

# **APLIKASI AUTOMATIC CROPPING PAS FOTO**

## **TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana  
Jurusan Teknik Informatika**



**Nama : Septian Aditya  
NIM : 07523061**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA  
2012**

**LEMBAR PENGESAHAN PEMBIMBING**

**APLIKASI AUTOMATIC CROPPING PAS FOTO**



**Izzati Muhimmah, ST, MSc., PhD.**

**LEMBAR PENGESAHAN PENGUJI**  
**APLIKASI AUTOMATIC CROPPING PAS FOTO**

Oleh :

**Nama : Septian Aditya**  
**NIM : 07523061**

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu  
Syarat untuk Memproleh Gelar Sarjana Jurusan Teknik Informatika  
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 23 April 2011

Tim Penguji,

**Izzati Muhimmah, ST, MSc., PhD.**

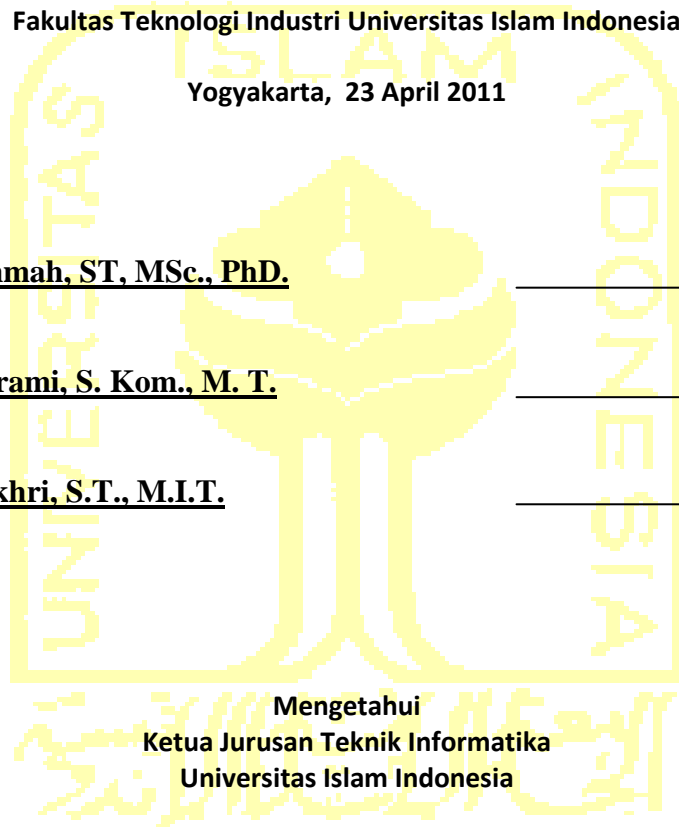
Ketua

**Affan Mahtarami, S. Kom., M. T.**

Anggota I

**Zainudin Zuhri, S.T., M.I.T.**

Anggota II



Mengetahui  
Ketua Jurusan Teknik Informatika  
Universitas Islam Indonesia

**Yudi Prayudi, S.Si., M.Kom.**

**LEMBAR PERNYATAAN KEASLIAN  
HASIL TUGAS AKHIR**

Saya yang bertanda tangan di bawah ini,

Nama : Septian Aditya

NIM : 07523061

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian dari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 23 April 2012

Septian Aditya

## HALAMAN PERSEMBAHAN

*Alhamdulillahillobbil'alamiin,  
Segala puji syukur atas nikmat Allah SWT. Terima kasih atas segala kemudahan  
yang Engkau berikan sehingga hamba dapat menyelesaikan skripsi ini.*

*Ucapan terima kasih sedalam-dalamnya saya ucapkan untuk :*

*Dosen-dosen Teknik Informatika FTI UII, terima kasih untuk semua ilmu  
berharga yang diberikan kepada saya.*

*Ibu Izzy yang selalu sabar membimbing saya dalam skripsi dan memberikan saya  
ilmu yang sangat berguna.*

*Teman-teman INCLUDE, teman seperjuangan, mahasiswa tingkat akhir penuh  
harapan gemilang. 4 tahun yang indah. Sukses selalu teman-teman !!*

*Satyo, Noppa, Ilham, Sistha, Rista, Mentari, Latep, Faiz. Teman bermain dan  
berdiskusi, tanpa adanya kalian kuliah ini tidak pernah terasa sepi.*

*Iguk, teman sekaligus guru bagi saya. Yang selalu memberi saya masukan dan  
solusi. Terima kasih teman.*

*Pada Lycan yang selalu memberi semangat dan perhatian untuk segala hal.*

*Terima kasih ☺*

*Untuk Adik-adikku Novindi Dwi Rahmawati dan Azmi Rahmadani. Terima kasih  
dan tetap semangat kuliahnya. ☺*

*Yang paling utama, saya persembahkan tugas akhir ini untuk Bapak Sulistiyo dan Ibu Inni Hikmatin tercinta, terima kasih atas cinta tak terbatas dan limpahan dukungan yang tak terkira.*

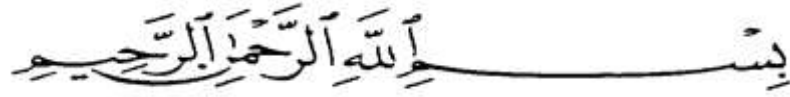
## **MOTTO**

Bersyukurlah dengan apa yang kita miliki. Dan hasil apa yang kita dapatkan adalah buah hasil yang kita kerjakan.

Semua masalah pasti ada jalannya walaupun itu sulit.

Ketika menghadapi suatu masalah sulit, orang sukses akan mencari jalan keluar dan orang malas cenderung menyerah.

## KATA PENGANTAR



*Assalamualaikum Wr. Wb.*

*Puji* syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan tugas akhir ini. Shalawat dan salam semoga selalu tercurah kepada junjungan kita Nabi Muhammad SAW, inspirasi akhlak dan pribadi mulia.

Tugas akhir ini yang berjudul “Aplikasi *Automatic Cropping* Pas Foto” disusun sebagai salah satu syarat untuk mendapatkan gelar sarjana S1 pada jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

Tak lupa pada kesempatan kali ini saya mengucapkan terimakasih kepada berbagai pihak yang telah membantu menyelesaikan tugas akhir ini. Ucapan terimakasih saya sampaikan kepada :

1. Allah S.W.T yang telah memberikan rahmat dan hidayah-Nya.
2. Bapak Gumbolo Hadi Susanto selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Ibu Izzati Muhimmah, ST, MSc., PhD. selaku Dosen Pembimbing. Terima kasih banyak atas bimbingan, dukungan, dan arahan selama pelaksanaan tugas akhir.
4. Bapak Sulistiyo dan Ibu Inni Hikmatin tercinta yang telah memberi semangat dan dukungan baik moril maupun materil.
5. Teman-teman terbaik yang mendukung dan menemaniku selama pengerjaan tugas akhir : Bithana Paragustin, Yogie Aditya, Satyo Agung,



Noppa, Iwan Faried, Ilham Yuli, Sistha Widita, Mentari Ambar, Marista Ernawati.

6. Teman-temanku yang tak mungkin saya sebutkan satu persatu disini, terimakasih banyak atas dukungannya selama ini.

Dalam pelaksanaan dan pembuatan program serta laporan tugas akhir ini saya menyadari bahwa masih banyak kekurangan-kekurangan baik yang disadari maupun yang tidak disadari, oleh karena itu saya sangat mengharapkan kritik dan saran serta masukan dari para pembaca.

Semoga laporan tugas akhir ini memberikan manfaat bagi pembaca maupun bagi kepustakaan ilmu baik pada Jurusan Teknik Informatika Universitas Islam Indonesia maupun bagi dunia ilmu pengetahuan Teknologi Informasi di Indonesia.

*Wassalamualaikum Wr. Wb.*

Yogyakarta, Juni 2012

Septian Aditya

## ABSTRAKSI

Semakin meningkatnya jumlah mahasiswa Universitas Islam Indonesia seiring dengan semakin dikenalnya UII. Dengan semakin banyaknya yang mendaftar masuk, maka dibutuhkan semakin banyak tenaga dan waktu untuk proses pemotretan pas foto dalam registrasi mahasiswa. Oleh karena itu maka dibuatlah sebuah aplikasi yang dapat mendeteksi wajah dan memotong (*cropping*) secara otomatis. Aplikasi tersebut mendeteksi letak wajah dengan cara mendeteksi warna kulit menggunakan deteksi tepi dengan metode Sobel dan model warna HSV. Tujuan dari aplikasi ini adalah dapat mengurangi penggunaan waktu dan tenaga untuk proses pemotongan pas foto registrasi mahasiswa baru Universitas Islam Indonesia. Dengan demikian maka waktu dan tenaga bisa digunakan untuk mengerjakan hal lain. Setelah Aplikasi dibangun dan diuji, hasil yang didapatkan bahwa aplikasi dapat membantu pemotongan dengan waktu yang lebih cepat.

Kata Kunci : *automatic, cropping, deteksi*

## TAKARIR

<i>Automatic Cropping</i>	Teknik memotong secara otomatis.
<i>BufferedImage</i>	Sebuah tipe suatu gambar dimana nilai pixel gambar tersebut dapat diubah.
<i>Cracking Tool</i>	
<i>Cropping</i>	Teknik memotong sebuah gambar.
<i>Database</i>	Basis data.
<i>Edge Detection</i>	Pendeteksian tepi gambar.
<i>File</i>	Data.
<i>Flowchart</i>	Diagram alir.
<i>Grayscale</i>	Model warna gambar dimana warna gambar hanya hitam, putih dan abu-abu.
<i>Histogram</i>	Tampilan grafis dari tabulasi frekuensi yang digambarkan dengan grafis batangan.
<i>HSV</i>	Model warna yang digunakan untuk mencari warna kulit.
<i>Input</i>	Masukkan.
<i>Interface</i>	Keluaran.
<i>Internet</i>	Sistem global dari seluruh jaringan komputer yang saling terhubung untuk melayani miliaran pengguna di seluruh dunia.
<i>Java</i>	Sebuah bahasa pemrograman.
<i>Make up</i>	Riasan.
<i>Netbeans</i>	Text editor untuk membuat suatu aplikasi.
<i>Output</i>	Keluaran.
<i>Password</i>	Kata sandi.
<i>Pixel</i>	Unsur gambar atau representasi sebuah titik terkecil dalam sebuah gambar grafis yang dihitung per inci.
<i>Prototype</i>	Model konsep atau contoh awal untuk kebutuhan

	pengujian.
<i>Reduce Noise</i>	Pengurangan derau.
<i>RGB</i>	Model warna yang digunakan untuk menghitung nilai HSV.
<i>Resize</i>	Pengubahan ukuran.
<i>Sobel</i>	Salah satu metode pengedeksian tepi.
<i>Software</i>	Perangkat lunak.
<i>Text Editor</i>	Editor yang digunakan untuk menampung suatu bahasa pemrograman tertentu.
<i>User</i>	Pengguna.
<i>Website</i>	Situs.

