BAB III
DESAIN DAN PERANCANGAN

Pada Bab 3 akan dijelaskan mengenai perancangan sistem perangkat lunak untuk mencapai tujuan dari tugas akhir. Perancangan yang akan dijelaskan pada bab ini meliputi perancangan data, perancangan proses dan perancangan antar muka. Selain itu akan dijelaskan juga desain metode secara umum pada sistem.

3.1 Perancangan Data

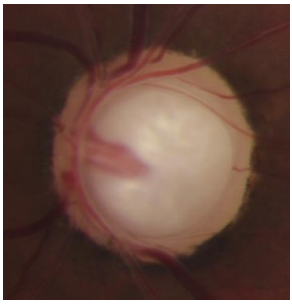
Perancangan data merupakan bagian yang terpenting dalam pengoperasian perangkat lunak karena diperlukan data yang tepat agar perangkat lunak dapat beroperasi dengan benar. Pada bagian ini akan dijelaskan mengenai perancangan data yang dibutuhkan untuk membangun perangkat lunak pendeteksi penyakit glaukoma dari citra fundus retina mata. Data yang diperlukan dalam pengoperasian perangkat lunak adalah data masukan (*input*) dan data keluaran (*output*) yang memberikan hasil proses pengoperasian perangkat lunak untuk pengguna.

3.1.1 Data Masukan

Data masukan adalah data awal yang akan diproses pada sistem pendeteksi penyakit glaukoma dari citra fundus retina mata. Data yang diproses berupa data citra yang direpresentasikan oleh tiga buah matriks dua dimensi (*channel red*, *green*, dan *blue*) dengan ukuran yang variatif (tidak ada batasan dalam ukuran citra).

Dalam tugas akhir ini, digunakan dataset citra retina mata yang didapat dari *database* RIM-ONE yang dapat diunduh langsung dari *http://medimrg.webs.ull.es/research/retinal-imaging/rim-one/*. Dataset berisi citra retina mata yang sudah dipotong sehingga citra hanya terfokus pada daerah yang terdapat optic disk yang hendak dievaluasi. Label kelas pada dataset terdiri

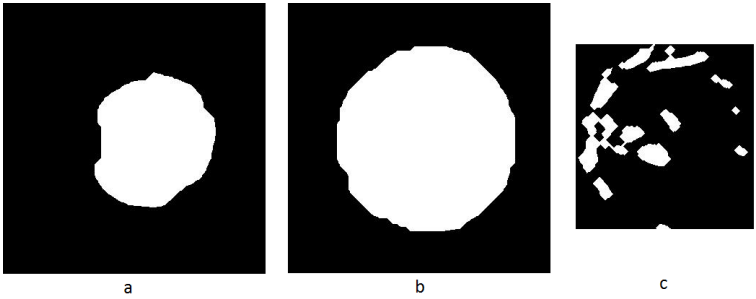
dari dua macam, yaitu normal dan glaukoma. Berikut contoh citra inputan yang digunakan.



Gambar 3.1 Citra Retina Mata Glaukoma

3.1.2 Data Luaran

Data luaran dari sistem ini adalah hasil klasifikasi dari data masukan dengan label glaukoma untuk kelas 1 dan label normal untuk kelas 0. Selain itu, didapat juga hasil segmentasi optic disk, optic cup, dan pembuluh darah dari citra masukan.



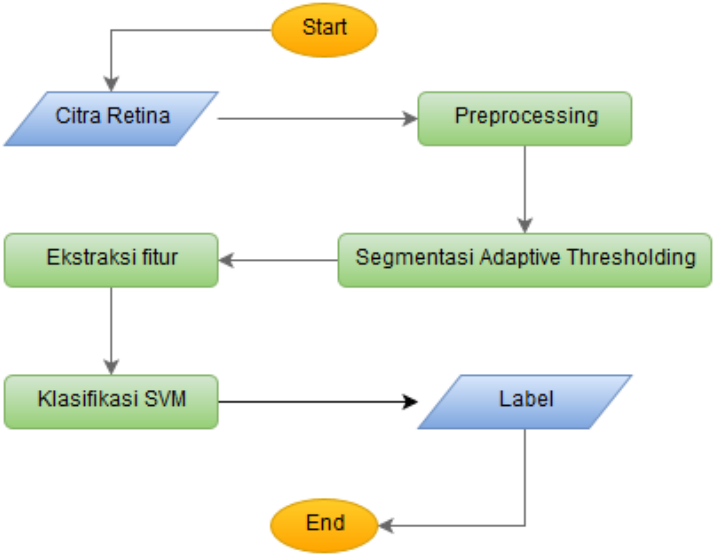
Gambar 3.2 Contoh hasil segmentasi (a) optic cup, (b) optic disk, (c) pembuluh darah di daerah optic disc

Gambar 3.2 menunjukkan contoh tampilan hasil segmentasi optic cup, optic disk, dan pembuluh darah dari citra retina mata pada Gambar 3.1.

3.2 Desain Metode Secara Umum

Pada tugas akhir ini akan dibangun suatu sistem untuk mendeteksi penyakit glaukoma dari citra masukan dengan menggunakan metode segmentasi *adaptive thresholding* dan metode klasifikasi *support vector machine*. Tahap pertama yang dilakukan opleh sistem adalah tahap *preprocessing*. Pada tahap ini dilakukan proses standarisasi, morfologi erosi dan dilasi pada segmentasi optic cup dan optic disk, serta bottom hat filtering, morfologi erosi dan dilasi pada segmentasi pembuluh darah. Tahap kedua adalah tahap segmentasi dengan menggunakan metode otsu. Tahap ketiga adalah tahap ekstraksi fitur. Fitur yang digunakan pada tugas akhir ini adalah CDR, ISNT pembuluh darah, dan ISNT NRR. Tahap yang terakhir adalah tahap klasifikasi dengan menggunakan pengklasifikasi *support vector machine*.

Bagan dari sistem yang dibangun ditunjukkan oleh Gambar 3.3.



Gambar 3.3 Diagram Alir Model Sistem

3.3 Perancangan Proses

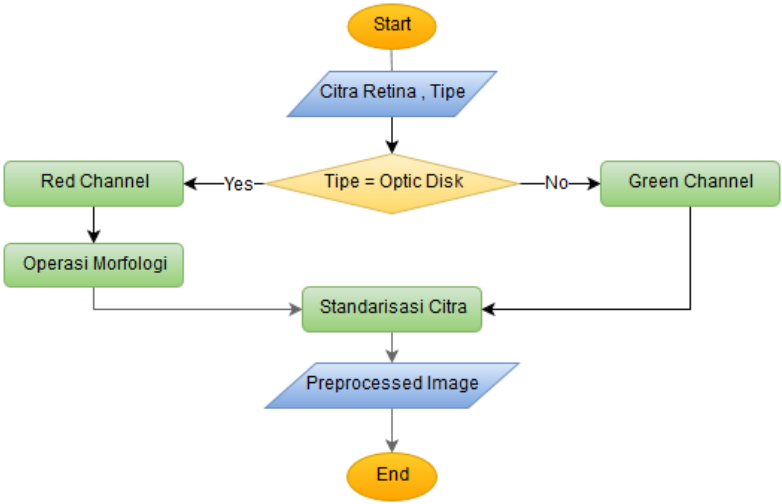
Berikut ini adalah rancangan dari sistem pendeteksi penyakit glaukoma dari citra fundus retina mata:

3.3.1 Tahap Preprocessing

Pada tahap ini, dilakukan proses awal sebelum masuk tahap segmentasi. Tahap ini dipecah menjadi dua proses yaitu preprocessing optic cup dan optic disk serta preprocessing pembuluh darah.

3.3.1.1 Preprocessing Optic Cup dan Optic Disk

Pada tahap ini terbagi menjadi dua proses yaitu, proses morfologi dan standarisasi. Berikut diagram alur dari tahap ini.



