CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

This project use Python programming languange. This project use entropy values for analyze the authenticity of a signature by comparing the entropy value of training and test data, and use precission and recall method for analyze. Size of the signature image samples are 500x500 pixels.

First, set the Folder Path location, and then the program will run from Load the Image from folder.

```
1. FOLDER_PATH = '/home/ivan/Project/data1'
2.
3. Def load_image_from_folder(folder):
4.     images = []
5.     for datattd in os.listdir(folder):
6.         img = os.path.join(folder, datattd)
```

The program will loop the image data in the folder path, then merging the path, folder, and path of each image. If the image path has not reached the last one, the entropy value is calculated and also the time required.

```
7. If img is not None :
8.    t1 = time.time()
9.    print("entropy =" + str(entropy(openImage(img))))
10.   t2 = time.time()
11.   print("Waktu untuk menghitung entropy = %.2f detik)
```

In line 9, there is a function called entropy and openImage. Function entropy is function that contains the entropy value calculation. And the function openImage contains the conversion process from RGB to Binary image.

Below is the code for convert RGB into binary images:

```
12.   def openImage(datattd):
13.       img = Image.open(datattd).convert('RGB')
14.       width, height = img.size
15.
16.   imgs = img.load()
17.   pixel = []
18.   for y in range(height):
19.       for x in range(width):
```

```
20.         R,G,B = imgs[x,y]
21.    grayscale = (0.2989 * R) + (0.5870 * G) + (0.1140 * B)
22.
23.    if grayscale >= 110:
24.       pixel.append(255)
25.    else:
26.       pixel.append(0)
27.
28.    return pixel
```

The openImage function begins by saving the image data converted to RGB, then saving the image width and height data, and save image data that has been loaded. Looping as much as the height and width of the image, if the grayscale value is >= 110 then the pixel array is entered in 255, else, will be entered 0 to get the binary image, for processed the entropy value.

Below is the code for calculating the entropy values :

```
29.    def entropy(pixel):
30.        count = []
31.    for i in range(256):
32.    count.append(0)
33.
34.    for i in range (len(pixel)):
35.        count[pixel[i]]+=1
36.
37.    temp[]
38.    for i in range(256):
39.        if(count[i]>0):
40.           temp.append(count[i]/len(pixel))
41.
42.    total_entropy = 0
43.    for i in range(len(temp)):
44.        temp[i] = ((-temp[i]) * (HitungLog(temp[i])))
45.        total_entropy += temp[i]
46.    return total_entropy
```

In the entropy function, an additional function is needed. There are hitungLog function and ln function. On line 44, hitungLog is called to calculate the temp value, where the content of hitungLog is a formula that calls the ln function inside. Below is the ln and hitungLog function code :

```
47.    def ln(x):
48.     n = 1000.0
49.    return n * ((x ** (1/n)) - 1)
50.
```

```
51.    def hitungLog(x):
52.    return ln(x)/ln(2)
```

5.2 Testing

Below is the graphic of the average entropy value from training data original signatures from 15 respondents. This data use ballpoint pen 0.5. The value range from 0 – 0.27. The average entropy of the original signature sample in the training test produces a relatively high average value in each image, But there is 1 being the lowest in the second signature sample.

