CHAPTER 4
ANALYSIS AND DESIGN

4.1 Desain

A) Workflow Program in General.

Illustration 4.1: Main Flowchart

Illustration 4.1 is the program workflow flowchart, beginning the input image and the image from the database must be preprocessed, after that the

extraction of each image's features, after extraction the feature is obtained then compare the results of the adaptive chaincode from both images.

B) Preprocessing Flowchart

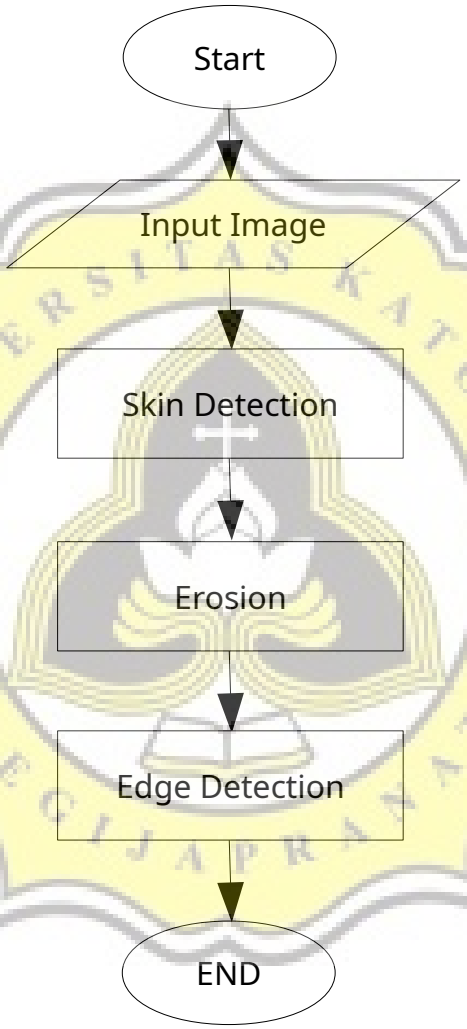


Illustration 4.2: Image PreProcessing Flowchart

Illustration 4.2 is the flowchart from the preprocessing image. It starts with the detection of the skin and then thickens the image and ends with edge detection.

C) Feature Extraction Flowchart

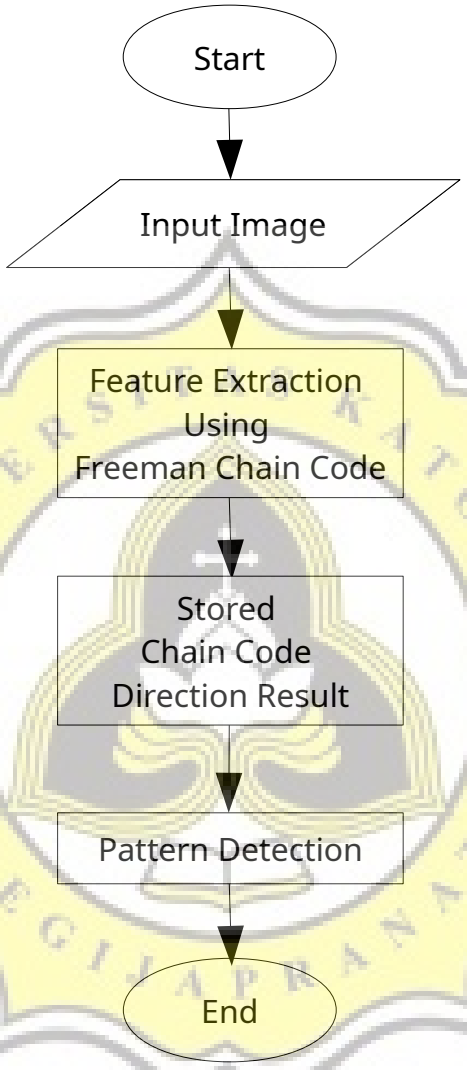


Illustration 4.3: Feature Extraction Using Freeman Chain Code

Illustration 4.3 is the flowchart of feature extraction, the first extraction feature is done using the Adaptive Freeman Chain Code algorithm, then the results are stored into an arraylist and the pattern is determined.

