

CHAPTER V

Implementation and Testing

5.1 Implementation

5.1.1 Main Program

This project will run `TampilanAwal.java` as the first interface. User can type the equation in the blank field, from rank 2 to rank 4. After press the button OK, the program will execute the equation with 3 algorithm, and produce some coordinates then will be save in tree as data structure.

```
.....  
public void run()  
{  
    num1 = Float.parseFloat(a.getText());  
    num2 = Float.parseFloat(b.getText());  
    num3 = Float.parseFloat(c.getText());  
    num4 = Float.parseFloat(d.getText());  
    num5 = Float.parseFloat(e.getText());  
    pohonKu = new MyTree(num1, num2, num3, num4, num5);  
    pohonKu.buatPohonAlgo();  
}  
.....|
```

Picture 5.1 : Main Program.

5.1.2 Algorithm

3 algorithm will run simultaneously. Each algorithm have several methods. The methods will calculate the equation to produce the coordinates. Each coordinates consist of x, y, and the algorithm. Type of algorithm in code is String, x and y are float. And will become a graph in integer, because pixels cannot receive float number.

```

.....
do
{
    if (Fxr*Fa<0.Of)
    {
        bawah=xr;
        Fb=Fxr;
    }
    else
    {
        atas=xr;
        Fa=Fxr;
    }
    hasilNt = Float.toString(xr);
    hasilFt = Float.toString(Fxr);
    System.out.println();
    System.out.println("X " + hasilNt + " Y " + hasilFt);
    pohonAlg.add(xr, Fxr, alg);
    n++;
    Fa=fa(x4,x3,x2,x1,x,atas);
    Fb=fb(x4,x3,x2,x1,x,bawah);
    xr=(atas+bawah)/2.Of;
    Fxr=fxr(x4,x3,x2,x1,x,xr);
}
while (n<=8);
.....

```

Picture 5.2 Algorithm : Bisection

```

.....
do
{
    n2=n1-((fn1*n10)/(fn1-fn0));
    y=(float)((x4*(Math.pow(n2,4)))+(
    x3*(Math.pow(n2,3)))+(
    x2*(Math.pow(n2,2)))+(
    x1*(Math.pow(n2,1))+x);
    temp=n1;
    n0=temp;
    n1=n2;
    fn1=fnsatu(x4,x3,x2,x1,x,n1);
    fn0=fnnol(x4,x3,x2,x1,x,n0);
    i++;
    hasilN2 = Float.toString(n2);
    hasilY = Float.toString(y);
    System.out.println();
    System.out.println("X " +
        hasilN2 + " Y " + hasilY);
    pohonAlg.add(n2, y, alg);
}
while (i<=10);
.....

```

Picture 5.3 Algorithm : Secant

```

.....
do
{
    xiT = xi-(Fxi/FTxi);
    Fxi = (float) ((x4*(Math.pow(xiT,4)))+
        (x3*(Math.pow(xiT,3)))+
        (x2*(Math.pow(xiT,2)))+
        (x1*(Math.pow(xiT,1)))+x);
    hasilNt = Float.toString(xiT);
    hasilFt = Float.toString(Fxi);
    System.out.println();
    System.out.println("X " + hasilNt +
        " Y " + hasilFt);
    pohonAlg.add(xiT, Fxi, alg);
    n++;
    xi=xiT;
    Fxi=fxi(x4,x3,x2,x1,x,xi);
    FTxi=ftxi(x4,x3,x2,x1,x,xi);
}
while (n<=5);
.....

```

Picture 5.4 Algorithm : Newton

5.1.3 Choose the Graph

From TampilanAwal.java, user will see several choice (radiobutton). User choose one of them, then can see the graph accordance with the choice. Every choice have action to find the coordinates in tree.

```

.....
class RadioButtonListener implements ActionListener
{
    public void actionPerformed (ActionEvent event)
    {
        if (t.isSelected())
        {
            ambilPohon = new MyBinaryTree();
            ambilPohon.cari(MyTree.root,t.getText());
            hasil=new Coba7();
            hasil.setVisible(true);
        }
    }
}
.....

```

Picture 5.5 : pilihan.java

5.1.4 Applet

The graph will appear in applet (java in web base). Applet made by manual code. This is no action between applet and algorithm which is chosen by user. When user choose the algorithm in pilihan.java (radiobutton), there is an action. Action to find the coordinates in tree. When the coordinates found, it will be shown beside the interface, means in terminal. To create a graph, use manual code. Insert the coordinates, of course in integer to Coba7.java. And will be processing in applet (Coba7.html)

```
.....  
public class Coba7 extends Applet  
{  
    int width, height;  
    public void init()  
    {  
        width = getSize().width;  
        height = getSize().height;  
        setBackground( Color.white );  
    }  
    public void paint(Graphics graphics)  
    {  
        graphics.drawLine(200,0,200,400);  
        graphics.drawLine(0,200, 400, 200);  
    }  
}  
.....
```

