CHAPTER V

IMPLEMENTATION AND TESTING

5.1. Implementation

This project took several stages in order to be a user can know the nominal currencies. This project can know the nominal money by matching histogram with euclidean distance and matching pattern with coefficient correlation.

5.1.1. Preprocessing

To get an image of the money, program must run a preprocessing process to get an image of the money by run the class of crop and resize.

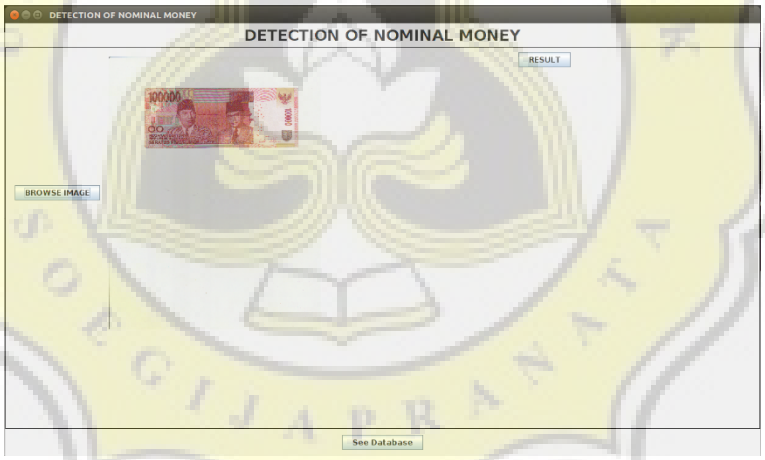


Figure 16. Main Display

```
class btnn3 implements ActionListener
{
        public void actionPerformed(ActionEvent event4)
        {
                cropuang c = new cropuang(gambar);
                resize r = new resize(c.getCrop());
```

Figure 17. Preprocessing Code

5.1.1.1. Crop Money

First, the input of the scan image detects the location of the money, then program will crop the money.

Figure 18. Cropping Money

Program will read the pixel from left, right, top and bottom of the scan image, when the pixel of red, green, blue of the image has s value under 230, program will run and record the pixel, then program will look for modus of each side of the money and make a new buffered image based on subtract the modus of left and right to become a new width of an image and subtract the modus of top and bottom to get a new height.

```
int startwidth = modkiri; //kiri
int endwidth = modkanan; //kanan
int startheight = modatas; //atas
int endheight = modbawah; //bawah

int selisihwidth = Math.abs(modkanan-modkiri);
System.out.println(selisihwidth);
int selisihheight = Math.abs(modbawah-modatas);
System.out.println(selisihheight);

if (selisihwidth<=selisihheight)
{
        int b = (modkanan-modkiri)-1;
        imageoutput = new BufferedImage(selisihheight,selisihwidth,BufferedImage.TYPE_INT_RGB);
        for(int i=modkiri;i<modkanan;i++)// width
        {
                for (int j=modatas;j<modbawah;j++)//height
                {
                  Color a = new Color(image.getRGB(i,j));

                  rednew = a.getRed();
                  greennew = a.getGreen();
                  bluenew = a.getBlue();

                  Color out = new Color(rednew, greennew, bluenew);

                  imageoutput.setRGB((j-modatas),b-(i-modkiri),out.getRGB());
                }
        }//rotate1.setRGB(j,a-i,image.getRGB(i,j));
}
```

Figure 19. Cropping Money Code

5.1.1.2. Resize

After program gets the cropped money, program will resize the money to compare the image of the money.

