

**APLIKASI PENDETEKSI JERAWAT DI WAJAH
DENGAN MENGGUNAKAN TEKNIK
PENGOLAHAN CITRA PADA FOTO**



Disusun Oleh :

N a m a : Dwi Prasetyo

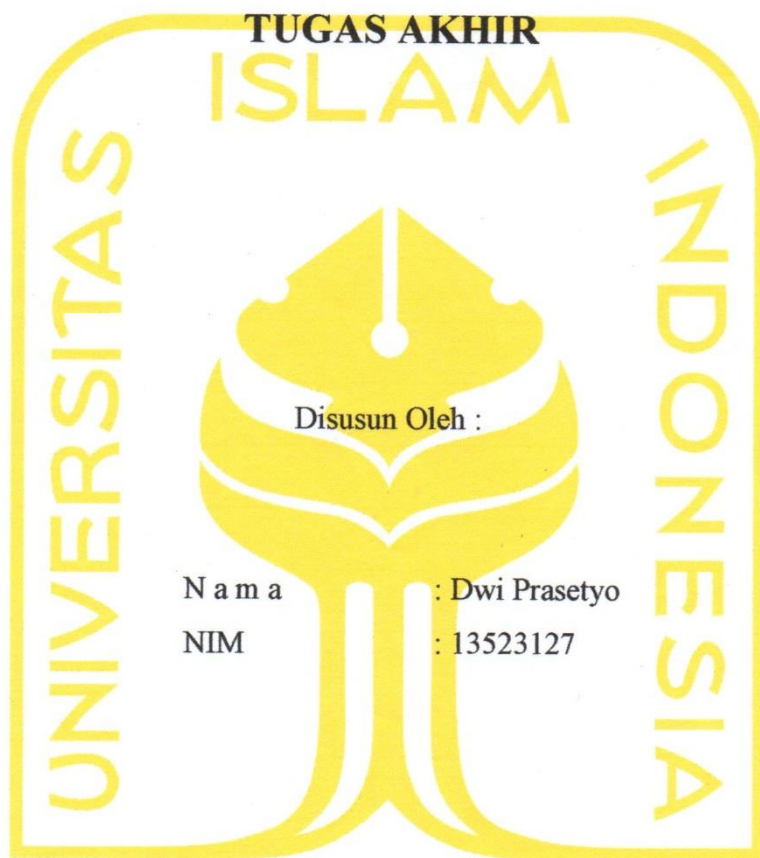
NIM : 13523127

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

2018

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**APLIKASI PENDETEKSI JERAWAT DI WAJAH
DENGAN MENGGUNAKAN TEKNIK
PENGOLAHAN CITRA PADA FOTO**



الجامعة الإسلامية
Yogyakarta, 25 Juni 2018

Pembimbing 1,

Pembimbing 2,

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

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

HALAMAN PENGESAHAN DOSEN PENGUJI

**APLIKASI PENDETEKSI JERAWAT DI WAJAH
DENGAN MENGGUNAKAN TEKNIK
PENGOLAHAN CITRA PADA FOTO
TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk
memperoleh gelar Sarjana Teknik Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 1 Agustus 2018

Tim Penguji

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



Anggota 1

Arrie Kurniawardhani, S.Si., M.Kom



Anggota 2

Rahadian Kurniawan, S.Kom., M.Kom..



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 di bawah ini:

Nama : Dwi Prasetyo

NIM : 13523127

Tugas akhir dengan judul:

APLIKASI PENDETEKSI JERAWAT DI WAJAH DENGAN MENGGUNAKAN TEKNIK PENGOLAHAN CITRA PADA FOTO

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, 25 Juni 2018



(Dwi Prasetyo)

HALAMAN PERSEMBAHAN

Tugas Akhir ini saya persembahkan untuk **Ibu** dan **Bapak** yang selalu memberikan dukungan dan kepercayaan, terima kasih banyak.

HALAMAN MOTTO

Fainna ma'al 'usri yusron. Innama'al 'Usrii Yusran

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan. Sesungguhnya sesudah kesulitan itu ada kemudahan”

(QS. Al-Insyirah : 5-6)

When I have Fitbar, I feel worry free~

(Iklan Fitbar, 2017)

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah, penulis panjatkan puji dan syukur kehadirat Allah SWT yang telah memberikan rahmat, hidayah, dan karunia-Nya, sehingga Laporan Tugas Akhir ini dapat penulis selesaikan. Tak lupa shalawat serta salam penulis haturkan kepada junjungan kita Nabi Muhammad SAW, yang telah membawa kita dari zaman jahiliyah menuju jaman terang benderang.

Tugas Akhir dengan judul “**Aplikasi Pendeteksi Jerawat di Wajah Dengan Menggunakan Teknik Pengolahan Citra pada Foto**” dibuat sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di Program Studi Teknik Informatika Universitas Islam Indonesia. Tugas Akhir ini dapat terselesaikan berkat bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada:

1. Bapak Fathul Wahid, selaku Rektor Universitas Islam Indonesia.
2. Bapak Hari Purnomo, selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Raden Teduh Dirgahayu, selaku Ketua Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Orang tua dan keluarga penulis atas segala doa dan dukungan selama penulis mengerjakan Tugas Akhir.
5. Ibu Izzati Muhimmah, selaku Dosen Pembimbing 1 Tugas Akhir di Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
6. Ibu Arrie Kurniawardhani, selaku Dosen Pembimbing 2 Tugas Akhir di Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
7. Bapak Kholid Haryono, selaku Dosen Pembimbing Akademik di Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
8. Bapak Hari Setiaji, selaku *Project Manager* dan Dosen Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
9. Bapak Rahadian Kurniawan, selaku Dosen Penguji Tugas Akhir di Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
10. Seluruh Dosen Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang telah memberi ilmu.

11. Fadhillah Abriyani dan Nurul Fatikah Muchlis yang turut membantu proses pembuatan Laporan Tugas Akhir.
12. Segenap keluarga besar teman-teman Program Studi Teknik Informatika Universitas Islam Indonesia yang saya banggakan.

Terimakasih kepada semua pihak yang telah membantu terselesaikannya penulisan laporan Tugas Akhir ini, semoga segala bentuk dukungan dibalas oleh Allah SWT.

Penulis menyadari bahwa laporan ini masih belum sempurna, karena keterbatasan kemampuan dan pengalaman. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan Laporan Tugas Akhir ini. Akhir kata, penulis berharap agar laporan ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 25 Juni 2018

(Dwi Prasetyo)

SARI

Kosmetik merupakan salah satu produk yang memiliki banyak konsumen yang mayoritasnya adalah kaum hawa, tak heran jika semakin banyak brand-brand kosmetik bermunculan baik brand global maupun *brand* lokal. AVO merupakan salah satu produsen kosmetik asal Indonesia yang sedang berkembang. Sebagai produsen kosmetik yang baru berdiri, menarik konsumen merupakan sebuah hal yang wajib dilakukan. Salah satunya adalah skin advisor yang terdapat pada website AVO sebagai media interaksi konsumen untuk mengevaluasi produk AVO yang sesuai dengan kondisi kulit konsumen. Salah satu fitur skin advisor yang belum diterapkan adalah menganalisis jerawat pada citra secara otomatis. Dengan menggunakan teknologi pengolahan citra digital, karakteristik dari jerawat pada citra dapat dimodelkan untuk digunakan sebagai media pengenalan atau deteksi.

Metode yang digunakan dalam penelitian ini berbasis pada segmentasi kulit wajah dan jerawat yang memanfaatkan morfologi (bentuk) yang berfokus pada objek jerawat. Penelitian ini memanfaatkan metode *connected-component labeling* untuk proses segmentasi serta *filter Laplacian of Gaussian* untuk mencari kandidat jerawat. Luas *area*, *eccentricity*, *mean intensity* serta standar deviasi dimanfaatkan sebagai parameter *threshold* untuk ekstraksi ciri kandidat jerawat hingga proses terakhir *marking* menggunakan *filter canny*. Pengujian dilakukan dengan *single decision threshold* untuk mengukur tingkat keberhasilan sistem dalam mendeteksi jerawat. Hasil pengujian deteksi jerawat didapat nilai *sensitivity* 0,464 dan nilai *specificity* sebesar 0,971.

