



TUGAS AKHIR - KI141502

Deteksi Penyakit Glaukoma pada Citra Fundus Retina Mata Menggunakan Adaptive Thresholding dan Support Vector Machine

AHMAD MUSTOFA
NRP 51112100100

Dosen Pembimbing I
Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.

Dosen Pembimbing II
Bilqis Amaliah, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA



TUGAS AKHIR - KI141502

Deteksi Penyakit Glaukoma pada Citra Fundus Retina Mata Menggunakan Adaptive Thresholding dan Support Vector Machine

AHMAD MUSTOFA
NRP 51112100100

Dosen Pembimbing I
Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.

Dosen Pembimbing II
Bilqis Amaliah, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016

(Halaman ini sengaja dikosongkan)



FINAL PROJECT - KI141502

Deteksi Penyakit Glaukoma pada Citra Fundus Retina Mata Menggunakan Adaptive Thresholding dan Support Vector Machine

AHMAD MUSTOFA

NRP 51112100100

Supervisor I

Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.

Supervisor II

Bilqis Amaliah, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS

FACULTY OF INFORMATION TECHNOLOGY

SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY

SURABAYA 2016

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

**Deteksi Penyakit Glaukoma pada Citra Fundus Retina Mata
Menggunakan Adaptive Thresholding dan Support Vector
Machine**

TUGAS AKHIR

**Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada**

**Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi**

Institut Teknologi Sepuluh Nopember

Oleh:

AHMAD MUSTOFA

NRP: 5112 100 100

Disetujui oleh Pembimbing Tugas Akhir:

Prof. Ir. Handayani Tjandrasa, M.Sc.,
Ph.D. (pembimbing 1)

NIP: 19490823 197603 2 001

Bilqis Amaliah, S.Kom., M.Kom.

NIP: 19750914 200112 2 002 (pembimbing 2)

**SURABAYA
JUNI, 2016**

(Halaman ini sengaja dikosongkan)

DETEKSI PENYAKIT GLAUKOMA PADA CITRA FUNDUS RETINA MATA MENGGUNAKAN ADAPTIVE THRESHOLDING DAN SUPPORT VECTOR MACHINE

Nama : Ahmad Mustofa
NRP : 5112100100
Jurusan : Teknik Informatika, FTIf-ITS
Dosen Pembimbing I : Prof. Ir. Handayani Tjandrasa, M.Sc.,
Ph.D.
Dosen Pembimbing II : Bilqis Amaliah, S.Kom., M.Kom.

Abstrak

Glaukoma adalah penyebab kebutaan kedua terbanyak setelah katarak. Berbeda dengan katarak, kebutaan yang disebabkan oleh glaukoma bersifat permanen. Hal ini karena glaukoma disebabkan oleh tekanan besar pada bola mata yang menyebabkan tersumbatnya pembuluh darah yang menuju ke syaraf mata sehingga syaraf mata tidak mendapatkan suplai darah yang cukup dan akhirnya akan mengalami kerusakan. Gejala glaukoma yang timbul biasanya tidak dapat dirasakan secara langsung. Sehingga perlu dilakukan pemeriksaan mata terlebih dahulu untuk mengetahui adanya glaukoma.

Pada pengerjaan tugas akhir ini, dibangun sebuah perangkat lunak untuk mendeteksi penyakit glaukoma pada citra fundus retina mata. Tahap pertama dalam pengerjaan tugas akhir ini adalah proses preprocessing citra. Tahap preprocessing terbagi menjadi preprocessing optic cup, preprocessing optic disk, dan preprocessing pembuluh darah. Kemudian akan dilakukan proses segmentasi optic cup, optic disk, dan pembuluh darah dengan menggunakan metode adaptive thresholding. Setelah proses segmentasi selesai, maka fitur Cup to Disk Ratio (CDR), ISNT Neuro Retinal Rim (NRR), dan ISNT pembuluh darah akan diekstraksi dari masing-masing citra yang telah tersegmentasi. Ketiga fitur tersebut kemudian dijadikan masukan pada pengklasifikasi support vector machine dengan menggunakan metode pencarian hyperplane sequential minimal optimization dan

fungsi kernel linear. Dengan menggunakan data yang diambil dari database RIM-ONE, didapatkan nilai akurasi rata-rata sebesar 80%.

Kata kunci: glaukoma , optic cup, optic disk, segmentasi, klasifikasi

GLAUCOMA DETECTION ON RETINAL FUNDUS IMAGES USING ADAPTIVE THRESHOLDING AND SUPPORT VECTOR MACHINE

Name : Ahmad Mustofa
NRP : 5112100100
Department : Informatics, FTIf-ITS
SupervisorI : Prof. Ir. Handayani Tjandrasa, M.Sc.,
Ph.D.
SupervisorII : Bilqis Amaliah, S.Kom., M.Kom.

Abstract

Glaucoma is the second most common cause of blindness after cataract. Unlike the cataract, blindness caused by glaucoma is permanent. Since glaucoma is caused by great pressure on the eyeball that results in blockage of the blood vessels leading to the nerves of the eye such that the eye nerves do not get enough blood supply and eventually will be damaged. Symptoms of glaucoma that arise usually can not be felt directly. So we need an eye examination to detect glaucoma.

In this final project, a software to detect glaucoma on retinal fundus images are built. The first stage in this final project is image preprocessing. Preprocessing stage is divided into optic cuppreprocessing, optic diskpreprocessing, and blood vessels preprocessing. Then optic cup, optic disc and blood vesselssegmentation are performed using adaptive thresholding. After the segmentation process is complete, the Cup to Disk Ratio (CDR), ISNT Neuro Retinal Rim (NRR), and ISNT blood vessels features will be extracted from each image that has been segmented. These three features are then used as inputs in support vector machine classifier using sequential minimal optimization method to search hyperplane and linear kernel function. By using

the data drawn from the database RIM-ONE, the average of system's accuracy is 80%.

Keywords: glaucoma , optic cup, optic disk, segmentation, classification

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “*Deteksi Penyakit Glaukoma Pada Citra Fundus Retina Mata Menggunakan Adaptive Thresholding Dan Support Vector Machine*” dengan tepat waktu.

Dalam pelaksanaan dan pembuatan tugas akhir ini penulis mendapatkan banyak bantuan dari berbagai pihak. Pada kesempatan ini, ucapan terima kasih penulis berikan kepada:

1. Allah SWT, karena atas limpahan rahmat-Nya, penulis diberikan kemudahan dan kelancaran dalam mengerjakan tugas akhir ini.
2. Orang tua dan keluarga penulis yang senantiasa memberikan doa dan dukungan kepada penulis untuk menyelesaikan pengerjaan tugas akhir ini.
3. Ibu Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D. dan Ibu Bilqis Amaliah, S.Kom., M.Kom. sebagai dosen pembimbing yang telah memberikan banyak arahan dan bantuan sehingga penulis dapat menyelesaikan tugas akhir ini.
4. Bapak Imam Kuswardayan, S.Kom., M.T. selaku dosen wali yang telah memberikan bimbingan dan nasihat selama masa perkuliahan.
5. Teman-teman *user* TA dan Administrator Laboratorium Komputasi Cerdas dan Visi, yang telah memberikan bantuan dalam penyelesaian tugas akhir ini.
6. Semua pihak yang telah membantu terselesaikannya tugas akhir ini.

Penulis menyadari adanya banyak kekurangan dalam pengerjaan tugas akhir baik dari segi program maupun laporan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun untuk penyempurnaan tugas akhir ini. Akhir kata, penulis meminta maaf bila terdapat kesalahan dalam penulisan

laporan tugas akhir ini. Semoga hasil dari tugas akhir ini dapat memberikan manfaat bagi pembaca pada umumnya dan penulis pada khususnya.

Surabaya, Juni 2016

Ahmad Mustofa

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak	vii
<i>Abstract</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
1 BAB I	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metodologi	2
1.6 Sistematika Penulisan.....	3
2 BAB II.....	5
2.1 Glaukoma	5
2.2 <i>Optic Disc</i> dan <i>Optic Cup</i>	6
2.3 Aturan ISNT	6
2.4 <i>Cup to Disc Ratio</i>	7
2.5 Neuro Retinal Rim.....	8
2.6 <i>Histogram Smoothing</i>	8
2.7 Bottom Hat Transormation.....	8
2.8 Otsu Thresholding	8
2.9 Median Filtering.....	9
2.10 Morfologi Erosi dan Dilasi.....	10
2.11 <i>Support Vector Machine</i>	10
2.11.1 Sequential Minimum Optimization	13
2.11.2 Least Squares.....	13
2.11.3 Soft Margin.....	13

2.11.4	Kernel Trick	14
2.12	Confusion Matrix	16
2.13	<i>k-Fold Cross Validation</i>	17
3	BAB III	19
3.1	Perancangan Data	19
3.1.1	Data Masukan.....	19
3.1.2	Data Luaran	20
3.2	Desain Metode Secara Umum	21
3.3	Perancangan Proses	22
3.3.1	Tahap Preprocessing.....	23
3.3.2	Tahap Segmentasi.....	25
3.3.3	Tahap Ekstraksi Fitur	29
3.3.4	Tahap Klasifikasi.....	30
3.4	Perancangan Antarmuka Perangkat Lunak.....	31
4	BAB IV	35
4.1	Lingkungan Implementasi	35
4.2	Implementasi	35
4.2.1	Implementasi Preprocessing.....	35
4.2.2	Implementasi Segmentasi.....	37
4.2.3	Implementasi Ekstraksi Fitur.....	41
4.2.4	Implementasi Klasifikasi	44
4.2.5	Implementasi User Interface.....	45
5	BAB V	49
5.1	Lingkungan Uji Coba	49
5.2	Data Uji Coba.....	50
5.3	Uji Coba Sistem.....	50
5.4	Skenario Uji Coba	54
5.4.1	Skenario Uji Coba 1	54
5.4.2	Skenario Uji Coba 2	56
5.4.3	Skenario Uji Coba 3	58

