

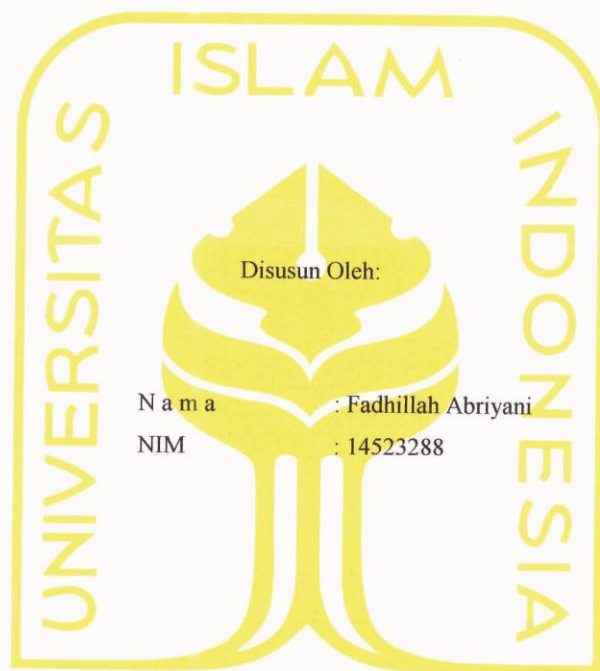
# **DETEKSI KERIPUT PADA CITRA WAJAH DENGAN TEKNIK PENGOLAHAN CITRA**



Disusun Oleh:

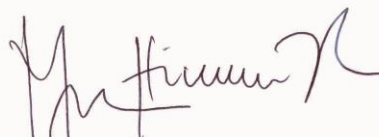
N a m a : Fadhillah Abriyani  
NIM : 14523288


**PROGRAM STUDI TEKNIK INFORMATIKA - PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
2018**

**HALAMAN PENGESAHAN DOSEN PEMBIMBING****DETEKSI KERIPUT PADA CITRA WAJAH DENGAN  
TEKNIK PENGOLAHAN CITRA****TUGAS AKHIR**

Yogyakarta, 18 Juli 2018

Pembimbing I, Pembimbing II,

  
( Izzati Muhimmah S.T., M.Sc., Ph.D. )

  
( Arrie Kurniawardhani S.Si., M.Kom )

## HALAMAN PENGESAHAN DOSEN PENGUJI

**DETEKSI KERIPUT PADA CITRA WAJAH DENGAN  
TEKNIK PENGOLAHAN CITRA  
TUGAS AKHIR**

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 15 Agustus 2018

Tim Penguji

Izzati Muhimmah, S.T., M.Sc., Ph.D

Anggota 1

Arrie Kurniawardhani, S.Si., M.Kom

Anggota 2

Zainudin Zukhri, ST., M.I.T

Mengetahui,

Ketua Program Studi Teknik Informatika - Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan berikut:

Nama : Fadhillah Abriyani

NIM : 14523288

Tugas akhir dengan judul:

### **DETEKSI KERIPUT PADA CITRA WAJAH DENGAN TEKNIK PENGOLAHAN CITRA**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 18 Juli 2018

  
( Fadhillah Abriyani )

## HALAMAN PERSEMBAHAN

*Alhamdulillahirabbil'alamin, Alhamdulillahirabbil'alamin, Alhamdulillahirabbil'alamin.*

Segala puji bagi Allah SWT yang telah mengizinkan dan memudahkan saya untuk bisa menyelesaikan pendidikan dan Tugas Akhir saya di UII.

Tugas Akhir ini saya persembahkan kepada:

Ibu Izzati Muhimmah, ST., M.Sc., PH.D

Ibu Arrie Kurniawardhani S.Si., M.Kom

Bapak Alm.Sutrisno KS ayah saya tercinta yang selalu saya rindukan

Ibu Suis ibu terbaik yang ada dihidup saya

Kakak Fitri Saraswati dan Adik Taufik Hidayat

Dan semua yang telah mendukung, mendoakan dan menjadi alasan utama saya untuk tetap berjuang, walau saya tau itu tidak mudah.

Terima Kasih

## HALAMAN MOTO

- وَقَالَ رَبُّكُمْ ادْعُونِي أَسْتَجِبْ لَكُمْ - ٦٠ -

*Dan Tuhan-mu Berfirman, "Berdoalah kepada-Ku, niscaya akan Aku kabulkan untukmu." (Q.S. Al-Mu'min : 60)*

*"Setiap kali kamu merasa beruntung, percayalah doa ibumu telah didengar"*

-unkown-

*"Pecayalah, takdir Allah selalu baik, sangat baik,  
kita saja yang belum melihatnya"*

-@alvisyhm-

*"Jalani aja, nanti bakal selesai sendiri kok"*

-diri sendiri-

## KATA PENGANTAR

### *Assalamu'alaikum Warahmatullahi Wabarakatuh*

Alhamdulillahirabbil'alamin, puji syukur senantiasa dipanjatkan kehadiran Allah SWT, yang telah memberikan rahmat dan hidayah-Nya baik berupa kenikmatan maupun kesehatan lahir dan batin, sehingga pelaksanaan pengerjaan proyek Tugas Akhir dapat berjalan dengan baik dan lancar sampai dengan adanya penyusunan laporan ini.

Pelaksanaan Tugas Akhir ini merupakan salah satu mata kuliah wajib dan sebagai salah satu syarat akhir yang harus dipenuhi untuk memperoleh gelar sarjana dari jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia. Adapun Tugas Akhir yang dikerjakan adalah mengenai **Deteksi Keriput Pada Citra Wajah Dengan Teknik Pengolahan Citra**.

Penyusun menyadari bahwa pelaksanaan Tugas Akhir dan penyusunan laporan ini tidak lepas dari bimbingan, dorongan dan bantuan baik materi maupun non materi dari berbagai pihak, sehingga proyek yang telah direncanakan dapat terealisasi dengan baik dan dapat diselesaikan dengan tepat waktu. Oleh karena itu, perkenankanlah penulis menghaturkan ucapan terima kasih kepada:

1. Allah SWT, karena telah diberikan segala karunia-Nya, sehingga selama kegiatan Tugas Akhir dapat berlangsung dengan baik dan lancar.
2. Rasulullah SAW, manusia terbaik yang pernah diciptakan, yang dengan teladannya membawa manusia dari jaman kegelapan hingga jaman terang benderang seperti saat ini.
3. Ayah Sutrisno KS (Alm) dan Ibu Suis selaku orang tua, ndut, mpit dan keluarga, yang tak pernah lelah memberikan doa, dukungan baik moral dan finansial, serta dukungannya.
4. Bapak Nandang Sutrisno, SH., M.Hum., LL.M., Ph.D. selaku Rektor Universitas Islam Indonesia periode antar waktu 2018 dan Bapak Fathul Wahid, S.T., M.Sc., Ph.D. selaku Rektor Universitas Islam Indonesia periode 2018-2022.
5. Bapak Dr. Drs. Imam Djati Widodo, M.Eng.Sc. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia periode 2014-2018 dan Bapak Prof. Dr. Ir. Hari Purnomo., M.T. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia periode 2018-2022.
6. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Prodi Studi Teknik Informatika Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia
7. Bapak Hendrik, S.T., M.Eng. selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

8. Ibu Izzati Muhimmah S.T., M.Sc., Ph.D. selaku Dosen Pembimbing satu Tugas Akhir dan Ibu Arrie Kurniawardhani S.Si., M.Kom. selaku Dosen Pembimbing dua di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang dengan sabar membimbing dan memberikan arahan dalam proses pengerjaan proyek Tugas Akhir ini.
9. Ibu dr. Rosmelia, M.Kes., Sp.KK dan pihak PT AVO Beauty yang sudah bersedia menjadi narasumber dan pihak penguji sistem.
10. Datin dan Nadya teman dari semester 1, Maria teman ter pintar dan ter polosku, terimakasih sudah mengajarku di setiap semesternya, dan teruntuk Nurul terima kasih sudah bersedia menjemput dan berjuang bersama menyelesaikan Tugas Akhir ini.
11. Muara Bungo Squad terkhusus untuk Ajengs, Koko, Dita dan Teta terima kasih sudah memilih Jogja dan menemani hari-hari panjangku dikota ini.
12. Elang dan Mas Dwi, terima kasih sudah bersedia menjadi teman, mentor, dan abang yang baik.
13. Teman-teman KKN Fira, Ici, Sari, Tata, Papiw, Panjul, Indra dan Anwar terima kasih sudah menambah cerita baru dalam dunia perkuliahanku.
14. Dan untuk semua pihak yang telah ikhlas membantu dan direpotkan oleh saya terima kasih, semoga kalian selalu dilancarkan segala urusan dan rezekinya oleh Allah SWT.

Tidak lupa penulis memohon maaf apabila selama melaksanakan Tugas Akhir terdapat kekhilafan dan kesalahan yang disadari maupun tidak disadari, semoga laporan akhir Tugas Akhir ini dapat bermanfaat bagi semua pihak yang membutuhkan sebagai referensi, Aamiin.

***Wassalammu'alaikum Warahmatullahi Wabarakatuh***

Yogyakarta, 18 Juli 2018

( Fadhillah Abriyani )



## SARI

Kulit merupakan bagian terluar dari tubuh manusia yang sangat berperan penting sebagai pelindung tubuh dari benturan, goresan dan penyinaran. Bagian kulit yang paling rentan mengalami penyinaran matahari adalah bagian wajah, akibatnya kulit wajah sering mengalami kerusakan, salah satunya yaitu keriput. Keriput adalah lipatan, garis atau kerutan yang muncul pada bagian wajah dan leher, keriput muncul akibat berkurangnya kolagen pada kulit, sehingga menyebabkan kulit mengendur dan kehilangan elastisitas. Umumnya kemunculan keriput pada wajah merupakan salah satu tanda penuaan, namun tidak jarang keriput muncul pada usia yang relatif muda atau sering disebut sebagai penuaan dini.

Keriput pada citra dapat diamati dengan fitur lengkung (bukan gumpalan), bersifat kontinu (bukan serangkaian lengkungan yang terputus-putus), dan tidak saling berpotongan satu sama lain. Ciri garis keriput dapat dideteksi dengan fitur-fitur yang terdapat pada Matlab, seperti filter *Gabor*, *Hessian*, *Sobel* dan *Canny*.

Penelitian ini menggunakan segmentasi warna *HSV*, *K-Means* dan filter *Imfill* untuk memisahkan objek warna kulit dengan warna *non*-kulit, selanjutnya pengambilan area keriput dilakukan dengan metode *Persentase*. Untuk mendeteksi garis keriput, dilakukan ekstraksi ciri dengan menggunakan metode *Canny* dengan *threshold* untuk mendeteksi garis-garis pada wajah. Untuk menghilangkan garis-garis *noise* (bukan keriput), hasil citra metode *Canny* dieliminasi dengan mencari nilai rata-rata luas objek garis keriput pada setiap area keriput. Hasil deteksi oleh sistem kemudian diuji dengan menggunakan uji validasi *Single Decision Threshold* yang menghasilkan nilai *sensitivity* yang diperoleh dari pengujian dengan AVO adalah 61,5%, nilai *specificity* adalah 98,1%, dan nilai *accuracy* adalah 96,1%, sedangkan untuk nilai *sensitivity* yang diperoleh dari pengujian dengan dokter ahli adalah 40,6%, nilai *specificity* adalah 96,9%, dan nilai *accuracy* adalah 94,2% dengan performa sistem keseluruhan sebesar 5,247 detik.

Kata kunci: keriput, *HSV*, *Canny Edge Detection*, *Single Decision Threshold*.

## GLOSARIUM

<i>Citra Biner</i>	Citra atau image dengan skala nilai 0 dan 1
<i>Citra Grayscale</i>	Citra dengan skala abu-abu yang memiliki nilai 0 hingga 255
<i>Noise</i>	Objek-objek yang tidak diinginkan
<i>Prototype</i>	Perancangan atau model sederhana sebuah sistem
<i>Regionprops</i>	Fungsi matlab yang digunakan untuk mendapatkan nilai-nilai vector dari objek citra Biner
<i>Threshold</i>	Nilai ambang batas

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	ii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR .....	vii
SARI.....	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR .....	xv
BAB I PENDAHULUAN.....	1
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 .....	3
1.7 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	6
2.1 Kulit.....	6
2.2 Keriput.....	6
2.2.1 Pengertian Keriput.....	6
2.2.2 Area Keriput .....	7
2.2.3 Faktor-Faktor Penyebab Keriput .....	8
2.2.4 Ciri Klinis Keriput .....	9
2.3 Pengolahan Citra .....	9
2.3.1 Citra Digital .....	10
2.3.2 Jenis Citra .....	10
2.3.3 Ruang Warna .....	11
2.4 <i>Preprocessing</i> .....	13

2.4.1	<i>Cropping</i> Citra.....	13
2.4.2	Resize Citra.....	14
2.4.3	Rekonstruksi Citra .....	15
2.5	Segmentasi Citra.....	15
2.6	Perbaikan Citra .....	16
2.7	Ekstraksi Ciri Bentuk .....	17
2.7.1	Deteksi Tepi.....	18
2.7.2	<i>Area</i> (Luas) .....	19
2.8	Uji Validasi Sistem.....	19
2.9	Penelitian Serupa.....	21
BAB III METODOLOGI PENELITIAN .....		23
3.1	Pengumpulan Data.....	23
3.2	Tahap Analisis .....	23
3.2.1	Analisis Permasalahan .....	23
3.2.2	Analisis Data.....	23
3.2.3	Analisis Kebutuhan <i>Input</i> .....	24
3.2.4	Analisis Kebutuhan Proses .....	24
3.2.5	Analisis Kebutuhan <i>Output</i> .....	24
3.2.6	Analisis Pengujian Sistem .....	24
3.3	Perancangan Perangkat Lunak .....	25
3.3.1	Diagram Konteks .....	25
3.3.2	Gambaran Umum Sistem.....	25
3.3.3	<i>Preprocessing</i> .....	27
3.3.4	Segmentasi Citra.....	27
3.3.5	Perbaikan Citra .....	28
3.3.6	Ekstraksi Ciri .....	28
3.3.7	<i>Marking</i> .....	28
3.4	Hasil Perancangan Antarmuka .....	28
3.4.1	Perancangan Halaman Utama.....	29
3.4.2	Perancangan <i>FileChooser</i> .....	30
BAB IV HASIL DAN PEMBAHASAN .....		31
4.1	Data Citra.....	31
4.2	<i>Preprocessing</i> .....	31
4.2.1	<i>Cropping</i> Citra.....	31

4.2.2	<i>Resize</i> .....	32
4.3	Segmentasi Citra.....	34
4.3.1	Segmentasi Warna Kulit dan Pengembangan ( <i>Threshold</i> ).....	34
4.4	Perbaikan Citra .....	40
4.5	Ekstraksi Ciri.....	42
4.5.1	Area Keriput .....	42
4.5.2	<i>Canny</i> dan <i>Area</i> .....	51
4.6	<i>Marking</i> .....	64
4.7	Hasil Ekstraksi Ciri Keriput .....	66
4.8	Implementasi Tampilan Antarmuka .....	71
4.8.1	Tampilan Awal Sistem .....	71
4.8.2	Tampilan <i>FileChooser</i> .....	71
4.8.3	Tampilan <i>Load</i> Foto dan Hasil Deteksi .....	72
4.9	Hasil Deteksi .....	72
4.10	Hasil Uji Validasi Sistem .....	73
4.11	Penyebab Kegagalan Sistem.....	78
4.12	Kelebihan dan Kekurangan Sistem .....	80
BAB V KESIMPULAN DAN SARAN .....		81
5.1	Kesimpulan.....	81
5.2	Saran.....	82
DAFTAR PUSTAKA .....		83
LAMPIRAN.....		86

## DAFTAR TABEL

Tabel 2.1 Analisis ciri fitur .....	9
Tabel 2.2 Tabel <i>confusion matrix</i> .....	20
Tabel 2.3 Tabel perbandingan penelitian serupa .....	21
Tabel 4.1 Perhitungan ekstraksi ciri garis keriput ( <i>Area</i> ) .....	66
Tabel 4.2 Tabel pengujian sistem (AVO) .....	74
Tabel 4.3 Tabel <i>confusion matrix</i> hasil pengujian (AVO) .....	75
Tabel 4.4 Tabel pengujian sistem (Dokter ahli) .....	75
Tabel 4.5 Tabel <i>confusion matrix</i> hasil pengujian (Dokter ahli) .....	77
Tabel 4.6 Tabel performa sistem .....	77

## DAFTAR GAMBAR

Gambar 2.1 Struktur kolagen kulit wajah .....	7
Gambar 2.2 Ilustrasi kulit wajah kekurangan kolagen.....	7
Gambar 2.3 Area keriput.....	8
Gambar 2.4 Model warna <i>RGB</i> .....	11
Gambar 2.5 Sistem koordinat <i>HSV</i> .....	12
Gambar 2.6 Alur kerja proses <i>Cropping</i> citra.....	14
Gambar 2.7 Contoh <i>Cropping</i> citra dengan aplikasi paint .....	14
Gambar 2.8 Contoh citra dengan filter <i>Sobel</i> (kiri) dan <i>Canny</i> (kanan).....	19
Gambar 3.1 Diagram konteks .....	25
Gambar 3.2 Gambaran umum sistem.....	26
Gambar 3.3 Perancangan Halaman Utama Sistem .....	29
Gambar 3.4 Perancangan Halaman <i>FileChooser</i> .....	30
Gambar 4.1 Contoh citra.....	31
Gambar 4.2 Hasil citra <i>Crop</i> : (a). Citra sebelum di <i>Crop</i> (b). Citra sesudah <i>Crop</i> .....	32
Gambar 4.3 Diagram alur proses <i>Resize</i> citra.....	33
Gambar 4.4 <i>Sintaks</i> proses <i>Resize</i> citra.....	33
Gambar 4.5 Diagram alur proses segmentasi citra dengan <i>HSV</i> .....	35
Gambar 4.6 <i>Sintaks</i> proses segmentasi citra dengan <i>HSV</i> .....	35
Gambar 4.7 Hasil segmentasi citra dengan <i>HSV</i> .....	36
Gambar 4.8 Diagram alur proses segmentasi citra dengan <i>K-Means Clustering</i> .....	37
Gambar 4.9 <i>Sintaks</i> proses segmentasi citra dengan <i>K-Means Clustering</i> .....	38
Gambar 4.10 Hasil segmentasi citra dengan <i>K-Means Clustering</i> .....	38
Gambar 4.11 Diagram alur proses <i>Imfill</i> .....	39
Gambar 4.12 <i>Sintaks</i> proses filter <i>Imfill</i> .....	39
Gambar 4.13 Citra hasil filter <i>Imfill</i> .....	40
Gambar 4.14 Diagram alur proses penajaman citra.....	41
Gambar 4.15 <i>Sintaks</i> proses filter <i>Laplacian</i> .....	41
Gambar 4.16 Hasil perbaikan citra: (a). Citra <i>Laplacian</i> .....	42
Gambar 4.17 Diagram alur proses pengambilan area dahi .....	43
Gambar 4.18 Diagram alur proses pengambilan area sisi tengah mata .....	44
Gambar 4.19 Diagram alur proses pengambilan area sisi mata kiri .....	45
Gambar 4.20 Diagram alur proses pengambilan area sisi mata kanan .....	46

Gambar 4.21 Diagram alur proses pengambilan area kantung mata kiri.....	47
Gambar 4.22 Diagram alur proses pengambilan area kantung mata kanan.....	48
Gambar 4.23 <i>Sintaks</i> proses pengambilan area keriput .....	50
Gambar 4.24 Hasil citra: (a). Area keriput (b). Area keriput pada citra asli .....	50
Gambar 4.25 Diagram alur proses filter <i>Canny</i> .....	51
Gambar 4.26 Diagram alur proses ekstraksi ciri ( <i>Area</i> ) dahi .....	52
Gambar 4.27 <i>Sintaks</i> proses ekstraksi ciri dahi .....	53
Gambar 4.28 Hasil ekstraksi ciri: (a). <i>Canny</i> area dahi (b). Area 1 dahi (c). Area 2 dahi.....	53
Gambar 4.29 Diagram alur proses ekstraksi ciri ( <i>Area</i> ) sisi tengah mata .....	54
Gambar 4.30 <i>Sintaks</i> proses ekstraksi ciri sisi tengah mata.....	55
Gambar 4.31 Hasil ekstraksi ciri: (a). <i>Canny</i> area sisi tengah mata .....	55
Gambar 4.32 Diagram alur proses ekstraksi ciri ( <i>Area</i> ) sisi kiri mata.....	56
Gambar 4.33 Diagram alur proses ekstraksi ciri ( <i>Area</i> ) sisi kanan mata.....	57
Gambar 4.34 <i>Sintaks</i> proses ekstraksi ciri sisi kiri dan kanan mata .....	59
Gambar 4.35 Hasil ekstraksi ciri: (a). <i>Canny</i> area sisi kiri mata (b). Area 1 sisi kiri mata .....	59
Gambar 4.36 Hasil ekstraksi ciri: (a). <i>Canny</i> area sisi kanan mata (b). Area 1 sisi.....	59
Gambar 4.37 Diagram alur proses ekstraksi ciri ( <i>Area</i> ) kantung mata kiri .....	60
Gambar 4.38 Diagram alur proses ekstraksi ciri ( <i>Area</i> ) kantung mata kanan.....	61
Gambar 4.39 <i>Sintaks</i> proses ekstraksi ciri kantung kiri dan kanan mata.....	63
Gambar 4.40 Hasil ekstraksi ciri: (a). <i>Canny</i> area kantung kiri mata (b). Area 1 kantung .....	63
Gambar 4.41 Hasil ekstraksi ciri: (a). <i>Canny</i> area kantung kanan mata (b). Area 1 kantung kanan mata (c). Area 2 kantung kanan mata (d). Area 3 kantung kanan mata .....	63
Gambar 4.42 Diagram alur proses <i>marking</i> .....	64
Gambar 4.43 <i>Sintaks</i> proses <i>marking</i> .....	65
Gambar 4.44 Hasil penandaan keriput.....	66
Gambar 4.45 Grafik luas ( <i>Area</i> ) dahi .....	67
Gambar 4.46 Grafik luas ( <i>Area</i> ) sisi tengah mata.....	68
Gambar 4.47 Grafik luas ( <i>Area</i> ) sisi mata kiri.....	69
Gambar 4.48 Grafik luas ( <i>Area</i> ) sisi mata kanan.....	69
Gambar 4.49 Grafik luas ( <i>Area</i> ) kantung mata kiri .....	70
Gambar 4.50 Grafik luas ( <i>Area</i> ) kantung mata kanan .....	70
Gambar 4.51 Tampilan awal sistem.....	71
Gambar 4.52 Tampilan <i>filechooser</i> sistem .....	71
Gambar 4.53 Tampilan <i>load</i> foto pada sistem .....	72



Gambar 4. 54 Tampilan hasil deteksi .....	72
Gambar 4.55 Citra hasil deteksi: (a). Hasil deteksi sempurna (b). Hasil deteksi .....	73
Gambar 4.56 Citra hasil segmentasi: (a). Citra asli (b). Hasil citra tidak sempurna .....	78
Gambar 4.57 Citra pengambilan area keriput tidak sempurna .....	79
Gambar 4.58 Citra hasil ekstraksi ciri (a). Citra asli (b). Citra hasil deteksi keriput.....	80

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Kulit merupakan organ tubuh manusia yang memiliki luas paling besar dan terletak paling luar yang berperan penting bagi manusia, selain itu kulit juga menjadi aset utama dan prioritas bagi kaum wanita (Wahyuningtyas et al., 2015). Kulit berfungsi sebagai pelindung dari kerusakan fisik seperti benturan, goresan, dan penyinaran, akibatnya kulit sering mengalami kerusakan dan kelainan. Kulit wajah merupakan bagian tubuh yang sering mengalami kerusakan karena sering terkena polusi dan paparan sinar matahari secara langsung.

Salah satu masalah kerusakan kulit pada wajah adalah keriput. Keriput adalah lipatan, garis atau kerutan yang muncul pada bagian wajah dan leher, umumnya kemunculan keriput pada wajah merupakan salah satu tanda penuaan, namun tidak jarang keriput muncul pada usia yang relatif muda atau sering disebut sebagai penuaan dini. Penyebab munculnya keriput pada usia muda adalah penyinaran matahari, lingkungan luar yang tidak mendukung, kebiasaan merokok, penurunan berat badan yang terlalu cepat, dan pemakaian kosmetik yang salah.

Dampak yang ditimbulkan akibat munculnya keriput pada wajah di antaranya kulit wajah mengendur karena kehilangan elastisitas, wajah terlihat lebih tua dari umur seharusnya, dan menambah biaya perawatan kulit wajah. Keriput juga dapat menyebabkan aspek psikososial seseorang terganggu, akibat adanya berbagai perasaan seperti tersingkir dan rendah diri, seolah menderita penyakit kronis, kurang memperdulikan kesehatan diri, dan mendapat diskriminasi di tempat kerja (ETCADM, 2017). Hal-hal tersebut memicu banyak orang untuk mengunjungi klinik kecantikan dan mengkonsumsi berbagai jenis obat untuk memudahkan keriput.

Untuk mengatasi permasalahan dampak dari keriput di atas, penelitian ini bekerja sama dengan pihak PT AVO merancang sebuah sistem yang dapat mendeteksi letak keriput pada wajah melalui foto pengguna, sistem akan mempermudah pengguna untuk mengetahui letak garis keriput pada wajah, sehingga penanganan keriput akan lebih efektif. Pengguna akan fokus mengobati keriput pada area wajah yang keriputnya terlihat cukup banyak, hal ini diharapkan dapat membantu proses penanganan yang tepat untuk mencegah dan mengurangi keriput semakin banyak terlihat. Sistem ini akan dirancang dengan *Digital Image Processing* yang nantinya dapat mendeteksi keriput di bagian wajah yang rawan munculnya keriput seperti bagian dahi, sisi tengah antara mata dan hidung, sudut luar mata, serta kantung mata. Nantinya

sistem ini akan diterapkan pada aplikasi *skin advisor* di *website AVO* untuk dapat mendeteksi dan menampilkan garis-garis keriput pada wajah pelanggan melalui foto yang telah diunggah.

## 1.2 Rumusan Masalah

Berdasarkan permasalahan pada latar belakang di atas, rumusan masalah yang ditentukan adalah:

- a. Bagaimana membedakan bagian wajah manusia dengan *background* ?
- b. Bagaimana cara untuk mengambil area-area munculnya keriput pada wajah ?
- c. Berapa nilai luas (*Area*) yang digunakan untuk mengenali objek keriput ?

## 1.3 Batasan Masalah

Batasan masalah yang ditentukan agar pengerjaan tugas akhir ini tidak menyimpang adalah sebagai berikut:

- a. Sistem dirancang dengan bantuan fitur-fitur pada Matlab.
- b. Format citra yang digunakan adalah JPG, PNG, dan JPEG.
- c. Citra yang digunakan harus citra yang memuat wajah dengan pencahayaan yang merata dan fokus menghadap lurus ke depan.
- d. Sistem hanya dapat mendeteksi masalah keriput yang terdapat pada wajah manusia.
- e. Sistem hanya mendeteksi keriput pada bagian yang telah ditentukan yaitu bagian dahi, sisi tengah antara mata dan hidung, sudut luar mata, dan kantung mata.
- f. Hasil akhir dari perancangan sistem ini adalah sistem berbasis desktop.

## 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan di atas, tujuan dari penelitian ini adalah menghasilkan sebuah sistem berbasis desktop yang dapat mendeteksi letak garis keriput pada bagian-bagian keriput yang telah ditentukan.

## 1.5 Manfaat Penelitian

Berdasarkan latar belakang yang telah diuraikan di atas, penelitian ini diharapkan dapat memberi manfaat sebagai berikut:

- a. Untuk membangun basis pengetahuan mengenali citra wajah.
- b. Untuk membangun basis pengetahuan mengenali ciri garis-garis keriput pada citra wajah.

- c. Dapat diterapkan pada aplikasi *skin advisor* yang terdapat pada *website* AVOSKIN BEAUTY agar proses pendeteksian keriput dapat dilakukan secara otomatis melalui citra wajah.
- d. Dapat digunakan sebagai tolak ukur untuk penelitian selanjutnya dalam penggunaan teknik pengolahan citra dalam kepentingan akademis mahasiswa informatika, kesehatan dan kecantikan.

## 1.6 Metodologi Penelitian

Adapun langkah-langkah yang diterapkan untuk mengembangkan sistem dalam penelitian ini adalah sebagai berikut:

- a. Tahap Pengumpulan Data
  - 1. Studi pustaka: dilakukan dengan mencari data-data dari buku, artikel, dan jurnal yang berkaitan dengan penyakit kulit khususnya keriput, deteksi wajah dan keriput yang dapat dijadikan sebagai referensi untuk tugas akhir.
  - 2. Wawancara dilakukan dengan Dr. Rosmelia., M.Kes., Sp. KK selaku dokter dokter ahli penyakit kulit dan kelamin fakultas kedokteran UIL.
  - 3. Observasi: dilakukan dengan mencoba langsung alat *Magic Mirror Skin Anayzer* di klinik kecantikan *Be Queen Skin Care* yang dapat memberikan rentang nilai mengenai masalah kulit wajah.
  - 4. Pencarian citra: citra yang mempunyai ciri keriput dengan kualitas yang baik didapat dari database AVO dan citra dari Internet.
- b. Tahap Pembuatan Sistem
  - 1. Analisis Kebutuhan Sistem
 

Pada tahap ini dilakukan proses pemodelan terhadap kebutuhan sistem untuk identifikasi garis keriput pada wajah yang dimulai dari analisis kebutuhan input, analisis kebutuhan output, dan analisis kebutuhan perangkat lunak dari sistem yang akan dirancang.
  - 2. Perancangan Sistem
 

Pada tahap ini dilakukan perancangan sistem yang bertujuan untuk mengembangkan sistem deteksi keriput pada citra wajah.
  - 3. Pengujian Kerja Sistem
 

Analisis kerja sistem akan diuji keakuratannya dengan mencocokkan hasil deteksi keriput dari sistem dengan hasil deteksi keriput dari dokter dan jurnal. Hasil

klasifikasi dari dua rater akan dibandingkan dan dihitung dengan menggunakan *Confusion Matrix*.

## **1.7 Sistematika Penulisan**

Penulisan Tugas Akhir ini disusun secara sistematis yang terdiri dari beberapa bab dengan subbab tertentu. Adapun uraian singkat mengenai isi pada laporan ini adalah sebagai berikut:

### **Bab I Pendahuluan**

Berisi latar belakang yang menjadi alasan dilakukan penelitian serta berisi penjelasan mengenai keriput pada wajah dan deteksi wajah. Kemudian dari latar belakang masalah dibuatlah rumusan masalah yang akan menjadi acuan dalam penyelesaian masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian yang berisi mengenai langkah dan metode untuk mencapai tujuan penelitian dan terdapat sistematika penulisan yang berfungsi sebagai acuan dalam penulisan Tugas Akhir.

### **Bab II Landasan Teori**

Berisi uraian tentang teori-teori atau tinjauan pustaka yang menjadi dasar penelitian. Pada bab ini memuat referensi yang berasal dari berbagai buku, artikel dan jurnal mengenai keriput, deteksi wajah, segmentasi citra, dan filter *Canny*.

### **Bab III Analisis Permasalahan**

Berisi mengenai analisis permasalahan data yang diambil, analisis metode deteksi wajah, analisis metode deteksi garis keriput, analisis kebutuhan sistem yang terdiri dari analisis kebutuhan masukan dan kebutuhan proses, analisis rancangan antarmuka serta analisis kinerja sistem.

