

CHAPTER V

IMPLEMENTATION AND TESTING

5.1 Implementation

This system running with java, so GUI use as interface of this program. This program has 3 important steps. They are process to read file txt, like as : template matrix, path image, and input value sobel. And process grayscale image use for process sobel detection. And then process sobel detection with 3x5 or 5x5. These are program listing below :

```
import javax.swing.*;
import java.awt.*;
import java.io.*;
import java.awt.Image;
import java.awt.event.*;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import java.io.BufferedWriter;
import java.io.FileWriter;

public class EdgesDetection extends JFrame implements ActionListener
{
    JButton button3;
    JLabel label1, label2;
    public EdgesDetection()
    {
        setTitle("Sobel Edges Detection");
        setBounds(200, 70, 700, 100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
        Container contentArea= getContentPane();
        JPanel panel2= new JPanel();
        panel2.setPreferredSize(new Dimension(480,500));
        panel2.setBorder(BorderFactory.createEtchedBorder());
        JPanel panel5= new JPanel();
        contentArea.add(panel2, BorderLayout.CENTER);
        label1= new JLabel();
        label1.setText("No Picture");
        panel2.add(label1);
        panel5.add(panel2);
        contentArea.add(panel5);

        JPanel panel4= new JPanel();
        contentArea.add(panel4, BorderLayout.SOUTH);
        button3= new JButton("Edges");
        button3.addActionListener(this);
        panel4.add(button3);
        pack();
    }
}
```

This method is used to make components of GUI interface. Create a view by showing two images, the original image and the image after result edges detection like make a comparison image.

```

public void actionPerformed(ActionEvent event)
{
    String value[]=new String[50];
    int plus=0;
    try
    {
        String tulis="";
        BufferedReader config=new BufferedReader(new FileReader("Conf.txt"));
        while((tulis=config.readLine())!=null)
        {
            value[plus]=tulis;
            //System.out.println(value[plus]);
            plus=plus+1;
        }
    }
    catch (Exception e)
    {
        System.out.println("path file tidak ada");
    }

    ImageIcon icon = new ImageIcon(value[1]);
    Image img = icon.getImage();
    Image newImg2 = img.getScaledInstance(500, 500,Image.SCALE_SMOOTH);
    ImageIcon icon3 = new ImageIcon(newImg2);
    label1.setText("");
    label1.setIcon(icon3);
}

```

This is process use for read user input from Conf.txt. In reading from txt do the checks each row. And then the results of the read from txt store in array. So that contents of the arrays will be a value input from txt. After that original image put on one of the display on the GUI interface. Original image was taken from user input in Conf.txt that is entered into the array.

```

int x=0;
int y=0;
int red = 0;
int green = 0;
int blue = 0;

try
{
    BufferedImage baca = ImageIO.read(new File(value[1]));
    int width = baca.getWidth();
    int height = baca.getHeight();
    int pixels[] = new int[width*height];
    int[][] output = new int[width][height];
    BufferedImage gray = new BufferedImage(width,height,BufferedImage.TYPE_BYTE_GRAY);
    Sobel tiga= new Sobel();
    Sobel2 lima= new Sobel2();

    for (int xx=0; xx< width; xx++)
    {
        try
        {
            for (int yy = 0; yy < height; yy++)
            {
                int color = baca.getRGB(xx, yy);
                red = getRed(color);
                green = getGreen(color);
                blue = getBlue(color);
                color=(red+green+blue)/3;
                gray.setRGB(xx,yy,color);
            }
        }

        catch (Exception e)
        {
            System.out.println("error di ambil nilai RGB");
        }
    }
}

