CHAPTER V
IMPLEMENTATION AND TESTING

5.1. Implementation

5.1.1. User Interface Design

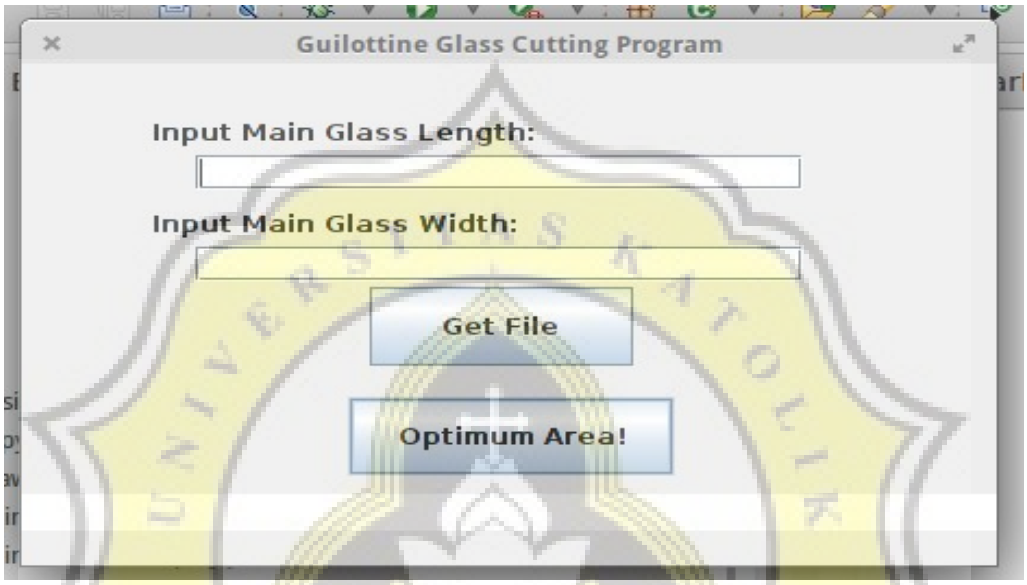

Figure 5 : User Interface Guillotine Glass Cutting Program

1. User input main glass length and main glass width with number. .
2. Then choose one of the button
   Get File – for processing only guillotine cut
   Optimum Area – for processing guillotine cut and generating optimum solution
3. After button clicked then open file window will shown

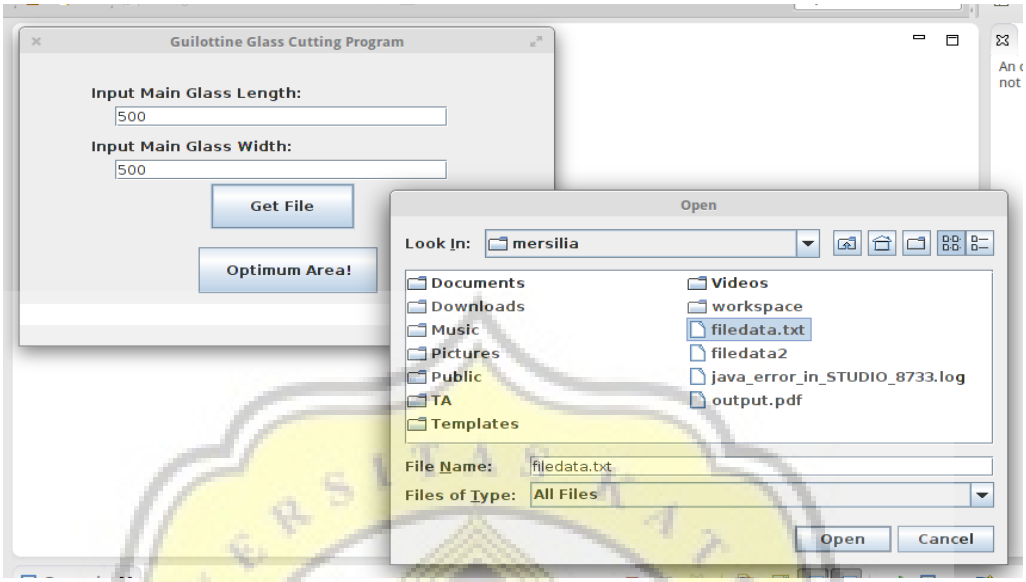

Figure 6 : OpenFile Window

5.1.2. Input Data Processing

In this part of the program will read the data in .txt file then change data from String data type into Integer data type

