



**RANCANG BANGUN PENDETEKSI GOLONGAN DARAH  
MENGUNAKAN METODE COMPONENT LABELLING**



**Oleh :**

**MAS ULUN NAJA**

**13.41020.0033**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**2017**

## ABSTRAK

Selama ini proses pemeriksaan golongan darah dilakukan dengan metode ABO yaitu dengan cara meneteskan 3 antisera pada 3 sampel darah yang sama diberi antisera A,B, dan AB. Setelah diaduk maka sampel tersebut akan mengalami aglutinasi yaitu proses penggumpalan darah akibat adanya aglutinin yang sifatnya menggumpalkan aglutinogen. Setelah terjadi aglutinasi maka harus dianalisis dengan standar tertentu sehingga golongan darah dapat diketahui. Proses ini masih dilakukan dengan pertolongan laboratorium yang memerlukan ketelitian dan terkadang terjadi kesalahan baca karena keterbatasan mata manusia, maka diperlukan suatu proses otomatisasi yang dapat membantu kerja manusia untuk melakukan analisis golongan darah. Pada penelitian ini metode yang digunakan untuk mendukung metode ABO adalah metode *component labelling*.

*Component labelling* dipilih dikarenakan metode ini dapat memberikan inisial atau label dan memberikan informasi luas piksel pada suatu obyek citra yang dideteksi, sehingga metode ini sesuai untuk menganalisis aglutinasi pada sampel darah. *Component labelling* akan memberikan label yang sama pada sekumpulan piksel yang membentuk obyek yang saling berdekatan pada suatu citra sehingga aglutinasi pada sampel darah dapat dianalisis dengan baik. Setelah aglutinasi dianalisis menggunakan *component labelling*, luas piksel pada sampel darah dan letak koordinat akan dihitung dan dibandingkan sesuai dengan teori metode ABO sehingga golongan darah dapat diketahui.

Sampel darah didapatkan dengan acak sebanyak 24 sampel darah lalu diuji dengan program yang telah dibuat. Hasilnya 100% sampel dapat dianalisis dengan benar dengan ketentuan pengambilan sampel di atas preparat harus benar, ambang batas *threshold* 80, kualifikasi obyek besar dengan luas diatas 100 piksel, warna dasar obyek adalah putih sehingga tidak ada gangguan pada citra digital, sehingga hasil yang didapat akan sesuai harapan.

**Kata Kunci :** *Image Processing*, OpenCv, Component Labelling, Pendeteksi

Golongan Darah.

## DAFTAR ISI

### Halaman

HALAMAN JUDUL.....	i
HALAMAN SYARAT.....	ii
MOTTO.....	iii
HALAMAN PERSEMBAHAN.....	iv
HALAMAN PENGESAHAN .....	v
HALAMAN PERNYATAAN.....	vi
ABSTRAK .....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI .....	x
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Sistematika Penulisan .....	3
BAB II LANDASAN TEORI .....	5
2.1 Darah Manusia.....	5
2.2 Teori Pengujian Gololan Darah.....	5

2.3	<i>Computer Vision</i> .....	7
2.4	<i>Open Computer Vision (OpenCV)</i> .....	8
2.5	<i>Grayscale</i> .....	10
2.6	<i>Threshold</i> .....	11
2.7	<i>Metode Component Labelling</i> .....	12
2.7.1	<i>Spoke Filter</i> .....	13
2.7.2	<i>Automatic Scale Selection</i> .....	14
2.8	<i>Kamera</i> .....	14
BAB III METODE PENELITIAN .....		14
3.1	<i>Metode Penelitian</i> .....	15
3.2	<i>Prosedur Penelitian</i> .....	16
3.2.1	<i>Program Deteksi Obyek Aglutinasi dengan Metode Component Labelling</i> .....	17
3.2.2	<i>Program Deteksi Posisi Koordinat X</i> .....	18
3.2.3	<i>Program Mengurutkan Ukuran Obyek Aglutinasi Dari Besar Ke Kecil Menggunakan Metode Bubble Sort</i> .....	19
3.2.4	<i>Program Mendapatkan Hasil Golongan Darah</i> .....	20
3.3	<i>Flowchart Program</i> .....	21
BAB IV HASIL DAN PEMBAHASAN.....		25
4.1	<i>Kebutuhan Sistem</i> .....	25
4.2	<i>Pengujian Nilai Threshold</i> .....	25
4.2.1	<i>Tujuan Pengujian Nilai Threshold</i> .....	26
4.2.2	<i>Alat Yang Digunakan Pada Pengujian Nilai Threshold</i> .....	26
4.2.3	<i>Prosedur Pengujian Nilai Threshold</i> .....	26

4.2.4	Hasil Pengujian Nilai Threshold.....	27
4.3	Pengujian Program Deteksi Aglutinasi dengan Metode <i>Component Labelling</i> .....	28
4.3.1	Tujuan Pengujian Program Deteksi Aglutinasi dengan Metode <i>Component Labelling</i> .....	28
4.3.2	Alat yang Digunakan pada Pengujian Program Deteksi Aglutinasi dengan Metode <i>Component Labelling</i> .....	28
4.3.3	Prosedur Pengujian Program Deteksi Aglutinasi dengan Metode <i>Component Labelling</i> .....	29
4.3.4	Hasil Pengujian Program Deteksi Aglutinasi dengan Metode <i>Component Labelling</i> .....	29
4.4	Pengujian Program Deteksi Posisi Koordinat X .....	32
4.4.1	Tujuan Pengujian Program Deteksi Posisi Koordinat X .....	33
4.4.2	Alat yang Digunakan pada Pengujian Program Deteksi Posisi Koordinat X .....	33
4.4.3	Prosedur Pengujian Program Deteksi Posisi Koordinat X .....	33
4.4.4	Hasil Pengujian Program Posisi Koordinat X .....	34
4.5	Pengujian Program Mengurutkan Ukuran Obyek Aglutinasi dari Besar ke Kecil .....	35
4.5.1	Tujuan Pengujian Program Mengurutkan Ukuran Obyek Aglutinasi dari Besar ke Kecil .....	35
4.5.2	Alat yang Digunakan pada Pengujian Program Mengurutkan Ukuran Obyek Aglutinasi dari Besar ke Kecil .....	35
4.5.3	Prosedur Pengujian Program Mengurutkan Ukuran Obyek Aglutinasi dari Besar ke Kecil .....	36

4.5.4	Hasil Pengujian Program Mengurutkan Ukuran Obyek Aglutinasi dari Besar ke Kecil .....	36
4.6	Pengujian Program Mendapatkan Hasil Golongan Darah .....	38
4.6.1	Tujuan Pengujian Program Mendapatkan Hasil Golongan Darah .....	38
4.6.2	Alat yang Digunakan pada Pengujian Program Mendapatkan Hasil Golongan Darah .....	38
4.6.3	Prosedur Pengujian Program Mendapatkan Hasil Golongan Darah .....	39
4.6.4	Hasil Pengujian Program Mendapatkan Hasil Golongan Darah .....	39
BAB V PENUTUP .....		50
5.1	Kesimpulan .....	50
5.2	Saran .....	50
DAFTAR PUSTAKA .....		52
BIODATA PENULIS .....		53
LAMPIRAN .....		54

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Transfusi darah hanya bisa dilakukan bila golongan darah antara penerima dan pendonor sejenis. Pada dunia kedokteran, golongan darah manusia dibagi 4, yaitu : A, B, AB dan O. Dengan demikian dalam transfusi darah juga dilakukan pengujian untuk menentukan golongan darah manusia (Andiani & Akmaliah, 2011).

Selama ini, untuk pemeriksaan golongan darah sering digunakan metode ABO. Pada metode ini sampel darah diletakkan pada 3 preparat. Preparat pertama diberi cairan antisera A dan preparat kedua diberi cairan antisera B, dan preparat ketiga diberi antisera AB. Setelah beberapa lama terjadi aglutinasi pada sampel tersebut. Aglutinasi adalah proses penggumpalan darah yang terjadi karena di dalam antisera terdapat aglutinin yang sifatnya menggumpalkan aglutinogen. Bentuk aglutinasi ini dibandingkan dengan standar yang ada untuk ditentukan golongan darahnya. Proses ini masih dilakukan dengan pertolongan laboratorium yang memerlukan ketelitian. Agar pemeriksaan dan dokumentasi secara digital dapat lebih mudah dilakukan.

Proses otomatisasi yang dimaksud adalah proses untuk menerapkan metode ABO kedalam *image processing*. Sehingga *image processing* akan menggantikan pengamatan manusia dalam melihat aglutinasi yang terjadi pada sampel darah. *Image Processing* adalah prosedur mengubah gambar ke dalam bentuk digital dan

melaksanakan beberapa operasi di dalamnya, untuk mendapatkan gambar yang baik dan mengambil beberapa informasi yang bermanfaat. Matematis pengolahan citra didefinisikan sebagai pengolahan gambar dua dimensi oleh komputer, gambar didefinisikan sebagai fungsi dua variabel seperti sumbu (x, y) seperti pada matrik (Basavaprasad & Ravi, 2014).

Karena *image processing* memiliki banyak metode yang dapat diterapkan, maka di dalam proyek ini haruslah mengambil metode yang paling tepat di bandingkan dengan metode yang lainnya. Diproyek ini penulis menggunakan metode *component labelling* yang dirasa paling tepat karena fungsinya untuk mendeteksi suatu obyek *blob* atau gumpalan pada suatu citra biner.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang diatas, dapat dirumuskan permasalahannya adalah “Bagaimana menerapkan metode *Component Labelling* untuk mendeteksi golongan darah”.

## 1.3 Batasan Masalah

Dalam perancangan ini, terdapat beberapa batasan masalah, antara lain :

1. Menggunakan *software* open cv
2. Menggunakan metode *Component Labeling*
3. Menggunakan *spokefilter* 4 tetangga
4. Tidak memperhitungkan rhesus pada golongan darah
5. dan Menggunakan bahasa pemrograman C atau C++



## 1.4 Tujuan

Tujuan dari penelitian ini adalah

1. Merancang pembacaan otomatis golongan darah menggunakan metode *component labelling*.
2. dan Dokumentasi secara digital.

