

# **LAMPIRAN**

## **Listing Program**

```

package RGB;
import java.util.Random;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
/**
 * @author Andreas Octorianto
 * RGB ANOMALOSCOPE
 */
public class Midlet extends MIDlet {

    private Display display;
    private MainCanvas canvas;
    private MenuCanvas menu;
    private Alert r;

    public Midlet() {
        display = Display.getDisplay(this);
        canvas = new MainCanvas();
        menu = new MenuCanvas();
    }

    public void startApp() {
        display.setCurrent(menu);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }

    public void showResult() {
        String Result = null;
        if (canvas.count == 15) {
            if (canvas.r_score >= 40 && canvas.g_score >= 40) {
                Result = "Anda terbukti Buta Warna. Mohon hubungi dokter untuk lebih lanjut. Terima kasih telah menggunakan aplikasi ini";
            }
            if (canvas.r_score >= 40 && canvas.g_score <= 40) {
                Result = "Anda terbukti Buta Warna pada warna merah. Mohon hubungi dokter untuk lebih lanjut. Terima kasih telah menggunakan aplikasi ini";
            }
            if (canvas.r_score <= 40 && canvas.g_score >= 40) {
                Result = "Anda terbukti Buta Warna pada warna merah. Mohon hubungi dokter untuk lebih lanjut. Terima kasih telah menggunakan aplikasi ini";
            }
            if (canvas.r_score <= 40 && canvas.g_score <= 40) {
                Result = "Anda tidak terbukti Buta Warna. Terima kasih telah menggunakan aplikasi ini";
            }
        }
    }
}

```

```

        r = new Alert("Result", Result, null, AlertType.INFO);
        r.setTimeout(Alert.FOREVER);
        display.setCurrent(r);
    } else {
        Result = "Anda belum menyelesaikan test!";
        r = new Alert("Result", Result, null, AlertType.INFO);
        r.setTimeout(Alert.FOREVER);
        display.setCurrent(r);
    }
}

public void exit() {
    destroyApp(true);
    notifyDestroyed();
}

public void next() { display.setCurrent(canvas); }

public void back () {
    canvas.r_score = 0; //reset
    canvas.g_score = 0;
    canvas.count = 0;
    display.setCurrent(menu);
}

class MainCanvas extends Canvas implements CommandListener {
    private final Command back = new Command("back", Command.BACK, 1);
    /* Score */
    int r_score = 0;
    int g_score = 0;
    /* Counter */
    int count = 0;
    /* current sample color */
    Random r = new Random();
    int Color_r = r.nextInt(256);
    int Color_g = r.nextInt(256);
    int Color_b = 0;
    int rgbColor = 0x00ffff00;
    /* current index */
    int ndx = 1;

    protected void paint(Graphics g) {
        this.addCommand(back);
        this.setCommandListener(this);
        int w = getWidth();
        int h = getHeight();
        //ukuran warna soal
        int sample_w = w - 1;
        int sample_h = h - 48;
        int sample_y = 2;
    }
}

```

```

//ukuran warna berubah
int tick_w = w - 1;
int tick_h = h - 128;
int tick_y = 2;
//ukuran indikator
int b_y = sample_y + sample_h + 4;
int g_y = b_y + 14;
//latar belakang layar
g.setColor(0, 0, 0);
g.fillRect(0, 0, w, h);
//warna untuk yang berubah
g.setColor(rgbColor);
g.fillRect(2, sample_y, sample_w, sample_h);
//Warna untuk soal
g.setColor(Color_r, Color_g, Color_b);
g.fillRect(2, tick_y, tick_w, tick_h);
//warna untuk penunjuk perubahan
int yellow = (rgbColor >> 8) & 0xff;
g.setColor(255, 255, 0);
g.fillRect(20, g_y, yellow / 4, 10);
int red1 = (rgbColor >> 16) & 0xff;
//tulisan pada canvas penunjuk soal
g.setColor(255, 255, 255);
g.drawString(Integer.toString(count), 36, b_y - 3, Graphics.LEFT |
Graphics.TOP);
g.drawString(Integer.toString(yellow), 18, g_y - 3, Graphics.RIGHT |
Graphics.TOP);
}

public void keyRepeated(int key) {
    keyPressed(key); //Handle repeat as in pressed.
}

protected void keyPressed(int key) {
    int action = getGameAction(key);
    int dir = 0;
    switch (action) {
    case RIGHT:
        dir += 1;
        break;
    case LEFT:
        dir -= 1;
        break;
    case FIRE: //jawaban berbeda score tidak bertambah
        if (count == 15){
            showResult();
        }
        count++;
        Random s = new Random();
        Color_r = s.nextInt(256);