5.5	Evaluasi	60
6	BAB VI.....	65
6.1	Kesimpulan.....	65
6.2	Saran.....	65
7	Daftar Pustaka.....	67
	BIODATA PENULIS	69

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 <i>Masks</i> yang digunakan untuk mendapatkan luasan di daerah a Superior; b Temporal; c Inferior; dan d Nasal dari optic disc	7
Gambar 2.2 Contoh alternatif hyperplane	11
Gambar 2.3 Fungsi Φ Memetakan Data ke Ruang Vektor Lebih Tinggi sehingga Kedua Kelas Dapat Dipisahkan Secara Linier oleh Sebuah <i>Hyperplane</i> [8].....	14
Gambar 3.1 Citra Retina Mata Glaukoma	20
Gambar 3.2 Contoh hasil segmentasi (a) optic cup, (b) optic disk, (c) pembuluh darah di daerah optic disc.....	21
Gambar 3.3 Diagram Alir Model Sistem	22
Gambar 3.4 Diagram Alir Preprocessing Optic Cup dan Optic Disk	23
Gambar 3.5 Diagram Blood Preprocessing	25
Gambar 3.6 Diagram Alur Segmentasi Optic Cup	26
Gambar 3.7 Diagram Alur Segmentasi Optic Disk	27
Gambar 3.8 Diagram Alur Segmentasi Pembuluh Darah.....	28
Gambar 3.9 Diagram Alur Neuro Retinal Rim	29
Gambar 3.10 Diagram Alur Klasifikasi.....	30
Gambar 3.11 Tatap muka utama	31
Gambar 3.12 Tatap muka setelah berkas selesai dipilih.....	32
Gambar 3.13 Tampilan akhir aplikasi	33
Gambar 5.1 Citra masukan sistem.....	50
Gambar 5.2 Hasil <i>preprocessing optic disk</i>	51
Gambar 5.3 Hasil <i>preprocessing optic cup</i>	51
Gambar 5.4 Hasil <i>preprocessing</i> pembuluh darah	52
Gambar 5.5 Hasil segmentasi <i>optic disk</i>	52
Gambar 5.6 Hasil segmentasi <i>optic cup</i>	53

Gambar 5.7 Hasil segmentasi pembuluh darah	53
Gambar 5.8 Degradasi pada segmentasi optic cup.....	61
Gambar 5.9 Kesalahan pada segmentasi optic cup.	62
Gambar 5.10 Gambaran dari (a) histogram dari preprocessed image dan (b) smoothed histogram dari (a)	62

DAFTAR TABEL

Tabel 2.1 Jenis <i>Kernel</i> yang Umum Digunakan dalam SVM	16
Tabel 2.2 <i>Confusion Matrix</i> Dua Kelas.....	17
Tabel 5.1 Hasil skenario 1 dengan Histogram Smoothing	55
Tabel 5.2 Hasil skenario 1 tanpa Histogram Smoothing	55
Tabel 5.3 Hasil skenario 2 dengan metode SMO	56
Tabel 5.4 Hasil skenario 2 dengan metode LS	57
Tabel 5.5 Hasil skenario 2 dengan metode QP.....	57
Tabel 5.6 Hasil skenario 3 dengan fungsi kernel <i>linear</i>	58
Tabel 5.7 Hasil skenario 3 dengan fungsi kernel <i>quadratic</i>	59
Tabel 5.8 Hasil skenario 3 dengan fungsi kernel <i>polynomial orde 3</i>	59
Tabel 5.9 Hasil skenario 3 dengan fungsi kernel RBF	60

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1 Latar Belakang

Glaukoma adalah penyebab kebutaan kedua terbanyak setelah katarak di seluruh dunia. Berbeda dengan katarak, kebutaan yang disebabkan oleh glaukoma bersifat permanen. Berdasarkan data WHO 2010, diperkirakan sebanyak 3.2 juta orang mengalami kebutaan akibat glaukoma [1]. Disamping itu, perkembangan teknologi informasi dan komunikasi yang pesat harusnya bisa dimanfaatkan untuk mempermudah tenaga kesehatan untuk melaksanakan tugasnya. Manfaat itulah yang diharapkan dari tugas akhir ini, yaitu mempermudah tenaga kesehatan untuk mendeteksi penyakit glaukoma dari citra retina mata pasien.

Dalam tugas akhir ini, fitur yang akan digunakan untuk deteksi glaukoma adalah CDR (Cup to Disc Ratio), luas NRR (Neuro Retinal Rim), dan luas pembuluh darah di daerah optic disc dalam ISNT [2]. Metode yang akan digunakan untuk segmentasi optic disc dan optic cup adalah adaptive thresholding, sedangkan untuk klasifikasi digunakan metode support vector machine.

Hasil yang diharapkan dari tugas akhir ini adalah sebuah aplikasi yang mampu mengklasifikasi apakah masukan yang berupa citra retina mata menderita glaukoma atau tidak. Aplikasi ini juga diharapkan dapat mempermudah tenaga medis dalam mendiagnosis penyakit glaukoma pada pasien.

1.2 Rumusan Masalah

Berikut beberapa hal yang menjadi rumusan masalah dalam tugas akhir ini :

1. Bagaimana melakukan segmentasi optic cup, optic disc, dan pembuluh darah dari citra retina mata?
2. Bagaimana sistem dapat mendeteksi penderita glaukoma dari citra retina mata?

1.3 Batasan Masalah

Berikut beberapa hal yang menjadi batasan masalah dalam pengerjaan tugas akhir ini:

1. Kelas retina mata yang dapat diklasifikasi adalah kelas mata normal dan kelas mata glaukoma.
2. Implementasi program dilakukan pada lingkungan komputer desktop dengan menggunakan matlab.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut.

1. Merancang dan membuat aplikasi yang bisa melakukan segmentasi optic cup, optic disc, dan pembuluh darah dari citra retina mata.
2. Merancang dan membuat aplikasi yang bisa mendeteksi penderita glaukoma dari citra retina mata.

1.5 Metodologi

Tahap yang dilakukan untuk menyelesaikan tugas akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Penulisan proposal ini merupakan tahap awal dalam pengerjaan tugas akhir. Pada proposal ini, penulis mengajukan gagasan pendeteksian penyakit glaukoma pada citra fundus retina mata menggunakan metode *adaptive thresholding* dan *support vector machine*.

2. Studi Literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur sejumlah referensi tentang deteksi penderita glaukoma dengan masukan citra retina mata. Informasi dan studi literatur tersebut didapat dari buku, internet, dan materi-materi kuliah yang berhubungan dengan metode yang digunakan.

3. Implementasi

Pada tahap ini dibangun perangkat lunak sesuai dengan rancangan yang diajukan pada proposal. Pembangunan perangkat lunak diimplementasikan sesuai dengan konsep yang telah didapatkan saat studi literatur.

4. Pengujian dan Evaluasi

Pada tahapan ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan juga untuk mengevaluasi jalannya sistem, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

5. Penyusunan Buku Tugas Akhir

Tahap ini merupakan tahap dokumentasi dari tugas akhir. Buku tugas akhir berisi dasar teori, perancangan, implementasi, serta hasil uji coba dan evaluasi dari aplikasi yang dibangun.

1.6 Sistematika Penulisan

Buku tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

1. Bab I. Pendahuluan

Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan tugas akhir.

2. Bab II. Tinjauan Pustaka

Bab tinjauan pustakan berisi penjelasan mengenai dasar teori yang mendukung pengerjaan tugas akhir.

3. Bab III. Analisis dan Perancangan

Bab analisis dan perancangan berisi penjelasan mengenai analisis kebutuhan, perancangan sistem dan perangkat yang digunakan dalam pengerjaan tugas akhir serta urutan pelaksanaan proses.

4. Bab IV. Implementasi

Bab implementasi berisi pembangunan aplikasi pendeteksi penyakit glaukoma dari citra retina mata menggunakan matlab.

5. Bab V. Uji Coba dan Evaluasi

Bab ini berisi hasil evaluasi aplikasi dengan menggunakan aplikasi yang dibangun. Juga disertakan analisis dari hasil evaluasi perangkat lunak.

6. Bab VI. Kesimpulan dan Saran

Bab kesimpulan dan saran berisi kesimpulan hasil penelitian. Selain itu, bagian ini berisi saran untuk pengerjaan lebih lanjut atau permasalahan yang dialami dalam proses pengerjaan tugas akhir.

BAB II

TINJAUAN PUSTAKA

Bab tinjauan pustaka berisi penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Glaukoma

Glaukoma merupakan penyebab kebutaan kedua terbanyak setelah katarak di seluruh dunia. Berbeda dengan katarak, kebutaan yang diakibatkan glaukoma bersifat permanen, atau tidak dapat disembuhkan. Hal ini menjadi tantangan tersendiri dalam upaya pencegahan dan penanganan kasus glaukoma. Berdasarkan data WHO 2010, diperkirakan sebanyak 3.2 juta orang mengalami kebutaan akibat glaukoma.

Glaukoma adalah penyakit mata dimana terjadi kerusakan saraf optik yang diikuti gangguan pada lapang pandangan yang khas. Kondisi ini utamanya diakibatkan oleh tekanan bola mata yang tinggi yang biasanya disebabkan oleh hambatan pengeluaran cairan bola mata (humour aqueous). Penyebab lain kerusakan saraf optic, antara lain gangguan suplai darah ke saraf optic dan kelemahan /masalah saraf optiknya sendiri.

Glaukoma dapat diklasifikasikan menjadi glaukoma primer, glaukoma sekunder, dan glaukoma kongenital. Glaukoma primer adalah glaukoma yang tidak diketahui penyebabnya. Glaukoma primer sudut terbuka biasanya merupakan glaukoma kronis, sedangkan glaukoma primer sudut tertutup bisa berupa glaukoma sudut tertutup akut atau kronis. Glaukoma sekunder adalah glaukoma yang timbul sebagai akibat dari penyakit mata lain, trauma, pembedahan, penggunaan kortikosteroid yang berlebihan atau penyakit sistemik lainnya. Glaukoma kongenital adalah

glaukoma yang ditemukan sejak dilahirkan dan biasanya disebabkan oleh sistem saluran pembuangan di dalam mata tidak berfungsi dengan baik sehingga menyebabkan pembesaran mata bayi. Di samping itu glaukoma dengan kebutaan total disebut juga sebagai glaukoma absolut. [1]

2.2 *Optic Disc dan Optic Cup*

Optic disc didefinisikan sebagai bagian dari saraf optik yang membentang dari bagian mielin saraf tepat dibelakang sklera sampai permukaan retina. Umumnya berbentuk sedikit oval dengan diameter vertikal sekitar 9% lebih besar dari diameter horizontal. Rata – rata pada orang normal, diameter vertikalnya mendekati 1500 mikrometer [2].