### **Bab IV Metodologi Penelitian**

Berisi mengenai diagram alur sistem, diagram alur proses deteksi wajah dan keriput pada citra wajah, penjelasan mengenai metode deteksi wajah, penjelasan mengenai deteksi keriput, dan penjelasan mengenai rancangan antarmuka yang dirancang.

### **Bab V Hasil dan Pembahasan**

Berisi pembahasan mengenai hasil dari setiap proses yang ada pada sistem sesuai dengan penyelesaian masalah yang diambil, dan pengujian validasi serta implementasi perhitungan deteksi garis keriput yang digunakan. Pengujian citra dilakukan dengan metode pengolahan citra, yang akan disajikan dalam bentuk *Confusion Matrix*.

## **Bab VI Kesimpulan dan Saran**

Berisi kesimpulan yang menyimpulkan apakah tujuan penelitian yang dilakukan sudah tercapai atau belum, dan berisi saran yang membangun agar penelitian dapat dikembangkan lagi oleh pihak yang ingin dan berkepentingan untuk memperbaiki kekurangan pada penelitian ini.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Kulit**

Kulit terdiri dari dua lapisan utama yaitu lapisan *epidermis* dan *dermis*. Lapisan *epidermis* merupakan lapisan terluar kulit yang terdiri atas epitel berlapis gepeng dengan lapisan tanduk (Kalangi, 2013). Ada dua jenis kulit yang melapisi tubuh yaitu kulit tebal yang melapisi telapak tangan, kaki dan jari, dan kulit tipis yang melapisi permukaan tubuh lain.

Kulit berfungsi sebagai pelindung dari kerusakan fisik seperti benturan, goresan, dan penyinaran. Akibatnya, kulit sering kali mengalami kerusakan dan kelainan. Kulit wajah merupakan bagian tubuh yang sering mengalami kerusakan, karena sering terkena polusi dan paparan sinar matahari secara langsung.

Salah satu masalah kerusakan kulit pada wajah adalah keriput, umumnya kemunculan keriput pada wajah merupakan salah satu tanda penuaan, namun tidak jarang keriput muncul pada usia yang relatif muda atau sering disebut sebagai penuaan dini.

#### **2.2 Keriput**

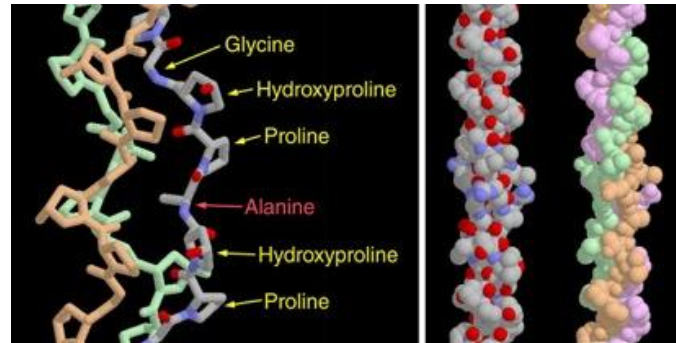
Penjelasan mengenai keriput dapat dilihat pada bagian di bawah ini:

##### **2.2.1 Pengertian Keriput**

Keriput atau kerutan adalah garis, lipatan, yang muncul pada kulit (Alodokter, 2016). Keriput terjadi karena berkurangnya *kolagen* pada kulit, selain itu munculnya keriput terjadi akibat adanya perubahan-perubahan faktor yang menunjang kulit di antaranya:

- a. Serabut *kolagen* dan serabut *elastin* yang berfungsi untuk menjaga kelenturan kulit berubah menjadi kaku dan tidak lentur, sehingga kehilangan daya elastisitasnya.
- b. Tulang dan otot mengalami *atrofi*, jaringan lemak subkutan berkurang disertai lapisan kulit yang tipis, sehingga menyokong terbentuknya kerutan-kerutan dan lipatan-lipatan atau garis halus kulit yang nyata.
- c. Pengaruh kontraksi otot-otot mimik yang tidak diikuti oleh kontraksi kulit yang sesuai, sehingga mengakibatkan garis-garis halus keriput muncul terutama disekitar mulut, mata dan dahi.

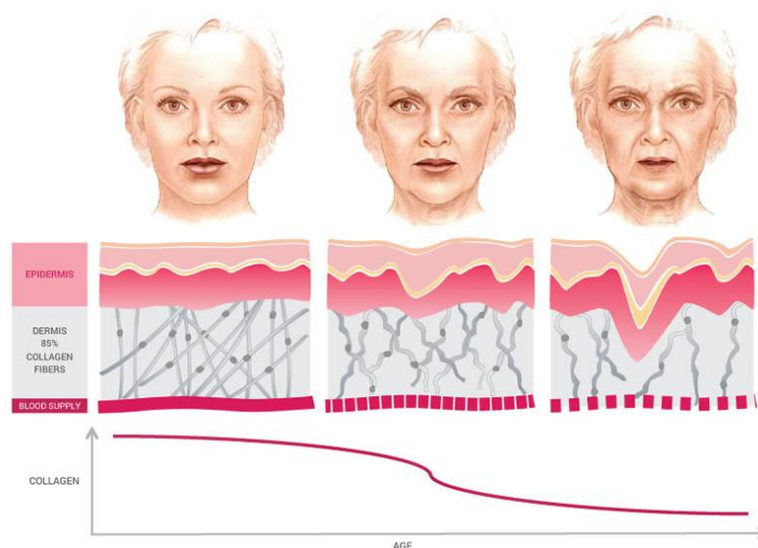
*Kolagen* adalah senyawa protein rantai panjang yang tersusun atas *asam amino*, *alanine*, *arginn*, *lisin*, *glisin*, *prolin*, serta *hiroksiprolin* (Atmaja et al., 2012). Struktur *kolagen* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Struktur kolagen kulit wajah

Sumber : Alevelbiology (2010)

*Kolagen* dan *elastin* adalah komponen utama lapisan *dermis*. Hilangnya serat-serat ini berdampak buruk terhadap kelembaban dan ketegangan kulit, sehingga menimbulkan kerut atau keriput. Kulit yang kekurangan *kolagen* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Ilustrasi kulit wajah kekurangan kolagen

Sumber : Epilighnewskin (2015)

### 2.2.2 Area Keriput

Pada umumnya ada dua jenis keriput yang muncul pada wajah yaitu keriput kasar dan keriput halus. Untuk keriput kasar biasanya muncul pada daerah garis-garis ekspresi pada



wajah, seperti sudut luar mata, dahi dan garis vertikal di kedua sisi mulut. Sedangkan untuk keriput halus berupa lekukan atau garis dangkal yang biasanya muncul pada daerah pergerakan muka seperti mata, mulut dan atas bibir. Sedangkan (Horng et al., 2001) menjelaskan keriput dapat terlihat jelas pada beberapa daerah wajah di antaranya:

- a. Pada dahi terdapat alur horizontal
- b. Pada sudut luar mata terdapat kerutan
- c. Pada pipi terlihat tulang pipi yang jelas, kantung pipi yang berbentuk bulan sabit dan garis dalam di antara pipi dan bibir atas.

Area keriput yang digunakan dalam penelitian ini mengikuti area keriput yang telah ditentukan pada penelitian sebelumnya, yaitu meliputi area dahi, area sisi tengah mata, area sisi mata kiri dan kanan, serta area kantung mata kiri dan kanan. Area keriput dari penelitian sebelumnya dapat dilihat pada Gambar 2.3.

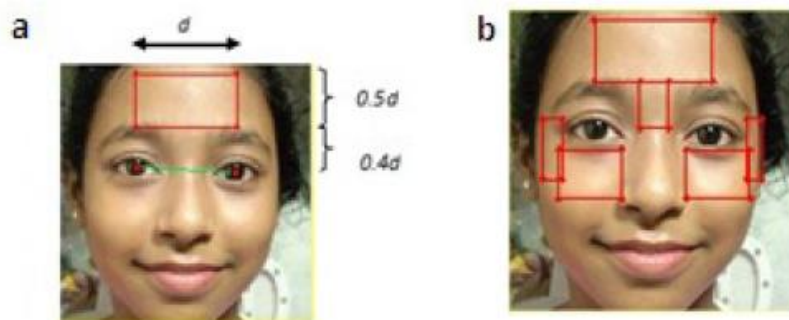


Fig. 3. (a) Forehead detection (b) Wrinkle area on face image

Gambar 2.3 Area keriput

Sumber: Jana et al., (2015)

Area pipi tidak dimasukkan karena sulit membedakan antara garis keriput dengan garis senyum, menurut penjelasan Dr.Rosmelia., M.Kes., Sp. KK menyebutkan bahwa untuk mengetahui garis senyum antara pipi dan bibir atas termasuk garis keriput atau tidak, harus dibandingkan dengan usia seseorang. Hal ini bertujuan untuk dapat melihat kedalaman garis senyum sesuai atau tidak dengan usianya, proses tersebut tidak dilakukan dalam penelitian ini.

### 2.2.3 Faktor-Faktor Penyebab Keriput

Keriput merupakan salah satu kelainan yang terjadi pada kulit menua, ada dua faktor yang mempengaruhi proses menua pada kulit yaitu faktor intrinsik dan faktor ekstrinsik. Faktor intrinsik adalah faktor-faktor dari dalam tubuh yang berpengaruh pada proses kulit menua,

yang meliputi keturunan (*genetik*), *rasial* dan *hormonal*. Sedangkan faktor ekstrinsik adalah faktor-faktor yang dapat menyebabkan proses menua dini pada kulit, sehingga menampilkan wajah yang terlihat lebih tua dari usia sebenarnya, yang termasuk kedalam faktor-faktor ekstrinsik di antaranya faktor lingkungan seperti sinar matahari, kelembaban udara, dan radikal bebas (Jusuf, 2005).

Medicinesia menyebutkan ada beberapa faktor yang dapat mempengaruhi munculnya keriput di antaranya genetika, ekspresi wajah, efek sinar matahari, radikal bebas, efek hormone, penggunaan otot, efek gravitasi, merokok, tidur dan penurunan berat badan yang ekstrem (Medicinesia, 2010). Penelitian kulit pada manusia menunjukkan bahwa kandungan *kolagen* pada kulit yang terpapar sinar matahari 20% lebih rendah dibandingkan kulit yang tidak terpapar sinar matahari.

#### 2.2.4 Ciri Klinis Keriput

Menurut Batool & Chellappa (2014) dalam penelitiannya berasumsi mengenai karakteristik geometri dan gambaran kerutan, kerutan menyebabkan gradien intensitas akibat sorotan gabor, kerutan muncul berupa fitur lengkung (bukan gumpalan), keriput bersifat kontinu (bukan serangkaian lengkungan yang terputus-putus), dan keriput kurang mungkin untuk saling berpotongan satu sama lain.

Ciri klinis keriput dapat dideteksi dengan fitur-fitur yang terdapat pada Matlab, filter yang akan digunakan akan disesuaikan dengan data dan kebutuhan sistem. Beberapa fitur yang diterapkan pada sistem terdapat dalam Tabel 2.1.

Tabel 2.1 Analisis ciri fitur

Ciri Klinis (Bentuk)	Rumus Ciri Fitur Bentuk
1. Terdapat alur atau garis halus horizontal pada daerah dahi	Ekstraksi ciri fitur bentuk yaitu: <i>Filter Canny</i> <i>Area</i>
2. Terdapat lipatan pada sudut luar mata	
3. Terdapat alur atau garis halus vertikal pada bagian tengah antara mata dan hidung bagian atas	

### 2.3 Pengolahan Citra

Untuk memperbaiki kualitas citra dari derau (*noise*), kurang tajam, kabur (*blurring*), warna terlalu kontras dan lain sebagainya dibutuhkan pengolahan citra. Pengolahan citra adalah pengolahan suatu citra dengan menggunakan komputer secara khusus, untuk menghasilkan

suatu citra yang lain (Fadlisyah, 2007). Menurut Ahmad (2005) mengatakan terminologi pengolahan citra digunakan jika hasil pengolahan data berupa citra berbentuk citra lain yang mengandung atau memperkuat informasi khusus pada citra hasil pengolahan sesuai dengan tujuan pengolahannya.

### 2.3.1 Citra Digital

Citra atau *image* adalah suatu cahaya pada bidang dua dimensi atau dapat didefinisikan sebagai fungsi dua dimensi  $f(x,y)$ .  $X$  dan  $y$  adalah koordinat spasial dan amplitude  $f$  pada setiap pasangan  $(x,y)$  yang disebut intensitas (*gray level*) citra pada titik tersebut, jika  $x$  dan  $y$  berhingga (*finite*) dan diskrit (tidak kontiu) maka disebut citra digital. Citra digital terdiri dari sejumlah elemen berhingga yang masing-masing mempunyai lokasi dan nilai, setiap elemen  $x$  dan  $y$  disebut sebagai elemen citra atau peel (piksel) (Iriyanto & Zaini, 2014).

### 2.3.2 Jenis Citra

Ada tiga jenis citra yang umum digunakan dalam pemrosesan citra, berikut adalah jenis citra berdasarkan nilai pikselnya:

#### Citra Warna

Citra warna atau *RGB* merupakan jenis citra yang menyajikan warna dalam bentuk komponen *Red* (merah), *Green* (hijau), dan *Blue* (biru). Setiap komponen warna menggunakan delapan bit yang memiliki nilai berkisar antara 0 sampai 255. Dengan demikian, kemungkinan warna yang dapat disajikan mencapai  $255 \times 255 \times 255$  atau 16.581.375 warna (Nafi'iyah, 2015).

#### Citra Grayscale

Citra *Grayscale* atau citra berskala keabuan merupakan jenis citra yang menangani gradasi warna hitam dan putih, yang menghasilkan efek warna abu-abu. Intensitas warna pada citra jenis ini berkisar antara 0 sampai 255, nilai 0 menyatakan hitam dan nilai 255 menyatakan putih. Citra *Grayscale* umumnya memiliki kedalaman piksel sebesar 8 bit tetapi ada juga citra *Grayscale* yang menggunakan 16 bit untuk penggunaan yang memerlukan ketelitian yang tinggi.

#### Citra Biner

Citra *Biner* merupakan citra yang hanya memiliki sebuah nilai setiap pikselnya yaitu 0 dan 1, setiap piksel terdiri dari warna hitam atau putih. Citra jenis ini hanya terdiri dari 1 bit

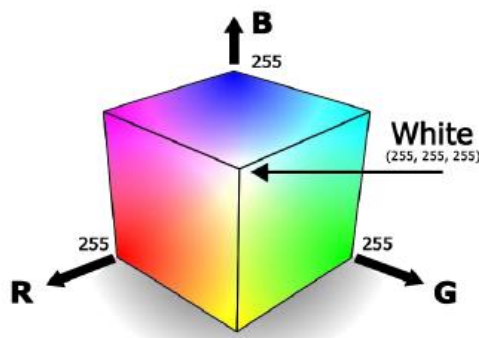
atau jika terdapat 8 bit di dalamnya maka kisaran nilainya antara nilai 0 dan 255. Citra *Biner* banyak digunakan dalam pemrosesan citra untuk memperoleh tepi bentuk suatu objek seperti teks, sidik jari atau gambar arsitektur.

### 2.3.3 Ruang Warna

Gonzales & Woods (2002) mendefinikan ruang warna atau sistem warna sebagai suatu spesifikasi sistem koordinat dan suatu subruang dalam sistem tersebut dengan setiap warna dinyatakan dengan satu titik di dalamnya. Tujuan ruang warna dibentuk adalah untuk memfasilitasi spesifikasi warna dalam bentuk suatu standar. Ruang warna yang digunakan dalam penelitian ini adalah sebagai berikut:

#### **RGB**

Model warna *RGB* adalah model warna berdasarkan konsep penambahan kuat cahaya primer yaitu *Red* (merah), *Green* (hijau), dan *Blue* (biru). Masing-masing nilai *RGB* memiliki nilai antara 0 sampai 255, jika dikombinasikan maka menghasilkan nilai  $256 \times 256 \times 256$  atau 16.777.216 kombinasi warna yang dapat dibentuk (Swedia & Cahyanti, 2010). Model warna *RGB* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Model warna *RGB*

Sumber: (Kolkur et al., 2016)

Salah satu cara yang mudah untuk menghitung nilai warna dan menafsirkan dalam model warna *RGB* adalah dengan melakukan normalisasi terhadap ketiga komponen warna tersebut. Cara melakukan normalisasi ditunjukkan pada Persamaan 2.1 sampai 2.3.

$$r = \frac{R}{R + G + B} \quad (2.1)$$

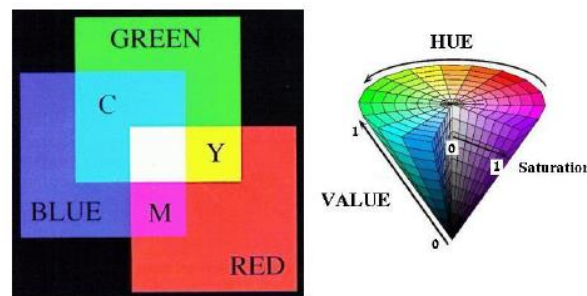
$$g = \frac{G}{R + G + B} \quad (2.2)$$

$$b = \frac{B}{R + G + B} \quad (2.3)$$

Normalisasi dilakukan untuk citra yang ditangkap dengan penerangan yang berbeda-beda. *RGB* digunakan hampir pada semua aplikasi karena ruang warna ini tidak memerlukan transformasi untuk menampilkan informasi ke layar monitor.

### HSV

Model warna *HSV* mendefinisikan warna dalam terminologi *Hue*, *Saturation*, dan *Value*. *Hue* menyatakan warna sebenarnya seperti merah, violet dan kuning. *Hue* digunakan untuk membedakan warna-warna dan menentukan kemerahan (*redness*). *Saturation* menyatakan tingkat kemurnian suatu warna, yaitu mengindikasikan seberapa banyak warna putih yang diberikan pada warna. Sedangkan *value* merupakan atribut yang menyatakan banyak cahaya yang diterima oleh mata tanpa memperdulikan warna (Swedia & Cahyanti, 2010) *HSV* mempunyai dasar *hexacone* berwarna hitam dengan nilai  $HSV = (0, 0, 0)$ , koordinat *HSV* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Sistem koordinat *HSV*

Sumber: (Wibowo, 2011)

Terdapat beberapa cara untuk mendapatkan nilai *H*, *S* dan *V* berdasarkan *R*, *G*, dan *B*, cara yang paling sederhana menurut (Acharya & Ray, 2005) ditunjukkan pada Persamaan 2.4 sampai 2.6.

$$H = \tan\left(\frac{3(G - B)}{(R - G) + (R - B)}\right) \quad (2.4)$$

$$S = 1 - \left( \frac{\min(R, G, B)}{V} \right) \quad (2.5)$$

$$V = \left( \frac{R + G + B}{3} \right) \quad (2.6)$$

Namun cara di atas akan membuat nilai *hue* tidak terdefinisi jika nilai *saturation* bernilai 0, cara yang dapat dilakukan adalah memasukan rumus persamaan 1, selanjutnya dapat memasukan rumus ditunjukkan pada Persamaan 2.7 sampai 2.10.

$$V = \max(r, g, b) \quad (2.7)$$

$$S = \begin{cases} 0, & \text{jika } V = 0 \\ 1 - \frac{\min(r, g, b)}{v}, & V = 0 \end{cases} \quad (2.8)$$

$$H = \begin{cases} 0, & \text{jika } S = 0 \\ \frac{60 \times (g \times b)}{S \times V}, & \text{jika } V = r \\ 60 \times \left[ 2 + \frac{b - r}{S \times V} \right], & \text{jika } V = g \\ 60 \times \left[ 4 + \frac{r - g}{S \times V} \right], & \text{jika } V = b \end{cases} \quad (2.9)$$

$$H = H + 360 \text{ jika } H < 0 \quad (2.10)$$

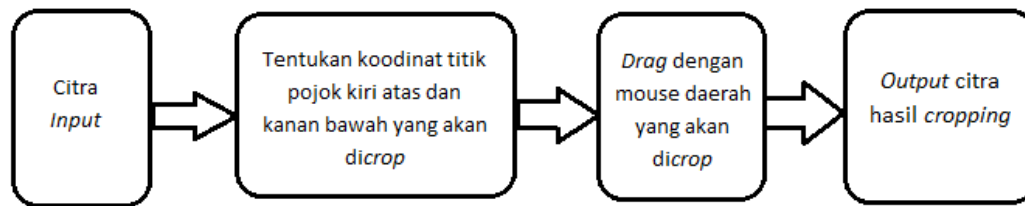
## 2.4 Preprocessing

*Preprocessing* citra dilakukan guna untuk memperbaiki citra yang akan digunakan yang dapat berupa pemotongan gambar, perubahan ukuran dan lain-lain.

### 2.4.1 Cropping Citra

*Cropping* adalah proses pemotongan citra pada koordinat tertentu pada area citra yang bertujuan untuk mengambil elemen citra yang diinginkan pada citra digital. Pada pemotongan citra digunakan dua koordinat yang berfungsi untuk memotong bagian dari citra, yaitu koordinat awal yang merupakan awal koordinat bagi citra hasil pemotongan dan koordinat akhir yang merupakan titik koordinat khir dari citra hasil pemotongan. Citra *Cropping* akan membentuk bangun segi empat dan setiap piksel pada area koordinat tertentu akan disimpan

dalam citra yang baru (Muwardi & Fadlil, 2017). Alur kerja proses *Cropping* citra dapat dilihat pada Gambar 2.6.



Gambar 2.6 Alur kerja proses *Cropping* citra

*Cropping* citra dilakukan secara manual dengan menggunakan *tools Crop* yang terdapat pada aplikasi Paint. Hal ini dikarenakan antar satu citra dengan citra lainnya memiliki ukuran yang berbeda, selain itu ukuran lebar dan tinggi *background* pada setiap citra juga berbeda. Dengan adanya proses *Cropping* akan mempermudah sistem matlab mengambil area keriput dengan tepat dan memungkinkan setiap ukuran citra dapat digunakan pada sistem matlab. Proses *Cropping* citra dapat dilihat pada Gambar 2.7.



Gambar 2.7 Contoh *Cropping* citra dengan aplikasi paint

#### 2.4.2 Resize Citra

*Resize* citra adalah proses mengubah besar ukuran citra digital dalam piksel. Proses *Resize* ini bertujuan untuk menyamakan citra masukan yang akan diproses dalam sistem Matlab. Perubahan ukuran citra dilakukan untuk mengecilkan ukuran citra agar komputer mampu memproses citra dengan cepat, ukuran citra yang diubah dapat berubah menjadi lebih

kecil atau bahkan lebih besar dari file aslinya. Ukuran citra yang digunakan pada penelitian ini adalah 580x580, ukuran tersebut dipilih agar tidak menghabiskan waktu lama saat penghitungan banyak piksel pada citra.

### 2.4.3 Rekonstruksi Citra

Rekonstruksi citra penyusunan atau penggambaran kembali seperti keadaan semula. Salah satu proses rekonstruksi citra pada matlab adalah filter *Imfill (Image Fill)*. *Imfill* dilakukan guna untuk memperbaiki pengolahan citra hasil proses Binerisasi maupun deteksi tepi yang belum baik. *Imfill 'holes'* akan menghilangkan lubang-lubang (*holes*) pada citra dengan melakukan pengisian piksel pada latar belakang dari tepi gambar *Biner*, lubang merupakan sekumpulan piksel *background* yang tidak dapat dicapai atau tidak tertutup.

## 2.5 Segmentasi Citra

Segmentasi citra adalah pemisahan objek yang satu dengan objek yang lain dalam suatu citra atau antara objek dengan latar yang terdapat dalam sebuah citra (Cahyan et al., 2011). Segmentasi citra pada umumnya berdasarkan pada *discontinuity* dan *similarity* dari intensitas piksel. Pendekatan *discontinuity* adalah mempartisi citra jika terdapat perubahan intensitas secara tiba-tiba (*edge based*), sedangkan pendekatan *similarity* adalah mempartisi citra menjadi daerah-daerah yang memiliki kesamaan sifat tertentu (*region based*), contohnya *thresholding*, *region growing*, *region splitting* dan *merging*, dan *clustering*.

### *K-Means Clustering*

*Clustering* didefinisikan sebagai proses sekumpulan objek kedalam kelas-kelas objek yang sama atau pengelompokan (August et al., 2012). Metode *K-Means* adalah salah satu teknik yang digunakan dalam segmentasi berbasis *clustering*. *K-Means Clustering* menurut (Masatu et al., 2014) adalah suatu algoritma pengelompokan objek berdasar pada atribut ke dalam pembagi  $k$ . *Clustering* ini merupakan suatu varian algoritma yang melakukan maksimalisasi kemungkinan, tujuannya adalah untuk menentukan  $k$ . Diberi asumsi bahwa format atribut objek itu adalah suatu garis vektor ruang, tujuannya untuk memperkecil total perbedaan *intra-cluster* atau fungsi (Gonzales dan Woods, 2002) yang ditunjukkan pada Persamaan 2.11.

$$V = \sum_{i=1}^k \sum_{j \in S_i} |x_j - \mu_i|^2 \quad (2.11)$$



Dengan ada  $k$  cluster  $i$   $S$ ,  $i = 1, 2, \dots, k$  dan  $\mu_i$  adalah pusat luasan atau titik dari semua poin-poin. Proses *cluster* dimulai dengan mengidentifikasi data yang akan dicluster.  $X_{ij}$  ( $i=1, \dots, n$ ;  $j=1, \dots, m$ ) dengan  $n$  adalah banyak data yang akan dicluster dan  $m$  adalah banyak variabel. Pada awal iterasi pusat setiap *cluster* bebas (sembarang),  $C_{kj}$  ( $k=1, \dots, k$ ;  $j=1, \dots, m$ ). kemudian dihitung jarak antara setiap data dengan setiap pusat *cluster*.

Untuk melakukan perhitungan jarak data ke-1 ( $x_i$ ) pada pusat *cluster* ke- $k$  ( $c_k$ ) dapat digunakan formula jarak *Euclidean*. Suatu data akan menjadi anggota dari *cluster* ke- $k$  apabila jarak data tersebut ke pusat *cluster* ke- $k$  bernilai paling kecil jika dibandingkan dengan jarak ke pusat kluster lainnya, selanjutnya kelompok data-data yang menjadi anggota pada setiap *cluster*. Nilai pusat *cluster* yang baru dapat dihitung dengan cara mencari nilai rata-rata dari data-data yang menjadi anggota pada *cluster* tersebut seperti yang ditunjukkan pada Persamaan 2.12.

$$c_{kj} = \frac{\sum_{i=1}^p x_{ij}}{p} \quad (2.12)$$

Nilai  $x_{ij} \in$  *cluster* ke- $k$ ,  $p$  = banyak anggota *cluster* ke- $k$ .

## 2.6 Perbaikan Citra

Perbaikan citra dilakukan guna untuk memperbaiki kualitas citra menjadi lebih baik dari sebelumnya, hal ini bertujuan agar menghasilkan citra yang sesuai dengan kebutuhan sistem dan proses pengolahan citra dapat berjalan dengan baik. Ada banyak jenis perbaikan citra yang dilakukan, namun yang dilakukan pada penelitian ini hanya *Laplacian*.

### *Laplacian*

*Laplacian* atau disebut juga metode *Laplacian of Gaussian (LoG)* didapat dengan menentukan bagian tepi citra menggunakan orde turunan kedua, operator *Laplace* menghasilkan kepekaan terhadap *noise* pada tiap bagian piksel dari citra. Turunan dari  $f(x, y)$  ditunjukkan pada Persamaan 2.13.

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.13)$$

Persamaan di atas menghasilkan turunan kedua dari fungsi  $f(x, y)$  yang ditunjukkan pada Persamaan 2.14.

$$\nabla^2 f(x, y) = -4 f(x, y) + f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) \quad (2.14)$$

Persamaan di atas didapat dengan melakukan konvolusi yang dilambangkan dengan bintang (\*) matriks *Laplacian* yang ditunjukkan pada Persamaan 2.15.

$$\nabla^2 f(x, y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} f(x-1, y+1) & f(x, y+1) & f(x+1, y+1) \\ f(x-1, y) & f(x, y) & f(x+1, y) \\ f(x-1, y-1) & f(x, y-1) & f(x+1, y-1) \end{bmatrix} \quad (2.15)$$

Kemudian dilakukan filter *Gaussian* yang ditunjukkan pada Persamaan 2.16.

$$g(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.16)$$

Operator *Laplacian of Gauss* didapat dari konvolusi yang ditunjukkan pada Persamaan 2.17.

$$\nabla^2 g(x, y) = \left( \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right) * e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.17)$$

Dengan  $\sigma$  = nilai standar deviasi distribusi ( $\sigma = 1,4$ ) dan  $e$  = nilai eksponensial (2,71828183) (Junior et al., 2014). Namun filter *Gaussian* tidak digunakan dalam penelitian ini, karena fungsi dari filter *Gaussian* digunakan untuk mengaburkan citra dan menghilangkan *noise* yang jika diterapkan dapat menghilangkan garis-garis keriput halus pada citra.

## 2.7 Ekstraksi Ciri Bentuk

Ekstraksi ciri merupakan tahapan untuk mengetahui ciri dari objek yang akan dicari agar dapat dibedakan dengan objek lainnya. Ciri tersebut akan digunakan sebagai parameter untuk membedakan objek pada tahapan identifikasi dan klasifikasi. Ekstraksi ciri yang umumnya

dilakukan yaitu ekstraksi ciri bentuk, tekstur, warna, dan ukuran. Namun pada penelitian ini, penelitian ini hanya menggunakan ciri bentuk dan luas.

### 2.7.1 Deteksi Tepi

Tepi adalah perubahan nilai intensitas derajat keabuan yang tiba-tiba membesar dalam jarak waktu yang singkat (Hidayatno et al., 2015), tepi dapat mencirikan batas-batas objek dan berguna untuk proses segmentasi dan identifikasi citra. Tujuan deteksi tepi adalah untuk meningkatkan penampakan garis objek atau daerah pada sebuah citra. Terdapat beberapa jenis deteksi tepi yang digunakan, tetapi yang digunakan dalam penelitian ini adalah *Canny*.