D) Comparation of Similarity Flowchart

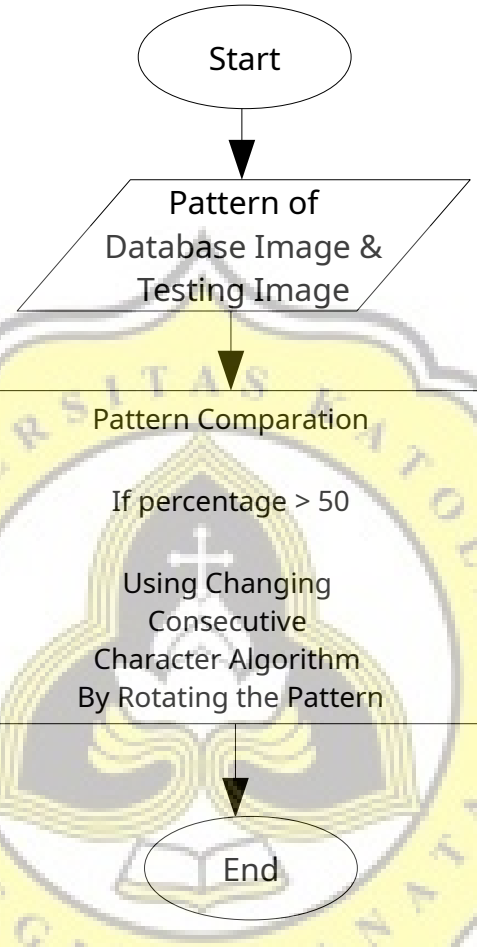


Illustration 4.4: Comparation of Similarity Flowchart

Illustration 4.4 is a flowchart for comparison of pattern similarity, limited by a percentage above 50 will be considered as eye / mouth. Table 4.3 is example of the results of the adaptive chaincode of input images and training images.

4.2 Analisys

To get optimal results there are several pleliminary studies that have been conducted. Skin detection, consisting of 2 compositions used to deal with Indonesian people's skin. Adaptive Freeman Chain Code which functions to obtain short Freeman Chain Code to proceed to the pattern matching process. Area Of Interest to sort out objects that are objects of research. Possibility of face to determine the face area in the wedding image so that the eye and mouth pattern can be found. Determine the Eyes and Mouth Position to determine the possible position of the eyes and mouth on the face. Adaptive Consecutive Changing Characters is done to overcome the similarity of patterns with different angles between testing images and images in the database. Below are explanations of preliminary studies;

1) Skin Detection

The first method in the preprocessing step in this research is skin detection. According to *Abdellatif Hajraoui, Mohamed Sabri, (2014)* to classify pixels from RGB images to two classes (skin and non-skin) can use 4 kinds of space colors, namely RGB color space, Normalize RGB color space, HSV color space, and YcbCr color space. The result of this step is a binary image.

Table 4.1 is a table of the provisions of each color space :

Table 4.1: Table of the provision of RGB , Normalize RGB , HSV, YCbCr color space

RGB	Normalize RGB	HSV	YCbCr
R>95 and G>40 and B>20	R>95 and G>40 and B>20	$0 \leq H \leq 50$	$97.5 \leq Cb \leq 142.5$
(max (R,G,B) - min(R,G,B))>15	(max (R,G,B) - min(R,G,B))>15	$0.2 \leq S \leq 0.68$	$134 \leq Cr \leq 176$
R-G >15	R-G >15	$0.35 \leq V \leq 1$	
R>G	R>G		

R>B	R>B		
	$0.36 \leq r \leq 0.465$		
	$0.28 \leq g \leq 0.363$		
**Normalize RGB Formula With :* $r=\frac{R}{R+G+B}$ and $g=\frac{G}{R+G+B}$			

Examples of the implementation are in the illustration 4.5.

