

CHAPTER 4

IMPLEMENTATION AND TESTING

4.1 Implementation

This project uses the java programming language and is supported by netbeans. This project uses the LVQ algorithm to determine the smallest value to be classified, including the class of raw / ripe guava fruit. 260x260pixel cut sample size. Image processing of guava fruit to be classified through many stages.

```

1.   ProsesLVQ();
2.   }
3.   catch (FileNotFoundException ex) {
4.
Logger.getLogger(Form.class.getName()).log(Level.SEVERE,
null, ex);
5.   }
6.
7.   finally
8.   {
9.       try {
10.          inputFile.close();
11.          } catch (IOException ex)
12.          {
Logger.getLogger(Form.class.getName()).log(Level.SEVERE,
null,
ex);
13.          }
14.      }
15.  }

```

In the LVQ algorithm, the initial process starts from calculating the weights obtained from the first 2 data. after that it starts counting the class to 0 if the results have been obtained continue to look for the value of class 1.

```

16.          private void ProsesLVQ()
17.      {
18.          // menentukan awal bobot dari 2 data pertama
19.
20.          for (int i = 0; i < a; i++)
21.          {
22.              kls0 = 0.0f;
23.              kls1 = 0.0f;
24.
25.              if(i >= 2)
26.              {
27.                  //
                proses    menghitung kls 0

```

```
28.         for (int x = 0; x < 3; x++)
29.         {
30.             floa
t           temp1, temp2;
temp       =
31.         1   simpanData[i][x];
temp       =
32.         2   simpanData[0][x];
```



```

33.         tampungkls0[x] = temp1 - temp2;
34.         kls0 += (float) Math.pow((temp1
35.         temp2), pangkat);
36.         //System.out.println(kls0);
37.     }
38.     //proses menghitung kls 1
39.     System.out.println("-----
40.     ----
41.     ");
42.     for (int x = 0; x < 3; x++)
43.     {
44.         float temp1, temp2;
45.         temp1 = simpanData[i][x];
46.         temp2 = simpanData[1][x];
47.         tampungkls1[x] = temp1 - temp2;
48.         kls1 += (float) Math.pow((temp1
49.         temp2), pangkat);
50.         //System.out.println(kls1);
51.     }
52.     System.out.println("-----
53.     ----
54.     ");
55.     kls0 = (float) Math.sqrt(kls0);
56.     kls1 = (float) Math.sqrt(kls1);
57.     // proses mencari nilai terkecil
58.     updateBobot(i);
59.     // b = b + 1;
60. }
61. }

```

The two results obtained from the values of class 0 and 1 are looking for the smallest value if it is found then update the weight. can be seen on line 54 after the smallest value is obtained, the weight will be updated so that it belongs to which class

4.2 Testing



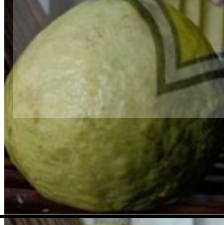

The testing phase was carried out with 100 test data for guava fruit images that were randomly selected by class. In the classification system, the level of ripeness of guava fruit uses extraction RGB and HSV colors are carried out by a training and testing process that aims to determine the level of conformity of the classification process with the training data that has







been defined as the data used for comparison with the image in the testing process. The following is an image of guava fruit along with the HSV value defined as a classification:



that the classification results of the level of fruit maturity guava using RGB and HSV color extraction using the LVQ method provides output with an accuracy level 60%. Where the results of the classification of guava fruit using 10 test data produces 6 images which have the same maturity level as the level the original ripe

Table 4.1 : System test results

No	Guava Fruit Test Data	Real Maturity Level	H	S	V	Maturity Level of Classification Results
1.		Raw	67	46	57	Raw
2.		Ripe	67	51	47	Raw
3.		Ripe	60	51	30	Raw
4.		Raw	60	53	40	Raw

5.		Ripe	40	52	47	Ripe
6.		Ripe	71	57	50	Raw
7.		Ripe	53	47	44	Raw
8.		Raw	60	60	47	Raw
9.		Ripe	54	61	45	Ripe
10.		Raw	59	47	54	Raw