## 1.5 Sistematika Penulisan

### BAB I PENDAHULUAN

Pada Bab ini akan dijelaskan tentang latar belakang , maksud penelitian, rumusan masalah, batasan masalah, dan tujuan dari penelitian ini.

### BAB II LANDASAN TEORI

Pada tahap ini akan dibahas teori secara singkat tentang teori-teori yang akan menjadi pembahasan utama pada penelitian ini seperti teori darah dan metode.

### BAB III METODE PENELITIAN

Pada bab ini akan dibahas tentang tahapan dalam melakukan pendeteksi golongan darah dengan metode *component labelling* yang telah ditentukan. Tahapan yang dilalui antara lain adalah :

1. Metode Penelitian
2. Prosedur Penelitian
3. Pendeteksian Obyek Menggunakan Metode *Component Labelling*
4. dan *Flowchart*

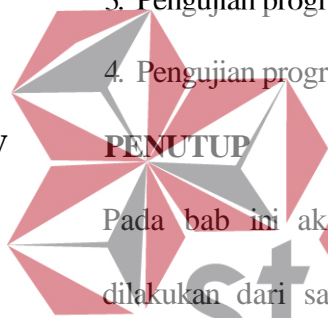
## **BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini akan dibahas pengujian sistem terhadap kemampuan mendeteksi golongan darah menggunakan metode component labelling. Untuk pengujiannya terdiri dari kebutuhan sistem, pengujian program. Pada pengujian program dibahas penggunaan metode untuk pendeteksian dan *tracking* obyek yang terdiri dari :

1. Pengujian program deteksi aglutinasi.
2. Pengujian program deteksi posisi koordinat X.
3. Pengujian program mengurutkan ukuran obyek.
4. Pengujian program mendapatkan hasil golongan darah

## **BAB V PENUTUP**

Pada bab ini akan diambil kesimpulan dari penelitian yang telah dilakukan dari saran untuk pengembangan di waktu yang akan datang.



INSTITUT BISNIS  
& INFORMATIKA  
**stikom**  
SURABAYA

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Darah Manusia**

Darah merupakan cairan yang bersirkulasi dalam tubuh manusia dan vertebrata yang berfungsi untuk mengirimkan zat-zat dan oksigen yang dibutuhkan oleh jaringan tubuh, serta mengangkut bahan kimia hasil metabolisme, selain itu darah juga berfungsi untuk pertahanan tubuh terhadap virus atau bakteri.

#### **2.2 Teori Pengujian Golongan Darah**

Dalam dunia kedokteran golongan darah manusia dibagi menjadi empat, yaitu A, B, AB, dan O. Pembagian ini dilakukan karena adanya perbedaan jenis karbohidrat dan protein pada permukaan membran sel darah merah. Untuk mengetahui jenis golongan darah seseorang perlu dilakukan uji laboratorium. Selama ini untuk pengujian golongan darah sering digunakan metode ABO, yang prosesnya dilakukan secara manual atau dengan cara meneteskan tiga jenis cairan atau reagen pada sampel darah. Jenis golongan darah sangat penting pada saat transfusi darah, seseorang harus menerima darah dari golongan darah yang sama dengan pendonor.

Dalam proses pengujian sampel darah menggunakan metode ABO, sampel darah akan ditetaskan suatu reagen, kemudian pada sampel darah akan terjadi proses aglutinasi atau penggumpalan darah. Penggumpalan darah disebabkan karena adanya interaksi antibodi dengan antigen yang terikat pada eritrosit.

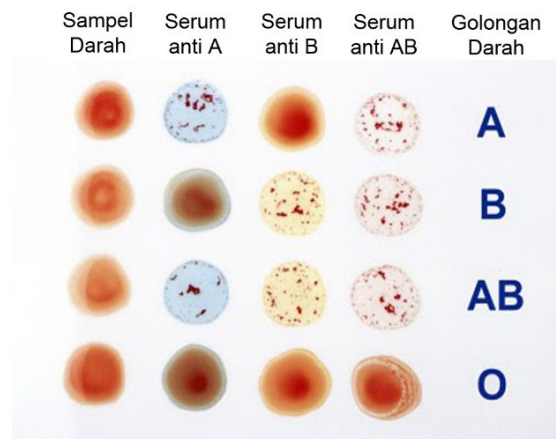
Darah memiliki antigen dan antibodi, dimana setiap masing-masing antigen dan antibodi terdiri dari A dan B, untuk lebih jelas melihat karakteristik golongan darah berdasarkan antigen dan antibodi dapat dilihat pada Tabel 2.1.

Tabel 2.1 Karakteristik Golongan Darah Sistem ABO

<b>Gol. Darah</b>	<b>Antigen</b>	<b>Antibodi</b>
A	A	B
B	B	A
AB	A dan B	-
O	-	A dan B

Selama ini pengujian darah dilakukan secara manual yaitu dengan cara meneteskan cairan reagen ke sampel darah untuk melihat tingkat penggumpalan darah menggunakan mata. Metode ini hanya dapat dilakukan oleh orang yang ahli dibidang ini. (Melati, Passarella, Primartha, & Murdiansyah, 2011).

Antisera atau antigen yang dipakai pada penentuan jenis golongan darah ini adalah antisera A dengan warna biru, antisera B dengan warna kuning, dan antisera AB tidak berwarna. Dalam sel darah manusia terdapat aglutinogen yang jika ditetesi dengan antisera akan menghasilkan penggumpalan. Hal ini terjadi karena di dalam antisera terdapat aglutinin yang sifatnya menggumpalkan aglutinogen. Gambar 2.1 merupakan gambar tipe golongan darah berdasarkan sistem ABO yang telah diberikan antisera A, B, dan AB.



Gambar 2.1 Sampel Golongan Darah

Pada Gambar 2.1 dapat dilihat penggumpalan yang terjadi pada setiap golongan darah itu berbeda-beda. Karena setiap golongan darah mempunyai suatu zat anti tertentu. Seperti pada golongan darah A mempunyai anti B, jika golongan darah A diberikan antisera A maka darah tersebut akan menggumpal, sedangkan untuk darah golongan B mempunyai anti A. darah golongan AB mempunyai anti A dan anti B dan golongan darah O tidak mempunyai zat anti.

### 2.3 Computer Vision

*Computer Vision* adalah disiplin ilmu yang mempelajari bagaimana merekonstruksi, menginterpretasikan, dan memahami sebuah tampilan 3 dimensi dan tampilan 2 dimensinya dalam hal sifat dari struktur tampilan tersebut. *computer vision* berkaitan dengan pemodelan dan meniru penglihatan manusia dengan menggunakan perangkat lunak dan perangkat keras pada komputer. *Computer vision* menggabungkan ilmu pengetahuan dalam bidang ilmu Komputer, teknik

elektro, matematika, fisiologi, dan ilmu kognitif. Diperlukan ilmu dari semua bidang tersebut untuk memahami dan simulasikan pengoperasian sistem penglihatan manusia (Helmirawan, 2012).

Untuk mendeteksi kemampuan manusia dalam menangkap informasi, sebuah teknologi *computer vision* harus terdiri dari banyak fungsi pendukung yang berfungsi secara penuh. Fungsi-fungsi pendukung tersebut antara lain :

1. Proses penangkapan citra/ gambar (*image acquisition*).
2. Proses pengolahan citra (*image processing*).
3. Analisa data citra (*image analysis*).
4. Proses pemahaman data citra (*image understanding*).

## 2.4 Open Computer Vision (OpenCV)

OpenCV adalah sebuah *library* yang berisi fungsi- fungsi pemrograman untuk teknologi *Computer vision* secara *real time*. OpenCV bersifat *open source*, bebas digunakan untuk hal-hal yang bersifat akademis dan komersial. Di dalamnya terdapat *interface* untuk C++, C, Python, dan Java juga dapat berjalan pada *Operation Sistem* Windows, Linux, Android, dan Mac. Terdapat lebih dari 2500 algoritma dalam OpenCV, digunakan lebih dari 40 ribu orang. Penggunaannya antara lain pada seni interaktif, inspeksi tambang, menampilkan peta di *web* melalui teknologi robotika.

Pada awalnya OpenCV ditulis dengan menggunakan bahasa C namun sekarang secara menyeluruh sudah menggunakan antarmuka bahasa C++ dan seluruh pengembangannya terdapat dalam format bahasa C++. Contoh aplikasi dari

OpenCV yaitu interaksi manusia dan Komputer, identifikasi, segmentasi, dan pengenalan obyek, pengenalan wajah, pengenalan gerak, gerakan diri, dan pemahaman gerak, struktur dari gerakan, kalibrasi stereo dan beberapa kamera serta komputasi mendalam, *robotic* (Helmirawan, 2012).

Fitur-fitur yang terdapat pada OpenCV antara lain:

1. Manipulasi data image (alokasi, rilis, duplikasi, pengaturan, konversi).
2. Image dan I/O video (masukan berbasis file dan kamera, keluaran *image/video file*).
3. Manipulasi matriks dan vector serta aljabar linier (Produk, solusi, eigenvalues, SVD).
4. Beragam struktur data dinamis (daftar, baris, grafik).
5. Dasar pengolahan citra (filter, deteksi tepi, deteksi sudut, pengambilan sampel dan interpolasi, konversi warna, operasi morfologi, histogram).
6. Analisis struktur (komponen yang berhubungan, pengolahan kontur, transformasi jarak, variasi momen, transformasi Hough, perkiraan polygonal, menyesuaikan garis, Delaunay triangulation).
7. Kalibrasi kamera (menemukan dan menelusuri pola kalibrasi, dasar estimasi matriks, estimasi homografi, korespondensi stereo).
8. Analisis gerakan (*optical flow*, segmentasi gerakan, penelusuran).
9. Pengenalan obyek (metode eigen, HMM).
10. *Graphical User Interface* Atau GUI (menampilkan image/ video, penanganan mouse dan *keyboard, scroll-bars*).
11. Pelabelan image (garis, polygon, gambar teks).