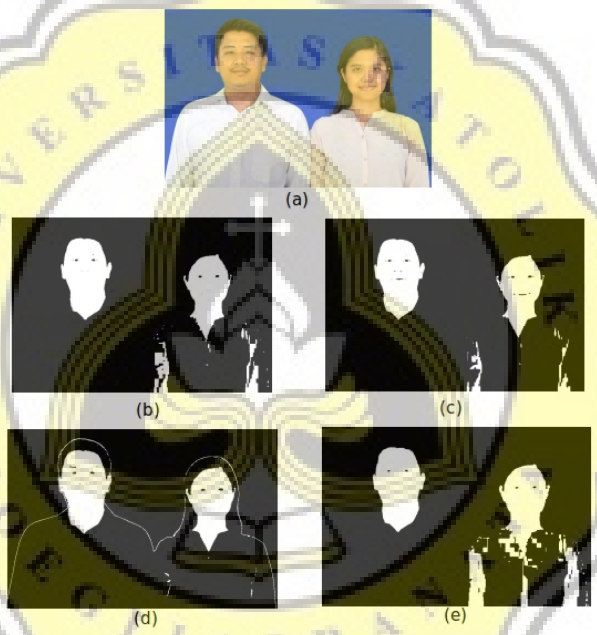


Illustration 4.5: a) Original Image
b) RGB Color Space Image
c) Normalize RGB Color Space Image
d) HSV Color Space Image
e) YCbCr Color Space Image

image source : *https://nyonyamalas.com/blog/2015/09/22/12-dokumen-syarat-administrasi-catatan-sipil-pernikahan-file-doc-available/*

From illustration 4.5 this research using the Normalize RGB Eye color space because the resulting binary image shows the shape of the eyes and mouth. The following conditions are used in the program:

Sample pixel : RGB (123,64,23)

```
If ( Pixel satisfies the condition C1 )
    If ( Pixel satisfies the condition C2 )
        Then skin class
    else
        Not skin class
else
    Not skin class
```

With :

	From The Sample Pixel RGB (123,64,23)
Condition C1	
R>95 and G>40 and B>20	**TRUE** 123>95 &64>40 & 23>20
max R,G,B - min R,G,B >15	**TRUE** 123-23>15
R-G >15	**TRUE** - 59 > 15
R>G and R>B	**TRUE** 123>64 & 123>23
Condition C2	
$0.36 \leq \frac{R}{R+G+B} \leq 0.465$	**FALSE** **0.36** <= 0.58 <= **0.465**
And	
$0.28 \leq \frac{G}{R+G+B} \leq 0.363$	**TRUE** **0.28** <= 0.3 <= **0.363**

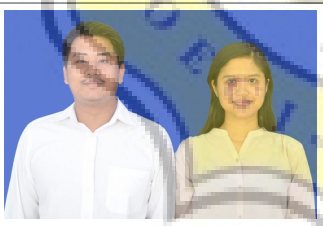
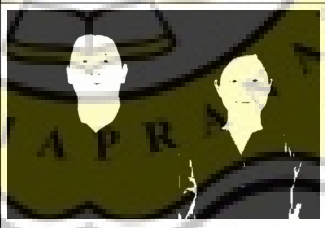
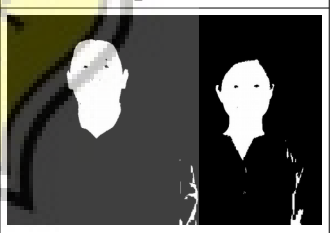
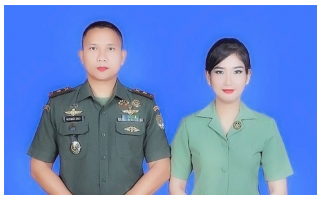
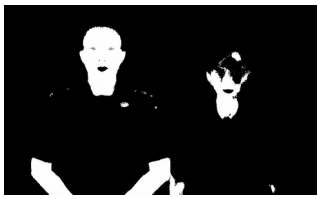
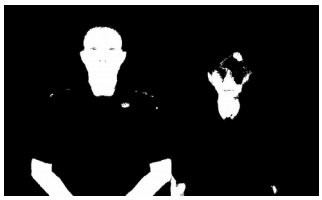
So from the example pixel above, the pixel does not belong to the skin class because in the second condition the value of R / R+G+B = 0.5.

