



TUGAS AKHIR -SM141501

**IMPLEMENTASI JARINGAN SARAF TIRUAN UNTUK
PENGENALAN *PLASMODIUM FALCIPARUM* PADA
CITRA SEL DARAH MERAH**

**ALFIANI PUTRI
NRP 0611144000069**

**Dosen Pembimbing
Prof. Dr. Mohammad Isa Irawan, MT**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember
Surabaya 2018**



FINAL PROJECT -SM141501

***IMPLEMENTATION OF NEURAL NETWORKS FOR
RECOGNITION OF PLASMODIUM FALCIPARUM ON
RED BLOOD CELLS IMAGES***

Alfiani Putri
NRP 06111440000069

Supervisor
Prof. Dr. Mohammad Isa Irawan, MT

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics, Computing, and Data Science
Sepuluh Nopember Institute of Technology
Surabaya 2018

LEMBAR PENGESAHAN

**IMPLEMENTASI JARINGAN SARAF TIRUAN UNTUK
PENGENALAN *PLASMODIUM FALCIPARUM*
PADA CITRA SEL DARAH MERAH**

***IMPLEMENTATION OF NEURAL NETWORKS FOR
RECOGNITION OF PLASMODIUM FALCIPARUM
ON RED BLOOD CELLS IMAGES***

TUGAS AKHIR

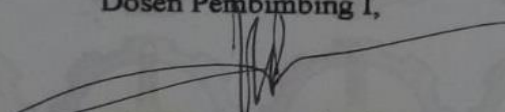
Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :

ALFIANI PUTRI
NRP. 06111440000069


Menyetujui,

Dosen Pembimbing I,


Prof. Dr. M. Isa Irawan, MT
NIP. 19631225 198903 1 001

Mengetahui,

**Ketua Departemen Matematika
FMKSD ITS**


Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003

Surabaya, Juli 2018

**IMPLEMENTASI JARINGAN SARAF TIRUAN UNTUK
PENGENALAN *PLASMODIUM FALCIPARUM*
PADA CITRA SEL DARAH MERAH**

Nama : Alfiani Putri
NRP : 06111440000069
Departemen : Matematika
Dosen Pembimbing : Prof. Dr. M. Isa Irawan, MT

ABSTRAK

WHO (*World Health Organization*) mencatat sekitar 429.000 kasus malaria yang menyebabkan kematian di dunia. Hal ini dikarenakan ahli parasitologi yang sedikit di daerah endemik malaria, sehingga membuat pasien sulit untuk segera mendapat penanganan dini. Penelitian berjudul "Implementasi Jaringan Saraf Tiruan untuk Pengenalan *Plasmodium Falciparum* pada Citra Sel Darah Merah" telah melakukan analisis otomatis berdasarkan citra sel darah dalam mendiagnosa penyakit malaria dengan lebih cepat dan akurat. Pada penelitian ini digunakan metode jaringan saraf tiruan model *Multi Perceptron*, dengan dua tahapan. Tahap pertama, yaitu proses identifikasi *plasmodium falciparum* diawali dengan pengolahan citra digital untuk mendapatkan fitur area sel darah merah dan fitur area akromia sentral. Tahap kedua, yaitu proses deteksi sel darah stadium *gametocyte* berbentuk pipih. Proses deteksi *gametocyte* diawali dengan *pre-processing*, segmentasi dan operasi morfologi untuk mencari nilai eksentrisitas sel darah yang digunakan sebagai masukan JST Multi Perceptron. Nilai akurasi untuk identifikasi sel terinfeksi *plasmodium falciparum* pada stadium *gametocyte* dengan menggunakan JST Multi Perceptron adalah sebesar 92.3%.

Kata Kunci : Pengenalan *Plasmodium Falciparum*, Jaringan Saraf Tiruan, Model Multi Perceptron

“Halaman ini sengaja dikosongkan”

**IMPLEMENTATION OF NEURAL NETWORKS FOR
RECOGNITION OF PLASMODIUM FALCIPARUM
ON RED BLOOD CELLS IMAGES**

Name of Student : Alfiani Putri
NRP : 06111440000069
Department : Mathematics
Supervisor : Prof. Dr. M. Isa Irawan, MT

ABSTRACT

WHO (World Health Organization) recorded about 429,000 cases of malaria that cause death in the world. This is because a few parasitologists in the malaria endemic area, making it difficult for patients to get immediate treatment. The study, titled "Implementation of Neural Networks for Recognition of Plasmodium Falciparum on Red Blood Cells Images" has performed automatic analysis based on the image of blood cells in diagnosing malaria more quickly and accurately. In this research, Multi Perceptron model of artificial neural network using two stages. The first stage, which is the identification process plasmodium falciparum begins with digital image processing to obtain features of red blood cell area and features a central acromia area. The second stage, namely the process of blood cell staging gametocyte-shaped flat. The gametocyte detection process begins with pre-processing, segmentation and morphological operations to find the eccentricity of blood cells used as a Multi Perceptron JST input. The accuracy value for the identification of plasmodium falciparum infected cells at gametocyte stage using Multi Perceptron JST was 92.3%.

Keywords : Recognition of Plasmodium Falciparum, Neural Networks, Multi Perceptron Model

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji dan syukur bagi Allah SWT yang memiliki apa yang ada di langit dan di bumi atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul

“IMPLEMENTASI JARINGAN SARAF TIRUAN UNTUK PENGENALAN *PLASMODIUM FALCIPARUM* PADA CITRA SEL DARAH MERAH”

sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika FMKSD Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Dalam menyelesaikan Tugas Akhir ini, penulis telah banyak mendapat bantuan serta masukan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis menyampaikan terima kasih kepada:

1. Keluarga tercinta yang senantiasa memberikan dukungan dan doa dengan ikhlas.
2. Bapak Prof. Dr. Mohammad Isa Irawan, MT selaku Dosen Pembimbing Tugas Akhir yang telah banyak memberikan bimbingan, arahan serta motivasi sehingga Tugas Akhir ini dapat terselesaikan.
3. Bapak Dr. Imam Mukhlash, S.Si, MT selaku Ketua Departemen Matematika ITS.
4. Ibu Dr. Dra. Mardijah, M.T. selaku Dosen Wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Departemen Matematika ITS.
5. Bapak Dr. Mahmud Yunus, M.Si, Bapak Drs. Daryono Budi Utomo, MSi, Ibu Dr. Dwi Ratna Sulistyaningrum,

S.Si, MT, dan Bapak Dr. Budi Setiyono, S.Si, MT selaku Dosen Penguji yang telah memberikan saran demi perbaikan Tugas Akhir ini.

6. Bapak Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi S1 Departemen Matematika ITS.
7. Seluruh jajaran dosen dan staf Departemen Matematika ITS.
8. Teman-teman Departemen Matematika ITS angkatan 2014 yang saling mendukung dan memotivasi.
9. Semua pihak yang tidak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

Apabila dalam Tugas Akhir ini ada kekurangan, penulis mengharapkan kritik dan saran dari pembaca. Semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum Wr. Wb.

Surabaya, 26 Juli 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan Tugas Akhir.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Penelitian Terdahulu.....	7
2.2 Plasmodium Falciparum.....	9
2.3 Pengolahan Citra Digital.....	10
2.3.1 Pencuplikan Citra.....	11
2.3.2 Citra Berwarna dan Citra Grayscale.....	12
2.3.3 Thresholding.....	13
2.3.4 Operasi Morphologi.....	14
2.3.5 Pelabelan Komponen.....	19
2.3.6 Ekstraksi Fitur.....	19
2.3.7 Eccentricity.....	20
2.4 Jaringan Saraf Tiruan.....	21
2.5 Jaringan Multi Perceptron.....	25
BAB III METODOLOGI PENELITIAN.....	27
3.1 Diagram Metodologi.....	27
3.2 Studi Literatur.....	28
3.3 Pengumpulan Data.....	28

3.4	Perancangan Sistem.....	28
3.5	Implementasi Sistem.....	30
3.6	Pengujian dan Evaluasi Sistem.....	30
3.7	Penarikan Kesimpulan.....	30
3.8	Penulisan Laporan Tugas Akhir.....	31
BAB IV	PERANCANGAN SISTEM.....	33
4.1	Pengumpulan Dataset.....	33
4.2	Pengolahan Data.....	33
4.2.1	<i>Pre-Processing</i>	35
4.2.2	Segmentasi.....	36
4.2.3	Operasi Morphologi.....	38
4.2.4	Ekstraksi Fitur.....	39
4.3	Nilai Input pada Proses <i>Training</i>	40
4.3.1	Proses Identifikasi <i>Plasmodium</i> <i>Falciparum</i>	40
4.3.2	Proses <i>Training</i> dengan Multi Perceptron	42
BAB V	IMPLEMENTASI SISTEM.....	49
5.1	Lingkungan <i>Hardware</i> dan <i>Software</i>	49
5.2	Implementasi Praproses Data	49
5.3	Implementasi JST Model Multi Perceptron.....	54
5.3.1	Proses <i>Training</i>	54
5.3.2	Proses <i>Testing</i>	57
BAB VI	UJI COBA DAN EVALUASI SISTEM.....	59
6.1	Dataset Uji Coba.....	59
6.2	Proses <i>Testing</i> dengan Multi Perceptron.....	60
6.3	Hasil <i>Testing</i> JST.....	62
6.4	Uji Coba Model	64
6.4.1	Uji Coba Sel Darah Terinfeksi <i>Plasmodium</i> <i>Falciparum</i>	64
6.4.2	Uji Coba Sel Darah Normal.....	66
6.4	Tabel Analisis Hasil.....	69
BAB VII	KESIMPULAN DAN SARAN.....	81
7.1	Kesimpulan.....	81
7.2	Saran.....	82

DAFTAR PUSTAKA.....	83
LAMPIRAN A.....	85
LAMPIRAN B.....	93
LAMPIRAN C.....	95
BIODATA PENULIS.....	111

“Halaman ini sengaja dikosongkan”

DAFTAR GAMBAR

Gambar 2.1.	Parasit Plasmodium Falciparum. (a) <i>thick blood film</i> , (b) <i>thin blood film</i>	10
Gambar 2.2.	Proses <i>sampling</i> dan kuantisasi. (a) Citra Digital, (b) Citra Digital Disampling Menjadi 14 Baris dan 12 Kolom, (c) Citra Digital Hasil Sampling Berukuran 14 x 12 Piksel.....	12
Gambar 2.3.	Contoh Proses Dilasi pada Citra Biner.....	16
Gambar 2.4.	Contoh Proses Erosi pada Citra Biner.....	17
Gambar 2.5.	(a) Citra biner, (b) Citra dengan <i>filling holes</i> sebagian (c) Citra dengan <i>filling holes</i> keseluruhan.....	18
Gambar 2.6.	Sebuah citra biner dan urutan pemberian label pada obyek-obyeknya.....	19
Gambar 2.7.	Geometri Elips.....	20
Gambar 2.8.	Sel Syaraf (Neuron).....	22
Gambar 2.9.	<i>Feedforward Neural Network</i>	23
Gambar 2.10.	<i>Recurrent Neural Network</i>	24
Gambar 2.11.	Arsitektur Multi Perceptron.....	25
Gambar 3.1.	Diagram Alir Metodologi Penelitian.....	27
Gambar 3.2.	Diagram Alir Rancangan Sistem.....	29
Gambar 4.1.	Koordinat titik pojok bagian citra yang <i>dicrop</i>	34
Gambar 4.2.	Sel darah terinfeksi <i>plasmodium falciparum</i>	35
Gambar 4.3.	Sel darah normal.....	35
Gambar 4.4.	(a) Citra RGB, (b) Citra <i>greyscale</i>	36
Gambar 4.5.	Pemberian Nilai <i>Thresholding</i>	37
Gambar 4.6.	Hasil Proses Segmentasi.....	38
Gambar 4.7.	(a) Hasil <i>cropping</i> , (b) Hasil Segmentasi, (c) Hasil proses <i>clear border</i>	38
Gambar 4.8.	(a) Hasil proses <i>clear border</i> , (b) Hasil setelah melalui proses <i>filling holes</i>	39

Gambar 4.9.	Flowchart tahap <i>training</i> model Multi Perceptron.....	45
Gambar 5.1.	Desain menu Home program identifikasi citra <i>plasmodium falciparum</i>	54
Gambar 5.2.	Tampilan program <i>Training</i> Multi Perceptron...	56
Gambar 5.3.	Tampilan program <i>Testing</i> Multi Perceptron....	58
Gambar 6.1.	Jumlah data pelatihan dan pengujian.....	59
Gambar 6.2.	Flowchart tahap <i>testing</i> Multi Perceptron Untuk identifikasi sel terinfeksi <i>plasmodium falciparum</i>	61
Gambar 6.3.	Tampilan program input data.....	65
Gambar 6.4.	Tampilan program Identifikasi Data.....	65
Gambar 6.5.	Tampilan program Identifikasi <i>Plasmodium Falciparum</i> dan deteksi sel stadium <i>gametocyte</i>	66
Gambar 6.6.	Tampilan program input data.....	67
Gambar 6.7.	Tampilan program Identifikasi Data.....	67
Gambar 6.8.	Tampilan program Identifikasi Sel Darah Normal.....	68

DAFTAR TABEL

Tabel 4.1.	Tabel nilai fitur dalam proses identifikasi Sel darah terinfeksi <i>plasmodium falciparum</i>	41
Tabel 4.2.	Tabel nilai fitur dalam proses identifikasi Sel darah normal/bukan <i>plasmodium falciparum</i>	42
Tabel 4.3.	Hasil pelatihan jaringan Multi Perceptron.....	46
Tabel 5.1.	Spesifikasi <i>Hardware</i> dan <i>Software</i>	49
Tabel 6.1.	Hasil pengujian jaringan Multi Perceptron.....	62
Tabel 6.2.	Hasil pengujian jaringan Multi Perceptron <i>stadium gametocyte</i>	64
Tabel 6.3.	Data hasil pengujian jaringan Multi Perceptron.....	69
Tabel 6.4.	Data hasil pengujian deteksi stadium <i>gametocyte</i>	77

“Halaman ini sengaja dikosongkan”

BAB I

PENDAHULUAN

Pada bab ini dijelaskan hal-hal yang melatar belakangi munculnya permasalahan yang dibahas dalam Tugas Akhir ini. Kemudian permasalahan tersebut disusun ke dalam suatu rumusan masalah. Selanjutnya dijabarkan juga batasan masalah untuk mendapatkan tujuan yang diinginkan serta manfaat yang dapat diperoleh. Adapun sistematika penulisan Tugas Akhir ini akan diuraikan di bagian akhir bab ini.

1.1 Latar Belakang

Pada umumnya, penyakit malaria disebabkan oleh protozoa dari genus *Plasmodium* yaitu *plasmodium falciparum*, *plasmodium vivax*, *plasmodium ovale*, dan *plasmodium malariae*. Penyakit ini menyebar pada manusia melalui gigitan nyamuk *Anopheles*. *Plasmodium falciparum* dan *Plasmodium vivax* merupakan jenis plasmodium yang paling sering dijumpai pada kasus penyebaran penyakit malaria pada sel darah merah. Di Indonesia, secara umum spesies yang paling sering ditemukan adalah *Plasmodium falciparum* dan *Plasmodium vivax*, *Plasmodium malariae* jarang ditemukan di Indonesia bagian timur, sedangkan *Plasmodium ovale* lebih jarang lagi. Penemuannya pernah dilaporkan dari Flores, Timor dan Irian Jaya. Pada tahun 2013 penyebab malaria yang tertinggi adalah *Plasmodium falciparum* (56%), kemudian *Plasmodium vivax* (44%), sedangkan *Plasmodium ovale* dan *Plasmodium malariae* tidak dilaporkan. Di tahun yang sama, populasi penduduk Indonesia tercatat 249.865.631, dari sejumlah populasi tersebut yang dicurigai malaria tercatat 3.197.890, kasus positif terinfeksi malaria dilaporkan terdapat 343.527 dan tercatat 45 kematian [1].

Menurut penelitian yang dilakukan di Balai Litbang P2B2 Banjarnegara, di daerah-daerah endemis malaria, pada dasarnya masih sangat sedikit ahli parasitologi yang berada di

daerah tersebut, sehingga pengamatan mikroskopis yang dilakukan lebih sering dilakukan secara konvensional dengan hasil keakuratan diagnosis terjangkau penyakit malaria pada pasien masih sangat rendah, akibatnya banyak pasien terlambat untuk pemberian pengobatan [2].

Berdasarkan fakta tersebut, dibutuhkan metode untuk merancang suatu program simulasi pendeteksian, dimana proses pembuatan program melibatkan seorang pakar namun nantinya program simulasi pendeteksian tersebut dapat dioperasikan oleh *non*-pakar. Penyakit malaria dapat didiagnosa dengan mencari parasit pada tetesan darah penderita. Pada penelitian ini, akan dilakukan pendeteksian *Plasmodium falciparum* melalui pengolahan citra digital dimana data yang berupa citra mikroskopis sel darah yang terinfeksi *Plasmodium falciparum* dapat diambil informasinya. Kemudian dapat dilakukan identifikasi dan pengelompokan citra ke dalam suatu kelas. Pengenalan *Plasmodium falciparum* digunakan untuk menghitung jumlah parasit *Plasmodium falciparum* yang berada pada citra sel darah merah berdasarkan stadiumnya, yang nantinya akan dihasilkan keputusan apakah sel darah tersebut terjangkau parasit *Plasmodium falciparum* atau tidak.

Dalam penelitian ini akan digunakan proses pengolahan citra digital untuk pengenalan *Plasmodium falciparum* pada citra sel darah, dan hasil pengolahan citra akan dianalisis menggunakan metode jaringan saraf tiruan. Metode jaringan saraf tiruan (JST) diterapkan tak lain karena metode ini memiliki iterasi yang cukup banyak, dan memiliki kemampuan sangat baik dalam mengeneralisasi masalah, sehingga hasil yang didapat sangat akurat [3]. Jaringan saraf tiruan (JST) yang digunakan dalam penelitian ini adalah model *Multi Perceptron*. Model Multi Perceptron merupakan jaringan saraf sederhana yang dapat digunakan untuk mengklasifikasikan suatu tipe pola tertentu dan dikenal dengan pemisahan secara linier [4].

Identifikasi jaringan *Multi Perceptron* akan menunjukkan hasil bahwa sel darah tidak mengandung parasit malaria apabila di dalam sel darah tersebut tidak terinfeksi parasit malaria yaitu *Plasmodium falciparum*, sedangkan hasil identifikasi jaringan *Multi Perceptron* yang menunjukkan sel darah mengandung parasit malaria yaitu *Plasmodium falciparum* akan diproses lebih lanjut dengan mengidentifikasi stadium perkembangan *plasmodium falciparum* menggunakan JST *Multi Perceptron* berdasarkan ekstraksi ciri morfologi. *Plasmodium falciparum* pada dasarnya memiliki tiga stadium, yaitu stadium *tropozoit*, stadium *skizon*, dan stadium *gametosit*. Sel darah merah yang terinfeksi parasit *Plasmodium falciparum* menunjukkan sel darah tersebut mengandung parasit malaria.

Penelitian terdahulu yang menggunakan model jaringan saraf tiruan *Multi Perceptron* adalah Mike Susmikanti (2010) pada penelitiannya yang berjudul "Pengenalan Pola Berbasis Jaringan Syaraf Tiruan Dalam Analisa CT Scan Tumor Otak Beligna" penelitian ini mengidentifikasi dan mengklasifikasikan jenis tumor jinak berdasarkan citra hasil CT scan dengan menggunakan JST model *Multi Perceptron*.

Berdasarkan penelitian sebelumnya, penelitian ini bertujuan untuk mengidentifikasi ada atau tidaknya parasit *Plasmodium falciparum* penyebab malaria yang paling mematikan dengan menggunakan salah satu metode jaringan saraf tiruan yaitu model *Multi Perceptron*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, dapat ditarik rumusan masalah dari Tugas akhir ini adalah bagaimana mengimplementasikan jaringan saraf tiruan model *Multi Perceptron* untuk pengenalan *Plasmodium Falciparum* pada citra sel darah merah.

1.3 Batasan Masalah

Permasalahan dalam dunia nyata sangatlah luas oleh karena itu diberikan suatu batasan masalah pada Tugas Akhir ini antara lain:

1. Data citra yang digunakan adalah sel darah merah yang terjangkit parasit malaria berjenis *Plasmodium falciparum* dan data citra sel darah merah yang normal.
2. Parasit *Plasmodium falciparum* yang akan dikenali pada sel darah merah berbasis jaringan saraf tiruan, menggunakan metode *Multi Perceptron*.
3. Data citra darah yang digunakan pada penelitian ini merupakan hasil citra mikroskopis digital dari apusan darah tepi yang didapat dari Laboratorium Perawatan Koya Barat Distrik Muara Tami Kota Jayapura Propinsi Papua.
4. *Plasmodium falciparum* pada citra sel darah merah diambil menggunakan mikroskop dengan perbesaran 100 kali.
5. Simulasi menggunakan software MATLAB.

