

USULAN TUGAS AKHIR

**KLASIFIKASI CITRA GLAUKOMA DENGAN ANN
BERDASARKAN PEMBULUH DARAH DALAM CITRA
FUNDUS RETINA MENGGUNAKAN PERBANDINGAN
METODE SEGMENTASI OTSU-THRESHOLDING DAN
DETEKSI TEPI CANNY**



Oleh:

**Teguh Ardian Samudra
F1D016084**

PROGRAM STUDI TEKNIK INFORMATIKA

**FAKULTAS TEKNIK
UNIVERSITAS MATARAM
2021**

USULAN TUGAS AKHIR

KLASIFIKASI CITRA GLAUKOMA DENGAN ANN BERDASARKAN PEMBULUH DARAH DALAM CITRA FUNDUS RETINA MENGGUNAKAN PERBANDINGAN METODE SEGMENTASI OTSU-THRESOLDING DAN DETEKSI TEPI CANNY

Telah diperiksa dan disetujui oleh Tim Pembimbing :

1. Pembimbing Utama



Gibran Satya Nugraha, S.Kom., M.Eng.

Tanggal: 28 Juni 2021

NIP : 19920323 201903 1 012

2. Pembimbing Pendamping



Fitri Bimantoro, ST., M.Kom.

Tanggal: 28 Juni 2021

NIP : 19860622 201504 1 002

Mengetahui,
Sekretaris Program Studi Teknik Informatika
Fakultas Teknik
Universitas Mataram

Andy Hidayat Jatmika, S.T., M.Kom.

NIP : 198312092012121001

ABSTRAK

Glaukoma merupakan salah satu penyakit mata yang dapat mengakibatkan kebutaan permanen yang disebabkan oleh meningkat-nya *Intraocular Pressure* (IOP) pada mata. Beberapa metode yang digunakan oleh dokter untuk mendeteksi glaukoma antara lain adalah *Optical Nerve Hypoplasia Stereo Photographs* (ONHSPs), teknik pencitraan canggih seperti *Optical Coherence Tomography* (OCT), *Scanning Laser Polarimetry* (SLP), dan *Confocal Scanning Laser Ophthalmoscopy* (CSLO) untuk menghasilkan gambar referensi mata dan struktur internal-nya. Namun, metode-metode tersebut membutuhkan banyak biaya dan pengawasan dari para ahli. Untuk mengatasi masalah tersebut, telah banyak penelitian di bidang *machine learning* yang dilakukan, salah satunya yaitu segmentasi pembuluh darah. Pembuluh darah memiliki peran penting untuk mendeteksi berbagai macam penyakit, dikarenakan memiliki banyak ciri-ciri alamiah seperti ketebalan, panjang dan warna. Melihat pentingnya peran segmentasi pembuluh darah, dalam penelitian ini dilakukan perbandingan metode segmentasi *otsu-thresholding* dan deteksi tepi *canny* yang bertujuan untuk mengetahui metode segmentasi pembuluh darah yang memberikan hasil lebih baik berdasarkan parameter akurasi, sensitivitas, dan spesifisitas dalam mendiagnosis citra glaukoma dan normal. *Dataset* yang digunakan adalah RIM-ONE r2 dan RIM-ONE r3. Kemudian akan dilakukan ekstraksi fitur dari *dataset* tersebut dengan menggunakan 5 fitur *Gray Level Co-occurrence Matrices* yaitu *angular second moment*, *contrast*, *inverse different moment*, *entropy*, dan *correlation*, serta 6 fitur statistik yaitu rerata intensitas, deviasi standar, *skewness*, energi, *entropy*, dan kehalusan. Setelah itu, fitur-fitur yang didapatkan akan digunakan sebagai masukan pada proses klasifikasi menggunakan metode *Artificial Neural Network backpropagation*. Untuk menguji performa klasifikasi, digunakan metode *Shuffle Sampling* dan *K-fold Cross Validation*.

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
USULAN TUGAS AKHIR	ii
ABSTRAK	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR	v
DAFTAR TABEL.....	vi
BAB I PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori	17
2.2.1 Pengolahan Citra Digital	17
2.2.2 Segmentasi	17
2.2.3 Ekstraksi Fitur	22
2.2.4 Klasifikasi	26
2.2.5 Pengujian Performa Klasifikasi.....	30
2.2.6 Evaluasi Performa Klasifikasi.....	31
BAB III METODOLOGI PENELITIAN	32
3.1 Alat dan Bahan Penelitian	32
3.2 Proses Penelitian.....	33
3.3 Perancangan Sistem.....	34
3.3.1 Tahap <i>Preprocessing</i>	36
3.3.2 Segmentasi	37
3.3.3 Ekstraksi Fitur	47
3.3.4 Klasifikasi	53
3.3.5 Performa Pengujian Klasifikasi.....	59
3.3.6 Evaluasi Performa Klasifikasi.....	60
3.4 Jadwal Kegiatan	63
DAFTAR PUSTAKA	65

DAFTAR GAMBAR

Gambar 2.1	Contoh Matriks Piksel Berukuran 5x5	18
Gambar 2.2	Area untuk Mengonversi Arah Tepi ke Dalam	19
Gambar 2.3	Penentuan Nilai Ambang.....	20
Gambar 2.4	Contoh Histogram pada Citra dengan Tekstur yang Berbeda	24
Gambar 2.5	Arsitektur Lapisan Tunggal.....	27
Gambar 2.6	Arsitektur Jaringan <i>Multilayer</i>	27
Gambar 2.7	Fungsi Aktivasi <i>Sigmoid Biner</i>	28
Gambar 2.8	Fungsi Aktivasi <i>Sigmoid Bipolar</i>	28
Gambar 2.9	<i>K-fold Cross Validation</i> dengan $k = 5$	30
Gambar 3.1	Citra Fundus Retina RIM-ONE r2	32
Gambar 3.2	Citra Fundus Retina RIM-ONE r3	33
Gambar 3.3	Diagram Alir Proses Penelitian	33
Gambar 3.4	Diagram Perancangan Sistem.....	35
Gambar 3.5	Citra Fundus Asli 2144x1424 Piksel.....	36
Gambar 3.6	Citra Fundus Hasil <i>Cropping</i> 609x717 Piksel.....	36
Gambar 3.7	Proses <i>Resize</i> Menjadi 300x300 Piksel	36
Gambar 3.8	Hasil Konversi Ruang Warna dari RGB ke <i>Grayscale</i>	37
Gambar 3.9	Contoh Matriks 4x4.....	37
Gambar 3.10	Grafik Histogram Matriks A	38
Gambar 3.11	Matriks Hasil <i>Otsu</i>	43
Gambar 3.12	Contoh Penerapan Filter <i>Gaussian</i>	44
Gambar 3.13	Contoh Matriks dengan <i>Kernel gx</i>	44
Gambar 3.14	Hasil Konvolusi <i>Kernel gx</i>	44
Gambar 3.15	Contoh Matriks dengan <i>Kernel gy</i>	44
Gambar 3.16	Hasil Konvolusi <i>Kernel gy</i>	45
Gambar 3.17	Hasil Perhitungan Matriks Arah Tepi	45
Gambar 3.18	Hasil Matriks Gradien	45
Gambar 3.19	Hasil Matriks <i>Non Maximus Suppression</i>	46
Gambar 3.20	Hasil Matriks <i>Hysteresis Thresholding</i>	46
Gambar 3.21	Contoh Perkalian Matriks.....	46
Gambar 3.22	Hasil Kali Matriks	47
Gambar 3.23	Contoh Matriks GLCM	47
Gambar 3.24	Pembentukan Matriks Ordo $n \times n$	47
Gambar 3.25	Komposisi Piksel.....	48
Gambar 3.26	Jumlah Pasangan Piksel.....	48
Gambar 3.27	Normalisasi Matriks	48
Gambar 3.28	Hasil Normalisasi Matriks.....	48
Gambar 3.29	Arah Sudut Matriks GLCM (a) 45° , (b) 90° , (c) 135°	49
Gambar 3.30	Contoh Matriks 4x4.....	51
Gambar 3.31	Grafik Histogram Matriks A	51
Gambar 3.32	Arsitektur ANN	54

DAFTAR TABEL

Tabel 2.1	<i>State of the Art</i> Penelitian.....	12
Tabel 2.2	Tabel Konfusi.....	31
Tabel 3.1	Data <i>Dummy</i>	53
Tabel 3.2	Bias dan Bobot Awal dari <i>Input Layer</i> ke <i>Hidden Layer</i>	55
Tabel 3.3	Bias dan Bobot Awal dari <i>Hidden Layer</i> ke <i>Output Layer</i>	55
Tabel 3.4	Bias dan Bobot Akhir dari <i>Input Layer</i> ke <i>Hidden Layer</i>	58
Tabel 3.5	Bias dan Bobot Akhir dari <i>Hidden Layer</i> ke <i>Output Layer</i>	59
Tabel 3.6	<i>Output</i> Data Hasil.....	59
Tabel 3.7	<i>Dataset</i> pada fold-1	59
Tabel 3.8	<i>Dataset</i> pada fold-2	60
Tabel 3.9	<i>Dataset</i> pada fold-3	60
Tabel 3.10	<i>Dataset</i> pada fold-4	60
Tabel 3.11	<i>Dataset</i> pada fold-5	60
Tabel 3.12	Pengujian dengan Metode <i>Shuffle Sampling</i>	60
Tabel 3.13	Contoh Pengujian <i>Cross Validation</i>	61
Tabel 3.14	Hasil Evaluasi Performa Klasifikasi	63
Tabel 3.15	Jadwal Kegiatan Penelitian	64

BAB I

PENDAHULUAN

1.1 Latar belakang

Glaukoma merupakan salah satu penyakit mata pada manusia yang menyebabkan kebutaan pada rentang usia 40 hingga 80 tahun [1]. Organisasi Kesehatan Dunia telah menyatakan Glaukoma sebagai penyebab kebutaan terbesar kedua di dunia dan mencakup 15% dari kasus kebutaan di dunia [2]. Glaukoma disebabkan karena adanya peningkatan tekanan cairan yang disebut *Intraocular Pressure* (IOP) pada saraf optik. Tekanan *intraocular* meningkat karena tersumbat-nya sistem drainase mata. Peningkatan tekanan *intraocular* di dalam mata dapat merusak saraf optik yang terletak di dalam retina. Jika kerusakan saraf optik akibat peningkatan tekanan *intraocular* terus berlanjut, glaukoma akan menyebabkan kehilangan penglihatan secara permanen [3].

Beberapa metode yang digunakan oleh dokter untuk mendeteksi glaukoma antara lain adalah *Optical Nerve Hypoplasia Stereo Photographs* (ONHSPs), teknik pencitraan canggih seperti *Optical Coherence Tomography* (OCT), *Scanning Laser Polarimetry* (SLP), dan *Confocal Scanning Laser Ophthalmoscopy* (CSLO) untuk menghasilkan gambar referensi mata dan struktur internal-nya. Namun, metode-metode tersebut membutuhkan banyak biaya dan pengawasan dari para ahli [1] [4].

Untuk mengatasi masalah tersebut, telah banyak penelitian yang dilakukan dengan memanfaatkan teknik-teknik dalam bidang pengolahan citra dan *machine learning*, salah satunya adalah segmentasi pembuluh darah [1]. Segmentasi pembuluh darah berperan penting dalam proses klasifikasi, seperti contoh pada penelitian dengan judul “*Earlier glaucoma detection using blood vessel segmentation and classification*”, yang menggunakan *Support Vector Machine* (SVM) dan *Adaptive Neuro Fuzzy Inference System* (ANFIS) sebagai *classifier* dan memperoleh akurasi, sensitivitas, dan spesifisitas secara berurutan sebesar 91,67%, 90%, dan 93.33% untuk metode SVM, dan untuk metode ANFIS mendapatkan akurasi, sensitivitas, dan spesifisitas secara berurutan sebesar

97.77%, 85.7%, dan 92% [3]. Nilai sensitivitas dan spesifisitas yang dihasilkan oleh dua *classifier* tersebut tidak sebaik akurasi-nya. Hal tersebut sangat dipengaruhi oleh kualitas hasil segmentasi citra, sehingga nilai fitur yang didapatkan tidak merepresentasikan kelas yang sebenarnya, dan berimbas pada prediksi yang salah [5]. Selain itu jumlah *dataset* yang digunakan juga cukup berpengaruh, karena semakin banyak *dataset* yang digunakan maka *classifier* akan lebih banyak melakukan pembelajaran sehingga memperkecil kemungkinan kesalahan dalam melakukan prediksi [6].

Pembuluh darah dapat digunakan untuk mendiagnosis berbagai macam penyakit seperti diabetes, glaukoma, degenerasi makula, hipertensi, dan lain-lain, karena pada pembuluh darah memiliki banyak ciri-ciri alamiah seperti ketebalan, panjang, dan warna [7][8][9]. Untuk itu, berbagai macam metode segmentasi telah dikembangkan, seperti metode deteksi tepi canny dan metode *otsu-thresholding*. Deteksi tepi canny memastikan objek yang disegmentasikan terbebas dari *noise*, kelebihan lainnya adalah metode ini dapat mendeteksi bagian tepi dengan probabilitas kesalahan yang kecil [10]. Ketika diimplementasikan pada objek yang memiliki bagian tepi yang tegas seperti pada pembuluh darah dalam citra fundus retina, metode ini sangat tepat untuk digunakan. Pembuluh darah memiliki warna yang lebih gelap dibandingkan dengan *optic disc* pada citra fundus retina yang sudah diubah menjadi citra *grayscale*. *Otsu-thresholding* adalah salah satu metode segmentasi yang memberikan nilai *error* yang kecil pada hasil segmentasi-nya. Selain itu, kelebihan metode ini adalah proses komputasi pada segmentasi-nya berlangsung cepat dan dapat dikombinasikan dengan metode yang lain seperti operasi morfologi [11]. Kelebihan lainnya dari metode ini yaitu kemampuannya dalam mensegmentasikan objek yang besar dan tersebar merata [10]. Di dalam citra fundus retina terdapat beberapa objek seperti *optic disc* dan pembuluh darah yang tersebar pada bagian *optic disc* tersebut. Pembuluh darah memiliki ukuran yang cukup besar, sehingga metode *otsu-thresholding* cocok digunakan dalam mensegmentasikan pembuluh darah dalam citra fundus retina. Pada beberapa penelitian, kedua metode ini mendapatkan akurasi keberhasilan yang cukup tinggi dengan rata-rata akurasi di atas 90%. Penelitian yang membandingkan kedua

metode ini sudah cukup banyak dilakukan, namun belum ada yang melakukan perbandingan pada deteksi glaukoma berdasarkan pembuluh darah.

Untuk mengetahui metode segmentasi yang paling cocok digunakan dalam mendiagnosis glaukoma berdasarkan pembuluh darah, maka dalam penelitian ini akan dilakukan ekstraksi fitur berbasis tekstur yang dimana fitur tersebut akan dijadikan masukan dalam proses klasifikasi citra fundus normal dan glaukoma. Beberapa ekstraksi fitur berbasis tekstur seperti *Gray Level Co-occurrence Matrices* (GLCM) dan statistik memberikan nilai akurasi diatas 95%, seperti pada penelitian tentang ekstraksi pembuluh darah untuk mendeteksi penyakit *diabetic retinopathy*, yang menggunakan GLCM sebagai ekstraksi fitur dan memperoleh akurasi dan sensitivitas cukup baik sebesar 97.76% dan 96.77% [12]. Dan penelitian lainnya yang menggunakan ekstraksi fitur statistik dan fitur *wavelet* tekstur untuk mendeteksi glaukoma dan mendapatkan hasil akurasi yang cukup tinggi sebesar 96.70% [13]. Kemudian salah satu metode klasifikasi yang dapat digunakan adalah *Artificial Neural Network* (ANN) dengan algoritma *Multilayer Perceptron-Back Propagation* (MLP-BP), seperti pada penelitian dengan judul “*Automated Glaucoma Detection System based on Wavelet Energy features and ANN*” dengan tingkat akurasi yang didapatkan cukup tinggi yaitu sebesar 97.60% [14].

Berdasarkan uraian di atas dan pentingnya kualitas hasil segmentasi dalam proses klasifikasi, maka pada penelitian ini akan dilakukan perbandingan antara dua metode segmentasi citra yaitu metode segmentasi deteksi tepi canny dan metode *otsu-thresholding* untuk mengetahui segmentasi yang paling tepat digunakan untuk mendiagnosis glaukoma berdasarkan pembuluh darah dengan ekstraksi fitur tekstur menggunakan fitur gabungan statistik dan *Gray Level Co-occurrence Matrix* (GLCM) serta klasifikasi menggunakan ANN algoritma MLP-BP .

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah bagaimana hasil perbandingan metode segmentasi antara deteksi tepi canny dan *otsu-thresholding* dalam mendiagnosis glaukoma berdasarkan pembuluh darah pada citra fundus retina menggunakan klasifikasi ANN.

1.3 Batasan Masalah

Berdasarkan latar belakang yang ada, diberikan batasan-batasan masalah sebagai berikut:

1. Digunakan dua buah *dataset* untuk citra fundus normal dan glaukoma yaitu RIM-ONE r2, dan RIM-ONE r3 yang didapatkan dari <http://medimrg.webs.ull.es/research/downloads>.
2. Metode segmentasi yang dibandingkan adalah deteksi tepi canny dan *otsu-thresholding*.
3. Citra fundus yang digunakan adalah citra fundus yang telah di-*crop* secara manual.
4. Ekstraksi fitur menggunakan fitur GLCM (*angular second moment, contrast, inverse different moment, entropy, dan correlation*) dan statistik (rerata intensitas, deviasi standar, *skewness*, energi, entropi, dan kehalusan).
5. Klasifikasi menggunakan metode ANN *backpropagation*.
6. Performa *classifier* diuji berdasarkan parameter akurasi, sensitivitas, dan spesifisitas.

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah untuk mengetahui metode mana yang memberikan hasil lebih baik berdasarkan parameter akurasi, sensitivitas, dan spesifisitas dari kedua metode tersebut.

1.5 Manfaat

Manfaat yang dapat diperoleh dari penelitian ini adalah dapat mengetahui metode segmentasi pembuluh darah yang memberikan hasil lebih baik antara deteksi tepi canny dan *otsu-thresholding* dalam mendiagnosis glaukoma

berdasarkan pembuluh darah pada citra fundus retina menggunakan klasifikasi ANN, serta diharapkan dapat menjadikan penelitian ini sebagai acuan untuk penelitian-penelitian yang akan datang.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penyusunan penelitian ini adalah sebagai berikut:

1. Bab I Pendahuluan

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan laporan tugas akhir.

2. Bab II Tinjauan Pustaka dan Dasar Teori

Bab ini membahas tentang penelitian-penelitian terdahulu dan teori-teori yang berkaitan dengan topik penelitian seperti metode segmentasi pembuluh darah deteksi tepi *canny* dan *otsu-thresholding*, ekstraksi fitur statistik dan *Gray Level Co-occurrence Matrix* (GLCM), dan *Artificial Neural Network* (ANN).

3. Bab III Metodologi Penelitian

Bab ini menjelaskan tentang metodologi yang akan digunakan untuk penelitian meliputi *preprocessing*, segmentasi, ekstraksi fitur dan klasifikasi.

4. Bab IV Hasil dan Pembahasan

Bab ini menyajikan hasil dari penelitian yang telah dilakukan dan pembahasan dari hasil penelitian. Meliputi hasil dari tiap segmentasi citra fundus, hasil dari klasifikasi fitur, nilai akurasi, sensitivitas dan spesifisitas.