#### *Canny*

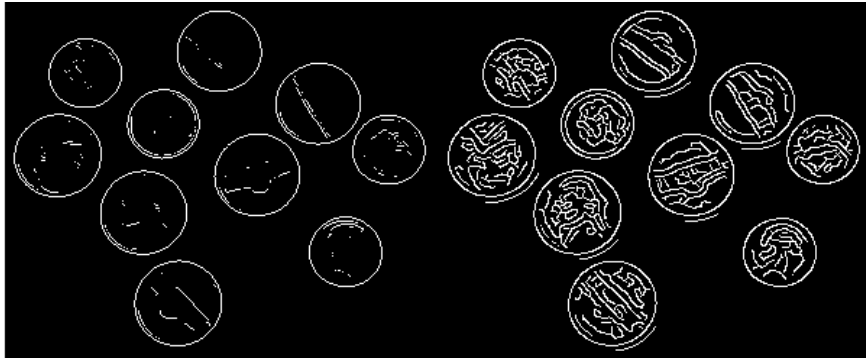
Deteksi tepi *Canny* ditemukan oleh Marr dan Hildreth dalam penelitian mengenai pemodelan persepsi visual manusia dan dikembangkan oleh John F Canny pada tahun 1986. Pendekatan algoritma *Canny* dapat dilakukan dengan konvolusi fungsi gambar dengan operator *Gaussian* dan turunannya (Hidayatno et al., 2015). Berikut turunan pertama dari fungsi citra yang dikonvolusikan (\*) dengan fungsi *Gaussian* yang ditunjukkan pada Persamaan 2.18.

$$g(x, y) = D[gauss(x, y) * f(x, y)] \quad (2.18)$$

Fungsi di atas ekuivalen dengan fungsi citra yang dikonvolusikan (\*) dengan turunan pertama dari fungsi *Gaussian* yang ditunjukkan pada Persamaan 2.19.

$$g(x, y) = D[gauss(x, y)] * f(x, y) \quad (2.19)$$

Kedua fungsi di atas memungkinkan *Canny* dapat menggabungkan tingkat kehalusan dan pendeteksian tepi kedalam suatu konvolusi dalam satu dimensi dengan dua arah yang berbeda yaitu vertikal dan horizontal. Ada beberapa kriteria deteksi tepi paling optimum yang dapat dipenuhi *Canny* di antaranya mampu untuk meletakkan dan menandai semua tepi yang ada sesuai dengan kriteria parameter konvolusi yang dilakukan, memberi fleksibilitas yang sangat tinggi untuk menentukan ketebalan tepi yang diinginkan, mampu melokalisasi dengan baik, dan memberikan respon yang jelas untuk setiap tepi (Ginting, 2010). Gambar perbedaan deteksi tepi dengan metode *Sobel* dan *Canny* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Contoh citra dengan filter *Sobel* (kiri) dan *Canny* (kanan)

Sumber: Matlab

### 2.7.2 Area (Luas)

*Area* atau Luas merupakan salah satu fungsi yang terdapat pada fungsi *Regionprops*. *Region Propertie (Regionprops)* adalah metode yang digunakan untuk mengukur berbagai properti dari setiap *Region* yang telah diberi label dalam matriks label (Pramana, 2014). Ada beberapa fungsi lain yang dihasilkan dari *regionprops* di antaranya *Area*, panjang *Major Axis*, dan panjang *Minor Axis*.

*Area* merupakan banyak piksel sebenarnya yang berada pada wilayah tertentu. Untuk mendapatkan nilai *Area* suatu objek dilakukan dengan memilih objek dengan menggunakan label, nilai dari label tersebut diperoleh dari hasil *Connected-Component* dalam citra *Biner*.

## 2.8 Uji Validasi Sistem

Uji validasi sistem dilakukan dengan metode *Single Decision Threshold*. Metode ini pada dasarnya memuat informasi yang digunakan untuk membandingkan hasil deteksi oleh sistem dengan hasil deteksi nilai sebenarnya oleh ahli. Dalam metode ini terdapat tabel yang mempunyai sejumlah baris data uji yang diprediksi benar dan tidak benar oleh model klasifikasi yang digunakan untuk menentukan kinerja sebuah model klasifikasi. Ada empat istilah yang digunakan dalam *Single Decision Threshold* yaitu:

- a. *True Positif (TP)* adalah apabila nilai sebenarnya dan nilai prediksi menghasilkan hasil positif, contohnya jika nilai sebenarnya “garis keriput”, maka sistem memutuskan “garis keriput”
- b. *True Negative (TN)* adalah jika nilai sebenarnya dan nilai prediksi menghasilkan hasil negatif, contoh jika nilai sebenarnya “bukan garis keriput”, maka sistem memutuskan “bukan garis keriput”

- c. *False Positive (FP)* adalah jika nilai sebenarnya bernilai negatif, tetapi sistem menghasilkan hasil yang positif. Contohnya jika nilai sebenarnya “bukan garis keriput”, tetapi sistem memutuskan “garis keriput”
- d. *False Negative (FN)* adalah jika nilai sebenarnya bernilai positif, tetapi sistem menghasilkan hasil yang negatif. Contohnya jika nilai sebenarnya “garis keriput”, tetapi sistem memutuskan “bukan garis keriput”.

Tabel 2.2 Tabel *confusion matrix*

		Nilai Sebenarnya	
		<i>True</i>	<i>False</i>
Nilai Prediksi	<i>True</i>	<i>TP</i>	<i>FP</i>
	<i>False</i>	<i>FN</i>	<i>TN</i>

Nilai *sensitivity* dan *specivicity* dapat dihitung setelah nilai *True Positive (TP)*, *False Positive (FP)*, *False Negative(FN)* dan *True Negative (TN)* didapatkan. *Sensitivity* adalah parameter untuk mengukur persentase data positif yang teridentifikasi dengan benar (garis keriput yang terdeteksi sistem sama dengan pakar), sedangkan *specificity* adalah paramater untuk mengukur persentase data negatif yang teridentifikasi dengan benar (sistem tidak mendeteksi objek bukan garis keriput dari kandidat) (Shortliffe & Cimino, 2010). Rumus nilai *sensitivity* dan *specivicity* dapat dilihat pada Persamaan 2.20 dan 2.21.

$$Sensitivity = \frac{TP}{TP + FN} \quad (2.20)$$

$$Specificity = \frac{TN}{TN + FP} \quad (2.21)$$

Nilai prediksi adalah nilai yang dihasilkan oleh sistem, sedangkan nilai sebenarnya adalah nilai yang dihasilkan oleh para ahli yaitu Ibu dr. Rosmelia, M.Kes., Sp.KK selaku dokter spesialis kulit dan kelamin serta Bapak Aris Nurul Huda selaku staff PT AVO.

Akurasi sistem dapat dihitung jika *True Positive (TP)*, *False Positive (FP)*, *False Negative FN)* dan *True Negative (TN)* telah didapatkan, akurasi adalah nilai yang menggambarkan seberapa akurat sistem dapat mendeteksi data secara benar (Nur, 2018). Rumus nilai akurasi dapat dilihat pada Persamaan 2.22.

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.22)$$

## 2.9 Penelitian Serupa

Ada beberapa penelitian terkait deteksi keriput yang pernah dilakukan sebelumnya, penelitian pertama dilakukan oleh Hayatin (2016), penelitian ini bertujuan untuk mengklasifikasikan kelompok usia berdasarkan citra wajah dengan menggunakan fitur penting yaitu face anthropometry dan kerutan (*wrinkle*). Kerutan dideteksi menggunakan metode *Sobel* dengan *threshold*, kemudian dihitung lebar dan kedalaman kerutan. Penelitian ini lebih menfokuskan pada klasifikasi kelompok usia, sehingga hasilnya tidak menampilkan fitur kerutan yang didapat.

Penelitian kedua dilakukan oleh Batool & Chellappa (2012), penelitian ini mendeteksi keriput dengan menggunakan proses menandai titik (*MPP*), keriput dilokalisasi dengan mengambil sampel *MPP* menggunakan *Reversible Algoritma Jump Markov Chain Monte Carlo (RJMCMC)*. Penelitian ini tidak melakukan proses pemisahan wajah dengan *background* dan tidak baik digunakan pada citra dengan kerutan yang tidak jelas serta memiliki intensitas gradien yang rendah.

Batool & Chellappa (2014) melakukan penelitian lagi terkait deteksi keriput untuk memperbaiki metode deteksi keriput sebelumnya yang menggunakan metode *MPP*. Dalam penelitian ini Nazre dan Rama menggunakan filter *Gabor* dengan morfologi gambar dan kendala geometris untuk mendeteksi garis-garis keriput pada wajah. Penelitian ini mampu menampilkan garis-garis keriput dalam pada wajah dengan baik namun hanya fokus mendeteksi keriput pada bagian dahi saja.

Dari beberapa penelitian yang telah dilakukan sebelumnya, masalah yang ditemukan dalam tidak ada proses pemisahan wajah dengan *background*, pendeteksian keriput pada wajah belum mampu menampilkan garis-garis keriput pada keseluruhan wajah dan menggunakan citra dengan garis kerutan yang cukup dalam. Perbandingan penelitian yang telah ada dengan penelitian yang akan dirancang dapat dilihat pada Tabel 2.3.

Tabel 2.3 Tabel perbandingan penelitian serupa

Pengembangan	Metode Deteksi Keriput yang digunakan	Pemisahan Objek Kulit Wajah dengan <i>Background</i>	Pengambilan Area Keriput	Mendeteksi Garis Keriput (Dangkal atau Dalam)	Mendeteksi Keriput pada seluruh Wajah	Menampilkan Keriput pada seluruh Wajah
--------------	---------------------------------------	--	--------------------------	---	---------------------------------------	--

Batool & Chellappa (2012)	Proses menandai titik ( <i>MPP</i> )	-	-	Dalam	√	√
Hayatin (2016)	Sobel dengan Threshold	√	√	Dalam	√	-
Batool & Chellappa (2014)	Gabor	-	-	Dalam	-	-
Tugas Akhir ini	Canny dengan Threshold	√	√	Dangkal	√	√

Penelitian yang akan dikerjakan saat ini diharapkan mampu memperbaiki masalah-masalah di atas, di antaranya yaitu melakukan pemisahan objek kulit wajah dengan *background*, pengambilan area keriput untuk meminimalisir terdeteksinya objek *noise*, mampu mendeteksi garis keriput dangkal yang muncul akibat penuaan dini dan menggunakan bagian wajah secara keseluruhan sehingga mampu menampilkan garis keriput di area-area munculnya keriput pada wajah secara utuh.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Pengumpulan Data**

Data yang digunakan dalam penelitian ini terdiri dari beberapa foto dengan ciri kulit keriput. Tiga foto digunakan sebagai data *training* yang bertujuan untuk mengidentifikasi ciri atau karakteristik citra wajah dan garis keriput, hasil dari proses identifikasi ini akan menjadi data acuan untuk menguji data sampel yang lain. Data uji yang digunakan dalam penelitian ini berjumlah 39 foto dengan latar belakang dan posisi wajah menghadap lurus kedepan, foto tersebut didapat secara random dari Internet dengan pertimbangan citra berwarna (*RGB*).

#### **3.2 Tahap Analisis**

Tahap analisis adalah langkah yang diterapkan guna untuk menganalisis kebutuhan dan keperluan terkait penelitian, dari tahap ini dapat diketahui kebutuhan dan tahapan-tahapan yang diperlukan dalam memecahkan masalah terkait tujuan dari penelitian yang akan dilakukan. Tahap analisis yang dilakukan dalam penelitian ini antara lain analisis permasalahan, analisis data, analisis kebutuhan *input*, analisis kebutuhan proses dan analisis kebutuhan *output*.

##### **3.2.1 Analisis Permasalahan**

Tahap analisis permasalahan dilakukan dengan menganalisis permasalahan yang akan diselesaikan dalam penelitian ini, di antaranya memisahkan objek warna kulit dengan *non*-kulit, mengambil area-area keriput, mendeteksi garis keriput yang terdapat pada citra, dan menampilkan hasil deteksi pada citra asli. Tahap ini diterapkan untuk memilih dan menentukan metode yang tepat untuk memecahkan masalah-masalah di atas.

##### **3.2.2 Analisis Data**

Tahap analisis data dilakukan dengan menganalisis citra yang akan diinput ke dalam sistem. Tahap ini diterapkan untuk mengetahui rentang nilai warna yang digunakan untuk memisahkan objek kulit wajah dengan *background*, menentukan metode yang tepat untuk mendeteksi garis keriput, dan memilih rentang luas yang digunakan untuk mengeliminasi garis keriput pada citra.



### 3.2.3 Analisis Kebutuhan *Input*

Dari analisis yang telah dilakukan, kebutuhan *input* yang diperlukan dalam sistem deteksi keriput pada citra wajah ini berupa:

- a. Sebuah file foto warna (*RGB*) yang mengandung citra wajah dengan tipe .jpg, jpeg atau .png
- b. Wajah pada citra harus menghadap lurus ke depan dan memiliki ciri keriput
- c. Pencahayaan pada citra harus merata.

Citra yang digunakan pada penelitian ini diambil secara *random* dari Internet yang terdiri dari citra pria dan wanita. Sebelum citra diinput kedalam sistem, citra melewati proses *Preprocessing* untuk memperbaiki citra yang akan digunakan yaitu berupa tahap pemotongan dan perubahan ukuran citra.

### 3.2.4 Analisis Kebutuhan Proses

Dari analisis yang telah dilakukan, proses yang terdapat pada sistem deteksi keriput pada citra wajah adalah sebagai berikut:

- a. Proses masukan citra digital.
- b. Proses memperkecil ukuran citra untuk mempermudah dan mempercepat proses pengolahan.
- c. Proses pemisahan objek warna kulit dengan *non-kulit*.
- d. Proses membagi area wajah berdasarkan area keriput wajah
- e. Proses ekstraksi ciri bentuk keriput
- f. Proses eliminasi luas garis keriput pada citra
- g. Proses menampilkan hasil deteksi garis keriput kedalam citra asli (citra masukan)
- h. Proses perhitungan tingkat akurasi menggunakan *Single Decision Threshold*.

### 3.2.5 Analisis Kebutuhan *Output*

Dari analisis yang telah dilakukan, kebutuhan *output* yang dari sistem deteksi keriput pada citra wajah ini berupa menampilkan informasi yang berisi letak garis-garis keriput pada citra wajah yang ditandai dengan garis berwarna biru.

### 3.2.6 Analisis Pengujian Sistem

Pengujian sistem dilakukan dengan menggunakan metode *Single Decision Threshold*, pada tabel *Single Decision Threshold* akan dihitung nilai *sensitivity*, *specificity* dan *accuracy* untuk mengetahui tingkat akurasi sistem. Hasil deteksi oleh sistem akan dibandingkan dengan

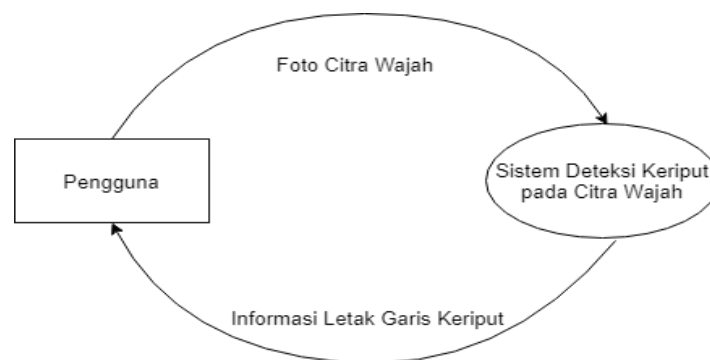
hasil validasi ahli dari dr. Rosmelia, M.Kes., Sp.KK selaku dokter spesialis kulit dan kelamin serta validasi dari pihak PT AVO.

### 3.3 Perancangan Perangkat Lunak

Pada tahap perancangan perangkat lunak ini dibuat rancangan sementara berupa *prototype* sistem secara detail. Metode perancangan yang digunakan adalah dengan menggunakan diagram konteks dan *Flowchart* umum, hal ini bertujuan agar pembuatan sistem lebih terstruktur dan mudah untuk diselesaikan.

#### 3.3.1 Diagram Konteks

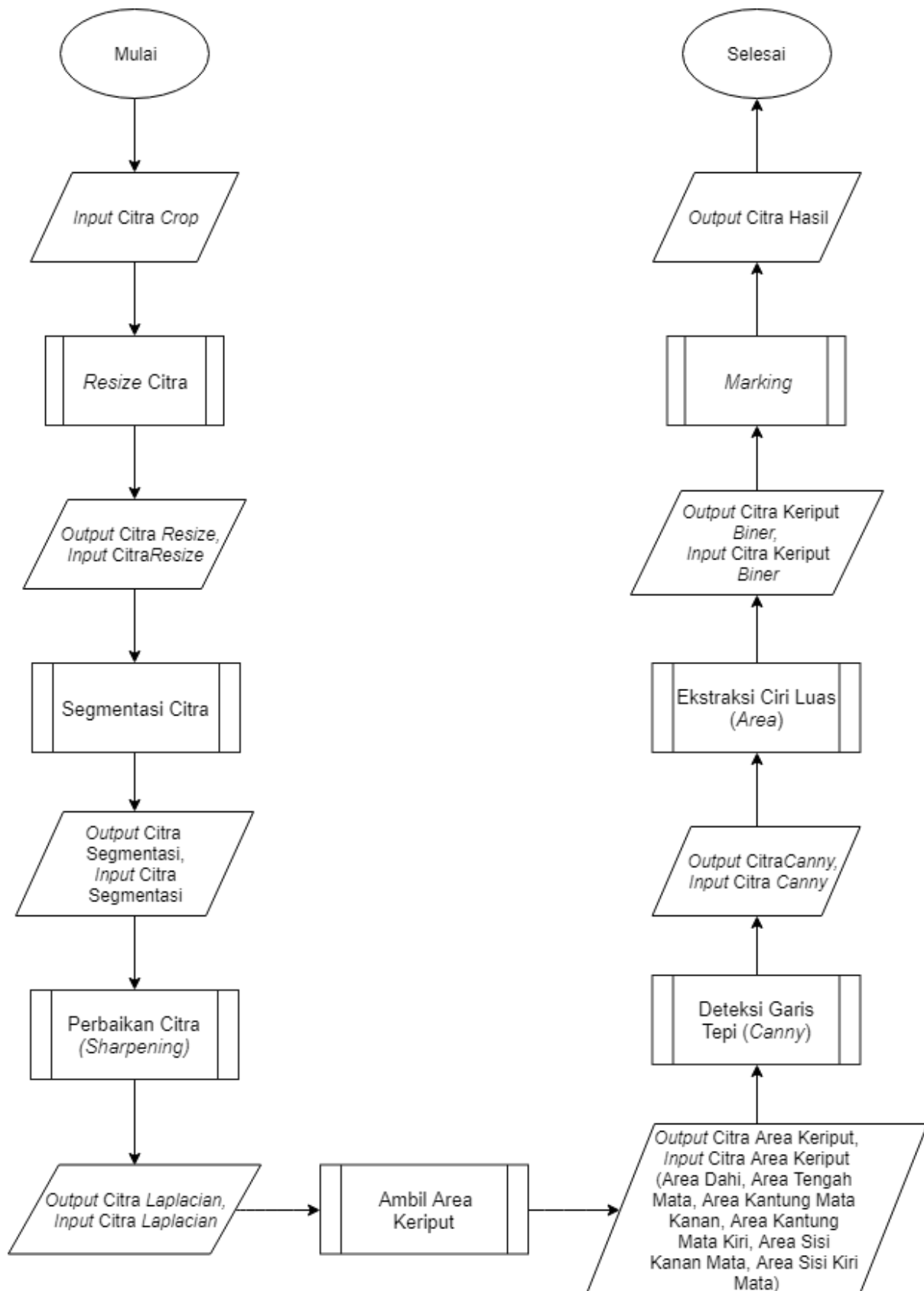
Diagram konteks digunakan untuk menggambarkan proses interaksi sistem dengan pengguna secara umum. Pada sistem ini diperlukan masukan dari pengguna berupa file foto yang memuat wajah pada citra dengan syarat menghadap lurus ke depan dan memiliki ciri garis keriput, selanjutnya file foto akan diproses menggunakan metode pengolahan citra yang meliputi operasi segmentasi citra, perbaikan citra, dan ekstraksi ciri. Hasil dari proses tersebut berupa citra wajah dengan garis-garis keriput yang sudah ditandai. Diagram konteks dari sistem yang akan dikembangkan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram konteks

#### 3.3.2 Gambaran Umum Sistem

Gambaran umum sistem deteksi keriput pada citra wajah terdapat pada Gambar 3.2.



Gambar 3.2 Gambaran umum sistem

Gambar 3.2 menjelaskan langkah pertama yang dilakukan oleh sistem adalah memilih gambar atau citra yang akan diidentifikasi. Citra yang dimasukkan adalah citra yang sudah mengalami proses pemotongan gambar (*Cropping*) sebelumnya, agar pengambilan area keriput yang akan dideteksi sesuai dengan ketentuan pada sistem, selanjutnya citra hasil pemotongan gambar diinput ke dalam sistem. Pada tahap *Preprocessing*, sistem akan melakukan proses perubahan ukuran citra (*Resize*) untuk mengecilkan ukuran citra, kemudian dilakukan segmentasi citra untuk memisahkan objek warna kulit dengan *background* menggunakan beberapa metode segmentasi citra yang akan menghasilkan citra dengan warna kulit saja. Citra akan mengalami perbaikan untuk mempertajam garis-garis wajah pada citra, hasil citra penajaman akan dibagi untuk mengambil area-area keriput yang telah ditentukan. Semua area keriput akan diproses melalui ekstraksi ciri dengan menggunakan metode deteksi tepi dan eliminasi luas (*Area*) garis keriput, hasil dari ekstraksi ciri tersebut akan ditampilkan dan diberi penandaan (*Marking*) pada citra asli.

### 3.3.3 *Preprocessing*

*Preprocessing* citra terdiri atas beberapa proses yang bertujuan untuk memperbaiki citra yang akan digunakan. Proses-proses tersebut di antaranya sebagai berikut:

#### ***Cropping***

Tahap *Cropping* dilakukan untuk memotong baris dan kolom citra, agar saat proses pengambilan area keriput di dalam sistem Matlab, area yang diambil sesuai dengan area yang telah ditentukan. Ukuran pemotongan citra tidak selalu sama, menyesuaikan citra yang akan digunakan, hal ini dikarenakan masing-masing citra mempunyai tinggi dan lebar *background* yang berbeda-beda.

#### ***Resize***

Tahap *Resize* dilakukan untuk menyamakan ukuran citra masukan yang akan diproses dalam sistem Matlab. Ukuran citra yang diubah dapat berubah menjadi lebih kecil atau bahkan lebih besar dari file aslinya. Perubahan ukuran citra dilakukan untuk mengecilkan ukuran citra agar sistem mampu memproses citra dengan cepat.

### 3.3.4 **Segmentasi Citra**

Segmentasi citra terdiri atas beberapa proses yang bertujuan untuk memisahkan objek warna kulit dengan non-kulit. Ada beberapa tahap yang dilakukan dalam proses ini, tahap

pertama adalah mengkonversi warna pada citra, sehingga hasil yang didapat adalah citra dengan warna kulit saja. Untuk memperbaiki tahap pertama proses segmentasi citra dilakukan proses pengelompokan warna pada citra, citra akan dibagi dalam beberapa kelompok sesuai dengan rentang warnanya, kelompok dengan nilai terbesar merupakan kelompok citra dengan warna kulit yang akan disimpan dan ditampilkan hasilnya. Tahap terakhir untuk menyempurnakan proses segmentasi citra dilakukan tahap penutupan celah-celah lubang yang masih terlihat pada citra, hasil dari ketiga tahap di atas berupa citra dengan objek yang tergolong warna kulit saja.

### 3.3.5 Perbaikan Citra

Setelah proses segmentasi citra selesai, dilakukan proses perbaikan citra dengan mempertajam garis-garis pada wajah, hal ini bertujuan agar garis-garis yang termasuk keriput halus terlihat jelas dan mudah untuk dideteksi.

### 3.3.6 Ekstraksi Ciri

Ekstraksi ciri terdiri atas beberapa tahap yang bertujuan untuk mendeteksi garis keriput pada citra wajah. Tahap pertama dilakukan proses pengambilan area keriput, area keriput yang diambil disesuaikan dengan area keriput yang telah ditentukan pada penelitian sebelumnya, yaitu area dahi, area sisi tengah mata, area sisi mata kiri dan kanan, serta area kantung mata kiri dan kanan. Setelah area keriput didapatkan, pada tahap kedua masing-masing area diterapkan metode deteksi garis tepi untuk mendeteksi garis keriput pada area tersebut, masing-masing area akan diberi *threshold* yang berbeda dikarenakan setiap citra mempunyai ciri garis tepi yang berbeda. Tahap terakhir dilakukan tahap eliminasi yang dilakukan untuk mengeliminasi garis wajah berdasarkan *Area* atau luas garis. Eliminasi luas (*Area*) garis wajah dilakukan dengan cara mengambil nilai luas garis yang termasuk garis keriput, sehingga hasil yang didapat berupa garis keriput saja tanpa ada *noise* dan garis bingkai objek.

### 3.3.7 Marking

Setelah garis-garis keriput terdeteksi, dilakukan proses *Marking* atau penandaan pada citra asli, agar hasil bisa dilihat oleh pengguna.

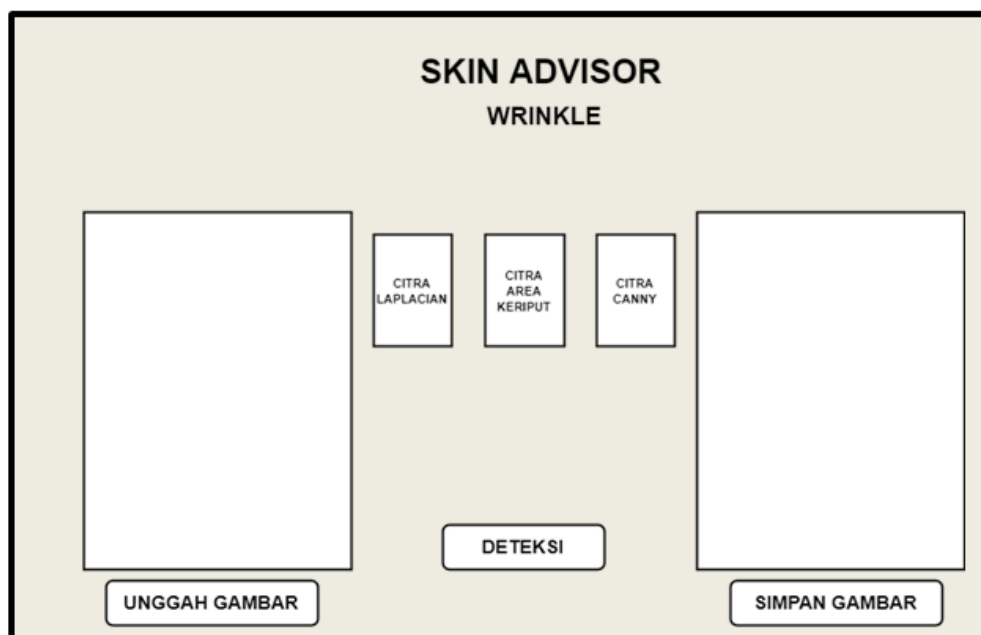
## 3.4 Hasil Perancangan Antarmuka

Sistem dilihat dari sudut pandang pengguna, masukan dari sistem ini adalah *file* citra RGB dengan format JPG, JPEG dan PNG. Pengguna dapat memilih sendiri *file* citra yang akan

digunakan, *file* citra dapat dipilih melalui sebuah kotak dialog yang mengarah pada *drive* atau *folder* penyimpanan citra. *Output* dari sistem deteksi keriput ini berupa lokasi-lokasi garis keriput yang berhasil dideteksi yang ditandai dengan garis berwarna biru. Hasil citra yang sudah diberi penanda sebagai garis keriput dapat disimpan pada sebuah *file*, sehingga proses deteksi tidak perlu diulang apabila ingin melihat hasilnya kembali.

### 3.4.1 Perancangan Halaman Utama

Halaman utama sistem ini menampilkan beberapa *button* untuk membantu proses kerja sistem, dua kotak *preview* besar untuk menampilkan citra sebelum dan sesudah diproses, tiga kotak *preview* kecil untuk menampilkan proses deteksi keriput dari citra asli, dan di setiap kotak *preview* diberi keterangan citra atau proses, agar tidak membingungkan pengguna. Perancangan halaman utama sistem dapat dilihat pada Gambar 3.3.



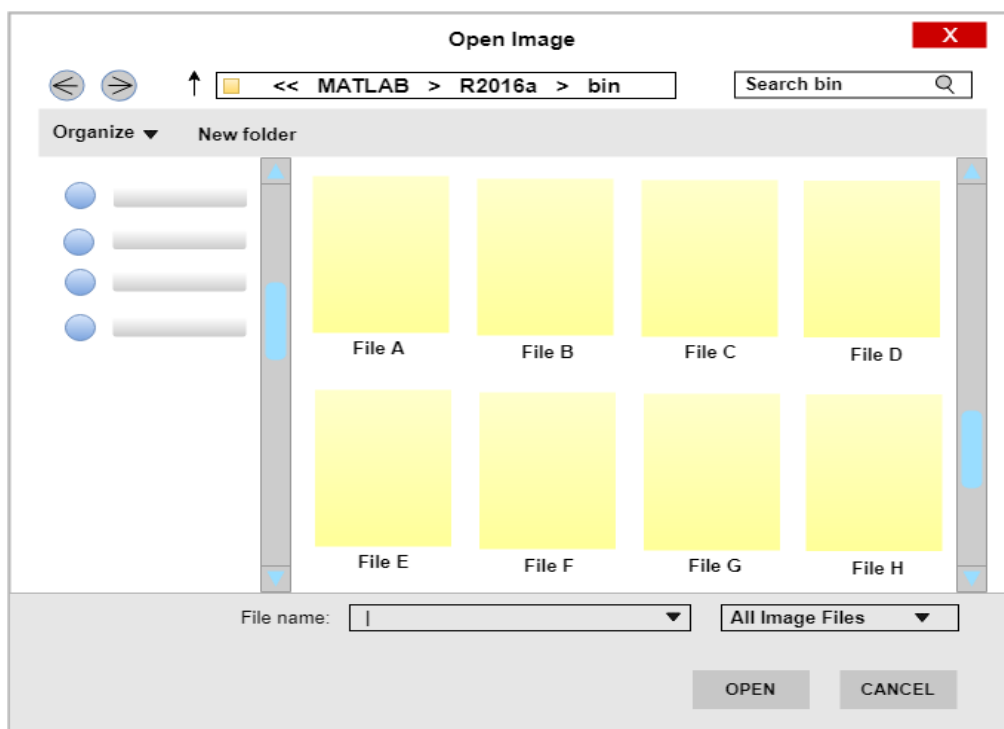
Gambar 3.3 Perancangan Halaman Utama Sistem

*Button Unggah Gambar* berfungsi untuk membantu mengarahkan pengguna ke suatu *drive* atau *folder* yang terdapat *file* citra foto yang akan diproses. *File* citra tersebut akan di-*load* dalam kotak *preview* citra asli, selanjutnya *button Deteksi* digunakan untuk menjalankan perintah deteksi keriput pada citra yang telah diunggah. Masing-masing hasil dari proses yang dilakukan akan ditampilkan dalam kotak *preview* **Citra Laplacian**, **Citra Area Keriput** dan **Citra Canny**. Setelah semua proses dilakukan, hasil citra yang telah dideteksi akan

ditampilkan dalam kotak *preview* **Citra Hasil Deteksi**. Untuk menyimpan citra yang telah dideteksi gunakan *button* **Simpan Gambar**, dan pengguna akan diarahkan ke *drive* atau *folder* penyimpanan yang diinginkan.

### 3.4.2 Perancangan *FileChooser*

Halaman *filechooser* digunakan untuk membantu pengguna dalam mencari dan memilih *file* foto yang akan diproses oleh sistem. Pada halaman ini terdapat informasi tentang nama *file*, tipe *file*, dan lokasi di mana *file* tersebut disimpan. Perancangan halaman *filechooser* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Perancangan Halaman *FileChooser*

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Data Citra

Data citra yang digunakan dalam penelitian ini merupakan citra yang didapat secara *random* dari Internet. Citra yang diambil harus memuat citra wajah wanita atau pria dengan kisaran umur 20 sampai 40 tahun dan memiliki ciri keriput dengan format .jpg, .jpeg atau png. Citra yang diambil sebagai data uji berjumlah 39 citra, contoh citra yang digunakan dapat dilihat pada Gambar 4.1.



Gambar 4.1 Contoh citra

#### 4.2 Preprocessing

Proses-proses yang terdapat pada modul ini adalah sebagai berikut:

##### 4.2.1 Cropping Citra

Tahap *Cropping* dilakukan untuk memotong baris dan kolom citra, agar saat proses pengambilan area keriput di dalam sistem Matlab, area yang diambil sesuai dengan area yang telah ditentukan. Gambaran alur proses *Cropping* citra tidak digambarkan kedalam *Flowchart* karena proses dilakukan di luar sistem, proses *Cropping* dilakukan dengan menggunakan *tools Crop* yang terdapat pada aplikasi Paint. Ukuran pemotongan citra tidak selalu sama, menyesuaikan citra yang akan digunakan. Hal ini dikarenakan masing-masing citra mempunyai tinggi dan lebar *background* yang berbeda-beda, contoh hasil pemotongan citra terdapat pada Gambar 4.2.