Kata Kunci : Deteksi Jerawat, Laplacian of Gaussian, *sensitivity*, *specificity*

GLOSARIUM

<i>Background</i>	Latar belakang citra
<i>Flowchart</i>	Diagram Alir
<i>Grayscale</i>	Citra dengan format warna keabuan
<i>Interface</i>	Antarmuka pengguna dengan sistem
<i>Input</i>	Masukkan
<i>Labeling</i>	Pemberian tanda pada piksel citra biner
<i>Output</i>	Keluaran
<i>Pixel</i>	Representasi sebuah titik kecil dalam sebuah citra

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO.....	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.....	2
1.5. Manfaat.....	2
1.6. Metode Penelitian.....	2
1.7. Sistematika Penulisan.....	3
BAB II LANDASAN TEORI.....	5
2.1. Jerawat.....	5
2.2. Pengolahan Citra	6
2.3. Model Warna.....	7
2.3.1. RGB.....	7
2.3.2. HSV	7
2.4. Citra Biner	9
2.5. <i>Connected-Component Labeling</i>	9
2.6. Segmentasi Kulit	10
2.7. Laplacian of Gaussian	11
2.8. Ekstraksi Ciri.....	13
2.8.1. Area dan <i>Perimeter</i>	13

2.8.2. <i>Eccentricity</i>	13
2.9. Standar Deviasi.....	14
2.10. Teori Pengujian Sistem	14
BAB III METODE PENELITIAN	17
3.1. Pengumpulan Data	17
3.2. Analisis Kebutuhan Sistem	18
3.2.1. Analisis Kebutuhan <i>Input</i>	18
3.2.2. Analisis Kebutuhan Proses.....	19
3.2.3. Analisis Kebutuhan <i>Output</i>	19
3.2.4. Analisis Kebutuhan Antarmuka	19
3.3. Rancangan Antarmuka	19
3.4. Analisis Pengujian Sistem	21
BAB IV ARSITEKTUR SISTEM	22
4.1. Gambaran Umum	22
4.3. Resize Citra	23
4.4. Segmentasi Kulit	25
4.5. Perbaikan Citra.....	26
4.6. Ekstraksi Ciri.....	27
4.7. <i>Marking</i>	33
BAB V HASIL DAN PEMBAHASAN	35
5.1. Implementasi Perangkat Lunak	35
5.1.1. <i>Resize Citra</i>	35
5.1.2. Segmentasi Kulit	36
5.1.3. Perbaikan Citra.....	37
5.1.4. Ekstraksi Ciri.....	38
5.1.5. <i>Marking</i>	40
5.2. Implementasi Antarmuka	41
5.3. Pengujian Sistem	43
5.4. Pengujian Kecepatan Sistem	45
5.5. Perbandingan Dengan Sistem Sejenis	46
5.6. Kelebihan dan Kekurangan Sistem	48
BAB VI KESIMPULAN DAN SARAN	50
6.1. Kesimpulan.....	50
6.2. Saran.....	50

DAFTAR PUSTAKA.....	51
LAMPIRAN.....	52

DAFTAR TABEL

Tabel 2.1 Ciri Klinis Jerawat	6
Tabel 2.2 Tabel <i>Contingency</i>	15
Tabel 4.1 Uji Parameter <i>Resize</i>	24
Tabel 4.2 Perbandingan Kecepatan Sistem.....	24
Tabel 4.3 Ciri Jerawat Data <i>Training I</i>	28
Tabel 4.4 Ciri Jerawat Data <i>Training II</i>	29
Tabel 4.5 Parameter Percobaan Ciri Jerawat	29
Tabel 4.6 Hasil Uji Percobaan Ekstraksi Ciri	30
Tabel 4.7 Parameter Area dan <i>Eccentricity</i>	30
Tabel 4.8 Hasil Uji Perubahan Parameter Area dan <i>Eccentricity</i>	31
Tabel 4.9 Ciri Jerawat Yang Tidak Terdeteksi	31
Tabel 4.10 Parameter Intensitas Warna	32
Tabel 4.11 Hasil Pengujian Ekstraksi Ciri	33
Tabel 4.12 Parameter Ekstraksi Ciri	33
Tabel 4.13 Perbandingan Kecepatan Proses <i>Edge Detection</i>	34
Tabel 5.1 Hasil Pengujian Sistem	43
Tabel 5.2 Tabel <i>Contingency</i> Pengujian	45
Tabel 5.3 Hasil Pengujian Kinerja Sistem	45
Tabel 5.4 Perbandingan Hasil Sistem Dengan Ponds.....	47

DAFTAR GAMBAR

Gambar 2.1 Jerawat	5
Gambar 2.2 Sistem koordinat model warna HSV	7
Gambar 2.3 Hasil <i>Labeling</i> ; (a) Citra Biner. (b) Citra Berlabel	9
Gambar 2.4 Segmentasi Wajah.....	10
Gambar 2.5 Bentuk LoG; (a) Kernel. (b) Plot 3D	12
Gambar 2.6 Hasil Deteksi Citra; (a) <i>Grayscale</i> . (b) LoG	12
Gambar 3.1 Data Citra; (a) Hadap Depan. (b) Hadap Samping	17
Gambar 3.2 Data <i>Training</i> ; (a) I. (b) II.....	17
Gambar 3.3 Kebutuhan <i>Input</i>	18
Gambar 3.4 Kebutuhan <i>Output</i>	19
Gambar 3.5 Rancangan Antarmuka	20
Gambar 3.6 Rancangan <i>File Input</i>	21
Gambar 4.1 <i>Flowchart</i> Utama	22
Gambar 4.2 Hasil Deteksi; (a) Citra Awal. (b) Citra <i>Resize</i>	23
Gambar 4.3 Perbandingan Citra Hasil Segmentasi; (a) Metode HSV. (b) Metode Otsu.....	25
Gambar 4.4 Perbaikan Citra Data <i>Training</i> I; (a) Citra LoG. (b) Citra Penandaan Jerawat....	27
Gambar 4.5 Perbaikan Citra Data <i>Training</i> II; (a) Citra LoG. (b) Citra Penandaan Jerawat ..	27
Gambar 4.6 Hasil Percobaan Ekstraksi Ciri; (a) Data <i>Training</i> I. (b) Data <i>Training</i> II	30
Gambar 4.7 Hasil Ekstraksi Ciri; (a) Data <i>Training</i> I dan (b) Data <i>Training</i> II.....	32
Gambar 4.8 Perbandingan <i>Edge Detection</i> ; (a) Canny, (b) Prewitt dan (c) Sobel	33
Gambar 5.1 Kode Program <i>Resize</i> Citra	35
Gambar 5.2 Kode Program Segmentasi Kulit.....	36
Gambar 5.3 Tahap Segmentasi Kulit; (a) <i>Grayscale</i> . (b) Biner. (c) <i>Imfill</i> . (d) Hasil.....	37
Gambar 5.4 Kode Program Perbaikan Citra	37
Gambar 5.5 Tahap Perbaikan Citra; (a) Citra <i>Grayscale</i> (b) Citra LoG.....	38
Gambar 5.6 Kode Program Ekstraksi Ciri Luas dan Bentuk	38
Gambar 5.7 Hasil Ekstraksi Ciri Luas dan Bentuk	39
Gambar 5.8 Kode Program Ekstraksi Ciri Warna	39
Gambar 5.9 Hasil Ekstraksi Ciri Warna	40
Gambar 5.10 Kode Program <i>Marking</i>	40
Gambar 5.11 Tahap <i>Marking</i> ; (a) Canny. (b) Hasil.....	41
Gambar 5.12 Antarmuka Kosong	41

Gambar 5.13 <i>File Input</i>	42
Gambar 5.14 Antarmuka Aktif	42
Gambar 5.15 Perbandingan Penandaan; (a) Pakar. (b) Sistem	43
Gambar 5.16 Perbandingan Hasil Deteksi Data 1; (a) Sistem. (b) Ponds.....	46
Gambar 5.17 Grafik Perbandingan Hasil Deteksi Sistem Dengan Ponds	47
Gambar 5.18 Perbandingan Hasil Deteksi Data 2; (a) Sistem. (b) Ponds.....	48

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kosmetik merupakan salah satu produk yang memiliki banyak konsumen yang mayoritasnya adalah kaum hawa, tak heran jika semakin banyak *brand-brand* kosmetik bermunculan baik *brand* global maupun *brand* lokal. AVO merupakan salah satu produsen kosmetik asal Indonesia yang sedang berkembang, meskipun masih anak baru di dunia kosmetik tetapi nama AVO sudah cukup dikenal masyarakat, terbukti dari banyaknya penjualan produk yang mencapai 2000 paket per bulannya (Dethira, 2016).

