
            frame.setVisible(true);
        });
    }

    class ShapePanel extends JPanel {
        public void renderShape() {
            repaint();
        }

        @Override
        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            if (!noD) {
                Graphics2D g2 = (Graphics2D) g;
                g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
                RenderingHints.VALUE_ANTIALIAS_ON);

                Shape shape;
                String selected = (String) primitive.getSelectedItem();
                if ("Rectangle".equals(selected)) {
                    shape = new Rectangle2D.Double(100, 50, 200, 100);

```

```

    } else if ("Ellipse".equals(selected)) {
        shape = new Ellipse2D.Double(100, 50, 200, 100);
    } else {
        Font font = new Font("Serif", Font.BOLD, 24);
        String text = "Ramdani\nNIM : 2350081055";

        // Multiline text render manual
        shape = null;
        String[] lines = text.split("\n");
        int y = 100;
        for (String line : lines) {
            GlyphVector gv = font.createGlyphVector(g2.getFontRenderContext(),
line);

            Shape lineShape = gv.getOutline(100, y);
            if (shape == null) {
                shape = lineShape;
            } else {
                Area area = new Area(shape);
                area.add(new Area(lineShape));
                shape = area;
            }
            y += 40; // Spasi antar baris
        }
    }

    String paintType = (String) paint.getSelectedItemAt();
    if ("Red".equals(paintType)) {
        g2.setPaint(Color.RED);
    } else if ("Gradient".equals(paintType)) {
        g2.setPaint(new GradientPaint(100, 50, Color.BLUE, 300, 150,
Color.YELLOW, true));
    } else {
        BufferedImage bi = new BufferedImage(10, 10,
BufferedImage.TYPE_INT_RGB);
        Graphics2D big = bi.createGraphics();
        big.setColor(Color.YELLOW);

```

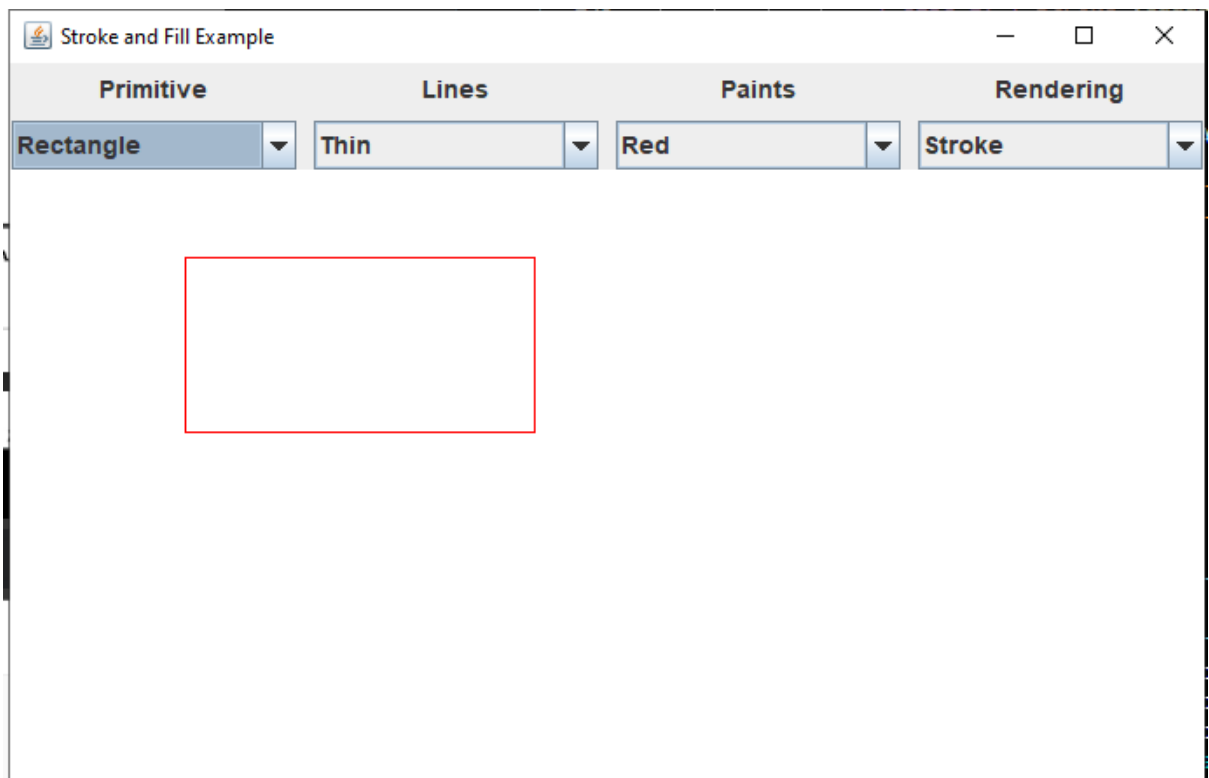
```

        big.fillRect(0, 0, 10, 10);
        big.setColor(Color.BLUE);
        big.fillRect(1, 1, 5, 5);
        big.dispose();
        g2.setPaint(new TexturePaint(bi, new Rectangle(10, 10)));
    }