(a)

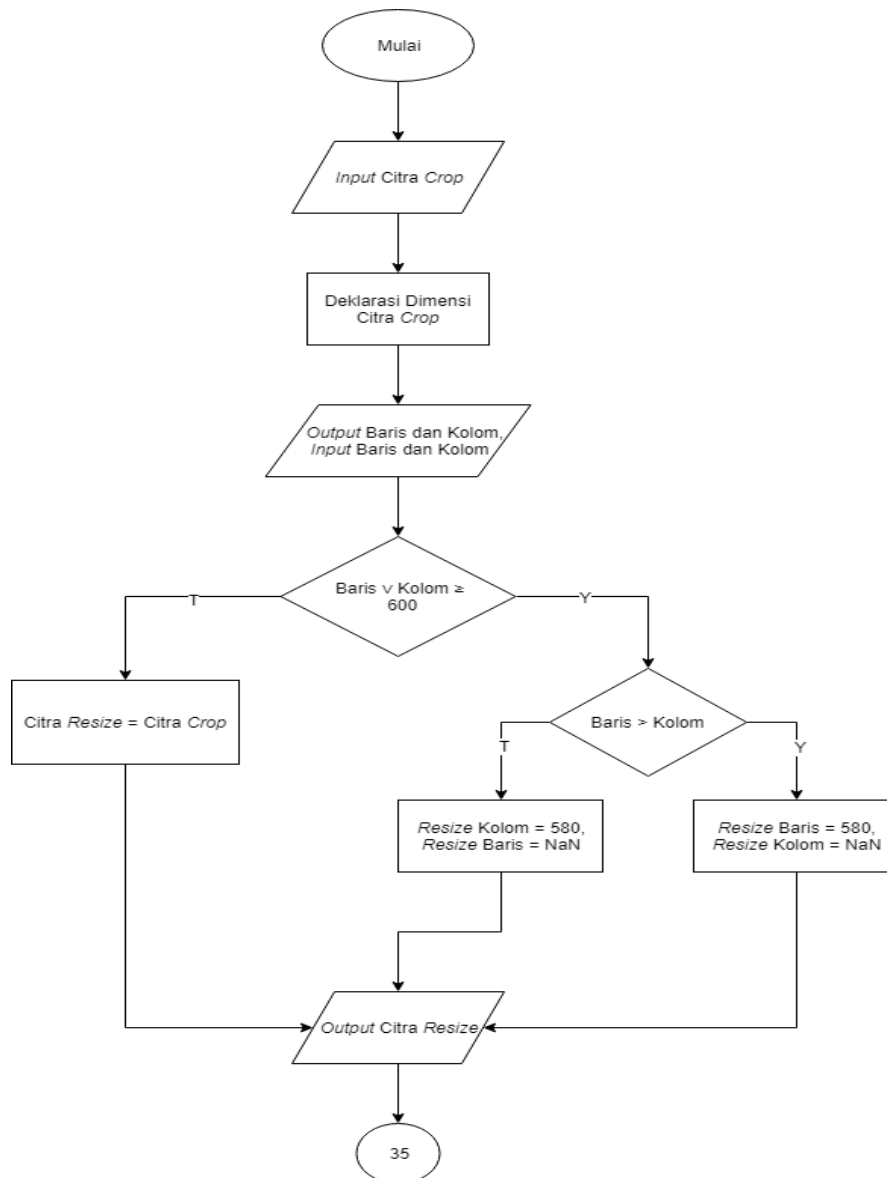


(b)

Gambar 4.2 Hasil citra *Crop*: (a). Citra sebelum di*Crop* (b). Citra sesudah *Crop*

#### 4.2.2 *Resize*

Sebelum diproses ketahap selanjutnya, citra harus melewati proses *Resize* guna untuk memperkecil ukuran citra, perubahan ukuran citra dilakukan untuk mengecilkan ukuran citra agar komputer mampu memproses citra dengan cepat. Ukuran citra yang digunakan pada penelitian ini adalah 580x580, ukuran ini dipilih agar ukuran citra tidak terlalu besar yang akan menyebabkan proses deteksi berjalan lama atau ukuran citra terlalu kecil yang menyebabkan citra akan pecah saat diproses sistem, sehingga garis-garis wajah terutama garis keriput tidak terlihat jelas. *Flowchart* dan *Sintaks* proses *Resize* dapat dilihat pada Gambar 4.3 dan 4.4.

Gambar 4.3 Diagram alur proses *Resize* citra

```

% %===== Resize Citra =====%

[bar kol dlm] = size(citra);

if (bar > kol)
    maxLength = bar;
    if (maxLength >= 600);
        citra = imResize(citra, [580 NaN]);
    end;
else
    maxLength = kol;
    if (maxLength >= 600);
        citra = imResize(citra, [NaN 580]);
    end;
end;

```

Gambar 4.4 Sintaks proses *Resize* citra

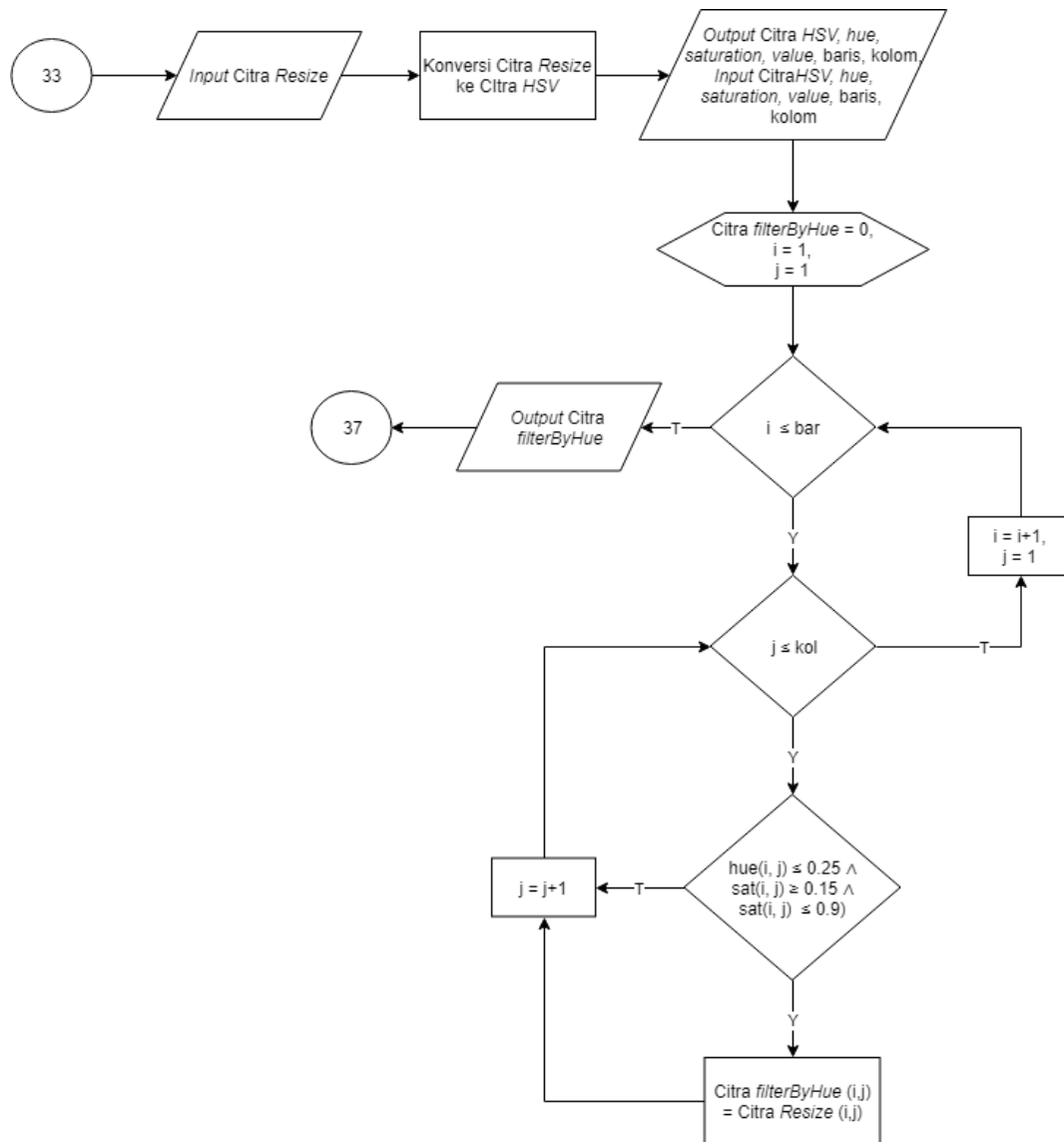
### 4.3 Segmentasi Citra

Segmentasi citra dilakukan untuk memisahkan objek warna kulit dengan *non*-kulit, proses-proses yang terdapat pada modul ini adalah sebagai berikut:

#### 4.3.1 Segmentasi Warna Kulit dan Pengembangan (*Threshold*)

Proses ini dilakukan untuk membedakan citra kulit dan *non*-kulit pada citra wajah. Citra wajah *RGB* diubah terlebih dahulu menjadi citra *HSV* dan diberi *threshold*, konversi warna *HSV* dipilih karena konversi warna yang paling baik untuk mendeteksi warna kulit dalam keadaan pencahayaan normal (Nugraha et al., 2017), *HSV* akan memisahkan *luminance* (cahaya) terhadap komponen warna mengingat terdapat pencahayaan yang berbeda-beda pada setiap warna kulit orang akibat cahaya yang ada disekitarnya, hal ini tidak dapat dilakukan *RGB* karena komponen cahaya dan warna yang terdapat citra akan dipadukan menjadi satu.

*Threshold* citra *HSV* yang mengindikasikan kulit berkisar antara 0.0-0.25 untuk nilai *H* (*hue*), dan 0.15-0.9 untuk nilai *S* (*saturation*), rentang nilai diambil karena merupakan nilai *HSV* yang mengindikasikan warna kulit Asia (Mujahidin, 2012). Hal ini terbukti rentang nilai tersebut cocok dan bisa digunakan pada semua citra uji, jika nilai diubah akan terdapat beberapa bagian dari citra yang berlubang atau tidak terdeteksi. Nilai *V* (*value*) tidak digunakan pada proses ini, karena warna kulit *non*-hitam dan *non*-putih hanya menggunakan *H* dan *S*, nilai *H* dan *S* yang diterima sudah mewakili warna kulit manusia (Mulyani & Propeliena, 2014). Citra *non*-kulit akan direpresentasikan dengan piksel hitam, sedangkan untuk citra kulit akan direpresentasikan dengan piksel putih. *Flowchart*, *sintaks* dan hasil proses segmentasi citra dengan *HSV* dapat dilihat pada Gambar 4.5 sampai 4.7.



Gambar 4.5 Diagram alur proses segmentasi citra dengan *HSV*

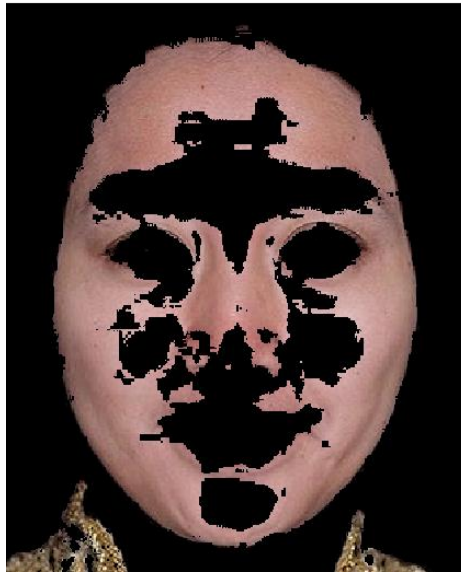
```

%===== RGB TO HSV =====%
[bar kol dlm] = size(citra);

citrahsv = rgb2hsv(citra);

hue = citrahsv(:, :, 1);
sat = citrahsv(:, :, 2);
val = citrahsv(:, :, 3);
filterByHue = uint8(zeros(bar, kol, dlm));
for i = 1 : bar
    for j = 1 : kol
        if (hue(i, j) <= 0.25 && sat(i, j) >= 0.15 && sat(i, j) <= 0.9)
            filterByHue(i, j, :) = citra(i, j, :);
        end
    end
end
end
  
```

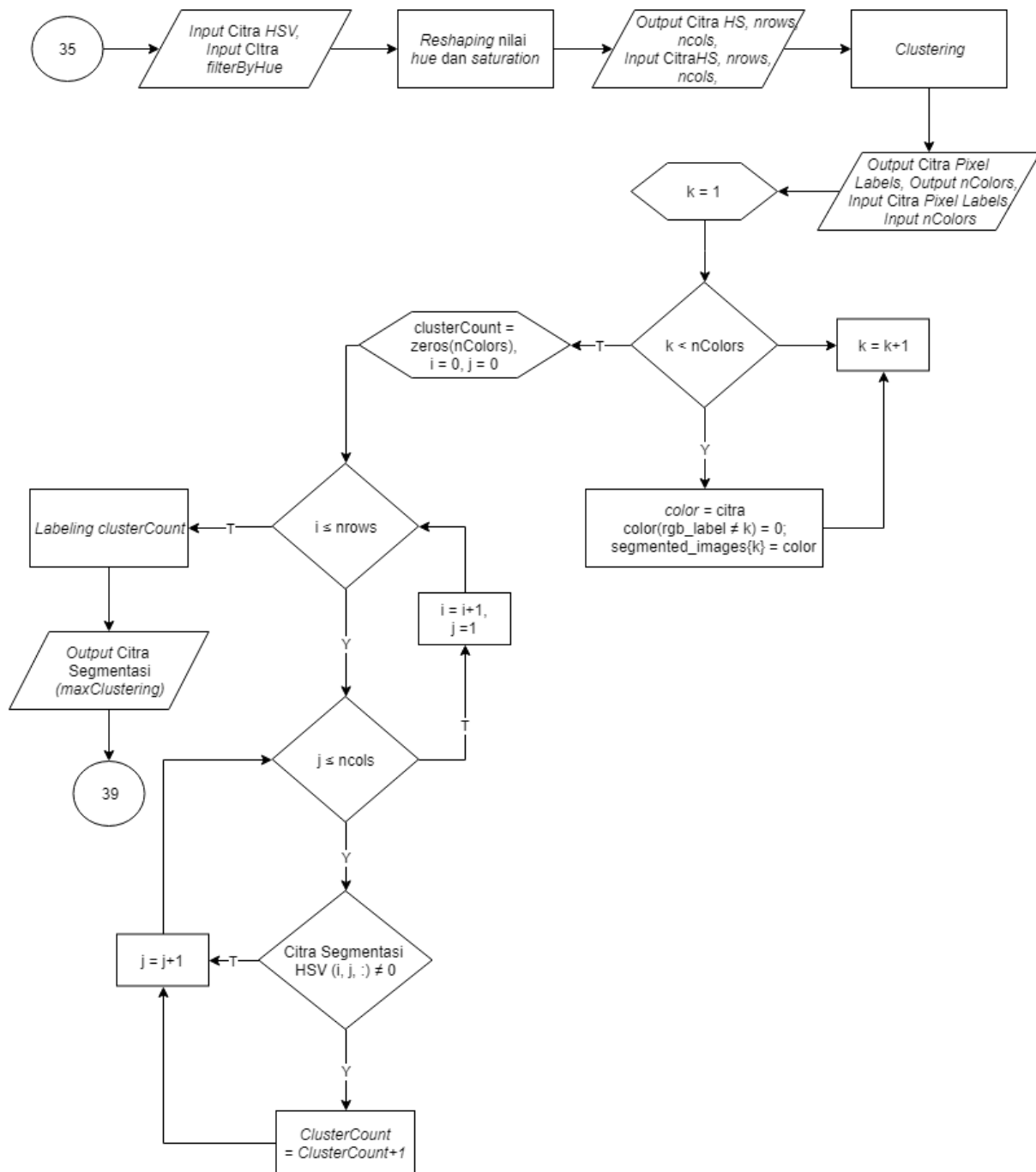
Gambar 4.6 *Sintaks* proses segmentasi citra dengan *HSV*



Gambar 4.7 Hasil segmentasi citra dengan *HSV*

Untuk menghasilkan citra kulit yang lebih baik, dilakukan proses segmentasi citra dengan metode *K-Means Clustering*. Proses ini akan melakukan pengelompokan objek sesuai dengan rentang warnanya, dalam penelitian ini citra hasil segmentasi *HSV* akan dibagi menjadi dua *cluster* dengan *centroid* yang akan ditentukan secara random dari data yang dimiliki citra. Dua *cluster* dipilih untuk memisahkan antara warna objek kulit dengan *non-kulit* seperti *background* dan warna rambut. Setelah *cluster* terbentuk, diambil nilai *cluster* yang memiliki anggota terbanyak untuk disimpan dan ditampilkan sebagai citra hasil segmentasi dengan *K-Means*.

Metode *K-Means* dipilih karena merupakan metode yang memiliki nilai akurasi lebih baik daripada metode *Decision Tree* (Valentino et al., 2017). Selain itu, proses *clustering* dapat dipakai saat tidak mengetahui bagaimana data tersebut harus dikelompokkan. *Flowchart*, *sintaks* dan hasil proses segmentasi dengan metode *K-Means Clustering* dapat dilihat pada Gambar 4.8 sampai 4.10.



Gambar 4.8 Diagram alur proses segmentasi citra dengan *K-Means Clustering*

```

%===== Clustering K-means =====%

%1. Mengambil nilai h dan s
hs = double(citraHSV(:,:,1:2));
nrows = size(hs,1);
ncols = size(hs,2);
hs = reshape(hs,nrows*ncols,2);

%2. Menentukan banyak Cluster
nColors = 2;
[cluster_idx, cluster_center] =
kmeans(hs,nColors,'distance','sqEuclidean','Replicates',2);

```

```

piksel_labels = reshape(cluster_idx,nrows,ncols);
RGB = label2rgb(piksel_labels);

%3. Menampilkan hasil segmentasi clustering
segmented_images = cell(1,2);
rgb_label = repmat(piksel_labels,[1 1 3]);

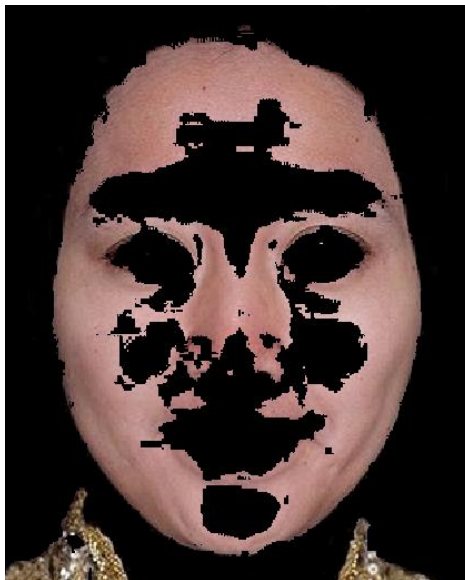
for k = 1:nColors
    color = citra;
    color(rgb_label ~= k) = 0;
    segmented_images{k} = color;
end

clusterCount = zeros(nColors);
for i = 1:nrows
    for j = 1:ncols
        if filterByHue(i, j, :) ~= 0
            clusterCount(piksel_labels(i,j)) =
clusterCount(piksel_labels(i,j)) + 1;
        end
    end
end

%4. Mengambil nilai cluster yang mempunyai objek terbanyak(maks)
[maxVal maxClusterIndex] = max(clusterCount);

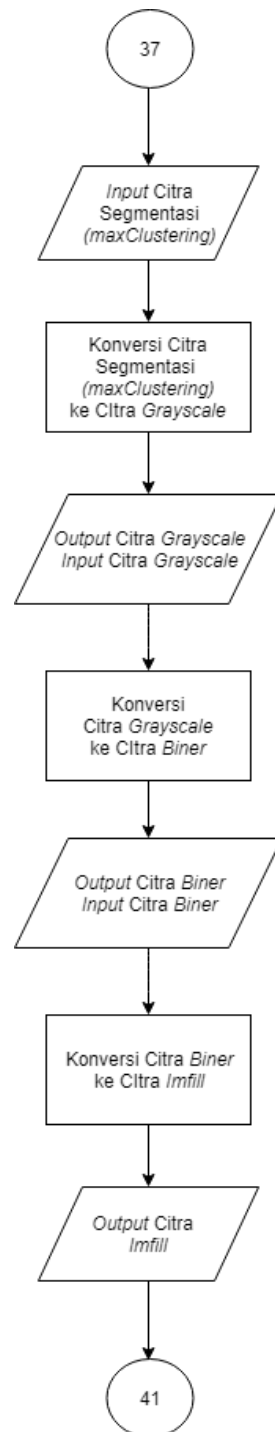
```

Gambar 4.9 Sintaks proses segmentasi citra dengan *K-Means Clustering*



Gambar 4.10 Hasil segmentasi citra dengan *K-Means Clustering*

Untuk menyempurnakan hasil citra dari proses segmentasi *K-Means* yang masih terdapat celah atau lubang, citra diberi filter *Imfill* untuk menutup lubang-lubang yang terdapat pada citra hasil segmentasi. *Flowchart*, *sintaks* dan hasil proses filter *Imfill* dapat dilihat pada Gambar 4.11 sampai 4.13.

Gambar 4.11 Diagram alur proses *Imfill*

```

%===== Proses Imfill =====%

citraRgb = rgb2gray(segmented_images{maxClusterIndex(1)});
level = graythresh(citraRgb);
citraImfil = im2bw(citraRgb, level);
citraImfil = imfill(citraImfil, 'holes');

```

Gambar 4.12 Sintaks proses filter *Imfill*

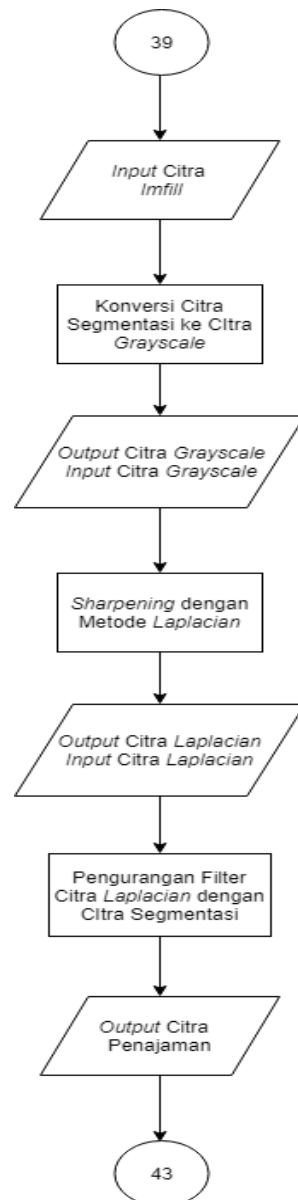




Gambar 4.13 Citra hasil filter *Imfill*

#### 4.4 Perbaikan Citra

Proses perbaikan citra dilakukan untuk mempertajam citra hasil segmentasi yang diproses dengan metode *Laplacian of Gaussian (LoG)*, tetapi pada penelitian ini *Gaussian* tidak digunakan karena akan memberi efek *blur* (buram) pada citra. Citra hasil proses *Imfill* akan diberi filter *Laplacian* dengan nilai *Alpha Laplace default* sebesar 0.2 untuk mempertajam garis-garis pada citra wajah, nilai *Alpha Laplace default* dipilih karena sudah dapat mengangkat detail citra dengan baik tanpa ada gangguan *noise* yang berarti. Untuk melihat hasil penajaman dilakukan pengurangan citra asli dengan citra hasil filter *Laplacian*. *Flowchart*, *sintaks* dan hasil proses pemberian filter *Laplacian* dapat dilihat pada Gambar 4.14 sampai 4.16.



Gambar 4.14 Diagram alur proses penajaman citra

```

%===== Laplacian =====%

sharpFilter = fspecial('laplacian', 0.2);
sharp = imfilter(fullwajahGray, sharpFilter, 'replicate');

%===== Laplace To Logical =====%

fullwajahL = logical(sharp);
figure,imshow(fullwajahL), title('Full Wajah');

%===== Pengurangan Citra Gray & Citra Laplace =====%

pengurangan = (fullwajahGray)-(sharp);
  
```

Gambar 4.15 Sintaks proses filter Laplacian



Gambar 4.16 Hasil perbaikan citra: (a). Citra *Laplacian*  
(b). Hasil pengurangan citra asli dengan filter *Laplacian*

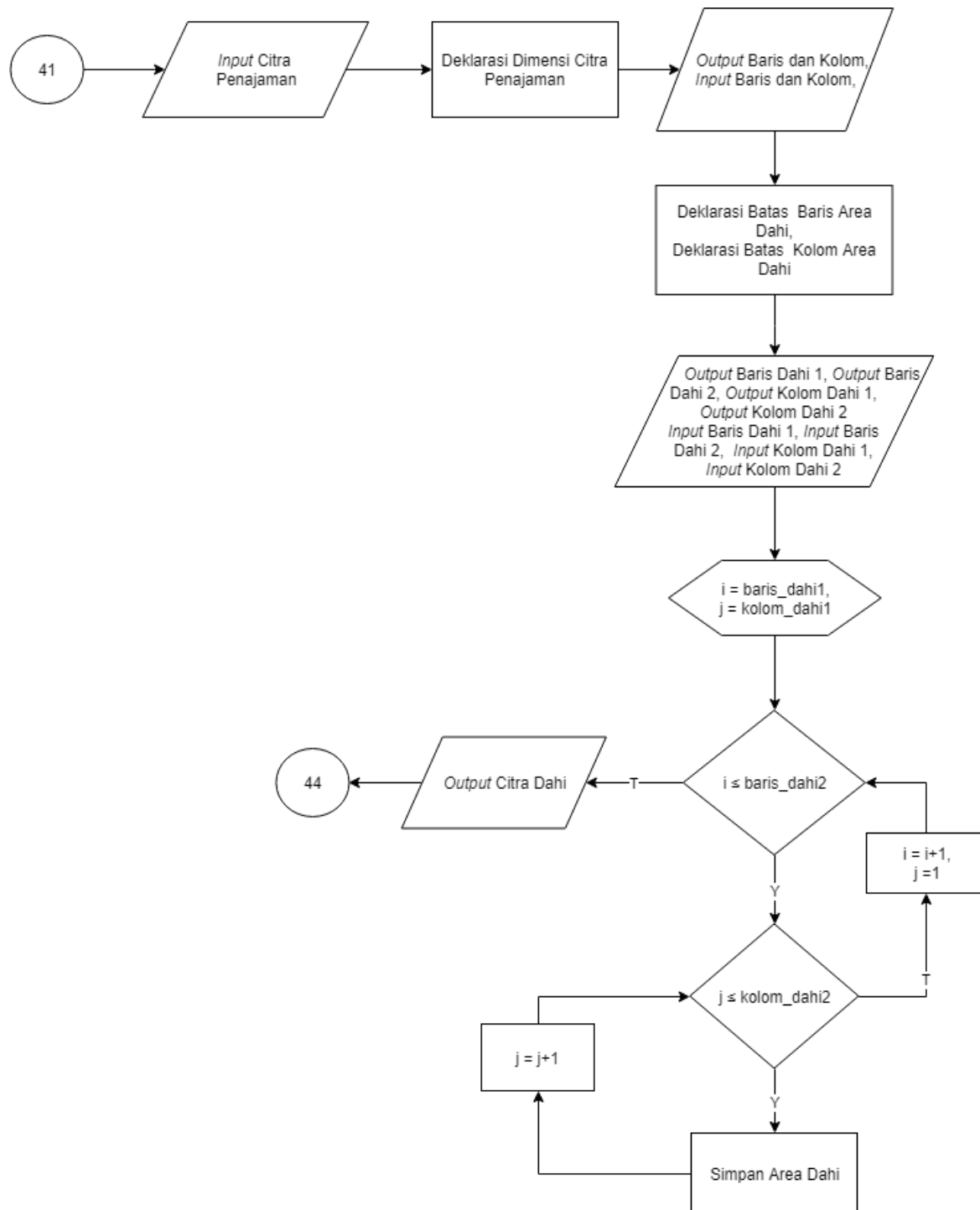
## 4.5 Ekstraksi Ciri

Proses-proses yang terdapat pada modul ini adalah sebagai berikut:

### 4.5.1 Area Keriput

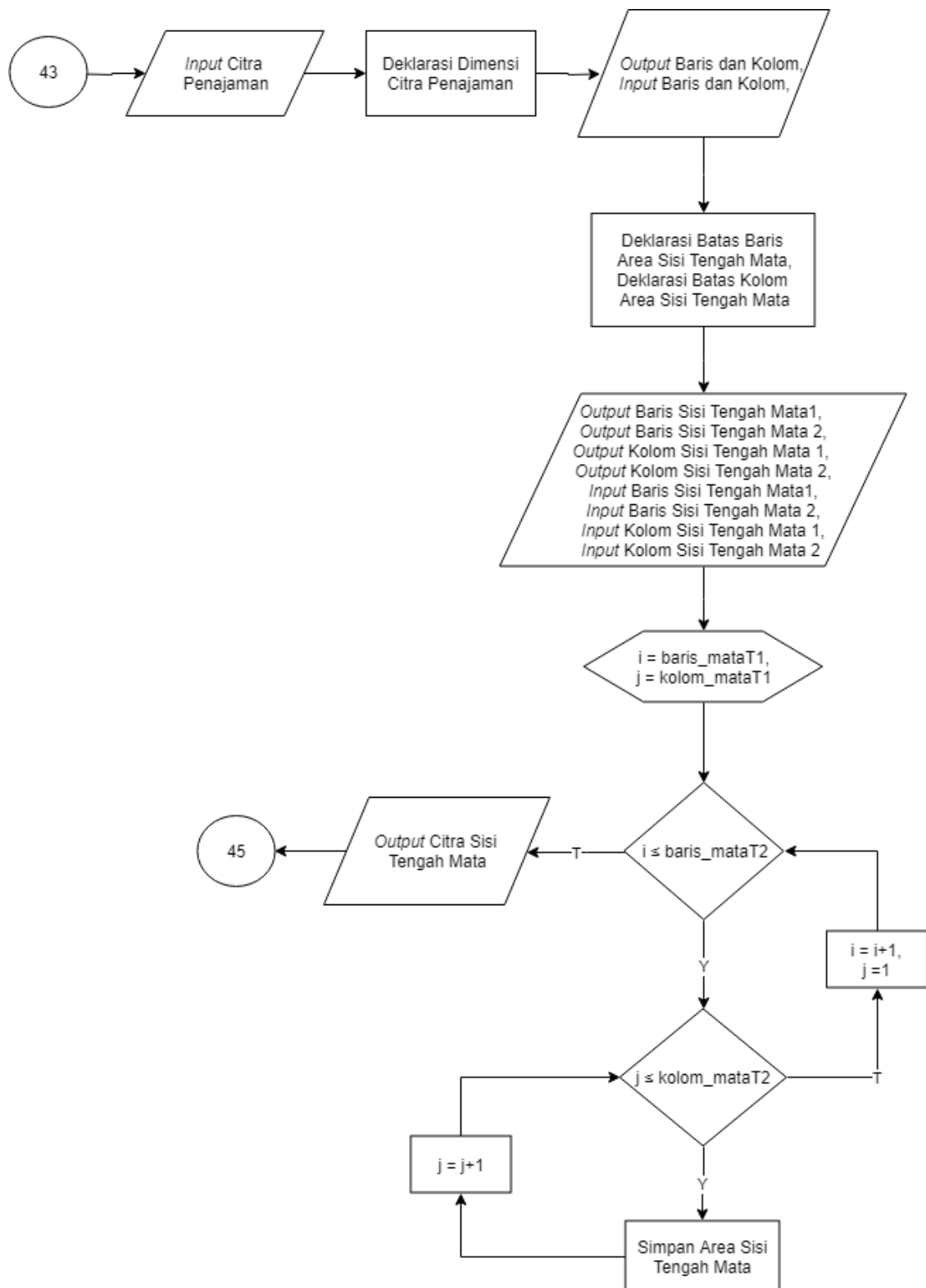
Pemilihan area-area keriput disesuaikan dengan penelitian yang telah dilakukan sebelumnya, pengambilan area keriput dilakukan dengan metode *Persentase*, metode ini dipilih karena merupakan metode yang paling mudah diterapkan dan disesuaikan dengan citra. Metode lain yang pernah dilakukan sebelumnya antara lain metode *Active Contour*. Metode ini mampu mengambil area keriput namun tidak sempurna karena tepi area keriput yang diambil tidak rapi. Metode lain yaitu pemotongan gambar dengan fitur *Crop* pada Matlab juga berhasil mengambil area keriput dengan baik, namun akan sulit pada tahap penggabungan area keriput kembali.