Sebagai produsen kosmetik yang baru berdiri, menarik konsumen merupakan sebuah hal yang wajib dilakukan. Banyak cara yang sudah dilakukan untuk menarik hati konsumen, diantaranya selalu menghadirkan produk-produk baru yang dibutuhkan oleh para konsumen, memanfaatkan media sosial sampai menggandeng *public figure* untuk mempromosikan produk AVO, dan memberikan layanan konsultasi secara *offline* maupun *online* untuk para konsumennya.

Selain itu, AVO juga sudah memiliki website yang di dalamnya terdapat fitur Skin Advisor. Skin Advisor merupakan salah satu fitur yang digunakan sebagai media interaksi konsumen untuk mengevaluasi produk AVO yang sesuai dengan kondisi kulit konsumen, namun prosesnya masih dilakukan dengan cara menjawab beberapa pertanyaan. Sementara salah satu produsen kosmetik yang sudah ternama yang merupakan pesaing AVO, sudah memiliki website dengan fitur Skin Advisor yang mampu menganalisis masalah kulit secara otomatis melalui foto konsumen.

Salah satu analisis masalah kulit yang terdapat pada fitur Skin Advisor yang dimiliki oleh pesaing AVO adalah deteksi jerawat melalui foto wajah yang diunggah sendiri oleh konsumen. Untuk semakin memanjakan dan menarik konsumen yang lebih banyak, AVO ingin mengikuti jejak pesaingnya untuk membangun sebuah fitur Skin Advisor yang dapat menganalisis masalah kulit khususnya jerawat pada wajah konsumen secara otomatis.

Dari penjabaran masalah di atas, penulis mendapat kesempatan untuk berkolaborasi dengan pihak AVO untuk melakukan penelitian dan pembangunan aplikasi yang dapat mendeteksi masalah kulit wajah khususnya jerawat pada foto wajah secara otomatis. Nantinya aplikasi ini akan dikembangkan dan diterapkan di website AVO sebagai fitur Skin Advisor.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, berikut masalah-masalah yang dapat dirumuskan:

1. Bagaimana memisahkan wajah dengan *background* pada citra?
2. Informasi apa yang digunakan untuk mengenali objek jerawat?
3. Metode apa yang digunakan dalam pengenalan jerawat pada citra digital?

1.3. Batasan Masalah

Untuk menjaga fokus penelitian Tugas Akhir ini, berikut batasan masalah yang perlu diperhatikan:

1. Penelitian memfokuskan jerawat pada wajah.
2. Posisi wajah menghadap ke depan atau samping dengan pencahayaan yang merata.
3. Penelitian memfokuskan warna kulit putih hingga kuning langsat.

1.4. Tujuan

Berdasarkan rumusan masalah yang telah ditetapkan, tujuan penelitian ini adalah menghasilkan aplikasi yang dapat memberikan informasi letak jerawat di wajah berupa penandaan pada citra digital.

1.5. Manfaat

Manfaat dari penelitian ini adalah terciptanya metode serta informasi yang digunakan untuk mengenali jerawat pada citra digital, sehingga dapat digunakan untuk mendeteksi masalah kulit lain.

1.6. Metode Penelitian

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

a. Tahap Pengumpulan Data

1. Observasi

Tahap observasi dilakukan dengan mengumpulkan foto-foto yang memenuhi kualifikasi untuk mengamati pola-pola yang ada, sehingga dapat dirumuskan gambaran umum dari permasalahan.

2. Studi Pustaka

Studi pustaka dilakukan dengan mencari berbagai referensi yang berkaitan dengan penelitian yang dilakukan. Adapun topik-topik yang menjadi referensi meliputi pengolahan citra digital, komputer visi, pemodelan kulit pada citra dan penyakit jerawat.

b. Tahap Pengembangan Sistem

1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahapan yang dilakukan untuk memodelkan kebutuhan-kebutuhan yang diperlukan sistem, agar sistem dapat mengenali jerawat pada citra.

2. Perancangan

Tahap ini menjelaskan bagaimana perancangan sistem yang akan dibuat. Rancangan sistem berupa diagram alir (*flowchart*) dan rancangan antarmuka (*interface*).

3. Implementasi

Pada tahap ini, sistem yang telah dirancang kemudian diimplementasikan dengan menggunakan MATLAB R2014a.

4. Pengujian dan Evaluasi

Sistem yang telah diimplementasikan akan diuji untuk mengukur keberhasilan sistem dalam mengenali jerawat. Setelah didapat hasil pengujian sistem, kemudian dilakukan perbandingan hasil dengan sistem sejenis.

1.7. Sistematika Penulisan

Tugas Akhir ini disusun secara sistematis untuk mempermudah pembaca dalam memahami tulisan ini dengan membuatnya menjadi bagian-bagian yang saling berhubungan. Secara garis besar, sistematika penulisan Tugas Akhir ini sebagai berikut:

BAB I PENDAHULUAN

Bagian ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metode penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bagian ini berisi teori-teori yang digunakan sebagai dasar penelitian. Bab ini memuat referensi yang digunakan seperti penjelasan dari teori-teori seperti jerawat, model warna, segmentasi, *filtering* maupun metode yang digunakan untuk menguji sistem.

BAB III METODE PENELITIAN

Bagian ini berisi metode serta perancangan dari aplikasi pendeteksi jerawat yang akan dibuat. Adapun muatannya seperti pengumpulan data, analisis, diagram perancangan serta perancangan antarmuka sistem.

BAB IV HASIL DAN PEMBAHASAN

Bagian ini berisi pembahasan sistem yang telah diimplementasikan, pengujian mengenai hasil dari aplikasi pendeteksi jerawat yang telah dibuat dengan menggunakan metode *single decision threshold*, serta perbandingan hasil dengan sistem sejenis.

BAB V KESIMPULAN DAN SARAN

Bagian ini berisi kesimpulan yang menunjukkan apakah tujuan dari penelitian ini sudah tercapai dan saran agar penelitian dapat dikembangkan lagi oleh pihak yang ingin melakukan penelitian lebih lanjut.

BAB II

LANDASAN TEORI

2.1. Jerawat

Jerawat merupakan salah satu penyakit kulit yang mudah menyerang manusia. Dalam dunia medis, jerawat disebut dengan akne. Menurut Straus (1993) dalam buku *Dermatology in General Medicine*, akne vulgaris yaitu jenis akne yang paling sering dijumpai dan *miscellaneous types* yang merupakan akne yang mempunyai banyak klasifikasi (Djuanda dkk., 1999). Akne Vulgaris adalah penyakit kulit yang terjadi akibat peradangan menahun folikel pilosebacea yang ditandai adanya komedo, papul, nodus dan kista pada tempat predileksinya (wajah, bahu, lengan bagian atas dan punggung) (Djuanda dkk., 1999).



Gambar 2.1 Jerawat

Sumber: <http://www.internationalcopd.org/tag/treating-acne/>

Akne digolongkan sebagai Efloresensi, yaitu kelainan kulit yang dapat dilihat dengan mata telanjang (secara objektif), dan dapat diperiksa dengan perabaan (Murtiastutik dkk., 2007). Secara umum, terdapat beberapa tipe lesi untuk mengenali akne vulgaris seperti berikut (Murtiastutik dkk., 2007):

Papula : Bintik-bintik kecil berwarna merah berupa peninggian kulit yang solid dengan diameter < 1 cm.

Pastula : Memiliki warna dasar kemerahan berupa vesikula yang berisi pus yang berada di atas kulit yang meradang.

Nodula : Berbentuk kubah dan hampir sama dengan papula dengan diameter > 1 cm. Disebut juga Tumor yang merujuk pada nodula ganas yang memiliki diameter > 2 .

Kista : Disebut juga Krusta, yaitu pengeringan cairan tubuh bercampur epitel debris bakteri.

Parut : Parut merupakan bentuk peradangan terakhir dari akne, sehingga akan terbentuk parut.

Tipe-tipe lesi yang disebutkan diatas dirangkum menjadi ciri klinis jerawat. Ciri klinis jerawat tersebut diterjemahkan menjadi fitur komputasi untuk mengenali jerawat pada citra di sistem, seperti yang dapat dilihat pada Tabel 2.1.