5. Bab V Kesimpulan dan Saran

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan serta saran yang diberikan sebagai acuan untuk mengembangkan dan membuat penelitian yang lebih baik lagi.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini akan dilakukan perbandingan dari dua metode segmentasi yaitu deteksi tepi canny dan *otsu-thresholding*. Deteksi tepi canny dan *otsu-thresholding* merupakan metode segmentasi yang cukup populer dan sudah digunakan oleh beberapa peneliti dalam melakukan segmentasi pembuluh darah. Seperti pada penelitian tentang deteksi glaukoma dengan menggunakan analisis ISNT yang menerapkan deteksi tepi canny sebagai segmentasi pembuluh darah yang berada di dalam area *optic disc* dan membandingkan tepinya dengan area *optic cup* apakah bersimpangan atau tidak, kemudian dijadikan sebagai acuan pada proses klasifikasi. Namun pada penelitian tersebut hanya menjelaskan proses segmentasi tanpa mencantumkan ekstraksi fitur yang digunakan [1].

Penelitian oleh Chin-Chen Chang dkk yang dalam penelitiannya digunakan metode segmentasi berbasis *line operator* dan deteksi tepi menggunakan metode canny untuk mendeteksi pembuluh darah pada retina. Pada penelitian ini, dilakukan peningkatan kontras saluran hijau yang selanjutnya dilakukan *inverts* untuk membuat pembuluh darah lebih terang dari latar belakang, kemudian dibuat peta tepi yang dilambangkan dengan *seed map* pertumbuhan pembuluh darah dengan menggunakan detektor canny, dan terakhir dilakukan segmentasi piksel pembuluh darah oleh *line operator* dari *seed map* dan menghasilkan akurasi rata-rata sebesar 94%. Namun pada penelitian ini hanya membahas proses segmentasi yang dilakukan dan tidak membahas ekstraksi fitur dan klasifikasi [7].

J. Sivakamasundari dkk yang melakukan penelitian mengenai perbandingan segmentasi canny yang menggunakan *preprocessing* morfologi citra dengan metode canny tanpa *preprocessing* dan juga metode canny dengan *preprocessing* berbasis peregangan kontras. Perbandingan skenario metode segmentasi tersebut dilakukan pada citra fundus pembuluh darah normal dan abnormal. Hasil dari penelitian menunjukkan bahwa segmentasi citra fundus

pembuluh darah menggunakan metode canny dengan *preprocessing* morfologi citra mendapatkan rata-rata recall terbesar yaitu 82% pada citra fundus normal dan abnormal, disusul dengan segmentasi canny menggunakan peregangan kontras dengan rata-rata recall 80.5%, dan terakhir segmentasi canny tanpa *preprocessing* mendapat rata-rata recall 80.3%. Pada penelitian ini digunakan ekstraksi fitur berbasis statistik dengan fitur-fitur seperti *entropy*, *energy*, *contrast*, *homogeneity for matching*, dan *retrieval*. Hasil rata-rata recall yang rendah disebabkan karena *dataset* yang digunakan hanya sebanyak 20 untuk citra fundus mata normal dan 20 untuk citra fundus mata abnormal serta fitur yang digunakan juga sedikit [15].

Penelitian oleh Rajdeep Dhar dkk yang menganalisis metode canny dan *laplacian of gaussian* dalam deteksi tepi citra. Dalam makalah ini dilakukan perbandingan segmentasi canny dan *laplacian* dalam berbagai varietas gambar retina, seperti retina yang sehat, terkena penyakit, pembuluh darah, *optic disc* dan lain-lain, dengan kesimpulan bahwa hasil dari deteksi tepi canny memiliki kinerja lebih baik dibandingkan dengan *laplacian of gaussian*. Penelitian ini membahas langkah-langkah segmentasi secara lengkap namun hanya berfokus pada hasil perbandingan segmentasi citra fundus retina tanpa membahas ekstraksi fitur dan klasifikasi [16].

Kemudian penelitian yang dilakukan oleh Sunita Sarangi dkk yang pada penelitian ini dilakukan pendeteksian kelainan pembuluh darah pada retina dengan menggunakan detektor tepi *sobel* dan canny. Hasil dari kedua metode tersebut dibandingkan dalam parameter gambar yang berbeda dan kesalahan histogram. Pada penelitian ini didapatkan kesimpulan bahwa operator canny lebih akurat dalam mendeteksi pembuluh darah dibandingkan dengan operator *sobel* dilihat dari tingkat kesalahan histogram yang lebih kecil yaitu sebesar 0.0015 pada operator canny dan sebesar 0.0142 pada operator *sobel*. Kekurangan dari penelitian ini adalah data yang digunakan hanya berupa sepuluh sampel *image* mata normal dan diabetes dan tidak membahas ekstraksi fitur dan klasifikasi [17].

Penelitian oleh Shouvik Chakraborty dkk yang membandingkan beberapa metode untuk segmentasi pembuluh darah. Pada penelitian ini diusulkan metode

baru yang kemudian dibandingkan dengan metode-metode segmentasi tepi yang sudah ada sebelumnya, salah satu nya adalah deteksi tepi canny dan mendapatkan hasil akurasi sebesar 90%. Kekurangan dari penelitian ini hanya membahas metode segmentasi yang dimana hasil segmentasi-nya akan dibandingkan dengan versi *ground truth* serta data yang digunakan hanya berjumlah 40 buah citra fundus. Penelitian ini juga tidak membahas proses ekstraksi fitur dan klasifikasi [18].

Kemudian beberapa penelitian tentang *otsu-thresholding* seperti penelitian oleh Khan Bahadarkhan dkk yang menggunakan metode *otsu* untuk mendeteksi *diabetic retinopathy* (DR). Pada penelitian ini citra fundus retina diubah ke *green channel* kemudian digunakan *Contrast Limited Adaptive Histogram Equalization* (CLAHE) dan *morphological filters* untuk meningkatkan kualitas dan menghilangkan *low frequency noise*, kemudian dilakukan *otsu-thresholding* sehingga didapatkan akurasi rata-rata sebesar 95%. Pada penelitian ini hanya digunakan 40 *image* untuk DRIVE *dataset* dan 20 *image* untuk STARE *dataset* dan tidak dicantumkan ekstraksi fitur dan klasifikasi yang digunakan [8].

Penelitian yang dilakukan oleh Jyotiprava Dash dkk dengan menerapkan metode *otsu-thresholding* untuk mengekstraksi pembuluh darah yang kemudian digunakan *morphological cleaning* untuk menghilangkan *noise* yang ada dan didapatkan akurasi rata-rata sebesar 95%. Metode *otsu* dipilih karena memiliki kinerja segmentasi yang baik dan penerapannya yang mudah, akan tetapi pada penelitian ini didapatkan sensitivitas yang rendah sebesar 72% untuk DRIVE *database* dan 73% untuk STARE *database* [9].

Penelitian oleh Wiharto dkk yang melakukan segmentasi pembuluh darah pada citra fundus retina. Metode yang digunakan terbagi menjadi tiga tahap yaitu *preprocessing*, segmentasi, dan pengujian. Tahap pertama dilakukan *preprocessing* untuk memperbaiki kualitas citra fundus dengan metode CLAHE dan median filter pada citra saluran hijau. Kemudian tahap kedua berupa segmentasi menggunakan beberapa metode yaitu, *frangi filter*, *2D-convolution filtering*, *median filtering*, *otsu's thresholding*, operasi morfologi, dan pengurangan *background*. Terakhir adalah menguji sistem menggunakan *dataset*

DRIVE dan STARE dan didapatkan hasil akurasi rata-rata sebesar 89.04%. Akurasi yang rendah pada penelitian ini disebabkan karena sedikitnya *dataset* yang digunakan yaitu berupa 20 citra fundus retina yang tidak tersegmentasi dan 20 citra fundus retina yang tersegmentasi secara manual [19].

Penelitian oleh Owais Ali dkk yang melakukan perbandingan dari tiga algoritma segmentasi pembuluh darah yang baru-baru ini dikembangkan dalam hal akurasi, sensitivitas, dan spesifisitas pada dua *dataset* yaitu DRIVE dan STARE yang tersedia untuk umum. Algoritma yang dibandingkan dalam penelitian ini adalah metode segmentasi berbasis SUACE dan *tyler coye* terintegrasi, *frangi* filter dan metode segmentasi berbasis *otsu-thresholding*, dan pendekatan segmentasi pembuluh jaringan saraf konvolusional berbasis *UNET*. Dari penelitian tersebut didapatkan kesimpulan bahwa algoritma segmentasi pembuluh darah berbasis *frangi* dan *otsu thresholding* menghasilkan akurasi tertinggi dengan akurasi rata-rata sebesar 94.17%. Kekurangan dari penelitian ini adalah rendahnya nilai sensitivitas yang didapatkan yaitu sebesar 46% untuk DRIVE *database* dan sebesar 49% untuk STARE *database*. Nilai dari sensitivitas yang rendah disebabkan karena sedikitnya *dataset* yang digunakan yaitu 40 *image* untuk DRIVE *dataset* dan 20 *image* untuk STARE *dataset* yang mengakibatkan nilai *false negative* yang tinggi [20].

Penelitian oleh Nur Rafidah Sabri dkk yang melakukan perbandingan metode segmentasi berbasis *thresholding* untuk deteksi penyakit *diabetic retinopathy*. Pada penelitian ini, dilakukan perbandingan tiga metode berbasis *thresholding* untuk mengidentifikasi *mikroaneurisma* (MAs) dan pembuluh darah, salah satunya *otsu thresholding*. Penelitian ini menggunakan tiga *dataset* yaitu DRIVE, E-Optha dan Messidor sehingga didapatkan akurasi rata-rata sebesar 92.18% untuk *database* DRIVE. Namun pada penelitian ini didapatkan nilai spesifisitas yang rendah, dengan nilai rata-rata sebesar 63%. Ini dikarenakan sedikitnya *dataset* yang digunakan, yaitu 20 sampel *image*. Penelitian ini juga tidak menjelaskan ekstraksi fitur dan klasifikasi yang digunakan [21].

Selanjutnya, ekstraksi fitur yang diusulkan pada penelitian ini adalah ekstraksi fitur GLCM dan statistik. Ekstraksi fitur tersebut sudah pernah

dilakukan sebelumnya, seperti penelitian tentang segmentasi pembuluh darah pada retina untuk mendiagnosis penyakit *diabetic retinopathy* yang memadukan ekstraksi fitur GLCM dan *daubechies wavelet transform* sebagai ekstraksi fiturnya, pada penelitian ini citra fundus retina diklasifikasikan sebagai citra fundus normal atau abnormal menggunakan ekstraksi fitur GLCM yang diperoleh menggunakan *daubechies wavelet transform* dan menunjukkan hasil yang lebih baik dari tiga metode lainnya dengan tingkat akurasi dan sensitivitas berturut-turut sebesar 97.76% dan 96.77% [12].

Kemudian pada penelitian untuk mendiagnosis penyakit glaukoma dengan menggunakan kombinasi ekstraksi fitur statistik dan *wavelet-based textural* yang dihitung dari *aproksimasi subband* hijau dan biru dari daerah *optic disc* untuk dilakukan karakterisasi pada *optic disc* secara menyeluruh dan mempertimbangkan tujuh fitur statistik untuk klasifikasi glaukoma, dan didapatkan akurasi rata-rata sebesar 95% [13].

Untuk klasifikasi pada penelitian ini diusulkan dengan menggunakan metode ANN. Metode klasifikasi ini sudah sering digunakan, seperti pada penelitian dengan judul “*Automated Glaucoma Detection System based on Wavelet Energy features and ANN*” yang menggunakan ANN *backpropagation* sebagai metode klasifikasinya. Klasifikasi ANN pada penelitian tersebut menggunakan dua fase yaitu *forward phase* dan *backward phase* dan menghasilkan akurasi rata-rata sebesar 97.60% [14].

Kemudian penelitian oleh Endi Permata dkk yang melakukan penelitian tentang klasifikasi penyakit glaukoma menggunakan metode klasifikasi ANN. Dimana pada penelitian ini, citra fundus yang telah melalui tahap *preprocessing* akan dilakukan segmentasi *optic cup* dan *optic disk*, kemudian dilakukan ekstraksi ciri dari hasil segmentasi dan terakhir dilakukan klasifikasi menggunakan metode ANN dan dibagi menjadi tiga kategori yaitu glaukoma ringan, sedang dan parah dengan nilai rata-rata akurasi, sensitivitas, dan spesifisitas secara berurutan sebesar 94.2%, 99%, dan 98% [22].

Penelitian lainnya yaitu penelitian oleh Ahmad Mustofa dkk yang melakukan penelitian tentang deteksi glaukoma pada citra fundus retina dengan

menggunakan *adaptive thresholding* dan *support vector machine*. Pada penelitian ini pertama-tama dilakukan tahap *preprocessing* yang dibagi menjadi *preprocessing optic cup*, *preprocessing optic disk*, dan *preprocessing pembuluh darah*. Kemudian dilakukan proses segmentasi *optic cup*, *optic disk*, dan pembuluh darah dengan menggunakan metode *adaptive thresholding*. Setelah itu fitur *Cup to disk ratio* (CDR), ISNT *Neuro Retinal Rim* (NRR), dan ISNT pembuluh darah diekstraksi dari masing-masing citra yang telah tersegmentasi. Ketiga fitur tersebut kemudian dijadikan masukan pada tahap klasifikasi *support vector machine* dengan menggunakan metode pencarian *hyperplane sequential minimal optimization* dan fungsi kernel linear. Dengan menggunakan data yang diambil dari database RIM-ONE, dan didapatkan nilai akurasi rata-rata sebesar 80% [23].

Kemudian penelitian oleh Ayu Ambarwati dkk yang melakukan analisis kinerja metode deteksi tepi. Pada penelitian ini dilakukan analisis kinerja terhadap operator deteksi tepi sobel, prewitt, roberts dan canny menggunakan *otsu-thresholding* berdasarkan nilai *threshold*, waktu proses, dan pengamatan visual dan didapatkan kesimpulan bahwa operator canny menghasilkan tepian yang tipis dan halus dibandingkan dengan operator lain dan tidak menghilangkan informasi penting yang terkandung pada gambar [24].

Kemudian pada makalah yang melakukan *review* algoritma segmentasi pembuluh darah yang dimana pada makalah ini dilakukan analisa berbagai algoritma dan teknik segmentasi pembuluh darah pada citra fundus retina mata dalam proses diagnosis penyakit *diabetic retinopathy* dengan kesimpulan bahwa dari ulasan dari tiap-tiap metode yang ada dalam makalah ini memiliki kekurangan dan kelebihan masing-masing dan dapat dipilih dan diterapkan pada penelitian terkait [25].

Dari beberapa referensi tersebut maka dapat disusun suatu *state of the art* untuk mendukung pelaksanaan penelitian yang akan dilakukan seperti pada Tabel 2.1.

Tabel 2.1 *State of the Art* Penelitian

No.	Judul Penelitian	Area Penelitian					Ket.
		Canny	Otsu	GLCM	Statistik	ANN	
1	<i>Detection of Glaucoma from Retinal Fundus Images by analyzing ISNT Measurement and features of Optic Cup and Blood Vessels</i>	✓					Hasil akhir dari penelitian ini adalah algoritma yang diusulkan dapat mengelompokkan citra normal dan glaukoma dengan efektif.
2	<i>A Novel Retinal Blood Vessel Segmentation Method Based on Line Operator and Edge Detector</i>	✓					Dari penelitian yang dilakukan didapatkan akurasi rata-rata sebesar 94% dari 60 total dataset.
3	<i>Content Based Human Retinal Image Retrieval Using Vascular Feature Extraction</i>	✓					Dari penelitian ini didapatkan hasil bahwa segmentasi dengan <i>preprocessing</i> mendapatkan rata-rata <i>recall</i> yang lebih tinggi dari segmentasi tanpa <i>preprocessing</i> yaitu sebesar 82%
4	<i>An Analysis of Canny and Laplacian of Gaussian Image Filters in Regard to Evaluating Retinal Image</i>	✓					Kesimpulan dari penelitian ini adalah hasil dari deteksi tepi canny memiliki kinerja lebih baik dibandingkan dengan <i>laplacian of gaussian</i> dari berbagai varietas gambar.
5	<i>A comparative</i>	✓					Pada penelitian

	<i>evaluation of diabetic retinal vascular structures using edge detection techniques</i>						ini didapatkan kesimpulan bahwa operator canny lebih akurat dalam mendeteksi pembuluh darah dibandingkan dengan operator <i>sobel</i> dilihat dari tingkat kesalahan histogram yang lebih kecil yaitu sebesar 0.0015
6	<i>Gradient Approximation in Retinal Blood Vessel Segmentation</i>	✓					Hasil dari penelitian ini adalah deteksi tepi canny menempati posisi kedua setelah metode yang diusulkan dengan perbedaan akurasi sebesar 0.01%.
7	<i>A Morphological Hessian Based Approach for Retinal Blood Vessels Segmentation and Denoising Using Region Based Otsu Thresholding</i>		✓				Pada penelitian ini didapatkan akurasi sebesar 95% dengan total <i>dataset</i> sebanyak 60 citra fundus retina
8	<i>Retinal Blood Vessel Segmentation Using Otsu Thresholding With Principal Component Analysis</i>		✓				Pada penelitian ini didapatkan akurasi, sensitivitas, dan spesifisitas secara berurutan sebesar 95%, 73%, dan 98% dengan total <i>dataset</i> 60 citra fundus retina

9	<i>Blood Vessels Segmentation in Retinal Fundus Image using Hybrid Method of Frangi Filter, Otsu Thresholding and Morphology</i>		✓				Pada penelitian ini didapatkan akurasi sebesar 89.04% dengan total dataset sebanyak 40 citra fundus retina
10	<i>A Comparative Study of Automatic Vessel Segmentation Algorithms</i>		✓				Pada penelitian ini didapatkan kesimpulan bahwa algoritma segmentasi pembuluh darah berbasis <i>frangi</i> dan <i>otsu thresholding</i> menghasilkan akurasi tertinggi dengan akurasi rata-rata sebesar 94.17%.
11	<i>Performance Comparison Using Thresholding Based Method for Diabetic Retinopathy</i>		✓				Dari penelitian yang dilakukan didapatkan akurasi rata-rata sebesar 92%
12	<i>Retinal Blood Vessels Extraction and Detection of Exudates Using Wavelet Transform and PNN Approach for the Assessment of Diabetic Retinopathy</i>			✓			Pada penelitian ini citra yang diklasifikasikan menggunakan ekstraksi fitur GLCM yang diperoleh menggunakan <i>daubechies wavelet transform</i> menunjukkan hasil yang lebih baik dari tiga metode lainnya dengan tingkat akurasi dan sensitivitas

							berturut-turut sebesar 97.76% dan 96.77%
13	<i>Glaucoma Detection from Retinal Images Using Statistical and Textural Wavelet Features</i>				✓		Dari penelitian yang dilakukan didapatkan akurasi rata-rata sebesar 96.7%
14	<i>Automated Glaucoma Detection System based on Wavelet Energy features and ANN</i>					✓	Hasil akhir dari penelitian ini adalah dapat mengelompokkan citra menjadi tiga kategori yaitu glaukoma ringan, sedang dan parah dengan nilai rata-rata akurasi, sensitivitas, dan spesifisitas secara berurutan sebesar 94.2%, 99%, dan 98%
15	Klasifikasi Glaukoma Menggunakan <i>Neural Network Backpropagation</i> (2016)					✓	Hasil akhir dari penelitian ini berupa sistem yang dapat mengklasifikasi citra glaukoma dengan akurasi rata-rata sebesar 94.2%
16	Deteksi Penyakit Glaukoma pada Citra Fundus Retina Menggunakan <i>Adaptive Thresholding</i> dan <i>Support Vector Machine</i>		✓				Dari penelitian yang dilakukan didapatkan kesimpulan bahwa performa yang terbaik didapat ketika menggunakan <i>kernel linear</i> dengan akurasi sebesar 80%

17	Segmentasi Citra Digital Menggunakan <i>Thresholding Otsu</i> untuk Analisa Perbandingan Deteksi Tepi	✓	✓				Dari penelitian yang dilakukan didapatkan kesimpulan bahwa operator Canny memberikan hasil tepian yang lebih tipis dan halus dibandingkan dengan metode lain serta tidak menghilangkan informasi penting gambar namun membutuhkan banyak waktu dalam komputasi nya
18	<i>Review</i> Algoritma Segmentasi Pembuluh Darah Pada Citra Fundus Retina Mata Untuk Membantu Diagnosis <i>Diabetic Retinopathy</i>	✓	✓				Dari hasil <i>review</i> yang dilakukan didapatkan hasil bahwa metode yang tertera bukan membuktikan metode yang terbaik untuk digunakan dalam penelitian di bidang <i>diabetic retinopathy</i> dan setiap metode yang ada memiliki kelebihan dan kekurangan nya masing-masing
19	Klasifikasi Citra Glaukoma dengan ANN Berdasarkan Pembuluh Darah dalam Citra Fundus Retina	✓	✓	✓	✓	✓	Penelitian yang akan dilakukan

	Menggunakan Perbandingan Metode Segmentasi <i>Otsu-Thresholding</i> dan Deteksi Tepi Canny						
--	--	--	--	--	--	--	--

2.2 Dasar Teori

2.2.1 Pengolahan Citra Digital

Secara umum, istilah pengolahan citra digital dinyatakan sebagai “pemrosesan gambar berdimensi dua melalui komputer digital”. Menurut Efford, pengolahan citra digital merupakan istilah umum bagi berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Foto adalah contoh gambar berdimensi dua yang dapat diolah dengan mudah, di mana setiap foto dalam bentuk citra digital dapat diolah menggunakan perangkat lunak tertentu seperti mengatur gelap terang suatu citra atau memisahkan suatu objek dengan latar belakangnya [26].

