CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

This project requires training data and testing data. Of the two processes, there are similarities in the processes used, namely determining the image to be processed, converting the image to grayscale, changing image from grayscale to binary and calculating the average of R,G,B.

5.1.1 Function determines image to be processes.

```
1. public void setImage(BufferedImage input) {
2.   realImage = input;
3.   setSize();
4.   imageToGray();
5.   grayToBinary();
6.   hitungMeanR_G_B();
7. }
```

On this function, used to call several functions. In the second line used for explain that input is defined as realImage, it will be used for next function. Lines three until four contain the commands to call functions setSize and imageToGray. Then on the fifth line used for call function grayToBinary. On the sixth line used for call function hitungMeanR_G_B.

5.1.2 Function finding original image size.

```
1. private void setSize() {
2.   tinggiCitra = realImage.getHeight();
3.   lebarCitra = realImage.getWidth();
4. }
```

Based on the function above, used for get the original image size. On the second line contains commands to return height dimensions with double precision and stored in high variable images. On the third line contain commands to return width dimensions with double precision and stored in high variable image.

5.1.3 Original image to grayscale conversion function.

```
1. private void imageToGray() {
2.   double red, green, blue;
3.   int gray;
4.   Color before, after;
5.
6.    BufferedImage   output   =   new   BufferedImage(lebarCitra,
   tinggiCitra, BufferedImage.TYPE_BYTE_GRAY);
7.
8.    for (int y = 0; y < tinggiCitra; y++) {
9.      for (int x = 0; x < lebarCitra; x++) {
10.
11.      before = new Color(realImage.getRGB(x, y));
12.
13.      // Calculate RGB to gray
14.      // with lumonisity algorithm
15.      red = (double) (before.getRed() * 0.2989);
16.      green = (double) (before.getGreen() * 0.5870);
17.      blue = (double) (before.getBlue() * 0.1140);
18.      gray = (int) (red + green + blue);
19.
20.      after = new Color(gray, gray, gray);
21.
22.      output.setRGB(x, y, after.getRGB());
23.    }
24.    }
25.    grayImage = output;
26.    }
```

Based on function 5.1.3, used for change the original image. On the second until four line used to declare a variable to be used. Sixth line contains commands to take lebarCitra, tinggiCitra data stored in the output variable. On the eight until twenty-four lines used for processing image where each R,G,B is multiplied by a certain value then the results are stored in the grayImage variable so that it can be used in other function.

5.1.4 Grayscale image to binary conversion function.

```
1. private void grayToBinary() {
2.   Color before, after;
3.   ArrayObjek.clear();
4.
5.    BufferedImage   output   =   new   BufferedImage(lebarCitra,
   tinggiCitra, BufferedImage.TYPE_BYTE_GRAY);
6.
7.    for (int y = 0; y < tinggiCitra; y++) {
8.     for (int x = 0; x < lebarCitra; x++) {
9.       before = new Color(this.grayImage.getRGB(x, y));
10.
11.    if (before.getBlue() < 251) {
12.      after = new Color(255, 255, 255);
13.          ArrayObjek.add(String.valueOf(x)      +      ","      +
   String.valueOf(y));
14.    } else {
15.      after = new Color(0, 0, 0);
16.    }
17.    output.setRGB(x, y, after.getRGB());
18.  }
19.    }
20.    binaryImage = output;
21.    }
```

Pada fungsi ini, digunakan untuk mengubah grayscale menjadi binary. Pada baris 2 digunakan untuk deklarasi variabel, baris 3 berisi perintah untuk mengosongkan ArrayObjek. Kemudian pada baris selanjutnya sama dengan mengubah gambar asli ke grayscale, lalu pada baris ke 11-16 berisi perintah jika before.getBlue() < 251 lalu menambahkan data x dan y kedalam arrayObjek (pada baris 13), dan hasil dari perintah tersebut disimpan di variabel binaryImage.

On this function, used to change grayscale to binary. On the second line used for declare variable, third line contains command for clear the ArrayObjek. Then on the next line is the same as changing the original image to grayscale, then on the 11-16 lines contains command if before.getBlue() < 251 then add data x and y into arrayObjek(on the 13 line), and the result is stored in the binary image variable.

5.1.5 Calculate the mean R,G,B function

```
1. private void hitungMeanR_G_B() {
2.   int length = ArrayObjek.size();
3.   double red = 0;
4.   double green = 0;
5.   double blue = 0;
6.   int x, y;
7.   Color before;
8.   String xy[];
9.
10. for (int i = 0; i < length; i++) {
11.   xy = ArrayObjek.get(i).toString().split(",");
12.
13.   //System.out.println(xy[0] + "-" + xy[1]);
14.   x = Integer.parseInt(xy[0]);
15.   y = Integer.parseInt(xy[1]);
16.
17.   before = new Color(this.realImage.getRGB(x, y));
18.
19.   red += before.getRed();
20.   blue += before.getBlue();
21.   green += before.getGreen();
22. }
23. red /= length;
24. blue /= length;
25. green /= length;
26.
27. meanR = red;
28. meanG = green;
29. meanB = blue;
30.    }
```

Based on function hitungMeanR_G_B used for calculate mean R,G,B on each image. On the second until eight lines used for the variable declaration to be used. On the 19-21 line contains command to find the R,G,B value on each pixel. 23-25 line contains command to find the mean R,G,B from each image, then on the 27-29 line mean value of R,G,B stored on variable meanR, meanG, meanB.

5.1.6 Button Training Data

```
542    // initialization
       String[] fileName = {""};
544    String[] ClassFile = {""};
545
546    if (j <= total_data_train_jn) {
547        fileName[0] = (path + "/segar".concat(Integer.toString(j).concat(".jpg")));
548        imageInput = ImageIO.read(new File(path + "/segar".concat(Integer.toString(j).concat(".jpg"))));
549        ClassFile[0] = "Segar";
550    } else {
551        fileName[0] = (path + "/tidaksegar".concat(Integer.toString(j - total_data_train_jn).concat(".jpg")));
552        imageInput = ImageIO.read(new File(path + "/tidaksegar".concat(Integer.toString(j - total_data_train_jn).concat(".jpg"))));
553        ClassFile[0] = "Tidak Segar";
554    }
```

Illustration 5.1: Button training data

On this training data button, 546-549 lines contains command if the file path is fresh, and using the jpg extension will be classified as Fresh in the class column. And 550-553 contains command if the file path is not fresh and using the jpg extension will be classified as Not Fresh in the class column.

5.1.7 Function Training Data

```
1. public ProsesGambar dataTraining(BufferedImage input, String
   classNameJ) {
2.   ProsesGambar image = new ProsesGambar();
3.
4.   image.setImage(input);
5.   double[] fitur = {image.getR(), image.getG(), image.getB()};
6.   valueFeature = new ArrayDataTiapFitur(fitur[0], fitur[1],
   fitur[2]);
7.    for (int i = 0; i < class_.length; i++) {
8.      if(class_[i].toLowerCase().equals(classNameJ.toLowerCase()))
   {
9.       this.dataset.get(i).addData(valueFeature);
10.    break;
11.   }
12. }
13. return image;
14.    }
```

On this function, used for processing training data. Second line serves to create an ProsesGambar object. Third line is used to input the image into the setImage function in the ProsesGambar class. Then on the fifth-sixth line contains command for inputting the mean R,G,B results into the feature array, and those feature being inserted into valueFeature variable. Then lines 7-13 are used to populate the dataset array according to the valueFeature variable.