1.4 Tujuan

Berdasarkan permasalahan yang telah dirumuskan, tujuan dari tugas akhir ini adalah :

1. Membuat suatu dasar *prototipe* yang dapat mengimplementasikan dan mengetahui kapabilitas metode jaringan saraf tiruan model *Multi Perceptron* untuk pengenalan *Plasmodium falciparum* pada citra sel darah merah.
2. Menerapkan metode JST *Multi Perceptron* untuk menghitung jumlah parasit *Plasmodium falciparum* yang terkandung pada citra sel darah merah.

1.5 Manfaat

Manfaat yang diperoleh dari Tugas Akhir ini antara lain:

1. Dapat dijadikan dasar untuk pengembangan konsep *On-site rapid diagnostic* yang dapat membantu tenaga medis di pedalaman kepulauan Indonesia di luar Jawa dalam mendiagnosa penyakit malaria.

2. Dapat digunakan sebagai alternatif di bagian laboratorium kesehatan untuk mengenali citra-citra mikroskopis dengan tujuan mengidentifikasi keberadaan suatu parasit tertentu di dalam suatu sel darah.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika penulisan didalam Tugas Akhir ini adalah sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang pembuatan Tugas Akhir, rumusan dan batasan permasalahan yang dihadapi dalam penelitian Tugas Akhir, tujuan dan manfaat pembuatan Tugas Akhir serta sistematika penulisan Tugas Akhir.

2. BAB II TINJAUAN PUSTAKA

Bab ini berisi teori dasar yang mendukung dalam Tugas Akhir ini, antara lain penelitian terdahulu, penjelasan *plasmodium falciparum*, pengolahan citra digital, jaringan saraf tiruan, dan jaringan *Multi Perceptron*.

3. BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tahap pengerjaan dalam menyelesaikan Tugas Akhir yang terdiri dari studi literatur, pengumpulan data, perancangan sistem, implementasi sistem, pengujian dan evaluasi sistem, serta penarikan kesimpulan dan penulisan laporan Tugas Akhir.

4. BAB IV PERANCANGAN SISTEM

Bab ini menjelaskan tahap persiapan pengolahan data hingga proses konstruksi sistem menggunakan metode jaringan saraf tiruan model *Multi Perceptron* sebagai acuan dalam implementasi sistem.

5. **BAB V IMPLEMENTASI SISTEM**

Bab ini membahas proses implementasi dengan menggunakan bahasa pemograman MATLAB berdasarkan rancangan sistem yang telah dibuat pada bab sebelumnya.

6. **BAB VI UJI COBA DAN EVALUASI SISTEM**

Bab ini membahas tahap-tahap uji coba berdasarkan implementasi sistem yang telah dibuat beserta pengujian kinerja sistem.

7. **BAB VII KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan Tugas Akhir yang diperoleh dari bab uji coba dan evaluasi sistem serta saran untuk pengembangan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini diuraikan mengenai dasar teori yang digunakan dalam penyusunan Tugas Akhir ini. Dasar teori yang dijelaskan dibagi menjadi beberapa sub bab yaitu penelitian terdahulu, penjelasan *Plasmodium Falciparum*, pengolahan citra digital, jaringan saraf tiruan, dan jaringan *Multi Perceptron*.

2.1 Penelitian Terdahulu

Pada beberapa penelitian sebelumnya dengan tema yang terkait dengan Tugas Akhir ini adalah penelitian yang relevan dengan tema yang diambil yaitu Adi pamungkas dkk [5] melakukan penelitian tentang penghitungan otomatis jumlah sel darah merah dan identifikasi fase *plasmodium falciparum* menggunakan metode *Morphological Operations*. Penelitian tersebut dilakukan untuk menghitung jumlah sel darah merah total, menghitung jumlah parasit yaitu plasmodium falciparum, dan mengidentifikasi fase perkembangan *plasmodium falciparum*. Proses pemisahan sel bertumpuk dilakukan dengan menggunakan operasi morfologi. Sedangkan proses penghitungan dan identifikasi fase plasmodium falciparum dilakukan dengan melakukan ekstraksi ciri morfologi parasit. Hasil akurasi yang didapat dari penelitian tersebut dengan koefisien korelasi yang digunakan 0,997 adalah 87,5% pada penghitungan jumlah parasit dan 87,5% pada identifikasi jenis *plasmodium falciparum*.

Mike Susmikanti melakukan penelitian tentang cara mengidentifikasi dan mengklasifikasikan jenis tumor jinak berdasarkan citra hasil CT scan dengan menggunakan Jaringan Saraf Tiruan model *Perceptron*. Data penelitian yang berupa citra dari hasil CT Scan otak ataupun MRI dikonversi kedalam bentuk digital standard menggunakan teknik pengolahan citra. Hasil konversi digital, menggunakan PCA menghasilkan

karakteristik dominan sehingga mewakili pola citra tersebut. Karakteristik dominan ini digunakan pada jaringan syaraf tiruan. Dalam identifikasi pola, pada tahap pembelajaran dengan pengawasan digunakan metode Multi Perceptron. Identifikasi diberikan untuk 120 sampel dengan hasil ketelitian sebesar 95%. [6].

Daniel Maitethia Memeu dkk [7] melakukan penelitian tentang sebuah sistem untuk mendeteksi parasit plasmodium menggunakan citra dari apusan darah tipis dengan metode *Artificial Neural Network (ANN) multilayer back propagation*. Sebanyak 205 citra digunakan untuk melatih dan menguji kinerja sistem. Sistem ini memiliki akurasi 95% dalam mendeteksi keberadaan parasit plasmodium. Segmentasi citra menggunakan jaringan syaraf tiruan yang di-training dengan nilai-nilai pixel dari warna RGB. Hasil deteksi parasit plasmodium menunjukkan bahwa JST classifier dilatih hanya dengan fitur warna RGB dan hasilnya lebih baik daripada ketika classifier di-training pada fitur warna RGB dan fitur warna HSI. Oleh karena itu disimpulkan bahwa deteksi parasit plasmodium dapat dilakukan secara memadai oleh classifier jaringan saraf yang di-training dengan fitur warna RGB pada citra parasit.

Muhammad Athoillah dkk [8] melakukan penelitian tentang model klasifikasi dengan algoritma SVM yang dimodifikasi dengan *multiple kernel learning* dan diaplikasikan untuk *image retrieval*. *Support Vector Machine (SVM)* adalah salah satu teknik yang dapat menyelesaikan masalah klasifikasi dengan baik. Untuk mengklasifikasi data yang tidak terpisah secara linear, algoritma ini harus dimodifikasi dengan *kernel learning*. Hasil penelitian ini menunjukkan bahwa SVM dengan *multi kernel learning* memiliki akurasi yang baik dan waktu komputasi yang singkat. Hasil ini juga dibandingkan dengan ketika menggunakan *single kernel Polynomial* dan RBF kernel. Hasilnya menunjukkan bahwa RBF kernel tidak sesuai untuk masalah klasifikasi seperti ini yang ditandai

dengan akurasi yang rendah dan lamanya waktu komputasi. Hal ini menunjukkan bahwa tidak setiap kernel sesuai untuk masalah klasifikasi.

Nur Ain dkk [9] melakukan penelitian tentang klasifikasi pada *Plasmodium falciparum* dengan menggunakan metode *Filter Gabor* dan *Multi Class Support Vector (SVM)* dengan menggunakan metode “*one-against-all*” untuk klasifikasi stadium dari penyakit malaria. Hasil menunjukkan akurasi terbaik sebesar 73,33% pada 150 citra *Plasmodium falciparum* dengan parameter dari *Filter Gabor* $N = 8, \lambda = 7, \theta = 25^\circ, \varphi = 0, b = 1, \text{ dan } \Upsilon = 0,5$.

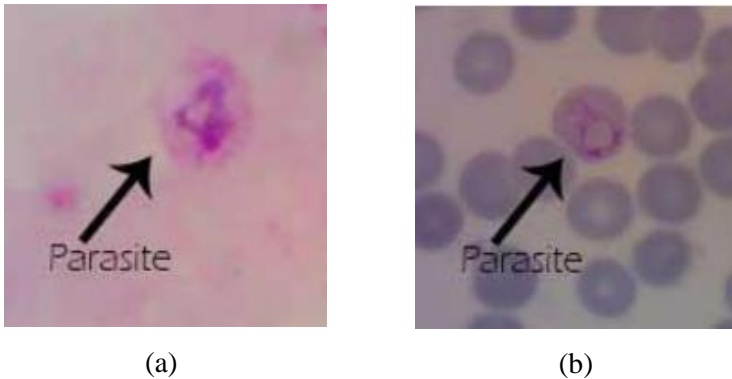
2.2 Plasmodium Falciparum

Plasmodium falciparum adalah protozoa parasit yang pathogen yang sering dan didistribusikan secara luas sebagian besar menyebabkan malaria. Manusia merupakan hospes perantara parasite ini, sedangkan hospes definitifnya adalah nyamuk *Anopheles* betina. *Plasmodium falciparum* menyebabkan penyakit malaria *falciparum*, dapat juga disebut malaria tropika. [10]

Plasmodium falciparum pada dasarnya memiliki tiga stadium, yaitu ;

- Stadium *tropozoit* (tropozoit muda / bentuk cincin,).
- Stadium *skizon* (Skizon muda, skizon tua)
- Stadium *gametosit* (gametosit jantan, gametosit betina).

Plasmodium falciparum, memiliki distribusi geografis terluas, mulai dari wilayah beriklim dingin, subtropik hingga daerah tropik. Demam terjadi setiap 48 jam atau setiap hari ketiga, pada siang atau sore. Masa inkubasi *plasmodium falciparum* antara 12 sampai 17 hari dan salah satu gejala adalah pembengkakan limpa atau *splenomegaly* [10].



Gambar 2.1. Parasit Plasmodium Falciparum
(a) *thick blood film* (b) *thin blood film* [10]

2.3 Pengolahan Citra Digital

Dalam pengolahan maupun pengenalan citra, masalah persepsi visual, yaitu apa yang dapat dilihat oleh mata manusia, mempunyai peranan penting. Penentuan apa yang dapat dilihat tidak hanya ditentukan oleh manusia itu sendiri. Mata merupakan bagian sistem visual manusia. Sistem visual ini sangat sulit dipelajari, terlebih jika ingin menyingkap proses yang melatarbelakangi timbulnya suatu persepsi, seperti pada peristiwa pengenalan (*recognition*).

Pada dasarnya, citra merupakan suatu fungsi intensitas cahaya dalam bidang dua dimensi. Karena intensitas yang dimaksud berasal dari sumber cahaya, dan cahaya adalah suatu bentuk energi, maka berlaku keadaan di mana fungsi intensitas terletak di antara :

$$0 < f(x, y) < \infty$$

Pada dasarnya, citra yang dilihat terdiri atas berkas-berkas cahaya yang dipantulkan oleh benda-benda di sekitarnya. Jadi secara alamiah, fungsi intensitas cahaya merupakan fungsi sumber cahaya yang menerangi objek, serta jumlah cahaya yang dipantulkan oleh objek, atau ditulis [6]:

$$f(x, y) = i(x, y) \cdot r(x, y) \quad (2.1)$$

di mana : $0 < i(x, y) < \infty$ (iluminasi sumber cahaya)

$$0 < r(x, y) < 1 \quad (\text{koefisien pantul objek})$$

Fungsi intensitas f pada suatu titik (x, y) disebut derajat keabuan atau *gray level* (l), dengan terletak di antara:

$$L_{min} \leq l \leq L_{max}$$

Dengan demikian:

$$L_{min} = i_{min} \cdot r_{min}$$

$$L_{max} = i_{max} \cdot r_{max}$$

Selang (L_{min}, L_{max}) sering disebut sebagai skala keabuan. Pada representasi suatu citra hitam putih secara numerik, biasanya selang digeser menjadi : $(0, L)$, dengan 0 menyatakan hitam dan L menyatakan putih. Semua bilangan yang terletak di antara 0 dan L merupakan derajat keabuan.

2.3.1 Pencuplikan Citra dan Kuantisasi Derajat Keabuan

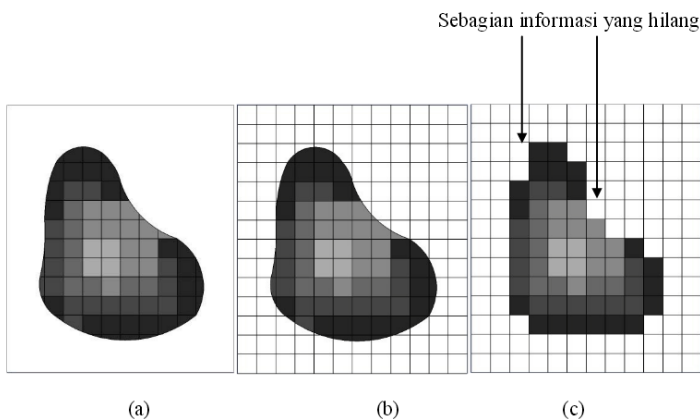
Suatu citra agar dapat direpresentasikan secara numerik, maka citra harus didigitalisasi, baik terhadap ruang (koordinat (x, y)) maupun terhadap skala keabuannya ($f(x, y)$). Proses digitalisasi koordinat (x, y) dikenal sebagai pencuplikan citra (*image sampling*), sedangkan proses digitalisasi skala keabuan ($f(x, y)$) disebut sebagai kuantisasi derajat keabuan (*gray level quantization*).

Pencuplikan merupakan proses pengambilan informasi dari citra analog yang memiliki panjang dan lebar tertentu untuk membaginya ke beberapa blok kecil. Blok-blok tersebut disebut sebagai piksel. Kuantisasi adalah proses pemberian nilai derajat keabuan di setiap titik piksel yang merupakan representasi dari warna asli dari citra analog dengan rentang nilai keabuan adalah 0 – 255. Hasil pencuplikan dan kuantisasi adalah sebuah matriks bilangan real.

Sebuah citra ($f(x, y)$) akan didekati oleh cuplikan-cuplikan yang seragam jaraknya dalam bentuk matriks $M \times N$ di mana indeks baris dan kolomnya menyatakan suatu titik

pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar atau piksel) menyatakan tingkat keabuan pada titik tersebut. Untuk sebuah citra digital, setiap piksel memiliki nilai integer yakni *gray level* yang menunjukkan amplitudo atau intensitas dari piksel tersebut. Citra merupakan fungsi dua dimensi yang kedua variabelnya yaitu nilai amplitudo dan koordinatnya merupakan nilai integer. Notasi matriks citra digital dapat dinyatakan sebagai berikut [18]:

$$f(x, y) = \begin{bmatrix} f(0, 0) & \dots & f(0, N - 1) \\ \vdots & \ddots & \vdots \\ f(M - 1, 0) & \dots & f(M - 1, N - 1) \end{bmatrix} \quad (2.2)$$



Gambar 2.2. Proses *sampling* dan kuantisasi. (a) Citra Digital, (b) Citra Digital Disampling Menjadi 14 Baris dan 12 Kolom, (c) Citra Digital Hasil Sampling Berukuran 14 x 12 Piksel [18]

2.3.2 Citra Berwarna dan Citra Grayscale

Citra berwarna atau citra RGB (*Red-Green-Blue*) merupakan warna dasar yang dapat diterima oleh mata manusia. Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari ketiga warna dasar RGB. Setiap titik pada citra warna membutuhkan data sebesar 3 byte.

Setiap warna dasar memiliki intensitas tersendiri dengan nilai minimum nol (0) dan nilai maksimum 255 (8 bit). RGB didasarkan pada teori bahwa mata manusia peka terhadap panjang gelombang 630nm (merah), 530 nm (hijau), dan 450 nm (biru).

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, artinya nilai dari $Red = Green = Blue$. Nilai-nilai tersebut digunakan untuk menunjukkan intensitas warna. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* berbeda dengan citra "hitam-putih", dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu "hitam" dan "putih" saja. Pada citra grayscale warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra *grayscale* seringkali merupakan perhitungan dari intensitas cahaya pada setiap piksel pada spektrum elektromagnetik *single band*. Citra *grayscale* disimpan dalam format 8 bit untuk setiap sample piksel, yang memungkinkan sebanyak 256 intensitas. Untuk mengubah RGB menjadi *grayscale* menggunakan persamaan berikut ini:

$$gray = \alpha(Red) + \beta(Green) + \gamma(Blue), \quad (2.3)$$

dengan $\alpha + \beta + \gamma = 1$

Pada dasarnya resolusi citra secara matematis direpresentasikan dengan matriks dimana setiap nilai elemen matriksnya merupakan nilai dari intensitas warna baik itu dalam RGB atau dalam *grayscale*.

2.3.3 Thresholding

Proses *threshold* akan menghasilkan citra biner, yaitu citra yang memiliki dua nilai tingkat keabuan (hitam dan putih). Secara umum proses pengembangan citra *grayscale* untuk menghasilkan citra biner adalah sebagai berikut: [17]

$$g(x, y) = \begin{cases} 1 & , f(x, y) \geq T \\ 0 & , f(x, y) < T \end{cases} \quad (2.4)$$

dengan $g(x,y)$ adalah citra biner dari citra *grayscale* $f(x,y)$, dan T menyatakan nilai ambang (*threshold*). Nilai T memegang peranan yang sangat penting dalam proses *thresholding*. Kualitas citra biner sangat tergantung pada nilai T yang digunakan [17]. Berdasarkan persamaan 2.1 di atas, proses *thresholding* akan merubah piksel dengan nilai di atas nilai *threshold* akan menjadi piksel putih (nilai=1) dan merubah piksel dengan nilai di bawah nilai *threshold* menjadi piksel hitam (nilai=0).

Dalam penelitian ini, proses *thresholding* digunakan untuk mensegmentasi atau memisahkan daerah-daerah dalam citra yang menjadi objek penelitian dari daerah-daerah yang tidak diperlukan. Daerah yang disegmentasi dalam citra sel darah merah tunggal adalah daerah sel darah merah, daerah akromia sentral sel darah merah, dan *background*. Nilai ambang T (*threshold*) dihitung secara otomatis berdasarkan citra masukan digunakan Metode Otsu. Pendekatan yang digunakan oleh metode otsu adalah dengan melakukan analisis diskriminan, yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis diskriminan akan memaksimumkan variabel tersebut agar dapat memisahkan obyek dengan latar belakang [17]. Nilai ambang yang akan dicari dinyatakan dengan k . Nilai k berkisar antara 1 sampai L , dengan $L=255$.

Probabilitas untuk piksel i dinyatakan dengan:

$$p_i = \frac{n_i}{N} \quad (2.5)$$

Keterangan : n_i = jumlah piksel dengan tingkat keabuan L
 N = banyaknya piksel pada citra.

2.3.4 Operasi Morphologi

Pengolahan citra morfologi dikenal sebagai sebuah teknik untuk analisis struktur spasial yang bertujuan untuk mengekstraksi atau memodifikasi informasi tentang bentuk dan struktur objek di dalam gambar dan juga untuk menganalisa bentuk dan format objek pada citra biner dan citra *grayscale*. Operasi morfologi dilakukan dengan membandingkan objek yang diteliti dengan sebuah matriks yang disebut dengan struktur elemen. Matriks yang digunakan pada struktur elemen dapat menyerupai cakram, garis, lingkaran, segi enam, dan lain-lain tergantung pada kebutuhan pengguna dalam proses pengolahan citra digital. Operasi-operasi pada matematika morfologi dilakukan dengan tujuan untuk identifikasi objek, eliminasi kebisingan citra, dan kebutuhan lainnya [18].

Di dalam metode matematika morfologi citra biner, terdapat 4 buah operasi utama, yaitu: dilasi, erosi, *opening*, dan *closing*, yaitu:

a. Dilasi

Dilasi adalah transformasi morfologi yang mengombinasikan dua himpunan dengan menggunakan penambahan vector dari elemen-elemen himpunan. Jika A dan B adalah himpunan dalam ruang Euclide dimensi $N(E^N)$ dengan A dan B masing-masing mempunyai elemen $a = (a_1, a_2, \dots, a_n)$ dan $b = (b_1, b_2, \dots, b_n)$, maka dilasi dari A dan B adalah himpunan dari semua kemungkinan jumlah vector sepasang elemen yang satu berasal dari A dan yang lainnya dari B .

