



TUGAS AKHIR - KI091391
IMPLEMENTASI PENGKLASIFIKASI SEGMENTASI
VASKULAR RETINA MATA DENGAN METODE M-
MEDIODS MULTIVARIAT

WILDA IMAMA SABILLA
5110100212

Dosen Pembimbing I
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.

Dosen Pembimbing II
Rully Soelaiman, S.Kom, M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2014



**FINAL PROJECT - KI091391
IMPLEMENTATION OF RETINAL VASCULAR
SEGMENT BASED CLASSIFIER USING
MULTIVARIATE M-METHODS**

**WILDA IMAMA SABILLA
5110100212**

**Supervisor
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
Rully Soelaiman, S.Kom, M.Kom.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
Sepuluh Nopember Institute of Technology
Surabaya, 2014**

IMPLEMENTASI PENGKLASIFIKASI SEGMENTASI VASKULAR RETINA MATA DENGAN METODE M- MEDIODS MULTIVARIAT

Nama Mahasiswa : Wilda Imama Sabilla
NRP : 5110 100 212
Jurusan : Teknik Informatika FTIF ITS
Dosen Pembimbing 1 : Dr. Eng. Chastine Fatichah, S.Kom.,
M.Kom.
Dosen Pembimbing 2 : Rully Soelaiman, S.Kom., M.Kom.

Abstrak

Neovaskularisasi adalah berkembangnya pembuluh darah baru di dalam mata. Pembuluh darah ini merupakan pembuluh darah yang abnormal, memiliki dinding pembuluh yang tipis, lemah, dan mudah pecah. Neovaskularisasi dapat terbentuk pada berbagai lokasi di dalam mata termasuk pada retina, sehingga citra retina dapat digunakan untuk mendeteksi neovaskularisasi secara otomatis. Pendeteksian dilakukan dengan melakukan klasifikasi terhadap segmen vaskular retina mata sebagai segmen vaskular normal atau abnormal.

Tugas akhir ini mengimplementasikan salah satu metode pengklasifikasi yang dapat digunakan untuk klasifikasi segmen vaskular retina yaitu m-Medioids multivariat. Metode m-Medioids multivariat terdiri dari dua tahap. Tahap pertama adalah perbaikan ruang fitur menggunakan Local Fisher Discriminant Analysis. Tahap kedua merupakan tahap klasifikasi dengan metode Learning Vector Quantization. Sebelum dilakukan klasifikasi, terlebih dahulu dilakukan pra-proses dengan metode masking untuk memisahkan antara latar belakang gambar dan objek retina. Selanjutnya dilakukan segmentasi pembuluh darah menggunakan transformasi wavelet yaitu dengan Isotropic Undecimated Wavelet Transform. Tahap awal sebelum klasifikasi adalah ekstraksi ciri untuk menghasilkan vektor fitur yang

digunakan sebagai pembeda segmen vaskular normal dan abnormal. Sebelum akhirnya dilakukan klasifikasi segmen vaskular dengan metode m-Medioids multivariat.

Hasil uji coba dari hasil segmentasi vaskular dari citra retina pada basis data DRIVE menghasilkan nilai akurasi mencapai 95,04% dengan perbandingan citra ground truth. Sedangkan hasil klasifikasi segmen vaskular normal dan abnormal dengan citra retina dari basis data STARE menggunakan metode m-Medioids multivariat menunjukkan akurasi terbaik sebesar 96.2%. Sehingga dapat disimpulkan bahwa metode segmentasi vaskular retina dan klasifikasi segmen vaskular retina yang digunakan pada Tugas akhir ini mampu melakukan segmentasi dan klasifikasi dengan baik.

Kata kunci: Klasifikasi, m-Medioids Multivariat, Segmentasi, Vaskular Retina.

IMPLEMENTATION OF RETINAL VASCULAR SEGMENT BASED CLASSIFIER USING MULTIVARIATE M-MEDIODS

Student's Name : Wilda Imama Sabilla
Student's ID : 5110 100 212
Department : Informatics Engineering, FTIF-ITS
First Advisor : Dr. Eng. Chastine Fatichah, S.Kom.,
M.Kom.
Second Advisor : Rully Soelaiman, S.Kom., M.Kom.

Abstract

Neovascularization is development of new blood vessels in the eyes. These new blood vessels are abnormal blood vessels, have the vessel walls are thin, weak, and easily broken. Neovascularization can be formed at various locations within the eyes including the retina, hence the retinal image can be used to detect neovascularization. Detection can be done by classifying retinal vascular segments as a normal or an abnormal vascular segment.

The final project is to implement one of the methods classifiers that can be used for classification of retinal vascular segments namely m-Medioids multivariate. M-Medioids multivariate method consists of two stages. The first stage is the improvement feature space using local fisher discriminant analysis. The second stage is the stage of classification with learning vector quantization method. Before the classification, preprocessing is done by masking method to separate the object of the retina from the background image. Vascular segmentation is then performed using the wavelet transform namely undecimated isotropic wavelet transform. The initial phase of classification is feature extraction to produce a feature vector which is used to distinguish normal and abnormal vascular

segment. Finally classification of vascular segments is done using m-Medioids multivariate methods.

The results of the evaluation of vascular segmentation with retinal image from DRIVE database generate the accuracy value of 95.04% with ground truth image comparison. While the results of the classification of normal and abnormal vascular segments by retinal image from STARE database using multivariate m-Medioids methods shows the best accuracy of 96.2%. Therefore it can be concluded that the method used for retinal vascular segmentation and retinal vascular segments classification in this final project is reliable for segmentation and classification.

Keywords: classification, multivariate m-Medioids, retinal vascular, segmentation.

LEMBAR PENGESAHAN

IMPLEMENTASI PENGKLASIFIKASI SEGMENTASI VASKULAR RETINA MATA DENGAN METODE M- MEDIODS MULTIVARIAT

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Pada

Bidang Studi Komputasi Cerdas Visual
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

WILDA IMAMA SABILLA
NRP: 5110 100 212

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Dr. Eng. Chastine Faticah, S.Kom., M.Kom.
NIP: 197512202001122002 (Pembimbing 1)
2. Rully Soelaiman, S.Kom., M.Kom.
NIP: 197002131994021001 (Pembimbing 2)

SURABAYA
JUNI 2014

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'amin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul ***“IMPLEMENTASI PENGKLASIFIKASI SEGMENT VASKULAR RETINA MATA DENGAN METODE M-MEDIODS MULTIVARIAT”***.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Rully Soelaiman, S.Kom., M.Kom. dan Ibu Dr. Eng. Chastine Fatichah, S.Kom., M.Kom. selaku pembimbing yang telah memberikan motivasi, nasehat, bimbingan dan bantuan kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS dan segenap dosen Teknik Informatika yang telah memberikan ilmunya selama penulis berkuliah di Teknik Informatika ITS.
3. Para penulis artikel ilmiah yang digunakan pada Tugas akhir ini, pendiri Google, pendiri Mathworks, serta berbagai referensi lain, yang telah membantu mempermudah penyelesaian Tugas Akhir ini.

4. Ayah dan Ibu dari penulis, Bapak Mokh. Irijik dan Ibu Wahyuni, yang senantiasa memberikan dukungan moral, material, serta doa. Serta menjadi motivasi penulis untuk selalu menjalankan kuliah dengan baik dan menyelesaikan Tugas Akhir ini.
5. Kawan-kawan angkatan 2010 yang menemani, membantu, dan memotivasi selama penulis menjalankan seluruh tugas perkuliahan hingga penulis sampai pada tahap pengerjaan Tugas Akhir ini.
6. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juni 2014

Wilda Imama Sabilla

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak	vii
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR KODE SUMBER	xxvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Permasalahan	2
1.3. Batasan Masalah	3
1.4. Tujuan dan Manfaat	3
1.5. Metodologi.....	4
1.6. Sistematika Laporan.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1. Praproses Citra Retina.....	7
2.1.1 Perluasan area objek retina.....	8
2.1.2 <i>Masking</i>	9
2.2. Segmentasi Vaskular Retina	10
2.2.1 2-D <i>Morlet Wavelet Transform</i>	11
2.2.2 Normalisasi Fitur.....	12
2.2.3 <i>Gaussian mixture model Bayesian classifier</i>	13
2.2.4 <i>Differential Filtering</i>	15
2.2.5 Proses Morfologi.....	17
2.2.6 <i>Isotropic Undecimated Wavelet Transform</i>	18
2.2.7 <i>Thresholding</i>	19
2.2.8 <i>Cleaning</i>	21
2.3. Ekstraksi Fitur Vaskular Retina	22
2.3.1 <i>Area</i>	23
2.3.2 <i>Energy</i>	23
2.3.3 <i>Mean Gradient</i>	23

2.3.4	<i>Standard Deviation Gradient</i>	24
2.3.5	<i>Mean Intensity</i>	24
2.3.6	<i>Intensity Variation</i>	25
2.3.7	<i>Vessel Segment</i>	25
2.4.	Klasifikasi Vaskular Normal dan Abnormal dengan M- Mediods Multivariat.....	25
2.4.1	<i>Local Fisher Discriminant Analysis</i>	26
2.4.2	<i>Learning Vector Quantization</i>	27
BAB III PERANCANGAN		31
3.1.	Perancangan Praproses dan Segmentasi Vaskular	31
3.1.1	Program Utama.....	39
3.1.2	<i>Masking</i>	39
3.1.3	Transformasi Wavelet.....	40
3.1.4	<i>Cleaning</i>	41
3.2.	Perancangan Ekstraksi Fitur Segmen Vaskular Retina	42
3.2.1	Program Utama.....	47
3.2.2	Fitur <i>Area</i>	48
3.2.3	Fitur <i>Energy</i>	49
3.2.4	Fitur <i>Mean Gradient</i>	49
3.2.5	Fitur <i>Standard Deviation Gradient</i>	50
3.2.6	Fitur <i>Mean Intensity</i> dan <i>Intensity Variation</i>	51
3.2.7	Fitur <i>Vessel Segment</i>	52
3.3.	Perancangan Klasifikasi Segmen Vaskular Retina	53
3.3.1	Program <i>Local Fisher Discriminant Analysis</i>	62
3.3.2	Program <i>Learning Vector Quantization</i>	64
BAB IV IMPLEMENTASI		67
4.1.	Lingkungan Implementasi.....	67
4.2.	Implementasi	67
4.2.1.	Implementasi Praproses dan Segmentasi Citra Retina	67

4.2.2. Implementasi Ekstraksi Fitur Segmen Vaskular Retina	71
4.2.3. Implementasi Klasifikasi Segmen Vaskular Retina	75
BAB V UJI COBA DAN EVALUASI	81
5.1. Lingkungan Uji Coba	81
5.2. Uji Coba Kebenaran Citra Hasil Segmentasi	81
5.2.1 Data Uji Coba Kebenaran Citra Hasil Segmentasi	82
5.2.2 Skenario Uji Coba Kebenaran Citra Hasil Segmentasi	82
5.2.3 Uji Coba dan Evaluasi Kebenaran Citra Hasil Segmentasi	83
5.3. Uji Coba Klasifikasi Segmen Vaskular Normal dan Abnormal	85
5.3.1. Data Uji Coba Kebenaran Klasifikasi Segmen Vaskular Normal dan Abnormal	85
5.3.2. Skenario Uji Coba Kebenaran Klasifikasi Segmen Vaskular Normal dan Abnormal	89
5.3.3. Uji Coba dan Evaluasi Kebenaran Klasifikasi Segmen Vaskular Normal dan Abnormal	91
5.3.4. Uji Coba dan Evaluasi Kebenaran Klasifikasi Berdasarkan Jumlah Dimensi Hasil Reduksi	94
5.3.5. Uji Coba dan Evaluasi Kebenaran Klasifikasi Berdasarkan Jumlah <i>Mediods</i>	96
5.3.6. Uji Coba dan Evaluasi Kebenaran Klasifikasi Berdasarkan Nilai <i>Learning Rate</i>	98
5.4. Uji Coba Waktu Eksekusi Perangkat Lunak	100
5.4.1. Data Uji Coba Waktu Eksekusi Perangkat Lunak	100
5.4.2. Skenario Uji Coba Waktu Eksekusi Perangkat Lunak	100
5.4.3. Uji Coba dan Evaluasi Waktu Eksekusi Perangkat Lunak	101

BAB VI KESIMPULAN DAN SARAN.....	103
6.1. Kesimpulan	103
6.2. Saran	104
LAMPIRAN A	107

DAFTAR GAMBAR

Gambar 2. 1	Citra retina dari basis data STARE.....	7
Gambar 2. 2	Citra retina dalam <i>inverse green channel</i> dan citra hasil praproses	8
Gambar 2. 3	Citra <i>masking</i> dan hasil <i>shrinking</i>	10
Gambar 2. 4	Citra retina hasil 2-D Morlet Wavelet Transform	12
Gambar 2. 5	Citra hasil segmentasi dengan metode yang ditawarkan Soares dkk.....	14
Gambar 2. 6	(a) Citra pembuluh yang diperbaiki (b) Citra hasil filter FoDoG dengan orientasi 0°	16
Gambar 2. 7	Citra <i>centreline</i> pembuluh	16
Gambar 2. 8	Citra hasil segmentasi dengan metode yang ditawarkan Fraz dkk	17
Gambar 2. 9	Hasil Citra Retina dengan <i>Two Dimensional Isotropic Undecimated Wavelet Transform</i>	20
Gambar 2. 10	Hasil citra retina setelah proses <i>thresholding</i>	20
Gambar 2. 11	Citra hasil akhir segmentasi.....	22
Gambar 3. 1	Bagan gambaran umum praproses dan segmentasi citra.....	32
Gambar 3. 2	<i>Pseudocode</i> program utama.....	39
Gambar 3. 3	<i>Pseudocode</i> program <i>masking</i>	40
Gambar 3. 4	<i>Pseudocode</i> proses transformasi wavelet (bagian pertama).....	40
Gambar 3. 5	<i>Pseudocode</i> proses transformasi wavelet (bagian kedua)	41
Gambar 3. 6	<i>Pseudocode</i> program <i>cleaning</i>	42
Gambar 3. 7	Bagan gambaran umum proses ekstraksi fitur segmen vaskular retina	43
Gambar 3. 8	<i>Pseudocode</i> program utama untuk ekstraksi fitur	48
Gambar 3. 9	<i>Pseudocode</i> program menghitung <i>area</i>	48
Gambar 3. 10	<i>Pseudocode</i> program menghitung <i>energy</i>	49

Gambar 3. 11	<i>Pseudocode</i> program menghitung <i>mean gradient</i>	50
Gambar 3. 12	<i>Pseudocode</i> program menghitung <i>standard deviation gradient</i> (bagian pertama).....	50
Gambar 3. 13	<i>Pseudocode</i> program menghitung <i>standard deviation gradient</i> (bagian kedua)	51
Gambar 3. 14	<i>Pseudocode</i> program menghitung <i>mean intensity dan intensity variation</i> (bagian pertama)	51
Gambar 3. 15	<i>Pseudocode</i> program menghitung <i>mean intensity dan intensity variation</i> (bagian kedua)	52
Gambar 3. 16	<i>Pseudocode</i> program menghitung <i>mean vessel segment</i>	53
Gambar 3. 17	Bagan gambaran umum proses klasifikasi segmen vaskular retina.....	54
Gambar 3. 18	<i>Pseudocode</i> program <i>Local Fisher Discriminant Analysis</i> (bagian pertama).....	62
Gambar 3. 19	<i>Pseudocode</i> program <i>Local Fisher Discriminant Analysis</i> (bagian kedua).....	63
Gambar 3. 20	<i>Pseudocode</i> fungsi <i>Local Fisher Discriminant Analysis</i> (bagian pertama).....	63
Gambar 3. 21	<i>Pseudocode</i> fungsi <i>Local Fisher Discriminant Analysis</i> (bagian kedua).....	64
Gambar 3. 22	<i>Pseudocode Learning Vector Quantization</i>	65
Gambar 5. 1	Contoh data citra yang merupakan TN.....	93
Gambar 5. 2	Contoh data citra yang merupakan FP.....	93
Gambar 5. 3	Contoh data citra yang merupakan TP.....	94
Gambar 5. 4	Nilai <i>error</i> untuk berbagai <i>learning rate</i> dari 24 data <i>training</i> optimal	99
Gambar 5. 5	Nilai <i>error</i> untuk berbagai <i>learning rate</i> dari 24 data <i>training</i> tidak optimal.....	99

DAFTAR KODE SUMBER

Kode Sumber 4. 1 Implementasi proses masking dengan <i>thresholding</i> dan <i>shrinking</i>	68
Kode Sumber 4. 2 Implementasi pengubahan citra asli ke ruang warna skala keabuan	69
Kode Sumber 4. 3 Implementasi transformasi wavelet.....	69
Kode Sumber 4. 4 Implementasi k fungsi dilatasi kernel wavelet	70
Kode Sumber 4. 5 Implementasi <i>cleaning</i> (bagian pertama)	70
Kode Sumber 4. 6 Implementasi <i>cleaning</i> (bagian kedua).....	71
Kode Sumber 4. 7 Implementasi fitur <i>area</i>	71
Kode Sumber 4. 8 Penggabungan citra hasil segmentasi dan citra asli.....	72
Kode Sumber 4. 9 Implementasi fitur <i>energy</i>	72
Kode Sumber 4. 10 Implementasi fitur <i>mean gradient</i>	73
Kode Sumber 4. 11 Implementasi <i>standard deviation gradient</i> ..	73
Kode Sumber 4. 12 Implementasi fitur <i>mean intensity</i> dan <i>intensity variation</i> (bagian pertama)	74
Kode Sumber 4. 13 Implementasi fitur <i>mean intensity</i> dan <i>intensity variation</i> (bagian kedua).....	75
Kode Sumber 4. 14 Implementasi fitur <i>vessel segments</i>	75
Kode Sumber 4. 15 Implementasi program <i>Local Fisher Discriminant Analysis</i>	76
Kode Sumber 4. 16 Implementasi fungsi <i>Local Fisher Discriminant Analysis</i> (bagian pertama).....	77
Kode Sumber 4. 17 Implementasi fungsi <i>Local Fisher Discriminant Analysis</i> (bagian kedua)	78

Kode Sumber 4. 18 Implementasi fungsi <i>Learning Vector Quantization</i> (bagian pertama).....	78
Kode Sumber 4. 19 Implementasi fungsi <i>Learning Vector Quantization</i> (bagian kedua)	79

DAFTAR TABEL

Tabel 3. 1	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Praproses dan Perbaikan Kontras Citra Retina (Bagian Pertama)	33
Tabel 3. 2	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Praproses dan Perbaikan Kontras Citra Retina (Bagian Kedua).....	34
Tabel 3. 3	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Praproses dan Perbaikan Kontras Citra Retina (Bagian Ketiga).....	35
Tabel 3. 4	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Praproses dan Perbaikan Kontras Citra Retina (Bagian Keempat)	36
Tabel 3. 5	Daftar Fungsi yang Digunakan Pada <i>Pseudocode</i> Praproses dan Perbaikan Kontras Citra Retina (Bagian Pertama)	37
Tabel 3. 6	Daftar Fungsi yang Digunakan Pada <i>Pseudocode</i> Praproses dan Perbaikan Kontras Citra Retina (Bagian Kedua).....	38
Tabel 3. 7	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Ekstraksi Fitur Segmen Vaskular (Bagian Pertama)	44
Tabel 3. 8	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Ekstraksi Fitur Segmen Vaskular (Bagian Kedua) ..	45
Tabel 3. 9	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Ekstraksi Fitur Segmen Vaskular (Bagian Ketiga) ..	46
Tabel 3. 10	Daftar Fungsi yang Digunakan Pada <i>Pseudocode</i> Ekstraksi Fitur Segmen Vaskular (Bagian Pertama)	46
Tabel 3. 11	Daftar Fungsi yang Digunakan Pada <i>Pseudocode</i> Ekstraksi Fitur Segmen Vaskular (Bagian Kedua) ..	47

Tabel 3. 12	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Klasifikasi Segmen Vaskular (Bagian Pertama)	54
Tabel 3. 13	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Klasifikasi Segmen Vaskular (Bagian Kedua).....	55
Tabel 3. 14	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Klasifikasi Segmen Vaskular (Bagian Ketiga)	56
Tabel 3. 15	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Klasifikasi Segmen Vaskular (Bagian Keempat).....	57
Tabel 3. 16	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Klasifikasi Segmen Vaskular (Bagian Kelima)	58
Tabel 3. 17	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Klasifikasi Segmen Vaskular (Bagian Keenam).....	59
Tabel 3. 18	Daftar Variabel yang Digunakan Pada <i>Pseudocode</i> Klasifikasi Segmen Vaskular (Bagian Ketujuh)	60
Tabel 3. 19	Daftar Fungsi yang Digunakan Pada <i>Pseudocode</i> Klasifikasi Segmen Vaskular (Bagian Pertama)	60
Tabel 3. 20	Daftar Fungsi yang Digunakan Pada <i>Pseudocode</i> Klasifikasi Segmen Vaskular (Bagian Kedua).....	61
Tabel 5. 1	<i>Confusion matrix</i> untuk segmentasi citra vaskular retina	82
Tabel 5. 2	Hasil perbandingan akurasi segmentasi dengan berbagai metode	84
Tabel 5. 3	Daftar citra pada 20 data <i>training</i> optimal (bagian pertama)	86
Tabel 5. 4	Daftar citra pada 20 data <i>training</i> optimal (bagian kedua).....	87
Tabel 5. 5	Daftar citra pada 24 data <i>training</i> optimal (bagian pertama)	87
Tabel 5. 6	Daftar citra pada 24 data <i>training</i> optimal (bagian kedua).....	88
Tabel 5. 7	Daftar citra pada 20 data <i>training</i> tidak optimal.....	88
Tabel 5. 8	Daftar citra pada 24 data <i>training</i> tidak optimal.....	89

Tabel 5. 9	<i>Confusion matrix</i> untuk klasifikasi segmen vaskular	90
Tabel 5. 10	Hasil percobaan dengan 4 jenis data <i>training</i>	92
Tabel 5. 11	<i>Matrix confusion</i> dari data <i>training</i> optimal.....	92
Tabel 5. 12	Hasil percobaan dengan 24 data <i>training</i> optimal dengan perubahan jumlah dimensi hasil reduksi	95
Tabel 5. 13	Hasil percobaan dengan 24 data <i>training</i> tidak optimal dengan perubahan dimensi reduksi.....	95
Tabel 5. 14	Hasil percobaan dengan 24 data <i>training</i> optimal dengan perubahan jumlah <i>mediods</i>	97
Tabel 5. 15	Hasil percobaan dengan 24 data <i>training</i> tidak optimal dengan perubahan jumlah <i>mediods</i>	97
Tabel 5. 16	Hasil Uji Coba Waktu Eksekusi Perangkat Lunak.	101
Tabel 5. 17	Waktu Eksekusi yang Dibutuhkan untuk Berbagai Jumlah <i>Mediods</i>	102
Tabel A. 1	Citra Hasil Segmentasi dan Citra <i>Ground Truth</i> dari Basis Data DRIVE (Bagian Pertama)	107
Tabel A. 2	Citra Hasil Segmentasi dan Citra <i>Ground Truth</i> dari Basis Data DRIVE (Bagian Kedua).....	108
Tabel A. 3	Citra Hasil Segmentasi dan Citra <i>Ground Truth</i> dari Basis Data DRIVE (Bagian Ketiga)	109
Tabel A. 4	Citra Hasil Segmentasi dan Citra <i>Ground Truth</i> dari Basis Data DRIVE (Bagian Keempat).....	110
Tabel A. 5	Citra Hasil Segmentasi dan Citra <i>Ground Truth</i> dari Basis Data DRIVE (Bagian Kelima)	111
Tabel A. 6	Citra Hasil Segmentasi dan Citra <i>Ground Truth</i> dari Basis Data DRIVE (Bagian Keenam).....	112
Tabel A. 7	Citra Hasil Segmentasi dan Citra <i>Ground Truth</i> dari Basis Data DRIVE (Bagian Ketujuh)	113
Tabel A. 8	Citra Hasil Segmentasi dan Citra <i>Ground Truth</i> dari Basis Data DRIVE (Bagian Kedelapan).....	114
Tabel A. 9	Citra Hasil Segmentasi dan Citra <i>Ground Truth</i> dari Basis Data DRIVE (Bagian Kesembilan).....	115

Tabel A. 10 Citra Hasil Segmentasi dan Citra <i>Ground Truth</i> dari Basis Data DRIVE (Bagian Kesepuluh)	116
Tabel A. 11 Citra dari Basis Data STARE (Bagian Pertama) ...	117
Tabel A. 12 Citra dari Basis Data STARE (Bagian Kedua).....	118
Tabel A. 13 Citra dari Basis Data STARE (Bagian Ketiga).....	119
Tabel A. 14 Citra dari Basis Data STARE (Bagian Keempat) ..	120
Tabel A. 15 Citra dari Basis Data STARE (Bagian Kelima)	121
Tabel A. 16 Citra dari Basis Data STARE (Bagian Keenam)...	122
Tabel A. 17 Citra dari Basis Data STARE (Bagian Ketujuh) ...	123
Tabel A. 18 Citra dari Basis Data STARE (Bagian Kedelapan)	124
Tabel A. 19 Citra dari Basis Data STARE (Bagian Kesembilan)	125
Tabel A. 20 Citra dari Basis Data STARE (Bagian Kesepuluh)	126
Tabel A. 21 Citra dari Basis Data STARE (Bagian Kesebelas) ..	127
Tabel A. 22 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan 3 Dimensi Hasil Reduksi	128
Tabel A. 23 Hasil Uji Coba dari 20 Data <i>Training</i> Optimal dengan 3 Dimensi Hasil Reduksi	129
Tabel A. 24 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan 3 Dimensi Hasil Reduksi	130
Tabel A. 25 Hasil Uji Coba dari 20 Data <i>Training</i> Tidak Optimal dengan 3 Dimensi Hasil Reduksi	131
Tabel A. 26 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan 4 Dimensi Hasil Reduksi	132
Tabel A. 27 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan 5 Dimensi Hasil Reduksi	133

Tabel A. 28 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan 6 Dimensi Hasil Reduksi.....	134
Tabel A. 29 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan 4 Dimensi Hasil Reduksi.....	135
Tabel A. 30 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan 5 Dimensi Hasil Reduksi.....	136
Tabel A. 31 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan 6 Dimensi Hasil Reduksi.....	137
Tabel A. 32 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan 3 Dimensi Hasil Reduksi dan 10 Buah <i>Mediods</i>	138
Tabel A. 33 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan 3 Dimensi Hasil Reduksi dan 8 Buah <i>Mediods</i>	139
Tabel A. 34 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan 3 Dimensi Hasil Reduksi dan 6 Buah <i>Mediods</i>	140
Tabel A. 35 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan 3 Dimensi Hasil Reduksi dan 10 Buah <i>Mediods</i>	141
Tabel A. 36 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan 3 Dimensi Hasil Reduksi dan 8 Buah <i>Mediods</i>	142
Tabel A. 37 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan 3 Dimensi Hasil Reduksi dan 6 Buah <i>Mediods</i>	143
Tabel A. 38 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan berbagai <i>Learning Rate</i> (Bagian Pertama)	144
Tabel A. 39 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan berbagai <i>Learning Rate</i> (Bagian Kedua)...	144

Tabel A. 40 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan berbagai <i>Learning Rate</i> (Bagian Ketiga)...	145
Tabel A. 41 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan berbagai <i>Learning Rate</i> (Bagian Keempat)	145
Tabel A. 42 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan berbagai <i>Learning Rate</i> (Bagian Kelima)..	146
Tabel A. 43 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan berbagai <i>Learning Rate</i> (Bagian Keenam)	146
Tabel A. 44 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan berbagai <i>Learning Rate</i> (Bagian Ketujuh)	147
Tabel A. 45 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan berbagai <i>Learning Rate</i> (Bagian Kedelapan)	147
Tabel A. 46 Hasil Uji Coba dari 24 Data <i>Training</i> Optimal dengan berbagai <i>Learning Rate</i> (Bagian Kesembilan)	148
Tabel A. 47 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan berbagai <i>Learning Rate</i> (Bagian Pertama)	148
Tabel A. 48 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan berbagai <i>Learning Rate</i> (Bagian Kedua)...	149
Tabel A. 49 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan berbagai <i>Learning Rate</i> (Bagian Ketiga)...	149
Tabel A. 50 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan berbagai <i>Learning Rate</i> (Bagian Keempat)	150
Tabel A. 51 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan berbagai <i>Learning Rate</i> (Bagian Kelima)..	150
Tabel A. 52 Hasil Uji Coba dari 24 Data <i>Training</i> Tidak Optimal dengan berbagai <i>Learning Rate</i> (Bagian Keenam)	151

Tabel A. 53 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Ketujuh) 151

Tabel A. 54 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Kedelapan) 152

Tabel A. 55 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Kesembilan) 152

BAB I

PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari Tugas Akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan Tugas Akhir.