Modul-modul yang terdapat pada OpenCV antara lain:

1. cv - fungsi utama OpenV.
2. cvaux - fungsi penolong OpenCV.
3. cxcore - pendukung struktur data dan aljabar linier.
4. highgui - fungsi GUI .

## 2.5 Grayscale

Untuk melakukan perubahan suatu gambar *full color* (RGB) menjadi suatu citra *grayscale* (gambar keabuan), metode yang umum digunakan, yaitu:

dimana :

$$(R + G + B)/3$$

R : Unsur warna merah  
G : Unsur warna hijau  
B : Unsur warna biru

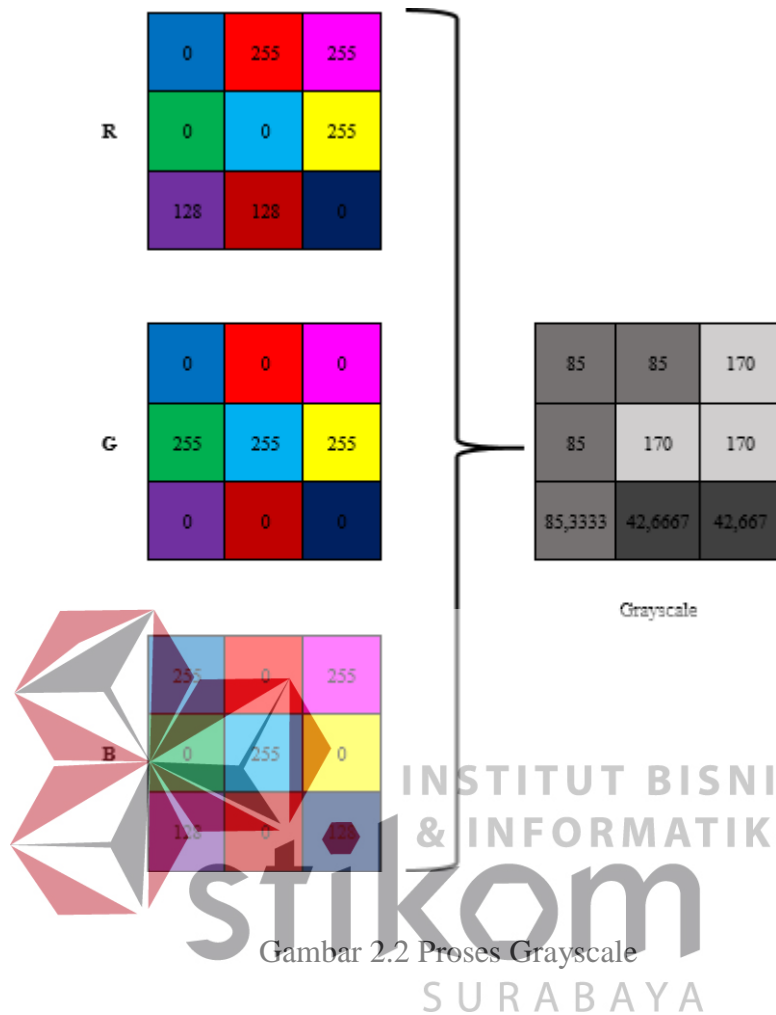


INSTITUT BISNIS  
& INFORMATIKA  
**stikom**  
SURABAYA

Nilai yang dihasilkan dari persamaan diatas akan diinput ke masing- masing unsur warna dasar citra *grayscale*.

Untuk ilustrasi proses *grayscale* lihat gambar 2.2 di bawah ini.





Gambar 2.2 Proses Grayscale

Gambar 2.2 menunjukkan terdapat 3 layer unsur warna dasar R, G, dan B nilai-nilai di tiga layer tersebut di bagi 3 untuk mendapatkan nilai *grayscale*.

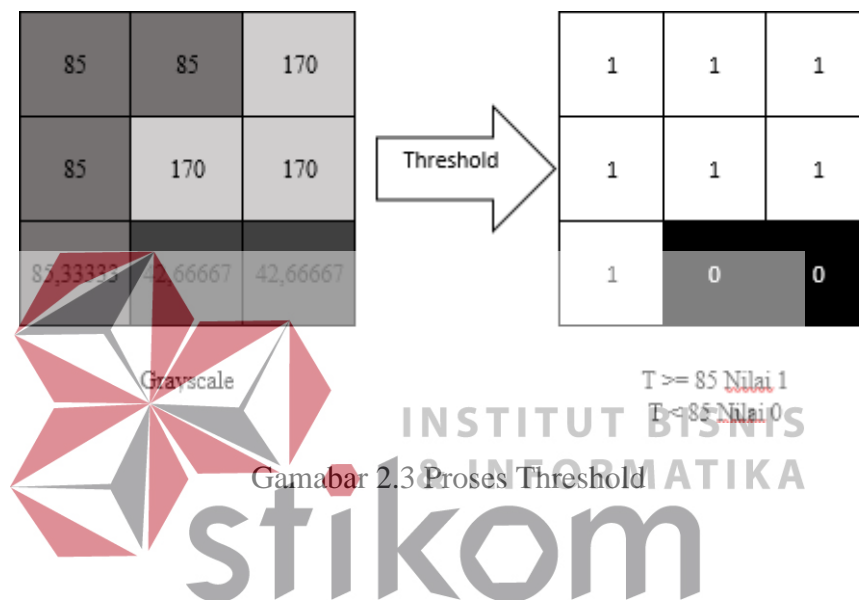
## 2.6 Threshold

Secara umum proses *thresholding* terhadap citra *grayscale* bertujuan menghasilkan citra biner, secara matematis dapat ditulis sebagai berikut.

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

Dengan  $g(x,y)$  adalah citra biner dari citra grayscale  $f(x,y)$ , dan  $T$  menyatakan nilai *threshold* (Purba, D. 2010). Nilai  $T$  ditentukan dengan menggunakan metode *thresholding* global.

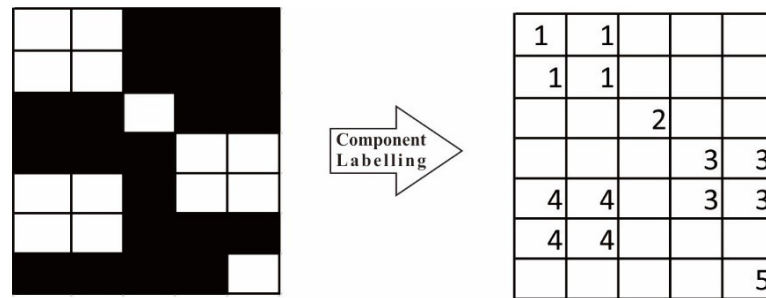
Sebagai ilustrasi proses threshold lihat pada gambar 2.3 di bawah.



Ilustrasi di atas menunjukkan nilai *grayscale* yang bernilai  $\geq 85$  akan bernilai 1 atau berwarna putih, dan sedangkan yang bernilai  $< 85$  bernilai 0 atau berwarna hitam.

## 2.7 Metode *Component Labelling*

Metode *component labeling* atau *Labeling* adalah suatu proses pemberian label yang sama pada sekumpulan piksel pembentuk obyek yang saling berdekatan pada suatu citra. Proses *Component Labelling* dilakukan pada citra biner, seperti diilustrasikan pada gambar 2.4 (Hadi & Samara, 2012).

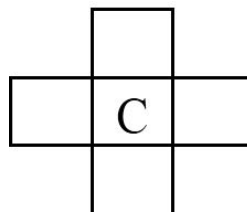


Gambar 2.4 Setelah Diproses Dalam Metode *Component Labelling*

### 2.7.1 *Spoke Filter*

*Spoke Filter* adalah metode yang ditemukan oleh minor dan sklansky ini dapat mendeteksi *blob* dalam berbagai ukuran dengan menggunakan sebuah filter yang dinamakan *spoke filter* atau biasa juga disebut *Adaptive Spatial Erosion Filter*.

Dengan menggunakan deteksi 4 tetangga lihat pada gambar 2.3, yaitu suatu pola yang sudah ditentukan yang memiliki 4 lengannya dan 1 *counter* di tengahnya. Menghitung berapakah *blob* atau gumpalan didalam suatu gambar.,



Gambar 2.5 4 Tetangga

Berikut adalah langkah kerja 4 tetangga dalam metode CCL

1. *Counter* = 0
2. Cari yang bernilai 1
3. Jika ada nilai 1, *counter* di *increment*

4. Jika sudah berisi angka 1 pada seluruh lengan, nilai pada lengan di set sama dengan nilai *counter*
5. dan Jika seluruh lengan bernilai nol, kembali ke langkah no 2

### 2.7.2 Automatic scale selection

*Automatic Scale Selection* prinsip dari metode ini adalah mengansumsika n sebuah *scale level*, dimana beberapa kombinasi dari turunan normalisas inya mengasumsikan sebuah nilai maksimum atas *scale* tersebut, yang mencerminka n ukuran dari *blob* yang sesuai. Metode ini dikemukakan oleh Lindeberg.

## 2.8 Kamera

Kamera diperlukan sebagai pengambil input utama dari sistem ini, yaitu sampel darah. Pada perancangan ini kamera yang digunakan adalah kamera yang mempunyai kemampuan mengambil gambar sebesar 5 *megapixels* dan kemampuan merekam gambar minimum sebesar 640x480 *pixels*.

Karena semakin detail gambar yang ditangkap kamera maka reaksi yang di hasilkan sampel darah yang telah di campur dengan reagen akan dapat terlihat dengan baik.