## DAFTAR ISI

HALAMAN JUDUL .....	<b>Error! Bookmark not defined.</b>
LEMBAR PENGESAHAN PEMBIMBING .....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR.....	<b>Error! Bookmark not defined.</b>
HALAMAN PERSEMBAHAN .....	<b>Error! Bookmark not defined.</b>
MOTTO .....	vii
KATA PENGANTAR .....	viii
ABSTRAKSI .....	x
TAKARIR.....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL .....	xvi
BAB I PENDAHULUAN.....	<b>Error! Bookmark not defined.</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian .....	2
1.5 Manfaat Penelitian .....	2
1.6 Metodologi Penelitian .....	2
1.6.1 Pengumpulan Data.....	3
1.6.2 Pengembangan Sistem.....	3
1.7 Sistematika Penulisan .....	4
BAB II LANDASAN TEORI.....	6
2.1 Pengolahan Citra .....	6
2.2 Biometrika.....	7
2.2.1 Biometrika Wajah.....	8
2.3 Sistem Warna RGB ( <i>Red Green Blue</i> ).....	9
2.4 Sistem Warna HSV ( <i>Hue Saturation Value</i> ).....	10
2.5 Pendeteksian Tepi .....	13
2.5.1 Sobel Operator.....	13

2.6	Pengenalan Piksel Warna Kulit.....	14
<b>BAB III ANALISIS KEBUTUHAN &amp; PERANCANGAN PERANGKAT LUNAK .....</b>		
3.1	Analisis Kebutuhan Sistem .....	17
3.1.1	Metode Analisis .....	17
3.1.2	Hasil Analisis.....	17
3.2	Perancangan Perangkat Lunak .....	18
3.2.1	Metode Perancangan.....	18
3.2.2	Hasil Perancangan .....	18
3.2.3	Perancangan Antar Muka .....	27
3.3	Analisis Pengujian Perangkat Lunak .....	28
3.3.1	Pengujian Pendeteksian Warna Kulit .....	28
3.3.2	Pengujian Aplikasi.....	28
3.3.3	Penanganan Kesalahan .....	29
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>		
4.1	Perangkat Keras dan Lunak yang Digunakan .....	30
4.1.1	Perangkat Keras .....	30
4.1.2	Perangkat Lunak .....	30
4.2	Implementasi Perangkat Lunak .....	30
4.2.1	Implementasi Proses .....	30
4.2.2	Implementasi Antarmuka Sistem.....	38
4.3	Analisis Kinerja Perangkat Lunak .....	42
4.3.1	Pengujian Pendeteksian Warna Kulit .....	42
4.3.2	Pengujian Aplikasi.....	49
4.3.3	Penanganan Kesalahan Masukkan Sistem.....	50
4.4	Kelebihan Aplikasi.....	52
4.5	Kekurangan Aplikasi.....	52
<b>BAB V KESIMPULAN DAN SARAN .....</b>		
5.1	Kesimpulan .....	53
5.2	Saran.....	53
<b>DAFTAR PUSTAKA</b>		
<b>LAMPIRAN</b>		

## DAFTAR GAMBAR

Gambar 2.1 Model Warna RGB .....	10
Gambar 2.2 Model Warna HSV .....	12
Gambar 2.3 Transformasi Gambar dari RGB ke HSV .....	13
Gambar 2.4 Warna Kulit Putih .....	16
Gambar 2.5 Warna Kulit Hitam.....	16
Gambar 3.1 <i>Flowchart</i> Aplikasi <i>Automatic Cropping</i> Pas Foto.....	22
Gambar 3.2 <i>Flowchart</i> Proses <i>Resize</i> Aplikasi <i>Automatic Cropping</i> Pas Foto..	23
Gambar 3.3 <i>Flowchart</i> Proses Deteksi Tepi .....	24
Gambar 3.4 <i>Flowchart</i> Proses Deteksi kulit.....	26
Gambar 3.5 <i>Flowchart</i> Proses <i>Cropping</i> .....	27
Gambar 3.6 Rancangan Antarmuka Aplikasi .....	28
Gambar 4.1 Tampilan Awal Aplikasi .....	39
Gambar 4.2 Tampilan Banyaknya Masukkan Gambar.....	39
Gambar 4.3 Tampilan Pilih Satu Gambar.....	40
Gambar 4.4 Tampilan Pilih Lebih Dari Satu Gambar .....	40
Gambar 4.5 Tampilan Tampilan Tampil Gambar .....	41
Gambar 4.6 Tampilan Pilih Direktori Tujuan.....	41
Gambar 4.7 Tampilan Hasil Pemotongan ( <i>Cropping</i> ).....	42
Gambar 4.8 Penanganan Kesalahan File Gambar. ....	50
Gambar 4.9 Penanganan Kesalahan Direktori Tujuan.....	51
Gambar 4.10 Penanganan Kesalahan Masukkan Batas Atas dan Batas Bawah yang Masih Kosong .....	51
Gambar 4.11 Penanganan Kesalahan Masukkan Batas Atas dan Batas Bawah Selain Angka .....	52

**DAFTAR TABEL**

Tabel 4.1 Tabel Hasil Deteksi Kulit .....	48
Tabel 4.1 Tabel Hasil Pengujian Aplikasi .....	49



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dengan semakin terkenalnya Universitas Islam Indonesia (UII) sebagai salah satu universitas di Indonesia, semakin banyak pula yang mendaftarkan diri untuk menuntut ilmu di Universitas Islam Indonesia. Calon mahasiswa yang mendaftar di Universitas Islam Indonesia harus melewati berbagai proses untuk dapat masuk. Tahapan-tahapan yang harus dilewati antara lain adalah pengambilan foto untuk melengkapi identitas diri calon mahasiswa. Pengambilan foto tersebut dilakukan oleh Badan Sistem Informasi (BSI) Universitas Islam Indonesia. Setelah pengambilan foto, selanjutnya foto akan dilakukan proses pemotongan (*cropping*) secara manual agar ukuran foto menjadi berukuran 3x4. Semakin bertambahnya pendaftar setiap tahunnya, tentu akan menambah waktu dan tenaga juga dalam proses pemotongan foto.

Untuk dapat menghemat waktu dan tenaga, maka dibuatlah sebuah aplikasi yang dapat menggantikan proses pemotongan tersebut. Aplikasi ini dapat melakukan proses pemotongan foto secara otomatis dengan ukuran 3x4. Dengan pendeteksian wajah melalui warna kulit, aplikasi ini memungkinkan menempatkan wajah tetap pada posisi tengah ketika proses pemotongan dilakukan.

Badan Sistem Informasi (BSI) adalah sebuah badan di bawah Universitas Islam Indonesia yang membidangi sistem informasi di Universitas Islam Indonesia. BSI bertugas mengembangkan teknologi, baik dalam bentuk perangkat keras maupun perangkat lunak. Sebagai contoh perangkat keras, BSI bertugas menyediakan akses internet di seluruh wilayah kampus UII berupa *hotspot*. Untuk perangkat lunak, BSI mengembangkkan sendiri sistem informasi akademik yang dinamakan Unisys dan juga seluruh *website-website* UII. (<http://bsi.uui.ac.id>, 2012)

## 1.2 Rumusan Masalah

Bagaimana membuat aplikasi *automatic cropping* pas foto.

## 1.3 Batasan Masalah

Untuk memfokuskan penelitian, maka dibuatlah batasan masalah pada penelitian yaitu sebagai berikut :

- a. Foto hasil pemotongan (*cropping*) foto hanya berukuran 3x4.
- b. Pencahayaan dari obyek foto harus terang dan jelas.
- c. Lokasi wajah diperoleh dari pendeteksian piksel warna kulit.
- d. Lokasi pendeteksian warna kulit adalah 2/3 atas foto.
- e. Pendeteksian kulit tidak dapat mendeteksi kulit dengan *make up* tebal dan terkena cahaya terlalu terang.

## 1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk membangun aplikasi yang dapat secara otomatis memotong ukuran sebuah foto menjadi ukuran 3x4 secara dinamis.

## 1.5 Manfaat Penelitian

Penelitian ini memiliki manfaat, yaitu :

- a. Dapat menghemat efisiensi waktu dan tenaga untuk pemotongan pas foto yang biasanya dilakukan secara manual.
- b. Sebagai media pembelajaran dan penerapan terhadap ilmu pencitraan.
- c. Sebagai *prototype* pembuatan aplikasi yang lebih bermanfaat dalam pendeteksian wajah.

## 1.6 Metodologi Penelitian

Metode penelitian adalah suatu langkah yang digunakan untuk menyelesaikan masalah dalam penelitian tersebut.

### 1.6.1 Pengumpulan Data

Pada tahap ini, peneliti akan mengumpulkan data yang akan digunakan untuk penelitian. Metode pengumpulan data yang digunakan oleh peneliti adalah sebagai berikut :

a. Studi Pustaka

Metode studi pustaka meliputi landasan teori pengolahan citra, pendeteksian tepi, sistem warna RGB (*Red Green Blue*) dan HSV (*Hue Saturation Value*) dan pendeteksian warna kulit.

b. Referensi *Internet*

Referensi *internet* digunakan untuk pengayaan referensi penelitian yang dilakukan oleh penulis.

### 1.6.2 Pengembangan Sistem

Tahapan selanjutnya yang dilakukan setelah pengumpulan data adalah pengembangan sistem. Tahapan pengembangan adalah sebagai berikut :

a. Analisis Data

Mengumpulkan data tentang pendeteksian wajah dan bahasa pemrograman Java dengan menggunakan aplikasi *Netbeans 7.0*.

b. Desain

Membuat *flowchart* untuk memudahkan perancangan sistem dan tampilan antarmuka (*interface*).

c. Pengkodean

Membuat aplikasi yang telah dirancang dengan menggunakan aplikasi *Netbeans 7.0*.

d. Pengujian

Pengujian yang dilakukan terhadap aplikasi yang dibangun meliputi kemampuan sistem untuk mendeteksi warna kulit secara otomatis dan kecepatan pemotongan foto.

## 1.7 Sistematika Penulisan

Dalam penyusunan laporan tugas akhir ini, sistematika dibagi menjadi beberapa bagian sebagai berikut :

### BAB I PENDAHULUAN

Membahas tentang Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, Metodologi Penelitian dan Sistematika Penulisan.

### BAB II LANDASAN TEORI

Membahas teori-teori yang mendukung dalam perancangan aplikasi *automatic cropping* pas foto. Teori tersebut meliputi tentang definisi pengolahan citra, deteksi tepi menggunakan *Sobel* operator, koordinat warna RGB (*Red Green Blue*), HSV (*Hue Saturation Value*), teori deteksi warna kulit.

### BAB III ANALISIS KEBUTUHAN & PERANCANGAN PERANGKAT LUNAK

Bab ini memuat tentang analisis sistem, analisis kebutuhan *input*, perancangan antarmuka, proses, *output*. Tahapan perancangan sistem terdiri dari perancangan *flowchart* dan antarmuka (*interface*).

### BAB IV HASIL DAN PEMBAHASAN

Memaparkan tentang hasil dan pembahasan aplikasi yang dirancang meliputi *input*, proses, *output*, dan pengujian aplikasi. Analisis kinerja aplikasi akan dievaluasi berdasarkan pengujian program.

### BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari analisis kinerja pada bagian sebelumnya. Saran-saran yang perlu diperhatikan berdasarkan keterbatasan-

keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama pengembangan aplikasi.

## **BAB II**

### **LANDASAN TEORI**

Pada bab ini akan diuraikan beberapa landasar teori yang terbagi dalam enam sub-bab yang membahas mengenai pengolahan citra, biometrika, sistem warna RGB (*Red Green Blue*), sistem ruang warna HSV (*Hue Saturation Value*), pendeteksian tepi (*edge detection*), dan pengenalan piksel warna kulit.

#### **2.1 Pengolahan Citra**

Mata adalah indera terbaik yang dimiliki manusia sehingga citra (gambar) memegang peranan penting dalam perspektif manusia. Namun mata manusia memiliki keterbatasan dalam menangkap sinyal elektromagnetik. Komputer dan mesin pencitraan lainnya dapat menangkap hampir seluruh sinyal elektromagnetik mulai dari gamma hingga gelombang radio. Mesin pencitraan dapat bekerja dari sumber yang tidak sesuai, tidak cocok, atau tidak dapat ditangkap oleh penglihatan manusia. Hal inilah yang menyebabkan pengolahan citra digital memiliki kegunaan dan spektrum aplikasi yang sangat luas. Teknologi pengolahan citra dapat masuk ke berbagai bidang seperti kedokteran, industri, pertanian, geologi, kelautan, dan lain sebagainya. Kehadiran teknologi pengolahan citra memberikan kemajuan yang sangat luar biasa pada bidang-bidang tersebut. Ke depan penerapan teknologi pengolahan citra digital ini akan terus meluas dan hal ini merupakan tantangan tersendiri bagi para penekun dan peneliti di bidang ini. Kategori pengolahan citra dapat dibagi ke dalam tiga kategori yakni kategori rendah, menengah, dan tinggi (Darma Putra, 2010).

Kategori rendah melibatkan operasi-operasi sederhana seperti prapengolahan citra untuk mengurangi derau, pengaturan kontras, dan pengaturan ketajaman citra. Pengolahan citra kategori rendah ini memiliki *input* dan *output* berupa citra (Darma Putra, 2010).

Pengolahan kategori menengah melibatkan operasi-operasi seperti segmentasi dan klarifikasi citra. Proses pengolahan citra menengah melibatkan

*input* berupa citra dan *output* berupa atribut (fitur) citra yang dipisahkan dari citra *input*. Pengolahan citra kategori tinggi melibatkan proses pengenalan dan deskripsi citra (Darma Putra, 2010).