Gambar 3.4 Diagram Alir Preprocessing Optic Cup dan Optic Disk

3.3.1.1.1 Operasi Morfologi

Pada proses ini dilakukan operasi morfologi erosi dan dilasi dengan menggunakan elemen

struktur berbentuk disk dengan ukuran 25 untuk menutup gap pembuluh darah pada *red channel*.

3.3.1.1.2 Standarisasi Citra

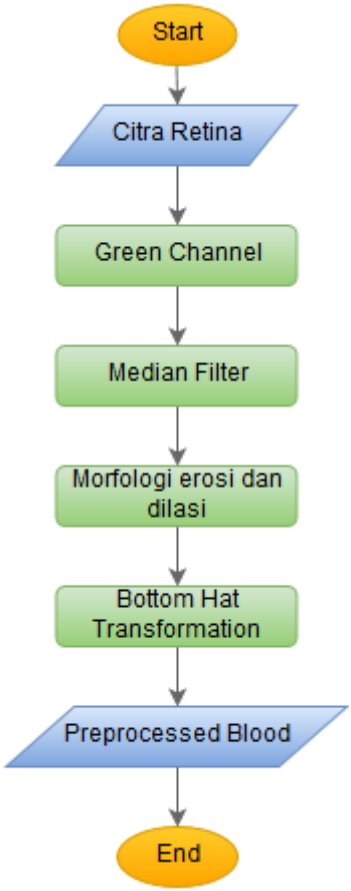
Pada tahap ini, dilakukan proses standarisasi citra dengan pseudocode sebagai berikut.

```
1  Standarize(I,tipe)
2      MEAN = mean(I)
3       Std = StandardDeviation(I)
4      if tipe == 'disk'
5          I = I - MEAN
6          Iter = 5
7      else
8          I = I - (MEAN+Std)
9          Iter = 7
10     end if
11     for x = 1 to iter
12         temp = nonzero(I)
13         I = I - minimum(temp)
14     end for
15     return I
```

3.3.1.2 Preprocessing Pembuluh Darah

Pada tahap ini, proses yang paling penting adalah proses *bottom hat transformation*. Hal ini dikarenakan pembuluh darah pada umumnya merupakan daerah tergelap pada citra retina mata, sehingga proses bottom hat transformation akan dapat mendeteksi daerah yang merupakan pembuluh darah.

Keluaran dari tahap ini akan menjadi masukan pada tahap segmentasi pembuluh darah. Berikut alur proses dari tahap *preprocessing* pembuluh darah.



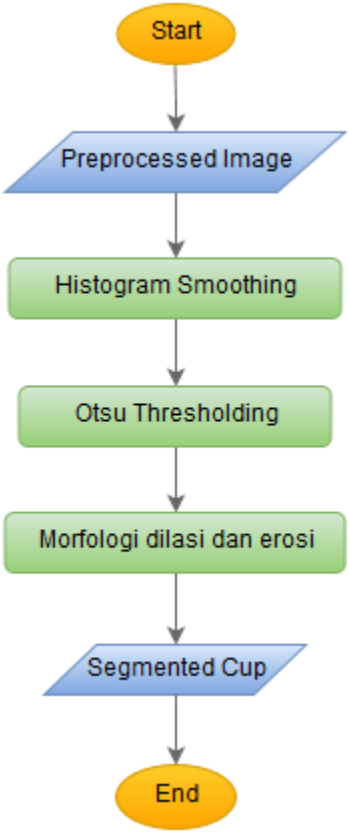
Gambar 3.5 Diagram Blood Preprocessing

3.3.2 Tahap Segmentasi

Pada tahap ini terbagi menjadi tiga bagian, yaitu segmentasi pembuluh darah, segmentasi optik disk, dan segmentasi optic cup.

3.3.2.1 Segmentasi Optic Cup

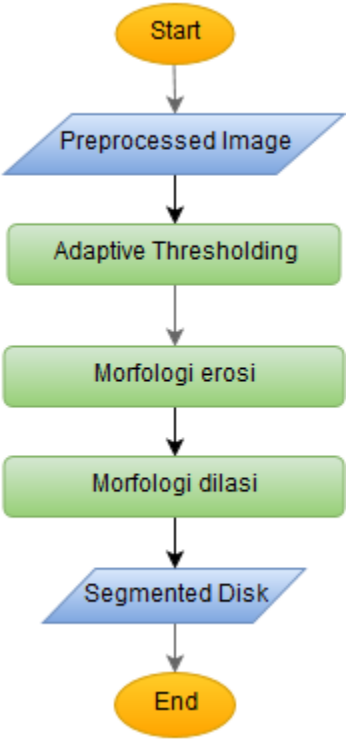
Pada tahap ini, dilakukan proses segmentasi optic cup dari citra yang telah melalui tahap preprocessing. Berikut diagram alurnya.



Gambar 3.6 Diagram Alur Segmentasi Optic Cup

3.3.2.2 Segmentasi Optic Disk

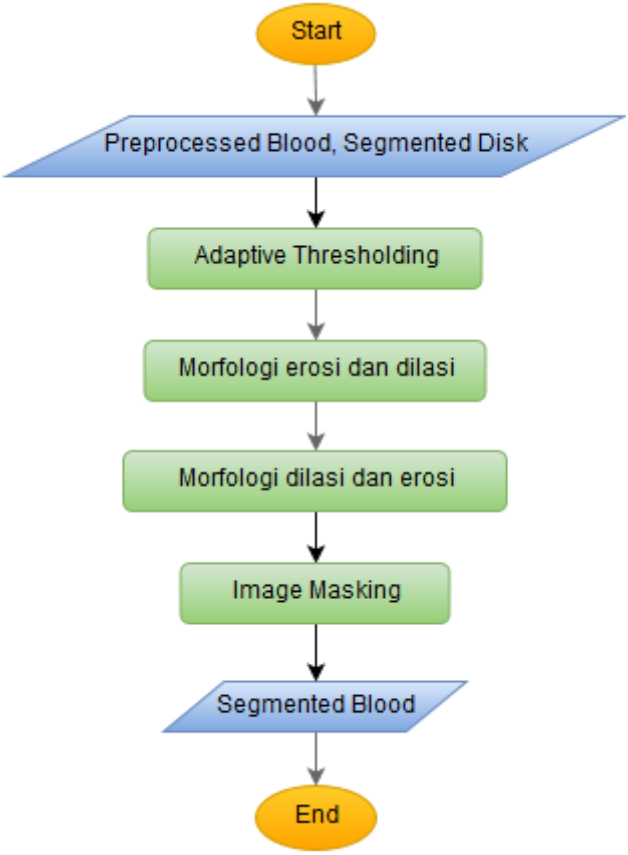
Pada tahap ini, dilakukan proses segmentasi optic disk dari citra yang telah melalui tahap preprocessing. Berikut diagram alurnya.



Gambar 3.7 Diagram Alur Segmentasi Optic Disk

3.3.2.3 Segmentasi Pembuluh Darah

Pada tahap ini dilakukan segmentasi pembuluh darah dari citra yang telah melalui tahap preprocessing. Berikut diagram alur dari tahap ini.



Gambar 3.8 Diagram Alur Segmentasi Pembuluh Darah

3.3.3 Tahap Ekstraksi Fitur

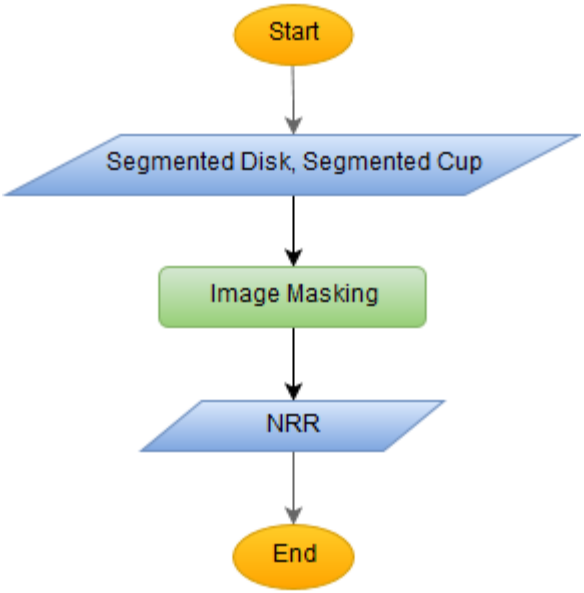
Pada tahap ini akan dilakukan proses ekstraksi fitur yang akan dipakai pada tahap klasifikasi. Fitur-fitur yang akan dipakai adalah *cup to disk ratio*, ISNT *neuro retinal rim*, ISNT pembuluh darah.