```

This is process to read image to get the width and height. After that take the RGB values of each pixel, to change into grayscale value of each of the red, green, and blue in the average, so the colors are originally colored to gray. Then gray value fill into bufferedimage with type byte, because bufferedimage store gray value then type on bufferimage is byte.

```

int counter=0;

gray.getRaster().getPixels(0,0,width,height,pixels);

for(int xz=0; xz < width; xz++ )
{
    for(int yz=0 ; yz < height ; yz++ )
    {
        output[xz][yz] = pixels[counter];
        counter= counter + 1;
        //System.out.println(output[i][j]);
    }
}

counter=0;
int hasil[][]=new int[width][height];
int sob3=2;
int itung=0;
if(Integer.parseInt(value[0])==3)
{
    while(value[sob3]!=null)
    {
        itung=itung+1;
        sob3=sob3+1;
    }

    if(itung!=3)
    {
        System.out.println("matrik tidak sesuai 3x3");
    }
    else
    {
        tiga.setSatu(Integer.parseInt(value[2]));
        tiga.setDua(Integer.parseInt(value[3]));
        tiga.setTiga(Integer.parseInt(value[4]));
        hasil=tiga.Edges(output,width,height);
    }
}
else if(Integer.parseInt(value[0])==5) //cek sobel 5x5
{
    while(value[sob3]!=null)
    {
        itung=itung+1;
        sob3=sob3+1;
    }
    if(itung!=10)
    {
        System.out.println("matrik tidak sesuai 5x5");
    }
    else
    {
        lima.setSatu(Integer.parseInt(value[2]));
        lima.setDua(Integer.parseInt(value[3]));
        lima.setTiga(Integer.parseInt(value[4]));
        lima.setEmpat(Integer.parseInt(value[5]));
        lima.setLima(Integer.parseInt(value[6]));
        lima.setEnam(Integer.parseInt(value[7]));
        lima.setTujuh(Integer.parseInt(value[8]));
        lima.setDelapan(Integer.parseInt(value[9]));
        lima.setSembilan(Integer.parseInt(value[10]));
        lima.setSepuluh(Integer.parseInt(value[11]));
        hasil=lima.Edges(output,width,height);
    }
}
}

```

This process use for take the value of the pixel after process grayscale, then fill into array multi dimensional which will be carried out the edges detection. Take value from array txt which contains input, when input 3 then execute class Sobel (3x3 matrix) and when input 5 then execute class Sobel2 (matrix 5x5). And also the process send and request value to other class.

```

        for(int ii = 0 ; ii < width ; ii++ )
        {
            for(int jj = 0 ; jj < height ; jj++ )
            {
                pixels[counter] =hasil[ii][jj];
                counter = counter + 1;
            }
        }

        gray.getRaster().setPixels(0,0,width,height,pixels);

        ImageIcon icon5 = new ImageIcon(gray);
        Image newImg = icon5.getImage().getScaledInstance(500, 500, Image.SCALE_SMOOTH);
        ImageIcon icon2 = new ImageIcon(newImg);
        label2.setText("");
        label2.setIcon(icon2);
    }
    catch (Exception e)
    {
        System.out.println("error baca image");
    }
}

public static int getRed(int color) {
    return (color & 0x00ff0000) >> 16;
}
;
public static int getGreen(int color) {
    return (color & 0x0000ff00) >> 8;
}
;
public static int getBlue(int color) {
    return (color & 0x000000ff) >> 0;
}
;

public static void main(String args [])
{
    EdgesDetection mainframe= new EdgesDetection();
}
}

```

The code above use for store value new pixel after process sobel edges detection and then image after process edges detection put on one of the display on the GUI interface.

```

public class Sobel
{
    int rumus1=0;
    int rumus2=0;
    int rumus3=0;

    public void setSatu(int a)
    {
        rumus1=a;
    }
    public void setDua(int a)
    {
        rumus2=a;
    }
    public void setTiga(int a)
    {
        rumus3=a;
    }

    public int[][] Edges(int pix[][],int widht, int height)
    {
        int horizontal[][]=new int[widht][height];
        int vertikal[][]=new int[widht][height];
        int gabung[][]=new int[widht][height];

        for (int wew=0; wew<widht; wew++)
        {
            for (int tt=0; tt<height; tt++)
            {
                if(wew==0 || wew==widht-1 || tt==0 || tt==height-1)
                    horizontal[wew][tt] = vertikal[wew][tt] = gabung[wew][tt] =0;
                else
                {
                    horizontal[wew][tt] = rumus1*pix[wew+1][tt-1] + rumus2*pix[wew+1][tt] + rumus3*pix[wew+1][tt+1]-
                    rumus1*pix[wew-1][tt-1] - rumus2*pix[wew-1][tt] - rumus3*pix[wew-1][tt+1];
                    vertikal[wew][tt] = rumus1*pix[wew-1][tt+1] + rumus2*pix[wew][tt+1] + rumus3*pix[wew+1][tt+1]-
                    rumus1*pix[wew-1][tt-1] - rumus2*pix[wew][tt-1] - rumus3*pix[wew+1][tt-1];
                    gabung[wew][tt] = Math.abs( horizontal[wew][tt]) + Math.abs( vertikal[wew][tt]);
                }
            }
        }

        return gabung;
    }
}