Teknologi pengamanan suatu sistem mengalami kemajuan pesat akibat dari pesatnya perkembangan pengolahan citra pada bidang biometrika. Sebagai contoh pemanfaatan sidik jari, iris, wajah, dan biometrika yang lainnya untuk sistem identifikasi seseorang (Darma Putra, 2010).

## 2.2 Biometrika

Menurut Eko Nugroho, di tengah kemajuan jaringan komunikasi dan mobilitas alat, dibutuhkan sebuah metode yang handal untuk mengidentifikasi seseorang. Dalam perkembangan sistem identifikasi sampai dengan awal tahun 2000, terdapat 2 metode yang dipakai, yaitu :

1. Sistem identifikasi berdasarkan kepemilikan (*possession based* atau “*what you have*”). Proses keamanan didasarkan pada suatu benda yang dimiliki seseorang, misal kartu kredit. Kelemahan pada sistem ini adalah apabila alat identifikasi tersebut hilang, maka orang lain yang menemukannya dapat menyalahgunakan fungsi alat tersebut untuk kepentingan pribadi.
2. Sistem pindentifikasi berdasarkan pengetahuan (*knowledge* atau “*what you know*”). Biasanya, sistem identifikasi jenis ini menggunakan *password* sebagai media utama. Kelemahan sistem ini adalah apabila *password* terlalu pendek, hal ini beresiko untuk dapat ditebak oleh orang lain, dan bila terlalu panjang akan mudah terlupa. Menurut ahli keamanan, saat ini sudah ada *cracking tool* yang mampu memindai kata maupun menebak *password* berupa kombinasi huruf dan angka. Sebagai contoh, “LockCraft”, salah satu program penjebol sandi. Dengan program ini, hanya dibutuhkan waktu sekitar 48 jam untuk dapat menemukan seluruh arsip *password* dalam suatu perusahaan.

Untuk mengatasi masalah tersebut, para ahli mencari cara lain untuk lebih aman. Cara yang kini dikembangkan adalah dengan menggunakan biometrika, yaitu suatu keadaan fisik tertentu maupun perilaku tertentu unik yang ada pada

seseorang. Teknologi biometrika merupakan suatu teknologi baru yang memiliki fungsi utama untuk mengenali manusia melalui sidik jari, mata, wajah, atau bagian tubuh yang lain. Biometrika berasal dari kata bios, yang berarti kehidupan, dan metron, yang berarti ukuran. Biometrika merupakan teknologi untuk mengenali seseorang secara unik. Dengan didukung oleh faktor biaya penggunaan teknologi yang semakin terjangkau dan fleksibilitas teknologi ini, teknologi ini dirasa akan dapat menggusur penggunaan kata sandi (*password*) ataupun kartu (misal kartu kredit) sebagai alat otentifikasi maupun identifikasi. Teknologi identifikasi ini juga disebut dengan teknologi “*what you are*”. Keunggulan sistem biometrika ini adalah sebagai berikut :

1. Biometrika tidak dapat hilang (fisik) atau terlupa (prilaku), kecuali karena faktor trauma.
2. Biometrika sulit untuk ditiru ataupun dipindah tangankan ke pihak lain.
3. Biometrika mengharuskan orang yang bersangkutan untuk ada di tempat identifikasi dilakukan.

### **2.2.1 Biometrika Wajah**

Menurut Eko Nugroho, wajah merupakan kontur kulit manusia yang terdapat pada bagian kepala depan. Pada dasarnya, wajah manusia tidak ada yang sama persis, sekalipun pada anak kembar, pasti ada perbedaannya. Seiring kemajuan teknologi terapan yang dikembangkan adalah bahwa komputer dapat mengenali pemiliknya, sehingga bila sebuah komputer dibuka, komputer tersebut dapat mengenali pengguna komputer tersebut. Tentu saja apabila orang yang membuka akses bukan pemiliknya, maka komputer tersebut tidak akan membuka akses. Ada banyak aspek dari wajah yang dapat dipergunakan sebagai alat pembeda wajah seseorang dengan yang lainnya, antara lain :

1. Bentuk Geometri Wajah (*Face Morphometrics* )

Dengan identifikasi melalui geometri wajah, maka akan ditetapkan fitur-fitur wajah yang akan dipergunakan sebagai patokan pengukuran seperti mata, hidung, mulut, dan lain-lain. dengan mengukur jarak antar fitur



tersebut, akan ditetapkan karakteristik yang unik dari ukuran geometrik seseorang.

2. Pola Kulit Wajah (*Face Skin Surface*)

Dengan metode idenfikasi melalui kulit wajah, kontur kulit seseorang dipergunakan sebagai alat identifikasi seseorang.

3. Temperature pada Wajah (*Face Thermogram*)

Karena wajah setiap orang memiliki suhu tertentu yang diakibatkan karena panas tubuh, maka pola temperatur pada wajah dapat dipergunakan sebagai alat identifikasi seseorang.

Keunggulan sistem biometrika wajah adalah sebagai berikut :

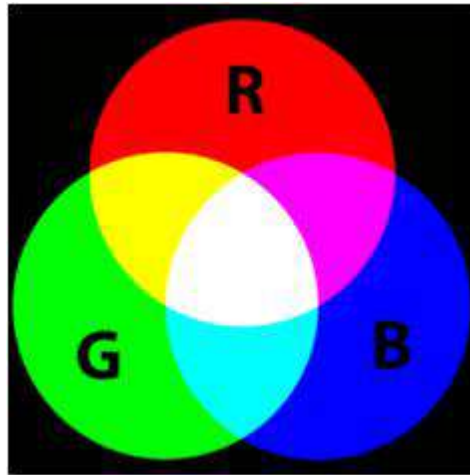
1. Sulit dipalsukan oleh orang yang ingin menerobos sistem keamanan dengan biometrika wajah tersebut.
2. Sistem ini mampu mengidentifikasi umur orang yang diidentifikasi.
3. Sistem ini mampu mengidentifikasi jenis kelamin.
4. Sistem ini mampu sekaligus melihat wajah orang yang diidentifikasi.
5. Biaya peralatan relatif murah.
6. Termasuk jenis yang relatif tidak mengganggu.

Kekurangan dari sistem biometrika wajah adalah sebagai berikut :

1. Wajah orang mudah berubah, baik karena umur maupun karena faktor lain.
2. Diperlukan update *database* secara periodik.
3. Kebanyakan orang kurang suka wajahnya difoto atau direkam.

### 2.3 Sistem Warna RGB (*Red Green Blue*)

Model warna RGB adalah sebuah model warna dimana warna-warna primer merah (*red*), hijau (*green*), dan biru terang (*blue light*) dimasukkan bersama dalam berbagai cara untuk mereproduksi sebuah warna. Nama RGB diambil dari inisial tiga warna merah (*red*), hijau (*green*), dan biru (*blue*) (Darma Putra, 2010). Model warna RGB dapat dilihat pada **Gambar 2.1**.



**Gambar 2.1** Model Warna RGB

Tujuan utama dari model warna RGB adalah untuk penginderaan, penggambaran dan menampilkan gambar pada sistem elektronik, seperti televisi dan komputer (Darma Putra, 2010).

Model warna RGB adalah pencampuran ketiga warna primer untuk membentuk warna final baru. Untuk membentuk warna dengan RGB, tiga warna primer (merah, hijau, biru) harus ditumpangkan. Satu dari ketiga warna tersebut disebut komponen dari warna, dan yang lainnya dapat mempunyai intensitas pada campuran. Intensitas nol untuk setiap komponen memberikan warna yang lebih gelap, sedangkan intensitas penuh memberikan warna terang (Darma Putra, 2010). Sebuah warna pada model warna RGB mendeskripsikan berapa banyak masing-masing dari warna merah, hijau, biru yang terdapat pada warna tersebut. Nilai dari warna-warna tersebut antara nol sampai dengan nilai maksimum tergantung dari aplikasi. Dalam komputasinya, warna-warna komponen terdaftar sebagai integer (bilangan bulat) dengan nilai antara 0 sampai 255 (Darma Putra, 2010).

#### **2.4 Sistem Ruang Warna HSV (*Hue Saturation Value*)**

Model HSV (*Hue Saturation Value*) menunjukkan ruang warna dalam tiga komponen utama yaitu *hue*, *saturation*, dan *value*.

*Hue* menunjukkan jenis warna (seperti merah, biru, atau kuning) atau corak warna yaitu tempat-tempat warna tersebut ditemukan dalam spektrum warna. Nilai *hue*

berupa sudut, sudut 0 sampai 360 derajat. Biasanya sudut 0 adalah merah, 60 derajat adalah kuning, 120 derajat adalah hijau, 180 derajat adalah cyan, 240 derajat adalah biru dan 300 adalah warna magenta (Darma Putra, 2010).

Saturasi (*saturation*) dari suatu warna adalah ukuran seberapa besar kemurnian dari warna tersebut. Sebagai contoh suatu warna yang semuanya merah tanpa putih adalah saturasi penuh. Jika ditambahkan putih ke merah, hasilnya akan berwarna-warni dan warna digeser dari merah ke merah muda (pink). Hue masih tetap merah tapi nilai saturasinya berkurang. Saturasi biasanya bernilai dari 0 sampai 1. 0 menunjukkan abu-abu dan 1 menunjukkan warna primer murni (Darma Putra, 2010).

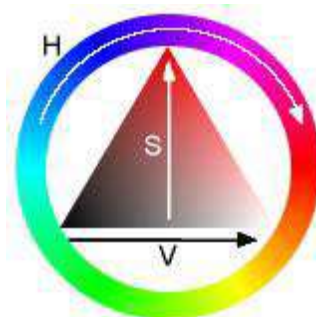
Komponen terakhir dari HSV adalah *value* atau disebut juga intensitas (*intensity*) yaitu ukuran seberapa besar kecerahan dari suatu warna atau seberapa besar cahaya datang dari suatu warna. *Value* dapat bernilai dari 0 sampai 100%. Suatu warna dengan nilai *value* 100% akan terlihat cerah dan sebaliknya, suatu warna dengan nilai *value* 0 akan terlihat gelap. Model warna HSV dapat dilihat dalam **Gambar 2.2**.

Perhitungan konversi RGB menjadi HSV dapat dirumuskan sebagai berikut :

$$H = \tan \left[ \frac{3(G - B)}{(R - G) + (R - B)} \right]$$

$$S = 1 - \frac{\min(R, G, B)}{V} \quad \dots\dots\dots(2.1)$$

$$V = \frac{R + G + B}{3}$$



**Gambar 2.2** Model Warna HSV

Namun pada rumus diatas , apabila  $S=0$  maka  $H$  tidak dapat ditentukan. untuk itu diperlukan normalisasi RGB terlebih dahulu dengan rumus berikut :

$$\begin{aligned} r &= \frac{R}{R + G + B} \\ g &= \frac{G}{R + G + B} \dots\dots\dots(2.2) \\ b &= \frac{B}{R + G + B} \end{aligned}$$

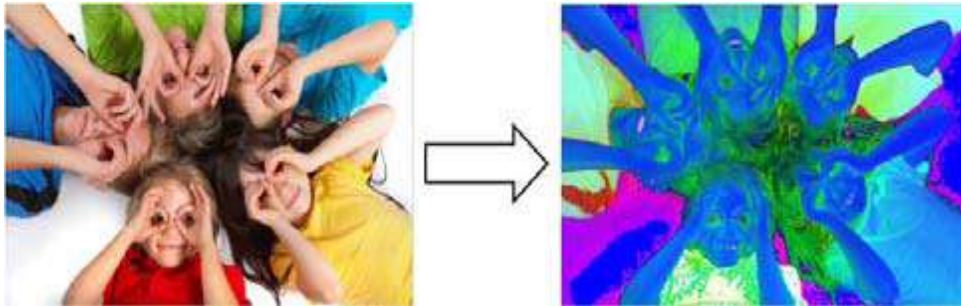
Dengan memanfaatkan nilai  $r, g, b$  yang telah dinormalisasi, maka rumus transformasi RGB ke HSV adalah sebagai berikut :

$$\begin{aligned} V &= \max(r, g, b) \\ S &= \begin{cases} 0 & \text{jika } V = 0 \\ V - \frac{\min(r, g, b)}{V} & \text{jika } V > 0 \end{cases} \dots\dots\dots(2.3) \end{aligned}$$

$$H = \begin{cases} 0 & \text{jika } S = 0 \\ \frac{60 \times (g - b)}{S \times V} & \text{jika } V = r \\ 60 \times \left[ 2 + \frac{(b - r)}{S \times V} \right] & \text{jika } V = g \\ 60 \times \left[ 4 + \frac{(r - g)}{S \times V} \right] & \text{jika } V = b \end{cases} \dots\dots\dots(2.4)$$

$$H = H + 360 \quad \text{jika } H < 0$$

Berikut ini adalah contoh konversi sebuah gambar dari RGB ke HSV :



**Gambar 2.3** Tranformasi Gambar dari RGB ke HSV

## 2.5 Pendeteksian Tepi (*Edege Detection*)

Pendeteksian tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi di dalam citra. Tujuan pendeteksian tepi adalah meningkatkan penampakan garis batas suatu daerah atau di dalam citra (Eko Nugroho, 2005).