Area-area keriput yang diambil adalah area dahi, area sisi tengah mata, area sisi mata kiri dan kanan, serta area kantung mata kiri dan kanan. Pada gambar area seluruh wajah diberi nilai *persentase* sebesar 100%, kemudian ditentukan baris awal, baris akhir area, kolom awal area dan kolom akhir area. *Flowchart*, *sintaks* dan hasil proses pengambilan area keriput dapat dilihat pada Gambar 4.17 sampai 4.24.



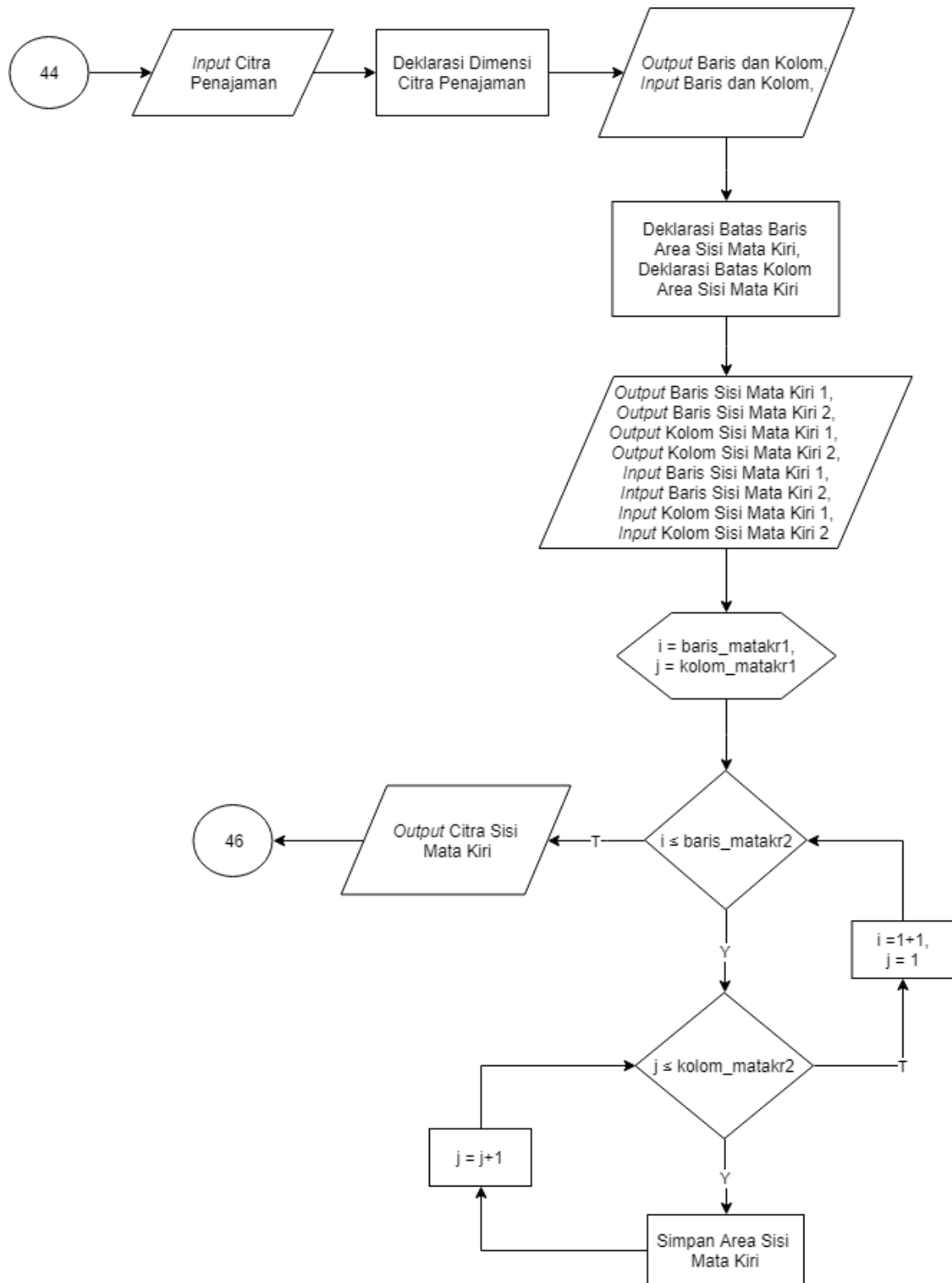
Gambar 4.17 Diagram alur proses pengambilan area dahi

Untuk baris area dahi diberi nilai *persentase* 20-30 dari 100% dan kolom area dahi 25-70 dari 100%. Nilai ini dipilih agar area dahi yang diambil tidak terlalu luas, sehingga akan mengenai rambut yang terdapat pada bagian atas, kiri dan kanan dahi serta rambut alis pada wajah.



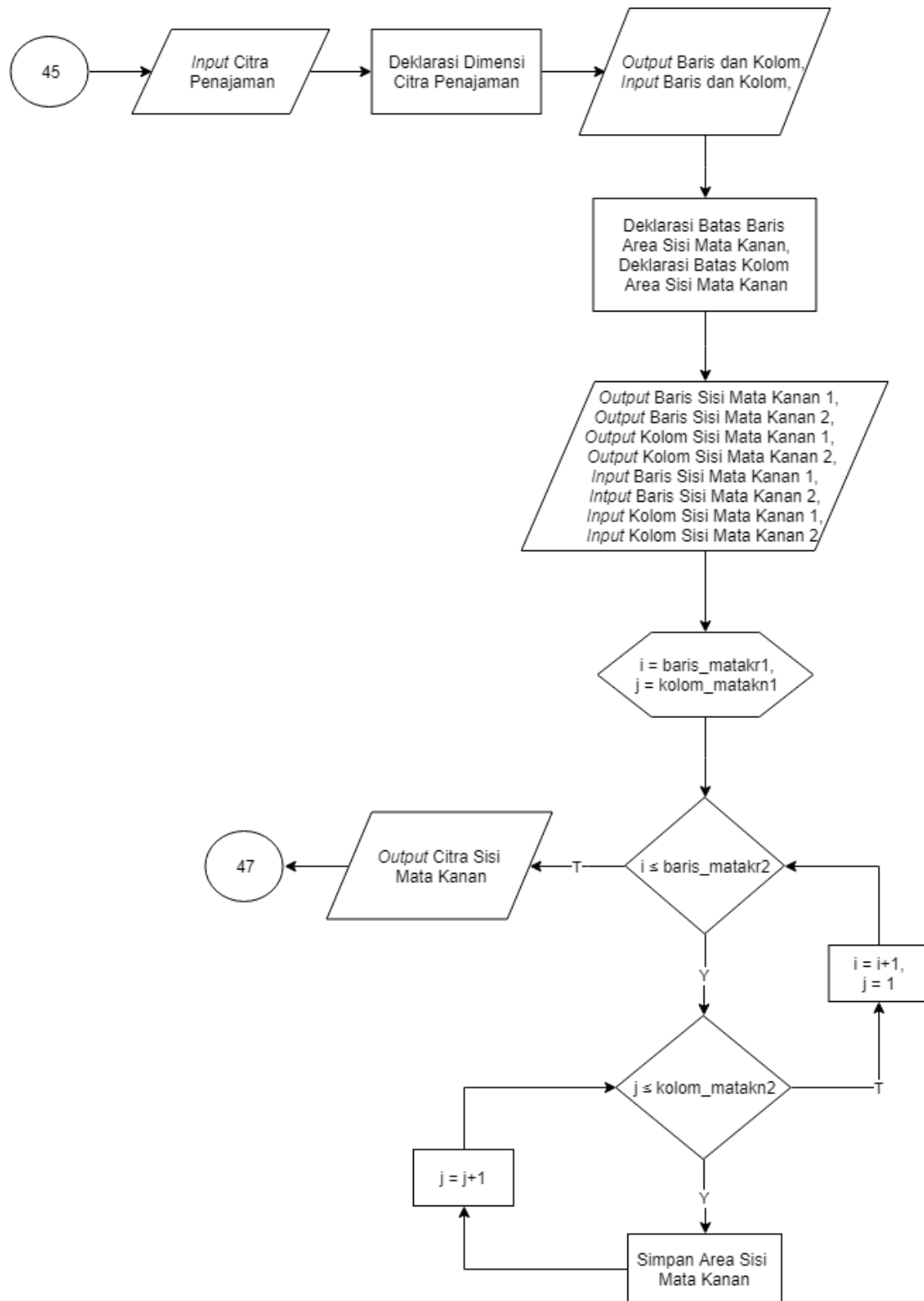
Gambar 4.18 Diagram alur proses pengambilan area sisi tengah mata

Untuk baris area sisi tengah mata diberi nilai *persentase* 34-43 dari 100% dan kolom area sisi tengah mata 44-55 dari 100%. Nilai ini dipilih agar area sisi tengah mata yang diambil tidak terlalu luas, sehingga akan mengenai bagian sudut mata.



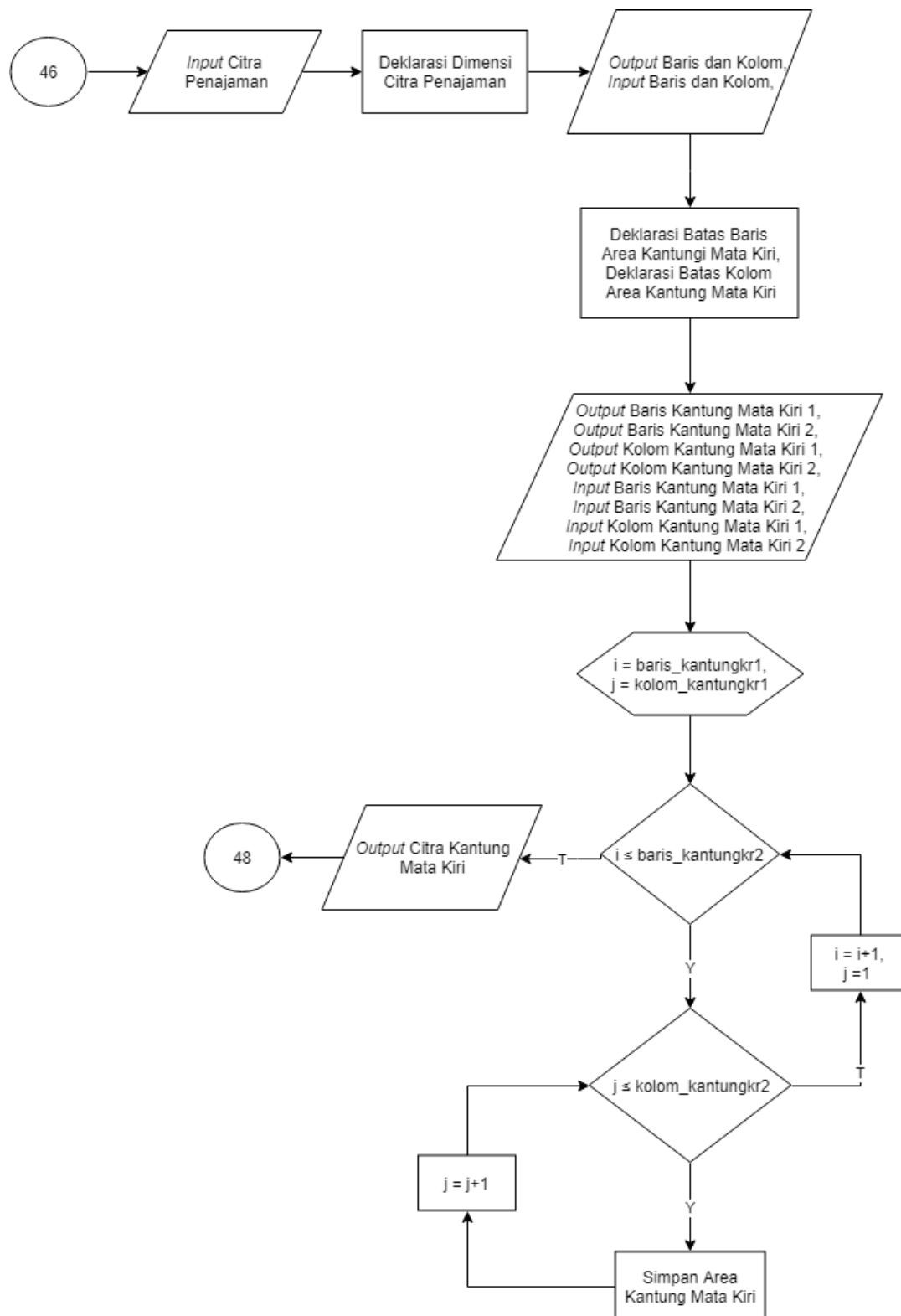
Gambar 4.19 Diagram alur proses pengambilan area sisi mata kiri

Untuk baris area sisi mata kiri diberi nilai *persentase* 38-44 dari 100% dan kolom area sisi mata kiri 15-25 dari 100%. Nilai ini dipilih agar area sisi mata kiri yang diambil tidak terlalu luas, sehingga akan mengenai bagian mata dan rambut bagian kiri.



Gambar 4.20 Diagram alur proses pengambilan area sisi mata kanan

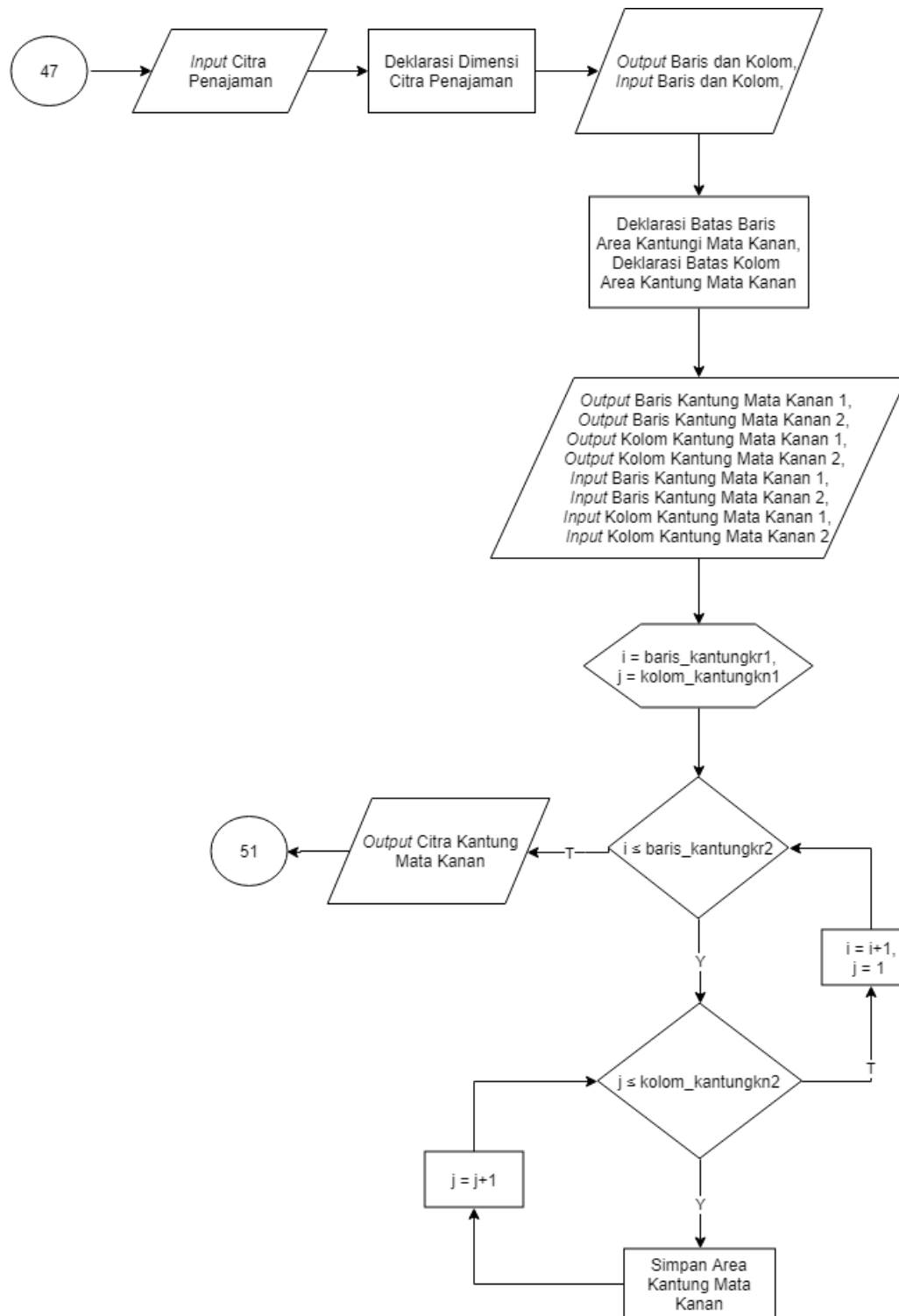
Untuk baris area sisi mata kanan menggunakan *persentase* baris area sisi mata kiri dan kolom area sisi mata kanan 75-85 dari 100%. Nilai ini dipilih agar area sisi mata kanan yang diambil tidak terlalu luas, sehingga akan mengenai bagian mata dan rambut bagian kanan.



Gambar 4.21 Diagram alur proses pengambilan area kantung mata kiri

Untuk baris area kantung mata kiri diberi nilai *persentase* 44.5-55 dari 100% dan kolom area kantung mata kiri 20-43 dari 100%. Nilai ini dipilih agar area kantung mata kiri yang diambil tidak terlalu luas, sehingga akan mengenai bagian mata dan hidung.





Gambar 4.22 Diagram alur proses pengambilan area kantung mata kanan

Untuk baris area kantung mata kanan menggunakan *persentase* baris area kantung mata kiri dan kolom area kantung mata kanan 55-80 dari 100%. Nilai ini dipilih agar area kantung mata kanan yang diambil tidak terlalu luas, sehingga akan mengenai bagian mata dan hidung.

```

%===== Ambil Area Keriput =====%

%1. Ambil Dahi
[bar kol] = size(pengurangan);
citraDahi = uint8(zeros(bar,kol));

baris_dahi1 = round(bar*20/100);
baris_dahi2 = round(bar*30/100);

kolom_dahi1 = round(kol*25/100);
kolom_dahi2 = round(kol*70/100);

for i = baris_dahi1 : baris_dahi2
    for j = kolom_dahi1 : kolom_dahi2
        citraDahi(i,j)= pengurangan(i,j);
    end
end

figure, imshow(citraDahi),title 'DAHI';

%2. Ambil Sisi Tengah Mata
citraMataT = uint8(zeros(bar,kol));

baris_mataT1 = round(bar*34/100);
baris_mataT2 = round(bar*43/100);

kolom_mataT1 = round(kol*44/100);
kolom_mataT2 = round(kol*55/100);

for i = baris_mataT1 : baris_mataT2
    for j = kolom_mataT1 : kolom_mataT2
        citraMataT(i,j) = pengurangan(i,j);
    end
end

figure, imshow(citraMataT), title 'Bagian Tengah Mata';

%3. Ambil Sisi Mata Kiri
citraMataKr = uint8(zeros(bar,kol));

baris_matakr1 = round(bar*38/100);
baris_matakr2 = round(bar*44/100);

kolom_matakr1 = round(kol*15/100);
kolom_matakr2 = round(kol*25/100);

for i = baris_matakr1 : baris_matakr2
    for j = kolom_matakr1 : kolom_matakr2
        citraMataKr(i,j) = pengurangan(i,j);
    end
end

figure, imshow(citraMataKr), title 'Mata Kiri';

%4. Ambil Sisi Mata Kanan
citraMataKn = uint8(zeros(bar,kol));

kolom_matakn1 = round(kol*75/100);
kolom_matakn2 = round(kol*85/100);

```

```

for i = baris_matakr1 : baris_matakr2
    for j = kolom_matakn1 : kolom_matakn2
        citraMataKn(i,j) = pengurangan(i,j);
    end
end

figure, imshow(citraMataKn), title 'Mata Kanan';

%5. Kantung Mata Kiri
citraKantungKr = uint8(zeros(bar,kol));

baris_kantungkr1 = round(bar*44.5/100);
baris_kantungkr2 = round(bar*55/100);

kolom_kantungkr1 = round(kol*20/100);
kolom_kantungkr2 = round(kol*43/100);

for i = baris_kantungkr1 : baris_kantungkr2
    for j = kolom_kantungkr1 : kolom_kantungkr2
        citraKantungKr(i,j) = pengurangan(i,j);
    end
end

figure, imshow(citraKantungKr), title 'Kantung Mata Kiri';

%6. Kantung Mata Kanan
citraKantungKn = uint8(zeros(bar,kol));

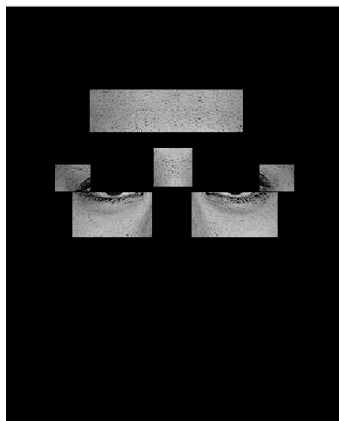
kolom_kantungkn1 = round(kol*55/100);
kolom_kantungkn2 = round(kol*80/100);

for i = baris_kantungkr1 : baris_kantungkr2
    for j = kolom_kantungkn1 : kolom_kantungkn2
        citraKantungKn(i,j) = pengurangan(i,j);
    end
end

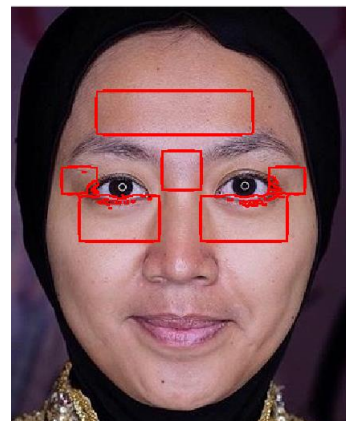
figure, imshow(citraKantungKn), title 'Kantung Mata Kanan';

```

Gambar 4.23 Sintaks proses pengambilan area keriput



(a)

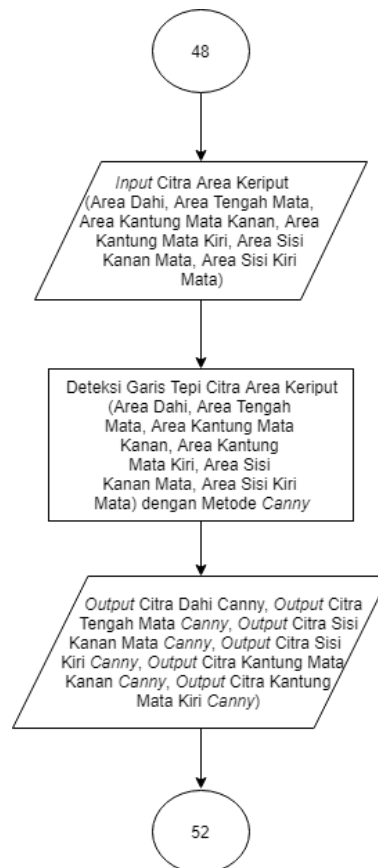


(b)

Gambar 4.24 Hasil citra: (a). Area keriput (b). Area keriput pada citra asli

#### 4.5.2 Canny dan Area

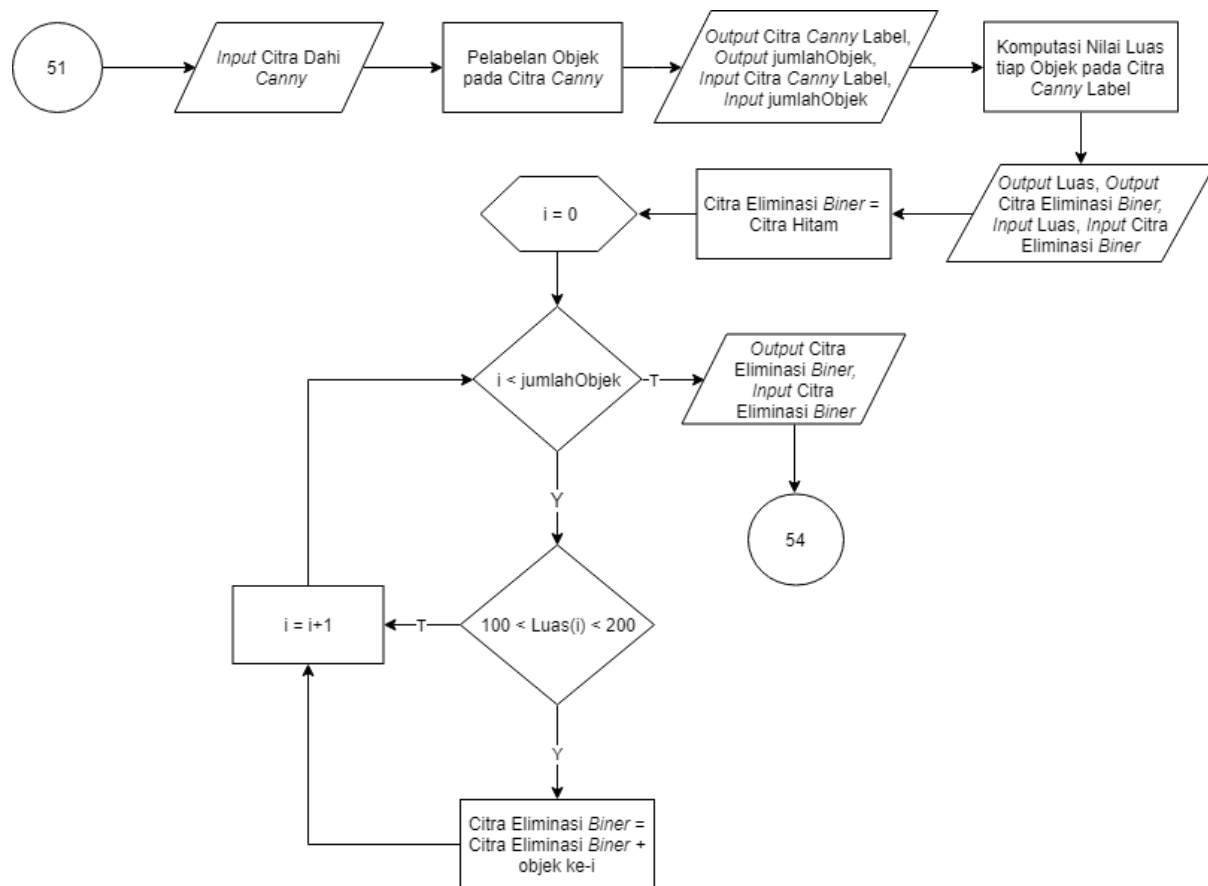
Setelah area-area keriput didapatkan, masing-masing area diberi filter *Canny* dengan nilai *threshold* yang berbeda-beda. *Canny* dipilih karena metode ini mampu menampilkan garis-garis wajah secara detail jika dibandingkan dengan metode deteksi tepi lainnya, *Canny* mampu melokalisasi dengan baik, dan memberikan respon yang jelas untuk setiap tepi (Ginting, 2010). Metode ini juga mampu menampilkan garis-garis yang diindikasikan sebagai garis halus keriput. Pemberian filter *Canny* akan menghasilkan garis-garis pada wajah, kemudian untuk mengambil garis keriputnya saja dilakukan proses eliminasi berdasarkan luas (*Area*) garis keriput pada masing-masing citra area keriput. *Flocwchart*, *sintaks* dan hasil proses filter *Canny* dan eliminasi berdasarkan luas (*Area*) dapat dilihat pada Gambar 4.25 sampai 4.41.



Gambar 4.25 Diagram alur proses filter *Canny*

Masing-masing area keriput diberi filter *Canny* dengan nilai *threshold* yang berbeda-beda, disesuaikan dengan garis-garis yang diindikasikan sebagai garis keriput halus pada citra. Area dahi diberi filter *Canny* dengan *threshold* 0.08, area sisi tengah mata diberi filter *Canny* dengan *threshold* 0.08, area sisi mata kiri dan kanan diberi filter *Canny* dengan *threshold* 0.06, area kantung mata kiri dan kanan diberi filter *Canny* dengan *threshold* 0.10. Semakin kecil

nilai *threshold* yang diberikan, maka semakin banyak pula garis tepi yang terdeteksi dan sebaliknya. Garis-garis yang diindikasikan sebagai garis keriput pada hasil citra proses *Canny* akan dieliminasi sesuai dengan luas (*Area*) masing-masing garis, luas (*Area*) garis dilihat dari dataset dua citra uji yang digunakan sebagai acuan.