1.1. Latar Belakang

Neovaskularisasi adalah berkembangnya pembuluh darah baru di dalam mata. Pembuluh darah baru yang berkembang merupakan pembuluh darah yang abnormal, memiliki dinding pembuluh yang tipis, lemah, dan mudah pecah [1]. Ketika pembuluh darah ini pecah, akan terjadi pendarahan pada daerah mata dan dapat berakibat pada kebutaan. Oleh karena itu diperlukan pendeteksian dini terhadap keberadaan neovaskularisasi untuk mengurangi resiko kebutaan.

Neovaskularisasi dapat terbentuk pada berbagai lokasi di dalam mata termasuk pada retina, sehingga citra retina dapat digunakan untuk mendeteksi neovaskularisasi. Pendeteksian dilakukan dengan mengklasifikasikan segmen vaskular retina mata sebagai segmen vaskular normal atau abnormal. Mata sehat hanya memiliki segmen vaskular normal, sedangkan pada mata yang tidak sehat akan ditemui segmen vaskular abnormal.

Salah satu metode pengklasifikasi yang dapat digunakan adalah m-Medioids multivariat. m-Medioids multivariat adalah sebuah metode untuk memodelkan dan mengklasifikasikan data yang berdistribusi multivariat serta memiliki pola tertentu. m-Medioids multivariat dikembangkan berdasarkan observasi bahwa semua distribusi data dalam pola dapat direpresentasikan oleh perwakilan data (*medioids*) yang terbagi dan terpisah dengan baik. Pengklasifikasi ini terdiri dari dua modul utama, yaitu pembuatan model dan klasifikasi data. Model m-Medioids multivariat yang dibuat dimaksudkan untuk memenuhi variasi distribusi yang

dimiliki oleh pola-pola tertentu. Selanjutnya klasifikasi akan dilakukan menggunakan model yang dibuat tersebut [2]. Karena segmen vaskular retina mata yang akan diklasifikasikan memiliki pola tertentu dan berdistribusi multivariat, maka m-Medioids multivariat adalah metode klasifikasi yang sesuai untuk digunakan.

Tugas Akhir ini mengimplementasikan pengklasifikasi segmen vaskular retina mata dengan menggunakan metode m-Medioids multivariat. Sebelum dilakukan klasifikasi, terlebih dahulu dilakukan praproses dengan metode *masking* untuk memisahkan antara latar belakang gambar dan objek retina. Selanjutnya dilakukan segmentasi pembuluh darah pada citra retina, serta ekstraksi ciri untuk menghasilkan vektor fitur yang digunakan sebagai pembeda segmen vaskular normal dan abnormal. Kemudian dilakukan klasifikasi segmen vaskular dengan metode m-Medioids multivariat.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Memahami konsep praproses dan segmentasi untuk menghasilkan citra segmentasi vaskular retina.
2. Memahami konsep ekstraksi fitur pada vaskular retina.
3. Memahami konsep m-Medioids multivariat sebagai metode pengklasifikasi.
4. Merancang sistem yang dapat melakukan segmentasi dan ekstraksi fitur vaskular retina, serta klasifikasi dengan metode m-Medioids multivariat.
5. Mengimplementasikan sistem yang telah dirancang pada segmen vaskular retina mata.
6. Menyusun uji coba segmentasi citra retina serta uji coba pengklasifikasi segmen vaskular retina mata menggunakan metode m-Medioids multivariat.

1.3. Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu:

1. Data yang digunakan untuk uji segmentasi pembuluh darah adalah data citra retina dari basis data DRIVE yang pada <http://www.isi.uu.nl/Research/Databases/DRIVE/> [3].
2. Data yang digunakan untuk data latih dan data uji klasifikasi segmen vaskular berupa citra retina dari basis data STARE (<http://www.ces.clemson.edu/~ahoover/stare>) [4].
3. Segmen vaskular retina mata diklasifikasikan ke dalam dua kelas, yaitu segmen vaskular normal dan segmen vaskular abnormal.
4. Implementasi menggunakan perangkat lunak MATLAB.

1.4. Tujuan dan Manfaat

Tujuan pengerjaan Tugas Akhir ini adalah:

1. Mengetahui penerapan konsep praproses dan segmentasi untuk menghasilkan citra segmentasi vaskular retina.
2. Mengetahui penerapan konsep ekstraksi fitur pada vaskular retina.
3. Mengetahui penerapan konsep m-Methods multivariat sebagai metode pengklasifikasi.
4. Menghasilkan rancangan sistem yang dapat melakukan segmentasi dan ekstraksi fitur vaskular retina, serta klasifikasi dengan metode m-Methods multivariat.
5. Mengimplementasikan rancangan sistem pada segmen vaskular retina mata.
6. Mengevaluasi kinerja metode segmentasi dan ekstraksi fitur, serta pengklasifikasi m-Methods multivariat pada segmen vaskular retina mata dengan melakukan uji coba.

Sedangkan manfaat pengerjaan Tugas Akhir ini adalah mempermudah dan mempercepat proses klasifikasi segmen vaskular retina normal dan abnormal secara otomatis. Diharapkan juga tugas akhir ini dapat mengurangi tingkat kesalahan dalam klasifikasi sehingga hasil yang diperoleh lebih akurat daripada proses manual.

1.5. Metodologi

Metodologi yang digunakan pada pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.

Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Proposal Tugas Akhir yang diajukan memiliki gagasan untuk melakukan implementasi pengklasifikasi segmen vaskular pada retina mata dengan menggunakan metode m-Mediods multivariat.

2. Studi literatur

Pada tahap ini dilakukan pencarian, pengumpulan, pembelajaran dan pemahaman informasi dan literatur yang diperlukan untuk pembuatan implementasi pada tahap praproses citra retina, segmentasi citra retina sehingga menghasilkan segmen vaskular retina, ekstraksi fitur serta klasifikasi dengan metode m-Mediods multivariat. Dasar informasi yang diperlukan pada pembuatan implementasi ini diantaranya mengenai citra secara umum dan algoritma tahap praproses. Proses segmentasi dengan penerapan *undecimate isotropic wavelet transform* dan operasi morfologi. Serta metode pengklasifikasi segmen vascular pada retina dengan m-Mediods multivariat. Informasi dan literatur didapatkan dari buku dan sumber-sumber informasi lain yang berhubungan.

3. Perancangan perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

4. Implementasi perangkat lunak

Implementasi merupakan tahap membangun rancangan sistem yang telah dibuat. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah sistem yang sesuai dengan apa yang telah direncanakan.

5. 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.

6. Penyusunan buku Tugas Akhir.

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.6. Sistematika Laporan

Buku Tugas Akhir ini disusun dengan sistematika sebagai berikut:

BAB I. PENDAHULUAN

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

BAB II. DASAR TEORI

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

BAB III. PERANCANGAN PERANGKAT LUNAK

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *pseudocode*.

BAB IV. IMPLEMENTASI

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

BAB V. UJI COBA DAN EVALUASI

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

BAB VI. KESIMPULAN DAN SARAN

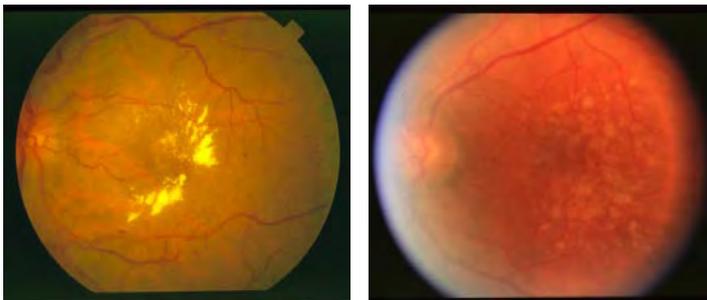
Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan algoritma yang diajukan pada pengimplementasian perangkat lunak. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1. Praproses Citra Retina

Citra retina mata dapat memberikan informasi tentang struktur vaskular retina mata. Dengan mengamati perubahan yang terjadi pada struktur vaskular retina mata, maka citra retina dapat digunakan untuk mendeteksi ketidaknormalan yang terjadi pada retina. Namun tidak semua citra retina dapat secara langsung diproses hingga ditemukan ketidaknormalan pada vaskular retina. Beberapa citra memiliki kualitas yang kurang karena sulit dibedakan antara *foreground* dan *background* seperti beberapa citra retina dari basis data STARE[4] yang ditunjukkan pada Gambar 2.1.



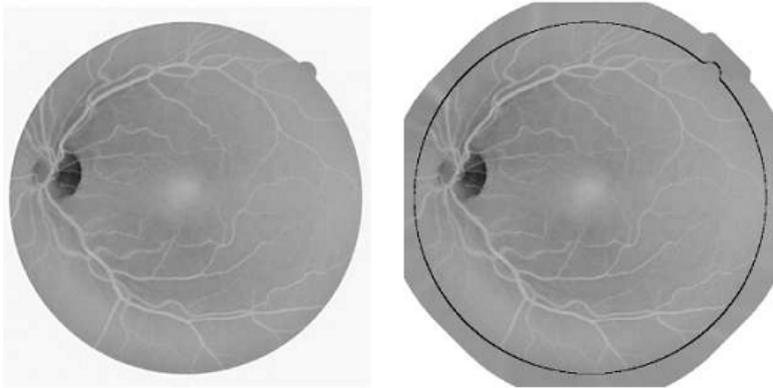
Gambar 2. 1 Citra retina dari basis data STARE

Untuk menghasilkan kualitas gambar yang baik dibutuhkan proses perbaikan kualitas citra terlebih dahulu dan disebut praproses citra retina. Terdapat beberapa metode

praproses, antara lain dengan memperluas area objek retina dan dengan metode *masking*.

2.1.1 Perluasan area objek retina

Soares dkk menggunakan metode praproses untuk mengurangi tingkat kesalahan deteksi objek retina secara iteratif. Praproses ini bertujuan untuk menghapus kontras yang tinggi antara objek retina dengan latar belakang gambar. Algoritma praproses dimulai dengan menentukan piksel latar belakang yang bertetangga dengan piksel objek retina. Kemudian piksel latar belakang tersebut diubah dengan nilai rerata dari piksel tetangga yang merupakan objek retina. Proses akan terus diulang sehingga menghasilkan area objek retina yang lebih luas dari sebelumnya. Sebelum dilakukan praproses terlebih dahulu citra retina diubah ke dalam menjadi citra dalam *inverse green channel* [10]. Citra dalam *inverse green channel* dan citra hasil praproses ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Citra retina dalam *inverse green channel* dan citra hasil praproses

2.1.2 Masking

Salah satu metode praproses adalah *masking* yang terdiri dari dua metode yakni *thresholding* dan *shrinking*. *Masking* adalah melakukan segmentasi pada citra sehingga terjadi pemisahan antara objek (*foreground*) dan latar belakang (*background*). Pemisahan ini perlu dilakukan karena *background* dari citra retina tidak benar-benar hitam sehingga memungkinkan dianggap sebagai objek.

Metode yang digunakan adalah metode *thresholding*, dengan nilai batas (*threshold*) yang berbeda untuk setiap citra tergantung pada kondisi kecerahan dan kontras citra. Terlebih dahulu citra retina diubah dari ruang warna RGB ke ruang warna keabuan. Metode *thresholding* kemudian akan mengubah citra keabuan menjadi citra biner. Suatu piksel citra yang skala keabuannya kurang dari *threshold* akan dianggap sebagai *background* yang berwarna hitam. Sedangkan piksel citra yang skala keabuannya lebih dari atau sama dengan *threshold* merupakan objek yang berwarna putih.

Guna memperbaiki FOV (*field of view*) untuk proses selanjutnya, dibutuhkan penyusutan citra biner hasil *masking* (*shrinking*). *Shrinking* dilakukan dengan operasi morfologi erosi [5]. Pada morfologi biner, citra dilihat sebagai himpunan dari ruang Euclidean E untuk setiap dimensi. Ide dasar dari morfologi biner adalah untuk memeriksa citra biner A yang berada pada ruang Euclidean E dengan bentuk lain yang telah didefinisikan B . Erosi dari citra biner A dengan elemen struktur B didefinisikan oleh Persamaan 2.1.

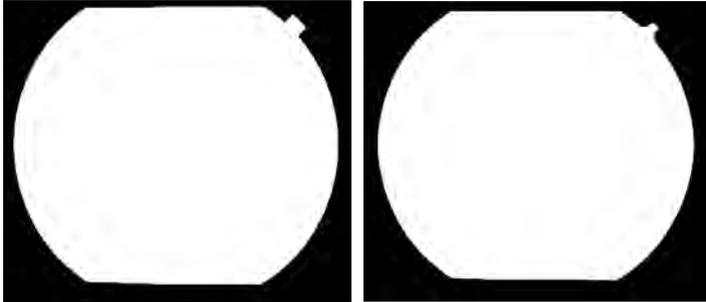
$$A \ominus B = \{z \in E | B_z \subseteq A\} \quad (2.1)$$

Dimana B_z adalah translasi B dengan vector z seperti yang ditunjukkan pada Persamaan 2.2.

$$B_z = \{b + z | b \in B\}, \forall z \in E. \quad (2.2)$$

Ketika elemen struktur B memiliki sebuah pusat, dan pusat tersebut ditempatkan pada asal dari E , maka erosi A oleh B merupakan kedudukan dari titik-titik yang dicapai oleh pusat B ketika B bergerak dalam A [5].

Gambar 2.3 menunjukkan citra hasil dari proses *masking* berserta citra hasil *shrinking*.



Gambar 2. 3 Citra *masking* dan hasil *shrinking*

2.2. Segmentasi Vaskular Retina

Klasifikasi segmen vaskular retina menggunakan bagian pembuluh darah tanpa membutuhkan bagian lain pada retina. Karena data citra yang dijadikan masukan berupa citra retina keseluruhan dengan semua bagian masih ada, maka dibutuhkan suatu metode untuk mengekstrak bagian pembuluh darah dari citra retina keseluruhan. Untuk menghasilkan bagian pembuluh darah dari citra retina dilakukan segmentasi pembuluh darah.

Segmentasi pembuluh darah retina dapat menyediakan sebuah pemetaan dari pembuluh darah di retina. Segmentasi ini dapat memudahkan penilaian karakteristik pembuluh darah yang dibutuhkan pada proses klasifikasi. Segmentasi dapat dilakukan dengan berbagai metode.

Soares dkk menggunakan metode *supervised* dengan menerapkan 2-D *Morlet wavelet* dan *Gaussian mixture model Bayesian classifier*. Transformasi wavelet digunakan untuk

membuat vektor fitur yang membedakan piksel ke dalam kelas piksel vaskular dan piksel bukan vaskular. Kemudian vektor fitur dari beberapa data yang telah diketahui akan dinormalisasi. Hasil vektor fitur tersebut akan diklasifikasikan dengan *Gaussian mixture model* untuk menghasilkan citra segmentasi dengan membagi piksel ke dalam kelas vaskular dan bukan vaskular. [10]. Metode yang ditawarkan akan dijelaskan pada subbab 2.2.1 hingga 2.2.3.

2.2.1 2-D Morlet Wavelet Transform

Suatu bidang nyata $\mathbb{R} \times \mathbb{R}$ dinotasikan sebagai \mathbb{R}^2 dan vektor direpresentasikan sebagai huruf tebal, contoh $\mathbf{x}, \mathbf{b} \in \mathbb{R}^2$. Jika $f \in L^2$ adalah citra direpresentasikan sebagai kuadrat fungsi yang didefinisikan pada \mathbb{R}^2 . Transformasi wavelet kontinyu $T_\psi(\mathbf{b}, \theta, a)$ didefinisikan pada Persamaan 2.3.

$$T_\psi(\mathbf{b}, \theta, a) = C_\psi^{-1/2} \frac{1}{a} \int \psi^*(a^{-1}r_{-\theta}(\mathbf{x} - \mathbf{b}))f(x)d^2\mathbf{x} \quad (2.3)$$

dimana T_ψ , ψ , \mathbf{b} , θ , dan a merupakan konstanta normalisasi, *analyzing wavelet*, vector perpindahan, sudut rotasi, dan parameter dilatasi (skala) secara berturut-turut. ψ^* adalah *complex conjugate* dari ψ [10].

Dari beberapa *analyzing wavelet* yang tersedia digunakan 2-D *Morlet wavelet* karena kemampuan pemilihan arah dalam mendeteksi fitur orientasi dan pengaturan frekuensi tertentu. 2-D *Morlet wavelet* didefinisikan pada Persamaan 2.4.

$$\psi_M(\mathbf{x}) = \exp(j\mathbf{k}_0\mathbf{x})\exp\left(-\frac{1}{2}|\mathbf{A}\mathbf{x}|^2\right) \quad (2.4)$$

dimana $j = \sqrt{-1}$ dan $\mathbf{A} = \text{diag}[\epsilon^{-1/2}, 1]$, $\epsilon \geq 1$ adalah sebuah matriks diagonal dengan ukuran 2×2 yang mendefinisikan *anisotropy* dari filter. *Morlet wavelet* merupakan *complex*

exponential modulated Gaussian, dimana \mathbf{k}_0 adalah vektor frekuensi dari *complex exponential* [10].

Parameter ϵ yang digunakan adalah 8, sehingga elongasi filter dan $\mathbf{k}_0 = [0,3]$. Untuk setiap nilai skala, dipilih nilai modulus maksimum untuk berbagai orientasi yang ditunjukkan pada Persamaan 2.5.

$$M_\psi(b, a) = \max_\theta |T_\psi(b, \theta, a)| \quad (2.5)$$

Transformasi *Morlet wavelet* dihitung untuk θ dari 0 hingga 170 derajat dengan jarak tiap 10 derajat dan diambil nilai maksimumnya [10]. Citra retina hasil 2-D *Morlet Wavelet Transform* ditunjukkan pada Gambar 2.4.



Gambar 2. 4 Citra retina hasil 2-D Morlet Wavelet Transform

2.2.2 Normalisasi Fitur

Diberikan sifat dasar dimensi dari fitur membentuk ruang fitur dari transformasi wavelet, namun fitur ini memungkinkan kesalahan klasifikasi untuk segmentasi. Untuk memperbaiki fitur dilakukan normalisasi dengan Persamaan 2.6.

$$\hat{v}_i = \frac{v_i - \mu_i}{\sigma_i} \quad (2.6)$$

dimana \hat{v}_i adalah nilai fitur ke- i untuk setiap piksel, μ_i adalah rata-rata nilai dari fitur ke- i , dan σ_i standar deviasi dari fitur ke- i . Transformasi normalisasi diaplikasikan secara terpisah untuk ruang fitur masing-masing citra [10].

2.2.3 Gaussian mixture model Bayesian classifier

Gaussian mixture model (GMM) *classifier* merupakan *Bayesian classifier* dimana setiap fungsi densitas probabilitas kelas-kondisional dideskripsikan sebagai kombinasi linier dari fungsi Gaussian. *Bayes classification rule* untuk vektor \mathbf{v} dapat ditunjukkan pada probabilitas posterior pada Persamaan 2.7.

Tentukan C_1 jika $P(C_1|\mathbf{v}) > P(C_2|\mathbf{v})$; selain itu C_2 (2.7)

Dengan aturan Bayes pada Persamaan 2.8.

$$P(C_i|\mathbf{v}) = \frac{p(\mathbf{v}|C_i)p(C_i)}{p(\mathbf{v})} \quad (2.8)$$

dimana $p(\mathbf{v}|C_i)$ adalah fungsi densitas probabilitas kelas-kondisional atau *likelihood*, $P(C_i)$ adalah probabilitas prior dari kelas C_i , dan $p(\mathbf{v})$ fungsi densitas probabilitas dari \mathbf{v} . $P(C_i)$ diestimasi sebagai N_i/N , yaitu rasio dari sampel pada kelas ke- i pada himpunan *training*. *Class likelihood* dideskripsikan sebagai kombinasi linear dari fungsi Gaussian pada Persamaan 2.9.

$$p(\mathbf{v}|C_i) = \sum_{j=1}^{k_i} p(\mathbf{v}|j, C_i)P_j \quad (2.9)$$

dimana k_i adalah jumlah *Gaussian modeling likelihood* i , P_j adalah bobot dari Gaussian j , dan setiap $p(\mathbf{v}|j, C_i)$ adalah distribusi Gaussian dengan d -dimensi. Untuk setiap kelas i , dilakukan estimasi parameter Gaussian k_i dan bobot

menggunakan algoritma *Expectation-Maximization* (EM). Hasil akhir segmentasi ditunjukkan pada Gambar 2.5.



Gambar 2. 5 Citra hasil segmentasi dengan metode yang ditawarkan Soares dkk

Fraz dkk juga menawarkan metode segmentasi dengan menggunakan *First-order Derivative of Gaussian* (FoDoG) yang dikombinasikan dengan bentuk pembuluh darah dan peta orientasi untuk menghasilkan citra segmentasi [11]. Metode yang ditawarkan akan dijelaskan pada subbab 2.2.4 dan 2.2.5.

Variasi level keabuan pada latar belakang citra terlebih dahulu dinormalisasi sehingga menghasilkan citra pembuluh yang diperbaiki. Filter FoDoG diaplikasikan dalam empat orientasi terhadap citra yang diperbaiki. Pixel kandidat yang berupa *centerline* pembuluh kemudian dideteksi menggunakan evaluasi *derivative sign* dan *average derivative value* pada empat citra *matched filter response* (MFR). Pixel *centerline* pembuluh diidentifikasi pada setiap arah untuk pixel kandidat menggunakan perhitungan statistic. Bentuk dan peta orientasi dari vaskular retina diperoleh dengan *bit plane slicing* pada citra yang diperbaiki. *Centerline* pembuluh yang terdeteksi kemudian dikombinasikan dengan bentuk dan peta orientasi untuk menghasilkan citra segmentasi.

2.2.4 Differential Filtering

Filter FoDoG yang digunakan untuk mendeteksi pembuluh darah pada citra retina didefinisikan pada Persamaan 2.10.

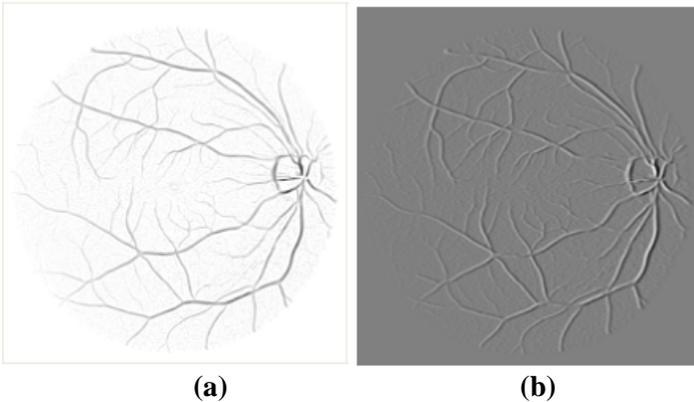
$$f(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \text{ untuk } |x| \leq (t, \sigma), |y| \leq L/2 \quad (2.10)$$

Dan turunan pertama $g(x, y)$ dari $f(x, y)$ didefinisikan pada Persamaan 2.11.

$$g(x, y) = -\frac{x}{\sqrt{2\pi}\sigma^3} \exp\left(-\frac{x^2}{2\sigma^2}\right) \text{ untuk } |x| \leq (t, \sigma), |y| \leq L/2 \quad (2.11)$$

Dimana σ merepresentasikan filter skala, L adalah panjang *neighborhood* pada sumbu y , dan t konstan dan biasanya bernilai 3 karena lebih dari 99% area di bawah kurva Gaussian berada pada interval $[-3\sigma, 3\sigma]$ [11].

Terdapat variasi intensitas bertahap di latar belakang gambar retina dari pinggiran retina menuju daerah makula sentral. Citra pembuluh tersebut diperbaiki dengan menghitung kelengkungan maksimum pokok dan dekomposisi Eigen dari matriks Hessian untuk setiap pixel pada citra retina yang diilustrasikan pada Gambar 2.6 (a). Ketika filter turunan pertama diterapkan ortogonal terhadap orientasi utama pembuluh, maka akan ada nilai-nilai positif pada satu sisi pembuluh dan nilai-nilai negatif di sisi lain. Intensitas lokal maksimum pembuluh umumnya terjadi pada piksel pusat pembuluh, sehingga himpunan piksel yang terhubung sesuai dengan intensitas maksimum pada pembuluh diidentifikasi sebagai piksel *centerline*. Filter kernel FoDoG diterapkan pada citra pembuluh yang diperbaiki dalam empat orientasi 0° , 45° , 90° dan 135° [11]. Citra dengan orientasi 0° ditunjukkan pada Gambar 2.6 (b).



Gambar 2. 6 (a) Citra pembuluh yang diperbaiki (b) Citra hasil filter FoDoG dengan orientasi 0°

Piksel kandidat untuk *centerline* pembuluh akan ditandai dengan mencari kombinasi spesifik dari turunan pada pembuluh serta dari empat buah orientasi filter yang digunakan. Himpunan segmen *centerline* pada suatu direksi tertentu akan menghasilkan citra hasil filter yang ditunjukkan pada gambar 2.7.



Gambar 2. 7 Citra *centreline* pembuluh

2.2.5 Proses Morfologi

Peningkatan kontras dari struktur pembuluh darah terhadap latar belakang citra retina dilakukan dengan menerapkan operator morfologi pada *green channel*. Transformasi *morphological top-hat* ditunjukkan pada Persamaan 2.12 dimana I_{th}^{θ} adalah transformasi citra *top-hat* dengan orientasi *structuring element* pada θ derajat, I adalah citra yang diproses, dan S_e adalah *structuring element* dengan morfologi *opening*, \circ dan θ adalah rotasi angular dari *structuring element* [11].

$$I_{th}^{\theta} = I - (I \circ S_e^{\theta}) \quad (2.12)$$

Kemudian dilakukan *bit plane slicing* yang membagi citra ke dalam bagian tertentu. Pemisahan citra ini berguna untuk analisa kepentingan dari tiap bagian citra. Citra *grayscale* hasil dari operasi *top-hat* dapat digunakan untuk merepresentasikan *bit plane*. Jumlahan dari operasi *top-hat* menghasilkan citra skeleton dari pembuluh darah. Citra hasil akhir segmentasi diperoleh dengan *region growing* yang dimulai dari *centerline* pembuluh dan menggunakan citra skeleton sebagai agregat *threshold* [11]. Hasil akhir segmentasi ditunjukkan pada Gambar 2.8.



Gambar 2. 8 Citra hasil segmentasi dengan metode yang ditawarkan Fraz dkk

Sedangkan metode segmentasi vaskular retina pada Tugas Akhir ini dilakukan dalam tiga tahap. Tahap-tahap tersebut yaitu transformasi wavelet dengan *Isotropic Undecimated Wavelet Transform*, dilanjutkan dengan *thresholding*, dan tahap akhir adalah *cleaning*. Metode ini akan dijelaskan pada subbab 2.2.6 hingga 2.2.8.

2.2.6 *Isotropic Undecimated Wavelet Transform*

Transformasi wavelet yang paling luas digunakan adalah *decimated bi-orthogonal wavelet transform* (DWT). DWT telah berhasil diimplementasikan untuk tujuan kompresi, namun DWT menunjukkan hasil yang kurang optimal untuk aplikasi *filtering*, dekonvolusi, deteksi, ataupun analisis data. Oleh karena itu banyak peneliti yang memilih menggunakan transformasi wavelet kontinyu meskipun terjadi banyak redudansi pada transformasi dan tidak mungkin dilakukan rekonstruksi sempurna. Untuk beberapa aplikasi dimana rekonstruksi diperlukan, beberapa peneliti menggunakan pendekatan *intermediate*, dengan mempertahankan konstruksi *filter bank* dengan algoritma *fast and dyadic*, tetapi mengeliminasi tahap penipisan (*decimation*) pada transformasi wavelet orthogonal [6].

Untuk beberapa citra yang terdiri dari objek isotropik, digunakan *Isotropic Undecimated Wavelet Transform*. Algoritma *Isotropic Undecimated Wavelet Transform* (IUWT) digunakan untuk data citra dimana objek lebih atau kurang isotropik dibandingkan dengan kebanyakan citra lainnya. Kebutuhan yang harus dipenuhi agar dapat menghasilkan analisis yang baik adalah:

- 1) Filter yang digunakan harus simetrik ($\bar{h}[k] = h[k]$ dan $\bar{g}[k] = g[k]$)
- 2) Pada dua dimensi ataupun dimensi yang lebih tinggi, h , g , ψ , ϕ harus mendekati isotropik.

Filter tidak harus ortogonal atau bi-ortogonal sehingga akan menguntungkan ketika ingin mendesain secara bebas. Untuk

alasan komputasional, $h[k, l]$ akan dipisah menjadi $h[k]h[l]$. Pemisahan ini bukan merupakan kondisi yang diharuskan, namun dapat mempercepat kalkulasi yang dibutuhkan pada dataset [6].