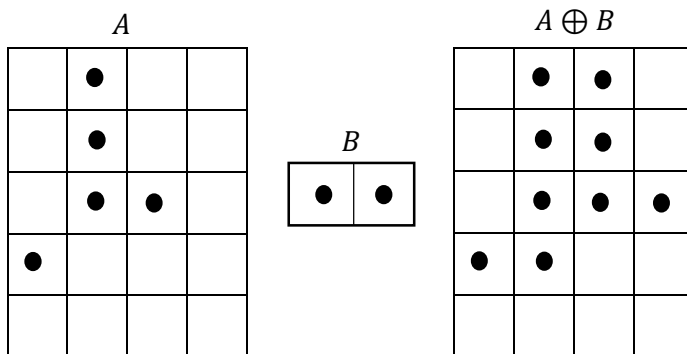
Dilasi dari A dan B ditulis $A \oplus B$ dan didefinisikan dengan persamaan 2.6.

$$A \oplus B = \{c \in E^N \mid c = a + b, \forall a \in A \text{ dan } b \in B\} \quad (2.6)$$

Sebagai contoh dari proses dilasi, misalkan A adalah matriks dari sebuah citra digital biner, dan B merupakan sebuah matriks struktur elemen yang akan digunakan pada proses dilasi matematika morfologi.

$$\begin{aligned}
 A &= \{(0,1), (1,1), (2,1), (2,2), (3,0)\} \\
 B &= \{(0,0), (0,1)\} \\
 A \oplus B &= \{(0,1), (1,1), (2,1), (2,2), (3,0), (0,2), \\
 &\quad (2,2), (2,3), (3,1)\}
 \end{aligned}$$

Maka proses dilasi citra dapat diilustrasikan seperti pada Gambar 2.3



Gambar 2.3 Contoh Proses Dilasi pada Citra Biner

b. Erosi

Erosi adalah morfologi dual dari dilasi. Erosi adalah transformasi morfologi yang mengombinasikan dua himpunan dengan menggunakan operasi penghilangan elemen-elemen himpunan. Jika A dan B adalah himpunan dalam ruang Euclidean dimensi $N(E^N)$, maka erosi A dan B adalah himpunan semua elemen x dimana $x + b \in A$ untuk setiap $b \in B$.

Erosi adalah A oleh B ditulis $A \ominus B$ dan didefinisikan dengan persamaan 2.7.

$$A \ominus B = \{x \in E^N \mid x + b \in A, \forall b \in B\} \quad (2.7)$$

Sebagai contoh dari proses dilasi, misalkan A adalah matriks dari sebuah citra digital biner, dan B merupakan

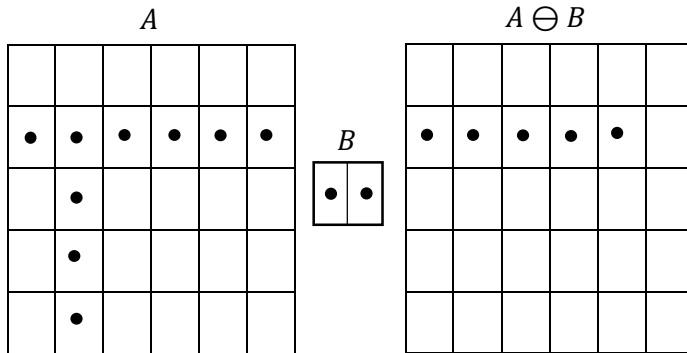
sebuah matriks struktur elemen yang akan digunakan pada proses dilasi matematika morfologi.

$$A = \{(1,0), (1,1), (1,2), (1,3), (1,4), (1,5), (2,1), (3,1), (4,1), (5,1)\}$$

$$B = \{(0,0), (0,1)\}$$

$$A \ominus B = \{(1,0), (1,1), (1,2), (1,3), (1,4)\}$$

Maka proses erosi citra dapat diilustrasikan seperti pada Gambar 2.4



Gambar 2.4 Contoh Proses Erosi pada Citra Biner

c. *Opening* dan *Closing*

Dalam penggunaannya, dilasi dan erosi biasanya digunakan secara berpasangan, baik dilasi dari suatu citra dilanjutkan erosi dari hasil dilasi atau erosi citra dilanjutkan dilasi. Operator *opening* pada citra dengan elemen struktur *disk* berguna untuk menghaluskan kontur, memutuskan panah atau garis yang perlu dan menghilangkan daerah atau sudut yang tajam. Sedangkan *closing* pada citra dengan elemen struktur *disk* berguna untuk menghaluskan kontur, menghubungkan garis-garis yang patah dan garis tipis yang panjang, menghilangkan lubang-lubang kecil dan mengisi kekosongan pada kontur.

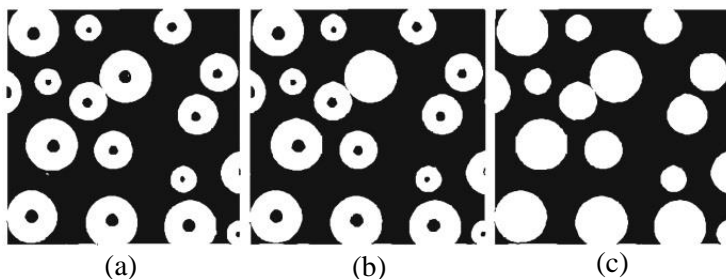
Secara definitif *opening* adalah operasi erosi yang diikuti oleh dilasi dengan menggunakan elemen penstruktur yang sama. *Opening* citra B oleh elemen struktur K ditulis $B \circ K$ dan didefinisikan pada persamaan 2.8.

$$B \circ K = (B \ominus K) \oplus K \quad (2.8)$$

Sedangkan *closing* adalah operasi dilasi yang diikuti oleh erosi dengan menggunakan elemen penstruktur yang sama. *Closing* dari citra B oleh elemen struktur K ditulis $B \cdot K$ dan didefinisikan pada persamaan 2.9.

$$B \cdot K = (B \oplus K) \ominus K \quad (2.9)$$

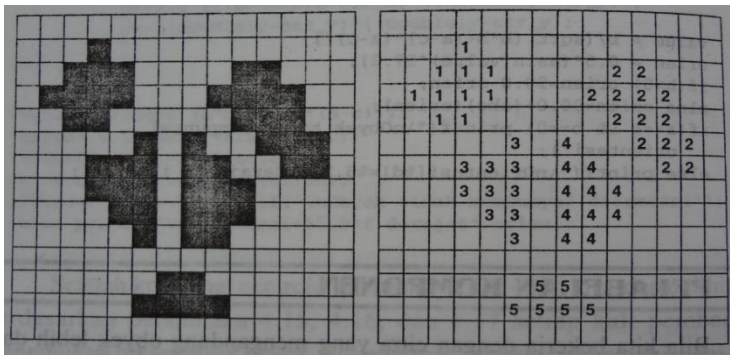
Operasi morfologi yang digunakan dalam tugas akhir ini adalah membersihkan border obyek (*clear border*) dan *filling holes*. Aplikasi rekonstruksi morfologi *clear border* ini untuk membersihkan border obyek digunakan untuk menghilangkan noise-noise kecil yang menyentuh border citra. Kunci dalam aplikasi ini adalah memilih marker dan mask citra yang tepat untuk mendapatkan efek yang diharapkan [17]. Sedangkan, *filling holes* ini digunakan untuk mengisi bagian tengah yang berlubang. Sebuah lubang (*holes*) didefinisikan sebagai daerah *background* yang dikelilingi oleh batas piksel *foreground* yang terhubung [19].



Gambar 2.5 (a) Citra biner, (b) Citra dengan *filling holes* sebagian (c) Citra dengan *filling holes* keseluruhan

2.3.5 Pelabelan Komponen

Salah satu operasi yang umum dan penting untuk dilakukan dalam pengolahan citra adalah menemukan komponen terkoneksi dalam citra. Hal ini didasarkan pada kenyataan bahwa suatu komponen terkoneksi dapat merupakan bagian yang mewakili sebuah obyek dalam citra yang mempunyai obyek lebih dari satu. Dengan memeriksa koneksitas dari suatu kumpulan piksel, maka kumpulan piksel ini dapat dianggap sebagai obyek tunggal, bila ia tidak lagi tersambung dengan piksel atau kumpulan piksel lainnya dalam citra biner [14].



Gambar 2.6 Sebuah citra biner dan urutan pemberian label pada obyek-obyeknya [14].

Dengan pelabelan ini ciri-ciri mendasar dari masing-masing obyek dapat dihitung secara sendiri-sendiri. Dalam penelitian ini, pelabelan komponen digunakan untuk mencari atau mengidentifikasi sel darah merah yang berbentuk pensil.

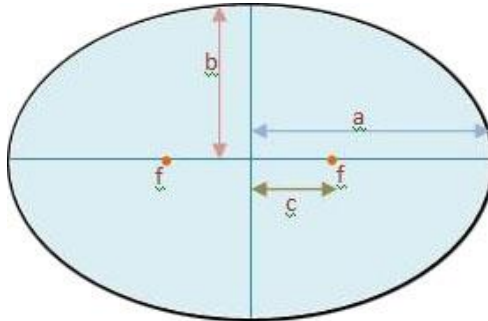
2.3.6 Ekstraksi Fitur

Ekstraksi fitur (*feature extraction*) merupakan bagian fundamental dari analisis citra. Fitur adalah karakteristik unik dari suatu obyek [17]. Karakteristik fitur yang baik sebisa mungkin memenuhi persyaratan berikut:

1. Dapat membedakan suatu objek dengan yang lainnya (*discrimination*).
2. Memperhatikan kompleksitas komputasi dalam memperoleh fitur. Kompleksitas komputasi yang tinggi akan menjadi beban tersendiri dalam menemukan suatu fitur.
3. Tidak terikat (*independence*), dalam arti bersifat invarian terhadap berbagai transformasi.
4. Jumlahnya sedikit, karena fitur yang jumlahnya sedikit akan dapat menghemat waktu komputasi dan ruang penyimpanan untuk proses selanjutnya (proses pemanfaatan fitur).

2.3.7 Eccentricity

Eccentricity adalah rasio jarak antara titik pusat suatu obyek elips dengan panjang sumbu utamanya.



Gambar 2.7 Geometri Elips

Hubungan-hubungan yang berlaku di antara besaran-besaran di atas adalah:

$$e = \frac{c}{a} \quad (2.10)$$

dengan e adalah eksentrisitas elips (*Eccentricity*), a adalah setengah sumbu panjang elips (semi-major axis), c adalah jarak fokus elips (focal length) dan f adalah titik fokus

elips (foci). Elips yang memiliki nilai eksentrisitas 0 sebenarnya adalah lingkaran, sementara elips yang memiliki nilai eksentrisitas 1 adalah segmen garis.

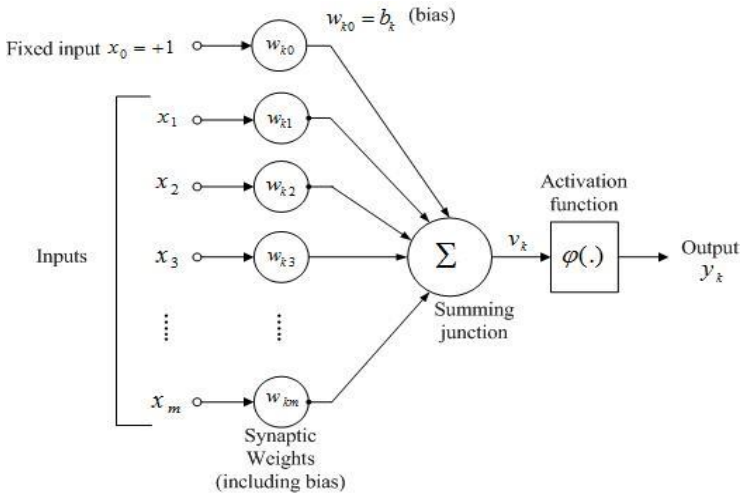
Eccentricity dalam tugas akhir ini digunakan sebagai masukan jaringan saraf tiruan Multi Perceptron untuk mengidentifikasi plasmodium falciparum dengan cara mencari sel darah merah yang berbentuk stadium gametocyte (pipih).

2.4 Jaringan Saraf Tiruan

Jaringan syaraf tiruan adalah salah satu cabang dari *machine learning*. JST merupakan teknik yang digunakan untuk membangun program yang cerdas dengan pemodelan yang mensimulasikan cara kerja jaringan syaraf pada otak manusia. Jadi, JST menggunakan konsep kerja dari syaraf otak manusia untuk menyelesaikan perhitungan pada komputer. JST menyerupai otak manusia dalam dua hal, yaitu:

1. Pengetahuan diperoleh jaringan melalui proses belajar.
2. Kekuatan hubungan antar sel syaraf (neuron) yang dikenal sebagai bobot-bobot sinaptik digunakan untuk menyimpan pengetahuan.

Seperti otak manusia, fungsi dari jaringan ditentukan oleh hubungan antar neuron. Hubungan antar neuron ini disebut bobot (*weight*). Untuk mendapatkan fungsi tertentu dapat dilakukan dengan pelatihan (*training*) dengan menyesuaikan nilai bobot dari masing-masing neuron. Satu sel syaraf terdiri dari tiga bagian, yaitu fungsi penjumlah (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*). Pada **Gambar 2.8.** dapat dilihat model dari sel syaraf (neuron) [10].



Gambar 2.8. Sel Syaraf (*Neuron*)

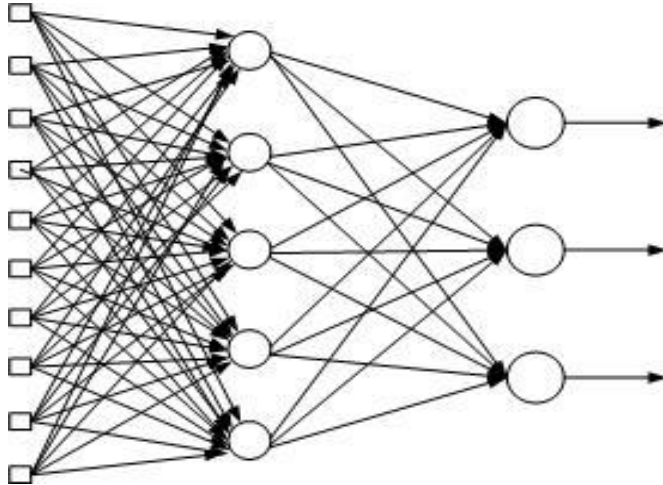
Pada umumnya, JST dilatih (*trained*) agar *input* mengarah ke target *output* yang spesifik. Jadi jaringan dilatih terus menerus hingga mencapai kondisi di mana *input* sesuai dengan target yang telah ditentukan. Pelatihan di mana setiap *input* diasosiasikan dengan target yang telah ditentukan disebut pelatihan terarah (*supervised learning*).

Secara umum, terdapat dua jenis arsitektur jaringan syaraf tiruan, yaitu:

1) *Feedforward Neural Networks*

Sebuah jaringan yang sederhana mempunyai struktur *feedforward* di mana sinyal bergerak dari *input* kemudian melewati lapisan tersembunyi dan akhirnya mencapai unit *output*. Tipe jaringan *feedforward* mempunyai sel syaraf yang tersusun dari beberapa lapisan. Lapisan *input* memberikan pelayanan dengan mengenalkan suatu nilai dari suatu variabel. Lapisan tersembunyi dan lapisan *output* sel syaraf terhubung satu sama lain dengan lapisan sebelumnya. Kemungkinan yang timbul adalah adanya

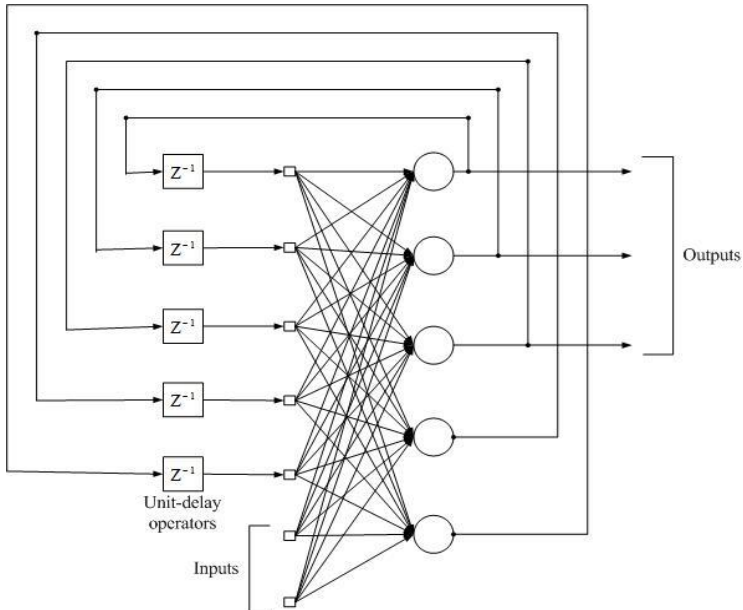
hubungan dengan beberapa unit dari lapisan sebelumnya atau terhubung semuanya dengan baik. Pada **Gambar 2.9.** dapat dilihat arsitektur dari *feedforward neural networks* [12].



Gambar 2.9. *Feedforward Neural Networks*

2) *Recurrent Neural Networks*

Recurrent neural networks adalah jaringan yang mempunyai minimal satu *feedback loop*. Sebagai contoh, suatu *recurrent neural networks* bisa terdiri dari satu lapisan neuron tunggal dengan masing-masing neuron memberikan kembali *output*-nya sebagai *input* pada semua neuron yang lain. Pada **Gambar 2.10.** dapat dilihat arsitektur dari *Recurrent Neural Networks* [12].



Gambar 2.10. *Recurrent Neural Networks*

Untuk mendapatkan tingkat kecerdasan yang diinginkan maka jaringan syaraf tiruan harus melalui proses pembelajaran. Pembelajaran (*learning*) adalah proses yang melibatkan serangkaian nilai *input* menjadi *input* jaringan secara berurutan dan bobot jaringan disesuaikan sehingga akan diperoleh nilai yang sama dengan nilai *output*-nya. Jaringan syaraf tiruan membagi metode belajar menjadi dua macam, yaitu :

1) Pembelajaran terawasi (*supervised learning*)

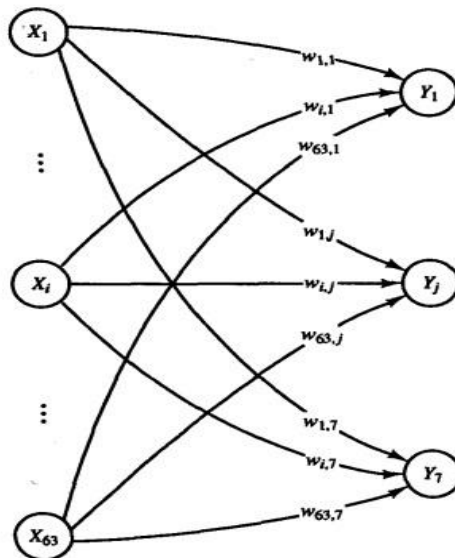
Metode belajar ini memerlukan pengawasan dari luar atau pelabelan data sampel yang digunakan dalam proses belajar. Jaringan belajar dari sekumpulan pola masukan dan keluaran sehingga pada saat pelatihan diperlukan pola yang terdiri dari vektor masukan dan vektor target yang diinginkan. Vektor masukan dimasukkan ke dalam jaringan yang kemudian menghasilkan vektor keluaran

yang selanjutnya dibandingkan dengan vektor target. Selisih kedua vektor tersebut menghasilkan galat (*error*) yang digunakan sebagai dasar untuk mengubah matriks koneksi sedemikian rupa sehingga galat semakin mengecil pada siklus berikutnya.

- 2) Pembelajaran tak terawasi (*unsupervised learning*)
Metode belajar ini menggunakan data yang tidak diberi label dan tidak memerlukan pengawasan dari luar. Data disajikan kepada JST dan membentuk kluster internal yang mereduksi data masukan ke dalam kategori klasifikasi tertentu. Tujuan dari pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu.

2.5 Jaringan Multi Perceptron

Multi Perceptron termasuk salah satu bentuk jaringan saraf yang sederhana.



Gambar 2.11. Arsitektur Multi Perceptron [4]

Multi Perceptron biasanya digunakan untuk mengklasifikasikan suatu tipe pola tertentu yang sering dikenal dengan pemisahan secara linier. Pada dasarnya, Multi Perceptron pada jaringan saraf dengan suatu lapisan memiliki bobot yang bisa diatur. Algoritma yang digunakan oleh aturan Multi Perceptron ini mengatur parameter-parameter bebasnya melalui proses pembelajaran.

Multi Perceptron sebagai cikal bakal jaringan saraf tiruan bekerja dengan cara menerima sejumlah (n) masukan berupa vektor $(x_1, x_2, x_3, \dots, x_n)$ dan menghasilkan sebuah keluaran dengan nilai 1, 0 dan -1.

Bentuk jaringan Multi Perceptron adalah sesuai **Gambar 2.11**. Secara sistematis pemrosesan informasi pada Multi Perceptron dirumuskan oleh persamaan fungsi jumlah berikut:

$$y_{in_j} = b_j + \sum_{i=1}^n x_i w_{ij} \quad (2.11)$$

dengan $j = 1, \dots, m$.