5.1.8 Function Testing Data

```
102   public String dataTesting(BufferedImage input) {
103
104       //Take data sample test
105       ArrayDataTimpFitur dataTesting;
106       ProsesGambar imageTesting = new ProsesGambar();
107       imageTesting.setImage(input);
108         System.out.println(input);
109       double[] fitur = {imageTesting.getR(),imageTesting.getG(),imageTesting.getB()};
110       dataTesting = new ArrayDataTimpFitur(fitur[0], fitur[1], fitur[2]);
111       double[] sampel = dataTesting.getMatrix();
112
113       //Create ArrayDataMeanVar Mean & Variance
114       ArrayList<ArrayDataMeanVar> cl = new ArrayList<>();
115
116       System.out.println("Jumlah class : " + class.length);
117       System.out.println("Jumlah class : " + class[0]);
118
119       for(int i = 0; i < class.length; i++) {
120           System.out.println("Class ke : " + i + " " + class[i]);
121           System.out.println("Dataset " + dataset.get(i).toString());
122           cl.add(new ArrayDataMeanVar(class[i], dataset.get(i).getMatrix()));
123       }
124
125       //Create array likelihood
126       ArrayList<ArrayDataLikelihood> lk = new ArrayList<>();
127       for(int j = 0; j < class.length; j++) {
128           lk.add(new ArrayDataLikelihood(class[j], cl.get(j), sampel));
129       }
130
131       //Create array, post prior
132       ArrayList<ArrayDataPosterior> pt = new ArrayList<>();
133       for(int k = 0; k < class.length; k++) {
134           pt.add(new ArrayDataPosterior(class[k], lk.get(k), getProbClass(k)));
135       }
136
137       //Final output class prediction
138       ArrayDataPosterior chosen = pt.get(0);
139       for(int w = 1; w < class.length; w++) {
140           if(chosen.getProbability() < pt.get(w).getProbability()) {
141               chosen = pt.get(w);
142           }
143       }
144
145       this.allPosterior = pt;
146
147
148
149       System.out.println("Class Pred is : " + chosen.getNameClass());
150       return chosen.getNameClass();
151   }
```

Illustration 5.2: Fungsi Data Testing

Based on function above, used to processing testing data. Line 106 creates an ProsesGambar object. Line 107 is used to input the image into the setImage function in the ProsesGambar class. Then lines 109-110 contain commands for inputting the mean R,G,B results into the feature array, and those feature being entered into the valueFeature variable. Then line 114 is used to create an arraylist which will be used to calculate the mean and variance of each class in the training data, then on lines 119-123 is used to add data to the ArrayDataMeanVar arraylist according to the class and data. On lines 126-129 contains commands for adding data to the ArrayDataLikelihood array according to class, mean and variance. Lines 132-135 are used to add data to the ArrayDataPosterior arraylist according

to the class and likelihood calculation results. Then on lines 138-143 contains commands to display the final classification resulst.

5.1.9 Function looks for Mean and Varian

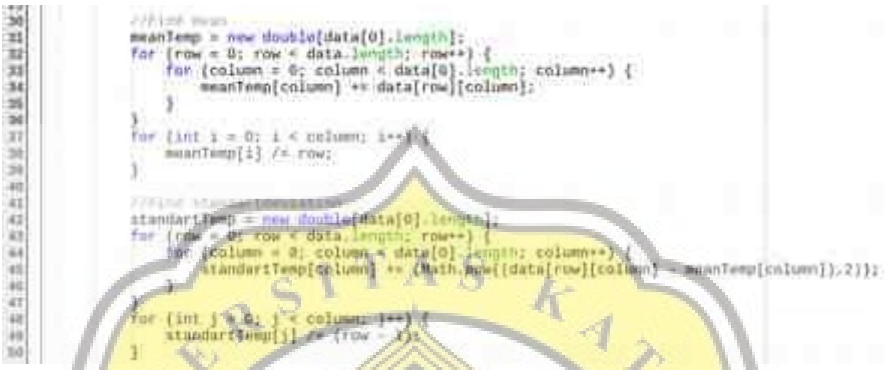


Illustration 5.3: Function mencari mean dan varian

The formula for finding the mean and variance for each class on training data can be seen in illustration 5.3. lines 31-39 contain commands for calculating the mean, and lines 42-50 contain commands for calculating the variance.

5.1.10 Function looks for probability

Illustration 5.4: Fungsi Mencari Probabilitas

Lines 27-28 used for to take the variance value, the mean and store it in the variance and mean variables. On the line 29 take testing data value and store it in value_ variable. Line 31 used for change variance value to standard deviation by using formula $\sqrt{variance}$. Then, on the 34 line is a function to calculate the probability of each class.

5.1.11 Function looks for final probability value.

```
21
22  public ArrayDataPosterior(String class_, ArrayDataLikelihood likelihood, double probClass) {
23      double tempProbPosterior = probClass;
24
25      //Kalikan semua probabilitas fitur untuk mendapatkan posterior
26      for(int i = 0; i < likelihood.getPFeatureClass().length; i++) {
27          tempProbPosterior *= likelihood.getPFeatureClass()[i];
28      }
29      this.nameClass = class_;
30      this.ProbPosterior = tempProbPosterior;
31  //      dataq.add(tempProbPosterior);
32
33      //Untuk ke
34      this.printProbability();
35  }
```

Illustration 5.5: Mencari probabilitas akhir

Illustration 5.5 is a function to find the final probability value.

5.2 Testing

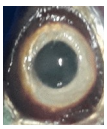
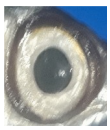
In the testing process, conducted three times with 48 testing data with a composition of eight fresh and non-fresh fish eye image, eight fresh and non-fresh gills, eight fresh and not fresh meat color image. Then used 50,100 and 150 training data which aim to implement the results of the program using Naive Bayes algorithm.

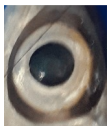
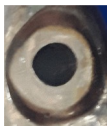
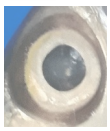
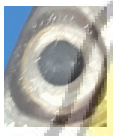
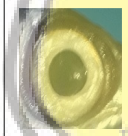
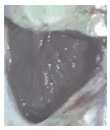
5.2.1 Testing 1

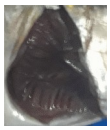
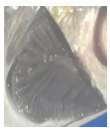
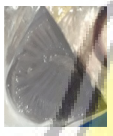
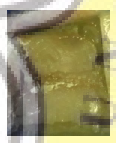
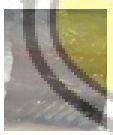
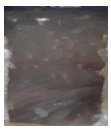
a Testing 1 Parameter

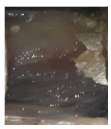
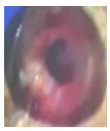
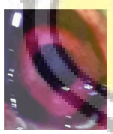
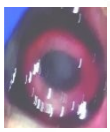
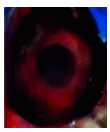
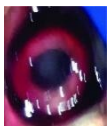
This is the first test table using one parameter:

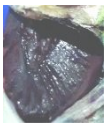
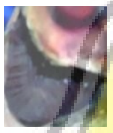
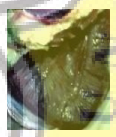
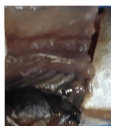
Table 5.1: One parameter testing table

No	Input	Mean			Probability Value		Image	Output
		Red	Green	Blue	F	NF		
1		122,58	117,25	110,33	3.2636770312488506E-6	4.6412780800884197E-7	F	F
2		143,75	148,64	148,64	3.7751350412877664E-8	6.199949949946974E-10	F	F

3		119,24	112,91	106,80	3.5511051574198456E-6	7.24655922899192E-7	F	F
4		118,97	110,42	103,06	3.455499372638653E-6	9.12312466431341E-7	F	F
5		141,38	139,97	134,80	2.602322395777415E-7	7.364181631297155E-9	F	F
6		135,44	138,49	136,70	3.580656039201237E-7	9.763897654556822E-9	F	F
7		126,13	130,00	129,97	1.206688264329287E-6	5.346307744117931E-8	F	F
8		143,98	148,94	148,81	3.556700968566803E-8	5.801082293569649E-10	F	F
9		103,63	101,08	107,03	2.675402586923913E-6	1.381662635405866E-6	F	F
10		93,72	94,88	98,54	1.282518880104012E-6	1.982096880101004E-6	F	NF
11		123,21	113,01	105,26	3.319913871959836E-6	6.675280965383367E-7	F	F
12		103,66	101,08	107,03	2.678757338485086E-6	1.380862249802421 1E-6	F	F

13		112,16	110,63	112,69	3.5023304368694363E-6	7.267337213763175E-7	F	F
14		107,47	103,73	101,15	2.934164544024801E-6	1.517558998522632E-6	F	F
15		117,54	112,36	106,22	3.591943782252404E-6	7.959044760700094E-7	F	F
16		117,29	112,09	105,95	3.592797454952096E-6	8.183241752448024E-7	F	F
17		99,72	93,12	92,15	1.306700945235172 2E-6	2.234225801331316E-6	F	NF
18		135,76	131,23	124,05	1.017289753870617 5E-6	5.002571591413183E-8	F	F
19		111,70	106,76	101,10	3.264471015076350 5E-6	1.304844071338868 7E-6	F	F
20		96,92	91,24	86,35	8.136332223868095E-7	2.288826225751552 6E-6	F	NF
21		95,49	90,21	85,53	6.901274739997 2E-7	2.279126074628589 3E-6	F	NF
22		96,16	89,78	87,91	7.913374012608586E-7	2.303056100082046E-6	F	NF

23		97,62	91,19	88,95	9.546726175676622E-7	2.2981816750289757E-6	F	NF
24		91,34	82,43	74,96	1.7003103118890426E-7	1.8021984410624583E-6	F	NF
25		98,97	55,25	65,93	7.754987823351549E-9	5.560284169782326E-7	NF	NF
26		107,20	104,52	94,43	2.4898975051968583E-6	1.7084885963520355E-6	NF	F
27		93,21	89,90	91,53	8.002142064077176E-7	2.2459061700143747E-6	NF	NF
28		99,96	94,28	100,70	1.7696882316964376E-6	1.9114547737663463E-6	NF	NF
29		76,47	21,67	63,69	4.372404012525958E-12	2.3674622674658517E-8	NF	NF
30		78,55	37,52	75,52	3.1991243863550855E-10	1.8014615402589325E-7	NF	NF
31		36,19	10,28	22,03	1.0978524693802138E-18	1.0430807228002911E-11	NF	NF
32		78,08	36,89	75	2.6342254122225427E-10	1.6631045218433931E-7	NF	NF

33		89,44	79,12	90,36	3.0914790559609796E-7	2.0251117910749455E-6	NF	NF
34		74,54	72,44	75,98	2.0682481908348465E-8	1.0703077348166412E-6	NF	NF
35		69,67	71,69	80,79	1.3567607106164755E-6	8.152743800549914E-8	NF	F
36		94,83	81,42	92,88	5.5794626606651963E-7	2.1479542674374034E-6	NF	NF
37		77,61	82,15	90,10	2.9618604010874055E-6	1.5025226352019168E-6	NF	F
38		99,72	99,33	105,34	2.2064572729643677E-6	1.5528756423789271E-6	NF	F
39		83,29	81,11	85,43	1.7648241412298587E-7	1.8708191710138786E-6	NF	NF
40		57,46	61,60	61,21	1.8902192097133 21E-10	1.8047411289255654E-7	NF	NF
41		110,56	101,64	100,35	2.9198319666041887E-6	1.5440913136384 65E-6	NF	F
42		95,48	82,30	76,03	2.3198469199225332E-7	1.90234388533413E-6	NF	NF

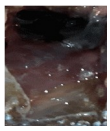
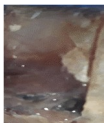
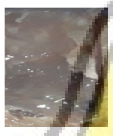
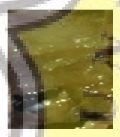
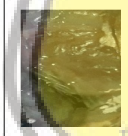
43		68,81	57,92	56,49	3.2061036434519237E-10	2.02249741483476E-7	NF	NF
44		116,95	103,47	97,47	2.9208028953502673E-6	1.364230699695198E-6	NF	F
45		95,49	90,21	85,53	4.889060328226918E-8	1.215989412877089E-6	NF	NF
46		91,63	74,57	67,48	1.8725436702903774E-8	8.955407707576777E-7	NF	NF
47		71,25	61,06	55,18	5.251729418035252E-10	2.367209217434950 8E-7	NF	NF
48		90,95	82,46	77,79	2.0675184328986435E-7	1.926518371459945E-6	NF	NF

Table Information :

F = Fresh

NF = Not Fresh

Based on table 5.1 the test result using 50 training data on one parameter showed that 35 data were successfully classified correctly and 13 data were incorrect. And calculated using the precision-recall formula to get an accuracy value of 72,9% with the following calculation:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

Illustration 5.6: Presicion-recall function

$$Accuracy = \frac{(17+18)}{(17+7+6+18)} = \frac{35}{48} = 72{,}91\,\%$$

Information :

True Positive (TP) = The prediction of fresh results is indeed fresh

True Negative (NF) = The predictions are not fresh are really not fresh

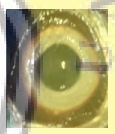
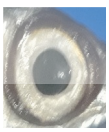
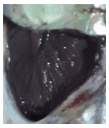
False Positive (FP) = Fresh predictions are not fresh

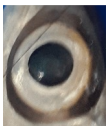
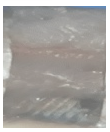
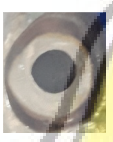
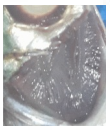
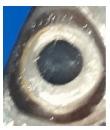
False Negative (FN) = Not Fresh predictions turn out to be fresh

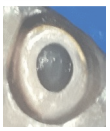
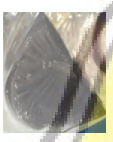
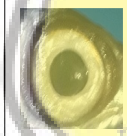
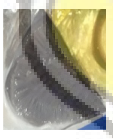
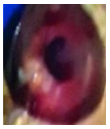
b Testing 3 Parameter