Tabel 2.1 Ciri Klinis Jerawat

No	Ciri Klinis	Fitur Komputasi
1	Papula , diameter < 1 cm	Ekstraksi ciri: - <i>Area</i> , untuk mengukur luas setiap objek
2	Nodula , berbentuk kubah (Kebundaran)	Ekstraksi ciri: - <i>Eccentricity</i> , untuk mengukur nilai bundar setiap objek
3	Pastula , berwarna kemerahan	Ekstraksi ciri: - <i>Mean Intensity</i> , untuk menghitung rata-rata intensitas warna objek di setiap <i>layer</i> warna (RGB) - Standar Deviasi, untuk menghitung simpangan baku <i>mean intensity</i> setiap objek (jerawat)

2.2. Pengolahan Citra

Citra (*image*) didefinisikan sebagai fungsi dua dimensi $f(x, y)$, di mana x dan y adalah koordinat *spatial* dan nilai f pada tiap pasang koordinat (x, y) disebut intensitas (*intensity*) atau tingkat keabuan (*gray level*) dari gambar pada titik tersebut (Gonzalez & Woods, 2002). Citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit agar dapat diolah komputer, di mana citra yang dihasilkan inilah yang disebut citra digital (*digital image*) (Munir, 2004).

Pengolahan citra (*image processing*) adalah pemrosesan citra menggunakan komputer menjadi citra yang memiliki kualitas lebih baik agar mudah diinterpretasi oleh manusia atau

mesin (komputer), di mana masukkannya (*input*) adalah citra dan keluarannya (*output*) juga citra yang mempunyai kualitas lebih baik (Munir, 2004).

2.3. Model Warna

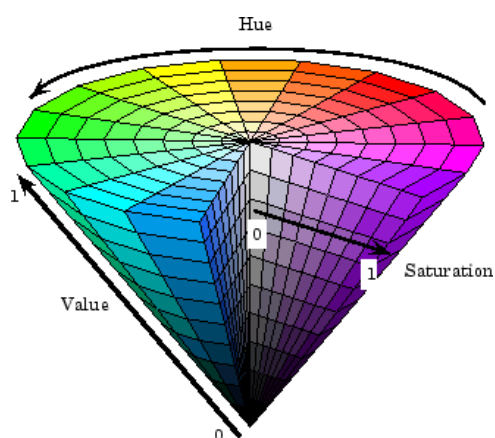
Model warna merupakan cara untuk menspesifikasi warna menjadi beberapa standar. Model warna digambarkan sebagai sistem koordinat pada subruang dengan model tertentu yang mengandung warna, di mana setiap warna diwakili oleh satu titik (Gonzalez & Woods, 2002). Adapun masing-masing warna digunakan sebagai standar untuk hardware tertentu, seperti RGB untuk monitor dan kamera atau HSV yang digunakan untuk keperluan pengolahan citra.

2.3.1. RGB

Warna-warna yang diterima mata manusia adalah hasil kombinasi cahaya dengan panjang gelombang yang berbeda. Penelitian menunjukkan bahwa rentang warna *red* (R), *green* (G) dan *blue* (B) atau biasa disingkat RGB merupakan rentang warna yang paling lebar, yang kemudian dinamakan warna pokok (*primaries*) (Munir, 2004). Menurut teori Young (1802), warna-warna lain dapat dihasilkan dengan mencampurkan warna-warna pokok dengan perbandingan tertentu (Munir, 2004).

2.3.2. HSV

Model warna HSV disebut juga HSI merupakan model warna yang terbentuk dari 3 komponen yaitu *hue*, *saturation* dan *value* (*intensity* dalam HSI).



Gambar 2.2 Sistem koordinat model warna HSV

Sumber: <https://www.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html>

Berdasarkan Gambar 2.2, sistem koordinat dari model warna HSV berbentuk kerucut terbalik. Terdapat 3 komponen yang membentuknya yaitu (Gonzalez & Woods, 2002):

Hue : Komponen yang menunjukkan jenis warna sebenarnya seperti merah, kuning atau hijau.

Saturation : Komponen untuk mengukur tingkat kemurnian dari warna sebenarnya.

Value : Disebut juga *intensity*, yaitu komponen untuk mengukur tingkat kecerahan warna.

Komponen-komponen dalam nilai HSV diperoleh dari perhitungan model warna RGB, oleh karena itu citra berwarna pada model warna HSV dipecah menjadi 3 seperti halnya pada model warna RGB (Ahmad, 2005). Untuk mengkonversi model warna dari RGB ke HSV, dapat dihitung dengan rumus sebagai berikut:

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

$$V = \max(r, g, b) \quad (2.2)$$

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

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

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

Untuk menghitung nilai tiap komponen HSV, dilakukan normalisasi dari setiap komponen RGB seperti Persamaan (2.1). Setelah itu dilakukan perhitungan untuk mencari

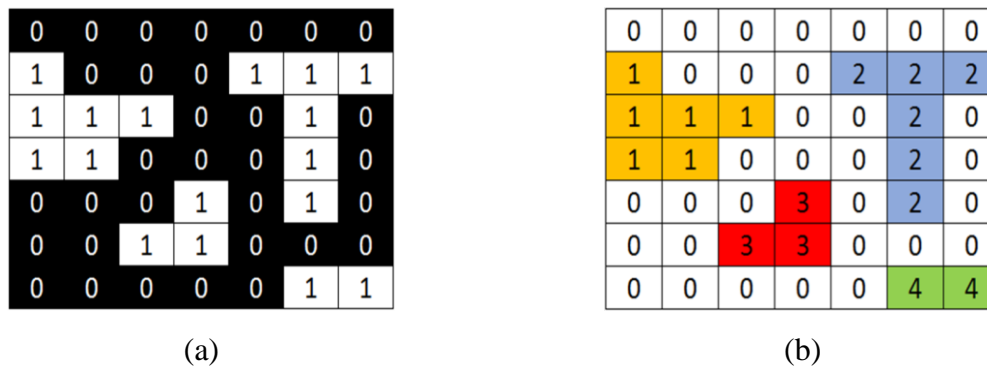
nilai *V* (*value*) dengan rentang hasil 0 sampai 1, *S* (*saturation*) dengan rentang hasil 0 sampai 1 dan *H* (*hue*) dengan rentang hasil 0 sampai 360.

2.4. Citra Biner

Dalam citra biner, setiap piksel hanya mempunyai 2 kemungkinan nilai, seperti *false* dan *true*, di mana di dalam matriks disimpan dengan nilai 0 (*false*) dan 1 (*true*). Citra biner dianggap sebagai citra intensitas bertipe khusus yang berisi hitam dan putih. Selain itu, citra biner dapat dinyatakan sebagai citra ber-indeks yang hanya memiliki 2 warna (Wijaya & Prijono, 2007).

2.5. Connected-Component Labeling

Connected-component labeling merupakan teknik operasi untuk mengklasifikasi *region* atau objek yang terdapat pada citra. Algoritma dari *connected-component labeling* adalah dengan memberikan label pada komponen piksel-piksel yang saling terhubung. Kumpulan piksel yang saling terhubung dianggap sebagai objek tunggal dan diberikan label yang berbeda pada setiap objeknya (Ahmad, 2005), seperti yang terlihat pada Gambar 2.3.



Gambar 2.3 Hasil *Labeling*; (a) Citra Biner. (b) Citra Berlabel

Sumber: <http://k-sience.blogspot.com/2017/06/deteksi-jumlah-objek-dengan-metode.html>

Pada dasarnya *connected-component labeling* memanfaatkan teori *connectivity* antar piksel pada citra. Piksel-piksel dalam sebuah *region* saling terhubung (*connected*) ditentukan jika mereka memiliki hubungan ketetanggaan dan tingkat keabuan yang sama (Gonzalez & Woods, 2002). Sebuah piksel dikatakan bertetangga jika mematuhi aturan *adjacency* atau memiliki kedekatan dengan piksel lain. Menurut Gonzalez dan Woods (2002), *adjacency* piksel terbagi menjadi 3 jenis:

1. **4-adjacency**. 2 piksel p dan q dengan keduanya bernilai V merupakan 4-adjacency jika q merupakan himpunan dari $N_4(p)$.
2. **8-adjacency**. 2 piksel p dan q dengan keduanya bernilai V merupakan 8-adjacency jika q merupakan himpunan dari $N_8(p)$.
3. **m-adjacency (mixed adjacency)**, piksel p dan q dengan keduanya bernilai V merupakan 8-adjacency jika :
 - a. q merupakan himpunan dari $N_4(p)$, atau
 - b. q merupakan himpunan dari $N_D(p)$ dan himpunan $N_4(p) \cap N_4(q)$ tidak memiliki piksel bernilai V .

