

**APLIKASI DETEKSI AWAL KELAINAN PADA *ERITROSIT*
MENGUNAKAN PENGOLAHAN CITRA DIGITAL**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan

Memperoleh gelar Sarjana Teknik



Disusun oleh:

FARIZZA ANANTA

NIM. 0710633063 - 63

KEMENTERIAN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2012

**APLIKASI DETEKSI AWAL KELAINAN PADA *ERITROSIT*
MENGUNAKAN PENGOLAHAN CITRA DIGITAL**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar sarjana teknik



Disusun Oleh:

FARIZZA ANANTA
NIM: 0710633063 - 63

Telah diperiksa dan disetujui oleh
Dosen Pembimbing

Pembimbing I

Pembimbing II

Adharul Muttaqin, ST., MT.

NIP.19760121 200501 1 001

Ali Mustofa, ST., MT.

NIP. 19710601 200003 1 001

**APLIKASI DETEKSI AWAL KELAINAN PADA *ERITROSIT*
MENGUNAKAN PENGOLAHAN CITRA DIGITAL**

Disusun Oleh :

FARIZZA ANANTA

NIM: 0710633063 - 63

Skripsi ini telah diuji dan dinyatakan lulus

pada tanggal 10 Juli 2012

Majelis Penguji :

Muhammad Aswin, Ir., MT.

NIP. 19640626 199002 1 001

R. Arief Setyawan, ST., MT.

NIP. 19750819 199903 1 001

Waru Djuriatno, ST., MT.

NIP. 19690725 199702 1 001

Mengetahui :

Ketua Jurusan Teknik Elektro

Dr. Ir. Sholeh Hadi Pramono, MS.

NIP. 19580728 198701 1 001

PENGANTAR

Dengan mengucapkan Alhamdulillah Rabbil'alamin, puji syukur kepada Allah SWT yang telah melimpahkan rahmat, ridho, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Aplikasi Deteksi Awal Kelainan Pada *Eritrosit* Dengan Menggunakan Pengolahan Citra Digita” sebagai salah satu syarat yang diajukan untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.

Penulis menyadari bahwa penulisan skripsi ini tidak akan terselesaikan tanpa adanya bantuan dari berbagai pihak karena itu ucapan terima kasih penulis sampaikan kepada semua pihak yang telah dengan rela meluangkan waktunya untuk membantu penulis dalam proses penulisan skripsi dan juga dengan ikhlas memberikan bimbingan, arahan, masukan, dan semangat sehingga penulis dapat menyelesaikan skripsi ini. Oleh karena itu pada kesempatan ini, penulis mengucapkan terima kasih kepada:

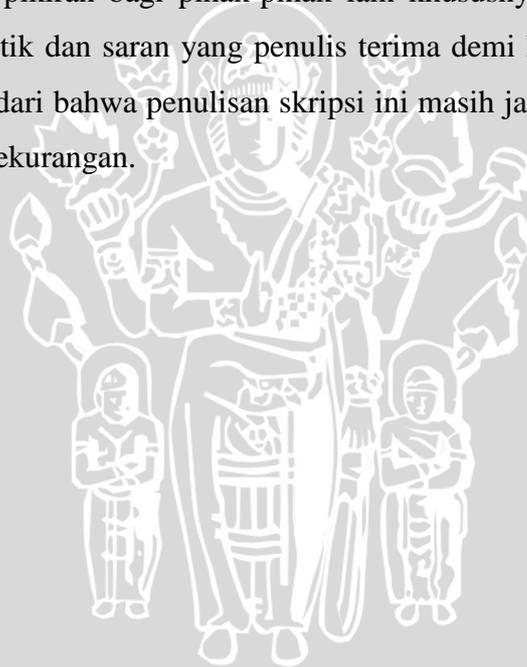
1. Bapak Dr. Ir. Sholeh Hadi Pramono., MS. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
2. Bapak M. Azis Muslim, S.T., M.T., Ph.D selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
3. Bapak Moch. Rif'an. ST., MT. selaku Ketua Program Studi Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.
4. Bapak Waru Djuriatno ST., MT. selaku Ketua Kelompok Dosen Keahlian Rekayasa Komputer Jurusan Teknik Elektro Universitas Brawijaya.
5. Bapak Adharul Muttaqin ST., MT. dan Bapak Ali Mustofa ST., MT. selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyusunan skripsi ini.
6. Ibu Rusmi Ambarwati ST., MT. selaku dosen penasehat akademik yang telah memberikan nasehat dan arahan dalam proses akademik penulis.
7. Dr. Gunawan, C.Ht yang telah memberis arahan dan membantu dalam proses penyusunan skripsi ini.
8. Mama Ir. Veronica Dwi Pihatningsih, Mami Ir. Dahliana Lusi Ratnasari, Abah Ir. Slamet Rivai MBA, Papi Afif Muchsin, Adek Nur Rivsi Muti'ah Rahmanti atas segala dukungan dan do'a serta dorongan semangat hingga terselesaikannya skripsi ini.

9. Fajar Rahmadhy beserta keluarga yang selalu memberi dukungan dan semangat kepada penulis selama proses pengerjaan skripsi ini,
10. Rekan dalam pengerjaan skripsi Ferdiansyah Gustiawan, Asri Samsiar, Fanny Kharisma, Brian QLP, Galih Chandra M, Mas Steven Lewi, Mas Ginanjar Binantoro yang telah membantu memberi solusi pada permasalahan yang muncul dan memberi masukan dalam proses pengerjaan skripsi ini.
11. Seluruh teman-teman angkatan 2007 (CORE '07) yang selalu bersama dalam keceriaan.
12. Balqis, Alfin, Fika, Andan, Fachmy dan Marhein yang merupakan sahabat penulis.
13. Semua pihak yang telah membantu penyelesaian skripsi ini baik secara langsung dan tidak langsung yang tidak dapat penulis sebutkan satu per satu.

Penulis berharap semoga skripsi ini bermanfaat bagi pembaca dan dapat memberikan sumbangan pikiran bagi pihak-pihak lain khususnya mahasiswa jurusan teknik elektro. Segala kritik dan saran yang penulis terima demi kesempurnaan skripsi ini karena penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna dan masih banyak memiliki kekurangan.

Malang, Juni 2012

Penyusun



DAFTAR ISI

PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	ix
ABSTRAK	xi
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Sistematika Penulisan.....	2
BAB II	4
DASAR TEORI	4
2.1 Darah	4
2.2 Sel Darah Merah (<i>Eritrosit</i>).....	5
2.3 Kelainan <i>Eritrosit</i>	6
2.4 Citra.....	7
2.4.1 Citra Analog	7
2.4.2 Citra Digital.....	8
2.5 Pixel dan Resolusi Pixel.....	8
2.6 Jenis-Jenis Citra Digital	8
2.6.1 Citra Biner	8
2.6.2 Citra <i>Grayscale</i>	9
2.6.3 Citra Warna	9
2.7 Pengolahan Citra Digital.....	9
2.7.1 <i>Connected Component Labeling</i>	9
2.8 Segmentasi	10
2.8.1 Metode Otsu	10
2.9 Pengolahan Tranformasi Geometri	12
2.9.1 Resize	12
2.9.2 Cropping.....	13



2.10	Deteksi Tepi	13
2.10.1	Metode Prewitt	15
BAB III.....		17
3.1	Studi Literatur	17
3.2	Penentuan Spesifikasi Alat.....	17
3.3	Perancangan dan Implementasi Sistem.....	17
3.4	Pengambilan Data	18
3.5	Pengujian dan Analisis	18
3.6	Kesimpulan dan Saran.....	21
BAB IV.....		22
PERANCANGAN DAN IMPLEMENTASI.....		22
4.1	Perancangan Secara Umum.....	22
4.1.1	Blok Sistem Diagram	22
4.1.2	Cara Kerja Aplikasi	23
4.2	Perancangan Perangkat Lunak	23
4.2.1	<i>Preprocess</i>	24
4.2.2	Proses Awal.....	27
4.2.2.1	Sub Proses Pengambilan Objek.....	29
4.2.2.2	Perancangan Sub Pengecekan Warna.....	33
4.2.2.3	Sub Proses Deteksi Tepi.....	34
4.2.3	Proses Akhir	36
4.3	Implementasi Sistem	44
4.3.1	Lingkungan Implementasi.....	44
4.2	Implementasi <i>Interface</i> (Antarmuka).....	45
4.2.1	Implementasi Antarmuka <i>Load Image</i>	45
4.2.2	Implementasi Antarmuka Objek Sel Darah Merah.....	46
4.2.3	Implementasi Antarmuka Analisis.....	46
BAB V		47
5.1	Pengujian Pengambilan Objek	47
5.2	Pengujian Identifikasi Objek.....	50
5.3	Analisis Faktor Kegagalan	52
BAB VI.....		54
PENUTUP.....		54

6.1 Kesimpulan	54
6.2 Saran.....	54
DAFTAR PUSTAKA.....	56



DAFTAR GAMBAR

Gambar 2.1	Gambar sel darah merah dari sampng dan dari atas.....	5
Gambar 2.2	Sel darah merah.....	6
Gambar 2.3	Kelainan bentuk <i>eritrosit</i>	7
Gambar 2.4	<i>Connected component labeling</i>	10
Gambar 2.5	Citra hasil metode otsu.....	11
Gambar 2.6	Sebuah citra di-cropping sebesar width x height	13
Gambar 2.7	Gambar tepi step	14
Gambar 2.8	Gambar tepi ramp.....	14
Gambar 2.9	Gambar tepi line.....	14
Gambar 2.10	Gambar tepi step-line	15
Gambar 2.11	Gambar deteksi tepi prewitt	16
Gambar 3.1	Rencana sistem pendeteksi kelainan darah secara keseluruhan..	18
Gambar 4.1	Blok diagram sistem.....	22
Gambar 4.2	Detail desain aplikasi	24
Gambar 4.3	<i>Flowchart Preprocess</i>	25
Gambar 4.4	<i>Flowchart</i> proses awal	28
Gambar 4.5	<i>Flowchart</i> pengambilan objek.....	29
Gambar 4.6	<i>Flowchart</i> pengecekan warna	33
Gambar 4.7	<i>Flowchart</i> deteksi tepi.....	34
Gambar 4.8	<i>Flowchart</i> analisis objek	37
Gambar 4.9	<i>Flowchart</i> analisis objek	38
Gambar 4.10	Antarmuka <i>load image</i>	45
Gambar 4.11	Antarmuka objek sel darah merah.....	46
Gambar 4.12	Antarmuka Analisis.....	46

DAFTAR TABEL

Tabel 5.1	Tabel Pengujian Pengambian Objek.....	47
Tabel 5.2	Tabel Pengujian Identifikasi Objek.....	51
Tabel 5.3	Tabel Hasil Analisis Identifikasi Objek.....	51



ABSTRAK

FARIZZA ANANTA, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Juni, 2012. *Aplikasi Deteksi Awal Kelainan Pada Eritrosit Menggunakan Pengolahan Citra Digital*. Dosen Pembimbing: Adharul Muttaqin, ST., MT. dan Ali Mustofa, ST., MT.

Darah terdiri dari bagian padat yaitu sel darah putih (*leukosit*), sel darah merah (*eritrosit*), trombosit dan bagian cairan yang berwarna kekuningan yang disebut plasma. Perubahan metabolisme dalam tubuh dapat menyebabkan kerusakan pada bentuk *eritrosit* normal. Kelainan bentuk pada *eritrosit* ini dimanfaatkan para ahli untuk proses awal deteksi adanya penyakit atau gangguan dalam tubuh. Untuk mengetahui adanya kelainan bentuk pada *eritrosit* dapat diketahui dengan cara melakukan pemeriksaan ke laboratorium klinik yang ada. Dengan adanya teknologi digital diharapkan dapat mempermudah dan mempercepat proses analisis dokter terhadap pasien yang berpotensi adanya kelainan pada sel darah merah dan sebagai suatu data untuk menunjang analisis berikutnya.

Skripsi ini membahas tentang bagaimana cara mendeteksi adanya kelainan bentuk pada sel darah merah. Dimulai dengan memisahkan objek sel darah merah dari *background* yang kemudian masing-masing dari objek sel darah merah diberi label dengan proses *connected component labeling*. Proses yang selanjutnya dilakukan adalah *cropping* yang dilakukan pada setiap objek ditambah dengan deteksi tepi untuk menentukan karakteristik berdasarkan perbandingan nilai varian dari lingkaran dalam dan lingkaran luar.

Pengujian terhadap 20 sampel citra darah didapatkan 90% sampel yang dapat terambil dan diidentifikasi dengan benar. Kegagalan identifikasi antara lain disebabkan adanya objek yang bertumpuk yang dikenali sebagai satu objek.

Kata kunci: *Eritrosit, Image Processing, Kelainan.*

BAB I PENDAHULUAN

1.1 Latar Belakang

Darah merupakan salah satu bagian penting dalam tubuh. Darah terdiri dari bagian padat yaitu sel darah putih (*leukosit*), sel darah merah (*eritrosit*), trombosit dan bagian cairan yang berwarna kekuningan yang disebut plasma.

Sel darah merah (*eritrosit*) berfungsi untuk mengangkut oksigen yang diperlukan oleh tubuh. *Eritrosit* normal mempunyai bentuk bikonkaf dengan diameter sekitar 7-8 mikron dengan ketebalan bagian tepi sekitar 2 mikron dan bagian tengah sekitar 1 mikron. Perubahan metabolisme dalam tubuh dapat menyebabkan kerusakan pada bentuk *eritrosit* normal. Kelainan bentuk pada *eritrosit* ini dimanfaatkan para ahli untuk proses awal deteksi adanya penyakit atau gangguan dalam tubuh. Untuk mengetahui adanya kelainan bentuk pada *eritrosit* dapat diketahui dengan cara melakukan pemeriksaan ke laboratorium klinik yang ada. Namun cara ini membutuhkan tingkat ketelitian yang tinggi pada pengamatnya.

Dengan berkembangnya teknologi pengolahan citra digital, dapat dibuat suatu software yang dapat membantu deteksi adanya kelainan bentuk pada *eritrosit*. Deteksi adanya kelainan bentuk pada *eritrosit* dilakukan dengan memanfaatkan teknologi *image processing*. Untuk mengetahuinya adanya kelainan bentuk pada sel darah merah dapat dilihat dari karakteristik objek. Bentuk sel darah merah yang bikonkaf apabila dilihat secara dua dimensi dapat dikatakan seperti dua lingkaran yang mempunyai jari-jari yang berbeda. Dari bentuk tersebut dapat dicari jarak dari dua lingkaran, apabila jarak tidak memenuhi dalam *range* yang telah ditetapkan maka sel darah merah tersebut berpotensi adanya kelainan dan sebaliknya.