Keterangan : b_j = bias pada unit

x_i = unit input

w_{ij} = vektor bobot

y_{in_j} = fungsi jumlah pada unit output

dengan dikenakan persamaan fungsi aktivasi *step bipolar* sebagai berikut:

$$f(y_{in}) = \begin{cases} 1 & , \text{jika } y_{in_j} > \theta \\ 0, & , \text{jika } -\theta \leq y_{in_j} \leq \theta \\ -1, & , \text{jika } y_{in_j} < -\theta \end{cases} \quad (2.12)$$

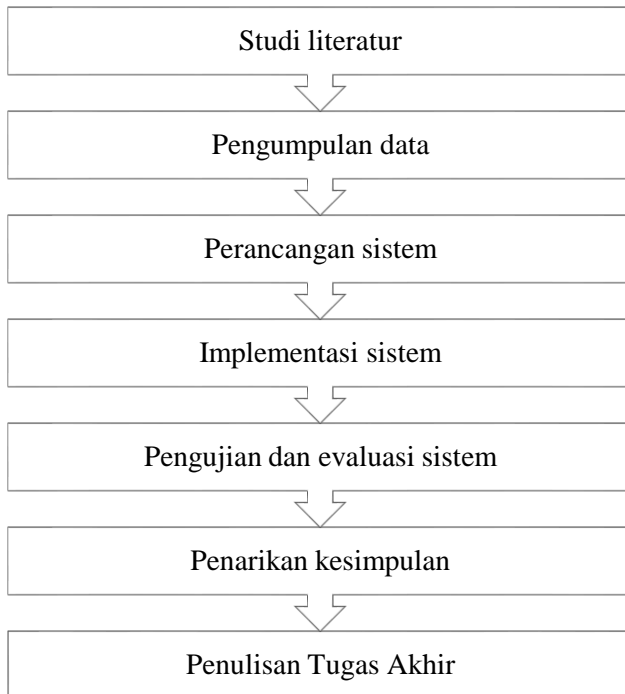
Multi Perceptron harus melalui tahap pembelajaran. Tahap pembelajaran dilakukan dengan cara mengubah-ubah nilai bobot sampai pada kondisi adaptasi yang diinginkan [4].

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang tahap pengerjaan dalam menyelesaikan Tugas Akhir ini sehingga penelitian ini dapat dirancang sistematis dan diatur dengan sebaik-baiknya.

3.1 Diagram Metodologi

Gambaran tahap-tahap dalam penelitian pada Tugas Akhir ini disajikan sebagai diagram alir pada **Gambar 3.1**.



Gambar 3.1. Diagram Alir Metodologi Penelitian

3.2 Studi Literatur

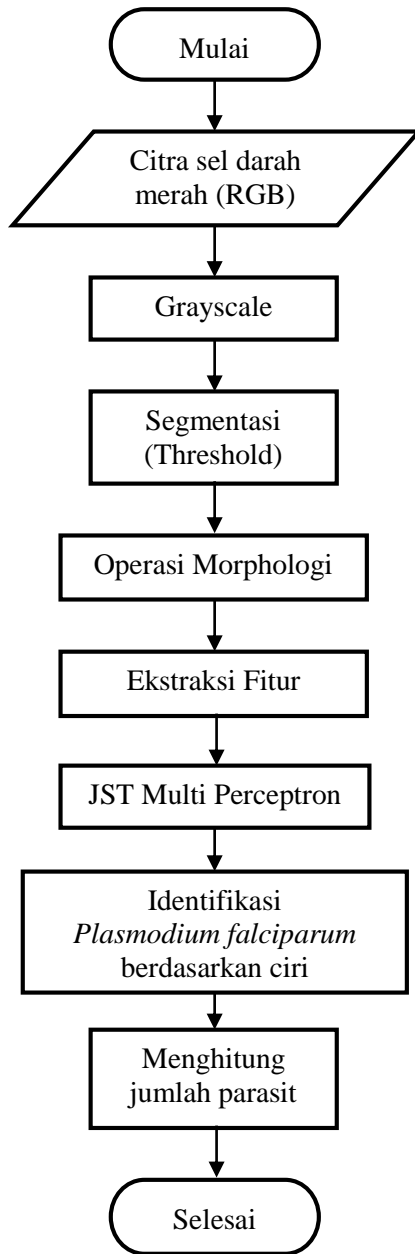
Pada tahap ini meliputi identifikasi permasalahan dan mencari referensi yang menunjang penelitian. Referensi yang dipakai adalah buku-buku literatur, jurnal ilmiah, tugas akhir atau thesis yang berkaitan dengan permasalahan, maupun artikel dari internet.

3.3 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan citra digital yang mengandung objek-objek sesuai batasan masalah dan dibagi menjadi dua kelompok yaitu sebagai data *training* dan data *testing*. Data citra darah yang digunakan pada penelitian ini merupakan hasil citra mikroskopis digital dari hapusan darah tepi yang didapat dari Laboratorium Perawatan Koya Barat Distrik Muara Tami Kota Jayapura Propinsi Papua.

3.4 Perancangan Sistem

Pada tahap ini dilakukan perancangan desain arsitektur jaringan saraf tiruan menggunakan model *Multi Perceptron* yang dijadikan acuan untuk implementasi sistem pada Tugas Akhir ini. Diagram alur rancangan sistem dapat dilihat pada **Gambar 3.2**.



Gambar 3.2. Diagram Alir Rancangan Sistem

3.5 Implementasi Sistem

Pada tahap ini dilakukan implementasi sistem dengan menggunakan perangkat lunak MATLAB. Input sistem berupa citra sel darah merah sesuai batasan masalah. Output sistem berupa hasil identifikasi stadium *Plasmodium falciparum* dan jumlah *Plasmodium falciparum* yang terkandung dalam citra sel darah merah. Sebelum diproses oleh JST *Multi Perceptron*, citra input diolah di dalam praproses yaitu proses *cropping* untuk memfokuskan objek yang akan dikenali. Setelah itu, citra dikonversi menjadi citra *gray scale* dan dilakukan proses pengolahan citra digital. Kemudian dilakukan proses JST *Multi Perceptron* berupa *training* dan dilanjutkan dengan proses *testing*.

3.6 Pengujian dan Evaluasi Sistem

Sistem yang telah dibuat dan melalui langkah-langkah di atas, selanjutnya akan melalui tahap uji coba dan evaluasi. Dalam tahap ini akan diujicobakan beberapa citra yang terjangkit parasit *Plasmodium falciparum*. Proses ini untuk membantu memahami cara penggunaan simulasi program yang telah dibuat, menganalisis, dan evaluasi sehingga dapat memberikan saran untuk pengembangan penelitian selanjutnya. Evaluasi juga dilakukan untuk mendapatkan sebuah hasil yang diinginkan.

3.7 Penarikan Kesimpulan

Setelah sistem selesai diimplementasikan dan dilakukan pengujian serta evaluasi, maka tahap selanjutnya adalah penarikan kesimpulan dari keseluruhan tahap yang sudah dilakukan dan pemberian saran terkait kekurangan hasil penelitian untuk pengembangan pengenalan citra digital berikutnya.

3.8 Penulisan Laporan Tugas Akhir

Tahap akhir setelah semua tahap dijalankan adalah penulisan laporan Tugas Akhir. Konten laporan Tugas Akhir sesuai dengan hasil yang telah didapatkan dari proses awal pengumpulan data sampai dengan penarikan kesimpulan.

“Halaman ini sengaja dikosongkan”

BAB IV PERANCANGAN SISTEM

Bab ini menjelaskan rancangan desain sistem yang digunakan sebagai acuan untuk implementasi sistem. Desain sistem menggambarkan proses rancang bangun secara terperinci dari awal tahap pengumpulan data hingga proses konstruksi sistem menggunakan metode Jaringan Saraf Tiruan model *Multi Perceptron*, serta penjelasan mengenai cara untuk mendapatkan data keluaran yang sesuai dengan tujuan dari penelitian Tugas Akhir ini.

4.1 Pengumpulan Dataset

Pengumpulan data sampel penelitian meliputi perolehan sel darah merah normal dan sel darah merah terinfeksi *plasmodium falciparum* yang telah diuji laboratorium. Citra sel darah merah normal dan sampel sel darah merah terinfeksi *plasmodium falciparum* bersumber dari Laboratorium Perawatan Koya Barat Distrik Muara Tami Kota Jayapura Propinsi Papua.

Data yang didapatkan adalah hasil *capture* preparat hapusan darah menggunakan mikroskop digital yang langsung terhubung ke komputer/laptop. Data tersebut nantinya akan dibagi ke dalam dua kelompok yaitu data *training* dan data *testing*.

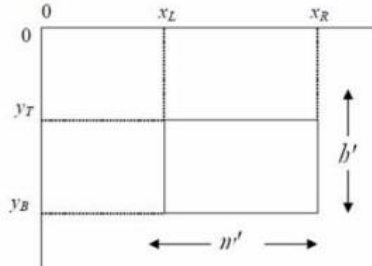
4.2 Pengolahan Data pada Proses *Training*

Citra digital preparat sel darah merah yang diperoleh dari hasil *capture* di *crop* sehingga didapatkan citra sel darah merah tunggal. Proses *cropping* adalah memotong satu bagian dari citra sehingga diperoleh citra yang berukuran lebih kecil. Proses ini bertujuan untuk memisahkan objek yang satu dengan objek yang lain dalam suatu gambar untuk mempercepat proses selanjutnya. Rumus yang digunakan untuk menggunakan operasi ini adalah [15] :

$$x' = x - x_L \quad \text{untuk } x = x_L \text{ sampai } x_R \quad (4.1)$$

$$y' = y - y_L \quad \text{untuk } y = y_r \text{ sampai } y_B \quad (4.2)$$

Pada Gambar 4.1 tampak (x_L, y_r) dan (x_R, y_B) masing-masing adalah koordinat titik pojok kiri dan pojok kanan bawah bagian citra yang hendak dicrop.



Gambar 4.1. Koordinat titik pojok bagian citra yang dicrop

Keterangan : h' (*high*) = tinggi citra.

w' (*width*) = lebar citra.

Ukuran citra berubah menjadi :

$$w' = x_R - x_L \quad (4.3)$$

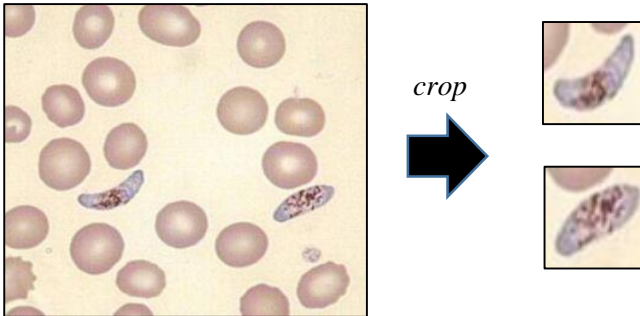
$$b' = y_B - y_r \quad (4.4)$$

dan transformasi baliknya adalah :

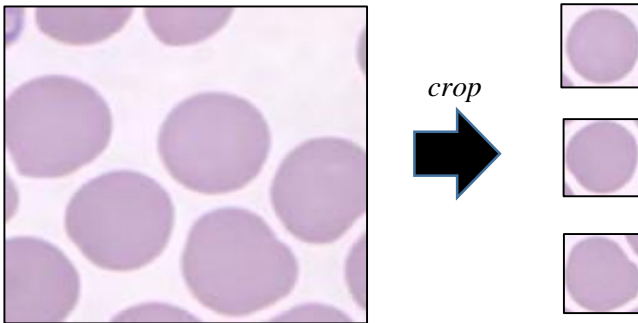
$$x' = x - x_L \quad \text{untuk } x' = 0 \text{ sampai } w' - 1 \quad (4.5)$$

$$y' = y - y_L \quad \text{untuk } y' = 0 \text{ sampai } b' - 1 \quad (4.6)$$

Citra sampel yang digunakan meliputi citra sel darah merah tunggal yang dikelompokkan menjadi 2 kelompok, yakni sel darah terinfeksi *plasmodium falciparum* seperti pada **Gambar 4.2** dan sel darah normal atau tidak terinfeksi *plasmodium falciparum* seperti pada **Gambar 4.3**.



Gambar 4.2. Sel darah terinfeksi *plasmodium falciparum*



Gambar 4.3. Sel darah normal

4.2.1 Pre-Processing

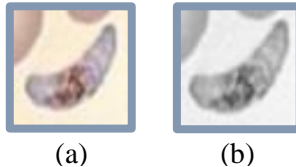
Pre-processing ini meliputi proses *greyscale* menggunakan MATLAB yang bertujuan untuk mengubah citra RGB (*Red, Green, Blue*) menjadi citra yang memiliki derajat keabuan 0-255.

Citra *greyscale* setiap pikselnya mempunyai warna gradasi mulai dari putih sampai hitam. Rentang tersebut berarti bahwa setiap piksel dapat diwakili oleh 8 bit, atau 1 byte. Dengan demikian maka persamaannya dapat dituliskan sebagai berikut :

$$I_{BW}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B}{3} \quad (4.7)$$

Keterangan : $I_R(x, y)$ = nilai piksel *Red* titik (x, y)
 $I_G(x, y)$ = nilai piksel *Green* titik (x, y)
 $I_B(x, y)$ = nilai piksel *Blue* titik (x, y)
 $I_{BW}(x, y)$ = nilai piksel *grayscale* titik (x, y)

Data yang diolah pada proses *grayscale* ini adalah data sel darah merah tunggal hasil *crop*.



Gambar 4.4. (a) Citra RGB, (b) Citra *greyscale*

4.2.2 Segmentasi

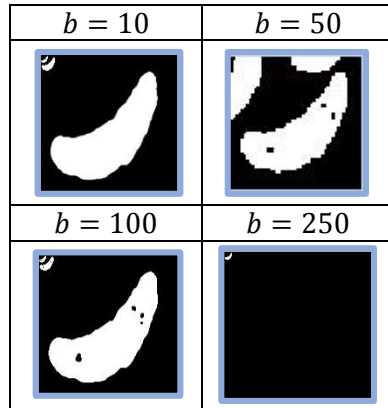
Proses selanjutnya adalah citra disegmentasi untuk memisahkan sel darah merah dengan *background* citra. Segmentasi dilakukan dengan proses *thresholding*. *Thresholding* digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Proses *thresholding* ini pada dasarnya adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan *thresholding* dengan derajat keabuan dapat digunakan rumus [15]:

$$x = \frac{w}{b} \quad \text{dengan } b = 0,1,2, \dots, 255 \quad (4.8)$$

Keterangan : w = nilai derajat keabuan sebelum *thresholding*
 x = nilai derajat keabuan setelah *thresholding*

dengan menggunakan *thresholding* maka derajat keabuan bisa diubah sesuai keinginan dengan tetap menyesuaikan kondisi dari jenis citra input, pada tugas akhir ini digunakan

derajat keabuan $b = 50$ karena nilai tersebut yang paling sesuai untuk mengenali citra input. Hal ini tampak pada Gambar 4.5 berikut :



Gambar 4.5. Pemberian Nilai *Thresholding*

Adanya pemberian nilai ambang (*threshold*) tertentu dapat menghasilkan citra biner, yaitu citra yang memiliki nilai 0 (hitam) dan 1 sampai 255 (putih). *Binary image* merupakan hasil pengolahan dari *black and white image*, dengan menggunakan fungsi sebagai berikut :

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 255 & I_{BW}(x, y) \geq T \end{cases} \quad (4.9)$$

dan dalam bentuk *floating point* sebagai berikut:

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 1 & I_{BW}(x, y) \geq T \end{cases} \quad (4.10)$$

Keterangan : $I_{BW}(x, y)$ = nilai piksel *Gray* titik (x, y)
 $I_{Bin}(x, y)$ = nilai piksel *Binary* titik (x, y)
 T = nilai *Threshold*

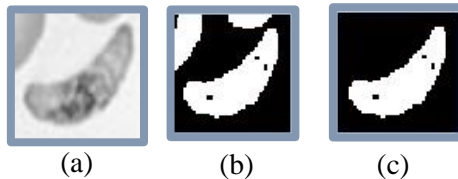
Citra biner hasil proses segmentasi terlihat pada **Gambar 4.6**.



Gambar 4.6. Hasil Proses Segmentasi

4.2.3 Operasi Morphologi

Operasi morfologi yang digunakan dalam penelitian ini adalah operasi *clear border* dan operasi *filling holes*. Operasi *clear border* diperlukan untuk menghilangkan citra yang tidak dibutuhkan yang menempel atau bersinggungan dengan border citra hasil proses *cropping*. Contoh penggunaan *clear border* pada penelitian ini adalah seperti **Gambar 4.7**.



Gambar 4.7. (a) Hasil *cropping*, (b) Hasil Segmentasi, (c) Hasil proses *clear border*

Operasi morfologi selanjutnya adalah operasi *filling holes*. Operasi *filling holes* dibutuhkan untuk melakukan penghitungan ukuran atau area sel darah merah dan area akromia sentral. Sel darah merah dengan hemoglobin rendah memiliki area akromia sentral yang lebih lebar dari normal, sehingga menyebabkan lubang saat segmentasi karena area akromia sentral memiliki warna yang sama dengan background. Oleh karena itu dibutuhkan operasi *filling holes* untuk mengisi lubang tersebut. Berikut algoritma yang digunakan untuk operasi *filling holes* [19] :

1. Misalkan B adalah struktur elemen dari N_1, N_2, N_3 tergantung pada obyek-obyek yang saling terkoneksi.
2. Pilih sebuah piksel p dari warna yang berada di dalam area background.
3. Inisialisasi X_0 untuk dijadikan *array* pada piksel background, kecuali di $X_0[p] = 1$.
4. Gunakan persamaan:

$$X_k = (X_{k-1} \oplus B) \cap A^c \text{ dengan } k = 1, 2, 3, \dots \quad (4.11)$$
 Sampai kondisi $X_k = X_{k-1}$ dimana area *filling holes* adalah $A \cup X_k$

Proses operasi *filling holes* tampak pada **Gambar 4.8**.



Gambar 4.8. (a) Hasil proses clear border
(b) Hasil setelah melalui proses *filling holes*

Proses *filling holes* dilanjutkan dengan perhitungan area sel darah merah dan area akromia sentral, yaitu dengan menghitung jumlah piksel warna putih. Area akromia sentral dapat diketahui dengan mengurangkan hasil perhitungan ukuran sel darah merah dengan jumlah piksel warna putih sebelum proses *filling holes*.

4.2.4 Ekstraksi Fitur

Pada tugas akhir ini, ada 4 nilai fitur yang dipilih berdasarkan penelitian dibidang medis untuk mengidentifikasi sel darah terinfeksi *plasmodium falciparum*, sebagai berikut :

- a. Area sel darah merah, yaitu harga skalar yang menyatakan jumlah keseluruhan piksel sel darah merah setelah *filling holes*.
- b. Area akromia sentral, yaitu harga skalar yang menyatakan jumlah keseluruhan piksel daerah akromia sentral.
- c. Perbandingan, merupakan hasil bagi dari nilai fitur area akromia sentral dan nilai fitur sel darah merah.
- d. Nilai *eccentricity*, digunakan sebagai masukan jaringan saraf tiruan Multi Perceptron untuk mengidentifikasi *gametocyte* dengan cara mencari sel darah merah yang berbentuk pipih.

4.3 Nilai Input pada Proses *Training*

Proses berjalannya sistem secara umum terbagi menjadi dua tahap yaitu proses *training* dan *testing*. Output dari proses *training* ini adalah bobot w dan bias b . Bobot dan bias yang dihasilkan dari proses *training* ini akan digunakan untuk melakukan pengenalan objek pada proses *testing*. Proses *testing* memerlukan input berupa citra dan akan memberikan output berupa hasil pengenalan objek dalam citra. Berikut penjabaran proses berjalannya sistem pada tiap tahap.

4.3.1 Proses Identifikasi *Plasmodium Falciparum*

Fitur yang digunakan untuk pelatihan jaringan Multi Perceptron dalam mengidentifikasi *plasmodium falciparum* adalah nilai-nilai fitur yang didapatkan dari proses pengolahan citra digital yang telah dilakukan, meliputi proses *cropping*, proses *grayscale* yang dilanjutkan dengan proses segmentasi, operasi *clear border*, dan operasi *filling holes*. Nilai-nilai fitur tersebut adalah fitur area sel darah merah, fitur area akromia sentral dan perbandingan fitur ukuran sel darah merah dengan fitur area akromia sentral.