Mixed adjacency merupakan modifikasi dari 8-adjacency. Pada dasarnya m-adjacency digunakan untuk menghilangkan ambiguitas yang sering terjadi ketika menggunakan 8-adjacency.

2.6. Segmentasi Kulit

Segmentasi merupakan prosedur untuk mempartisi gambar menjadi bagian-bagian tertentu (Gonzalez & Woods, 2002). Segmentasi merupakan salah satu proses yang rumit dalam bidang pengolahan citra digital. Contoh kasus segmentasi adalah memisahkan objek wajah dengan *background* pada citra digital.

Penelitian ini yaitu mengenali jerawat pada wajah, dimana objek yang difokuskan adalah wajah. Peran segmentasi pada penelitian ini adalah untuk memisahkan wajah dengan *background*. Tujuan dari segmentasi wajah agar objek selain jerawat pada wajah tidak terdeteksi. Contoh hasil dari segmentasi wajah dapat dilihat pada Gambar 2.4.



Gambar 2.4 Segmentasi Wajah

Sumber: <https://web.fe.up.pt/~tavares/projectos/DEFOBJ/defobj%20results.html>

Untuk mensegmentasi wajah, dapat dilakukan dengan pendekatan pemodelan warna kulit. Menurut penelitian yang dilakukan oleh Mujahidin (2012), pemodelan warna kulit dapat dilakukan dengan menetapkan nilai ambang (*threshold*) pada format warna HSV. Nilai ambang HSV untuk warna kulit adalah sebagai berikut:

$$0.0 \leq H \leq 0.25$$

$$0.15 \leq S \leq 0.9$$

$$0.2 \leq V \leq 0.95$$

Nilai ambang tersebut kemudian dijadikan kondisi untuk mengubah citra berwarna (RGB) menjadi citra biner. Citra biner yang dimaksud adalah dengan mengubah warna objek yang tersegmentasi menjadi berwarna putih (1), sedangkan *background* menjadi berwarna hitam (0). Pengubahan menjadi citra biner bertujuan mempermudah proses penghapusan *noise* dengan memanfaatkan metode *connected-component*, sehingga membuat objek yang tersegmentasi menjadi lebih rapi (Mujahidin, 2012).

2.7. Laplacian of Gaussian

Laplacian merupakan operator turunan kedua yang berfungsi untuk mendeteksi tepi yang dilacak dengan menemukan titik potong dengan sumbu x. Akan tetapi, Laplacian sangat sensitif terhadap *noise* (derau). Oleh karena itu, diperlukan *filter* untuk melemahkan *noise* dengan menggunakan Gaussian. Jadi Laplacian of Gaussian merupakan kombinasi dari *filter* Gaussian dan deteksi tepi Laplacian. Karakteristik dari deteksi tepi Laplacian of Gaussian adalah sebagai berikut (Ahmad, 2005):

1. Kovolusi citra dengan *filter* Gaussian.
2. Penguatan tepi (*sharpening*) dengan deteksi tepi Laplacian.
3. Kriteria pelacakan dengan menemukan titik potong dengan sumbu x dalam fungsi turunan kedua (Laplacian) yang bersesuaian dengan puncak fungsi turunan pertama.
4. Lokasi tepi dapat ditentukan dengan resolusi subpiksel menggunakan interpolasi linear.

Untuk mendapat hasil dari operator Laplacian of Gaussian, $h(x, y)$, maka dapat dirumuskan dengan operasi konvolusi berikut (Ahmad, 2005).

$$h(x, y) = \nabla^2 [g(x, y) \times f(x, y)] \quad (2.5)$$

Selanjutnya menggunakan aturan turunan untuk konvolusi, didapat:

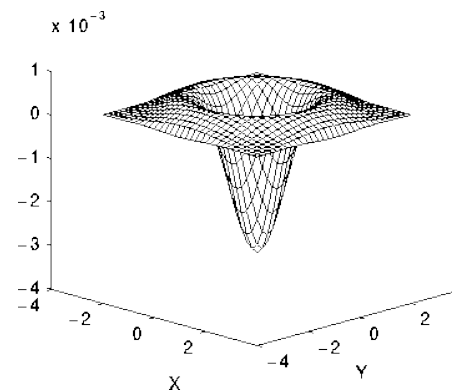
$$h(x, y) = [\nabla^2 g(x, y)] \times f(x, y) \quad (2.6)$$

$$\nabla^2 g(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right) e^{-\frac{(x^2 + y^2)}{2\sigma^2}} \quad (2.7)$$

Persamaan (2.7) merupakan rumus dari Laplacian of Gaussian. Laplacian of Gaussian disebut juga operator tepi Meksiko, karena koordinat x dan y membentuk lekukan berbentuk tepi Meksiko, seperti yang terlihat pada Gambar 2.5.

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

(a)



(b)

Gambar 2.5 Bentuk LoG; (a) Kernel. (b) Plot 3D

Sumber: <https://www.quora.com/What-happens-to-the-image-when-Laplacian-of-Gaussian-is-applied>



(c)



(d)

Gambar 2.6 Hasil Deteksi Citra; (a) *Grayscale*. (b) LoG

Sumber: <https://www.quora.com/What-happens-to-the-image-when-Laplacian-of-Gaussian-is-applied>

Gambar 2.6 diatas merupakan Hasil deteksi tepi pada citra menggunakan Laplacian of Gaussian, di mana *input* berupa citra *grayscale* yang kemudian menghasilkan *output* citra hitam putih dengan tepi yang telah terdeteksi.

2.8. Ekstraksi Ciri

Ekstraksi ciri merupakan tahapan yang dilakukan untuk memperoleh ciri atau informasi tertentu dari suatu objek. Informasi yang didapat dari suatu objek dapat digunakan untuk mengenali objek serupa maupun untuk membedakan dengan objek lainnya. Contoh penggunaan ekstraksi ciri adalah untuk mengukur luas suatu objek.

2.8.1. Area dan *Perimeter*

Area merupakan jumlah piksel-piksel yang tersusun pada suatu wilayah atau objek. Area memiliki satuan piksel karena piksel-piksel yang tersusun membentuk suatu luasan. Area dapat mencerminkan ukuran atau berat objek sesungguhnya pada beberapa benda pejal dengan bentuk yang seragam, tetapi tidak demikian dengan benda yang berongga (Ahmad, 2005).

Perimeter (keliling) merupakan bagian terluar dari suatu objek yang bersinggungan dengan piksel-piksel diluar objek tersebut. Nilai *perimeter* didapat dengan menghitung jumlah piksel pada batas objek (Ahmad, 2005).

2.8.2. *Eccentricity*

Eccentricity merupakan nilai yang menentukan kebundaran (elips) suatu objek. *Eccentricity* memiliki rentang 0 sampai 1. Objek dengan bentuk yang mendekati garis lurus memiliki nilai mendekati 1, sedangkan objek dengan bentuk mendekati lingkaran memiliki nilai mendekati 0. Nilai *eccentricity* dapat dirumuskan seperti Persamaan **Error! Reference source not found.**

$$e = \sqrt{1 - \frac{b^2}{a^2}} \quad (2.8)$$

Keterangan :

e = Nilai *eccentricity*

a = Mayor *axis*

b = Minor *axis*

2.9. Standar Deviasi

Standar deviasi atau simpangan baku merupakan ukuran tingkat penyimpangan data yang banyak digunakan dalam analisis data. Ukuran yang dimaksud merupakan ukuran penyimpangan terhadap nilai mean (rata-rata), di mana nilai dari simpangan baku merupakan harga akar positif dari selisih *item* data dengan rata-rata nilai sampel (Supangat, 2007). Formulasi simpangan baku untuk data yang belum dikelompokkan adalah sebagai berikut :

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}; \text{Untuk data} > 100 \quad (2.9)$$

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}; \text{Untuk data} \leq 100 \quad (2.10)$$

Sedangkan untuk data yang sudah dikelompokkan ke dalam daftar distribusi frekuensi diformulasikan sebagai berikut :

$$s = \sqrt{\frac{\sum_{i=1}^k f_i (x_i - \bar{x})^2}{n}}; \text{Untuk data} > 100 \quad (2.11)$$

$$s = \sqrt{\frac{\sum_{i=1}^k f_i (x_i - \bar{x})^2}{n - 1}}; \text{Untuk data} \leq 100 \quad (2.12)$$

Semakin besar nilai simpangan baku, maka tingkat keragaman data akan semakin menjauhi nilai rata-ratanya. Sedangkan semakin kecil nilai simpangan baku, maka tingkat keragaman data akan semakin mendekati nilai rata-ratanya. Jika didapat nilai simpangan baku = 0, dapat diartikan bahwa data memiliki nilai yang sama dengan rata-ratanya, sehingga tidak didapati penyimpangan terhadap nilai rata-ratanya (Supangat, 2007).