3.3.3.1 Cup to Disk Ratio

Untuk mendapatkan fitur *Cup to Disk Ratio* (CDR) adalah dengan menggunakan persamaan 2.2 dimana luas *optic cup* adalah luas dari daerah yang berhasil tersegmentasi pada proses segmentasi optic cup dan luas *optic disk* adalah luas dari daerah yang berhasil tersegmentasi pada tahap segmentasi *optic disk*.

3.3.3.2 ISNT Neuro Retinal Rim

Neuro retinal rim (NRR) adalah daerah di dalam *optic disk* yang bukan merupakan *optic cup*. Berikut ini adalah diagram alur untuk mendapatkan daerah NRR.



Gambar 3.9 Diagram Alur Neuro Retinal Rim

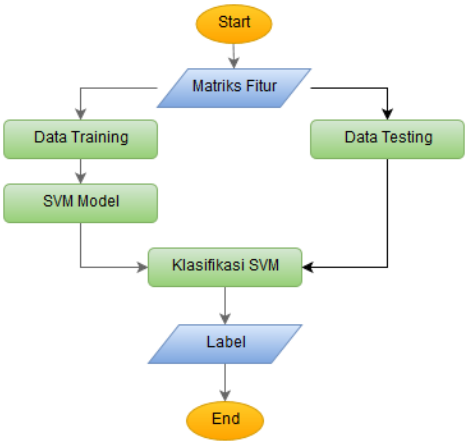
Setelah NRR didapatkan, maka penghitungan ISNT dilakukan dengan menggunakan persamaan 2.1.

3.3.3.3 ISNT Pembuluh Darah

Pada tahap ini, pembuluh darah yang berhasil tersegmentasi dihitung ISNT-nya dengan menggunakan persamaan 2.1.

3.3.4 Tahap Klasifikasi

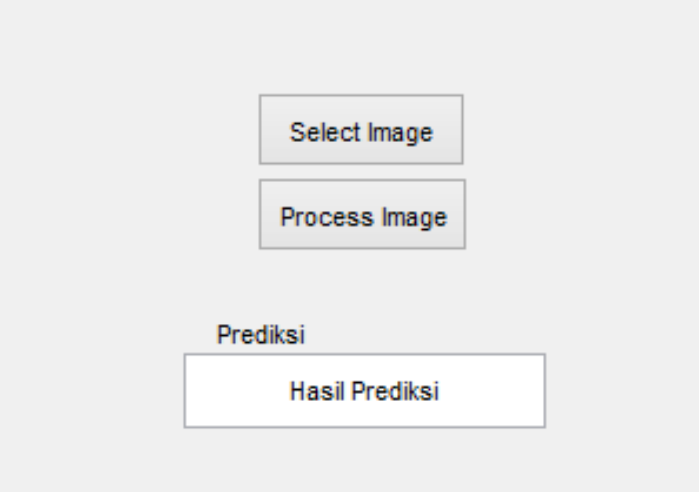
Fitur CDR, ISNT NRR, dan ISNT pembuluh darah hasil dari tahap ekstraksi fitur menjadi masukan pada tahap klasifikasi. Sebelum dilakukan proses klasifikasi, ketiga fitur tersebut dibagikan menjadi data training dan data testing. Data training adalah data yang digunakan untuk membuat model klasifikasi yang digunakan. Sedangkan data testing digunakan untuk proses uji coba model klasifikasi yang telah dibuat. Metode klasifikasi yang digunakan adalah support vector machine. Hasil dari tahap klasifikasi ini adalah kelas glaukoma yang merepresentasikan angka 1, dan kelas normal yang merepresentasikan angka 0 untuk tiap data masukan. Berikut diagram alur dari tahap ini.



Gambar 3.10 Diagram Alur Klasifikasi

3.4 Perancangan Antarmuka Perangkat Lunak

Perancangan antarmuka perangkat lunak merupakan penggambaran halaman antarmuka yang akan digunakan pengguna ketika berinteraksi dengan aplikasi. Gambar 3.11 menunjukkan rancangan antarmuka untuk tombol inputan dan keluaran hasil prediksi.

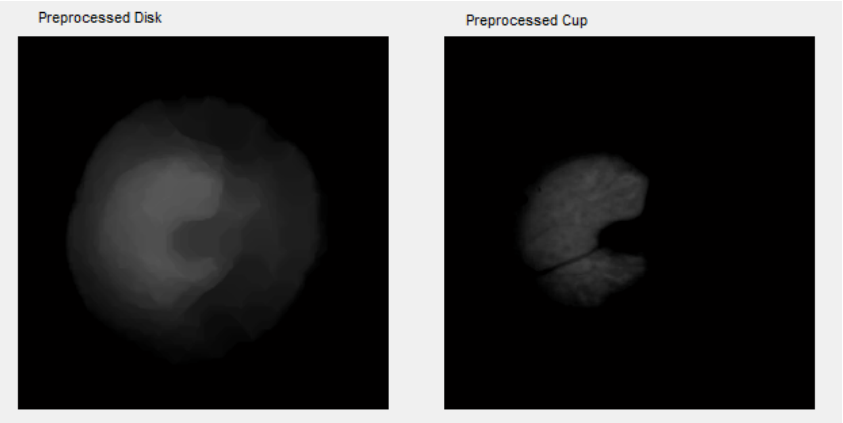


Gambar 3.11 Antarmuka Masukan dan Keluaran Hasil Prediksi

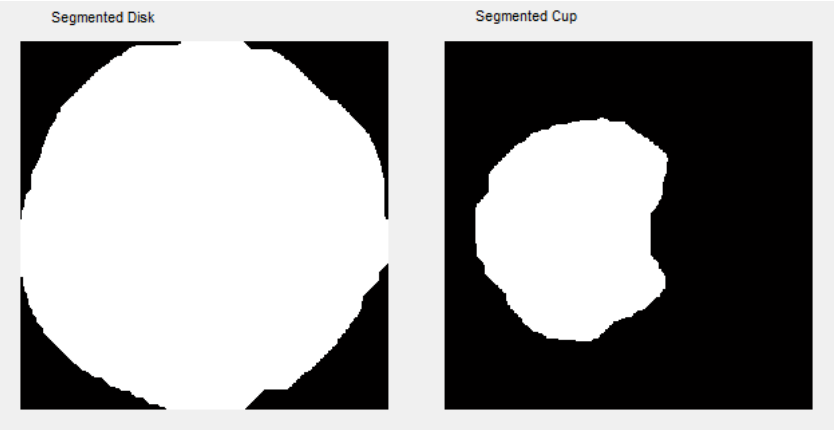
Tombol *Select Image* digunakan untuk memilih citra retina sebagai masukan. Ketika tombol ini diklik akan muncul kotak dialog buka yang bertujuan untuk memilih berkas yang akan diproses. Setelah berkas selesai dipilih, maka akan ditampilkan di *axes* (*image holder* pada matlab) *Source Image*. Berikut antarmuka untuk menampilkan berkas yang dipilih.