Untuk mendeteksi tepi-tepi pada citra ini dapat digunakan metode Sobel, Prewitt, Robert, Laplacian of a Gaussian, Canny, dan lain-lain (Eko Nugroho, 2005).

### 2.5.1 Sobel Operator

Salah satu cara untuk menghindari gradien yang dihitung pada titik interpolasi dari piksel-piksel yang terlibat adalah dengan menggunakan jendela 3x3 untuk perhitungan gradien, sehingga perkiraan gradien berada tepat di tengah jendela (Usman Ahmad, 2005).

Menurut Usman Ahmad, operator Sobel adalah operator yang paling banyak digunakan sebagai pelacak tepi karena kesederhanaan dan keampuhannya. Selain itu Sobel Operator merupakan pengembangan metode Robert dengan menggunakan filter HPF (*High Pass Filter*) yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode sobel adalah kemampuan untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi. Misalkan piksel-piksel disekitar piksel  $p_0 = (x,y)$  seperti dibawah ini :

P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>
P <sub>8</sub>	(x,y)	P <sub>4</sub>
P <sub>7</sub>	P <sub>6</sub>	P <sub>5</sub>

Maka berdasarkan susunan piksel-piksel tetangga tersebut, besaran gradien yang dihitung dengan metode Sobel adalah sebagai berikut :

$$M = \sqrt{s_x^2 + s_y^2} \dots\dots\dots(2.5)$$

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

$$\begin{aligned} s_x &= (p_3 + cp_4 + p_5) - (p_1 + cp_8 + p_7) \\ s_x &= (p_1 + cp_2 + p_3) - (p_7 + cp_6 + p_5) \end{aligned} \dots\dots\dots(2.6)$$

Dengan c suatu konstanta bernilai 2. Di dalam perhitungan gradien pada operator pelacak tepi lainnya,  $s_x$  dan  $s_y$  dapat diimplementasikan menggunakan jendela :

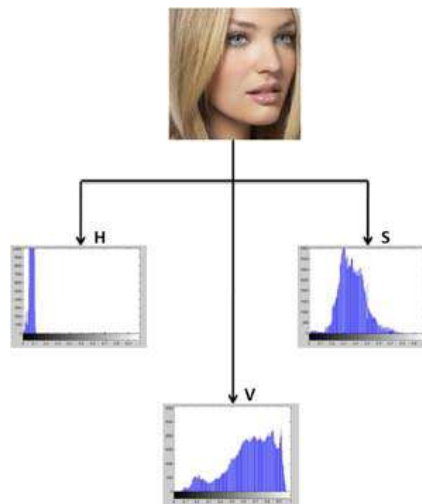
$$s_y = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array} \quad s_x = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \dots\dots\dots(2.7)$$

Jika diperhatikan, metode Sobel menempatkan penekanan atau pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat jendela. Dengan demikian pengaruh piksel-piksel tetangga akan berbeda sesuai dengan letaknya terhadap titik dimana gradien dihitung. Dari susunan nilai-nilai pembobotan pada jendela juga terlihat bahwa perhitungan terhadap gradien merupakan gabungan dari posisi mendatar dan posisi vertikal (Usman Ahmad, 2005).

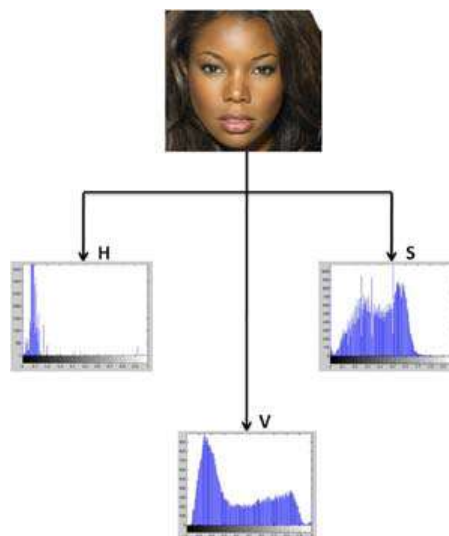
## 2.6 Pengenalan Piksel Warna Kulit

Untuk pengenalan piksel warna kulit, metode yang digunakan pada penelitian ini terdapat pada jurnal *Detection of Pornographic Digital Images* dan

*Human Face Detection in Cluttered Color images Using Skin Color and Edge Information.* Pada jurnal ini disebutkan bahwa HSV (*Hue Saturation Value*) memberikan hasil terbaik dalam pendeteksian piksel warna kulit. Warna kulit dapat dimodelkan menggunakan histogram. Analisa histogram pada model warna HSV dilakukan untuk menentukan nilai ambang pada warna kulit. Histogram ini diperlukan untuk mengidentifikasi nilai rentang dengan tujuan untuk mengelompokkan seseorang atau beberapa orang dalam satu gambar. Gambar warna kulit putih dan hitam beserta histogram HSV dapat dilihat pada gambar berikut.



**Gambar 2.4** Warna Kulit Putih



**Gambar 2.5** Warna Kulit Hitam

Histogram yang diamati sebelumnya menunjukkan nilai-nilai yang pada orang dengan ras yang berbeda. Setelah dilakukan analisa mendalam, maka ditentukan nilai ambang sebagai berikut :

$$0 < H < 0.25$$

$$0.15 < S < 0.9$$

$$0.2 < V < 0.95$$

Dimana nilai H, S dan V dalam jarang dari 0 sampai 1.



## **BAB III**

### **ANALISIS KEBUTUHAN & PERANCANGAN PERANGKAT LUNAK**

#### **3.1 Analisis Kebutuhan Sistem**

Analisis kebutuhan merupakan langkah awal untuk menentukan perangkat lunak yang akan dibuat.

##### **3.1.1 Metode Analisis**

Tahap ini digunakan untuk mengetahui semua kebutuhan dalam pengembangan perangkat lunak yang akan dibuat.

##### **3.1.2 Hasil Analisis**

###### **a. Analisis Kebutuhan Masukan Sistem**

Berdasarkan analisis yang dilakukan penulis, masukan yang dibutuhkan sistem adalah file foto yang akan diproses, URL (*Uniform Resource Locator*) tujuan dimana hasil foto yang telah diproses, masukan batas atas dan bawah foto untuk dipotong. Batas atas dan bawah merupakan persentase dari kotak deteksi wajah yang akan tampil.

###### **b. Analisis Kebutuhan Proses**

Berdasarkan analisis yang dilakukan penulis, proses-proses yang dilakukan adalah sebagai berikut :

- 1) Proses pengambilan file yang telah user masukkan dan mengubah tipe dari file menjadi *BufferedImage*.
- 2) Proses mengubah ukuran foto menjadi lebih kecil.
- 3) Proses pendeteksian wajah, meliputi :
  - a) Proses pendeteksian tepi dengan metode Sobel.
  - b) Proses pengambilan nilai RGB (*Red Green Blue*) dari foto.

- c) Proses perhitungan untuk mencari nilai HSV (*Hue Saturation Value*) dari nilai RGB (*Red Green Blue*) yang telah didapat.
  - d) Proses pendeteksian wajah dari nilai HSV (*Hue Saturation Value*) dan Sobel.
  - e) Proses *cropping* foto berdasarkan masukan batas atas dan bawah oleh *user*.
- 4) Proses penyimpanan file yang telah dipotong ke dalam URL (*Uniform Resource Locator*) yang telah dimasukkan oleh *user*.

### c. Analisis Keluaran Sistem

Keluaran yang dihasilkan oleh aplikasi *Automatic Cropping* Pas Foto adalah file foto yang telah berukuran 3x4 berdasarkan masukan batas atas dan bawah oleh *user*.

### d. Kebutuhan Antarmuka

Antarmuka merupakan penghubung antara pengguna dengan sistem. Pada aplikasi *Automatic Cropping* Pas Foto ini antarmuka yang digunakan adalah antarmuka yang sederhana dan dapat menampilkan tampilan foto yang telah dideteksi agar dapat memasukkan batas atas dan bawah.

## 3.2 Perancangan Perangkat Lunak

### 3.2.1 Metode Perancangan

Metode perancangan perangkat lunak untuk aplikasi ini adalah diagram alir (*Flowchart*).

### 3.2.2 Hasil Perancangan

Hasil perancangan sistem untuk aplikasi *Automatic Cropping* Pas Foto ini berupa *Flowchart*. Penjelasan dari hasil rancangan dapat dilihat sebagai berikut :

**a. *Flowchart***

*Flowchart* adalah sebuah diagram visual yang menyajikan informasi mengenai operasi-operasi yang dilakukan sistem dan urutan proses dari inisialisasi (mulai) hingga selesai. Dalam perancangan sistem sistem/perangkat lunak, *flowchart* mendefinisikan suatu proses atau menganalisis apabila terjadi kesalahan. *Flowchart* mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* utama aplikasi *Automatic Cropping Pas Foto* pada perangkat lunak dapat dilihat pada **Gambar 3.1**.

Pada **Gambar 3.1** dijelaskan alur program secara garis besar pada aplikasi *Automatic Cropping Pas Foto*. Proses tersebut dimulai dengan memasukkan file gambar yang akan dipotong. Pengguna akan memilih tipe masukkan file yang telah disediakan, yaitu masukkan satu file gambar atau masukkan lebih dari satu gambar. Setelah file tersebut dimasukkan, akan dilakukan proses *resize* gambar untuk ditampilkan pada kotak gambar. Setelah itu file akan dideteksi tepi menggunakan metode Sobel. Setelah nilai-nilai tepi diperoleh, kemudian gambar akan dideteksi untuk mencari warna kulit pada gambar. Gambar yang telah dideteksi kemudian akan ditampilkan pada kotak gambar agar dapat dilihat oleh pengguna. Pengguna juga diminta untuk memilih file direktori tujuan untuk tempat penyimpanan gambar setelah dipotong nantinya. Setelah itu pengguna dapat memasukkan nilai batas atas dan batas bawah. Setelah memasukkan batas atas dan bawah, pengguna dapat melihat hasil gambar yang telah dipotong pada jendela hasil. Setelah hasil sesuai dengan keinginan pengguna, gambar akan telah dipotong akan di simpan pada direktori tujuan. Untuk masukkan lebih dari satu file, proses diatas akan diulang untuk file berikutnya sampai dengan file terakhir yang dipilih.