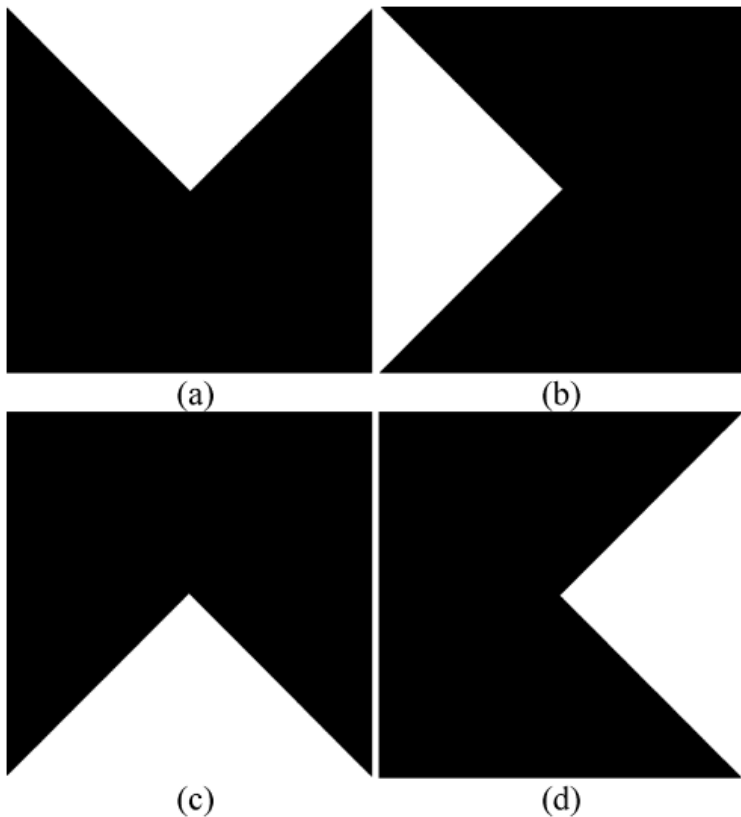
Sedangkan *optic cup* adalah sebuah lekukan berbentuk mangkok yang dibentuk oleh invaginasi dari gelembung optik yang berkembang menjadi lapisan sensorik dan berpigmen dari retina [3].

Pada umumnya *optic disc* merupakan daerah yang paling terang dalam citra retina, sedangkan *optic cup* merupakan daerah paling terang di daerah *optic disc*. [paper utama]

2.3 **Aturan ISNT**

ISNT adalah sebuah aturan yang menitikberatkan pada 4 daerah pada optic disc, yaitu Inferior, Superior, Nasal, dan Temporal. Keempat daerah inilah yang memiliki peran dalam membedakan mata yang normal dan yang tidak [4], baik itu jika dinilai dari keberadaan pembuluh darah maupun optic cup di masing-masing daerah [5]. Berikut adalah persamaan untuk menghitung nilai ISNT. [6]

$$ISNT = \frac{Luas\ pada\ quadran\ inferior + Luas\ pada\ quadran\ superior}{Luas\ pada\ quadran\ nassal + Luas\ pada\ quadran\ temporal} \quad (2.1)$$



Gambar 2.1 *Masks* yang digunakan untuk mendapatkan luasan di daerah a Superior; b Temporal; c Inferior; dan d Nasal dari optic disc

2.4 *Cup to Disc Ratio*

Cup to Disc Ratio (CDR) adalah perbandingan dari *optic cup* dan *optic disc* baik dari segi diameter vertikal, diameter horizontal, ataupun luas. CDR adalah salah satu fitur yang populer digunakan dalam pengevaluasian glaukoma [5]. Dalam tugas akhir ini, CDR yang digunakan adalah rasio jejari dari *optic*

cup dan optic disk yang didapatkan dengan rumus sebagai berikut.

$$CDR = \sqrt{\text{Area of optic cup} / \text{Area of optic disk}} \quad (2.2)$$

2.5 Neuro Retinal Rim

Neuro Retinal Rim (NRR) adalah daerah yang tersisa jika daerah optic cup dihapus dari optic disc. Bentuk dari NRR dapat digunakan untuk mengukur kerusakan pada optic disc [5].

2.6 Histogram Smoothing

Histogram smoothing adalah sebuah metode untuk memperhalus histogram dari sebuah image dengan filter tertentu. Dalam tugas akhir ini filter yang digunakan adalah filter gaussian dengan rerata 50, standar deviasi 6, dan ukuran window 1x100.

2.7 Bottom Hat Transformation

Bottom hat transformation adalah sebuah operasi morfologi yang dapat digunakan untuk mendapatkan objek yang gelap pada background yang terang. Berikut persamaan dari *bottom hat transformation*.

$$B_{hat}(I) = ((I \oplus b) \ominus b) - I \quad (2.3)$$

dimana \oplus adalah operasi dilasi dan \ominus adalah operasi erosi. [6]

2.8 Otsu Thresholding

Dalam *image processing*, metode otsu digunakan untuk melakukan *thresholding* berbasis *clustering* sehingga didapatkan batas yang dapat dijadikan acuan untuk melakukan segmentasi pada suatu citra. Dalam metode ini jumlah kelas yang dapat ditangani hanya ada 2, yaitu objek dan latar belakang. Konsep utama dari metode ini adalah dengan memilih nilai batas yang menghasilkan kelas dengan varian intra kelasnya terkecil dan

varian antar kelasnya terbesar. Rumus untuk menghitung varian antar kelas adalah sebagai berikut. [7]

$$\sigma_b^2(t) = \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2 \quad (2.4)$$

dimana $\omega_0(t) = \sum_0^{t-1} p(i) \quad (2.5)$ dan $\omega_1(t) = \sum_t^{255} p(i) \quad (2.6)$

serta $\mu_0(t) = \frac{\sum_0^{t-1} ip(i)}{\omega_0} \quad (2.7)$ dan $\mu_1(t) = \frac{\sum_t^{255} ip(i)}{\omega_1} \quad (2.8)$

dengan $p(i) = \frac{\text{histogram}(i)}{\text{jumlah pixel}} \quad (2.9)$

Berikut ini algoritma dari metode otsu.

- Hitung histogram dan peluang dari setiap level intensitas
- Inisialisasi nilai awal $\omega_i(0)$ dan $\mu_i(0)$
- For t = 1 to 256
 - Update ω_i dan μ_i
 - Hitung $\sigma_b^2(t)$
- Pilih t dengan nilai $\sigma_b^2(t)$ maksimum

2.9 Median Filtering

Median filtering adalah sebuah metode *filtering* yang pada umumnya digunakan untuk menghilangkan *noise*. Berikut pseudocode dari median filtering dua dimensi.

1	MedianFiltering(I, size)
2	foreach p = pixels in I
3	p = median(part(I(position(p).x - size, position(p).y + size))
4	end foreach
5	return I

2.10 Morfologi Erosi dan Dilasi

Operasi morfologi erosi dan dilasi adalah metode yang biasa dipakai dalam image processing dengan tujuan – tujuan tertentu. Dalam tugas akhir ini, operasi morfologi erosi dan dilasi digunakan untuk menutup gap yang dihasilkan oleh pembuluh darah pada saat melakukan segmentasi optic disc dan optic cup.

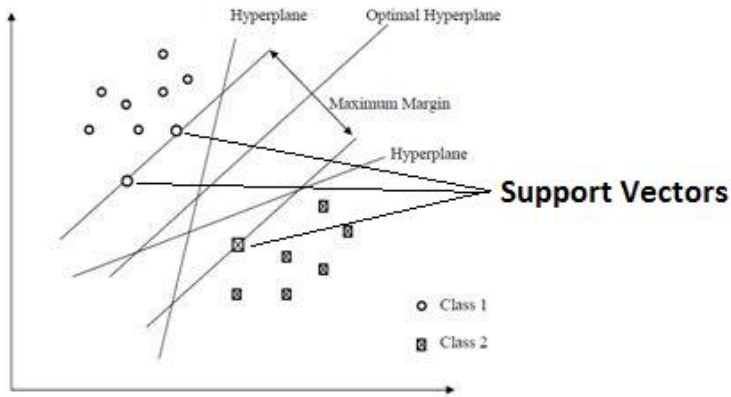
Operasi morfologi erosi dan dilasi membutuhkan elemen struktur yang berbeda-beda sesuai kebutuhan. Elemen struktur yang digunakan pada tugas akhir ini adalah elemen struktur berbentuk disk dengan ukuran 50.

2.11 Support Vector Machine

Support Vector Machine (SVM) dikembangkan oleh oser, Guyon, Vapnik dan pertama kali dipresentasikan pada tahun 1992. Prinsip dasar SVM adalah *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada *problem non-linear* dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi.

Konsep dasar SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua kelas pada *input space*. Gambar 2.2 merupakan beberapa contoh dari kemungkinan *hyperplane* yang digunakan untuk memisahkan kelas satu dengan lainnya [8].

Hyperplane pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur *margin hyperplane* dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* dengan data terdekat dari masing-masing kelas. Data yang paling dekat dengan *hyperplane* disebut sebagai *support vectors*.



Sumber gambar : www.sine.ni.com

Gambar 2.2 Contoh alternatif hyperplane

Diberikan data masukan $x_i \in \mathbb{R}^d$ dan masing-masing kelas dinotasikan $y_i \in \{-1, +1\}$ untuk $i = 1, 2, 3, \dots, n$ dimana n adalah banyaknya data. Fungsi *hyperplane* dibuat dengan persamaan

$$w \cdot x + b = 0 \quad (2.10)$$

Data x_i yang termasuk kelas -1 dapat dirumuskan sebagai data yang memenuhi pertidaksamaan

$$w \cdot x_i + b \leq -1 \quad (2.11)$$

sedangkan data x_i yang termasuk kelas +1 dirumuskan sebagai data yang memenuhi pertidaksamaan

$$w \cdot x_i + b \geq +1 \quad (2.12)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya dengan rumus

$$\frac{1}{||w||^2} \quad (2.13)$$

Hal tersebut dapat diselesaikan dengan mencari titik minimal persamaan 2.14 dengan memperhatikan *constraint* 2.15.

$$\min \tau(w) = \frac{1}{2} ||w||^2 \quad (2.14)$$

$$y_i(x_i \cdot w + b) - 1 \geq 0, \forall i \quad (2.15)$$

Masalah ini dapat dipecahkan dengan berbagai teknik komputasi di antaranya *Lagrange Multiplier*.

$$L(w, b, \alpha) = \frac{1}{2} ||w||^2 - \sum_{i=1}^n \alpha_i (y_i ((x_i \cdot w + b) - 1)) \quad (2.16)$$

dimana α adalah *Lagrange multipliers*, yang bernilai nol atau positif ($\alpha_i \geq 0$). Nilai optimal dari persamaan (2.16) dapat dihitung dengan meminimalkan L terhadap w dan b , dan memaksimalkan L terhadap α_i . Dengan memperhatikan bahwa pada titik optimal $L=0$, persamaan (2.16) dapat dimodifikasi sebagai maksimisasi *problem* yang hanya mengandung α_i saja, sebagaimana persamaan (2.17) dan (2.18)

maximize:

$$\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i x_j \quad (2.17)$$

Subject to:

$$\alpha_i \geq 0 \ (i = 1, 2, \dots, n) \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.18)$$

Hasil dari perhitungan ini diperoleh α_i yang kebanyakan bernilai positif. Data yang berkorelasi dengan α_i yang positif inilah yang disebut sebagai *support vector*.