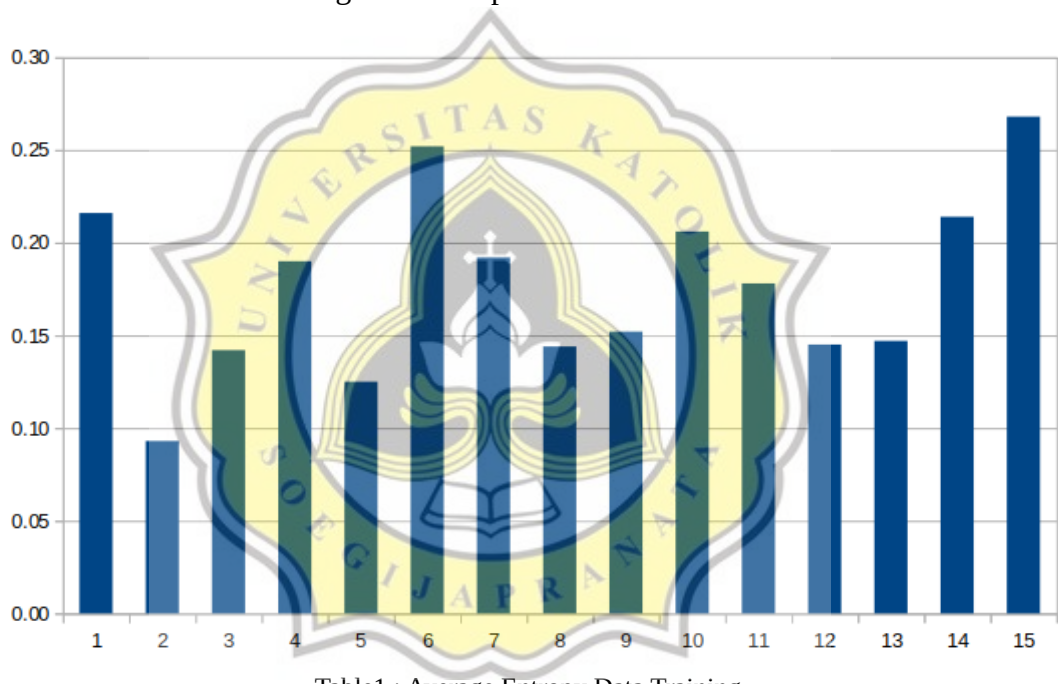

Table1 : Average Entropy Data Training

Below is the graphic that show the average entropy value of original signatures from 15 respondents. This data samples, use ballpoint pens 0.1, 0.3, 0.4, 0.5, and 0.8. The value of the original signatures from 15 respondents range from 0 to 0.27. From the results of the average entropy of the original signature is arguably lower than the training results, it cause using a different thickness of the pen and that affect the etropy results.

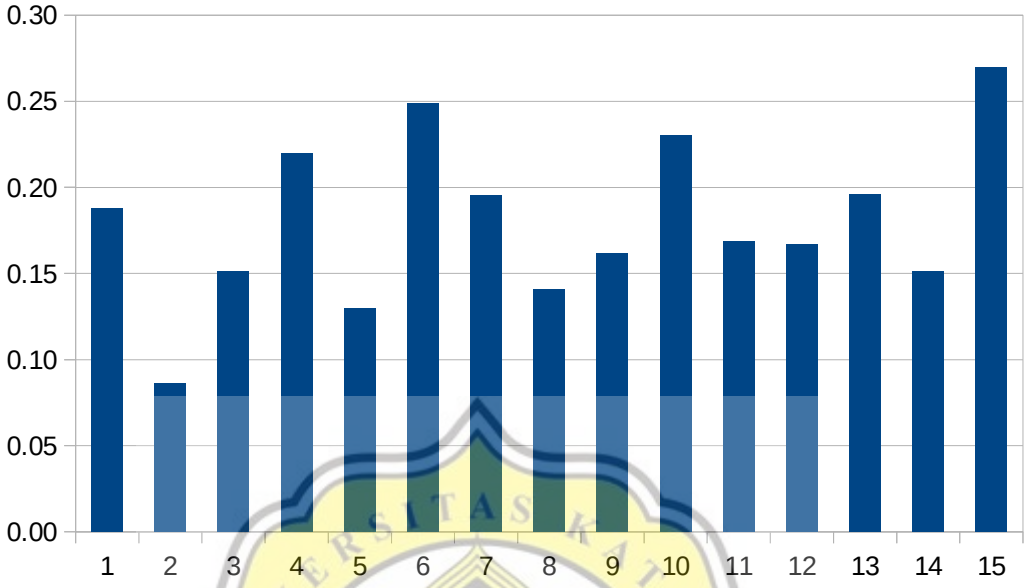

Table 2 : Average Entropy Original Signatures

The average results of the fake signature images from 15 respondent also shown in the following graphic :

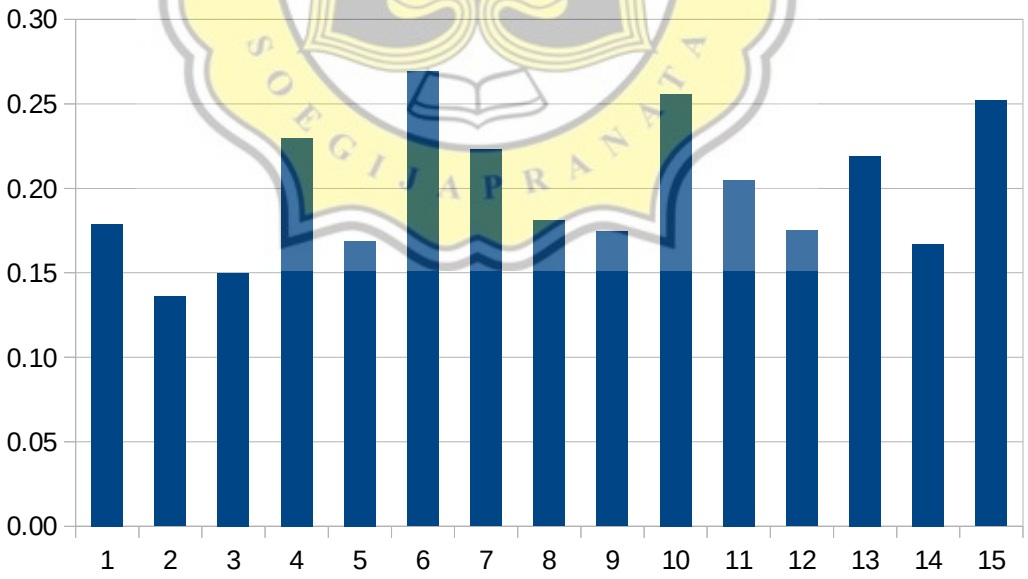

Table 3 : Average Entropy Fake Signatures

Range of averages value of entropy from fake signatures image is from 0 to 0.26. The average entropy value of original and fake signatures are variated. However, almost all respondents showed the average result os the original signature was smaller than the fake signatures. Some have only a little range, but some have a far range. Below is an analysis table for the average emtropy of training data, test data, the fake signature data from 15 respondents.