Diharapkan hasil dari pengolahan citra ini dapat mempermudah dan mempercepat proses analisis dokter terhadap pasien yang berpotensi adanya kelainan pada sel darah merah dan sebagai suatu data untuk menunjang analisis berikutnya.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dijelaskan pada latar belakang, maka rumusan masalah ditekankan pada:

1. Bagaimana cara memisahkan satu citra *eritrosit* yang ada dalam citra darah.
2. Bagaimana cara mengidentifikasi adanya kelainan bentuk pada tiap *eritrosit* yang terdapat pada citra darah.

1.3 Batasan Masalah

Beberapa hal yang menjadi batasan masalah dalam pembuatan program ini antara lain:

1. Citra yang diolah adalah hasil pemotretan darah dengan menggunakan kamera dan mikroskop khusus tanpa membahas proses pemotretannya.
2. Kelainan yang dibahas hanya sebatas kelainan bentuk pada *eritrosit* tanpa mengklarifikasikan jenis kelainan.
3. Pengenalan kelainan *eritrosit* berdasarkan anatomi sel darah merah yang terdapat dalam literatur.
4. Perancangan meliputi algoritma mulai dari pembacaan citra digital hingga deteksi adanya kelainan pada darah.
5. Bahasa pemrograman yang dipakai adalah Microsoft Visual Studio .NET C# dengan bantuan A Forge framework.

1.4 Tujuan

Tujuan dari skripsi ini adalah untuk merancang dan membuat suatu aplikasi yang mampu mendeteksi adanya kelainan pada *eritrosit* sebagai alat bantu yang berfungsi untuk mempermudah dan mempercepat analisis dokter dan juga sebagai data rekam medis yang menunjang analisis dokter selanjutnya.

1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan laporan skripsi ini adalah sebagai berikut.

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II Dasar Teori

Membahas dasar teori yang digunakan dalam perancangan dan pembuatan perangkat lunak.

BAB III Metodologi

Menjelaskan metode yang digunakan dalam pengerjaan skripsi berikut langkah – langkah yang akan diambil.

BAB IV Perancangan dan Implementasi

Menjelaskan langkah langkah perancangan aplikasi untuk mendeteksi adanya kelaianan bentuk pada *eritrosit* berikut penjelasan algoritma yang digunakan dan penjelasan dari setiap langkah-langkahnya.

BAB V Pengujian

Membahas pengujian dari sistem yang telah dibuat berikut analisis dari hasilnya.

BAB VI Kesimpulan dan Saran

Berisi Kesimpulan dan Saran

BAB II DASAR TEORI

2.1 Darah

Darah merupakan medium transport tubuh, yang mempunyai fungsi utama mengangkut oksigen yang diperlukan oleh sel-sel di seluruh tubuh. Darah juga menyuplai jaringan tubuh dengan nutrisi, mengangkut zat-zat sisa metabolisme, dan mengandung berbagai bahan penyusun sistem imun yang bertujuan mempertahankan tubuh dari berbagai penyakit. Volume darah manusia sekitar 2%-10% berat badan normal dan berjumlah sekitar 5 liter. Keadaan jumlah darah pada tiap-tiap orang tidak sama, bergantung pada usia, pekerjaan, serta keadaan jantung atau pembuluh darah. Darah terdiri atas 2 komponen utama, yaitu sebagai berikut:

1. Plasma darah, bagian cairan darah yang sebagian besar terdiri atas air, elektrolit, dan protein darah.
2. Butir-butir darah (*blood corpuscles*), yang terdiri atas komponen-komponen berikut ini:

- a. Eritrosit

Sel darah merah (*SDM-red blood cell*) adalah jenis sel darah yang paling banyak dan berfungsi membawa oksigen ke jaringan-jaringan tubuh lewat darah.

- b. Leukosit

Sel darah putih (*SDP-white blood cell*) berfungsi untuk membantu tubuh melawan berbagai penyakit infeksi sebagai bagian dari sistem kekebalan tubuh. Sel darah putih tidak berwarna, memiliki inti, dapat bergerak secara amoeboid, dan dapat menembus dinding kapiler / diapedesis.

- c. Trombosit

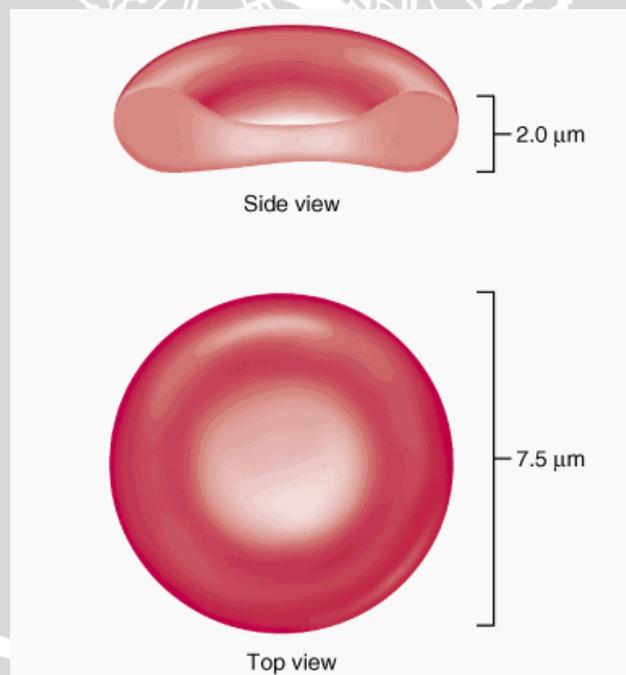
Trombosit adalah fragmen atau kepingan-kepingan tidak berinti dari sitoplasma megakariosit yang berukuran 1-4 mikron dan beredar dalam sirkulasi darah selama 10 hari. Fungsi utama trombosit adalah melindungi pembuluh darah terhadap kerusakan

endotel akibat trauma-trauma kecil yang terjadi sehari-hari dan mengawali penyembuhan luka pada dinding pembuluh darah.

Pada tugas akhir ini pendeteksian difokuskan pada sel darah merah (*eritrosit*).

2.2 Sel Darah Merah (*Eritrosit*)

Sel darah merah merupakan suatu sel yang berbentuk bikonkaf dengan diameter sekitar 7-8 mikron, dengan tebal bagian tepi 2 mikron dan pada bagian tengah tebalnya hanya 1 mikron atau kurang. Bentuk bikonkaf merupakan bentuk cakram dengan bentuk tengah agak gepeng, bila dilihat dari segi visual bentuk bikonkaf menyerupai bentuk sebuah lingkaran. Bikonkavitas memungkinkan gerakan oksigen masuk dan keluar sel secara cepat dengan jarak yang pendek antara membran dan inti sel. Warnanya kuning kemerah-merahan, karena di dalamnya mengandung suatu zat yang disebut hemoglobin. Untuk lebih jelas bentuk dari sel darah merah, dapat di lihat dibawah ini.



Gambar 2.1 Gambar Sel Darah Merah dari Samping dan dari Atas

Sumber: legacy.owensboro.kctcs.edu

Sel darah merah tidak mempunyai inti sel, mitokondria dan ribosom, serta tidak dapat bergerak. Sel ini tidak dapat melakukan mitosis, fosforilasi oksidatif sel, atau pembentukan protein.



Gambar 2.2 Sel Darah Merah

Sumber: NIH: Noguchi, Rodgers, and Schechter

Komponen eritrosit adalah sebagai berikut:

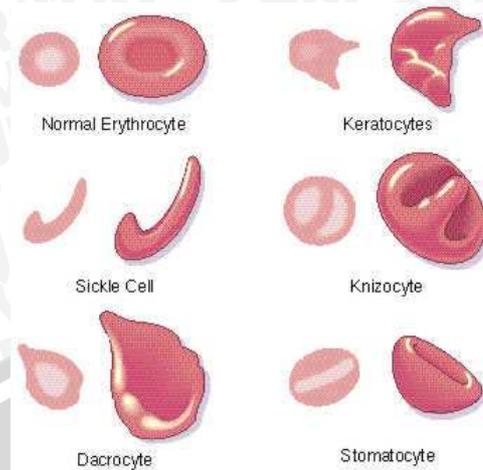
1. Membran eritrosit.
2. Sistem enzim: enzim G6PD (*Glucose 6-Phosphatedehydrogenase*).
3. Hemoglobin, komponennya terdiri atas:
 - a. Heme yang merupakan gabungan protoporfirin dengan besi;
 - b. Globin: bagian protein yang terdiri atas 2 rantai alfan dan 2 rantai beta.

Terdapat sekitar 300 molekul hemoglobin dalam setiap sel darah merah. Hemoglobin berfungsi untuk mengikat oksigen, satu gram hemoglobin akan bergabung dengan 1,34 ml oksigen.

Tugas akhir hemoglobin adalah menyerap karbondioksida dan ion hydrogen serta membawanya ke paru-paru tempat zat-zat tersebut dilepaskan hemoglobin.

2.3 Kelainan Eritrosit

Kelainan dan penyakit pada darah dan sistem peredaran darah dapat disebabkan oleh faktor keturunan dan non keturunan. Contoh bentuk abnormal eritrosit yaitu bentuk bulan sabit, bentuk hlem, bentuk target, bentuk seperti durian / irregular, bentuk pensil, bentuk tetesan atau teardrop dll.



Gambar 2.3 Kelainan Bentuk Eritrosit

Sumber: pathology.vcu.edu

2.4 Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebafai keluaran suatu perekaman data dapat bersifat optic berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisive, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan.

2.4.1 Citra Analog

Citra analog adalah citra yang bersifat kontinu, seperti pada gambar televisi, foto sinar-X, foto yang tercetak di kertas foto, lukisan, pemandangan alam, hasil CT scan, dan lain sebagainya. Citra analog tidak dapat direpresentasikan dalam komputer sehingga tidak bisa diproses secara langsung.

2.4.2 Citra Digital

Citra digital merupakan suatu larik dua dimensi atau suatu matriks yang elemen-elemennya menyatakan tingkat keabuan dari elemen gambar. Jadi informasi yang terkandung bersifat diskret. Citra digital tidak selalu merupakan hasil langsung data rekaman suatu sistem. Kadang-kadang hasil rekaman data bersifat kontinu seperti gambar pada monitor televisi, foto sinar-X, dan lain sebagainya. Dengan demikian untuk mendapatkan suatu citra digital diperlukan suatu proses konversi, sehingga citra tersebut selanjutnya dapat diproses dengan komputer.

2.5 Pixel dan Resolusi Pixel

Pada suatu citra definisi dari pixel adalah bagian terkecil dari suatu citra atau image yang tidak hanya mewakili satu titik melainkan berupa kotak dan juga dapat sebagai ukuran dari suatu citra. Resolusi pixel merupakan perhitungan jumlah pixel dalam suatu citra. Pada sebuah citra dengan tinggi N pixel dan lebar M pixel berarti memiliki resolusi sebesar $M \times N$ pixel.

2.6 Jenis-Jenis Citra Digital

Ada banyak cara untuk menyimpan citra digital di dalam memori. Cara penyimpanan menentukan jenis citra digital yang terbentuk. Beberapa jenis yang sering digunakan adalah citra biner, citra *grayscale*, dan citra warna.

2.6.1 Citra Biner

Citra biner hanya terdiri dari dua nilai pixel, yaitu hitam dan putih. Citra biner juga disebut dengan citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap pixel dari citra biner.

Gradasi warna:



Bit 0 = warna hitam

Bit 1 = warna putih

2.6.2 Citra Grayscale

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian RED = GREEN = BLUE, yang digunakan untuk tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan dan putih. Tingkat keabuan disini merupakan warna abu dengan berbagai tingkat dari hitam hingga mendekati putih.

2.6.3 Citra Warna

Setiap pixel dari citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = *Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 *byte*, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Citra warna disebut juga dengan *true color* karena mempunyai jumlah warna yang cukup besar sehingga dikatakan hampir mencakup semua warna di alam.

2.7 Pengolahan Citra Digital

Pada dasarnya pengolahan citra digital dilakukan untuk memproses suatu citra agar sesuai dengan keinginan kita dan juga agar dapat dilakukan analisis lebih lanjut mengenai citra yang menjadi masukan sebab pada suatu citra masukan biasanya terdapat suatu informasi yang perlu untuk dianalisis lebih lanjut. Proses pengolahan citra digital dapat diartikan secara umum sebagai pemrosesan citra dua dimensi oleh computer.

2.7.1 Connected Component Labeling

Connected component labeling adalah metode pengelompokan suatu objek berdasarkan *pixel* yang saling berhubungan. Setelah setiap objek dikelompokkan, maka pixel disetiap objek diberi label dengan level keabuan atau level warna yang berbeda.

Cara kerja dari metode ini adalah dengan memindai citra pixel demi pixel dari pojok kiri atas sampai pojok kanan bawah untuk mengidentifikasi daerah dimana pixel yang berhubungan.

Setelah selesai memberi label pada setiap objek yang ada, setiap label kemudian dikelompokkan dalam kelas tertentu, dan setiap label yang memiliki keunikan dikelompokkan ke dalam setiap kelas, dimana setiap kelas memiliki label yang berbeda.



Gambar 2.4 *Connected component labeling*

Sumber : 4.bp.blogspot.com.

2.8 Segmentasi

Segmentasi merupakan teknik membagi suatu citra menjadi beberapa daerah dimana setiap daerah memiliki kemiripan atribut. Proses segmentasi dilakukan agar dapat membedakan obyek yang diteliti dengan latar belakangnya.

Proses segmentasi secara sederhana dapat dilakukan dengan proses binerisasi dimana citra awal yang memiliki warna kemudian diubah menjadi citra grayscale selanjutnya dengan menggunakan teknik pengambangan yang umum digunakan maka akan dapat mencari nilai threshold yang sesuai dengan image atau citra yang sedang kita proses dengan teknik pengambangan ini maka akan menghasilkan citra biner. Dimana untuk menghasilkan citra biner

(2.1)

Dengan $g(x,y)$ adalah citra biner dari citra grayscale $f(x,y)$, dan T menyatakan nilai ambang. Nilai T memegang peranan penting dalam proses pengambangan. Kualitas hasil citra biner sangat bergantung pada nilai T yang digunakan.