Gambar 4.26 Diagram alur proses ekstraksi ciri (*Area*) dahi

Luas satu garis pada dahi dieliminasi dengan rentang nilai lebih dari 100 dan luas dua kurang dari 200, nilai ini dipilih agar garis-garis *noise* yang bukan garis keriput dan garis bingkai pada dahi tidak terambil.

```

%===== Canny Dahi =====%
%1. Canny Dahi
CannyDahi = edge(citraDahi, 'Canny', 0.08);

%2. BW LABEL DAHI
CannyDahiBw = logical(CannyDahi);
[bwCanny, numCanny] = bwlabel(CannyDahi, 8);

%===== Eliminasi Kandidat Keriput Dahi- Canny =====%

```

```

%1. Eliminasi Berdasarkan Luas Dan Bentuk
blobMeasurements = regionprops(bwCanny, 'Area');
Area1Dahi = [blobMeasurements.Area];

indexBlob = find(Area1Dahi > 100);
ambilBlob = ismember(bwCanny, indexBlob);

blobBW = ambilBlob > 0;
[labeledblobCannyDahi, numberOfBlobs] = bwlabel(blobBW);

%2. Eliminasi Area Luas
blobMeasurementsArea = regionprops(labeledblobCannyDahi,
'Area');
Area2Dahi = [blobMeasurementsArea.Area];

meanArea = mean(Area2Dahi);
stdArea = std(Area2Dahi);

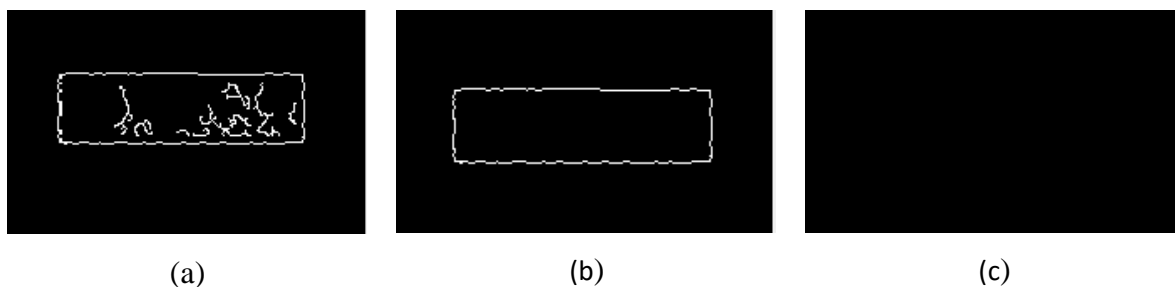
% indexArea = find(areaM > meanArea + stdArea);
indexArea = find(Area2Dahi < 200);
ambilAreaDahi = ismember(labeledblobCannyDahi, indexArea);

blobBW = ambilAreaDahi > 0;
[labelAreaBWDahi, numberAreaBW] = bwlabel(blobBW);

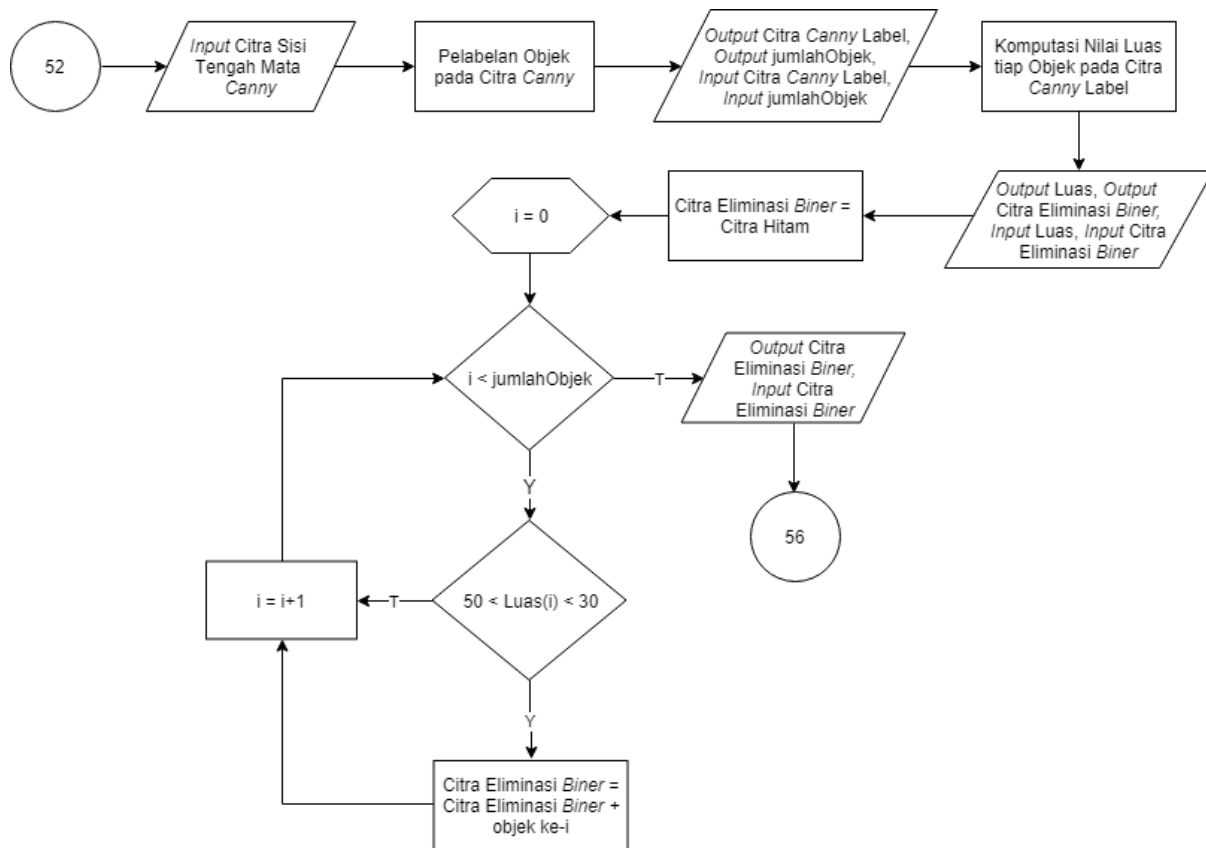
figure,
subplot(1,4,1), imshow(CannyDahi), title('Canny Dahi'),
subplot(1,4,2), imshow(bwCanny), title('BW Canny Dahi'),
subplot(1,4,3), imshow(labeledblobCannyDahi), title('Area 1
Dahi'),
subplot(1,4,4), imshow(labelAreaBWDahi), title('Area
2/Bingkai Dahi');

```

Gambar 4.27 Sintaks proses ekstraksi ciri dahi



Gambar 4.28 Hasil ekstraksi ciri: (a). Canny area dahi (b). Area 1 dahi (c). Area 2 dahi



Gambar 4.29 Diagram alur proses ekstraksi ciri (*Area*) sisi tengah mata

Luas satu garis sisi tengah mata dieliminasi dengan rentang nilai lebih dari 50 dan luas dua kurang dari 30, nilai ini dipilih agar garis-garis *noise* yang bukan garis keriput dan garis bingkai pada sisi tengah mata tidak terambil.

```

%===== Canny Sisi Tengah =====%
%1. Canny
CannySisiT = edge(citraMataT, 'Canny', 0.08);

%2. BW LABEL
CannySisiTBw = logical(CannySisiT);
[bwSisiT, numCanny] = bwlabel(CannySisiT, 8);

%===== Eliminasi Kandidat Keriput - Canny =====%

%1. Eliminasi Berdasarkan Luas Dan Bentuk

%Sisi Tengah Mata
blobMeasurements = regionprops(bwSisiT, 'Area');
Area1SisiT = [blobMeasurements.Area];

indexBlob = find(Area1SisiT > 50);
ambilBlob = ismember(bwSisiT, indexBlob);

blobBW = ambilBlob > 0;
  
```

```

[labeledblobCannySisiT, numberOfBlobs] = bwlabel(blobBW);

%2. Eliminasi Area Luas
blobMeasurementsArea = regionprops(labeledblobCannySisiT,
'Area');
Area2BT = [blobMeasurementsArea.Area];

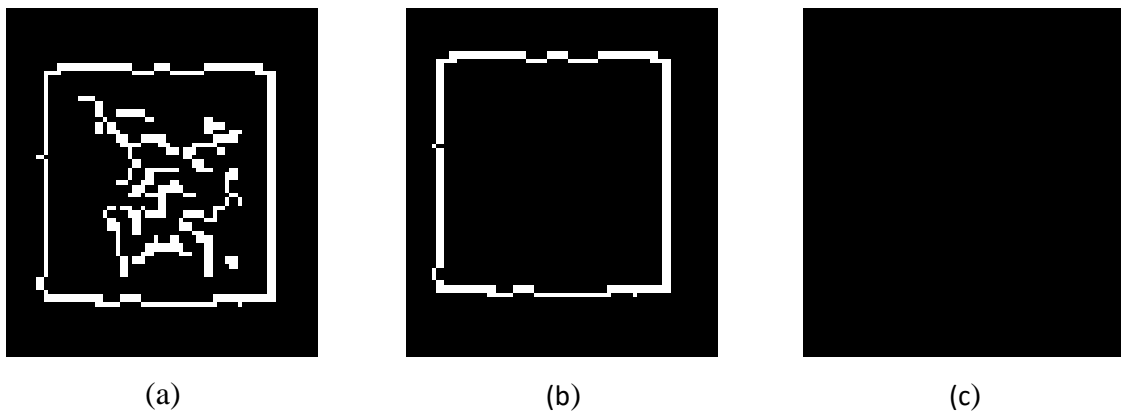
% indexArea = find(areaM > meanArea + stdArea);
indexArea = find(Area2BT < 30);
ambilAreaT = ismember(labeledblobCannySisiT, indexArea);

blobBW = ambilAreaT > 0;
[labelAreaBWT, numberAreaBW] = bwlabel(blobBW);

figure,
subplot(1,4,1), imshow(CannySisiT), title('Canny Sisi
Tengah'),
subplot(1,4,2), imshow(bwSisiT), title('BW Canny Sisi
Tengah'),
subplot(1,4,3), imshow(labeledblobCannySisiT), title('Area 1
Sisi Tengah'),
subplot(1,4,4), imshow(labelAreaBWT), title('Area 2/Bingkai
Sisi Tengah');

```

Gambar 4.30 *Sintaks* proses ekstraksi ciri sisi tengah mata



(a)

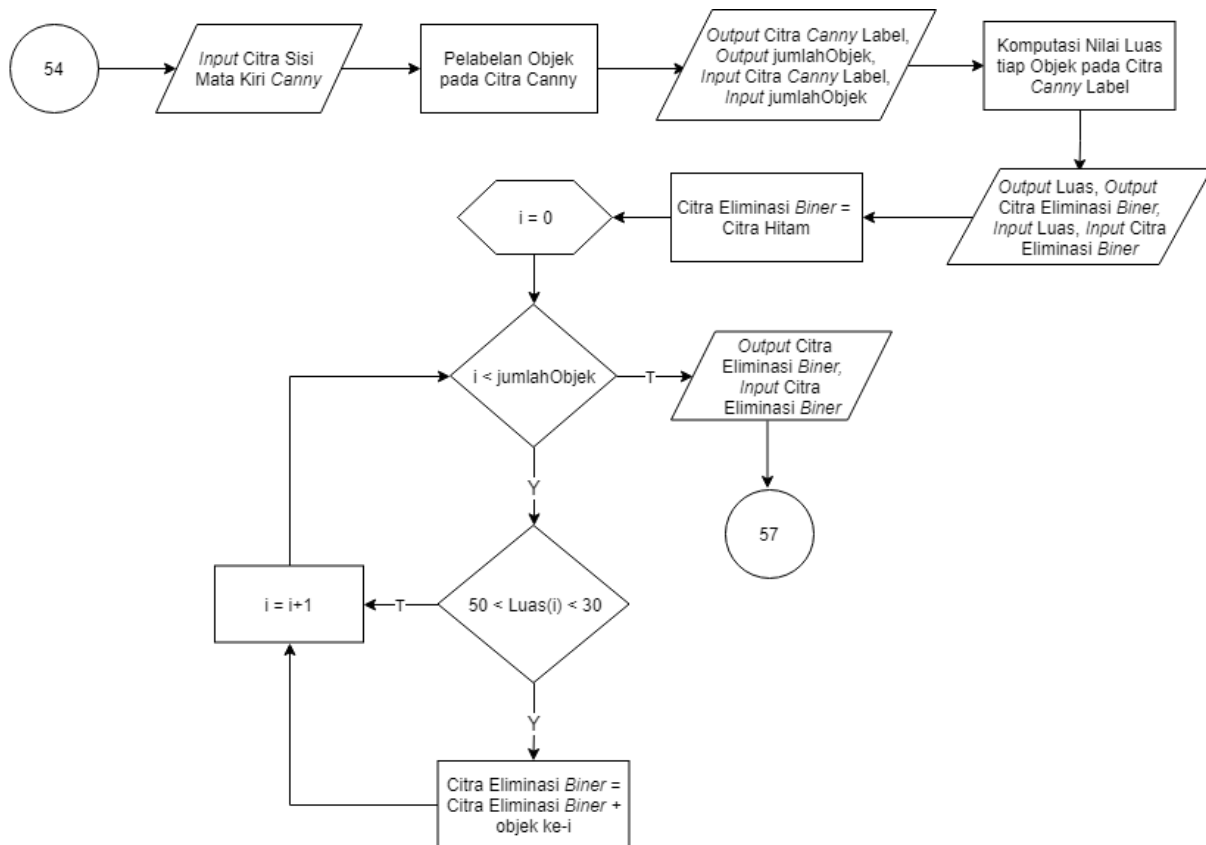
(b)

(c)

Gambar 4.31 Hasil ekstraksi ciri: (a). *Canny* area sisi tengah mata

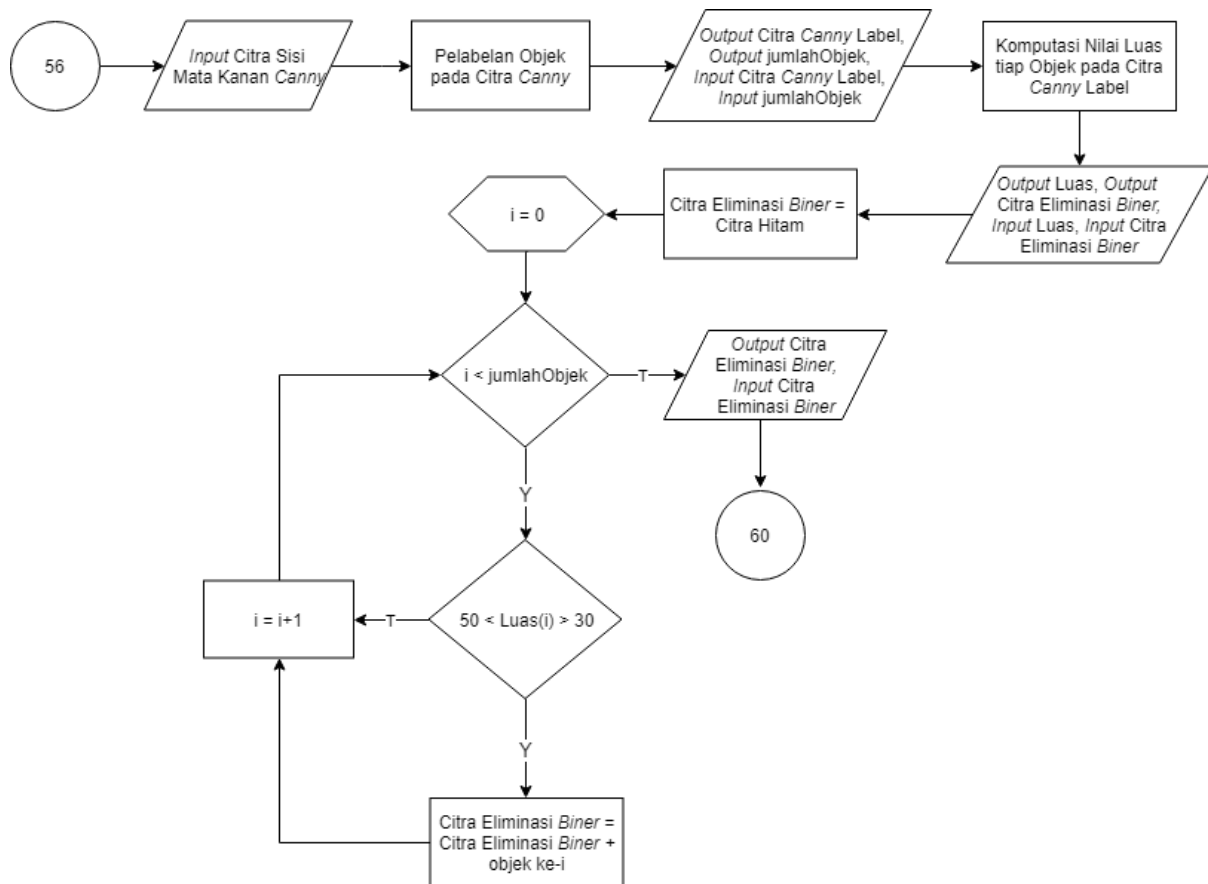
(b). Area 1 sisi tengah mata (c). Area 2 sisi tengah mata





Gambar 4.32 Diagram alur proses ekstraksi ciri (*Area*) sisi kiri mata

Luas satu garis sisi kiri mata dieliminasi dengan rentang nilai lebih dari 50 dan luas dua kurang dari 30, nilai ini dipilih agar garis-garis *noise* yang bukan garis keriput dan garis bingkai pada sisi kiri mata tidak terambil.



Gambar 4.33 Diagram alur proses ekstraksi ciri (*Area*) sisi kanan mata

Luas satu garis sisi kanan mata dieliminasi dengan rentang nilai lebih dari 50 dan luas dua kurang dari 30, nilai ini dipilih agar garis-garis *noise* yang bukan garis keriput dan garis bingkai pada sisi kanan mata tidak terambil.

```

%===== Canny Sisi Mata =====%
%1. Canny
CannyMataKr = edge(citraMataKr, 'Canny', 0.06);

CannyMataKn = edge(citraMataKn, 'Canny', 0.06);

%2. BW LABEL
CannyMataKrBw = logical(CannyMataKr);
[bwCannyMataKr, numCanny] = bwlabel(CannyMataKr, 8);

CannyMataKnBw = logical(CannyMataKn);
[bwCannyMataKn, numCanny] = bwlabel(CannyMataKn, 8);

%===== Eliminasi Kandidat Keriput - Canny =====%

%1. Eliminasi Berdasarkan Luas Dan Bentuk

%a. Sisi Mata Kiri
blobMeasurements = regionprops(bwCannyMataKr, 'Area');
Area1Mkr = [blobMeasurements.Area];
  
```

```

indexBlob = find(ArealMkr > 50);
ambilBlob = ismember(bwCannyMatakr, indexBlob);

blobBW = ambilBlob > 0;
[labeledblobCannyMatakr, numberOfBlobs] = bwlabel(blobBW);

%b. Sisi Mata Kanan
blobMeasurements = regionprops(bwCannyMatakkn, 'Area');
ArealMkn = [blobMeasurements.Area];

indexBlob = find(ArealMkn > 50);
ambilBlob = ismember(bwCannyMatakkn, indexBlob);

blobBW = ambilBlob > 0;
[labeledblobCannyMatakkn, numberOfBlobs] = bwlabel(blobBW);

%2. Eliminasi Berdasarkan Luas Dan Bentuk Bingkai

%a. Sisi Mata Kiri
blobMeasurements = regionprops(labeledblobCannyMatakr,
'Area');
Area2Mkr = [blobMeasurements.Area];

indexBlob = find(Area2Mkr < 30);
ambilBlob = ismember(labeledblobCannyMatakr, indexBlob);

blobBW = ambilBlob > 0;
[labeledAreaBWMatakr, numberOfBlobs] = bwlabel(blobBW);

%b. Sisi Mata Kanan
blobMeasurements = regionprops(labeledblobCannyMatakkn,
'Area');
Area2Mkn = [blobMeasurements.Area];

indexBlob = find(Area2Mkn < 30);
ambilBlob = ismember(labeledblobCannyMatakkn, indexBlob);

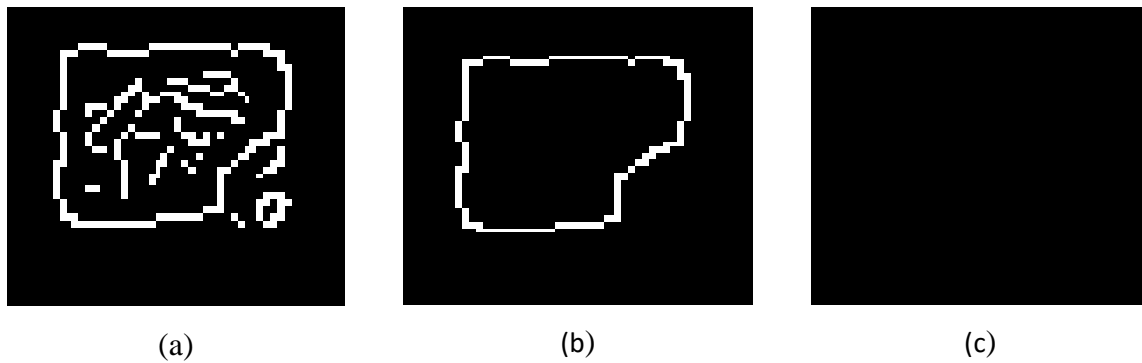
blobBW = ambilBlob > 0;
[labeledAreaBWMatakkn, numberOfBlobs] = bwlabel(blobBW);

figure,
subplot(2,4,1), imshow(CannyMatakr), title('Canny Sisi
Kiri'),
subplot(2,4,2), imshow(bwCannyMatakr), title('BW Sisi Kiri'),
subplot(2,4,3), imshow(labeledblobCannyMatakr), title('Area 1
Sisi Kiri');
subplot(2,4,4), imshow(labeledAreaBWMatakr), title('Area
2/Bingkai Sisi Kiri'),

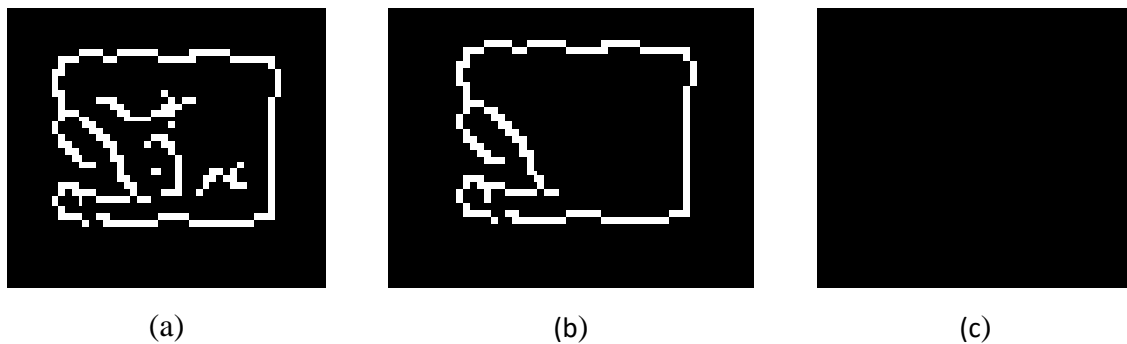
subplot(2,4,5), imshow(CannyMatakkn), title('Canny Sisi
Kanan'),
subplot(2,4,6), imshow(bwCannyMatakkn), title('BW Canny Sisi
Kanan'),
subplot(2,4,7), imshow(labeledblobCannyMatakkn), title('Area 1
Sisi Kanan');
subplot(2,4,8), imshow(labeledAreaBWMatakkn), title('Area
2/Bingkai Sisi Kanan');

```

Gambar 4.34 *Sintaks* proses ekstraksi ciri sisi kiri dan kanan mata

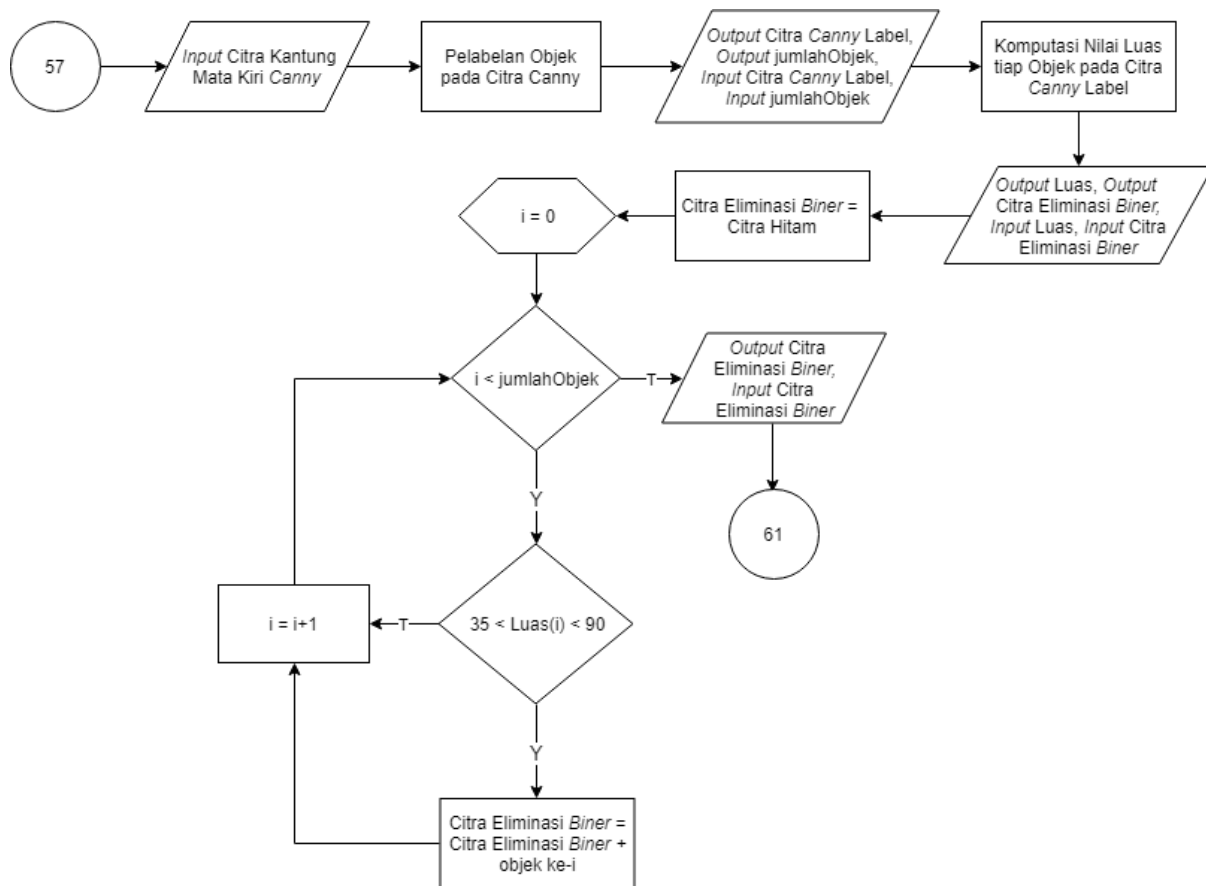


Gambar 4.35 Hasil ekstraksi ciri: (a). *Canny* area sisi kiri mata (b). Area 1 sisi kiri mata (c). Area 2 sisi kiri mata



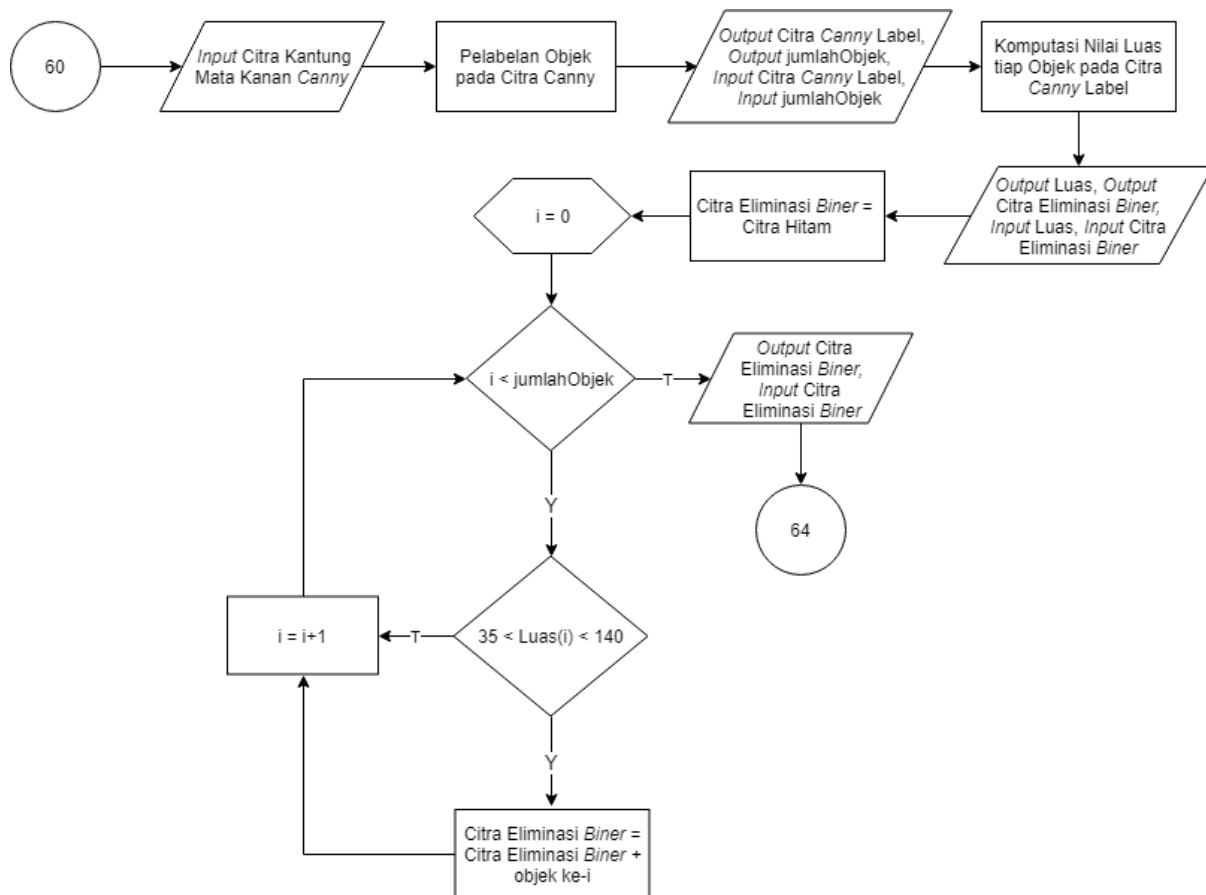
Gambar 4.36 Hasil ekstraksi ciri: (a). *Canny* area sisi kanan mata (b). Area 1 sisi kanan mata (c). Area 2 sisi kanan mata

Nilai luas area sisi tengah mata, sisi kiri dan kanan mata disamakan karena tidak ada garis yang terindikasi sebagai garis keriput, sehingga rentang nilai yang diambil digunakan untuk menghilangkan *noise* dan bingkai pada area tersebut.



Gambar 4.37 Diagram alur proses ekstraksi ciri (*Area*) kantung mata kiri

Luas satu garis kantung mata kiri dieliminasi dengan rentang nilai lebih dari 35 dan luas dua kurang dari 90, nilai ini dipilih agar garis-garis *noise* yang bukan garis keriput dan garis bingkai pada kantung mata kiri tidak terambil.