IUWT menggunakan Persamaan 2.13 hingga 2.15 untuk analisis *scaling* dan fungsi wavelet:

$$\phi_1(x) = \frac{1}{12} (|x - 2|^3 - 4|x - 1|^3 + 6|x|^3 - 4|x + 1|^3 + |x + 2|^3) \quad (2.13)$$

$$\phi(x, y) = \phi_1(x)\phi_1(y) \quad (2.14)$$

$$\frac{1}{4}\psi\left(\frac{x}{2}, \frac{y}{2}\right) = \phi(x, y) - \frac{1}{4}\phi\left(\frac{x}{2}, \frac{y}{2}\right) \quad (2.15)$$

Dimana $\phi_1(x)$ adalah *spline* dari *order* 3, dan fungsi wavelet didefinisikan sebagai perubahan antara dua resolusi. Filter yang berhubungan yakni h dan g didefinisikan pada Persamaan 2.16 hingga 2.18.

$$h^{(1D)}[k] = \frac{[1, 4, 6, 4, 1]}{16}, k = -2, \dots, 2 \quad (2.16)$$

$$h[k, l] = h^{(1D)}[k]h^{(1D)}[l] \quad (2.17)$$

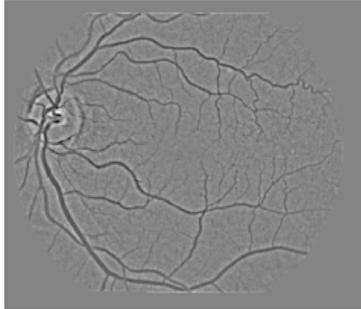
$$g[k, l] = \delta[k, l] - h[k, l] \quad (2.18)$$

Dimana δ didefinisikan sebagai $\delta[0, 0] = 1$ dan $\delta[k, l] = 0$ untuk semua (k, l) berbeda dari $(0, 0)$ [6]. Gambar 2.9 menunjukkan hasil dari citra retina setelah diproses dengan *Isotropic Undecimated Wavelet Transform*.

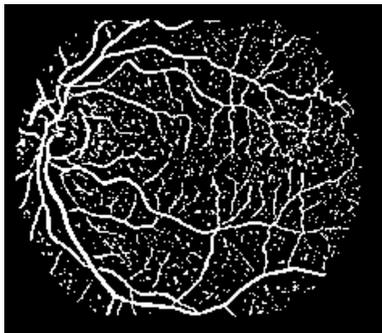
2.2.7 Thresholding

Metode *thresholding* digunakan untuk menghasilkan citra biner vaskular retina dari citra hasil *Isotropic Undecimated Wavelet Transform*. *Thresholding* dimulai dengan menentukan

prosentase piksel yang ingin dipertahankan sebagai segmen vaskular. Jika pada citra hasil wavelet menunjukkan bahwa piksel yang berupa segmen vaskular lebih gelap daripada piksel lainnya, maka piksel dengan intensitas gelap akan dianggap sebagai segmen vaskular sebanyak prosentase yang telah ditentukan. Jika yang terjadi sebaliknya, maka yang dianggap sebagai segmen vaskular adalah piksel dengan intensitas terang. Piksel yang tidak dianggap sebagai segmen vaskular akan dijadikan latar belakang gambar yang berwarna hitam. Sedangkan piksel yang merupakan segmen vaskular dijadikan objek yang berwarna putih [9]. Gambar 2.10 merupakan citra retina yang telah melalui proses *thresholding*.



Gambar 2. 9 Hasil Citra Retina dengan *Two Dimensional Isotropic Undecimated Wavelet Transform*



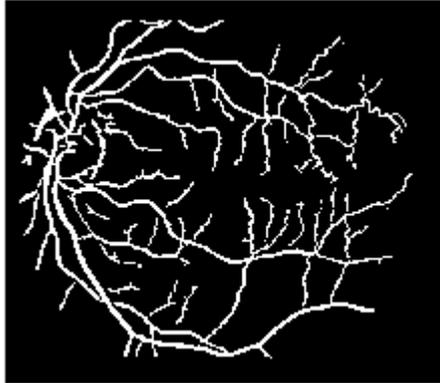
Gambar 2. 10 Hasil citra retina setelah proses *thresholding*

2.2.8 *Cleaning*

Setelah dilakukan *thresholding*, pembuluh darah retina telah berhasil diekstrak dari bagian retina lain dan menghasilkan citra biner yang memisahkan antara pembuluh darah retina dengan *background*. Tetapi citra hasil *thresholding* belum sepenuhnya baik, karena terdapat beberapa objek berukuran kecil yang terlihat seperti derau. Agar citra segmentasi lebih baik sehingga yang tersisa hanyalah objek yang benar-benar merupakan pembuluh darah maka dilakukan *cleaning*.

Cleaning merupakan suatu metode pembersihan agar citra hasil segmentasi menyisakan objek-objek yang merupakan pembuluh darah dan menghapus objek lainnya. Terdapat dua hal yang dilakukan pada proses *cleaning*, yaitu menghapus objek kecil dan mengisi lubang (*hole*) kecil [9].

Sebelum menjalankan proses *cleaning*, tentukan dulu nilai *threshold*. Terdapat dua nilai *threshold* yang dibutuhkan, yaitu *min_object_size* yang menyatakan luas area minimum suatu objek dapat dianggap sebagai pembuluh darah dan *min_holes_size* yang menyatakan luas area minimum dari lubang yang diperbolehkan. Untuk menghapus objek kecil hitung terlebih dahulu area dari setiap objek berwarna putih, yang dianggap sebagai pembuluh darah. Apabila luas area objek tersebut kurang dari *min_object_size*, maka hapus objek tersebut. Sedangkan untuk mengisi lubang, hitung area lubang yang berwarna hitam. Ketika suatu area lubang memiliki luas kurang dari *min_holes_size*, isi lubang tersebut dengan cara mengubah piksel lubang menjadi piksel berwarna putih [9]. Hasil dari proses *cleaning* dapat dilihat pada Gambar 2.11 dan merupakan citra akhir segmentasi yang akan digunakan untuk proses klasifikasi.



Gambar 2. 11 Citra hasil akhir segmentasi

2.3. Ekstraksi Fitur Vaskular Retina

Sebuah sistem membutuhkan beberapa properti untuk membedakan segmen vaskular normal dan abnormal. Vaskular abnormal muncul sebagai kumpulan pembuluh darah yang lebih rapat dan biasanya terbentuk dari pembuluh-pembuluh darah kecil. Tahap ekstraksi segmen vaskular berusaha menemukan pembuluh darah sebanyak-banyaknya sehingga pembuluh darah kecil dan tipis dapat diikutsertakan pada tahap klasifikasi.

Agar segmen vaskular retina dapat diklasifikasikan ke dalam kelas normal ataupun abnormal, dibutuhkan himpunan fitur yang mewakili setiap segmen kandidat. Segmen vaskular retina adalah bagian pembuluh darah tertentu yang berada pada *region of interest* (ROI) dengan luas jendela $w \times w$. ROI berupa citra hasil segmentasi, luas jendela yang digunakan adalah luas sebuah citra hasil segmentasi. Sedangkan segmen kandidat adalah segmen vaskular pada citra hasil segmentasi yang dipertimbangkan untuk ekstraksi fitur atau klasifikasi [7].

Setiap segmen kandidat dianggap sebagai sampel untuk klasifikasi dan direpresentasikan oleh sebuah vektor fitur. Jika sebuah citra retina χ terdiri dari k segmen vaskular kandidat, maka citra retina dapat direpresentasikan sebagai $\chi =$

$\{v_1, v_2, v_3, \dots, v_k\}$ dimana v_j adalah vektor fitur untuk segmen kandidat ke- j dengan m fitur, sehingga $v = \{x_1, x_2, x_3, \dots, x_m\}$. Fitur-fitur yang digunakan untuk klasifikasi segmen vaskular normal dan abnormal dari citra biner hasil segmentasi adalah *area* (x_1), *energy* (x_2), *mean gradient* (x_3), *standard deviation gradient* (x_4), *mean intensity* (x_5), *intensity variation* (x_6), dan *vessel segment* (x_7) [7].

2.3.1 Area

Area yaitu jumlah piksel yang merupakan pembuluh darah di dalam jendela. Nilai *area* diperoleh dengan menghitung piksel yang bernilai 1, karena pada citra biner hasil segmentasi, pembuluh darah digambarkan dengan piksel berwarna putih. Pembuluh darah normal lebih tebal dibandingkan pembuluh darah abnormal, sehingga *area* pembuluh darah normal lebih luas dibandingkan pembuluh darah abnormal [7].

2.3.2 Energy

Energy yaitu intensitas dari piksel pembuluh darah pada segmen kandidat. Nilai *energy* didapatkan dengan menghitung jumlah intensitas *green channel* dari semua piksel yang merupakan pembuluh darah. Pembuluh darah normal cenderung lebih terang dibandingkan dengan pembuluh darah abnormal pada *inverse green channel* citra [7].

2.3.3 Mean Gradient

Mean gradient adalah nilai rata-rata *gradient magnitude* dari piksel pembuluh darah pada segmen kandidat. *Gradient* dari sebuah citra dengan menghitung perubahan intensitas dari setiap titik pada citra *gradient* terhadap titik yang sama pada citra asli dalam arah tertentu. *Gradient* dari citra memiliki dua komponen arah, yaitu (x,y) dimana fungsi *gradient* dari (x,y) adalah turunan

parsial pada arah x dan y . *Gradient magnitude* didefinisikan sebagai $\nabla f = \frac{\partial f}{\partial x} \hat{x} + \frac{\partial f}{\partial y} \hat{y}$ dengan $\frac{\partial f}{\partial x}$ adalah *gradient* pada arah x dan $\frac{\partial f}{\partial y}$ adalah *gradient* pada arah y .

Dalam kasus ini, nilai *green channel* dari citra retina yang akan dihitung dengan *gradient magnitude*. *Mean gradient* dihitung dengan menjumlahkan semua nilai *gradient magnitude* pada setiap piksel kemudian dibagi dengan luas jendela [7].

2.3.4 *Standard Deviation Gradient*

Standard deviation gradient adalah standar deviasi dari *gradient magnitude* dari piksel pembuluh darah pada segmen kandidat. *Standard deviation gradient* merupakan akar kuadrat dari hasil penjumlahan kuadrat selisih *mean gradient* dengan nilai *gradient magnitude* tiap piksel dibagi dengan luas jendela [7].

2.3.5 *Mean Intensity*

Mean intensity adalah normalisasi dari nilai *mean intensity* piksel pembuluh darah pada segmen kandidat. Sebelumnya dari citra asli diambil nilai intensitas pada *green channel*. Kemudian dihitung *mean intensity* dengan membagi jumlah nilai intensitas dari tiap piksel dengan luas jendela. *Standard deviation intensity* ditentukan dengan menghitung akar kuadrat dari jumlah kuadrat selisih *mean intensity* dan nilai intensitas tiap piksel dibagi dengan luas jendela. Normalisasi intensitas diperoleh dengan operasi pengurangan *mean intensity* dengan nilai intensitas tiap piksel, kemudian dibagi dengan *standard deviation intensity*. Hasil akhir dari *mean intensity* adalah rata-rata dari normalisasi intensitas [7].

2.3.6 *Intensity Variation*

Intensity variation adalah perbandingan antara *mean intensity* dan *standard deviation intensity* pada segmen kandidat yang telah diperoleh sebelumnya pada proses penghitungan *mean intensity* [7].

2.3.7 *Vessel Segment*

Vessel segment adalah jumlah segmen pembuluh darah pada kandidat segmen. Jumlah segmen ditentukan dengan menggunakan *connected component analysis*. *Connected component* bekerja dengan memeriksa apakah piksel tetangga memiliki intensitas yang sama dengan piksel tersebut. Jika piksel tetangga memiliki intensitas yang sama, maka piksel tetangga tersebut akan dilabeli sebagai piksel yang terdapat pada *region* dengan piksel tersebut. Pembuluh darah abnormal biasanya memiliki jumlah segmen yang lebih banyak dibandingkan pembuluh darah normal [7].

2.4. **Klasifikasi Vaskular Normal dan Abnormal dengan M-Mediods Multivariat**

Klasifikasi adalah sebuah algoritma *supervised* untuk menentukan kelas atau label dari sebuah data berdasarkan data-data lain yang sudah diketahui kelasnya. Algoritma klasifikasi dimulai dengan memodelkan segmen vaskular normal dan abnormal berdasarkan data yang sudah diketahui kelasnya. Kemudian model yang dibuat tersebut akan dipakai untuk mengenali segmen vaskular yang belum diketahui kelasnya.

Salah satu metode klasifikasi yang ada adalah metode m-Mediods Multivariat. Algoritma ini terdiri dari dua bagian, yaitu perbaikan representasi ruang fitur dengan menggunakan *Local Fisher Discriminant Analysis* (LFDA) serta pemodelan dan pengenalan dengan menggunakan *Learning Vector Quantization* (LVQ).

2.4.1 Local Fisher Discriminant Analysis

Local Fisher Discriminant Analysis (LFDA) merupakan pendekatan klasifikasi yang ditawarkan untuk meningkatkan representasi ruang fitur dengan mengembangkan transformasi *supervised*. Peningkatan representasi ruang fitur ini dilakukan dengan meningkatkan kesamaan antar data atau objek pada satu kelas (*intra class similarity*) dan mengurangi kesamaan antar objek yang terdapat pada kelas berbeda (*inter class similarity*). Aplikasi dari *supervised* LFDA akan mengidentifikasi arah utama pada ruang fitur yang menghasilkan diskriminasi maksimum antara kelas yang berbeda [7].

Jika $DB = \{v_1, v_2, \dots, v_n\}$ adalah himpunan n sampel pelatihan yang memiliki c kelas pembuluh darah $\{Normal, Abnormal\}$. *Scatter matrix* dalam kelas (*within*) dan antarkelas (*between*) masing-masing dihitung dengan Persamaan 2.19 dan Persamaan 2.20.

$$\mathfrak{K}_w = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n W_{i,j}^w (\|v_i, v_j\|) \quad (2.19)$$

$$\mathfrak{K}_b = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n W_{i,j}^b (\|v_i, v_j\|) \quad (2.20)$$

Dimana $\|.,.\|$ adalah fungsi jarak *Euclidean* dan $W_{i,j}^w$ serta $W_{i,j}^b$ akan dijelaskan pada Persamaan 2.21 dan Persamaan 2.22.

$$W_{i,j}^w = \begin{cases} \exp\left(\frac{\|v_i, v_j\|^2}{\varsigma_i \varsigma_j}\right) * \frac{1}{n_k} & \text{iff } v_i \wedge v_j \in C_k \\ 0 & \text{otherwise} \end{cases} \quad (2.21)$$

$$W_{i,j}^b = \begin{cases} \exp\left(\frac{\|v_i, v_j\|^2}{\varsigma_i \varsigma_j}\right) * \left(\frac{1}{n} - \frac{1}{n_k}\right) & \text{iff } v_i \wedge v_j \in C_k \\ \frac{1}{n} & \text{otherwise} \end{cases} \quad (2.22)$$

Dengan n_k adalah jumlah keanggotaan (*membership count*) dari kelas C_k dan ζ_i adalah jarak rata-rata dari sampel v_i dengan k tetangga terdekat. Nilai dari k ditentukan secara empirik dan diatur untuk $k=7$ [7].

Representasi ruang koefisien berdasarkan LFDA dihasilkan dengan melakukan generalisasi dekomposisi nilai eigen dari $\mathfrak{R}_b E = \lambda \mathfrak{R}_w E$ dimana λ adalah generalisasi nilai eigen dan E adalah vektor eigen. Peningkatan kualitas dari representasi ruang fitur segmen pembuluh darah dengan *localized Fisher discriminant direction* diperoleh dengan menggunakan persamaan $F = \{E_1, E_2, \dots, E_m\}$ dimana $\{E_1, E_2, \dots, E_m\}$ adalah vektor eigen yang disusun mulai dari nilai terbesar menuju nilai terkecil (*descending*) dengan nilai eigen $\{\lambda_1, \lambda_2, \dots, \lambda_m\}$ [7]. Representasi ruang fitur yang telah diperbaiki digunakan untuk membangun model dari pola pembuluh darah yang berbeda dan melakukan klasifikasi menggunakan model yang telah dibangun.

2.4.2 Learning Vector Quantization

Learning Vector Quantization (LVQ) adalah salah satu varian dari *neural network* yang mengombinasikan pembelajaran kompetitif dengan supervisi. LVQ dapat digunakan untuk klasifikasi dengan melakukan pemodelan serta pengenalan [8].

Pemodelan dengan pembuatan sebuah himpunan *training* yang terdiri dari pasangan Q vektor *training* dan target *output* yang berupa kelas dari objek atau data, yaitu

$$\{s^{(q)}: t^{(q)}\}, \quad q = 1, 2, \dots, Q,$$

dimana $s^{(q)}$ adalah vektor *training* dengan N dimensi, dan $t^{(q)}$ adalah vektor target *output* dengan M dimensi. M sama dengan jumlah kelas dan M harus lebih kecil daripada Q [8]. Vektor target didefinisikan pada Persamaan 2.23.

$$t_i^{(q)} = \begin{cases} 1, & \text{jika } s^{(q)} \text{ terdapat pada kelas } i \\ 0, & \text{jika } s^{(q)} \text{ selain kelas } i \end{cases} \quad (2.23)$$

LVQ terdiri atas lapisan (*layer*) kompetitif, termasuk *subnet* kompetitif dan *layer* linier. Pada *layer* pertama atau dapat disebut *hidden layer*, setiap *neuron* diberikan kepada sebuah kelas. Beberapa *neuron* pada *layer* pertama dapat berada pada kelas yang sama. Setiap *neuron* pada *layer* pertama yang memiliki kelas yang sama tersebut akan dipetakan sebuah *neuron* pada *layer* kedua. Jumlah *neuron* pada *layer* pertama, Q , setidaknya memiliki jumlah yang sama atau lebih besar dari jumlah *neuron* pada *layer* kedua, M .

Pada *layer* kompetitif, *neuron* pada *layer* pertama akan melakukan pembelajaran kepada vektor prototip yang memperbolehkan untuk melakukan klasifikasi sebuah region dari ruang input. Kedekatan antara vektor input dan setiap vector bobot dihitung menggunakan jarak *Euclidean* terdekat [8]. Sebuah *subnet* digunakan untuk mencari elemen terkecil dari input net yang ditunjukkan pada Persamaan 2.24.

$$n^{(1)} = \begin{bmatrix} \|x - W_{.1}^{(1)}\| \\ \|x - W_{.2}^{(1)}\| \\ \dots \\ \|x - W_{.Q}^{(1)}\| \end{bmatrix} \quad (2.24)$$

Dan himpunan yang memiliki nilai elemen output bernilai 1, menandakan bahwa vektor input terdapat pada suatu kelas, dan untuk lainnya isi dengan nilai 0. Subnet tersebut kemudian direpresentasikan sebagai nilai vektor dengan fungsi vektor pada Persamaan 2.25.

$$a^{(1)} = \text{compet}(n^{(1)}) \quad (2.25)$$

Beberapa *neuron* yang memiliki kelas sama pada *layer* pertama dianggap sebagai sub kelas. *Layer* kedua dari jaringan LVQ kemudian digunakan untuk mengombinasikan sub kelas ke dalam sebuah kelas tunggal. Hal ini dilakukan dengan

menggunakan matriks bobot $W^{(2)}$ yang memiliki elemen seperti ditunjukkan pada Persamaan 2.26.

$$w_{ij} = \begin{cases} 1, & \text{jika neuron ke } - i \text{ adalah sub kelas dari } j \\ 0, & \text{jika neuron ke } - i \text{ bukan sub kelas dari } j \end{cases} \quad (2.26)$$

Ketika $W^{(2)}$ telah ditetapkan, maka $W^{(2)}$ tidak akan diubah. Selain itu bobot $W^{(1)}$ pada *layer* kompetitif akan dimodelkan dengan aturan Kohonen LVQ.

Setiap iterasi pada vektor *training* pada jaringan LVQ diwakili oleh input x , dan jarak *Euclidean* dari vektor input untuk setiap vector prototip dihitung. Setiap *neuron* pada *layer* pertama akan melakukan kompetisi, *neuron* yang akan menjadi pemenang j^* adalah *neuron* yang memiliki jarak *Euclidean* terkecil. Kemudian elemen dari j^* akan diberikan nilai 1 sedangkan *neuron* lainnya bernilai 0. Aktivasi $a^{(1)}$ akan dikalikan dengan $W^{(2)}$ untuk dijadikan input pada jaringan $n^{(2)}$. Sehingga hasil akhir dari jaringan adalah $a^{(2)} = n^{(2)}$, karena fungsi transfer dari *neuron* output adalah fungsi identitas. $a^{(2)}$ juga hanya akan memiliki satu elemen yang tidak bernilai nol yaitu k^* , yang menandakan vektor input terdapat pada kelas k^* [8].

Aturan Kohonen digunakan untuk meningkatkan bobot dari *layer* pertama. Jika x diklasifikasikan dengan benar, maka vektor bobot $w_{j^*}^{(1)}$ dari *neuron* pemenang pada *layer* pertama bergerak menuju x . Tetapi jika x diklasifikasikan salah, maka jelas bahwa *neuron* pemenang yang memenangkan kompetisi adalah salah [8]. Aturan ini akan dijelaskan pada Persamaan 2.27 dan Persamaan 2.28.

$$\Delta w_{j^*}^{(1)} = \alpha (x - w_{j^*}^{(1)}) \quad \text{jika } a_{k^*}^{(2)} = t_{k^*} = 1 \quad (2.27)$$

$$\Delta w_{j^*}^{(1)} = -\alpha (x - w_{j^*}^{(1)}) \quad \text{jika } a_{k^*}^{(2)} = 1 = t_{k^*} \quad (2.28)$$

Pada metode klasifikasi *m*-Mediods multivariat jumlah *neuron* pada *layer* pertama atau *hidden layer* mewakili jumlah *mediods*. Sehingga *mediods* juga akan mewakili jumlah sub kelas dari kelas yang ada. *Hidden layer* pada jaringan LVQ berguna untuk melakukan *clustering* kedalam berbagai sub kelas sebelum nantinya objek atau data diklasifikasikan berdasarkan kelas yang ada.

BAB III PERANCANGAN

Pada bab ini akan dijelaskan perancangan perangkat lunak pengklasifikasi segmen vaskular retina. Perancangan akan dibagi menjadi tiga proses utama yaitu praproses citra dan segmentasi vaskular retina, proses ekstraksi fitur dari segmen vaskular retina, serta klasifikasi segmen vaskular normal dan abnormal. Untuk masing-masing proses utama akan dibagi menjadi proses-proses kecil yang terlibat didalamnya. Sebelumnya pada bab ini akan dijelaskan gambaran umum setiap program utama dalam bagan gambaran umum metode, selanjutnya untuk penjelasan lebih detail akan disajikan dalam *pseudocode*.

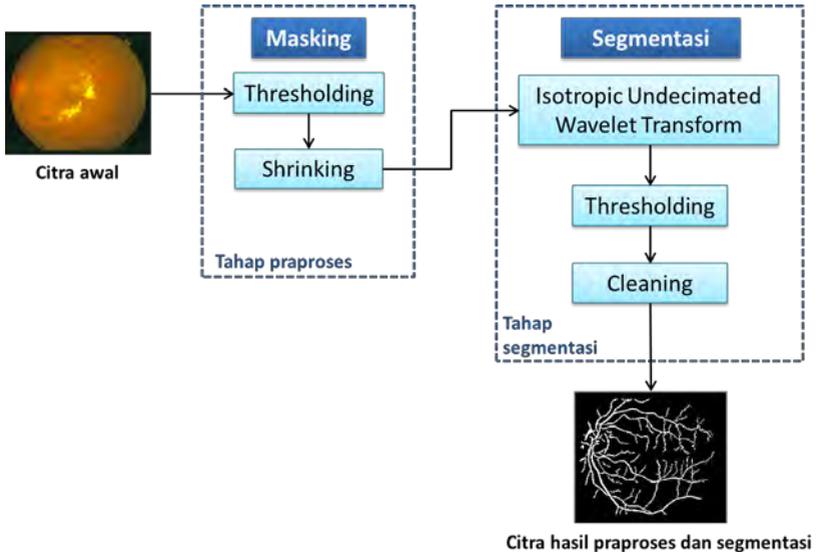
3.1. Perancangan Praproses dan Segmentasi Vaskular

Salah satu proses utama yang ada pada aplikasi ini adalah praproses dan segmentasi vaskular citra retina. Gambaran umum dari praproses dan segmentasi citra retina digambarkan pada Gambar 3.1.

Citra masukan akan diolah ke dalam pra proses yaitu proses *masking*. Pada *masking*, citra masukan terlebih dahulu dilakukan *thresholding*. Hasil citra *thresholding* kemudian diproses pada tahap *shrinking* untuk menghasilkan citra *masking* yang sudah disusutkan.

Citra *masking* yang sudah disusutkan dan citra asli retina dalam skala keabuan akan digunakan pada tahap segmentasi. Tahap segmentasi dimulai dengan melakukan transformasi wavelet pada citra masukan menggunakan *Isotropic Undecimated Wavelet Transform* sehingga menghasilkan citra pada skala keabuan dengan bagian pembuluh darah yang lebih jelas. Citra hasil transformasi wavelet selanjutnya diproses menggunakan *thresholding* untuk menghasilkan citra biner yang menunjukkan pembuluh darah retina. Pada akhirnya citra dilakukan *cleaning*

sehingga dihasilkan citra segmentasi yang lebih bersih dan menyisahkan bagian pembuluh darah retina saja.



Gambar 3. 1 Gambaran umum praproses dan segmentasi citra

Penjelasan praproses dan proses segmentasi citra retina mulai dijelaskan lebih detail dalam bentuk *pseudocode*. *Pseudocode* proses ini terdapat satu program utama yang bertujuan untuk memanggil secara keseluruhan program-program lain yang lebih kecil yang berisi proses-proses yang lebih kecil untuk dijalankan. Pada subbab-subbab berikutnya akan dijelaskan *pseudocode* masing-masing program. Tabel 3.1 sampai Tabel 3.4 berisi variabel-variabel yang akan digunakan sedangkan Tabel 3.5 dan Tabel 3.6 berisi fungsi-fungsi yang akan dipakai pada *pseudocode* program.

Tabel 3. 1 Daftar Variabel yang Digunakan Pada *Pseudocode* Praproses dan Perbaikan Kontras Citra Retina (Bagian Pertama)

No.	Nama Variabel	Tipe	Penjelasan
1.	area_holes	double	Menjelaskan luas area yang merupakan lubang hitam di sekitar objek berwarna putih pada citra biner segmentasi.
2.	area_objects	double	Menjelaskan luas area yang merupakan objek pembuluh darah berwarna putih di sekitar area berwarna hitam pada citra biner segmentasi.
3.	b3	int	Koefisien <i>spline</i> untuk filter pada <i>Isotropic Undecimated Wavelet Transform</i> .
4.	bw_mask	uint8	Citra biner yang merupakan hasil dari proses <i>masking</i> dengan ukuran 650 x 700 piksel.
5.	bw_segmented	uint8	Citra biner hasil akhir dari proses segmentasi yang berukuran 650 x 700 piksel.
6.	bw_wavelet	uint8	Citra dalam skala keabuan yang merupakan hasil dari transformasi wavelet dengan <i>Isotropic Undecimated Wavelet Transform</i> yang berukuran 650 x 700 piksel.
7.	cc_objects	struct	Berupa <i>struct</i> yang berisi atribut-atribut hasil dari analisis <i>connected component</i> .

Tabel 3. 2 Daftar Variabel yang Digunakan Pada *Pseudocode* Praproses dan Perbaikan Kontras Citra Retina (Bagian Kedua)

No.	Nama Variabel	Tipe	Penjelasan
8.	cc_objects .PixelIdx List	cell	Menyimpan indeks dari objek yang ditemukan dengan menggunakan analisis <i>connected component</i> .
9.	d	int	Nilai maksimum dari jumlahan semua nilai bw_mask baik dalam satu kolom maupun satu baris.
10.	d1	int	Nilai maksimum dari jumlahan semua nilai bw_mask dalam satu kolom.
11.	d2	int	Nilai maksimum dari jumlahan semua nilai bw_mask dalam satu baris.
12.	final_mask	uint8	Citra biner yang merupakan hasil <i>masking</i> yang telah dilakukan <i>shrinking</i> dengan ukuran 650 x 700 piksel.
13.	h	Cell	Filter kernel yang digunakan pada <i>Isotropic Undecimated Wavelet Transform</i> .
14.	h2	cell	Nilai sementara dari perluasan filter kernel yang digunakan pada <i>Isotropic Undecimated Wavelet Transform</i> .
15.	im	uint8	Citra asli dalam ruang warna RGB berukuran 650 x 700 piksel.

Tabel 3. 3 Daftar Variabel yang Digunakan Pada *Pseudocode* Praproses dan Perbaikan Kontras Citra Retina (Bagian Ketiga)

No.	Nama Variabel	Tipe	Penjelasan
16.	Img	uint8	Citra asli yang telah diubah ke dalam ruang warna skala keabuan dengan ukuran 650 x 700 piksel.
17.	Ind	int	Variabel untuk memeriksa <i>level</i> dari wavelet.
18.	inds	logical	Variabel akan menunjukkan nilai 1 jika luas area lubang kurang dari <code>iuwt_px_fill</code> , dan akan bernilai 0 jika sebaliknya.
19.	iuwt_dark	logical	Variabel ini akan bernilai <i>true</i> jika piksel pembuluh darah lebih gelap dibandingkan piksel-piksel di sekitarnya.
20.	iuwt_inpainting	logical	Variabel ini akan bernilai <i>true</i> jika piksel diluar daerah yang didefinisikan oleh <code>bw_mask</code> seharusnya diisi dengan nilai piksel terdekat pada citra asli.
21.	iuwt_px_fill	double	Nilai minimum suatu lubang (<i>hole</i>) untuk dipertahankan agar tidak diisi.
22.	iuwt_px_remove	double	Nilai minimum suatu objek untuk tetap dipertahankan agar tidak dihapus.
23.	iuwt_w_levels	int	Berupa vektor numerik yang menunjukkan nilai <i>default</i> dari tingkat wavelet yang akan digunakan.