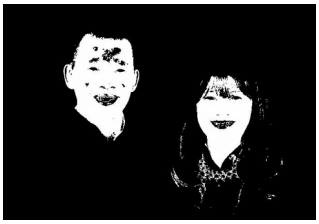
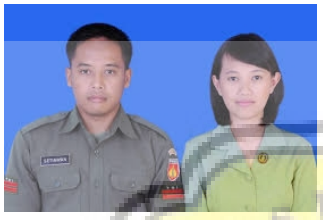
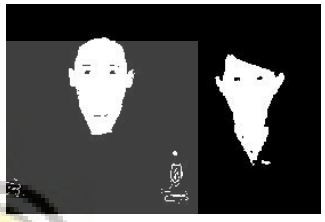
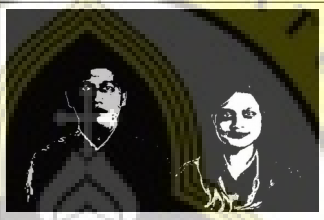
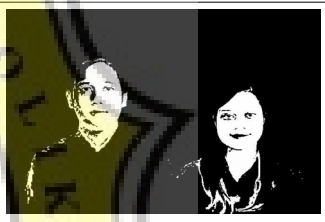
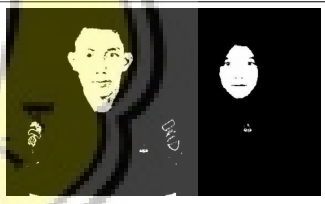
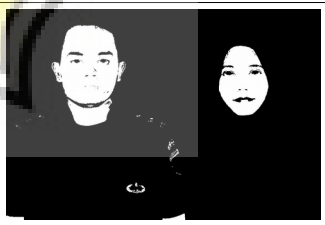
Preliminary study of skin detection consists of 2 compositions of Normalized RGB. This composition is determined to deal with dark-colored faces (table 4.2).

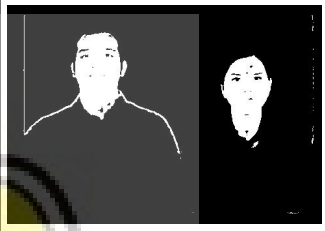
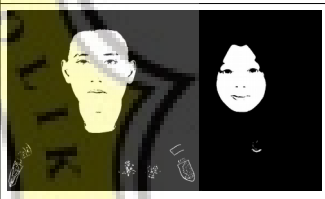
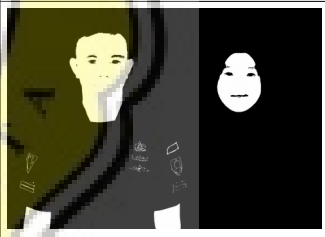
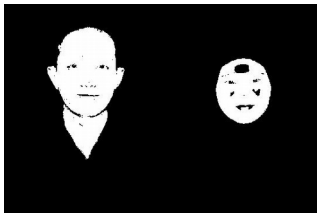
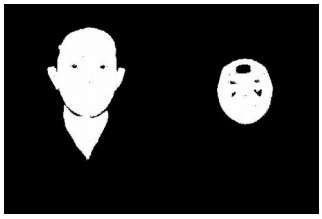
Table 4.2: Table of 2 Skin Detection Composition