2.11.1 Sequential Minimum Optimization

Sequential Minimum Optimization adalah sebuah metode dalam komputasi *Quadratic Programming* pada SVM dengan memecah komputasi *Quadratic Programming* menjadi beberapa sub kecil. Sehingga kompleksitas komputasi *Quadratic Programming* pada SVM dapat direduksi. [10]

2.11.2 Least Squares

Salah satu variasi metode untuk menyelesaikan komputasi *Quadratic Programming* pada SVM adalah dengan menggunakan metode *least squares*. Dimana persamaan 2.16 diubah menjadi persamaan berikut.

$$L(w, b, e, \alpha) = \frac{1}{2} ||w||^2 - \sum_{i=1}^n \alpha_i (y_i ((x_i \cdot w + b) - 1 + e_i)) \quad (2.19)$$

dimana e_i adalah nilai *least squares* dari y_i, x_i . [11]

2.11.3 Soft Margin

Pada umumnya, dua buah kelas pada *input space* tidak dapat dipisahkan secara sempurna. Hal ini menyebabkan *constraint* pada persamaan (2.15) tidak dapat terpenuhi, sehingga optimasi tidak dapat dilakukan. Untuk mengatasi masalah ini, SVM dirumuskan ulang dengan memperkenalkan teknik *soft margin*. Dalam *soft margin*, persamaan (2.15) dimodifikasi dengan memasukkan variabel *slack* ξ_i ($\xi_i > 0$) sehingga menjadi persamaan berikut.

$$y_i(x_i \cdot w + b) \geq 1 - \xi_i, \quad \forall i \quad (2.20)$$

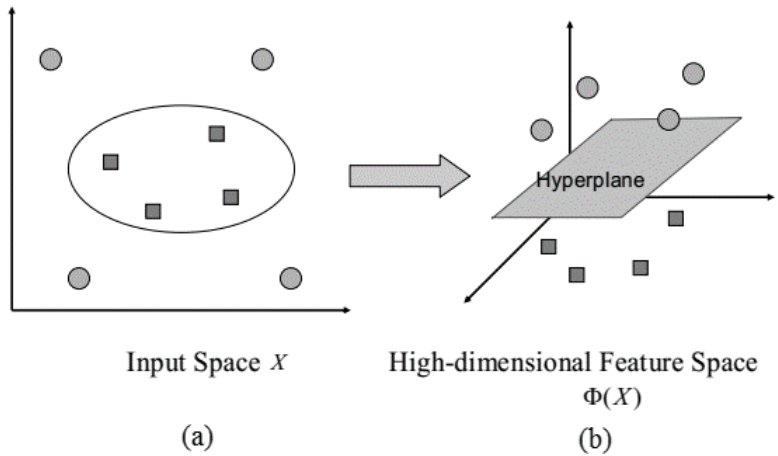
Dengan demikian persamaan (2.14) diubah menjadi persamaan berikut.

$$\min \tau(w, \xi) = \frac{1}{2} ||w||^2 + C \sum_{i=1}^n \xi_i \quad (2.21)$$

Parameter C dipilih untuk mengontrol *tradeoff* antara *margin* dan *error* klasifikasi ξ . Nilai C yang besar akan memberikan *penalty* yang lebih besar terhadap *error* klasifikasi [8].

2.11.4 Kernel Trick

Pada umumnya masalah dalam dunia nyata jarang yang bersifat *linear separable*, kebanyakan bersifat *non-linear*. Untuk menyelesaikan masalah *non-linear*, SVM dimodifikasi dengan memasukkan fungsi *kernel* [8].



Gambar 2.3 Fungsi Φ Memetakan Data ke Ruang Vektor Lebih Tinggi sehingga Kedua Kelas Dapat Dipisahkan Secara Linier oleh Sebuah *Hyperplane* [8].

Dalam *non-linear SVM*, pertama-tama data x dipetakan oleh fungsi $\Phi(x)$ ke ruang vektor yang berdimensi lebih tinggi. Pada ruang vektor yang baru ini, *hyperplane* yang memisahkan kedua kelas tersebut dapat dikonstruksikan. Pada Gambar 2.3 (a) diperlihatkan data pada kelas +1 yang direpresentasikan dalam bentuk lingkaran dan kelas -1 yang direpresentasikan dalam bentuk persegi berada pada *input space* berdimensi dua tidak dapat dipisahkan secara linier. Selanjutnya Gambar 2.3 (b) menunjukkan bahwa fungsi Φ memetakan tiap data pada *input space* tersebut ke ruang vektor baru yang berdimensi lebih tinggi (dimensi 3), dimana kedua kelas dapat dipisahkan secara linier oleh sebuah *hyperplane*. Notasi matematika dari pemetaan ini adalah sebagai berikut.

$$\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^q \text{ dengan } d < q \quad (2.22)$$

Pemetaan ini dilakukan dengan menjaga topologi data, dalam artian dua data yang berjarak dekat pada *input space* akan berjarak dekat juga pada *feature space* begitupun data yang berjarak jauh pada *input space* akan berjarak jauh pada *feature space*.

Selanjutnya proses pembelajaran pada SVM dalam menemukan titik-titik *support vector*, hanya bergantung pada *dot product* dari data yang sudah ditransformasikan pada ruang baru yang berdimensi lebih tinggi yaitu $\Phi(x_i) \cdot \Phi(x_j)$.

Karena umumnya transformasi Φ ini tidak diketahui, dan sangat sulit untuk dipahami, maka perhitungan *dot product* tersebut dapat digantikan dengan fungsi *kernel* $K(x_i, x_j)$ yang mendefinisikan secara implisit transformasi Φ . Hal ini disebut sebagai *kernel trick* yang dirumuskan berdasarkan persamaan berikut.

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (2.23)$$

Kernel trick memberikan berbagai kemudahan, karena dalam proses pembelajaran SVM, untuk menentukan *support vector*, kita hanya cukup mengetahui fungsi *kernel* yang dipakai, dan tidak perlu mengetahui wujud dari fungsi *non-linear* Φ . Jenis-jenis *kernel* yang umum digunakan dalam SVM dapat dilihat pada Tabel 2.1.

Tabel 2.1 Jenis *Kernel* yang Umum Digunakan dalam SVM

Jenis kernel	Definisi
Polynomial	$K(x_i, x_j) = (x_i \cdot x_j + 1)^p$
Gaussian	$K(x_i, x_j) = \exp\left(-\frac{\ x_i - x_j\ ^2}{2\gamma^2}\right)$

Selanjutnya hasil klasifikasi dari data x diperoleh dari persamaan berikut.

$$f(\Phi(x)) = w \cdot \Phi(x) + b \quad (2.24)$$

$$f(\Phi(x)) = \sum_{i=1, x_i \in SV}^n \alpha_i y_i \Phi(x) \cdot \Phi(x_i) + b \quad (2.25)$$

$$f(\Phi(x)) = \sum_{i=1, x_i \in SV}^n \alpha_i y_i K(x, x_i) + b \quad (2.26)$$

SV pada persamaan (2.25) dan (2.26) dimaksudkan dengan subset dari data *training* yang terpilih sebagai *support vectors*, dengan kata lain data x_i yang berkorespondensi pada $\alpha_i \geq 0$.

2.12 Confusion Matrix

Confusion matrix adalah suatu metode yang biasa digunakan untuk melakukan penghitungan performa suatu

algoritma pada konsep data *mining*. *Confusion matrix* memiliki informasi hasil prediksi dan aktual pada data yang telah diklasifikasi. Tabel 2.2 menunjukkan *confusion matrix* untuk dataset yang memiliki dua kelas.

Tabel 2.2 *Confusion Matrix* Dua Kelas

<div style="display: inline-block; transform: rotate(-45deg);"> Prediksi Aktual </div>	Prediksi	
	1	0
1	TP	FN
0	FP	TN

Nilai performa yang bisa dihitung menggunakan *confusion matrix* antara lain: akurasi, sensitivitas, dan presisi. Akurasi adalah hasil bagi dari jumlah prediksi yang terklasifikasi secara benar dibagi total data yang diklasifikasi seperti pada persamaan berikut.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.27)$$

Presisi adalah perbandingan dari jumlah data TP dengan total data TP dan FP seperti pada persamaan berikut.

$$Presisi = \frac{TP}{TP + FP} \quad (2.28)$$

Sensitivitas atau *recall* adalah perbandingan dari jumlah data TP dengan total data TP dan FN seperti pada persamaan berikut.

$$Sensitivitas \text{ atau } Recall = \frac{TP}{TP + FN} \quad (2.29)$$

2.13 *k-Fold Cross Validation*

Cross Validation adalah sebuah metode evaluasi dimana data yang digunakan untuk *testing* berbeda dengan data yang digunakan

untuk *training*. *k-Fold Cross Validation* adalah variasi dari metode *cross validation* dimana keseluruhan data yang ada dipisah menjadi k *subset*. Masing-masing *subset* kemudian secara bergantian digunakan sebagai data *testing* dengan $k-1$ *subset* yang lain digunakan sebagai data *training*. Performa akhir didapat dari rata-rata performa dari semua k percobaan. [9]