Gambar 2.6 Kamera

## BAB III

### METODE PENELITIAN

#### 3.1 Metode Penelitian

Metode penelitian tugas akhir ini adalah perancangan menggunakan *image processing* sebagai pengganti penglihatan manusia untuk pengamatan aglutinasi atau penggumpalan pada sampel darah yang telah dicampur dengan reagen/antisera A,B, dan AB. Pengamatan tersebut dilakukan oleh kamera yang kemudian gambar yang diambil diteruskan ke komputer untuk diproses

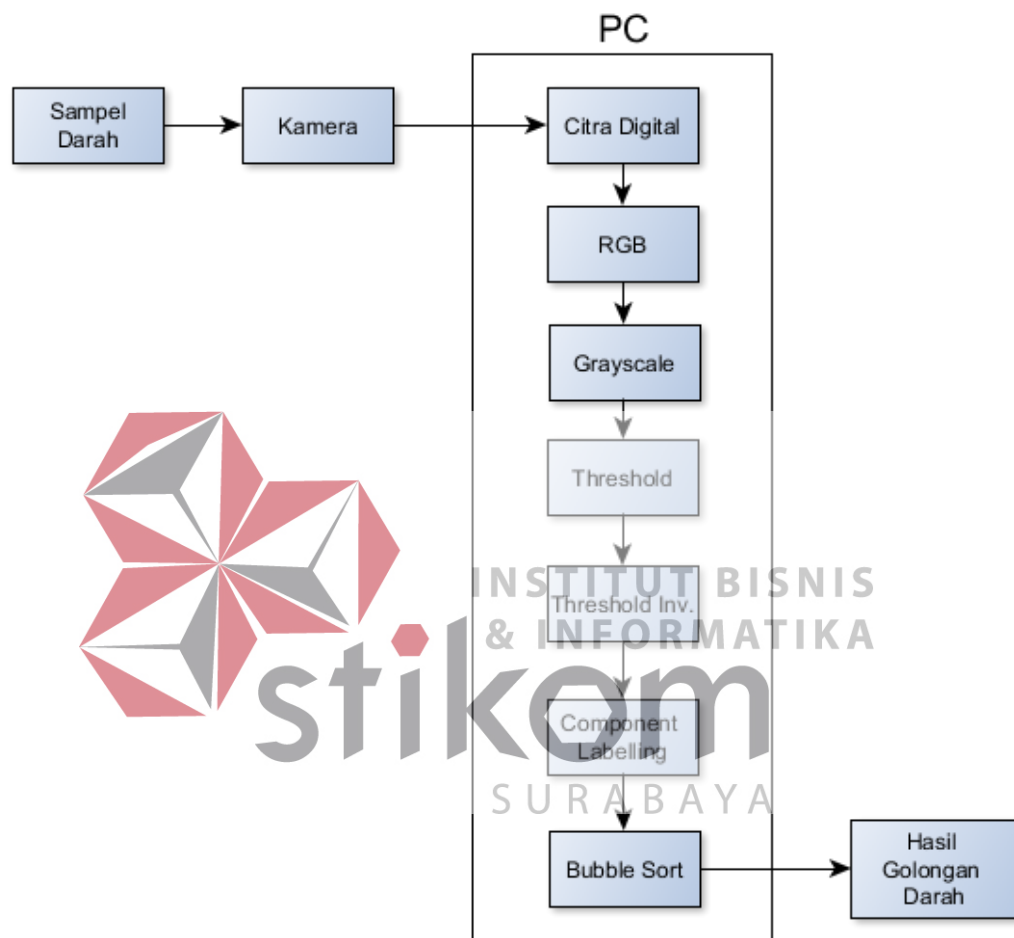


Gambar 3.1 Blok Diagram Sistem Pendeteksi Golongan Darah

Dari gambar 3.1 data yang diperoleh dari pengambilan gambar oleh kamera akan diolah komputer dengan *image processing* menggunakan metode component labelling untuk mendeteksi aglutinasi, setelah itu akan dibandingkan sesuai metode ABO daerah mana sajakah yang terdapat aglutinasi sehingga *output* akan didapatkan dan ditampilkan pada monitor.

### 3.2 Prosedur Penelitian

Prosedur penelitian yang digunakan pada pengerjaan Tugas Akhir ini ditunjukkan oleh Gambar 3.2.



Gambar 3.2 Prosedur Penelitian

Setelah citra gambar atau sampel didapatkan oleh kamera maka akan diproses oleh komputer, pertama kali gambar masih dalam bentuk RGB (*Red, Green, and Blue*) / berwarna akan ditransformasikan menjadi *grayscale* lalu akan diubah ke dalam bentuk *threshold* agar menjadi gambar biner atau bernilai 1 dan 0 dengan ambang batas tertentu sehingga gambar hanya terdiri dari hitam pada

bagian obyek dan putih pada bagian latar belakang sehingga untuk dapat menjalankan metode *component labelling* untuk mencari obyek dengan mencari nilai 1 pada pixel maka nilai threshold harus dibalik, yaitu dengan menggunakan *Invers* ( $X^{255}$ ) setelah mengalami proses *invers* maka nilai akan dibalik sehingga obyek akan menjadi putih atau bernilai 1 dan latar menjadi hitam atau bernilai 0. Lalu proses metode *component labelling* dapat dikerjakan yaitu dengan cara mendeteksi nilai 1 yang saling berdekatan hingga bertemu nilai 0 pada suatu citra digital dan memberi label pada obyek yang bernilai 1 itulah yang dinamakan *blob*. Setelah hasil metode *component labelling* selesai maka akan dibandingkan sesuai teori dasar pemeriksaan golongan darah yaitu metode ABO dan akan muncul hasilnya dari hasil ini sehingga dapat ditampilkan pada layar monitor.

Dalam penelitian ini akan dilakukan beberapa metode dalam mendeteksi sampel golongan darah. metode yang telah dilakukan dimulai dari program deteksi obyek aglutinasi dengan metode *component labelling*, program deteksi posisi koordinat X, program mengurutkan ukuran obyek aglutinasi dari besar ke kecil menggunakan metode *bubble sort*, dan program mendapatkan hasil golongan darah.

### **3.2.1 Program Deteksi Obyek Aglutinasi dengan Metode *Component Labelling***

Faktor keberhasilan proses *component labelling* adalah kunci untuk keberhasilan sistem ini dikarenakan *component labelling* menjadi pokok bahasan

apakah metode ini dapat diterapkan untuk dapat mendeteksi golongan darah pada sampel darah atau tidak, untuk menjalankan metode ini gambar citra RGB atau gambar warna harus dirubah kebentuk biner menggunakan thresholding sehingga gambar akan menjadi hitam putih tau bernilai 1 dan 0 dengan latar putih dan Obyek hitam namun untuk deteksi Obyek pada component labelling dilakukan dengan cara mencari nilai 1 pada matriks gambar maka nilai thresholding harus dirubah sehingga Obyeknya akan berwarna putih atau bernilai 1 dengan cara *inverse threshold* sehingga proses *component labelling* dapat dilakukan dengan cara mencari nilai satu yang berdekatan pada matriks gambar dan memberi label pada Obyek yang bernilai 1 ilustrasinya seperti ditunjukkan pada gambar 3.3.



Gambar 3.3 Gambar *Threshold* Menjadi Gambar *Blob*

### 3.2.2 Program Deteksi Posisi Koordinat X

Program ini dilakukan untuk mendeteksi posisi koordinat X suatu Obyek aglutinasi sampel darah. Pengujian ini bertujuan untuk mendeteksi posisi Obyek yang telah diberi label oleh component labelling, sehingga user dapat mengetahui posisi *centroid* obyek yang telah dideteksi.



### 3.2.3 Program Mengurutkan Ukuran Obyek Aglutinasi dari Besar ke Kecil menggunakan Metode *Bubble Sort*

Program ini dilakukan untuk mengurutkan ukuran obyek aglutinasi sampel darah dari besar ke kecil menggunakan metode *bubble sort*, pengujian ini bertujuan untuk mengetahui ukuran yang terbesar sampai yang terkecil dari obyek untuk memudahkan proses analisa dan meletakkan ukuran yang terbesar dipaling atas sehingga berguna untuk menentukan golongan darah O karena golongan ini mempunyai tiga ukuran obyek yang besar sehingga dapat memudahkan *user* dalam analisa. Pada pengujian ini dapat diilustrasikan pada gambar 3.4 dibawah.



