Lampiran N

Code Implementasi SplashScreen

Berikut ini adalah *code* implementasi SplashScreen.

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

public class SplashScreen extends JWindow{

  private ProgressBar progressBar;

  public SplashScreen(ImageIcon img){
    this(img, Color.blue, 15);
  }
  public SplashScreen(ImageIcon img, Color progBarColor, int progBarHeight){
    JLabel image = new JLabel();
    image.setIcon(img);
    image.setBorder(new javax.swing.border.LineBorder(Color.LIGHT_GRAY));
    progressBar = new ProgressBar(0,100,progBarColor);
    progressBar.setPreferredSize(
                         new Dimension(this.getWidth(), progBarHeight));
    progressBar.setVisible(true);
    progressBar.setBorder(
                   new javax.swing.border.LineBorder(Color.LIGHT_GRAY));
    this.add(image, BorderLayout.CENTER);
    this.add(progressBar, BorderLayout.SOUTH);
  }
  public SplashScreen(ImageIcon img, Color progBarColor, int progBarHeight,
                        String dispStr, Font dispStrFont, Color dispStrCol){

    JLabel image = new JLabel(img);
    image.setBorder(new javax.swing.border.LineBorder(Color.LIGHT_GRAY));
    progressBar = new ProgressBar(0, 100, progBarColor,
                             dispStr, dispStrFont, dispStrCol);
    progressBar.setPreferredSize(
                         new Dimension(this.getWidth(), progBarHeight));
    progressBar.setVisible(true);
    progressBar.setBorder(
                    new javax.swing.border.LineBorder(Color.LIGHT_GRAY));
    this.setLayout(new BorderLayout());
    this.add(image, BorderLayout.CENTER);
    this.add(progressBar, BorderLayout.SOUTH);
  }
  public SplashScreen(int width, int height){
    progressBar = new ProgressBar();
    progressBar.setPreferredSize(new Dimension(width, height));
    this.add(progressBar, BorderLayout.CENTER);
    this.setSize(width, height);
    this.setLocationRelativeTo(null);
  }
  public void incrementProgress(int incVal){
    progressBar.increment(incVal);
  }
  public void incrementProgress(int incVal, String displayedString){
    progressBar.increment(incVal, displayedString);
  }
```

```
  public ProgressBar getProgressBar(){
    return progressBar;
  }
}

class ProgressBar extends JComponent{

  private int value;
  private int maxValue;
  private int minValue;
  private Color progBarColor;
  private String displayedString = "";
  private Font displayedStringFont;
  private Color displayedStringColor;

  public ProgressBar(int minValue, int maxValue, Color progBarColor,
                    String dispStr, Font dispStrFont, Color dispStrCol){

    this(minValue, maxValue, progBarColor);
    this.displayedString = dispStr;
    this.displayedStringFont = dispStrFont;
    this.displayedStringColor = dispStrCol;
  }
  public ProgressBar (int minValue, int maxValue, Color progBarColor) {
    super ();
    this.value = minValue;
    this.minValue = minValue;
    this.maxValue = maxValue;
    this.progBarColor = progBarColor;
  }
  public ProgressBar (){
    super ();
    value = minValue;
    this.minValue = 0;
    this.maxValue = 100;
    this.progBarColor = Color.blue;
  }
  public void increment (int incVal, String displayedString){
    this.displayedString = displayedString;
    this.increment(incVal);
  }
  public void increment (int incVal){
    int ii = 0;
    while(ii < incVal){
      value ++;
      try {
        Thread.sleep (3);
      }catch (InterruptedException ex){}
      repaint();
      ii++;
    }
  }
  public void paintComponent (Graphics g){

    super.paintComponent(g);
    int w = getWidth ();
    int h = getHeight ();
    Graphics2D g2 = (Graphics2D) g;

    g2.setColor (this.progBarColor);
    g2.fillRect(0,0,getProgress(), getHeight());

    if(displayedStringColor != null)
      g2.setColor(displayedStringColor);
    if(displayedStringFont != null)
      g2.setFont(displayedStringFont);
    g2.drawString(displayedString,10,15);
  }
  public void setMaximumValue (int maxvalue){
    this.maxValue = maxValue;
  }
  public int getProgress(){
    if(maxValue == 0)
      return 0;
    return value * getWidth() / maxValue;
  }
  public int getMaximumValue (){
    return maxValue;
  }
  public void setValue(int val){
    value = val;
  }
  public int getValue(){
    return value;
  }
  public void setProgBarColor(Color col){
    progBarColor = col;
  }
  public Color getProgBarColor(){
    return progBarColor;
  }
}
```

Lampiran O

Code Implementasi

TEX2PDFConverter.java

Berikut ini adalah *code* implementasi TEX2PDFConverter.java.

```
import java.io.*;
import java.util.StringTokenizer;

public class TEX2PDFConverter{

  public static File convertTeX2PDF(File texFile, File includeDir)
                                                  throws IOException{
    String message = "", temp = "";
    String dir = texFile.getParent();
    String texName = texFile.getName();

    String command = "pdflatex \""+texFile.getAbsolutePath()+"\" "+
                     " -output-directory="+dir+
                     " -include-directory=\""+
                     includeDir.getAbsolutePath()+"\"";

    Process prcs = Runtime.getRuntime().exec(command);
    BufferedReader bufInp = new BufferedReader(
                                new InputStreamReader(
                                    prcs.getInputStream()));

    //Read the output from command
    //just wait until the pdf file is generated
    while ((temp = bufInp.readLine()) != null);

    StringTokenizer strtok = new StringTokenizer(texName,".");
    String pdfName = dir+"\\"+strtok.nextToken()+".pdf";
    File pdfResult = new File(pdfName);

    return pdfResult;
  }
}
```

Lampiran P

Code Implementasi

JavaPrologInterface.java

Berikut ini adalah *code* implementasi JavaPrologInterface.java.

```
import java.io.*;

public class JavaPrologInterface{

  public static String execPrologQuery(String query) throws IOException{
    String result = "", temp = "";
    Process prcs = Runtime.getRuntime().exec(query);
    BufferedReader bufInp = new BufferedReader(
                                new InputStreamReader(
                                    prcs.getInputStream()));
    // Read the output from command
    while ((temp = bufInp.readLine()) != null)
      result += temp+"\n";
    return result;
  }
}
```