2.8.1 Metode Otsu

Metode Otsu merupakan metode yang digunakan untuk menghitung nilai ambang (T) secara otomatis berdasarkan citra masukan. Tujuan dari metode otsu adalah membagi histogram citra gray level ke dalam dua daerah yang berbeda

secara otomatis tanpa membutuhkan bantuan user untuk memasukkan nilai ambang. Pendekatan yang dilakukan 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 memaksimalkan variabel tersebut agar dapat membagi objek dengan latar belakang.

Nilai ambang yang dicari pada suatu citra *grayscale* dilambangkan dengan k . Nilai k ini berkisar antara 1 sampai dengan L sedangkan L memiliki nilai maksimum 255. Probabilitas untuk pixel i dapat dinyatakan dengan persamaan :

$$p_i = \frac{n_i}{N} \tag{2.2}$$

Dengan n_i menyatakan jumlah pixel dengan tingkat keabuan i dan N menyatakan banyaknya pixel pada citra.

Pada nilai momen kumulatif ke nol, momen kumulatif ke satu, dan nilai rata-rata dapat dinyatakan pada persamaan di bawah ini :

$$m_0 = N \tag{2.3}$$

$$m_1 = \sum_{i=0}^{L-1} i p_i \tag{2.4}$$

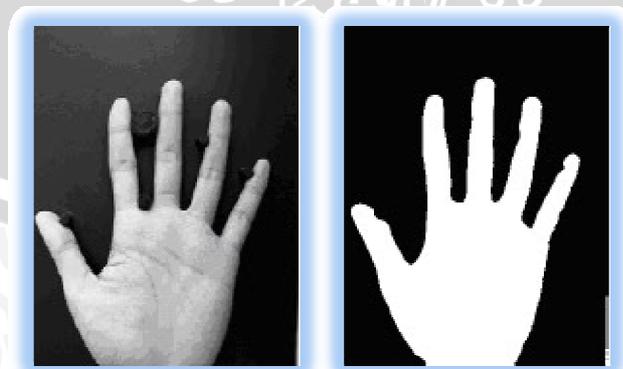
$$\bar{x} = \frac{m_1}{m_0} \tag{2.5}$$

Sedangkan untuk nilai ambang k dapat ditentukan dengan memaksimalkan persamaan :

$$J(k) = \sum_{i=0}^{k-1} p_i^2 + \sum_{i=k}^{L-1} p_i^2 \tag{2.6}$$

dengan :

$$p_i = \frac{n_i}{N} \tag{2.7}$$



Gambar 2.5 Citra hasil metode otsu

Sumber : 4.bp.blogspot.com.

2.9 Pengolahan Tranformasi Geometri

Citra terdiri dari piksel yang di dalamnya terdapat dua komponen yaitu koordinat piksel (x,y) dan intensitas warna. Untuk memenuhi kebutuhan pengolahan citra seringkali diperlukan operasi geometri baik ukuran maupun orientasinya seperti pencerminan (*flipping*), rotasi, pemotongan (*cropping*) penskalaan dengan interpolasi, transpose citra, efek gelombang, efek warp, efek swirl, dan efek glass. Dalam pembuatan aplikasi ini menggunakan teknik tranformasi geometri *resize, cropping*.

2.9.1 Resize

Operasi ini dimaksudkan untuk memperbesar atau memperkecil citra. Hal ini dapat dilakukan dengan mengintroduksikan parameter skala horisontal dan vertikal. Skala yang bernilai lebih dari 1 akan memperbesar citra, sedangkan jika sama dengan 1 menunjukkan citra asli, sedangkan jika kurang dari 1 akan memperkecil citra. Untuk mempertahankan perbandingan antara tinggi dan lebar citra atau *aspect ratio*, maka nilai $S_h = S_v$, tranformasi spasial yang dipakai adalah:

$$X' = S_h X \quad (2.8)$$

$$Y' = S_v Y \quad (2.9)$$

Dimana x' dan y' adalah posisi hasil scalling, x dan y adalah posisi asal, S_h adalah skala horisontal dan S_v adalah skala vertikal.

Dari persamaan persamaan diatas untuk setiap titik pada citra hasil dengan koordinat x' dan y' diketahui, dapat dicari koordinat titik asal x dan y . Untuk mengatasi nilai koordinat titik asal yang tidak bulat dapat digunakan interpolasi untuk menghasilkan citra yang mendekati aslinya. Perubahan posisi piksel juga mengakibatkan perubahan ukuran citra sesuai degan persamaan berikut:

$$\text{Width} = S_h w \quad (2.10)$$

$$\text{Height} = S_v h \quad (2.11)$$

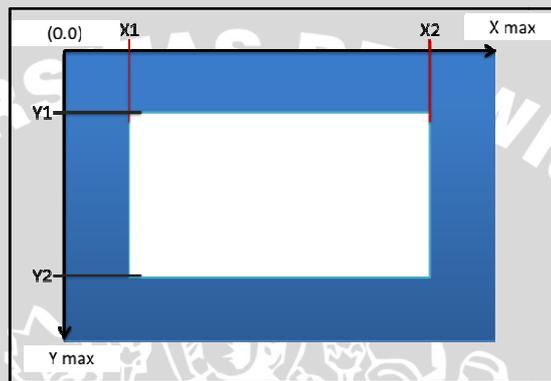
Dimana width' adalah lebar hasil dan height' adalah tinggi hasil, w adalah lebar asal, h adalah tinggi asal, S_h adalah faktor skala horisontal, dan S_v adalah faktor skaling vertical.

2.9.2 Cropping

Dalam dunia fotografi dan percetakan cropping adalah teknik pengolahan citra untuk memotong area pada gambar gambar atau citra. Cropping merupakan salah satu dasar teknik manipulasi gambar yang digunakan untuk menghilangkan bagian yang tidak di inginkan dari gambar, teknik ini digunakan untuk memotong suatu bagian dari citra sehingga diperoleh citra yang berukuran lebih kecil.

$$\text{Width} = X2 - X1 \quad (2.12)$$

$$\text{Height} = Y2 - Y1 \quad (2.13)$$



Gambar 2.6. Sebuah Citra di-cropping sebesar Width x Height

Sumber : Sutoyo, T., Mulyanto, Edy., Suhartono, Vincent., Nurhayati, Oky Dwi., Wijanarto. Teori Pengolahan Citra Digital. 2009.

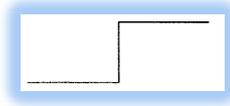
2.10 Deteksi Tepi

Pada suatu tepian citra mengandung informasi penting dari citra yang bersangkutan. Tepian citra dapat merepresentasikan bentuk, ukuran dari obyek-obyek yang ada pada citra selain itu juga terkadang terdapat informasi tentang tekstur dari suatu citra. Tepian dari suatu citra adalah posisi dimana intensitas pixel dari citra berubah dari nilai rendah ke nilai yang tinggi atau sebaliknya.

Tepian citra dapat dilihat melalui perubahan intensitas perubahan pixel pada suatu area. Berdasarkan perbedaan perubahan intensitas tersebut tepian dapat dibagi menjadi 4 jenis diantaranya :

1. Step

Tepian jenis step merupakan tepian citra yang terbentuk dari perubahan intensitas citra secara signifikan dari tinggi ke rendah ataupun sebaliknya. Tepian step dapat digambarkan seperti di bawah ini:



Gambar 2.7 Gambar tepi step

Sumber : Darma Putra. Pengolahan Citra Digital. 2010.

2. Ramp

Tepian jenis ini terbentuk dari perubahan intensitas citra secara perlahan. Perubahan secara perlahan dapat dilihat pada bentuk kurva yang semakin tinggi dengan perubahan kontinu. Tepian jenis *ramp* dapat dilihat pada gambar dibawah ini :

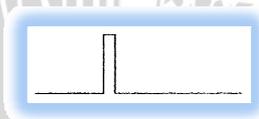


Gambar 2.8 Gambar tepi ramp

Sumber : Darma Putra. Pengolahan Citra Digital. 2010.

3. Line

Tepian jenis ini ditandai dengan perubahan intensitas secara drastis dari intensitas rendah-tinggi-rendah atau sebaliknya. Tepian *line* dapat ditunjukkan pada gambar di bawah ini :

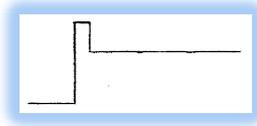


Gambar 2.9 Gambar tepi line

Sumber : Darma Putra. Pengolahan Citra Digital. 2010.

4. Step-line

Tepian *step-line* merupakan gabungan dari tepian jenis *step* dan *line*. Tepian jenis ini ditandai dengan peningkatan intensitas yang tajam dalam interval tertentu dan kemudian ditandai dengan penurunan yang tidak signifikan, sehingga perubahan selanjutnya berlangsung stabil.



Gambar 2.10. Gambar tepi step-line

Sumber : Darma Putra. Pengolahan Citra Digital. 2010.

Terdapat berbagai operator deteksi tepi yang telah dikembangkan berdasarkan turunan pertama (*first order derivative*) di antaranya operator *Robert*, *Sobel*, *Prewitt*, *Krisch* dan *Canny*. Konsep dasar perhitungan deteksi tepi menggunakan turunan pertama adalah dengan memanfaatkan perbedaan nilai suatu pixel dengan pixel tetangganya seperti persamaan di bawah ini:

$$(2.14)$$

$$(2.15)$$

Δ_y menyatakan perbedaan nilai pixel dalam arah y, sedangkan Δ_x menyatakan perbedaan nilai pixel dalam arah x. Dalam proses konvolusi, Δ_y diwakili dengan matrik 2 x 1 seperti dibawah ini :

$$(2.16)$$

Sedangkan Δ_x diwakili dengan matrik 1 x 2 seperti di bawah ini :

$$(2.18)$$

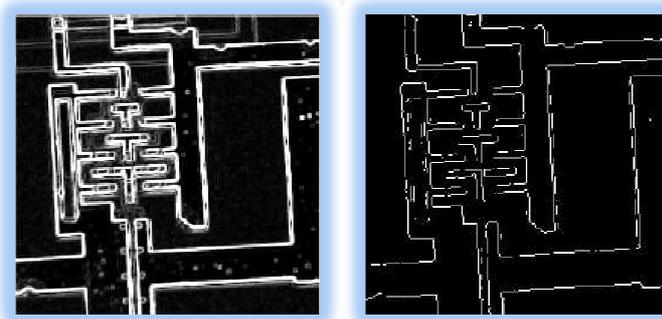
Matrik pada Δ_y dan Δ_x dapat digunakan untuk menghitung nilai gradien sebagai berikut :

$$(2.19)$$

2.10.1 Metode Prewitt

Pada algoritma metode prewitt sebenarnya mirip dengan Sobel karena operator yang digunakan adalah 3x3. Matriks ini juga dikonvolusikan dengan citra masukan dalam arah x dan y. Filter ini memiliki kelebihan “cepat” dibandingkan dengan Sobel. Tapi kernelnya hanya cocok untuk citra yang memiliki kontras tinggi dan derau yang sangat kecil. Misal pada susunan dari pixel-pixel di sekitar pixel $p_0 = (x,y)$ seperti pada gambar di bawah ini





Gambar 2.11 Gambar deteksi tepi prewitt

Sumber : Alasdair McAndrew. An Introduction to Digital Image Processing with Matlab. 2004

Dengan operator prewitt dapat menghitung besaran gradien dari persamaan :

$$\frac{\partial I}{\partial x} \quad (2.20)$$

Dimana nilai M adalah besaran gradien yang dihitung pada titik tengah jendela dan turunan parsial dihitung dengan :

$$(2.21)$$

$$(2.22)$$

Dengan menggunakan nilai konstanta c bernilai 1 maka dalam operator prewitt menjadi sebuah kernel seperti di bawah ini :

$$(2.23)$$

Tidak seperti operator sobel yang meletakkan penekanan dan pembobotan pada pixel-pixel yang lebih dekat dengan titik pusat akan tetapi operator prewitt pada ke-8 pixel tetangganya mempunyai pengaruh yang sama dalam perhitungan gradien.

BAB III

METODOLOGI

Metode penelitian yang digunakan dalam penyusunan skripsi perancangan aplikasi yang mampu membedakan adanya kelainan pada sel darah merah, sehingga dapat memilah antar sel darah merah normal dan sel darah abnormal.

3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

1. Membaca dan mempelajari buku – buku yang berhubungan dengan *image processing* dan patologi klinik.
2. Mempelajari metode pendeteksian objek pada citra.
3. Mempelajari teknik – teknik dasar pemrograman dengan menggunakan Microsoft Visual Studio .NET

3.2 Penentuan Spesifikasi Alat

Menentukan perangkat yang akan digunakan untuk menunjang aplikasi:

1. Perangkat Keras
 - unit laptop dengan spesifikasi Intel Core 2 Duo processor T6400 2.0GHz , DDR3 1 GB, VGA Intel Graphic Media Accelerator 4500M HD.
2. Perangkat Lunak
 - *Operating system* Windows 7 Ultimate
 - Microsoft visual studio .NET C#
 - A Forge Framework

3.3 Perancangan dan Implementasi Sistem

Secara garis besar desain yang akan dibuat untuk aplikasi pendeteksi kelainan bentuk pada *eritrosit* terdiri dari beberapa langkah. Berikut ini desain

rancangan sistem pendeteksi kelainan bentuk pada *eritrosit*:



Gambar 3.1 Rancangan sistem pendeteksi kelainan darah secara keseluruhan

Citra darah sebagai masukan dari aplikasi ini diproses di dalam sebuah perangkat keras yang di dalamnya terdapat perangkat lunak yang berfungsi sebagai pengolah citra diantaranya segmentasi, *cropping* dan deteksi tepi untuk menentukan apakah sampel citra sel darah merah normal atau tidak.

3.4 Pengambilan Data

Pengambilan data dilakukan untuk memperoleh data-data yang diperlukan untuk menyelesaikan skripsi ini. Data yang diperoleh merupakan data sekunder yang didapat dari berbagai macam buku referensi, jurnal dan internet yang telah divalidasi oleh dokter.

