**Laporan Praktikum**

**Bahasa Pemrograman 1**

**Dosen pengampu : (*Dede Husen, M.Kom.*)**

# Modul 5



**Nama : Muhammad Rizal Nurfirdaus**

**NIM : 20230810088**

**Kelas : TINFC – 2023 – 04**

**Teknik Informatika Fakultas Ilmu Komputer**

**Universitas Kuningan**

**PreTest**

1. Apakah Komponen AWT dan Swing dapat digunakan untuk membuat animasi 2D atau 3D?

Komponen AWT (Abstract Window Toolkit) dan Swing dalam Java memang dapat digunakan untuk membuat animasi 2D, seperti menggambar dan memindahkan objek di sebuah kanvas (Canvas). Namun, untuk animasi 3D, Java biasanya memerlukan library tambahan seperti Java 3D API atau library pihak ketiga seperti LWJGL dan JOGL. AWT dan Swing sendiri tidak menyediakan kemampuan bawaan untuk animasi 3D.

1. Buat program untuk penerapan jawaban No.1 tersebut dalam Java!

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

/\*\*

\*

\* @author Muhammad Rizal Nur F

\*/

import javax.swing.\*;

import java.awt.\*;

public class BP1\_M5\_PreTest\_Rizal extends JPanel {

private int x = 0; // Posisi x awal objek

private int y = 50; // Posisi y awal objek

public void moveBall() {

while (true) {

x += 5; // Gerak ke kanan

if (x > getWidth()) {

x = 0; // Reset posisi jika keluar layar

}

repaint(); // Gambar ulang

try {

Thread.sleep(50); // Delay animasi

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

g.setColor(Color.BLUE);

g.fillOval(x, y, 30, 30); // Gambar lingkaran

}

public static void main(String[] args) {

JFrame frame = new JFrame("Simple Animation");

BP1\_M5\_PreTest\_Rizal animation = new BP1\_M5\_PreTest\_Rizal();

frame.add(animation);

frame.setSize(400, 200);

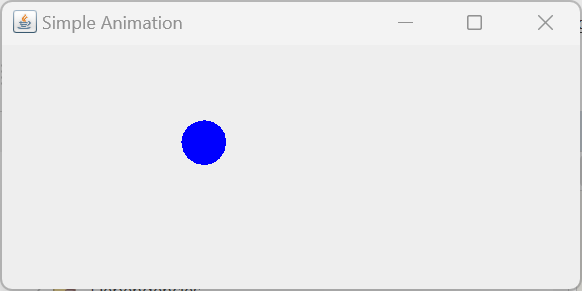
frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

animation.moveBall(); // Memulai animasi

}

}



Analisis : Program BP1\_M5\_PreTest\_Rizal adalah implementasi animasi 2D sederhana menggunakan Java Swing. Program ini menampilkan sebuah lingkaran biru yang bergerak secara horizontal dari kiri ke kanan di dalam jendela aplikasi. Jika lingkaran mencapai tepi kanan, posisinya akan direset ke awal, menciptakan animasi berulang. Animasi dikontrol oleh metode moveBall(), yang secara berkala memperbarui posisi lingkaran menggunakan thread dengan delay 50 milidetik untuk kelancaran gerakan.

**Praktikum**



/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

/\*\*

\*

\* @author Muhammad Rizal Nur F

\*/

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Graphics;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import java.util.List;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.SwingUtilities;

import javax.swing.Timer;

public class BP1\_M5\_P1\_Rizal extends JPanel {

private static final int L = 400, Y = 400;

private List<Mobil> mMobil;

public BP1\_M5\_P1\_Rizal() {

setBackground(new Color(0, 0, 0));

setLayout(null);

mMobil = new ArrayList<>();

mMobil.add(new Mobil(100, 200));

Timer timer = new Timer(50, new ActionListener() {

public void actionPerformed(ActionEvent e) {

for (Mobil mobil : mMobil) {

mobil.Bergerak();

}

repaint();

}

});

timer.start();

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

for (Mobil mobil : mMobil) {

mobil.GambarMobil(g);

}

}

@Override

public Dimension getPreferredSize() {

return new Dimension(L, Y);