Tabel 3. 4 Daftar Variabel yang Digunakan Pada *Pseudocode* Praproses dan Perbaikan Kontras Citra Retina (Bagian Keempat)

No.	Nama Variabel	Tipe	Penjelasan
24.	<code>iuwt_w_thresh</code>	double	Nilai <i>threshold</i> yang digunakan untuk diaplikasikan pada wavelet yang menunjukkan proporsi objek citra yang dipertahankan.
25.	<code>levels</code>	int	Berupa vektor numerik yang menunjukkan nilai dari tingkat wavelet yang digunakan.
26.	<code>padding</code>	char	Menunjukkan <i>padding</i> yang digunakan untuk filter pada <i>Isotropic Undecimated Wavelet Transform</i> .
27.	<code>s_in</code>	int	Berupa nilai dari citra asli yang telah dilakukan presisi <i>single</i> .
28.	<code>s_out</code>	int	Berupa <i>array</i> yang telah dihaluskan yang berhubungan dengan transformasi wavelet.
29.	<code>siz</code>	int	Nilai maksimum dari angka 5 dan hasil pembulatan ($d*0.025$) yang digunakan untuk pembuatan matriks berisi angka 1 pada tahap <i>shrinking</i> .
30.	<code>spacing</code>	int	Nilai yang digunakan untuk mengatur jarak untuk setiap koefisien pada kernel filter.
31.	<code>threshold</code>	double	Nilai ambang batas <i>masking</i> .
32.	<code>w</code>	int	Berupa <i>array</i> yang memiliki satu dimensi lebih banyak daripada variabel <code>img</code> .
33.	<code>wclass</code>	char	Menunjukkan kelas dari wavelet yang digunakan.

Tabel 3. 5 Daftar Fungsi yang Digunakan Pada *Pseudocode* Praproses dan Perbaikan Kontras Citra Retina (Bagian Pertama)

No.	Nama Fungsi	Penjelasan
1.	<code>bwconncomp</code>	Fungsi untuk menghitung jumlah objek yakni berupa piksel-piksel yang saling berhubungan pada suatu citra.
2.	<code>cellfun</code>	Fungsi untuk mengaplikasikan suatu fungsi untuk setiap <i>cell</i> pada suatu <i>array</i> .
3.	<code>cell2mat</code>	Fungsi untuk mengubah setiap nilai dalam <i>cell</i> pada suatu <i>array</i> dengan matriks tunggal.
4.	<code>dilate_wavelet_kernel</code>	Fungsi untuk melakukan dilasi pada kernel dengan memasukkan <i>spacing</i> di antara koefisien kernel filter.
5.	<code>dlg_clean_segmented</code>	Fungsi untuk menjalankan proses <i>cleaning</i> untuk perbaikan hasil segmentasi.
6.	<code>dlg_iuwt</code>	Fungsi untuk menjalankan proses transformasi wavelet.
7.	<code>double</code>	Fungsi untuk mengubah suatu nilai ke dalam presisi <i>double</i> .
8.	<code>find</code>	Fungsi untuk menemukan indeks suatu nilai yang bukan nol.
9.	<code>im2bw</code>	Fungsi untuk melakukan proses <i>thresholding</i> sehingga dapat dihasilkan suatu citra biner.
10.	<code>imerode</code>	Fungsi untuk melakukan operasi morfologi rotasi pada citra biner.

Tabel 3. 6 Daftar Fungsi yang Digunakan Pada *Pseudocode* Praproses dan Perbaikan Kontras Citra Retina (Bagian Kedua)

No.	Nama Fungsi	Penjelasan
11.	<code>imfilter</code>	Fungsi untuk melakukan n-dimensi filter pada suatu citra.
12.	<code>isscalar</code>	Fungsi untuk memeriksa apakah suatu masukan merupakan skalar.
13.	<code>isvector</code>	Fungsi untuk memeriksa apakah suatu nilai masukan merupakan vektor.
14.	<code>length</code>	Fungsi untuk menghitung panjang suatu vektor.
15.	<code>mask</code>	Fungsi untuk menjalankan proses <i>masking</i> .
16.	<code>max</code>	Fungsi untuk menemukan nilai maksimum dari kumpulan nilai yang ada.
17.	<code>numel</code>	Fungsi untuk menghitung jumlah elemen dari suatu <i>array</i> .
18.	<code>rgb2gray</code>	Fungsi untuk mengubah citra dalam ruang warna RGB menjadi citra dalam skala keabuan.
19.	<code>single</code>	Fungsi untuk mengubah nilai pada suatu vektor sehingga presisinya <i>single</i> .
20.	<code>size</code>	Fungsi untuk mencari ukuran dari suatu <i>array</i> .
21.	<code>squeeze</code>	Fungsi untuk menghapus dimensi tunggal.
22.	<code>zeros</code>	Fungsi untuk membuat matriks yang terdiri dari angka nol.

3.1.1 Program Utama

Program utama merupakan program yang memanggil fungsi-fungsi lain untuk menjalankan program secara keseluruhan. Proses-proses yang dilakukan dalam algoritma yang digunakan dipisahkan dalam fungsi-fungsi yang berbeda untuk memudahkan pengecekan program setiap prosesnya. *Pseudocode* program utama ditunjukkan pada Gambar 3.2.

Masukan	Citra retina dalam ruang warna RGB (variabel <i>im</i>)
Keluaran	Citra hasil akhir segmentasi (variabel <i>bw_segmented</i>)
<pre> 1. <i>img</i> ← <i>rgb2gray</i>(<i>im</i>) 2. <i>final_mask</i> ← <i>mask</i>(<i>img</i>) 3. <i>iuwt_dark</i> ← <i>true</i> 4. <i>iuwt_inpainting</i> ← <i>false</i> 5. <i>iuwt_w_levels</i> ← 2:3 6. <i>iuwt_w_thresh</i> ← 0.2 7. <i>iuwt_px_remove</i> ← 0.05 8. <i>iuwt_px_fill</i> ← 0.05 9. [<i>bw_wavelet</i>,<i>iuwt_dark</i>,<i>iuwt_w_levels</i>,<i>iuwt_w_thresh</i>] ←<i>dlg_iuwt</i>(<i>im</i>,<i>bw_mask</i>,<i>iuwt_dark</i>,<i>iuwt_w_levels</i>,<i>iuwt_w_thresh</i>,<i>iuwt_inpainting</i>) 10. [<i>bw_segmented</i>,<i>iuwt_px_remove</i>,<i>iuwt_px_fill</i>] ←<i>dlg_clean_segmented</i>(<i>bw_wavelet</i>,<i>iuwt_px_remove</i>,<i>iuwt_px_fill</i>,<i>bw_mask</i>) </pre>	

Gambar 3. 2 *Pseudocode* program utama

3.1.2 Masking

Program *masking* merupakan program yang melakukan pemisahan antara objek retina dengan latar belakang gambar (*background*). *Masking* terdiri dari dua proses yaitu *thresholding* dan *shrinking* yang menerapkan metode yang telah dijelaskan pada subbab 2.1.2. *Pseudocode* proses *masking* ditunjukkan pada Gambar 3.3.

Masukan	Citra retina dalam ruang warna RGB (variabel <i>im</i>)
Keluaran	Citra hasil akhir <i>masking</i> (variabel <i>final_mask</i>)
<ol style="list-style-type: none"> 1. $bw_mask \leftarrow im2bw(im, threshold)$ 2. $d1 \leftarrow \max(\text{sum}(bw_mask))$ 3. $d2 \leftarrow \max(\text{sum}(bw_mask, 2))$ 4. $d \leftarrow \max(d1, d2)$ 5. $siz \leftarrow \max(5, \text{round}(d * .025))$ 6. $final_mask \leftarrow \text{imerode}(bw_mask, \text{ones}(siz))$ 	

Gambar 3. 3 Pseudocode program *masking*

3.1.3 Transformasi Wavelet

Program transformasi wavelet mengaplikasikan algoritma *Isotropic Undecimated Wavelet Transform* serta melakukan *thresholding* untuk menghasilkan citra biner yang berupa pembuluh darah retina. *Pseudocode* ini mengaplikasikan metode yang telah dijelaskan pada subbab 2.2.6 dan subbab 2.2.7. *Pseudocode* proses transformasi wavelet ditunjukkan pada Gambar 3.4 serta fungsi dilasi kernel pada Gambar 3.5.

Masukan	Citra retina dalam ruang warna skala keabuan (variabel <i>img</i>), level wavelet (variabel <i>iuwt_w_levels</i>)
Keluaran	Level dari wavelet untuk gambar 2D (variabel <i>w</i>), array yang telah dihaluskan yang berhubungan dengan transformasi wavelet (variabel <i>s_out</i>)
<ol style="list-style-type: none"> 1. $padding \leftarrow 'symmetric'$ 2. If $\text{numel}(img) > 10000000$ 3. $wclass \leftarrow 'single'$ 4. $s_in \leftarrow \text{single}(img)$ 	

Gambar 3. 4 Pseudocode proses transformasi wavelet (bagian pertama)

```

5.  elseif
6.      wclass ← 'double'
7.      s_in ← double(img)
8.  end if
9.  w←zeros([size(img),length(levels)],wclass)
10. b3 ← [1 4 6 4 1] / 16
11. for ii←1 to levels(end)
12.     h←dilate_wavelet_kernel(b3,2^(ii-1)-1)
13.     s_out←imfilter(s_in,h'* h,padding)
14.     ind←find(levels←ii)
15.     If isscalar(ind)
16.         w(:, :, ind) ← s_in - s_out
17.     end if
18.     s_in ← s_out
19. end for
20. w ← squeeze(w)
21. function h2←
22. dilate_wavelet_kernel(h,spacing)
23. If ~isvector(h) && ~isscalar(spacing)
24.     error(['Invalid input to DILATE_
25.         WAVELET_KERNEL:'H must be a vector and
26.         SPACING must be a scalar'])
27. end if
28. h2←zeros(1,numel(h)+spacing*(numel(h)-1))
29. If size(h,1)>size(h,2)
30.     h2 ← h2'
31. end if
32. h2(1:spacing+1:end) ← h

```

Gambar 3.5 Pseudocode proses transformasi wavelet (bagian kedua)

3.1.4 *Cleaning*

Program *cleaning* mengaplikasikan proses *cleaning* yakni menghapus objek pembuluh darah berukuran kecil serta mengisi lubang berukuran kecil. *Pseudocode* ini mengaplikasikan metode yang telah dijelaskan pada subbab 2.2.8. *Pseudocode* proses *cleaning* wavelet ditunjukkan pada Gambar 3.6.

Masukan	Citra hasil transformasi wavelet (variabel <code>bw_wavelet</code>), minimum luas area objek (variabel <code>iuwt_px_remove</code>), minimum luas lubang (variabel <code>iuwt_px_fill</code>)
Keluaran	Citra hasil akhir segmentasi (variabel <code>bw_segmented</code>)
<pre> 1. if iuwt_px_remove > 0 2. cc_objects ← bwconncomp(bw_wavelet) 3. area_objects ← cellfun('size', 4. cc_objects.PixelIdxList, 1) 5. bw_segmented ← false(size(bw)) 6. inds ← area_objects >= iuwt_px_remove 7. bw_segmented(cell2mat(cc_objects. 8. PixelIdxList(inds))) ← true 9. else 10. bw_segmented ← bw_wavelet; 11. end if 12. if iuwt_px_fill > 0 13. cc_holes ← bwconncomp(~bw_clean) 14. area_holes ← cellfun('size', 15. cc_holes.PixelIdxList, 1) 16. inds ← area_holes < iuwt_px_fill 17. bw_segmented(cell2mat 18. (cc_holes.PixelIdxList(inds))) ← true; 19. end if </pre>	

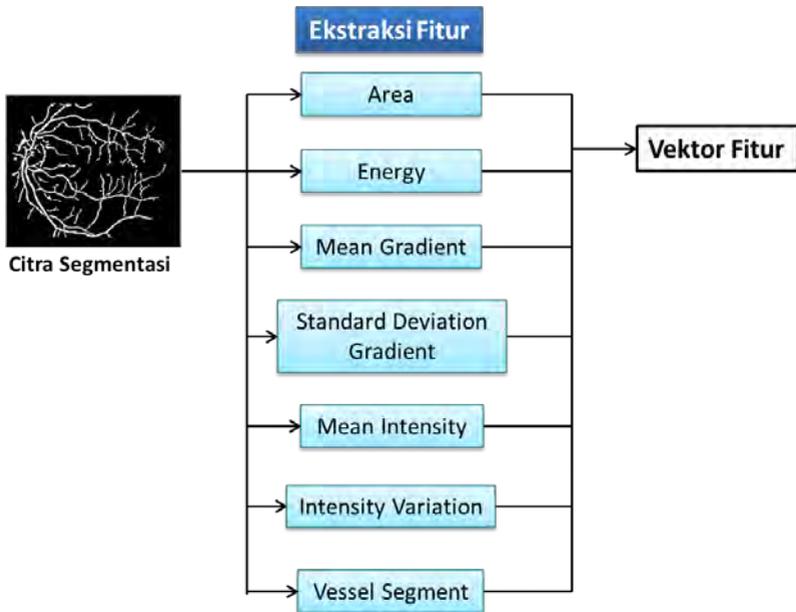
Gambar 3. 6 Pseudocode program *cleaning*

3.2. Perancangan Ekstraksi Fitur Segmen Vaskular Retina

Salah satu proses utama lainnya yang ada pada aplikasi ini adalah ekstraksi fitur segmen vaskular retina. Citra hasil segmentasi akan diekstrak sehingga menghasilkan fitur-fitur yang dapat dijadikan pembeda antara segmen vaskular normal dan abnormal seperti yang ditunjukkan pada gambar 3.7. Fitur-fitur yang digunakan yaitu *area*, *energy*, *mean gradient*, *standard*

deviation gradient, mean intensity, intensity variation, dan vessel segments.

Penjelasan proses ekstraksi fitur akan dijelaskan lebih detail dalam bentuk *pseudocode*. *Pseudocode* proses ini terdapat satu program utama yang bertujuan untuk memanggil secara keseluruhan program-program lain yang lebih kecil yang berisi proses-proses yang lebih kecil untuk dijalankan. Pada subbab-subbab berikutnya akan dijelaskan *pseudocode* masing-masing program. Tabel 3.7 sampai Tabel 3.9 berisi variabel-variabel yang akan digunakan sedangkan Tabel 3.10 dan 3.11 berisi fungsi-fungsi yang akan dipakai pada *pseudocode* program.



Gambar 3. 7 Gambaran umum proses ekstraksi fitur segmen vaskular retina

Tabel 3. 7 Daftar Variabel yang Digunakan Pada *Pseudocode* Ekstraksi Fitur Segmen Vaskular (Bagian Pertama)

No.	Nama Variabel	Tipe	Penjelasan
1.	area	int	Luas area yang merupakan pembuluh darah pada citra biner hasil segmentasi.
2.	bw_ segmented	uint8	Citra biner hasil akhir dari proses segmentasi yang berukuran 650 x 700 piksel.
3.	cc	struct	Berupa <i>struct</i> yang berisi atribut-atribut hasil dari <i>connected component analysis</i> .
4.	cc.NumObjects	int	Jumlah objek yang ditemukan dengan analisis <i>connected component</i> .
5.	count	int	Frekuensi dari piksel untuk setiap nilai intensitas.
6.	energy	double	Jumlah intensitas piksel yang merupakan pembuluh darah pada <i>green channel</i> dari citra asli.
7.	Gmag	double	Nilai yang menunjukkan <i>gradient magnitude</i> .
8.	Gdir	double	Nilai yang menunjukkan <i>gradieny direction</i> .
9.	im	uint8	citra retina dalam ruang warna RGB
10.	intvariation	double	Nilai yang menunjukkan variasi intensitas dari suatu citra.

Tabel 3. 8 Daftar Variabel yang Digunakan Pada *Pseudocode* Ekstraksi Fitur Segmen Vaskular (Bagian Kedua)

No.	Nama Variabel	Tipe	Penjelasan
11.	m	int	Nilai yang menunjukkan jumlah baris citra masukan.
12.	meangrad	double	Nilai yang menunjukkan <i>mean gradient</i> dari suatu citra.
13.	meanint	double	Nilai yang menunjukkan <i>mean intensity</i> dari semua piksel yang merupakan pembuluh darah.
14.	meanintensity	double	Nilai yang menunjukkan jumlah <i>mean intensity</i> dari suatu citra yang telah dinormalisasi.
15.	n	int	Nilai yang menunjukkan jumlah kolom citra hasil masukan.
16.	norm	double	Hasil normalisasi <i>mean intensity</i> .
17.	out	uint8	Citra hasil gabungan antara citra asli retina pada <i>green channel</i> dengan citra biner hasil segmentasi.
18.	stdgrad	double	Nilai <i>standard deviation gradient</i> dari suatu citra.
19.	stdint	double	Nilai <i>standard deviation intensity</i> dari semua piksel yang merupakan pembuluh darah.

Tabel 3. 9 Daftar Variabel yang Digunakan Pada *Pseudocode* Ekstraksi Fitur Segmen Vaskular (Bagian Ketiga)

No.	Nama Variabel	Tipe	Penjelasan
20.	sum	double	Jumlah dari <i>mean gradient</i> untuk semua piksel dalam suatu citra.
21.	vector_feature	double	Suatu matriks yang menyimpan semua fitur hasil ekstraksi citra biner hasil segmentasi.
22.	vessegm	int	Nilai yang menunjukkan jumlah objek (segmen) pembuluh darah dari hasil <i>connected component analysis</i> .
23.	x	int	Nilai dari intensitas atau level yang terdapat pada suatu citra.

Tabel 3. 10 Daftar Fungsi yang Digunakan Pada *Pseudocode* Ekstraksi Fitur Segmen Vaskular (Bagian Pertama)

No.	Nama Fungsi	Penjelasan
1.	bwconncomp	Fungsi untuk menghitung jumlah pada suatu citra.
2.	fitur1	Fungsi yang digunakan untuk menghitung fitur <i>area</i> .
3.	fitur2	Fungsi yang digunakan untuk menghitung fitur <i>energy</i>
4.	fitur3	Fungsi yang digunakan untuk menghitung fitur <i>mean gradient</i> .
5.	fitur4	Fungsi yang digunakan untuk menghitung fitur <i>standard deviation gradient</i> .

Tabel 3. 11 Daftar Fungsi yang Digunakan Pada *Pseudocode* Ekstraksi Fitur Segmen Vaskular (Bagian Kedua)

No.	Nama Fungsi	Penjelasan
6.	<code>fitur56</code>	Fungsi yang digunakan untuk menghitung fitur <i>mean intensity</i> dan <i>intensity variation</i> .
7.	<code>fitur7</code>	Fungsi yang digunakan untuk menghitung fitur <i>vessel segment</i> .
8.	<code>imgradient</code>	Fungsi yang digunakan untuk mendapatkan nilai <i>gradient magnitude</i> .
9.	<code>imhist</code>	Fungsi untuk menghasilkan histogram dari suatu citra keabuan.
10.	<code>min</code>	Fungsi untuk menemukan nilai minimum dari kumpulan data.
11.	<code>size</code>	Fungsi untuk mencari ukuran dari suatu <i>array</i> .
12.	<code>zeros</code>	Fungsi untuk membuat matriks yang terdiri dari angka nol.
13.	<code>sqrt</code>	Fungsi untuk menghitung akar kuadrat dari sebuah nilai.

3.2.1 Program Utama

Program utama merupakan program yang memanggil fungsi-fungsi lain untuk menjalankan program secara keseluruhan. Proses-proses yang dilakukan dalam algoritma yang digunakan dipisahkan dalam fungsi-fungsi yang berbeda untuk memudahkan pengecekan program setiap prosesnya. *Pseudocode* program utama ditunjukkan pada Gambar 3.8.

Masukan	Citra hasil akhir segmentasi (variabel <code>bw_segmented</code>), citra retina dalam ruang warna RGB (variabel <code>im</code>)
Keluaran	Vektor fitur (variabel <code>vector_feature</code>)
<pre> 1. [m,n] ← size(bw_segmented) 2. out ← zeros(m,n) 3. for i←1 to m 4. for j←1 to n 5. If bw_segmented(i,j)==1 6. out(i,j) ← im(i,j,3) 7. end if 8. end for 9. end for 10. [Gmag,Gdir] ← imgradient(out) 11. area ← fitur1(bw_segmented) 12. energy ← fitur2(out,m,n) 13. meangrad ← fitur3(Gmag,m,n) 14. stdgrad ← fitur4(Gmag,m,n,meangrad) 15. [meanintensity,intvariation] ← fitur56(out,m,n) 16. vessegm ← fitur7(bw_segmented) 17. vector_feature ← [area green meangrad stdgrad meanintensity intvariation vessegm]</pre>	

Gambar 3. 8 Pseudocode program utama untuk ekstraksi fitur

3.2.2 Fitur Area

Program fitur *area* merupakan program digunakan untuk menghitung nilai dari fitur *area* yang dijelaskan pada subbab 2.3.1. Pseudocode penghitungan *area* ditunjukkan Gambar 3.9.

Masukan	Citra hasil akhir segmentasi (variabel <code>bw_segmented</code>)
Keluaran	Luas area (variabel <code>area</code>)
<pre> 1. [count, x] ← imhist(bw_segmented) 2. area ← min(count)</pre>	

Gambar 3. 9 Pseudocode program menghitung *area*

3.2.3 Fitur *Energy*

Program fitur *energy* merupakan program digunakan untuk menghitung nilai dari fitur *energy* yang telah dijelaskan pada subbab 2.3.2. *Pseudocode* proses penghitungan fitur *energy* ditunjukkan pada Gambar 3.10.

Masukan	Citra gabungan antara citra asli dengan citra hasil segmentasi (variabel out)
Keluaran	Nilai energy dari piksel pembuluh darah (variabel energy)
<pre> 1. energy ← 0 2. for i ← 1 to m 3. for j ← 1 to n 4. energy ← energy + out(i, j) 5. end for 6. end for </pre>	

Gambar 3. 10 *Pseudocode* program menghitung *energy*

3.2.4 Fitur *Mean Gradient*

Program fitur *mean gradient* merupakan program digunakan untuk menghitung nilai dari fitur *mean gradient* yang telah dijelaskan pada subbab 2.3.3. *Pseudocode* proses penghitungan fitur *mean gradient* ditunjukkan pada Gambar 3.11.

Masukan	Nilai gradient magnitudo (variabel Gmag), jumlah baris citra (variabel m), jumlah kolom citra (variabel n)
Keluaran	Nilai mean gradient dari piksel pembuluh darah (variabel meangrad)
<pre> 1. sum ← 0 2. for i ← 1 to m 3. for j ← 1 to n 4. sum ← sum + Gmag(i,j) 5. end for 6. end for 7. meangrad ← sum / (m*n) </pre>	

Gambar 3. 11 Pseudocode program menghitung *mean gradient*

3.2.5 Fitur *Standard Deviation Gradient*

Program fitur *standard deviation gradient* merupakan program digunakan untuk menghitung nilai dari fitur *standard deviation gradient* yang telah dijelaskan pada subbab 2.3.4. Pseudocode proses penghitungan fitur *standard deviation gradient* ditunjukkan pada Gambar 3.12 dan 3.13.

Masukan	Nilai gradient magnitudo (variabel Gmag), jumlah baris citra (variabel m), jumlah kolom citra (variabel n), nilai mean gradient (variabel meangrad)
Keluaran	Nilai standard deviation gradient dari piksel pembuluh darah (variabel stdgrad)

Gambar 3. 12 Pseudocode program menghitung *standard deviation gradient* (bagian pertama)

```

1.  stdgrad ← 0
2.  for i←1 to m
3.      for j←1 to n
4.          stdgrad ← stdgrad + ((Gmag(i,j)-
5.              meangrad)^2)
6.      end for
7.  end for
8.  stdgrad ← sqrt(stdgrad/(m*n))

```

Gambar 3. 13 Pseudocode program menghitung *standard deviation gradient* (bagian kedua)

3.2.6 Fitur *Mean Intensity* dan *Intensity Variation*

Program fitur *mean intensity* dan *intensity variation* merupakan program digunakan untuk menghitung nilai dari fitur *mean intensity* dan *intensity variation* yang telah dijelaskan pada subbab 2.3.5 dan 2.3.6. Pseudocode proses penghitungan fitur *mean intensity* dan *intensity variation* ditunjukkan pada Gambar 3.14 dan 3.15.

Masukan	Citra gabungan antara citra asli dengan citra hasil segmentasi (variabel out), jumlah baris citra (variabel m), jumlah kolom citra (variabel n)
Keluaran	Nilai mean intensity (variabel meanintensity) dan nilai intensity variation (variabel intvariation) dari piksel pembuluh darah

Gambar 3. 14 Pseudocode program menghitung *mean intensity* dan *intensity variation* (bagian pertama)

```

1.  meanint ← 0;
2.  for i←1 to m
3.      for j←1 to n
4.          meanint ← meanint + out(i,j)
5.      end for
6.  end for
7.  meanint ← meanint/(m*n)
8.  stdint ← 0
9.  for i←1 to m
10.     for j←1 to n
11.         stdint ← stdint + ((out(i,j)-
12.             meanint)^2)
13.     end for
14. end for
15. stdint ← sqrt(stdint/(m*n))
16. norm ← zeros(i,j)
17. for i←1 to m
18.     for j←1 to n
19.         norm(i,j) ← (out(i,j)-
20.             meanint)/stdint
21.     end for
22. end for
23. meanintensity ← 0
24. for i←1 to m
25.     for j←1 to n
26.         meanintensity ← meanintensity +
27.             norm(i,j)
28.     end for
29. end for
30. intvariation ← meanint/stdint

```

Gambar 3.15 Pseudocode program menghitung *mean intensity* dan *intensity variation* (bagian kedua)

3.2.7 Fitur *Vessel Segment*

Program fitur *vessel segment* merupakan program digunakan untuk menghitung nilai dari fitur *vessel segment* yang

telah dijelaskan pada subbab 2.3.7. *Pseudocode* proses penghitungan fitur *vessel segment* ditunjukkan pada Gambar 3.16.

Masukan	Citra hasil segmentasi (variabel <code>bw_segmented</code>)
Keluaran	Nilai vessel segmen dari suatu citra hasil segmentasi (variabel <code>vessegm</code>)
<ol style="list-style-type: none"> 1. <code>cc ← bwconncomp(bw_segmented)</code> 2. <code>vessegm ← cc.NumObjects</code> 	

Gambar 3.16 *Pseudocode* program menghitung *mean vessel segment*

3.3. Perancangan Klasifikasi Segmen Vaskular Retina

Proses utama lain yang dilakukan pada akhir Tugas Akhir ini adalah proses klasifikasi segmen vaskular retina. Gambaran umum dari praproses dan segmentasi citra retina digambarkan pada Gambar 3.17.

Masukan untuk tahap ini adalah data hasil ekstraksi fitur vaskular retina yang berupa vektor fitur. Kemudian dari vektor fitur tersebut akan dilakukan perbaikan ruang fitur dengan metode *Local Fisher Discriminant Analysis* (LFDA). Tujuan dari perbaikan ruang fitur ini adalah untuk memperkecil jarak antar data pada kelas yang sama dan memperbesar jarak antar data pada kelas yang berbeda. Hasil dari LFDA adalah berupa vektor fitur yang diperbaiki dan memiliki dimensi yang lebih kecil dari vektor fitur sebelumnya.

Tahap selanjutnya yaitu klasifikasi yang terdiri dari tahap pemodelan dan tahap pengenalan menggunakan metode *Learning Vector Quantization* (LVQ). Sebelum dilakukan tahap pemodelan, terlebih dahulu ditentukan jumlah *mediods* yang akan digunakan. *Medioids* ini akan mewakili jumlah sub kelas dari kelas segmen vaskular normal dan abnormal yang akan direpresentasikan oleh jumlah *neuron* pada *hidden layer*. Setelah model terbentuk, maka data-data yang akan diuji akan dilakukan

pengenalan berdasarkan model yang telah dibuat tersebut. Pengenalan ini akan menghasilkan label kelas dari sebuah data citra retina yang menunjukkan segmen vaskular pada citra tersebut normal atau abnormal.



Gambar 3. 17 Gambaran umum proses klasifikasi segmen vaskular retina

Penjelasan proses klasifikasi segmen vaskular retina akan dijelaskan lebih detail dalam bentuk *pseudocode*. *Pseudocode* proses ini terdapat dua program, yaitu program perbaikan ruang fitur dengan *Local Fisher Discriminant Analysis* dan program untuk klasifikasi segmen vaskular dengan *Learning Vector Quantization*. Tabel 3.12 sampai Tabel 3.18 berisi variabel-variabel yang akan digunakan sedangkan Tabel 3.19 dan 3.20 berisi fungsi-fungsi yang akan dipakai pada *pseudocode* program.