887*335 400*200

Figure 20. Resizing Money

Resize image is done by scaling which obtained by dividing the new width/height with original width/height of the image. Then the value of scale will be inserted to a new BufferedImage with a new width and height.

```
System.out.println("Panjang : "+width+" Lebar : "+height);
float skalax=(float)widthbaru/width;
System.out.println(skalax);
float skalay=(float)heightbaru/height;
System.out.println(skalay);

imageresize = new BufferedImage(widthbaru,heightbaru,BufferedImage.TYPE_INT_RGB);

for(int a=0;a<heightbaru;a++)
{
        for(int b=0;b<widthbaru;b++)
        {
                int x=(int)(b/skalax);
                int y=(int)(a/skalay);
                Color c = new Color(image.getRGB(x,y));
                red=c.getRed();
                green=c.getGreen();
                blue=c.getBlue();
                imageresize.setRGB(b,a,c.getRGB());
        }
}
```

Figure 21. Resizing Code

5.1.2. Histogram

After preprocessing, program will read the input image, and run the class of matchinghistogram to count the similarity of images and database.

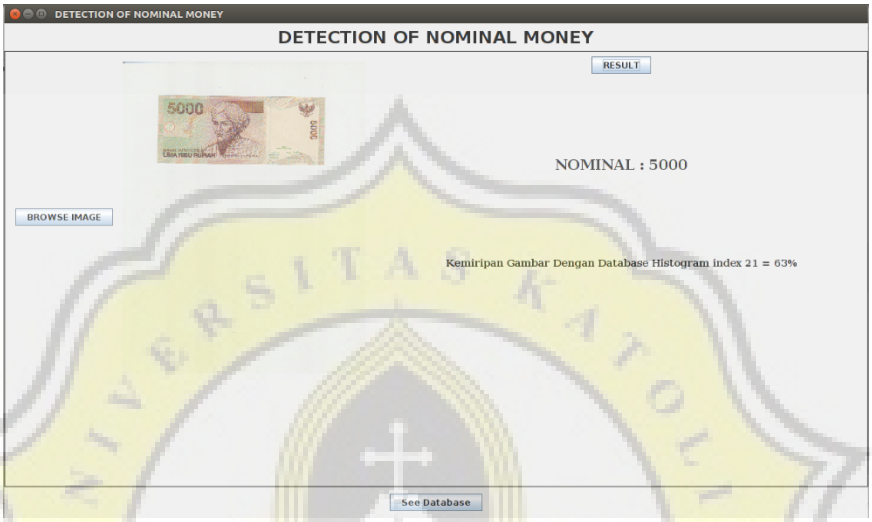


Figure 22. Main Display after getting the Euclidean Distance value

```
matchinghistogram a = new matchinghistogram(r.getHasilsize());
if(a.getHasil()!=0)
{
        if(a.getHasil()==1000)
                {
                        hasil.setText("NOMINAL : "+a.getHasil());
                        tampung.setText("Kemiripan Gambar Dengan Database Histogram index "+a.getNominal()
+a.getTampung()+"%");
                }
        else if(a.getHasil()==5000)
        {
                hasil.setText("NOMINAL : "+a.getHasil());
                tampung.setText("Kemiripan Gambar Dengan Database Histogram index "+a.getNominal()
+a.getTampung()+"%");
        }
```

Figure 23. Histogram Code

5.1.2.1. Matching Histogram

Calculation of the euclidean distance is done so that the program can find out the color values of each pixel of the image. The calculation is done by grouping data based on the value of the pixel. Then group the red, green and blue pixel of image input and image that stored in database in array. After that, subtract the red, green, and blue index with an image in database and sum the red, green, blue index. Then, get the root of sum index to find the value of euclidean distance.

```
int total, total1, total2, total3;
double hasil;
int count1=0;
for(int y=0; y<256 ; y++)
{
        total1=Math.abs(redd[y]-redd1[y]);
        total2=Math.abs(greenn[y]-greenn1[y]);
        total3=Math.abs(bluee[y]-bluee1[y]);
        total=total1+total2+total3;
        hasil =Math.sqrt(total);
        if(hasil<=16)
        {
                count1++;
        }
}
count = (count1 * 100 / 256);

if(count<=49)
{
        System.out.println(count+" %");
}
if(count>=50)
{
        System.out.println(count+" %");
}
        tampungmax[k][l]=count;
```

Figure 24. Euclidean Distance Code

5.1.3. Pattern Recognition

After get the result of nominal money based on histogram, program will run the pattern recognition by cropping pattern on the left side of money, mean, thresholding and sobel and count the similarity of pattern by pattern image and database of a pattern.