}

public class Mobil {

private static final int INCREMENT = 5;

private int x, y;

public Mobil(int x, int y) {

this.x = x;

this.y = y;

}

public void GambarMobil(Graphics g) {

g.setColor(Color.BLUE); // atap mobil

g.fillRect(x + 15, y - 20, 60, 35);

g.setColor(Color.YELLOW); // badan mobil

g.fillRect(x, y, 110, 30);

g.setColor(Color.RED); // roda mobil

g.fillOval(x + 15, y + 20, 20, 20);

g.fillOval(x + 60, y + 20, 20, 20);

}

public void Bergerak() {

if (x >= L) { // jika sudah mencapai batas

x = 1; // kembali ke posisi awal

} else {

x += INCREMENT; // bergerak maju

}

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

JFrame frame = new JFrame();

frame.getContentPane().add(new BP1\_M5\_P1\_Rizal());

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

frame.setTitle("Animasi Mobil Bergerak!");

frame.pack();

frame.setLocationRelativeTo(null);

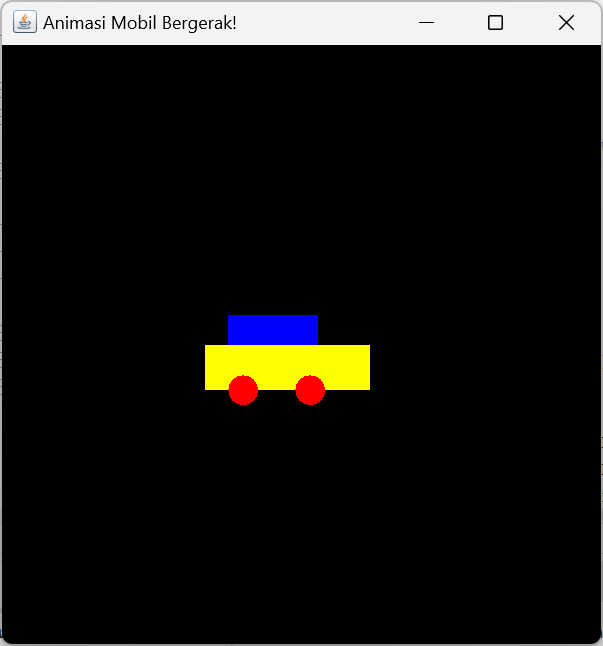
frame.setVisible(true);

}

});

}

}



Analisis : Kode di atas adalah implementasi sederhana dari aplikasi **Form Login** menggunakan Java Swing. Aplikasi ini memiliki antarmuka grafis dengan dua bidang teks untuk input *username* dan *password*, serta dua tombol: "LOGIN" dan "REGISTER". Saat tombol "LOGIN" diklik, metode Login() akan dipanggil untuk memeriksa apakah *username* dan *password* yang dimasukkan sesuai dengan nilai tetap ("admin" dan "admin111"). Jika cocok, pesan sukses akan ditampilkan menggunakan JOptionPane, sementara jika tidak, pesan kesalahan akan muncul. Desain antarmuka mencakup label, teks, dan ikon, yang diatur menggunakan javax.swing.GroupLayout.



**PostTest**

1. Buat Java class baru dengan nama BP1\_M5\_PostTest\_NamaAnda. Buat program untuk menghasilkan animasi bergerak sesuai dengan kreasi Anda.

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

/\*\*

\*

\* @author Muhammad Rizal Nur F

\*/

import javax.swing.\*;

import java.awt.\*;

public class BP1\_M5\_PostTest\_Rizal extends JPanel {

private int x = 0; // Posisi x awal objek

private int y = 0; // Posisi y awal objek

private int dx = 5; // Kecepatan horizontal

private int dy = 3; // Kecepatan vertikal

public void moveBall() {

while (true) {

x += dx; // Gerak horizontal

y += dy; // Gerak vertikal

// Pantul jika mencapai tepi layar

if (x < 0 || x > getWidth() - 30) dx = -dx;

if (y < 0 || y > getHeight() - 30) dy = -dy;

repaint(); // Gambar ulang

try {

Thread.sleep(30); // Delay animasi

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

g.setColor(Color.RED);

g.fillOval(x, y, 30, 30); // Gambar lingkaran

}

public static void main(String[] args) {

JFrame frame = new JFrame("PostTest Animation");