Langkah-langkah mendapatkan nilai-nilai fitur tersebut adalah sebagai berikut :

- a. Nilai fitur jumlah piksel sel darah didapatkan dengan cara menghitung jumlah piksel warna putih setelah dilakukan proses *filling holes*.
- b. Nilai fitur jumlah piksel area akromia sentral didapatkan dengan cara mengurangi jumlah piksel setelah dilakukan proses *filling holes* dengan sebelum dilakukan proses *filling holes*.
- c. Sedangkan perbandingan, merupakan hasil bagi dari nilai fitur jumlah piksel area akromia sentral dan nilai fitur jumlah piksel sel darah merah.

Nilai fitur yang digunakan dalam proses identifikasi *plasmodium falciparum* adalah sebagai berikut :

Tabel 4.1. Tabel nilai fitur dalam proses identifikasi
Sel darah terinfeksi *plasmodium falciparum*

No	Terinfeksi <i>Plasmodium Falciparum</i>		
	Fitur jumlah piksel sel darah	Fitur jumlah piksel area akromia sentral	Perbandingan fitur sel darah dan fitur area akromia sentral
1	743	0	0
2	415	0	0
3	406	6	0.014778325123
4	667	0	0
5	427	0	0
...

(Selengkapnya terdapat pada lampiran A)

Tabel 4.2. Tabel nilai fitur dalam proses identifikasi Sel darah normal/bukan *plasmodium falciparum*

No	Normal/Bukan <i>Plasmodium Falciparum</i>		
	Fitur jumlah piksel sel darah	Fitur jumlah piksel area akromia sentral	Perbandingan fitur sel darah dan fitur area akromia sentral
1	1553	0	0
2	1656	0	0
3	1566	0	0
4	1533	9	0.005870841487
5	1727	60	0.034742327735
...

(Selengkapnya terdapat pada lampiran A)

Nilai-nilai fitur yang berada pada Tabel 4.1 dan Tabel 4.2 akan digunakan sebagai nilai input pada proses *training* dengan metode JST Multi Perceptron. Sedangkan deteksi atau pencarian sel stadium *gametocyte* yang mempunyai ciri berbentuk pipih dilakukan dengan menggunakan JST Multi Perceptron berdasarkan nilai *eccentricity* sel darah merah. Pencarian nilai *eccentricity* tiap sel darah merah membutuhkan operasi pelabelan komponen. Dengan pelabelan komponen ini ciri-ciri mendasar dari masing-masing sel darah merah dapat dihitung.

4.3.2 Proses *Training* dengan Multi Perceptron

Bobot koneksi dari unit assosiator ke unit response (atau *output*) ditentukan melalui pelatihan (*learning rate*) Multi Perceptron dan nilai input. Input (x) yang digunakan untuk proses *training* berasal dari nilai-nilai fitur yang berada pada Tabel 4.1 dan Tabel 4.2.

Untuk setiap *input* training jaringan akan menghitung response dari unit *output*, kemudian jaringan akan menentukan apakah suatu error terjadi pada pola tersebut dengan (4.12) membandingkan *output* hasil perhitungan dengan nilai targetnya. Jaringan tersebut akan membedakan error antara *output* hasil perhitungan dengan target. Dalam kedua kasus tersebut tanda dari error menunjukkan bahwa bobot koneksi harus diubah dalam arah yang dinyatakan oleh nilai target. Namun demikian hanya bobot-bobot pada koneksi dari unit pengiriman sinyal selain 0 ke unit *output* yang akan disesuaikan nilainya, karena hanya sinyal tersebut yang menambah error. Jika error tidak terjadi maka bobot-bobot tersebut tidak akan diubah tetapi sebaliknya jika suatu error terjadi untuk pola *input* pelatihan tertentu, bobot-bobot akan diubah menurut rumus :

$$w_i \text{ (new)} = w_i \text{ (old)} + \alpha t x_i$$

Keterangan :

x_i = *input* ke - i

t = target yang nilainya +1 atau -1

α = kecepatan belajar (*learning rate*) yaitu

$$0 < \alpha \leq 1$$

W = bobot

jika error tidak terjadi, maka bobot-bobot tidak akan berubah.

Adapun algoritma Multi Perceptron [16] adalah sebagai berikut :

Langkah 0 : Inisialisasi:

- a. Bobot input variabel ke-i menuju ke neuron ke-j (w_{ij}) dan bobot bias menuju ke neuron ke-j (b_j):
(untuk sederhananya, set bobot dan bias dengan angka 0)
- b. Set learning rate α ($0 < \alpha \leq 1$)
- c. Set maksimum epoh (MaxEpoh)

Langkah 1 : Tetapkan epoh = 0.

Langkah 2 : Selama kondisi berhenti bernilai *false*, lakukan langkah-langkah berikut ini :

- a. Untuk masing-masing pasangan pembelajaran $s_k - t_k$ dengan $k=1,2,\dots,n$, kerjakan:

- (i). Set input dengan nilai sama dengan vektor *input*:

$$x_{ki} = s_{ki}; \text{ dengan } k=1,2,\dots,m \quad (4.13)$$

- (ii). Hitung respon untuk unit *output* :

$$y_{in_j} = b_j + \sum_{i=1}^m x_i w_{ij} ; \quad (4.14)$$

dengan $j=1,2,\dots,c$

$$f(y_{in}) = \begin{cases} 1 & \text{jika } y_{in} > \theta \\ 0, & \text{jika } -\theta \leq y_{in} \leq \theta \\ -1, & \text{jika } y_{in} < -\theta \end{cases} \text{ atau}$$

- (iii). Perbaiki bobot dan bias, jika terjadi kesalahan (*error*) pada pola ini :

jika $y \neq t_{kj}$, maka:

$$w_{ij} (\text{baru}) = w_{ij} (\text{lama}) + \alpha t x_{kj} \quad (4.15)$$

$$b_j (\text{baru}) = b_j (\text{lama}) + \alpha t_{kj} \quad (4.16)$$

jika tidak, maka:

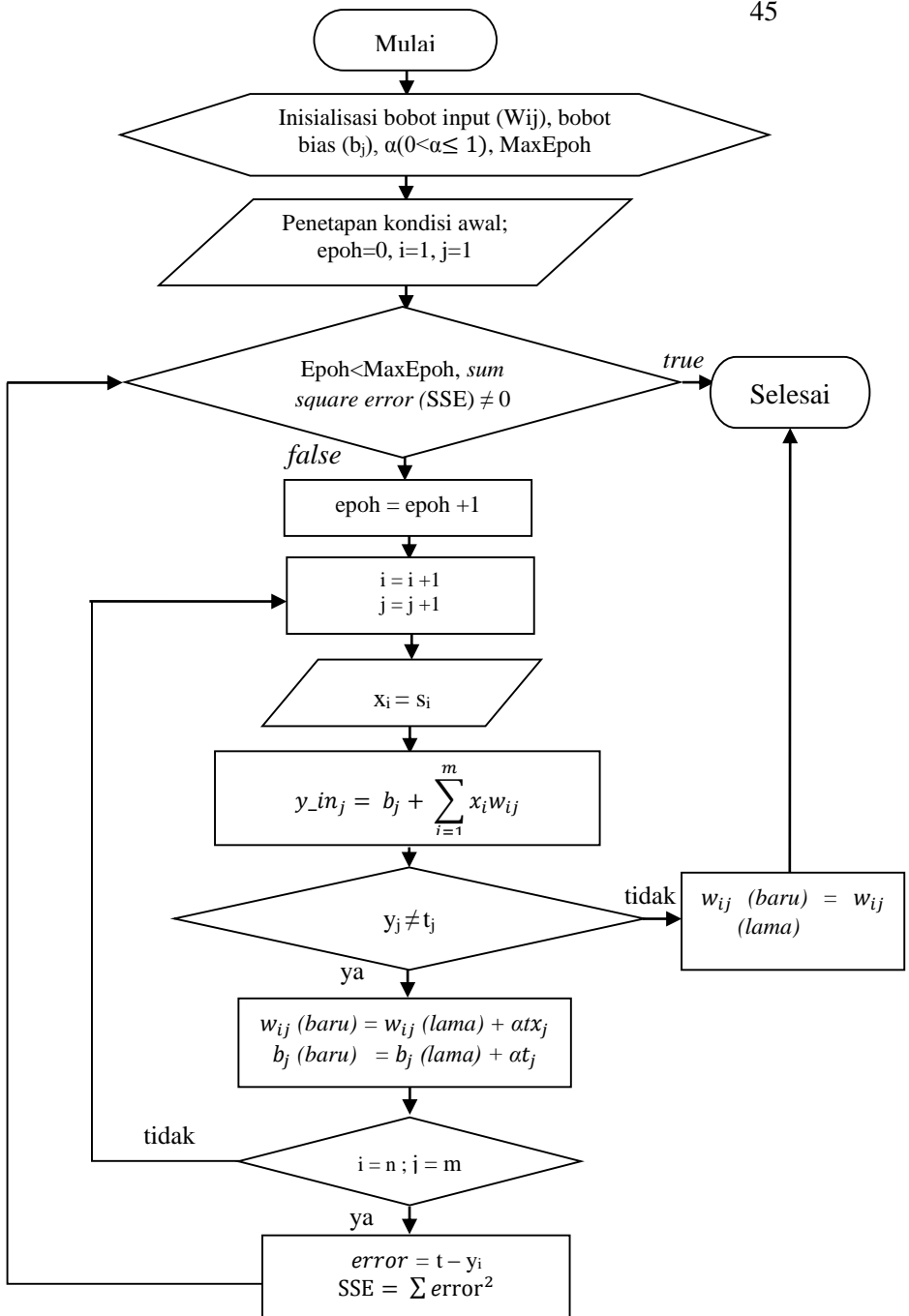
$$w_{ij} (\text{baru}) = w_{ij} (\text{lama}) \quad (4.17)$$

$$b_j (\text{baru}) = b_j (\text{lama}) \quad (4.18)$$

Tes kondisi berhenti :

Jika masih terjadi perubahan bobot atau jumlah kuadrat *error* ($\text{sum square error} \neq 0$) dan $\text{epoh} < \text{MaxEpoh}$, maka kondisi berhenti *false*, namun jika sudah tidak terjadi perubahan bobot ($\text{sum square error} = 0$) atau $\text{epoh} \geq \text{MaxEpoh}$, maka kondisi berhenti *true*.

Fitur hasil pengolahan citra menjadi *input* untuk JST. Model JST yang digunakan dalam penelitian ini adalah model Multi Perceptron. *Flowchart* tahap pembelajaran model Multi Perceptron disajikan pada **Gambar 4.9** berdasarkan algoritma di atas.



Gambar 4.9. Flowchart tahap training model Multi Perceptron

Data yang digunakan pada proses training JST Multi Perceptron untuk identifikasi *plasmodium falciparum* sebanyak 180 data citra sel darah merah tunggal, yang terdiri dari 100 citra sel darah penderita malaria dan 80 citra sel darah bukan penderita malaria.

Variabel manipulasi dalam penelitian ini adalah nilai laju pembelajaran dan nilai toleransi error atau ketelitian. Nilai laju pembelajaran yang digunakan adalah 0.1, 0.5 dan 1. Nilai toleransi error yang digunakan adalah 0.01, 0.0001 dan 0.000001, sedangkan nilai maksimum epoch yang digunakan adalah 100.

Data masukan untuk proses pelatihan ini adalah nilai fitur hasil pengolahan citra digital yang telah dilakukan sebelumnya. **Tabel 4.3** menunjukkan data hasil training jaringan Multi Perceptron untuk identifikasi plasmodium falciparum.

Tabel 4.3. Hasil pelatihan jaringan Multi Perceptron

Learning Rate	Error Tolerance	w1	w2	w3	b	epoch
0.1	0.000001	-124	54.6	0.1344	19.1	99
0.5	0.0001	-620	273	0.6724	95.5	99
1	0.01	-1240	546	1.3448	191	99

Proses pelatihan menghasilkan nilai w1, w2, w3 dan b, masing-masing adalah bobot akhir dari fitur pertama, bobot akhir dari fitur kedua, bobot akhir dari fitur ketiga dan bobot akhir bias. Bobot-bobot akhir tersebut nantinya akan digunakan dalam proses pengujian.

Sedangkan, data yang digunakan pada proses pelatihan JST untuk proses deteksi sel stadium *gametocyte* yang mempunyai ciri berbentuk pipih adalah sebanyak 20 citra plasmodium falciparum sebelum proses *cropping*, yang terdiri dari 4 citra *gametocyte* dan 16 citra bukan *gametocyte*. Data masukan untuk proses pelatihan deteksi sel stadium *gametocyte* adalah nilai *eccentricity* sel darah merah (lampiran B).

Variabel manipulasi dalam penelitian ini adalah nilai laju pembelajaran dan nilai toleransi error atau ketelitian. Nilai laju pembelajaran yang digunakan adalah 0.1, 0.5 dan 1. Nilai toleransi error yang digunakan adalah 0.01, 0.0001 dan 0.000001, sedangkan nilai maksimum epoch yang digunakan adalah 100. **Tabel 4.4** menunjukkan data hasil pelatihan jaringan Multi Perceptron untuk deteksi sel stadium *gametocyte*.

Tabel 4. 4. Hasil pelatihan jaringan Multi Perceptron stadium *gametocyte*

Learning Rate	Error Tolerance	w	b	epoch
0.1	0.000001	0.1588	-0.1	4
0.5	0.0001	0.7943	-0.5	4
1	0.01	1.5886	-1	4

Bobot akhir dan bobot bias tersebut nantinya akan digunakan dalam proses pengujian.

“Halaman ini sengaja dikosongkan”

BAB V IMPLEMENTASI SISTEM

Pada bab ini, dibahas mengenai langkah-langkah dalam pengimplemetasian sistem berdasarkan desain sistem yang telah dirancang.

5.1 Lingkungan *Hardware* dan *Software*

Lingkungan perancangan sistem dibangun dari dua lingkungan yaitu lingkungan *software* dan lingkungan *hardware*. Spesifikasi lingkungan perancangan sistem secara lengkap dapat dilihat pada **Tabel 5.1**.

Tabel 5.1. Spesifikasi *Hardware* dan *Software*

Lingkungan	Spesifikasi	
Hardware	Processor	Intel® Celeron® CPU 1007U @ 1.50GHz 1.50 GHz
	RAM	2 GB
Software	Sistem Operasi	Windows 8 Pro 32-bit
	Tools	MATLAB

5.2 Implementasi Praproses Data

Praproses data dilakukan dengan melakukan proses *cropping* pada citra input, sehingga didapatkan citra sel darah merah tunggal. Implementasi program dengan toolbox matlab untuk proses *cropping* adalah sebagai berikut :

```
axes(handles.axes1);  
image(handles.gambar)  
axis off  
guidata(hObject, handles);  
else  
    return;  
end
```

```

i=imcrop(handles.gambar);
a=imresize(i,[50 50]);
handles.a=a;
guidata(hObject, handles)
axes(handles.axes2);
axis off
imshow(a);

```

Setelah dilakukan proses *cropping*, selanjutnya melakukan proses pengolahan citra digital meliputi proses *grayscale*, segmentasi citra, operasi morfologi dan ekstraksi fitur. Implementasi program dengan toolbox matlab untuk mengkonversi citra RGB ke *grayscale* adalah sebagai berikut :

```

a=rgb2gray(handles.gambar1);
axes(handles.axes2);
imshow(a)

```

Selanjutnya citra sel darah merah tunggal hasil proses *grayscale* disegmentasi dengan proses *threshold*. Proses *thresholding* berfungsi untuk memisahkan sel darah merah dan *background*. Toolbox matlab menyediakan fungsi *graythresh* yang menghitung *threshold* menggunakan metode otsu. Sintaks untuk pemanggilan fungsi *graythresh* adalah :

```

[b,k]= size(a);
T=graythresh(a)
T=T*255

```

dimana a adalah citra input dan T adalah *threshold* yang dihasilkan.

Proses *threshold* akan menghasilkan citra biner, yaitu citra yang memiliki dua nilai tingkat keabuan (hitam dan putih). nilai ambang T (*threshold*) dapat dihitung secara otomatis menggunakan metode otsu berdasarkan citra masukan. Pendekatan yang digunakan oleh metode Otsu adalah dengan melakukan analisis diskriminan, yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Implementasi program dengan toolbox matlab untuk proses tersebut adalah sebagai berikut :

```
[label1, nlabel1] = bwlabel(Fill);
T=50;
for i = 1:b
    for j = 1:k
        if (a(i,j)>T )
            B(i,j)=0;
        else B(i,j)=1;
        end
    end
end
axes(handles.axes2); imshow(B);
```

Tahap selanjutnya, dilakukan operasi *morphology* berupa operasi *clear border* untuk menghilangkan noise yang menempel pada border citra. Implementasi program dengan toolbox matlab untuk proses tersebut adalah sebagai berikut :

```
clear=imclearborder(B);
axes(handles.axes2);
imshow(clear)
```

Selanjutnya dengan operasi *filling holes* untuk mengisi bentuk sel yang berlubang akibat proses segmentasi. Implementasi program dengan toolbox matlab untuk proses tersebut adalah sebagai berikut :

```
C=imfill(clear,'holes');
axes(handles.axes3);
imshow(C)
```

Operasi morfologi dibutuhkan untuk mendapatkan nilai fitur area sel darah merah dan fitur area akromia sentral, yaitu dengan cara menghitung jumlah piksel warna putih sebelum dan setelah dilakukan *filling holes* untuk mengidentifikasi sel darah terinfeksi *plasmodium falciparum*. Implementasi program dengan toolbox matlab untuk proses tersebut adalah sebagai berikut :

```
B (data);[m,n]= size(B);
count = 0;
for i = 1 : m;
    for j = 1 : n;
        if B(i,j) == 1;
            count = count + 1;
        end
    end
end
jumPixelPutih (data)= count

C=imfill(B,'holes');[r,s]= size(C);
count = 0;
for i = 1 : r;
    for j = 1 : s;
        if C(i,j) == 1;
            count = count + 1;
        end
    end
end
jumPixelTotal (data)= count

akromia (data)= jumPixelTotal(data)-
jumPixelPutih(data)
```

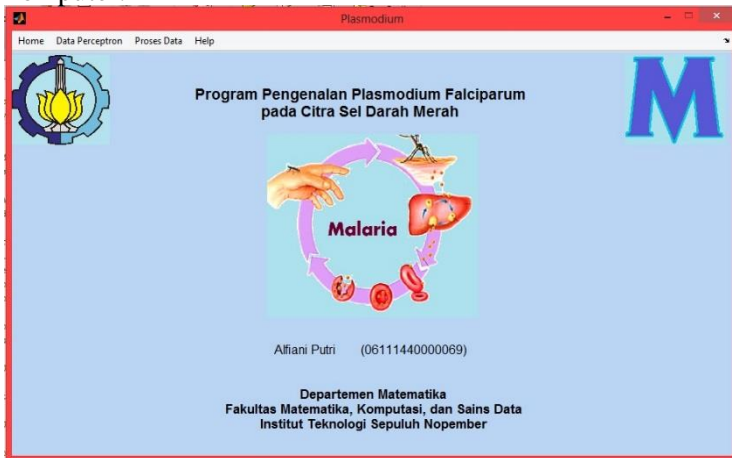
Sedangkan untuk mendeteksi atau melakukan pencarian sel stadium *gametocyte* berbentuk pipih dilakukan dengan menggunakan JST Multi Perceptron berdasarkan nilai *eccentricity* sel darah merah. Pencarian nilai *eccentricity* tiap sel darah merah membutuhkan operasi pelabelan komponen. Dengan pelabelan komponen ini ciri-ciri mendasar dari masing-masing sel darah merah dapat dihitung.

Nilai *eccentricity* sel stadium *gametocyte* dapat diketahui dengan toolbox matlab berikut :

```
[label2, nlabel2] = bwlabel(Zero);
ecce=[props.Eccentricity];
Perimeter_Ratio=(stats(k).Perimeter)/(2*(m+n));
Eccentricity = stats(k).Eccentricity;
if Perimeter_Ratio < 0.0405299
text(boundary(2,2)+10,boundary(1,1), 'Tropozoit');
elseif Perimeter_Ratio >= 0.0405299
if Eccentricity < 0.787774
text(boundary(2,2)+10,boundary(1,1), 'Schizont');
elseif Eccentricity >= 0.787774
text(boundary(2,2)+10,boundary(1,1), 'Gametocyte'); end
end
```


5.3 Implementasi JST Model Multi Perceptron

Desain *user interface* program identifikasi citra *plasmodium falciparum* di sel darah merah ini dapat dilihat pada **Gambar 5.1**. Pada desain *user interface* ini terdapat menu *Home*, *Data Multi Perceptron*, *Proses Data*, dan *Help*. File yang digunakan sebagai masukan dalam Tugas Akhir ini adalah file yang berekstensi jpg yang diambil langsung dari direktori komputer.