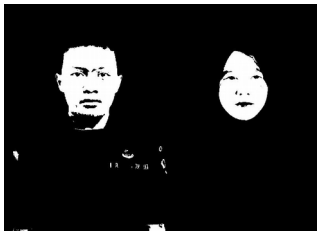
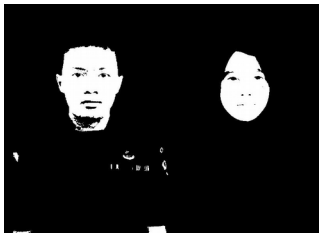
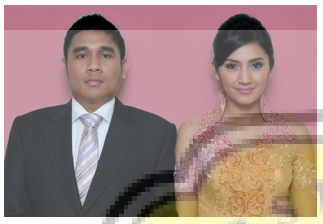
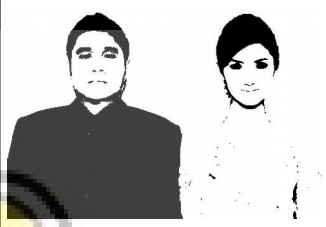
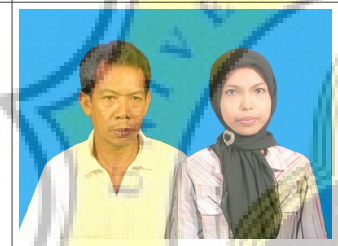
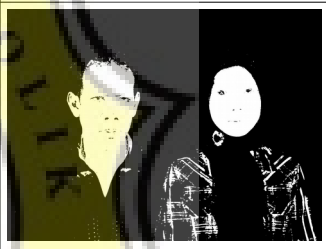
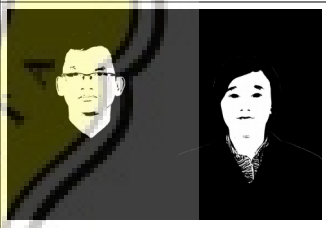
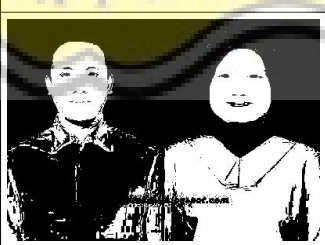
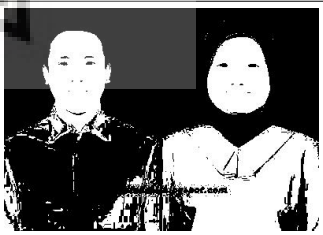
No	Composition 1	Composition 2
1	red>95	red>95
2	green>40	green>40
3	blue>20	blue>20
4	MaxRGB – MinRGB > 15	MaxRGB – MinRGB > 15
5	Abs(red-green) >15	Abs(red-green) >15
6	red > green	red > green
7	red > blue	red > blue
8	0.36 < r < 0.465	0.36 < r < **0.52**
9	0.28 < g < 0.363	**0.2** < g < 0.363

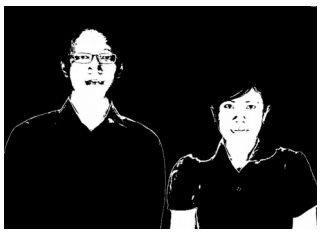
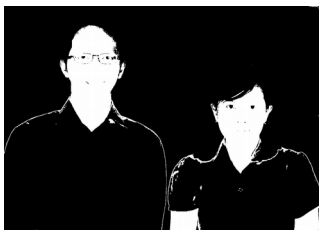
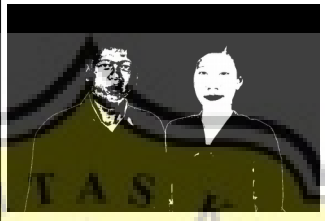
Table 4.3: Testing Skin Detection

No	Testing Image	Result of Composition 1	Result of Composition 2
1			
2			

3			
4			
5			
6			
7			

8			
9			
10			
11			
12			
13			

14			
15			
16			
17			
18			

19			
20			

From table 4.3, faces that have a dark color have been detected using the second composition, but when applied to faces that have a bright color the pattern of the eyes and mouth becomes unclear. Here is the percentage of successful skin detection with the provision of not removing eyes and mouth:

1. Composition 1, with 15 images have a clear eye and mouth pattern, 5 images failed. So the percentage of success is 15 / 20 x 100 = 75 %.
2. Composition 2, with 7 images have a clear eye and mouth pattern, 13 images failed. So the percentage of success is 7 / 20 x 100 = 35 %.

