CHAPTER 4
ANALYSIS AND DESIGN

4.1. Analysis

To perform face recognition requires a dataset. The dataset is used to become a data trainer, the dataset can be obtained by taking pictures via cellphone and webcam camera, here the dataset that I use is a photo of the faces of 5 different people, I use 5 photos of each face and each photo has 2 types of high and low resolution so that a total of 10 photos of each face. Here I use the webcam camera to take low resolution quality photos while my cell phone uses to take high resolution photos

4.2. Design

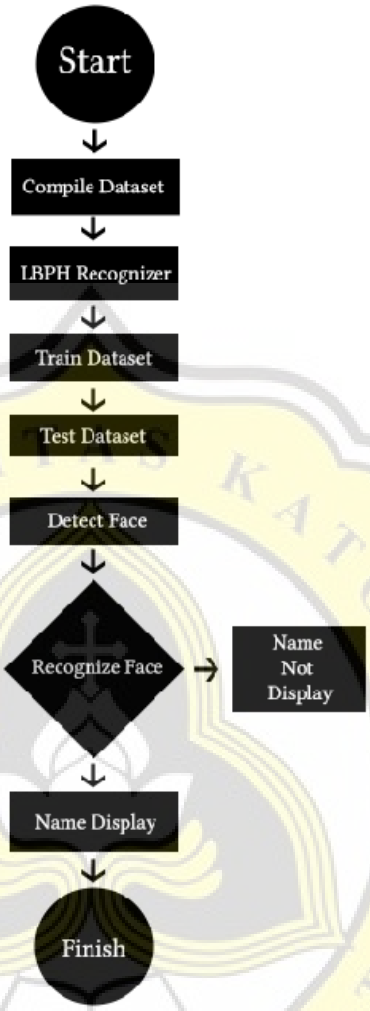

Figure 4.1 Flowchart

First, the dataset obtained will be processed into train data which will be used later. First the dataset is resized and converted to grayscale

Figure 4.2 Grayscale

After becoming grayscale the photo is processed using the haar cascade algorithm to detect the specified gain, the parts that I specify here are the eyes, face and mouth.

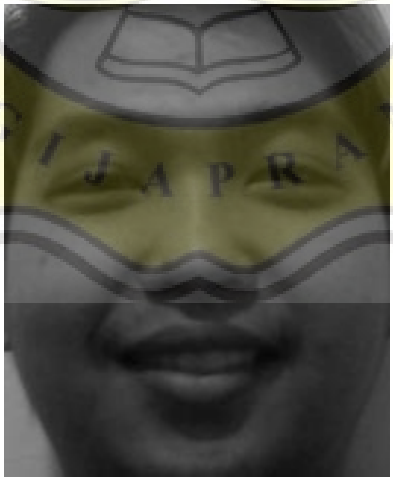

Figure 4.3 Face Crop

Figure 4.4 Eye Crop

Figure 4.5 Mouth Crop

The cropped dataset is then reprocessed using the LBHP face recognition method. The first way this method works is that each part of the cropped dataset will be taken 3x3 pixels from the top left end of the dataset. Furthermore, the dataset will be processed edge detection, for example as shown below:

3	6	7
1	5	5
8	3	2

Figure 4.6 Example 3x3 pixel

We will use the value in the middle as a threshold which is useful for determining the binary for the neighbor value at 3x3 pixels, if the neighbor value exceeds the threshold, the binary is 1, if the neighbor value is lower than the threshold, the binary is 0.

0	1	1
0		1
1	0	0

Figure 4.7 Example Binary 3x3

Then the binaries are arranged starting from the top left end and shifting sequentially in a clockwise direction, this binary will be converted to decimal and the decimal value will be the pixel center value.