Gambar 5.1. Desain menu *Home* program identifikasi citra *plasmodium falciparum*

5.3.1 Proses *Training*

1. Inisialisasi bobot dan bias. Untuk mempermudah, set bobot dan bias menjadi nol. Berikut deklarasi variabel awal :

```
w1=0;
w2=0;
w3=0;
b=0;
data=1;
epoh=1;
```

2. Hitung respon untuk unit *output* :

$$y_{in_j} = b_j + \sum_{i=1}^m x_i w_{ij}; \quad (5.1)$$

dengan $j=1,2,\dots,c$

Implementasi program untuk menentukan nilai unit *output* adalah sebagai berikut :

```
y_in=b+(X(i,1)*w1)+(X(i,2)*w2)+(X(i,3)*w3);
    if y_in>0
        y=1;
    else if y_in<0
        y=-1;
    else y=0;
        end
    end
```

3. *Update* bobot dan bias jika ada *error* untuk suatu pattern dengan Persamaan (4.4), (4.5), (4.6), dan (4.7).

Implementasi program untuk menentukan nilai *Update* bobot dan bias adalah sebagai berikut :

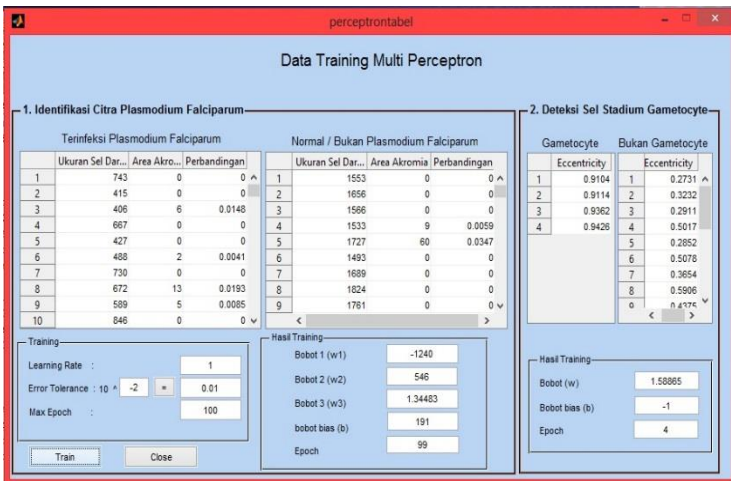
```
if (y~=T(i))
    w1lama=(a*T(i)*X(i,1));
    w2lama=(a*T(i)*X(i,2));
    w3lama=(a*T(i)*X(i,3));
    w1=w1+w1lama;
    w2=w2+w2lama;
    w3=w3+w3lama;
    b=b+(a*T(i));
else if (y==T(i))
    w1=w1;
    w2=w2;
    w3=w3;
    b=b;
    end
end
```

4. Test untuk kondisi STOP.

Implementasi program untuk menentukan kondisi berhenti adalah sebagai berikut :

```
SSE=0;
err=T(i)-y;
SSE=SSE+abs(err^2);
MSE=SSE/(m);
t=T(i);
```

Tampilan program Training tampak seperti pada **Gambar 5.2**, yang menampilkan fitur area sel darah merah, fitur area akromia sentral dan nilai perbandingan serta nilai *eccentricity* sel stadium *gametocyte*. Program Training juga menampilkan bobot fitur akhir, bobot bias akhir dan banyaknya epoch yang dibutuhkan dari proses pelatihan. Pengguna dapat merubah nilai laju pembelajaran (*learning rate*), nilai ketelitian atau *error* serta maksimal epoch. Hasil dari perubahan tersebut akan mengakibatkan perubahan pada bobot fitur akhir, bobot bias akhir dan banyaknya epoch yang akan terlihat pada panel hasil training.



Gambar 5.2. Tampilan program *Training* Multi Perceptron

5.3.1 Proses *Testing*

1. Hitung respon dari unit-unit output.

$$y_in_j = b + \sum_i x_i w_{ij} \quad (5.2)$$

Implementasi program untuk menentukan nilai unit *output* adalah sebagai berikut :

```

y_in(i)=b+(X(i,1)*w1)+(X(i,2)*w2)
+(X(i,3)*w3);
    if y_in(i)>0
        y(i)=1;
    else if y_in(i)<0
        y(i)=-1;
    else
        y(i)=0;
    end

```

Tampilan program *Testing* tampak seperti pada **Gambar 5.3**, menampilkan fitur area sel darah merah, fitur area akromia sentral, nilai perbandingan fitur dan nilai *eccentricity* sel darah merah data uji beserta hasil identifikasi jaringan saraf tiruan Multi Perceptron (y). Hasil identifikasi jaringan saraf tiruan Multi Perceptron (y) dicocokkan dengan target hasil diagnosis laboratorium. Banyaknya data yang benar digunakan untuk menghitung tingkat akurasi identifikasi jaringan saraf tiruan Multi Perceptron.

tabel

Data Testing Perceptron

1. Identifikasi Citra Plasmodium Falciparum

Training

Learning Rate =

Error =

Max Epoch =

Hasil Training

w1 = w3 =

w2 = b = Epoch =

Ukuran Sel Dar...	Area Akro...	Perbandin...	y	Target
1	743	0	0	1
2	415	0	0	1
3	406	6	0.0148	1
4	667	0	0	1
5	427	0	0	1
6	488	2	0.0041	1
7	730	0	0	1
8	672	13	0.0193	-1
9	589	5	0.0085	1
10	846	0	0	1
11	785	0	0	1
12	776	0	0	1

benar = buah

akurasi = %

2. Deteksi Sel Stadium Gametocyte

Hasil Training

w =

b = Epoch =

Eccentricity	y	Target
1	0.9104	1
2	0.9114	1
3	0.9362	1
4	0.9426	1
5	0.6392	-1
6	0.5071	-1
7	0.3344	-1

benar = buah

akurasi = %

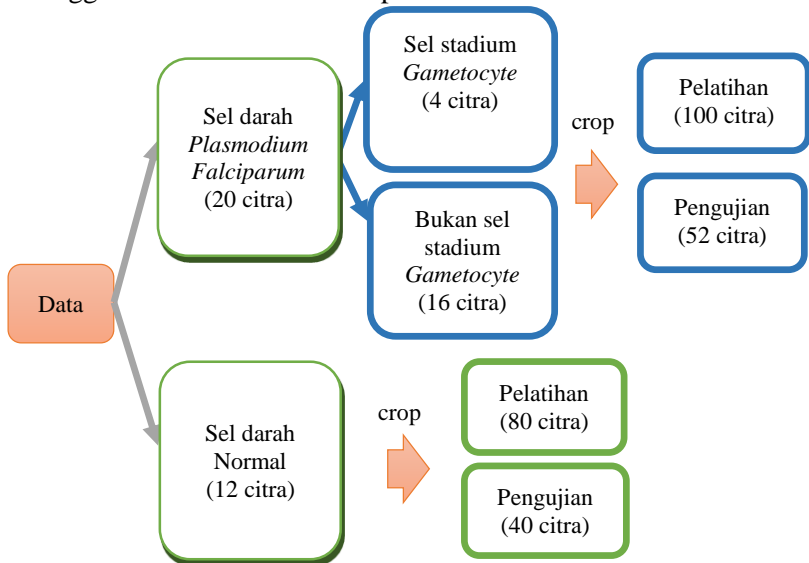
Gambar 5.3. Tampilan program *Testing* Multi Perceptron

BAB VI UJI COBA DAN EVALUASI SISTEM

Pada bab ini dijelaskan tahap-tahap uji coba berdasarkan implementasi sistem yang telah dibuat beserta pengujian kinerja sistem. Pengujian dilakukan terhadap dataset yang disimpan di dalam direktori komputer dan akan dihitung tingkat akurasi dari sistem.

6.1 Dataset Uji Coba

Fitur yang didapatkan dari *image processing* digunakan sebagai masukan untuk Jaringan Multi Perceptron. Tugas akhir ini menggunakan nilai fitur area sel darah merah, nilai fitur area akromia sentral dan nilai perbandingan fitur untuk identifikasi sel darah teinfeksi plasmodium falciparum, serta menggunakan nilai *eccentricity* untuk identifikasi sel stadium *gametocyte* menggunakan JST Multi Perceptron.



Gambar 6.1. Jumlah data pelatihan dan pengujian

Pada **Gambar 6.1** tampak bahwa jumlah keseluruhan data sebanyak 32 citra, yang terdiri dari 20 citra sel darah terinfeksi *plasmodium falciparum* dan 12 citra sel darah normal. Dari 20 citra sel darah terinfeksi *plasmodium falciparum* tersebut terdapat 4 citra yang termasuk dalam sel darah stadium *gametocyte* sedangkan 16 citra sisanya bukan merupakan sel darah stadium *gametocyte*.

Masing-masing data tersebut kemudian di *crop* untuk mendapatkan citra sel darah merah tunggal pada tiap data. Data yang digunakan sebagai masukan pelatihan JST untuk identifikasi *plasmodium falciparum* berjumlah 180 data, terdiri dari 100 citra sel darah terinfeksi *plasmodium falciparum* dan 80 citra sel darah normal. Jumlah pola masukan pada pengujian adalah jumlah sampel citra yang digunakan untuk menguji tingkat akurasi sistem. Jumlah keseluruhan data uji sebanyak 92 citra yang terdiri dari 52 citra sel darah terinfeksi *plasmodium falciparum* dan 40 citra sel darah normal.

6.2 Proses *Testing* dengan Multi Perceptron

Jaringan Multi Perceptron dapat digunakan untuk mengklasifikasikan pola masukan setelah melalui tahap pembelajaran. Langkah pengujian Multi Perceptron [16] adalah sebagai berikut:

Langkah 0: Inisialisasi bobot dan bias

(digunakan nilai bobot yang diperoleh dari algoritma pelatihan)

Langkah 1: Untuk setiap vector masukan x , lakukan langkah 2-4.

Langkah 2: Set aktivasi dari unit masukan

$$x_1 = s_i \quad (6.1)$$

Langkah 3: Hitung Total masukan ke unit keluaran

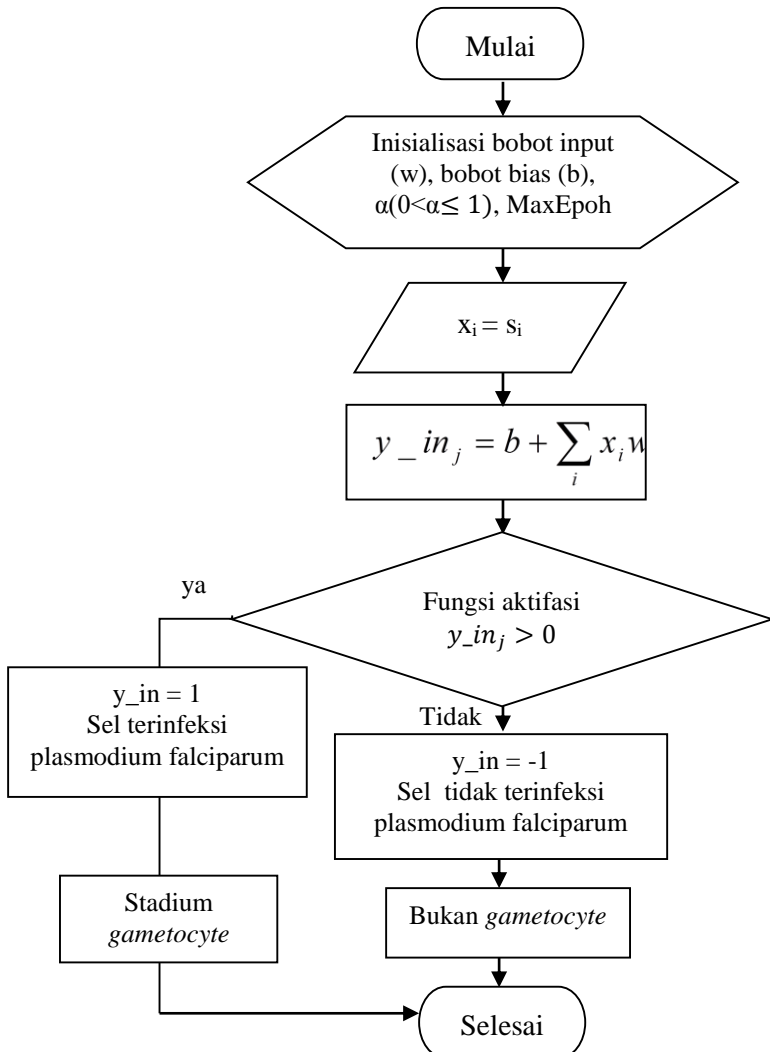
$$y_{in_j} = b + \sum x_i w_{ij} \quad (6.2)$$

Langkah 4: Gunakan Fungsi aktivasi

$$f(y_{in}) = \begin{cases} 1 & \text{jika } y_{in_j} > \theta \\ 0, & \text{jika } -\theta \leq y_{in_j} \leq \theta \\ -1, & \text{jika } y_{in_j} < -\theta \end{cases} \quad (6.3)$$

dengan nilai $f(y_{in})$ menjadi nilai keluaran dari unit keluaran.

Flowchart tahap pengujian model Multi Perceptron untuk pencarian sel darah berbentuk pensil disajikan pada **Gambar 6.2** berdasarkan algoritma di atas.



Gambar 6.2. *Flowchart* tahap testing Multi Perceptron untuk identifikasi sel terinfeksi *plasmodium falciparum*

6.3 Hasil Testing JST

Analisis hasil identifikasi JST terhadap target identifikasi (hasil laboratorium) digunakan untuk mengetahui tingkat akurasi sistem identifikasi *plasmodium falciparum* terhadap data uji. Jumlah hasil identifikasi yang sesuai dengan uji laboratorium dinyatakan dengan “jumlah sesuai” dan jumlah sampel per kategori dinyatakan dengan “banyaknya data”. Dari hasil tersebut, tingkat akurasi sistem dapat dihitung dengan persamaan berikut:

$$\text{Akurasi} = \frac{\text{jumlah sesuai}}{\text{banyaknya data}} \times 100\% \quad (6.4)$$

Proses pengujian dilakukan untuk mendapatkan tingkat akurasi dengan menguji kecocokan hasil identifikasi JST terhadap data target yang merupakan hasil diagnosa dari lab. Data yang digunakan pada proses pengujian jaringan Multi Perceptron berbeda dengan data untuk pelatihan. Data yang digunakan untuk proses pengujian sebanyak 92 data, terdiri dari 52 citra sel darah *plasmodium falciparum* dan 40 citra bukan sel darah *plasmodium falciparum*.

Bobot fitur dan bobot bias yang digunakan dalam proses pengujian jaringan Multi Perceptron ini adalah bobot akhir fitur dan bobot akhir bias yang diperoleh dari proses pelatihan atau pembelajaran. Hasil pengujian jaringan Multi Perceptron untuk identifikasi sel darah *plasmodium falciparum* dapat dilihat pada **tabel 6.1**.

Tabel 6.1. Hasil pengujian jaringan Multi Perceptron

Learning Rate	Error Tolerance	Akurasi
0.1	0.000001	89.1304 %
0.5	0.0001	89.1304 %
1	0.01	89.1304 %

Tingkat akurasi dari ketiga variabel manipulasi adalah 89.1304 % sehingga dapat dikatakan bahwa perubahan laju pembelajaran dan nilai error dalam penelitian ini tidak mempengaruhi hasil akurasi data. Hasil akurasi data dipengaruhi oleh nilai fitur yang digunakan untuk pelatihan.

Selanjutnya, data yang digunakan sebagai masukan pelatihan JST untuk proses deteksi sel stadium *gametocyte* dalam mengidentifikasi stadium *plasmodium falciparum* berjumlah 20 citra *plasmodium falciparum* sebelum proses *cropping*, yang terdiri dari 4 citra sel stadium *gametocyte* dan 16 citra bukan sel stadium *gametocyte*.

Tingkat akurasi proses deteksi sel stadium *gametocyte* dalam mengidentifikasi stadium *plasmodium falciparum* dapat diketahui dengan analisis hasil deteksi sel berbentuk pipih terhadap target identifikasi (hasil laboratorium). Jumlah hasil identifikasi yang sesuai dengan uji laboratorium dinyatakan dengan “jumlah sesuai” dan jumlah sampel per kategori dinyatakan dengan “banyaknya data”. Dari hasil tersebut, tingkat akurasi sistem dapat dihitung dengan persamaan berikut:

$$\text{Akurasi} = \frac{\text{jumlah sesuai}}{\text{banyaknya data}} \times 100\% \quad (6.5)$$

Data yang digunakan untuk proses pengujian jaringan Multi Perceptron untuk proses deteksi sel darah stadium *gametocyte* berbeda dengan data untuk pelatihan. Data yang digunakan untuk proses pengujian sebanyak 13 data, terdiri dari 4 citra sel darah sel darah stadium *gametocyte* dan 9 citra sel darah bukan stadium *gametocyte*.

Bobot dan bobot bias yang digunakan dalam proses pengujian jaringan Multi Perceptron ini adalah bobot akhir dan bobot akhir bias yang diperoleh dari proses pelatihan. Hasil

pengujian jaringan Multi Perceptron untuk deteksi sel sel darah stadium *gametocyte* dapat dilihat pada **tabel 6.2**.

Tabel 6.2. Hasil pengujian jaringan Multi Perceptron stadium *gametocyte*

Learning Rate	Error Tolerance	Akurasi
0.1	0.000001	92.3077 %
0.5	0.0001	92.3077%
1	0.01	92.3077%

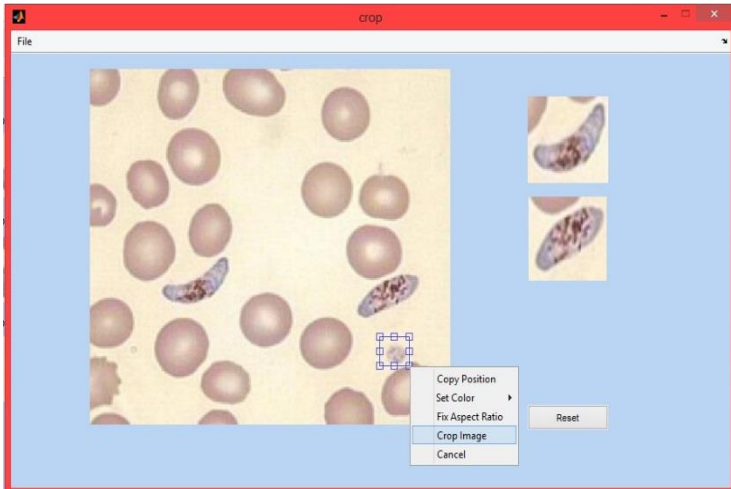
Tingkat akurasi dari ketiga variabel manipulasi adalah 92.3077% sehingga dapat dikatakan bahwa perubahan laju pembelajaran dan nilai error dalam penelitian ini tidak mempengaruhi hasil akurasi data. Hasil akurasi data dipengaruhi oleh nilai *eccentricity* data yang digunakan untuk pelatihan.

6.4 Uji Coba Model

Uji coba model dilakukan dengan menjalankan implementasi jaringan saraf tiruan dengan menggunakan model *Multi Perceptron* untuk pengenalan citra plasmodium *falciparum* di sel darah merah.

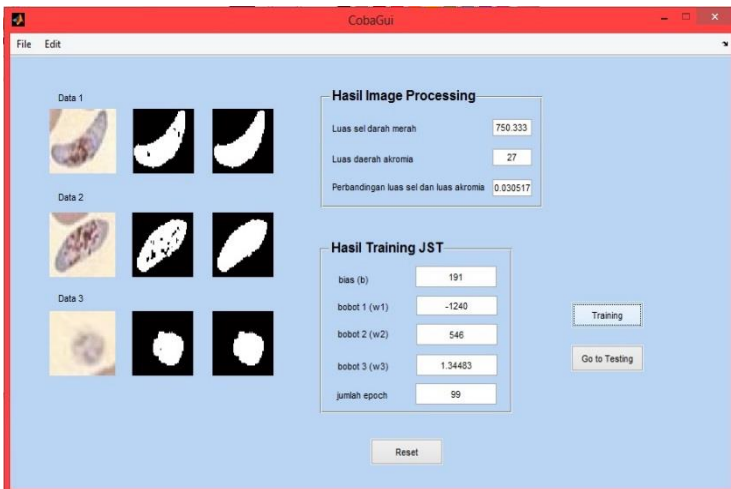
6.4.1 Uji Coba Sel Darah Terinfeksi *Plasmodium Falciparum*

Pada tampilan program Proses Data, program ini terdiri dari dua sub menu, yaitu program Input Data dan program Identifikasi Data. Program Input Data digunakan untuk meng-*crop* citra darah menjadi citra sel darah tunggal. Tampilan program *cropping* data tampak seperti pada **Gambar 6.3**.