Gambar 3.12 Antarmuka *Image Holder* Citra Masukan

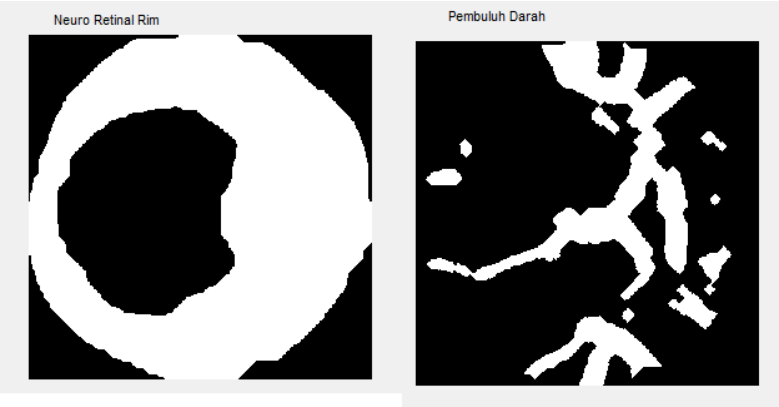
Setelah berkas berhasil dipilih, maka selanjutnya tombol *Process Image* ditekan. Tombol ini berfungsi untuk menjalankan semua proses pendeteksian mulai dari *preprocessing*, segmentasi, ekstraksi fitur, dan klasifikasi. Hasil *preprocessing* dan segmentasi dari *optic disk* akan ditampilkan pada *axes Preprocessed Disk* dan *Segmented Disk.* Sedangkan hasil *preprocessing* dan segmentasi dari *optic cup* akan ditampilkan pada *axes Preprocessed Cup* dan *Segmented Cup.* Selain itu, hasil segmentasi pembuluh darah akan ditampilkan pada *axes* Pembuluh Darah. Untuk hasil klasifikasi akan ditampilkan pada *text holder* Hasil Prediksi. Berikut tampilan aplikasi setelah tombol *Process Image* ditekan.



Gambar 3.13 Antarmuka *Image Holder* Hasil *Preprocessing* *Optic cup* dan *Optic Disk*



Gambar 3.14 Antarmuka *Image Holder* Hasil Segmentasi *Optic cup* dan *Optic Disk*



Gambar 3.15 Antarmuka *Image Holder* Hasil Segmentasi Pembuluh Darah dan *Neuro Retinal Rim*

(Halaman ini sengaja dikosongkan)

BAB IV
IMPLEMENTASI

Bab implementasi berisi pembahasan mengenai implementasi perangkat lunak berdasarkan perancangan yang telah dibuat. Tahap perancangan merupakan tahap dasar dari implementasi perangkat lunak.

4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat Keras
   a. Processor: Intel® Core™ i3 CPU-3240 @ 3.40GHz
   b. Memory (RAM) : 4096 MB
   c. Tipe Sistem: 64-bit
2. Perangkat Lunak
   a. Sistem operasi : *Window*s 8 Professional 64 bit.
   b. Perangkat Pengembang : Matlab R2014a
   c. Perangkat Pembantu : Microsoft Excel 2013

4.2 Implementasi

Sub bab implmentasi ini menjelaskan tentang implementasi proses yang sudah dijelaskan pada bab perancangan perangkat lunak.

4.2.1 Implementasi Preprocessing

4.2.1.1 Preprocessing Optic Disk dan Optic Cup

Tahap pertama pada proses ini adalah menentukan channel citra yang akan dipakai. *Optic disk* menggunakan *red channel* sedangkan *optic cup* menggunakan *green channel*. Setelah itu, jika *channel* yang terpilih adalah *red channel* maka akan dilakukan

operasi morfologi dilasi dan erosi dengan elemen stuktur berbentuk disk dengan radius 25. Setelah itu dilakukan proses standarisasi untuk memperjelas objek *optic disk* dan *optic cup* pada *channel* terpilih. Berikut ini adalah kode program implementasi tahap *preprocessing optic cup* dan *optic disk*.

```
function y = Preprocessing(x,tipe)
    y = double(x);
    y1 = double(x);
    if(strcmp(tipe,'disk') == 1)
        y = imdilate(y,strel('disk',25));
        y = imerode(y,strel('disk',25));
    end
    tmp = y1(:);
    xBar = mean(tmp)
    xStd = std(tmp)
    if(strcmp(tipe,'disk') == 1)
        y1 = y1-(xBar);
        y = y-(xBar);
    elseif (strcmp(tipe,'cup') == 1)
        y1 = y1-(xBar+xStd);
        y = y-(xBar+xStd);
    end
    if(strcmp(tipe,'disk') == 1)
        iter = 5;
    else
        iter = 7;
    end
    y = uint8(y);
    x = y;
    x(x==0) = max(max(y));
    for i = 1 : iter
        y = uint8(y - min(min(x)));
    end
end
```

Kode sumber 4.1 Implementasi Tahap Preprocessing Optic Cup dan Optic Disk

4.2.1.2 Preprocessing Pembuluh Darah

Pada tahap ini, *channel* yang digunakan adalah *green channel*. Ini dilakukan karena keberadaan pembuluh darah lebih terlihat pada *green channel*. Tahap selanjutnya adalah *median filtering* dengan ukuran *neighborhood window* 3x3. Setelah itu dilakukan operasi morfologi erosi dengan elemen struktur berbentuk disk dengan radius 5 dan dilasi dengan elemen struktur berbentuk disk dengan radius 3. Setelah itu dilakukan *bottom hat transformation* dengan elemen struktur berbentuk *diamond* dengan ukuran 20. Berikut ini adalah kode program implementasi tahap *preprocessing* pembuluh darah.

```
1 greenChan =  medfilt2(greenChan);
2 greenChan = imerode(greenChan,strel('disk',5));
3 greenChan = imdilate(greenChan,strel('disk',3));
4 gr = BottomHat(greenChan,strel('diamond',20));
```

Kode sumber 4.2 Implementasi Tahap Preprocessing Pembuluh Darah

```
1 function I = BottomHat(green,SE)
2     I = imdilate(green,SE);
3     I = imerode(I,SE);
4     I = I-green;
5 end
```

Kode sumber 4.3 Implementasi Bottom Hat Transformation

4.2.2 Implementasi Segmentasi

Pada tahap ini, dilakukan implementasi segmentasi dari citra yang telah melalui tahap *preprocessing*. Tahap ini terbagi menjadi 3 bagian, yaitu segmentasi *optic cup*, *optic disk*, dan pembuluh darah.

4.2.2.1 Segmentasi Optic Cup

Langkah pertama dari tahap ini adalah melakukan *smoothing histogram* dari citra yang telah melalui tahap *preprocessing*. Filter yang digunakan adalah filter gaussian dengan

rerata 50, standar deviasi 6, dan ukuran *window* 1x100. Setelah itu dilakukan thresholding dengan level yang diambil dari *smoothed histogram* dengan menggunakan metode otsu. Setelah itu dilakukan morfologi dilasi dan erosi menggunakan elemen struktur berbentuk disk dengan radius 50. *Optic cup* adalah objek yang memiliki area terluas dari objek-objek yang berhasil tersegmentasi. Berikut kode program implementasi segmentasi *optic cup.*

```
1  function [cup,area,green] = SegmenCup(I)
2      green = I(:,:,2);
3      I = Preprocessing(green,'cup');
4      level = HistogramSmoothing(I,'cup');
5      iBw = im2bw(I,level);
6      iBw = imdilate(iBw,strel('disk',50));
7      iBw = imerode(iBw,strel('disk',50));
8      stats = regionprops(iBw,'basic');
9      [b,idx] = sort([stats.Area],'ascend');
10     l = double(int64((mean2(b)+b(1))/2));
11     iBw = bwareaopen(iBw, l);
12     cup = iBw;
13     area = sum(sum(cup));
14 end
```

Kode sumber 4.4 Implementasi Segmentasi Optic Cup

```
1  function y = HistogramSmoothing(Image)
2      [p,val] = imhist(Image);
3      p = p./numel(Image);
4      mean = 50;
5      sigma = 6;
6      f = normpdf(0:100,mean,sigma);
7      p1 = conv(p,f);
8      y = otsu(p1);
9  End
```

