

## CHAPTER 4

### ANALYSIS AND DESIGN

#### 4.1 Analysis

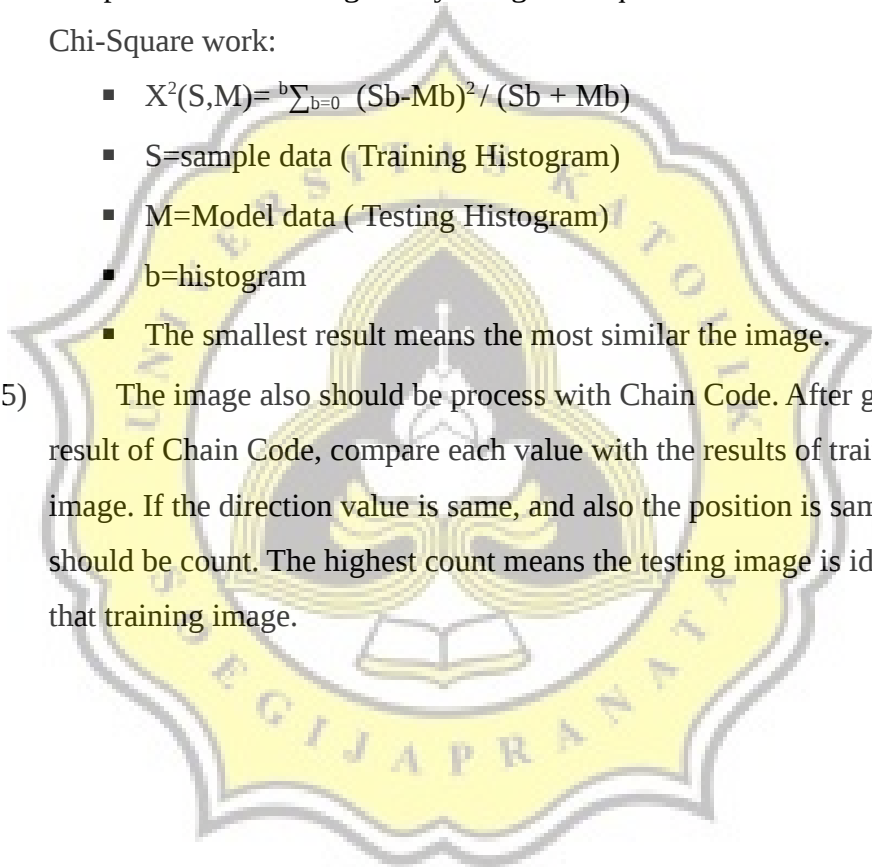
- 1) Input and collect train image data. Each train image must have the same size (500 x 500) and on grayscale color, so each train image should get image preprocessing first, then each image should have a Label name / category. The address and the label name of those image should be save in “daftarGambar.txt”.
- 2) Train all of the training image with ALBP. Read the “daftarGambar.txt” to get the image address. Each image should be process with ALBP, and the result (histogram) should save in Array 2D.  
How ALBP works?
  - Each pixel must be compared to each of its neighbors (8 neighbors), the center pixel become the comparasion value.
  - If the neighbors value is same/greater than center pixel, so it give ‘1’ value.
  - If the neighbors value is lower than center pixel, so it give ‘0’ value.
  - Each value is merged into 8bit binary. Example : 00110011, 11111111
  - The binary should be shifted one by one to the left (rotary), and the smallest value(decimal) stored in histogram.
- 3) Train all of the training image with Chain Code. Read the “daftarGambar.txt” to get the image address. Find the first black pixel, and mark it as Start point, from that point, detect the next connected black neighbor in a clockwise direction, if there are more than 1 neighbor, choose the first detected neighbor and then move to that neighbor and mark as visited. The neighbor that already visited, cannot be visit again. Record that direction of moving. When it traverse to the start point mark it as end point, and it’s mean that the Chain Code has already finish to detect the

shape(2d). If the Chain Code couldn't find / move, because there's nothing neighbor left, mark it as end point, and it's mean that the shape is a line.

- 4) Testing Image. Choose the image, then it should get image preprocessing (resize, grayscale, edge detection, thinning), after preprocessing the image should be process with ALBP, the result compared to each histogram by using Chi-Square method. This is how Chi-Square work:

- $X^2(S,M) = \sum_{b=0}^b (S_b - M_b)^2 / (S_b + M_b)$
- S=sample data ( Training Histogram)
- M=Model data ( Testing Histogram)
- b=histogram
- The smallest result means the most similar the image.