2.2.2 Segmentasi

Segmentasi merupakan proses yang bertujuan untuk mendapatkan atau membagi suatu objek pada citra digital yang telah dikelompokkan berdasarkan kemiripan atribut yang ada. Proses memilah objek pada citra ini bergantung pada banyaknya suatu objek yang terkandung di dalamnya. Semakin banyak objek yang terkandung pada suatu citra, maka proses pemilahan objek pada citra tersebut akan semakin kompleks. Secara prinsip, segmentasi digunakan untuk mendapatkan objek yang diinginkan dan dapat dilakukan sebagai langkah awal dari proses klasifikasi objek [26]. Pada penelitian ini digunakan segmentasi deteksi tepi canny dan *otsu-thresholding*.

1. Deteksi Tepi Canny

Deteksi tepi pada citra digital berfungsi untuk memperoleh tepian objek yang memanfaatkan perubahan nilai intensitas yang drastis pada batas dua area. Deteksi tepi pada umumnya menggunakan dua macam detektor, yaitu detektor

baris (H_y) dan detektor kolom (H_x). Operator canny merupakan operator deteksi tepi dengan tingkat kesalahan rendah dan terkenal optimal yang dikemukakan oleh John Canny pada tahun 1986. Deteksi ini memiliki beberapa langkah dalam implementasinya, seperti yang telah dijabarkan sebagai berikut [26]:

a. Langkah 1 :

Pertama, dilakukan proses penghalusan citra (smoothing) menggunakan *Gaussian Filter*, dengan menggunakan Persamaan 2.1.

$$G(i, j) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(i-u)^2 + (j-v)^2}{2\sigma^2}} \quad (2-1)$$

dimana :

$e = 2.71$ (konstanta euler)

σ = standar deviasi (sigma)

$\pi = 3.14$ (pi)

b. Langkah 2 :

Kemudian dilakukan proses untuk mendapatkan kekuatan tepi, selanjutnya gradien citra dapat dihitung menggunakan Persamaan (2-2).

$$|G| = |G_x| + |G_y| \quad (2-2)$$

c. Langkah 3 :

Langkah selanjutnya adalah perhitungan arah tepi dengan menggunakan Persamaan (2-3).

$$\theta = \tan^{-1}(G_y, G_x) \quad (2-3)$$

d. Langkah 4 :

Kemudian setelah arah tepi diperoleh, perlu menghubungkan antara arah tepi dengan sebuah arah yang dapat dilacak dari citra. Sebagai contoh, terdapat susunan piksel berukuran 5x5 seperti pada gambar 2.1. Dengan melihat piksel “a” tampak a memiliki 4 arah yaitu, 0° , 45° , 90° , dan 135° .

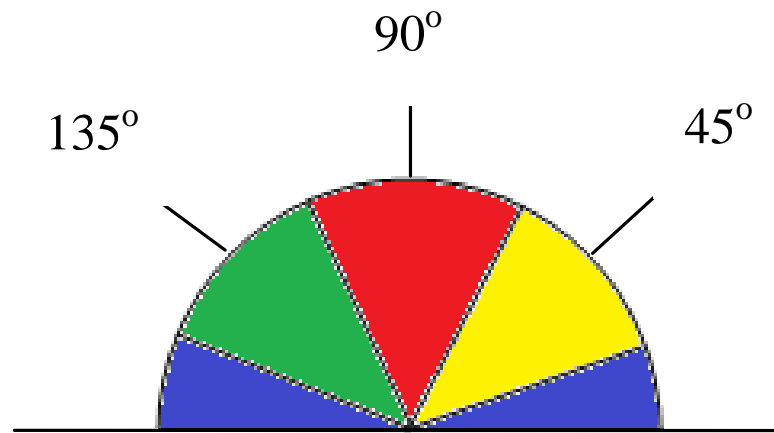
X	X	X	X	X
X	X	X	X	X
X	X	a	X	X
X	X	X	X	X
X	X	X	X	X

Gambar 2.1 Contoh Matriks Piksel Berukuran 5x5

Selanjutnya, arah tepi yang diperoleh dikelompokkan ke dalam salah satu dari keempat arah tersebut berdasarkan area yang tertera pada Gambar 2.2.

Konversi yang berlaku dapat dilihat pada Persamaan (2-4).

$$arah = \begin{cases} 0 & y_i^{(H)} = f^{(H)} \left(\sum_j w_{ij}^{(HI)} x_j \right) \\ 45 & \\ 90 & y_i = f^{(T)} \left(\sum_j w_{ij}^{(TH)} y_j^{(H)} \right) \\ 135 & \end{cases} \quad (2-4)$$



Gambar 2.2 Area untuk Mengonversi Arah Tepi ke Dalam kategori salah satu dari arah 0°, 45°, 90°, dan 135°

Semua arah tepi yang berada di kisaran 0° dan 22.5° serta 157.5° dan 180° (berwarna biru) diubah menjadi 0°. Semua arah tepi yang berada di kisaran 22.5° dan 67.5° (berwarna kuning) diubah menjadi 45°. Semua arah tepi yang berada di kisaran 67.5° dan 112.5° (berwarna merah) diubah menjadi 90°. Semua arah tepi yang berada di kisaran 112.5° dan 157.5° (berwarna hijau) diubah menjadi 135°.

e. Langkah 5 :

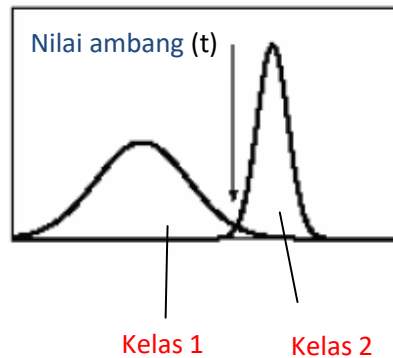
Setelah memperoleh nilai tepi, dilakukan penghilangan non-maksimum. Penghilangan non-maksimum dilakukan sepanjang tepi pada arah tepi dan menghilangkan piksel-piksel (mengatur piksel sebagai 0) yang tidak dianggap sebagai tepi sehingga diperoleh tepi yang tipis.

f. Langkah 6 :

Terakhir adalah proses yang disebut *hysteresis*. Pada proses ini, garis yang terlihat putus-putus pada tepi objek dihilangkan dengan menggunakan dua ambang T_1 dan T_2 . Semua piksel citra yang memiliki nilai lebih kecil dari T_1 akan diubah menjadi hitam (nilai 0). Selanjutnya semua piksel yang terhubung dengan piksel tersebut yang bernilai lebih besar dari T_2 akan diubah menjadi putih (nilai 255).

2. Metode *Otsu-Thresholding*

Metode ini merupakan metode yang dikemukakan oleh Nobuyuki Otsu pada tahun 1979, yang menentukan nilai ambang dengan cara membedakan dua kelompok yaitu objek dengan latar belakang, yang memiliki bagian saling bertumpukan berdasarkan histogram seperti pada Gambar 2.3 [26].



Gambar 2.3 Penentuan Nilai Ambang

Prinsip metode *otsu* adalah sebagai berikut [26]. Pertama-tama, probabilitas nilai intensitas i dalam histogram dihitung menggunakan Persamaan (2-5).

$$p(i) = \frac{n_i}{N}, p(i) \geq 0, \sum_{i=1}^{256} p(i) = 1 \quad (2-5)$$

Dengan n_i adalah jumlah piksel dengan intensitas i dan N adalah jumlah total piksel pada citra. Jika histogram dibagi menjadi objek dan latar belakang, maka pemberian bobot pada kedua kelas dinyatakan dengan Persamaan (2-6) dan (2-7).

$$w_1(t) = \sum_{i=1}^t p(i) \quad (2-6)$$

$$w_2(t) = \sum_{i=t+1}^L p(i) = 1 - w_1(t) \quad (2-7)$$

Dalam hal ini, L adalah jumlah aras keabuan. Rata-rata kedua kelas dapat dihitung dengan menggunakan Persamaan (2-9) dan (2-8).

$$m_1(t) = \sum_{i=1}^t i \cdot p(i) / W_1(t) \quad (2-8)$$

$$m_2(t) = \sum_{i=1}^t i \cdot p(i) / W_2(t) \quad (2-9)$$

Varians kedua kelas dinyatakan dengan Persamaan (2-10) dan (2-11).

$$\sigma_1^2(t) = \sum_{i=1}^t (1 - m_1)^2 \cdot \frac{p(i)}{W_1(t)} \quad (2-10)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^L (1 - m_2)^2 \cdot \frac{p(i)}{W_2(t)} \quad (2-11)$$

Varians total dapat dinyatakan dengan menggunakan Persamaan (2-12).

$$\sigma^2(t) = \sigma_W^2(t) + \sigma_B^2(t) \quad (2-12)$$

Dalam hal ini, σ_W^2 dinamakan sebagai *within-class variance* (WCV) dan σ_B^2 disebut dengan *between-class variance* (BCV). WVC dapat dinyatakan dengan menggunakan Persamaan (2-13).

$$\sigma_W^2(t) = W_1(t) \cdot \sigma_1(t)^2 + W_2(t) \cdot \sigma_2(t)^2 \quad (2-13)$$

Rumus di atas menunjukkan bahwa WCV adalah jumlah varians kelas secara individual yang telah diberi bobot dengan probabilitas kelas masing-masing. Adapun BCV dinyatakan dengan menggunakan Persamaan (2-14).

$$\sigma_B^2(t) = W_1 \cdot [m_1(t) - m_T]^2 + W_2 \cdot [m_2(t) - m_T]^2 \quad (2-14)$$

Dalam hal ini, m_T adalah rerata total yang dapat dihitung menggunakan Persamaan (2-15).

$$m_T = \sum_{i=1}^N i \cdot p(i) \quad (2-15)$$

Nilai ambang optimum dapat diperoleh dengan cara meminimumkan WCV atau dengan memaksimalkan BCV [26].

2.2.3 Ekstraksi Fitur

1. Gray Level Co-occurrence Matrices (GLCM)

Gray Level Co-occurrence Matrices (GLCM) merupakan ekstraksi fitur yang diusulkan pada tahun 1973 oleh Haralick. GLCM menggunakan perhitungan tekstur pada orde kedua yang memperhitungkan hubungan antar pasangan dua piksel citra asli. Ekstraksi fitur ini mendeskripsikan tekstur gambar dengan menunjukkan seberapa sering pasangan suatu piksel terjadi pada citra [13][26].

Untuk mendapatkan fitur GLCM, hanya beberapa fitur usulan Haralick yang digunakan yaitu *angular second moment* (ASM), *contrast*, *inverse different moment* (IDM), *entropy*, dan *correlation* [26][27].

ASM adalah ukuran homogenitas citra yang dapat dihitung dengan menggunakan Persamaan (2-16).

$$ASM = \sum_{i=1}^L \sum_{j=1}^L (GLCM(i, j))^2 \quad (2-16)$$

L menyatakan jumlah level yang digunakan dalam komputasi.

Kontras adalah ukuran keberadaan variasi aras keabuan piksel citra yang dapat dihitung dengan menggunakan Persamaan (2-17).

$$Kontras = \sum_{n=1}^L n^2 \{ \sum_{|i-j|=n} GLCM(i, j) \} \quad (2-17)$$

Inverse different moment (IDM) digunakan untuk mengukur homogenitas yang dapat dihitung dengan Persamaan (2-18).

$$IDM = \sum_{i=1}^L \sum_{j=1}^L \frac{(GLCM(i, j))^2}{1+(i-j)^2} \quad (2-18)$$

Entropi merupakan ukuran ketidakaturan aras keabuan di dalam citra. Nilai dikatakan tinggi jika elemen-elemen GLCM mempunyai nilai yang relatif sama dan dikatakan rendah jika elemen-elemen GLCM dekat dengan nilai 0 atau

1. Entropi dapat dihitung dengan Persamaan (2-19).

$$Entropi = - \sum_{i=1}^L \sum_{j=1}^L (GLCM(i, j) \log(GLCM(i, j))) \quad (2-19)$$

Korelasi merupakan ukuran ketergantungan linear antar nilai aras keabuan yang dapat dihitung menggunakan Persamaan (2-20).

$$Korelasi = \sum_{i,j=1}^L (GLCM(i,j)) \frac{(i-\mu)(j-\mu)}{\sigma^2} \quad (2-20)$$

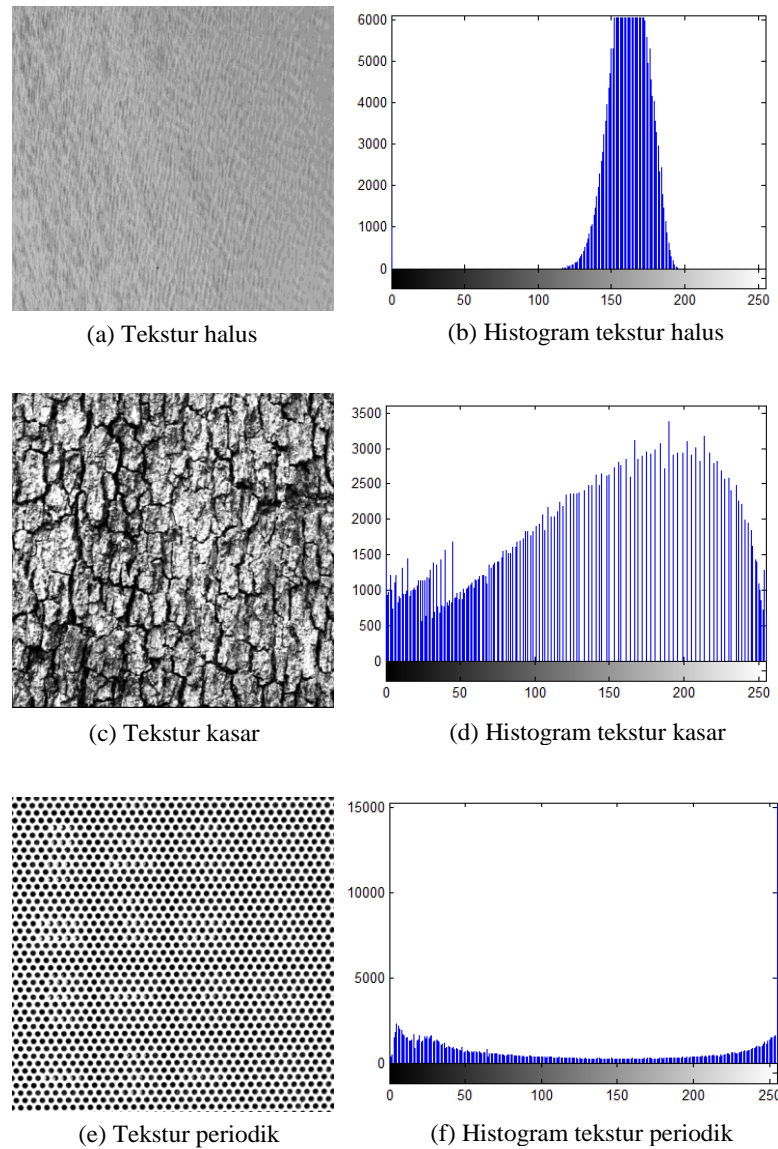
dengan ketentuan seperti pada Persamaan (2-21) dan Persamaan (2-22) berikut

$$\mu = \sum_{i,j=1}^L i * GLCM(i,j) \quad (2-21)$$

$$\sigma = \sum_{i,j=1}^L GLCM(i,j) (i - \mu)^2 \quad (2-22)$$

2. Ekstraksi Fitur Statistik

Metode yang sederhana untuk mendapatkan tekstur adalah dengan mendasarkan pada histogram. Berikut merupakan contoh histogram dari tiga buah citra yang memiliki tekstur berbeda yang dapat dilihat pada Gambar 2.4. Gambar 2.4(a) menunjukkan bahwa citra dengan tekstur halus memiliki perubahan intensitas yang sempit. Sebaliknya, citra dengan tekstur yang kasar memiliki kontras yang tinggi ditandai dengan jangkauan intensitas yang lebar yang dapat dilihat pada Gambar 2.4(c) dan Gambar 2.4(d). Menurut penglihatan, meskipun beraturan citra pada Gambar 2.4(e) termasuk kasar jika dibandingkan dengan citra pada Gambar 2.4(a).



Gambar 2.4 Contoh Histogram pada Citra dengan Tekstur yang Berbeda

Fitur pertama yang dihitung secara statis adalah rerata intensitas. Komponen ini dapat dihitung dengan menggunakan Persamaan (2-23)

$$m = \sum_{i=0}^{L-1} i \cdot p(i) \quad (2-23)$$

Dalam hal ini, i merupakan aras keabuan pada citra f dan $p(i)$ merupakan probabilitas kemunculan i , dan L merupakan aras keabuan tertinggi. Persamaan tersebut akan menghasilkan rerata kecerahan.

Fitur kedua berupa deviasi standar yang dapat dihitung menggunakan Persamaan (2-24)

$$\sigma = \sqrt{\sum_{i=0}^{L-1} (i - m)^2 p(i)} \quad (2-24)$$

Dalam hal ini, σ^2 menyatakan varian atau momen orde dua ter-normalisasi karena $p(i)$ merupakan fungsi peluang. Persamaan tersebut menghasilkan ukuran kontras.