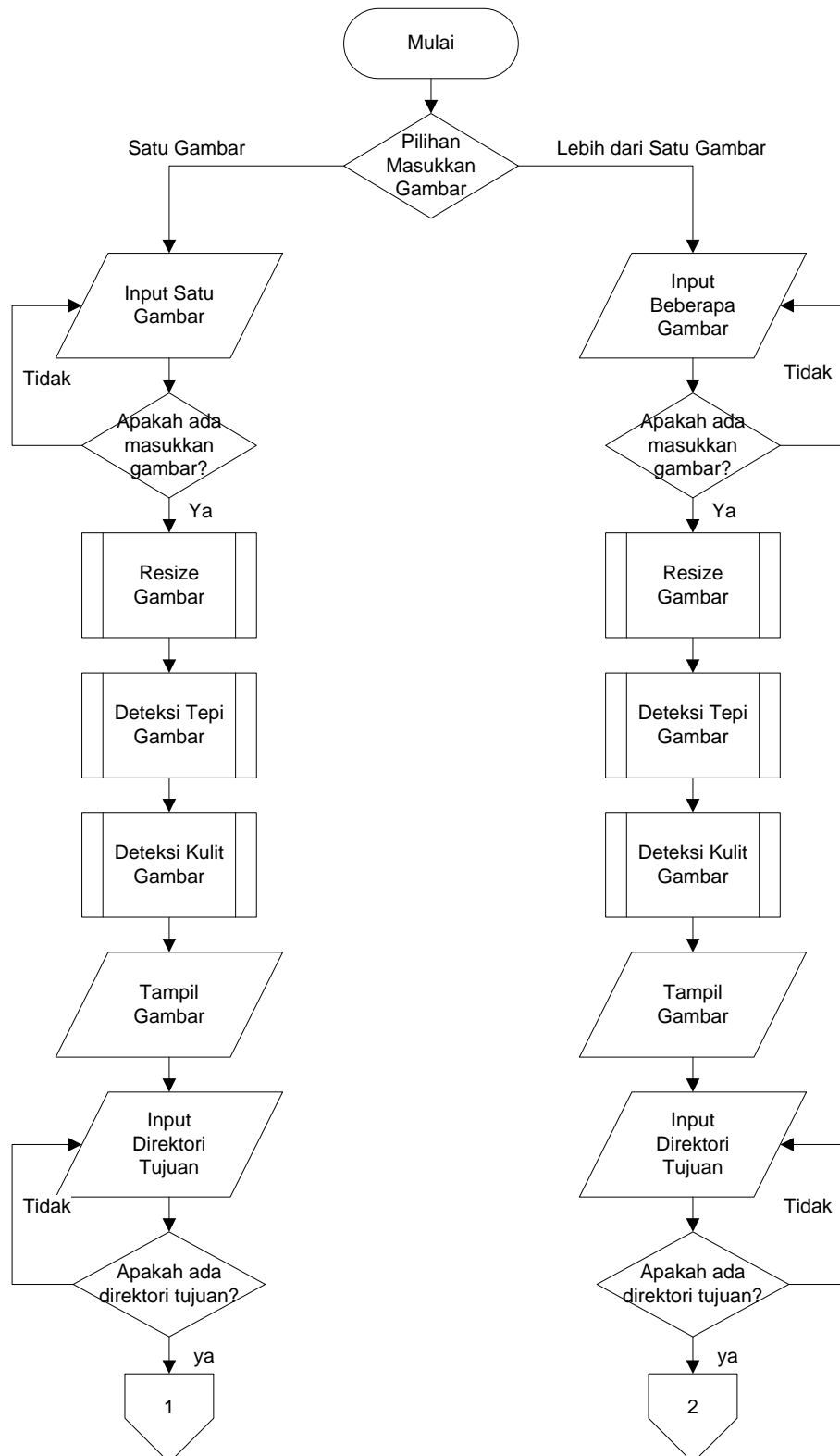
Untuk *flowchart* proses *resize* gambar dapat dilihat pada **Gambar 3.2**. Setelah file gambar dipilih, proses pertama yang akan dijalankan adalah proses *resize* gambar. Proses ini akan melakukan inisialisasi awal nilai tinggi dan lebar gambar. Kemudian akan mencari nilai paling besar dari gambar tersebut. Setelah itu nilai terbesar yang ditemukan akan dirubah menjadi 500 agar dapat muat

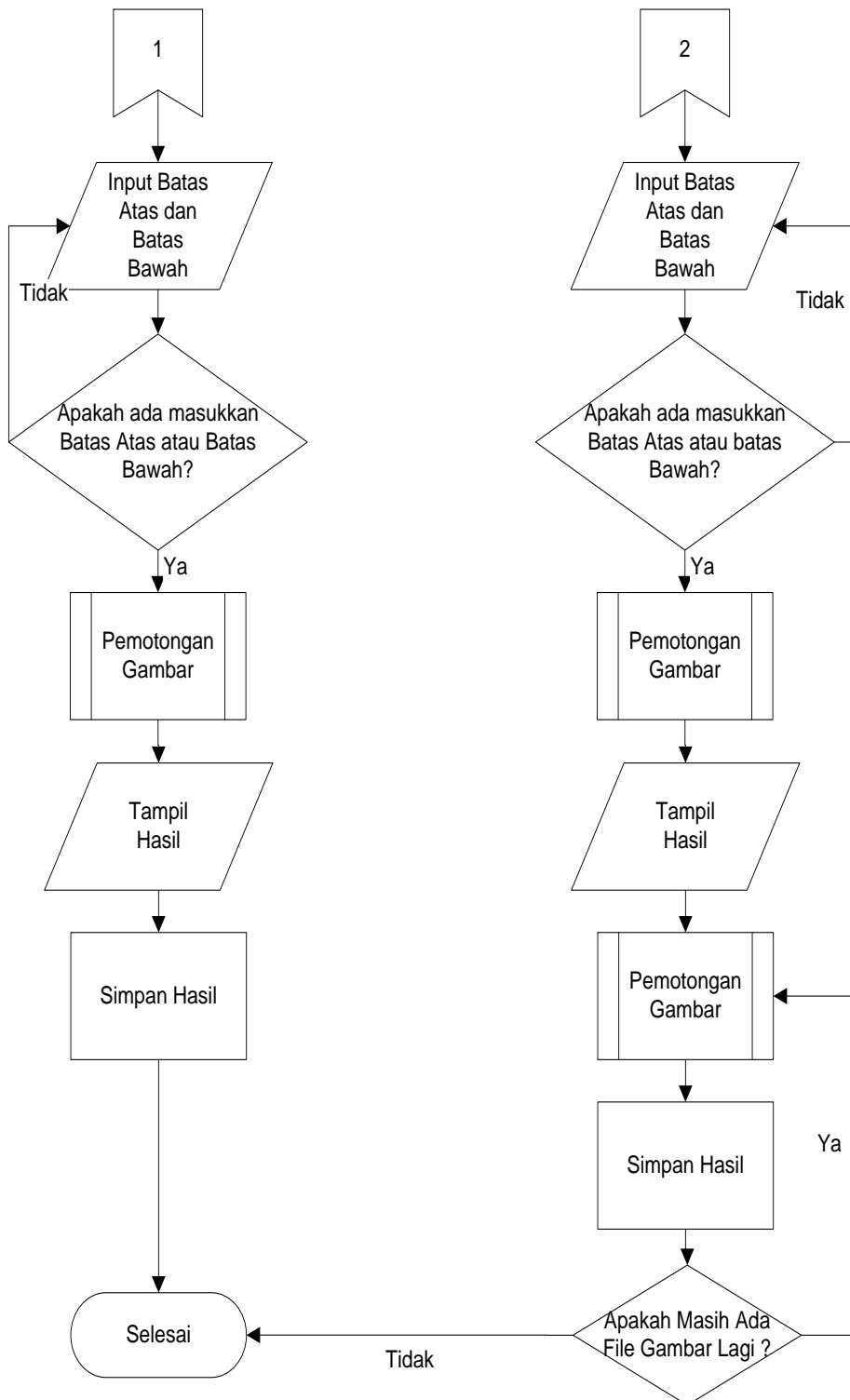
dalam kotak gambar. Untuk nilai lainnya aja dihitung menyesuaikan nilai yang baru.

Proses berikutnya adalah proses pendeteksian tepi. *Flowchart* proses pendeteksian tepi dapat dilihat pada **Gambar 3.3**. Proses pendeteksian tepi dimulai dengan inialisasi nilai tinggi dan lebar gambar. Kemudian inialisasi nilai  $i$  dan  $j$  untuk perulangan yaitu 0. Setelah itu kemudian dilakukan transformasi nilai RGB ke *grayscale*. Setelah proses selesai kemudian nilai-nilai *grayscale* yang telah dihitung digunakan untuk menghitung nilai tepi dengan menggunakan metode Sobel. Setelah nilai-nilai tepi dihitung, maka proses pendeteksian tepi telah selesai.

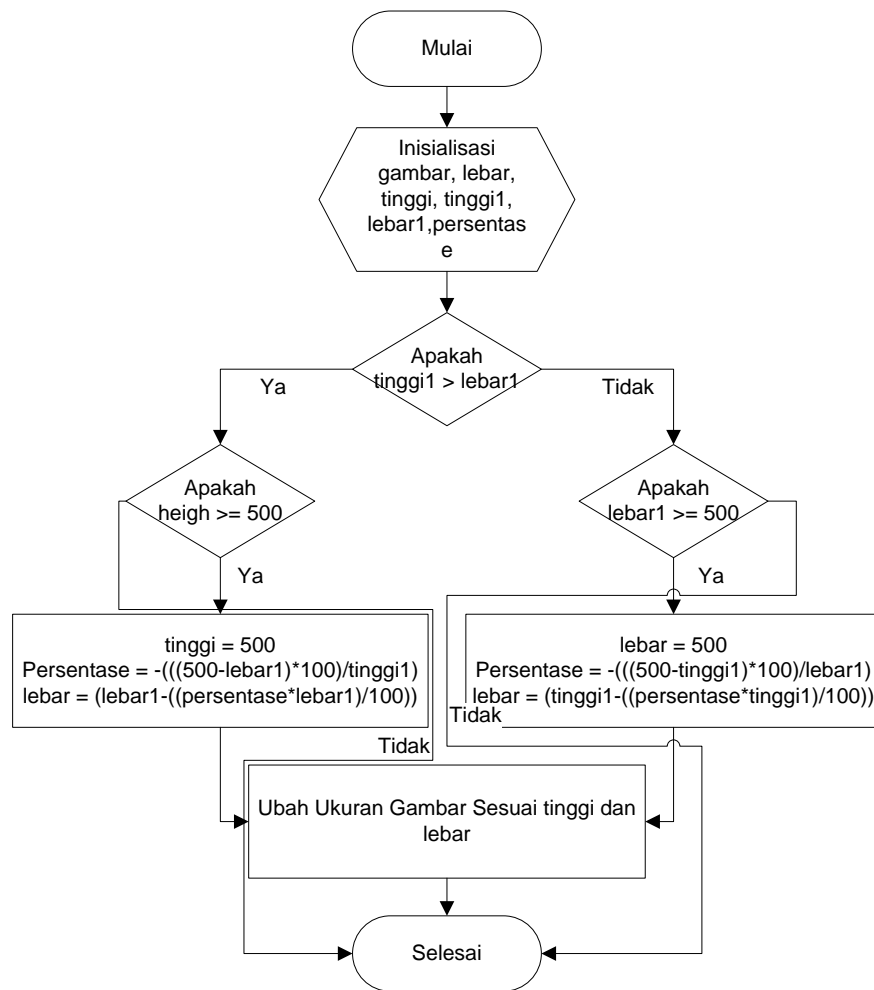
Untuk proses flowchart pendeteksian warna kulit dapat dilihat pada **Gambar 3.4**. Pada proses pendeteksian warna kulit, dimulai dengan inialisasi lebar dan tinggi gambar. Kemudian pengambilan nilai RGB pada setiap piksel gambar yang akan digunakan untuk mencari nilai HSV dari gambar tersebut. Setelah nilai HSV dihitung, kemudian nilai HSV tersebut digunakan untuk mendeteksi warna kulit yang ada pada gambar.

Proses terakhir pada aplikasi *Automatic Cropping* pas foto adalah proses pemotongan (*cropping*) yang dapat dilihat pada **Gambar 3.5**. Pada proses terakhir ini, pertama-tama akan dilakukan inialisasi batas atas, bawah, kanan dan kiri pada gambar yang telah dideteksi dan juga inialisasi masukkan batas atas dan batas bawah dari pengguna. Setelah itu akan dihitung batas atas, bawah, kanan dan kiri sesuai masukkan dari pengguna tadi. Setelah dihitung, gambar akan dipotong sesuai dengan ukuran tersebut. Selanjutnya gambar yang telah dipotong tersebut akan disimpan pada direktori tujuan yang telah dipilih oleh pengguna sebelumnya.