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

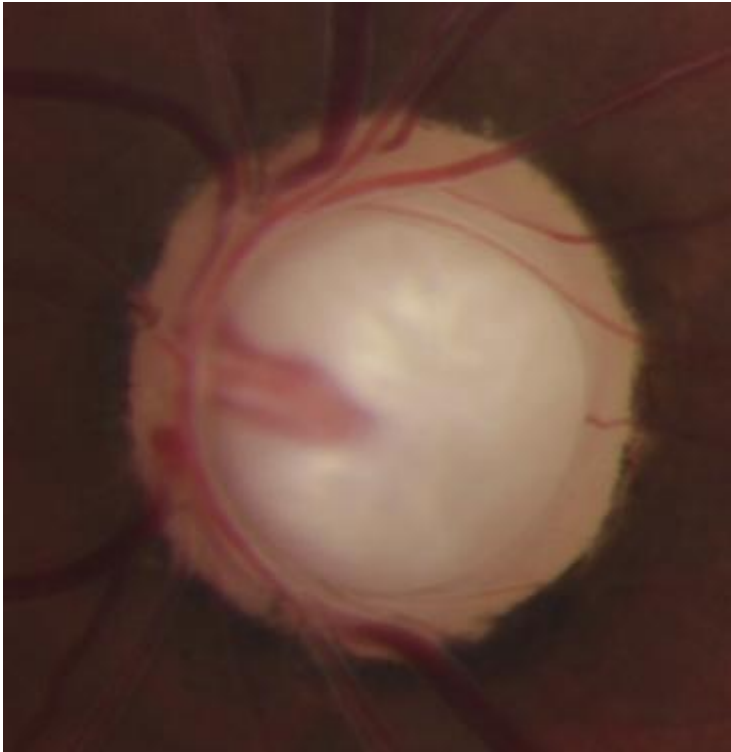
Perancangan data merupakan bagian yang terpenting dalam pengoperasian perangkat lunak karena diperlukan data yang tepat agar perangkat lunak dapat beroperasi dengan benar. Pada bagian ini akan dijelaskan mengenai perancangan data yang dibutuhkan untuk membangun perangkat lunak pendeteksi penyakit glaukoma 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

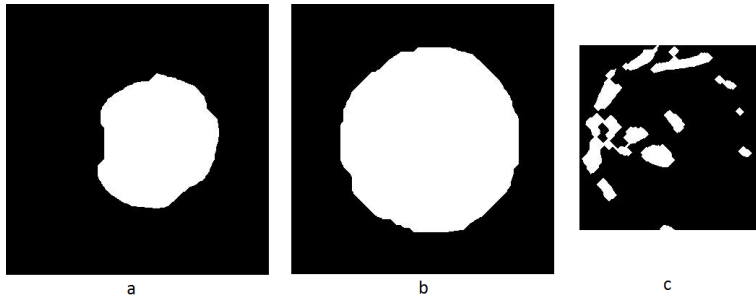
dari dua macam, yaitu normal dan glaukoma. Berikut contoh citra inputan yang digunakan.



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.



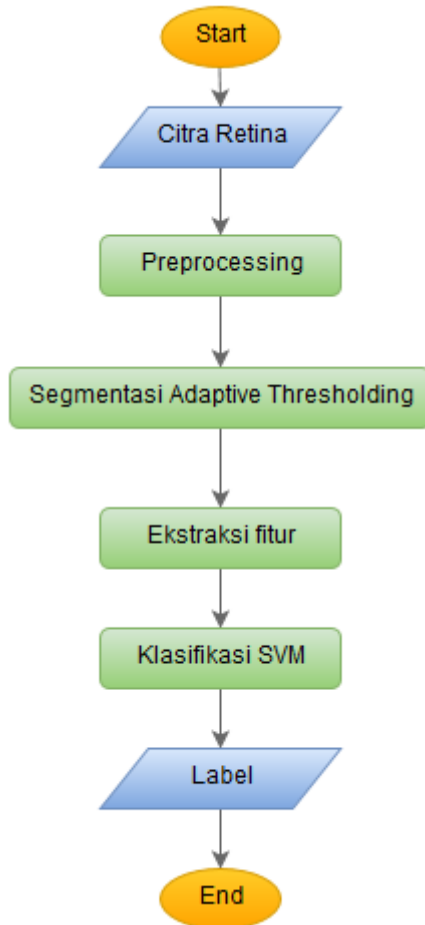
Gambar 3.2 Contoh hasil segmentasi (a) optic cup, (b) optic disk, (c) pembuluh darah di daerah optic disc

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 oleh 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

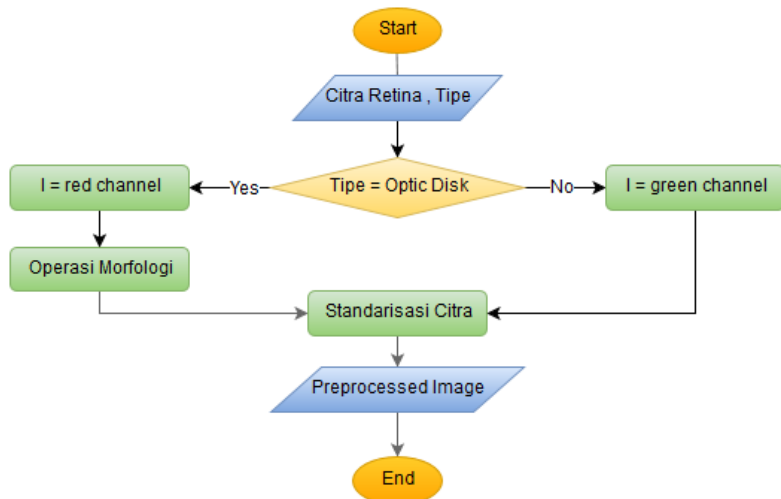
Berikut ini adalah rancangan dari sistem pendeteksi penyakit glaukoma dari citra fundus retina mata:

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

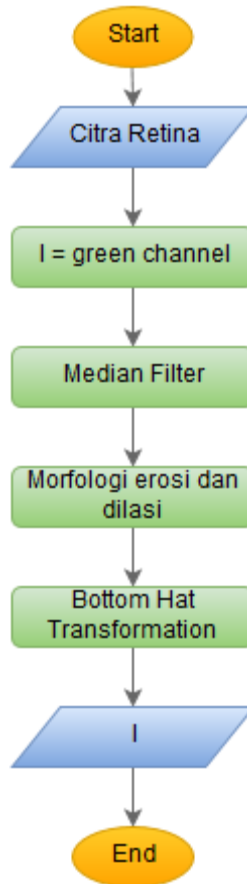
Pada tahap ini, dilakukan proses standarisasi citra dengan pseudocode sebagai berikut.

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 dara 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.



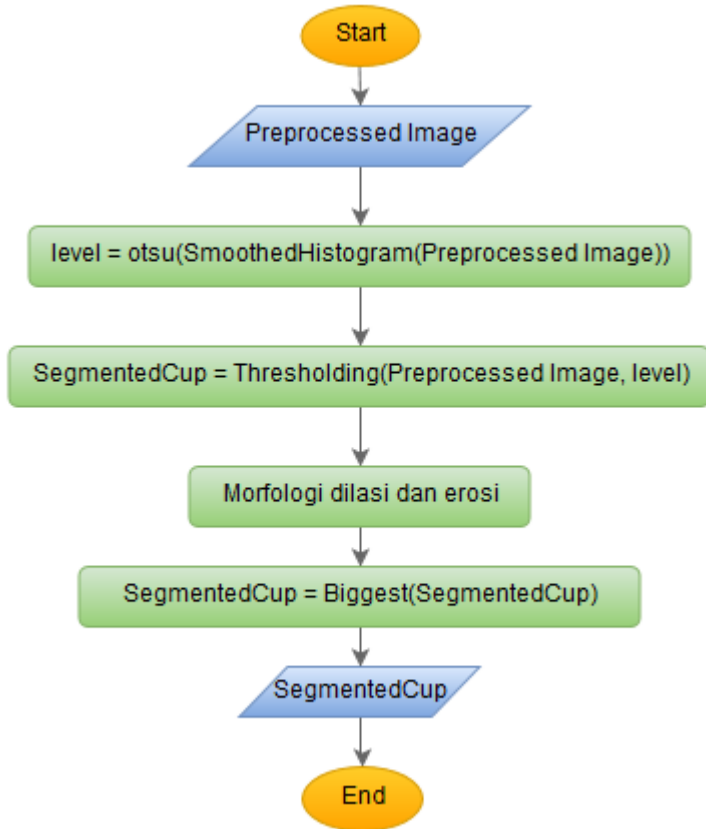
Gambar 3.5 Diagram Blood Preprocessing

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

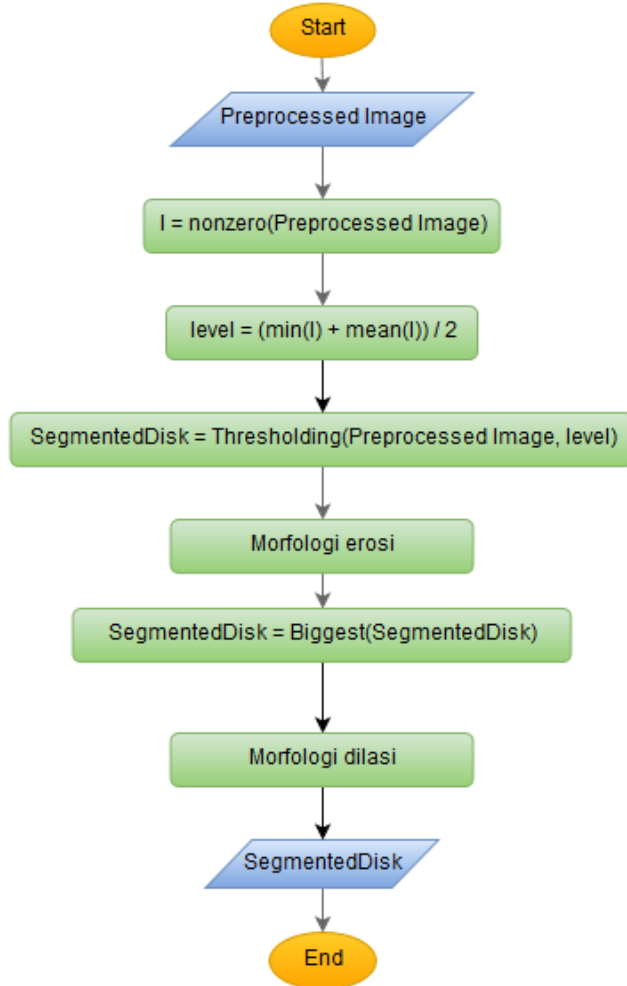
Pada tahap ini, dilakukan proses segmentasi optic cup dari citra yang telah melalui tahap preprocessing. Berikut diagram alurnya.