Fitur *skewness* merupakan ukuran ketidaksimetrisan terhadap rerata intensitas, yang dapat dihitung dengan menggunakan Persamaan (2-25)

$$skewness = \sum_{i=0}^{L-1} (i - m)^3 p(i) \quad (2-25)$$

Skewness biasa disebut dengan momen orde tiga ter-normalisasi. Nilai negatif menyatakan bahwa distribusi kecerahan cenderung condong ke kiri terhadap rerata dan nilai positif menyatakan bahwa distribusi kecerahan condong ke kanan terhadap rerata. Untuk normalisasi, nilai *skewness* dibagi dengan $(L-1)^2$.

Deskriptor energi adalah ukuran yang menyatakan distribusi intensitas piksel terhadap aras keabuan yang dapat dihitung dengan menggunakan Persamaan (2-26)

$$energi = \sum_{i=0}^{L-1} [p(i)]^2 \quad (2-26)$$

Citra yang seragam dengan satu nilai aras keabuan akan memiliki energi yang maksimum, yaitu sebesar 1. Secara umum, citra dengan sedikit aras keabuan akan memiliki energi yang lebih tinggi daripada yang memiliki banyak nilai aras keabuan. Energi sering disebut dengan keseragaman.

Entropi mengindikasikan kompleksitas citra. Entropi dapat dihitung menggunakan Persamaan (2-27)

$$entropi = - \sum_{i=0}^{L-1} p(i) \log_2(p(i)) \quad (2-27)$$

Semakin tinggi nilai entropi, maka citra tersebut akan semakin kompleks. Energi dan entropi berkecenderungan berkebalikan. Entropi juga merepresentasikan jumlah informasi yang terkandung di dalam suatu sebaran data.

Properti kehalusan bisa disertakan untuk mengukur tingkat kehalusan atau kekasaran intensitas pada citra, yang dapat dihitung menggunakan Persamaan (2-28)

$$R = 1 - \frac{1}{1 + \sigma^2} \quad (2-28)$$

Dalam hal ini, σ merupakan deviasi standar. Nilai R yang rendah menunjukkan bahwa citra memiliki intensitas yang kasar. Dalam menghitung kehalusan, varians perlu dinormalisasi sehingga nilainya berada dalam jangkauan $[0 \ 1]$ dengan cara membaginya dengan $(L-1)^2$.

2.2.4 Klasifikasi

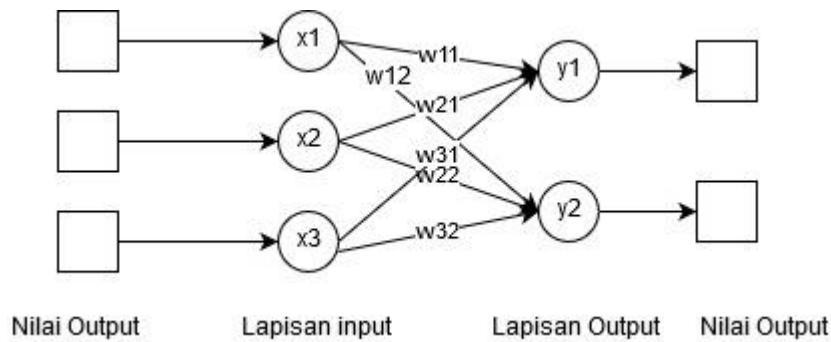
Artificial neural network (ANN) merupakan sistem pembelajaran yang merepresentasikan kinerja dari otak manusia, yang di mana setiap informasi *input* dan *output*-nya dimasukan kedalam sistem dan diproses dalam suatu neuron. Neuron-neuron tersebut dikumpulkan dalam lapisan-lapisan yang disebut dengan *neuron layer* dan terbagi menjadi 3 lapisan penyusun, yaitu [28]:

1. Lapisan *input* (*input layer*), pada *layer* ini unit *input* akan menerima pola data dari luar yang menggambarkan suatu permasalahan.
2. Lapisan tersembunyi (*hidden layer*), pada *layer* ini unit tersembunyi akan mengolah data dan menghasilkan *output* yang tidak dapat diamati secara langsung.
3. Lapisan *output* (*output layer*), pada *layer* ini unit *output* akan menghasilkan keluaran yang merupakan solusi ANN terhadap suatu masalah.

ANN memiliki beberapa arsitektur jaringan yang sering digunakan dalam suatu aplikasi antara lain [28]:

1. Jaringan Lapisan Tunggal (*Single Layer Network*)

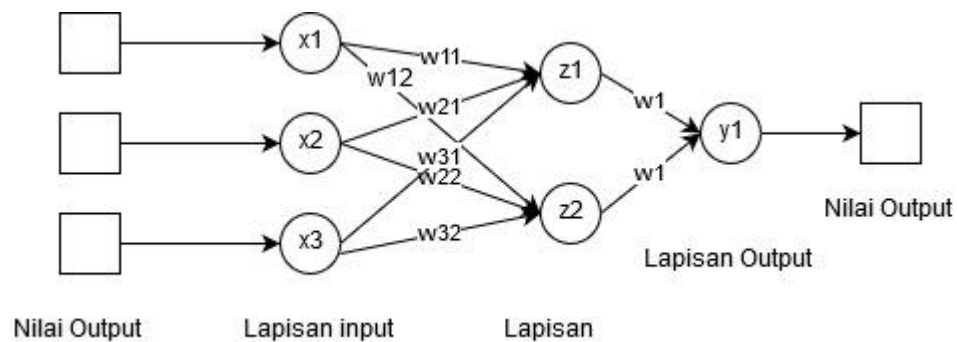
Jaringan ini terdiri dari 1 lapisan *input* dan 1 lapisan *output*. Setiap neuron yang ada pada *layer input* akan selalu terhubung dengan neuron yang ada pada *layer output*. Jaringan ini hanya menerima *input* dan langsung mengolahnya menjadi *output* tanpa melalui *hidden layer*. Jaringan lapisan tunggal dapat dilihat pada Gambar 2.5.



Gambar 2.5 Arsitektur Lapisan Tunggal

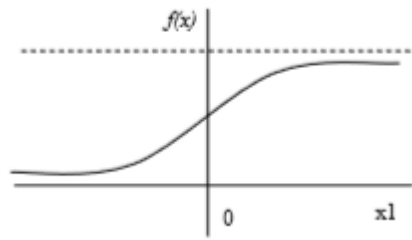
2. Jaringan Banyak Lapisan (*Multilayer Network*)

Jaringan lapisan ini terdiri dari tiga lapisan yaitu lapisan *input*, lapisan *output*, dan lapisan tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan jaringan dengan lapisan tunggal, namun memiliki kekurangan dimana proses pembelajaran cenderung membutuhkan waktu yang lebih lama. Jaringan banyak lapisan dapat dilihat pada Gambar 2.5.



Gambar 2.6 Arsitektur Jaringan *Multilayer*

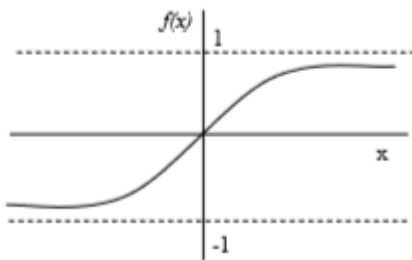
Pada klasifikasi *Artificial Neural Network backpropagation*, fungsi aktivasi yang digunakan harus memenuhi beberapa syarat yaitu: kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang sering digunakan dan telah memenuhi ketiga syarat tersebut adalah fungsi *sigmoid biner* yang memiliki range (0,1). Diberikan $F(x) = \frac{1}{1+e^{-x}}$ dengan turunan $f'(x) = f(x)(1 - f(x))$. Grafik fungsi *sigmoid biner* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Fungsi Aktivasi *Sigmoid Biner*

Fungsi lain yang sering digunakan adalah fungsi *sigmoid bipolar* yang bentuk fungsinya mirip dengan fungsi *sigmoid biner*, namun dengan *range* $(-1,1)$.

Diberikan $F(x) = \frac{2}{1+e^{-x}} - 1$ dengan turunan $f(x) = \frac{(1+f(x))(1-f(x))}{2}$. Grafik fungsi *sigmoid bipolar* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Fungsi Aktivasi *Sigmoid Bipolar*

Fungsi *sigmoid* memiliki nilai maksimum = 1. Maka untuk pola yang targetnya lebih dari 1, pola masukan dan keluaran harus terlebih dahulu ditransformasi sehingga semua polanya memiliki *range* yang sama seperti fungsi *sigmoid* yang dipakai. Alternatif lain adalah menggunakan fungsi aktivasi *sigmoid* hanya pada lapisan yang bukan lapisan keluaran. Pada lapisan keluaran, fungsi aktivasi yang dipakai adalah fungsi identitas[28] : $f(x) = x$

Pelatihan standar pada klasifikasi ANN *backpropagation* meliputi 3 tahap yaitu tahap maju, propagasi mundur, dan perubahan bobot. Algoritma pelatihan untuk jaringan dengan satu lapisan tersembunyi (dengan fungsi aktivasi *sigmoid biner*) adalah sebagai berikut[28][29][30]:

- Langkah 0 : Inisialisasi bobot dengan bilangan acak kecil antara 0 dan 1
- Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-8
- Langkah 2 : Untuk setiap data pelatihan, lakukan langkah 3-8

Tahap I : Propagasi Maju

Langkah 3 : Tiap *input unit* menerima sinyal dan meneruskannya ke *hidden unit* di atasnya

Langkah 4 : Hitung semua keluaran di unit tersembunyi $z_j (j = 1, 2, \dots, p)$ dengan menggunakan Persamaan (2-29) dan (2-30)

$$z_net_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2-29)$$

$$z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}} \quad (2-30)$$

Langkah 5 : Hitung semua keluaran jaringan di unit $y_k (k = 1, 2, \dots, m)$ dengan menggunakan Persamaan (2-31) dan (2-32)

$$y_net_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2-31)$$

$$y_k = f(y_net_k) = \frac{1}{1 + e^{-y_net_k}} \quad (2-32)$$

Tahap II : Propagasi mundur

Langkah 6 : Hitung faktor δ unit keluaran berdasarkan error di setiap unit keluaran $y_k (k = 1, 2, \dots, m)$ dengan Persamaan (2-33)

$$\delta_k = (t_k - y_k) f'(y_net_k) = (t_k - y_k) y_k (1 - y_k) \quad (2-33)$$

Hitung suku perubahan bobot w_{jk} dengan laju percepatan α dengan Persamaan (2-34)

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (2-34)$$

Langkah 7 : Hitung faktor δ unit tersembunyi berdasarkan error di setiap unit tersembunyi $z_j (j = 1, 2, \dots, p)$ dengan persamaan (2-35) dan (2-36)

$$\delta_net_j = \sum_{k=1}^m \delta_k w_{jk} \quad (2-35)$$

$$\delta_j = \delta_net_j f'(z_net_j) = \delta_net_j z_j (1 - z_j) \quad (2-36)$$

Hitung suku perubahan bobot v_{ji} dengan Persamaan (2-37)

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2-37)$$

dimana : $j = 1, 2, \dots, p$ dan $i = 0, 1, \dots, n$

Tahap III : Perubahan bobot

Langkah 8 : Hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran dengan Persamaan (2-38)

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2-38)$$

dimana: $k = 1, 2, \dots, m$ dan $j = 0, 1, \dots, p$

Perubahan bobot garis yang menuju ke unit tersembunyi dengan Persamaan (2-39)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2-39)$$

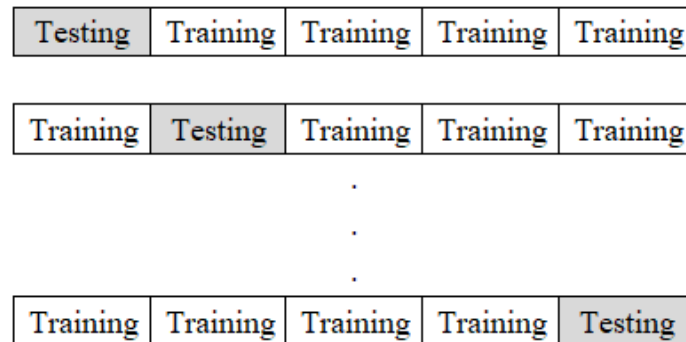
dimana: $j = 1, 2, \dots, p$ dan $i = 0, 1, \dots, n$

Langkah 9 : Uji kondisi berhenti (akhir iterasi).

2.2.5 Pengujian Performa Klasifikasi

1. *K-fold Cross Validation*

Metode ini membagi data menjadi dua bagian, yaitu data pelatihan dan data pengujian. Selanjutnya, sebuah subset data akan dijadikan data uji dan subset data lainnya akan dijadikan sebagai data latih, kemudian dilakukan proses silang dimana salah satu subset data pengujian sebelumnya akan dijadikan sebagai data pelatihan dan subset data pelatihan sebelumnya akan dijadikan sebagai data pengujian. Proses ini akan terus berlangsung hingga sebuah subset data akan menjadi data pelatihan masing-masing sebanyak satu kali dan proses akan diulangi sebanyak k , seperti pada Gambar 2.9 [31].



Gambar 2.9 *K-fold Cross Validation* dengan $k = 5$

2. *Shuffled Sampling*

Metode pembagian data ini merupakan pengujian citra yang dilakukan dengan membagi data menjadi dua bagian, yaitu data latih dan data uji dengan skenario perbandingan 70:30, 80:20, dan 90:10 yang ditentukan secara acak [32] .

2.2.6 Evaluasi Performa Klasifikasi

Untuk mengetahui seberapa handal suatu *classifier* dalam mengenali objek-objek dalam kelas yang berbeda, digunakan matriks konfusi berbentuk tabel dengan ukuran m baris dan m kolom. Matriks konfusi untuk dua kelas dapat dilihat pada Tabel 2.2 [31].

Tabel 2.2 Tabel Konfusi

		Predicted class	
		C ₁	C ₂
Actual class	C ₁	True positive (TP)	False negative (FN)
	C ₂	False positive (FP)	True negative (TN)

Dimana C₁ adalah kelas *positive* dan C₂ adalah kelas *negative*. Pada tabel di atas terdapat empat kemungkinan yaitu, *true positive* (TP) adalah banyaknya hasil klasifikasi benar pada suatu kelas bernilai *positive* (glaukoma), *false positive* (FP) adalah banyaknya hasil klasifikasi salah pada suatu kelas bernilai *positive* (glaukoma), *false negative* (FN) adalah banyaknya hasil klasifikasi salah pada suatu kelas bernilai *negative* (normal), dan *true negative* (TN) adalah banyaknya hasil klasifikasi benar pada suatu kelas bernilai *negative* (normal) [31].

Pengukuran evaluasi performa klasifikasi terdiri dari akurasi, sensitivitas, dan spesifisitas. Akurasi merupakan presentasi kesuksesan dari sebuah klasifikasi yang dapat dihitung menggunakan Persamaan (2-40). Sensitivitas merupakan presentasi *classifier* dalam memprediksi kelas *positive* yang dapat dihitung dengan menggunakan Persamaan (2-41). Spesifisitas merupakan presentasi *classifier* dalam memprediksi kelas *negative* yang dapat dihitung menggunakan Persamaan (2-42) [3][31].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2-40)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (2-41)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (2-42)$$

BAB III METODOLOGI PENELITIAN

3.1 Alat dan Bahan Penelitian

Alat dan bahan pada penelitian yang dilakukan berupa *software* dan *hardware* serta data-data yang dibutuhkan selama kegiatan

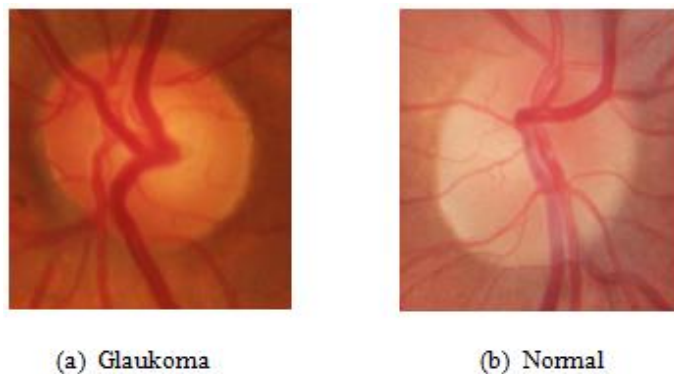
1. Alat penelitian

Alat-alat yang akan digunakan dalam melakukan penelitian ini adalah sebagai berikut:

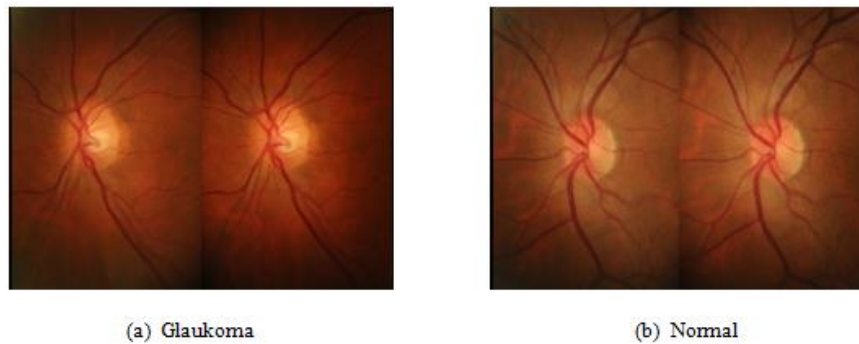
- *Processor Laptop* : Intel(R) Core(TM) i3-2377M CPU @ 1.50GHz
- *Sistem Operasi* : Windows 10 64-bit.
- *Bahasa Pemrograman* : Python

2. Bahan Penelitian

Bahan yang digunakan pada penelitian ini merupakan *dataset* RIM-ONE r2 berupa 255 data citra fundus mata normal dan 200 data citra fundus mata diduga dan positif glaukoma dan *dataset* RIM-ONE r3 berupa 85 data citra fundus mata normal dan 74 data citra fundus mata diduga dan positif glaukoma [33], keduanya merupakan *dataset* citra fundus yang dapat diunduh dari <http://medimrg.webs.ull.es/research/downloads/> dengan format file “.zip”. Contoh citra fundus retina dari RIM-ONE r2 dan RIM-ONE r3 dapat dilihat pada Gambar 3.1 dan Gambar 3.2. Citra fundus yang digunakan merupakan citra fundus retina yang telah melalui proses *cropping* secara manual dengan format .jpg dan resolusi sebesar 300 x 300 piksel untuk setiap *dataset*.



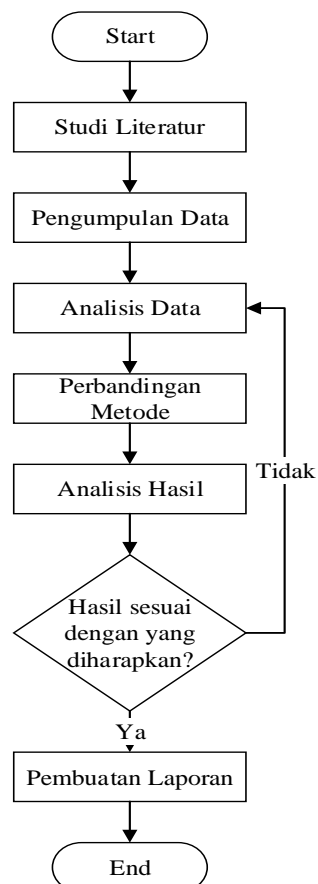
Gambar 3.1 Citra Fundus Retina RIM-ONE r2



Gambar 3.2 Citra Fundus Retina RIM-ONE r3

3.2 Proses Penelitian

Proses penelitian dilakukan untuk mengetahui tahapan-tahapan penelitian yang akan dilakukan, mulai dari studi literatur sampai dengan penyusunan laporan. Proses penelitian yang dilakukan pada penelitian ini digambarkan pada diagram alir pada Gambar 3.3.