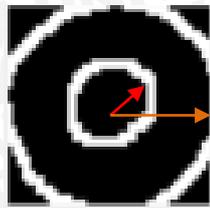
3.5 Pengujian dan Analisis

Pada tahap ini pengujian sistem dilakukan untuk memastikan bahwa sistem yang telah dirancang memiliki tingkat kesalahan atau error yang sangat kecil. Agar dapat diketahui apakah sistem yang telah dirancang dapat bekerja dengan baik sesuai dengan perancangan yang telah dibuat sebelumnya maka akan dilakukan beberapa pengujian yang berangkaian dan berurutan. Tahapan rangkaian pengujian adalah sebagai berikut:

1. Pengujian program pendeteksi kelainan pada sel darah merah.

Untuk mengetahui adanya kelainan pada sel darah merah, maka perlu diketahui penjumlahan nilai dari r_1 dan r_2 , dimana r_1 adalah

jarak pusat ke lingkaran dalam, sedangkan r2 adalah jarak titik pusat ke lingkaran luar.



Keterangan:

r1 : panah merah

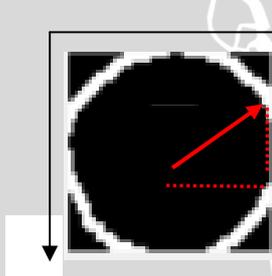
r2 : panah orange

Sebelum mencari nilai dari r1 dan r2, maka harus ditentukan titik tengah pusat dr lingkaran, dimana titik pusat tersebut adalah titik tengah citra, yang dapat dicari dengan menggunakan persamaan seperti di bawah ini:

$$\text{center_x} = \text{lebar citra} / 2 \tag{3.1}$$

$$\text{center_y} = \text{tinggi citra} / 2 \tag{3.2}$$

center_x menyatakan titik tengah untuk koordinat x, sedang center_y menyatakan titik tengah untuk koordinat y .Setelah ditemukan koordinat titik pusat lingkaran, maka selanjutnya adalah menemukan koordinat dari r1 dan r2 menggunakan persamaan berikut:



$$x = r \cos \alpha \tag{3.3}$$

$$y = r \sin \alpha \tag{3.4}$$

Setelah menemukan koordinat untuk r1 dan r2, maka langkah selanjutnya adalah mencari jarak koordinat (x,y) dari titik pusat dengan menggunakan persamaan dibawah ini:

$$x_n = \text{center_x} + x \tag{3.5}$$

$$y_n = \text{center_y} + y \tag{3.6}$$

Xn adalah jarak koordinat sumbu x dan Yn adalah jarak koordinat y. langkah selanjutnya adalah mencari nilai rata-rata r1 dan r2 dengan menggunakan persamaan berikut:

$$\bar{r} = \frac{\sum_{i=0}^{n=359} r_i}{n} \quad (3.7)$$

Persamaan diatas berlaku untuk r1 dan r2.

Langkah selanjutnya adalah mencari nilai varian r1 dan r2. Rumus yang digunakan untuk mencari nilai varian adalah sebagai berikut:

$$s^2 = \frac{\sum (r_i - \bar{r})^2}{n - 1} \quad (3.8)$$

Keterangan:

s^2 : nilai varian

r_i : data ke i

\bar{r} : nilai rata-rata

n : jumlah data

Setelah masing nilai varian telah ditemukan, maka selanjutnya adalah membagi nilai varian r1 dan nilai varian r2.

2. Pengujian program terhadap citra darah.
3. Pengujian pengambilan objek.

Untuk pengujian pengambilan objek, nilai error dapat dihitung dengan persamaan berikut:

$$\text{error} = \frac{\text{selisih perhitungan mata dengan program}}{\text{perhitungan dengan mata}} \times 100\%$$

Keterangan :

- Error = nilai error dari sistem untuk menentukan faktor kegagalan
- Selisih perhitungan = merupakan jumlah total selisih perhitungan sel darah merah dengan perhitungan secara mata dan perhitungan secara program

- Perhitungan dengan mata = merupakan jumlah sel darah merah yang dihitung dengan mata
4. Tingkat keberhasilan sistem dalam mengidentifikasi objek dapat dihitung dengan persamaan berikut:

$$\text{keberhasilan} = \frac{\text{total benar}}{\text{jumlah seluruh data}} \times 100\%$$

Keterangan :

- Keberhasilan = nilai keberhasilan dari sistem
 - Total benar = merupakan jumlah total nilai benar dari citra yang berhasil diidentifikasi
 - Jumlah seluruh data = merupakan jumlah seluruh data
5. Analisis faktor kegagalan dalam sistem.

3.6 Kesimpulan dan Saran

Pada tahap ini, diambil dari hasil pengujian dan analisis terhadap Aplikasi Deteksi Dini Kelainan Pada *Eritrosit*. Tahap Selanjutnya adalah membuat saran untuk perbaikan terhadap penelitian selanjutnya sehingga dapat menyempurnakan kekurangan-kekurangan yang ada.

BAB IV

PERANCANGAN DAN IMPLEMENTASI

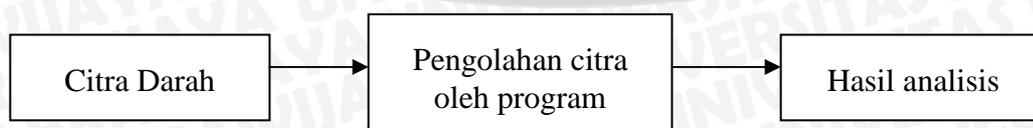
Pada bab ini akan membahas perancangan dan implementasi aplikasi deteksi dini adanya kelainan pada *eritrosit* yang dikerjakan dengan beberapa tahap. Tahap pertama merupakan tahap pre proses dimana proses ini untuk menyiapkan citra. Tahap kedua merupakan tahap pengambilan objek untuk memudahkan proses analisa yang terdapat pada proses selanjutnya. Tahap ketiga merupakan tahapan yang paling akhir dilakukan yang bertujuan untuk menganalisis dan menentukan apakah *eritrosit* tersebut normal atau terdapat kelainan.

4.1 Perancangan Secara Umum

Tahap awal yang dibutuhkan adalah membuat perancangan aplikasi secara global dimana tahap awal ini berfungsi sebagai acuan dalam proses pembuatan aplikasi deteksi dini adanya kelainan pada *eritrosit* yang akan dibuat. Pada tahap perancangan ini diawali terlebih dahulu dengan pendefinisian kegiatan pelaku atau *user* dalam menggunakan program aplikasi pendeteksi deteksi dini adanya kelainan pada *eritrosit* dengan menerapkan beberapa teknik pengolahan citra dan menjelaskan perangkat yang digunakan meliputi blok diagram sistem dan cara kerja aplikasi.

4.1.1 Blok Sistem Diagram

Pada sistem deteksi objek yang menunjukkan adanya kelainan pada darah merah terdiri dari beberapa komponen yang dapat digambarkan menjadi blok diagram seperti pada Gambar 4.1.



Gambar 4.1 Blok Diagram Sistem

Sumber : Perancangan

Fungsi dari masing-masing bagian pada diagram blok ini akan dijelaskan sebagai berikut:

1. Citra darah yang digunakan adalah hasil pemotretan darah dengan menggunakan kamera dan mikroskop khusus tanpa membahas proses pemotretannya.
2. Pengolahan citra mempunyai peranan yang sangat penting dalam sistem ini, karena pada langkah ini menentukan ada atau tidaknya kelainan bentuk pada darah merah.
3. Citra darah yang telah melalui proses pengolahan citra maka di dapat hasil akhir apakah terdapat kelainan bentuk pada darah merah atau tidak.

4.1.2 Cara Kerja Aplikasi

Cara kerja dari aplikasi deteksi dini adanya kelainan pada *eritrosit* ini dimulai dengan *me-load* data yang telah didapatkan dari sumber data sekunder, kemudian dilanjutkan dengan proses *resize* hingga pelabelan komponen sebagai pre proses untuk menyiapkan citra ke proses berikutnya. Setelah melalui pelabelan tiap objek dalam citra diambil dan dianalisis apakah terdapat kelaianan pada sel darah merah atau tidak.

Dalam pembuatan aplikasi ini, juga menggunakan komponen dari A Forge framework yang dibuat oleh Andrew Kirilov yang berfungsi untuk melakukan proses *resize* dan *cropping*.

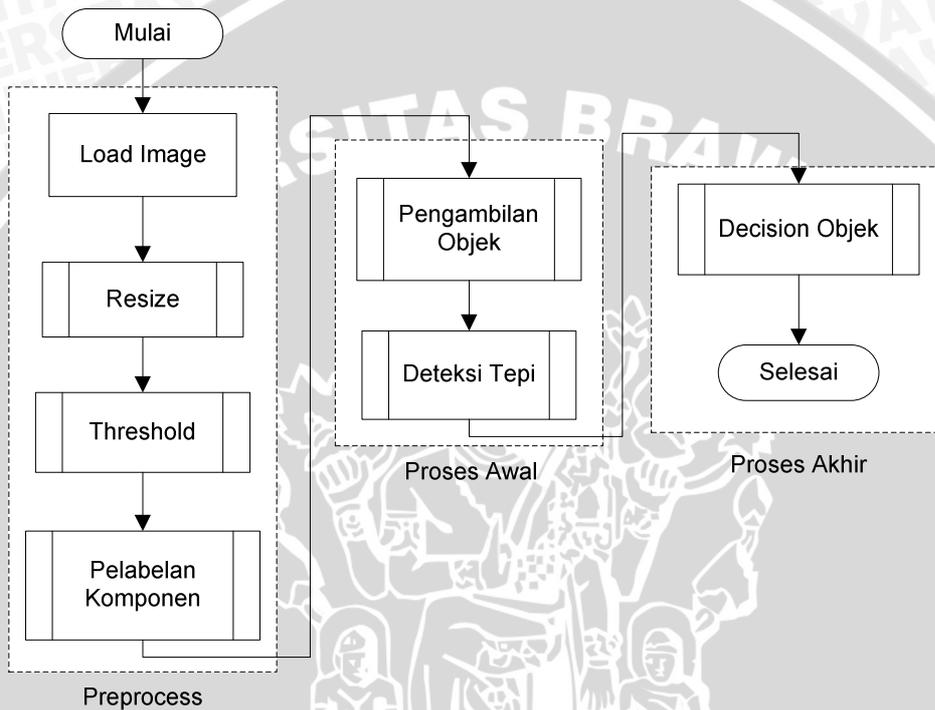
4.2 Perancangan Perangkat Lunak

Pada bagian perancangan ini perangkat lunak yang akan dibuat menggunakan bahasa pemrograman *Microsoft Visual Studio C#.NET 2010* dan sistem yang digunakan untuk membangun perangkat lunak ini dirancang dengan spesifikasi mampu melakukan hal-hal sebagai berikut :

1. Mengakses data citra yang telah tersimpan dalam harddisk computer.
2. Melakukan *resize* pada citra yang akan diolah menjadi ukuran 320x240.

3. Mengubah citra menjadi citra biner dengan melakukan proses threshold secara otomatis pada citra.
4. Memberikan label yang berbeda pada setiap objek.
5. Melakukan proses cropping untuk pengambilan tiap objek.
6. Melakukan proses deteksi tepi menggunakan operator prewitt.
7. Mampu menganalisis objek.

Untuk desain aplikasi secara umum akan ditunjukkan pada Gambar 4.2.

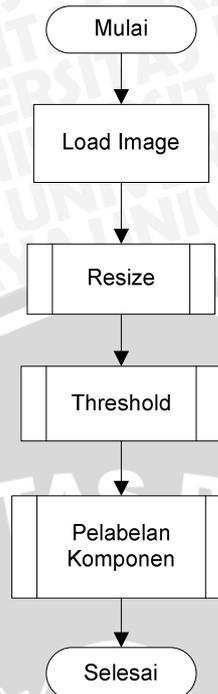


Gambar 4.2 Detail Desain Aplikasi

Sumber : Perancangan

4.2.1 Preprocess

Pada tahap ini citra darah akan melalui proses *resize* yang kemudian citra diubah menjadi citra biner dengan mencari nilai *threshold* secara otomatis dengan menggunakan metode otsu dan selanjutnya tiap objek akan ditandai. Flowchart *preprocess* seperti pada Gambar 4.3.



Gambar 4.3 Flowchart Preprocess

Sumber : Perancangan

Berikut penjelasan dari *flowchart preprocess* :

1. Hasil citra yang akan diproses masih memiliki ukuran yang beragam oleh karena itu citra akan di *resize* menjadi ukuran 320x240. Proses *resize* untuk menyamakan ukuran untuk mempercepat proses.
2. Setelah proses *resize* pada citra maka proses selanjutnya adalah proses segmentasi dimana proses ini dilakukan untuk mengubah citra mengubah citra warna menjadi citra biner. Proses binerisasi dilakukan dengan mendapatkan nilai ambang (*threshold*) secara otomatis menggunakan metode otsu.
3. Karena terdapat banyak objek dalam citra biner maka digunakan operasi pelabelan objek untuk mempermudah proses selanjutnya.

Implementasi dari proses *preprocess* pada bahasa C# adalah sebagai berikut:

- Proses Resizing

Data yang menjadi input pada aplikasi ini awalnya memiliki resolusi yang beragam karena itu perlu dilakukan proses *resize* pada

citra dengan tujuan untuk mengurangi iterasi dalam penyeleksian piksel dan juga mempercepat proses pada citra. Data citra input akan di resize ukurannya menjadi 320x240. Proses resize ini akan dilakukan dengan menggunakan bantuan Aforge Framework dan berikut listing program dalam syntax C# :

1. *Using Statement* yang digunakan untuk Aforge Framework


```
using AForge.Imaging.Filters;
```
 2. *Meresize* ukuran menjadi 320x240


```
ResizeBicubic resize = new ResizeBicubic(320, 240);
```
 3. Menerapkan *resize* pada citra


```
pictureBox1.Image = resize.Apply(resizePixel);
```
- Proses Segmentasi

Segmentasi dilakukan untuk proses binerisasi pada image dengan menggunakan nilai threshold. Proses mendapatkan nilai threshold dilakukan secara otomatis dengan metode otsu. Proses metode otsu ini menggunakan bantuan dari aplikasi yang dibuat Tolga Birdal dimana telah menerapkan metode otsu pada bahasa pemrograman C#

 1. Memanggil fungsi `getOtsuThreshold`

```
ot.Convert2GrayScaleFast(temp);
```
 2. Mendapatkan nilai grayscale dari citra


```
Color imgColor = resizePixel.GetPixel(x, y);
int img = (int)((imgColor.R * 0.3) + (imgColor.G * 0.59)
+ (imgColor.B * 0.11));
```
 3. Proses binerisasi dimana bila nilai grayscale lebih besar dari nilai threshold maka akan dibuat menjadi 255 dan sebaliknya akan dibuat menjadi 0


```
if (img >= otsuThreshold)
{ img = 255; }
Else
{ img = 0; }
```
 4. Menerapkan hasil binerisasi dan memberikan nilai pixel baru pada citra


```
Color newColor = Color.FromArgb(img, img, img);
resizePixel.SetPixel(x, y, newColor);
```

- Proses Pelabelan Komponen

Pelabelan image pada aplikasi ini juga menggunakan bantuan dari Aforge framework. Cara kerja dari pelabelan komponen adalah dengan cara membaca tiap piksel pada citra dari atas ke bawah kiri ke kanan, setelah ditemukan piksel yang pertama dan bukan piksel *background*, dilakukan pengecekan tetangga apakah sudah ditandai atau belum, jika sudah ditandai maka piksel tersebut ditandai sesuai dengan piksel yang berdekatan, jika tidak maka piksel ini diberi tanda baru. Pada proses pelabelan juga akan digunakan operasi untuk menghitung jumlah objek yang ada pada citra.