Kode sumber 4.5 Implementasi Fungsi Histogram Smoothing

```
1  function level = otsu(I)
2      histogramCounts = imhist(I);
3      total = sum(histogramCounts);
4      sumB = 0;
5      wKelas1 = 0;
6      maximum = 0.0;
7      histogramCounts = histogramCounts/total;
8      sum1 = dot( (0:255), histogramCounts);
9      total = sum(histogramCounts);
10     for i=1:256
11         wKelas1 = wKelas1 + histogramCounts(i);
12         if (wKelas1 == 0)
13             continue;
14         End
15         wKelas2 = total - wKelas1;
16         if (wKelas2 == 0)
17             break;
18         End
19         sumB = sumB + (i-1) * histogramCounts(i);
20         meanKelas1 = sumB / wKelas1;
21         meanKelas2 = (sum1 - sumB) / wKelas2;
22         stdAntarKelas = wKelas1 * wKelas2 * (meanKelas1 - meanKelas2) * (meanKelas1 - meanKelas2);
23         if ( stdAntarKelas >= maximum )
24             level = i-1;
25             maximum = stdAntarKelas;
26         End
27     End
28     level = level/255;
29 End
```

Kode sumber 4.6 Implementasi Metode Otsu

4.2.2.2 Segmentasi Optic Disk

Pada tahap ini, level untuk thresholding didapat dari jumlah nilai minimum dan rerata dari citra yang telah melalui tahap *preprocessing* dibagi 2. Setelah dilakukan operasi thresholding, dilakukan operasi morfologi erosi menggunakan elemen struktur berbentuk disk dengan radius 50. Setelah itu, objek yang memiliki area terbesar dipilih, dan kemudian dilakukan operasi morfologi dilasi dengan menggunakan elemen struktur berbentuk disk dengan radius 50. Berikut kode program dari tahap segmentasi *optic disk.*

```
1  function [disk , index, area] = DiscSegmentation(I)
2      red = I(:,:,1);
3      pre = Preprocessing(red,'disk');
4      level = pre(:);
5      level(level == 0) = max(level);
6      mini = min(level);
7      level = pre(:);
8      level(level == 0) = NaN;
9      rerata = nanmean(level);
10     rerata = (mini + rerata)/2;
11     iBw = pre>(mini + rerata)/2;
12     iBw = imerode(iBw,strel('disk',50));
13     stats = regionprops(iBw,'basic');
14     [b,idx] = sort([stats.Area],'descend')
15     index = stats(idx(1)).Centroid;
16     index = int32(index);
17     iBw = bwareaopen(iBw, b(1));
18     iBw = imdilate(iBw,strel('disk',50));
19     disk = iBw;
20     area = sum(sum(disk));
21 End
```

Kode sumber 4.7 Implementasi Segmentasi Optic Disk

4.2.2.3 Segmentasi Pembuluh Darah

Pada tahap ini, dilakukan operasi thresholding dengan nilai level 2.7 kali standar deviasi dari citra yang sudah melalui tahap *preprocessing*. Setelah itu dilakukan operasi morfologi erosi dan

dilasi dengan menggunakan elemen struktur berbentuk *diamond* dengan ukuran 5, kemudian dilakukan operasi morfologi dilasi dan erosi dengan elemen struktur yang sama. Setelah itu, dilakukan *masking* dengan *mask optic disk* untuk mendapatkan pembuluh darah yang hanya berada di daerah *optic disk*. Berikut ini adalah kode program dari tahap segmentasi pembuluh darah.

```
1 level = 2.7*std(std(double(gr)))/255;
2 y = im2bw(gr,level);
3 y = imopen(y,strel('diamond',5));
4 y = imclose(y,strel('diamond',5));
5 y = y.*disk;
```

Kode sumber 4.8 Implementasi Segmentasi Pembuluh Darah

4.2.3 Implementasi Ekstraksi Fitur

Pada tahap ini dilakukan implementasi ekstraksi fitur yang akan dipakai sebagai masukan pada tahap klasifikasi.

4.2.3.1 Cup to Disk Ratio

Berikut kode program untuk menghitung fitur *cup to disk ratio*.

```
1    diskArea = sum(sum(disk));
2    cupArea = sum(sum(cup));
3     CupToDiskRatio = cupArea/diskArea;
4     CupToDiskRatio = sqrt(CupToDiskRatio);
```

Kode sumber 4.9 Implementasi Ekstraksi Fitur CDR

4.2.3.2 ISNT Neuro Retinal Rim

Untuk mendapatkan fitur ini, tahap pertama yang dilakukan adalah dengan melakukan *cropping* pada *optic disk* dan *optic cup* untuk menyeragamkan ukuran NRR dan pembuluh darah. Berikut adalah kode program untuk melakukan *cropping*.

```
1 [row,col]=find(disk);
2 row_t = min(row);
3 col_t = min(col);
```

```
4  disk = imcrop(disk,[col_t row_t max(col)-col_t
   max(row)-row_t]);
5  cup = imcrop(cup,[col_t row_t max(col)-col_t
   max(row)-row_t]);
```

Kode sumber 4.10 Implementasi Proses Cropping

Setelah itu dilakukan penghitungan fitur ISNT NRR dengan kode program sebagai berikut.

```
1  function [imNrr,y] = NRR(imCup,imDisk)
2      mask = imread('mask.png');
3      mask = im2bw(mask);
4      imNrr = logical(imDisk .* (1-imCup));
5      [h,w] = size(imNrr);
6      mask1 = imresize(mask,[h w]);
7      temporal = sum(sum(imNrr.*mask1));
8      mask = imrotate(mask,90);
9      mask1 = imresize(mask,[h w]);
10     superior = sum(sum(imNrr.*mask1));
11     mask = imrotate(mask,90);
12     mask1 = imresize(mask,[h w]);
13     nasal = sum(sum(imNrr.*mask1));
14     mask = imrotate(mask,90);
15     mask1 = imresize(mask,[h w]);
16     inferior = sum(sum(imNrr.*mask1));
17     y = (superior+inferior)/(nasal+temporal);
18 end
```

Kode sumber 4.11 Implementasi Ekstraksi Fitur ISNT NRR

4.2.3.3 ISNT Blood Vessel

Untuk mendapatkan fitur ini, tahap pertama yang dilakukan adalah dengan melakukan *cropping* pada pembuluh darah untuk menyeragamkan ukuran dengan NRR. Berikut adalah kode program untuk melakukan *cropping*.

```
1  [row,col]=find(disk);
2  row_t = min(row);
```

```
3  col_t = min(col);
4  blood = imcrop(blood,[col_t row_t
   max(col)-col_t max(row)-row_t]);
```

Kode sumber 4.12 Implementasi Cropping Pembuluh Darah

Setelah itu dilakukan penghitungan fitur ISNT pembuluh darah dengan kode program sebagai berikut.

```
1  function y = ISNTBlood(imBlood)
2      mask = imread('mask.png');
3      mask = im2bw(mask);
4      [h,w] = size(imBlood);
5      mask1 = imresize(mask,[h w]);
6      temporal = sum(sum(imBlood.*mask1));
7      mask = imrotate(mask,90);
8      mask1 = imresize(mask,[h w]);
9      superior = sum(sum(imBlood.*mask1));
10     mask = imrotate(mask,90);
11     mask1 = imresize(mask,[h w]);
12     nasal = sum(sum(imBlood.*mask1));
13     mask = imrotate(mask,90);
14     mask1 = imresize(mask,[h w]);
15     inferior = sum(sum(imBlood.*mask1));
16     y = (superior+inferior)/(nasal+temporal);
17 end
```