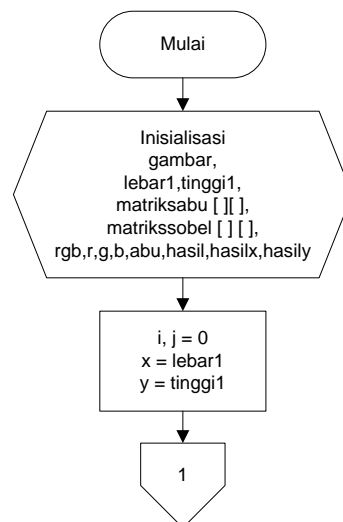


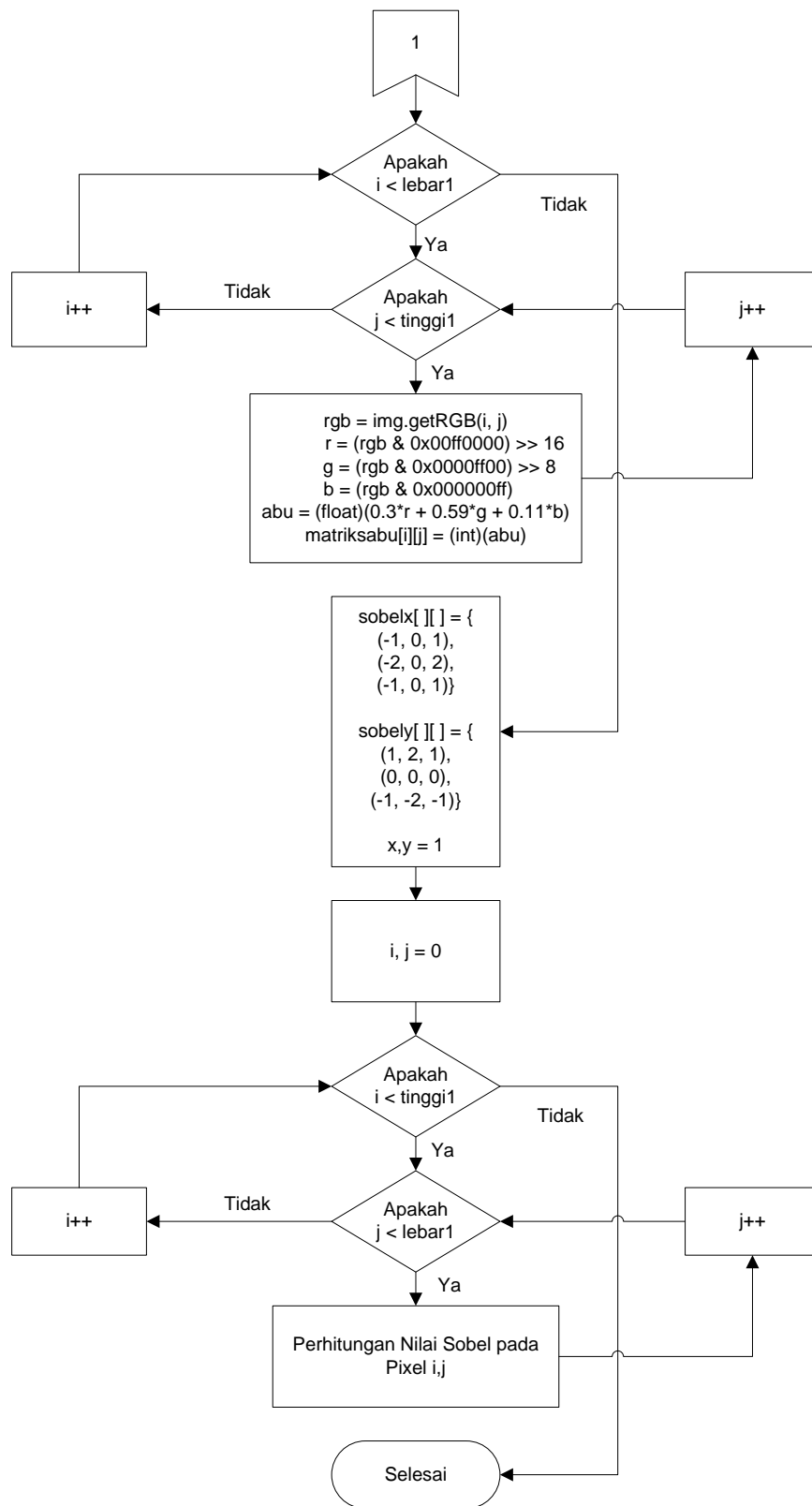


**Gambar 3. 1** Flowchart Utama Aplikasi Automatic Cropping Pas Foto



**Gambar 3. 2** Flowchart Proses Resize Aplikasi Automatic Cropping Pas Foto



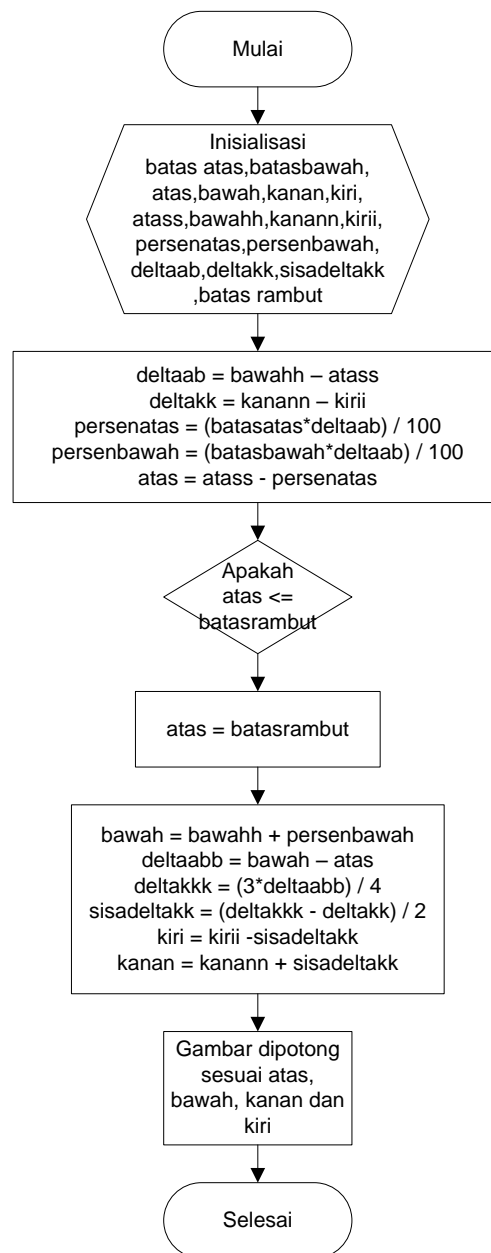


**Gambar 3. 3** Flowchart Proses Deteksi Tepi Aplikasi Automatic Cropping Pas Foto









**Gambar 3. 5** Flowchart Proses *Cropping* Aplikasi *Automatic Cropping* Pas Foto

### 3.2.3 Perancangan Antarmuka

Perancangan antarmuka aplikasi *Automatic Cropping* Pas Foto ini dibuat sederhana agar memudahkan pengguna dalam menggunakan aplikasi ini. Aplikasi ini terdiri dari menu dan kotak gambar. Menu berisi tombol ambil gambar, tombol pilih direktori tujuan, inputan batas atas dan bawah serta tombol *crop*. Tombol gambar untuk mengambil file gambar yang akan dipotong. Tombol pilih direktori tujuan untuk memilih direktori yang nantinya akan digunakan sebagai tempat

menyimpan file yang telah dipotong. Tombol crop untuk melakukan proses pemotongan foto dengan batas atas dan bawah sesuai yang telah dimasukkan oleh pengguna, kemudian secara otomatis menyimpan file hasil di direktori tujuan yang telah dipilih oleh pengguna. Rancangan antarmuka aplikasi *Automatic Cropping Pas Foto* adalah sebagai berikut :

**Gambar 3. 6** Rancangan Antarmuka Aplikasi Automatic Cropping Pas Foto

### 3.3 Analisis Pengujian Perangkat Lunak

#### 3.3.1 Pengujian Pendeteksian Warna Kulit

Pengujian pendeteksian warna kulit untuk mengetahui apakah pendeteksian warna kulit dapat berjalan dengan benar dan mengenali warna-warna kulit sebagaimana yang diharapkan.

#### 3.3.2 Pengujian Aplikasi

Pengujian waktu pemotongan untuk mengetahui apakah pemotongan yang dilakukan oleh aplikasi dapat menghemat waktu seperti yang diharapkan.

### **3.3.3 Penanganan Kesalahan**

Penanganan kesalahan dilakukan untuk mencegah *error* pada aplikasi yang dikarenakan penggunaan *user* yang tidak sesuai dengan aturan-aturan pada aplikasi.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Perangkat Keras dan Lunak yang Digunakan

##### 4.1.1 Perangkat Keras

Perangkat keras yang dibutuhkan untuk membangun aplikasi *Automatic Cropping* Pas Foto ini adalah sebagai berikut :

- 1) Processor Intel i7.
- 2) RAM 4GB.
- 3) VGA Nvidia GeForce Gt540M.

##### 4.1.2 Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam pembuatan aplikasi *Automatic Cropping* Pas Foto ini adalah sebagai berikut :

- 1) *Windows 7*

*Windows* merupakan sistem operasi yang dibuat oleh *Microsoft*. *Windows* sangat universal karena sebagian besar *software-software* yang ada dapat berjalan dan digunakan pada sistem operasi ini.

- 2) *Netbeans 7.0*

*Netbeans* adalah *text editor* untuk membuat sebuah aplikasi dengan bahasa pemrograman. Bahasa pemrograman yang dapat digunakan di dalam *Netbeans* antara lain adalah C, *Java* dan PHP. Aplikasi *Automatic Cropping* pas Foto ini dibuat menggunakan bahasa *Java*.

#### 4.2 Implementasi Perangkat Lunak

##### 4.2.1 Implementasi Proses

Proses-proses yang terdapat dalam sistem ini adalah sebagai berikut :

### a. Proses Pendeteksian Tepi

Proses pendeteksian tepi pada aplikasi *automatic cropping* pas foto ini dilakukan menggunakan metode pendeteksian Sobel. Proses pendeteksian tepi terbagi menjadi dua sub-proses, yaitu :

#### 1. Proses Transformasi RGB (*Red Green Blue*) ke *Grayscale*

Setelah file gambar diinputkan, kemudian nilai RGB (*Red Green Blue*) pada file gambar tersebut akan diambil untuk di rubah menjadi *grayscale*. Nilai *grayscale* tersebut nantinya akan digunakan untuk perhitungan deteksi menggunakan metode Sobel. Kode pemrograman dari proses transformasi ini adalah sebagai berikut :

```
public void rgbtogray() {
    int rgb, r, g, b;

    width = img.getWidth();
    height = img.getHeight();

    matriksgray = new int [width][height];
    matrikssobel = new double [width][height];

    for (int i = 0; i < width; i++) {
        for (int j = 0; j < height; j++) {
            rgb = img.getRGB(i, j);
            r = (rgb & 0x00ff0000) >> 16;
            g = (rgb & 0x0000ff00) >> 8;
            b = (rgb & 0x000000ff);
            float gray = (float)(0.3*r + 0.59*g + 0.11*b);
            matriksgray[i][j] = (int)(gray);
        }
    }
}
```

#### 2. Proses Pendeteksian dengan Sobel

Proses pendeteksian ini, nilai *grayscale* yang telah didapat akan dihitung dengan menggunakan metode Sobel untuk mendapatkan nilai tepi pada gambar tersebut. Kode pemrograman untuk proses ini adalah sebagai berikut :

```
public void sobel() {
int [][] sobelx = {
    {-1, 0, 1},
    {-2, 0, 2},
    {-1, 0, 1}
};
int [][] sobely = {
    {1, 2, 1},
    {0, 0, 0},
    {-1, -2, -1}
};
double hasilx, hasily;
double hasil = 0;
int x = 1, y = 1;

for (int i = 1; i < width-1; i++) {
    for (int j = 1; j < height-1; j++) {
        hasilx = ((matriksgray[i-1][j-1] * sobelx[x-1][y-1])
            + (matriksgray[i-1][j] * sobelx[x-1][y])
            + (matriksgray[i-1][j+1] * sobelx[x-1][y+1])
            + (matriksgray[i][j-1] * sobelx[x][y-1])
            + (matriksgray[i][j] * sobelx[x][y])
            + (matriksgray[i][j+1] * sobelx[x][y+1])
            + (matriksgray[i+1][j-1] * sobelx[x+1][y-1])
            + (matriksgray[i+1][j] * sobelx[x+1][y])
            + (matriksgray[i+1][j+1] * sobelx[x+1][y+1]));

        hasily = ((matriksgray[i-1][j-1] * sobely[x-1][y-1])
            + (matriksgray[i-1][j] * sobely[x-1][y])
            + (matriksgray[i-1][j+1] * sobely[x-1][y+1])
            + (matriksgray[i][j-1] * sobely[x][y-1])
            + (matriksgray[i][j] * sobely[x][y])
            + (matriksgray[i][j+1] * sobely[x][y+1])
            + (matriksgray[i+1][j-1] * sobely[x+1][y-1])
```



```

        + (matriksgray[i+1][j] * sobely[x+1][y])
        + (matriksgray[i+1][j+1] * sobely[x+1][y+1]));

    hasil = hasilx + hasily;
    if (hasil < 0) {
        hasil = -1 * hasil;
    }
    matrikssobel[i][j] = hasil;
}
}
}