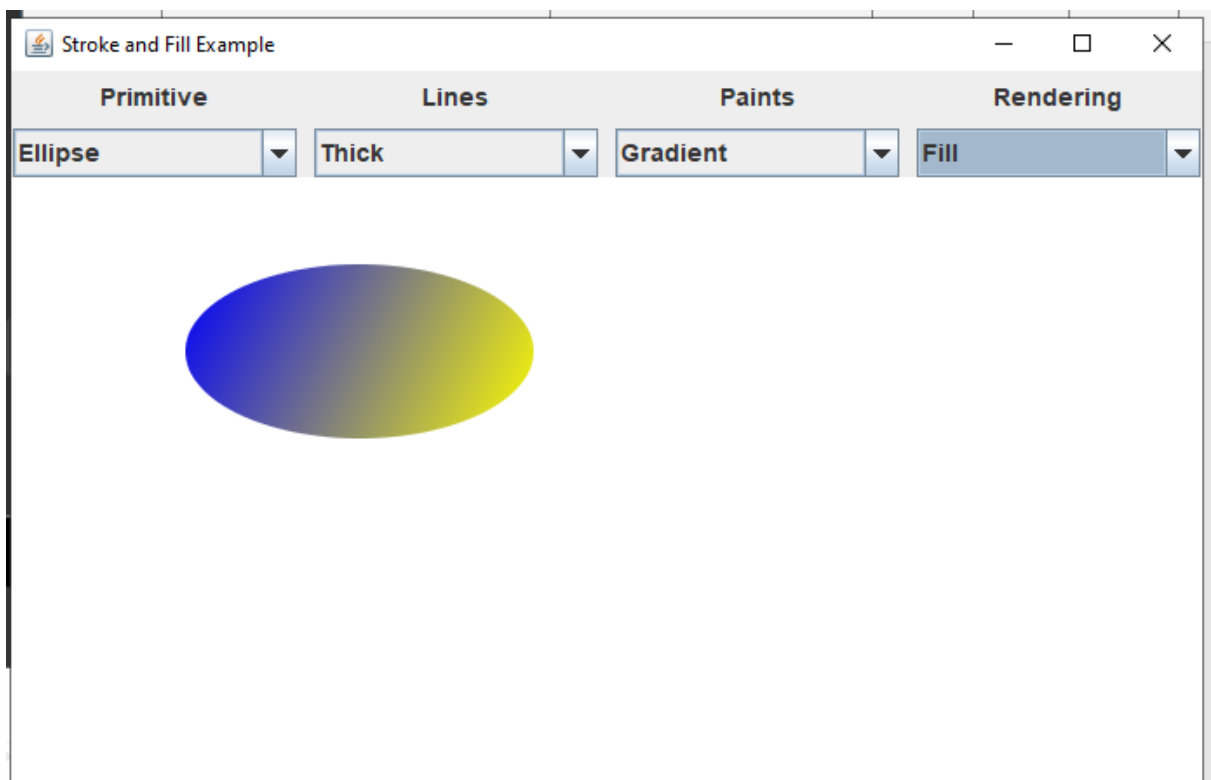
    String lineType = (String) line.getSelectedItemAt();
    if ("Thin".equals(lineType)) {
        g2.setStroke(new BasicStroke(1.0f));
    } else if ("Thick".equals(lineType)) {
        g2.setStroke(new BasicStroke(5.0f));
    } else {
        float[] dashes = { 10.0f};
        g2.setStroke(new BasicStroke(3.0f, BasicStroke.CAP_BUTT,
            BasicStroke.JOIN_BEVEL, 10.0f, dashes, 0.0f));
    }

    String strokeType = (String) stroke.getSelectedItemAt();
    if ("Fill".equals(strokeType)) {
        g2.fill(shape);
    } else if ("Stroke".equals(strokeType)) {
        g2.draw(shape);
    } else {
        g2.fill(shape);
        g2.setPaint(Color.BLACK);
        g2.draw(shape);
    }
}
}
}
}
}