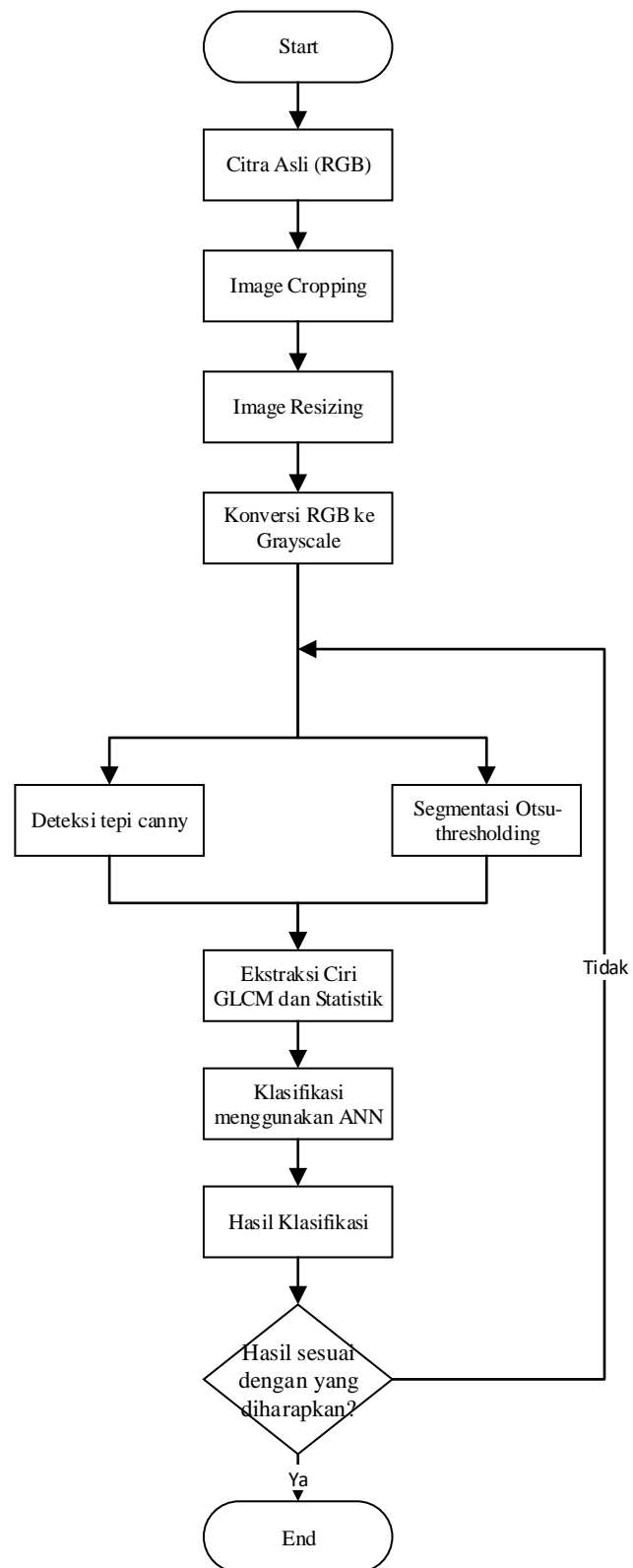


Gambar 3.3 Diagram Alir Proses Penelitian

Berdasarkan Gambar 3.3, ada beberapa tahap dalam proses penelitian ini, yaitu pertama melakukan studi literatur dimana pada tahap ini dilakukan dengan mempelajari buku-buku, karya ilmiah, jurnal-jurnal penelitian sebelumnya, serta sumber lain yang berkaitan dengan penelitian yang dilakukan. Selanjutnya merupakan proses pengumpulan data, dimana data yang digunakan didapatkan dari *dataset* yang tersedia di internet. Kemudian tahap selanjutnya adalah menganalisis data yang telah dikumpulkan. Setelah itu dilakukan perbandingan metode antara metode *otsu-thresholding* dan deteksi tepi *canny*. Selanjutnya dilakukan analisis hasil dimana apabila hasil tidak sesuai maka akan kembali ke tahap analisis data. Selanjutnya jika sistem telah berjalan dengan baik maka akan dilanjutkan ke tahap terakhir yaitu pembuatan laporan dari awal pengumpulan data sampai akhir pembuatan sistem.

3.3 Perancangan Sistem

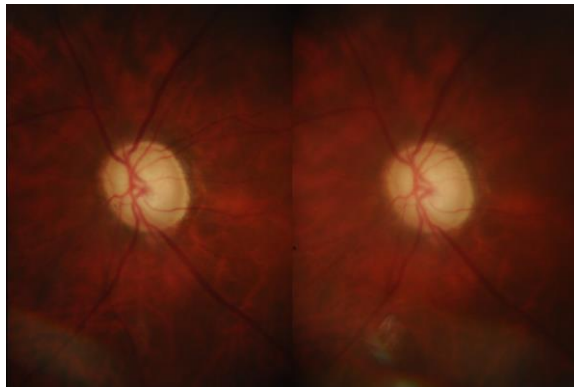
Pada perancangan sistem ini, dijelaskan bagaimana sistem dirancang mulai dari tahap pelatihan sistem hingga sistem dapat mengklasifikasi penyakit glaukoma. Proses perancangan sistem digambarkan pada Gambar 3.4.



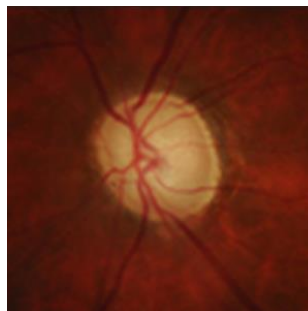
Gambar 3.4 Diagram Perancangan Sistem

3.3.1 Tahap *Preprocessing*

Untuk meningkatkan kualitas citra fundus masukan, pada tahap *preprocessing* ini dilakukan proses *cropping*, *resizing*, dan konversi warna. Proses *cropping* bertujuan untuk mengambil bagian citra fundus yang dibutuhkan dan membuang bagian citra fundus yang tidak dibutuhkan. Berikut merupakan contoh *cropping* yang dapat dilihat pada Gambar 3.5.

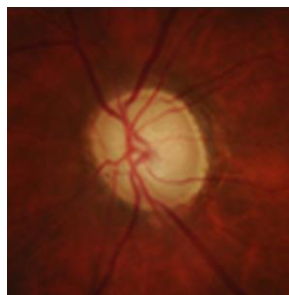


Gambar 3.5 Citra Fundus Asli 2144x1424 Piksel



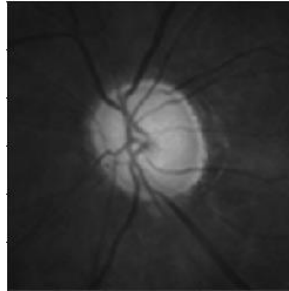
Gambar 3.6 Citra Fundus Hasil *Cropping* 609x717 Piksel

Resizing dilakukan untuk mengubah resolusi citra fundus menjadi 300x300 piksel. Proses *cropping* dan *resizing* ini dilakukan di luar sistem. Contoh hasil *resizing* dapat dilihat pada Gambar 3.7.



Gambar 3.7 Proses *Resize* Menjadi 300x300 Piksel

Setelah resolusi citra fundus diubah, selanjutnya melakukan konversi citra RGB menjadi *grayscale*. Proses *grayscale* adalah proses pertama yang dilakukan di dalam sistem. Contoh konversi ruang warna dapat dilihat pada Gambar 3.8.



Gambar 3.8 Hasil Konversi Ruang Warna dari RGB ke *Grayscale*

Proses *cropping*, *resizing*, dan konversi ruang warna dilakukan untuk mempermudah proses ekstraksi fitur yang akan dilakukan.

3.3.2 Segmentasi

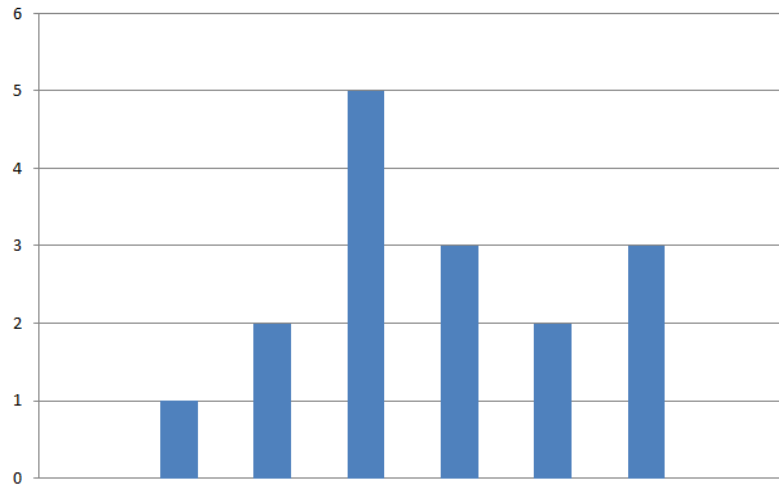
1. *Otsu-thresholding*

Berikut merupakan contoh perhitungan *otsu-thresholding*. Misalkan terdapat citra dengan matriks 4x4 seperti pada Gambar 3.9.

$$A = \begin{array}{|c|c|c|c|} \hline 1 & 2 & 4 & 6 \\ \hline 3 & 5 & 6 & 4 \\ \hline 3 & 6 & 3 & 3 \\ \hline 5 & 4 & 3 & 2 \\ \hline \end{array}$$

Gambar 3.9 Contoh Matriks 4x4

Langkah pertama yaitu mengecek histogram dari matriks A di atas. Dari matriks tersebut, didapatkan grafik histogram seperti pada Gambar 3.10.



Gambar 3.10 Grafik Histogram Matriks A

Setelah didapatkan grafik histogram, selanjutnya dihitung probabilitas nilai intensitas i dengan menggunakan Persamaan (2-5).

$$\begin{aligned}
 p(1) &= \left(\frac{0}{16}\right), p(2) = \left(\frac{1}{16}\right), p(3) = \left(\frac{2}{16}\right), p(4) = \left(\frac{5}{16}\right), p(5) = \left(\frac{3}{16}\right), p(6) = \left(\frac{2}{16}\right), p(7) \\
 &= \left(\frac{3}{16}\right), p(8) = \left(\frac{0}{16}\right) \\
 &= 0, 0.0625, 0.125, 0.3125, 0.1875, 0.125, 0.1875, 0
 \end{aligned}$$

Menghitung pembobotan masing-masing kelas dengan menggunakan Persamaan (2-6) dan (2-7).

$$\begin{aligned}
 w_1(t) &= w_1(1) = 0, w_1(2) = 0 + 0.0625, w_1(3) = 0 + 0.0625 + 0.125, w_1(4) = 0 + \\
 &0.0625 + 0.125 + 0.3125, w_1(5) = 0 + 0.0625 + 0.125 + 0.3125 + 0.1875, \\
 w_1(6) &= 0 + 0.0625 + 0.125 + 0.3125 + 0.1875 + 0.125, w_1(7) = 0 + 0.0625 \\
 &+ 0.125 + 0.3125 + 0.1875 + 0.125 + 0.1875, w_1(8) = 0 + 0.0625 + 0.125 \\
 &+ 0.3125 + 0.1875 + 0.125 + 0.1875 + 0.
 \end{aligned}$$

$$\begin{aligned}
 w_1(t) &= w_1(1) = 0, w_1(2) = 0.0625, w_1(3) = 0.1875, w_1(4) = 0.5, w_1(5) = 0.6875, \\
 w_1(6) &= 0.8125, w_1(7) = 1, w_1(8) = 1.
 \end{aligned}$$

$$\begin{aligned}
 w_2(t) &= w_2(1) = 1 - 0, w_2(2) = 1 - 0.0625, w_2(3) = 1 - 0.1875, w_2(4) = 1 - 0.5, \\
 w_2(5) &= 1 - 0.6875, w_2(6) = 1 - 0.8125, w_2(7) = 1 - 1, w_2(8) = 1 - 1.
 \end{aligned}$$

$$\begin{aligned}
 w_2(t) &= w_2(1) = 1, w_2(2) = 0.9375, w_2(3) = 0.8125, w_2(4) = 0.5, w_2(5) = 0.3125, \\
 w_2(6) &= 0.1875, w_2(7) = 0, w_2(8) = 0.
 \end{aligned}$$

Kemudian menghitung rerata kedua kelas dengan menggunakan Persamaan (2-8) dan (2-9).

$$m_1(1) = 0$$

$$\begin{aligned} m_1(2) &= (1 * 0/0.0625) + (2 * 0.0625/0.0625) \\ &= 0 + 2 \end{aligned}$$

$$m_1(2) = 2$$

$$\begin{aligned} m_1(3) &= (1 * 0/0.1875) + (2 * 0.0625/0.1875) + (3 * 0.125/0.1875) \\ &= 0 + 0.6667 + 2 \end{aligned}$$

$$m_1(3) = 2.6667$$

$$\begin{aligned} m_1(4) &= (1 * 0/0.5) + (2 * 0.0625/0.5) + (3 * 0.125/0.5) + (4 * 0.3125/0.5) \\ &= 0 + 0.25 + 0.75 + 2.5 \end{aligned}$$

$$m_1(4) = 3.5$$

$$\begin{aligned} m_1(5) &= (1 * 0/0.6875) + (2 * 0.0625/0.6875) + (3 * 0.125/0.6875) + (4 * 0.3125/0.6875) + (5 * 0.1875/0.6875) \\ &= 0 + 0.1818 + 0.5454 + 1.8181 + 1.3636 \end{aligned}$$

$$m_1(5) = 3.9089$$

$$\begin{aligned} m_1(6) &= (1 * 0/0.8125) + (2 * 0.0625/0.8125) + (3 * 0.125/0.8125) + (4 * 0.3125/0.8125) + (5 * 0.1875/0.8125) + (6 * 0.125/0.8125) \\ &= 0 + 0.1538 + 0.4615 + 1.5385 + 1.1538 + 0.9231 \end{aligned}$$

$$m_1(6) = 4.2307$$

$$\begin{aligned} m_1(7) &= (1 * 0/1) + (2 * 0.0625/1) + (3 * 0.125/1) + (4 * 0.3125/1) + (5 * 0.1875/1) + (6 * 0.125/1) + (7 * 0.1875/1) \\ &= 0 + 0.125 + 0.375 + 1.25 + 0.9375 + 0.75 + 1.3125 \end{aligned}$$

$$m_1(7) = 4.75$$

$$\begin{aligned} m_1(8) &= (1 * 0/1) + (2 * 0.0625/1) + (3 * 0.125/1) + (4 * 0.3125/1) + (5 * 0.1875/1) + (6 * 0.125/1) + (7 * 0.1875/1) + (8 * 0/1) \\ &= 0 + 0.125 + 0.375 + 1.25 + 0.9375 + 0.75 + 1.3125 + 0 \end{aligned}$$

$$m_1(8) = 4.75$$

$$m_2(1) = (1 * 0/1)$$

$$m_2(1) = 0$$

$$m_2(2) = (1 * 0/0.93750) + (2 * 0.0625/0.93750)$$

$$= 0 + 0.13333$$

$$m_2(2) = 0.13333$$

$$m_2(3) = (1 * 0/0.81250) + (2 * 0.0625/0.81250) + (3 * 0.125/0.81250)$$

$$= 0 + 0.15385 + 0.46154$$

$$m_2(3) = 0.61538$$

$$m_2(4) = (1 * 0/0.5) + (2 * 0.0625/0.5) + (3 * 0.125/0.5) + (4 * 0.3125/0.5)$$

$$= 0 + 0.25 + 0.75 + 2.5$$

$$m_2(4) = 3.5$$

$$m_2(5) = (1 * 0/0.31250) + (2 * 0.0625/0.31250) + (3 * 0.125/0.31250) + (4 * 0.3125/0.31250) + (5 * 0.1875/0.31250)$$

$$= 0 + 0.4 + 1.2 + 4 + 3$$

$$m_2(5) = 8.6$$

$$m_2(6) = (1 * 0/0.18750) + (2 * 0.0625/0.18750) + (3 * 0.125/0.18750) + (4 * 0.3125/0.18750) + (5 * 0.1875/0.18750) + (6 * 0.125/0.18750)$$

$$= 0 + 0.66667 + 2 + 0.66667 + 5 + 4$$

$$m_2(6) = 18.333$$

$$m_2(7) = 0$$

$$m_2(8) = 0$$

Varians kedua kelas dihitung menggunakan Persamaan (2-10) dan (2-11).