From the results of the analysis, this program will use skin detection method with **composition 1** so that the eyes and mouth can be detected.

2) Adaptive Freeman Chain Code Algorithm and Recognize the Position and Size of Objects

Freeman Chain Code consists of 2 types, the first is the 4-way chain code and the second is the 8-way chain code. In this research 8 chaincode directions were used. 4 and 8 directions are illustrated in the illustration 4.6.

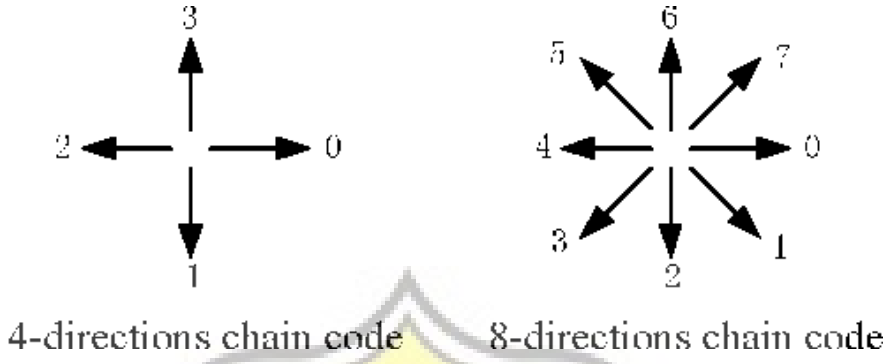


Illustration 4.6: 2 Kind of Freeman Chain Code

In this research Freeman Chain Code is used to determine the length and width of objects, the position of objects and the pattern of objects in the image resulting from edge detection. To determine the object's length, and the object's position with the chain code is explained by the following code:

```
if(cc.xawal>j-1){
    cc.xawal=j-1;
 }                          <-- Determine the first x

if(cc.yawal>i-1){
    cc.yawal=i-1;           <-- Determine the first y
 }


  if(cc.xakhir<j+1){
     cc.xakhir=j+1;         <-- Determine the last x
      }
if(cc.yakhir<i+1){
    cc.yakhir=i+1;          <-- Determine the last y
}
```

After doing the chain code

height = cc.xakhir - cc.xawal; <-- Determine the height

width = cc.yakhir - cc.yawal; <-- Determine the width

To determine the pattern of objects described as follows.

8 direction chaincode result:

0 0 0 0 0 0 0 1 1 1 2 2 2 3 3 3 4 4 4 4 4 5 5 5 5 4 4 4 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 7 6 6 6 7 7 7

Results of **Adaptive Chain Code** are the result of compression from Original Chain Code which has the same character in sequence.

0 1 2 3 4 5 4 5 6 7 6 7

3) Determine the Area of Interest

To determine the area of interest (eyes / mouth), this research have done a preliminary study to remove the noise from the picture.

Table 4.4: Preliminary Study for Area of Interest

No	Eye Size (height x width) (px)	No	Mouth Size (height x width) (px)
1	16 x 11	1	25 x 7
2	21 x 9	2	37 x 20
3	14 x 13	3	22 x 10
4	27 x 16	4	58 x 22
5	51 x 23	5	61 x 30
6	26 x 19	6	33 x 11
7	22 x 11	7	17 x 8
8	23 x 10	8	83 x 28
9	55 x 20	9	41 x 15
10	58 x 21	10	29 x 11

After do some research from several pictures, from table 4.4 its known that the minimum height of eyes and mouth is 14 px and the minimum width of eyes and mouth is 7 px. So constraint is used to eliminate small objects is:

```
1) if (height<=13 || width<=6) {
2)      //Black the area of object
3) }
```

The constraint uses the "OR" command to removes certain objects such as lines and dots.