- 5) The image also should be process with Chain Code. After getting the result of Chain Code, compare each value with the results of training image. If the direction value is same, and also the position is same, it should be count. The highest count means the testing image is identic with that training image.



## 4.2 Desain

### A) Flowchart workflow program in general

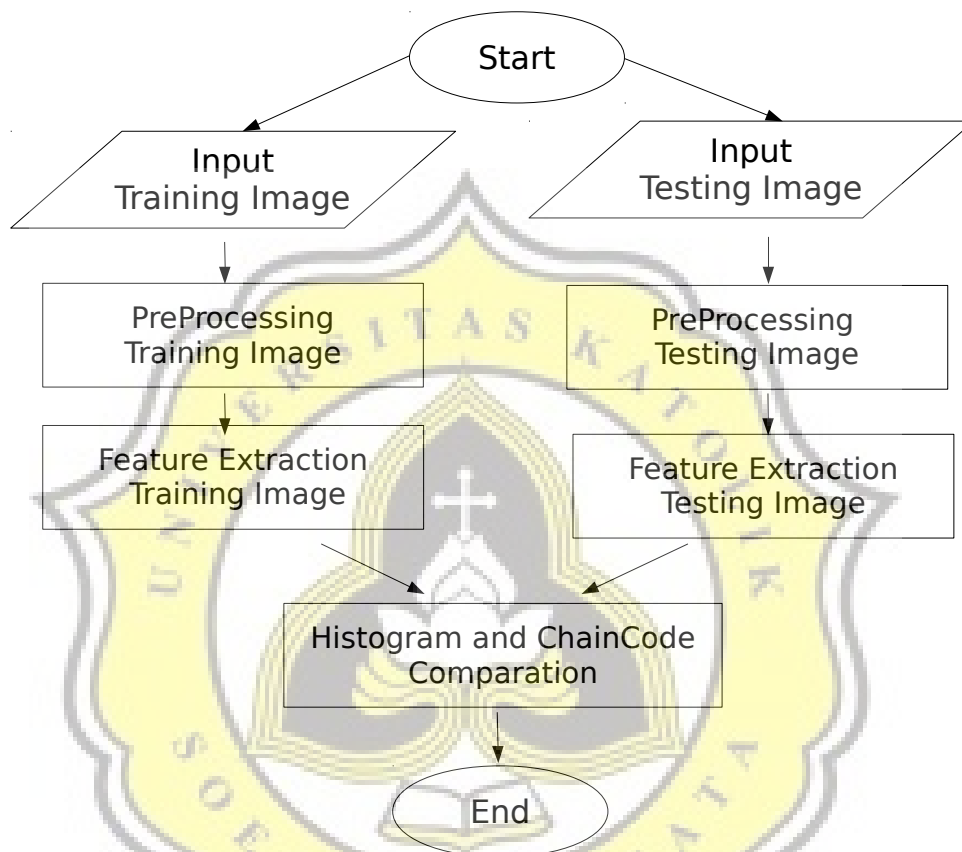


Illustration 4.1: Main Flowchart

Here is the detail from that flowchart, first of all input several training image for the system, the images need to be normalized, so every images should get image preprocessing, after it the feautre of train images should be extracted and the results should be stored. The next step, input the image that want to test. The testing image also should get image preprocessing. Then extract the feature from that image. After it, compare the histogram and chain code's result from the testing image to each histogram and chain code's result from the training image.