$$\sigma_1^2(1) = ((1 - 0)^2 * 0/0)$$

$$\sigma_1^2(1) = 0$$

$$\sigma_1^2(2) = ((1 - 2)^2 * 0/0.0625) + ((1 - 2)^2 * 0.0625/0.0625)$$

$$= 0 + 1$$

$$\sigma_1^2(2) = 1$$

$$\sigma_1^2(3) = ((1 - 2.6667)^2 * 0/0.1875) + ((1 - 2.6667)^2 * 0.0625/0.1875) + ((1 - 2.6667)^2 * 0.125/0.1875)$$

$$= 0 + 0.926 + 1.8519$$

$$\sigma_1^2(3) = 2.7779$$

$$\begin{aligned}\sigma_1^2(4) &= ((1 - 3.5)^2 * 0/0.5) + ((1 - 3.5)^2 * 0.0625/0.5) + ((1 - 3.5)^2 * 0.125/0.5) \\ &\quad + ((1 - 3.5)^2 * 0.3125/0.5) \\ &= 0 + 0.7812 + 1.5625 + 3.9052\end{aligned}$$

$$\sigma_1^2(4) = 6.25$$

$$\begin{aligned}\sigma_1^2(5) &= ((1 - 3.9089)^2 * 0/0.6875) + ((1 - 3.9089)^2 * 0.0625/0.6875) + ((1 - 3.9089)^2 * 0.125/0.6875) \\ &\quad + ((1 - 3.9089)^2 * 0.3125/0.6875) + ((1 - 3.9089)^2 * 0.1875/0.6875) \\ &= 0 + 0.7692 + 1.5385 + 3.8462 + 2.3077\end{aligned}$$

$$\sigma_1^2(5) = 8.4616$$

$$\begin{aligned}\sigma_1^2(6) &= ((1 - 4.2307)^2 * 0/0.8125) + ((1 - 4.2307)^2 * 0.0625/0.8125) + ((1 - 4.2307)^2 * 0.125/0.8125) \\ &\quad + ((1 - 4.2307)^2 * 0.3125/0.8125) + ((1 - 4.2307)^2 * 0.1875/0.8125) + ((1 - 4.2307)^2 * 0.125/0.8125) \\ &= 0 + 0.8029 + 1.6057 + 4.0144 + 2.4086 + 1.6057\end{aligned}$$

$$\sigma_1^2(6) = 10.4373$$

$$\begin{aligned}\sigma_1^2(7) &= ((1 - 4.75)^2 * 0/1) + ((1 - 4.75)^2 * 0.0625/1) + ((1 - 4.75)^2 * 0.125/1) + \\ &\quad ((1 - 4.75)^2 * 0.3125/1) + ((1 - 4.75)^2 * 0.1875/1) + ((1 - 4.75)^2 * 0.125/1) + ((1 - 4.75)^2 * 0.1875/1) \\ &= 0 + 0.8789 + 1.7578 + 4.3945 + 2.6367 + 1.7578 + 2.6367\end{aligned}$$

$$\sigma_1^2(7) = 14.0624$$

$$\begin{aligned}\sigma_1^2(8) &= ((1 - 4.75)^2 * 0/1) + ((1 - 4.75)^2 * 0.0625/1) + ((1 - 4.75)^2 * 0.125/1) + \\ &\quad ((1 - 4.75)^2 * 0.3125/1) + ((1 - 4.75)^2 * 0.1875/1) + ((1 - 4.75)^2 * 0.125/1) + ((1 - 4.75)^2 * 0.1875/1) + ((1 - 4.75)^2 * 0/1) \\ &= 0 + 0.8789 + 1.7578 + 4.3945 + 2.6367 + 1.7578 + 2.6367 + 0\end{aligned}$$

$$\sigma_1^2(8) = 14.0624$$

$$\begin{aligned}\sigma_2^2(1) &= ((1 - 0)^2 * 0.0625/1) + ((1 - 0)^2 * 0.125/1) + ((1 - 0)^2 * 0.3125/1) + ((1 - 0)^2 * 0.1875/1) \\ &\quad + ((1 - 0)^2 * 0.125/1) + ((1 - 0)^2 * 0.1875/1) + ((1 - 0)^2 * 0/1) \\ &= 1\end{aligned}$$

$$\begin{aligned}\sigma_2^2(2) &= ((1 - 0.13333)^2 * 0.125/0.9375) + ((1 - 0.13333)^2 * 0.3125/0.9375) + ((1 - 0.13333)^2 * 0.1875/0.9375) + ((1 - 0.13333)^2 * 0.125/0.9375) + ((1 - 0.13333)^2 * 0.1875/0.9375) + ((1 - 0.13333)^2 * 0/0.9375) \\ &= 0.7511\end{aligned}$$

$$\begin{aligned}\sigma_2^2(3) &= ((1 - 0.61538)^2 * 0.3125/0.8125) + ((1 - 0.61538)^2 * 0.1875/0.8125) + ((1 - 0.61538)^2 * 0.125/0.8125) + ((1 - 0.61538)^2 * 0.1875/0.8125) + ((1 - 0.61538)^2 * 0/0.8125) \\ &= 0.1479\end{aligned}$$

$$\begin{aligned}\sigma_2^2(4) &= ((1 - 3.5)^2 * 0.1875/0.5) + ((1 - 3.5)^2 * 0.125/0.5) + ((1 - 3.5)^2 * 0.1875/0.5) + ((1 - 3.5)^2 * 0/0.5) \\ &= 6.25\end{aligned}$$

$$\begin{aligned}\sigma_2^2(5) &= ((1 - 8.6)^2 * 0.125/0.3125) + ((1 - 8.6)^2 * 0.1875/0.3125) + ((1 - 8.6)^2 * 0/0.3125) \\ &= 57.760\end{aligned}$$

$$\begin{aligned}\sigma_2^2(6) &= ((1 - 18.333)^2 * 0.1875/0.1875) + ((1 - 18.333)^2 * 0/0.1875) \\ &= 300.44\end{aligned}$$

$$\sigma_2^2(7) = 0$$

$$\sigma_2^2(8) = 0$$

Menghitung rerata total dengan menggunakan Persamaan (2-15)

$$\begin{aligned}m_T &= (1 * 0) + (2 * 0.0625) + (3 * 0.125) + (4 * 0.3125) + (5 * 0.1875) + (6 * 0.125) + (7 * 0.1875) + (8 * 0) \\ &= 0 + 0.125 + 0.375 + 1.25 + 0.9375 + 0.75 + 1.3125 + 0\end{aligned}$$

$$m_T = 4.75$$

Menghitung BCV dengan menggunakan Persamaan (2-14)

$$\begin{aligned}\sigma_B^2(1) &= (0 * (0 - 4.75)^2) + (1 * (4.75 - 0)^2) \\ &= 22.562\end{aligned}$$

$$\begin{aligned}\sigma_B^2(2) &= (0.0625 * (2 - 4.75)^2) + (0.9375 * (4.9333 - 0.1333)^2) \\ &= 20.454\end{aligned}$$

$$\begin{aligned}\sigma_B^2(3) &= (0.1875 * (2.6667 - 4.75)^2) + (0.8125 * (5.2308 - 0.61538)^2) \\ &= 14.704\end{aligned}$$

$$\sigma_B^2(4) = (0.5 * (3.5 - 4.75)^2) + (0.5 * (6 - 3.5)^2)$$

$$\begin{aligned}
&= 1.5625 \\
\sigma_B^2(5) &= (0.6875 * (3.9089 - 4.75)^2) + (0.3125 * (6.6 - 8.6)^2) \\
&= 5.1182 \\
\sigma_B^2(6) &= (0.8125 * (4.2307 - 4.75)^2) + (0.1875 * (7 - 18.33)^2) \\
&= 34.814 \\
\sigma_B^2(7) &= (1 * (4.75 - 4.75)^2) + (0 * (0 - 0)^2) \\
&= 0 \\
\sigma_B^2(8) &= (1 * (4.75 - 4.75)^2) + (0 * (0 - 0)^2) \\
&= 0
\end{aligned}$$

Nilai *threshold* dapat diperoleh dengan memaksimumkan nilai BCV. Dalam contoh ini didapatkan nilai *threshold* sebesar 5. Sehingga didapatkan matriks hasil seperti pada Gambar 3.11.

$A =$

0	0	0	1
0	1	1	0
0	1	0	0
1	0	0	0

Gambar 3.11 Matriks Hasil *Otsu*

Untuk mengubah citra hasil segmentasi menjadi *grayscale* dilakukan perkalian matriks hasil dengan *grayscale* matriks awal.

2. Deteksi Tepi Canny

Berikut merupakan contoh perhitungan deteksi tepi canny

a. Filter *Gaussian*

Untuk menghitung atau menentukan nilai-nilai dalam setiap elemen dalam filter penghalus *Gaussian* yang akan dibentuk seperti Persamaan (2-1). Dalam hal ini σ merupakan standar deviasi dan piksel pada pusat (x, y) dengan bobot terbesar berupa 1. Sebagai contoh, bobotnya dapat diperoleh dengan contoh sebagai berikut:

$$(0,0) = e^{-0} = 1$$

$$(1,0) = G(0,1) = G(-1,0) = G(0,-1) = e^{-1/2} = 0,6065$$

$$(1,1) = G(1,-1) = G(-1,1) = G(-1,-1) = e^{-1} = 0,3679$$

$$(2,1) = G(1,2) = G(-2,1) = G(-2,-1) = e^{-5/2} = 0,0821$$

$$(2,0) = G(0,2) = G(0,-2) = G(-2,0) = e^{-2} = 0,1353$$

$$(2,2) = G(-2,-2) = G(-2,2) = G(2,-2) = e^{-4} = 0,0183$$

Langkah selanjutnya adalah melakukan proses *filtering* dengan menerapkan operasi konvolusi sebagai berikut:

1	0	1	1
0	2	3	1
1	2	1	0
1	0	0	1

Gambar 3.12 Contoh Penerapan Filter *Gaussian*

b. Menghitung Gradien

Menghitung gradien gambar yang telah diperhalus dengan menggunakan operator *sobel*. Untuk melakukan proses perhitungan gradien dengan menerapkan proses konvolusi dengan persamaan yang akan dibentuk seperti Persamaan (2-2).

1	0	1	1
0	2	3	1
1	2	1	0
1	0	0	1

 $kernel\ gx =$

-1	0	1
-2	0	2
-1	0	1

Gambar 3.13 Contoh Matriks dengan *Kernel gx*

1	0	1	1
0	6	-3	1
1	2	-4	0
1	0	0	1

Gambar 3.14 Hasil Konvolusi *Kernel gx*

1	0	1	1
0	2	3	1
1	2	1	0
1	0	0	1

 $kernel\ gy =$

1	2	1
0	0	0
-1	-2	-1

Gambar 3.15 Contoh Matriks dengan *Kernel gy*

1	0	1	1
0	-4	-1	1
1	6	8	0
1	0	0	1

Gambar 3.16 Hasil Konvolusi *Kernel* gy

c. Menghitung Sudut

Pada proses ini yaitu menghitung arah tepian yang ditentukan dengan menggunakan persamaan yang telah dibentuk pada Persamaan (2-3). Dengan hasil perhitungan arah tepi seperti pada Gambar 3.17.

45	45	45	45
45	34	18	45
45	72	63	45
45	45	45	45

Gambar 3.17 Hasil Perhitungan Matriks Arah Tepi

Setelah arah tepi diperoleh, maka perlu menghubungkan antara arah tepi dengan sebuah arah yang data dilacak dari citra, dengan pembagian arah yang berkisar antara:

$$0^0 = 0 - 22.5 \text{ dan } 157.5 - 180$$

$$45^0 = 22.5 - 67.5$$

$$90^0 = 67.5 - 112.5$$

$$135^0 = 112.5 - 157.5$$

45	45	45	45
45	45	0	45
45	90	45	45
45	45	45	45

Gambar 3.18 Hasil Matriks Gradien

d. *Non Maximus Suppression*

Pada proses ini, dilakukan penghilangan nilai-nilai yang tidak dianggap tepi dengan membandingkan antara nilai suatu piksel dengan nilai piksel tetangganya. Apabila nilainya lebih besar maka dipertahankan, apabila

lebih kecil maka diubah menjadi nol dengan hasil matriks seperti pada Gambar 3.19 berikut.

45	45	45	45
45	45	0	45
45	90	45	45
45	45	45	45

Gambar 3.19 Hasil Matriks *Non Maximus Suppression*

e. *Hysteresis Thresholding*

Pada tahap ini diterapkan dua buah *thresholding* yaitu *low thresholding* (T1) dan *high thresholding* (T2). Nilai yang kurang dari T1 akan diubah menjadi hitam (nilai 0) dan nilai yang lebih dari T2 diubah menjadi putih (nilai 1). Misalkan $T1 = 50$ dan $T2 = 80$. Sehingga didapatkan matriks hasil seperti pada Gambar 3.20.

0	0	0	0
0	0	0	0
0	1	0	0
0	0	0	0

Gambar 3.20 Hasil Matriks *Hysteresis Thresholding*

3. Merubah Matriks Hasil Menjadi *Grayscale*

Untuk mengubah matriks hasil segmentasi menjadi *grayscale*, dilakukan perkalian matriks hasil segmentasi dengan *grayscale* matriks awal. Seperti pada Gambar 3.21 dan Gambar 3.22.

1	2	4	6
3	5	6	4
3	6	3	3
5	4	3	2

0	0	0	1
0	1	1	0
0	1	0	0
1	0	0	0

Gambar 3.21 Contoh Perkalian Matriks

0	0	0	6
0	5	6	0
0	6	0	0
5	0	0	0

Gambar 3.22 Hasil Kali Matriks

3.3.3 Ekstraksi Fitur

1. Ekstraksi Fitur *Gray Level Co-Occurrence Matrix* (GLCM)

Proses ekstraksi fitur GLCM terdiri dari dua tahap yaitu pembentukan matriks GLCM dan perhitungan nilai fitur GLCM.

a. Pembentukan Matriks GLCM

Misalkan terdapat citra dengan matriks seperti Gambar 3.23.

A=

0	3	1	0
0	2	0	2
0	1	1	3
0	0	3	0

Gambar 3.23 Contoh Matriks GLCM

Langkah pertama yaitu membentuk matriks ordo $n \times n$, dimana n merupakan nilai elemen terbesar dari matriks A. Dalam hal ini nilai terbesar pada matriks A adalah 3, maka dibentuk matriks ordo 3×3 dan dimulai dari index 0 seperti pada Gambar 3.24.

	0	1	2	3
0				
1				
2				
3				

Gambar 3.24 Pembentukan Matriks Ordo $n \times n$

Selanjutnya mengecek matriks ketetanggaan sesuai dengan komposisi piksel seperti pada Gambar 3.25. Dan berdasarkan komposisi piksel tersebut, dilakukan perhitungan terhadap jumlah komposisi piksel yang memenuhi hubungan seperti pada Gambar 3.26.

	0	1	2	3
0	0,0	0,1	0,2	0,3
1	1,0	1,1	1,2	1,3
2	2,0	2,1	2,2	2,3
3	3,0	3,1	3,2	3,3

Gambar 3.25 Komposisi Piksel

	0	1	2	3
0	2	2	3	3
1	2	2	0	2
2	3	0	0	0
3	3	2	0	0

Gambar 3.26 Jumlah Pasangan Piksel

Kemudian setelah itu, dilakukan normalisasi agar jumlah seluruh elemennya sama dengan satu. Normalisasi dilakukan dengan cara menjumlahkan seluruh nilai elemen pada matriks dan digunakan sebagai pembagi seperti yang terlihat pada Gambar 3.27. Hasil dari normalisasi dapat dilihat dari Gambar 3.28.

	2	2	3	3
	2	2	0	2
	3	0	0	0
	3	2	0	0

Gambar 3.27 Normalisasi Matriks

0.083	0.083	0.125	0.125
0.083	0.083	0	0.083
0.125	0	0	0
0.125	0.083	0	0

Gambar 3.28 Hasil Normalisasi Matriks

Hal yang sama dilakukan untuk membuat matriks GLCM dengan sudut 45° , 90° , dan 135° namun dengan arah yang berbeda. Sudut 45° derajat dibuat dengan arah ketetanggaan diagonal-kanan, sudut 90° derajat dibuat

dengan arah ketetanggaan vertikal, dan sudut 135 derajat dibuat dengan arah ketetanggaan diagonal-kiri yang dapat dilihat pada Gambar 3.29.

0	3	1	0
0	2	0	2
0	1	1	3
0	0	3	0

(a)

0	3	1	0
0	2	0	2
0	1	1	3
0	0	3	0

(b)

0	3	1	0
0	2	0	2
0	1	1	3
0	0	3	0

(c)

Gambar 3.29 Arah Sudut Matriks GLCM (a) 45°, (b) 90°, (c) 135°

Setelah mendapatkan matriks yang telah di normalisasi, tahap selanjutnya yaitu melakukan perhitungan nilai fitur GLCM.

b. Perhitungan Nilai Matriks

Matriks GLCM yang telah di normalisasi pada tahap sebelumnya akan digunakan untuk menghitung 5 fitur GLCM yang terdiri dari *angular second moment (ASM)*, *contrast*, *inverse different moment (IDM)*, *entropy*, dan *correlation*.

1) Angular Second Moment (ASM)

Untuk menghitung ASM digunakan Persamaan (2-16)

$$ASM = (0.083)^2 + (0.083)^2 + (0.125)^2 + (0.125)^2 + (0.083)^2 + (0.083)^2 + (0)^2 + (0.083)^2 + (0.125)^2 + (0)^2 + (0)^2 + (0)^2 + (0.125)^2 + (0.083)^2 + (0)^2 + (0)^2$$

$$ASM = 0.104$$

2) Contrast

Untuk menghitung *contrast* digunakan Persamaan (2.17)

$$Contrast = ((0 - 0)^2 * 0.083) + ((0 - 1)^2 * 0.083) + ((0 - 2)^2 * 0.125) + ((0 - 3)^2 * 0.125) + ((1 - 0)^2 * 0.083) + ((1 - 1)^2 * 0.083) + ((1 - 2)^2 * 0) + ((1 - 3)^2 * 0.083) + ((2 - 0)^2 * 0.125) + ((2 - 1)^2 * 0) + ((2 - 2)^2 * 0) + ((2 - 3)^2 * 0) + ((3 - 0)^2 * 0.125) + ((3 - 1)^2 * 0.083) + ((3 - 2)^2 * 0) + ((3 - 3)^2 * 0)$$

$$Contrast = 4.08$$

3) *Inverse Different Moment (IDM)*

Untuk menghitung IDM digunakan Persamaan (2.18)

$$\begin{aligned} \text{IDM} = & \left(\frac{0.083}{1+(0-0)^2} \right) + \left(\frac{0.083}{1+(0-1)^2} \right) + \left(\frac{0.125}{1+(0-2)^2} \right) + \left(\frac{0.125}{1+(0-3)^2} \right) + \\ & \left(\frac{0.083}{1+(1-0)^2} \right) + \left(\frac{0.083}{1+(1-1)^2} \right) + \left(\frac{0}{1+(1-2)^2} \right) + \left(\frac{0.083}{1+(1-3)^2} \right) + \left(\frac{0.125}{1+(2-0)^2} \right) + \\ & \left(\frac{0}{1+(2-1)^2} \right) + \left(\frac{0}{1+(2-2)^2} \right) + \left(\frac{0}{1+(2-3)^2} \right) + \left(\frac{0.125}{1+(3-0)^2} \right) + \left(\frac{0.083}{1+(3-1)^2} \right) + \\ & \left(\frac{0}{1+(3-2)^2} \right) + \left(\frac{0}{1+(3-3)^2} \right) \\ \text{IDM} = & 0.359 \end{aligned}$$

4) *Entropy*

Untuk menghitung *entropy* dapat digunakan Persamaan (2.19)

$$\begin{aligned} \text{Entropy} = & (-0.083 * \log(0.083)) + (-0.083 * \log(0.083)) + (-0.125 * \\ & \log(0.125)) + (-0.125 * \log(0.125)) + (-0.083 * \log(0.083)) + (-0.083 * \\ & \log(0.083)) + (-0 * \log(0)) + (-0.083 * \log(0.083)) + (-0.125 * \log(0.125)) \\ & + (-0 * \log(0)) + (-0 * \log(0)) + (-0 * \log(0)) + (-0.125 * \log(0.125)) + (- \\ & 0.083 * \log(0.083)) + (-0 * \log(0)) + (-0 * \log(0)) \\ \text{Entropy} = & 0.99 \end{aligned}$$

5) *Correlation*

Untuk menghitung nilai *correlation* dapat digunakan Persamaan (2.20). Sebelum menghitung *correlation*, perlu dihitung nilai μ dengan persamaan (2-21) dan σ dengan Persamaan (2-22).

$$\begin{aligned} \mu = & (0 * 0.083) + (0 * 0.083) + (0 * 0.125) + (0 * 0.125) + (1 * 0.083) + (1 \\ & * 0.083) + (1 * 0) + (1 * 0.083) + (2 * 0.125) + (2 * 0) + (2 * 0) + (2 * 0) + \\ & (3 * 0.125) + (3 * 0.083) + (3 * 0) + (3 * 0) \end{aligned}$$

$$\mu = 1.123$$

$$\begin{aligned} \sigma = & (0.083 * (0 - 1.123)^2) + (0.083 * (0-1.123)^2) + (0.125 * (0 - 1.123)^2) + \\ & (0.125 * (0 - 1.123)^2) + (0.083 * (1 - 1.123)^2) + (0.083 * (1 - 1.123)^2) + (0 * \\ & (1 - 1.123)^2) + (0.083 * (1 - 1.123)^2) + (0.125 * (2 - 1.123)^2) + (0 * (2 - \\ & 1.123)^2) + (0 * (2 - 1.123)^2) + (0 * (2 - 1.123)^2) + (0.125 * (3 - 1.123)^2) + \\ & (0.083 * (3 - 1.123)^2) + (0 * (3 - 1.123)^2) + (0 * (3 - 1.123)^2) \end{aligned}$$

$$\sigma = 1.3574$$

$$\begin{aligned}
Correlation = & \left(0.083 \frac{(0-1.123)(0-1.123)}{1.3574^2}\right) + \left(0.083 \frac{(0-1.123)(1-1.123)}{1.3574^2}\right) + \\
& \left(0.125 \frac{(0-1.123)(2-1.123)}{1.3574^2}\right) + \left(0.125 \frac{(0-1.123)(3-1.123)}{1.3574^2}\right) + \\
& \left(0.083 \frac{(1-1.123)(0-1.123)}{1.3574^2}\right) + \left(0.083 \frac{(1-1.123)(1-1.123)}{1.3574^2}\right) + \left(0 \frac{(1-1.123)(2-1.123)}{1.3574^2}\right) + \\
& \left(0.083 \frac{(1-1.123)(3-1.123)}{1.3574^2}\right) + \left(0.125 \frac{(2-1.123)(0-1.123)}{1.3574^2}\right) + \left(0 \frac{(2-1.123)(1-1.123)}{1.3574^2}\right) + \\
& \left(0 \frac{(2-1.123)(2-1.123)}{1.3574^2}\right) + \left(0 \frac{(2-1.123)(3-1.123)}{1.3574^2}\right) + \left(0.125 \frac{(3-1.123)(0-1.123)}{1.3574^2}\right) + \\
& \left(0.083 \frac{(3-1.123)(1-1.123)}{1.3574^2}\right) + \left(0 \frac{(3-1.123)(2-1.123)}{1.3574^2}\right) + \left(0 \frac{(3-1.123)(3-1.123)}{1.3574^2}\right) \\
Correlation = & -0.3705
\end{aligned}$$