RESPONDENT	RATA2 TRAINING	RATA2 UJI	RATA2 TTD PALSU
1	0.22	0.19	0.18
2	0.09	0.09	0.14
3	0.14	0.15	0.15
4	0.19	0.22	0.23
5	0.13	0.13	0.17
6	0.25	0.25	0.27
7	0.19	0.20	0.22
8	0.14	0.14	0.18
9	0.15	0.16	0.17
10	0.21	0.23	0.26
11	0.18	0.17	0.21
12	0.15	0.17	0.18
13	0.15	0.20	0.22
14	0.21	0.15	0.17
15	0.27	0.27	0.25

Table 4 : Average Entropy 15 Respondent

From all the data obtained, an analysis is carried out to measure how effective and how accurate this signature verification by using the entropy value calculation method. Analysis is using precision recall calculation and K-Fold Cross Validation.

1. Precission & Recall

Precision is the level of accuracy between the information requested by the user and the answers given by the system. And the Recall is the success rate of the system in recovering information. In the Precision and Recall there are 4 item that will be used, they are True Positive (TP), True Negative (TN), False Positive (FP),

and False Negative (FN). With the context of signature verification using the entropy value, TP (True Positive is the predicted number of test data that are the original signatures and the system result shows the numbers of original signatures too. FP (False Positive) is the numbers that in the test data are predicted fake signatures, but the system results show the original signature numbers. FN (False Negative is the number when the test data value predicted is the original signatures, but the system result shows the number of fake signature results. (TN) True negative is when the the data predicted fake signatures, and the results is true fake signatures.

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Below is the Precision formula :

$$\frac{True\ Positive}{True\ Positive + False\ Positive}$$

And the Recall Formula is :

$$\frac{True\ Positive}{True\ Positive + False\ Negative}$$

The Formula of Accuracy is :

$$\frac{TP + TN}{TP + FP + FN + TN}$$

450 tandatangan asli, 150 tanda tangan palsu
RENTAN 0.03

RESP	TP	TN	FP	FN
1	21	5	9	5
2	27	7	3	3
3	14	3	16	7
4	13	6	17	4
5	30	4	0	6
6	14	5	16	5
7	24	4	6	6
8	24	5	6	5
9	15	2	15	8
10	22	7	8	3
11	16	5	14	5
12	25	2	5	8
13	9	6	21	4
14	11	6	19	4
15	26	4	4	6
TOTAL	291	71	159	79

Table 5: Precission & Recall Results

Below is the table results of True Positive, True Negative, False Positive, and False Negative :

From the table, the result of Precission, Recall and the Accuracy is :

$$\text{Precission} : \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{291}{291+159} = 64\ \%$$

$$\text{Recall} : \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{291}{291+79} = 78\ \%$$

$$\text{Accuracy} : \frac{291+71}{291+71+159+79} = \frac{362}{600} = 60{,}3\ \%$$

And then, The F-Measure or F1 Score Value can be calculated from the precision and recall values. F- Measure or F1 Score is the balance between precision and recall values. The Formula of F-Measure calculation is :

$$\text{F1 Score} = \frac{Precission * Recall}{Precission + Recall}$$

And the result is :

$$\text{F1 Score} : \frac{Precission * Recall}{Precission + Recall} = \frac{64 * 78}{64 + 78} = \frac{4992}{142} = 35{,}1\ \%$$

2. K-Fold Cross Validation

K-Fold Cross Validation is a validation technique for predict the accuracy of a model. The step is to divide the data into training data and testing data. In this analysis, the value of K = 5. There will be 5 folds.

The first experiments uses respondent data 1 until 5, by dividing into 5 subsets and input 75 data with 15 data each respondent :

Subset Uji	ASLI	PALSU	Akurasi
RESP 1	10	5	66.70%
RESP 2	13	2	86.60%
RESP 3	15	0	100%
RESP 4	15	0	100%
RESP 5	15	0	100%

Table 6: Percobaan 1 K-Fold Cross validation

The Second experiments uses respondent data 6 until 10, and dividing into 5 subsets and 75 data with 15 data each respondent :

RESP 6	5	10	33.30%
RESP 7	10	5	66.70%
RESP 8	15	0	100%
RESP 9	12	3	80%
RESP 10	9	6	60%

Table 7: Percobaan 2 K-Fold Cross Validation

The last experiments uses respondent data 11 until 15 :

RESP 11	13	2	86.70%
RESP 12	12	3	80%
RESP 13	15	0	100%
RESP 14	11	4	73%
RESP 15	9	6	60%

Table 8: Percobaan 3 K-Fold Cross Validation

From all the experiments, obtained the highest average accuracy is 90,6 % from experiment 1, and the lowest average accuracy is 68% from experiments 2. And the third experiments yield an average accuracy 79,9%.