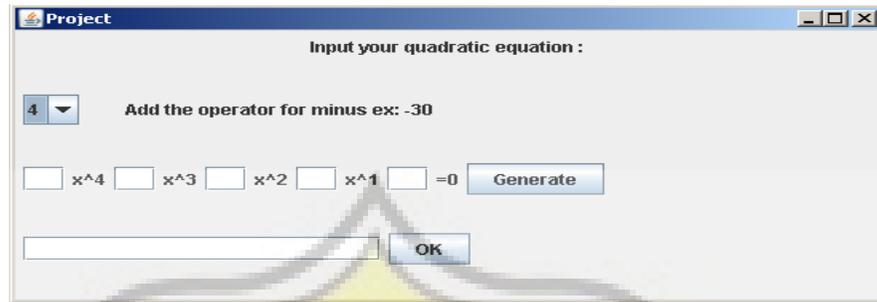
Picture 5.6 : Coba7.java

Here is some code to call the applet in Coba7.html :

```
<applet width=500 height=500 code="Coba7.class"> </applet>
```

5.2 Testing

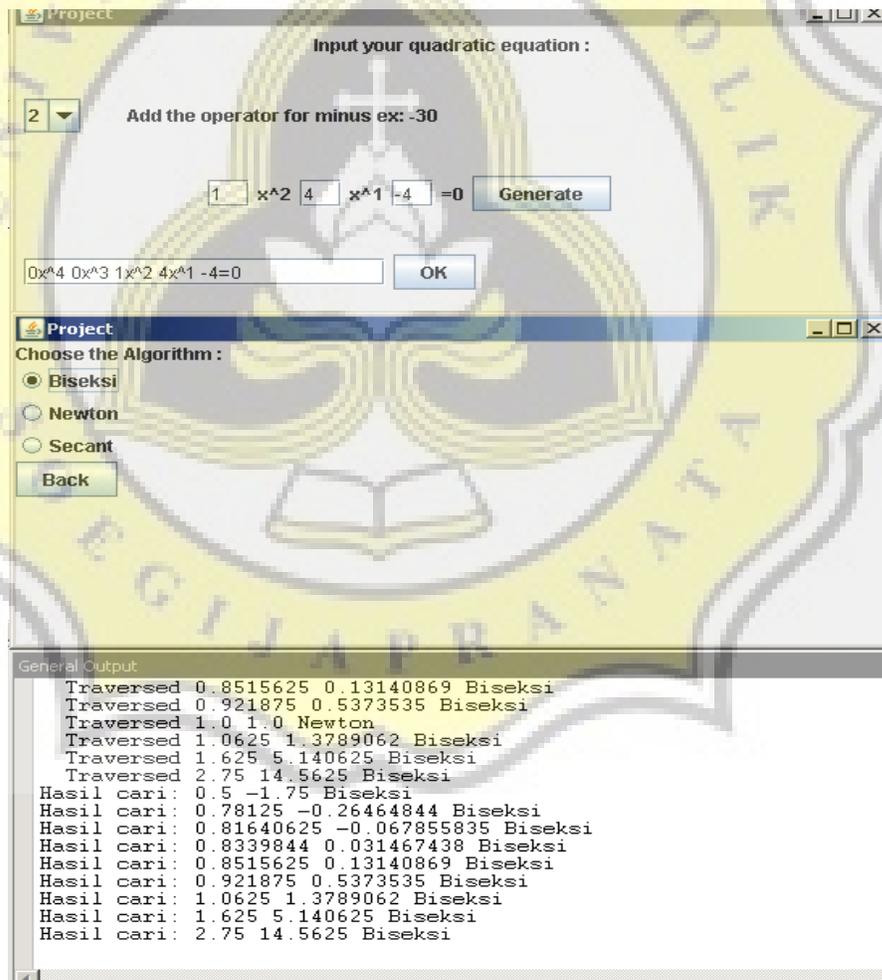
5.2.1 Form Input



The screenshot shows a window titled "Project" with the subtitle "Input your quadratic equation :". Below the subtitle, there is a dropdown menu with the value "4" and the text "Add the operator for minus ex: -30". Below this, there are five input fields for coefficients: x^4 , x^3 , x^2 , x^1 , and a constant term, followed by "=0". A "Generate" button is to the right of the constant field. At the bottom, there is an "OK" button.

