CHAPTER IV

ANALYSIS AND DESIGN

4.1. Analysis

There are many ways to determine the age of wood, example by looking at how many the annual rings on the wood. Therefore, this analysis was made, by processing the image of annual rings to know how many the number of annual rings. With grayscale method, edge detection, edge linking, thresholding, and chain code.

4.1.1. Use Case Diagram

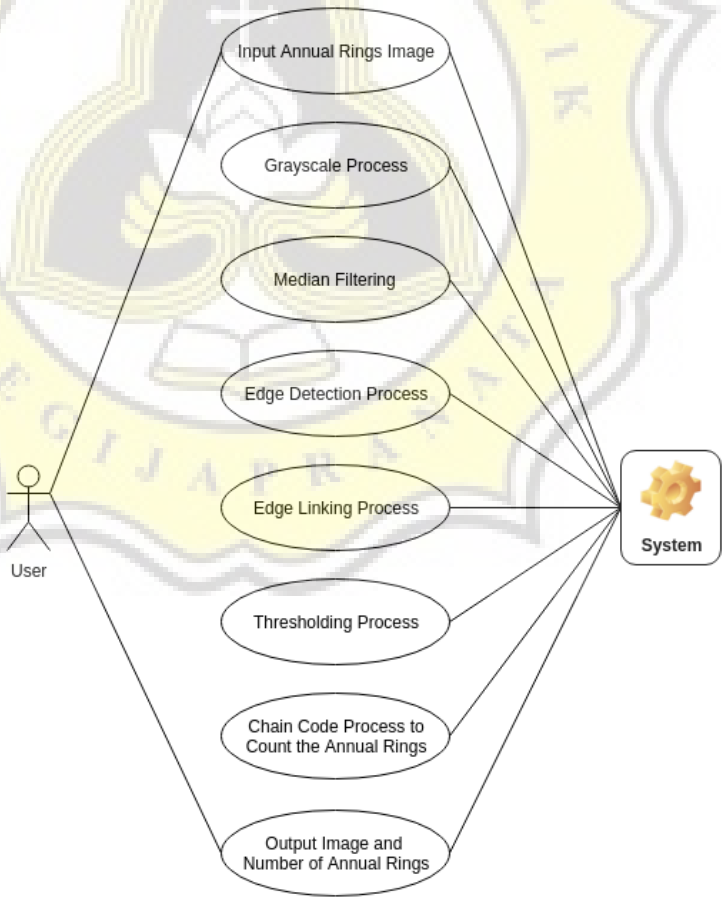

Figure 1. Use Case Diagram

Based on above diagram, the user must input a annual rings image. Input can be in png, jpg or jpeg. Without input from the user then this program will not operate. After the input is received by the system then system will begin to process the input, running the process one by one and give response to the user with the result. The result from the process is an annual rings image that already thresholded along with the sum of the annual rings.