BP1\_M5\_PostTest\_Rizal animation = new BP1\_M5\_PostTest\_Rizal();

frame.add(animation);

frame.setSize(400, 300);

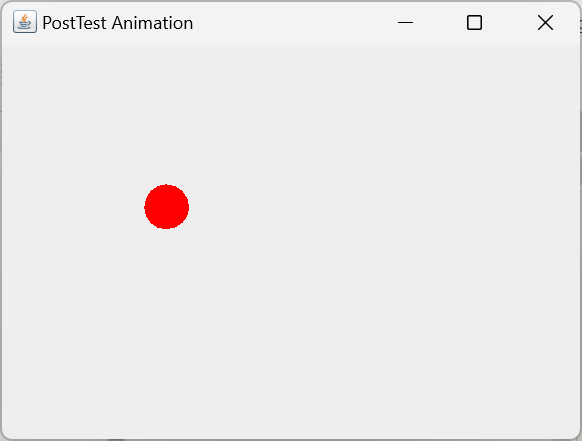
frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

animation.moveBall(); // Memulai animasi

}

}



Analisis : Program **BP1\_M5\_PostTest\_Rizal** adalah implementasi animasi 2D dengan lintasan diagonal menggunakan Java Swing. Objek lingkaran bergerak secara horizontal dan vertikal dengan kecepatan tertentu, lalu memantul saat mencapai batas jendela aplikasi. Gerakan ini diatur melalui pembaruan koordinat x dan y, sedangkan metode paintComponent() digunakan untuk menggambar ulang lingkaran setiap kali posisinya berubah.

**Tugas**

1. Buat Java class baru dengan nama BP1\_M5\_Tugas\_NamaAnda. Buat program dengan menghasilkan animasi bergerak dengan gabungan arah vertikal dan horizontal. (Boleh meneruskan dari hasil posttest).

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

/\*\*

\*

\* @author Muhammad Rizal Nur F

\*/

import javax.swing.\*;

import java.awt.\*;

import java.util.Random;

public class BP1\_M5\_Tugas\_Rizal extends JPanel {

private int x = 0; // Posisi x awal objek

private int y = 0; // Posisi y awal objek

private int dx = 5; // Kecepatan horizontal

private int dy = 3; // Kecepatan vertikal

private Color ballColor = Color.BLUE; // Warna lingkaran

public void moveBall() {

Random random = new Random();

while (true) {

x += dx; // Gerak horizontal

y += dy; // Gerak vertikal

// Pantul jika mencapai tepi layar

if (x < 0 || x > getWidth() - 30) {

dx = -dx;

ballColor = new Color(random.nextInt(256), random.nextInt(256), random.nextInt(256)); // Ganti warna

}

if (y < 0 || y > getHeight() - 30) {

dy = -dy;

ballColor = new Color(random.nextInt(256), random.nextInt(256), random.nextInt(256)); // Ganti warna

}

repaint(); // Gambar ulang

try {

Thread.sleep(30); // Delay animasi

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

g.setColor(ballColor);

g.fillOval(x, y, 30, 30); // Gambar lingkaran

}

public static void main(String[] args) {

JFrame frame = new JFrame("Tugas Animation");

BP1\_M5\_Tugas\_Rizal animation = new BP1\_M5\_Tugas\_Rizal();

frame.add(animation);

frame.setSize(500, 400);

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

frame.setVisible(true);

animation.moveBall(); // Memulai animasi

}

}



**Analisis :** Program BP1\_M5\_Tugas\_Rizal adalah pengembangan animasi 2D dengan gerakan diagonal yang menggabungkan efek pantulan dan perubahan warna objek. Lingkaran bergerak secara horizontal dan vertikal, memantul saat mencapai batas layar, sambil mengganti warna secara acak setiap kali terjadi pantulan. Warna lingkaran diubah menggunakan objek Random, menciptakan variasi visual yang menarik. Program ini memperlihatkan kombinasi animasi posisi dan manipulasi atribut visual untuk menghasilkan efek animasi yang lebih kompleks dan interaktif.