Gambar 4.38 Diagram alur proses ekstraksi ciri (*Area*) kantung mata kanan

Luas satu garis kantung mata kanan dieliminasi dengan rentang lebih dari 35, luas dua lebih dari 55 dan luas tiga kurang dari 140, nilai ini dipilih agar garis-garis *noise* yang bukan garis keriput dan garis bingkai pada kantung mata kanan tidak terambil. Area kantung mata kanan dieliminasi sebanyak tiga kali karena luas garis yang bukan garis keriput seperti garis tepi mata dan garis bingkai kantung mata kanan tetap terambil jika eliminasi hanya dilakukan dua kali.

```

%===== Canny Kantung Mata =====%
%1. Canny

CannyKantungkr = edge(citraKantungKr, 'Canny', 0.10);
CannyKantungkn = edge(citraKantungKn, 'Canny', 0.10);

%2. BW LABEL
CannyKantungKrBw = logical(CannyKantungkr);
[bwCannykkr, numCanny]=bwlabel(CannyKantungkr, 8);
CannyKantungKnBw = logical(CannyKantungkn);
[bwCannykkn, numCanny]=bwlabel(CannyKantungkn, 8);

```

```

===== Eliminasi Kandidat Keriput - Canny
=====

%1.  Eliminasi Berdasarkan Luas Dan Bentuk

%a1. Kantung Kiri
blobMeasurements = regionprops(bwCannykkr, 'Area');
Area1kkr = [blobMeasurements.Area];

indexBlob = find(Area1kkr > 35);
ambilBlob = ismember(bwCannykkr, indexBlob);

blobBW = ambilBlob > 0;
[labeledblobCannykkr, numberOfBlobs] = bwlabel(blobBW);

%a2. Kantung Kiri
blobMeasurements = regionprops(labeledblobCannykkr, 'Area');
Area2kkr = [blobMeasurements.Area];

indexBlob = find(Area2kkr < 90);
ambilBlob = ismember(labeledblobCannykkr, indexBlob);

blobBW = ambilBlob > 0;
[labeledBW2kkr, numberOfBlobs] = bwlabel(blobBW);

%2.  Eliminasi Berdasarkan Luas Dan Bentuk Bingkai

%b1. Kantung Kanan
blobMeasurements = regionprops(bwCannykkn, 'Area');
Area1kkn = [blobMeasurements.Area];

indexBlob = find(Area1kkn > 35);
ambilBlob = ismember(bwCannykkn, indexBlob);

blobBW = ambilBlob > 0;
[labeledblobCannykkn, numberOfBlobs] = bwlabel(blobBW);

%b2. Kantung Kanan
blobMeasurements = regionprops(labeledblobCannykkn, 'Area');
Area2kkn = [blobMeasurements.Area];

indexBlob = find(Area2kkn > 55 );
ambilBlob = ismember(labeledblobCannykkn, indexBlob);

blobBW = ambilBlob > 0;
[labeledBWkkn, numberOfBlobs] = bwlabel(blobBW);

%b3. Kantung Kanan
blobMeasurements = regionprops(labeledBWkkn, 'Area');
Area3kkn = [blobMeasurements.Area];

indexBlob = find(Area3kkn < 140);
ambilBlob = ismember(labeledBWkkn, indexBlob);

blobBW = ambilBlob > 0;
[labeledBW3kkn, numberOfBlobs] = bwlabel(blobBW);

figure,

```

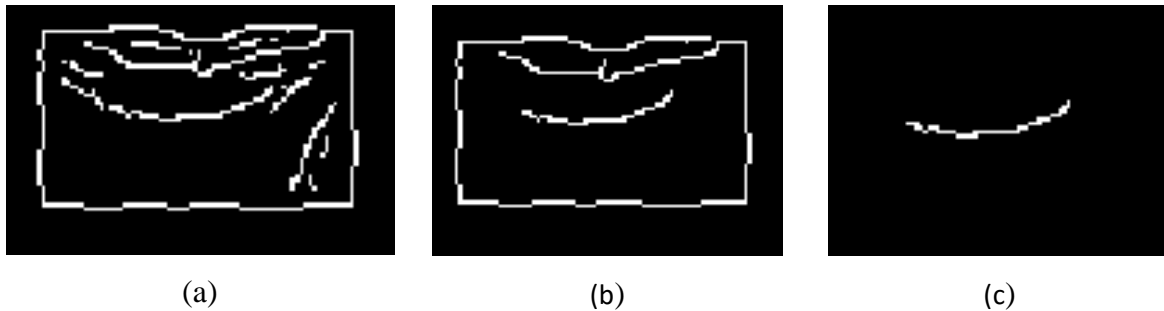
```

subplot(2,4,1), imshow(CannyKantungkr), title('Canny Kantung
Kiri'),
subplot(2,4,2), imshow(bwCannykkr), title('BW Canny Kantung
Kiri'),
subplot(2,4,3), imshow(labeledblobCannykkr), title('Area 1
Kantung Kiri');
subplot(2,4,4), imshow(labeledBW2kkr), title('Area 2/Bingkai
Kantung Kiri'),

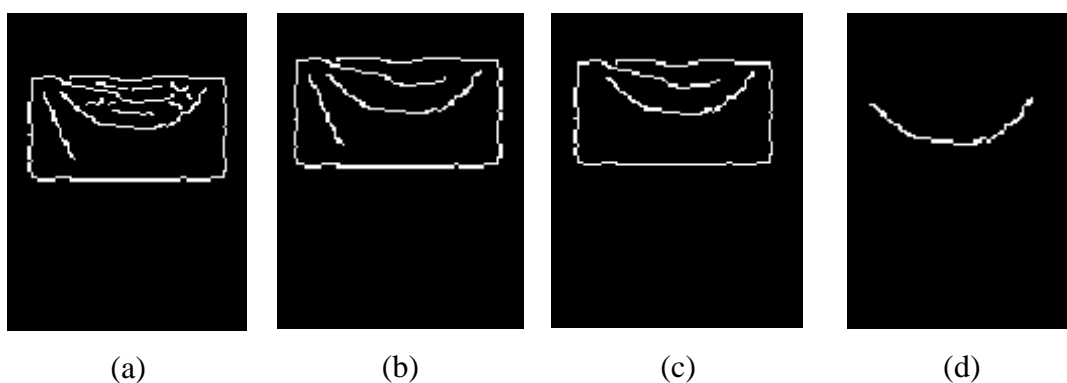
subplot(2,4,5), imshow(CannyKantungkn), title('Canny Kantung
Kanan');
subplot(2,4,6), imshow(bwCannykkn), title('BW Canny Kantung
Kanan');
subplot(2,4,7), imshow(labeledblobCannykkn), title('Area 1
Kantung Kanan'),
% area 2 tidak dimasukkan ke gambar
subplot(2,4,8), imshow(labeledBW3kkn), title('Area 2/Bingkai
Kantung Kanan');

```

Gambar 4.39 *Sintaks* proses ekstraksi ciri kantung kiri dan kanan mata



Gambar 4.40 Hasil ekstraksi ciri: (a). *Canny* area kantung kiri mata (b). Area 1 kantung kiri mata (c). Area 2 kantung kiri mata

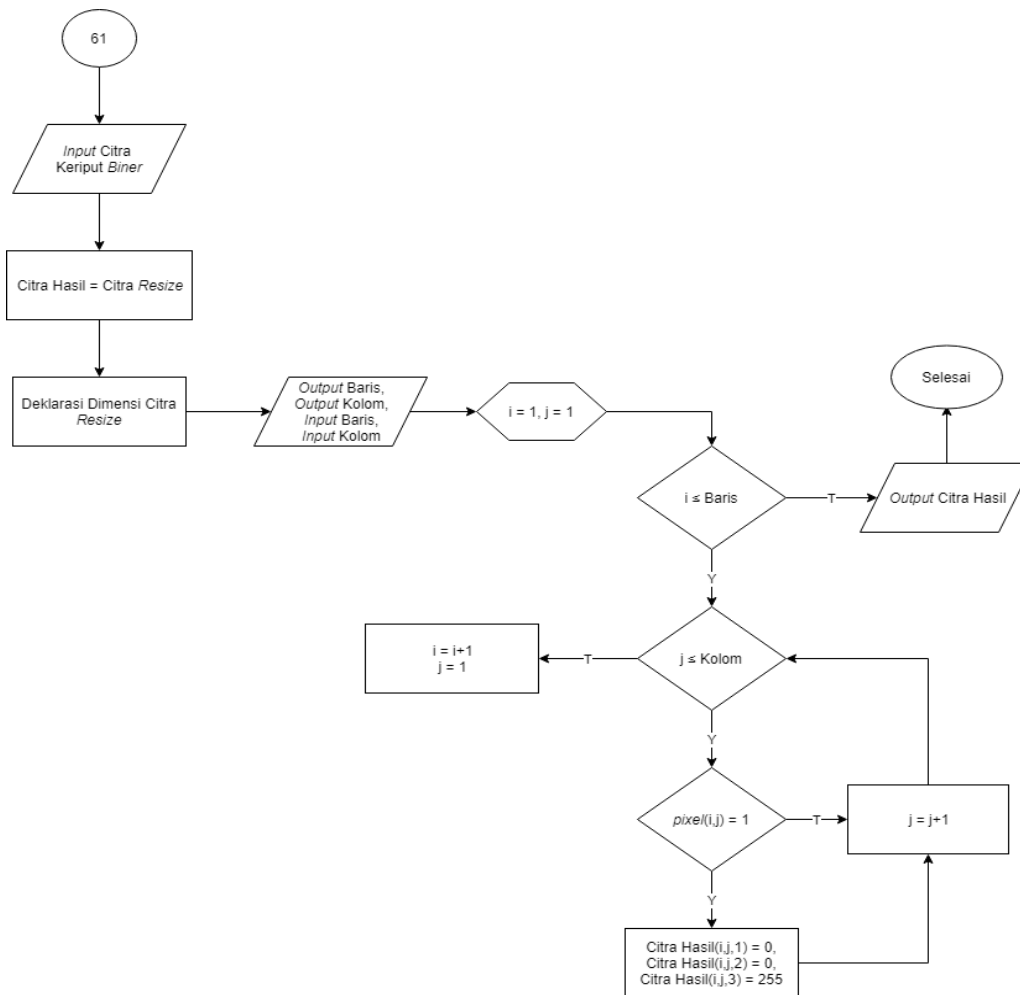


Gambar 4.41 Hasil ekstraksi ciri: (a). *Canny* area kantung kanan mata (b). Area 1 kantung kanan mata (c). Area 2 kantung kanan mata (d). Area 3 kantung kanan mata



#### 4.6 Marking

Setelah proses ekstraksi ciri dilakukan, hasil citra ekstraksi ciri semua area keriput dipindahkan ke dalam citra asli, dan ditandai dengan garis biru. Warna biru dipilih agar warna garis kontras dengan warna kulit wajah citra, sehingga penandaan akan jelas terlihat. *Flowchart*, *sintaks* dan hasil penandaan garis keriput pada citra wajah asli dapat dilihat pada Gambar 4.42 sampai 4.44.



Gambar 4.42 Diagram alur proses *marking*

```

%===== Marking =====%

hasil = citra;

for i = 1 : bar
    for j = 1 : kol
        if labelAreaBWDahi(i, j) > 0;
            hasil(i, j, 1) = 0;
            hasil(i, j, 2) = 0;
            hasil(i, j, 3) = 255;
        end;
    end;
end;

```

```

    end;
end;

for i = 1 : bar
    for j = 1 : kol
        if labeledAreaBWMatakr(i, j) > 0;
            hasil(i, j, 1) = 0;
            hasil(i, j, 2) = 0;
            hasil(i, j, 3) = 255;
        end;
    end;
end;

for i = 1 : bar
    for j = 1 : kol
        if labeledAreaBWMatakkn(i, j) > 0;
            hasil(i, j, 1) = 0;
            hasil(i, j, 2) = 0;
            hasil(i, j, 3) = 255;
        end;
    end;
end;

for i = 1 : bar
    for j = 1 : kol
        if labelAreaBWT(i, j) > 0;
            hasil(i, j, 1) = 0;
            hasil(i, j, 2) = 0;
            hasil(i, j, 3) = 255;
        end;
    end;
end;

for i = 1 : bar
    for j = 1 : kol
        if labeledBW2kkr(i, j) > 0;
            hasil(i, j, 1) = 0;
            hasil(i, j, 2) = 0;
            hasil(i, j, 3) = 255;
        end;
    end;
end;

for i = 1 : bar
    for j = 1 : kol
        if labeledBW3kkn(i, j) > 0;
            hasil(i, j, 1) = 0;
            hasil(i, j, 2) = 0;
            hasil(i, j, 3) = 255;
        end;
    end;
end;

hasil = uint8(hasil);
figure,
subplot(1,2,1),imshow(citra),title('Citra Asli')
subplot(1,2,2),imshow(hasil),title('Ambil Keriput');

```

Gambar 4.43 Sintaks proses marking



Gambar 4.44 Hasil penandaan keriput

#### 4.7 Hasil Ekstraksi Ciri Keriput

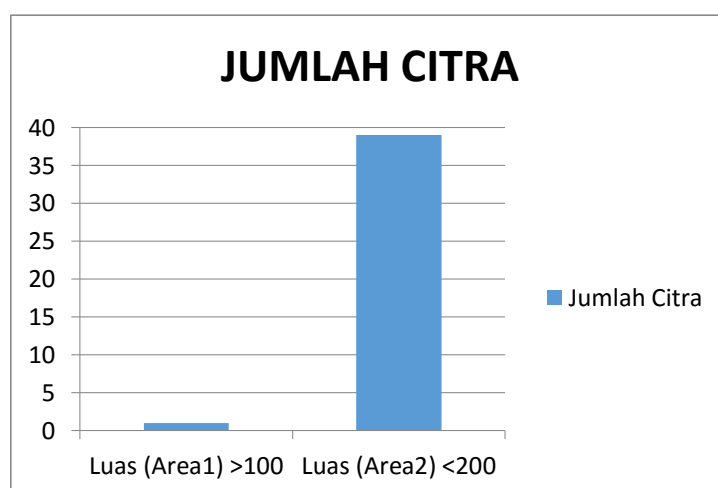
Berdasarkan hasil ekstraksi ciri citra garis keriput, maka di dapatkan nilai luas (*Area*) yang digunakan sebagai batasan untuk menentukan ciri dari garis keriput, nilai diambil dari dataset luas (*Area*) garis yang diindikasikan sebagai keriput pada masing-masing citra hasil proses deteksi tepi *Canny*. Rentang nilai digunakan untuk membedakan antara objek garis keriput dengan objek bukan garis keriput, semakin besar nilai area suatu objek garis, maka semakin luas atau panjang garis keriput terdeteksi. Hasil ekstraksi ciri garis keriput dapat dilihat pada Tabel 4.1 sebagai berikut:

Tabel 4.1 Perhitungan ekstraksi ciri garis keriput (*Area*)

Sampel	Area Dahi		Area Sisi Tengah Mata		Area Sisi Mata Kiri		Area Sisi Mata Kanan		Area Kantung Mata Kiri		Area Kantung Mata Kanan		
	1	2	1	2	1	2	1	2	1	2	1	2	3
Keriput 1	0	0	0	0	0	0	0	0	42	42	87	87	87
Keriput 2	0	0	0	0	0	0	0	0	73	73	100	100	100
Keriput 3	0	0	0	0	0	0	0	0	52	52	62	62	62
Keriput 4	0	0	0	0	0	0	0	0	38-53	38-53	60-61	60-61	60-61
Keriput 5	0	0	0	0	0	0	0	0	61-66	61-66	0	0	0
Keriput 6	0	0	0	0	0	0	0	0	66	66	60-80	60-80	60-80
Keriput 7	0	0	0	0	0	0	0	0	54	54	85	85	85
Keriput 8	0	0	0	0	0	0	0	0	88	88	70	70	70
Keriput 9	0	0	0	0	0	0	0	0	47	47	86	86	86
Keriput 10	0	0	0	0	0	0	0	0	39-77	39-77	69	69	69
Keriput 11	0	0	0	0	0	0	0	0	69	69	72	72	72
Keriput 12	0	0	0	0	0	0	0	0	49	49	68	68	68
Keriput 13	0	0	0	0	0	0	0	0	53-62	53-62	87	87	87
Keriput 14	0	0	0	0	0	0	0	0	46-75	46-75	60	60	60
Keriput 15	0	0	0	0	0	0	0	0	84	84	0	0	0
Keriput 16	0	0	0	0	0	0	0	0	36-49	36-49	98	98	98

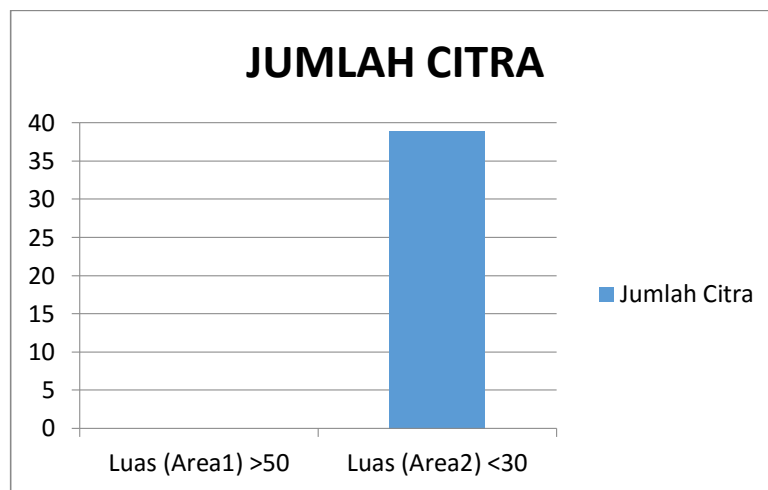
Keriput 17	0	0	0	0	0	0	0	0	61	61	56	56	56
Keriput 18	0	0	0	0	0	0	0	0	50	50	63	63	63
Keriput 19	0	0	0	0	0	0	0	0	49	49	67	67	67
Keriput 20	0	0	0	0	0	0	0	0	51	51	0	0	0
Keriput 21	0	0	0	0	0	0	0	0	0	0	72	72	72
Keriput 22	0	0	0	0	0	0	0	0	65	65	57	57	57
Keriput 23	0	0	0	0	0	0	0	0	36-62	36-62	67	67	67
Keriput 24	0	0	0	0	0	0	0	0	47-82	47-82	64-119	64-119	64-119
Keriput 25	0	0	0	0	0	0	0	0	38-83	38-83	81	81	81
Keriput 26	0	0	0	0	0	0	0	0	87	87	60	60	60
Keriput 27	0	0	0	0	0	0	0	0	68	68	57	57	57
Keriput 28	0	0	0	0	0	0	0	0	38-48	38-48	108	108	108
Keriput 29	0	0	0	0	0	0	0	0	64-68	64-68	57-138	57-138	57-138
Keriput 30	0	0	0	0	0	0	0	0	42	42	61	61	61
Keriput 31	0	0	0	0	0	0	0	0	77	77	87	87	87
Keriput 32	0	0	0	0	0	0	0	0	57	57	65	65	65
Keriput 33	0	0	0	0	0	0	0	0	60	60	59	59	59
Keriput 34	0	0	0	0	0	0	0	0	79	79	68	68	68
Keriput 35	0	0	0	0	0	0	0	0	54	54	66	66	66
Keriput 36	0	0	0	0	0	0	0	0	43	43	61	61	61
Keriput 37	0	0	0	0	0	0	0	0	43	43	66	66	66
Keriput 38	0	0	0	0	0	0	0	0	50-51	50-51	71	71	71
Keriput 39	111-137	111-137	0	0	0	0	0	0	55-66	55-66	61-91	61-91	61-91
	Min: 111		Min: 0		Min:0		Min: 0		Min: 36		Min: 56		
	Max: 137		Max: 0		Max: 0		Max: 0		Max: 87		Max: 138		
Nilai Area Minimum : 36									Nilai Area Maksimum : 138				

Sesuai dengan tabel di atas dapat disimpulkan bahwa masing-masing objek garis keriput di setiap area memiliki nilai luas (*Area*) yang berbeda sesuai dengan panjang dan lebar garis yang terdeteksi. Perbandingan rentang luas (*Area*) dahi pada *sintaks* dengan rentang luas (*Area*) dahi sebenarnya yang disajikan dalam bentuk grafik dapat dilihat pada Gambar 4.45.



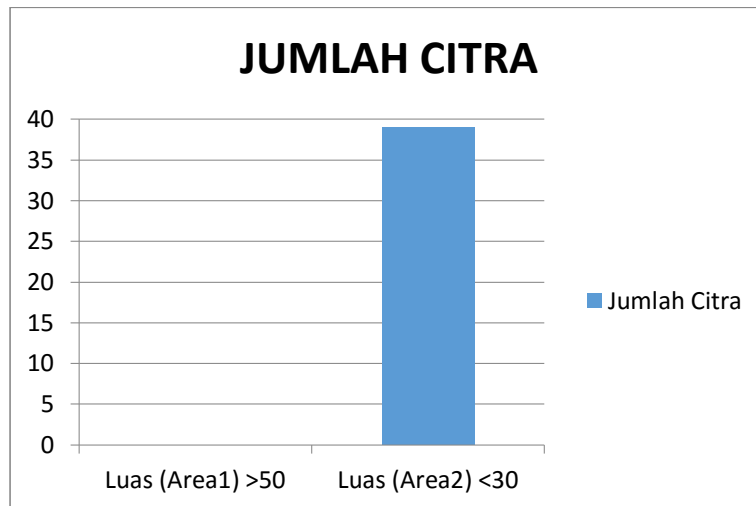
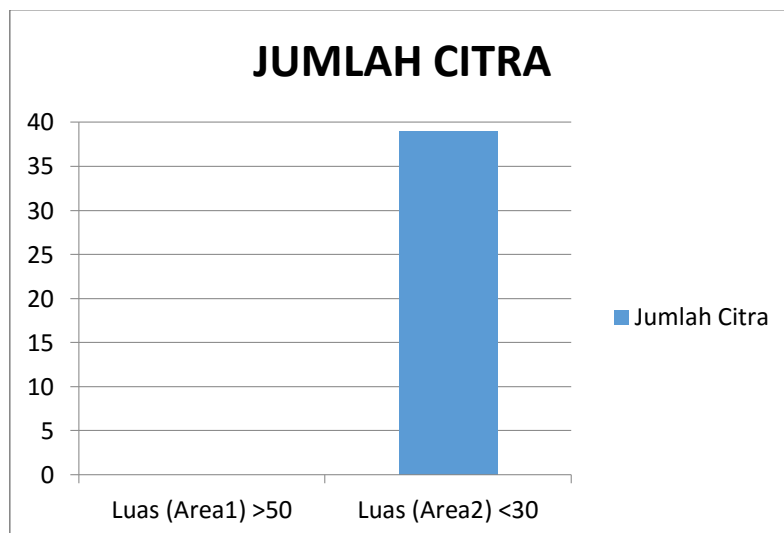
Gambar 4.45 Grafik luas (*Area*) dahi

Hasil grafik menunjukkan hanya satu citra yang masuk kedalam rentang nilai luas (*Area*) 1 yang diterapkan di sistem, nilai luas (*Area*) garis keriput pada dahi sebenarnya memiliki luas minimum 111 dan nilai maksimum 137. Hal ini dikarenakan hanya terdapat sebuah citra dari 39 citra uji yang memiliki ciri keriput pada bagian dahi, persentase kecocokan rentang nilai yang digunakan pada sistem dengan rentang nilai sebenarnya adalah 2,56%. Sedangkan untuk nilai luas (*Area*) 2, semua citra uji masuk kedalam rentang yang diterapkan sistem, rentang nilai luas (*Area*) 2 digunakan untuk menghilangkan *noise* pada citra, persentase kecocokan rentang nilai luas (*Area*) 2 yang digunakan pada sistem dengan rentang nilai sebenarnya adalah 100%. Perbandingan rentang luas (*Area*) sisi tengah mata pada *sintaks* dengan rentang luas (*Area*) sisi tengah mata sebenarnya yang disajikan dalam bentuk grafik dapat dilihat pada Gambar 4.46.

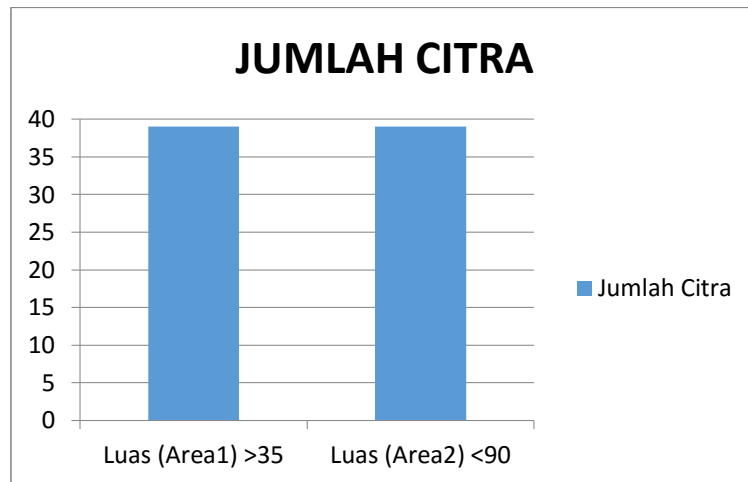


Gambar 4.46 Grafik luas (*Area*) sisi tengah mata

Hasil grafik menunjukkan tidak ada citra yang masuk dalam rentang nilai luas (*Area*) 1 yang diterapkan di sistem, hal ini dikarenakan tidak ada satupun garis yang diindikasikan sebagai garis keriput, sehingga luas (*Area*) sebenarnya pada sisi tengah mata bernilai 0, persentase kecocokan rentang nilai luas (*Area*) 1 yang digunakan pada sistem dengan rentang nilai sebenarnya adalah 0%. Sedangkan untuk nilai luas (*Area*) 2, semua citra uji masuk kedalam rentang yang diterapkan sistem, rentang nilai luas (*Area*) 2 digunakan untuk menghilangkan *noise* pada citra, persentase kecocokan rentang nilai luas (*Area*) 2 yang digunakan pada sistem dengan rentang nilai sebenarnya adalah 100%. Perbandingan rentang luas (*Area*) sisi mata kiri dan kanan pada *sintaks* dengan rentang luas (*Area*) sisi mata kiri dan kanan sebenarnya yang disajikan dalam bentuk grafik dapat dilihat pada Gambar 4.47 dan 4.48.

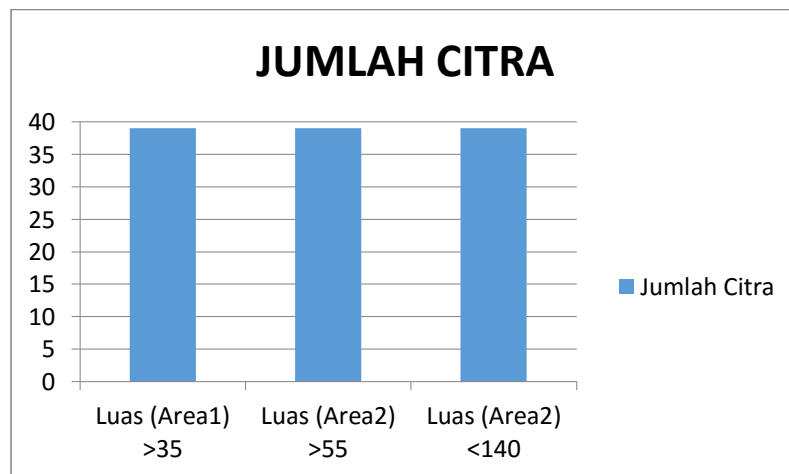
Gambar 4.47 Grafik luas (*Area*) sisi mata kiriGambar 4.48 Grafik luas (*Area*) sisi mata kanan

Untuk grafik luas (*Area*) sisi mata kiri dan kanan, hasil dari kedua grafik menunjukkan tidak ada citra yang masuk dalam rentang nilai luas (*Area*) 1 yang diterapkan di sistem dan semua citra uji masuk dalam rentang nilai luas (*Area*) 2, alasannya sama dengan alasan yang diterapkan pada hasil grafik luas (*Area*) sisi tengah mata. Persentase kecocokan rentang nilai luas (*Area*) 1 yang digunakan pada sistem dengan rentang nilai sebenarnya adalah 0%, sedangkan persentase kecocokan rentang nilai luas (*Area*) 2 yang digunakan pada sistem dengan rentang nilai sebenarnya adalah 100%. Perbandingan rentang luas (*Area*) kantung mata kiri pada *sintaks* dengan rentang luas (*Area*) kantung mata kiri sebenarnya yang disajikan dalam bentuk grafik dapat dilihat pada Gambar 4.49.



Gambar 4.49 Grafik luas (*Area*) kantung mata kiri

Hasil grafik menunjukkan semua citra uji yang terdiri dari 39 citra masuk kedalam rentang nilai luas (*Area*) 1 dan nilai luas (*Area*) 2 yang diterapkan di sistem, nilai luas (*Area*) garis keriput pada kantung mata kiri memiliki luas minimum 36 dan nilai maksimum 87, persentase kecocokan rentang nilai yang digunakan pada sistem dengan rentang nilai sebenarnya adalah 100%. Perbandingan rentang luas (*Area*) sisi tengah mata pada *sintaks* dengan rentang luas (*Area*) sisi tengah mata sebenarnya yang disajikan dalam bentuk grafik dapat dilihat pada Gambar 4.50.



Gambar 4.50 Grafik luas (*Area*) kantung mata kanan

Hasil grafik menunjukkan semua citra uji yang terdiri dari 39 citra masuk kedalam rentang nilai luas (*Area*) 1, luas (*Area*) 2, dan luas (*Area*) 2 yang diterapkan di sistem, nilai luas (*Area*) garis keriput pada kantung mata kiri memiliki luas minimum 56 dan nilai

maksimum 138. Persentase kecocokan rentang nilai yang digunakan pada sistem dengan rentang nilai sebenarnya adalah 100%.

## 4.8 Implementasi Tampilan Antarmuka

Implementasi tampilan antarmuka yang terdapat pada sistem dijelaskan pada bagian di bawah ini:

### 4.8.1 Tampilan Awal Sistem

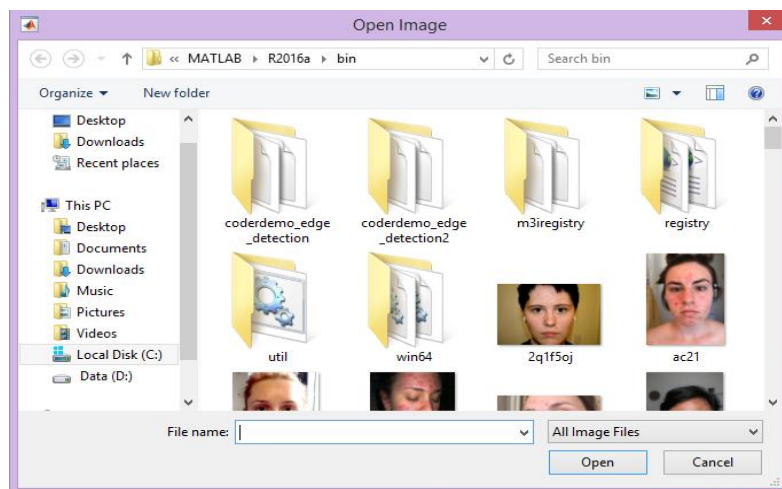
Tampilan awal sistem dapat dilihat pada Gambar 4.51.



Gambar 4.51 Tampilan awal sistem

### 4.8.2 Tampilan *FileChooser*

Tampilan *filechooser* dapat dilihat pada Gambar 4.52.

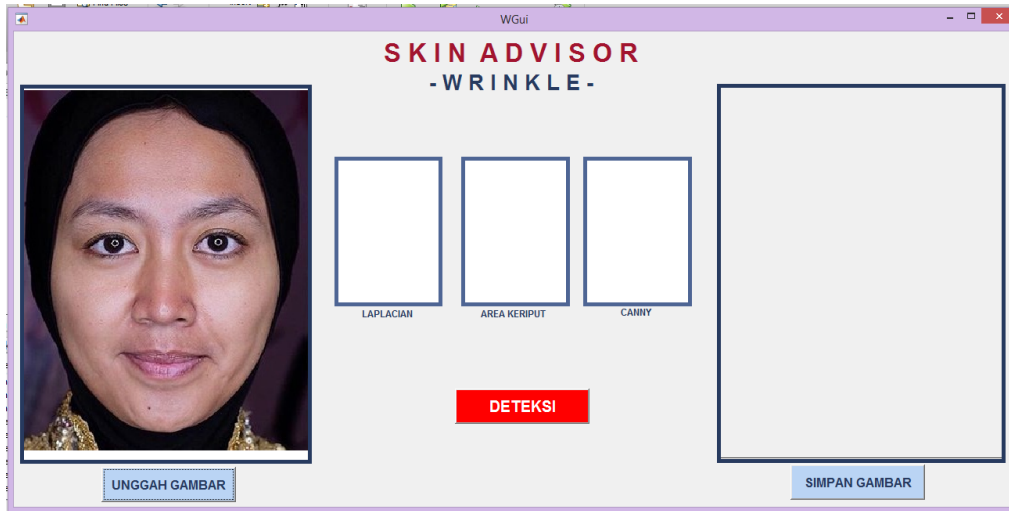


Gambar 4.52 Tampilan *filechooser* sistem

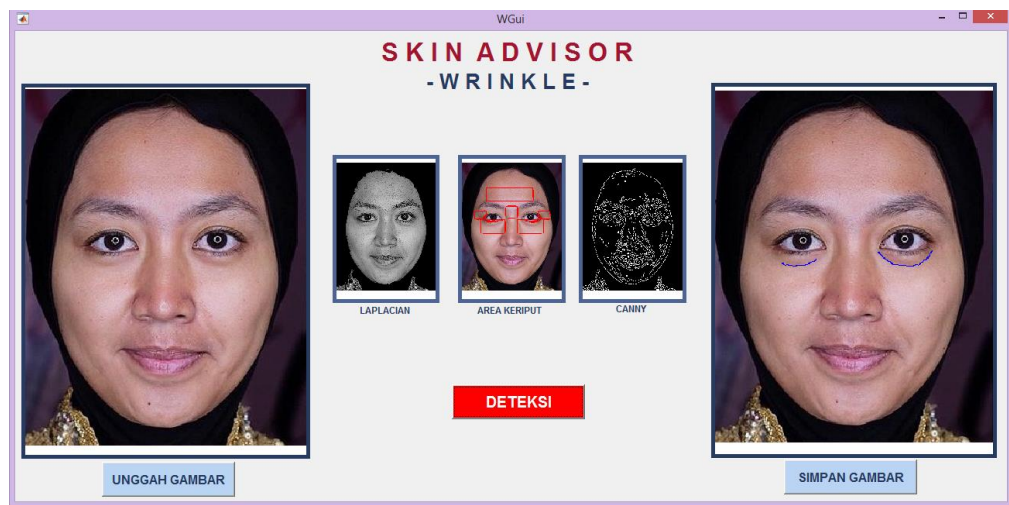


### 4.8.3 Tampilan *Load* Foto dan Hasil Deteksi

Tampilan *load* foto dan hasil deteksi pada sistem dapat dilihat pada Gambar 4.53 dan 4.54.