Kode sumber 4.13 Implementasi Ekstraksi Fitur ISNT Pembuluh Darah

4.2.4 Implementasi Klasifikasi

Tahap setelah ekstraksi fitur adalah klasifikasi menggunakan metode *support vector machine*. Sebelum dilakukan klasifikasi, data masukan dibagi ke dalam 2 kategori yaitu data untuk *training* dan data untuk *testing.* Data dipisahkan menggunakaan metode *K-fold cross validation* dengan k = 10 menjadi data *training* dan data *testing*. Berikut adalah kode

program implementasi klasifikasi menggunakan *support vector machine* dan *10-fold cross validation*.

```
1  k=10;
2  newDataset = Dataset;
3  newGroup = Group;
4  cvFolds = crossvalind('Kfold', newGroup, k);
   %# get indices of 10-fold CV
5  cp = classperf(newGroup);                    %#
   init performance tracker
6  dataset = newDataset;
7  metode = 'QP';
8  tic;
9  for i = 1:k                                  %#
   for each fold
10     testIdx = (cvFolds == i);                %#
   get indices of test instances
11     trainIdx = ~testIdx;                     %#
   get indices training instances
12
13     %# train an SVM model over training
   instances
14     svmModel = svmtrain(dataset(trainIdx,:),
   newGroup(trainIdx), ...
15                    'Autoscale',true,
   'Showplot',false, 'Method',metode, ...
16                    'BoxConstraint',2e-1,
   'Kernel_Function','rbf','rbf_sigma',1);
17     %# test using test instances
18     pred       =       svmclassify(svmModel,
   dataset(testIdx,:), 'Showplot',false);
19     %# evaluate and update performance object
20     cp = classperf(cp, pred, testIdx);
21 end
22 %# get accuracy
23 akurasi = cp.CorrectRate
24 %# get confusion matrix
```

```
25 %# columns:actual, rows:predicted, last-row: unclassified instances
26 confusionMatriks = cp.CountingMatrix
```

Kode sumber 4.14 Implementasi Klasifikasi

4.2.5 Implementasi User Interface

Pada tahap ini, dilakukan implementasi *user interface* aplikasi pendeteksi penyakit glaukoma dari citra fundus retina mata. Berikut kode program dari tahap implementasi *user interface* aplikasi ini.

```
1  function varargout = GUI_TA(varargin)
2  gui_Singleton = 1;
3  gui_State = struct('gui_Name',       mfilename, ...
4                     'gui_Singleton',  gui_Singleton, ...
5                     'gui_OpeningFcn', @GUI_TA_OpeningFcn, ...
6                     'gui_OutputFcn',  @GUI_TA_OutputFcn, ...
7                     'gui_LayoutFcn',  [] , ...
8                     'gui_Callback',   []);
9  if nargin && ischar(varargin{1})
10     gui_State.gui_Callback = str2func(varargin{1});
11 end
12 if nargout
13     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
14 else
15     gui_mainfcn(gui_State, varargin{:});
16 end
17 function GUI_TA_OpeningFcn(hObject, eventdata, handles, varargin)
18 handles.output = hObject;
19 guidata(hObject, handles);
20 function varargout = GUI_TA_OutputFcn(hObject, eventdata, handles)
```

```
varargout{1} = handles.output;
function pushbutton1_Callback(hObject, eventdata, handles)
[filename, user_canceled] = imgetfile;
if (user_canceled == 0)
    I = imread(filename);
    handles.I = I;
    guidata(hObject,handles);
    axes(handles.axes1);
    imshow(I);
end
function pushbutton2_Callback(hObject, eventdata, handles)
I = handles.I;
%processing image
preprocessedDisk = Preprocessing(I(:,:,1),'disk');
preprocessedCup = Preprocessing(I(:,:,2),'cup');
[disk,index,diskArea] = DiscSegmentation(I);
[cup,cupArea] = SegmenCup(I);
blood = BloodVessel(I(:,:,2),disk);
%cropping
[row,col]=find(disk);
row_t = min(row);
col_t = min(col);
disk = imcrop(disk,[col_t row_t max(col)-col_t max(row)-row_t]);
cup = imcrop(cup,[col_t row_t max(col)-col_t max(row)-row_t]);
blood = imcrop(blood,[col_t row_t max(col)-col_t max(row)-row_t]);
%viewing image
axes(handles.axes2);
imshow(blood);
axes(handles.axes3);
imshow(preprocessedDisk);
axes(handles.axes4);
```

```
52 imshow(disk);
53 axes(handles.axes5);
54 imshow(preprocessedCup);
55 axes(handles.axes6);
56 imshow(cup);
57 %extracting feature
58 CupToDiskRatio = sqrt(cupArea/diskArea);
59 bloodISNT = ISNTBlood(blood);
60 [imNrr,nrrISNT] = NRR(cup,disk);
61 newData = csvread('newdataTA_120.csv');
62 newDataset = [newData(:,1) newData(:,2)
   newData(:,3)];
63 newGroup = newData(:,4);
64 svmModel = svmtrain(newDataset, newGroup, ...
65                  'Autoscale',true,
   'Showplot',false, 'Method','SMO', ...
66                  'BoxConstraint',2e-1,
   'Kernel_Function','linear');
67 pred = svmclassify(svmModel, [CupToDiskRatio
   nrrISNT bloodISNT], 'Showplot',false);
68 if pred == 0
69     set(handles.edit3,'string','Glaukoma');
70 else
71     set(handles.edit3,'string','Normal');
72 end
73 function edit3_Callback(hObject, eventdata,
   handles)
74 function edit3_CreateFcn(hObject, eventdata,
   handles)
75 if ispc && isequal(get(hObject,'BackgroundColor'),
   get(0,'defaultUicontrolBackgroundColor'))
76     set(hObject,'BackgroundColor','white');
77 End
```

Kode sumber 4.15 Implementasi User Interface

(Halaman ini sengaja dikosongkan)

BAB V
UJI COBA DAN EVALUASI

Pada bab ini dijelaskan mengenai rangkaian uji coba perangkat lunak. Pada bab ini juga akan dibahas mengenai perhitungan akurasi dari perangkat lunak yang dibuat. Tujuan dari uji coba ini adalah untuk mengetahui apakah model yang telah dihasilkan sistem mampu mendeteksi penyakit glaukoma pada citra fundus retina mata dengan benar, mengetahui fungsi kernel pada *support vector machine* yang optimal, dan mengetahui metode yang dipakai untuk pencarian *hyperplane* pada *support vector machine* yang optimal untuk melakukan klasifikasi. Selain itu juga pada bab ini akan dipaparkan pembahasan yang meliputi lingkungan uji coba, data uji coba, skenario uji coba, hasil uji coba, dan evaluasi.

5.1 Lingkungan Uji Coba

Lingkungan uji coba yang akan digunakan untuk melakukan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat Keras
   a. Processor: Intel® Core™ i3 CPU-3240 @ 3.40GHz
   b. Memory (RAM) : 4096 MB
   c. Tipe Sistem: 64-bit
2. Perangkat Lunak
   a. Sistem operasi : *Window*s 8 Professional 64 bit.
   b. Perangkat Pengembang : Matlab R2014a
   c. Perangkat Pembantu : Microsoft Excel 2013

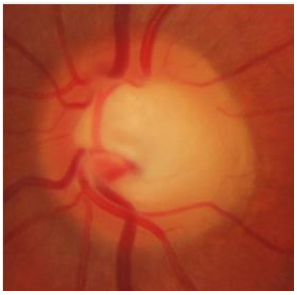
5.2 Data Uji Coba

Data yang digunakan untuk uji coba adalah citra retina mata yang diambil dari database RIM-ONE v2 yang terdiri dari 56 retina mata yang normal dan 64 retina mata yang glaukoma.

Uji coba dilakukan dengan menggunakan skema 10-*fold cross validation* yang dilakukan sebanyak 5 kali untuk masing-masing skenario. Hasil uji coba pada bab ini adalah hasil rata-rata dari 5 kali percobaan 10-*fold cross validation*.