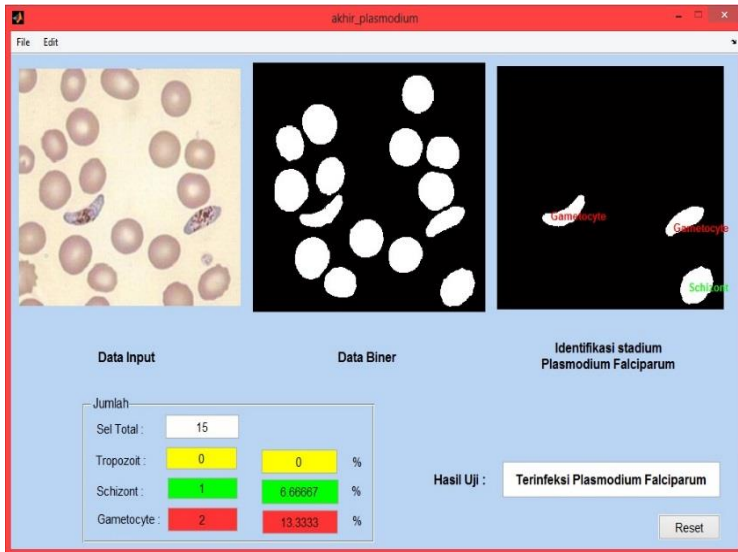
Gambar 6.3. Tampilan program input data

Tampilan program Identifikasi Data digunakan untuk mendapatkan nilai penghitungan hasil image processing dan hasil training JST Multi Perceptron. Tampilan program Identifikasi Data tampak seperti pada **Gambar 6.4.**



Gambar 6.4. Tampilan program Identifikasi Data

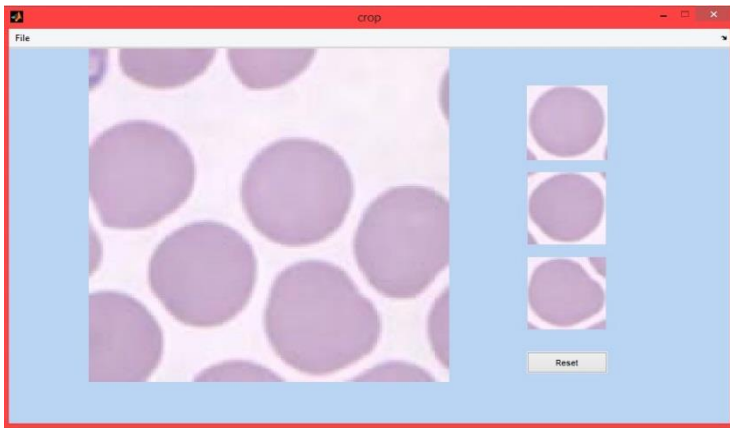
Tampilan akhir identifikasi program *plasmodium falciparum* akan tampak seperti **Gambar 6.5**. Pada program akan melakukan proses penghitungan jumlah total sel, jumlah sel stadium trophozoit, schizont, dan gametocyte berdasarkan hasil ekstraksi ciri morfologi parasit. Apabila terdapat sel darah stadium *gametocyte* yang berbentuk pipih, program akan menampilkan keterangan pada label hasil uji “Terinfeksi Plasmodium Falciparum”.



Gambar 6.5. Tampilan program Identifikasi Plasmodium Falciparum dan deteksi sel stadium *gametocyte*.

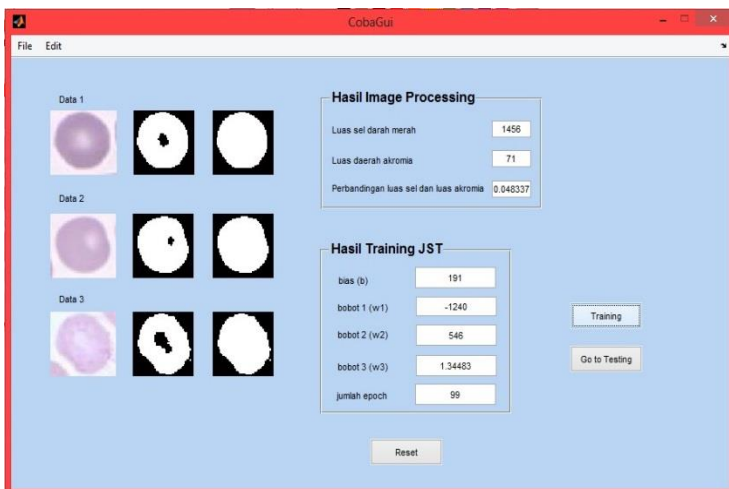
6.4.2 Uji Coba Sel Darah Normal

Program Input Data digunakan untuk meng-*crop* citra darah normal menjadi citra sel darah tunggal. Tampilan program *cropping* data tampak seperti pada **Gambar 6.6**



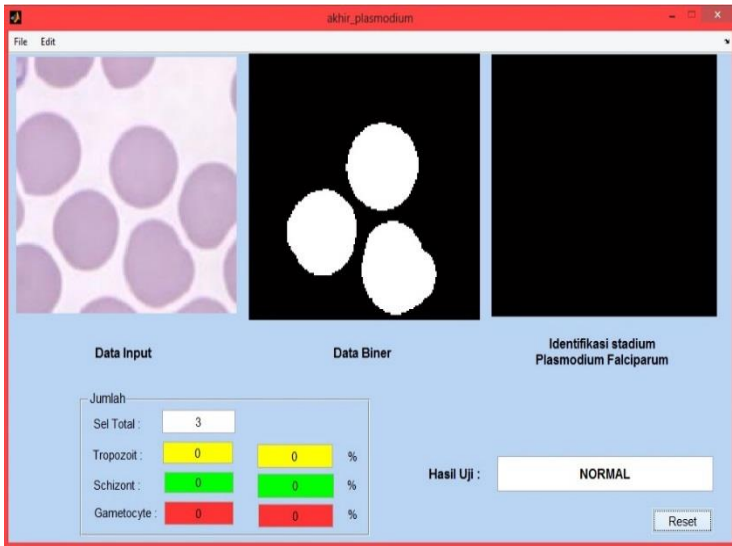
Gambar 6.6. Tampilan program input data

Tampilan program Identifikasi Data digunakan untuk mendapatkan nilai penghitungan hasil image processing dan hasil training JST Multi Perceptron. Tampilan program Identifikasi Data untuk sel darah normal tampak seperti pada **Gambar 6.7.**



Gambar 6.7. Tampilan program Identifikasi Data














Tampilan akhir identifikasi program *plasmodium falciparum* akan tampak seperti **Gambar 6.8**. Pada program akan melakukan proses penghitungan jumlah total sel, jumlah sel stadium trophozoit, schizont, dan gametocyte berdasarkan hasil ekstraksi ciri morfologi parasit. Apabila terdapat sel darah stadium *gametocyte* yang berbentuk pipih, program akan menampilkan keterangan pada label hasil uji “Terinfeksi Plasmodium Falciparum”. namun bila keberadaan sel darah stadium *gametocyte* yang berbentuk pipih tidak ditemukan maka program akan menampilkan keterangan pada label hasil uji “Normal”.



































Gambar 6.8. Tampilan program Identifikasi Sel Darah Normal












6.5 Tabel Analisis Hasil












Tabel 6.3. Data hasil pengujian jaringan Multi Perceptron












No	Citra	Hasil	Target	Keterangan
1		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
2		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
3		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
4		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
5		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
6		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
7		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
8		Bukan Plasmodium Falciparum	Plasmodium Falciparum	SALAH
9		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
10		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
11		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
12		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
13		Plasmodium Falciparum	Plasmodium Falciparum	BENAR












14		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
15		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
16		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
17		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
18		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
19		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
20		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
21		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
22		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
23		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
24		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
25		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
26		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
27		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
28		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
29		Plasmodium Falciparum	Plasmodium Falciparum	BENAR




30		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
31		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
32		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
33		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
34		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
35		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
36		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
37		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
38		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
39		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
40		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
41		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
42		Bukan Plasmodium Falciparum	Plasmodium Falciparum	SALAH
43		Plasmodium Falciparum	Plasmodium Falciparum	BENAR
44		Bukan Plasmodium Falciparum	Plasmodium Falciparum	SALAH
45		Plasmodium Falciparum	Plasmodium Falciparum	BENAR

46		Bukan Plasmodium Falciparum	Plasmodium Falciparum	SALAH
47		Bukan Plasmodium Falciparum	Plasmodium Falciparum	SALAH
48		Bukan Plasmodium Falciparum	Plasmodium Falciparum	SALAH
49		Bukan Plasmodium Falciparum	Plasmodium Falciparum	SALAH
50		Bukan Plasmodium Falciparum	Plasmodium Falciparum	SALAH
51		Bukan Plasmodium Falciparum	Plasmodium Falciparum	SALAH
52		Bukan Plasmodium Falciparum	Plasmodium Falciparum	SALAH
53		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
54		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
55		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
56		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR

57		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
58		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
59		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
60		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
61		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
62		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
63		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
64		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
65		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
66		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
67		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR

68		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
69		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
70		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
71		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
72		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
73		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
74		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
75		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
76		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
77		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
78		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR

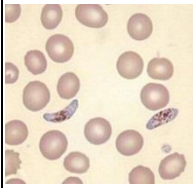
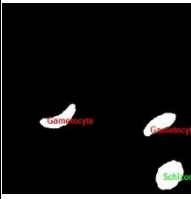
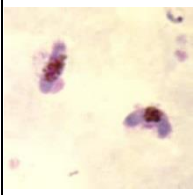
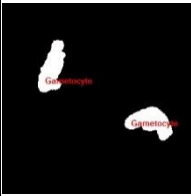
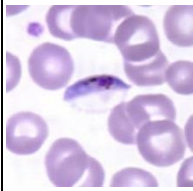

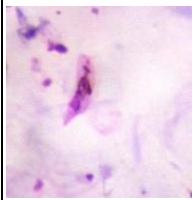

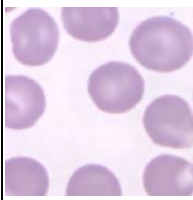
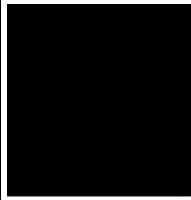
79		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
80		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
81		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
82		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
83		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
84		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
85		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
86		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
87		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
88		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
89		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR


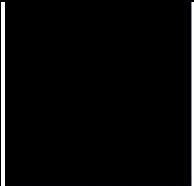
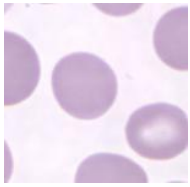
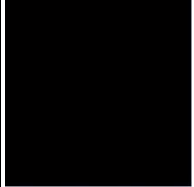
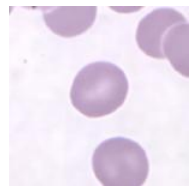
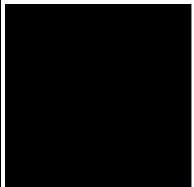
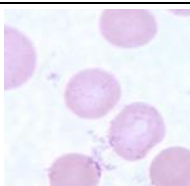
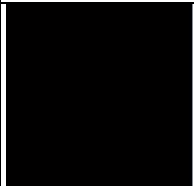
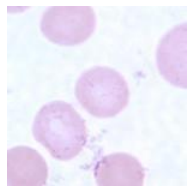
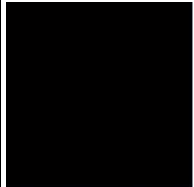
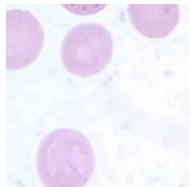
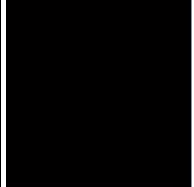
90		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
91		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR
92		Bukan Plasmodium Falciparum	Bukan Plasmodium Falciparum	BENAR

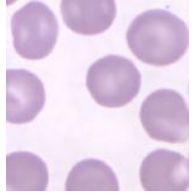

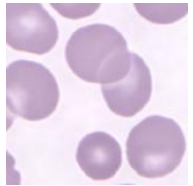
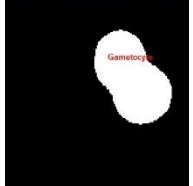
Dari **tabel 6.3** terlihat bahwa dari 92 data yang diujikan, terdapat 82 data yang benar dan 10 data yang salah sehingga tingkat akurasi jaringan Multi Perceptron adalah 89.1304%.

Setelah melalui proses pengujian jaringan Multi Perceptron, data yang menunjukkan hasil bahwa data tersebut merupakan citra sel darah yang terinfeksi *Plasmodium Falciparum* perlu diproses lebih lanjut apakah data tersebut merupakan sel darah stadium *gametocyte* atau bukan stadium *gametocyte* dengan mencari ada atau tidaknya sel darah berbentuk pipih.

Tabel 6.4. Data hasil pengujian deteksi stadium *gametocyte*

No	Citra	Hasil	Target	Keterangan
1			Terinfeksi	BENAR
2			Terinfeksi	BENAR
3			Terinfeksi	BENAR
4			Terinfeksi	BENAR
5			Normal	BENAR

6			Normal	BENAR
7			Normal	BENAR
8			Normal	BENAR
9			Normal	BENAR
10			Normal	BENAR
11			Normal	BENAR

12			Normal	BENAR
13			Terinfeksi	SALAH

Hasil pengujian data pada proses deteksi sel stadium *gametocyte* dapat dilihat pada **tabel 6.4**. Dari 13 citra yang telah di uji yakni sebanyak 4 citra *plasmodium falciparum* dan 9 citra sel darah normal sebelum proses cropping, didapatkan hasil pengujian bahwa hanya 1 data yang tidak sesuai dengan target hasil diagnosa dari lab. Sehingga hasil akurasi pendeteksian sel darah stadium *gametocyte* berbentuk pipih ini mencapai 92.3077%.

“Halaman ini sengaja dikosongkan”

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Dari uji coba dan evaluasi pada bab sebelumnya, dapat disimpulkan beberapa hal sebagai berikut:

1. Proses identifikasi *plasmodium falciparum* stadium *gametocyte* yang berbentuk pipih dilakukan dengan menggunakan JST Multi Perceptron berdasarkan nilai *eccentricity* sel darah merah. Sel darah yang memiliki sel *gametocyte* merupakan sel darah penderita malaria. Sistem yang dibuat dengan menggunakan metode jaringan saraf tiruan model Multi Perceptron memiliki nilai akurasi untuk identifikasi *plasmodium falciparum* mencapai 89.1% dan nilai akurasi untuk identifikasi sel darah stadium *gametocyte* berbentuk pipih mencapai 92.3%.
2. Proses penghitungan jumlah parasit *plasmodium falciparum* dilakukan dengan mendapatkan nilai dari fitur citra meliputi fitur area sel darah merah, fitur area akromia sentral, nilai perbandingan fitur dan nilai *eccentricity* digunakan sebagai input jaringan saraf tiruan (JST) model Multi Perceptron, dengan hasil akhir berupa penghitungan jumlah total sel, jumlah sel stadium *tropozoit*, *schizont*, dan *gametocyte* berdasarkan hasil ekstraksi ciri morfologi parasit. Apabila terdapat sel darah stadium *gametocyte* yang berbentuk pipih, program akan menampilkan keterangan pada label hasil uji “Terinfeksi Plasmodium Falciparum”. namun bila keberadaan sel darah stadium *gametocyte* yang berbentuk pipih tidak ditemukan maka program akan menampilkan keterangan pada label hasil uji “Normal”.

7.2 Saran

Saran yang diberikan untuk perbaikan dan kelanjutan dari penelitian ini adalah :

1. Melakukan penelitian yang membandingkan metode Multi Perceptron dengan metode-metode jaringan saraf tiruan lainnya.
2. Menambahkan dan mengembangkan suatu algoritma untuk proses klasifikasi stadium parasit *Plasmodium Falciparum* agar mendapatkan hasil yang lebih baik dan akurat.

DAFTAR PUSTAKA

- [1] WebMD 2013. Overview Malaria, Medical Reference from Healthwise, Amerika Serikat.
- [2] Nugraha, A.C. "Laporan Magang Diagnosis Malaria dengan Pemeriksaan Mikroskopis di Balai Litbang P2B2 Banjarnegara", 2012.
- [3] Budi Setiyawan. 2003. Pengantar Jaringan Syaraf Tiruan.
- [4] Fausset, L. (1991). Fundamentals of Neural Networks Architectures, Algorithms, and Applications. Prentice Hall.
- [5] Pamungkas, Adi. "The Automatic Counting Of The Number Of Red Blood Cells And Identification Of Plasmodium Falciparum Phase Using Morphological Operations". Faculty of Science and Mathematics, Diponegoro University, Semarang, Indonesia.
- [6] Susmikanti, Mike. 2010. Pengenalan Pola Berbasis Jaringan Syaraf Tiruan Dalam Analisa CT Scan Tumor Otak Beligna". Seminar Nasional Aplikasi Teknologi Informasi. Yogyakarta, 19 Juni 2010. **ISSN: 1907-5022**.
- [7] Memeu1, Daniel Maitethia, et. al. (2013). "Detection of Plasmodium Parasites from Images of Thin Blood Smears". **Open Journal of Clinical Diagnostics**, Vol 3. Hal. 183-194.
- [8] Athoillah, M., Irawan, M. I., dan Imah, E.M. (2015). "Support Vector Machine with Multiple Kernel Learning for Image Retrieval". International Conference on Information, Communication Technology and System (ICTS).
- [9] Banyal, Nur Ain, Surianti, Abd. Rachman Dayat. (2016). "Klasifikasi Citra Plasmodium Penyebab Penyakit Malaria Dalam Sel Darah Merah Manusia

- dengan Menggunakan Metode Multi lass Support Vector". **Jurnal Ilmiah Ilkom Volume 8 No 2**.
- [10] Tim Infolabmed Malaria, 2017. Pemeriksaan Laboratorium dan Identifikasi dilengkapi ATLAS Berwarna MALARIA Berbagai Fase.
- [11] Praidia, A. R. (2008). Pengenalan Penyakit Darah Menggunakan Teknik Pengolahan Citra dan Jaringan Syaraf Tiruan. Laporan Tugas Akhir. Departemen Teknik Elektro Fakultas Teknik Universitas Indoneisa.
- [12] Haykin, S. (1999). Neural Networks A Comprehensive Foundation Second Edition. Singapura : Prentice Hall.
- [13] Bahri, R. (2011). Perbandingan Algoritma Template Matcing dan Featur Extraction pada Optical Character Recognition.
- [14] Ahmad, Usman. 2005. Pengolahan Citra Digital dan Teknik Pemrogramannya. Edisi Pertama. Yogyakarta: Graha Ilmu. ISBN:979-756-072-6
- [15] Dougherty, Geoff. 2009. Digital Image Processing for Medical Applications. Published in the United States of America by Cambridge University Press, New York. **ISBN-13 978-0-511-53343-3**.
- [16] Kusumadewi, Sri. 2004. Membangun Jaringan Syaraf Tiruan Menggunakan Matlab & Excel Link. Yogyakarta: Graha Ilmu. **ISBN :979-3289-91-0**.
- [17] Putra, Darma. 2010. Pengolahan Citra Digital. Yogyakarta: ANDI. **ISBN: 978-979-29-1443-6**.
- [18] Gonzales, R. C., Woods, R.E. (2002). Digital Image Processing Second Edition. New Jersey : Prentice Hall.
- [19] Tcheslavski, Gleb V. 2009. Morphological Image Processing: Basic Algorithms. Spring 2009. <http://ee.lamar.edu/gleb/dip/index.htm>

LAMPIRAN A

Tabel A.1. Nilai fitur yang digunakan sebagai masukan JST untuk sel darah terinfeksi *Plasmodium Falciparum*

No.	Terinfeksi <i>Plasmodium Falciparum</i>		
	Fitur jumlah piksel sel darah	Fitur jumlah piksel area akromia sentral	Perbandingan fitur sel darah dan fitur area akromia sentral
1	743	0	0
2	415	0	0
3	406	6	0.0147783251231527
4	667	0	0
5	427	0	0
6	488	2	0.00409836065573771
7	730	0	0
8	672	13	0.0193452380952381
9	589	5	0.00848896434634975
10	846	0	0
11	785	0	0
12	776	0	0
13	612	2	0.00326797385620915
14	460	0	0
15	440	0	0
16	545	2	0.00366972477064220
17	451	0	0
18	602	0	0
19	639	0	0
20	545	0	0
21	345	0	0
22	363	0	0
23	312	0	0

24	535	4	0.00747663551401869
25	523	0	0
26	455	0	0
27	744	0	0
28	354	0	0
29	338	0	0
30	325	0	0
31	283	0	0
32	455	0	0
33	864	0	0
34	605	0	0
35	720	0	0
36	639	0	0
37	647	0	0
38	579	0	0
39	376	0	0
40	997	0	0
41	802	0	0
42	957	14	0.0146290491118077
43	153	0	0
44	596	1	0.00167785234899329
45	622	0	0
46	655	0	0
47	667	0	0
48	544	26	0.0477941176470588
49	568	0	0
50	503	0	0
51	394	9	0.0228426395939086
52	615	8	0.0130081300813008
53	995	0	0
54	343	0	0
55	753	0	0
56	905	0	0
57	528	5	0.00946969696969697