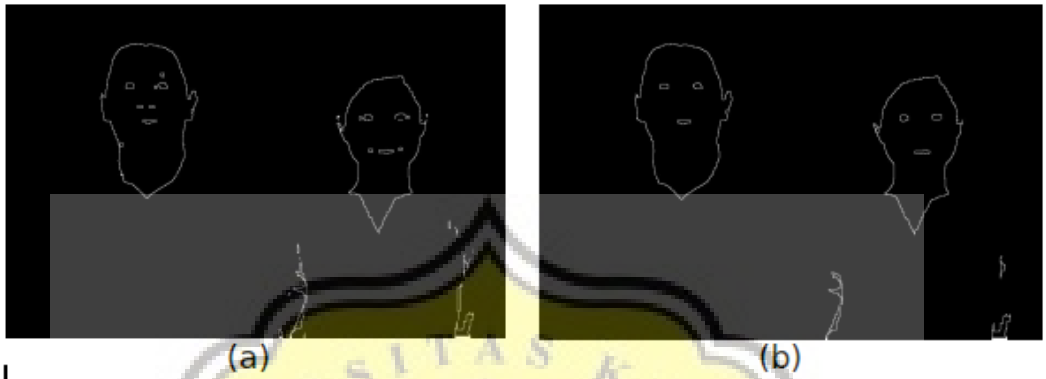


Illustration 4.7: a) Before AOI Process b) After AOI Process

Illustration 4.7, are examples of the results of the determine the area of interest. From the illustration above, it is known that the points that are not part of the research will be removed such as the nostrils, points above the man's eye, and points close to the woman's lips.