## B) Preprocessing Flowchart

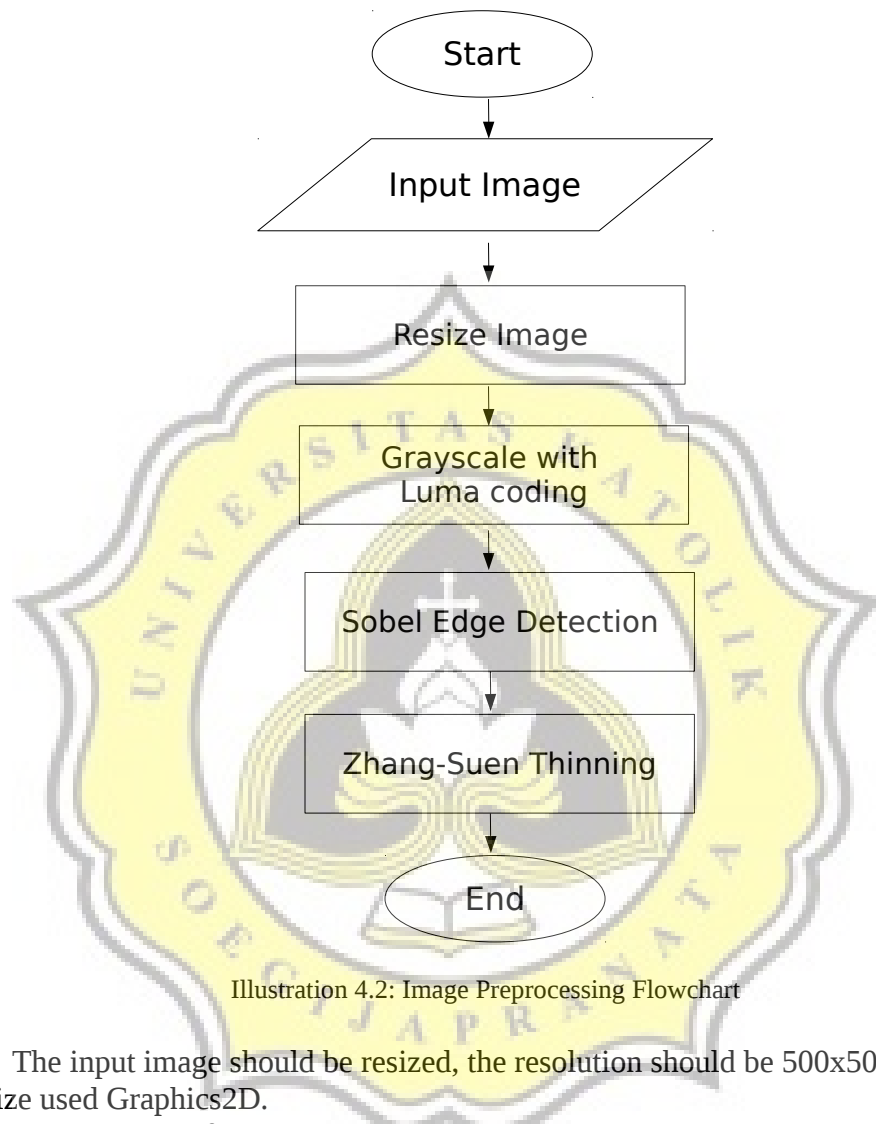


Illustration 4.2: Image Preprocessing Flowchart

The input image should be resized, the resolution should be 500x500. The resize used Graphics2D. After it, the color of image should be grayscale, the grayscale method is using luma coding grayscale.

$$Y' = 0.299 R + 0.587 G + 0.114 B$$

And then, apply the sobel edge detection on that image. Every 3x3 pixels will be multiplied by the Gx kernel, and also Gy kernel, then calculate the G value

$$G = \sqrt{G_x^2 + G_y^2}, \text{ G value would replace the value of center pixel.}$$

-1	0	+1
-2	0	+2
-1	0	+1

G<sub>x</sub>

+1	+2	+1
0	0	0
-1	-2	-1

G<sub>y</sub>

Illustration 4.3: Sobel Kernel

After apply sobel edge detection, the edge should have 1 pixel thinness, so the image should get Zhang-Suen Thinning Algorithm. The step is:

- I. Define  $A(P1)$  = the number of transitions from white to black, in the sequence  $P2, P3, P4, P5, P6, P7, P8, P9, P2$ .
- II. Define  $B(P1)$  = The number of black pixel neighbours of  $P1$ .
- III. Step 3

All pixels are tested and pixels satisfying all the following conditions (simultaneously) are just noted at this stage.

(0) The pixel is black and has eight neighbours

(1)  $2 \leq B(P1) \leq 6$

(2)  $A(P1) = 1$

(3) At least one of  $P2$  and  $P4$  and  $P6$  is white

(4) At least one of  $P4$  and  $P6$  and  $P8$  is white

P9	P2	P3
P8	P1	P4
P7	P6	P5

Illustration 4.4:  
Thinning

After iterating over the image and collecting all the pixels satisfying all step 3 conditions, all these condition satisfying pixels are set to white.

- IV. All pixels are again tested and pixels satisfying all the following conditions are just noted at this stage.

(0) The pixel is black and has eight neighbours

(1)  $2 \leq B(P1) \leq 6$

(2)  $A(P1) = 1$

(3) At least one of  $P2$  and  $P4$  and  $P8$  is white

(4) At least one of  $P2$  and  $P6$  and  $P8$  is white

- V. After iterating over the image and collecting all the pixels satisfying all step 2 conditions, all these condition satisfying pixels are again set to white.