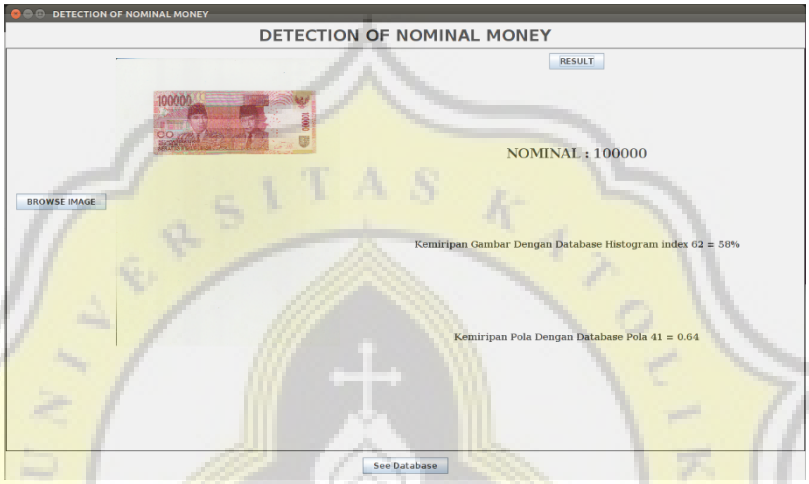


Figure 25. Main Display after getting Coefficient Correlation value

5.1.3.1. Crop

Crop class is used to find the specific pattern of the money image on the left side and crop the part that found by program.

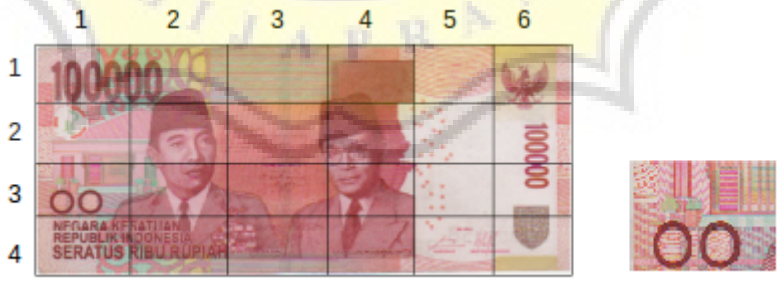


Figure 26. Cropping Pattern

Pattern of the money has found by crop the image by dividing the width and height and get the (3,1) location pixel then crop and insert in a new buffered image.

```
cropbaris=height/4;
cropkolom=width/6;

int z = cropbaris*2;


imagecrop = new BufferedImage(cropkolom,cropbaris,BufferedImage.TYPE_INT_RGB);
for(int a=0;a<cropbaris;a++)
{
        for(int b=0;b<cropkolom;b++)
        {
                Color c = new Color(image.getRGB(b,z));
                imagecrop.setRGB(b,a,c.getRGB());

        }
        z++;
}
```

Figure 27. Cropping Pattern Code

5.1.3.2. Mean Filtering

Mean Filtering is used to eliminate some noise on the image.