Gambar 3.6 Diagram Alur Segmentasi Optic Cup

3.3.2.2 Segmentasi Optic Disk

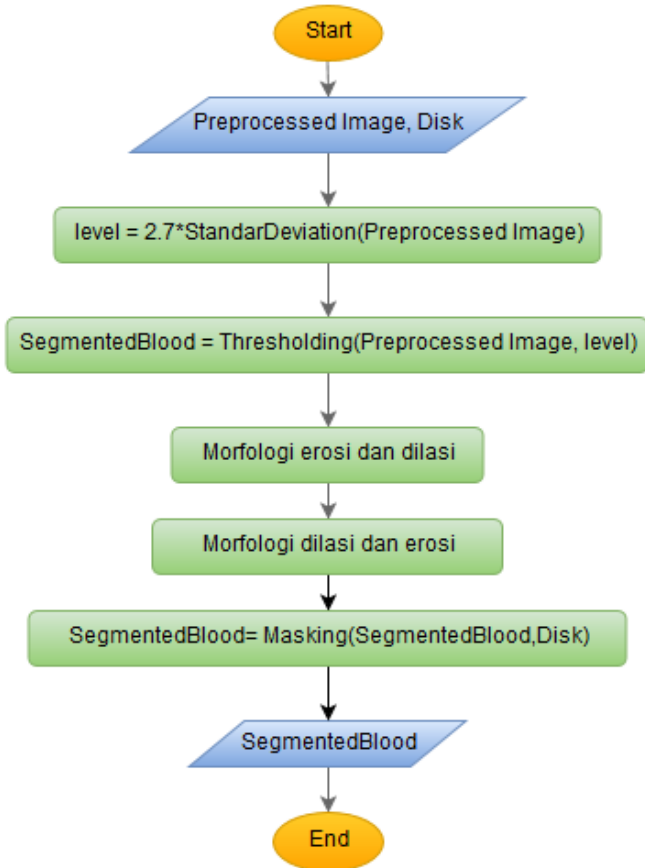
Pada tahap ini, dilakukan proses segmentasi optic disk dari citra yang telah melalui tahap preprocessing. Berikut diagram alurnya.



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

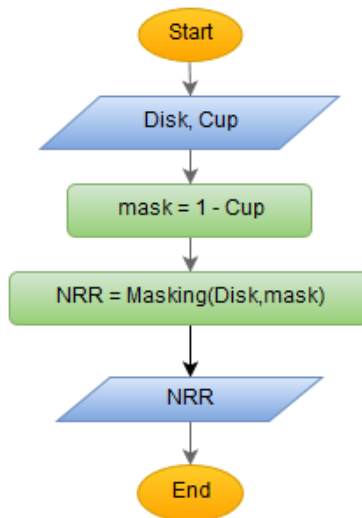
Pada tahap ini akan dilakukan proses ekstraksi fitur yang akan dipakai pada tahap klasifikasi. Fitur-fitur yang akan dipakai adalah *cup to disk ratio*, *ISNT neuro retinal rim*, *ISNT pembuluh darah*.

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

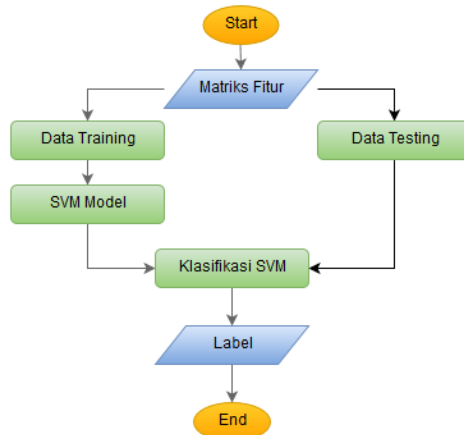
Setelah NRR didapatkan, maka penghitungan ISNT dilakukan dengan menggunakan persamaan 2.1.

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

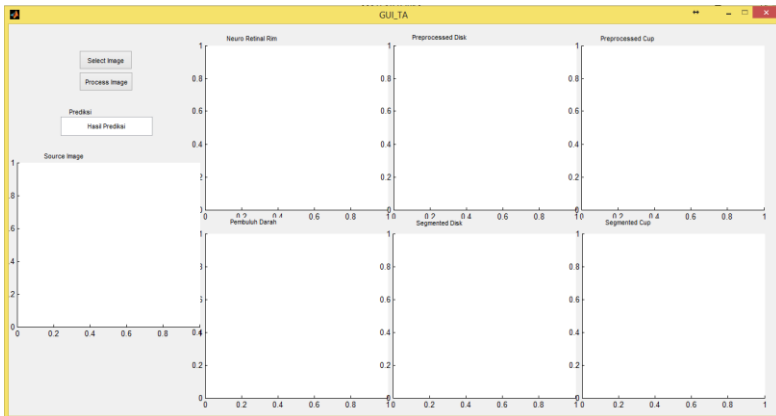
Fitur CDR, ISNT NRR, dan ISNT pembuluh darah hasil dari tahap ekstraksi fitur menjadi masukan pada tahap klasifikasi. Sebelum dilakukan proses klasifikasi, ketiga fitur tersebut dibagikan menjadi data training dan data testing. Data training adalah data yang digunakan untuk membuat model klasifikasi 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

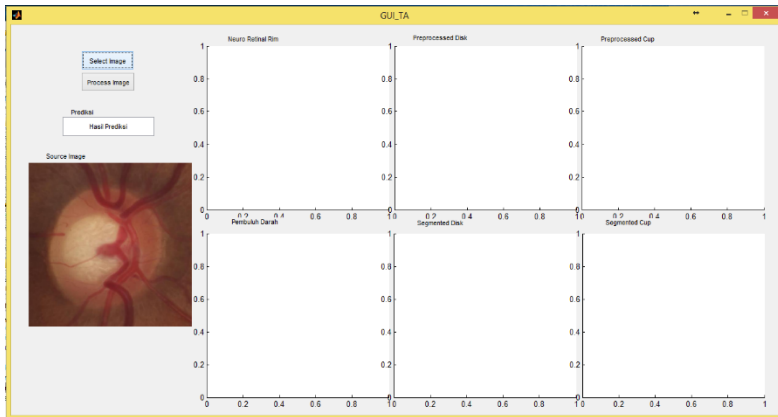
3.4 Perancangan Antarmuka Perangkat Lunak

Perancangan antarmuka perangkat lunak merupakan penggambaran halaman antarmuka yang akan digunakan pengguna ketika berinteraksi dengan aplikasi. Gambar 3.11 menunjukkan rancangan tatap muka perangkat lunak dalam tugas akhir ini.



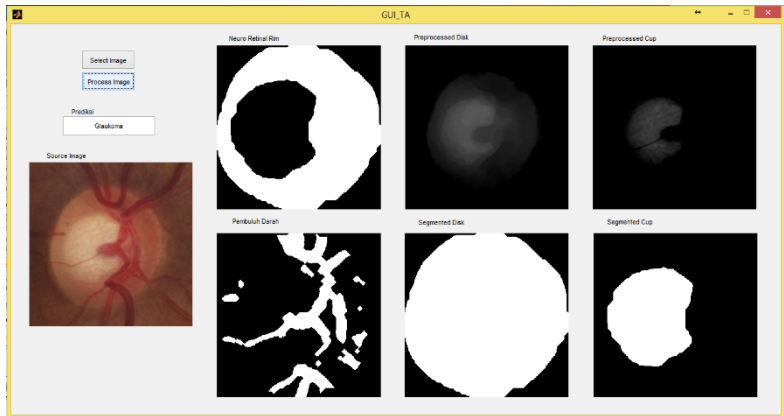
Gambar 3.11 Tatap muka utama

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 tampilan setelah berkas berhasil dipilih.



Gambar 3.12 Tatap muka setelah berkas selesai dipilih

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 Tampilan akhir aplikasi

(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 : *Windows 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*.

Kode sumber 4.1 Implementasi Tahap Preprocessing Optic Cup dan Optic Disk

1	<code>function y = Preprocessing(x,tipe)</code>
2	<code> y = double(x);</code>
3	<code> y1 = double(x);</code>
4	<code> if(strcmp(tipe,'disk') == 1)</code>
5	<code> y = imdilate(y,strel('disk',25));</code>
6	<code> y = imerode(y,strel('disk',25));</code>
7	<code> end</code>
8	<code> tmp = y1(:);</code>
9	<code> xBar = mean(tmp)</code>
10	<code> xStd = std(tmp)</code>
11	<code> if(strcmp(tipe,'disk') == 1)</code>
12	<code> y1 = y1-(xBar);</code>
13	<code> y = y-(xBar);</code>
14	<code> elseif (strcmp(tipe,'cup') == 1)</code>
15	<code> y1 = y1-(xBar+xStd);</code>
16	<code> y = y-(xBar+xStd);</code>
17	<code> end</code>
18	<code> if(strcmp(tipe,'disk') == 1)</code>
19	<code> iter = 5;</code>
20	<code> else</code>
21	<code> iter = 7;</code>
22	<code> end</code>
23	<code> y = uint8(y);</code>
24	<code> x = y;</code>
25	<code> x(x==0) = max(max(y));</code>
26	<code> for i = 1 : iter</code>
27	<code> y = uint8(y - min(min(x)));</code>
28	<code> end</code>

29	<code>end</code>
----	------------------

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.

Kode sumber 4.2 Implementasi Tahap Preprocessing Pembuluh Darah

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

Kode sumber 4.3 Implementasi Bottom Hat Transformation

1	<code>function I = BottomHat(green, SE)</code>
2	<code> I = imdilate(green, SE);</code>
3	<code> I = imerode(I, SE);</code>
4	<code> I = I-green;</code>
5	<code>end</code>

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*.

Kode sumber 4.4 Implementasi Segmentasi Optic Cup

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

Kode sumber 4.5 Implementasi Fungsi Histogram Smoothing

1	<code>function y = HistogramSmoothing(Image)</code>
2	<code>[p,val] = imhist(Image);</code>
3	<code>p = p./numel(Image);</code>
4	<code>mean = 50;</code>
5	<code>sigma = 6;</code>
6	<code>f = normpdf(0:100,mean,sigma);</code>

7	<code>p1 = conv(p,f);</code>
8	<code>y = otsu(p1);</code>
9	<code>End</code>

Kode sumber 4.6 Implementasi Metode Otsu

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

28	<code>level = level/255;</code>
29	<code>End</code>

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*.