This is example of .txt file writing format:

rectangle length ,rectangle width , rectangle ID number

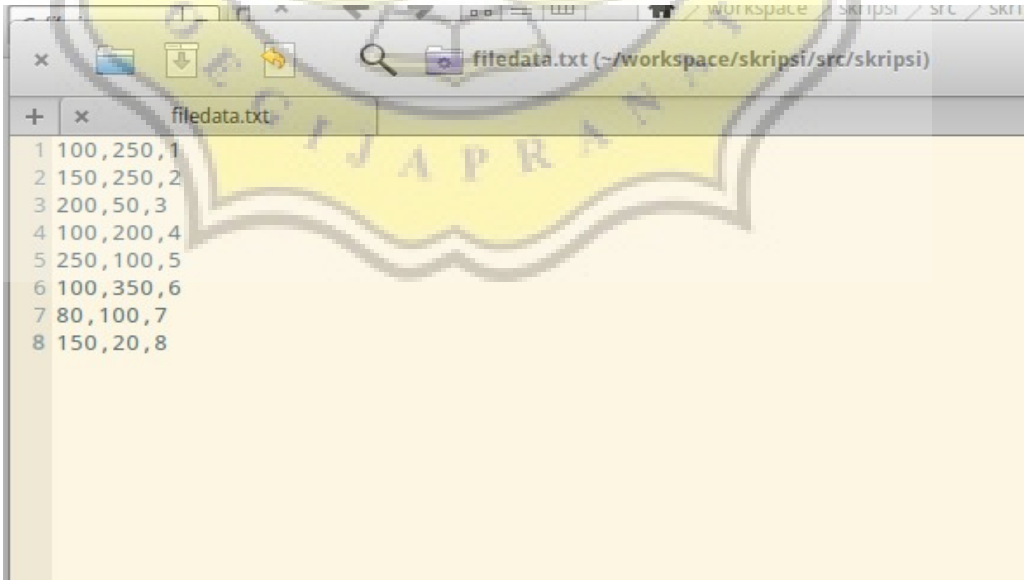

Figure 7 : .txt file format and example

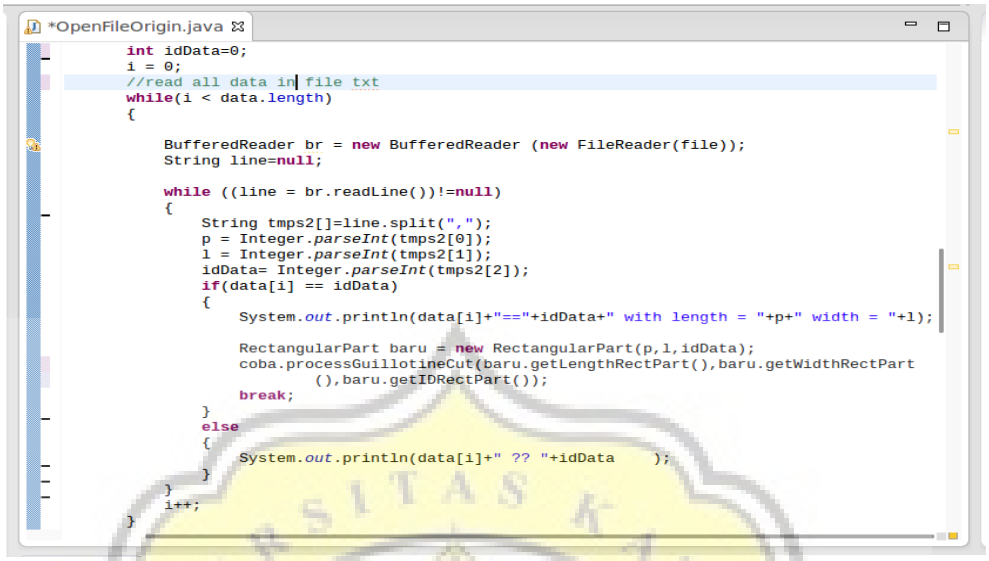

Figure 8 : read txt file and change it to integer data

data.length- data is an array of id number per line, and data.length is count how many ID number/line in txt file choosen before.

BufferedReader read file from

java.io.File file = fileChooser.getSelectedFile();

it read per line and then in each line String data separated by `line3.split(",")` for the length,width,ID number then save it temporary at array `tmps2[]`

String data changed into Integer data with `Integer.parseInt` then save length in p, width in l , ID data in iddata.