## C) Feature Extraction Flowchart

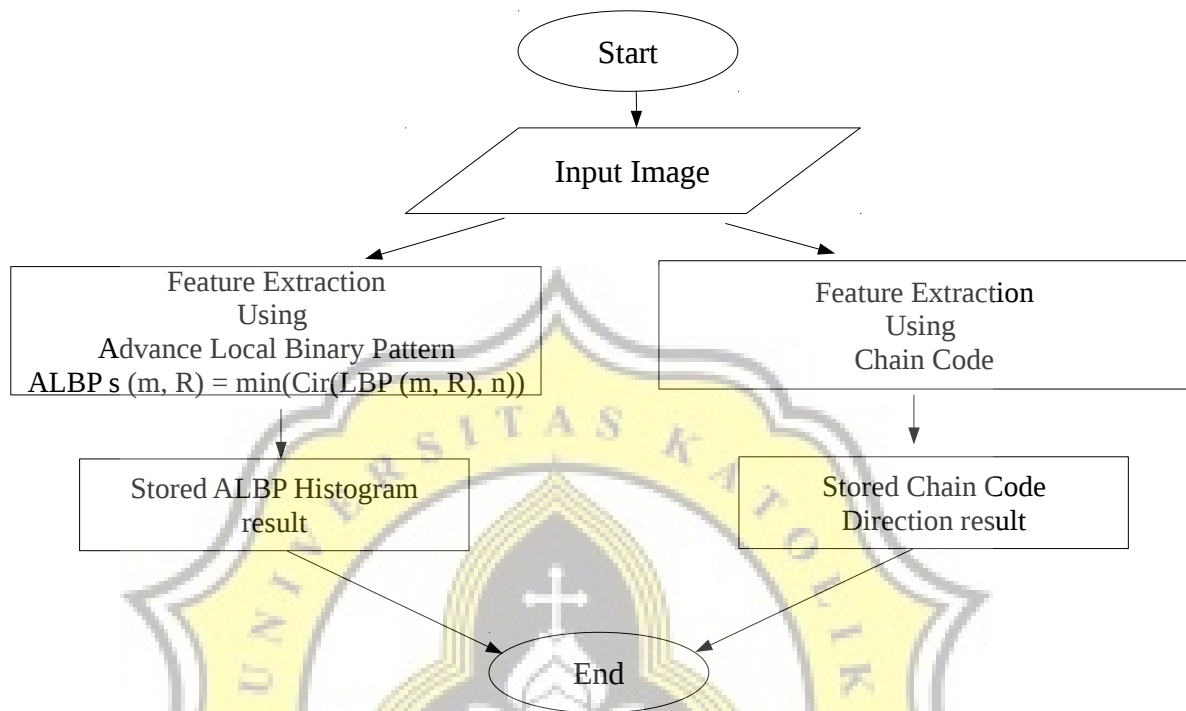


Illustration 4.5: Feature Extraction Flowchart

After the input image have already normalized / got image preprocessing, then the image can be extracted its features. In this research used 2 feature extraction, first is advanced Local Binary Pattern, and the another one is Chain Code. The feature extraction process using advanced Local Binary Pattern is each pixel is compare the center point with the neighbors, if the neighbors is equal or bigger than the center point, it give '1' value, if lower than it give '0' value. The value will be combined in a clockwise order. The different between ALBP and LBP is, after the value is combined, it will performs a circular anti-clockwise bitwise shift and the smallest value will be stored, in example, after the value is combined, it get '1110000', so the first step is to shift the value 1 step and it will become '1100001', shift again and it become '10000011', do it again '00000111', and so on until it perform full shift. On that example, from '11100000' the smallest value is '00000111', so '00000111' is stored.

$$LBP(m, R) = \sum_{i=0}^{p-1} u(t_i - t_c) 2^i \quad ALBP(m, R) = \min(Cir(LBP(m, R), n))$$

The way chain code works is, first find the start point, by trace from pixel (0,0) until the end corner of image (x,y), the first detected black pixel (0) would be a start point, then find the connected black pixel around the neighbors in a clockwise direction, and it would move to the first connected white pixel and record that direction, the visited pixel should be mark and not allowed to be visited again. Repeat it until its come to the start point and that would be an end,

it's mean that the shape is 2d. If the program cant find another path, but not yet reached the starting point, it means the last pixel is the end point, and it's means that shape is a line.