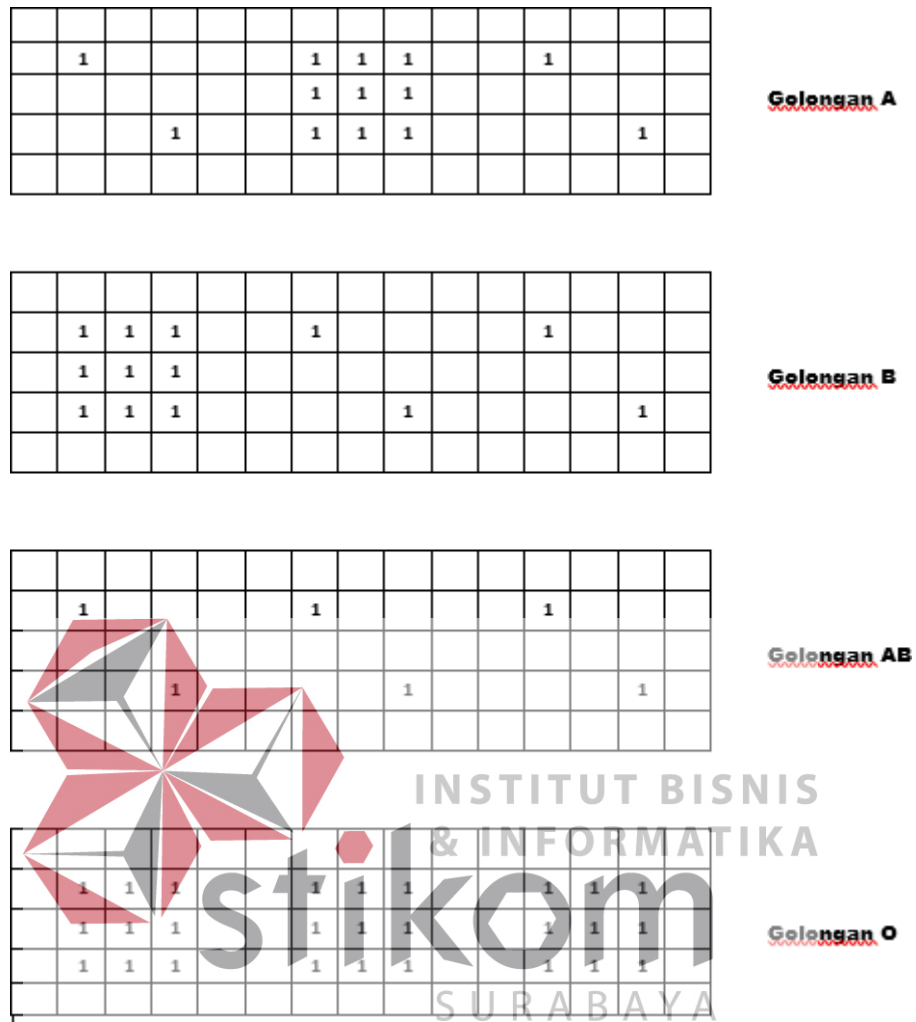
Gambar 3.4 Metode *Bubble Sort*

Pada ilustrasi gambar 3.4 adalah salah satu metode untuk membuat suatu nilai acak menjadi runtut, pada ilustrasi diatas terdapat nilai acak 1 sampai dengan 5 lalu algoritma yang dibuat akan menukar nilai bila nilai pada sisi kanannya lebih

kecil dari nilai tersebut sehingga hasilnya nilai menjadi runtut dari nilai 1 sampai 5.

### 3.2.4 Program Mendapatkan Hasil Golongan Darah

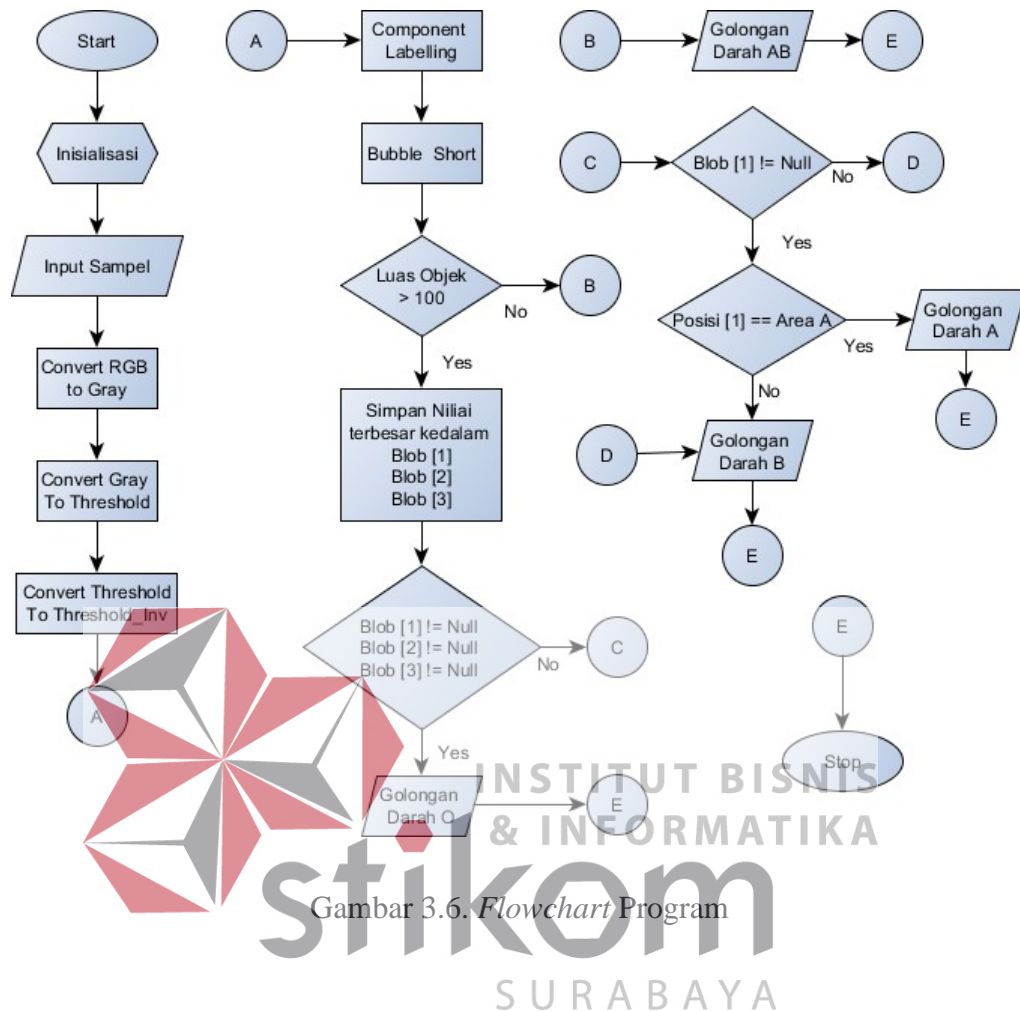
Program ini digunakan untuk mendapatkan hasil golongan darah dari citra sampel darah yang telah diproses oleh component labelling dan algoritma pendukung lainnya. Untuk golongan darah A dan B didapatkan dari hasil tracking Obyek terbesar dan melihat posisi koordinat X dari Obyek tersebut apakah berada pada area antigen A atau berada pada antigen B, jika Obyek terbesar berada pada area antigen B maka dapat disimpulkan golongan darah A, jika obyek terbesar berada pada area antigen A maka dapat disimpulkan golongan darah B, jika terdapat tiga obyek besar pada area antigen A, B, dan AB maka dapat disimpulkan golongan darah O B, dan jika tidak ada obyek besar pada area antigen A,B, dan AB maka dapat disimpulkan golongan darah AB. Ilustrasi *tracking* obyek menggunakan component labelling sesuai golongan darah dapat dilihat pada gambar 3.5.



Gambar 3.5 Ilustari Golongan Darah.

### 3.3 Flowchart Program

Pada penelitian ini mempunyai *flowchart* program seperti yang ditunjukkan pada Gambar 3.6.



Gambar 3.6. Flowchart Program

Penjelasan dari *flowchart* pada Gambar 3.7 adalah sebagai berikut :

1. Inisialisasi merupakan proses deklarasi variabel- variabel yang digunakan didalam program. Pada penelitian ini variabel- variabel seperti variabel yang digunakan untuk menampung gambar tangkapan kamera, dll.
2. Input sampel merupakan proses memasukkan gambar sampel darah dari directory ke program.
3. *Convert RGB To Gray* merupakan proses untuk merubah gambar RGB atau berwarna menjadi *grayscale*.

4. *Convert Gray To Threshold* untuk mengubah gambar *grayscale* menjadi gambar hitam putih yang bernilai biner 1 dan 0.
5. *Convert Threshold To Threshold\_Inv.* merupakan proses merubah nilai 1 menjadi 0 dan 0 menjadi 1 sehingga Obyek yang mula-mula berwarna hitam menjadi putih.
6. *Componen Labelling* merupakan proses untuk membuat label pada Obyek aglutinasi yang telah dideteksi dan mendapatkan besar Obyek pada sampel darah.
7. *Bubble sort* adalah proses mengurutkan nilai dari besar ke kecil atau sebaliknya sehingga Obyek aglutinasi akan runtut dari ukuran besar ke ukuran kecil.
8. Percabangan  $\text{luas obyek} > 100$  digunakan untuk mencari ukuran obyek aglutinasi yang bernilai lebih dari 100.
9. Simpan nilai terbesar yaitu yang nilainya  $> 100$  kedalam Variabel Array Blob [1], Blob [2], dan Blob [3].
10. Jika variabel array Blob [1], Blob [2], dan Blob [3] tidak kosong maka akan di golongankan O.
11. *Output* kelayar golongan darah O.
12. *Output* kelayar golongan darah AB.
13. Percabangan untuk melihat apakah variabel array Blob [1] tidak kosong .

14. Percabangan untuk melihat posisi obyek aglutinasi yang disimpan pada variabel array Blob [1].
15. *Output* kelayar golongan darah B.
16. *dan Output* kelayar golongan darah A.



## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Kebutuhan Sistem

Sebelum melakukan deteksi dan *tracking* obyek dibutuhkan perangkat lunak yang dapat menunjang penelitian. Perangkat keras dan lunak yang digunakan dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Kebutuhan Perangkat Keras.

Perangkat Keras	Spesifikasi
Processor	Intel Pentium Processor T4400 (2.2 Ghz, 800 MHz FS3)
RAM	1GB
Sistem Operasi	Windows
Web Cam	8 Mega Pixel

Tabel 4.2 Kebutuhan Perangkat Lunak.

Perangkat Lunak	Uraian
OpenCV	Aplikasi yang digunakan untuk teknik <i>image processing</i> .

#### 4.2 Pengujian Nilai *Threshold*

Pengujian dilakukan untuk menetapkan nilai ambang atau *threshold* agar mendapatkan prosentase kebenaran yang diinginkan.

#### 421 Tujuan Pengujian Nilai *Threshold*

Tujuan dari pengujian ini adalah untuk mendapatkan nilai ambang batas atau *threshold* yang mendapatkan prosentase kebenaran paling tinggi.