5.1.3. Genetic Algorithm Implementation

Genetic algorithm handle data shuffling, waste area generating, and found the optimum result from the 1000 generation

5.1.3.1. Data Shuffling

Data shuffling will used only if the user choose button “Optimum Area!” in user interface. (Figure 5) because the “Get file” button only implementing guillotine cut.

Method shuffleArray(data) handling the shuffling operation for shuffling the sequence of ID data in array data.

```
static void shuffleArray(int[] ar)
{
      Random rnd = ThreadLocalRandom.current()
      for (int i = ar.length - 1; i > 0; i--)
      {
            int index = rnd.nextInt(i + 1);
            int a = ar[index];
            ar[index] = ar[i];
            ar[i] = a;
      }
}
```

Data shuffling is in the OpenFileGenerate.java

5.1.3.2. Waste Area Generating

Genetic algorithm processing 1000 times for this project (processL) , each rocess will reading again the .txt file then swapped data length and width by shuffleArray(revert) before execute shuffleArray(data) for the next process. Swap process makes the combination of rectangle posision rotating 90 degree.

Figure 9 : swapping length and width

and then after width and length swapping, *idData* saved ID number from .txt file. This program read .txt file, shuffling ID number then save in array *data*, read data .txt again, swap the length and width of rectangle and saved ID number in array *idData*, execute the guillotine cut and then save the waste area total.

Figure 10 : execute guillotine cut and save waste area

Waste area get by execute *getEmptySheet()* from guillotine cut algorithm. `matrixLength[j]=coba.getEmptySheet();` *matrixLength[ ]* length declaration must equal or higher than the *processL* length so all of the waste area can be saved.

Next is generating waste area, by search the lowest number of waste area.

minsave=0;

Figure 11 : get lower waste area

when the lower waste area found then save it in *minsave*

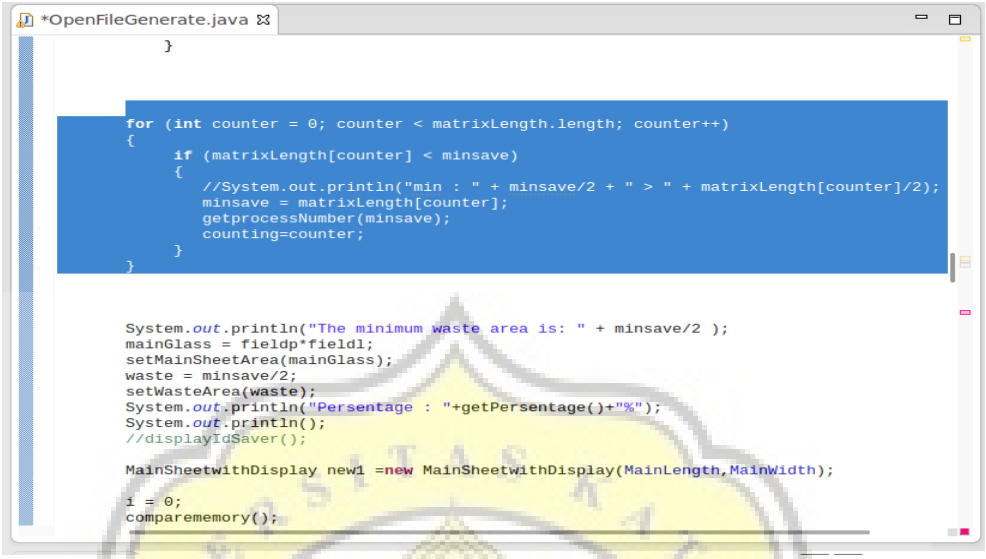

Figure 12 : get lowest waste area

lowest waste area found and save it in *minsave.*

Waste area generating is in the OpenFileGenerate.java

5.1.4 Guillotine Cut Algorithm Implementation

Guillotine cut concept in this project must follows this conditions,

The valid rectangle entered :

logic 1 ; If the rectangular width <= free area main sheet width

logic 2; If the rectangular length <= free area main sheet length

The valid rectangle arrangement condition based on guillotine cut:

- Fill the main sheet (matrix data) from left-above
- Then fill under after check the space if it is still fit, if not then create a vertical line , start placed again from left-above.