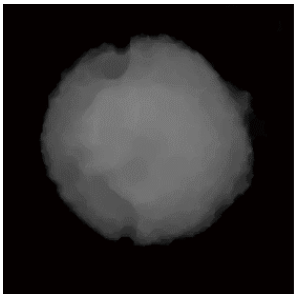
5.3 Uji Coba Sistem

Pada bagian ini akan ditampilkan hasil dari masing-masing proses pada sistem pendeteksi penyakit glaukoma dari citra retina mata.

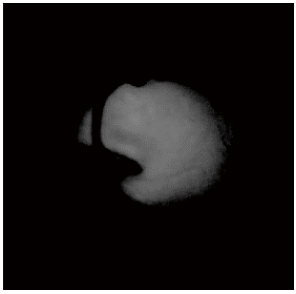


Gambar 5.1 Citra masukan sistem

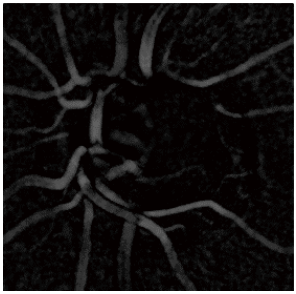
Tahap pertama yang dilalui citra masukan adalah tahap *preprocessing*. Gambar 5.2, 5.3, dan 5.4 secara berturut turut adalah hasil dari tahap *preprocessing optic disk*, *optic cup*, dan pembuluh darah.



Gambar 5.2 Hasil *preprocessing optic disk*

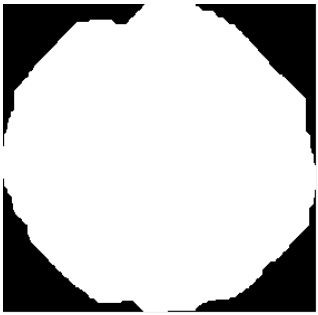


Gambar 5.3 Hasil *preprocessing optic cup*

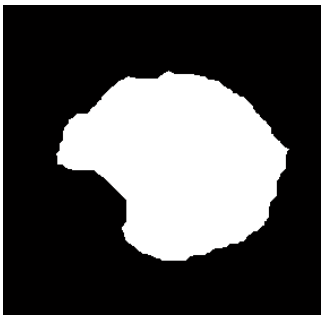


Gambar 5.4 Hasil *preprocessing* pembuluh darah

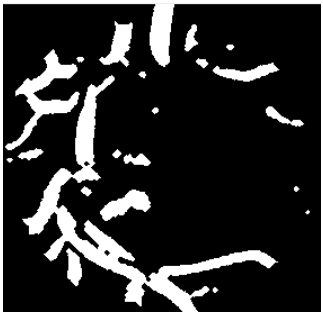
Setelah tahap ini, masing-masing citra keluaran akan masuk pada tahap segmentasi. Gambar 5.5, 5.6 dan 5.7 secara berturut turut adalah hasil dari tahap segmentasi *optic disk*, *optic cup*, dan pembuluh darah.



Gambar 5.5 Hasil segmentasi *optic disk*



Gambar 5.6 Hasil segmentasi *optic cup*



Gambar 5.7 Hasil segmentasi pembuluh darah

Setelah citra berhasil disegmentasi, maka akan dihitung nilai dari masing-masing fitur yang akan dipakai oleh sistem. Berikut ini hasil dari tahap ekstraksi fitur dari sampel citra masukan.

- Cup to Disk Ratio = 0.6245
- ISNT Neuro Retinal Rim = 0.9987
- ISNT Pembuluh Darah = 0.9195

Setelah fitur didapat, maka kelas dari citra masukan akan dihitung dengan menggunakan svm model yang sudah didapatkan saat training data. Hasil klasifikasi dari citra yang diuji cobakan ini adalah kelas glaukoma.

5.4 Skenario Uji Coba

Sebelum melakukan uji coba, perlu ditentukan skenario yang akan digunakan dalam uji coba. Melalui skenario ini, perangkat akan diuji apakah sudah berjalan dengan benar dan bagaimana performa masing-masing skenario, serta membandingkan skenario manakah yang memiliki performa lebih baik. Terdapat dua skenario uji coba pada perangkat lunak ini, antara lain:

1. Penghitungan performa dengan dan tanpa menggunakan proses *smoothing* histogram pada proses segmentasi optic cup.
2. Penghitungan performa dengan mengubah metode pencarian *hyperplane* pada pengklasifikasi *support vector machine* pada proses klasifikasi. Metode yang digunakan antara lain *Sequential Minimal Optimization* (SMO), *Quadratic Programming* (QP), dan *Least Squares* (LS).
3. Penghitungan performa dengan mengubah fungsi kernel *support vector machine* pada proses klasifikasi. Fungsi kernel yang diujikan yaitu *linear, quadratic, polynomial orde 3*, dan RBF (*Gaussian Radial Basis Function*).

5.4.1 Skenario Uji Coba 1

Skenario uji coba 1 adalah penggunaan metode *smoothing* histogram pada proses segmentasi *optic cup*. Akan dibandingkan performa hasil klasifikasi dengan dan tanpa menggunakan *smoothing* histogram pada proses segmentasi *optic cup*. Tabel 5.1 dan 5.2 secara berturut-turut menampilkan hasil uji coba dari sistem yang menggunakan proses histogram *smoothing* dan tanpa histogram *smoothing* menggunakan 10 *folds cross validation* sebanyak 5 kali percobaan.

Tabel 5.1 Hasil akurasi uji coba skenario 1

Percobaan	Akurasi dengan Histogram Smoothing (%)	Akurasi dengan Histogram Smoothing (%)
1	79	77
2	79	73
3	81	77
4	82	73
5	79	74
Rata-rata	80	75

Tabel 5.2 Hasil presisi uji coba skenario 1

Percobaan	Presisi dengan Histogram Smoothing (%)	Presisi dengan Histogram Smoothing (%)
1	83	80
2	81	81
3	84	81
4	85	77
5	82	78
Rata-rata	83	79

Tabel 5.3 Hasil recall uji coba skenario 1

Percobaan	Recall dengan Histogram Smoothing (%)	Recall dengan Histogram Smoothing (%)
1	77	75
2	80	66
3	80	73
4	80	72
5	78	72
Rata-rata	79	72

Berdasarkan hasil yang telah didapatkan pada uji coba skenario 1, performa yang lebih bagus didapat ketika histogram *smoothing* digunakan pada proses segmentasi *optic cup* dengan akurasi sebesar 80%, *presisi* 83% dan *recall* 79%.

5.4.2 Skenario Uji Coba 2

Skenario uji coba 1 adalah penggunaan metode pencarian *hyperplane* yang berbeda pada pengklasifikasi *support vector machine*. Tabel 5.3, 5.4, dan 5.5 secara berturut-turut menampilkan hasil dari 10 *folds cross validation* sebanyak 5 kali uji coba dari metode *Sequential Minimal Optimization* (SMO), *Least Squares* (LS), dan *Quadratic Programming* (QP).

Tabel 5.4 Hasil akurasi uji coba skenario 2

Percobaan	Akurasi SMO (%)	Akurasi QP (%)	Akurasi LS (%)
1	79	80	78
2	79	79	80
3	81	80	80
4	82	78	78

Percobaan	Akurasi SMO (%)	Akurasi QP (%)	Akurasi LS (%)
5	79	81	79
Rata-rata	80	80	79

Tabel 5.5 Hasil presisi uji coba skenario 2

Percobaan	Presisi SMO (%)	Presisi QP (%)	Presisi LS (%)
1	83	83	83
2	81	83	84
3	84	84	84
4	85	83	83
5	82	85	83
Rata-rata	83	84	84

Tabel 5.6 Hasil recall uji coba skenario 2

Percobaan	Recall SMO (%)	Recall QP (%)	Recall LS (%)
1	77	78	75
2	80	77	77
3	80	77	77
4	80	75	75
5	78	78	77
Rata-rata	79	77	76

Berdasarkan hasil yang telah didapatkan pada uji coba skenario 2, didapat bahwa nilai akurasi tertinggi adalah 80 % dengan metode SMO dan QP. Metode yang akan digunakan oleh sistem adalah metode SMO karena memiliki nilai *recall* yang lebih baik dari QP.