Gambar 4.53 Tampilan *load* foto pada sistem



Gambar 4. 54 Tampilan hasil deteksi

## 4.9 Hasil Deteksi

Hasil deteksi garis keriput dapat dilihat pada Gambar 4.55.



(a)



(b)

Gambar 4.55 Citra hasil deteksi: (a). Hasil deteksi sempurna (b). Hasil deteksi tidak sempurna

Gambar di atas adalah hasil deteksi yang dihasilkan oleh sistem, deteksi keriput berhasil dilakukan dengan sempurna pada citra (a), sistem mampu mendeteksi semua garis yang termasuk garis keriput. Sedangkan citra (b) menampilkan citra dengan deteksi tidak sempurna, karena bagian garis keriput pada sisi kantung mata kiri tidak terdeteksi.

#### 4.10 Hasil Uji Validasi Sistem

Setelah melakukan proses ekstraksi ciri dengan filter Canny dan regionprops (area), selanjutnya dilakukan proses pengujian data uji untuk validasi sistem. Pengujian dilakukan dengan metode *Single Decision Threshold* dari 39 data uji, hasil citra dari sistem digabungkan dengan citra digital dari pihak ahli yaitu dokter dan AVO, kemudian nilai *TP*, *TN*, *FP* dan *TN* dihitung dari banyak garis yang beririsan.

**True Positive** (TP) berarti banyak garis keriput yang terdeteksi di sistem dan ditandai pada citra pakar, **True Negative** (TN) berarti banyak garis keriput kandidat yang tidak terdeteksi di sistem, **False Positive** (FP) berarti banyak garis keriput yang terdeteksi di sistem tapi tidak ditandai pada citra pakar, **False Negative** (FN) berarti banyak garis keriput yang tidak terdeteksi di sistem tapi ditandai pada citra pakar.

Setelah tabel pengujian terdapat tabel *contingency* yang berisi nilai *sensitivity* dan *specivicity* dari sistem keseluruhan. **Sensitivity** adalah parameter untuk mengukur persentase data positif yang teridentifikasi dengan benar (garis keriput yang terdeteksi sistem sama dengan pakar). **Specificity** adalah parameter untuk mengukur persentase data negatif yang teridentifikasi dengan benar (sistem tidak mendeteksi objek bukan garis keriput dari kandidat). Tabel hasil uji validasi citra antara sistem dan AVO dapat dilihat pada Tabel 4.2 dan 4.3.

Tabel 4.2 Tabel pengujian sistem (AVO)

N o	Nama File	True Positive	True Negative	False Positive	False Negative	Sensitivity	Specificit y	Accuracy
1	Keriput 1	1	62	2	0	1	0.969	0.969
2	Keriput 2	2	43	0	0	1	1	1
3	Keriput 3	0	78	2	2	0	0.975	0.951
4	Keriput 4	2	73	3	1	0.667	0.961	0.949
5	Keriput 5	2	28	0	1	0.667	1	0.968
6	Keriput 6	2	51	1	1	0.667	0.981	0.964
7	Keriput 7	1	62	2	0	1	0.969	0.969
8	Keriput 8	2	38	0	0	1	1	1
9	Keriput 9	0	65	2	2	0	0.970	0.942
10	Keriput 10	1	38	3	0	1	0.927	0.929
11	Keriput 11	2	55	0	0	1	1	1
12	Keriput 12	0	38	2	1	0	0.95	0.927
13	Keriput 13	2	33	1	0	1	0.971	0.972
14	Keriput 14	1	48	2	0	1	0.96	0.961
15	Keriput 15	0	50	1	0	0	0.980	0.980
16	Keriput 16	0	57	3	0	0	0.95	0.95
17	Keriput 17	2	27	0	0	1	1	1
18	Keriput 18	2	22	0	0	1	1	1
19	Keriput 19	2	40	0	2	0.5	1	0.955
20	Keriput 20	1	27	0	1	0.5	1	0.966
21	Keriput 21	1	46	0	1	0.5	1	0.979
22	Keriput 22	2	46	0	0	1	1	1
23	Keriput 23	3	50	2	2	0.6	0.962	0.930
24	Keriput 24	2	56	2	0	1	0.966	0.967
25	Keriput 25	2	71	1	0	1	0.986	0.986
26	Keriput 26	2	56	0	1	0.667	1	0.983
27	Keriput 27	2	39	0	0	1	1	1
28	Keriput 28	3	21	1	0	1	0.955	0.96
29	Keriput 29	4	46	0	2	0.667	1	0.962
30	Keriput 30	2	40	0	0	1	1	1
31	Keriput 31	2	47	0	2	0.5	1	0.961
32	Keriput 32	2	49	0	4	0.333	1	0.927
33	Keriput 33	2	44	0	0	1	1	1
34	Keriput 34	2	41	0	0	1	1	1
35	Keriput 35	2	44	0	1	0.667	1	0.979
36	Keriput 36	2	62	0	2	0.5	1	0.970

No	Nama File	True Positive	True Negative	False Positive	False Negative	Sensitivity	Specificity	Accuracy
37	Keriput 37	2	20	0	0	1	1	1
38	Keriput 38	3	48	0	13	0.188	1	0.797
49	Keriput 39	2	54	5	3	0.4	0.915	0.875
<b>TOTAL</b>		<b>67</b>	<b>1815</b>	<b>35</b>	<b>42</b>	<b>0.615</b>	<b>0.981</b>	<b>0.961</b>

Perhitungan nilai *sensitivity*, *specificity*, dan *accuracy* dapat dilihat pada Persamaan 2.20. sampai 2.22. Nilai *TP*, *TN*, *FP* dan *FN* yang digunakan adalah nilai total dari *TP*, *TN*, *FP* dan *FN* yang terdapat pada Tabel 4.2. Tabel hasil pengujian dari pengujian dapat dilihat pada Tabel 4.3.

Tabel 4.3 Tabel *confusion matrix* hasil pengujian (AVO)

		Pakar		
		Keriput	Non-Keriput	
Sistem	Keriput	67	35	
	Non-Keriput	42	1815	
		Sensitivity	Specificity	Accuracy
		0.615	0.981	<b>0.961</b>

Tabel 4.2 menjelaskan perbandingan hasil citra dari sistem dengan hasil penandaan oleh AVO, sedangkan Tabel 4.3 menjelaskan nilai *sensitivity* dari sistem sebesar 0,615 atau bernilai 61,5% (dalam satuan persen), hasil yang didapat cukup baik untuk nilai kebenaran deteksi garis keriput, untuk nilai *specificity* dari sistem bernilai 0,981 atau 98,1% (dalam satuan persen), sedangkan untuk nilai *accuracy* yang didapat oleh sistem adalah 0,961 atau 96,1% (dalam satuan persen). Hasil yang didapat sangat baik karena kesalahan sistem dalam mendeteksi objek yang bukan termasuk garis keriput jarang ditemukan. Tabel hasil uji validasi citra antara sistem dan dokter ahli dapat dilihat pada Tabel 4.4 dan 4.5 berikut:

Tabel 4.4 Tabel pengujian sistem (Dokter ahli)

No	Nama File	True Positive	True Negative	False Positive	False Negative	Sensitivity	Specificity	Accuracy
1	Keriput 1	0	63	3	0	0	0.955	0.955
2	Keriput 2	0	45	2	1	0	0.957	0.938
3	Keriput 3	0	78	2	3	0	0.975	0.940
4	Keriput 4	0	75	5	0	0	0.938	0.938

N o	Nama File	True Positive	True Negative	False Positive	False Negative	Sensitivit y	Specificit y	Accuracy
5	Keriput 5	1	29	1	2	0.333	0.967	0.909
6	Keriput 6	0	53	3	0	0	0.946	0.946
7	Keriput 7	0	63	2	0	0	0.969	0.969
8	Keriput 8	0	40	2	0	0	0.952	0.952
9	Keriput 9	0	65	2	1	0	0.970	0.956
10	Keriput 10	0	39	4	0	0	0.907	0.907
11	Keriput 11	2	55	0	0	1	1	1
12	Keriput 12	1	37	1	2	0.333	0.974	0.927
13	Keriput 13	1	34	2	1	0.5	0.944	0.921
14	Keriput 14	1	48	2	2	0.333	0.96	0.925
15	Keriput 15	1	49	0	1	0.5	1	0.980
16	Keriput 16	2	55	1	0	1	0.982	0.983
17	Keriput 17	0	29	2	1	0	0.935	0.906
18	Keriput 18	1	23	1	0	1	0.958	0.96
19	Keriput 19	0	42	2	0	0	0.955	0.955
20	Keriput 20	0	28	1	3	0	0.966	0.875
21	Keriput 21	0	47	1	0	0	0.979	0.979
22	Keriput 22	0	48	2	0	0	0.96	0.96
23	Keriput 23	3	50	1	2	0.6	0.980	0.946
24	Keriput 24	1	57	3	0	1	0.95	0.951
25	Keriput 25	2	71	1	1	0.667	0.986	0.973
26	Keriput 26	2	56	0	0	1	1	1
27	Keriput 27	2	39	1	2	0.5	0.975	0.932
28	Keriput 28	1	23	2	0	1	0.92	0.923
29	Keriput 29	2	48	2	4	0.333	0.96	0.893
30	Keriput 30	0	42	2	1	0	0.955	0.933
31	Keriput 31	2	47	0	2	0.5	1	0.961
32	Keriput 32	2	49	0	3	0.400	1	0.944
33	Keriput 33	0	46	2	3	0	0.958	0.902
34	Keriput 34	2	41	0	2	0.5	1	0.956
35	Keriput 35	2	44	0	0	1.000	1	1
36	Keriput 36	1	63	1	5	0.167	0.984	0.914
37	Keriput 37	2	20	0	2	0.5	1	0.917
38	Keriput 38	3	48	0	7	0.300	1	0.879
49	Keriput 39	2	54	2	6	0.25	0.964	0.875
<b>TOTAL</b>		<b>39</b>	<b>1843</b>	<b>58</b>	<b>57</b>	<b>0.406</b>	<b>0.969</b>	<b>0.942</b>

Perhitungan nilai *sensitivity*, *specificity*, dan *accuracy* dapat dilihat pada Persamaan 2.20. sampai 2.22. Nilai *TP*, *TN*, *FP* dan *FN* yang digunakan adalah nilai total dari *TP*, *TN*, *FP* dan *FN* yang terdapat pada Tabel 4.4. Tabel hasil pengujian dari pengujian dapat dilihat pada Tabel 4.5.

Tabel 4.5 Tabel *confusion matrix* hasil pengujian (Dokter ahli)

		Pakar		
		Keriput	Non-Keriput	
Sistem	Keriput	39	58	
	Non-Keriput	57	1843	
		Sensitivity	Specificity	Accuracy
		0.406	0.969	0.942

Tabel 4.4 menjelaskan perbandingan hasil citra dari sistem dengan hasil penandaan oleh dokter ahli, sedangkan Tabel 4.5 menjelaskan nilai *sensitivity* dari sistem sebesar 0,406 atau bernilai 40,6% (dalam satuan persen), hasil yang didapat kurang baik untuk nilai kebenaran deteksi garis keriput karena nilai kebenaran tidak mampu melewati 50%, untuk nilai *specificity* dari sistem bernilai 0,969 atau 96,9% (dalam satuan persen), hasil yang didapat sangat baik karena kesalahan sistem dalam mendeteksi objek yang bukan termasuk garis keriput jarang ditemukan, sedangkan untuk nilai *accuracy* yang didapat oleh sistem adalah 0,942 atau 94,2% (dalam satuan persen).

Kesimpulan dari kedua pengujian di atas adalah, sistem mendapat nilai pengujian yang lebih baik dari hasil pengujian dengan AVO daripada hasil pengujian dengan dokter ahli. Hal ini dikarenakan perbedaan letak dan panjang penandaan garis keriput, saat citra digabungkan tidak ada area yang beririsan. Selain itu dokter ahli berpendapat bahwa garis keriput pada citra tidak semua dapat dikatakan sebagai garis keriput akibat penuaan dini (garis keriput normal), sehingga ada beberapa citra yang tidak ditandai sama sekali garis keriputnya oleh dokter ahli. Hasil performa sistem keseluruhan dapat dilihat pada Tabel 4.6.

Tabel 4.6 Tabel performa sistem

Proses	Rata-Rata Performa Sistem dalam Satuan Waktu (Detik)
Resize	0.008
Segmentasi Kulit	1.090
Perbaikan Citra	0.521

Proses	Rata-Rata Performa Sistem dalam Satuan Waktu (Detik)
Ambil Area Keriput	1.263
Ekstraksi Ciri	2.154
Marking	0.211
<b>Total</b>	<b>5.247</b>

Tabel 4.6 menunjukkan rata-rata performa sistem, sistem memiliki kecepatan keseluruhan 5,247 detik, nilai tersebut termasuk nilai performa sistem yang cukup cepat mengingat banyak proses yang dilakukan oleh sistem. Sistem berjalan lambat jika citra yang digunakan masih berukuran besar meskipun sudah melalui tahap *Resize*, proses yang paling cepat dilakukan sistem adalah proses *Resize*, sedangkan proses yang paling lambat dilakukan sistem adalah proses ekstraksi ciri, karena sistem harus mendeteksi garis keriput di masing-masing area keriput yang telah diambil.

#### 4.11 Penyebab Kegagalan Sistem

Tidak semua citra uji dapat diproses sempurna oleh sistem, meskipun citra yang digunakan merupakan citra ideal yang telah memenuhi syarat sebagai citra input. Hal itu disebabkan oleh beberapa hal. Pertama, sistem gagal dalam mendeteksi citra wajah akibat kesamaan warna antara objek kulit dengan objek non-kulit, faktor efek pencahayaan yang terlalu terang pada foto juga menyebabkan wajah tidak terdeteksi. Kegagalan proses segmentasi citra dapat dilihat pada Gambar 4.56.



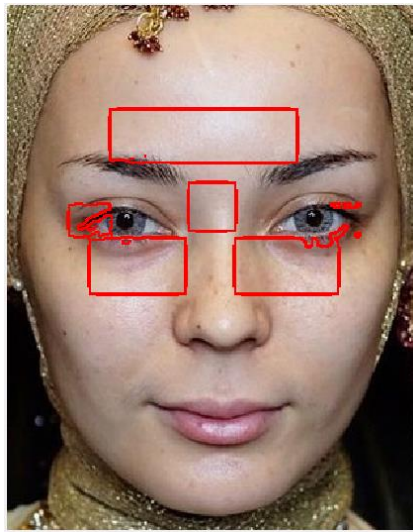
(a)



(b)

Gambar 4.56 Citra hasil segmentasi: (a). Citra asli (b). Hasil citra tidak sempurna

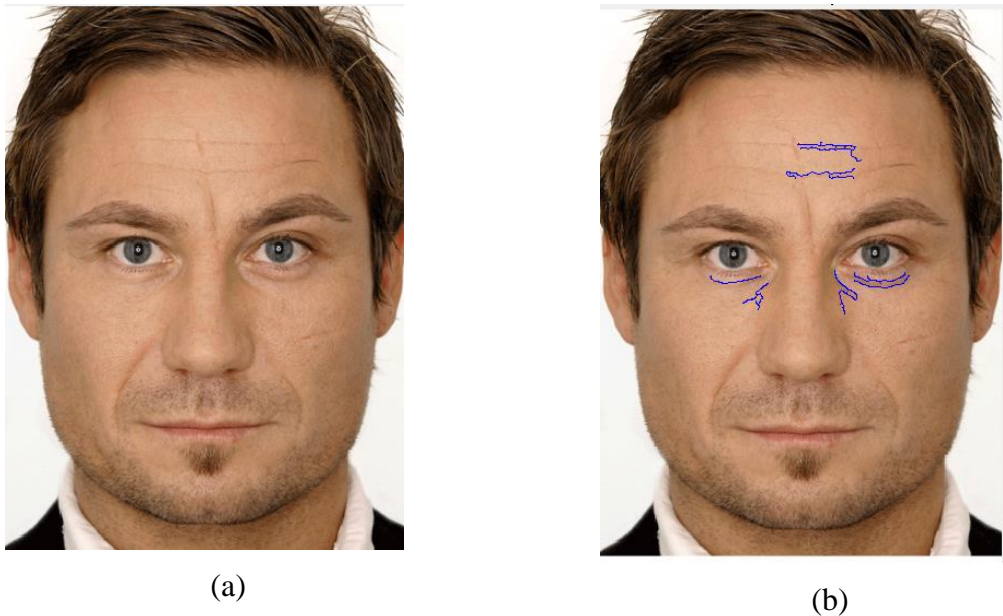
Kedua, sistem gagal dalam proses mengambil area keriput dengan tepat, hal ini disebabkan oleh faktor pengambilan area yang masih bersifat statis, sehingga ukuran tidak bisa menyesuaikan dan tidak semua citra cocok dengan ukuran yang digunakan. Namun, karena pengambilan area keriput tidak bergeser terlalu jauh, citra tetap bisa digunakan sebagai data uji. Selain itu, proses pemotongan gambar yang kurang tepat juga bisa menyebabkan kesalahan dalam mengambil area keriput. Kegagalan pengambilan area keriput dapat dilihat pada Gambar 4.57.



Gambar 4.57 Citra pengambilan area keriput tidak sempurna

Dan yang terakhir, sistem gagal mendeteksi garis keriput pada area yang telah diambil, hal ini disebabkan oleh faktor pengambilan area yang kurang tepat atau luas area garis keriput yang terdeteksi sama dengan luas area garis bukan keriput (*noise*). Kegagalan proses ekstraksi ciri garis keriput dapat dilihat pada Gambar 4.58.





Gambar 4.58 Citra hasil ekstraksi ciri (a). Citra asli (b). Citra hasil deteksi keriput tidak sempurna

#### 4.12 Kelebihan dan Kekurangan Sistem

Adapun kelebihan dari sistem deteksi keriput pada citra wajah dengan teknik pengolahan citra adalah sebagai berikut:

- a. Dapat membedakan objek wajah dan *background* citra dengan seleksi warna kulit
- b. Dapat mendeteksi garis keriput di semua *area* keriput pada wajah
- c. Dapat mendeteksi garis keriput ringan akibat penuaan dini pada wajah
- d. Memberikan informasi mengenai ciri warna kulit dan luas *area* garis keriput.

Sedangkan untuk kekurangan dari sistem deteksi keriput pada citra wajah dengan teknik pengolahan citra adalah sebagai berikut:

- a. Citra harus melewati proses *Cropping* manual terlebih dahulu agar pengambilan area keriput sesuai dengan ukuran yang telah ditentukan
- b. Masih terdapat area-area tertentu yang tidak terdeteksi garis keriput karena citra yang digunakan tidak memiliki ciri keriput, di antaranya area sisi tengah mata, area sisi kanan dan kiri mata, serta area kantung mata kanan dan kiri.
- c. Kualitas citra yang digunakan tidak semua berkualitas bagus dan memenuhi syarat sebagai citra dengan ciri keriput penuaan dini, karena citra yang digunakan didapat secara random dari Internet.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Dari semua proses yang telah dilakukan pada tugas akhir ini, mulai dari tahap analisis, perancangan hingga pengujian. Kesimpulan yang didapat dari penelitian ini adalah:

- a. Pada pengujian secara umum, program memiliki tingkat keberhasilan yang cukup tinggi untuk citra dengan kondisi ideal. Kondisi ideal yang dimaksud adalah citra *RGB* yang mengandung wajah menghadap lurus kedepan dan ciri keriput, serta citra dengan pencahayaan yang baik.
- b. Sistem mampu memisahkan bagian wajah dan *background* dengan baik untuk citra yang memenuhi syarat sebagai citra input dengan proses segmentasi citra.
- c. Sistem mampu mengambil area keriput dengan tepat untuk citra yang sudah melewati proses *Cropping* terlebih dahulu dengan metode *Presentase*.
- d. Sistem mampu mendeteksi dan menampilkan letak garis keriput pada wajah untuk citra yang memenuhi syarat sebagai citra input, tetapi ketepatan deteksi garis keriput belum sempurna.
- e. Luas (*Area*) objek garis yang diindikasikan sebagai garis keriput pada masing-masing area adalah area dahi 111-137, area kantung mata kiri 36-87 dan area kantung mata kanan 56-138. Untuk area sisi tengah mata, sisi kiri dan kanan mata tidak ditemukan objek yang diindikasikan sebagai garis keriput pada citra uji, sehingga diberi nilai 0.
- f. Penelitian ini dapat dijadikan acuan untuk penelitian selanjutnya karena memuat informasi mengenai metode dan rentang nilai *threshold* yang digunakan untuk mendeteksi objek wajah dan keriput.
- g. Nilai *sensitivity* yang diperoleh dari pengujian dengan AVO adalah 61,5%, nilai *specificity* adalah 98,1%, dan nilai *accuracy* adalah 96,1%. Sedangkan untuk nilai *sensitivity* yang diperoleh dari pengujian dengan dokter ahli adalah 40,6%, nilai *specificity* adalah 96,9% dan nilai *accuracy* adalah 94,2%..
- h. Nilai *sensitivity* di bawah 50% oleh dokter disebabkan oleh penandaan letak dan panjang garis keriput antara citra sistem dengan citra dokter yang berbeda, selain itu proses ekstraksi ciri yang kurang maksimal dan penggunaan citra yang tidak memiliki garis keriput di semua area menyebabkan garis keriput tidak terdeteksi.
- i. Performa sistem cukup cepat yaitu 5,247 detik dari keseluruhan proses yang dilakukan.

## 5.2 Saran

Pada penelitian ini terdapat banyak kekurangan yang perlu diperbaiki, adapun saran untuk penelitian selajutnya adalah:

- a. Penelitian ini menggunakan citra yang diambil secara random dari Internet. Untuk penelitian selajutnya jika mempunyai waktu pengerjaan yang lebih lama sebaiknya menggunakan citra yang diambil sendiri oleh peneliti agar syarat citra sesuai dengan permasalahan yang diteliti, untuk keriput dapat menggunakan objek wanita umur 30 tahun yang pekerjaannya sering terpapar sinar matahari.
- b. Proses *Cropping* citra dapat dilakukan secara otomatis di sistem matlab, sehingga citra dapat langsung diinput ke dalam sistem.
- c. Pengambilan area keriput di wajah dapat konsisten untuk semua citra, sehingga citra tidak perlu melewati proses *Cropping* terlebih dahulu.
- d. Untuk proses ekstraksi ciri garis keriput dapat menambahkan ekstraksi ciri garis keriput lain seperti ciri warna dan kedalaman garis keriput.

## DAFTAR PUSTAKA

- Acharya, T., & Ray, A.K. 2005. Image Processing: Principles and Applications. John Wiley & Sons, Inc., Canada.
- Alevelbiology. (2010). Structure of Collagen Tissue & Molecule \_ A-Level Biology Revision Notes. Retrieved April 15, 2018, from <https://alevelbiology.co.uk/notes/structure-of-collagen/>
- Alodokter. (2016). Keriput - Gejala, penyebab dan mengobati - Alodokter. Retrieved April 15, 2018, from <https://www.alodokter.com/keriput>
- Atmaja, N. S., Marwiyah, & Setyowati, E. (2012). Pengaruh Kosmetika Anti Aging Wajah Terhadap Hasil Perawatan Kulit Wajah. *Journal of Beauty and Beauty Health Education*, 1(1).
- August, J., Saida, S. J., Srinivas, L., Sivaram, R., & Guntur, T. (2012). An Efficient K-Means and C-Means Clustering Algorithm for Image Segmentation, *I(2278)*, 84–87.
- Batool, N., & Chellappa, R. (2012). Modeling and Detection of Wrinkles in Aging Human Faces Using Marked Point Processes. *Computer Vision -- ECCV 2012*, 7584, 178–188.
- Batool, N., & Chellappa, R. (2014). Fast detection of facial wrinkles based on Gabor features using image morphology and geometric constraints. *Pattern Recognition*, 48, 1–32.
- Cahyan, P. A., Aswin, M., & Mustofa, A. (2011). SEGMENTASI CITRA DIGITAL DENGAN MENGGUNAKAN ALGORITMA WATERSHED DAN LOWPASS FILTER SEBAGAI PROSES AWAL. *SEGMENTASI CITRA*, 1(1202), 1–6.
- Epilighnewskin. (2015). PDO Threads BEAUTY LIFT - Epiligh New Skin Clinic. Retrieved April 15, 2018, from <https://www.epilighnewskin.co.uk/?treatments=facial-lifting-threads>
- ETCADM. (2017). Modalitas Atasi Keriput – apps. Retrieved April 15, 2018, from [www.ethicaldigest.com/laporan-khusus/modalitas-atasi-keriput](http://www.ethicaldigest.com/laporan-khusus/modalitas-atasi-keriput)
- Ginting, E. D. (2010). Deteksi Tepi Menggunakan Metode Canny Dengan Matlab Untuk Membedakan Uang Asli Dan Uang Palsu. *Gunadarma*, 22(3), 2–13.
- Hayatin, N. (2016). Klasifikasi kelompok usia berdasarkan citra wajah menggunakan algoritma neural network dengan fitur face anthropology dan kedalaman kerutan. *TEKNOLOGI*, 6(2), 61–67.
- Hidayatno, A., Isnanto R, R., & Niam, B. (2015). Analisis Deteksi Tepi Pada Citra Berdasarkan Perbaikan Kualitas Citra. *Universitas Diponegoro*, 2–7.
- Hornig, W. B., Lee, C. P., & Chen, C. W. (2001). Classification of age groups based on facial

- features. *Tamkang Journal of Science and Engineering*, 4(3), 183–192.
- Jana, R., Datta, D., & Saha, R. (2015). Age estimation from face image using wrinkle features. *Procedia Computer Science*, 46(2015), 1754–1761.
- Junior, R. A., Nurhasanah, & Sanubary, I. (2014). Perbandingan Penggunaan Beberapa Metode Deteksi Tepi pada Pengolahan Citra Radiologi Fraktur Tulang. *Prisma Fisika*, 5(3), 117–121.
- Jusuf, N. K. (2005). Kulit Menua. *Majalah Kedokteran Nusantara*, 38(2), 184–189.
- Kalangi, S. J. R. (2013). Histofisiologi kulit. *Jurnal Biomedik (JBM)*, 5(1), 12–19.
- Kolkur, S., Kalbande, D., Shimpi, P., Bapat, C., & Jatakia, J. (2016). Human Skin Detection Using RGB, HSV and YCbCr Color Models, 137(1), 324–332.
- Masatu, D. A., Soesanti, I., & Nugroho, H. A. (2014). Penerapan Algoritma Kompresi Jpeg Dan Metode Fuzzy C-Means Pada Kompresi Citra Berbasis. *Penelitian Teknik Elektro Dan Teknologi Informasi*, 1(1), 7–11.
- Medicinesia. (2010). Keriput – Medicinesia. Retrieved February 28, 2018, from <https://www.medicinesia.com/kedokteran-klinis/kulit/keriput/>
- Mujahidin, S. (2012). Aplikasi Perhitungan Jumlah Orang Dalam Satu Foto. *Tugas Akhir*. Yogyakarta: Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.
- Mulyani, W. S., & Propeliena, R. (2014). SEGMENTASI CITRA WARNA KULIT MANUSIA DENGAN DETEKSI WARNA HSV UNTUK MENDETEKSI GAMBAR PORNOGRAFI.
- Muwardi, F., Fadlil, A., Studi, P., Elektro, T., Industri, F. T., Dahlan, U. A., & Umbulharjo, S. H. (2017). Pengolahan Citra Dan Pengklasifikasi Jarak, 3(2), 124–131.
- Nafi'iyah, N. (2015). Algoritma Kohonen dalam Mengubah Citra Graylevel Menjadi Citra Biner. *Jurnal Ilmiah Teknologi Informasi Asia*, 9(2), 49–55.
- Nugraha, T. A., Soim, S., & Hadi, I. (2017). HSV PADA VIRTUAL MOUSE. *Seminar Nasional Teknologi Informasi, Bisnis, Dan Desa 2017*, 1(1), 446–452.
- Nur, A. (2018). SEL POLEN BERDASARKAN CITRA MIKROSKOPIS DIGITAL. *Tugas Akhir*. Yogyakarta: Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.
- Pramana, C. J. (2014). Implementasi Metode Thresholding dan Metode Regionprops untuk Mendeteksi Marka Jalan Secara Live Video. *Jurnal Universitas Dian Nuswantoro Semarang*, 1(1), 1–9.
- Swedia, E. R., & Cahyanti, M. (2010). Algoritma Tranformasi Ruang Warna. *Visual Bassic6*,

*Visual Basic.NET Dan Java*, 1–7.

Valentino, F., Adji, Teguh, B., & Permanasari, Adhistya, E. (2017). Komparasi Metode Decision Tree dan K-Means Clustering Dalam Mengatasi Masalah Cold-start Pengguna Baru, 268–273.

Wahyuningtyas, R. S., Pratiwi, H. S., Studi, P., Informatika, T., Teknik, F., & Tanjungpura, U. (2015). Sistem Pakar Penentuan Jenis Kulit Wajah Wanita Menggunakan Metode Naïve Bayes. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, 1(1), 2–6.

Wibowo, J. S. (2011). Deteksi dan Klasifikasi Citra Berdasarkan Warna Kulit Menggunakan HSV. *Jurnal Teknologi Informasi DINAMIK*, 16(2), 118–123.

## LAMPIRAN

FORM-TA/TF-A3



UNIVERSITAS ISLAM INDONESIA  
Jurusan Teknik Informatika FTI

### SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Fadhilah Abriyani

No. Mhs. : \_\_\_\_\_

Judul TA : Deteksi Keriput pada Citra Wajah

- 1 Karena batasananya kulit orang asia, tambahkan data training / uji yang berkulit putih (orang china / Jepang), tidak hanya yang sawo matang.
- 2 Untuk gambar uji lain (dari internet), misal gambar anak kecil, akurasi deteksi perlu ditingkatkan.

Nilai kemajuan Tugas Akhir: \_\_\_\_\_ (0 - 100)  
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 5 Juni 2018

Dosen,

  
Septia Rani  
(nama terang)

Dilampirkan pada Laporan TA yang diajukan untuk pendadaran