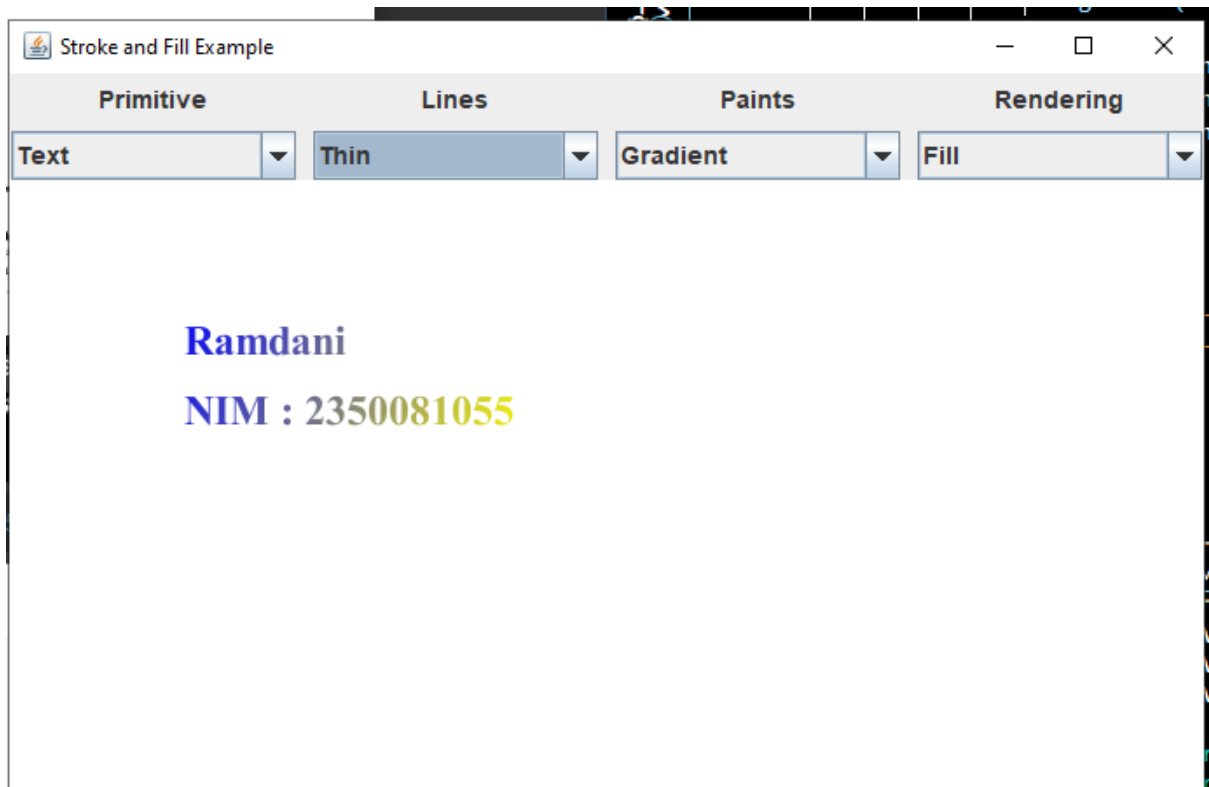
```



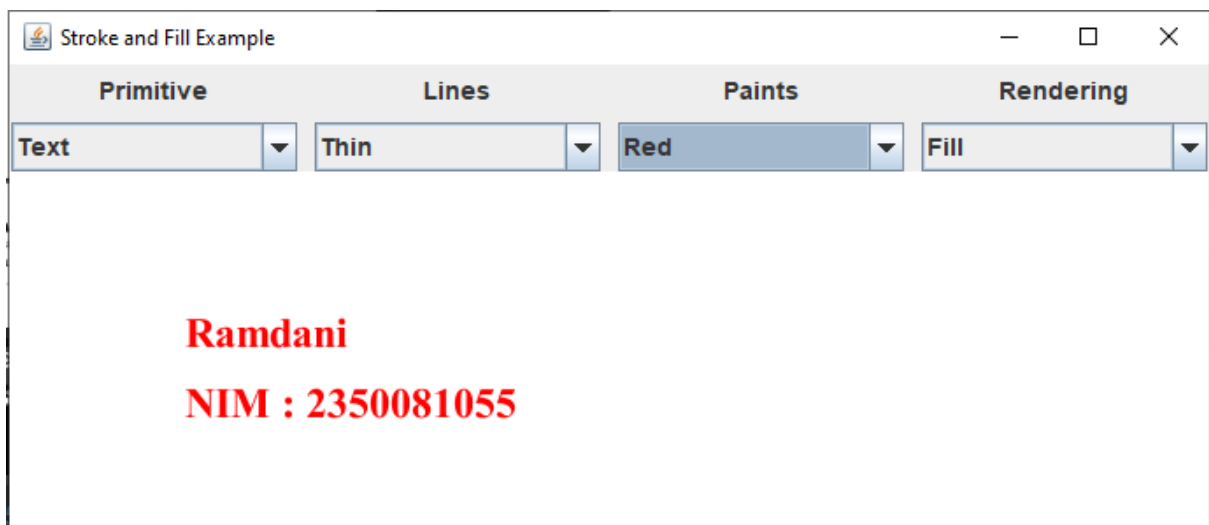
gambar I- 11 StrokeAndFill



gambar I- 12 StrokeAndFill



gambar I- 13 StrokeAndFill



gambar I- 14 StrokeAndFill

1.9.C. Analisa

Program Latihan ke 9 ini adalah aplikasi GUI Java menggunakan Swing yang memungkinkan pengguna memilih berbagai opsi untuk menggambar dan mewarnai bentuk 2D secara interaktif. StrokeAndFill adalah kelas utama yang membuat jendela dengan panel kontrol berisi empat JComboBox untuk memilih tipe bentuk (persegi panjang, elips, atau teks), tipe garis (tipis, tebal, putus-

putus), tipe pewarnaan (merah solid, gradien, tekstur), dan metode rendering (stroke, fill, atau keduanya). Panel ShapePanel bertanggung jawab menggambar bentuk sesuai pilihan pengguna dengan menerapkan antialiasing untuk kualitas visual yang lebih baik.

BAB II. TUGAS PRAKTIKUM

2.1. Tugas 6.1

2.1.A. Source Code