4.1.2. System Working Order

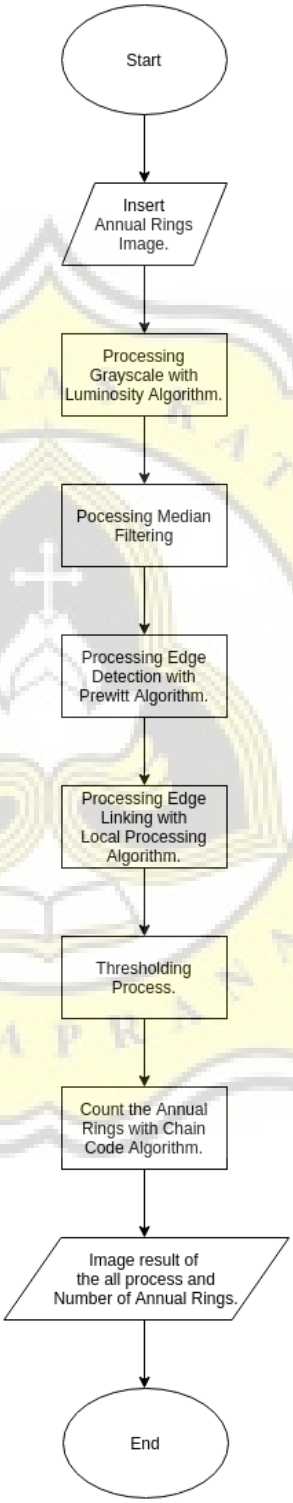

Figure 2. System Working Order

Program will present GUI and wait for the user to input the image. After the image is input then the program will directly start the process, the image will be processed with grayscale process to convert the image into grayscale, then edge detection process to detect the edge, then edge linking to optimize the edge detection process and threshold process.

After the image is processed then will be continuing with calculation of annual rings process. This program will present the image of all process that already processed before and the sum of those annual rings.

4.1.2.1. Grayscale

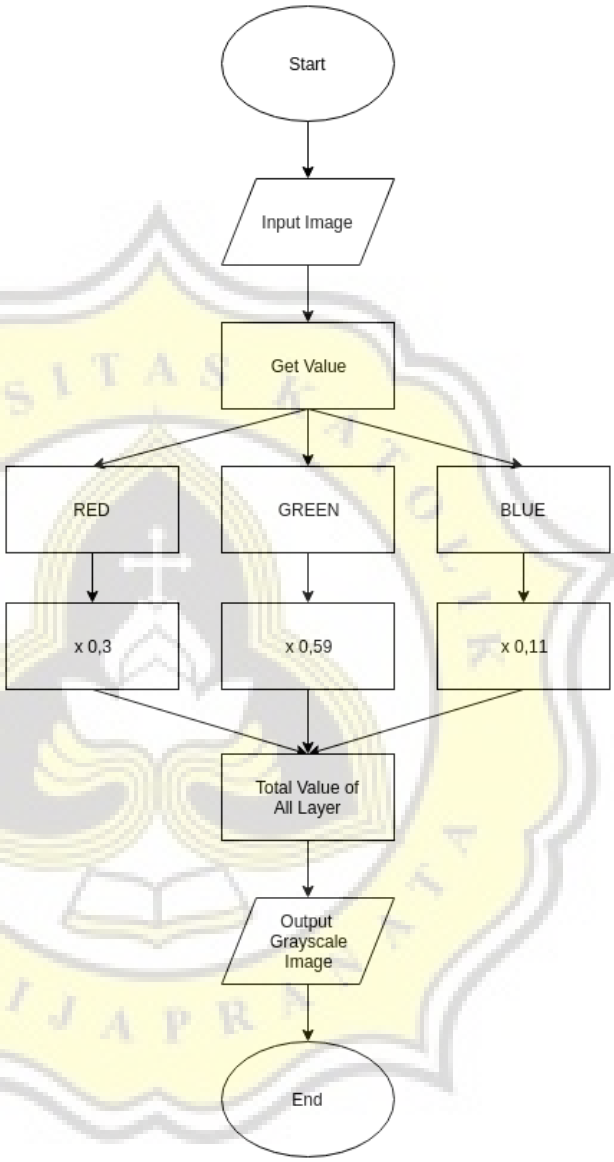

Figure 3. Flow Chart Grayscale

To change object into grayscale in this program using luminosity algorithm, to take value in every layer and operate with the formula below :

(0,3 x RED) + (0,59 x GREEN) + (0,11 x BLUE).

Result value will be entered into the layers red, green, and blue. So that 3 layers has the same value then the image will be grayscale.