```

The code above use to sobel detection with matrix 3x3, value sobel take from user input from class EdgesDetection and then send new value back to class EdgesDetection. Process edges detection carried out one by one from array content pixel value after grayscale.

```
public class Sobel2
```

```
int rumus1=0;
int rumus2=0;
int rumus3=0;
int rumus4=0;
int rumus5=0;
int rumus6=0;
int rumus7=0;
int rumus8=0;
int rumus9=0;
int rumus10=0;
public void setSatu(int a)
{
    rumus1=a;
}
public void setDua(int a)
{
    rumus2=a;
}
public void setTiga(int a)
{
    rumus3=a;
}
public void setEmpat(int a)
{
    rumus4=a;
}
public void setLima(int a)
{
    rumus5=a;
}
public void setEnam(int a)
{
    rumus6=a;
}
public void setTujuh(int a)
{
    rumus7=a;
}
public void setDelapan(int a)
{
    rumus8=a;
}
public void setSembilan(int a)
{
    rumus9=a;
}
public void setSepuluh(int a)
{
    rumus10=a;
}
}
```

```
public int[][] Edges(int pix[],int widht, int height)
{
```

```
int horizontal[][]=new int[widht][height];
int vertikal[][]=new int[widht][height];
int gabung[][]=new int[widht][height];
for (int wew=0; wew<widht; wew++)
```

```
{
    for (int tt=0; tt<height; tt++)
```

```
{
    if(wew==0 || wew==widht-1 || tt==0 || tt==height-1 || wew==1 || wew==widht-2 || tt==1 || tt==height-2)
        horizontal[wew][tt]=vertikal[wew][tt]=gabung[wew][tt]=0;
```

```
else
```

```
{
```

```
horizontal[wew][tt] = rumus1*pix[wew+2][tt-2] + rumus2*pix[wew+2][tt-1] + rumus3*pix[wew+2][tt] + rumus4*pix[wew+2][tt+1] +
rumus5*pix[wew+2][tt+2] + rumus6*pix[wew+1][tt-2] + rumus7*pix[wew+1][tt-1] + rumus8*pix[wew+1][tt] + rumus9*pix[wew+1][tt+1] +
rumus10*pix[wew+1][tt+2] - rumus1*pix[wew-2][tt-2] - rumus2*pix[wew-2][tt-1] - rumus3*pix[wew-2][tt] - rumus4*pix[wew-2][tt+1] -
rumus5*pix[wew-2][tt+2] - rumus6*pix[wew-1][tt-2] - rumus7*pix[wew-1][tt-1] - rumus8*pix[wew-1][tt] - rumus9*pix[wew-1][tt+1] -
rumus10*pix[wew-1][tt+2];
```

```
vertikal[wew][tt] = rumus1*pix[wew+2][tt-2] + rumus2*pix[wew+1][tt-2] + rumus3*pix[wew][tt-2] + rumus4*pix[wew-1][tt-2] +
rumus5*pix[wew-2][tt-2] + rumus6*pix[wew+2][tt-1] + rumus7*pix[wew+1][tt-1] + rumus8*pix[wew][tt-1] + rumus9*pix[wew-1][tt-1] +
rumus10*pix[wew-2][tt-1] - rumus1*pix[wew+2][tt+2] - rumus2*pix[wew+1][tt+2] - rumus3*pix[wew][tt+2] - rumus4*pix[wew-1][tt+2] -
rumus5*pix[wew-2][tt+2] - rumus6*pix[wew+2][tt+1] - rumus7*pix[wew+1][tt+1] - rumus8*pix[wew][tt+1] - rumus9*pix[wew-1][tt+1] -
rumus10*pix[wew-2][tt+1];
```

```
gabung[wew][tt] = Math.abs( horizontal[wew][tt]) + Math.abs( vertikal[wew][tt]);
}
}
```

```
return gabung;
}
```

The code above use to sobel detection with template matrix 5x5,value sobel take from user input from class EdgesDetection and then send new value back to class EdgesDetection. Process edges detection carried out one by one from array content pixel value after grayscale.