Kode sumber 4.7 Implementasi Segmentasi Optic Disk

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

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.

Kode sumber 4.8 Implementasi Segmentasi Pembuluh Darah

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

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*.

Kode sumber 4.9 Implementasi Ekstraksi Fitur CDR

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

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*.

Kode nomor 4.10 Implementasi Proses Cropping

1	<code>[row,col]=find(disk);</code>
2	<code>row_t = min(row);</code>
3	<code>col_t = min(col);</code>
4	<code>disk = imcrop(disk,[col_t row_t max(col)-col_t max(row)-row_t]);</code>
5	<code>cup = imcrop(cup,[col_t row_t max(col)-col_t max(row)-row_t]);</code>

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

Kode nomor 4.11 Implementasi Ekstraksi Fitur ISNT NRR

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

18	<code>end</code>
----	------------------

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*.

Kode sumber 4.12 Implementasi Cropping Pembuluh Darah

1	<code>[row,col]=find(disk);</code>
2	<code>row_t = min(row);</code>
3	<code>col_t = min(col);</code>
4	<code>blood = imcrop(blood,[col_t row_t max(col)-col_t max(row)-row_t]);</code>

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

Kode sumber 4.13 Implementasi Ekstraksi Fitur ISNT Pembuluh Darah

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

16	<code>y = (superior+inferior)/(nasal+temporal);</code>
17	<code>end</code>

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 menggunakan 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*.

Kode sumber 4.14 Implementasi Klasifikasi

1	<code>k=10;</code>
2	<code>newDataset = Dataset;</code>
3	<code>newGroup = Group;</code>
4	<code>cvFolds = crossvalind('Kfold', newGroup, k);</code> <code>%# get indices of 10-fold CV</code>
5	<code>cp = classperf(newGroup);</code> <code>%#</code> <code>init performance tracker</code>
6	<code>dataset = newDataset;</code>
7	<code>metode = 'QP';</code>
8	<code>tic;</code>
9	<code>for i = 1:k</code> <code>%#</code> <code>for each fold</code>
10	<code>testIdx = (cvFolds == i);</code> <code>%#</code> <code>get indices of test instances</code>
11	<code>trainIdx = ~testIdx;</code> <code>%#</code> <code>get indices training instances</code>
12	
13	<code>%# train an SVM model over training</code> <code>instances</code>
14	<code>svmModel = svmtrain(dataset(trainIdx,:),</code> <code>newGroup(trainIdx), ...</code>

15	<code>'Autoscale',true,</code> <code>'Showplot',false, 'Method',metode, ...</code>
16	<code>'BoxConstraint',2e-1,</code> <code>'Kernel_Function','rbf','rbf_sigma',1);</code>
17	<code>## test using test instances</code>
18	<code>pred = svmclassify(svmModel,</code> <code>dataset(testIdx,:), 'Showplot',false);</code>
19	<code>## evaluate and update performance object</code>
20	<code>cp = classperf(cp, pred, testIdx);</code>
21	<code>end</code>
22	<code>## get accuracy</code>
23	<code>akurasi = cp.CorrectRate</code>
24	<code>## get confusion matrix</code>
25	<code>## columns:actual, rows:predicted, last-row:</code> <code>unclassified instances</code>
26	<code>confusionMatriks = cp.CountingMatrix</code>

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.

Kode sumber 4.15 Implementasi User Interface

1	<code>function varargout = GUI_TA(varargin)</code>
2	<code>gui_Singleton = 1;</code>
3	<code>gui_State = struct('gui_Name', mfilename, ...</code>
4	<code> 'gui_Singleton', gui_Singleton,</code> <code> ...)</code>
5	<code> 'gui_OpeningFcn',</code> <code>@GUI_TA_OpeningFcn, ...)</code>
6	<code> 'gui_OutputFcn',</code> <code>@GUI_TA_OutputFcn, ...)</code>
7	<code> 'gui_LayoutFcn', [] , ...)</code>
8	<code> 'gui_Callback', []);</code>
9	<code>if nargin && ischar(varargin{1})</code>

10	gui_State.gui_Callback = str2func(varargin{1});
11	end
12	if nargin
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)
21	varargout{1} = handles.output;
22	function pushbutton1_Callback(hObject, eventdata, handles)
23	[filename, user_canceled] = imgetfile;
24	if (user_canceled == 0)
25	I = imread(filename);
26	handles.I = I;
27	guidata(hObject, handles);
28	axes(handles.axes1);
29	imshow(I);
30	end
31	function pushbutton2_Callback(hObject, eventdata, handles)
32	I = handles.I;
33	%processing image
34	preprocessedDisk = Preprocessing(I(:,:,1),'disk');
35	preprocessedCup = Preprocessing(I(:,:,2),'cup');
36	[disk,index,diskArea] = DiscSegmentation(I);
37	[cup,cupArea] = SegmenCup(I);
38	blood = BloodVessel(I(:,:,2),disk);
39	%cropping
40	[row,col]=find(disk);

41	row_t = min(row);
42	col_t = min(col);
43	disk = imcrop(disk,[col_t row_t max(col)-col_t max(row)-row_t]);
44	cup = imcrop(cup,[col_t row_t max(col)-col_t max(row)-row_t]);
45	blood = imcrop(blood,[col_t row_t max(col)-col_t max(row)-row_t]);
46	%viewing image
47	axes(handles.axes2);
48	imshow(blood);
49	axes(handles.axes3);
50	imshow(preprocessedDisk);
51	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

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 : *Windows* 8 Professional 64 bit.
 - b. Perangkat Pengembang : Matlab R2014a
 - c. Perangkat Pembantu : Microsoft Excel 2013

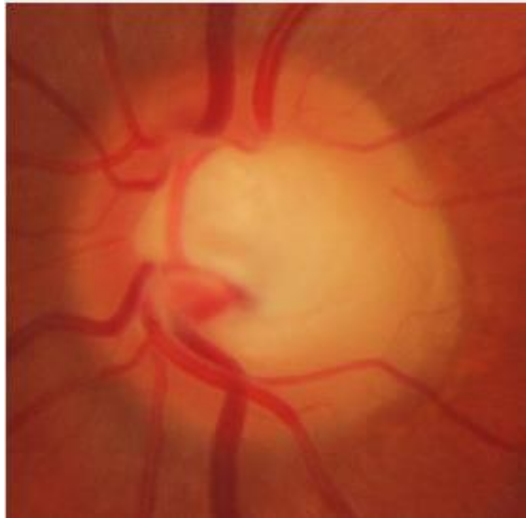
5.2 Data Uji Coba

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*.

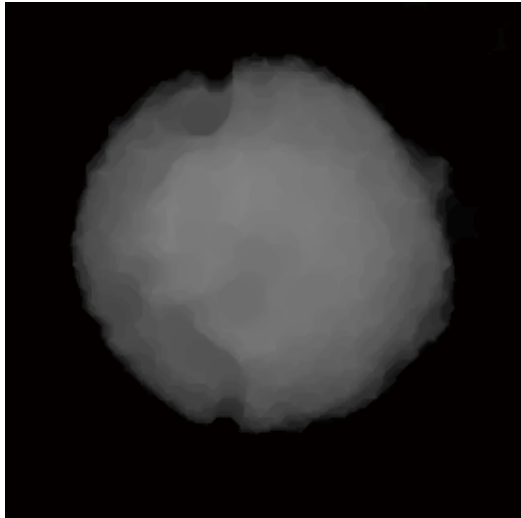
5.3 Uji Coba Sistem

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

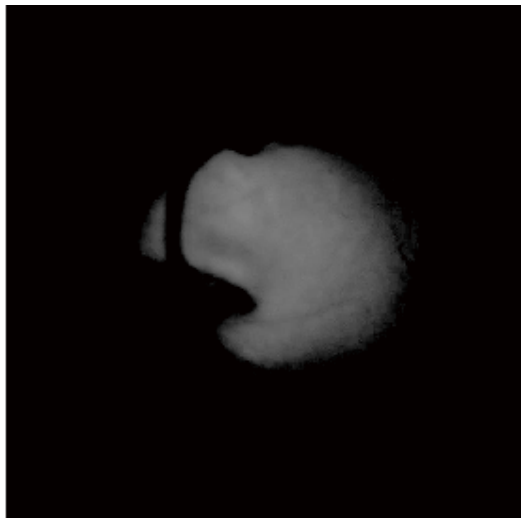


Gambar 5.1 Citra masukan sistem

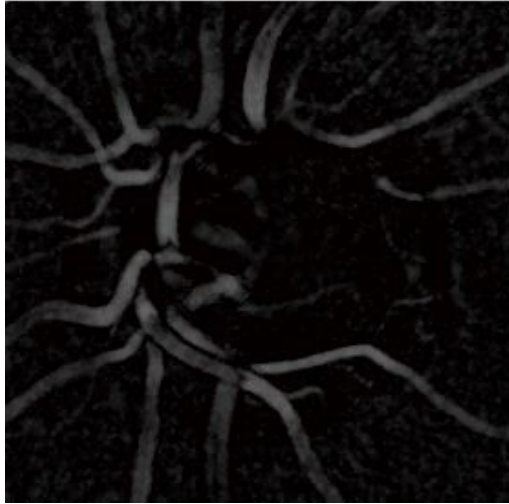
Tahap pertama yang dilalui citra masukan adalah tahap *preprocessing*. Gambar 5.2, 5.3, dan 5.4 secara berturut turut adalah hasil dari tahap *preprocessing optic disk*, *optic cup*, dan pembuluh darah.



Gambar 5.2 Hasil *preprocessing optic disk*



Gambar 5.3 Hasil *preprocessing optic cup*

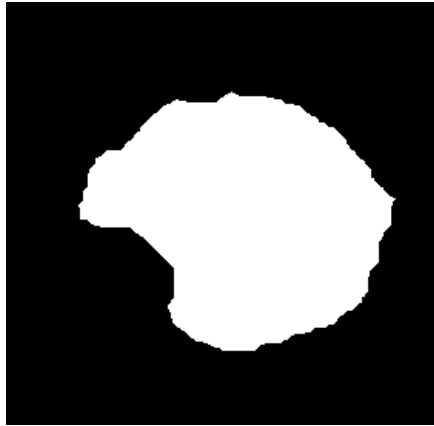


Gambar 5.4 Hasil *preprocessing* pembuluh darah

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.