```
import java.awt.*;
import javax.swing.*;

public class Tugas6_1_WarnaPanel extends JFrame {
    public Tugas6_1_WarnaPanel() {
        setTitle("Panel Warna dan Keterangan");
        setLayout(new GridLayout(5, 2, 10, 10)); // 10 warna

        // Array warna dan nama
        Color[] colors = {
            Color.RED, Color.BLUE, Color.GREEN, Color.YELLOW,
            Color.ORANGE,
            Color.PINK, Color.CYAN, Color.MAGENTA, Color.GRAY,
            Color.LIGHT_GRAY
        };

        String[] names = {
            "Merah", "Biru", "Hijau", "Kuning", "Oranye",
            "Pink", "Cyan", "Magenta", "Abu-abu", "Abu terang"
        };

        // Tambahkan panel berwarna dan labelnya
        for (int i = 0; i < colors.length; i++) {
            JPanel panel = new JPanel();
            panel.setBackground(colors[i]);
            panel.setBorder(BorderFactory.createLineBorder(Color.BLACK));

            panel.add(new JLabel(names[i]));
            add(panel);
        }
    }
}
```

```

setSize(500, 400);
setLocationRelativeTo(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new Tugas6_1_WarnaPanel());
}
}

```

2.1.B. Hasil



gambar 1- Tugas 6.1

2.1.C. Analisa

Kode Program Tugas 1 ini membuat sebuah aplikasi GUI sederhana menggunakan Java Swing yang menampilkan 10 panel warna dengan nama warna masing-masing di dalam sebuah jendela JFrame. Layout menggunakan GridLayout dengan 5 baris dan 2 kolom, memberi jarak 10 piksel antar panel secara horizontal

dan vertikal.

2.2. Tugas 6.2

2.2.A. Source Code

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Tugas6_2_ButtonWarna extends JFrame {
    private Color currentColor = Color.WHITE;

    public Tugas6_2_ButtonWarna() {
        setTitle("Tampilkan Rectangle Berwarna");
        setLayout(new BorderLayout());

        // Panel tombol
        JPanel buttonPanel = new JPanel();
        JButton merahBtn = new JButton("Merah");
        JButton hijauBtn = new JButton("Hijau");
        JButton biruBtn = new JButton("Biru");

        buttonPanel.add(merahBtn);
        buttonPanel.add(hijauBtn);
        buttonPanel.add(biruBtn);
        add(buttonPanel, BorderLayout.NORTH);

        // Panel untuk menggambar rectangle
        JPanel drawPanel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                g.setColor(currentColor);
                g.fillRect(100, 50, 200, 100);
            }
        };
        add(drawPanel, BorderLayout.SOUTH);
    }
}
```

```

    }
};

drawPanel.setPreferredSize(new Dimension(400, 300));
drawPanel.setBackground(Color.WHITE);
add(drawPanel, BorderLayout.CENTER);

// ActionListener
merahBtn.addActionListener(e -> {
    currentColor = Color.RED;
    drawPanel.repaint();
});

hijauBtn.addActionListener(e -> {
    currentColor = Color.GREEN;
    drawPanel.repaint();
});

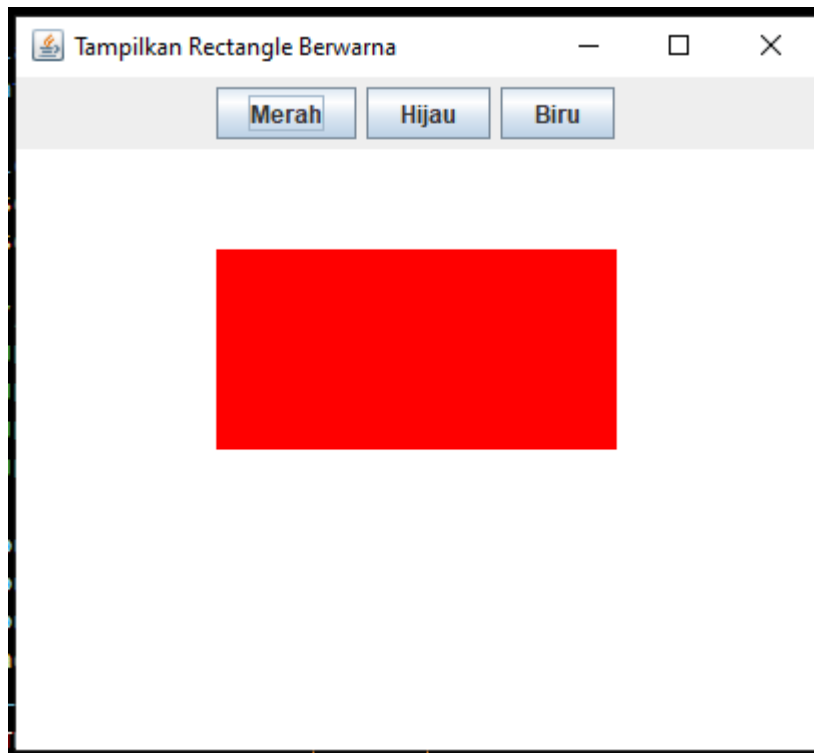
biruBtn.addActionListener(e -> {
    currentColor = Color.BLUE;
    drawPanel.repaint();
});

pack();
setLocationRelativeTo(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
}

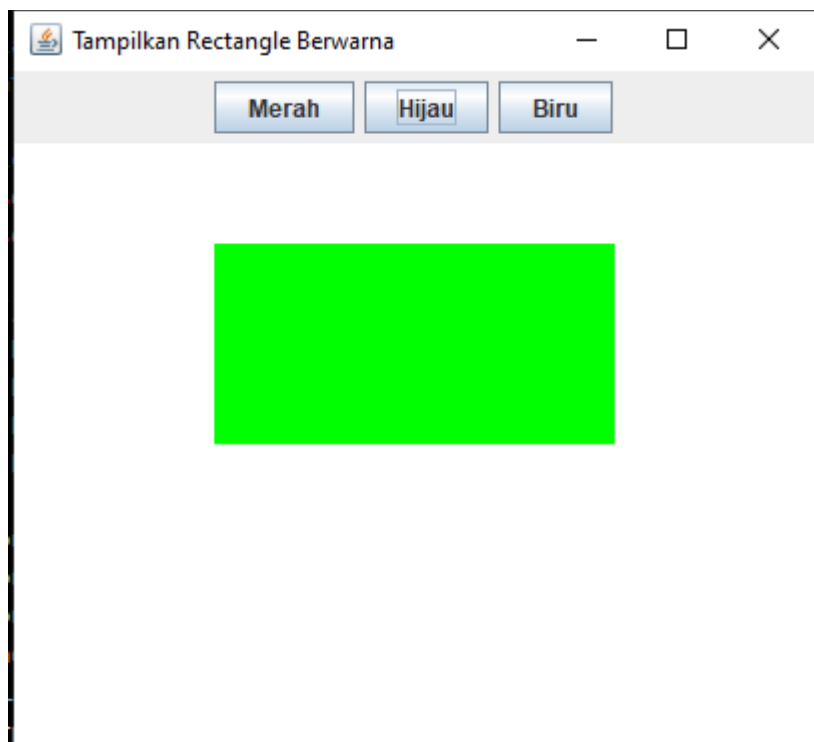
public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new Tugas6_2_ButtonWarna());
}
}