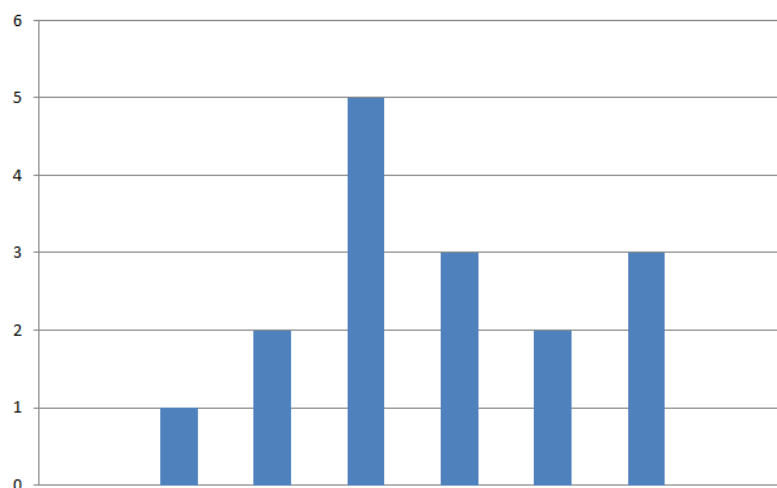
2. Ekstraksi Fitur Statistik

Berikut merupakan contoh perhitungan ekstraksi fitur statistik. Misalkan terdapat citra dengan matriks 4x4 seperti pada Gambar 3.30.

$$A = \begin{array}{|c|c|c|c|} \hline 1 & 2 & 4 & 6 \\ \hline 3 & 5 & 6 & 4 \\ \hline 3 & 6 & 3 & 3 \\ \hline 5 & 4 & 3 & 2 \\ \hline \end{array}$$

Gambar 3.30 Contoh Matriks 4x4

Langkah pertama yaitu mengecek histogram dari matriks A di atas. Dari matriks tersebut, didapatkan grafik histogram seperti pada Gambar 3.31.



Gambar 3.31 Grafik Histogram Matriks A

Setelah itu menghitung rerata intensitas berdasarkan grafik histogram di atas menggunakan Persamaan (2-23).

$$\begin{aligned}
 m &= (0 * 0) + (1 * 0.0625) + (2 * 0.125) + (3 * 0.3125) + (4 * 0.1875) + (5 * 0.125) + (6 * 0.1875) + (7 * 0) \\
 &= 0 + 0.0625 + 0.25 + 0.9375 + 0.75 + 0.625 + 1.125 + 0 \\
 m &= 3.75
 \end{aligned}$$

Menghitung deviasi standar dengan menggunakan Persamaan (2-24).

$$\begin{aligned}
 \sigma^2 &= ((0 - 3.75)^2 * 0) + ((1 - 3.75)^2 * 0.0625) + ((2 - 3.75)^2 * 0.125) + ((3 - 3.75)^2 * 0.3125) + ((4 - 3.75)^2 * 0.1875) + ((5 - 3.75)^2 * 0.125) + ((6 - 3.75)^2 * 0.1875) + ((7 - 3.75)^2 * 0) \\
 &= 0 + 0.4727 + 0.3828 + 0.1758 + 0.0117 + 0.1953 + 0.9492 + 0 \\
 \sigma^2 &= 2.1875 \text{ (varians)} \\
 \sigma &= \sqrt{2.1875} \\
 &= 1.4790
 \end{aligned}$$

Menghitung *skewness* dengan menggunakan Persamaan (2-25).

$$\begin{aligned}
 Skewness &= ((0 - 3.75)^3 * 0) + ((1 - 3.75)^3 * 0.0625) + ((2 - 3.75)^3 * 0.125) + ((3 - 3.75)^3 * 0.3125) + ((4 - 3.75)^3 * 0.1875) + ((5 - 3.75)^3 * 0.125) + ((6 - 3.75)^3 * 0.1875) + ((7 - 3.75)^3 * 0) \\
 &= 0 + (-1.2998) + (-0.6699) + (-0.1318) + 0.0029 + 0.2441 + 2.1357 + 0 \\
 &= 0.2812 / (8-1)^2 \\
 &= 0.0057
 \end{aligned}$$

Menghitung energi dengan menggunakan Persamaan (2-26).

$$\begin{aligned}
 Energi &= (0)^2 + (0.0625)^2 + (0.125)^2 + (0.3125)^2 + (0.1875)^2 + (0.125)^2 + (0.1875)^2 + (0)^2 \\
 &= 0 + 0.0039 + 0.0156 + 0.0976 + 0.0352 + 0.0156 + 0.0352 + 0 \\
 Energi &= 0.2031
 \end{aligned}$$

Menghitung *entropi* dengan menggunakan Persamaan (2-27).

$$\begin{aligned}
 Entropi &= - ((0 * \log_2(0)) + (0.0625 * \log_2(0.0625)) + (0.125 * \log_2(0.125)) + (0.3125 * \log_2(0.3125)) + (0.1875 * \log_2(0.1875)) + (0.125 * \log_2(0.125)) + (0.1875 * \log_2(0.1875)) + (0 * \log_2(0))) \\
 &= - (0 + -0.25 + -0.375 + -0.5244 + -0.4528 + -0.375 + -0.4528 + 0)
 \end{aligned}$$

$$Entropi = 2.43$$

Menghitung kehalusan (R) dengan menggunakan Persamaan (2-28). Sebelum menghitung kehalusan (R), varians perlu dinormalisasi.

$$R = 1 - \left(\frac{1}{1+0.04464} \right)$$

$$R = 0.042375$$

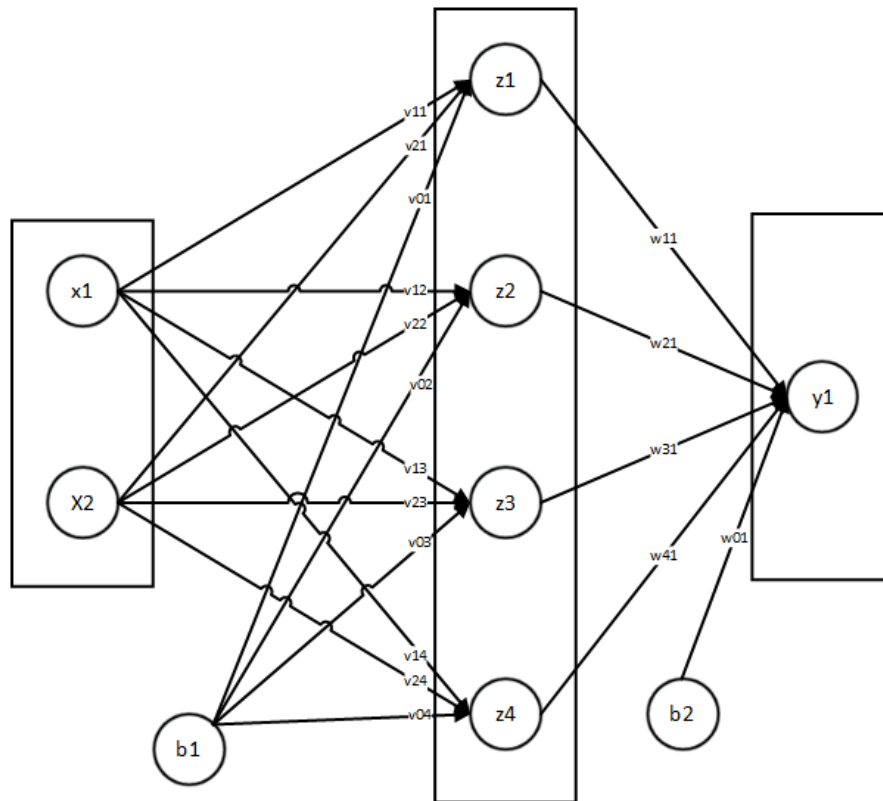
3.3.4 Klasifikasi

Pada penelitian ini digunakan metode *Artificial Neural Network* sebagai *classifier*. Berikut merupakan contoh perhitungan metode *backpropagation* pada pendeteksian penyakit glaukoma dengan data pelatihan *dummy* seperti pada Tabel 3.1. Data *dummy* tersebut merupakan data pelatihan yang terdiri dari 4 data yang terbagi menjadi ke dalam dua kelas berdasarkan target normal atau glaukoma, yaitu 1 (glaukoma), dan 0 (normal).

Tabel 3.1 Data *Dummy*

No.	Input 1	Input 2	Target
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

Arsitektur ANN yang akan digunakan pada contoh ini dapat dilihat pada Gambar 3.32.



Gambar 3.32 Arsitektur ANN

Arsitektur yang digunakan untuk contoh perhitungan ini adalah *multilayer* yang terdiri dari 1 *input layer*, 1 *hidden layer* dan 1 *output layer*. Pada contoh tersebut terdapat 2 neuron pada *input layer*, 4 neuron pada *hidden layer*, dan 1 neuron pada *output layer*.

Proses pelatihan data menggunakan metode ini memiliki beberapa ketentuan, berikut merupakan ketentuan-ketentuan untuk contoh penelitian menggunakan metode ini.

1. Batas *error* = 0.0001
2. Batas *epoch* = 2000
3. *Learning rate* = 0.01
4. Fungsi aktivasi yang digunakan adalah *sigmoid biner*.

Algoritma metode ini terdiri dari 10 langkah yang dibagi menjadi 3 fase yaitu

Langkah 0: Inisialisasi bobot awal dari *input* awal ke *hidden layer* pertama yang disajikan dalam Tabel 3.2 berikut.

Tabel 3.2 Bias dan Bobot Awal dari *Input Layer* ke *Hidden Layer*

Dari- Ke-	Bias (B1)	X1	X2
W1	0.28183	0.1377	0.0045
W2	0.7558	0.1031	0.0380
W3	0.61836	0.0317	0.1135
W4	0.2505	0.0964	0.0786

Bias dan bobot awal dari *input layer* ke *hidden layer* pertama dinyatakan dengan V_{01} sampai dengan V_{24} , seperti pada Tabel 3.2 di atas. Kemudian untuk bias dan bobot awal dari *hidden layer* ke *output layer* dinyatakan dengan W_{01} sampai dengan W_{41} seperti pada Tabel 3.3 berikut.

Tabel 3.3 Bias dan Bobot Awal dari *Hidden Layer* ke *Output Layer*

Dari- Ke-	Bias (B2)	W1	W2	W3	W4
Y	0.2097	0.0936	0.3335	0.1324	0.0187

Langkah 1: Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 sampai 9. Kondisi penghentian terpenuhi jika $error < 0.0001$ atau $epoch > 2000$.

Langkah 2: Untuk setiap data pelatihan, lakukan langkah 3 sampai 8.

Tahap I : Propagasi Maju

Langkah 3: Tiap unit masukan ($x_i, i = 1, 2, \dots, n$) menerima sinyal dan meneruskannya ke unit selanjutnya, yaitu lapisan tersembunyi. Yang digunakan pada contoh ini adalah data pertama dengan $X_1 = 1, X_2 = 0$.

Langkah 4: Hitung semua keluaran pada lapisan tersembunyi ($Z_j, j = 1, 2, \dots, p$) menggunakan Persamaan (2-29).

$$\begin{aligned} Z_{net1} &= 0.28183 + (1 * 0.1377) + (0 * 0.0045) \\ &= 0.41953 \end{aligned}$$

$$\begin{aligned} Z_{net2} &= 0.7558 + (1 * 0.1031) + (0 * 0.0380) \\ &= 0.8589 \end{aligned}$$

$$\begin{aligned} Z_{net3} &= 0.61836 + (1 * 0.0317) + (0 * 0.1135) \\ &= 0.65006 \end{aligned}$$

$$\begin{aligned} Z_{net4} &= 0.2505 + (1 * 0.0964) + (0 * 0.0786) \\ &= 0.3469 \end{aligned}$$

Untuk menghitung sinyal keluaran tiap neuron *hidden layer*, digunakan fungsi aktivasi *sigmoid biner* seperti pada Persamaan (2-30).

$$Z_1 = \frac{1}{1+e^{-0.41953}} = 0.603372$$

$$Z_2 = \frac{1}{1+e^{-0.8589}} = 0.702412$$

$$Z_3 = \frac{1}{1+e^{-0.65006}} = 0.657009$$

$$Z_4 = \frac{1}{1+e^{-0.3469}} = 0.585857$$

Langkah 5: Hitung semua keluaran jaringan di lapisan *output* ($Y_k, k = 1, 2, \dots, m$) menggunakan Persamaan (2-31).

$$\begin{aligned} Y_{net1} &= 0.2097 + (0.603372 * 0.0936) + (0.702412 * 0.3335) + (0.657009 * \\ &\quad 0.1324) + (0.585857 * 0.0187) \\ &= 0.598374 \end{aligned}$$

Untuk menghitung sinyal *output*-nya digunakan fungsi aktivasi *sigmoid biner* seperti pada Persamaan (2-32).

$$Y_1 = \frac{1}{1+e^{-0.598374}} = 0.645269$$

Tahap II : Propagasi mundur

Langkah 6: Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran ($y_k, k = 1, 2, \dots, m$) menggunakan Persamaan (2-33).

$$\begin{aligned} \delta_1 &= (t_1 - y_1) * f'(y_{net1}) \\ &= (1 - 0.645269) * f'(0.598374) \\ &= 0.081197 \end{aligned}$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki W_{jk}) dengan *learning rate* (α), menggunakan Persamaan (2-34).

$$\Delta W_{11} = 0.01 * 0.081197 * 0.603372 = 0.000489920$$

$$\Delta W_{21} = 0.01 * 0.081197 * 0.702412 = 0.000570337$$

$$\Delta W_{31} = 0.01 * 0.081197 * 0.657009 = 0.000533472$$

$$\Delta W_{41} = 0.01 * 0.081197 * 0.585857 = 0.000475698$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai w_{0k}) menggunakan Persamaan (2-34).

$$\Delta W_{01} = 0.01 * 0.081197 = 0.00081197$$

Langkah 7: Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi ($Z_j, j = 1, 2, \dots, p$) menggunakan Persamaan (2-35).

$$\delta_{net_1} = 0.081197 * 0.0936 = 0.007600039$$

$$\delta_{net_2} = 0.081197 * 0.3335 = 0.027079200$$

$$\delta_{net_3} = 0.081197 * 0.1324 = 0.010750483$$

$$\delta_{net_4} = 0.081197 * 0.0187 = 0.001518384$$

Faktor δ unit tersembunyi *hidden layer* pertama dihitung menggunakan Persamaan (2-36):

$$\begin{aligned}\delta_1 &= 0.007600039 * f'(0.41953) \\ &= 0.239314\end{aligned}$$

$$\begin{aligned}\delta_2 &= 0.027079200 * f'(0.8589) \\ &= 0.209029\end{aligned}$$

$$\begin{aligned}\delta_3 &= 0.010750483 * f'(0.65006) \\ &= 0.225348\end{aligned}$$

$$\begin{aligned}\delta_4 &= 0.001518384 * f'(0.3469) \\ &= 0.242628\end{aligned}$$

Hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}) dari *input layer* ke *hidden layer* pertama menggunakan Persamaan (2-37)

$$\Delta V_{11} = 0.01 * 0.239314 * 1 = 0.00239314$$

$$\Delta V_{12} = 0.01 * 0.209029 * 1 = 0.00209029$$

$$\Delta V_{13} = 0.01 * 0.225348 * 1 = 0.00225348$$

$$\Delta V_{14} = 0.01 * 0.242628 * 1 = 0.00242628$$

$$\Delta V_{21} = 0.01 * 0.239314 * 0 = 0$$

$$\Delta V_{22} = 0.01 * 0.209029 * 0 = 0$$

$$\Delta V_{23} = 0.01 * 0.225348 * 0 = 0$$

$$\Delta V_{24} = 0.01 * 0.242628 * 0 = 0$$

Menghitung koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai V_{0j}) menggunakan Persamaan (2-37).

$$\Delta V_{01} = 0.01 * 0.239314 = 0.00239314$$

$$\Delta V_{02} = 0.01 * 0.209029 = 0.00209029$$

$$\Delta V_{03} = 0.01 * 0.225348 = 0.00225348$$

$$\Delta V_{04} = 0.01 * 0.242628 = 0.00242628$$

Tahap III : Perubahan bobot

Langkah 8: Tiap-tiap unit *output* ($k = 1, 2, \dots, m$) memperbaiki bias dan bobotnya ($j = 0, 1, 2, \dots, p$) menggunakan Persamaan (2-38).

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk}$$

$$W_{01}(\text{baru}) = 0.2097 + 0.00081197 = 0.21051197$$

$$W_{11}(\text{baru}) = 0.0936 + 0.000489920 = 0.094089920$$

$$W_{21}(\text{baru}) = 0.3335 + 0.000570337 = 0.334070337$$

$$W_{31}(\text{baru}) = 0.1324 + 0.000533472 = 0.132933472$$

$$W_{41}(\text{baru}) = 0.0187 + 0.000475698 = 0.019175698$$

Tiap-tiap unit tersembunyi ($Z_j, j = 1, 2, 3, \dots, p$) memperbaiki bias dan bobotnya ($j = 0, 1, 2, 3, \dots, n$) menggunakan Persamaan (2-39).

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij}$$

$$V_{01}(\text{baru}) = 0.28183 + 0.00239314 = 0.28422314$$

$$V_{11}(\text{baru}) = 0.1377 + 0.00239314 = 0.14009314$$

$$V_{12}(\text{baru}) = 0.0045 + 0.00209029 = 0.00659029$$

Demikian juga untuk $V_{13}(\text{baru})$ sampai $V_{24}(\text{baru})$ dihitung dengan cara yang sama untuk memperoleh semua $V_{ij}(\text{baru})$

Langkah 9: Kondisi pelatihan berhenti, jika $\text{error} \leq 0.0001$ atau jumlah *epoch* mencapai 2000.

Software Jupyterlab digunakan untuk pembuatan jaringan *backpropagation* yang sesuai dengan inisialisasi awal sehingga diperoleh bias dan bobot akhir yang dapat dilihat pada Tabel 3.4 dan Tabel 3.5.

