CHAPTER 4
ANALYSIS AND DESIGN

4.1 Analysis

Otsu is a segmentation algorithm that mainly used for discrete images. While the problem in this project is camouflage images usually not suitable if segmented with Otsu method, because those images usually contains similar color between object and background. So the discrete images must be used alongside camouflage images for data comparison on Otsu method. Manual thresholding which its threshold value determined by user is also used in this project, for result comparison with Otsu method.

4.2 Design

4.2.1. FLOWCHART OF PROGRAM

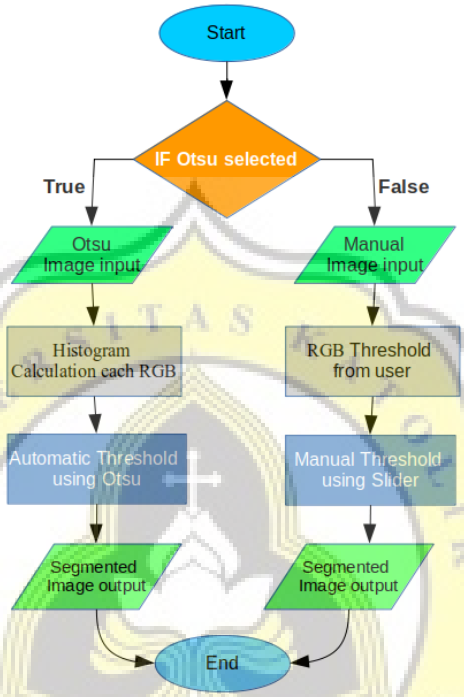

Illustration 4.1: Flowchart of Program Steps

Data sample which prepared for this research as much as 40 images that consists of 20 discrete images and 20 camouflage images, all images have resolution from 256x256 pixels until 1366×768 pixels.

If Otsu method is chosen, the next step is image input, then histogram calculation is done at each color channel (R,G,B) with data range 0-255. After histogram calculation is finished, Otsu calculation is done as automatic thresholding.

If Otsu method is not chosen, then manual thresholding is chosen. The next step is image input, then specified RGB threshold from user. Sliders are used as determination of threshold values from 0-255, so this method is called as manual thresholding.

The detailed process of these two methods will be explained on the next flowcharts. After image processed in Otsu method or Manual threshold, original image and segmented image are showed in GUI as the output.