Tabel 3. 12 Daftar Variabel yang Digunakan Pada *Pseudocode* Klasifikasi Segmen Vaskular (Bagian Pertama)

No.	Nama Variabel	Tipe	Penjelasan
1.	A	double	Nilai pangkat dari variabel dibagi terhadap bilangan natural (matriks <i>affinity</i>).
2.	c	int	Matriks yang menyimpan kelas dari data vektor fitur.

Tabel 3. 13 Daftar Variabel yang Digunakan Pada *Pseudocode* Klasifikasi Segmen Vaskular (Bagian Kedua)

No.	Nama Variabel	Tipe	Penjelasan
3.	d	int	Jumlah baris dari nilai fitur pada vektor fitur.
4.	distance2	double	Nilai kuadrat jarak dari variabel x_c .
5.	eigvec	double	Nilai generalisasi dari vektor eigen.
6.	eigval_matrix	double	Nilai generalisasi dari nilai eigen.
7.	error	int	Matriks yang menandai letak data yang mengalami kesalahan klasifikasi.
8.	flag	logical	Bernilai benar jika variabel <code>localscale</code> bukan nol.
9.	G	double	Nilai perkalian x_c dengan A dan transpose dari x_c .
10.	in	double	Transpos dari nilai fitur pada vektor fitur hasil reduksi LFDA yang digunakan untuk <i>testing</i> klasifikasi.
11.	index	int	Indeks terurut dari variabel <code>distance2</code> .
12.	kNNDist2	double	Nilai terurut jarak tetangga terdekat.
13.	localscale	double	Perkalian antara variabel <code>sigma</code> dan transpose dari variabel <code>sigma</code> .
14.	LR	double	Nilai <i>learning rate</i> yang dibutuhkan untuk masukan LVQ.

Tabel 3. 14 Daftar Variabel yang Digunakan Pada *Pseudocode* Klasifikasi Segmen Vaskular (Bagian Ketiga)

No.	Nama Variabel	Tipe	Penjelasan
15.	mediod	int	Jumlah <i>mediods</i> atau <i>neuron</i> pada <i>hidden layer</i> .
16.	n	int	Jumlah kolom dari nilai fitur pada vektor fitur.
17.	nc	int	Jumlah kolom dari variabel x_c .
18.	net	network	Model dari jaringan <i>neural network</i> yang terbentuk menggunakan algoritma LVQ.
19.	out	int	Nilai hasil dari klasifikasi yang menunjukkan kelas dari data <i>testing</i> .
20.	perf	double	Nilai <i>error</i> klasifikasi.
21.	r	int	Jumlah dimensi hasil reduksi.
22.	sigma	double	Nilai akar kuadrat dari variabel $kNNdist2$.
23.	sort_eigval	double	Nilai terurut untuk nilai eigen.
24.	sort_eigval_index	double	Nilai terurut untuk indeks dari nilai eigen.
25.	sorted	double	Nilai terurut dari variabel $distance2$.
26.	t	int	Transpos dari nilai target yang menunjukkan kelas dari vektor fitur hasil reduksi LFDA.
27.	T3	double	Matriks transformasi hasil LFDA sejumlah 3 dimensi.

Tabel 3. 15 Daftar Variabel yang Digunakan Pada *Pseudocode* Klasifikasi Segmen Vaskular (Bagian Keempat)

No.	Nama Variabel	Tipe	Penjelasan
28.	T4	double	Matriks transformasi hasil LFDA sejumlah 4 dimensi.
29.	T5	double	Matriks transformasi hasil LFDA sejumlah 5 dimensi.
30.	T6	double	Matriks transformasi hasil LFDA sejumlah 6 dimensi.
31.	targ	int	Transpos dari nilai target yang menunjukkan kelas dari vektor fitur hasil reduksi LFDA untuk <i>testing</i> klasifikasi.
32.	test	double	Data <i>testing</i> yang menjadi masukan pada LVQ, dipilih berdasarkan jumlah dimensi yang digunakan.
33.	test3	double	Data <i>testing</i> yang memiliki jumlah dimensi sama dengan 3.
34.	test4	double	Data <i>testing</i> yang memiliki jumlah dimensi sama dengan 4.
35.	test5	double	Data <i>testing</i> yang memiliki jumlah dimensi sama dengan 5.
36.	test6	double	Data <i>testing</i> yang memiliki jumlah dimensi sama dengan 6.
37.	tr3	double	Perkalian antara vektor fitur yang akan dijadikan data <i>training</i> dengan T3.

Tabel 3. 16 Daftar Variabel yang Digunakan Pada *Pseudocode* Klasifikasi Segmen Vaskular (Bagian Kelima)

No.	Nama Variabel	Tipe	Penjelasan
38.	tr4	double	Perkalian antara vektor fitur yang akan dijadikan data <i>training</i> dengan T4.
39.	tr5	double	Perkalian antara vektor fitur yang akan dijadikan data <i>training</i> dengan T5.
40.	tr6	double	Perkalian antara vektor fitur yang akan dijadikan data <i>training</i> dengan T6.
41.	train	double	Data <i>training</i> yang menjadi masukan pada LVQ, dipilih berdasarkan jumlah dimensi yang digunakan.
42.	train3	double	Data <i>training</i> yang memiliki jumlah dimensi sama dengan 3.
43.	train4	double	Data <i>training</i> yang memiliki jumlah dimensi sama dengan 4.
44.	train5	double	Data <i>training</i> yang memiliki jumlah dimensi sama dengan 5.
45.	train6	double	Data <i>training</i> yang memiliki jumlah dimensi sama dengan 6.
46.	ts3	double	Perkalian antara vektor fitur yang akan dijadikan data <i>testing</i> dengan T3.

Tabel 3. 17 Daftar Variabel yang Digunakan Pada *Pseudocode* Klasifikasi Segmen Vaskular (Bagian Keenam)

No.	Nama Variabel	Tipe	Penjelasan
47.	ts4	double	Perkalian antara vektor fitur yang akan dijadikan data <i>testing</i> dengan T4.
48.	ts5	double	Perkalian antara vektor fitur yang akan dijadikan data <i>testing</i> dengan T5.
49.	ts6	double	Perkalian antara vektor fitur yang akan dijadikan data <i>testing</i> dengan T6.
50.	tsb	double	<i>Beetwen class catter matrix.</i>
51.	tsw	double	<i>Within class catter matrix.</i>
52.	training	double	Bagian dari variabel <i>vector_feature</i> yang digunakan sebagai data <i>training</i> .
53.	vector_feature	double	Suatu matriks yang menyimpan semua fitur hasil ekstraksi citra biner hasil segmentasi.
54.	x	double	Transpos dari nilai fitur pada vektor fitur hasil reduksi LFDA.
55.	x	double	Nilai fitur dari variabel <i>vector_feature</i> .
56.	x _c	double	Nilai fitur dari variabel <i>vector_feature</i> yang terdapat pada kelas yang sama.
57.	x _{c1}	double	Penjumlahan pada setiap kolom dari nilai x _c .

Tabel 3. 18 Daftar Variabel yang Digunakan Pada *Pseudocode* Klasifikasi Segmen Vaskular (Bagian Ketujuh)

No.	Nama Variabel	Tipe	Penjelasan
58.	x_{c2}	double	Penjumlahan pada setiap baris dari kuadrat nilai x_c .
59.	Y	int	Nilai kelas dari data pada <code>vector_feature</code> .
60.	Y'	int	Tranpose dari variabel Y .

Tabel 3. 19 Daftar Fungsi yang Digunakan Pada *Pseudocode* Klasifikasi Segmen Vaskular (Bagian Pertama)

No.	Nama Fungsi	Penjelasan
1.	<code>diag</code>	Fungsi yang digunakan untuk menampilkan matriks diagonal.
2.	<code>eig</code>	Fungsi yang digunakan untuk menghitung generalisasi nilai eigen dan vektor eigen.
3.	<code>eigs</code>	Fungsi yang digunakan untuk menghitung beberapa generalisasi nilai eigen dan vektor eigen sesuai dengan jumlah dimensi reduksi.
4.	<code>gsubtract</code>	Fungsi yang digunakan untuk melakukan pengurangan elemen dari matriks secara general.
5.	<code>LFDA</code>	Fungsi yang digunakan untuk menjalankan algoritma LFDA.
6.	<code>lvqnet</code>	Fungsi yang digunakan untuk membangun jaringan <i>neural network</i> dengan menggunakan algoritma LVQ.

Tabel 3. 20 Daftar Fungsi yang Digunakan Pada *Pseudocode* Klasifikasi Segmen Vaskular (Bagian Kedua)

No.	Nama Fungsi	Penjelasan
7.	perform	Fungsi yang digunakan untuk menghitung <i>error</i> klasifikasi dari data <i>testing</i> .
8.	repmat	Fungsi yang digunakan untuk menyalin nilai variabel tertentu.
9.	size	Fungsi untuk menghitung ukuran matriks.
10.	sort	Fungsi yang digunakan untuk melakukan pengurutan data mulai data terkecil hingga data terbesar.
11.	sum	Fungsi yang digunakan untuk penjumlahan.
12.	train	Fungsi yang digunakan untuk membuat model <i>neural network</i> LVQ berdasarkan data <i>training</i> tertentu.
13.	transpose	Fungsi yang digunakan untuk melakukan transpos data.
14.	unique	Fungsi yang digunakan untuk mencari semua nilai berbeda dari suatu variabel.
15.	view	Fungsi yang digunakan untuk menampilkan model dari jaringan yang dibangun.
16.	zeros	Fungsi yang digunakan untuk membuat matriks yang terdiri dari angka nol.

3.3.1 Program *Local Fisher Discriminant Analysis*

Program *Local Fisher Discriminant Analysis* merupakan program yang akan menghasilkan perbaikan vektor fitur. Program ini akan menghasilkan vektor fitur dengan dimensi yang berbeda-beda, yaitu tiga dimensi, empat dimensi, lima dimensi, dan enam dimensi. Program ini juga akan memanggil fungsi *Local Fisher Discriminant Analysis*. *Pseudocode* ini menerapkan metode yang dijelaskan pada subbab 2.4.1. *Pseudocode* program ditunjukkan pada Gambar 3.18 dan Gambar 3.19.

Masukan	Vektor fitur (variabel <code>vector_feature</code>)
Keluaran	Data training dan testing dengan tiga dimensi (variabel <code>train3, test3</code>), empat dimensi (variabel <code>train4, test4</code>), lima dimensi (variabel <code>train5, test5</code>), enam dimensi (variabel <code>train6, test6</code>)
<pre> 1. training ← vector_feature(1:20, :) 2. X ← transpose(training(:, 2:8)) 3. Y ← training(:, 1) 4. [T3, T4, T5, T6] ← LFDA(X, Y) 5. tr3 ← training(:, 2:8) * T3 6. train3 ← [training(:, 1) tr3] 7. ts3 ← data(1:53, 2:8) * T3 8. test3 ← [data(1:53, 1) ts3] 9. tr4 ← training(:, 2:8) * T4 10. train4 ← [training(:, 1) tr4] 11. ts4 ← data(1:53, 2:8) * T4 12. test4 ← [data(1:53, 1) ts4] 13. tr5 ← training(:, 2:8) * T5 14. train5 ← [training(:, 1) tr5] 15. ts5 ← data(1:53, 2:8) * T5 </pre>	

Gambar 3. 18 *Pseudocode* program *Local Fisher Discriminant Analysis* (bagian pertama)

```

16. test5 ← [data(1:53,1) ts5]
17. tr6 ← training(:,2:8)*T6
18. train6 ← [training(:,1) tr6]
19. ts6 ← data(1:53,2:8)*T6
20. test6 ← [data(1:53,1) ts6]

```

Gambar 3. 19 Pseudocode program *Local Fisher Discriminant Analysis* (bagian kedua)

Program *Local Fisher Discriminant Analysis* membutuhkan fungsi LFDA untuk dapat menjalankan algoritma LFDA. *Pseudocode* dari fungsi LFDA ditunjukkan pada Gambar 3.20 dan Gambar 3.21.

Masukan	Nilai fitur dan nilai kelas dari variabel <i>vector_feature</i> (variabel X, variabel Y)
Keluaran	Matriks transformasi hasil LFDA sejumlah 3 dimensi, 4 dimensi, 5 dimensi, dan 6 dimensi (variabel T3, T4, T5, T6)
<pre> 1. [d n] ← size(X) 2. tSb ← zeros(d,d) 3. tSw ← zeros(d,d) 4. for c ← unique(Y') 5. Xc ← X(:,Y==c) 6. Nc ← size(Xc,2) 7. Xc2 ← sum(Xc.^2,1) 8. distance2 ← repmat(Xc2,nc,1)+ repmat(Xc2',1,nc)-2*Xc'*Xc 9. [sorted,index] ← sort(distance2) 10. kNNdist2 ← sorted(kNN+1,:) 11. sigma ← sqrt(kNNdist2) 12. localscale ← sigma'*sigma 13. flag ← (localscale~=0) 14. A ← zeros(nc,nc) </pre>	

Gambar 3. 20 Pseudocode fungsi *Local Fisher Discriminant Analysis* (bagian pertama)

```

15.     A(flag) ← exp(-distance2(flag)./
        localscale(flag))
16.     Xc1 ← sum(Xc,2)
17.     G ← Xc*(repmat(sum(A,2),[1 d]).*Xc') -
        Xc*A*Xc'
18.     tSb ← tSb+G/n+Xc*Xc'*(1-nc/n)+
        Xc1*Xc1'/n
19.     tSw ← tSw+G/nc
20.     end for
21.     X1 ← sum(X,2)
22.     tSb ← tSb-X1*X1'/n-tSw
23.     tSb ← (tSb+tSb')/2
24.     tSw ← (tSw+tSw')/2
25.     If r==d
26.         [eigvec,eigval_matrix] ← eig(tSb,tSw)
27.     else
28.         [eigvec,eigval_matrix] ←
29.         eigs(tSb,tSw,r, 'la')
30.     end if
31.     eigval ← diag(eigval_matrix)
32.     [sort_eigval,sort_eigval_index]←
        sort(eigval)
33.     T0 ← eigvec(:,sort_eigval_index(end:-1:1))
34.     T3 ← T0(:,1:3)
35.     T4 ← T0(:,1:4)
36.     T5 ← T0(:,1:5)
37.     T6 ← T0(:,1:6)

```

Gambar 3. 21 Pseudocode fungsi *Local Fisher Discriminant Analysis* (bagian kedua)

3.3.2 Program *Learning Vector Quantization*

Program *Learning Vector Quantization* merupakan program yang akan memodelkan data *training* sesuai dengan jumlah *mediods* atau *neuron* pada *hidden layer* dan nilai *learning rate*. Setelah itu program akan melakukan klasifikasi atau pengenalan dengan menggunakan data *testing*. *Pseudocode* ini menerapkan metode yang dijelaskan pada subbab 2.4.2.

Pseudocode program *Learning Vector Quantization* ditunjukkan pada Gambar 3.22.

Masukan	Data <i>training</i> dan data <i>testing</i> sesuai jumlah dimensi yang digunakan (variabel <i>train</i> , variabel <i>test</i>)
Keluaran	Nilai <i>error</i> klasifikasi (variabel <i>perf</i>)
<pre> 1. x ← transpose(train(:,1:r)) 2. t ← transpose(train(:,r+1:r+2)) 3. net ← lvqnet(medioid,LR,'learnlv1') 4. net ← train(net,x,t) 5. view(net) 6. in ← transpose(test(:,1:3)) 7. targ ← transpose(testing(:,4:5)) 8. out ← net(in) 9. perf ← perform(net,out,targ) 10. error ← gsubtract(targ,out) </pre>	

Gambar 3. 22 *Pseudocode* program *Learning Vector Quantization*

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1. Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi adalah program MATLAB 8.1.0 (R2013a) yang dipasang pada sistem operasi Windows 7.

4.2. Implementasi

Pada bagian ini akan dijelaskan implementasi setiap subbab yang terdapat pada bab sebelumnya yaitu bab perancangan perangkat lunak. Bagian implementasi ini juga akan dibagi menjadi 3 bagian, yaitu bagian praproses dan segmentasi citra retina, bagian ekstraksi fitur segmen vaskular retina, dan bagian klasifikasi segmen vaskular normal dan abnormal.

4.2.1. Implementasi Praproses dan Segmentasi Citra Retina

Pada bagian ini akan dijelaskan implementasi dari praproses dan segmentasi citra retina. Implementasi akan dijelaskan mulai dari praproses yang berupa *masking*. Setelah itu dilanjutkan dengan segmentasi pembuluh darah pada citra retina menggunakan transformasi wavelet dan *cleaning*.

4.2.1.1 Masking

Masukan dari program *masking* berupa citra asli retina yang berukuran 650 x 700 piksel dengan tipe data uint8. Citra tersebut akan diproses sehingga menghasilkan citra biner *masking* yang memisahkan antara objek retina dengan latar belakang gambar. *Masking* terdiri dari dua proses yaitu *thresholding* dan *shrinking*. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.1.2. Implementasi dalam matlab ditunjukkan pada Kode Sumber 4.1.

1	<code>bw_mask = im2bw(im, threshold);</code>
2	<code>figure, imshow(bw_mask);</code>
3	<code>d1 = max(sum(bw_mask));</code>
4	<code>d2 = max(sum(bw_mask, 2));</code>
5	<code>d = max(d1, d2);</code>
6	<code>siz = max(5, round(d * .025));</code>
7	<code>final_mask = imerode(bw_mask, ones(siz));</code>
8	<code>figure, imshow(final_mask);</code>
9	<code>imwrite(bw_mask, 'mask.bmp');</code>

Kode Sumber 4. 1 Implementasi proses masking dengan *thresholding* dan *shrinking*

4.2.1.2 Transformasi Wavelet

Implementasi dari transformasi wavelet mengaplikasikan algoritma *Isotropic Undecimated Wavelet* serta melakukan *thresholding* untuk menghasilkan citra biner yang berupa pembuluh darah retina. Masukan untuk transformasi wavelet adalah berupa citra hasil *masking* serta citra retina yang telah diubah ke dalam ruang warna skala keabuan. Sehingga sebelum dilakukan transformasi, masukan citra diubah ke dalam ruang warna skala keabuan seperti pada Kode Sumber 4.2 Implementasi dalam matlab untuk proses transformasi wavelet ditunjukkan pada Kode Sumber 4.3. Sedangkan untuk fungsi dilatasi kernel wavelet yang dibutuhkan pada transformasi wavelet dituliskan pada Kode

Sumber 4.4. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.1.3.

1	<code>img = rgb2gray(img);</code>
---	-----------------------------------

Kode Sumber 4. 2 Implementasi perubahan citra asli ke ruang warna skala keabuan

1	<code>padding = 'symmetric';</code>
2	<code>if numel(img) > 10000000</code>
3	<code> wclass = 'single';</code>
4	<code> s_in = single(img);</code>
5	<code>else</code>
6	<code> wclass = 'double';</code>
7	<code> s_in = double(img);</code>
8	<code>end</code>
9	<code>w = zeros([size(img) length(levels)],</code> <code> wclass);</code>
10	<code>b3 = [1 4 6 4 1] / 16;</code>
11	<code>for ii = 1:levels(end)</code>
12	<code> h = dilate_wavelet_kernel(b3, 2^(ii-1)-</code> <code> 1);</code>
13	<code> s_out = imfilter(s_in, h' * h, padding);</code>
14	<code> ind = find(levels == ii);</code>
15	<code> if isscalar(ind)</code>
16	<code> w(:, :, ind) = s_in - s_out;</code>
17	<code> end</code>
18	<code> s_in = s_out;</code>
19	<code>end</code>
20	<code>w = squeeze(w);</code>

Kode Sumber 4. 3 Implementasi transformasi wavelet

1	function h2 = dilate_wavelet_kernel(h, spacing)
2	if ~isvector(h) && ~isscalar(spacing)
3	error(['Invalid input to DILATE_WAVELET_KERNEL: 'H must be a vector and SPACING must be a scalar']);
4	end
5	h2 = zeros(1,numel(h)+spacing*(numel(h)-1));
6	if size(h,1) > size(h,2)
7	h2 = h2';
8	end
9	h2(1:spacing+1:end) = h;

Kode Sumber 4. 4 Implementasi k fungsi dilatasi kernel wavelet

4.2.1.3 *Cleaning*

Implementasi dari program *cleaning* mengaplikasikan proses *cleaning* yakni menghapus objek pembuluh darah berukuran kecil serta mengisi lubang berukuran kecil. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.1.4. Implementasi dalam matlab untuk proses *cleaning* ditunjukkan pada Kode Sumber 4.5 dan Kode Sumber 4.6.

1	if iuwt_px_remove > 0
2	cc_objects = bwconncomp(bw_wavelet);
3	area_objects = cellfun('size', cc_objects.PixelIdxList, 1);
4	bw_segmented = false(size(bw));
5	inds = area_objects >= iuwt_px_remove;
6	bw_segmented(cell2mat(cc_objects.PixelId xList(inds))) = true;
7	else
8	bw_segmented = bw_wavelet;
9	end

Kode Sumber 4. 5 Implementasi *cleaning* (bagian pertama)

10	<code>if iuwt_px_fill > 0</code>
11	<code> cc_holes = bwconncomp(~bw_segmented);</code>
12	<code> area_holes = cellfun('size', cc_holes.PixelIdxList, 1);</code>
13	<code> inds = area_holes < iuwt_px_fill;</code>
14	<code> bw_segmented(cell2mat(cc_holes.PixelIdxL ist(inds))) = true;</code>
15	<code>end</code>

Kode Sumber 4. 6 Implementasi *cleaning* (bagian kedua)

4.2.2. Implementasi Ekstraksi Fitur Segmen Vaskular Retina

Pada bagian ini akan dijelaskan implementasi dari proses ekstraksi fitur segmen vaskular retina. Implementasi ini bertujuan untuk menghasilkan fitur-fitur yang dapat dijadikan pembeda antara segmen vaskular normal dan abnormal. Fitur-fitur yang digunakan yaitu *area*, *energy*, *mean gradient*, *standard deviation gradient*, *mean intensity*, *intensity variation*, dan *vessel segments*.

4.2.2.1 Fitur *area*

Implementasi fitur *area* bertujuan untuk menemukan nilai *area* pembuluh darah dari suatu citra retina. Masukan untuk program ini adalah berupa citra biner hasil segmentasi. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.2. Implementasi dalam matlab untuk proses penghitungan fitur *area* ditunjukkan pada Kode Sumber 4.7.

1	<code>[count, x] = imhist(bw_segmented);</code>
2	<code>area = min(count);</code>

Kode Sumber 4. 7 Implementasi fitur *area*

4.2.2.2 Fitur *energy*

Implementasi fitur *energy* bertujuan untuk menemukan nilai intensitas piksel pembuluh darah pada *green channel* dari suatu citra retina. Masukan untuk program ini adalah berupa citra biner hasil segmentasi serta citra asli retina yang diambil nilai intensitasnya pada *green channel*. Sebelumnya dilakukan terlebih dahulu penggabungan antara citra hasil segmentasi dan citra asli seperti yang ditunjukkan pada Sumber Kode 4.8. Implementasi dalam matlab untuk proses penghitungan fitur *energy* ditunjukkan pada Kode Sumber 4.9. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.3.

1	<code>[m,n]=size(bw_segmented);</code>
2	<code>out = zeros(m,n);</code>
3	<code>for i = 1:m</code>
4	<code> for j = 1:n</code>
5	<code> if(bw_segmented(i,j)==1)</code>
6	<code> out(i,j) = im(i,j,3);</code>
7	<code> end</code>
8	<code> end</code>
9	<code>end</code>

Kode Sumber 4. 8 Penggabungan citra hasil segmentasi dan citra asli

1	<code>energy=0;</code>
2	<code>for i=1:m</code>
3	<code> for j=1:n</code>
4	<code> energy = energy + out(i,j);</code>
5	<code> end</code>
6	<code>end</code>

Kode Sumber 4. 9 Implementasi fitur *energy*

4.2.2.3 Fitur *mean gradient*

Implementasi fitur *mean gradient* bertujuan untuk menemukan nilai rata-rata *gradient magnitude* piksel pembuluh

darah dari suatu citra retina. Masukan untuk program ini adalah nilai *gradient magnitude* dari setiap piksel pada *green channel* citra retina. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.4. Implementasi dalam matlab untuk proses penghitungan fitur *mean gradient* ditunjukkan pada Kode Sumber 4.10.

1	sum=0;
2	for i=1:m
3	for j=1:n
4	sum = sum + Gmag(i,j);
5	end
6	end
7	meangrad = sum/(m*n);

Kode Sumber 4. 10 Implementasi fitur *mean gradient*

4.2.2.4 Fitur *standard deviation gradient*

Implementasi fitur *standard deviation gradient* bertujuan untuk menemukan nilai standar deviasi *gradient magnitude* piksel pembuluh darah dari suatu citra retina. Masukan untuk program ini adalah nilai *gradient magnitude* dari setiap piksel pada *green channel* citra retina serta nilai *mean gradient*. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.5. Implementasi dalam matlab untuk proses penghitungan fitur *standard deviation gradient* ditunjukkan pada Kode Sumber 4.11.

1	stdgrad=0;
2	for i=1:m
3	for j=1:n
4	stdgrad = stdgrad + ((Gmag(i,j)- meangrad)^2);
5	end
6	end
7	stdgrad = sqrt(stdgrad/(m*n));

Kode Sumber 4. 11 Implementasi *standard deviation gradient*

4.2.2.5 Fitur *mean intensity* dan *intensity variation*

Implementasi fitur *mean intensity* dan *intensity variation* bertujuan untuk menemukan nilai rata-rata dan rasio antara rata-rata dengan standard deviasi dari intensitas piksel pembuluh darah pada *green channel* suatu citra retina. Masukan untuk program ini adalah berupa penggabungan antara citra biner hasil segmentasi serta citra asli retina yang diambil nilai intensitasnya pada *green channel*. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.6. Implementasi dalam matlab untuk proses penghitungan fitur *mean intensity* dan *intensity variation* ditunjukkan pada Kode Sumber 4.12 dan Kode Sumber 4.13.

1	<code>meanint=0;</code>
2	<code>for i=1:m</code>
3	<code> for j=1:n</code>
4	<code> meanint = meanint + out(i,j);</code>
5	<code> end</code>
6	<code>end</code>
7	<code>meanint = meanint/(m*n);</code>
8	<code>stdint=0;</code>
9	<code>for i=1:m</code>
10	<code> for j=1:n</code>
11	<code> stdint = stdint + ((out(i,j)-</code> <code> meanint)^2);</code>
12	<code> end</code>
13	<code>end</code>
14	<code>stdint = sqrt(stdint/(m*n));</code>
15	<code>norm = zeros(i,j);</code>
16	<code>for i=1:m</code>
17	<code> for j=1:n</code>
18	<code> norm(i,j) = (out(i,j)-meanint)/</code> <code> stdint;</code>
19	<code> end</code>
20	<code>end</code>

Kode Sumber 4. 12 Implementasi fitur *mean intensity* dan *intensity variation* (bagian pertama)

21	<code>meanintensity=0;</code>
22	<code>for i=1:m</code>
23	<code> for j=1:n</code>
24	<code> meanintensity = meanintensity + norm(i,j);</code>
25	<code> end</code>
26	<code>end</code>
27	<code>intvariation = meanint/stdint;</code>

Kode Sumber 4. 13 Implementasi fitur *mean intensity* dan *intensity variation* (bagian kedua)

4.2.2.6 Fitur *vessel segments*

Implementasi fitur *vessel segments* bertujuan untuk menemukan jumlah segmen pembuluh darah dari suatu citra retina. Masukan untuk program ini adalah citra biner hasil segmentasi. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.7. Implementasi dalam matlab untuk proses penghitungan fitur *vessel segments* ditunjukkan pada Kode Sumber 4.14.

1	<code>cc = bwconncomp(imsegm);</code>
2	<code>vessegm = cc.NumObjects;</code>

Kode Sumber 4. 14 Implementasi fitur *vessel segments*

4.2.3. Implementasi Klasifikasi Segmen Vaskular Retina

Pada bagian ini akan dijelaskan implementasi dari klasifikasi segmen vaskular retina. Implementasi akan dijelaskan mulai dari tahap perbaikan ruang fitur dengan menggunakan algoritma LFDA. Setelah itu dilanjutkan dengan pemodelan dan pengenalan segmen vaskular citra retina menggunakan algoritma LVQ. Hasil akhir dari implementasi ini berupa label kelas dari segmen vaskular retina.

4.2.3.1 Program *Local Fisher Discriminant Analysis*

Implementasi program *Local Fisher Discriminant Analysis* akan memperbaiki representasi ruang fitur yang terdapat pada vektor fitur. Masukan untuk program ini adalah vektor fitur hasil dari metode ekstraksi fitur beserta label kelasnya. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.3.1. Implementasi dalam matlab untuk program *Local Fisher Discriminant Analysis* ditunjukkan pada Kode Sumber 4.15.