```

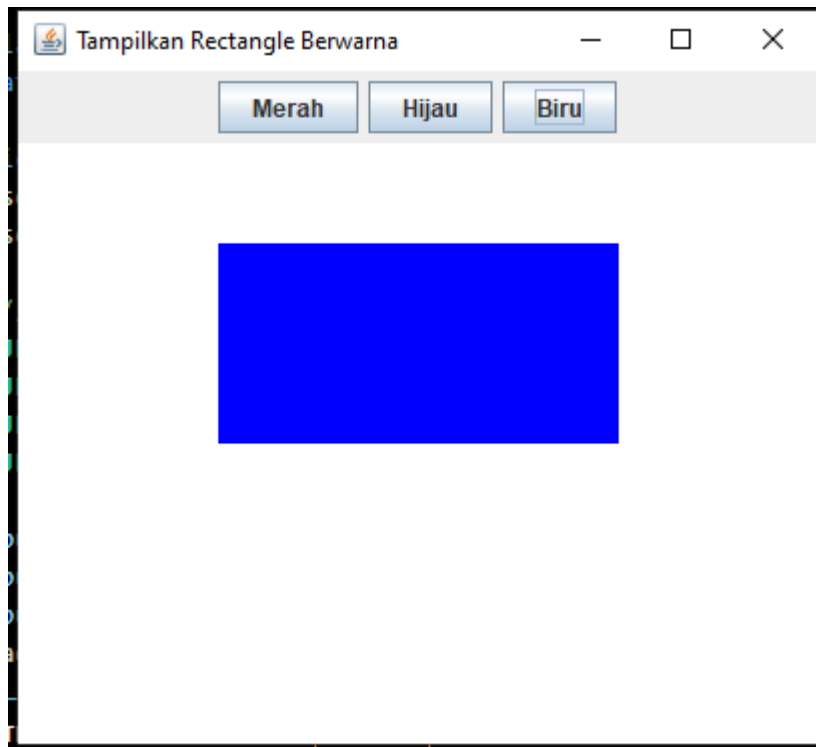
2.2.B. Hasil



gambar I- 15 Tugas6_2_ButtonWarna



gambar I- 16 Tugas6_2_ButtonWarna



gambar I- 17 Tugas6_2_ButtonWarna

2.2.C. Analisa

Kode ini adalah aplikasi GUI sederhana menggunakan Java Swing yang menampilkan tiga tombol berwarna "Merah", "Hijau", "Biru" dan sebuah panel gambar untuk menggambar persegi panjang berwarna sesuai tombol yang ditekan. Kelas Tugas6_2_ButtonWarna memperluas JFrame dan menggunakan BorderLayout dengan panel tombol di bagian atas NORTH dan panel gambar di tengah CENTER.