4) Determine the Possibility of Face

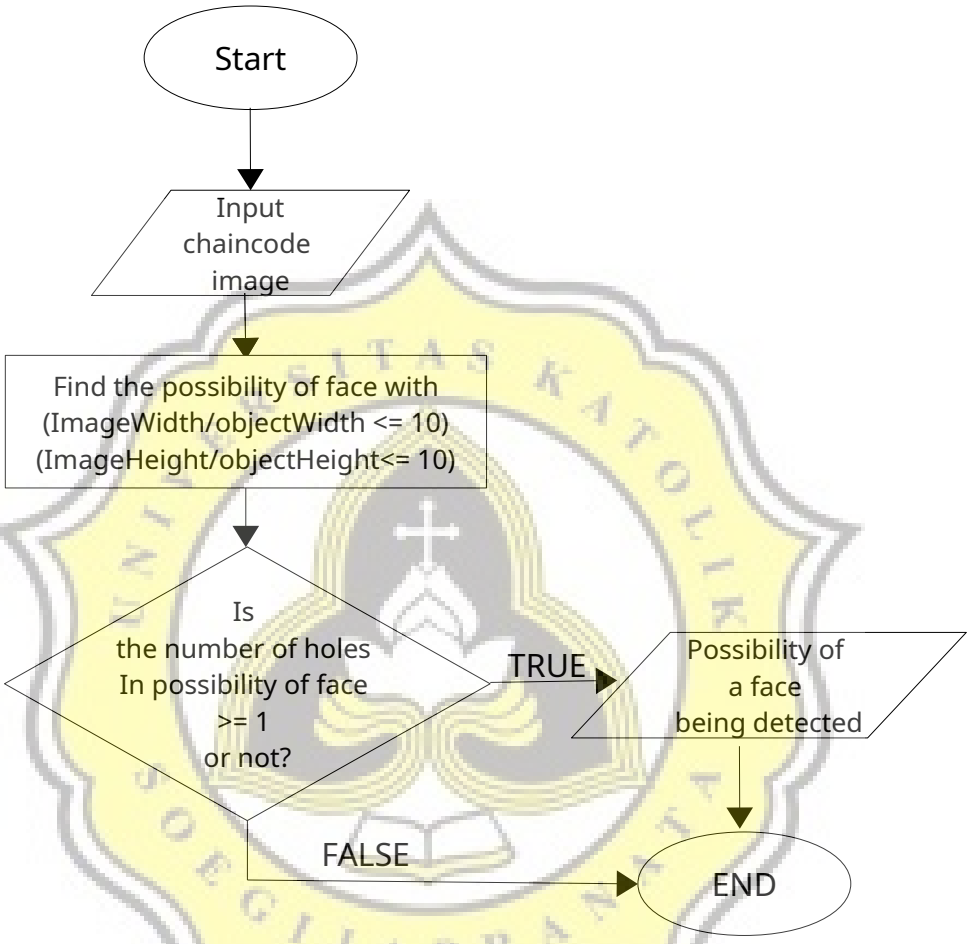


Illustration 4.8: Flowchart for determine the possibilities of face

Illustration 4.8 is a flowchart to determine the face area, with 2 constraints. In the first constraint, if the width of the object is divided by the width of the image <10 and the height of the object divided by the height of the image <10 then the object is a possible face region. The number 10 is obtained from the results of trials on 20 images, that the object which is the possibility of the head on the marriage image has a size that is not too small.

Then the second constraint for detection of possible face region is the number of holes in it, if the number of holes >= 1 then the object is the face area. So, by looking at the above conditions, it is known that 1 object will be compared to all the starting and ending points of the possible object head. If there are no holes, the object is not a face area.

5) Determine the Eyes and Mouth Position

First, if the object's position is above 2/5 of the face width and is not on the vertical center lines of the face, it will be compared with the eye object.

Secondly, if the object's position is in the vertical centerline of the face and is under 2/5 of the width of the face, it will be compared with the mouth object.

Constraints 2/5 was obtained from a trial of 10 images where the proportional location of the eyes was above 2/5 the size of the head

Illustration 4.9 is the visual image of determine the eyes and mouth position.

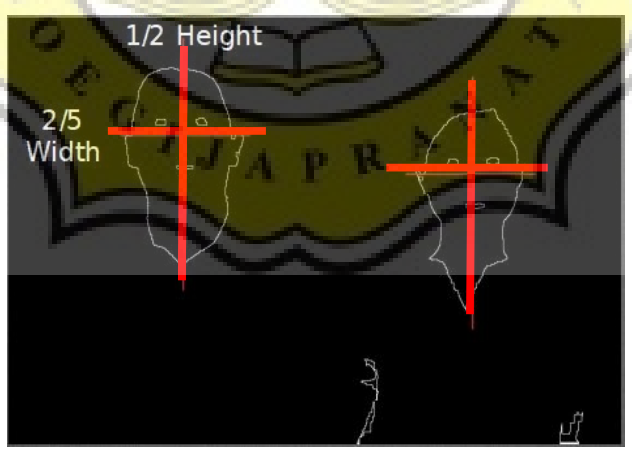


Illustration 4.9: Eyes and Mouth Position

6) Pattern Matching using Consecutive Changing Characters

This research using Changing Consecutive Characters Algorithm algorithms that have been modified by **shifting and rotating patterns**. Example:

Input : 0 1 2 1 3 4 5 6 7 Database : 0 1 3 4 5 6 7

0 1 2 3 4 5 4 6 7

To detect the similarity of patterns used 2 methods:

1. Shifting

From the input pattern, (the length of input pattern - 1) will be shifted

example of the results of shifting :

a. 1 2 1 3 4 5 6 7 0

b. 2 1 3 4 5 6 7 0 1

c. 1 3 4 5 6 7 0 1 2

....

h. 7 0 1 2 1 3 4 5 6

2. Rotating

By adding 1 value for each individual pattern from the input pattern above, rotating data will be obtained in accordance with the direction of the chaincode as follows:

a. First rotated pattern : 1 2 3 2 4 5 6 7 0

b. Then do shifting like step 1.

c. Then rotating again becomes 2 3 4 3 5 6 7 0 1

d. Do shifting like step 1

e. Repeat steps a, b as much 4 times (total rotating is 7 times)

from steps 1 and 2 the pattern will be compared to the pattern in the database with a maximum pattern length of 1.5 from the length of the input pattern. Then the percentage of similarity between each change in input pattern and database pattern is calculated. From the sample data above, the similarity of input data is closer to the database 0 1 3 4 5 6 7