#### 422 Alat Yang Digunakan Pada Pengujian Nilai *Threshold*

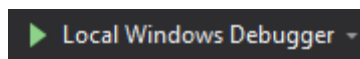
Untuk melakukan pengujian ini, diperlukan alat antara lain:

1. *Personal Computer*
2. Kamera *Webcam*
3. Layar monitor
4. Program OpenCV

#### 423 Prosedur Pengujian Nilai *Threshold*

1. Aktifkan *Personal Komputer*.
2. Hubungkan *webcam* pada *port usb computer*.
3. Jalankan program OpenCV pada Visual Studio.
4. Posisikan kamera pada obyek yang diberi dasar putih agar obyek dapat terlihat jelas diatas kaca preparat.
5. Ubah nilai *threshold*, pada proyek ini penulis menggunakan nilai 50-100.
6. Tekan tombol F5 untuk melakukan *debuging* atau jalankan program

dengan klik



pada menu bar untuk

menjalankan program.



## 424 Hasil Pengujian Nilai Threshold

Tabel 4.3 Pengujian Nilai *Threshold*

NO	NAMA	Kelamin	Gol.	Gol. Darah dan Nilai T 50-100						Hitung Benar Nilai T 50-100					
				50	60	70	80	90	100	50	60	70	80	90	100
1	Subjek 1	L	B	B	B	B	B	B	B	1	1	1	1	1	1
2	Subjek 2	L	A	A	A	A	A	A	O	1	1	1	1	1	0
3	Subjek 3	L	O	O	O	O	O	O	O	1	1	1	1	1	1
4	Subjek 4	L	B	B	B	B	B	B	B	1	1	1	1	1	1
5	Subjek 5	L	A	A	A	A	A	A	X	1	1	1	1	1	0
6	Subjek 6	L	B	B	B	B	B	B	X	1	1	1	1	1	0
7	Subjek 7	P	O	O	O	O	O	O	X	1	1	1	1	1	0
8	Subjek 8	L	O	O	O	O	O	O	X	1	1	1	1	1	0
9	Subjek 9	P	O	B	B	B	O	O	X	0	0	0	1	1	0
10	Subjek 10	P	B	B	B	B	B	A	X	1	1	1	1	0	0
11	Subjek 11	L	O	O	O	O	O	O	O	1	1	1	1	1	1
12	Subjek 12	L	B	B	B	B	B	O	X	1	1	1	1	0	0
13	Subjek 13	L	AB	AB	AB	AB	AB	AB	X	1	1	1	1	1	0
14	Subjek 14	L	A	A	A	A	A	A	X	1	1	1	1	1	0
15	Subjek 15	L	O	O	O	O	O	O	X	1	1	1	1	1	0
16	Subjek 16	L	O	O	O	O	O	O	X	1	1	1	1	1	0
17	Subjek 17	L	B	B	B	B	B	B	X	1	1	1	1	1	0
18	Subjek 18	L	O	O	O	O	O	O	X	1	1	1	1	1	0
19	Subjek 19	L	B	AB	AB	B	B	B	B	0	0	1	1	1	1
20	Subjek 20	P	O	O	O	O	O	O	X	1	1	1	1	1	0
21	Subjek 21	P	O	B	B	O	O	O	X	0	0	1	1	1	0
22	Subjek 22	P	A	A	A	A	A	A	X	1	1	1	1	1	0
23	Subjek 23	L	O	O	O	O	O	O	O	1	1	1	1	1	1
24	Subjek 24	P	B	B	B	B	B	B	B	1	1	1	1	1	1
TOTAL BENAR										21	21	23	24	22	7

Pengertian simbol :

1. A : Golongan darah A
2. B : Golongan darah B
3. O : Golongan darah O

4. AB : Golongan darah AB
5. X : Tidak terdeteksi
6. 1 : Benar
7. 0 : Salah
8. L : Laki-laki
9. P : Perempuan

Dari table 4.3 diatas terdapat 24 sampel dan hanya 1 nilai *threshold* yang benar semua yaitu nilai *threshold* 80.

#### 4.3 **Pengujian Program Deteksi Aglutinasi Dengan Metode *Component Labelling***

Pengujian bertujuan untuk melakukan deteksi obyek aglutinasi sampel darah dan memberi label disetiap obyek aglutinasi sampel darah.

##### 4.3.1 **Tujuan Program Deteksi Aglutinasi Dengan Metode *Component Labelling***

Tujuan dari pengujian ini untuk mengetahui proses *component labelling* bekerja dengan baik.

##### 4.3.2 **Alat yang Digunakan Pada Pengujian Program Deteksi Aglutinasi Dengan Metode *Component Labelling***

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain:

1. *Personal Computer*
2. *Kamera Webcam*

3. Layar monitor
4. Program OpenCV

#### 433 Prosedur Pengujian Program Deteksi Aglutinasi Dengan Metode

##### *Component Labelling*

Langkah-langkah untuk melakukan pengujian program deteksi warna adalah sebagai berikut:

1. Aktifkan Personal Komputer.
2. Hubungkan *webcam* pada *port* *usb computer*.
3. Jalankan program OpenCV pada Visual Studio.
4. Posisikan kamera pada obyek yang diberi dasar putih agar obyek dapat terlihat jelas diatas kaca preparat.
5. Tekan tombol F5 untuk melakukan *debugging* atau jalankan program dengan klik  pada menubar untuk menjalankan program.

#### 434 Hasil Pengujian Program Deteksi Aglutinasi Dengan Metode

##### *Component Labelling*

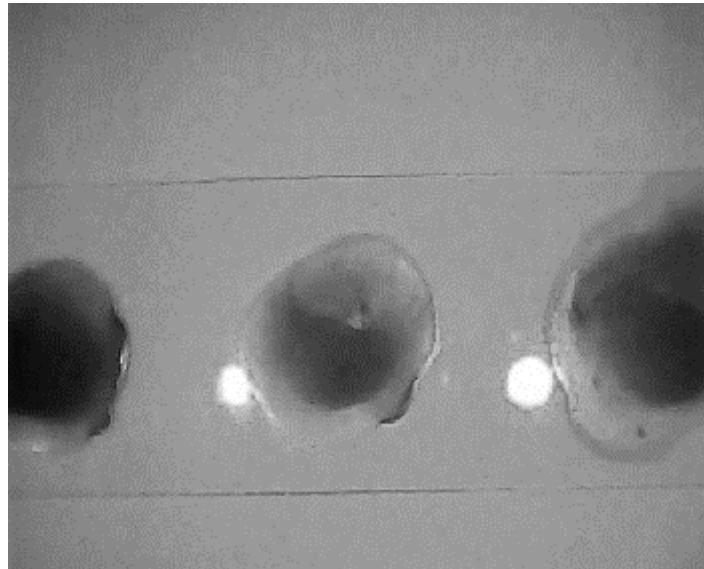
Tampilan hasil *capture* sebelum ada objek dibawah kamera ditunjukkan pada gambar 4.1 dan 4.2 sedangkan *capture* setelah diberi obyek ada pada gambar 4.3, 4.4, dan 4.5.



Gambar 4.1 Gambar Asli Sebelum Ada Obyek



Gambar 4.2 Gambar Asli Setelah Ada Obyek



Gambar 4.3 Gambar Gray Setelah Ada Obyek



Gambar 4.4 Gambar Threshold Setelah Ada Obyek

```

Blob [ 1 ]= 2
Blob [ 2 ]= 7
Blob [ 3 ]= 23
Blob [ 4 ]= 2
Blob [ 5 ]= 1
Blob [ 6 ]= 1
Blob [ 7 ]= 2
Blob [ 8 ]= 1
Blob [ 9 ]= 12
Blob [ 10 ]= 1
Blob [ 11 ]= 2
Blob [ 12 ]= 153
Blob [ 13 ]= 246
Blob [ 14 ]= 243

```



Gambar 4.5 Hasil Pembacaan Program Component Labelling.

Pada gambar 4.5 dapat dilihat bahwa terdapat 14 *blob* atau obyek yang terdeteksi. Obyek yang terdeteksi tersebut adalah gumpalan darah yang bernilai 1 atau berwarna putih sehingga *component labelling* memberikan tanda dengan nomor secara otomatis dengan cara *increment* atau menambahkan satu setiap menemukan suatu obyek yang bernilai 1.

#### 4.4 Pengujian Program Deteksi Posisi Koordinat X

Pengujian program deteksi koordinat X ini digunakan untuk mengetahui posisi koornat X pada *centroid* objek aglutinasi pada sampel darah.

#### 441 Tujuan Pengujian Program Deteksi Posisi Koordinat X

Pengujian ini bertujuan untuk melakukan pendeteksian posisi koornat X pada *centroid* objek aglutinasi pada sampel darah.

#### 442 Alat yang Digunakan pada Pengujian Program Deteksi Posisi Koordinat X

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain:

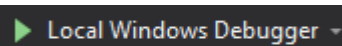
1. *Personal Computer*
2. Kamera *Webcam*
3. Layar Monitor
4. dan Program OpenCV.

#### 443 Prosedur Pengujian Program Deteksi Posisi Koordinat X

Langkah-langkah untuk melakukan pengujian program deteksi posisi koordinat X adalah sebagai berikut:

1. Aktifkan Personal Komputer.
2. Hubungkan webcam pada *port usb computer*.
3. Jalankan program OpenCV pada Visual Studio.
4. Posisikan kamera pada obyek yang diberi dasar putih agar obyek dapat terlihat jelas diatas kaca preparat.
5. dan Tekan tombol F5 untuk melakukan debugging atau jalankan program

dengan klik



pada menubar untuk menjalankan program.

#### 444 Hasil Pengujian Program Deteksi Posisi Koordinat X

Tampilan hasil pembacaan koordinat X pada program dapat dilihat pada gambar 4.6.



```

Blob [ 1 ]= 8
X :-1.#IND

Blob [ 2 ]= 18
X :51.4697

Blob [ 3 ]= 4
X :-1.#IND

Blob [ 4 ]= 1
X :-1.#IND

Blob [ 5 ]= 2
X :-1.#IND

Blob [ 6 ]= 1
X :-1.#IND

Blob [ 7 ]= 1
X :-1.#IND

Blob [ 8 ]= 2
X :-1.#IND

Blob [ 9 ]= 4
X :-1.#IND

Blob [ 10 ]= 1
X :-1.#IND

Blob [ 11 ]= 146
X :172.04

Blob [ 12 ]= 4
X :184.5

Blob [ 13 ]= 246
X :27.6347

Blob [ 14 ]= 236
X :328.017

```

Gambar 4.6 Hasil Pembacaan Program Koordinat X.