2.3. Tugas Gabungan 6-1 dan 6-2

2.3.A. Source Code

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Tugas6_Gabungan6_1_6_2 extends JFrame {
    private Color currentColor = Color.WHITE;
    private DrawPanel drawPanel;

    public Tugas6_Gabungan6_1_6_2() {
        setTitle("Tugas 6 - Warna & Tombol Aksi");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



```

setLayout(new BorderLayout(10, 10));

// --- Panel Warna jadi Tombol (Tugas 6-1 sebagai tombol) ---
JPanel warnaPanel = new JPanel(new GridLayout(5, 2, 5, 5));
Color[] colors = {
    Color.RED, Color.BLUE, Color.GREEN, Color.YELLOW,
    Color.ORANGE,
    Color.PINK, Color.CYAN, Color.MAGENTA, Color.GRAY,
    Color.LIGHT_GRAY
};
String[] names = {
    "Merah", "Biru", "Hijau", "Kuning", "Oranye",
    "Pink", "Cyan", "Magenta", "Abu-abu", "Abu terang"
};

for (int i = 0; i < colors.length; i++) {
    JButton colorBtn = new JButton(names[i]);
    colorBtn.setBackground(colors[i]);
    colorBtn.setOpaque(true);
    colorBtn.setBorderPainted(false);
    colorBtn.setForeground(getContrastColor(colors[i]));
    colorBtn.setFont(new Font("SansSerif", Font.BOLD, 14));

    Color c = colors[i]; // variabel final untuk lambda
    colorBtn.addActionListener(e -> {
        currentColor = c;
        drawPanel.repaint();
    });

    warnaPanel.add(colorBtn);
}

// --- Panel Gambar Rectangle ---
drawPanel = new DrawPanel();
drawPanel.setPreferredSize(new Dimension(400, 200));
drawPanel.setBackground(Color.WHITE);

```

```

// --- Gabungkan ---
add(warnaPanel, BorderLayout.NORTH);
add(drawPanel, BorderLayout.CENTER);

pack();
setLocationRelativeTo(null);
setVisible(true);
}

// Panel custom untuk menggambar rectangle
private class DrawPanel extends JPanel {
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(currentColor);
        g.fillRect(100, 50, 200, 100);
    }
}

// Fungsi untuk mendapatkan warna kontras (putih/ hitam) agar teks
tombol terbaca jelas
private Color getContrastColor(Color color) {
    double y = (299 * color.getRed() + 587 * color.getGreen() + 114 *
color.getBlue()) / 1000;
    return y >= 128 ? Color.BLACK : Color.WHITE;
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new Tugas6_Gabungan6_1_6_2());
}
}

```

2.3.B. Hasil



gambar I- 18 Tugas6_Gabungan6_1_6_2

2.3.C. Analisa

Kode ini menggabungkan dua konsep sebelumnya, yakni menampilkan deretan tombol berwarna dan panel gambar yang menggambar persegi panjang berwarna sesuai tombol yang dipilih, ke dalam satu jendela JFrame menggunakan layout BorderLayout. Bagian atas NORTH berisi warnaPanel yang merupakan grid 5x2 tombol, masing-masing mewakili warna tertentu dengan latar belakang berwarna dan teks yang otomatis disesuaikan kontrasnya agar terbaca putih atau hitam. Saat sebuah tombol ditekan, warna yang terkait disimpan dalam variabel `currentColor` dan panel gambar `DrawPanel` yang berada di tengah CENTER akan merepaint persegi panjang dengan warna tersebut. `DrawPanel` adalah kelas dalam inner class yang mengoverride `paintComponent` untuk menggambar persegi panjang dengan warna saat ini.

BAB III. TUGAS AKHIR

3.1. Tugas Akhir 1

3.1.A. Source Code

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class PopupMenuExample extends JFrame {
    private JPopupMenu popupMenu;
    private JPanel panel;

    public PopupMenuExample() {
        setTitle("Using JPopupMenu");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        panel = new JPanel();
        panel.setComponentPopupMenu(createPopupMenu());

        panel.addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                if (e.isPopupTrigger()) popupMenu.show(panel, e.getX(),
e.getY());
            }

            public void mouseReleased(MouseEvent e) {
                if (e.isPopupTrigger()) popupMenu.show(panel, e.getX(),
e.getY());
            }
        });

        add(panel);
        setLocationRelativeTo(null); // agar muncul di tengah layar
    }
}
```

```

    }

    private JPopupMenu createPopupMenu() {
        popupMenu = new JPopupMenu();

        JRadioButtonMenuItem blueItem = new
JRadioButtonMenuItem("Blue");
        JRadioButtonMenuItem yellowItem = new
JRadioButtonMenuItem("Yellow");
        JRadioButtonMenuItem redItem = new
JRadioButtonMenuItem("Red");