```

#### b. Proses Pendeteksian Warna Kulit

Dalam aplikasi *Automatic Cropping Pas Foto*, proses pendeteksian warna kulit terbagi menjadi lima subproses, yaitu :

##### 1. Proses Pengambilan Nilai RGB (*Red Green Blue*) pada Gambar

File gambar yang telah dimasukkan akan diproses untuk mendapatkan nilai RGB (*Red Green Blue*) dari gambar tersebut. Nilai RGB (*Red Green Blue*) nantinya akan digunakan untuk mencari nilai HSV (*Hue Saturation Value*) gambar tersebut. Kode pemrograman dari proses ini adalah sebagai berikut :

```

public void rgb(){
int rgb, red, green, blue;

width = image.getWidth();
height = image.getHeight();

r = new int [width][height];
g = new int [width][height];
b = new int [width][height];

for (int i = 0; i < width; i++) {
    for (int j = 0; j < height; j++) {
        rgb = image.getRGB(i, j);
        red = (rgb & 0x00ff0000) >> 16;
        green = (rgb & 0x0000ff00) >> 8;
    }
}
}

```

```

        blue = (rgb & 0x000000ff);

        r[i][j] = red;
        g[i][j] = green;
        b[i][j] = blue;
    }
}
}

```

## 2. Proses Perhitungan Nilai HSV dari Nilai RGB

Setelah nilai RGB didapatkan, kemudian nilai tersebut digunakan untuk mencari nilai HSV. Pencarian nilai HSV dilakukan menggunakan perhitungan yang telah dibahas dalam lantasan teori. Berikut ini adalah kode pemrograman untuk mencari nilai HSV dari nilai RGB :

```

public void hsv(){
float delta, red, green, blue;
float H = 0, S = 0, V;
float max, min;

h = new float [width][height];
s = new float [width][height];
v = new float [width][height];

for (int i = 0; i < width; i++) {
    for (int j = 0; j < height; j++) {
        delta = r[i][j] + g[i][j] + b[i][j];
        red = r[i][j]/delta;
        green = g[i][j]/delta;
        blue = b[i][j]/delta;

        max = Math.max(red, Math.max(green, blue));
        min = Math.min(red, Math.min(green, blue));

        V = max;
        if (V == 0) {
            S = 0;

```

```

    } else if (V > 0) {
        S = (V-min)/V;
    } else {
    }

    if (S == 0) {
        H = 0;
    } else if (V == red) {
        H = (green - blue)/(S*V);
    } else if (V == green) {
        H = ((blue - red)/(S*V))+2;
    } else if (V == blue) {
        H = ((red - green)/(S*V))+4;
    }

    H *= 60;

    if (H < 0) {
        H += 360;
    } else {
    }

    H = H/360;

    h[i][j] = H;
    s[i][j] = S;
    v[i][j] = V;
    }
}

```

### 3. Proses Pencarian Warna Kulit pada Gambar

Nilai HSV yang telah didapatkan akan digunakan untuk pendeteksian warna kulit pada gambar. Selain nilai HSV, proses ini juga membutuhkan nilai piksel dari proses deteksi tepi menggunakan Sobel operator. Kode pemrograman untuk proses pendeteksian ini adalah sebagai berikut :

```

public void deteksi_kulit(){
statuskulit = new int [width][height];
int ymid = height-(height/3);

for (int i = 0; i < width; i++) {
    for (int j = 0; j < height; j++) {
        if ((h[i][j] > 0.0) && (h[i][j] < 0.25)) {
            if ((s[i][j] > 0.15) && (s[i][j] < 0.9)) {
                if ((v[i][j] > 0.2) && (v[i][j] < 0.95)) {
                    if (matrikssobel[i][j] < 125) {
                        statuskulit[i][j] = 1;
                    } else {
                        statuskulit[i][j] = 0;
                    }
                } else {
                    statuskulit[i][j] = 0;
                }
            } else {
                statuskulit[i][j] = 0;
            }
        } else {
            statuskulit[i][j] = 0;
        }
    }
}

for (int i = 0; i < width; i++) {
    for (int j = ymid; j < height; j++) {
        statuskulit[i][j] = 0;
    }
}
}

```

#### 4. Proses *Reduce Noise*

Proses ini digunakan untuk menyingkirkan *noise* yang ikut terdeteksi pada gambar. Berikut ini adalah kode pemrograman dari proses reduce noise :

```

public void reduce_noise(){
int counter;

for (int i = 1; i < (width-1); i++) {
    for (int j = 1; j < (height-1); j++) {
        counter = 0;

        counter = statuskulit[i-1][j-1]+statuskulit[i-1][j]
            +statuskulit[i-1][j+1]+statuskulit[i][j-1]
            +statuskulit[i][j] +statuskulit[i][j+1]
            +statuskulit[i+1][j-1]+statuskulit[i+1][j]
            +statuskulit[i+1][j+1];

        if (counter >= 5) {
            statuskulit[i][j] = 1;
        } else {
            statuskulit[i][j] = 0;
        }
    }
}
}

```

##### 5. Proses Pencarian Batas Atas, Bawah, Kanan dan Kiri Wajah

Proses ini digunakan untuk mencari koordinat batas-batas dari wajah yang telah terdeteksi. Nantinya nilai-nilai ini akan digunakan untuk membuat kotak deteksi pada gambar dan perhitungan batas foto yang dipotong.

```

public void tepiwajah(){
tepininx = height;
tepinimaxx = 0;
tepiniminy = width;
tepinimaxy = 0;

filter = new int [width][height];

for (int i = 0; i < width; i++) {
    for (int j = 0; j < height; j++) {

```

```
filter[i][j] = 0;
if (statuskulit[i][j] == 1) {
    if (i > tepimaxx) {
        tepimaxx = i;
    } else if (i < tepiminx) {
        tepiminx = i;
    } else if (j > tepimaxy) {
        tepimaxy = j;
    } else if (j < tepiminy) {
        tepiminy = j;
    }
    } else {
    }
}

for (int i = tepiminx; i <= tepimaxx; i++) {
    filter[i][tepiminy] = 1;
    filter[i][tepmaxy] = 1;
}

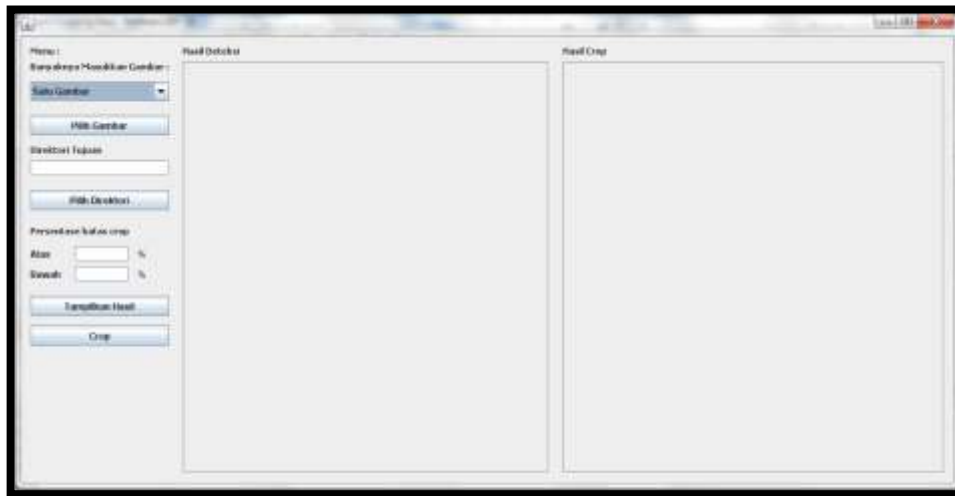
for (int i = tepiminy; i <= tepimaxy; i++) {
    filter[tepiminx][i] = 1;
    filter[tepmaxy][i] = 1;
}
}
```

#### 4.2.2 Implementasi Antarmuka Sistem

##### a. Tampilan Awal Aplikasi

Tampilan awal aplikasi *automatic cropping pas foto* dapat dilihat pada **Gambar**

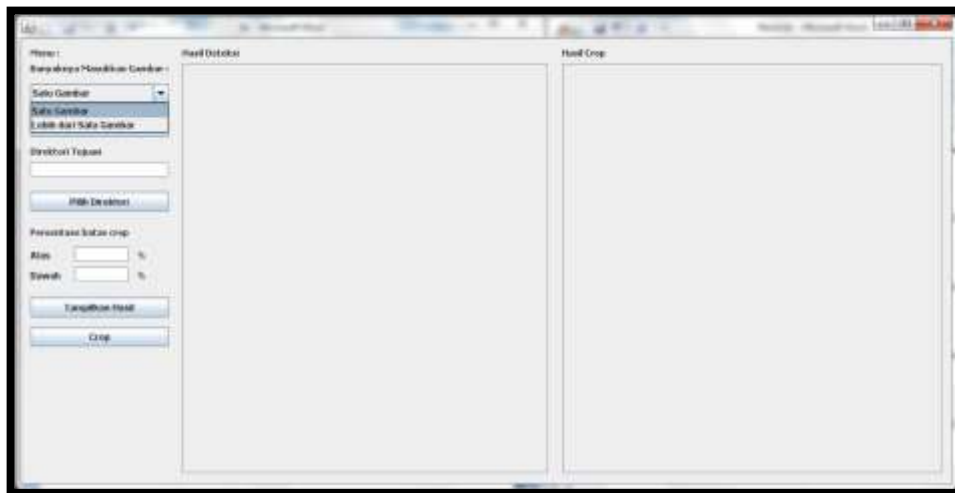
#### 4.1.



**Gambar 4. 1** Tampilan Awal Aplikasi

**b. Tampilan Banyaknya Masukkan Gambar**

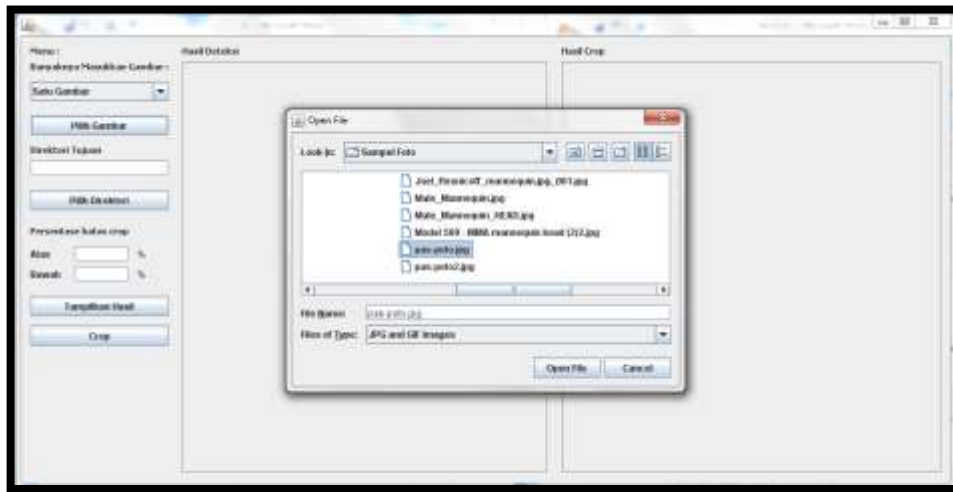
Tampilan banyaknya masukkan gambar dapat dilihat pada **Gambar 4.2**.