Tabel 3.4 Bias dan Bobot Akhir dari *Input Layer* ke *Hidden Layer*

Dari- Ke-	Bias (B1)	X1	X2
W1	0.4162	-0.1872	-0.1152
W2	2.3763	-1.6563	-1.4001
W3	0.3468	0.9591	-0.6164
W4	0.4129	-0.202	-0.1227

Tabel 3.5 Bias dan Bobot Akhir dari *Hidden Layer* ke *Output Layer*

Dari- Ke-	Bias (B2)	W1	W2	W3	W4
Y	1.5028	0.5573	-2.1845	0.6774	0.5430

Dari hasil yang sudah didapatkan sebelumnya, nilai akan dibulatkan ke *integer* terdekat (*threshold*) dan menghasilkan Tabel 3.6.

Tabel 3.6 *Output Data Hasil*

Data ke-	<i>Output</i>	Hasil <i>Threshold</i>	Target
1	0.01421309	0	0
2	0.9953228	1	1
3	0.9952612	1	1
4	0.99730814	1	1

Untuk mendapatkan tingkat akurasi dari hasil pelatihan, jumlah hasil *threshold* yang sesuai target dibagi dengan jumlah data. Tingkat akurasi = $(4/4) * 100 \% = 100 \%$.

3.3.5 Performa Pengujian Klasifikasi

Pada penelitian ini digunakan dua metode untuk mencari performa pengujian klasifikasi, dimana *dataset* RIM-ONE r3 menggunakan metode *k-fold cross validation* dikarenakan jumlah *dataset* yang sedikit, sedangkan *dataset* RIM-ONE r2 menggunakan metode *shuffle sampling* dengan perbandingan 70:30, 80:20, dan 90:10, karena jumlah *dataset* yang lebih banyak.

1. *K-fold Cross Validation*

Nilai *k* yang digunakan adalah 5, dengan jumlah *dataset* berupa 85 data citra fundus mata normal dan 74 data citra fundus mata diduga dan positif glaukoma, sehingga didapatkan pembagian *dataset* seperti pada Tabel 3.7 sampai Tabel 3.11 dimana N adalah citra fundus mata normal dan G adalah citra fundus mata diduga dan positif glaukoma.

Tabel 3.7 *Dataset* pada fold-1

Testing	Training	Training	Training	Training
17 N	17 N	17 N	17 N	17 N
15 G	15 G	15 G	15 G	14 G

Tabel 3.8 *Dataset* pada fold-2

Training	Testing	Training	Training	Training
17 N	17 N	17 N	17 N	17 N
15 G	15 G	15 G	15 G	14 G

Tabel 3.9 *Dataset* pada fold-3

Training	Training	Testing	Training	Training
17 N	17 N	17 N	17 N	17 N
15 G	15 G	15 G	15 G	14 G

Tabel 3.10 *Dataset* pada fold-4

Training	Training	Training	Testing	Training
17 N	17 N	17 N	17 N	17 N
15 G	15 G	15 G	15 G	14 G

Tabel 3.11 *Dataset* pada fold-5

Training	Training	Training	Training	Testing
17 N	17 N	17 N	17 N	17 N
15 G	15 G	15 G	15 G	14 G

3. *Shuffled Sampling*

Pembagian data pada *dataset* RIM-ONE r2 ini dibagi menjadi tiga bagian, yaitu berupa 70% data latih dan 30% data uji, 80% data latih dan 20% data uji, dan 90% data latih dan 10% data uji dengan *dataset* berupa 255 data citra fundus mata normal dan 200 data citra fundus mata diduga dan positif glaukoma. Sehingga didapatkan pembagian *dataset* seperti pada Tabel 3.12.

Tabel 3.12 Pengujian dengan Metode *Shuffle Sampling*.

<i>Dataset</i>	Training	Testing
70 : 30	179 N 140 G	76 N 60 G
80 : 20	204 N 160 G	51 N 40 G
90 : 10	230 N 180 G	25 N 20 G

3.3.6 Evaluasi Performa Klasifikasi

Pada penelitian ini ada dua performa klasifikasi yang dilakukan yaitu performa klasifikasi pada data *k-fold cross validation* dan pada pembagian data

shuffle sampling. Pengukuran performa klasifikasi terdiri dari akurasi, sensitivitas, dan spesifisitas yang dapat dihitung menggunakan Persamaan (2-40), (2-41) dan Persamaan (2-42). Contoh hasil pengujian *k-fold cross validation* per-fold dapat dilihat pada Tabel 3.13.

Tabel 3.13 Contoh Pengujian *Cross Validation*

Pengujian pada	TP	FP	TN	FN
<i>Fold-1</i>	14	1	16	1
<i>Fold-2</i>	14	1	14	3
<i>Fold-3</i>	13	2	16	1
<i>Fold-4</i>	13	2	16	1
<i>Fold-5</i>	13	1	15	2
Total	67	7	77	8

Setelah dilakukan perhitungan per-fold, jumlah total dari hasil prediksinya digunakan untuk mengukur akurasi, sensitivitas, dan spesifisitas, sehingga didapatkan hasil sebagai berikut.

TP (glaukoma dideteksi glaukoma) = 67

FP (glaukoma dideteksi normal) = 7

TN (normal dideteksi normal) = 77

FN (normal dideteksi glaukoma) = 8

Akurasi = $(67 + 77) / (67 + 7 + 77 + 8)$
= 91%

Sensitivitas = $67 / (67 + 8)$
= 89%

Spesifisitas = $77 / (77 + 7)$
= 92%

Kemudian contoh perhitungan evaluasi performa klasifikasi untuk metode *shuffle sampling* adalah sebagai berikut

Dataset dengan perbandingan 70:30, dimana 70% * 200 (total citra fundus glaukoma) dan 70% * 255 (total citra fundus normal) digunakan sebagai data latih, dan 30% * 200 (total citra fundus glaukoma) dan 30% * 255 (total citra

fundus normal) digunakan sebagai data uji sehingga didapatkan perhitungan sebagai berikut.

$$\text{TP (glaukoma dideteksi glaukoma)} = 56$$

$$\text{FP (glaukoma dideteksi normal)} = 4$$

$$\text{TN (normal dideteksi normal)} = 72$$

$$\text{FN (normal dideteksi glaukoma)} = 4$$

$$\begin{aligned}\text{Akurasi} &= (56 + 72) / (56 + 4 + 72 + 4) \\ &= 94\%\end{aligned}$$

$$\begin{aligned}\text{Sensitivitas} &= 56 / (56 + 4) \\ &= 93\%\end{aligned}$$

$$\begin{aligned}\text{Spesifisitas} &= 72 / (72 + 4) \\ &= 95\%\end{aligned}$$

Dataset dengan perbandingan 80:20, dimana 80% * 200 (total citra fundus glaukoma) dan 80% * 255 (total citra fundus normal) digunakan sebagai data latih, dan 20% * 200 (total citra fundus glaukoma) dan 20% * 255 (total citra fundus normal) digunakan sebagai data uji sehingga didapatkan perhitungan sebagai berikut.

$$\text{TP (glaukoma dideteksi glaukoma)} = 36$$

$$\text{FP (glaukoma dideteksi normal)} = 4$$

$$\text{TN (normal dideteksi normal)} = 48$$

$$\text{FN (normal dideteksi glaukoma)} = 3$$

$$\begin{aligned}\text{Akurasi} &= (36 + 48) / (36 + 4 + 48 + 3) \\ &= 92\%\end{aligned}$$

$$\begin{aligned}\text{Sensitivitas} &= 36 / (36 + 3) \\ &= 92\%\end{aligned}$$

$$\begin{aligned}\text{Spesifisitas} &= 48 / (48 + 4) \\ &= 92\%\end{aligned}$$

Dataset dengan perbandingan 90:10, dimana 90% * 200 (total citra fundus glaukoma) dan 90% * 255 (total citra fundus normal) digunakan sebagai data latih, dan 10% * 200 (total citra fundus glaukoma) dan 10% * 255 (total citra

fundus normal) digunakan sebagai data uji sehingga didapatkan perhitungan sebagai berikut.

$$\text{TP (glaukoma dideteksi glaukoma)} = 17$$

$$\text{FP (glaukoma dideteksi normal)} = 3$$

$$\text{TN (normal dideteksi normal)} = 22$$

$$\text{FN (normal dideteksi glaukoma)} = 3$$

$$\begin{aligned} \text{Akurasi} &= (17 + 22) / (17 + 3 + 22 + 3) \\ &= 87\% \end{aligned}$$

$$\begin{aligned} \text{Sensitivitas} &= 17 / (17 + 3) \\ &= 85\% \end{aligned}$$

$$\begin{aligned} \text{Spesifisitas} &= 22 / (22 + 3) \\ &= 88\% \end{aligned}$$

Dari perhitungan di atas didapatkan hasil seperti pada Tabel 3.14.

Tabel 3.14 Hasil Evaluasi Performa Klasifikasi

	TP	FP	TN	FN	Akurasi	Sensitivitas	Spesifisitas
<i>K-fold cross validation</i>	67	7	77	8	0.91	0.89	0.92
70 : 30	56	4	72	4	0.94	0.93	0.95
80 : 20	36	4	48	3	0.92	0.92	0.92
90 : 10	17	3	22	3	0.87	0.85	0.88

3.4 Jadwal Kegiatan

Estimasi waktu yang digunakan dalam pelaksanaan penelitian perbandingan metode segmentasi ini yaitu selama kurang lebih 5 bulan. Jadwal kegiatan penelitian dapat dilihat pada Tabel 3.15.

Tabel 3.15 Jadwal Kegiatan Penelitian

No.	Kegiatan	Waktu(Bulan)						Keterangan
		I	II	III	IV	V	VI	
1	Studi Literatur							Studi literatur yang berkaitan dengan topik penelitian
2	Pengumpulan dan Analisis Data							Pengumpulan dan analisis data penelitian
3	Perancangan Sistem							Perancangan Sistem yang akan dibuat
4	Pengujian							Pengujian sistem yang telah dibuat
5	Dokumentasi							Dokumentasi penelitian

DAFTAR PUSTAKA

- [1] K. Thakkar, K. Chauhan, A. Sudhalkar, R. Gulati, and M. S. Ophthalmologist, "Detection of Glaucoma from Retinal Fundus Images by analysing ISNT Measurement and features of Optic Cup and Blood Vessels," *Int. J. Eng. Technol. Sci. Res. IJETSR* www.ijetsr.com ISSN, vol. 4, no. 7, pp. 2394–3386, 2017.
- [2] A. A. Salam, T. Khalil, M. U. Akram, A. Jameel, and I. Basit, "Automated detection of glaucoma using structural and non structural features," *Springerplus*, vol. 5, no. 1, 2016.
- [3] E. Deepika and S. Maheswari, "Earlier glaucoma detection using blood vessel segmentation and classification," *Proc. 2nd Int. Conf. Inven. Syst. Control. ICISC 2018*, no. Icisc, pp. 484–490, 2018.
- [4] S. Maheshwari, R. B. Pachori, and U. R. Acharya, "Automated Diagnosis of Glaucoma Using Empirical Wavelet Transform and Correntropy Features Extracted from Fundus Images," *IEEE J. Biomed. Heal. Informatics*, vol. 21, no. 3, pp. 803–813, May 2017.
- [5] N. Sharma, A. Ray, S. Sharma, K. Shukla, S. Pradhan, and L. Aggarwal, "Segmentation and classification of medical images using texture-primitive features: Application of BAM-type artificial neural network," *J. Med. Phys.*, vol. 33, no. 3, pp. 119–126, Jul. 2008.
- [6] A. R. Ajiboye, R. Abdullah-Arshah, H. Qin, and H. Isah-Kebbe, "Evaluating the Effect of Dataset Size on Predictive Model Using Supervised Learning Technique," *Int. J. Softw. Eng. Comput. Sci.*, vol. 1, no. February, pp. 75–84, 2015.
- [7] C. C. Chang, C. C. Lin, P. Y. Pai, and Y. C. Chen, "A novel retinal blood vessel segmentation method based on line operator and edge detector," *IIH-MSP 2009 - 2009 5th Int. Conf. Intell. Inf. Hiding Multimed. Signal Process.*, pp. 299–302, 2009.
- [8] K. BahadarKhan, A. A. Khaliq, and M. Shahid, "A Morphological Hessian Based Approach for Retinal Blood Vessels Segmentation and Denoising Using Region Based Otsu Thresholding," *PLoS One*, vol. 11, no. 9, pp. 1–

19, 2016.

- [9] J. Dash and N. Bhoi, "Retinal Blood Vessel Segmentation Using Otsu Thresholding With Principal Component Analysis," *Proc. 2nd Int. Conf. Inven. Syst. Control. ICISC 2018*, no. Icisc, pp. 933–937, 2018.
- [10] Ali Abdo Mohammed Al-Kubati, J. A. M. Saif, A. Taher, and M. A., "Evaluation of Canny and Otsu Image Segmentation," *Int. Conf. Emerg. Trends Comput. Electron. Eng.*, pp. 23–25, 2012, [Online]. Available: <http://psrcentre.org/images/extraimages/312651.pdf>.
- [11] W. X. Kang, Q. Q. Yang, and R. P. Liang, "The Comparative Research on Image Segmentation Algorithms," *Proc. 1st Int. Work. Educ. Technol. Comput. Sci. ETCS 2009*, vol. 2, pp. 703–707, 2009.
- [12] C. Nivetha, S. Sumathi, and M. Chandrasekaran, "Retinal Blood Vessels Extraction and Detection of Exudates Using Wavelet Transform and PNN Approach for the Assessment of Diabetic Retinopathy," *Proc. 2017 IEEE Int. Conf. Commun. Signal Process. ICCSP 2017*, pp. 1962–1966, 2018.
- [13] L. Abdel-Hamid, "Glaucoma Detection from Retinal Images Using Statistical and Textural Wavelet Features," *J. Digit. Imaging*, vol. 33, no. 1, pp. 151–158, 2020.
- [14] R. Gayathri, P. V. Rao, and S. Aruna, "Automated glaucoma detection system based on wavelet energy features and ANN," *Proc. 2014 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2014*, pp. 2808–2812, 2014.
- [15] J. Sivakamasundari, G. Kavitha, V. Natarajan, and S. Ramakrishnan, "Content Based Human Retinal Image Retrieval Using Vascular Feature Extraction," *Asian Conf. Intell. Inf. Database Syst.*, pp. 468–476, 2012.
- [16] R. Dhar, R. Gupta, and K. L. Baishnab, "An Analysis of Canny and Laplacian of Gaussian Image Filters in Regard to Evaluating Retinal Image," *Proceeding IEEE Int. Conf. Green Comput. Commun. Electr. Eng. ICGCCEE 2014*, 2014.
- [17] S. Sarangi, A. Mohapatra, and S. K. Sabut, "A Comparative Evaluation of Diabetic Retinal Vascular Structures Using Edge Detection Techniques," *Int. J. Telemed. Clin. Pract.*, vol. 1, no. 1, p. 111, 2015.

- [18] S. Chakraborty, S. Chatterjee, N. Dey, A. S. Ashour, and F. Shi, "Gradient approximation in retinal blood vessel segmentation," *2017 4th IEEE Uttar Pradesh Sect. Int. Conf. Electr. Comput. Electron. UPCON 2017*, pp. 618–623, 2017.
- [19] Wiharto and Y. S. Palgunadi, "Blood Vessels Segmentation in Retinal Fundus Image Using Hybrid Method of Frangi Filter , Otsu Thresholding and Morphology," vol. 10, no. 6, pp. 417–422, 2019.
- [20] O. Ali, N. Muhammad, Z. Jadoon, B. M. Kazmi, N. Muzamil, and Z. Mahmood, "A Comparative Study of Automatic Vessel Segmentation Algorithms," *2020 3rd Int. Conf. Comput. Math. Eng. Technol. Idea to Innov. Build. Knowl. Econ. iCoMET 2020*, pp. 1–6, 2020.
- [21] N. R. Sabri, H. Yazid, and S. A. Rahim, "Performance comparison using thresholding based method for diabetic retinopathy," *ISCAIE 2019 - 2019 IEEE Symp. Comput. Appl. Ind. Electron.*, pp. 228–233, 2019.
- [22] E. Permata, R. Munarto, and G. A. . Indra, "Klasifikasi Glaukoma Menggunakan Neural Network Backpropagation," in *Prosiding SENTIA 2016*, 2016, vol. 8, pp. 158–163.
- [23] A. Mustofa, H. Tjandrasa, and B. Amaliah, "Deteksi Penyakit Glaukoma pada Citra Fundus Retina Menggunakan Adaptive Thresholding dan Support vector machine," vol. 5, no. 2, pp. 3–6, 2016.
- [24] A. Ambarwati, R. Passarella, and Sutarno, "Segmentasi Citra Digital Menggunakan Thresholding Otsu untuk Analisa Perbandingan Deteksi Tepi," *Annu. Res. Semin. 2016*, vol. 2, no. 1, pp. 216–226, 2016.
- [25] R. E. Putra, H. Tjandrasa, and N. Suciati, "Review Algoritma Segmentasi Pembuluh Darah Pada Citra Fundus Retina Mata Untuk Membantu Diagnosis Diabetic Retinopathy," *JUTI J. Ilm. Teknol. Inf.*, vol. 10, no. 2, p. 66, 2012.
- [26] A. Kadir and A. Susanto, *Teori dan Aplikasi Pengolahan Citra*. 2013.
- [27] N. Nurhalimah, I. G. P. Suta Wijaya, and F. Bimantoro, "Klasifikasi Kain Songket Lombok Berdasarkan Fitur GLCM dan Moment Invariant Dengan Teknik Pengklasifikasian Linear Discriminant Analysis (LDA)," *J. Teknol.*

- Informasi, Komputer, dan Apl. (JTIKA)*, vol. 2, no. 2, pp. 173–183, 2020.
- [28] Y. A. Lesnussa, S. Latuconsina, and E. R. Persulesy, “Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Memprediksi Prestasi Siswa SMA (Studi kasus: Prediksi Prestasi Siswa SMAN 4 Ambon),” *J. Mat. Integr.*, vol. 11, no. 2, p. 149, 2015.
 - [29] A. A. S. M. K. Maharani and F. Bimantoro, “Pengenalan Pola Tulisan Tangan Aksara Sasak Menggunakan Metode Linear Discriminant Analysis dan Jaringan Syaraf Tiruan Jenis Backpropagation,” *J. Teknol. Informasi, Komputer, dan Apl. (JTIKA)*, vol. 2, no. 2, pp. 237–247, 2020.
 - [30] B. B. Wiko Putra, M. A. Albar, and B. Irmawati, “Penerapan Algoritma Jaringan Saraf Tiruan Metode Backpropagation untuk Memprediksi Jumlah Nilai Ekspor di Provinsi NTB,” *J. Teknol. Informasi, Komputer, dan Apl. (JTIKA)*, vol. 1, no. 2, 2019.
 - [31] G. S. Nugraha, I. Soesanti, and S. Wibirama, “Deteksi Glaukoma Berdasarkan Kesamaan Ciri Tekstur Optic Cup dan Optic Disc pada Citra Fundus Retina,” Universitas Gadjah Mada, 2017.
 - [32] A. Bisri and R. Rachmatika, “Integrasi Gradient Boosted Trees dengan SMOTE dan Bagging untuk Deteksi Kelulusan Mahasiswa,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 8, no. 4, p. 309, 2019.
 - [33] C. Pena-Betancor *et al.*, “Estimation of the relative amount of hemoglobin in the cup and neuroretinal rim using stereoscopic color fundus images,” *Investig. Ophthalmol. Vis. Sci.*, vol. 56, no. 3, pp. 1562–1568, 2015.