58	339	0	0
59	492	0	0
60	798	0	0
61	891	0	0
62	766	0	0
63	502	0	0
64	615	0	0
65	839	1	0.00119189511323004
66	594	9	0.0151515151515152
67	814	0	0
68	872	0	0
69	597	0	0
70	577	0	0
71	401	20	0.0498753117206983
72	946	2	0.00211416490486258
73	830	0	0
74	867	0	0
75	200	0	0
76	464	0	0
77	834	0	0
78	887	0	0
79	749	0	0
80	217	0	0
81	499	0	0
82	351	0	0
83	422	0	0
84	656	0	0
85	575	0	0
86	606	2	0.00330033003300330
87	971	0	0
88	388	11	0.0283505154639175
89	873	0	0
90	441	15	0.0340136054421769
91	737	0	0

92	425	0	0
93	1271	0	0
94	1086	66	0.0607734806629834
95	1056	111	0.105113636363636
96	1012	58	0.0573122529644269
97	1000	0	0
98	1119	56	0.0500446827524576
99	1092	17	0.0155677655677656
100	1272	0	0

Tabel A.2. Nilai fitur yang digunakan sebagai masukan JST untuk sel darah normal / bukan *Plasmodium Falciparum*

No.	Normal / Bukan <i>Plasmodium Falciparum</i>		
	Fitur jumlah piksel sel darah	Fitur jumlah piksel area akromia sentral	Perbandingan fitur sel darah dan fitur area akromia sentral
1	1553	0	0
2	1656	0	0
3	1566	0	0
4	1533	9	0.00587084148727984
5	1727	60	0.0347423277359583
6	1493	0	0
7	1689	0	0
8	1824	0	0
9	1761	0	0
10	1658	64	0.0386007237635706
11	1577	0	0
12	1617	0	0
13	1493	100	0.0669792364367046
14	1679	20	0.0119118522930316
15	1617	0	0
16	1454	59	0.0405777166437414
17	1675	91	0.0543283582089552
18	1379	0	0
19	1419	0	0
20	1589	0	0
21	1574	42	0.0266836086404066
22	1829	87	0.0475669764898852
23	1779	13	0.00730747611017426
24	1677	129	0.0769230769230769
25	1618	0	0

26	1378	0	0
27	1546	0	0
28	1544	0	0
29	1664	0	0
30	1658	0	0
31	1572	0	0
32	1482	0	0
33	1727	0	0
34	1457	0	0
35	1464	25	0.0170765027322404
36	1361	0	0
37	1252	7	0.00559105431309904
38	1456	53	0.0364010989010989
39	1425	0	0
40	1401	0	0
41	1519	0	0
42	1601	0	0
43	1441	0	0
44	1478	0	0
45	1559	0	0
46	1426	0	0
47	1411	0	0
48	1343	0	0
49	1534	0	0
50	1301	0	0
51	1516	0	0
52	1577	0	0
53	1617	0	0
54	1493	100	0.0669792364367046
55	1679	20	0.0119118522930316
56	1617	0	0
57	1454	59	0.0405777166437414
58	1675	91	0.0543283582089552
59	1379	0	0

60	1419	0	0
61	1589	0	0
62	1658	0	0
63	1572	0	0
64	1482	0	0
65	1727	0	0
66	1457	0	0
67	1464	25	0.0170765027322404
68	1361	0	0
69	1252	7	0.00559105431309904
70	1456	53	0.0364010989010989
71	1425	0	0
72	1401	0	0
73	1519	0	0
74	1601	0	0
75	1441	0	0
76	1553	0	0
77	1656	0	0
78	1533	9	0.00587084148727984
79	1727	60	0.0347423277359583
80	1689	0	0

“Halaman ini sengaja dikosongkan”

LAMPIRAN B

Tabel B.1. Nilai fitur yang digunakan sebagai masukan JST untuk deteksi sel berbentuk pipih dalam mengidentifikasi sel darah stadium *gametocyte*

No.	<i>Gametocyte</i>
	<i>Eccentricity</i>
1	0.910412871713725
2	0.911414526081118
3	0.936219977998672
4	0.942609722547409

Tabel B.2. Nilai fitur yang digunakan sebagai masukan JST untuk mengidentifikasi sel darah bukan stadium *gametocyte*

No.	Bukan <i>Gametocyte</i>
	<i>Eccentricity</i>
1	0.273121228819448
2	0.323225313906088
3	0.291089791970135
4	0.501718023906406
5	0.285170999133156
6	0.507769590793557
7	0.365355687592022
8	0.590647974806593
9	0.437505077788748
10	0.131145058773877
11	0.593856790577553
12	0.237128539362227
13	0.482632072934556
14	0.345506973665943
15	0.551828826503995

16	0.470332456605050
17	0.437511488297913
18	0.256401954607435
19	0.398536260871262
20	0.574568114780680
21	0.291783069491936

LAMPIRAN C

Source Code

a. Listing Form Training

```
%Identifikasi plasmodium falciparum
clear;clc;
myform= guidata(gcbo);
a=str2double(get(myform.edit14,'string'))
error=str2double(get(myform.edit16,'string'))
maxepoh=str2double(get(myform.edit17,'string'
))
load mikro.mat
load nonmikro.mat
X(1:100,:)= fitur_mikro;
X(101:180,:)=fitur_n;
X
load target.mat
T=t
[m,n]=size(X)
w1=0;
w2=0;
w3=0;
b=0;
data=1;
epoh=1;
MSE=1;

while (epoh<maxepoh && MSE>error)
    SSE=0;
    for i=1:m
        data;

y_in=b+(X(i,1)*w1)+(X(i,2)*w2)+(X(i,3)*w3);
        if y_in>0
            y=1;
        else if y_in<0
            y=-1;
        else y=0;
```

```

        end
    end

    err=T(i)-y;
    SSE=SSE+abs(err^2);
    MSE=SSE/(m);
    t=T(i);
    if (y~=T(i))
        w1lama=(a*T(i)*X(i,1));
        w2lama=(a*T(i)*X(i,2));
        w3lama=(a*T(i)*X(i,3));
        w1=w1+w1lama;
        w2=w2+w2lama;
        w3=w3+w3lama;
        b=b+(a*T(i));
    else if (y==T(i))
        w1=w1;
        w2=w2;
        w3=w3;
        b=b;
    end
end

data=1;
data=data+i;

end

epoh=epoh+1;
end
epoh=epoh-1;
set(myform.edit18,'string',w1);
set(myform.edit19,'string',w2);
set(myform.edit20,'string',w3);
set(myform.edit21,'string',b);
set(myform.edit22,'string',epoh);
E=error;
save ('F:\Mathematics File\Semester 8
^_^Tugas Akhir\Tugas\GUI

```

```

MALARIA\percept.mat', 'epoh', 'w1', 'w2', 'w3', 'b
', 'a', 'E', 'maxepoh')

%deteksi sel gametocyte
load eccentricity.mat
load noneccentricity.mat
x(1:4,:)= fitur_ecce;
x(5:25,:)=fitur_nonecce;
x
load targetecce.mat
T=t
[m,n]=size(x)
w=0;
b=0;
%a=1;
data=1;
epoh=1;
MSE=1;

while (epoh<maxepoh && MSE>error)
    fprintf('-----')
    epoh
    fprintf('data      y_in      y
T      error      SSE      w1
w2      w3      b      MSE\n')
    SSE=0;
    for i=1:m
        data;

        y_in=b+(x(i,1)*w);
        if y_in>0
            y=1;
        else if y_in<0
            y=-1;
        else y=0;
        end
    end

    err=T(i)-y;

```

```

SSE=SSE+abs(err^2);
MSE=SSE/(m);
t=T(i);
if (y~=T(i))
    wlama=(a*T(i)*x(i,1));
    w=w+wlama;
    b=b+(a*T(i));
else if (y==T(i))
    w=w;
    b=b;
end
end

data=1;
data=data+i;

fprintf('%d      %d      %d      %d      %d\n',
data-1,y_in,y,T(i),err,SSE,w,b,MSE)
end

epoh=epoh+1;
end
epoh=epoh-1
w
b

set(myform.edit9,'string',w);
set(myform.edit10,'string',b);
set(myform.edit11,'string',epoh);
save ('F:\Mathematics File\Semester 8
^_^Tugas Akhir\Tugas\GUI
MALARIA\perceptecce.mat','epoh','w','b')

```

b. Listing *Form Testing*

```

% --- Outputs from this function are returned
to the command line.
function varargout = tabel_OutputFcn(hObject,
eventdata, handles)

% Get default command line output from
handles structure
varargout{1} = handles.output;
clc;

%untuk tes uji identifikasi plasmodium
falciparum
load tes.mat
load tesplasmodium.mat
load percept.mat
set(handles.edit3, 'string', w1);
set(handles.edit4, 'string', w2);
set(handles.edit5, 'string', w3);
set(handles.edit6, 'string', b);
set(handles.edit7, 'string', a);
set(handles.edit8, 'string', E);
set(handles.edit9, 'string', maxepoh);
set(handles.edit10, 'string', epoh);
z=[fitur_ tes y]
set(handles.uitable1, 'Data', z);

%untuk uji deteksi sel gametocyte
load perceptecce.mat
load targeteccentricity.mat
load targetecce.mat
load teseccce.mat
set(handles.edit11, 'string', w);
set(handles.edit12, 'string', b);
set(handles.edit13, 'string', epoh);
L=[fitur_ tecce Y]
set(handles.uitable2, 'Data', L);

```

```

% --- Executes on button press in
pushbutton1.
function pushbutton1_Callback(hObject,
eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
clc;

%untuk tes uji
load tes.mat
load targettes.mat
load tesnew.mat
z=[fitur_tes y t]
set(handles.uitable1, 'Data', z);
load percept.mat
X(1:92,:)= fitur_tes;
X
data=1;
[m,n]=size(X);
count=0;
for dataa=1:m
for i=1:m

y_in(i)=b+(X(i,1)*w1)+(X(i,2)*w2)+(X(i,3)*w3)
;
    if y_in(i)>0
        y(i)=1;
    else if y_in(i)<0
        y(i)=-1;
    else
        y(i)=0;
    end
    end
end
y_in(dataa)=y_in(i);
y(dataa)=y(i);
end
for i=1:m
    T=t;
    if (y(i)==T(i))

```

```

        count = count + 1;
    end
end
benar=count
akurasi= (benar/m)*100
set(handles.edit1, 'string', benar);
set(handles.edit2, 'string', akurasi);

%Deteksi sel gametocyte
load targeteccentricity.mat
load tesecece.mat
load targettesecece.mat
L=[fitur_tecce Y t]
set(handles.uitable2, 'Data', L);

load perceptecce.mat
load targeteccentricity.mat
load tesecece.mat
x(1:13, :)= fitur_tecce;
x
data=1;
[m,n]=size(x);
count=0;
for dataa=1:m
for i=1:m

    y_in(i)=b+(x(i,1)*w);
    if y_in(i)>0
        y(i)=1;
    else if y_in(i)<0
        y(i)=-1;
    else
        y(i)=0;
    end
end

end

end
end

```



```
for i=1:m
    T=t;
    if (Y(i)==T(i))
        count = count + 1;
    end
end
benar=count
akurasi= (benar/m)*100
set(handles.edit16,'string',benar);
set(handles.edit17,'string',akurasi);
```

c. Listing *Form* Program Identifikasi Data

```

% -----
function menu_edit_biner_Callback(hObject,
eventdata, handles)
% hObject    handle to menu_edit_biner (see
GCBO)
clc;
%data
a=rgb2gray(handles.gambar1);
axes(handles.axes2);
imshow(a)

[b,k]= size(a);
T=graythresh(a)
T=T*255

for i = 1:b
    for j = 1:k
        if (a(i,j)>T )
            B(i,j)=0;
        else B(i,j)=1;
        end
    end
end
axes(handles.axes2);
imshow(B);

clear=imclearborder(B);
axes(handles.axes2);
imshow(clear)

% -----
function menu_Edit_Filling_Callback(hObject,
eventdata, handles)
% hObject    handle to menu_Edit_Filling (see
GCBO)
a=rgb2gray(handles.gambar1);

```

```

[b,k]= size(a);
T=graythresh(a)
T=T*255

for i = 1:b
    for j = 1:k
        if (a(i,j)>T )
            B(i,j)=0;
        else B(i,j)=1;
        end
    end
end
clear=imclearborder(B);
C=imfill(clear, 'holes');
axes(handles.axes3);
imshow(C)

% Hitung Jumlah pixel putih
[m,n]= size(clear);
count = 0;
for i = 1 : m;
    for j = 1 : n;
        if clear(i,j) == 1;
            count = count + 1;
        end
    end
end
jumPixelPutih1 = count

% Hitung Jumlah pixel putih setelah filling
holes
[r,s]= size(C);
count = 0;
for i = 1 : r;
    for j = 1 : s;
        if C(i,j) == 1;
            count = count + 1;
        end
    end
end

```

```

    end
end
jumPixelTotal1 = count

% Hitung jumlah daerah akromia sentral
akromial= jumPixelTotal1-jumPixelPutih1
%perbandingan akromiat dan luas total
perbandingan1=akromial/jumPixelTotal1

a=rgb2gray(handles.gambar2);
[b,k]= size(a);
T=graythresh(a)
T=T*255

for i = 1:b
    for j = 1:k
        if (a(i,j)>T )
            B(i,j)=0;
        else B(i,j)=1;
        end
    end
end
clear=imclearborder(B);
C=imfill(clear, 'holes');
axes(handles.axes6);
imshow(C)

jumPixelTotal=(jumPixelTotal1+jumPixelTotal2+
jumPixelTotal3)/3;
akromia=(akromial+akromia2+akromia3)/3;
perbandingan=(perbandingan1+perbandingan2+per
bandingan3)/3;
fitur =[jumPixelTotal;akromia;perbandingan]
%fitur_gui=fitur'
%save gui.mat
set(handles.edit1, 'string', jumPixelTotal);
set(handles.edit2, 'string', akromia);
set(handles.edit3, 'string', perbandingan);

```

```

% -----
function
menu_proses_training_Callback(hObject,
 eventdata, handles)
% hObject    handle to menu_proses_training
% (see GCBO)
load percept.mat
set(handles.edit6, 'string', b);
set(handles.edit7, 'string', w1);
set(handles.edit8, 'string', w2);
set(handles.edit9, 'string', w3);
set(handles.edit10, 'string', epoh);

% -----
function
menu_testing_terinfeksi_Callback(hObject,
 eventdata, handles)
% hObject    handle to
% menu_testing_terinfeksi (see GCBO)
% akhir_plasmodium;

% --- Executes on button press in
% pushbutton5.
function pushbutton5_Callback(hObject,
 eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
load percept.mat
set(handles.edit6, 'string', b);
set(handles.edit7, 'string', w1);
set(handles.edit8, 'string', w2);
set(handles.edit9, 'string', w3);
set(handles.edit10, 'string', epoh);

```

d. Listing *Form Program Deteksi Sel Stadium Gametocyte*

```

% -----
function Menu_Edit_Proces_Callback(hObject,
eventdata, handles)
% hObject    handle to Menu_Edit_Proces (see
GCBO)
a=rgb2gray(handles.a);
[b,k]= size(a);
T=graythresh(a)
T=T*255
for i = 1:b
    for j = 1:k
        if (a(i,j)>T )
            B(i,j)=0;
        else B(i,j)=1;
        end
    end
end
B;
guidata(hObject, handles)
axes(handles.axes2);
axis off
imshow(B);

clear=imclearborder(B);
guidata(hObject, handles)
axes(handles.axes2);
axis off
imshow(clear);

Fill=imfill(clear,'holes');
guidata(hObject, handles)
axes(handles.axes2);
axis off
imshow (Fill);

[label1, nlabel1] = bwlabel(Fill);
Threshold=500;

```

```

Zero=zeros(size (Fill));
for ii = 1 : nlabel1
    objekii =(labell == ii);
    luasobjii = sum(sum(objekii));

    if luasobjii > Threshold
%untuk menghilangkan noise kecil
        Zero=Zero+objekii;

        handles.Zero=Zero;
        guidata(hObject, handles)
        axes(handles.axes2);
        axis off
        imshow (Zero);
        imshow (Zero);
    end
end

set(handles.edit4, 'String', '0')
L = bwlabel(Zero);
Jumlah_Sel_Total = max(max(L))
nilai = num2str(Jumlah_Sel_Total);
set(handles.edit4, 'String', nilai);
% -----
function Menu_Edit_Uji_Callback(hObject,
 eventdata, handles)
% hObject    handle to Menu_Edit_Uji (see
GCBO)
clc;
[label2, nlabel2] = bwlabel(handles.Zero);
props = regionprops(label2, 'eccentricity')
ecce=[props.Eccentricity];
eccentricity=[ecce']

X=eccentricity;
load perceptecce.mat

for i=1:nlabel2

```

```

    y_in(i)=b+(X(i,1)*w);
    if y_in(i)>0
        y(i)=1;
    else if y_in(i)<0
        y(i)=-1;
    end
end
end
y=[y']
idx = ( y==1);
im = ismember(label2,find(idx));
guidata(hObject, handles)
axes(handles.axes3);
axis off
imshow (im);
    set(handles.edit1,'String','0')
set(handles.edit2,'String','0')
set(handles.edit3,'String','0')

a = handles.a;
[B,L] = bwboundaries(im,'noholes');
stats =
regionprops(L,'Centroid','Area','Perimeter','
Eccentricity');

[m,n] = size(a);
for k = 1:length(B)
    boundary = B{k};
    Area_Ratio = (stats(k).Area)/(m*n);
    Perimeter_Ratio =
(stats(k).Perimeter)/(2*(m+n));
    Eccentricity = stats(k).Eccentricity;

    Metric =
4*pi.*Area_Ratio/(Perimeter_Ratio.^2);
% 4. Perimeter-Eccentricity
if Perimeter_Ratio < 0.0405299
text(boundary(2,2)+10,boundary(1,1),'Tropozoi
t','Color','y','FontWeight','bold');

```



```

        val1 =
str2double(get(handles.edit1, 'String'));
        set(handles.edit1, 'String', 1+val1);

elseif Perimeter_Ratio >= 0.0405299
    if Eccentricity < 0.787774
text(boundary(2,2)+10,boundary(1,1), 'Schizont
', 'Color', 'g', 'FontWeight', 'bold');
        val2 =
str2double(get(handles.edit2, 'String'));
set(handles.edit2, 'String', 1+val2);

elseif Eccentricity >= 0.787774
text(boundary(2,2)+10,boundary(1,1), 'Gametocy
te', 'Color', 'r', 'FontWeight', 'bold');
        val3 =
str2double(get(handles.edit3, 'String'));

set(handles.edit3, 'String', 1+val3);
        end
    end

set(handles.edit9, 'String', 'Terinfeksi
Plasmodium Falciparum');
end

for x =
str2double(get(handles.edit1, 'String'));
    y =
str2double(get(handles.edit2, 'String'));
    z =
str2double(get(handles.edit3, 'String'));

        if x==0 && y==0 && z==0

set(handles.edit9, 'String', 'NORMAL');
        end
end

```

BIODATA PENULIS



Bernama lengkap Alfiani Putri dengan nama panggilan Al. Lahir di Pasuruan, 1 Maret 1995. Jenjang pendidikan formal yang ditempuh yaitu SDN Kalirejo Bangil (2001-2007), SMPN 2 Bangil (2007-2010), SMAN 1 Bangil (2010-2013).

Pada tahun 2014 penulis diterima di Departemen Matematika ITS melalui jalur seleksi bersama masuk perguruan tinggi negeri (SBMPTN) untuk menempuh pendidikan S1 selama empat tahun. Di Departemen Matematika ITS penulis mengambil bidang minat Ilmu Komputer (Computer Science). Penulis juga aktif berorganisasi di KM ITS, yaitu sebagai staff Departemen Dalam Negeri (DAGRI) di Unit Kegiatan Mahasiswa Cinta Rebana ITS (UKM CR ITS) (2014-2015), sebagai staff Applied Science Department (ASCI) di Himpunan Mahasiswa Matematika ITS (Himatika ITS) (2015-2016), dan penulis juga aktif dalam kepanitiaan acara tingkat Nasional yaitu Olimpiade Matematika ITS (OMITS ITS) sebagai Problem Maker Crew. Penulis juga melaksanakan kerja praktek di PT. Pertamina (Persero) Surabaya.

Jika ingin memberikan saran, kritik, dan diskusi mengenai Tugas Akhir ini, bisa menghubungi melalui email alfianiputri95@gmail.com. Semoga bermanfaat.

“Halaman ini sengaja dikosongkan”