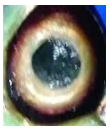
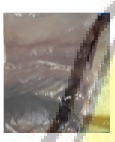
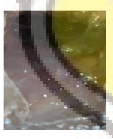
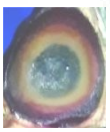
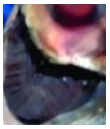
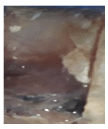
This is the first test table using three parameter:

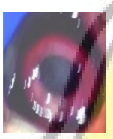
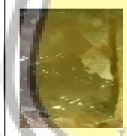
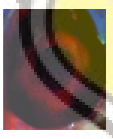
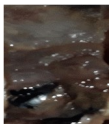
Table 5.2: Three parameter testing table

No	Input	Probability Value		Probability Value		Image	Output
		F	NF	F	NF		
1		3.2636770312488506E-6	4.6412780800884197E-7	2.415260187802645E-6	1.3600054149153413E-6	F	F
		2.675402586923913E-6	1.381662635405866E-6				
		1.3067009452351722E-6	2.2342258013313 16E-6				
2		3.7751350412877664E-8	6.199949949946974E-10	7.791866614625023E-7	6.775808636700435E-7	F	F
		1.282518880104012E-6	1.982096880101004E-6				
		1.0172897538706175E-6	5.002571591413183E-8				

3		3.5511051574198456E-6	7.24655922899192E-7	3.3784966814851932E-6	8.9900936359213 24E-7	F	F
		3.3199138719593836E-6	6.67528096538336 7E-7				
		3.2644710150763505E-6	1.3048440713388687E-6				
4		3.455499372638653E-6	9.12312466431341E-7	2.3159633311170183E-6	1.5273336473284382E-6	F	F
		2.678757338485086E-6	1.3808622498024211E-6				
		8.136332223868095E-7	2.2888262257515526E-6				
5		2.602322395777415E-7	7.364181631297155E-9	1.4842300501489663E-6	1.0044079925454012E-6	F	F
		3.5023304368694363E-6	7.267337213763175E-7				
		6.9012747399972E-7	2.2791260746285893E-6				
6		3.5806560392011237E-7	9.763897654556822E-9	1.3611891830685909E-6	1.2767929987530782E-6	F	F

		2.934164544024801E-6	1.517558998522632E-6				
		7.913374012608586E-7	2.303056100082046E-6				
7		1.206688266432928 7E-6	5.346307744117931E-8	1.917768222084331 7E-6	1.049183076180055E-6	F	F
		3.591943782252404E-6	7.959044760700094E-7				
		9.546726175676622E-7	2.298181675028975 7E-6				
8		3.556700968566803E-8	5.801082293569649E-10	1.266131831942194E-6	8.737009081788726E-7	F	F
		3.592797454952009 6E-6	8.183241752448024E-7				
		1.7003103118890426E-7	1.802198441062458 3E-6				
9		7.754987823351549E-9	5.560284169782326E-7	1.078911620007879 5E-6	1.375077173897214 4E-6	NF	NF
		3.091479059609796E-7	2.025111791074945 5E-6				

		2.9198319666041887E-6	1.544091313638465E-6				
10		2.4898975051968583E-6	1.7084885963520355E-6	9.1418822636582E-7	1.5603800721676022E-6	NF	NF
		2.0682481908348465E-8	1.0703077348166412E-6				
		2.3198469199225332E-7	1.90234388533413E-6				
11		8.002142064077176E-7	2.2459061700143747E-6	7.1909850912951 27E-7	8.4322778316778 33E-7	NF	NF
		1.3567607106164755E-6	8.152743800549914E-8				
		3.2061036434519237E-10	2.02249741483476E-7				
12		1.7696882316964376E-6	1.9114547737663463E-6	1.7494791310373003E-6	1.8078799136329828E-6	NF	F
		5.5794626660651963E-7	2.1479542674374034E-6				
		2.9208028953502673E-6	1.36423069695198E-6				

13		4.372404012525958E-12	2.3674622674658517E-8	1.003585125591229E-6	9.140622235845549E-7	NF	NF
		2.9618604010874055E-6	1.5025226352019168E-6				
		4.889060328226918E-8	1.215989412877089E-6				
14		3.199124386355085 5E-10	1.8014615402589325E-7	7.418342073686358E-7	8.761875223874993E-7	NF	NF
		2.2064572729643677E-6	1.5528756423789271E-6				
		1.8725436702903774E-8	8.955407707576777E-7				
15		1.0978524693802138E-18	1.0430807228002911E-11	5.900252902196242E-8	7.025168411882005E-7	NF	NF
		1.7648241412298587E-7	1.8708191710138786E-6				
		5.251729418035252E-10	2.3672092174349508E-7				
16		2.6342254122225427E-10	1.6631045218433931E-7	6.906809591735265E-8	7.577676455122804E-7	NF	NF

		1.890219209713321E-10	1.8047411289255654E-7				
		2.0675184328986435E-7	1.9265183714599 45E-6				

Table Information :

F = Fresh

NF = Not Fresh

Based on the test results using 50 data training with three parameters (Table 5.2) shows the results that 15 data successfully classified correctly and one data is wrong. And calculated using the precision-recall formula to get an accuracy value of 93,75% with the following calculation:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

$$Accuracy = \frac{(8+7)}{(8+0+1+7)} = \frac{15}{16} = 93{,}75\%$$

Information :

True Positive (TP) = The prediction of fresh results is indeed fresh

True Negative (NF) = The predictions are not fresh are really not fresh

False Positive (FP) = Fresh predictions are not fresh

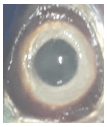
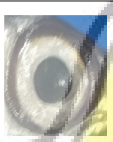
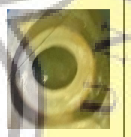
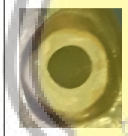
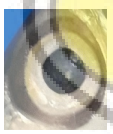
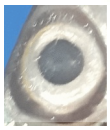
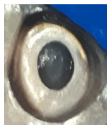
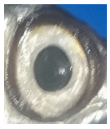
False Negative (FN) = Not Fresh predictions turn out to be fresh

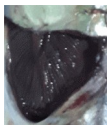
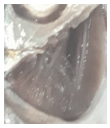
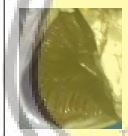
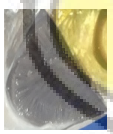
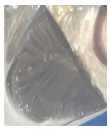
5.2.2 **Testing 2**

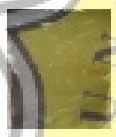
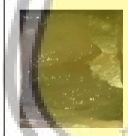
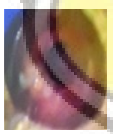
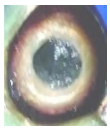
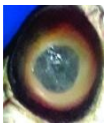
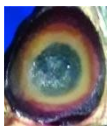
a. Testing 1 Parameter

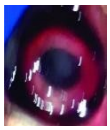
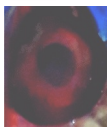
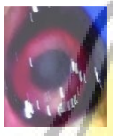
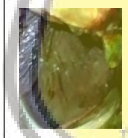
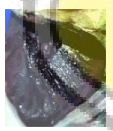
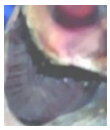
This is the second test table using one parameter:

Table 5.3: One parameter testing table

No	Input	Mean			Nilai Probabilitas		Image	Output
		Red	Green	Blue	F	NF		
1		122,58	117,25	110,33	3.379335940430306E-6	2.274897071668028E-7	F	F
2		143,75	148,64	148,64	4.286277093641 02E-8	1.213056585605 8015E-10	F	F
3		119,24	112,91	106,80	3.669450778263 3893E-6	3.929038574897 6725E-7	F	F
4		118,97	110,42	103,06	3.560551353281302E-6	5.234135462221825E-7	F	F
5		141,38	139,97	134,80	2.863348919661883E-7	1.901618691262 3885E-9	F	F
6		135,44	138,49	136,70	3.886084986662 3116E-7	2.647732762456 3457E-9	F	F
7		126,13	130,00	129,97	1.279035357225818E-6	1.853120249509 5344E-8	F	F
8		143,98	148,94	148,81	4.043477182445632E-8	1.126174004778 7663E-10	F	F

9		103,63	101,08	107,03	2.855008082946851E-6	9.515277261840812E-7	F	F
10		93,72	94,88	98,54	1.42141496286225E-6	1.6636215868516995E-6	F	NF
11		123,21	113,01	105,26	3.420677707838823E-6	3.5426319201389736E-7	F	F
12		103,66	101,08	107,03	2.8582255106701903E-6	9.506265179736463E-7	F	F
13		112,16	110,63	112,69	3.669788481888311E-6	4.069441943219375 7E-7	F	F
14		107,47	103,73	101,15	3.081731885729386E-6	1.034362844749662 4E-6	F	F
15		117,54	112,36	106,22	3.715696895855482 2E-6	4.419741051096645E-7	F	F
16		117,29	112,09	105,95	3.716707476671750 4E-6	4.576202972351579E-7	F	F
17		99,72	93,12	92,15	1.4035337131530735E-6	1.9017118453121962E-6	F	NF
18		135,76	131,23	124,05	1.085219180402322E-6	1.6693454411368752E-8	F	F

19		111,70	106,76	101,10	3.3939290031435476E-6	8.353261869651393E-7	F	F
20		96,92	91,24	86,35	8.827462287769126E-7	2.1011521758180095E-6	F	NF
21		95,49	90,21	85,53	7.537677128132886E-7	2.1453410722778695E-6	F	NF
22		96,16	89,78	87,91	8.623805111179671E-7	2.132700233658448E-6	F	NF
23		97,62	91,19	88,95	1.033571721724134E-6	2.065129381263857 6E-6	F	NF
24		91,34	82,43	74,96	1.895385463113386E-7	1.985459368332244 3E-6	F	NF
25		98,97	55,25	65,93	8.437379462659222E-9	7.324356899048205E-7	NF	NF
26		107,20	104,52	94,43	2.607359191693568 5E-6	1.21529120139388E-6	NF	F
27		93,21	89,90	91,53	8.865287281 32773E-7	2.07545382754718E-6	NF	NF
28		99,96	94,28	100,70	1.907595531238902E-6	1.513289008760915E-6	NF	NF

29		76,47	21,67	63,69	5.679174978513428E-12	4.394517946900631E-8	NF	NF
30		78,55	37,52	75,52	4.013042218977528E-10	2.906884569971146E-7	NF	NF
31		36,19	10,28	22,03	2.342464590146485E-18	2.975089242071751E-11	NF	NF
32		78,08	36,89	75	3.318437908547625 3E-10	2.708523445447614E-7	NF	NF
33		89,44	79,12	90,36	3.506530782496558 4E-7	2.120626309582545 8E-6	NF	NF
34		74,54	72,44	75,98	2.531078324333787E-7	2.021241590906995 4E-6	NF	NF
35		69,67	71,69	80,79	2.097655413332601 7E-8	1.311631548680059 6E-6	NF	F
36		94,83	81,42	92,88	6.147102822968982E-7	2.084710061640110 5E-6	NF	NF
37		77,61	82,15	90,10	1.763902137319579 5E-7	1.841262386393643 5E-6	NF	NF
38		99,72	99,33	105,34	2.387241730623541E-6	1.135467262833257E-6	NF	F

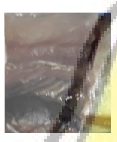
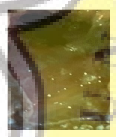
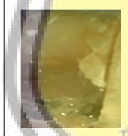
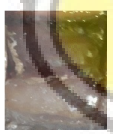
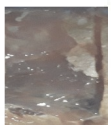
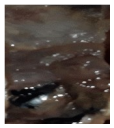
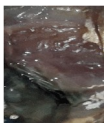
39		83,29	81,11	85,43	2.0772857162241603E-7	2.0838191856831934E-6	NF	NF
40		57,46	61,60	61,21	2.837666586024975E-10	3.1627796939754985E-7	NF	NF
41		110,56	101,64	100,35	3.0405681741232383E-6	1.0538956169352673E-6	NF	F
42		95,48	82,30	76,03	2.531078324333787E-7	2.0212415909069954E-6	NF	NF
43		68,81	57,92	56,49	4.2849779384694566E-10	3.4134863776215 66E-7	NF	NF
44		116,95	103,47	97,47	3.0029346603406796E-6	8.8781281383886 72E-7	NF	F
45		95,49	90,21	85,53	5.4502427703103 48E-8	1.4753169409846 08E-6	NF	NF
46		91,63	74,57	67,48	2.1617417788073328E-8	1.1714331313181722E-6	NF	NF
47		71,25	61,06	55,18	6.846682831222191E-10	3.8668451107777994E-7	NF	NF
48		90,95	82,46	77,79	2.3114154297255555E-7	2.0957575992256605E-6	NF	NF

Table Information :

F = Fresh

NF = Not Fresh

In the test table above, using 100 training data on one parameter the results are 36 data classified correctly and 12 data incorrectly. Therefore calculated using the presicion-recall formula to get an accuracy value of 75% with the following calculation:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

$$Accuracy = \frac{(17+19)}{(17+7+5+19)} = \frac{36}{48} = 75\%$$

Information :

True Positive (TP) = The prediction of fresh results is indeed fresh

True Negative (NF) = The predictions are not fresh are really not fresh

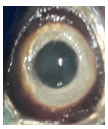
False Positive (FP) = Fresh predictions are not fresh

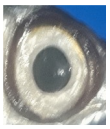
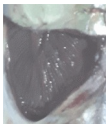
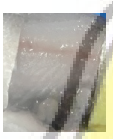
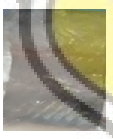
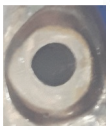
False Negative (FN) = Not Fresh predictions turn out to be fresh

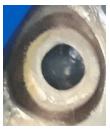
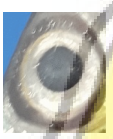
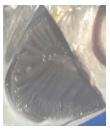
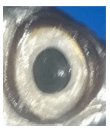
b. Testing 3 Parameter

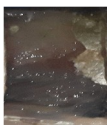
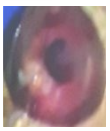
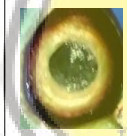
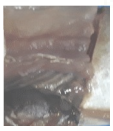
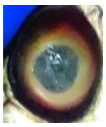
This is the second test table using one parameter:

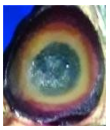
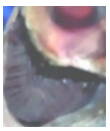
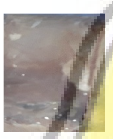
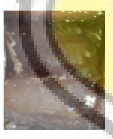
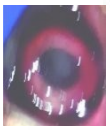
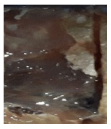
Table 5.4: Three parameter testing table

No	Input	Probability Value		Probability Value		Image	Output
		F	NF	F	NF		
1		3.379335940430306E-6	2.274897071668028E-7	2.545959245510077E-6	1.026909759554360 2E-6	F	F
		2.855008082946851E-6	9.515277261840812E-7				

		1.4035337131530735E-6	1.9017118453121962E-6				
2		4.28627709364102E-8	1.2130565856058015E-10	1.253317071632286E-6	8.401575206315341E-7	F	F
		1.42141496286225E-6	1.6636215868516995E-6				
		1.085219180402322E-6	1.6693454411368752E-8				
3		3.6694507782633893E-6	3.9290385748976725E-7	3.494685829748586E-6	5.274977454896014E-7	F	F
		3.420677707838823E-6	3.5426319201389736E-7				
		3.3939290031435476E-6	8.353261869651393E-7				
4		3.560551353281302E-6	5.234135462221825E-7	2.433841030909468 5E-6	1.191730746671279 4E-6	F	F
		2.8582255106701903E-6	9.5062651797364 63E-7				
		8.827462287769126E-7	2.1011521758180095E-6				

5		2.863348919661883E-7	1.9016186912623885E-9	1.5699636955559292E-6	8.513956284303565E-7	F	F
		3.669788481888311E-6	4.0694419432193757E-7				
		7.537677128132886E-7	2.1453410722778695E-6				
6		3.886084986662311 6E-7	2.6477327624563457E-9	1.4442402985525096E-6	1.056570270390189E-6	F	F
		3.081731885729386E-6	1.0343628447496624E-6				
		8.62380511179671E-7	2.132700233658448E-6				
7		1.279035357225818E-6	1.8531202495095344E-8	2.0094346582684 78E-6	8.418782296228725E-7	F	F
		3.715696895855482 2E-6	4.419741051096645E-7				
		1.033571721724134E-6	2.0651293812638576E-6				
8		4.043477182445632E-8	1.1261740047787663E-10	1.3155602649358484E-6	8.1439742765596E-7	F	F

		3.7167074766717504E-6	4.576202972351579E-7				
		1.895385463113386E-7	1.9854593683322443E-6				
9		8.437379462659222E-9	7.324356899048205E-7	1.1332195439451844E-6	1.302319205474211E-6	NF	NF
		3.5065307824965584E-7	2.1206263095825458E-6				
		3.0405681741232383E-6	1.0538956169352673E-6				
10		2.6073591916935685E-6	1.21529120139388E-6	9.621786310041352E-7	1.5571885801166982E-6	NF	NF
		2.5310783243333787E-7	2.0212415909069954E-6				
		2.5310783243333787E-7	2.0212415909069954E-6				
11		8.86528728132773E-7	2.07545382754718E-6	3.0264459335331534E-7	1.2428113379964656E-6	NF	NF
		2.0976554133326017E-8	1.3116315486800596E-6				

		4.2849779384694566E-10	3.413486377621566E-7				
12		1.907595531238902E-6	1.513289008760915E-6	1.8417468246254933E-6	1.4952706280799644E-6	NF	F
		6.147102822968982E-7	2.0847100616401105E-6				
		3.0029346603406796E-6	8.878128138388672E-7				
13		5.679174978513428E-12	4.394517946900631E-8	7.696610687001331E-8	1.1201748356157527E-6	NF	NF
		1.7639021373195795E-7	1.8412623863936435E-6				
		5.450242770310348E-8	1.475316940984608E-6				
14		4.013042218977528E-10	2.906884569971146E-7	8.03086817544504E-7	8.65862950382848E-7	NF	NF
		2.387241730623541E-6	1.135467262833257E-6				
		2.161741778807332 8E-8	1.171433131318172 2E-6				

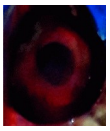
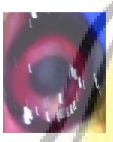
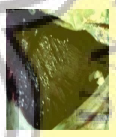
15		2.34246459 0146485E-18	2.97508924 2071751E-11	6.94710799 6929357E-8	8.23511149 217798E-7	NF	NF
		2.07728571 62241603E-7	2.08381918 56831934E-6				
		6.84668283 1222191E-10	3.86684511 07777994E-7				
16		3.31843790 85476253E-10	2.70852344 5447614E-7	7.72523844 7400427E-8	8.94295971 0559905E-7	NF	NF
		2.83766658 6024975E-10	3.16277969 39754985E-7				
		2.31141542 97255555E-7	2.09575759 92256605E-6				

Table Infomation :

F = Fresh

NF = Not Fresh

When using 100 training data with three parameters, there are results that 15 data are classified as correct and only one data is incorrect. Then calculated using the precision-recall formula to get an accuracy value of 72,9% with the following calculation:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

$$Accuracy = \frac{(8+7)}{(8+0+1+7)} = \frac{15}{16} = 93{,}75\,\%$$

Information :

True Positive (TP) = The prediction of fresh results is indeed fresh

True Negative (NF) = The predictions are not fresh are really not fresh

False Positive (FP) = Fresh predictions are not fresh

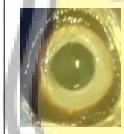
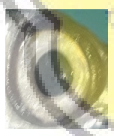
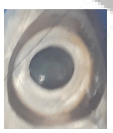
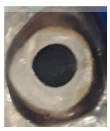
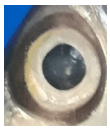
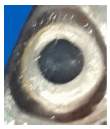
False Negative (FN) = Not Fresh predictions turn out to be fresh

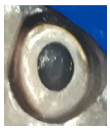
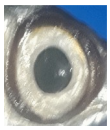
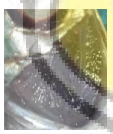
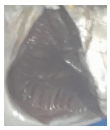
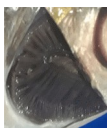
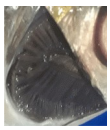
5.2.3 Testing 3

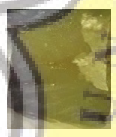
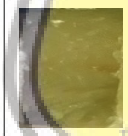
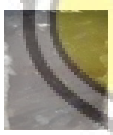
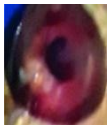
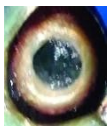
a. Testing 1 Parameter

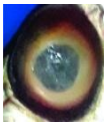
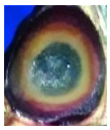
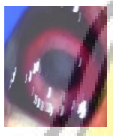
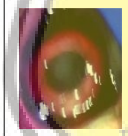
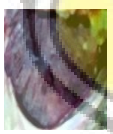
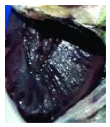
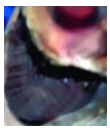
This is the third test table using one parameter:

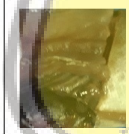
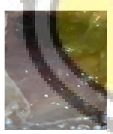
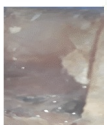
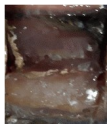
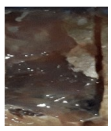
Table 5.5: One parameter testing table

No	Input	Mean			Nilai Probabilitas		Image	Output
		Red	Green	Blue	F	NF		
1		122,58	117,25	110,33	2,266146007812076 7E-6	1,291961804743111 4E-7	F	F
2		143,75	148,64	148,64	3,664059202158702E-8	1,030063827796757 1E-10	F	F
3		119,24	112,91	106,80	2,588775336106490 7E-6	2,256692794405196 1E-7	F	F
4		118,97	110,42	103,06	2,622582867783 55E-6	3,015034194122523E-7	F	F
5		141,38	139,97	134,80	1,992482764251493E-7	1,261018454590758E-9	F	F
6		135,44	138,49	136,70	2,695680258844255 7E-7	1,826214415519148 9E-9	F	F

7		126,13	130,00	129,97	8,348971721390833E-7	1,1914805356843132E-8	F	F
8		143,98	148,94	148,81	3,476376629786519E-8	9,607481057144936E-11	F	F
9		103,63	101,08	107,03	2,4216118857154485E-6	6,087002162434428E-7	F	F
10		93,72	94,88	98,54	1,5360477331504138E-6	1,1773076158176364E-6	F	F
11		123,21	113,01	105,26	2,4034408993484096E-6	1,9887835746300125E-7	F	F
12		103,66	101,08	107,03	2,4233659022520835E-6	6,079642302760974E-7	F	F
13		112,16	110,63	112,69	2,634197589170889E-6	2,4550586276304365E-7	F	F
14		107,47	103,73	101,15	2,594708466343718E-6	6,455558965583374E-7	F	F
15		117,54	112,36	106,22	2,659371689828407E-6	2,5637247377009857E-7	F	F
16		117,29	112,09	105,95	2,6722573890391966E-6	2,658289450965785E-7	F	F

17		99,72	93,12	92,15	1,5694540733604508E-6	1,3343018624243987E-6	F	F
18		135,76	131,23	124,05	6,931951071928658E-7	9,80453434856808E-9	F	F
19		111,70	106,76	101,10	2,709954550657485E-6	5,05554576587918E-7	F	F
20		96,92	91,24	86,35	1,1369797647179923E-6	1,5753740689033246E-6	F	NF
21		95,49	90,21	85,53	1,0139342906688073E-6	1,6437641592919894E-6	F	NF
22		96,16	89,78	87,91	1,1137523744354133E-6	1,6006241883988588E-6	F	NF
23		97,62	91,19	88,95	1,268481235751247E-6	1,5124857038911 69E-6	F	NF
24		91,34	82,43	74,96	3,6995938722642526E-7	1,8248523901282566E-6	F	NF
25		98,97	55,25	65,93	3,3917756108352396E-8	9,535064151723339E-7	NF	NF
26		107,20	104,52	94,43	2,3560753067404213E-6	7,764462621507177E-7	NF	F

27		93,21	89,90	91,53	1,1200548919446743E-6	1,5538603727179556E-6	NF	NF
28		99,96	94,28	100,70	1,8968715004716493E-6	1,0204785455213 69E-6	NF	F
29		76,47	21,67	63,69	1,0145231169413389E-10	1,2870771300383 04E-7	NF	NF
30		78,55	37,52	75,52	2,8833747894103767E-9	5,1477425370748 97E-7	NF	NF
31		36,19	10,28	22,03	8,04836841882609E-16	4,5286436079717 05E-10	NF	NF
32		78,08	36,89	75	2,483075531989153E-9	4,9011073447885 15E-7	NF	NF
33		89,44	79,12	90,36	5,595065120211251E-7	1,7607551739070495E-6	NF	NF
34		74,54	72,44	75,98	7,700143733581849E-8	1,6139386900881787E-6	NF	NF
35		69,67	71,69	80,79	6,231818787656558E-8	1,4703295481093 42E-6	NF	NF
36		94,83	81,42	92,88	8,512647794881545E-7	1,6116837359936744E-6	NF	NF

37		77,61	82,15	90,10	3,195915773972919E-7	1,6435783876781743E-6	NF	NF
38		99,72	99,33	105,34	2,1626275328577727E-6	7,495890964446559E-7	NF	F
39		83,29	81,11	85,43	3,7735151752309327E-7	1,8513916968682921E-6	NF	NF
40		57,46	61,60	61,21	2,1791583932870472E-9	5,74441091329135E-7	NF	NF
41		110,56	101,64	100,35	2,5684886384450185E-6	6,481446676314135E-7	NF	F
42		95,48	82,30	76,03	4,604775870950364E-7	1,7812255955314242E-6	NF	NF
43		68,81	57,92	56,49	3,2776466033725315E-9	6,389194224344442E-7	NF	NF
44		116,95	103,47	97,47	2,4789423451907386E-6	5,273212416552401E-7	NF	F
45		95,49	90,21	85,53	1,0139342906688073E-6	1,6437641592919894E-6	NF	NF
46		91,63	74,57	67,48	1,459223405136991E-7	1,591039198748477E-6	NF	NF

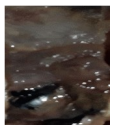
47		71,25	61,06	55,18	4,8124465298827644E-9	6,9865190104038E-7	NF	NF
48		90,95	82,46	77,79	4,266103625320162E-7	1,8698351387499258E-6	NF	NF

Table Information :

F = Fresh

NF = Not Fresh

Based on table 5.5 test results using 150 training data on one parameter 38 data are successfuly classified corectly and 10 data are incorrect. Therefore calculated using the precision-recall formula to get an accuracy value of 79,16% with the following calculation:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

$$Accuracy = \frac{(19+19)}{(19+5+5+19)} = \frac{38}{48} = 79{,}16\%$$ Information :

True Positive (TP) = The prediction of fresh results is indeed fresh

True Negative (NF) = The predictions are not fresh are really not fresh

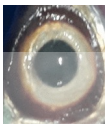
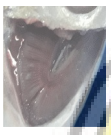
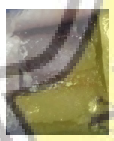
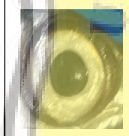
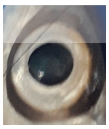
False Positive (FP) = Fresh predictions are not fresh

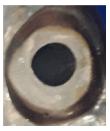
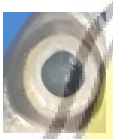
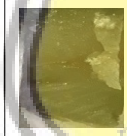
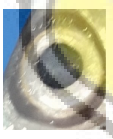
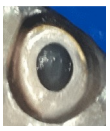
False Negative (FN) = Not Fresh predictions turn out to be fresh

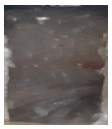
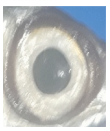
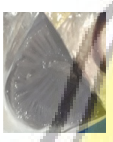
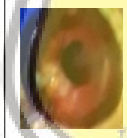
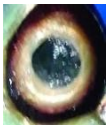
b. Testing 3 Parameter