2.10. Teori Pengujian Sistem

Metode yang digunakan untuk menguji sistem adalah metode *Single Decision Threshold*, yaitu tabel yang terdiri atas banyaknya baris data uji yang diprediksi benar dan tidak

benar dengan citra yang telah ditandai. Metode ini digunakan untuk mengukur tingkat keberhasilan sistem dalam mengenali jerawat pada citra. Pada dasarnya metode ini mengandung informasi yang membandingkan hasil deteksi jerawat oleh sistem (nilai prediksi) dengan hasil penandaan jerawat oleh pakar (nilai sebenarnya). Dalam penelitian ini, pakar yang dimaksud adalah AVO.

Ada empat istilah yang digunakan dalam *Single Decision Threshold*, diantaranya (Shortliffe & Cimino, 2010):

1. **True Positive** (TP) adalah apabila nilai sebenarnya dan nilai prediksi menghasilkan hasil yang positif.
2. **True Negative** (TN) adalah apabila nilai sebenarnya dan nilai prediksi menghasilkan hasil yang negatif.
3. **False Positive** (FP) adalah apabila nilai sebenarnya bernilai negatif, tetapi prediksi menghasilkan hasil yang positif.
4. **False Negative** (FN) adalah apabila nilai sebenarnya bernilai positif, tetapi prediksi menghasilkan nilai yang negatif.

Tabel 2.2 Tabel *Contingency*

		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>

Jumlah dari nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN) pada Tabel 2.2, akan digunakan untuk mengukur nilai *sensitivity* dan *specificity*.

Sensitivity disebut juga **True-Positive Rate** (TPR), merupakan parameter untuk mengukur presentase data positif yang teridentifikasi dengan benar, yaitu jerawat yang terdeteksi sistem sama dengan jerawat yang ditandai pakar. Untuk menghitung nilai *sensitivity*, diformulasikan seperti Persamaan (2.13) (Shortliffe & Cimino, 2010):

$$Sensitivity = \frac{TP}{TP + FN} \quad (2.13)$$

Specificity disebut juga **True-Negative Rate** (TNR), merupakan parameter untuk mengukur presentase data negatif yang teridentifikasi dengan benar, yaitu sistem tidak

mendeteksi objek bukan jerawat. Untuk menghitung nilai *specificity*, diformulasikan seperti Persamaan (2.14) (Shortliffe & Cimino, 2010):

$$Specificity = \frac{TN}{TN + FP} \quad (2.14)$$

Skala dari nilai *sensitivity* dan *specificity* adalah 0-1. Semakin besar nilainya maka dapat diartikan bahwa sistem dapat mengenali jerawat dengan baik, sedangkan jika semakin kecil nilainya berarti sistem belum optimal dalam menganali jerawat.

BAB III

METODE PENELITIAN

3.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini terdiri dari 36 foto dengan karakteristik wajah *close-up* tunggal menghadap depan atau samping dengan latar belakang (*background*) yang berbeda dengan warna kulit wajah seperti pada Gambar 3.1. Foto-foto tersebut diperoleh dari internet yang telah divalidasi oleh AVO. Dua foto digunakan sebagai data *training* dengan tujuan untuk mengidentifikasi ciri-ciri atau karekteristik dari jerawat pada citra. Hasil dari proses identifikasi ini akan digunakan sebagai data acuan untuk menguji data sampel lain.

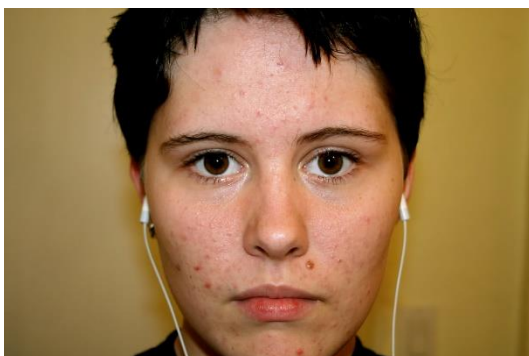


(a)



(b)

Gambar 3.1 Data Citra; (a) Hadap Depan. (b) Hadap Samping



(a)



(b)

Gambar 3.2 Data *Training*; (a) I. (b) II

Kedua data yang diberikan oleh AVO pada Gambar 3.2 tersebut digunakan sebagai data *training* untuk mengidentifikasi jerawat, dengan pertimbangan jerawat pada citra terlihat jelas. Data *training* I memiliki resolusi 1599 x 1066 piksel, sedangkan data *training* II memiliki resolusi 500 x 333 piksel sehingga memenuhi kebutuhan *input* sistem.

3.2. Analisis Kebutuhan Sistem

Tahap analisis adalah langkah dalam mengamati kebutuhan dan keperluan yang berkaitan dengan penelitian. Tahapan analisis yang digunakan dalam penelitian ini antara lain analisis kebutuhan *input*, analisis kebutuhan proses, analisis kebutuhan *output* dan analisis kebutuhan antarmuka.

3.2.1. Analisis Kebutuhan *Input*

Hasil yang didapat dari diskusi dengan AVO adalah *input* yang diperlukan dalam sistem ini berupa satu buah *file* foto wajah *close-up* berwarna (RGB) yang berjerawat dengan wajah tampak depan atau samping, contohnya seperti Gambar 3.3. Rincian kebutuhan *input* sistem adalah sebagai berikut:

1. Format Citra

Citra menggunakan format .jpg dan .png yang merupakan format paling banyak dijumpai.

2. Ukuran Citra

Ukuran panjang (kolom) atau lebar (baris) citra tidak kurang dari 400 *pixel* agar definisi citra wajah dan jerawat terlihat jelas (tidak *blur*).



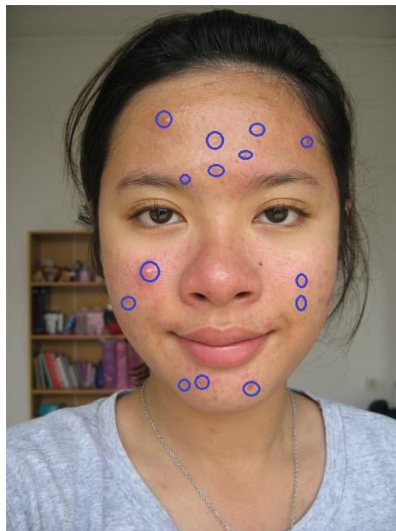
Gambar 3.3 Kebutuhan *Input*

3.2.2. Analisis Kebutuhan Proses

Analisis kebutuhan proses adalah suatu pengamatan yang mengacu pada kebutuhan dalam proses pembentukan sistem yang akan dibangun. Dalam penelitian ini proses yang dibutuhkan adalah sistem dapat mendeteksi jerawat pada citra *input*.

3.2.3. Analisis Kebutuhan Output

Analisis kebutuhan *output* merupakan tahapan analisis dalam menentukan keperluan untuk *output* dari suatu sistem atau penelitian. Analisis kebutuhan *output* juga didapat dari diskusi dengan AVO. *Output* yang dibutuhkan dari sistem pendeteksi jerawat berupa citra wajah dengan jerawat yang sudah ditandai oleh sistem seperti contohnya pada Gambar 3.4.