Dapat dilihat pada gambar 4.6 terdapat 14 *blob* dan masing-masing *blob* mempunyai posisi koordinat X, koordinat X tersebut diambil dari posisi *centroid* pada *blob* atau obyek, pada gambar dapat dilihat bahwa *blob* yang ukurannya kecil tidak dapat tampilan nilai koordinat X dikarenakan ukurannya yang terlalu kecil dan tidak masuk dalam kualifikasi ukuran minimum *blob*.



#### **4.5      Pengujian Program Mengurutkan Ukuran Obyek Aglutinasi Dari Besar Ke Kecil**

Pengujian ini dilakukan untuk mengurutkan ukuran Obyek aglutinasi sampel darah dari besar ke kecil menggunakan metode *bubble sort*.

##### **4.5.1    Tujuan Program Mengurutkan Ukuran Obyek Aglutinasi Dari Besar Ke Kecil**

Pengujian ini bertujuan untuk mengetahui ukuran yang terbesar sampai yang terkecil dari Obyek untuk memudahkan proses analisa dan meletakkan ukuran yang terbesar dipaling atas sehingga berguna untuk menentukan golongan darah O karena golongan ini mempunyai tiga ukuran Obyek yang besar sehingga dapat memudahkan *user* dalam analisa.


##### **4.5.2    Alat yang Digunakan pada Pengujian Program Mengurutkan Ukuran Obyek Aglutinasi Dari Besar Ke Kecil**

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain:

1. *Personal Computer*
2. *Kamera Webcam*
3. *Layar Monitor*
4. *dan Program OpenCV*

#### 453      **Prosedur Pengujian Program Mengurutkan Ukuran Obyek Aglutinasi Dari Besar Ke Kecil**

Langkah-langkah untuk melakukan pengujian terhadap jarak obyek dengan kamera adalah sebagai berikut:

1. Aktifkan Personal Komputer.
2. Hubungkan *web cam* pada *port usb computer*.
3. Jalankan program OpenCV pada Visual Studio.
4. Posisikan kamera pada obyek yang diberi dasar putih agar obyek dapat terlihat jelas diatas kaca preparat.
5. Tekan tombol F5 untuk melakukan *debuging* atau jalankan program dengan klik  pada menu bar untuk menjalankan program.

#### 454      **Hasil Pengujian Program Mengurutkan Ukuran Obyek Aglutinasi Dari Besar Ke Kecil**

Pengujian mengurutkan ukuran obyek aglutinasi dari besar ke kecil dengan metode *bubble sort* dan hasil dari pengujian dapat dilihat pada gambar 4.8.



```

Blob [ 1 ]= 14
X :194.5

Blob [ 2 ]= 18
X :50.0556

Blob [ 3 ]= 1
X :-1.#IND

Blob [ 4 ]= 2
X :-1.#IND

Blob [ 5 ]= 2
X :-1.#IND

Blob [ 6 ]= 2
X :-1.#IND

Blob [ 7 ]= 6
X :-1.#IND

Blob [ 8 ]= 2
X :-1.#IND

Blob [ 9 ]= 1
X :-1.#IND

Blob [ 10 ]= 152
X :169.852

Blob [ 11 ]= 242
X :26.5588

Blob [ 12 ]= 2
X :-1.#IND

Blob [ 13 ]= 241
X :326.97

```

INSTITUT BISNIS  
INFORMATIKA  
stikom  
SURABAYA

Gambar 4.7 Pembacaan Obyek Aglutinasi Dengan Ukuran Yang Masih Acak

```

Urutan ke-1 : 242
Urutan ke-2 : 241
Urutan ke-3 : 152
Urutan ke-4 : 18
Urutan ke-5 : 14
Urutan ke-6 : 6
Urutan ke-7 : 2
Urutan ke-8 : 2
Urutan ke-9 : 2
Urutan ke-10 : 2
Urutan ke-11 : 2
Urutan ke-12 : 1
Urutan ke-13 : 1

```

Gambar 4.8 Pembacaan Obyek Aglutinasi Setelah Diproses *Bubble Sort*

Dapat dilihat pada gambar 4.8 *blob* masih tidak berurutan sesuai luas piksel obyek dan pada gambar 4.9 *blob* telah diurutkan sesuai luas piksel obyek

yang bernilai besar hingga yang terkecil, dapat dilihat *blob* dengan nilai luas terbesar adalah 242 piksel dan nilai luas terkecil adalah 1.

#### **4.6 Pengujian Program Mendapatkan Hasil Golongan Darah**

Pengujian ini dilakukan untuk mendapatkan hasil golongan darah dari citra sampel darah yang telah diproses oleh *component labelling*.

##### **4.6.1 Tujuan Pengujian Program Mendapatkan Hasil Golongan Darah**

Pengujian ini bertujuan untuk mengetahui untuk mendapatkan hasil golongan darah dari citra sampel darah yang telah diproses oleh *component labelling*.

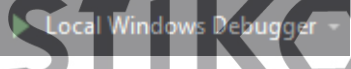
##### **4.6.2 Alat yang Digunakan pada Pengujian Program Mendapatkan Hasil Golongan Darah**

Untuk melakukan pengujian ini, diperlukan beberapa alat antara lain:

1. *Personal Computer*
2. *Kamera Webcam*
3. *Layar Monitor*
4. dan Program *OpenCV*.

#### 463      **Prosedur Pengujian Program Mendapatkan Hasil Golongan Darah**

Langkah-langkah untuk melakukan pengujian terhadap jarak obyek dengan kamera adalah sebagai berikut:

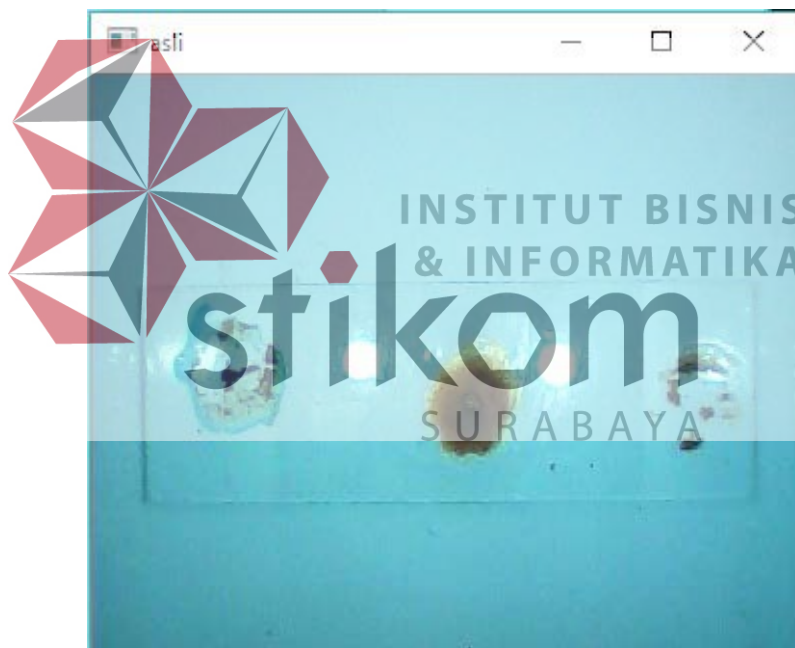
1. Aktifkan Personal Komputer.
2. Hubungkan *webcam* pada *port usb computer*.
3. Jalankan program OpenCV pada Visual Studio.
4. Posisikan kamera pada obyek yang diberi dasar putih agar obyek dapat terlihat jelas diatas kaca preparat.
5. Posisikan ketinggian kamera dan obyek 12 cm agar obyek dapat terlihat utuh oleh hasil *capture* kamera.
6. dan Tekan tombol F5 untuk melakukan *debuging* atau jalankan program dengan klik  pada menubar untuk menjalankan program.

#### 464      **Hasil Pengujian Program Mendapatkan Hasil Golongan Darah**

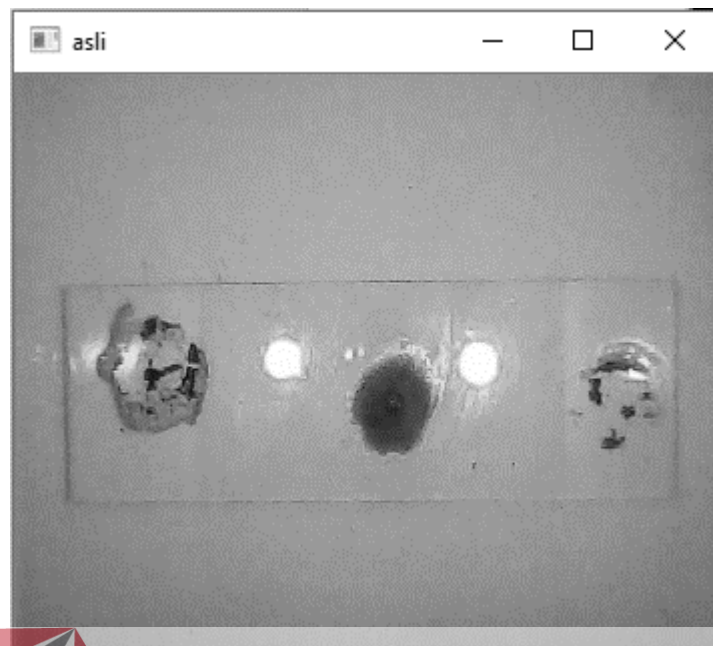
Hasil pengujian golongan darah dilakukan dengan 24 sampel darah yang diambil dari manusia melalui pembuluh darah pipa kapiler lalu di teteskan pada bidang preparat dan diberi tetesan antisera/antigen A,B, dan AB .