1. Memberi label objek pada citra

```
ConnectedComponentsLabeling f = new  
ConnectedComponentsLabeling();
```

2. Menerapkan fungsi labeling pada citra

```
pictureBox3.Image = f.Apply(cc1);
```

3. Menghitung jumlah objek pada citra

```
hitung = f.ObjectCount;
```

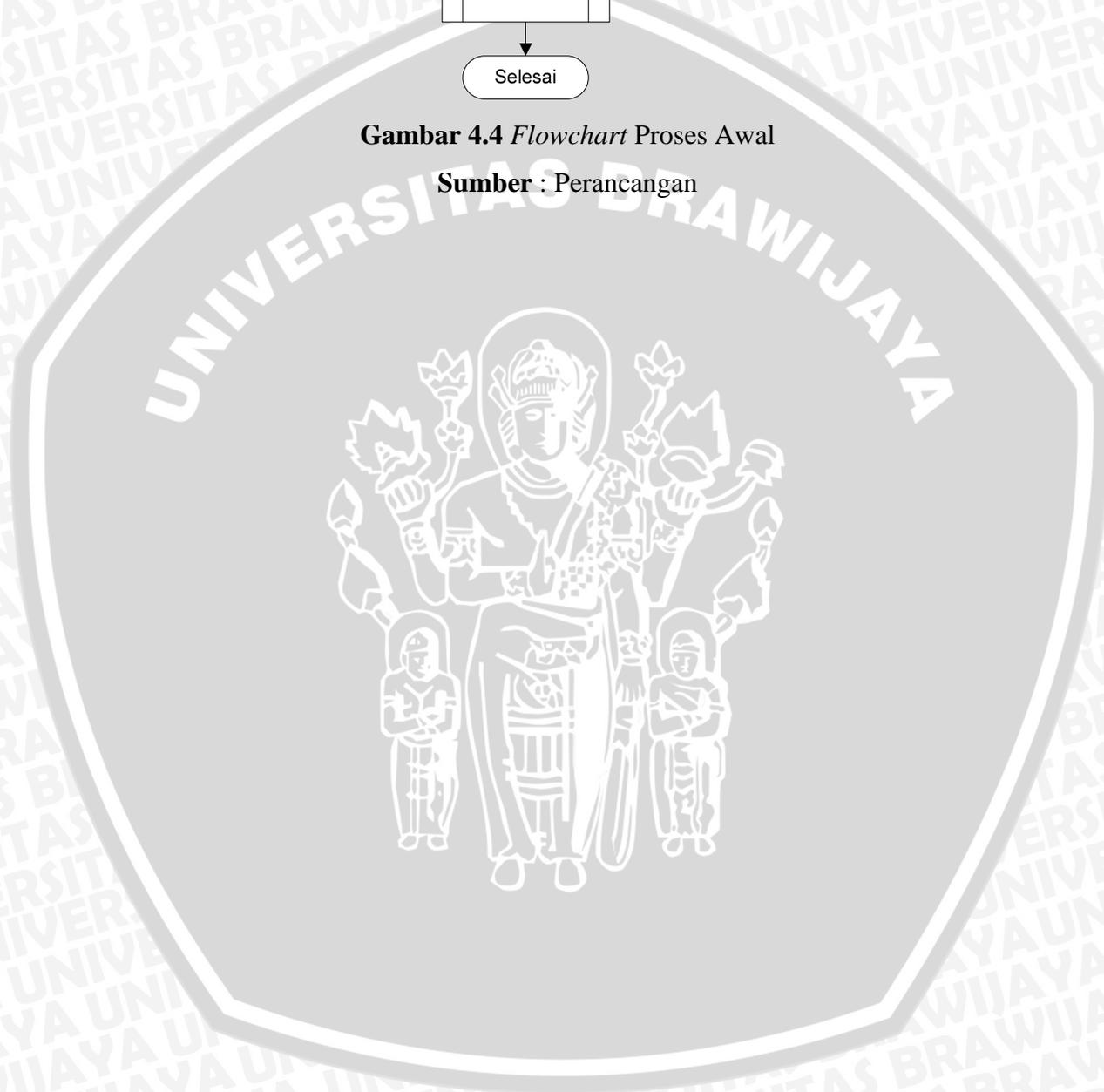
4.2.2 Proses Awal

Proses tahap awal ini dilakukan setelah output dari hasil *preprocess* telah ditandai. Proses pertama dalam tahap ini adalah *cropping*, proses ini dilakukan untuk mengambil tiap objek yang terdapat dalam citra. Setelah itu dilakukan pencarian deteksi tepi untuk tiap objek. Proses-proses pada tahap ini dilakukan untuk menyiapkan citra lebih sempurna untuk mempermudah proses analisa berikutnya. Flowchart perancangan proses awal seperti pada Gambar 4.4.

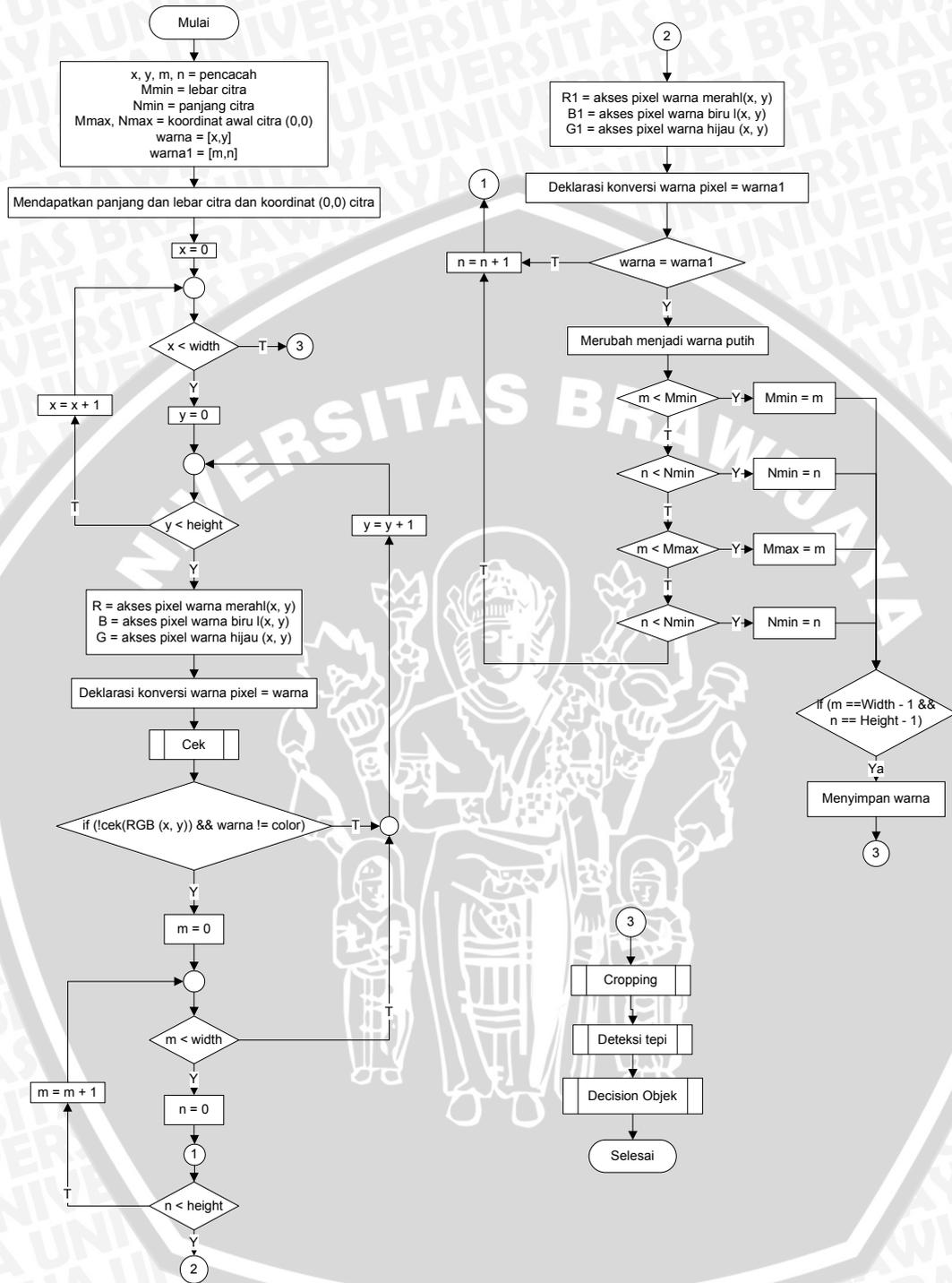


Gambar 4.4 Flowchart Proses Awal

Sumber : Perancangan



4.2.2.1 Sub Proses Pengambilan Objek



Gambar 4.5 Flowchart pengambilan objek

Sumber : Perancangan



Penjelasan *flowchart*:

1. Karena pada proses sebelumnya objek telah diberi label dengan warna sebagai pembedanya, maka proses pengambilan objek dimulai dengan menentukan warna awal objek yang akan diambil.
2. Ketika warna ditemukan, maka melakukan pengecekan apakah warna ini sudah tersimpan atau belum, jika belum maka proses iterasi akan berlanjut, jika warna yang ditemukan sudah tersimpan, maka proses ini akan berulang sampai menemukan warna baru.
3. Proses iterasi terus berlanjut, apabila dia menemukan warna yang sama, maka akan diubah menjadi *pixel* putih, jika tidak maka akan diubah menjadi *pixel* hitam.
4. Selanjutnya dilakukan proses pencarian batas-batas koordinat pada objek untuk proses *cropping*.
5. Proses ini akan terus berulang sampai tidak ditemukan lagi warna baru.

Implementasi proses pengambilan objek pada bahasa C# adalah sebagai berikut:

- Proses Pengambilan Objek

Setelah objek-objek diberi label, maka proses selanjutnya yang dilakukan adalah pengambilan objek atau *cropping* untuk mempermudah proses analisis berikutnya. Berikut penjelasannya serta listing programnya:

1. Menentukan koordinat (X,Y) pada objek dan selanjutnya mengkonversi nilai pixel yang didapat ke nilai integer.

```
for (int x = 0; x < objek.Width; x++)
{
    for (int y = 0; y < objek.Height; y++)
    {
        int R = (objek.GetPixel(x, y).R);
        int G = (objek.GetPixel(x, y).G);
        int B = (objek.GetPixel(x, y).B);

        warna = Color.FromArgb(R, G, B).ToArgb();
    }
}
```

- }
2. Melakukan pengecekan warna apakah warna yang didapat sudah tersimpan dalam array atau belum, jika sudah maka proses berulang sampai menemukan warna yang baru. Jika tidak maka proses berlanjut melakukan iterasi dimulai dari titik (0,0) untuk mencari objek yang memiliki warna yang sama maka akan diubah menjadi pixel putih, jika tidak maka akan lanjut ke proses selanjutnya. Proses terus berulang sampai tidak ditemukan lagi warna baru.

```
if(!cek(objek.GetPixel(x,y))&&warna!=Color.Black.ToArgb())
{
    for (int m = 0; m < objek.Width; m++)
    {
        for (int n = 0; n < objek.Height; n++)
        {
            int R1 = (objek.GetPixel(m, n).R);
            int G1 = (objek.GetPixel(m, n).G);
            int B1 = (objek.GetPixel(m, n).B);

            warna1 = Color.FromArgb(R1, G1, B1).ToArgb();
            if (R == R1 && G == G1 && B == B1)
            {
                objek.SetPixel(m, n, Color.White);
            }
        }
    }
}
```

- Proses Cropping

Cropping yang dilakukan ini merupakan lanjutan dari proses pengambilan objek karena setelah menentukan objek maka objek yang berwarna putih akan dipisahkan dari objek yang lain dengan cara *di-cropping*.