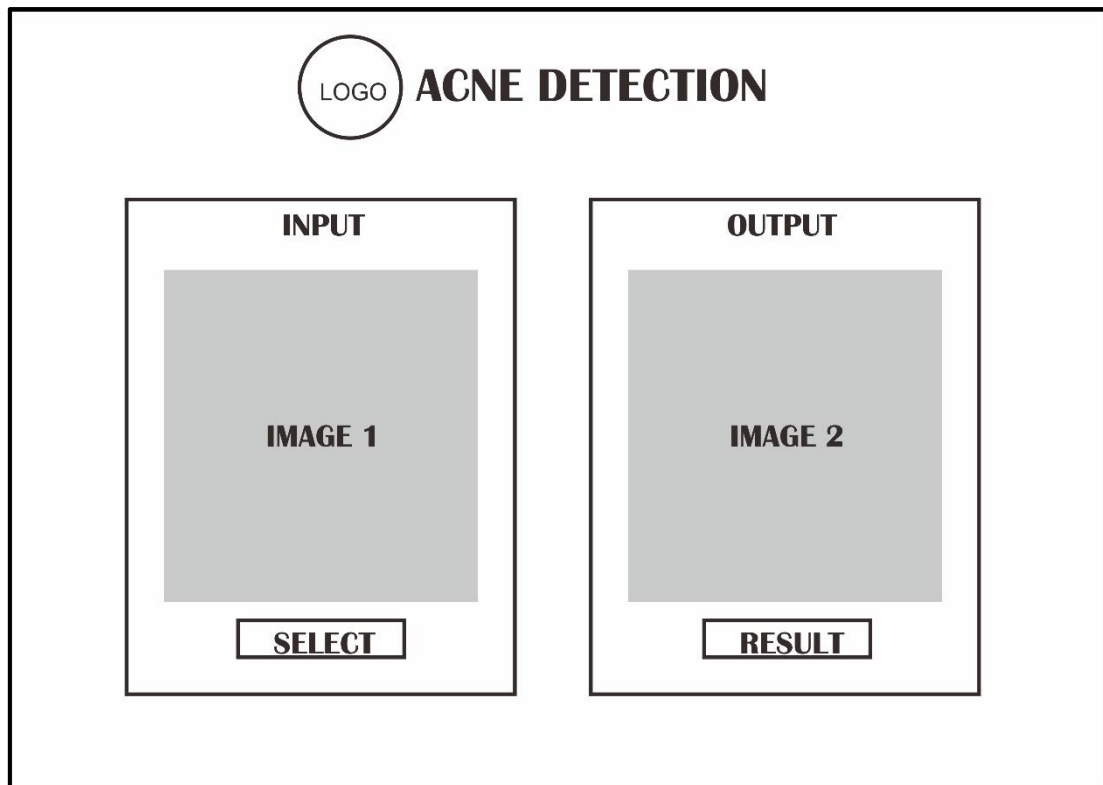
Gambar 3.4 Kebutuhan *Output*

3.2.4. Analisis Kebutuhan Antarmuka

Analisis kebutuhan antarmuka adalah tahapan analisis dalam menentukan keperluan untuk membangun antarmuka. Kebutuhan antarmuka pada penelitian ini adalah dua tombol yaitu tombol untuk memilih dan memasukkan citra dan tombol untuk memproses hasil deteksi citra yang telah dimasukkan. Citra yang perlu ditampilkan pada antarmuka adalah citra masukkan dan citra hasil. AVO juga ingin terdapat logo mereka pada antarmuka.

3.3. Rancangan Antarmuka

Antarmuka merupakan jembatan antara *user* (pengguna) dengan sistem untuk memudahkan pengguna berinteraksi dengan sistem. Rancangan antarmuka pada penelitian ini memiliki dua halaman yaitu halaman utama dan *file input*.



Gambar 3.5 Rancangan Antarmuka

Terdapat dua komponen utama dari rancangan antarmuka halaman utama pada Gambar 3.5 yaitu panel INPUT, PROCESS dan OUTPUT. Rincian dari antarmuka tersebut dapat dilihat pada poin-poin berikut:

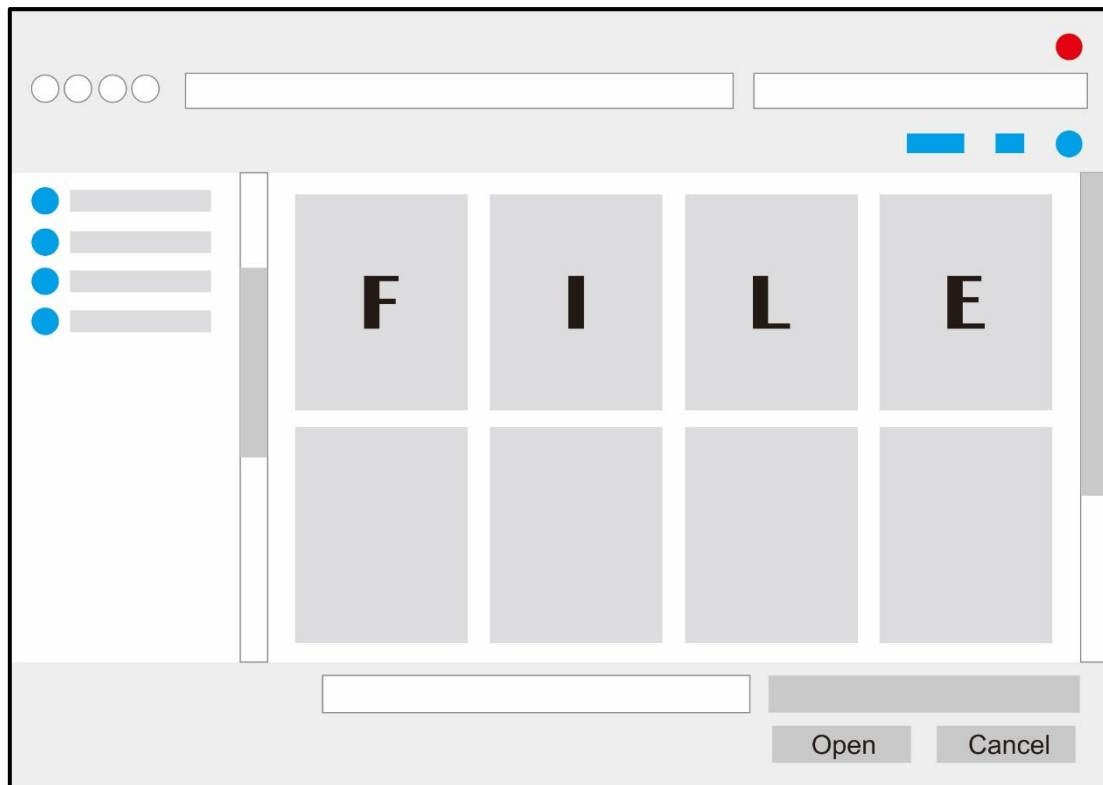
1. INPUT

Pada panel INPUT terdapat panel IMAGE 1 untuk menampilkan gambar setelah *user* memasukkan gambar yang dipilih dengan menekan tombol SELECT yang berfungsi menampilkan *file input*.

2. OUTPUT

Pada panel OUTPUT memiliki struktur yang sama dengan panel INPUT. Yang membedakan adalah fungsinya, yaitu tombol RESULT untuk memproses masukkan yang kemudian akan ditampilkan pada panel IMAGE 2.

Saat tombol SELECT ditekan, maka akan muncul Window File Input untuk memilih data citra yang akan digunakan untuk masukkan aplikasi. Adapun rancangannya dapat dilihat pada Gambar 3.6.