Setelah sampel telah diberi antisera maka diletakkan dibawah kamera dengan jarak antara kamera dan obyek adalah 12 cm sehingga seluruh bagian obyek dapat terlihat saat *capture* kamera. Lalu jalankan program untuk golongan darah A dan B didapatkan dari hasil tracking obyek terbesar dan melihat posisi

koordinat X dari obyek tersebut apakah berada pada area antigen A atau berada pada antigen B, jika obyek terbesar berada pada area antigen B maka dapat disimpulkan golongan darah A seperti pada gambar 4.9, 4.10, dan 4.11. Jika obyek terbesar berada pada area antigen A maka dapat disimpulkan golongan darah B seperti pada gambar 4.12, 4.13, dan 4.14. Jika terdapat tiga obyek besar pada area antigen A, B, dan AB maka dapat disimpulkan golongan darah O B ,dan jika tidak ada obyek besar pada area antigen A,B, dan AB maka dapat disimpulkan golongan darah AB.



Gambar 4.9 Sampel Darah A



Gambar 4.10 Sampel Darah A Dengan Proses *Grayscale*



Gambar 4.11 Sampel Darah A Dengan Proses *Threshold*

```

-----
                        Hasil
-----
Ukuran Blob = 123
  A      B      AB
-----
  |      |      |
  |      X      |
  |      |      |
-----
Blob berada pada koordinat X :187.952 ,maka masuk kedalam
Golongan : A

```

Gambar 4.12 Hasil Pembacaan Program Pada Sampel Darah A

Pada gambar 4.12 dapat dilihat bahwa letak blob besar dengan ukuran 123 piksel berada pada area B di koordinat X 187,952 sehingga dapat disimpulkan bahwa sampel tersebut golongan A.



Gambar 4.13 Sampel Darah B





Gambar 4.14 Sampel Darah B Dengan Proses *Grayscale*



Gambar 4.15 Sampel Darah B Dengan Proses *Threshold*

```

-----
                        Hasil
-----
Ukuran Blob = 192
  A      B      AB
-----
  x      |      |
-----
Blob berada pada koordinat X :80.0354 ,maka masuk kedalam
Golongan : B

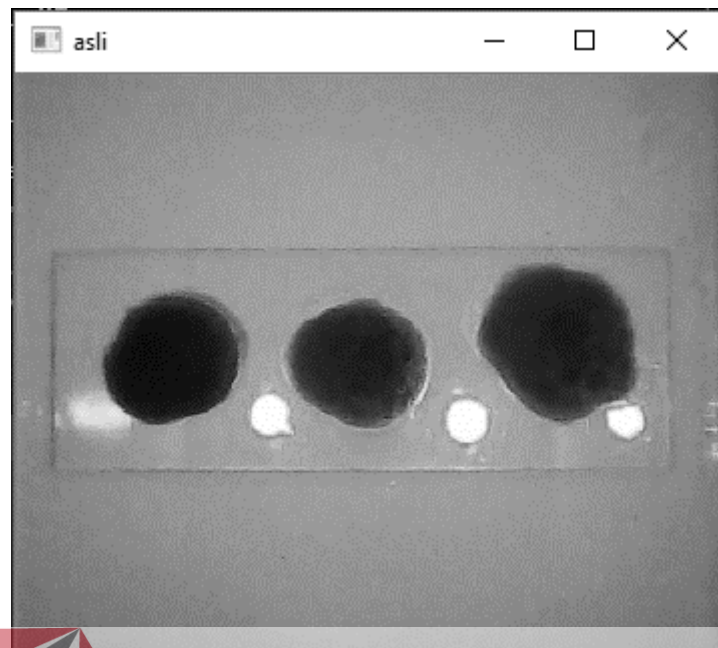
```

Gambar 4.16 Hasil Pembacaan Program Pada Sampel Darah B

Pada gambar 4.16 dapat dilihat bahwa letak *blob* besar dengan ukuran 192 piksel berada pada area A di koordinat X 80,0354 sehingga dapat disimpulkan bahwa sampel tersebut golongan B.



Gambar 4.17 Sampel Darah O



Gambar 4.18 Sampel Darah O Dengan Proses *Grayscale*



Gambar 4.19 Sampel Darah O Dengan Proses *Threshold*

```

-----
                        Hasil
-----
Urutan ke-1 :245
Urutan ke-2 :190
Urutan ke-3 :180

  A      B      AB
-----
  X      X      X

-----

Ketiga blob besar, maka masuk kedalam
Golongan : 0

```

Gambar 4.20 Hasil Pembacaan Program Pada Sampel Darah O

Pada gambar 4.20 dapat dilihat bahwa terdapat 3 *blob* besar dengan ukuran 245 piksel, 190 piksel, dan 180 piksel bila terdapat 3 *blob* besar pada sampel dapat disimpulkan bahwa sampel tersebut golongan O.



Gambar 4.21 Sampel Darah AB



Gambar 4.22 Sampel Darah AB Dengan Proses *Grayscale*



Gambar 4.23 Sampel Darah AB Dengan Proses *Threshold*



Gambar 4.24 Hasil Pembacaan Program Pada Sampel Darah AB

Pada gambar 4.24 dapat dilihat bahwa tidak terdapat *blob* besar di ketiga area maka sampel dapat disimpulkan bahwa sampel tersebut golongan AB.

Setelah dilakukan pengujian program dengan keterangan 24 sampel darah yang didapatkan secara acak, ambang batas threshold 80 dari 0 hingga 255, dan ukuran *blob* besar lebih dari 100p dapat dilihat di table 4.3 dibawah ini.

Tabel 4.4 Hasil Pengujian 24 Sampel

NO	NAMA	SEX	TYPE	PROGRAM	STATUS
1	Subjek 1	L	B	B	Benar
2	Subjek 2	L	A	A	Benar
3	Subjek 3	L	O	O	Benar
4	Subjek 4	L	B	B	Benar
5	Subjek 5	L	A	A	Benar
6	Subjek 6	L	B	B	Benar
7	Subjek 7	P	O	O	Benar
8	Subjek 8	L	O	O	Benar
9	Subjek 9	P	O	O	Benar
10	Subjek 10	P	B	B	Benar
11	Subjek 11	L	O	O	Benar
12	Subjek 12	L	B	B	Benar

13	Subjek 13	L	AB	AB	Benar
14	Subjek 14	L	A	A	Benar
15	Subjek 15	L	O	O	Benar
16	Subjek 16	L	O	O	Benar
17	Subjek 17	L	B	B	Benar
18	Subjek 18	L	O	O	Benar
19	Subjek 19	L	B	B	Benar
20	Subjek 20	P	O	O	Benar
21	Subjek 21	P	O	O	Benar
22	Subjek 22	P	A	A	Benar
23	Subjek 23	P	B	B	Benar
24	Subjek 24	L	O	O	Benar

Dengan melihat daftar sampel pada table 4.4 diatas dapat disimpulkan bahwa hasil uji program pada 24 sampel 100% benar.



## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil pengujian didapatkan beberapa hasil sebagai berikut :

1. Deteksi obyek atau *blob* menggunakan metode *component labelling* untuk mendeteksi golongan darah dapat diterapkan dengan baik dengan syarat pengambilan sampel darah dan tes golongan darah dengan antigen dilakukan dengan baik dan benar.
2. Koordinat X dapat terdeteksi dengan baik jika ukuran luas obyek atau *blob* tidak terlalu kecil.
3. Program deteksi golongan darah menggunakan metode *component labelling* ini dapat bekerja dengan baik jika ambang batas threshold bernilai 80.
4. Dan Program deteksi golongan darah menggunakan metode *component labelling* ini dapat bekerja dengan baik jika kualifikasi *blob* besar adalah  $>100$ .

#### 5.2. Saran

Berikut ini terdapat beberapa saran yang penulis berikan untuk peneliti berikutnya apabila ingin mengembangkan penelitian yang telah dibuat agar menjadi lebih baik adalah sebagai berikut:

1. Peneliti berikutnya diharapkan dapat mengembangkan program yang dapat membaca rhesus darah.



2. Peneliti berikutnya diharapkan dapat mengembangkan alat otomasinya untuk mencampurkan antigen pada sampel darah sehingga selain dapat membaca golongan darah sistem juga dapat melakukan prosedur pencampuran antigen secara otomatis.
3. Penelitian berikutnya diharapkan dapat mengembangkan program dengan tampilan *visual* GUI yang mempunyai nilai estetika baik sehingga interaksi awam dan komputer dapat dilakukan dengan mudah.



## DAFTAR PUSTAKA

- Andiani, N., & Akmaliah, I. F. (2011). Alat Pendeteksi Golongan Darah Manusia Berbasis Mikrontroler 89S51. *SANATIKA*, 292.
- Basavaprasad, B., & Ravi, M. (2014). *A Study On The Importance Of Image Processing And Its Applications. Ijret: International Journal of Research in Engineering and Technology*, 155.
- Hadi, S., & Samara, Y. R. (2012). Deteksi Objek Kendaraan Pada Citra Dijital Jalan Raya Menggunakan Metode Visi Komputer. 215.
- Helmirawan. (2012). Rancang Bangun Dan Analisis Sistem Pemantau Lalu Lintas Menggunakan OpenCV Dengan Algoritma Canny Dan Blob Detection. *FT UI*.
- Melati, E., Passarella, R., Primartha, R., & Murdiansyah, A. (2011). Desain dan Pembuatan Alat Pendeteksi Golongan Darah Menggunakan Mikrokontroler. *jurnal Generic*, 48.
- Perkasa, T. R., Widyantara, H., & Susanto, P. (2014). Rancang Bangun Pendeteksi Gerak Menggunakan Metode Image Subtraction Pada Single Board Computer (SBC). *Journal Of Control And Network System*, 90.