- The guillotine cut will pass the rectangle which not fit enter free area in main sheet

Matrix example with data file .txt

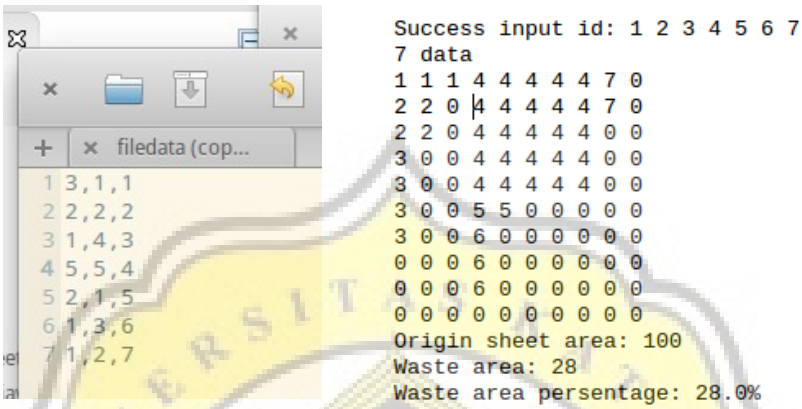

Figure 13 & 14 : .txt file & example matrix data output with guillotine cut condition

Guillotine cut process is in the MainSheet.java (`void processGuillotineCut(int lRect, int wRect, int id)`)

5.1.5. Drawing Rectangles

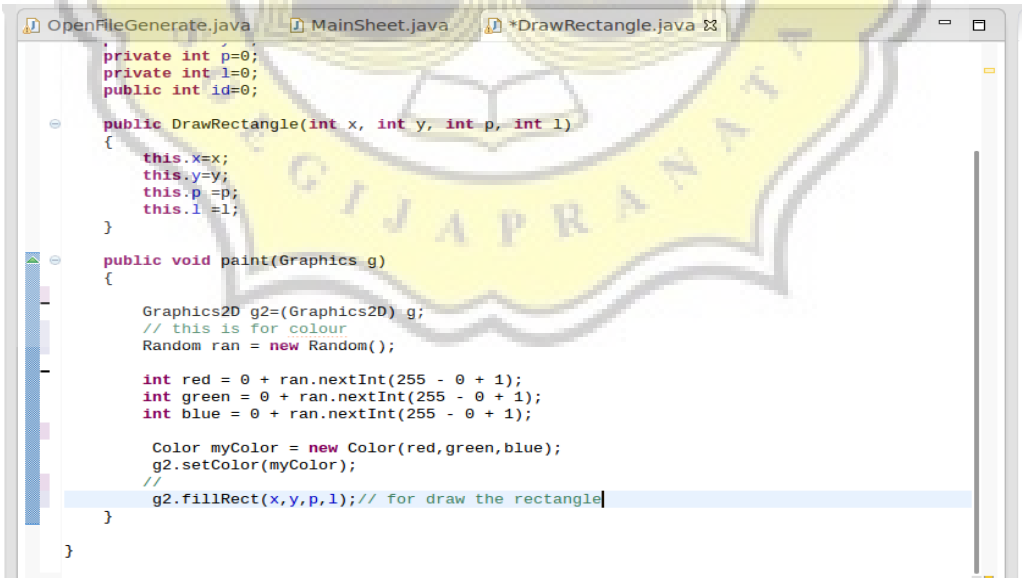

Figure 15 : Drawing rectangle implementation

Drawing rectangle is one of the feature from java

```
import java.awt.geom.Rectangle2D;
import javax.swing.*;
```

To draw the rectangle must have x coordinat, y coordinat, rectangle width, and rectangle length.

fillRect it used for draw Rectangle with colour inside and this project use the random color for each rectangles.

X and Y coordinat get from the formula of last rectangle placed and guillotine cut condition.

Drawing rectangle is in the DrawRectangle.java

5.1.6. Guillotine Cut Layout Algorithm Implementation

Guillotine Cut Layout Algorithm implement by add

```
DrawRectangle rectangle = new
DrawRectangle(0,0,lRect,wRect);
trial.add(rectangle);
trial.setVisible(true);
```

into guillotine-cut function, but for implement easier this project create class guillotine cut with display in MainSheetwithDisplay.java

5.2. Testing

This project run in eclipse Kepler, tested with selected filedata.txt with following format and filled main sheet glass length x width 500 x 500 pixel