1. Mendefinisikan awal nilai Xmin dan Ymin dimana dua nilai ini menentukan koordinat citra sesuai dengan panjang dan lebar citra.

```
int Xmin = seleksiObjek.Width;
int Ymin = seleksiObjek.Height;
```

2. Definisi awal nilai Xmax,Ymax dimana dua nilai ini dimulai dari koordinat (0,0)

```
int Xmax = 0;
int Ymax = 0;
```

3. Kemudian dilakukan proses pengecekan nilai (x,y) terhadap Xmin, Ymin, Xmax, Ymax dan proses ini dilakukan masih dalam proses iterasi saat proses pengecekan nilai pixel RGB

```
if (x < Xmin) Xmin = x;
if (y < Ymin) Ymin = y;
if (x > Xmax) Xmax = x;
if (y > Ymax) Ymax = y;
```

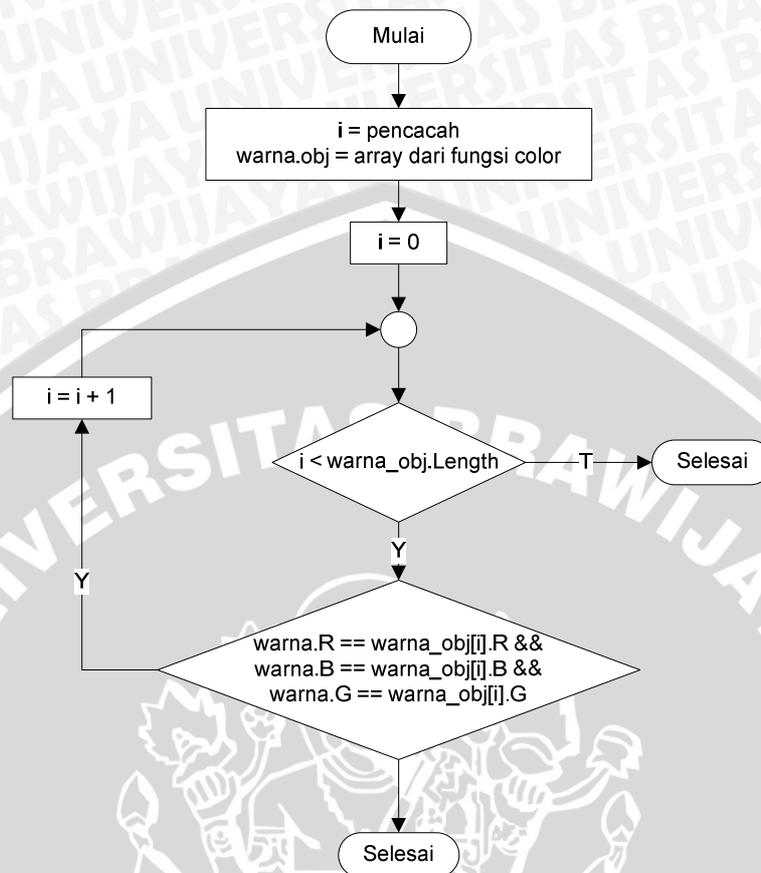
4. Selanjutnya menggunakan fungsi cropping dengan menggunakan parameter nilai-nilai Xmin, Ymin, Xmax, Ymax

```
Crop detectionObjek = new Crop(new Rectangle(Mmin, Nmin,
(Mmax - Mmin), (Nmax - Nmin)));
```

5. Menerapkan fungsi *cropping* pada pictureBox yang telah disediakan.

```
if (pictureBox8.Image == null)
{
    pictureBox8.Image = prewitt(crop_Img);
    decision(new Bitmap(pictureBox8.Image));
}
else if (pictureBox9.Image == null)
{
    pictureBox9.Image = prewitt(crop_Img);
    decision(new Bitmap(pictureBox9.Image));
}
else if (pictureBox10.Image == null)
{
    pictureBox10.Image = prewitt(crop_Img);
    decision(new Bitmap(pictureBox10.Image));
}
```

4.2.2.2 Perancangan Sub Pengecekan Warna



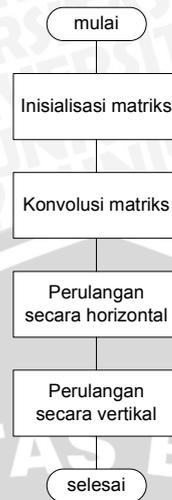
Gambar 4.6 Flowchart pengecekan warna

Sumber : Perancangan

Penjelasan *flowchart*:

- 1 Warna yang telah didapat dari proses iterasi pada pengambilan objek, dimasukkan ke dalam fungsi cek untuk melakukan pengecekan.
- 2 Warna yang telah didapat dicek apakah sudah tersimpan dalam array atau belum, jika sudah maka fungsi cek bernilai true, jika belum maka bernilai false.

4.2.2.3 Sub Proses Deteksi Tepi



Gambar 4.7 Flowchart deteksi tepi

Sumber : Perancangan

Penjelasan *flowchart*:

1. Proses Deteksi Tepi dimulai dengan inialisasi matriks berupa kernel prewitt yang akan digunakan untuk konvolusi. Kernel ini berukuran 3x3 yang terdiri atas 2 buah kernel, untuk horizontal (Gx) dan vertikal(Gy).
2. Selanjutnya dilakukan konvolusi matriks terhadap citra. Lalu konvolusi ini diterapkan dengan perulangan secara horizontal dan vertikal sampai keseluruhan citra.
3. Pertama dilakukan konvolusi kernel vertikal (Gy) dengan perulangan secara horizontal.
4. Kedua dilakukan konvolusi kernel horizontal (Gx) dengan perulangan secara vertikal.
5. Ketiga dilakukan konvolusi kernel vertikal (Gy) dengan perulangan secara vertikal. Dan terakhir dilakukan konvolusi kernel horizontal (Gx) dengan perulangan secara horizontal.

Implementasi proses deteksi tepi menggunakan bahasa C# dijelaskan sebagai berikut, deteksi tepi yang digunakan pada aplikasi ini menggunakan operator prewitt. Berikut penjelasan dan listing programnya:

1. Mendefinisikan matriks g_x dan g_y yang digunakan operator prewitt

```
int[,] gx = new int[,] { { -1, -1, -1 }, { 0, 0, 0 }, { 1, 1, 1 }  
};  
int[,] gy = new int[,] { { -1, 0, 1 }, { -1, 0, 1 }, { -1, 0, 1 }  
};
```

2. Deklarasi citra untuk deteksi tepi

```
Bitmap prewitt, prewitt1;  
prewitt = img;  
prewitt1 = new Bitmap(img.Width, img.Height);
```

3. Melakukan iterasi perulangan matriks g_x dan g_y secara horizontal dan vertikal. Perulangan dimulai dengan koordinat (1,1) sampai dengan batas ukuran panjang dan lebar citra. Pendefinisian nilai baru x dan y serta c sebagai dari konvolusi, karena operasi ini menggunakan konvolusi.

```
for (int x = 1; x < prewitt.Height - 1; x  
{  
    for (int y = 1; y < prewitt.Width - 1; y++)  
    {  
        float new_x = 0, new_y = 0;  
        float c;  
    }  
}
```

4. Proses konvolusi dilakukan dengan cara iterasi. Mula-mula menentukan nilai pixel yang akan di deteksi tepi. Setelah nilai RGB pada citra dengan koordinat (x,y) ditemukan maka dilakukan pencarian nilai rata-rata RGB. Untuk menemukan nilai x dan y yang baru dilakukan penjumlahan array pada matriks g_x setiap 1 piksel dan mengalikan dengan hasil dari rata-rata RGB pada citra, begitu juga dengan matriks g_y . Iterasi berdasarkan panjang dan lebar citra.

```
for (int hw = -1; hw < 2; hw++)  
{  
    for (int wi = -1; wi < 2; wi++)  
    {  
        c = (prewitt.GetPixel(y + wi, x + hw).B +  
            prewitt.GetPixel(y + wi, x + hw).R +  
            prewitt.GetPixel(y + wi, x + hw).G) / 3;  
        new_x += gx[hw + 1, wi + 1] * c;
```

```
        new_y += gy[hw + 1, wi + 1] * c;  
    }  
}
```

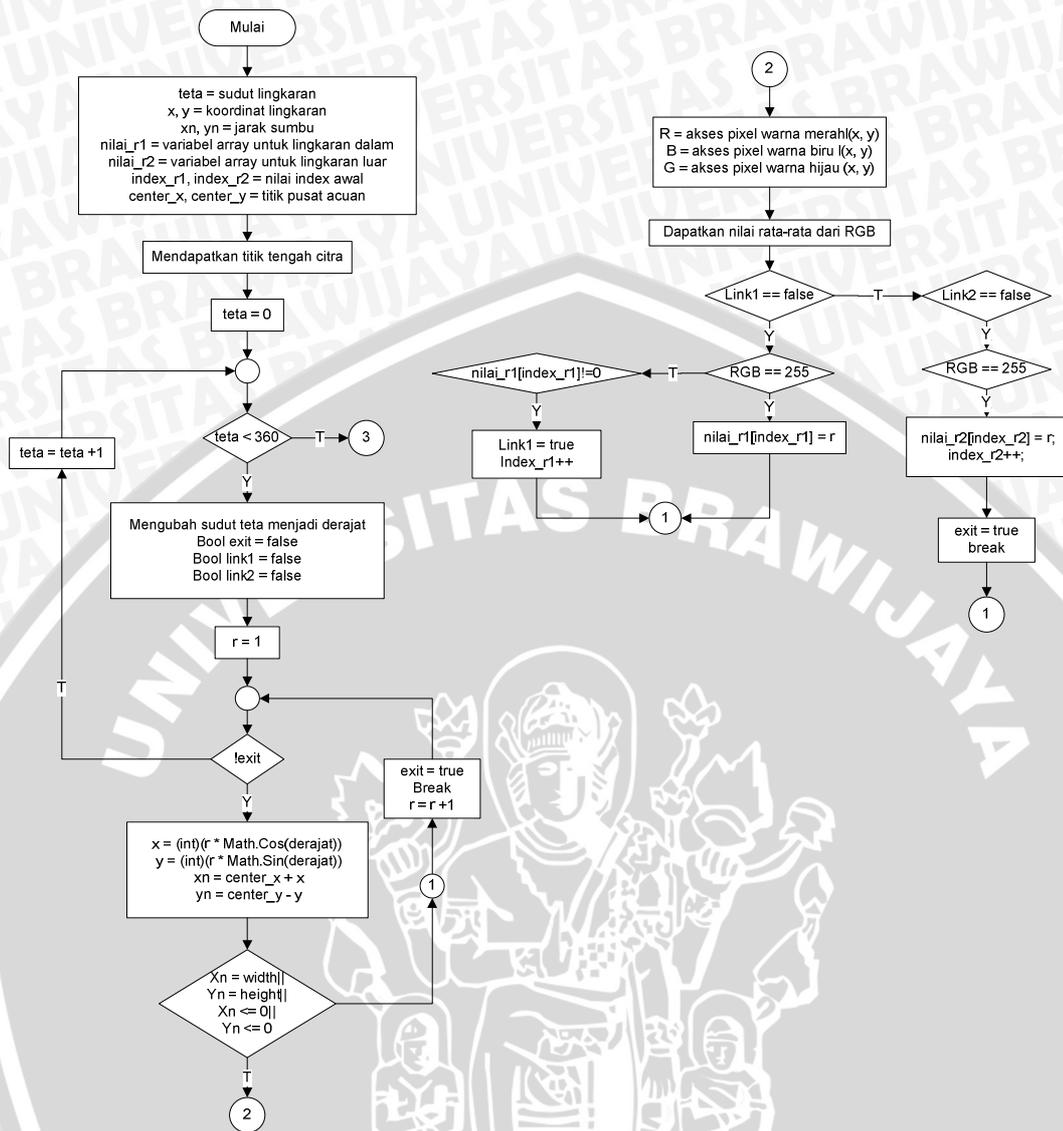
5. Pengecekan untuk membuat garis tepi pada objek

```
if (new_x * new_x + new_y * new_y > 128 * 128)  
    prewitt1.SetPixel(y, x, Color.White);  
else  
    prewitt1.SetPixel(y, x, Color.Black);
```

4.2.3 Proses Akhir

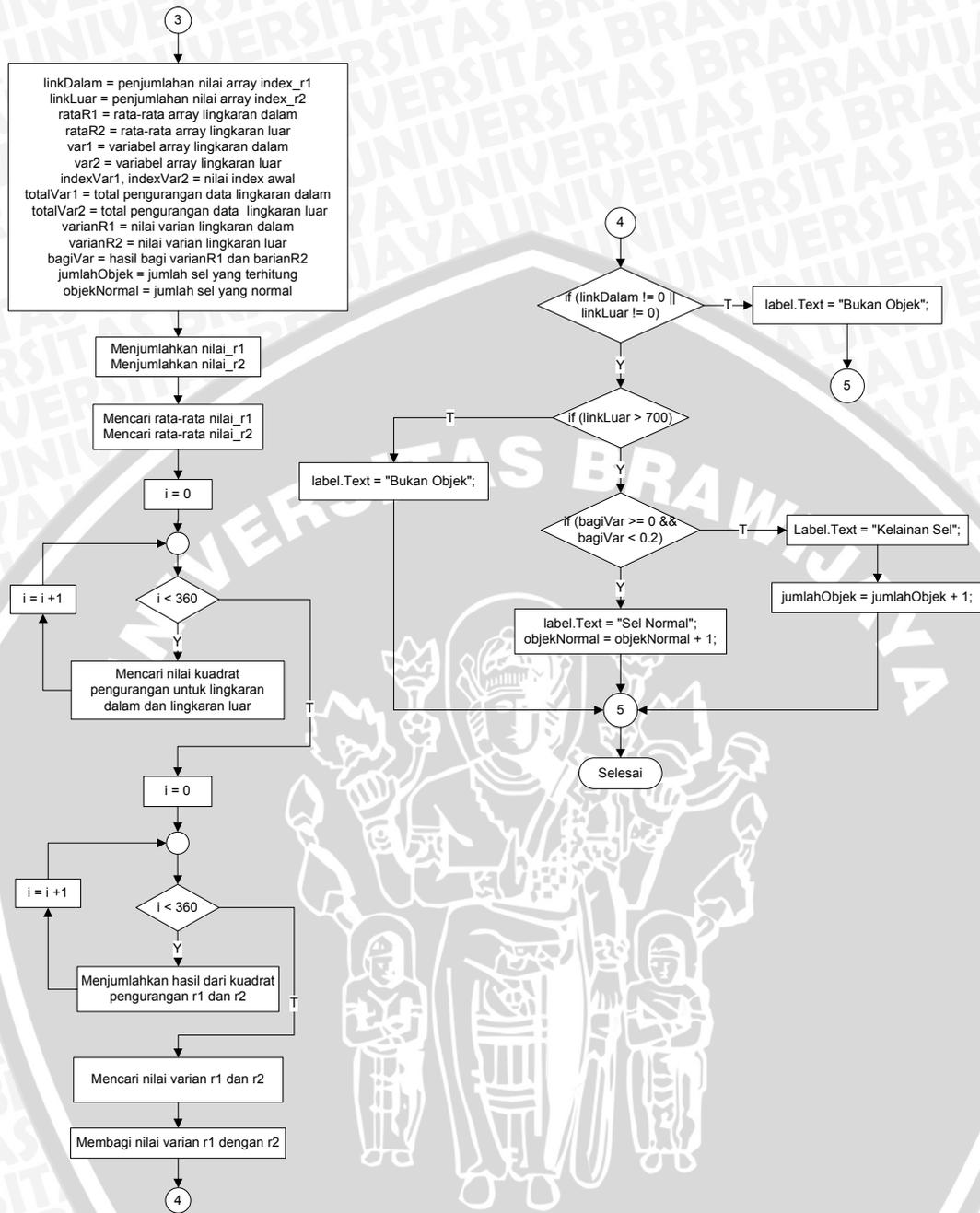
Tahap ini merupakan tahap akhir dalam aplikasi deteksi dini adanya kelainan pada *eritrosit*. Hasil dari objek yang telah diproses di tahap sebelumnya, selanjutnya dianalisis apakah terdapat kelainan pada sel darah merah atau tidak.