```

```

        Color_g = s.nextInt(256);
        break;

    case UP: //jawaban sama score bertambah
        if(count == 15){
            showResult();
        }
        int colorG = (rgbColor >> (ndx*8)) & 0xff;
        int colorR = (rgbColor >> ((ndx+1)*8)) & 0xff;
        if (Color_r > Color_g && colorR < (Color_r+3) && colorR > (Color_r-3))
        {r_score+=10;}
        if (Color_g > Color_r && colorG < (Color_g+3) && colorG > (Color_g-3))
        {g_score+=10;}
        count++;
        Random d = new Random();
        Color_r = d.nextInt(256);
        Color_g = d.nextInt(256);
        if(r_score >= 150) {r_score = 150;}
        if(g_score >= 150) {g_score = 150;}
        break;

    default:
        return;
    }

    //limit
    if(count > 15) { count = 15; }
    int x = (rgbColor >> (ndx*8)) & 0xff;
    int y = (rgbColor >> (ndx+1)*8) & 0xff;
    int v = x/y;
    v += (dir * 0x20);
    if(v < 0) { v = 0; }
    if(v > 255) { v = 255; }
    int mask1 = 0xff << (ndx * 8);
    int mask2 = 0xff << ((ndx+1)*8);
    int mask = mask1|mask2;
    rgbColor = (rgbColor & ~mask) | (v << (ndx * 8) | (v << (ndx+1)*8));
    repaint();
}

    public void commandAction(Command c, Displayable d) {
        if(c == back) { back(); }
    }
}

class MenuCanvas extends Canvas implements CommandListener{
    private final Command next = new Command ("Ok", Command.OK, 1);
    private final Command exit = new Command ("Exit", Command.EXIT, 1);
    protected void paint(Graphics g) {

```

```

        this.addCommand(next);
        this.addCommand(exit);
        this.setCommandListener(this);
        g.setColor(0, 0, 0);
        g.fillRect(0, 0, getWidth(), getHeight());
        g.setColor(255,255,255);
        g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,      Font.STYLE_BOLD,
Font.SIZE_LARGE));
        g.drawString("Selamat          Datang",          115,          30,
Graphics.HCENTER/Graphics.BASELINE);
        g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,      Font.STYLE_BOLD,
Font.SIZE_LARGE));
        g.drawString("dalam          Tes          Buta          Warna.",          115,          45,
Graphics.HCENTER/Graphics.BASELINE);
        g.setFont(Font.getFont(Font.FACE_SYSTEM,              Font.STYLE_PLAIN,
Font.SIZE_MEDIUM));
        g.drawString("untuk memilih jawaban BERBEDA tombol tengah", 120, 70,
Graphics.HCENTER/Graphics.BASELINE);
        g.setFont(Font.getFont(Font.FACE_SYSTEM,              Font.STYLE_PLAIN,
Font.SIZE_MEDIUM));
        g.drawString("untuk jawaban SAMA tekan panah atas", 115, 85,
Graphics.HCENTER/Graphics.BASELINE);
        g.setFont(Font.getFont(Font.FACE_SYSTEM,              Font.STYLE_BOLD,
Font.SIZE_LARGE));
        g.drawString("Apakah Ingin Memulai Test ?", 120, 125,
Graphics.HCENTER/Graphics.BASELINE);
    }
    public void commandAction(Command c, Displayable d) {
        if (c == exit) { exit(); }
        if (c == next) { next(); }
    }
}
}
}

```