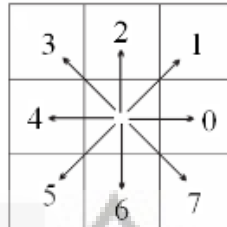


Illustration 4.6: Chain Code Direction

#### D) Chi Square Flowchart

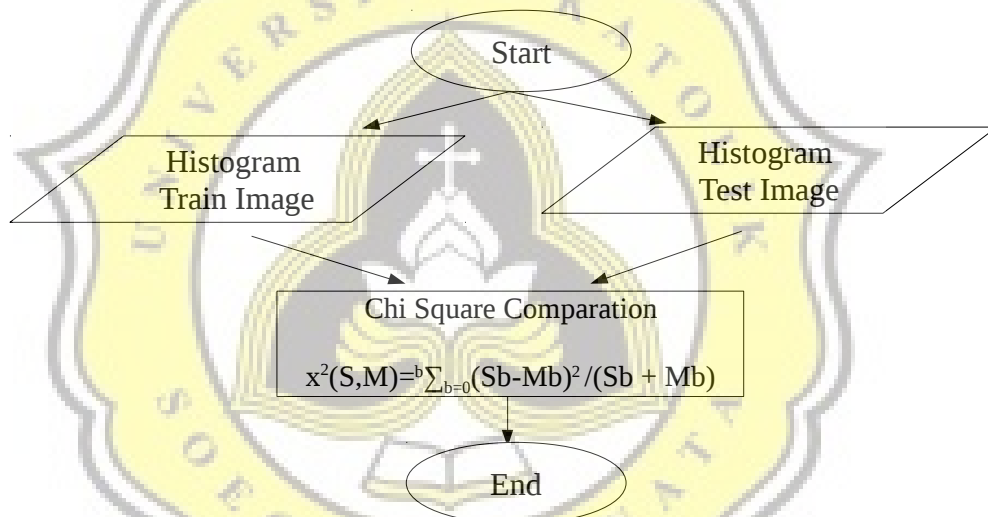


Illustration 4.7: Chi Square Flowchart

The final step from this is project is comparing the histogram from train image with test image by using Chi-Square. The process is :

- get the histogram value from train image and test image
- example data, train Image ALBP histogram : 0=5, 1=6, 2=0, 3=1, 4=10
- example data, test image ALBP histogram : 0=1, 1=3, 2=5, 3=1, 4=10
- calculate each index with chisquare, then sum it.
- And do it for the next train image histogram
- the smallest chi square value, that means the most similar the image.

$$X^2(S, M) = \sum_{b=1}^B \frac{(S_b - M_b)^2}{S_b + M_b}$$

## E) Chain Code Comparison

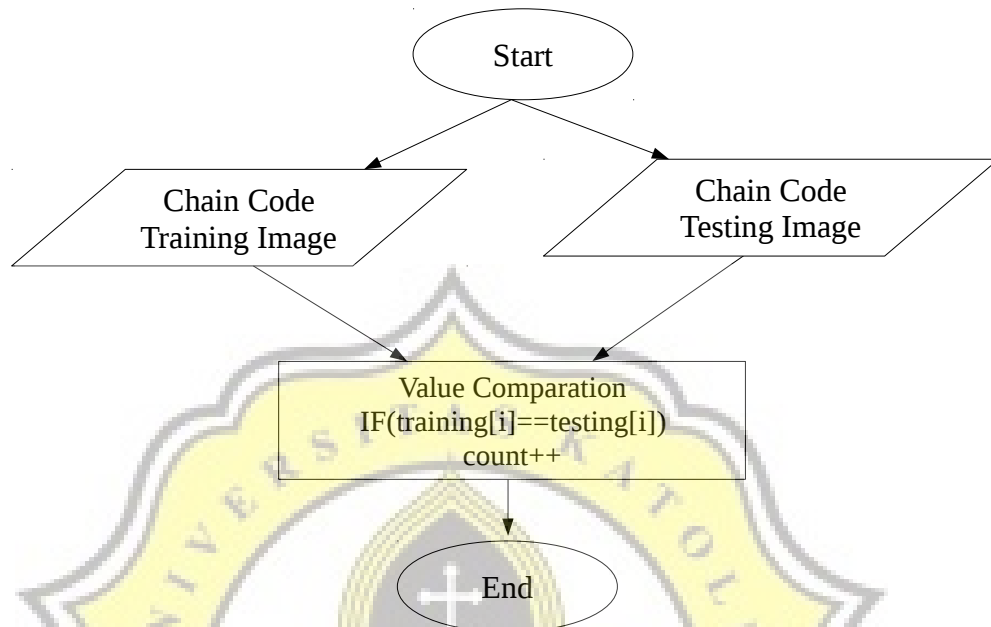


Illustration 4.8: ChainCode Comparataion Flowchart

Each step direction from testing image would be compare to each step direction on training image, if the value is same, it would be count. Example : the chain code from Testing Image is : 0,0,0,6,6,4,4,4,2,2 and the Training Image is : 0,0,0,6,6,4,4,4,4,2,1. So it has 9 same values from 10 values, it has 90% similarity.