Gambar 4.8 Flowchart analisis objek
Sumber : Perancangan





Gambar 4.9 Flowchart analisis objek

Sumber : Perancangan

Penjelasan flowchart:

1. Proses pertama yang dilakukan adalah mencari titik tengah citra objek dengan cara membagi dua panjang dan lebar citra objek sehingga diperoleh titik tengah objek.

2. Selanjutnya lakukan proses iterasi untuk menghitung jarak dari titik pusat objek ke tepi objek dan ini dilakukan hingga mencapai 360 derajat. Langkah awal sebelum menghitung jarak adalah dengan mengubah terlebih dahulu teta menjadi derajat karena pada C# sudut awalnya dibaca sebagai radian.
3. Melakukan proses iterasi hingga menemukan pixel putih yang pertama, kemudian menghitung jarak dari pusat ke pixel putih pertama. Nilai dari pusat ke pixel putih pertama disimpan dalam `index_r1`.
4. Setelah menemukan pixel putih yang pertama, melakukan iterasi lagi hingga menemukan pixel putih yang kedua, menghitung jarak dari pusat ke pixel putih yang kedua kemudian menyimpan nilai jarak ke dalam `index_r2`.
5. Setelah nilai jarak disimpan dalam `index` selanjutnya mencari total jarak `r1` dan `r2`.
6. Proses selanjutnya adalah mencari nilai varian untuk masing-masing `r1` dan `r2` yang kemudian membandingkan nilai varian tersebut untuk menentukan apakah sel tersebut normal atau mengalami kelainan.
7. Apabila terdapat lebih dari 80% sel darah merah tersebut normal, maka dapat dikatakan bahwa sampel darah tersebut normal, jika sebaliknya maka terdapat kelainan pada sampel darah.

Implementasi analisis objek menggunakan bahasa C# dijelaskan sebagai berikut, untuk menentukan apakah sel darah merah normal atau tidak dapat dicari dengan cara mencari nilai jarak dari titik pusat objek ke tepi citra. Bentuk sel darah merah yang bikonkaf apabila dilihat secara dua dimensi dapat dikatakan seperti dua lingkaran yang mempunyai jari-jari yang berbeda. Dari bentuk tersebut dapat dicari jarak dari dua lingkaran, apabila jarak tidak memenuhi dalam *range* yang telah ditetapkan maka sel darah merah tersebut berpotensi adanya kelainan dan sebaliknya.

1. Citra input untuk proses analisis objek

```
Bitmap New;
```

```
New = new Bitmap(ImageDecision.Width, ImageDecision.Height);
```

2. Deklarasi variable

```
//variabel sudut teta
double teta = 0;
//koordinat (x,y) dan jarak sumbu (xn,xy)
int x, y, xn, yn;
//variabel array untuk r1 dan r2
int[] nilai_r1 = new int[360]; //lingkaran dalam
int[] nilai_r2 = new int[360]; //lingkaran luar
//nilai awal index
int index_r1 = 0;
int index_r2 = 0;
//menentukan titik pusat acuan
int center_x = ImageDecision.Width / 2;
int center_y = ImageDecision.Height / 2;
```

- ## 3. Proses pertama kali yang dilakukan untuk mencari jarak dari titik pusat ke tepi lingkaran pertama adalah mengubah satuan teta ke satuan derajat. Kemudian melakukan hitungan matematis untuk mencari koordinat lingkaran pertama ditandai dengan ditemukannya piksel putih yang pertama. Setelah nilai jarak titik pusat ke lingkaran pertama disimpan dalam array, maka proses masih berlanjut untuk menemukan jarak titik pusat ke lingkaran kedua dimana ditandai ditemukannya piksel putih pada garis tepi objek.

```
//perulangan mencari jarak tepi ke titik pusat
for (teta = 0; teta < 360; teta = teta + 1)
{
    double derajat = (Math.PI / 180) * teta;
    bool exit = false;
    bool link1 = false;
    bool link2 = false;

    for (int r = 1; !exit; r++)
    {
        //perhitungan menemukan koordinat x
        x = (int)(r * Math.Cos(derajat));
        //perhitungan menemukan koordinat y
        y = (int)(r * Math.Sin(derajat));
        //jarak koordinat sumbu x
        xn = center_x + x;
```

```
//jarak koordinat sumbu y
yn = center_y - y;
//kondisi bila menemukan piksel putih lebih dari ukuran citra
perulangan akan berhenti
if (xn >= ImageDecision.Width || yn >= ImageDecision.Height ||
xn <= 0 || yn <= 0)
{
    exit = true;
    break;
}
int R = (ImageDecision.GetPixel(xn, yn).R);
int G = (ImageDecision.GetPixel(xn, yn).G);
int B = (ImageDecision.GetPixel(xn, yn).B);

int RGB = ((int)R + (int)G + (int)B) / 3;

//menyimpan nilai jarak dari titik pusat ke lingkaran pertama
if (link1 == false)
{
    if (RGB == 255)
    {
        nilai_r1[index_r1] = r;
    }
    else if (nilai_r1[index_r1] != 0)
    {
        link1 = true;
        index_r1++;
    }
}
//menyimpan nilai jarak dari titik pusat ke lingkaran kedua
else if (link2 == false)
{
    if (RGB == 255)
    {
        nilai_r2[index_r2] = r;
        index_r2++;
        exit = true;
    }
}
}
```

- Setelah jarak dari titik pusat ke lingkaran pertama dan kedua sudah ditemukan, maka proses selanjutnya yang dilakukan adalah menjumlah nilai-nilai array yang tersimpan.

```
//menghitung jumlah array
//lingkaran pertama
linkDalam = nilai_r1.Sum();
//lingkaran kedua
linkLuar = nilai_r2.Sum();
```

- Setelah menjumlahkan nilai-nilai array, maka proses berikutnya adalah mencari nilai varian untuk masing-masing array yang kemudian membandingkan nilai varian lingkaran pertama dan lingkaran kedua untuk enentukan apakah sel darah merah tersebut normal atau mengalami kelainan.

```
//deklarasi untuk mencari nilai varian
double rataR1 = 0;
double[] var1 = new double[360];
double[] var2 = new double[360];
int indexVar1 = 0;
int indexVar2 = 0;
double totalVar1 = 0;
double totalVar2 = 0;

//Mencari rata-rata
rataR1 = linkDalam / 360;
rataR2 = linkLuar / 360;

//Mencari varian
for (int i = 0; i < 360; i++)
{
    var1[indexVar1] = Math.Pow(nilai_r1[i] - rataR1, 2);
    indexVar1++;
    var2[indexVar2] = Math.Pow(nilai_r2[i] - rataR2, 2);
    indexVar2++;
}
for (int i = 0; i < 360; i++)
{
    totalVar1 += var1[i];
    totalVar2 += var2[i];
}
```

```
}  
varianR1 = Math.Ceiling(totalVar1 / (360 - 1));  
varianR2 = Math.Ceiling(totalVar2 / (360 - 1));
```

```
bagiVar = varianR1 / varianR2;
```

6. Hasil akhir dari proses ini adalah aplikasi dapat menentukan apakah sel darah merah atau tidak dilihat dari hasil perbandingan nilai varian lingkaran pertama dan kedua. Apabila hasil perbandingan bernilai antara 0 – 0,2 maka sel darah merah tersebut normal, jika diluar itu maka terdapat kelainan pada sel darah merah.

```
if (linkDalam != 0 || linkLuar != 0)  
{  
    if (linkLuar > 700)  
    {  
        if (bagiVar >= 0.01 && bagiVar < 0.2)  
        {  
            label2.Text = "Sel Normal";  
            objekNormal = objekNormal + 1;  
        }  
        Else  
        {  
            label2.Text = "Kelainan Sel";  
        }  
        jumlahObjek = jumlahObjek + 1;  
    }  
    Else  
    {  
        label2.Text = "Bukan Objek";  
    }  
}  
Else  
{  
    label2.Text = "Bukan Objek";  
}
```

```
range = (float)(objekNormal / jumlahObjek);  
range = range * 100;  
if (range >= 80)  
{
```

```
listView1.Items.Clear();
listView1.Items.Add(new ListViewItem(new string[] {
    "Jumlah Sel darah merah: ", jumlahObjek.ToString() }));
listView1.Items.Add(new ListViewItem(new string[] {
    "Jumlah Sel Normal: ", objekNormal.ToString() }));
listView1.Items.Add(new ListViewItem(new string[] {
    "Sel darah Merah Normal" }));
}
else
{
    listView1.Items.Clear();
    listView1.Items.Add(new ListViewItem(new string[] {
        "Jumlah Sel darah merah: ", jumlahObjek.ToString() }));
    listView1.Items.Add(new ListViewItem(new string[] {
        "Jumlah Sel Normal: ", objekNormal.ToString() }));
    listView1.Items.Add(new ListViewItem(new string[] {
        "Terdapat kelainan pada Sel Darah Merah" }));
}
}
```

4.3 Implementasi Sistem

Setelah tahap perancangan dibuat maka untuk selanjutnya adalah tahap implementasi. Pada tahap implementasi ini merupakan proses transformasi dari proses perancangan yang telah dibuat sebelumnya dimana hasil perancangan ditransformasikan ke dalam bentuk bahasa pemrograman (*coding*) yang sesuai dengan sintaks yang ada pada bahasa pemrograman C# NET.

4.3.1 Lingkungan Implementasi

Aplikasi dibuat dengan menggunakan Microsoft Visual Studio C# dengan Aforge.NET framework. Sistem diimplementasikan dengan menggunakan spesifikasi sebagai berikut:

1. Perangkat keras (Komputer)

Spesifikasi :

Processor : Intel Core i7-740QM, 1.73GHz

Memory : 2 GB DDR3

OS : Windows 7

VGA : ATI Mobility Radeon HD 5470

2. Perangkat Lunak
 - Sistem Operasi : Windows 7 Ultimate
 - Bahasa Pemrograman : Microsoft visual studio .NET C#
 - Framework : Aforge .NET Framework

4.2 Implementasi *Interface* (Antarmuka)

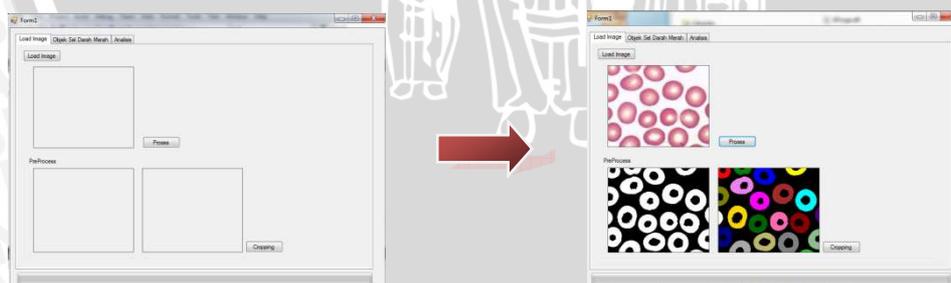
Program deteksi dini kelainan pada *eritrosit* ini mempunyai 3 tampilan utama, berikut tampilan:

1. *Load Image*
2. Objek Sel Darah Merah
3. Analisis

Bentuk antarmuka program ini didesain sesederhana mungkin dan berurutan. Ini bertujuan agar pemakai tidak bingung dalam menggunakan program ini

4.2.1 Implementasi Antarmuka *Load Image*

Tampilan antarmuka dari *load image* dapat dilihat pada gambar dibawah ini. Terdapat 3 pictureBox. PictureBox yang berada pada bagian atas tampilan berisi citra yang akan diproses. Sedang 2 pictureBox yang berada dibawah merupakan hasil citra yang telah melewati pre proses.

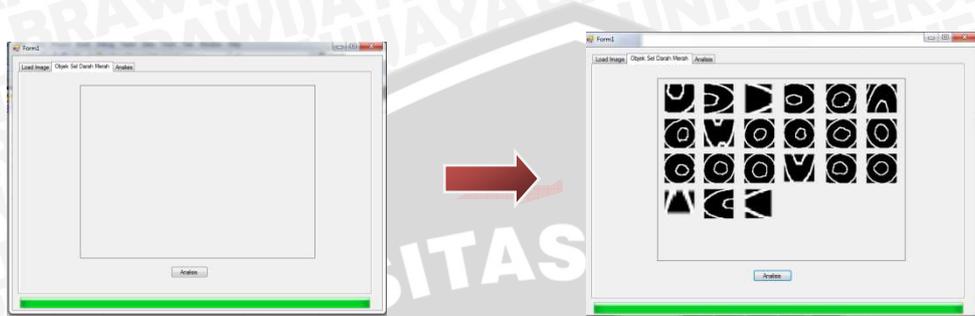


Gambar 4.10 Antarmuka *Load Image*

Sumber : Implementasi

4.2.2 Implementasi Antarmuka Objek Sel Darah Merah

Tampilan antarmuka dari objek sel darah merah dapat dilihat pada gambar dibawah ini. Tampilan ini berisi objek-objek sel darah merah yang telah di *cropping*.

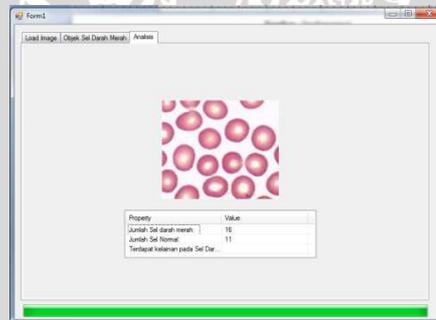


Gambar 4.11 Antarmuka Objek Sel darah Merah

Sumber : Implementasi

4.2.3 Implementasi Antarmuka Analisis

Tampilan antarmuka dari objek sel darah merah dapat dilihat pada gambar dibawah ini. Tampilan ini berisi hasil dari analisis citra sel darah merah.