5.2 Testing

To test the program works properly, writer make a test. This test consist of 14 point for total point if the program is work properly. In this test, every item will be tested, if the result is match with what expected or purpose, then it will get 2 point, if it does not match then it will get 1 point, and if the program does not work, it will get 0 point.

No	Action	Result	Score
1	Run the program in computer	Program running	2
2	Show message when input is not filled	Message out with information which part is not filled	2
3	Input matrix,path image, sobel value	Show result image by matrix, path image, and sobel value choose	2
4	Click Edges button	Process Edges detection work correctly	2
5	Output Edges detection	Edges detection goes well, but when width and height image is not same / not even, make result of a untidy	1
6	Change input matrix and path image	Result image change	2
7	Change input sobel value	Result image change correclty	2
		Total testing point	13
		Maximum point	14
		Percentage	93%

Table 5.1 Testing Table

5.3 Application Interface

This is the application interface that will be appear when program is running and after user enter the inputs, then program simulate the result.



Figure 5.1 Application Interface

The button Edges use for execute program to process image with sobel detection, after user input matrix, path image, and value sobel. Output GUI is contains of the original image and image after edges detection.

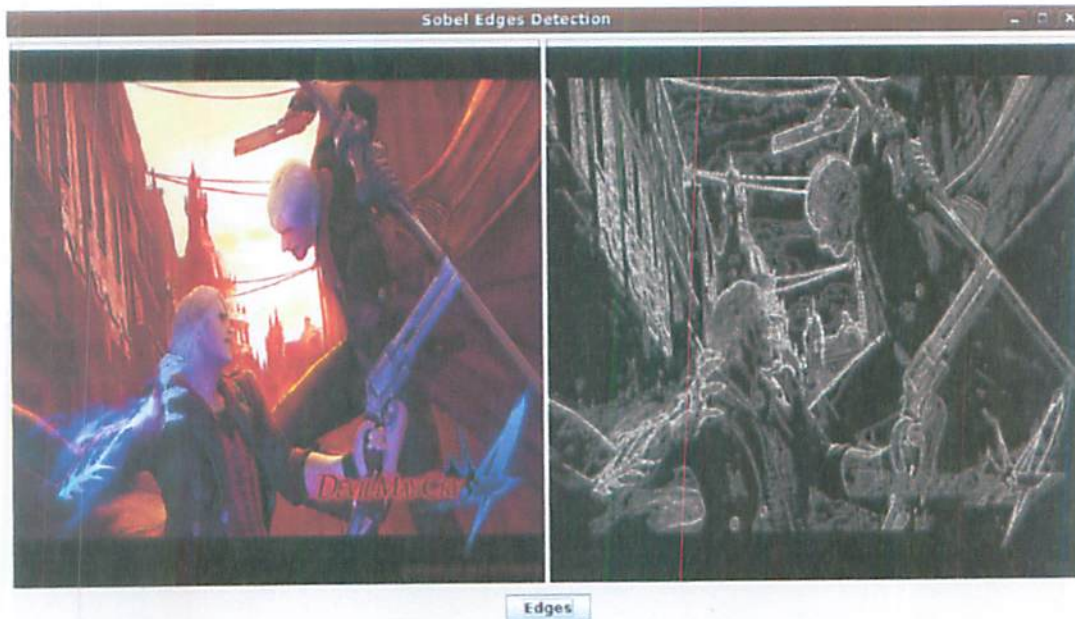


Figure 5.2 Result Sobel Matrix 3x3

This is result image after process sobel edges detection with matrix 3x3. It can be seen that the image in the form of edge, image objects more visible. After process edges detection object in the image can be easier to recognize.



Figure 5.3 Result Sobel Matrix 5x5

This is result image after process sobel edges detection with matrix 5x5. It can be seen that the image in the form of edge more visible then process with matrix 3x3, image objects more visible and more detail. After process edges detection object in the image can be easier to recognize.