Figure 28. Mean Filtering Pattern

Several steps to process the image with mean filtering is taking a 3*3 pixel neighbor and get an average pixel to get the mean filter. Output of the image is quite blurred, because of filtering that smoothing the image.

```
red[0] = (redbaru[j-1][i-1]*kernel[0][0]);
red[1] = (redbaru[j-1][i]*kernel[0][1]);
red[2] = (redbaru[j-1][i+1]*kernel[0][2]);
red[3] = (redbaru[j][i-1]*kernel[1][0]);
red[4] = (redbaru[j][i]*kernel[1][1]);
red[5] = (redbaru[j][i+1]*kernel[1][2]);
red[6] = (redbaru[j+1][i-1]*kernel[2][0]);
red[7] = (redbaru[j+1][i]*kernel[2][1]);
red[8] = (redbaru[j+1][i+1]*kernel[2][2]);
hasilred = (red[0]+red[1]+red[2]+red[3]+red[4]+red[5]+red[6]+red[7]+red[8])/9;

green[0] = (greenbaru[j-1][i-1]*kernel[0][0]);
green[1] = (greenbaru[j-1][i]*kernel[0][1]);
green[2] = (greenbaru[j-1][i+1]*kernel[0][2]);
green[3] = (greenbaru[j][i-1]*kernel[1][0]);
green[4] = (greenbaru[j][i]*kernel[1][1]);
green[5] = (greenbaru[j][i+1]*kernel[1][2]);
green[6] = (greenbaru[j+1][i-1]*kernel[2][0]);
green[7] = (greenbaru[j+1][i]*kernel[2][1]);
green[8] = (greenbaru[j+1][i+1]*kernel[2][2]);
hasilgreen = (green[0]+green[1]+green[2]+green[3]+green[4]+green[5]+green[6]+green[7]+green[8])/9;

blue[0] = (bluebaru[j-1][i-1]*kernel[0][0]);
blue[1] = (bluebaru[j-1][i]*kernel[0][1]);
blue[2] = (bluebaru[j-1][i+1]*kernel[0][2]);
blue[3] = (bluebaru[j][i-1]*kernel[1][0]);
blue[4] = (bluebaru[j][i]*kernel[1][1]);
blue[5] = (bluebaru[j][i+1]*kernel[1][2]);
blue[6] = (bluebaru[j+1][i-1]*kernel[2][0]);
blue[7] = (bluebaru[j+1][i]*kernel[2][1]);
blue[8] = (bluebaru[j+1][i+1]*kernel[2][2]);
hasilblue = (blue[0]+blue[1]+blue[2]+blue[3]+blue[4]+blue[5]+blue[6]+blue[7]+blue[8])/9;
```

Figure 29. Mean Filtering Code

5.1.3.3. Thresholding

Thresholding is used to segment the image background and foreground. Thresholding can make an image become two binary images.

Figure 30. Thresholding Pattern

In this program, using 128 threshold values and make the pixel value under 128 become 0 (black) and pixel above 128 become 255 (white). So, the image just have a two color which is black and white.

```
if(total>0&&total<127)
        total=0;
else
        total=255;
```

Figure 31. Thresholding Code

5.1.3.4. Edge Detection

Edge detection used to find an edge of an image.

Figure 32. Edge Detection Pattern

To get an edge on the image, take a 3*3 pixel neighbor and multiply with 3*3 pixel image. Then, if the result above 255, make the color into 255 and if the result under 0, then multiply the result by -1 so there is no color that are not in range image pixel. After getting the result of a kernel, add the multiply kernel x axis and y axis to get an image result.

```
redx[0] = (redbaru[j-1][i-1] * kernelx[0][0]);
redx[1] = (redbaru[j-1][i] * kernelx[0][1]);
redx[2] = (redbaru[j-1][i+1] * kernelx[0][2]);
redx[3] = (redbaru[j][i-1] * kernelx[1][0]);
redx[4] = (redbaru[j][i] * kernelx[1][1]);
redx[5] = (redbaru[j][i+1] * kernelx[1][2]);
redx[6] = (redbaru[j+1][i-1] * kernelx[2][0]);
redx[7] = (redbaru[j+1][i] * kernelx[2][1]);
redx[8] = (redbaru[j+1][i+1] * kernelx[2][2]);


int totalx = (redx[0]+redx[1]+redx[2]+redx[3]+redx[4]+redx[5]+redx[6]+redx[7]+redx[8]);

redy[0] = (redbaru[j-1][i-1] * kernely[0][0]);
redy[1] = (redbaru[j-1][i] * kernely[0][1]);
redy[2] = (redbaru[j-1][i+1] * kernely[0][2]);
redy[3] = (redbaru[j][i-1] * kernely[1][0]);
redy[4] = (redbaru[j][i] * kernely[1][1]);
redy[5] = (redbaru[j][i+1] * kernely[1][2]);
redy[6] = (redbaru[j+1][i-1] * kernely[2][0]);
redy[7] = (redbaru[j+1][i] * kernely[2][1]);
redy[8] = (redbaru[j+1][i+1] * kernely[2][2]);

int totaly = (redy[0]+redy[1]+redy[2]+redy[3]+redy[4]+redy[5]+redy[6]+redy[7]+redy[8]);

if(totalx < 0 )
{
        totalx = totalx * -1;
}
if(totaly < 0 )
{
        totaly = totaly * -1;
}

baru = totalx + totaly;
if(baru > 255)
{
        baru = 255 ;
}
```