**Gambar 4. 2** Tampilan Banyaknya Masukkan Gambar

**c. Tampilan Pilih Satu Gambar**

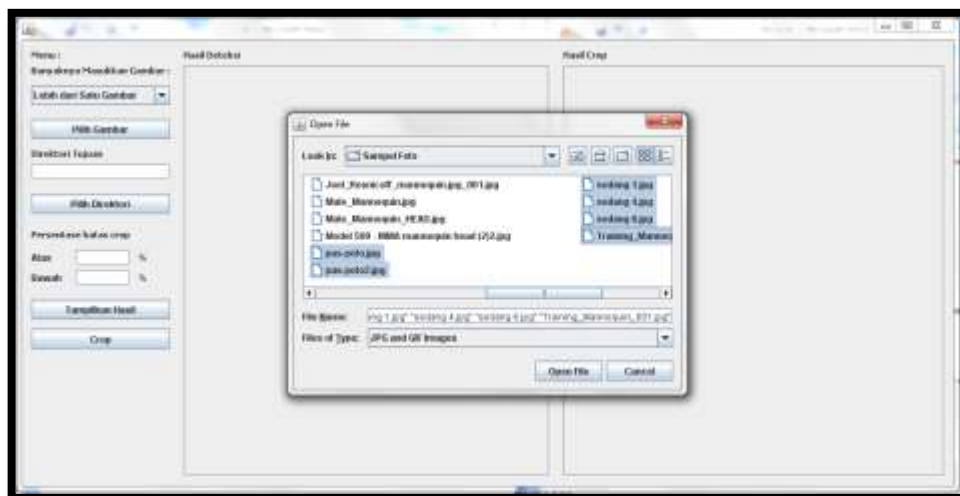
Jika pada banyaknya gambar pengguna memilih satu gambar maka akan tampak tampilan seperti **Gambar 4.3**.



**Gambar 4. 3** Tampilan Pilih Satu Gambar

**d. Tampilan Pilih Lebih dari Satu Gambar**

Jika pada banyaknya gambar pengguna memilih satu gambar maka akan tampak tampilan seperti **Gambar 4.4**.



**Gambar 4. 4** Tampilan Pilih Lebih dari Satu Gambar

**e. Tampilan Tampil Gambar**

Setelah gambar dipilih, gambar akan ditampilkan seperti **Gambar 4.5**.





**Gambar 4. 5** Tampilan Pilih Gambar

**f. Tampilan Pilih Direktori Tujuan**

Tampilan pilih direktori tujuan akan tampak seperti **Gambar 4.6**.



**Gambar 4. 6** Tampilan Pilih Direktori Tujuan

**g. Tampilan Hasil Pemotongan (Cropping)**

Tampilan hasil pemotongan dapat dilihat pada **Gambar 4.7**.





**Gambar 4. 7** Tampilan Hasil Pemotongan (Cropping)

### 4.3 Analisis Kinerja Perangkat Lunak









Pada analisis kinerja perangkat lunak ini aplikasi *automatic* cropping pas foto akan diuji untuk mengetahui kinerja dari aplikasi. Pengujian yang dilakukan adalah sebagai berikut :







#### 4.3.1 Pengujian Pendeteksian Warna Kulit







Pengujian pendeteksian warna kulit ini dilakukan untuk mengetahui apakah pendeteksian warna kulit tersebut dapat berjalan. Selain itu pengujian ini juga dilakukan untuk mengetahui kriteria kulit yang tidak dapat terdeteksi. Pengujian pendeteksian warna kulit dilakukan pada beberapa sampel foto yang dapat dilihat pada tabel dibawah ini :







No	Gambar	Hasil Deteksi Kulit	Keterangan
1			Terdeteksi







2			Terdeteksi
3			Terdeteksi
4			Terdeteksi
5			Terdeteksi

6			Terdeteksi
7			Terdeteksi
8			Terdeteksi
9			Terdeteksi

10			Tidak Terdeteksi
11			Terdeteksi
12			Tidak Terdeteksi

13			Terdeteksi
14			Terdeteksi
15			Terdeteksi

16			Tidak Terdeteksi
17			Terdeteksi
18			Terdeteksi

19			Terdeteksi
20			Terdeteksi
21			Tidak Terdeteksi

Tabel 4.1 Tabel Hasil Deteksi Kulit



Setelah dilakukan pengujian pendeteksian kulit pada tabel sebelumnya, maka didapatkan hasil sebagai berikut :

1. Warna kulit dapat dideteksi.
2. Warna kulit dengan *make up* tebal tidak dapat terdeteksi.

#### 4.3.2 Pengujian Aplikasi

Pengujian aplikasi ini dilakukan untuk mengetahui jumlah maksimal foto dan waktu yang diperlukan untuk proses pemotongan foto tersebut, apakah aplikasi dapat berjalan efisien seperti yang diharapkan. Setelah dilakukan pengujian aplikasi pada jumlah foto dan waktu pemotongan, didapatkan hasil sebagai berikut :

**Tabel 4.1** Tabel Hasil Deteksi Kulit

Jumlah Foto	Waktu Proses
1	4 detik
5	6 detik
10	11 detik
50	52 detik
100	1 menit 44 detik
500	8 menit 44 detik
1000	17 menit 32 detik
5000	1 jam 33 menit 48 detik

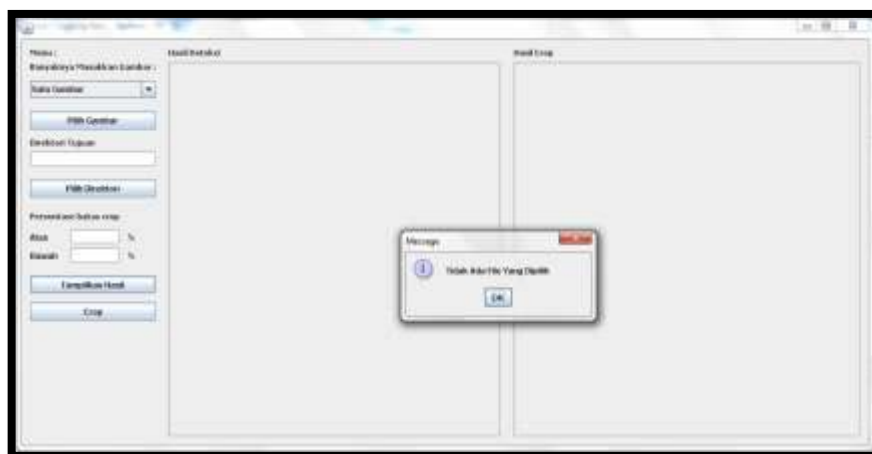
Pada tabel 4.1 dapat dilihat banyak foto dan waktu yang dibutuhkan untuk proses pemotongannya. Foto yang digunakan memiliki ukuran 354x472, 450x600, 500x678, 1063x1500 dan 1938x2907. Untuk proses pemotongan 1 foto, aplikasi ini memakan waktu 4 detik. Untuk proses pemotongan 5000 foto memakan waktu 1 jam 33 menit 48 detik. Rata-rata waktu yang dibutuhkan untuk proses pemotongan 1 foto dalam 5000 foto adalah 1 detik. Berdasarkan hasil pada pengujian aplikasi yang dilakukan, didapatkan hasil bahwa kinerja aplikasi sangat membantu dari segi waktu dan tenaga.

### 4.3.3 Penanganan Kesalahan Masukkan Sistem

Penanganan kesalahan ini dilakukan supaya proses dapat dilakukan tanpa adanya kesalahan masukan dari pengguna yang tidak sesuai dengan prosedur aplikasi tersebut. Penanganan kesalahan pada aplikasi *automatic cropping* pas foto ini dapat dilihat sebagai berikut :

#### 1. Peringatan File Gambar Belum Dipilih

Proses pemotongan (*cropping*) akan dijalankan dengan menekan tombol crop. Ketika tombol tersebut ditekan tetapi file gambar belum dipilih oleh user, maka akan muncul peringatan seperti **Gambar 4.8**.



**Gambar 4. 8** Penanganan Kesalahan File Gambar

#### 2. Peringatan Direktori Tujuan Belum Dipilih

Proses pemotongan (*cropping*) akan dijalankan dengan menekan tombol crop. Ketika tombol tersebut ditekan tetapi direktori tujuan belum dipilih oleh pengguna, maka akan muncul peringatan seperti **Gambar 4.9**.



**Gambar 4. 9** Penanganan Kesalahan Direktori Tujuan

3. Peringatan Masukkan Batas Atas dan Batas Bawah Belum Diisi  
Ketika tombol crop ditekan sedangkan masukkan batas atas dan batas bawah masih kosong, maka akan muncul peringatan seperti **Gambar 4.10**.



**Gambar 4. 10** Penanganan Kesalahan Masukkan Batas Atas dan Batas Bawah yang Masih Kosong

4. Peringatan Masukkan Batas Atas dan batas Bawah Bukan Bilangan  
Ketika tombol crop ditekan sedangkan masukkan batas atas dan batas bawah bukan berupa angka, maka akan muncul peringatan seperti **Gambar 4.11**.



**Gambar 4. 11** Penanganan Kesalahan Masukkan Batas Atas dan Batas Bawah Selain Angka

#### **4.4 Kelebihan Aplikasi**

Kelebihan yang terdapat pada aplikasi *Automatic Cropping Pas Foto* ada, sebagai berikut :

1. Dapat mendeteksi warna kulit pada gambar.
2. Dapat melakukan pemotongan foto pada banyak gambar secara cepat.
3. Kesalahan masukkan oleh pengguna dapat tertangani dengan baik.

#### **4.5 Kekurangan Aplikasi**

Kekurangan yang terdapat pada aplikasi *Automatic Cropping Pas Foto* adalah, sebagai berikut :

1. Aplikasi hanya dapat digunakan pada gambar dengan ukuran dibawah 2000 piksel.
2. Pendeteksian letak wajah berdasarkan pendeteksian warna piksel kulit, belum dapat mendeteksi letak mata dan bibir.
3. Aplikasi belum dapat mendeteksi lebih dari satu wajah.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Setelah melewati pengujian pada aplikasi *Automatic Cropping* Pas Foto yang dibuat sebagai penelitian tugas akhir ini, maka didapat kesimpulan sebagai berikut :

1. Aplikasi *Automatic Cropping* Pas Foto telah dapat mendeteksi sebagian besar warna kulit.
2. Dengan adanya aplikasi *Automatic Cropping* Pas Foto dapat membantu proses pemotongan (*cropping*) pas foto dari segi efisiensi waktu dan tenaga.
3. Aplikasi ini memiliki penanganan kesalahan yang cukup baik.
4. Aplikasi ini dapat dijadikan sebagai prototype untuk membuat aplikasi pengenalan wajah.

#### **5.2 Saran**

Untuk lebih mengembangkan aplikasi *Automatic Cropping* Pas Foto ini, hendaknya dapat ditambahkan :

1. Aplikasi dapat dikembangkan agar dapat memproses gambar dengan ukuran besar lebih dari 2000 piksel.
2. Dapat ditambahkan pendeteksian letak mata dan bibir untuk pengenalan wajah yang lebih baik.
3. Dapat dikembangkan agar dapat mendeteksi lebih dari satu wajah pada satu gambar.

**DAFTAR PUSTAKA**

- Ahmad, Usman. 2005. Pengolahan Citra Digital Dan Teknik Pemrogramannya. Graha Ilmu. Yogyakarta.
- Basilio, J. A. M., et al. 2011. Detection of Pornographic Digital Images. International Journal of Computers: Issue 2, Volume 5. Mexico.
- Darwin, Ian. 2004. Java™ Cookbook™. O'Reilly. United State of America.
- K, Sandeep & A. N., Rajagopalan. Human Face Detection in Cluttered Color Images Using Skin Detection Color and Edge Information. Department of Electrical Engineering Indian Institute of Technology. Madras Chennai, 600 036, India.
- Nugroho, Eko. 2009. Biometrika Mengenal Sistem Identifikasi Masa Depan. Andi. Yogyakarta.
- Putra, Darma. 2010. Pengolahan Citra Digital. Andi, Yogyakarta.
- Suwarga. 2009. Dasar Pemrograman Komputer. Andi, Yogyakarta.
- Teixiera, Hugo Vidal. 2008. High-Quality Image Resize with Java. <<http://www.componenthouse.com/article-20>>. Diakses 12 Oktober 2011.
- Wijono, Matius., dkk. 2004. Java™ 2 SE dengan JBuilder®. Andi, Yogyakarta.