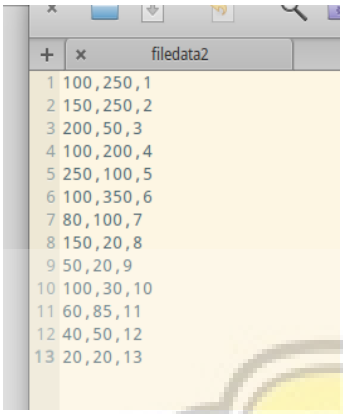

Figure 16 : .txt file for testing

First testing by choose button “Get File” this is the result

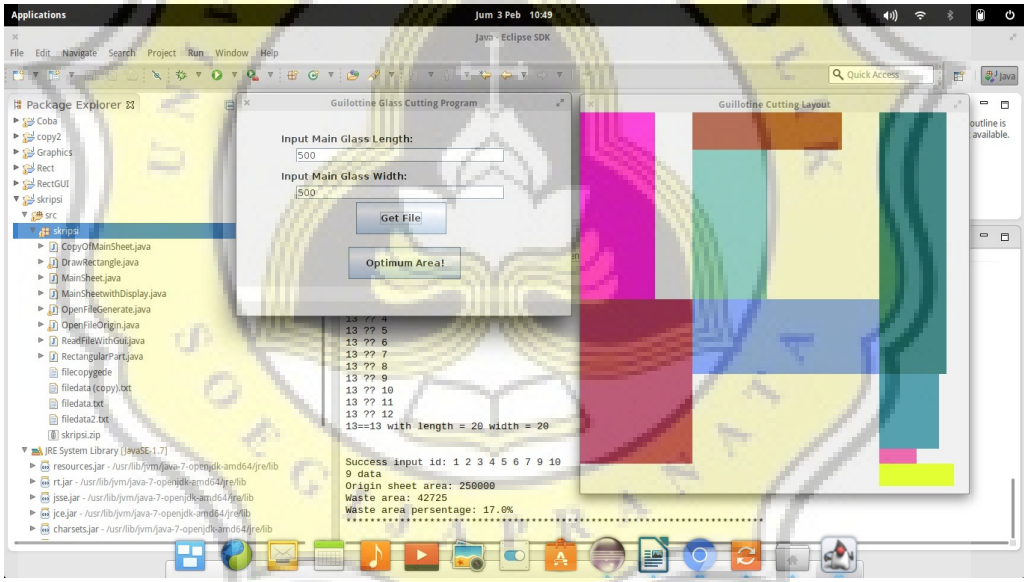

Figure 17 : Guillotine cut layout result

Result above only shows the guillotine cut and not all rectangles entered to the sheet and still in origin sequence of ID number, only 9 data entered.

Compare after implementing both algorithm this is the result if user choose button “Optimum Area!”(Left picture)

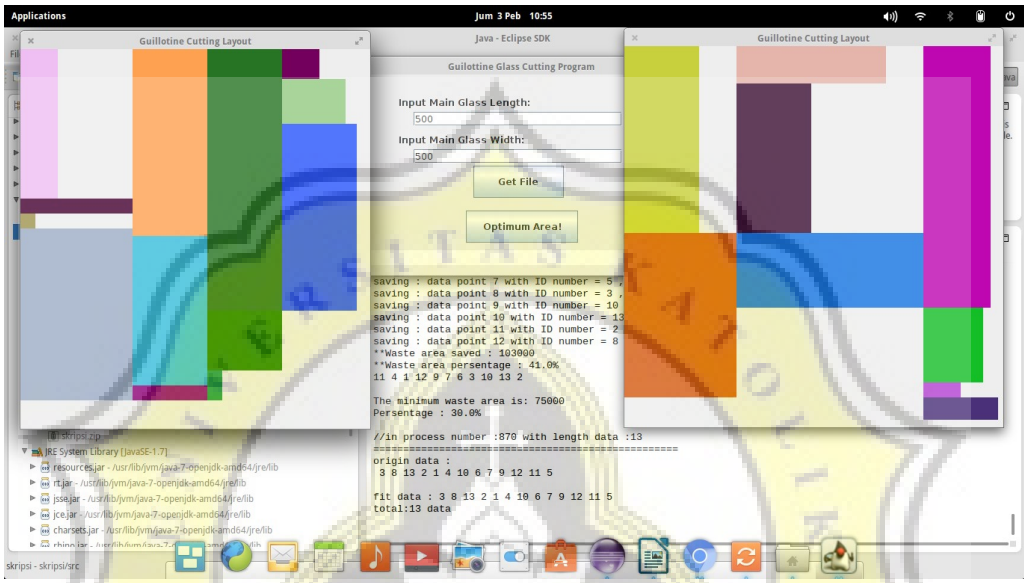

Figure 18:Generated guillotine cut layout result compare

All the data entered to the main glass sheet in the same sheet size.