4.1.2.2. Median Filtering

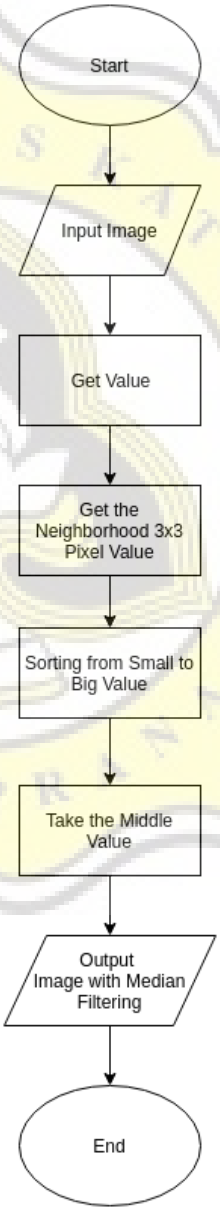

Figure 4. Median Filtering Flow Chart

Median filtering is used to remove noise at image. By taking all value of the neighborhood 3x3 and sorted it from small to large. Then take the middle value and becomes the new value of that pixel.

4.1.2.3. Edge Detection

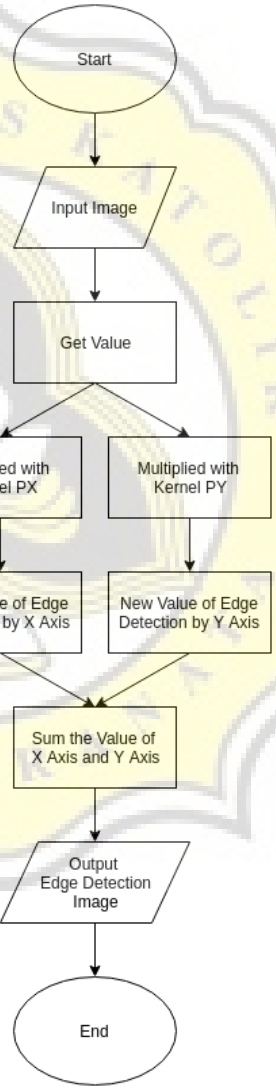

Figure 5. Edge Detection Flow Chart

Edge detection can make the edge / line in the pictures could be more visible than other objects. In this program uses prewitt kernel for edge detection :

1	1	1
0	0	0
-1	-1	-1

Kernel Py

-1	0	1
-1	0	1
-1	0	1

Kernel Px

Figure 6. Edge Detection Kernel

After pixel multiplied by the kernel will then become the new pixel value.

4.1.2.4. Edge Linking

Edge linking is optimality from edge detection process to connect every doted pixels. This program is using local processing edge linking. With comparison the pixel itself into it's neighbor, there are 2 options to process this, 3x3 or 5x5

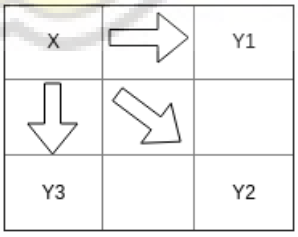

Figure 7. Edge Linking 3x3

X = Processing Pixel

Y = The Neighbours

Before compare the pixel with the neighbor, first we have to find the pixel angle direction by taking value from edge detection by X Axis and Y axis.

With :

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_x}{G_y}\right)$$

The condition that have to be fulfilled are followings :

$$\left|\nabla f(x,y) - \nabla f(x',y')\right| \leq T$$

- Comparing the pixel value with the neighbors.
- T is the positive threshold limit.

$$\left|\alpha(x,y) - \alpha(x',y')\right| < A$$

- Comparing the edge pixel angle direction with the neighbors.
- A is the positive threshold limit.

If this condition is fulfilled, then a connection between 2 pixels will be established.