        ButtonGroup group = new ButtonGroup();
        group.add(blueItem);
        group.add(yellowItem);
        group.add(redItem);

        popupMenu.add(blueItem);
        popupMenu.add(yellowItem);
        popupMenu.add(redItem);

        // Tambahkan aksi ketika warna dipilih
        blueItem.addActionListener(e -> panel.setBackground(Color.BLUE));
        yellowItem.addActionListener(e ->
panel.setBackground(Color.YELLOW));
        redItem.addActionListener(e -> panel.setBackground(Color.RED));

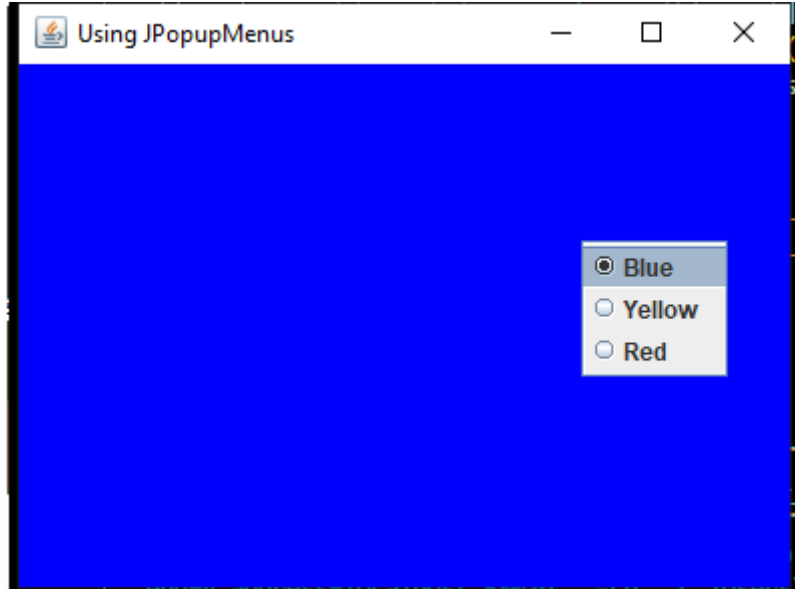
        return popupMenu;
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            new PopupMenuExample().setVisible(true);
        });
    }
}

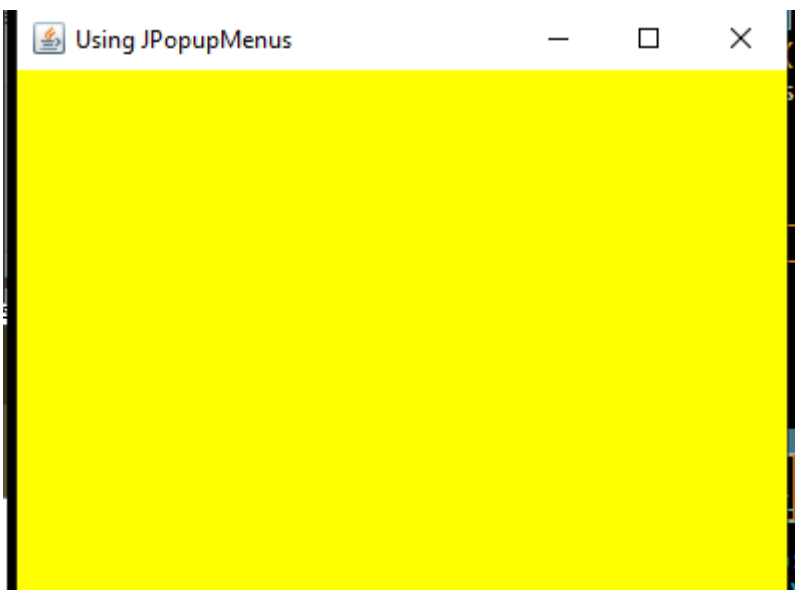
```

```
});  
}  
}
```

3.1.B. Hasil



gambar I- 19 Tugas Akhir 1 Blue



gambar I- 20 Tugas Akhir 1 Yellow

3.1.C. Analisa

Tugas Akhir ke 1 ini membuat aplikasi GUI sederhana menggunakan Java Swing yang menampilkan sebuah panel dengan menu popup (klik kanan) berisi tiga pilihan warna: Biru, Kuning, dan Merah. Saat pengguna klik kanan di dalam panel, JPopupMenu muncul dengan opsi warna yang berupa JRadioButtonMenuItem, sehingga hanya satu

warna yang bisa dipilih sekaligus. Pilihan warna dikelompokkan dalam ButtonGroup agar bersifat eksklusif.

3.2. Tugas Akhir 2

3.2.A. Source Code

```
import javax.swing.*;
import java.awt.*;
import java.awt.geom.AffineTransform;
import java.awt.geom.GeneralPath;

public class StarCirclePattern extends JPanel {

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        drawStarCircle((Graphics2D) g);
    }

    private void drawStarCircle(Graphics2D g2d) {
        int centerX = getWidth() / 2;
        int centerY = getHeight() / 2;
        int radius = 100;