Picture 5.7 Form Input

5.2.2 Choose the Algorithm

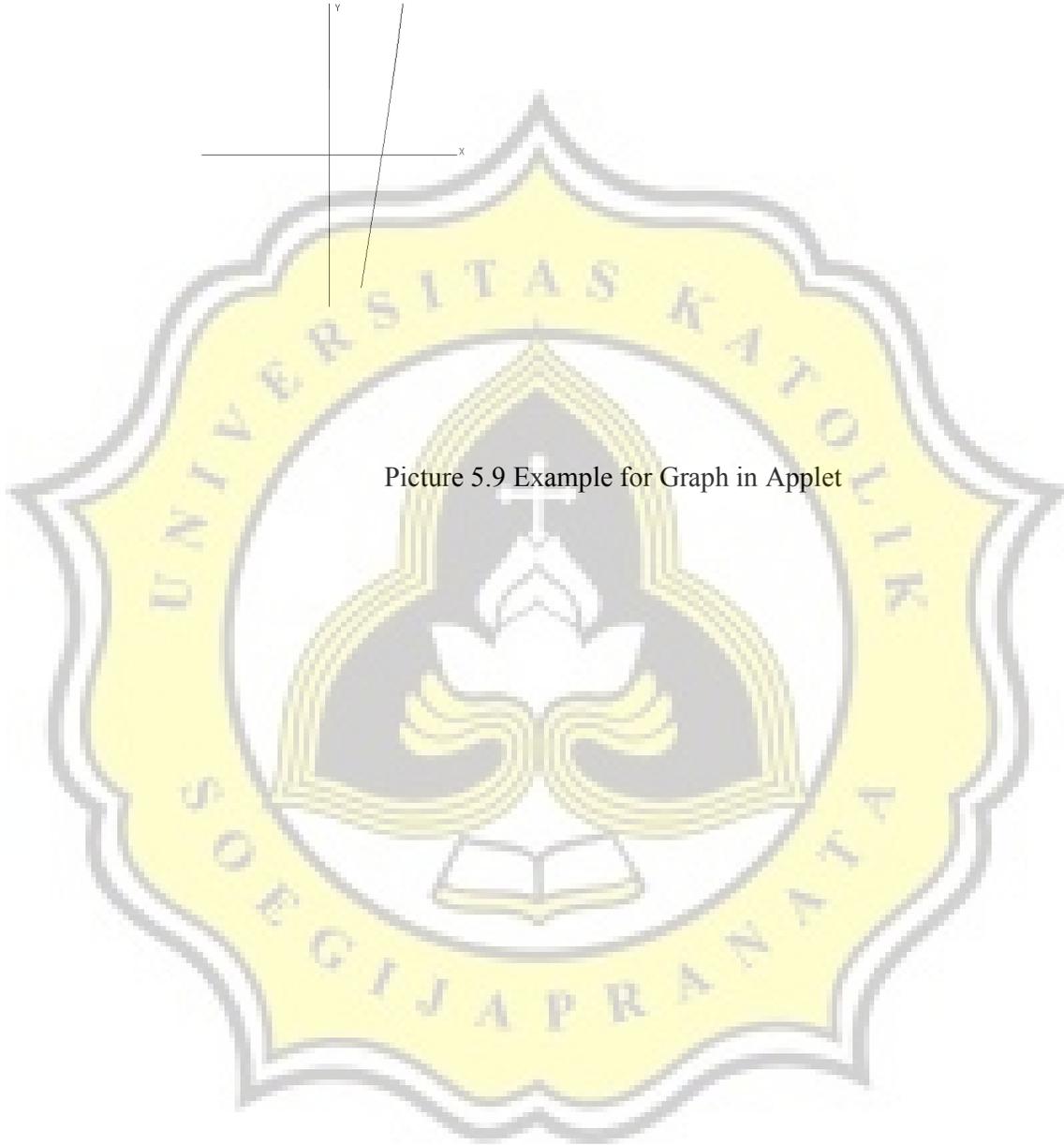
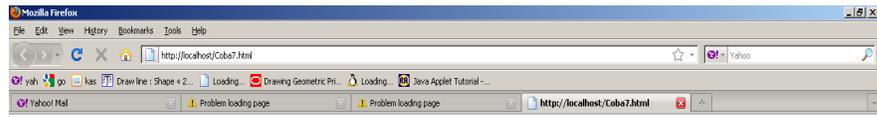


The screenshot shows the "Project" window with the subtitle "Input your quadratic equation :". The dropdown menu now shows "2" and the text "Add the operator for minus ex: -30". The coefficient fields are: x^2 (1), x^1 (4), and constant (-4), followed by "=0". A "Generate" button is to the right. Below the fields, the generated equation is displayed: $0x^4 0x^3 1x^2 4x^1 -4=0$. An "OK" button is below the equation. Below the main window, a smaller window titled "Project" is visible, showing "Choose the Algorithm :". The "Biseksi" radio button is selected. Other options are "Newton" and "Secant". A "Back" button is at the bottom. At the bottom of the main window, a "General Output" window displays the following text:

```
Traversed 0.8515625 0.13140869 Biseksi
Traversed 0.921875 0.5373535 Biseksi
Traversed 1.0 1.0 Newton
Traversed 1.0625 1.3789062 Biseksi
Traversed 1.625 5.140625 Biseksi
Traversed 2.75 14.5625 Biseksi
Hasil cari: 0.5 -1.75 Biseksi
Hasil cari: 0.78125 -0.26464844 Biseksi
Hasil cari: 0.81640625 -0.067855835 Biseksi
Hasil cari: 0.8339844 0.031467438 Biseksi
Hasil cari: 0.8515625 0.13140869 Biseksi
Hasil cari: 0.921875 0.5373535 Biseksi
Hasil cari: 1.0625 1.3789062 Biseksi
Hasil cari: 1.625 5.140625 Biseksi
Hasil cari: 2.75 14.5625 Biseksi
```

Picture 5.8 Result in Coordinates

5.2.3 Applet



Picture 5.9 Example for Graph in Applet