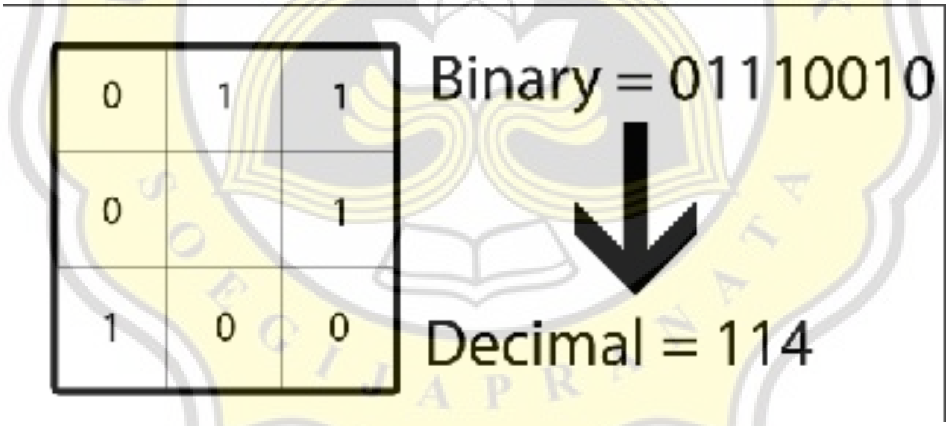

Figure 4.8 Binary to Decimal

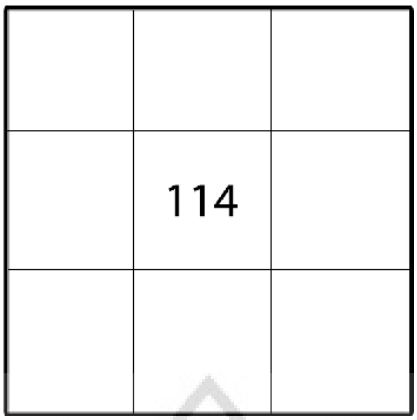

Figure 4.9 Pixel Center Value

After the process is complete, then the image will be divided into several more grids, where each grid contains decimals from the previous 3x3 pixel process. Decimals that are in 1 grid will be concatenate and the final results will sort starting from the left and become a histogram in the image.

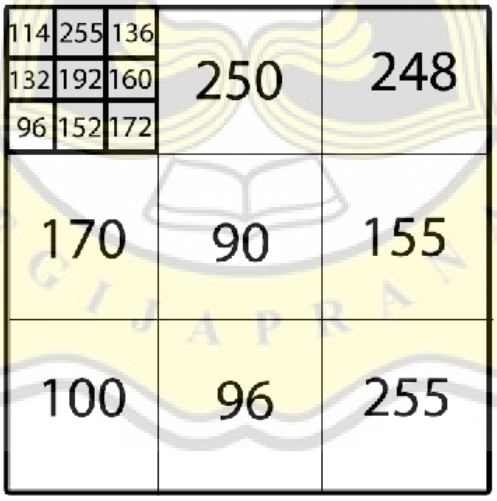

Figure 4.10 Before Concate

156	250	248
170	90	155
100	96	255

Figure 4.11 After Concate

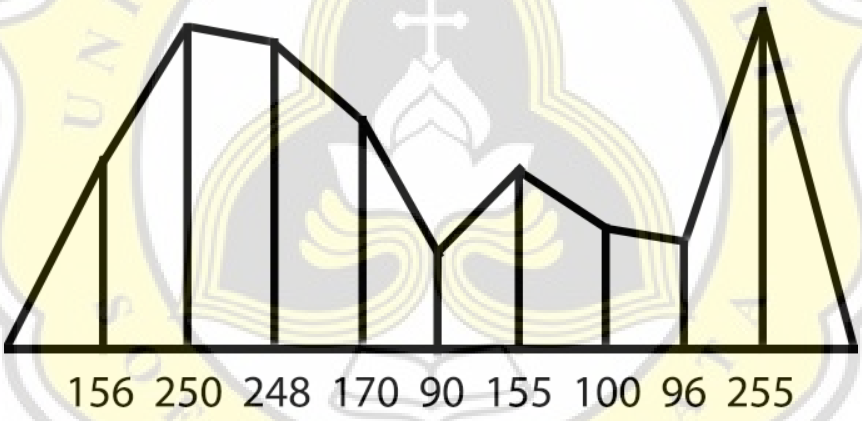

Figure 4.12 Histogram From Every Grid

Furthermore, other datasets will also go through the process described earlier, after going through the process and getting the histogram results, the next process is calculating the histogram distance comparison using the euclidean distance method, calculating the distance comparison between one dataset and another is useful for producing the final histogram that will be used as a benchmark/label for the dataset folder, here is the formula for the euclidean distance:

$$d = \sqrt{[(x2 - x1)^2 + (y2 - y1)^2]}$$

Figure 4.13 Euclidean Distance Formula

In Figure 4.13 d is the symbol of the Euclidean distance, which will be obtained from the results of calculating the roots. x1,y1 are the values of the first histogram and x2,y2 are the values of the second histogram.

d = The Euclidean Distance

x1,y1 = The variable of the first histogram

x2,y2 =The variable of the second histogram