        // Set rendering hints for smooth graphics
        g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

        // Create 5-pointed star
        Shape star = createStar(0, 0, 30, 60, 5);

        // Draw 36 stars rotated around the center
        for (int i = 0; i < 36; i++) {
            double angle = Math.toRadians(i * 10);
```

```

        AffineTransform at = new AffineTransform();
        at.translate(centerX + Math.cos(angle) * radius, centerY +
Math.sin(angle) * radius);
        at.rotate(angle);

        // Set random color
        g2d.setColor(new Color((int)(Math.random() * 0x1000000)));
        g2d.fill(at.createTransformedShape(star));
    }
}

private Shape createStar(double x, double y, double innerRadius, double
outerRadius, int numRays) {
    GeneralPath path = new GeneralPath();
    double angle = Math.PI / numRays;

    for (int i = 0; i < numRays * 2; i++) {
        double r = (i % 2 == 0) ? outerRadius : innerRadius;
        double theta = i * angle;
        double dx = x + Math.cos(theta) * r;
        double dy = y + Math.sin(theta) * r;
        if (i == 0) {
            path.moveTo(dx, dy);
        } else {
            path.lineTo(dx, dy);
        }
    }
    path.closePath();
    return path;
}

public static void main(String[] args) {
    JFrame frame = new JFrame("Drawing 2D Shapes");

```

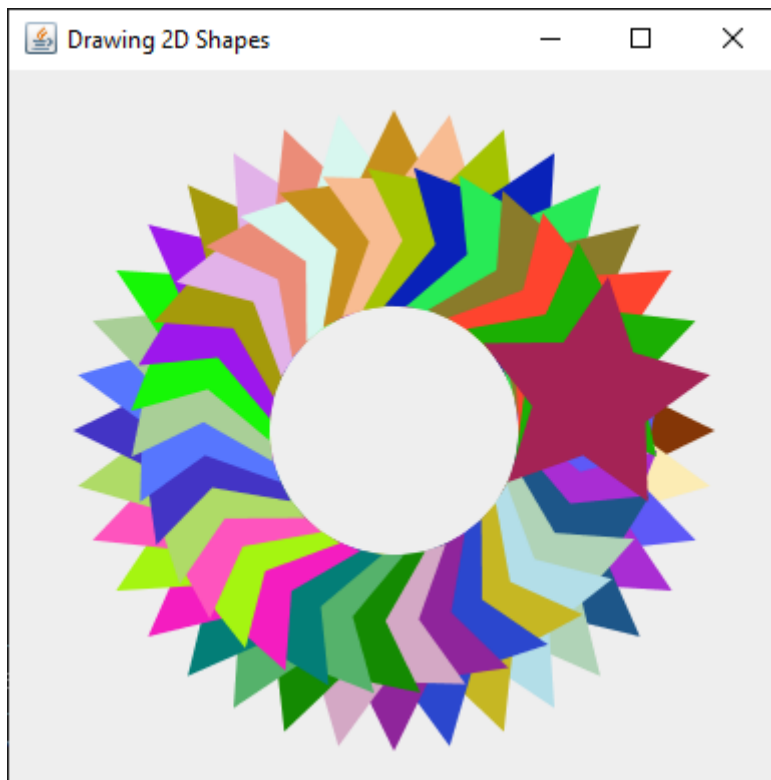


```

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 400);
        frame.add(new StarCirclePattern());
        frame.setLocationRelativeTo(null); // Center on screen
        frame.setVisible(true);
    }
}

```

3.2.B. Hasil



gambar I- 21 StarCirclePattern

3.2.C. Analisa

Program untuk Tugas akhir ke 2 ini membuat pola dekoratif berupa bintang lima sisi yang diatur melingkar sebanyak 36 buah mengelilingi titik tengah panel. Setiap bintang digambar dengan warna acak dan diputar sesuai posisi sudutnya agar efeknya dinamis dan menarik secara visual. Fungsi `createStar` menghasilkan bentuk bintang dengan parameter radius dalam dan luar, serta jumlah "rays" (sisi) bintang.

BAB IV. KESIMPULAN

Pada praktikum modul 6 ini membahas cara membuat dan memanipulasi gambar dua dimensi menggunakan kelas-kelas dari paket `java.awt` dan `java.awt.geom`, seperti menggambar bentuk dasar (lingkaran, persegi panjang, garis, busur), mengatur warna, gradien, dan tekstur, serta mengontrol ketebalan dan gaya garis (stroke). Materi ini juga mencakup teknik transformasi bentuk seperti rotasi dan translasi, penggunaan rendering hints untuk hasil visual yang lebih halus, serta penggabungan bentuk kompleks. Selain itu, integrasi dengan Swing memungkinkan pembuatan antarmuka grafis interaktif yang responsif terhadap aksi pengguna, sehingga Java 2D Graphics menjadi fondasi penting untuk membuat aplikasi grafis dan visualisasi yang dinamis dan menarik.