4.2.2. OTSU METHOD FLOWCHART

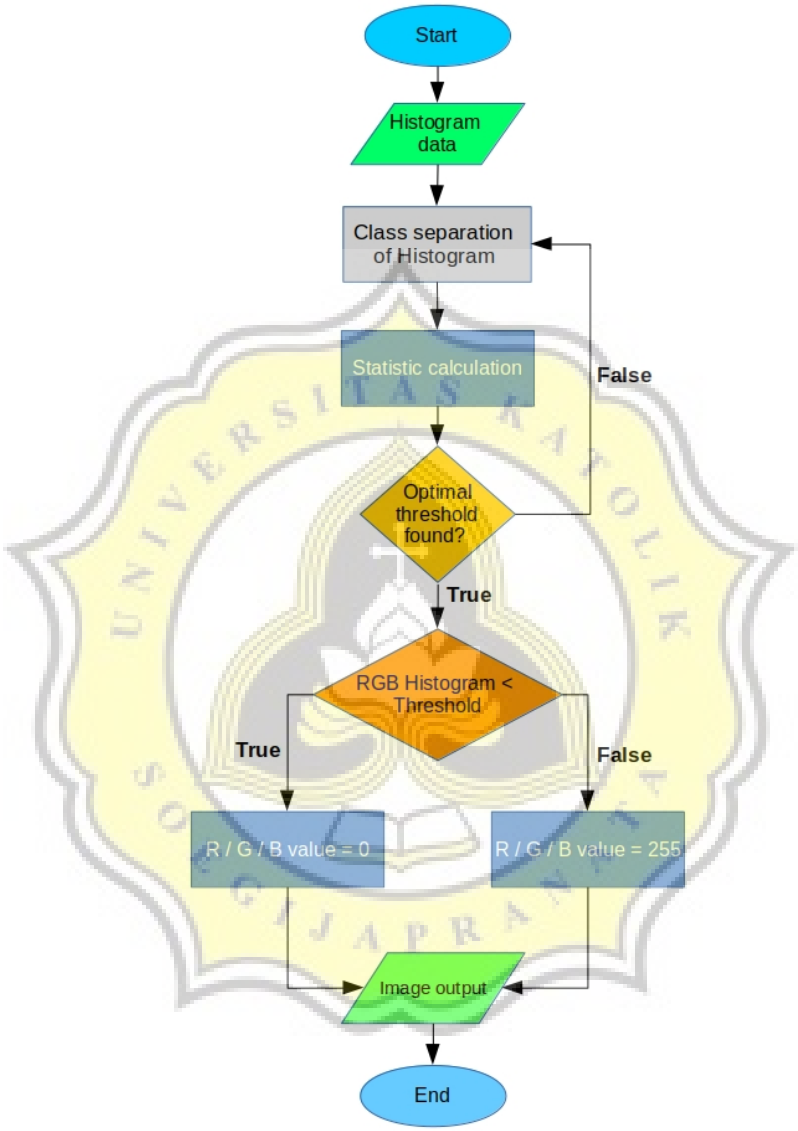

Illustration 4.2: Otsu Flowchart

Based on flowchart above, the first step is histogram data usage as data input. Then histogram is divided into two classes based on threshold: Background & Foreground, from 0 until 255 looped threshold value. Next step is statistic calculation including the weight, mean, variance, and between class variance of each color channel histogram. In the last calculation step, maximum value of

between class variance is searched in all threshold (T) 0-255. While the within class variance is not calculated in this project, because it is an optional method which has minimum value.

Here is formula of Otsu method:

Background

Weight $W_b = \frac{f_i}{N}$

Mean $\mu_b = \frac{\sum i.fi}{\sum fi}$

Variance $\sigma_b^2 = \frac{\sum fi(i-\mu_b)}{\sum fi}$

Foreground (Object)

Weight $W_f = \frac{f_i}{N}$

Mean $\mu_f = \frac{\sum i.fi}{\sum fi}$

Variance $\sigma_f^2 = \frac{\sum fi(i-\mu_f)}{\sum fi}$

Between Class Variance $\sigma_B^2 = W_b W_f (\mu_b - \mu_f)^2$

Within Class Variance $\sigma_W^2 = W_b \sigma_b^2 + W_f \sigma_f^2$

Information:

fi = Frequency of pixels on each color level i

N = Total of pixels

i = Color level

The maximum value of between class variance is used as optimal threshold. Suppose the maximum value is found at T = 130, then color value of optimal threshold = 129 (counted from 0 in iteration value of looping). Optimal threshold is used for binarization: If color value on histogram < optimal threshold then color value is changed to 0. If color value >= optimal threshold then color value is changed to 255. Finally binarization results from three separated histogram are merged into one image as an output.

4.2.3. MANUAL THRESHOLDING FLOWCHART

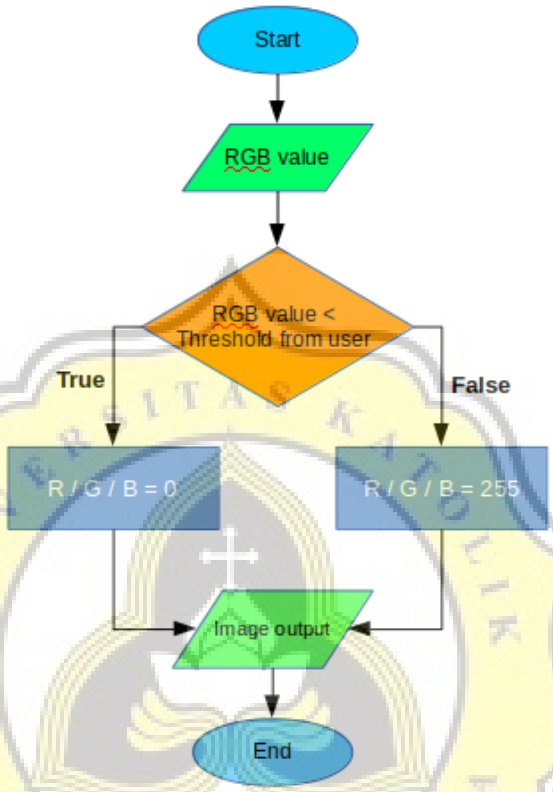

Illustration 4.3: Manual thresholding Flowchart

Manual thresholding only need RGB value as input, without histogram data. Threshold values are obtained from user input on each R,G,B slider (0-255). Those three values directly become a threshold for binarization, without statistic calculation because it does not search for optimal threshold like Otsu method. If the RGB value in the histogram < threshold then the color is changed to black (0). If the value of RGB >= threshold then the color is changed to white (255). Finally binarization results are merged into one image as an output.