- 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 skenario 1 dengan Histogram Smoothing

Percobaan	Akurasi	Time	Presisi	Recall
1	79%	0.076003	83%	77%
2	79%	0.080427	81%	80%
3	81%	0.117171	84%	80%
4	82%	0.095897	85%	80%
5	79%	0.079565	82%	78%
Rata-rata	80%	0.089813	83%	79%

Tabel 5.2 Hasil skenario 1 tanpa Histogram Smoothing

Percobaan	Akurasi	Time	Presisi	Recall
1	77%	0.089385	80%	75%
2	73%	0.086302	81%	66%
3	77%	0.105823	81%	73%
4	73%	0.094761	77%	72%
5	74%	0.11959	78%	72%
Rata-rata	75%	0.099172	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%.

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 5.3 Hasil skenario 2 dengan metode SMO

Percobaan	Akurasi	Time	Presisi	Recall
1	79%	0.076003	83%	77%
2	79%	0.080427	81%	80%
3	81%	0.117171	84%	80%
4	82%	0.095897	85%	80%
5	79%	0.079565	82%	78%
Rata-rata	80%	0.089813	83%	79%

Tabel 5.4 Hasil skenario 2 dengan metode LS

Percobaan	Akurasi	Time	Presisi	Recall
1	78%	0.042342	83%	75%
2	80%	0.035718	84%	77%
3	80%	0.037269	84%	77%
4	78%	0.033161	83%	75%
5	79%	0.034508	83%	77%
Rata-rata	79%	0.0366	84%	76%

Tabel 5.5 Hasil skenario 2 dengan metode QP

Percobaan	Akurasi	Time	Presisi	Recall
1	80%	0.243935	83%	78%
2	79%	0.239451	83%	77%
3	80%	0.275507	84%	77%
4	78%	0.231506	83%	75%
5	81%	0.232773	85%	78%
Rata-rata	80%	0.244634	84%	77%

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 waktu dan 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 5.6 Hasil skenario 3 dengan fungsi kernel *linear*

Percobaan	Akurasi	Time	Presisi	Recall
1	79%	0.076003	83%	77%
2	79%	0.080427	81%	80%
3	81%	0.117171	84%	80%
4	82%	0.095897	85%	80%
5	79%	0.079565	82%	78%
Rata-rata	80%	0.089813	83%	79%

Tabel 5.7 Hasil skenario 3 dengan fungsi kernel *quadratic*

Percobaan	Akurasi	Time	Presisi	Recall
1	78%	0.22826	80%	77%
2	77%	0.220579	78%	78%
3	78%	0.304377	80%	80%
4	76%	0.17547	76%	80%
5	75%	0.217376	78%	73%
Rata-rata	77%	0.229212	79%	78%

Tabel 5.8 Hasil skenario 3 dengan fungsi kernel *polynomial orde 3*

Percobaan	Akurasi	Time	Presisi	Recall
1	72%	1.054431	74%	72%
2	76%	0.653245	78%	77%
3	76%	0.821697	78%	77%
4	72%	0.702548	73%	73%
5	73%	0.829963	77%	72%
Rata-rata	74%	0.812377	76%	74%

Tabel 5.9 Hasil skenario 3 dengan fungsi kernel RBF

Percobaan	Akurasi	Time	Presisi	Recall
1	76%	0.079465	75%	83%
2	79%	0.077998	76%	89%
3	80%	0.118394	77%	89%
4	78%	0.088523	76%	86%
5	78%	0.078924	75%	86%
Rata-rata	78%	0.088661	76%	87%

Berdasarkan hasil yang telah didapatkan pada uji coba skenario 3, didapat bahwa nilai akurasi tertinggi adalah 80 % dengan fungsi kernel *linear*. Fungsi kernel *linear* kemudian akan digunakan oleh sistem untuk proses klasifikasi karena memiliki tingkat akurasi yang paling tinggi.

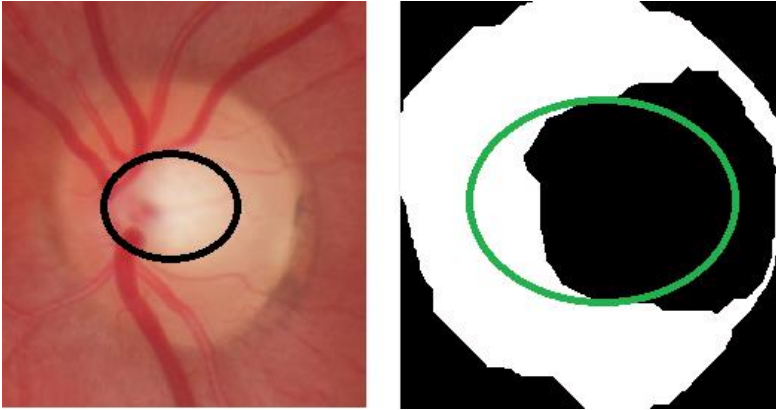
5.5 Evaluasi

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.



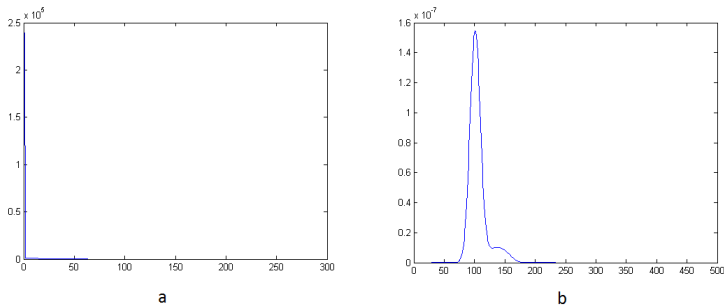
Gambar 5.8 Degradasi pada segmentasi optic cup.

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.

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.



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)

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

6.1 Kesimpulan

Kesimpulan yang diperoleh berdasarkan uji coba dan evaluasi yang telah dilakukan antara lain :

1. Keberadaan pembuluh darah dan pantulan cahaya dapat mengganggu proses segmentasi *optic disk* dan *optic cup*.
2. Proses *smoothing histogram* dapat meningkatkan performa sistem sebesar 5%.
3. Metode pencarian *hyperplane* yang menghasilkan performa terbaik pada *support vector machine* adalah *sequential minimal optimization*.
4. Fungsi kernel yang menghasilkan performa terbaik pada *support vector machine* adalah fungsi kernel *linear*.

6.2 Saran

Beberapa saran yang disampaikan terkait pengerjaan tugas akhir ini adalah :

- i. Perlu dikembangkan metode untuk menangani *noise* pantulan cahaya pada citra retina mata.
- ii. Perlu dikembangkan metode penghilangan pembuluh darah yang lebih baik pada segmentasi *optic cup*.
- iii. Diperlukan penelitian lebih lanjut untuk mendapatkan fitur lain yang lebih baik daripada fitur yang digunakan dalam tugas akhir ini.

(Halaman ini sengaja dikosongkan)

Daftar Pustaka

- [1] Pusat Data dan Informasi Kementerian Kesehatan RI, "InfoDATIN, Situasi dan Analisis Glaukoma," Pusat Data dan Informasi Kementerian Kesehatan RI, Jakarta Selatan, 2015.
- [2] Gillan and Stewart, "Optic Disc Basic," Ninewells Hospital & Medical School, 12 2009. [Online]. Available: http://www.optic-disc.org/tutorials/optic_disc_basics/page12.html. [Accessed 10 12 2015].
- [3] TheFreeDictionary, "Medical Online Dictionary," The American Heritage® Medical Dictionary, 2007. [Online]. Available: <http://medical-dictionary.thefreedictionary.com/optic+cup>. [Accessed 10 12 2015].
- [4] N. Harizman, C. Oliveira, A. Chiang, C. Tello, M. Marmor, R. Ritch and J. Liebmann, "The ISNT rule and differentiation of normal from glaucomatous eyes," *Arch Ophthalmol*, vol. 124, pp. 1579-1583, 2006.
- [5] A. Issac, P. M. Sarathi and M. K. Dutta, "An adaptive threshold based image processing technique for improved glaucoma detection and classification," *Computer Methods and Programs in Biomedicine*, vol. I, no. 22, pp. 229-244, 2015.
- [6] G. V. Tcheslavski, "Lamar University," 22 4 2010. [Online]. Available: <http://www.ee.lamar.edu/gleb/dip/10-3%20->

%20Morphological%20Image%20Processing.pdf.
[Accessed 8 6 2016].

- [7] World Heritage Encyclopedia, "Project Guttenberg Self-Publishing Press," [Online]. Available: http://www.gutenberg.us/articles/otsu's_method. [Accessed 8 6 2016].
- [8] A. S. Nugroho, A. B. Witarto and D. Handoko, "Support Vector Machine-Teori dan Aplikasinya dalam Bioinformatika," *IlmuKomputer.Com*, 2003.
- [9] J. Schneider, "Carnegie Mellon University School of Computer Science," Carnegie Mellon University, 7 February 1997. [Online]. Available: <http://www.cs.cmu.edu/~schneide/tut5/node42.html>. [Accessed 21 6 2016].
- [10] J. Nayak, R. A. U., P. S. Bhat, N. Shetty and T.-C. Lim, "Automated Diagnosis of Glaucoma Using Digital Fundus Image," *J Med Syst*, no. 33, pp. 337-346, 2009.

BIODATA PENULIS



Ahmad Mustofa, penulis dari buku tugas akhir ini lahir di kabupaten Sumenep tanggal 3 Juli 1994. Penulis telah menempuh pendidikan di SD Negeri Pangarangan IV (2000-2006), SMP Negeri 1 Sumenep (2006-2009), SMA Negeri 1 Sumenep (2009-2012) dan Teknik Informatika ITS Surabaya (2012-2016). Selama masa perkuliahan, penulis pernah menjadi asisten pada mata kuliah Interaksi

Manusia dan Komputer, Basis Data Terapan PIKTI ITS, Pengembangan Game Multiplatform PIKTI ITS, dan Pemrograman Android PIKTI ITS. Penulis juga aktif sebagai anggota organisasi Keluarga Muslim Informatika (KMI), Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS dan Jamaah Masjid Manurul Ilmi (JMMI) ITS. Penulis memilih bidang minat Komputasi Cerdas dan Visi (KCV) dan tertarik pada topik *image processing*, *signal processing*, dan kecerdasan buatan. Penulis dapat dihubungi melalui email mustofa.ahmad12@mhs.if.its.ac.id.