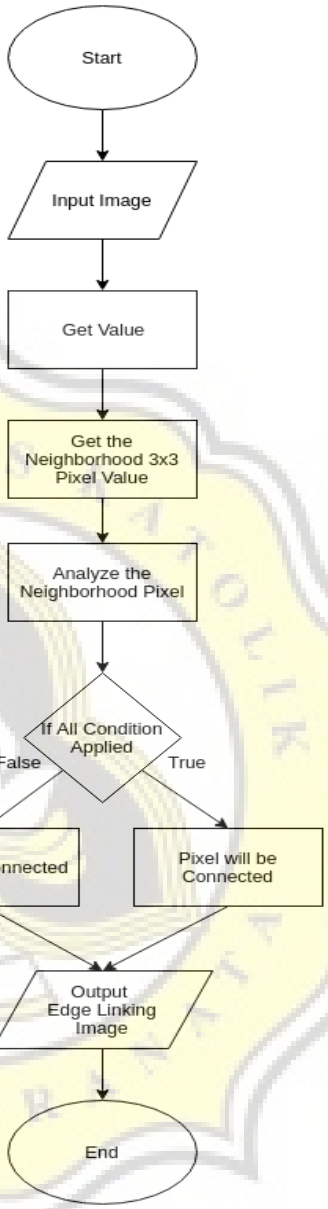

Figure 8. Edge Linking Flow Chart

4.1.2.5. Thresholding

Thresholding is used to convert image from certain range value into determined value, so that image pixel value only contains from above determined value.

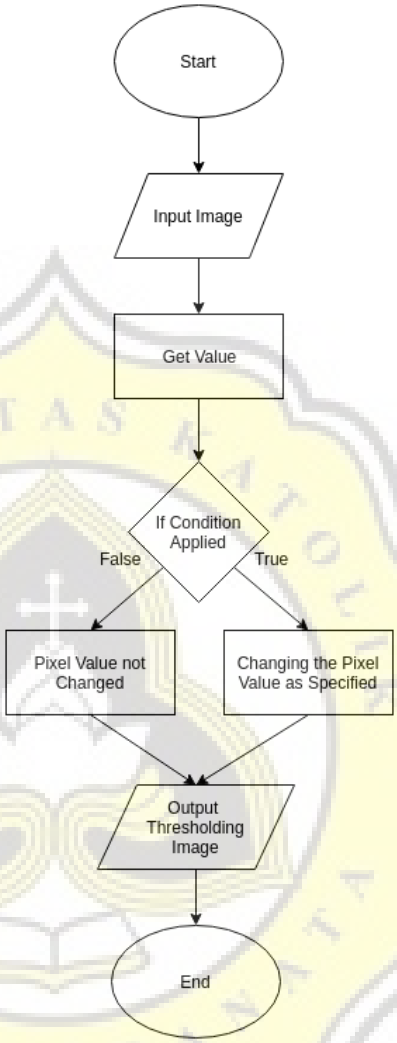

Figure 9. Thresholding Flow Chart

4.1.2.6. Chain Code

Chain code is an pattern recognition algorithm that identified pattern based on pixel direction. Before that should determine each direction code, then detect from the edge of the first pixel until the last pixel and make a note for all of the codes.

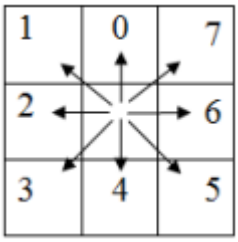

Figure 10. Chain Code Direction

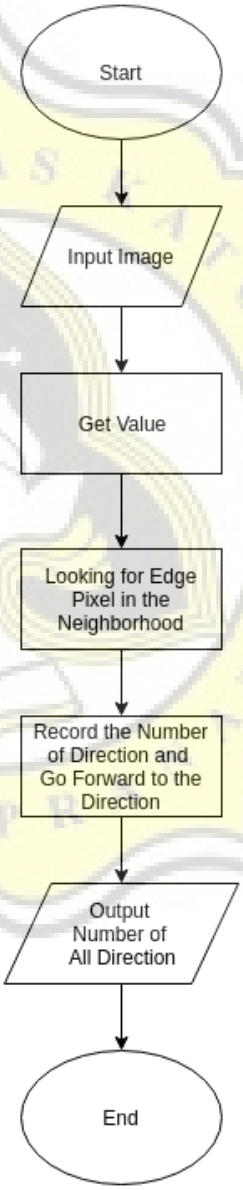

Figure 11. Chain Code Flow Chart for Edge Pixel

4.2. Design

4.2.1. Class Diagram

This is the explanation of some class that will be created :

1) Class woodGUI

   This is main class, will be run first.