1	<code>training = vector_feature(1:20,:);</code>
2	<code>X = transpose(training(:,2:8));</code>
3	<code>Y = training(:,1);</code>
4	<code>[T3,T4,T5,T6] = LFDA(X,Y);</code>
5	<code>tr3 = training(:,2:8)*T3;</code>
6	<code>train3 = [training(:,1) tr3];</code>
7	<code>ts3 = data(1:53,2:8)*T3;</code>
8	<code>test3 = [data(1:53,1) ts3];</code>
9	<code>tr4 = training(:,2:8)*T4;</code>
10	<code>train4 = [training(:,1) tr4];</code>
11	<code>ts4 = data(1:53,2:8)*T4;</code>
12	<code>test4 = [data(1:53,1) ts4];</code>
13	<code>tr5 = training(:,2:8)*T5;</code>
14	<code>train5 = [training(:,1) tr5];</code>
15	<code>ts5 = data(1:53,2:8)*T5;</code>
16	<code>test5 = [data(1:53,1) ts5];</code>
17	<code>tr6 = training(:,2:8)*T6;</code>
18	<code>train6 = [training(:,1) tr6];</code>
19	<code>ts6 = data(1:53,2:8)*T6;</code>
20	<code>test6 = [data(1:53,1) ts6];</code>

Kode Sumber 4. 15 Implementasi program *Local Fisher Discriminant Analysis*

4.2.3.2 Fungsi *Local Fisher Discriminant Analysis*

Implementasi fungsi *Local Fisher Discriminant Analysis* bertujuan untuk menghasilkan matriks transformasi berdasarkan jumlah dimensi yang dipilih. Masukan untuk fungsi ini adalah nilai fitur hasil dari vektor fitur dan label kelas untuk masing-masing data pada vektor fitur. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.3.1. Implementasi dalam matlab untuk fungsi *Local Fisher Discriminant Analysis* ditunjukkan pada Kode Sumber 4.16 dan Kode Sumber 4.17.

1	<code>[d n]=size(X);</code>
2	<code>tSb=zeros(d,d);</code>
3	<code>tSw=zeros(d,d);</code>
4	<code>for c=unique(Y')</code>
5	<code> Xc=X(:,Y==c);</code>
6	<code> nc=size(Xc,2);</code>
7	<code> Xc2=sum(Xc.^2,1);</code>
8	<code> distance2= repmat(Xc2,nc,1)+ repmat</code> <code> (Xc2',1,nc)-2*Xc'*Xc;</code>
9	<code> [sorted,index]=sort(distance2);</code>
10	<code> kNNDist2=sorted(kNN+1,:);</code>
11	<code> sigma=sqrt(kNNDist2);</code>
12	<code> localscale=sigma'*sigma;</code>
13	<code> flag=(localscale~=0);</code>
14	<code> A=zeros(nc,nc);</code>
15	<code> A(flag)=exp(-distance2(flag)./</code> <code> localscale(flag));</code>
16	<code> Xc1=sum(Xc,2);</code>
17	<code> G=Xc*(repmat(sum(A,2),[1 d]).*Xc')-</code> <code> Xc*A*Xc';</code>
18	<code> tSb=tSb+G/n+Xc*Xc'*(1-nc/n)+Xc1*Xc1'/n;</code>
19	<code> tSw=tSw+G/nc;</code>
20	<code>end</code>

Kode Sumber 4. 16 Implementasi fungsi *Local Fisher Discriminant Analysis* (bagian pertama)

21	<code>X1=sum(X,2);</code>
22	<code>tSb=tSb-X1*X1'/n-tSw;</code>
23	<code>tSb=(tSb+tSb')/2;</code>
24	<code>tSw=(tSw+tSw')/2;</code>
25	<code>if r==d</code>
26	<code> [eigvec,eigval_matrix]=eig(tSb,tSw)</code>
27	<code>else</code>
28	<code> [eigvec,eigval_matrix]=eigs</code> <code> (tSb,tSw,r,'la')</code>
29	<code>end</code>
30	<code>eigval=diag(eigval_matrix);</code>
31	<code>[sort_eigval,sort_eigval_index]=sort(eigval);</code>
32	<code>T0=eigvec(:,sort_eigval_index(end:-1:1));</code>
33	<code>T3 = T0(:,1:3);</code>
34	<code>T4 = T0(:,1:4);</code>
35	<code>T5 = T0(:,1:5);</code>
36	<code>T6 = T0(:,1:6);</code>

Kode Sumber 4. 17 Implementasi fungsi *Local Fisher Discriminant Analysis* (bagian kedua)

4.2.3.3 Klasifikasi dengan *Learning Vector Quantization*

Implementasi fungsi *Learning Vector Quantization* bertujuan untuk memodelkan jaringan *neural network* dari data *training* dan melakukan klasifikasi hingga menghasilkan kelas untuk data *testing*. Masukan untuk program ini adalah nilai fitur reduksi hasil dari algoritma LFDA dengan jumlah dimensi reduksi yang dipilih. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.3.2. Implementasi dalam matlab untuk fungsi *Learning Vector Quantization* ditunjukkan pada Kode Sumber 4.18 dan Kode Sumber 4.19.

1	<code>x = transpose(train(:,1:r));</code>
2	<code>t = transpose(train(:,r+1:r+2));</code>
3	<code>net = lvqnet(medioid,LR,'learnlvl');</code>

Kode Sumber 4. 18 Implementasi fungsi *Learning Vector Quantization* (bagian pertama)

4	<code>net = train(net,x,t);</code>
5	<code>view(net);</code>
6	<code>in = transpose(test (:,1:3));</code>
7	<code>targ = transpose(testing(:,4:5));</code>
8	<code>out = net(in);</code>
9	<code>perf = perform(net,out,targ);</code>
10	<code>error = gsubtract(targ,out);</code>

Kode Sumber 4. 19 Implementasi fungsi *Learning Vector Quantization* (bagian kedua)

BAB V

UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba serta analisis setiap pengujian.

5.1. Lingkungan Uji Coba

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi pembuatan klasifikasi segmen vaskular retina mata pada Tugas Akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Prosesor: Intel® Core™ i3-350M processor @ 2.26GHz.
 - b. Memory(RAM): 4 GB.
 - c. Tipe system: 64-bit sistem operasi.
2. Perangkat lunak
 - a. Sistem operasi: Windows 7 Ultimate.
 - b. Perangkat pengembang: MATLAB 8.1.0 (R2013a).

5.2. Uji Coba Kebenaran Citra Hasil Segmentasi

Pada uji coba ini akan dilakukan pengujian terhadap kebenaran dan kesesuaian metode praproses dan segmentasi pada sistem yang telah dibuat. Uji coba kebenaran bertujuan untuk menunjukkan bahwa program telah dapat berjalan sebagaimana seharusnya.

5.2.1 Data Uji Coba Kebenaran Citra Hasil Segmentasi

Data yang digunakan pada tahap uji coba kebenaran hasil segmentasi ini menggunakan 40 buah citra retina dari basis data DRIVE. Data yang digunakan berupa citra retina dalam ruang warna RGB dengan berformat *.tif* serta citra *ground truth* yang dihasilkan secara manual dengan format *.gif*.

5.2.2 Skenario Uji Coba Kebenaran Citra Hasil Segmentasi

Uji coba dimulai dengan mengubah citra dalam ruang RGB menjadi citra biner hasil segmentasi dengan menggunakan metode praproses yaitu *masking* dan metode segmentasi. Pada bagian *thresholding* yang merupakan salah satu tahap *masking*, nilai *threshold* yang digunakan untuk citra pada basis data STARE dan DRIVE berkisar antara 0.1 hingga 0.23. Nilai *threshold* ini diperoleh berdasarkan uji coba. Untuk salah satu metode segmentasi, yaitu transformasi wavelet dengan IUWT digunakan level wavelet yang sama untuk semua citra yaitu level 2 dan level 3. Sedangkan untuk *thresholding* pada bagian segmentasi, nilai *threshold* yang digunakan untuk semua citra adalah 0.2. Dan untuk nilai *min_object_size* dan *min_holes_size* yang digunakan adalah 0.05% dari jumlah piksel keseluruhan.

Citra hasil segmentasi dibandingkan dengan citra *ground truth* basis data DRIVE untuk diuji kebenarannya. Uji kebenaran citra hasil segmentasi dilakukan dengan evaluasi nilai *error* dan akurasi menggunakan *confusion matrix* yang ditampilkan pada Tabel 5.1.

Tabel 5. 1 *Confusion matrix* untuk segmentasi citra vaskular retina

Prediksi	Kenyataan	
	Segmen vaskular	Latar belakang
Segmen vaskular	TP	FP
Latar belakang	FN	TN

Berdasarkan *confusion matrix* pada Tabel 5.1, TP diperoleh ketika suatu piksel pada citra *ground truth* dan citra hasil segmentasi merupakan segmen vaskular. FP diperoleh ketika suatu piksel pada citra *ground truth* merupakan latar belakang gambar, namun piksel hasil segmentasi merupakan segmen vaskular. FN diperoleh ketika suatu piksel pada citra *ground truth* adalah segmen vaskular, namun piksel hasil segmentasi adalah latar belakang gambar. Sedangkan TN diperoleh ketika suatu piksel pada citra *ground truth* dan citra hasil segmentasi merupakan latar belakang gambar.

Error diperoleh dengan membandingkan jumlah piksel yang merupakan FP dan FN dengan jumlah semua piksel seperti yang ditunjukkan pada Persamaan 5.1.

$$error = \frac{FP+FN}{TP+FP+FN+TN} \quad (5.1)$$

Nilai *error* yang dihasilkan dari Persamaan 5.1 berbanding terbalik dengan nilai kebenaran. Semakin kecil *error* maka semakin tinggi nilai kebenaran, sebaliknya ketika nilai *error* semakin besar maka nilai kebenaran semakin rendah. Selain *error*, dapat pula digunakan akurasi yang ditunjukkan pada persamaan 5.2. Akurasi berbanding lurus dengan nilai kebenaran, dimana semakin tinggi akurasi maka semakin tinggi nilai kebenaran.

$$akurasi = \frac{TP+TN}{TP+FP+FN+TN} \quad (5.2)$$

5.2.3 Uji Coba dan Evaluasi Kebenaran Citra Hasil Segmentasi

Uji coba dilakukan dengan menghitung nilai *error* dari 40 buah citra yang terdapat pada basis data DRIVE. Berdasarkan hasil pengujian, nilai rata-rata *error* untuk 40 citra berdasarkan citra *ground truth* adalah 4.964% atau akurasi kebenaran

segmentasi mencapai 95.04%. Karena nilai *error* cukup rendah dan akurasi tinggi, maka nilai uji kebenaran tinggi. Oleh karena itu metode praproses serta metode segmentasi vaskular citra retina dianggap cukup baik dan program bekerja sebagaimana seharusnya.

Uji coba selanjutnya dilakukan dengan membandingkan metode segmentasi yang digunakan dengan metode segmentasi yang sudah ada. Metode pembanding yang digunakan adalah metode dari Soares dkk serta metode dari Fraz dkk. Soares dkk menggunakan metode praproses dengan perluasan area objek retina. Sedangkan metode segmentasi yang digunakan adalah metode *supervised* dengan menerapkan *2-D Morlet wavelet* dan *Gaussian mixture model Bayesian classifier*. Metode yang ditawarkan Soares dkk telah dijelaskan pada subbab 2.1.1, 2.2.1, 2.2.2 dan 2.2.3. Di lain pihak Fraz dkk menggunakan metode segmentasi dengan menggunakan *First-order Derivative of Gaussian* (FoDoG) yang dikombinasikan dengan bentuk pembuluh darah dan peta orientasi. Metode yang ditawarkan Fraz dkk telah dijelaskan pada subbab 2.2.4 dan 2.2.5.

Dengan menggunakan nilai akurasi, hasil yang diperoleh dengan menggunakan metode dari Soares dkk, Fraz dkk, serta metode yang digunakan pada Tugas Akhir dapat dilihat pada Tabel 5.2. Berdasarkan Tabel 5.2, akurasi yang diperoleh menggunakan metode yang digunakan pada Tugas Akhir ini lebih tinggi dibandingkan metode lainnya. Hal ini menunjukkan bahwa alur dari metode praproses dan segmentasi yang digunakan benar, sehingga citra segmentasi yang dihasilkan mampu mencapai akurasi yang sama baiknya dengan metode lainnya.

Tabel 5. 2 Hasil perbandingan akurasi segmentasi dengan berbagai metode

Metode	Akurasi (%)
Soares dkk	94.66
Fraz dkk	94.30
Metode yang ditawarkan	95.04

5.3. Uji Coba Klasifikasi Segmen Vaskular Normal dan Abnormal

Pada uji coba ini akan dilakukan pengujian terhadap kebenaran dan kesesuaian metode klasifikasi segmen vaskular retina baik tahap pemodelan maupun tahap pengenalan pada sistem yang telah dibuat. Uji coba kebenaran bertujuan untuk menunjukkan bahwa program telah dapat berjalan sebagaimana seharusnya.

5.3.1. Data Uji Coba Kebenaran Klasifikasi Segmen Vaskular Normal dan Abnormal

Data yang digunakan pada tahap uji coba kebenaran hasil klasifikasi berupa vektor fitur segmen vaskular beserta label kelas yang diperoleh dari 53 buah citra retina pada basis data STARE. 53 data yang dipilih memiliki label normal dan penyakit *Proliferative Diabetic Rethinopathy* (PDR). Data dengan label penyakit PDR digunakan karena penyakit ini menyebabkan struktur vaskular retina menjadi abnormal. Dari 53 data citra tersebut, 34 data terdapat pada kelas segmen vaskular normal, sedangkan 19 data lainnya terdapat pada kelas segmen vaskular abnormal.

Vektor fitur segmen vaskular kemudian diperbaiki dengan LFDA sehingga menghasilkan vektor fitur tereduksi dengan jumlah dimensi yaitu 3 dimensi, 4 dimensi, 5 dimensi, dan 6 dimensi. Untuk melakukan klasifikasi, terlebih dahulu dibuat suatu data *testing* dan data *training*. Data *testing* pada uji coba ini menggunakan keseluruhan data dari basis data STARE sejumlah 53 data. Sedangkan data *training* akan dipilih dari 53 citra yang berasal dari basis data STARE.

Pembuatan data *training* dilakukan dengan memilih 24 data dengan masing-masing 12 data yang terdapat pada kelas normal dan 12 data pada kelas abnormal. Dilakukan sebanyak 50 kombinasi pemilihan 24 data sebagai data *training*. Kemudian dari berbagai kombinasi tersebut dilakukan klasifikasi

menggunakan LVQ. Data *training* yang menghasilkan *error* terkecil akan dipilih sebagai 24 data *training* optimal. Sedangkan 24 data *training* tidak optimal dipilih dari salah satu data *training* yang tidak menghasilkan *error* terkecil. Dari 24 data tersebut kemudian dihapus masing-masing 2 data untuk kelas normal dan 2 data untuk kelas abnormal, sehingga dihasilkan 20 data *training* optimal dan 20 data *training* tidak optimal.

Uji coba kebenaran klasifikasi segmen vaskular akan menggunakan empat jenis data *training* yang telah dihasilkan sebelumnya, yaitu 20 data *training* optimal, 24 data *training* optimal, 20 data *training* tidak optimal, dan 24 data *training* tidak optimal. Data-data yang digunakan pada keempat jenis data *training* tersebut akan dijelaskan pada Tabel 5.3 hingga Tabel 5.8.

Tabel 5. 3 Daftar citra pada 20 data *training* optimal (bagian pertama)

No.	Nama Citra	Kelas
1.	img1.ppm	Normal
2.	img7.ppm	Normal
3.	img9.ppm	Normal
4.	img10.ppm	Normal
5.	img12.ppm	Normal
6.	img13.ppm	Normal
7.	img19.ppm	Normal
8.	img20.ppm	Normal
9.	img30.ppm	Normal
10.	img34.ppm	Normal
11.	img37.ppm	Abnormal
12.	img38.ppm	Abnormal
13.	img39.ppm	Abnormal
14.	img41.ppm	Abnormal
15.	img42.ppm	Abnormal
16.	img43.ppm	Abnormal
17.	img46.ppm	Abnormal

Tabel 5. 4 Daftar citra pada 20 data *training* optimal (bagian kedua)

No.	Nama Citra	Kelas
18.	img47.ppm	Abnormal
19.	img51.ppm	Abnormal
20.	img53.ppm	Abnormal

Tabel 5. 5 Daftar citra pada 24 data *training* optimal (bagian pertama)

No.	Nama Citra	Kelas
1.	img1.ppm	Normal
2.	img4.ppm	Normal
3.	img7.ppm	Normal
4.	img9.ppm	Normal
5.	img10.ppm	Normal
6.	img12.ppm	Normal
7.	img13.ppm	Normal
8.	img19.ppm	Normal
9.	img20.ppm	Normal
10.	img30.ppm	Normal
11.	img33.ppm	Normal
12.	img34.ppm	Normal
13.	img35.ppm	Abnormal
14.	img37.ppm	Abnormal
15.	img38.ppm	Abnormal
16.	img39.ppm	Abnormal
17.	img41.ppm	Abnormal
18.	img42.ppm	Abnormal
19.	img43.ppm	Abnormal
20.	img46.ppm	Abnormal
21.	img47.ppm	Abnormal

Tabel 5. 6 Daftar citra pada 24 data *training* optimal (bagian kedua)

No.	Nama Citra	Kelas
22.	img50.ppm	Abnormal
23.	img51.ppm	Abnormal
24.	img53.ppm	Abnormal

Tabel 5. 7 Daftar citra pada 20 data *training* tidak optimal

No.	Nama Citra	Kelas
1.	img2.ppm	Normal
2.	img11.ppm	Normal
3.	img14.ppm	Normal
4.	img17.ppm	Normal
5.	img19.ppm	Normal
6.	img23.ppm	Normal
7.	img25.ppm	Normal
8.	img28.ppm	Normal
9.	img30.ppm	Normal
10.	img32.ppm	Normal
11.	img36.ppm	Abnormal
12.	img37.ppm	Abnormal
13.	img38.ppm	Abnormal
14.	img40.ppm	Abnormal
15.	img42.ppm	Abnormal
16.	img45.ppm	Abnormal
17.	img46.ppm	Abnormal
18.	img50.ppm	Abnormal
19.	img52.ppm	Abnormal
20.	img53.ppm	Abnormal

Tabel 5. 8 Daftar citra pada 24 data *training* tidak optimal

No.	Nama Citra	Kelas
1.	img1.ppm	Normal
2.	img2.ppm	Normal
3.	img3.ppm	Normal
4.	img11.ppm	Normal
5.	img14.ppm	Normal
6.	img17.ppm	Normal
7.	img19.ppm	Normal
8.	img23.ppm	Normal
9.	img25.ppm	Normal
10.	img28.ppm	Normal
11.	img30.ppm	Normal
12.	img32.ppm	Normal
13.	img36.ppm	Abnormal
14.	img37.ppm	Abnormal
15.	img38.ppm	Abnormal
16.	img40.ppm	Abnormal
17.	img42.ppm	Abnormal
18.	img45.ppm	Abnormal
19.	img46.ppm	Abnormal
20.	img49.ppm	Abnormal
21.	img50.ppm	Abnormal
22.	img51.ppm	Abnormal
23.	img52.ppm	Abnormal
24.	img53.ppm	Abnormal

5.3.2. Skenario Uji Coba Kebenaran Klasifikasi Segmen Vaskular Normal dan Abnormal

Data yang digunakan pada uji coba memiliki dua kelas hasil, yaitu kelas normal dan kelas abnormal. Jika segmen vaskular pada data citra merupakan segmen vaskular normal maka akan menghasilkan kelas normal. Sedangkan jika segmen

vaskular pada data citra merupakan segmen vaskular abnormal maka akan menghasilkan kelas abnormal.

Uji coba dimulai dengan memilih data *training* yang akan digunakan dari empat jenis data *training* yang ada, yaitu 20 data *training* optimal, 24 data *training* optimal, 20 data *training* tidak optimal, atau 24 data *training* tidak optimal. Data *training* tersebut akan dimodelkan menggunakan algoritma LVQ dengan memilih jumlah dimensi hasil reduksi, jumlah *mediods* atau *neuron* pada *hidden later* LVQ, dan nilai *learning rate*. Dengan menggunakan model yang telah dihasilkan, data *testing* akan diklasifikasikan ke dalam dua kelas, yaitu kelas normal dan abnormal.

Uji kebenaran dilakukan dengan evaluasi performa menggunakan *confusion matrix* yang ditampilkan pada Tabel 5.9.

Tabel 5.9 *Confusion matrix* untuk klasifikasi segmen vaskular

Prediksi	Kenyataan	
	Abnormal	Normal
Abnormal	TN	FN
Normal	FP	TP

Berdasarkan Tabel 5.8, TN diperoleh ketika dalam kenyataan dan hasil prediksi segmen vaskular merupakan segmen vaskular. FN diperoleh ketika dalam kenyataan segmen vaskular merupakan segmen vaskular normal namun dideteksi sebagai segmen vaskular abnormal. FP diperoleh ketika dalam kenyataan segmen vaskular merupakan segmen vaskular namun dideteksi sebagai segmen vaskular normal. Sedangkan TP diperoleh ketika dalam kenyataan segmen vaskular merupakan segmen vaskular normal dan dideteksi sebagai segmen vaskular normal.

Nilai performa diperoleh dengan menghitung nilai *error* dan akurasi. *Error* diperoleh dengan menghitung jumlah data *testing* yang mengalami kesalahan klasifikasi dibandingkan

dengan jumlah semua data *testing* seperti pada Persamaan 5.1. Dan akurasi dihitung menggunakan Persamaan 5.2.

Agar hasil uji coba kebenaran klasifikasi segmen vaskular normal dan abnormal lebih akurat, maka uji coba akan dilakukan dengan beberapa kali perulangan. Dari perulangan tersebut akan diambil nilai *error* terkecil, nilai terbesar, rata-rata, varian, dan standar deviasi.

5.3.3. Uji Coba dan Evaluasi Kebenaran Klasifikasi Segmen Vaskular Normal dan Abnormal

Uji coba dilakukan dengan menghitung nilai *error* dari 53 data *testing*. Untuk data *training*, digunakan keempat jenis data yaitu 20 data *training* optimal, 24 data *training* optimal, 20 data *training* tidak optimal, dan 24 data *training* tidak optimal.

Hasil uji coba kebenaran diperoleh dengan menggunakan dimensi hasil reduksi sejumlah 3 dimensi, 12 buah *mediods*, dan *learning rate* sebesar 0.1. Nilai dari ketiga parameter ini dipilih berdasarkan percobaan dimana nilai tersebut membutuhkan waktu eksekusi yang relatif cepat dibandingkan dengan nilai lainnya. Uji coba kebenaran klasifikasi segmen vaskular normal dan abnormal diulang sebanyak 20 percobaan sehingga menghasilkan performa berdasarkan nilai yang ditunjukkan pada Tabel 5.10.

Berdasarkan hasil pengujian yang telah dilakukan, nilai *error* terbaik adalah 3.77% yang dihasilkan oleh 20 data *training* optimal dan 24 data *training* optimal. Sedangkan untuk 20 data *training* tidak optimal dan 24 data *training* tidak optimal nilai *error* lebih besar disbanding data optimal, *error* yang dihasilkan masing-masing adalah 13.21% dan 11.32%.

20 data *training* optimal dan 20 data *training* tidak optimal merupakan *subset* dari 24 data *training* optimal dan 24 data *training* tidak optimal. Berdaraskan uji coba dapat diketahui bahwa subset dari data *training* yang telah optimal mampu menghasilkan performa yang sama baiknya dengan *superset*-nya. Sedangkan jika data *training* tidak optimal, maka *subset* dari data

tersebut juga akan menghasilkan performa yang kurang baik bahkan menghasilkan *error* lebih besar dari yang dihasilkan *superset*-nya.

Tabel 5. 10 Hasil percobaan dengan 4 jenis data *training*

Nilai <i>Error</i>	Jenis data <i>training</i>			
	20 data optimal	24 data optimal	20 data tidak optimal	24 data tidak optimal
Minimum	0.0377	0.0377	0.1321	0.1132
Maksimum	0.1509	0.1132	0.2075	0.2075
Rata-rata	0.0868	0.0651	0.1575	0.1566
Varian	0.0010	0.0003	0.0004	0.0007
Standar Deviasi	0.0320	0.0178	0.0206	0.0274

Dari data *training* optimal tersebut, *confusion matrix* yang dihasilkan akan ditunjukkan pada Tabel 5.11.

Tabel 5. 11 *Matrix confusion* dari data *training* optimal

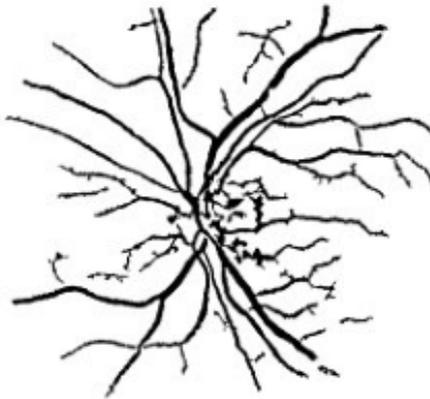
Prediksi	Kenyataan	
	Abnormal	Normal
Abnormal	17	0
Normal	2	34

Berdasarkan data dari basis data STARE, salah satu data yang merupakan TN adalah *img35.ppm* dengan citra hasil segmentasi seperti yang ditunjukkan pada Gambar 5.1. Sedangkan data yang merupakan FP adalah *img40.ppm* dengan citra hasil segmentasi seperti yang ditunjukkan pada Gambar 5.2. Untuk data yang merupakan TP adalah *img2.ppm* dengan citra hasil segmentasi seperti yang ditunjukkan pada Gambar 5.3. Sedangkan tidak ada

data yang termasuk FN, karena tidak terjadi kesalahan klasifikasi dimana segmen vaskular normal dideteksi sebagai abnormal.



Gambar 5. 1 Contoh data citra yang merupakan TN



Gambar 5. 2 Contoh data citra yang merupakan FP



Gambar 5. 3 Contoh data citra yang merupakan TP

Citra yang merupakan TN secara visual memiliki segmen vaskular yang cenderung lebih tipis dan memiliki banyak segmen kecil. Citra yang merupakan TP secara visual memiliki segmen vaskular yang lebih tebal serta jumlah segmen yang relatif lebih sedikit dibandingkan citra dengan vaskular abnormal. Sedangkan citra yang merupakan FP secara visual memiliki segmen yang lebih mirip dengan segmen vaskular normal. Citra retina pada Gambar 5.2 memiliki segmen vaskular yang cenderung tebal dibandingkan segmen vaskular abnormal, selain itu jumlah segmen vaskular juga cenderung sedikit. Hal ini memungkinkan terjadinya kesalahan klasifikasi lebih tinggi karena secara visual Gambar 5.2 lebih mirip dengan segmen vaskular normal.

5.3.4. Uji Coba dan Evaluasi Kebenaran Klasifikasi Berdasarkan Jumlah Dimensi Hasil Reduksi

Uji coba dilakukan dengan menghitung nilai *error* dari 53 data testing menggunakan beberapa nilai yang mewakili jumlah dimensi hasil reduksi. Jumlah dimensi yang digunakan yaitu 3 dimensi, 4 dimensi, 5 dimensi, dan 6 dimensi. Jumlah dimensi

yang digunakan untuk uji coba harus lebih kecil dibandingkan jumlah fitur pada vektor fitur yang berjumlah 7 dimensi.

Data *training* yang digunakan untuk uji coba adalah 24 data *training* optimal dan 24 data *training* tidak optimal. Hasil uji coba kebenaran diperoleh dengan menggunakan 12 buah *methods* dan *learning rate* sebesar 0.1. Nilai dari kedua parameter ini dipilih berdasarkan percobaan dimana nilai tersebut membutuhkan waktu eksekusi yang relatif cepat dibandingkan dengan nilai lainnya. Uji coba ini bertujuan untuk mengetahui pengaruh jumlah dimensi hasil reduksi terhadap nilai *error* klasifikasi, baik untuk data optimal maupun data tidak optimal.

Tabel 5. 12 Hasil percobaan dengan 24 data *training* optimal dengan perubahan jumlah dimensi hasil reduksi

Nilai <i>Error</i>	Jumlah Dimensi Hasil Reduksi			
	3 dimensi	4 dimensi	5 dimensi	6 dimensi
Minimum	0.0377	0.0377	0.0377	0.0377
Maksimum	0.1132	0.1509	0.1321	0.1509
Rata-rata	0.0650	0.0707	0.0839	0.0990
Varian	0.0003	0.0008	0.0008	0.0007
Standar Deviasi	0.0178	0.0286	0.0284	0.0273

Tabel 5. 13 Hasil percobaan dengan 24 data *training* tidak optimal dengan perubahan dimensi reduksi

Nilai <i>Error</i>	Jumlah Dimensi Hasil Reduksi			
	3 dimensi	4 dimensi	5 dimensi	6 dimensi
Minimum	0.1132	0.0943	0.0755	0.0755
Maksimum	0.2075	0.2264	0.1509	0.1509
Rata-rata	0.1566	0.1613	0.1226	0.1037
Varian	0.0007	0.0010	0.0005	0.0005
Standar Deviasi	0.0274	0.0326	0.0233	0.0240

Uji coba kebenaran klasifikasi segmen vaskular normal dan abnormal diulang sebanyak 20 percobaan sehingga menghasilkan performa berupa nilai *error* berdasarkan nilai yang ditunjukkan pada Tabel 5.12 dan 5.13. Berdasarkan hasil uji coba, perubahan jumlah dimensi hasil reduksi tidak berpengaruh pada data *training* yang optimal. Nilai *error* terbaik yang diperoleh sama yaitu 3.77% baik untuk reduksi 3 dimensi, 4 dimensi, 5 dimensi, maupun 6 dimensi. Sedangkan untuk data *training* yang tidak optimal, perubahan dimensi fitur mempengaruhi nilai *error*. Nilai *error* akan cenderung menurun seiring dengan penambahan jumlah dimensi hasil reduksi. Nilai *error* terkecil diperoleh pada jumlah dimensi hasil reduksi adalah 5 dimensi dan 6 dimensi. Nilai *error* tertinggi terjadi untuk jumlah dimensi hasil reduksi adalah 3 dimensi.