Gambar 4.12 Antarmuka Analisis

Sumber : Implementasi

BAB V PENGUJIAN

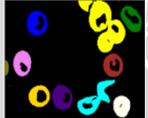
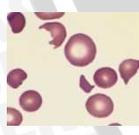
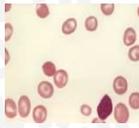
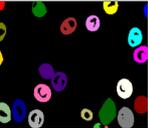
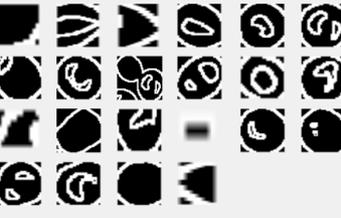
Pengujian dan analisis dilakukan untuk mengetahui apakah sistem telah bekerja sesuai perancangan, maka diperlukan serangkaian pengujian, yaitu pengambilan objek dan identifikasi objek. Pengujian ini dilakukan pada jumlah data sebanyak 20 data sampel.

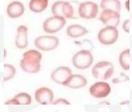
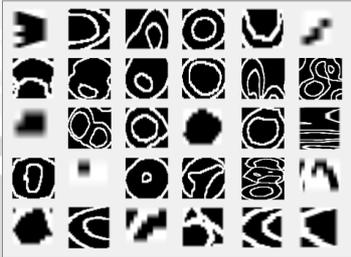
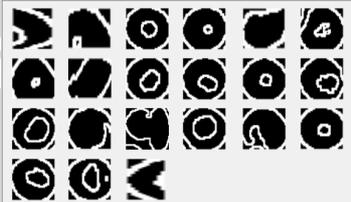
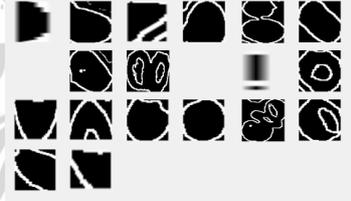
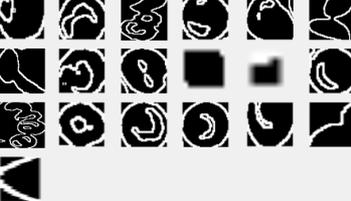
5.1 Pengujian Pengambilan Objek

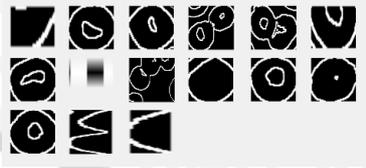
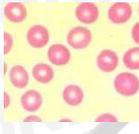
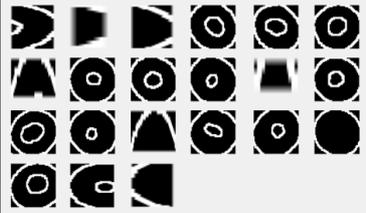
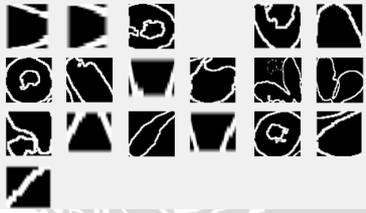
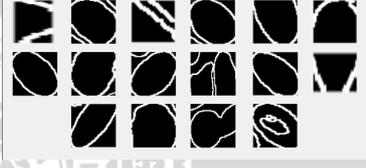
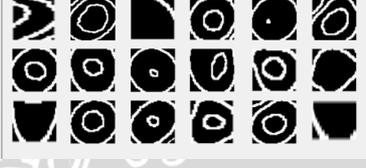
Pengujian ini bertujuan untuk mengetahui apakah program tersebut berkerja sesuai dengan fungsionalitasnya. Dalam hal ini apakah program dapat mendeteksi beberapa objek yang berada dalam satu citra dan juga telah memisahkan objek dengan latar belakangnya yang kemudian tiap objek telah diberi label yang berbeda-beda sehingga hasil akhir nanti tiap objek dapat diambil satu persatu. Berikut hasil pengujian dapat dilihat dari tabel 5.1

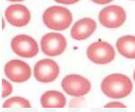
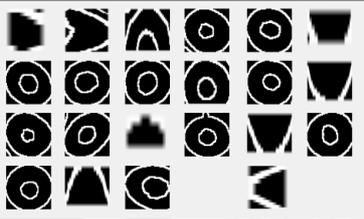
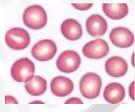
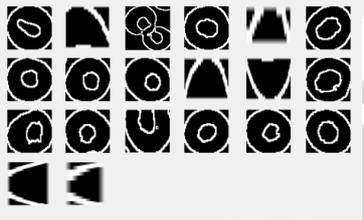
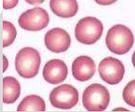
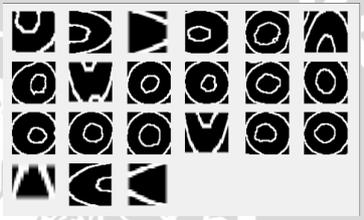
Tabel 5.1 Tabel Pengujian Pengambilan Objek

Sumber : Pengujian

Citra Uji	Objek Terdeteksi	TM	Objek yang Terambil	TS
		13		10
		12		12
		23		22

Citra Uji	Objek Terdeteksi	TM	Objek yang Terambil	TS
		29		29
		21		22
		18		20
		18		19
		19		21

Citra Uji	Objek Terdeteksi	TM	Objek yang Terambil	TS
		17		15
		21		21
		19		19
		17		17
		18		18
		15		16
		18		20

Citra Uji	Objek Terdeteksi	TM	Objek yang Terambil	TS
		21		23
		21		20
		21		21
		13		14
		10		11

Keterangan:

TM : Total dihitung secara mata

TS : Total dihitung secara program

5.2 Pengujian Identifikasi Objek

Pengujian identifikasi objek ini dilakukan dengan tujuan keberhasilan aplikasi untuk menentukan apakah objek termasuk sampel darah normal atau tidak. Berikut hasil pengujian dapat dilihat dari tabel 5.2

Tabel 5.2 Tabel Pengujian Identifikasi Objek

Sumber : Pengujian

Citra Uji	Sel Normal		Sel Rusak		Total	
	Ph	Pj	Ph	Pj	Ph	Pj
Sampel 1	6	1	4	6	10	9
Sampel 2	3	3	7	6	10	10
Sampel 3	1	3	16	17	17	17
Sampel 4	2	9	20	12	22	21
Sampel 5	13	12	7	5	20	16
Sampel 6	1	1	16	11	17	12
Sampel 7	1		20	10	21	16
Sampel 8	15	13	0	4	15	15
Sampel 9	11	5	4	9	15	12
Sampel 10	13	12	1	2	14	14
Sampel 11	4	6	9	7	13	12
Sampel 12	1	2	14	10	15	12
Sampel 13	4	12	14	9	18	15
Sampel 14	2	7	14	9	16	15
Sampel 15	0	6	19	11	19	15
Sampel 16	14	12	0	4	14	15
Sampel 17	14	12	1	2	15	14
Sampel 18	15	12	0	8	15	20
Sampel 19	9	8	2	2	11	11
Sampel 20	7	9	2	4	9	10

Keterangan:

Ph : Perhitungan oleh dokter ahli

Pj : Pengujian oleh software

Sampel dikatakan normal atau tidak dapat dilihat dari hasil pengujian oleh software. Sampel citra darah dapat dikatakan normal apabila terdapat $\geq 80\%$ sel darah merah yang sehat. Hasil analisis identifikasi objek dapat dilihat pada tabel 5.3

Tabel 5.3 Tabel Hasil Analisis Pengujian Identifikasi Objek

Sumber : Pengujian

Citra Uji	Teori	Hasil Deteksi
Sampel 1	Kelainan Sel	Kelainan Sel
Sampel 2	Kelainan Sel	Kelainan Sel

Citra Uji	Teori	Hasil Deteksi
Sampel 3	Kelainan Sel	Kelainan Sel
Sampel 4	Kelainan Sel	Kelainan Sel
Sampel 5	Kelainan Sel	Kelainan Sel
Sampel 6	Kelainan Sel	Kelainan Sel
Sampel 7	Kelainan Sel	Kelainan Sel
Sampel 8	Sel Normal	Sel Normal
Sampel 9	Kelainan Sel	Kelainan Sel
Sampel 10	Sel Normal	Sel Normal
Sampel 11	Kelainan Sel	Kelainan Sel
Sampel 12	Kelainan Sel	Kelainan Sel
Sampel 13	Kelainan Sel	Kelainan Sel
Sampel 14	Kelainan Sel	Kelainan Sel
Sampel 15	Kelainan Sel	Kelainan Sel
Sampel 16	Sel Normal	Sel Normal
Sampel 17	Sel Normal	Sel Normal
Sampel 18	Sel Normal	Sel Normal
Sampel 19	Sel Normal	Sel Normal
Sampel 20	Sel Normal	Sel Normal

Dari tabel diatas dapat dilihat bahwa aplikasi dapat mendeteksi objek sangat baik.

5.3 Analisis Faktor Kegagalan

Pada kenyataan aplikasi deteksi dini kelainan pada *eritrosit* ini tidak selalu berjalan optimal. Terdapat beberapa faktor yang dapat menyebabkan proses yang dilakukan menjadi gagal. Berikut adalah beberapa faktor yang menyebabkan kegagalan tersebut:

1. Adanya objek yang tidak terambil dengan sempurna, dikarenakan adanya objek yang memiliki koordinat yang sama.



2. Terdapat perbedaan hasil pembacaan sel darah secara teori dan program, hal ini dikarenakan beberapa faktor antara lain:
 - a. Adanya objek yang tumpuk, pada aplikasi ini tidak dapat memisahkan objek yang bertumpuk. Hal ini dikarenakan pada aplikasi ini tidak diterapkan algoritma untuk memisahkan objek.
 - b. Pada aplikasi ini tidak diterapkan algoritma untuk mengenali bentuk, karena pada aplikasi ini tidak menentukan bentuk dari sel darah normal, penentuan normal tidaknya sel hanya berdasarkan jarak dari inti sel ke tepi sel.



BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis sistem maka dapat diambil kesimpulan sebagai berikut:

1. Proses dalam aplikasi deteksi awal kelainan pada *eritrosit* terbagi menjadi 3 bagian yaitu preprocess, proses tahap awal dan proses tahap akhir
2. Metode yang digunakan untuk mendeteksi dan memisahkan citra *eritrosit* yang ada dalam citra darah adalah *Connected Component Labeling*, yaitu memberi label yang berbeda pada setiap objek yang kemudian diambil satu persatu menggunakan proses *cropping*.
3. Dari 20 citra darah yang diuji, terdapat 15 sampel yang citra objek dapat terambil dengan benar. Hal ini dikarenakan beberapa faktor seperti adanya objek yang bertumpuk dan noise, sehingga aplikasi mengenali sebagai satu objek.
4. Proses pengidentifikasian objek dilakukan dengan membandingkan nilai varian lingkaran dalam dan lingkaran luar, apabila nilai varian kurang dari 0,2 maka sel tersebut normal jika lebih maka sel tersebut mengalami kelainan.
5. Terjadi perbedaan antara penghitungan secara langsung suatu citra sel darah dibandingkan dengan perhitungan menggunakan program baik yang normal maupun yang mengalami kelainan. Berdasarkan hasil dari pengujian yang telah dilakukan kelainan sel darah merah berupa Sickle dan Stomatocyte memiliki nilai varian diatas 0,2.

6.2 Saran

Terdapat beberapa hal yang perlu dibenahi dalam program yang dibuat oleh penulis ini. Untuk mempermudah dalam mempelajari dan berguna sebagai bahan pertimbangan bagi mereka yang berminat pada perangkat lunak ini, penulis memberikan saran-saran umum sebagai berikut:



1. Jumlah penggunaan sampel yang lebih banyak sehingga hasil pengujian sistem secara keseluruhan dapat lebih akurat
2. Aplikasi belum memisahkan objek yang bertumpuk, untuk itu selanjutnya agar dapat digunakan metode lain untuk dapat memisahkan objek yang bertumpuk.
3. Penggunaan metode lain untuk dapat dibandingkan sehingga diperoleh metode gabungan yang lebih baik dalam hal untuk mendeteksi objek.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- [1] Ahmad ,Usman, ”Pengolahan Citra Digital dan Teknik Pemrogramannya”, Graha Ilmu, Yogyakarta, 2005.
- [2] Barbara Young, James. S. Lowe, “Wheater’s Functional Histology”, Churchill: Livinstone, 2006.
- [3] Courtesy of Albany. N. Y, Dr. Lawrence S. Stunmant M. D., Ph. D, New York State Department of Health.
- [4] Courtesy of Dr. Robert W. McKenna, Department of Pathology, University of Texas Southwestern Medical School, Dallas, TX
- [5] Handayani Wiwik, dan Haribowo Andi Sulisty, *Asuhan Keperawatan pada Klien dengan Gangguan Sistem Hematologi*, Indonesia, 2008.
- [6] Gustiawan Ferdiansyah, “Aplikasi Pendeteksi Objek Berpotensi Tumor Paru Pada Cita Hasil *Thorax* CT Scan Dengan Memanfaatkan Pengolahan Citra Digital”, Malang, 2011.
- [7] Hoffbrand A.V, dkk. *Kapita Selekte Hematologi*, ECG, Jakarta, 2005.
- [8] Kraft Kristine M. D, “The Complete (But Niot Obsessive) Hematopathology Guide”, Departemen of Pathology University of Minneosta of Medicine, 2011.
- [9] Muntasa, Arif, dan Purnomo, Mauridhi Hery.,”Konsep Pengolahan Citra Digital dan Ekstraksi Fitur”, Graha Ilmu, Yogyakarta, 2010.
- [10] Otsu N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979.
- [11] Putra, Darma,”Pengolahan Citra Digital”, Andi, Indonesia, 2010.
- [12] Wimba, Daud. Hematologi. <http://dikakigunungitu.blogspot.com/2009/03/hematologi-3-diagnosa-morfologi.html> (diakses tanggal 28 Juli 20011)
- [13] Riley, Roger S.,dkk. “How to Prepare & Interpret Peripheral Blood Smears”. <http://www.pathology.vcu.edu/education/PathLab/pages/hematopath/pbs.html> (diakses tanggal 28 Juli 2011)
- [14] Sutoyo,T,dkk, ”Teori Pengolahan Citra Digital”, Andi, Indonesia, 2009.
- [15] Basic techniques of Hematology (Basic Hematological Techniques): Department of Health microscope study clinic, Faculty of Medicine. University; 2548: 59-64.