2) Class GUI

   Class where contain all about interface user and to operate the program.

3) Class processGUI

   Class to operate continuity process and perform the final result.

4) Class grayscale

   Class to process a annual rings image into grayscale.

5) Class median

   Class to apply median filter to the annual rings image.

6) Class edetect

   Class to detect the edge of the annual rings image that already passed the grayscale process.

7) Class elinking

   Class to connect lines that separate due to noise.

8) Class threshold

   Class to do threshold to annual rings image that already detected before.

9) Class count

   Class to calculate the annual rings from previous threshold process.

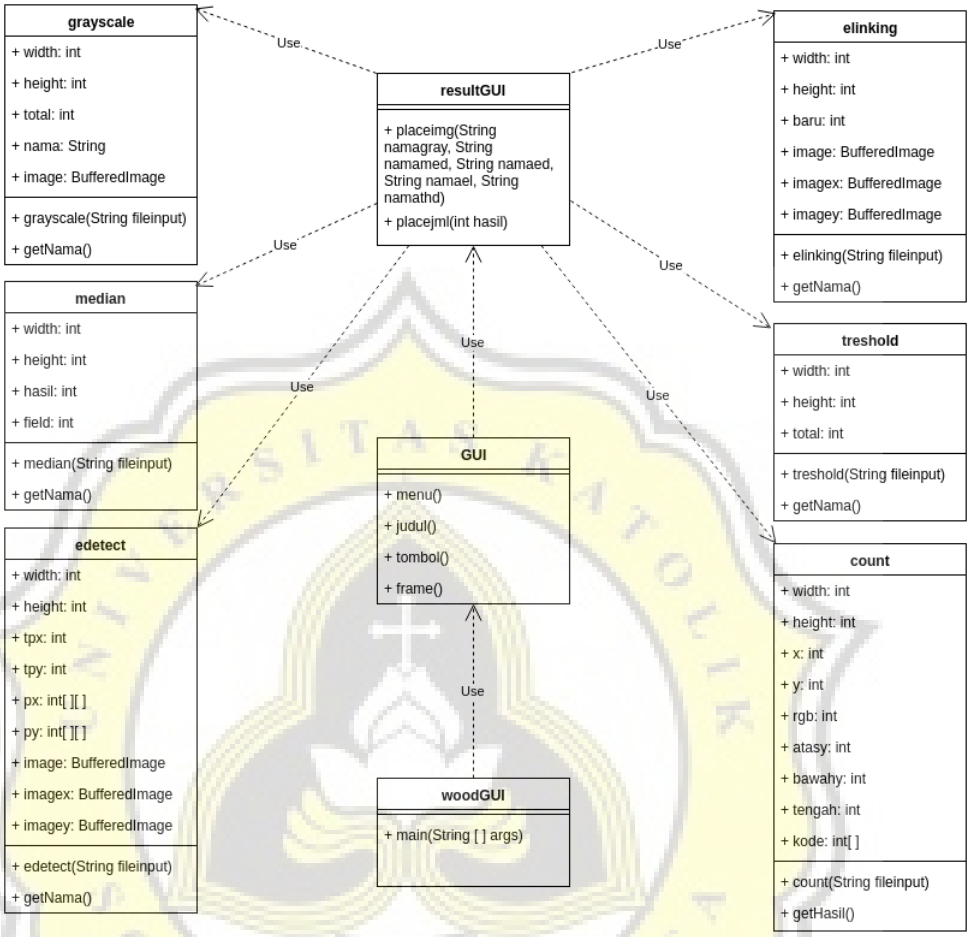

Figure 12. Class Diagram

To run the program, the user have to open “woodGUI” which is a test drive class and in that class make “GUI” class object. “GUI” class is a class that contains an user interface, in this class there are image files input that will be processed later on. When the process started, it will start to process the image one by one.

After the image processing is done, it will make “resultGUI” class object, in this stage the program will perform the final result from every stage, and the number of annual rings.