5.3.5. Uji Coba dan Evaluasi Kebenaran Klasifikasi Berdasarkan Jumlah *Mediods*

Uji coba dilakukan dengan menghitung nilai *error* dari 53 data *testing* menggunakan beberapa nilai yang mewakili jumlah *mediods*. Jumlah *mediods* yang digunakan adalah 6 *mediods*, 8 *mediods*, 10 *mediods*, dan 12 *mediods*. Jumlah *mediods* dipilih dengan nilai yang lebih besar dari jumlah kelas dari data.

Data *training* yang digunakan untuk uji coba adalah 24 data *training* optimal dan 24 data *training* tidak optimal. Hasil uji coba kebenaran diperoleh dengan menggunakan jumlah dimensi hasil reduksi adalah 3 dimensi dan *learning rate* yang digunakan sebesar 0.1. Nilai dari kedua parameter ini dipilih berdasarkan percobaan dimana nilai tersebut membutuhkan waktu eksekusi yang relatif cepat dibandingkan dengan nilai lainnya. Uji coba ini bertujuan untuk mengetahui pengaruh jumlah *mediods* terhadap nilai *error* klasifikasi, baik untuk data optimal maupun data tidak optimal. Uji coba kebenaran klasifikasi segmen vaskular normal dan abnormal diulang sebanyak 20 percobaan sehingga menghasilkan performa berdasarkan nilai *error* yang ditunjukkan pada Tabel 5.14 dan 5.15.

Tabel 5. 14 Hasil percobaan dengan 24 data *training* optimal dengan perubahan jumlah *mediods*

Nilai Error	Jumlah <i>mediods</i>			
	6 <i>mediods</i>	8 <i>mediods</i>	10 <i>mediods</i>	12 <i>mediods</i>
Minimum	0.0377	0.0377	0.0377	0.0377
Maksimum	0.1321	0.1321	0.1321	0.1132
Rata-rata	0.0669	0.0726	0.0716	0.0651
Varian	0.0011	0.0009	0.0008	0.0003
Standar Deviasi	0.0332	0.0307	0.0284	0.0178

Tabel 5. 15 Hasil percobaan dengan 24 data *training* tidak optimal dengan perubahan jumlah *mediods*

Nilai Error	Jumlah <i>mediods</i>			
	6 <i>mediods</i>	8 <i>mediods</i>	10 <i>mediods</i>	12 <i>mediods</i>
Minimum	0.1321	0.1321	0.1321	0.1132
Maksimum	0.1887	0.2075	0.2075	0.2075
Rata-rata	0.1528	0.1678	0.1585	0.1566
Varian	0.0002	0.0004	0.0008	0.0007
Standar Deviasi	0.0149	0.0202	0.0283	0.0274

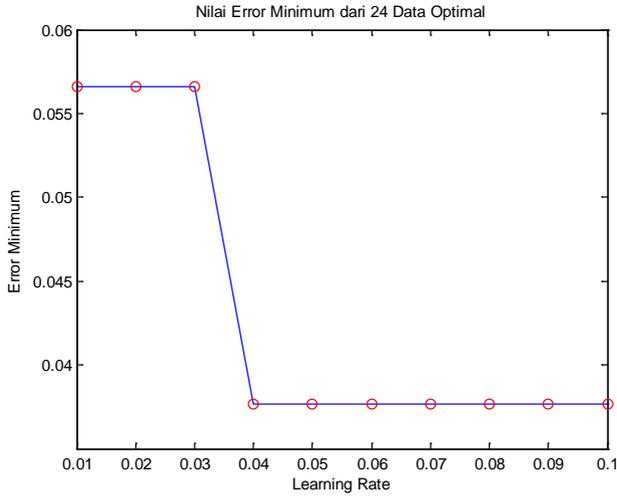
Berdasarkan hasil uji coba, perubahan jumlah *mediods* tidak mempengaruhi nilai *error* pada data *training* yang optimal. Nilai *error* terbaik yang diperoleh adalah 3.77% baik untuk jumlah *mediods* sebanyak 6 buah, 8 buah, 10 buah, maupun 12 buah *mediods*. Sedangkan untuk data *training* yang tidak optimal, perubahan jumlah *mediods* mempengaruhi nilai *error*. Dari Tabel 5.15 terlihat bahwa nilai *error* akan menurun seiring dengan bertambahnya jumlah *mediods*. Nilai *error* terbaik untuk 6, 8 dan 10 buah *mediods* yaitu 13.21%. Namun untuk jumlah *mediods* sebanyak 12 buah, nilai *error* terbaik yang dapat dihasilkan menurun menjadi 13.21%.

5.3.6. Uji Coba dan Evaluasi Kebenaran Klasifikasi Berdasarkan Nilai *Learning Rate*

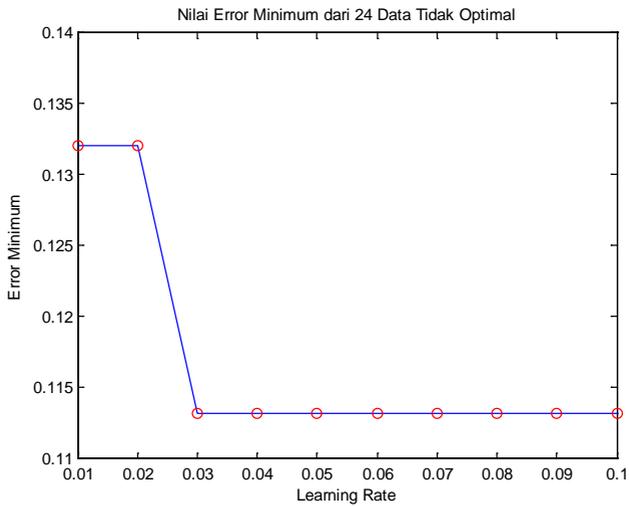
Uji coba dilakukan dengan menghitung nilai *error* dari 53 data *testing* dengan nilai *learning rate* antara 0.01 hingga 0.1. Nilai dari parameter *learning rate* dipilih berdasarkan percobaan dimana nilai optimal berada pada sekitar 0.1.

Data *training* yang digunakan untuk uji coba adalah 24 data *training* optimal dan 24 data *training* tidak optimal. Hasil uji coba kebenaran diperoleh dengan menggunakan jumlah dimensi hasil reduksi adalah 3 dimensi dan 12 buah *mediods*. Nilai dari kedua parameter ini dipilih berdasarkan percobaan dimana nilai tersebut membutuhkan waktu eksekusi yang relatif cepat dibandingkan dengan nilai lainnya. Uji coba ini bertujuan untuk mengetahui pengaruh nilai *learning rate* terhadap nilai *error* klasifikasi, baik untuk data optimal maupun data tidak optimal.

Uji coba kebenaran klasifikasi segmen vaskular normal dan abnormal diulang sebanyak 10 percobaan, nilai *error* terbaik dari 10 percobaan tersebut akan digunakan untuk analisis sehingga menghasilkan grafik yang ditunjukkan pada Gambar 5.4 dan Gambar 5.5. Berdasarkan hasil uji coba, nilai *learning rate* dapat mempengaruhi nilai *error* baik untuk data *training* optimal dan data *training* tidak optimal. Untuk 24 data *training* optimal, nilai *error* terbaik yang dihasilkan untuk *learning rate* antara 0.04 hingga 0.1 lebih baik dibandingkan nilai *error* untuk *learning rate* antara 0.01 hingga 0.03. Pada 24 data *training* tidak optimal nilai *error* yang dihasilkan juga akan terpengaruh. Nilai *error* untuk *learning rate* antara 0.03 hingga 0.1 lebih baik dibandingkan dengan nilai *error* yang dihasilkan dengan *learning rate* 0.01 dan 0.02. Berdasarkan kedua data *training* tersebut, nilai *error* akan semakin kecil dengan kenaikan *learning rate*. Nilai *learning rate* terbaik terdapat pada sekitar 0.1.



Gambar 5. 4 Nilai *error* untuk berbagai *learning rate* dari 24 data *training* optimal



Gambar 5. 5 Nilai *error* untuk berbagai *learning rate* dari 24 data *training* tidak optimal

5.4. Uji Coba Waktu Eksekusi Perangkat Lunak

Pada uji coba ini akan dilakukan pengujian terhadap waktu eksekusi perangkat lunak mulai dari tahap praproses hingga tahap klasifikasi. Uji coba waktu eksekusi bertujuan untuk mengetahui waktu yang dibutuhkan oleh perangkat lunak untuk memroses citra masukan hingga diperoleh label kelas hasil klasifikasi.

5.4.1. Data Uji Coba Waktu Eksekusi Perangkat Lunak

Data yang digunakan pada tahap uji coba waktu eksekusi perangkat lunak menggunakan citra retina dari basis data STARE. Data citra ini diproses hingga menghasilkan vektor fitur, kemudian vektor fitur ini direduksi, hasil reduksi dimensi dengan dimensi reduksi sebanyak 3 dimensi selanjutnya digunakan untuk uji coba ini.

5.4.2. Skenario Uji Coba Waktu Eksekusi Perangkat Lunak

Uji coba ini dilakukan dengan menghitung waktu eksekusi untuk setiap bagian proses dari perangkat lunak. Bagian proses tersebut adalah praproses dan segmentasi, proses ekstraksi fitur, serta proses klasifikasi. Waktu eksekusi dihitung berdasarkan waktu yang dibutuhkan oleh suatu citra mulai dari prasoses dan segmentasi, hingga citra tersebut berhasil dideteksi kelasnya sebagai vaskular normal atau vaskular abnormal.

Waktu eksekusi atau *running time* diperoleh dengan menggunakan Persamaan 5.3.

$$\begin{aligned}
 \text{running time} = & \text{waktu praproses} + \text{waktu segmentasi} + \\
 & \frac{\text{waktu ekstraksi fitur}}{\text{jumlah data}} + \frac{\text{waktu perbaikan fitur}}{\text{jumlah data}} + \\
 & \text{waktu training} + \frac{\text{waktutesting}}{\text{jumlah data}}
 \end{aligned} \tag{5.3}$$

Untuk *training* menggunakan LVQ, digunakan 24 data *training* optimal dengan jumlah dimensi hasil reduksi adalah 3 dimensi dan nilai *learning rate* adalah jumlah 0.1. Nilai dari jumlah dimensi hasil reduksi dan *learning rate* dipilih berdasarkan percobaan dimana nilai tersebut membutuhkan waktu eksekusi yang relatif cepat. Sedangkan untuk jumlah *mediods*, digunakan beberapa variasi yaitu 6 *mediods*, 8 *mediods*, 10 *mediods*, dan 12 *mediods*.

5.4.3. Uji Coba dan Evaluasi Waktu Eksekusi Perangkat Lunak

Uji coba dilakukan dengan menghitung *running time* menggunakan data citra *img2.ppm* untuk tahap praproses, segmentasi, ekstraksi fitur dan pengenalan (*testing*). Sedangkan untuk bagian *training* digunakan 24 data *training*.

Berdasarkan uji coba yang telah dilakukan, nilai *running time* yang diperoleh akan ditunjukkan pada Tabel 5.16.

Tabel 5. 16 Hasil Uji Coba Waktu Eksekusi Perangkat Lunak

Proses		<i>Running time</i> (detik)
Praproses		0.37
Segmentasi		3.48
Ekstraksi Fitur		1.02
Perbaikan Fitur		0.06
<i>Training LVQ</i>		
Jumlah <i>mediods</i>	6	9.28
	8	19.74
	10	32.08
	12	8.18
<i>Testing</i>		0.01

Waktu eksekusi tercepat terjadi pada bagian *testing*. Sedangkan waktu terlama yang dibutuhkan terjadi pada proses *training LVQ*.

Training LVQ membutuhkan waktu yang cukup lama karena membutuhkan beberapa kali perulangan (iterasi). Iterasi ini akan terus dilakukan hingga diperoleh model LVQ yang optimal. Selain itu jumlah *mediods* juga mempengaruhi waktu eksekusi. Ketika jumlah *mediods* dapat mewakili jumlah sub kelas dari kelas normal dan abnormal, maka pembuatan model yang optimal akan lebih cepat, sehingga waktu eksekusi juga lebih cepat. Sedangkan proses-proses lain selain *training* LVQ membutuhkan waktu yang relatif cepat karena tidak membutuhkan iterasi. Waktu eksekusi total yang dibutuhkan sebuah citra untuk menjalani semua proses dalam perangkat lunak adalah antara 13 detik hingga 37 detik seperti yang ditunjukkan pada Tabel 5.17.

Tabel 5. 17 Waktu Eksekusi yang Dibutuhkan untuk Berbagai Jumlah *Medioids*

Jumlah <i>medioids</i>	<i>Running time</i> (detik)
6 <i>medioids</i>	14.22
8 <i>medioids</i>	24.68
10 <i>medioids</i>	37.02
12 <i>medioids</i>	13.12

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap pembuatan program praproses dan segmentasi vaskular retina serta ekstraksi fitur dan klasifikasi segmen vaskular retina dapat diambil kesimpulan sebagai berikut:

1. Metode praproses dan metode segmentasi segmen vaskular yang digunakan telah memiliki alur proses yang benar, terbukti dengan nilai akurasi yang lebih baik dibandingkan dengan metode segmentasi yang telah ada sebelumnya.
2. Metode praproses dan metode segmentasi segmen vaskular dengan menggunakan *Isotropic Undecimated Wavelet Transform* relatif handal untuk menghasilkan segmentasi vaskular dari citra retina. Hal ini terbukti dengan hasil akurasi sebesar 95,4% untuk perbandingan citra hasil segmentasi dan citra *ground truth* dari basis data DRIVE.
3. Metode klasifikasi menggunakan m-Methods multivariat dapat memodelkan dan mengenali segmen vaskular retina dengan baik. Hasil akurasi terbaik untuk klasifikasi dengan metode m-Methods multivariat mencapai 96.2% atau *error* sebesar 3.77% untuk 53 buah data dari basis data STARE.
4. Untuk data *training* yang baik dan optimal, perubahan nilai dimensi reduksi dan jumlah *methods* tidak akan mempengaruhi hasil akurasi. Hal ini membuktikan bahwa klasifikasi menggunakan metode m-Methods multivariat

dengan algoritma klasifikasi LVQ relatif handal terhadap perubahan parameter yang dilakukan.

5. Dengan menggunakan data *training* yang baik dan optimal, maka subset dari data *training* tersebut juga akan menghasilkan klasifikasi yang baik.
6. Nilai *learning rate* dapat memberikan pengaruh terhadap hasil akurasi. Nilai *learning rate* terbaik berada pada sekitar nilai 0.1.
7. Waktu eksekusi yang dibutuhkan untuk sebuah citra dapat dideteksi sebagai citra normal atau abnormal antara 13 detik hingga 37 detik.

6.2. Saran

Saran yang diberikan untuk pengembangan aplikasi klasifikasi segmen vaskular retina pada Tugas Akhir ini antara lain:

1. Berdasarkan hasil uji coba, terdapat beberapa data dengan kelas abnormal yang diklasifikasi sebagai data normal. Oleh karena itu dapat dilakukan penambahan fitur yang mampu mengenali segmen vaskular abnormal dengan lebih baik.
2. Untuk meningkatkan hasil uji coba segmentasi, dapat dilakukan pengembangan metode segmentasi yang digunakan antara lain perbaikan transformasi wavelet.

DAFTAR PUSTAKA

- [1] crvo_my, "Neovascularization", Central Retinal Vein Occlusion, 2011. [Online]. Available:<https://sites.google.com/site/crvoquickstart/neovascularization>. [Accessed 21 Februari 2014].
- [2] Shahid Razzaq Shehzad Khalid, "Frameworks for multivariate m-Medioids based modeling and classification in Euclidean and general feature spaces," *Pattern Recognition*, vol. 45, no. 3, pp. 1092–1103, Maret 2012.
- [3] J.J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, B. van Ginneken, "Ridge based vessel segmentation in color images of the retina", *IEEE Transactions on Medical Imaging*, 2004, vol. 23, pp. 501-509.
- [4] Hoover, STA RE d atabase, 2013. [Online]. Available: <http://www.ces.clemson.edu/ahover/stare>. [Accessed 5 Februari 2014].
- [5] Wikipedia, "Erosion (morphology)", Wikipedia, 8 Juni 2014. [Online]. Available:[http://en.wikipedia.org/wiki/Erosion_\(morphology\)](http://en.wikipedia.org/wiki/Erosion_(morphology)). [Accessed 20 Februari 2014].
- [6] Jean-Luc Starck, Jalal Fadili, Fionn Murtagh, "The Undecimated Wavelet Decomposition and its Reconstruction", *IEEE Transaction on Image Processing*, vol. 16, pp.297-309, Februari 2007.
- [7] Shehzad Khalid, Anam Tariq, M. Younus Javed M. Usman Akram, "Detection of neovascularization in retinal images using multivariate m-Medioids based classifier," *Journal Computerized Medical Imaging and Graphics*, vol. 37, no. 5-6, pp. 346-357, September 2013.
- [8] K. Ming Leung, "Learning Vector Quantization", Department of Computer and Information Science, Polytechnic University, New York, April 2009.

- [9] Pete Bankhead, “ARIA: Automated Retinal Image Analyzer v1.0”, Centre for Vision and Vascular Science, Queen's University of Belfast, UK, Desember 2011.
- [10] J.V.B. Soares, J.J.G. Leandro, R.M. Cesar, H.F. Jelinek, M.J. Cree, “Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification,” *IEEE Transactions on Medical Imaging*, vol. 25, pp. 1214–1222, 2006.
- [11] M.M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, Christopher G. Owen, Alicja R. Rudnicka, S.A. Barman, “Retinal Vessel Extraction Using First-Order Derivative of Gaussian and Morphological Processing,” *Springer-Verlag*, pp. 410–420, 2010.

BIODATA PENULIS

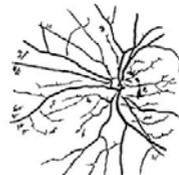


Wilda Imama Sabilla, lahir di Malang pada tanggal 29 Agustus 1992. Penulis telah menempuh pendidikan formal di SDN II Toyomarto (1998-2004), SMPN 1 Singosari (2004-2007), dan SMAN 1 Lawang (2007-2010). Pada tahun 2010 penulis diterima di S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Di jurusan Teknik Informatika, penulis mengambil bidang minat Komputasi Cerdas dan Visualisasi (KCV).

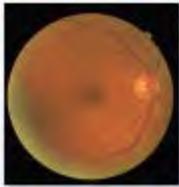
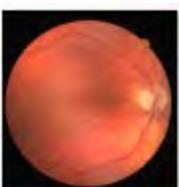
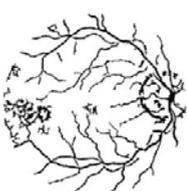
Selama masa kuliah, penulis aktif dalam pada organisasi mahasiswa tingkat jurusan, yaitu Himpunan Mahasiswa Teknik Computer-Informatika (HMTC). Di HMTC penulis pernah menjabat sebagai staff departemen kewirausahaan serta sekretaris departemen media informasi. Penulis juga mengikuti beberapa Latihan Ketrampilan Manajemen Mahasiswa (LKMM) yaitu praTD dan TD. Selain aktif dalam berorganisasi, penulis juga mengikuti beberapa perlombaan dan menjadi finalis diantaranya finalis IT Contest ITS Expo dalam kategori Pengembangan Game pada tahun 2012 dan finalis GEMASTIK 2013 kategori data mining. Penulis dapat dihubungi melalui *email*: wildaimama@gmail.com.

LAMPIRAN A

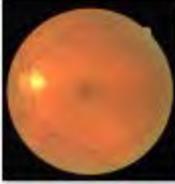
Tabel A. 1 Citra Hasil Segmentasi dan Citra *Ground Truth* dari Basis Data DRIVE (Bagian Pertama)

Citra Asli/ Nama Citra	Citra <i>Ground Truth</i>	Citra Hasil Segmentasi	<i>Error</i> (%)
 01_test.tif			4.453
 02_test.tif			4.449
 03_test.tif			5.320
 04_test.tif			4.879

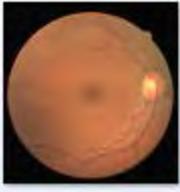
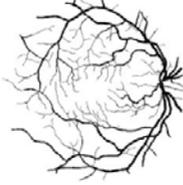
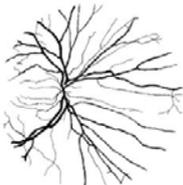
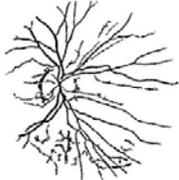
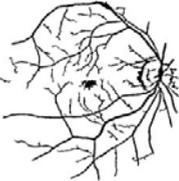
Tabel A. 2 Citra Hasil Segmentasi dan Citra *Ground Truth* dari Basis Data DRIVE (Bagian Kedua)

Citra Asli/ Nama Citra	Citra <i>Ground Truth</i>	Citra Hasil Segmentasi	<i>Error</i> (%)
 05_test.tif			4.703
 06_test.tif			4.855
 07_test.tif			5.542
 08_test.tif			6.020

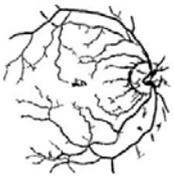
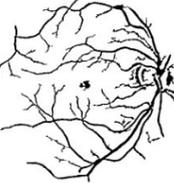
Tabel A. 3 Citra Hasil Segmentasi dan Citra *Ground Truth* dari Basis Data DRIVE (Bagian Ketiga)

Citra Asli/ Nama Citra	Citra <i>Ground Truth</i>	Citra Hasil Segmentasi	<i>Error</i> (%)
 09_test.tif			4.692
 10_test.tif			4.618
 11_test.tif			5.116
 12_test.tif			4.660

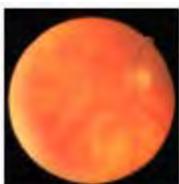
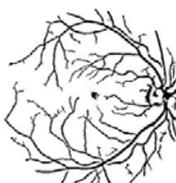
Tabel A. 4 Citra Hasil Segmentasi dan Citra *Ground Truth* dari Basis Data DRIVE (Bagian Keempat)

Citra Asli/ Nama Citra	Citra <i>Ground Truth</i>	Citra Hasil Segmentasi	<i>Error</i> (%)
 13_test.tif			5.236
 14_test.tif			4.818
 15_test.tif			4.888
 16_test.tif			4.597

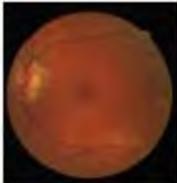
Tabel A. 5 Citra Hasil Segmentasi dan Citra *Ground Truth* dari Basis Data DRIVE (Bagian Kelima)

Citra Asli/ Nama Citra	Citra <i>Ground Truth</i>	Citra Hasil Segmentasi	<i>Error</i> (%)
 17_test.tif			4.680
 18_test.tif			4.418
 19_test.tif			3.890
 20_test.tif			4.030

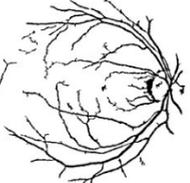
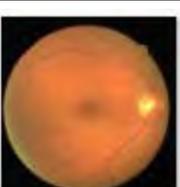
Tabel A. 6 Citra Hasil Segmentasi dan Citra *Ground Truth* dari Basis Data DRIVE (Bagian Keenam)

Citra Asli/ Nama Citra	Citra <i>Ground Truth</i>	Citra Hasil Segmentasi	<i>Error</i> (%)
 21_training.tif			4.633
 22_training.tif			5.124
 23_training.tif			6.906
 24_training.tif			5.167

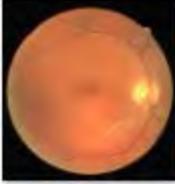
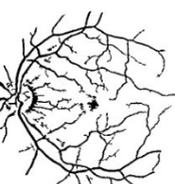
Tabel A. 7 Citra Hasil Segmentasi dan Citra *Ground Truth* dari Basis Data DRIVE (Bagian Ketujuh)

Citra Asli/ Nama Citra	Citra <i>Ground Truth</i>	Citra Hasil Segmentasi	<i>Error</i> (%)
 25_training.tif			5.602
 26_training.tif			5.043
 27_training.tif			5.191
 28_training.tif			5.328

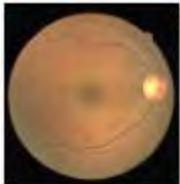
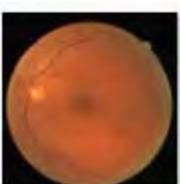
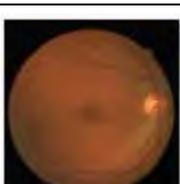
Tabel A. 8 Citra Hasil Segmentasi dan Citra *Ground Truth* dari Basis Data DRIVE (Bagian Kedelapan)

Citra Asli/ Nama Citra	Citra <i>Ground Truth</i>	Citra Hasil Segmentasi	<i>Error</i> (%)
 29_training.tif			5.062
 30_training.tif			4.651
 31_training.tif			5.310
 32_training.tif			4.485

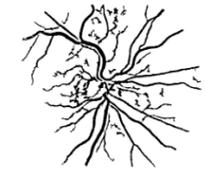
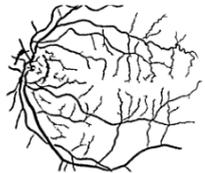
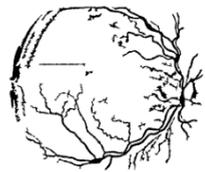
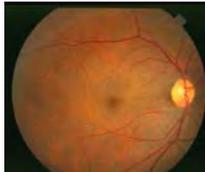
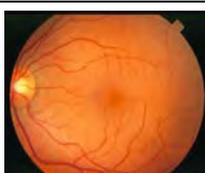
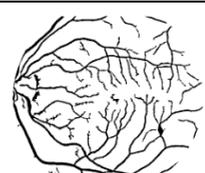
Tabel A. 9 Citra Hasil Segmentasi dan Citra *Ground Truth* dari Basis Data DRIVE (Bagian Kesembilan)

Citra Asli/ Nama Citra	Citra <i>Ground Truth</i>	Citra Hasil Segmentasi	<i>Error</i> (%)
 33_training.tif			4.399
 34_training.tif			7.275
 35_training.tif			4.575
 36_training.tif			5.392

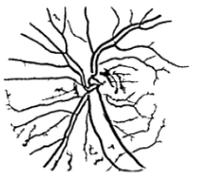
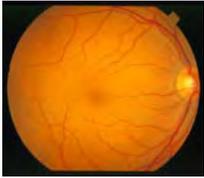
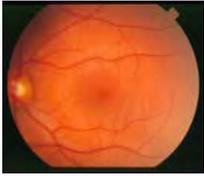
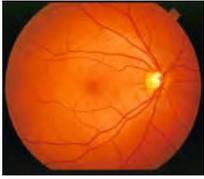
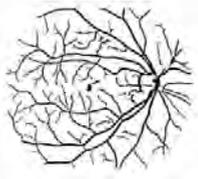
Tabel A. 10 Citra Hasil Segmentasi dan Citra *Ground Truth* dari Basis Data DRIVE (Bagian Kesepuluh)

Citra Asli/ Nama Citra	Citra <i>Ground Truth</i>	Citra Hasil Segmentasi	<i>Error</i> (%)
 37_training.tif			4.986
 38_training.tif			4.682
 39_training.tif			4.436
 40_training.tif			4.428

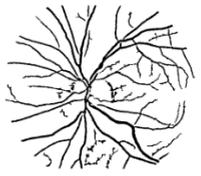
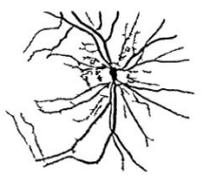
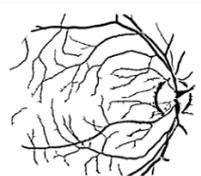
Tabel A. 11 Citra dari Basis Data STARE (Bagian Pertama)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img1.ppm			Normal
img2.ppm			Normal
img3.ppm			Normal
img4.ppm			Normal
img5.ppm			Normal

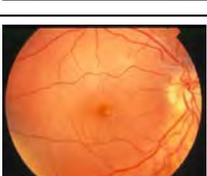
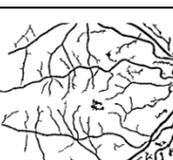
Tabel A. 12 Citra dari Basis Data STARE (Bagian Kedua)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img6.ppm			Normal
img7.ppm			Normal
img8.ppm			Normal
img9.ppm			Normal
img10.ppm			Normal