Figure 33. Edge Detection Code

5.1.3.5. Coefficient Correlation

Similarity calculation of a pattern with coefficient correlation is done by subtracting the sum pixel and average pixel of the image. Then multiply the subtract pixel with another image. After get the result of multiply, divide the result with the square root value of multiply sum pixel and average pixel of 2 images.

```
// (x-xbar)*(y-ybar)
atass = BigDecimal.valueOf(kaliatas);
System.out.println("HITUNG ATAS : "+atass);

// akar (x-xbar)2*(y-ybar)2
bawah11 = BigDecimal.valueOf(totalkuadrat1);
System.out.println("BAWAH 1 : "+bawah11);

bawah22 = BigDecimal.valueOf(totalkuadrat2);
System.out.println("BAWAH 2 : "+bawah22);

akarbawah = bawah11.multiply(bawah22);
BigDecimal bawah = new BigDecimal(Math.sqrt(akarbawah.doubleValue()));
System.out.println("HITUNG BAWAH : "+bawah);

hasilkoef = atass.divide(bawah,2,BigDecimal.ROUND_HALF_UP);
koefisien = hasilkoef.floatValue();

System.out.println("KOEFISIEN : "+koefisien);
tampung[k][l]=koefisien;
System.out.println(tampung[k][l]+"\n");
```

Figure 34. Coefficient Correlation Code

5.1.4. Decision

After program run all the process, program will make a decision by euclidean distance of histogram and coefficient correlation of pattern. If the result of euclidean distance is under 50% histogram similarity, program will not recognize the image as money. When the euclidean distance has the result above 50%, program will run the pattern recognition. If the size of pattern does not same as the database pattern, program will not recognize patterns. Pattern will be recognized if the result of coefficient correlation has value above 0.6. If the result of euclidean distance and coefficient correlation is different, then program will just detect the money by euclidean distance result.

5.2. Testing

5.2.1. Database Histogram

Database histogram contains images from all nominal rupiah from 1.000, 2.000, 5.000, 10.000, 20.000, 50.000 and 100.000 that divided into three conditions, condition 1 contain money that is categorized as good or new money, condition 2 contain money that is categorized as average or folded money and condition 3 contain money that is categorized bad or crumpled money based on the color of the money.

Table 1. Database Histogram

NOMINAL	CONDITION 1	CONDITION 2	CONDITION 3
1.000	 00.png	 01.png	 02.png
2.000	 10.png	 11.png	 12.png
5.000	 20.png	 21.png	 22.png
10.000	 30.png	 31.png	 32.png
20.000	 40.png	 41.png	 42.png
50.000	 50.png	 51.png	 52.png

100.000	 60.png	 61.png	 62.png

5.2.2. Database Pattern

Database pattern contains images of pattern from nominal rupiah 2.000, 10.000, 20.000, 50.000, 100.000 divided into three conditions, condition 1 contain pattern from image money from image scan that does not have a noise, condition 2 have a difference in position from condition 1 and have some noise on some pattern. Condition 3 have different in position from condition 1 and 2 and have a different noise from condition 2.