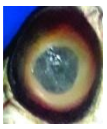
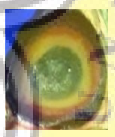
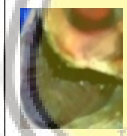
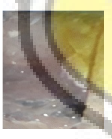
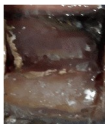
This is the third test table using three parameter:

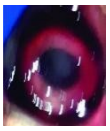
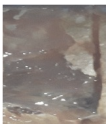
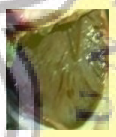
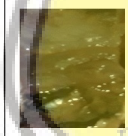
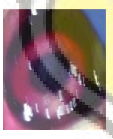
Table 5.6: Three parameter testing table

No	Input	Nilai Probabilitas		Nilai Probabilitas		Image	Output
		F	NF	F	NF		
1		2,2661460078120767E-6	1,2919618047431114E-7	2,0857373222959917E-6	6,907327530473843E-7	F	F
		2,4216118857154485E-6	6,0870021624344 28E-7				
		1,5694540733604508E-6	1,3343018624243987E-6				
2		3,6640592022158702E-8	1,0300638277967571E-10	7,5529447745495 57E-7	3,9573838551632803E-7	F	F
		1,5360477331504138E-6	1,1773076158176364E-6				
		6,931951071928658E-7	9,80453434856808E-9				
3		2,5887753361064907E-6	2,2566927944051961E-7	2,5673902620374 62E-6	3,1003407116381297E-7	F	F
		2,4034408993484096E-6	1,9887835746300125E-7				
		2,7099545506574 85E-6	5,05554576587918E-7				

4		2,62258286778355E-6	3,015034194122523E-7	2,0609761782512084E-6	8,282805728638915E-7	F	F
		2,4233659022520835E-6	6,079642302760974E-7				
		1,1369797647179923E-6	1,5753740689033246E-6				
5		1,992482764251493E-7	1,261018454590758E-9	1,2824600520882817E-6	6,301770135032078E-7	F	F
		2,634197589170889E-6	2,455058627630436 5E-7				
		1,0139342906688073E-6	1,6437641592919894E-6				
6		2,6956802588442557E-7	1,8262144155191489E-9	1,3260096222211856E-6	7,493354331242385E-7	F	F
		2,594708466343718E-6	6,455558965583374E-7				
		1,1137523744354133E-6	1,6006241883988588E-6				
7		8,348971721390833E-7	1,191480535 6843132E-8	1,587583365906246E-6	5,935909943393703E-7	F	F

		2,659371689828407E-6	2,5637247377009857E-7				
		1,268481235751247E-6	1,512485703891169E-6				
8		3,476376629786519E-8	9,607481057144936E-11	1,0256601808544957E-6	6,969258033451355E-7	F	F
		2,672257389039196 6E-6	2,658289450965785E-7				
		3,699593872264252 6E-7	1,824852390128256 6E-6				
9		3,391775610835239 6E-8	9,535064151723339E-7	1,0539709688581654E-6	1,1208020855702657E-6	NF	NF
		5,595065120211251E-7	1,760755173907049 5E-6				
		2,568488638445018 5E-6	6,481446676314135E-7				
10		2,356075306740421 3E-6	7,764462621507177E-7	9,645181103904253E-7	1,3905368492567736E-6	NF	NF
		7,700143733581849E-8	1,613938690088178 7E-6				

		4,604775870950364E-7	1,7812255955314242E-6				
11		1,1200548919446743E-6	1,5538603727179556E-6	3,9521690880820413E-7	1,2210364477539 14E-6	NF	NF
		6,231818787656558E-8	1,4703295481093 42E-6				
		3,2776466033725315E-9	6,3891942243444442E-7				
12		1,8968715004716493E-6	1,0204785455213 69E-6	1,7423595417168475E-6	1,0531611743900944E-6	NF	F
		8,5126477948815 45E-7	1,6116837359936744E-6				
		2,4789423451907386E-6	5,2732124165552401E-7				
13		1,0145231169413389E-10	1,2870771300383 04E-7	1,5520512340756 17E-7	1,1211084331434939E-6	NF	NF
		3,195915773972919E-7	1,6435783876781743E-6				
		1,0139342906688073E-6	1,6437641592919894E-6				

14		2,8833747894103767E-9	5,147742537074897E-7	7,458032657040804E-7	8,966499097231998E-7	NF	NF
		2,1626275328577727E-6	7,495890964446559E-7				
		1,459223405136991E-7	1,591039198748477E-6				
15		8,048368418826 09E-16	4,528643607971705E-10	1,273879882859376 4E-7	8,501654874231564E-7	NF	NF
		3,773515175230 9327E-7	1,851391696868292 1E-6				
		4,812446529882 7644E-9	6,986519010403 8E-7				
16		2,483075531989153E-9	4,901107344788515E-7	1,437575321524307 8E-7	9,781289881859707E-7	NF	NF
		2,179158393287047 2E-9	5,744410913291 35E-7				
		4,266103625320162E-7	1,869835138749925 8E-6				

Information Table

F = Fresh

NF = Not Fresh

Based on the test results using 150 training data with three parameter (Table 5.6) shows the results showed that 15 data successfully classified correctly and one data is wrong. Therefore calculated using the precision-recall formula to get an accuracy value of 93,75% with the following calculation:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

$$Accuracy = \frac{(8+7)}{(8+0+1+7)} = \frac{15}{16} = 93{,}75\,\%$$

Information :

True Positive (TP) = The prediction of fresh results is indeed fresh

True Negative (NF) = The predictions are not fresh are really not fresh

False Positive (FP) = Fresh predictions are not fresh

False Negative (FN) = Not Fresh predictions turn out to be fresh

5.2.4 Diagram of analysis

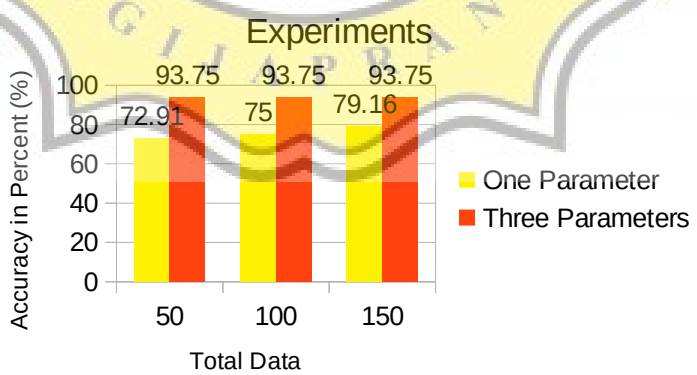


Illustration 5.7: Chart of Result

After doing three experiments using one parameter, illustration 5.6 gets an accuracy value of 72,9% with 50 training data. Then in the second experiment, the accuracy value of 100 training data is 75%. In the third experiment of 150 training

data, the results obtained 79,16% accuracy. Moreover, the three experiments using three parameters obtained the same value amounting to 93,75%.

Based on three experiments that use one parameter, several factors influence the algorithm's decision that the fish included in the either fresh or not fresh category, namely the mean value of R,G,B on each image, the standard deviation value, and the amount of in the training data. Furthermore, experiments using three parameters obtained the same accuracy value because each parameter's average probability value is sought in the experiment.