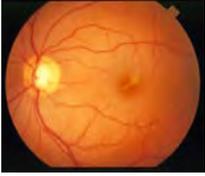
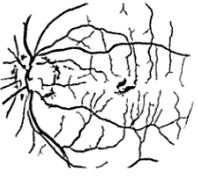
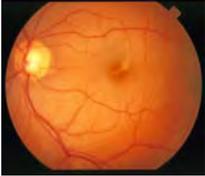
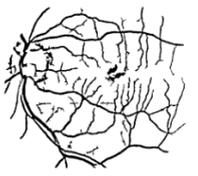
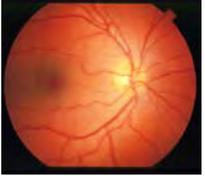
Tabel A. 13 Citra dari Basis Data STARE (Bagian Ketiga)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img11.ppm			Normal
img12.ppm			Normal
img13.ppm			Normal
img14.ppm			Normal
img15.ppm			Normal

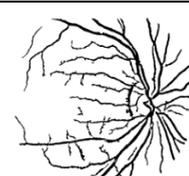
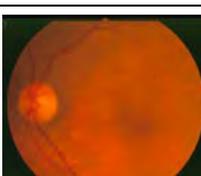
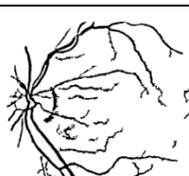
Tabel A. 14 Citra dari Basis Data STARE (Bagian Keempat)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img16.ppm			Normal
img17.ppm			Normal
img18.ppm			Normal
img19.ppm			Normal
img20.ppm			Normal

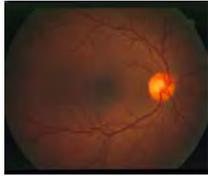
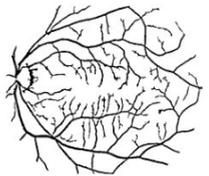
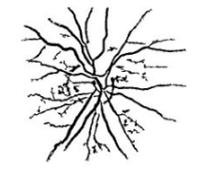
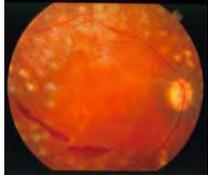
Tabel A. 15 Citra dari Basis Data STARE (Bagian Kelima)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img21.ppm			Normal
img22.ppm			Normal
img23.ppm			Normal
img24.ppm			Normal
img25.ppm			Normal

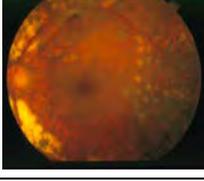
Tabel A. 16 Citra dari Basis Data STARE (Bagian Keenam)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img26.ppm			Normal
img27.ppm			Normal
img28.ppm			Normal
img29.ppm			Normal
img30.ppm			Normal

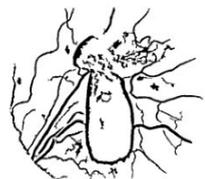
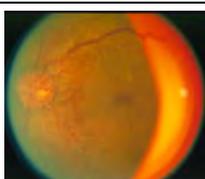
Tabel A. 17 Citra dari Basis Data STARE (Bagian Ketujuh)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img31.ppm			Normal
img32.ppm			Normal
img33.ppm			Normal
img34.ppm			Normal
img35.ppm			Abnormal

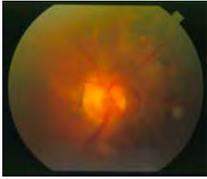
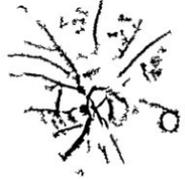
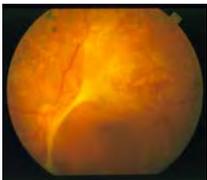
Tabel A. 18 Citra dari Basis Data STARE (Bagian Kedelapan)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img36.ppm			Abnormal
img37.ppm			Abnormal
img38.ppm			Abnormal
img39.ppm			Abnormal
img40.ppm			Abnormal

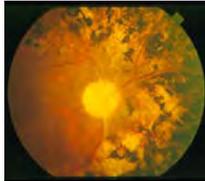
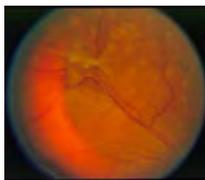
Tabel A. 19 Citra dari Basis Data STARE (Bagian Kesembilan)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img41.ppm			Abnormal
img42.ppm			Abnormal
img43.ppm			Abnormal
img44.ppm			Abnormal
img45.ppm			Abnormal

Tabel A. 20 Citra dari Basis Data STARE (Bagian Kesepuluh)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img46.ppm			Abnormal
img47.ppm			Abnormal
img48.ppm			Abnormal
img49.ppm			Abnormal
img50.ppm			Abnormal

Tabel A. 21 Citra dari Basis Data STARE (Bagian Kesebelas)

Nama citra	Citra asli	Citra Hasil Segmentasi	Kelas
img51.ppm			Abnormal
img52.ppm			Abnormal
img53.ppm			Abnormal

Tabel A. 22 Hasil Uji Coba dari 24 Data *Training* Optimal dengan 3 Dimensi Hasil Reduksi

Percobaan	Mediods	Learning Rate	Error
1	12	0.1	0.0566
2	12	0.1	0.0755
3	12	0.1	0.0566
4	12	0.1	0.0566
5	12	0.1	0.0566
6	12	0.1	0.0377
7	12	0.1	0.0755
8	12	0.1	0.0566
9	12	0.1	0.0566
10	12	0.1	0.0943
11	12	0.1	0.0566
12	12	0.1	0.0755
13	12	0.1	0.0755
14	12	0.1	0.1132
15	12	0.1	0.0566
16	12	0.1	0.0566
17	12	0.1	0.0755
18	12	0.1	0.0566
19	12	0.1	0.0755
20	12	0.1	0.0377
Minimum			0.0377
Maksimum			0.1132
Rata-rata			0.0651
Varian			0.0003
Standar Deviasi			0.0178

Tabel A. 23 Hasil Uji Coba dari 20 Data *Training* Optimal dengan 3 Dimensi Hasil Reduksi

Percobaan	Mediods	Learning Rate	Error
1	12	0.1	0.0755
2	12	0.1	0.1321
3	12	0.1	0.0943
4	12	0.1	0.0377
5	12	0.1	0.0566
6	12	0.1	0.0943
7	12	0.1	0.0943
8	12	0.1	0.1321
9	12	0.1	0.1132
10	12	0.1	0.0566
11	12	0.1	0.0566
12	12	0.1	0.1509
13	12	0.1	0.0755
14	12	0.1	0.0566
15	12	0.1	0.1321
16	12	0.1	0.0943
17	12	0.1	0.0566
18	12	0.1	0.0943
19	12	0.1	0.0566
20	12	0.1	0.0755
Minimum			0.0377
Maksimum			0.1509
Rata-rata			0.0867
Varian			0.0010
Standar Deviasi			0.0320

Tabel A. 24 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan 3 Dimensi Hasil Reduksi

Percobaan	Mediolds	Learning Rate	Error
1	12	0.1	0.1321
2	12	0.1	0.1321
3	12	0.1	0.1321
4	12	0.1	0.1698
5	12	0.1	0.2075
6	12	0.1	0.1321
7	12	0.1	0.1132
8	12	0.1	0.1698
9	12	0.1	0.1321
10	12	0.1	0.1887
11	12	0.1	0.1509
12	12	0.1	0.2075
13	12	0.1	0.1698
14	12	0.1	0.1887
15	12	0.1	0.1321
16	12	0.1	0.1509
17	12	0.1	0.1698
18	12	0.1	0.1509
19	12	0.1	0.1321
20	12	0.1	0.1698
Minimum			0.1132
Maksimum			0.2075
Rata-rata			0.1566
Varian			0.0007
Standar Deviasi			0.0274

Tabel A. 25 Hasil Uji Coba dari 20 Data *Training* Tidak Optimal dengan 3 Dimensi Hasil Reduksi

Percobaan	Mediods	Learning Rate	Error
1	12	0.1	0.1509
2	12	0.1	0.1509
3	12	0.1	0.1887
4	12	0.1	0.1698
5	12	0.1	0.1321
6	12	0.1	0.1321
7	12	0.1	0.1509
8	12	0.1	0.1509
9	12	0.1	0.1509
10	12	0.1	0.1509
11	12	0.1	0.2075
12	12	0.1	0.1509
13	12	0.1	0.1509
14	12	0.1	0.1321
15	12	0.1	0.1509
16	12	0.1	0.1887
17	12	0.1	0.1509
18	12	0.1	0.1887
19	12	0.1	0.1509
20	12	0.1	0.1509
Minimum			0.1321
Maksimum			0.2075
Rata-rata			0.1575
Varian			0.0004
Standar Deviasi			0.0206

Tabel A. 26 Hasil Uji Coba dari 24 Data *Training* Optimal dengan 4 Dimensi Hasil Reduksi

Percobaan	Mediods	Learning Rate	Error
1	12	0.1	0.0566
2	12	0.1	0.0377
3	12	0.1	0.0943
4	12	0.1	0.0566
5	12	0.1	0.0566
6	12	0.1	0.1132
7	12	0.1	0.0755
8	12	0.1	0.0377
9	12	0.1	0.0566
10	12	0.1	0.0755
11	12	0.1	0.0566
12	12	0.1	0.0943
13	12	0.1	0.0566
14	12	0.1	0.1509
15	12	0.1	0.0377
16	12	0.1	0.0377
17	12	0.1	0.0755
18	12	0.1	0.0755
19	12	0.1	0.0755
20	12	0.1	0.0943
Minimum			0.0377
Maksimum			0.1509
Rata-rata			0.0707
Varian			0.0008
Standar Deviasi			0.0286

Tabel A. 27 Hasil Uji Coba dari 24 Data *Training* Optimal dengan 5 Dimensi Hasil Reduksi

Percobaan	Mediods	Learning Rate	Error
1	12	0.1	0.0566
2	12	0.1	0.0377
3	12	0.1	0.0566
4	12	0.1	0.1132
5	12	0.1	0.0566
6	12	0.1	0.0943
7	12	0.1	0.0755
8	12	0.1	0.0566
9	12	0.1	0.1321
10	12	0.1	0.1132
11	12	0.1	0.0943
12	12	0.1	0.1132
13	12	0.1	0.0377
14	12	0.1	0.1132
15	12	0.1	0.0943
16	12	0.1	0.1132
17	12	0.1	0.0943
18	12	0.1	0.0566
19	12	0.1	0.0755
20	12	0.1	0.0943
Minimum			0.0377
Maksimum			0.1321
Rata-rata			0.0839
Varian			0.0008
Standar Deviasi			0.0284

Tabel A. 28 Hasil Uji Coba dari 24 Data *Training* Optimal dengan 6 Dimensi Hasil Reduksi

Percobaan	Mediolds	Learning Rate	Error
1	12	0.1	0.1132
2	12	0.1	0.0755
3	12	0.1	0.1321
4	12	0.1	0.1132
5	12	0.1	0.1321
6	12	0.1	0.0755
7	12	0.1	0.0377
8	12	0.1	0.0943
9	12	0.1	0.0943
10	12	0.1	0.1132
11	12	0.1	0.1321
12	12	0.1	0.0943
13	12	0.1	0.0943
14	12	0.1	0.0943
15	12	0.1	0.0943
16	12	0.1	0.1509
17	12	0.1	0.0755
18	12	0.1	0.1132
19	12	0.1	0.0943
20	12	0.1	0.0566
Minimum			0.0377
Maksimum			0.1509
Rata-rata			0.0990
Varian			0.0007
Standar Deviasi			0.0273

Tabel A. 29 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan 4 Dimensi Hasil Reduksi

Percobaan	Mediods	Learning Rate	Error
1	12	0.1	0.1509
2	12	0.1	0.1321
3	12	0.1	0.1509
4	12	0.1	0.1321
5	12	0.1	0.1321
6	12	0.1	0.2264
7	12	0.1	0.1698
8	12	0.1	0.1509
9	12	0.1	0.1698
10	12	0.1	0.2075
11	12	0.1	0.2075
12	12	0.1	0.2075
13	12	0.1	0.1509
14	12	0.1	0.1698
15	12	0.1	0.1698
16	12	0.1	0.1698
17	12	0.1	0.0943
18	12	0.1	0.1321
19	12	0.1	0.1698
20	12	0.1	0.1321
Minimum			0.0943
Maksimum			0.2264
Rata-rata			0.1613
Varian			0.0011
Standar Deviasi			0.0326

Tabel A. 30 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan 5 Dimensi Hasil Reduksi

Percobaan	Mediolds	Learning Rate	Error
1	12	0.1	0.1509
2	12	0.1	0.1509
3	12	0.1	0.1321
4	12	0.1	0.1509
5	12	0.1	0.1132
6	12	0.1	0.1321
7	12	0.1	0.1321
8	12	0.1	0.1132
9	12	0.1	0.1321
10	12	0.1	0.1321
11	12	0.1	0.1509
12	12	0.1	0.0943
13	12	0.1	0.0943
14	12	0.1	0.1132
15	12	0.1	0.1321
16	12	0.1	0.1509
17	12	0.1	0.0943
18	12	0.1	0.1132
19	12	0.1	0.0755
20	12	0.1	0.0943
Minimum			0.0755
Maksimum			0.1509
Rata-rata			0.1226
Varian			0.0005
Standar Deviasi			0.0233

Tabel A. 31 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan 6 Dimensi Hasil Reduksi

Percobaan	Mediolds	Learning Rate	Error
1	12	0.1	0.1132
2	12	0.1	0.1321
3	12	0.1	0.0755
4	12	0.1	0.1132
5	12	0.1	0.0755
6	12	0.1	0.0943
7	12	0.1	0.1509
8	12	0.1	0.0755
9	12	0.1	0.1509
10	12	0.1	0.1132
11	12	0.1	0.1321
12	12	0.1	0.0943
13	12	0.1	0.0943
14	12	0.1	0.1132
15	12	0.1	0.0943
16	12	0.1	0.0755
17	12	0.1	0.0943
18	12	0.1	0.0755
19	12	0.1	0.0943
20	12	0.1	0.1132
Minimum			0.0755
Maksimum			0.1509
Rata-rata			0.1038
Varian			0.0006
Standar Deviasi			0.0241

Tabel A. 32 Hasil Uji Coba dari 24 Data *Training* Optimal dengan 3 Dimensi Hasil Reduksi dan 10 Buah *Mediods*

Percobaan	<i>Mediods</i>	<i>Learning Rate</i>	<i>Error</i>
1	10	0.1	0.0377
2	10	0.1	0.0377
3	10	0.1	0.0566
4	10	0.1	0.0755
5	10	0.1	0.0566
6	10	0.1	0.0755
7	10	0.1	0.1132
8	10	0.1	0.0377
9	10	0.1	0.0566
10	10	0.1	0.0755
11	10	0.1	0.1321
12	10	0.1	0.0566
13	10	0.1	0.0566
14	10	0.1	0.1132
15	10	0.1	0.0943
16	10	0.1	0.0377
17	10	0.1	0.0566
18	10	0.1	0.0755
19	10	0.1	0.0755
20	10	0.1	0.1132
Minimum			0.0377
Maksimum			0.1321
Rata-rata			0.0717
Varian			0.0008
Standar Deviasi			0.0285

Tabel A. 33 Hasil Uji Coba dari 24 Data *Training* Optimal dengan 3 Dimensi Hasil Reduksi dan 8 Buah *Mediods*

Percobaan	<i>Mediods</i>	<i>Learning Rate</i>	<i>Error</i>
1	8	0.1	0.0377
2	8	0.1	0.0755
3	8	0.1	0.0566
4	8	0.1	0.0377
5	8	0.1	0.1321
6	8	0.1	0.0943
7	8	0.1	0.0755
8	8	0.1	0.0755
9	8	0.1	0.0566
10	8	0.1	0.0566
11	8	0.1	0.0377
12	8	0.1	0.0566
13	8	0.1	0.0566
14	8	0.1	0.1321
15	8	0.1	0.1132
16	8	0.1	0.1132
17	8	0.1	0.0943
18	8	0.1	0.0566
19	8	0.1	0.0566
20	8	0.1	0.0377
Minimum			0.0377
Maksimum			0.1321
Rata-rata			0.0726
Varian			0.0009
Standar Deviasi			0.0308

Tabel A. 34 Hasil Uji Coba dari 24 Data *Training* Optimal dengan 3 Dimensi Hasil Reduksi dan 6 Buah *Mediods*

Percobaan	<i>Mediods</i>	<i>Learning Rate</i>	<i>Error</i>
1	6	0.1	0.1321
2	6	0.1	0.0377
3	6	0.1	0.0755
4	6	0.1	0.0377
5	6	0.1	0.0755
6	6	0.1	0.0566
7	6	0.1	0.0755
8	6	0.1	0.1321
9	6	0.1	0.0943
10	6	0.1	0.0377
11	6	0.1	0.0377
12	6	0.1	0.0377
13	6	0.1	0.0377
14	6	0.1	0.0566
15	6	0.1	0.0377
16	6	0.1	0.0566
17	6	0.1	0.0377
18	6	0.1	0.1132
19	6	0.1	0.0566
20	6	0.1	0.1132
Minimum			0.0377
Maksimum			0.1321
Rata-rata			0.0669
Varian			0.0011
Standar Deviasi			0.0332

Tabel A. 35 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan 3 Dimensi Hasil Reduksi dan 10 Buah *Mediods*

Percobaan	<i>Mediods</i>	<i>Learning Rate</i>	<i>Error</i>
1	10	0.1	0.1321
2	10	0.1	0.1698
3	10	0.1	0.1887
4	10	0.1	0.1321
5	10	0.1	0.1698
6	10	0.1	0.1698
7	10	0.1	0.1321
8	10	0.1	0.1509
9	10	0.1	0.1321
10	10	0.1	0.1509
11	10	0.1	0.1321
12	10	0.1	0.1321
13	10	0.1	0.1321
14	10	0.1	0.2075
15	10	0.1	0.2075
16	10	0.1	0.1509
17	10	0.1	0.1887
18	10	0.1	0.2075
19	10	0.1	0.1509
20	10	0.1	0.1321
Minimum			0.1321
Maksimum			0.2075
Rata-rata			0.1585
Varian			0.0008
Standar Deviasi			0.0283

Tabel A. 36 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan 3 Dimensi Hasil Reduksi dan 8 Buah *Mediolds*

Percobaan	<i>Mediolds</i>	<i>Learning Rate</i>	<i>Error</i>
1	8	0.1	0.1509
2	8	0.1	0.1509
3	8	0.1	0.1887
4	8	0.1	0.1887
5	8	0.1	0.1321
6	8	0.1	0.1887
7	8	0.1	0.1509
8	8	0.1	0.1509
9	8	0.1	0.1698
10	8	0.1	0.1698
11	8	0.1	0.1509
12	8	0.1	0.1698
13	8	0.1	0.2075
14	8	0.1	0.1509
15	8	0.1	0.1698
16	8	0.1	0.1509
17	8	0.1	0.1887
18	8	0.1	0.1509
19	8	0.1	0.1887
20	8	0.1	0.1887
Minimum			0.1321
Maksimum			0.2075
Rata-rata			0.1679
Varian			0.0004
Standar Deviasi			0.0202

Tabel A. 37 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan 3 Dimensi Hasil Reduksi dan 6 Buah *Mediods*

Percobaan	<i>Mediods</i>	<i>Learning Rate</i>	<i>Error</i>
1	6	0.1	0.1321
2	6	0.1	0.1509
3	6	0.1	0.1509
4	6	0.1	0.1698
5	6	0.1	0.1509
6	6	0.1	0.1321
7	6	0.1	0.1698
8	6	0.1	0.1509
9	6	0.1	0.1321
10	6	0.1	0.1321
11	6	0.1	0.1509
12	6	0.1	0.1887
13	6	0.1	0.1509
14	6	0.1	0.1698
15	6	0.1	0.1509
16	6	0.1	0.1509
17	6	0.1	0.1509
18	6	0.1	0.1509
19	6	0.1	0.1698
20	6	0.1	0.1509
Minimum			0.1321
Maksimum			0.1887
Rata-rata			0.1528
Varian			0.0002
Standar Deviasi			0.0149

Tabel A. 38 Hasil Uji Coba dari 24 Data *Training* Optimal dengan berbagai *Learning Rate* (Bagian Pertama)

Percobaan	Mediods	Learning Rate	Error
1	12	0.01	0.0566
2	12	0.01	0.0755
3	12	0.01	0.0755
4	12	0.01	0.0755
5	12	0.01	0.0755
6	12	0.01	0.0755
7	12	0.01	0.0755
8	12	0.01	0.1132
9	12	0.01	0.0755
10	12	0.01	0.0755
Minimum Error			0.0566

Tabel A. 39 Hasil Uji Coba dari 24 Data *Training* Optimal dengan berbagai *Learning Rate* (Bagian Kedua)

Percobaan	Mediods	Learning Rate	Error
1	12	0.02	0.0755
2	12	0.02	0.0566
3	12	0.02	0.0755
4	12	0.02	0.0755
5	12	0.02	0.0755
6	12	0.02	0.0755
7	12	0.02	0.1321
8	12	0.02	0.0755
9	12	0.02	0.0755
10	12	0.02	0.0755
Minimum Error			0.0566

Tabel A. 40 Hasil Uji Coba dari 24 Data *Training* Optimal dengan berbagai *Learning Rate* (Bagian Ketiga)

Percobaan	Medioids	Learning Rate	Error
1	12	0.03	0.0566
2	12	0.03	0.0566
3	12	0.03	0.0566
4	12	0.03	0.0755
5	12	0.03	0.0755
6	12	0.03	0.1321
7	12	0.03	0.0556
8	12	0.03	0.0943
9	12	0.03	0.0556
10	12	0.03	0.0556
Minimum Error			0.0566

Tabel A. 41 Hasil Uji Coba dari 24 Data *Training* Optimal dengan berbagai *Learning Rate* (Bagian Keempat)

Percobaan	Medioids	Learning Rate	Error
1	12	0.04	0.0755
2	12	0.04	0.0566
3	12	0.04	0.0566
4	12	0.04	0.0566
5	12	0.04	0.0943
6	12	0.04	0.0566
7	12	0.04	0.1132
8	12	0.04	0.0566
9	12	0.04	0.0943
10	12	0.04	0.0377
Minimum Error			0.0377

Tabel A. 42 Hasil Uji Coba dari 24 Data *Training* Optimal dengan berbagai *Learning Rate* (Bagian Kelima)

Percobaan	Mediods	Learning Rate	Error
1	12	0.05	0.0566
2	12	0.05	0.0755
3	12	0.05	0.1132
4	12	0.05	0.0566
5	12	0.05	0.0755
6	12	0.05	0.0566
7	12	0.05	0.0377
8	12	0.05	0.0755
9	12	0.05	0.1132
10	12	0.05	0.0377
Minimum Error			0.0377

Tabel A. 43 Hasil Uji Coba dari 24 Data *Training* Optimal dengan berbagai *Learning Rate* (Bagian Keenam)

Percobaan	Mediods	Learning Rate	Error
1	12	0.06	0.0377
2	12	0.06	0.0566
3	12	0.06	0.0377
4	12	0.06	0.0377
5	12	0.06	0.0377
6	12	0.06	0.0377
7	12	0.06	0.0943
8	12	0.06	0.1321
9	12	0.06	0.0755
10	12	0.06	0.1132
Minimum Error			0.0377

Tabel A. 44 Hasil Uji Coba dari 24 Data *Training* Optimal dengan berbagai *Learning Rate* (Bagian Ketujuh)

Percobaan	Mediolds	Learning Rate	Error
1	12	0.07	0.0755
2	12	0.07	0.0377
3	12	0.07	0.1132
4	12	0.07	0.0755
5	12	0.07	0.0943
6	12	0.07	0.1132
7	12	0.07	0.0943
8	12	0.07	0.0755
9	12	0.07	0.1321
10	12	0.07	0.0566
Minimum Error			0.0377

Tabel A. 45 Hasil Uji Coba dari 24 Data *Training* Optimal dengan berbagai *Learning Rate* (Bagian Kedelapan)

Percobaan	Mediolds	Learning Rate	Error
1	12	0.08	0.0566
2	12	0.08	0.1132
3	12	0.08	0.0566
4	12	0.08	0.1132
5	12	0.08	0.0943
6	12	0.08	0.0943
7	12	0.08	0.0377
8	12	0.08	0.0755
9	12	0.08	0.0377
10	12	0.08	0.0943
Minimum Error			0.0377

Tabel A. 46 Hasil Uji Coba dari 24 Data *Training* Optimal dengan berbagai *Learning Rate* (Bagian Kesembilan)

Percobaan	Mediods	Learning Rate	Error
1	12	0.09	0.0377
2	12	0.09	0.0943
3	12	0.09	0.0566
4	12	0.09	0.0566
5	12	0.09	0.1321
6	12	0.09	0.0943
7	12	0.09	0.0755
8	12	0.09	0.0566
9	12	0.09	0.0566
10	12	0.09	0.0377
Minimum Error			0.0377

Tabel A. 47 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Pertama)

Percobaan	Mediods	Learning Rate	Error
1	12	0.01	0.1509
2	12	0.01	0.1698
3	12	0.01	0.1509
4	12	0.01	0.2075
5	12	0.01	0.1321
6	12	0.01	0.2075
7	12	0.01	0.1321
8	12	0.01	0.1321
9	12	0.01	0.1698
10	12	0.01	0.1698
Minimum Error			0.1321

Tabel A. 48 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Kedua)

Percobaan	Mediods	Learning Rate	Error
1	12	0.02	0.1698
2	12	0.02	0.1509
3	12	0.02	0.1509
4	12	0.02	0.1698
5	12	0.02	0.1321
6	12	0.02	0.1509
7	12	0.02	0.1698
8	12	0.02	0.1321
9	12	0.02	0.1509
10	12	0.02	0.1321
Minimum Error			0.1321

Tabel A. 49 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Ketiga)

Percobaan	Mediods	Learning Rate	Error
1	12	0.03	0.1698
2	12	0.03	0.1132
3	12	0.03	0.1321
4	12	0.03	0.1321
5	12	0.03	0.1321
6	12	0.03	0.1509
7	12	0.03	0.1509
8	12	0.03	0.1698
9	12	0.03	0.1887
10	12	0.03	0.1509
Minimum Error			0.1132

Tabel A. 50 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Keempat)

Percobaan	Mediods	Learning Rate	Error
1	12	0.04	0.1509
2	12	0.04	0.1698
3	12	0.04	0.1698
4	12	0.04	0.1509
5	12	0.04	0.1509
6	12	0.04	0.1509
7	12	0.04	0.1321
8	12	0.04	0.1698
9	12	0.04	0.1321
10	12	0.04	0.1132
Minimum Error			0.1132

Tabel A. 51 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Kelima)

Percobaan	Mediods	Learning Rate	Error
1	12	0.05	0.2075
2	12	0.05	0.1698
3	12	0.05	0.1509
4	12	0.05	0.1321
5	12	0.05	0.1321
6	12	0.05	0.1509
7	12	0.05	0.1698
8	12	0.05	0.1132
9	12	0.05	0.1321
10	12	0.05	0.1509
Minimum Error			0.1132

Tabel A. 52 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Keenam)

Percobaan	Mediolds	Learning Rate	Error
1	12	0.06	0.1509
2	12	0.06	0.1509
3	12	0.06	0.1698
4	12	0.06	0.1321
5	12	0.06	0.1509
6	12	0.06	0.1132
7	12	0.06	0.1321
8	12	0.06	0.1698
9	12	0.06	0.1509
10	12	0.06	0.1698
Minimum Error			0.1132

Tabel A. 53 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Ketujuh)

Percobaan	Mediolds	Learning Rate	Error
1	12	0.07	0.1887
2	12	0.07	0.1698
3	12	0.07	0.1698
4	12	0.07	0.1698
5	12	0.07	0.1509
6	12	0.07	0.1698
7	12	0.07	0.1132
8	12	0.07	0.1509
9	12	0.07	0.1321
10	12	0.07	0.1509
Minimum Error			0.1132

Tabel A. 54 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Kedelapan)

Percobaan	Mediods	Learning Rate	Error
1	12	0.08	0.1509
2	12	0.08	0.1887
3	12	0.08	0.1321
4	12	0.08	0.1321
5	12	0.08	0.1698
6	12	0.08	0.1509
7	12	0.08	0.1132
8	12	0.08	0.1509
9	12	0.08	0.1509
10	12	0.08	0.1887
Minimum Error			0.1132

Tabel A. 55 Hasil Uji Coba dari 24 Data *Training* Tidak Optimal dengan berbagai *Learning Rate* (Bagian Kesembilan)

Percobaan	Mediods	Learning Rate	Error
1	12	0.09	0.1698
2	12	0.09	0.1509
3	12	0.09	0.1321
4	12	0.09	0.1321
5	12	0.09	0.1321
6	12	0.09	0.1132
7	12	0.09	0.1132
8	12	0.09	0.1509
9	12	0.09	0.1509
10	12	0.09	0.1321
Minimum Error			0.1132