Table 2. Database Pattern

NOMINAL	CONDITION 1	CONDITION 2	CONDITION 3
2.000	 00.png	 01.png	 02.png
10.000	 10.png	 11.png	 12.png
20.000	 20.png	 21.png	 22.png
50.000	 30.png	 31.png	 32.png
100.000	 40.png	 41.png	 42.png

5.2.3. Sample

Testing 1

Scan image of testing1.png has an image with top position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 1.000 and have 60% similarity histogram with database histogram 1.000 in condition 1. Nominal 1.000 do not have any pattern.

Table 3. Testing Image 1

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing1.png	1.000	60 %	 00.png	-	-

Testing 2

Scan image of testing2.png has an image with center position in scanning the image. The result of euclidean distance of testing image is detected as nominal 2.000 and have 73% similarity histogram with database histogram 2.000 in condition 3. The result of coefficient correlation has 0.72 similarity pattern with nominal 2.000 in condition 3.

Table 4. Testing Image 2

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing2.png	2.000	73 %	 12.png	0.72	 02.png

Testing 3

Scan image of testing3.png has an image with top position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 5.000 and have 57% similarity histogram with database histogram 5.000 in condition 2. Nominal 5.000 do not have any pattern.

Table 5. Testing Image 3

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing3.png	5.000	57 %	 21.png	-	-

Testing 4

Scan image of testing4.png has an image with right center position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 10.000 and have 55% similarity histogram with database histogram 10.000 in condition 1. The result of coefficient correlation has 0.76 similarity pattern with nominal 10.000 in condition 2.

Table 6. Testing Image 4

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing4.png	10.000	55 %	 30.png	0.76	 11.png

Testing 5

Scan image of testing5.png has an image with center tilt position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 20.000 and have 73% similarity histogram with database histogram 20.000 in condition 1. The esult of coefficient correlation has 0.62 similarity pattern with nominal 10.000 in condition 2.

Table 7. Testing Image 5

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing5.png	20.000	73%	 40.png	0.62	 11.png

Testing 6

Scan image of testing6.png has an image with center position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 50.000 and have 83% similarity histogram with database histogram 50.000 in condition 1. The result of coefficient correlation has 0.65 similarity pattern with nominal 50.000 in condition 1.

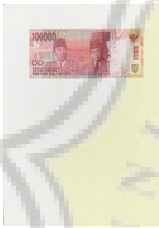
Table 8. Testing Image 6

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing6.png	50.000	83%	 50.png	0.65	 30.png

Testing 7

Scan image of testing7.png has an image with top position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 100.000 and have 50% similarity histogram with database histogram 100.000 in condition 3. The result of coefficient correlation has 0.64 similarity pattern with nominal 100.000 in condition 3.

Table 9. Testing Image 7

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing7.png	100.000	50%	 62.png	0.64	 42.png

Testing 8

Scan image of testing8.png has an image with center position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 1.000 and have 63% similarity histogram with database histogram 1.000 in condition 3. Nominal 1.000 do not have any pattern.

Table 10. Testing Image 8

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing8.png	1.000	63%	 02.png	-	-

Testing 9

Scan image of testing9.png has an image with bottom position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 5.000 and have 58% similarity histogram with database histogram 5.000 in condition 1. Nominal 5.000 do not have any pattern.

Table 11. Testing Image 9

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing9.png	5.000	58%	 20.png	-	-

Testing 10

Scan image of testing10.png has an image with top tilt position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 2.000 and have 70% similarity histogram with database histogram 2.000 in condition 2. The pattern on this image is not detected.

Table 12. Testing Image 10

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing10.png	2.000	70%	 11.png	Pattern Not Detected	

Testing 11

Scan image of testing11.png has an image with center position in vertical scanning the image. The result of euclidean distance of testing image is detected as nominal 5.000 and have 51% similarity histogram with database histogram 5.000 in condition 2. Nominal 5.000 do not have any pattern.