5.4.3 Skenario Uji Coba 3

Skenario uji coba 2 adalah penggunaan fungsi kernel yang berbeda dengan menggunakan metode penghitungan *hyperplane*

SMO pada pengklasifikasi *support vector machine*. Tabel 5.6, 5.7, 5.8, dan 5.9 secara berturut-turut menampilkan hasil dari 10 *folds cross validation* sebanyak 5 kali uji coba dari fungsi kernel *linear, quadratic, polynomial orde 3*, dan RBF (*Gaussian Radial Basis Function*).

Tabel 5.7 Hasil akurasi uji coba skenario 3

Percobaan	Akurasi Linear (%)	Akurasi Quadratic (%)	Akurasi Polynomial (%)	Akurasi RBF (%)
1	79	78	72	76
2	79	77	76	79
3	81	78	76	80
4	82	76	72	78
5	79	75	73	78
Rata-rata	80	77	74	78

Tabel 5.8 Hasil presisi uji coba skenario 3

Percobaan	Presisi Linear (%)	Presisi Quadratic (%)	Presisi Polynomial (%)	Presisi RBF (%)
1	83	80	74	75
2	81	78	78	76
3	84	80	78	77
4	85	76	73	76
5	82	78	77	75
Rata-rata	83	79	76	76

Tabel 5.9 Hasil recall uji coba skenario 3

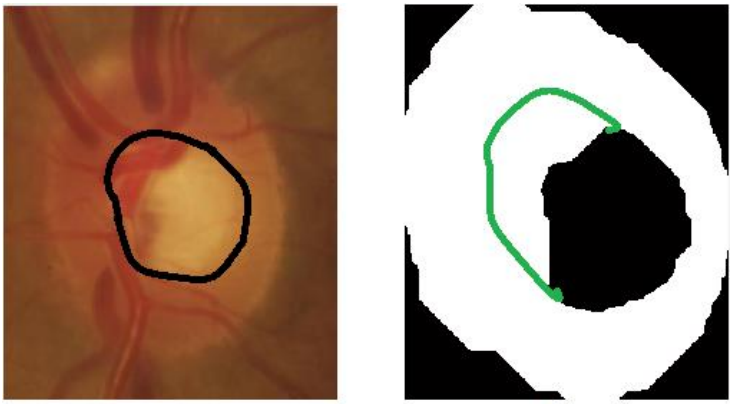
Percobaan	Recall Linear (%)	Recall Quadratic (%)	Recall Polynomial (%)	Recall RBF (%)
1	77	77	72	83

Percobaan	Recall Linear (%)	Recall Quadratic (%)	Recall Polynomial (%)	Recall RBF (%)
2	80	78	77	89
3	80	80	77	89
4	80	80	73	86
5	78	73	72	86
Rata-rata	79	78	74	87

Berdasarkan hasil yang telah didapatkan pada uji coba skenario 3, performa yang terbaik didapat ketika menggunakan kernel linear dengan akurasi sebesar 80%, *presisi* 83% dan *recall* 79%.. Fungsi kernel *linear* kemudian akan digunakan oleh sistem untuk proses klasifikasi karena memiliki tingkat akurasi yang paling tinggi.

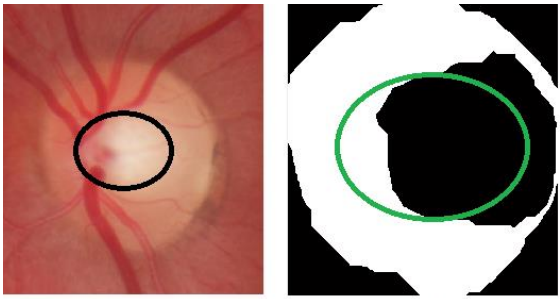
5.5 Evaluasi

Keberadaan pembuluh darah juga mempengaruhi hasil segmentasi *optic cup*. Pada *channel hijau* dari citra retina mata, pembuluh darah terlihat sangat jelas. Bahkan pada beberapa citra, pembuluh darah terlihat menutupi sebagian area dari *optic cup* sehingga bentuk *optic cup* yang tersegmentasi akan mengalami degradasi daripada yang semestinya. Gambar 5.8 menunjukkan contoh degradasi yang terjadi pada segmentasi *optic cup* yang terjadi karena keberadaan pembuluh darah.



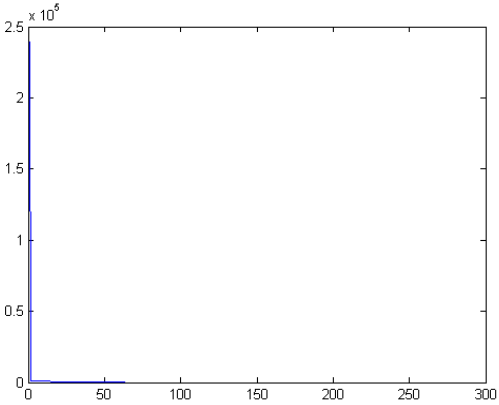
Gambar 5.8 Degradasi pada segmentasi *optic cup* (a) daerah *optic cup* yang seharusnya, (b) daerah *Neuro Retinal Rim* yang tersegmentasi.

Selain itu, adanya pantulan cahaya pada beberapa retina mata menyebabkan kesalahan segmentasi *optic cup* dan *optic disk*. Hal ini disebabkan karena daerah yang memantulkan cahaya terlihat lebih terang daripada yang seharusnya, sehingga daerah tersebut bisa terdeteksi sebagai *optic disk* ataupun *optic cup*. Dari dataset citra retina mata yang digunakan, kebanyakan pantulan cahaya lebih sering terjadi di daerah *optic* disk. Gambar 5.9 menunjukkan contoh kesalahan segmentasi *optic cup* yang terjadi karena adanya pantulan cahaya pada daerah *optic disk* yang terdeteksi sebagai *optic cup* dan keberadaan pembuluh darah.

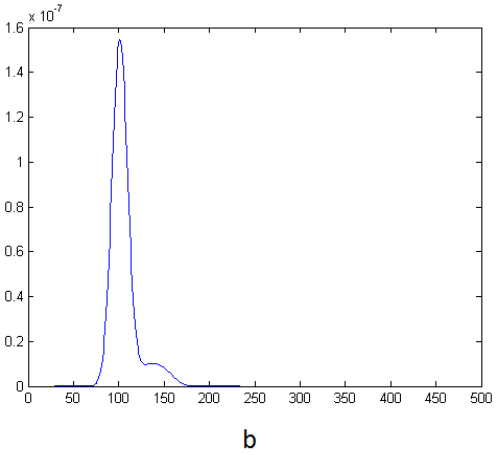


Gambar 5.9 Kesalahan pada segmentasi optic cup, (a) daerah optic cup yang seharusnya, (b) daerah *Neuro Retinal Rim* yang tersegmentasi.

Selain itu, hasil dari skenario uji coba 1 menunjukkan bahwa proses *histogram smoothing* pada segmentasi *optic cup* memberikan peningkatan performa yang cukup signifikan, yaitu sekitar 5%. Ini dapat dimaklumi karena dengan adanya *histogram smoothing*, penyebaran nilai piksel pada citra yang telah melalui tahap *preprocessing* menjadi lebih merata. Hal ini akan memudahkan dalam pencarian *threshold* pada metode otsu.



a



Gambar 5.10 Gambaran dari (a) histogram dari preprocessed image dan (b) smoothed histogram dari (a)

Hasil dari skenario uji coba 2 dan 3 menunjukkan metode dan fungsi kernel yang memberikan performa paling baik pada pengklasifikasi *support vector machine* adalah metode SMO dan fungsi kernel *linear*.

(Halaman ini sengaja dikosongkan)