Table 13. Testing Image 11

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing11.png	5.000	51%	 21.png	-	-

Testing 12

Scan image of testing12.png has an image with top position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 10.000 and have 100% similarity histogram with database histogram 10.000 in condition 1. The result of coefficient correlation has 1.0 similarity pattern with nominal 10.000 in condition 1.

Table 14.Testing Image 12

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing12.png	10.000	100%	 30.png	1.0	 10.png

Testing 13

Scan image of testing13.png has an image with bottom position in vertical scanning the image. The result of euclidean distance of testing image is detected as nominal 20.000 and have 82% similarity histogram with database histogram 20.000 in condition 1. The result of coefficient correlation has 0.91 similarity pattern with nominal 20.000 in condition 1.

Table 15. Testing Image 13

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing12.png	20.000	82%	 40.png	0.91	 20.png

Testing 14

Scan image of testing14.png has an image with bottom left position in vertical scanning the image. The result of euclidean distance of testing image is detected as nominal 50.000 and have 66% similarity histogram with database histogram 50.000 in condition 3. The result of coefficient correlation has 0.72 similarity pattern with nominal 50.000 in condition 3.

Table 16. Testing Image 14

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing14.png	50.000	66%	 52.png	0.72	 32.png

Testing 15

Scan image of testing15.png has an image with center tilt position in vertical scanning the image. The result of euclidean distance of testing image is detected as nominal 100.000 and have 66% similarity histogram with database histogram 100.000 in condition 2. The pattern on this image is not detected.

Table 17. Testing Image 15

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing15.png	100.000	66%	 61.png	Pattern not Detected	

Testing 16

Scan image of testing16.png has an image with center position in vertical scanning the image. This image is not detected as a money.

Table 18. Testing Image 16

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing16.png	Money not Detected				

Testing 17

Scan image of testing17.png has an image with center position in vertical scanning the image. The result of euclidean distance of testing image is detected as nominal 2.000 and have 71% similarity histogram with database histogram 2.000 in condition 3. The result of coefficient correlation has 0.81 similarity pattern with nominal 2.000 in condition 1.

Table 19. Testing Image 17

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing17.png	2.000	71%	 12.png	0.81	 00.png

Testing 18

Scan image of testing18.png has an image with center flip position in vertical scanning the image. The result of euclidean distance of testing image is detected as nominal 5.000 and have 58% similarity histogram with database histogram 5.000 in condition 3. Nominal 5.000 do not have any pattern.

Table 20. Testing Image 18

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing18.png	5.000	58%	 21.png	-	-

Testing 19

Scan image of testing19.png has an image with top left position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 100.000 and have 78% similarity histogram with database histogram 100.000 in condition 3. The result of coefficient correlation has 0.64 similarity pattern with nominal 100.000 in condition 3.

Table 21. Testing Image 19

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing19.png	100.000	78%	 62.png	0.64	 42.png

Testing 20

Scan image of testing20.png has an image with bottom position in horizontal scanning the image. The result of euclidean distance of testing image is detected as nominal 1.000 and have 62% similarity histogram with database histogram 1.000 in condition 1. Nominal 1.000 do not have any pattern.

Table 22. Testing Image 20

Scan Image	Nominal	Euclidean Distance	Database Histogram	Coefficient Correlation	Database Pattern
 testing20.png	1.000	62%	 00.png	-	-

Based on testing image above, result of detection nominal money using euclidean distance has 19 images right of 20 images or 95% accurate in matching the color of the image. The inaccurate of the euclidean distance is based on the result of the similarity of histogram is under 50% with image in the database. The result of the coefficient correlation has 9 images right of 12 images or 75% accurate similarity of pattern. The inaccurate of coefficient correlation is based on the location of the crop image. Tilted image will not be able to find a location of the money image because of the cropping pattern that only get a fixed location of the image.