Gambar 3.6 Rancangan *File Input*

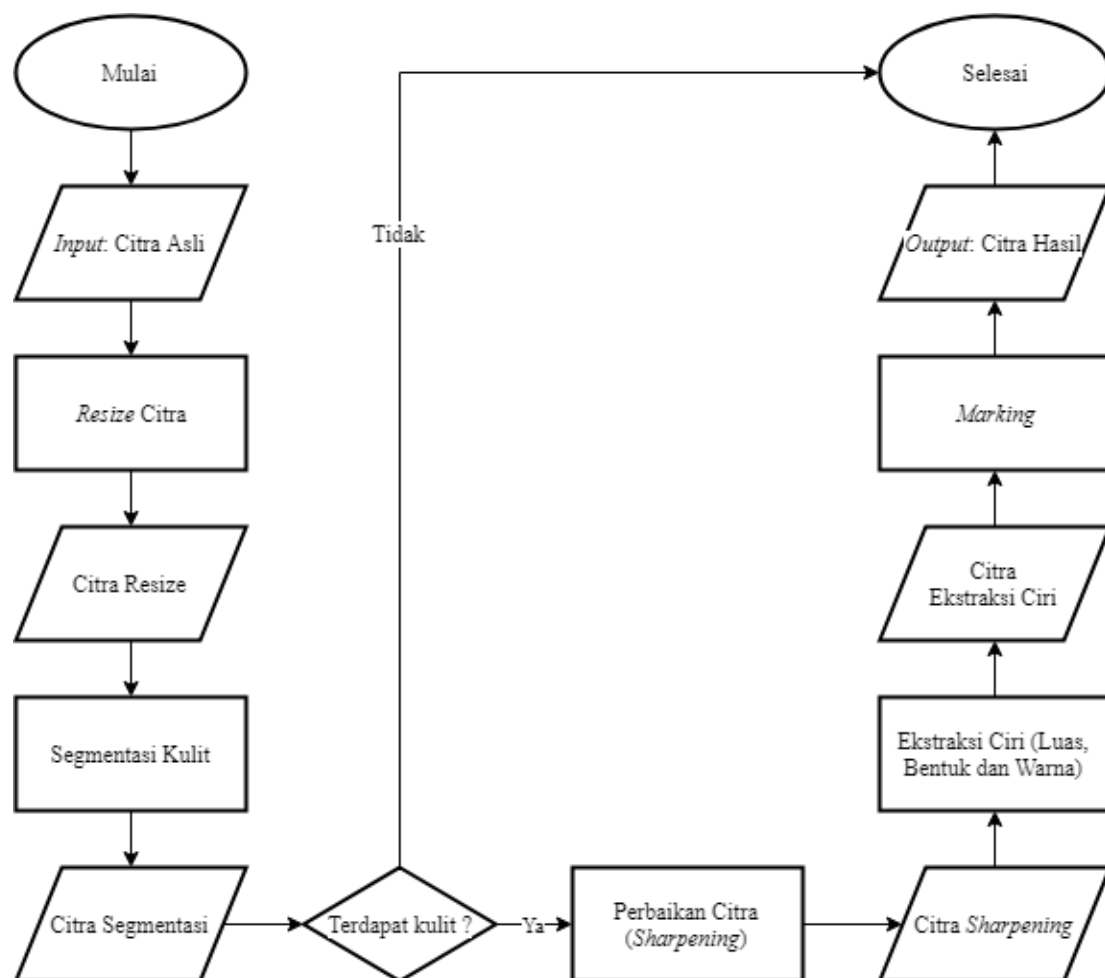
3.4. Analisis Pengujian Sistem

Pengujian ini dilakukan untuk mengetahui hasil klasifikasi sistem sudah sesuai dengan hasil yang diperoleh dari ahli. Pengujian ini menggunakan *Single Decision Threshold*. Setiap data akan diujikan dengan mengukur nilai *sensitivity* dan *specificity* untuk mengetahui seberapa baik sistem dalam mendeteksi jerawat.

BAB IV ARSITEKTUR SISTEM

4.1. Gambaran Umum

Gambaran umum merupakan peta sederhana dari rancangan sistem secara keseluruhan. Untuk mempermudah membaca proses-proses yang terjadi pada sistem, maka dibuat *flowchart* untuk menggambarkan alur proses mulai dari *input* hingga *output*, seperti yang terlihat pada Gambar 4.1.



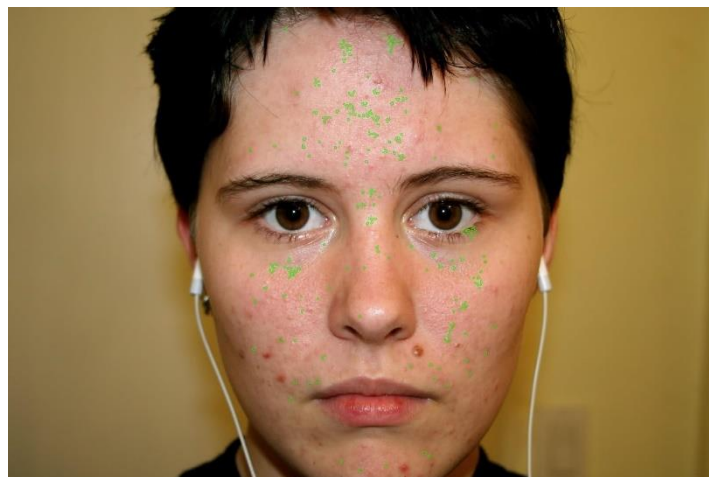
Gambar 4.1 *Flowchart* Utama

Sistem dimulai dari *input* sebuah citra asli berupa citra wajah di mana akan melalui 5 proses dan akan menghasilkan *output* akhir sebuah citra hasil berupa citra wajah dengan jerawat yang sudah ditandai. Adapun 5 proses tersebut diantaranya:

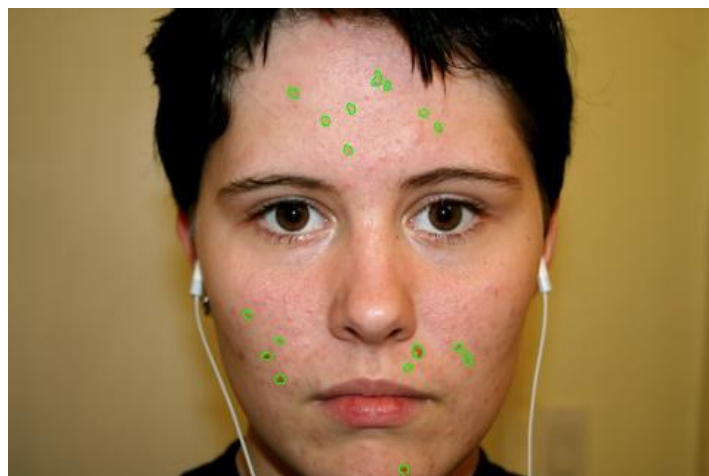
4.3. Resize Citra

Resize citra merupakan proses untuk mengubah ukuran dimensi citra menjadi lebih kecil jika dimensi citra terlalu besar. Proses ini bertujuan untuk mengurangi iterasi yang terlalu banyak dikarenakan dimensi citra yang besar. Besarnya dimensi citra juga berpengaruh terhadap proses perbaikan citra, oleh karena itu proses *resize* citra dapat menentukan standar ukuran citra.

Walaupun menjadi proses pertama yang dieksekusi dalam sistem, *resize* citra merupakan proses terakhir yang ditemukan. Hal itu dikarenakan data *training* I memiliki hasil deteksi yang lebih baik jika dilakukan perubahan ukuran lebih kecil dibanding ukuran yang sebenarnya. Ukuran dimensi citra berpengaruh terhadap keseluruhan proses yang terjadi pada sistem. Dua gambar yang sama dengan ukuran dimensi yang berbeda, akan memiliki hasil deteksi yang berbeda seperti pada Gambar 4.2.



(a)



(b)

Gambar 4.2 Hasil Deteksi; (a) Citra Awal. (b) Citra *Resize*

Hasil yang berbeda terjadi karena terdapat perubahan informasi-informasi pada citra akibat ukuran dimensi yang berbeda. Oleh karena itu, terdapat proses *resize* untuk mengubah ukuran dimensi citra jika terlalu besar. Sehingga rentang informasi untuk mengenali jerawat tidak berubah. *Input* membutuhkan citra dengan ukuran panjang atau lebar tidak kurang dari 400 piksel, sedangkan data *training* II memiliki ukuran dimensi terbesar (kolom) 500 piksel. Dari rentang 400 hingga 500 piksel, didapat ukuran **480 piksel** memiliki hasil uji terbaik dengan nilai *sensitivity* dan *specificity* tertinggi seperti pada Tabel 4.1.

Tabel 4.1 Uji Parameter *Resize*

Nama	Resize (Piksel)	True Positive	True Negative	False Positive	False Negative	Sensitivity	Specificity
Data <i>Training</i> I	470	9	183	8	6	0,600	0,958
	480	13	200	3	2	0,867	0,985
	490	10	214	5	5	0,667	0,977
Data <i>Training</i> II	470	7	105	1	3	0,700	0,991
	480	9	112	0	1	0,900	1,000
	490	8	119	0	2	0,800	1,000

Dalam sistem yang melakukan banyak iterasi, ukuran citra juga berpengaruh terhadap kecepatan sistem. Data *training* I memiliki resolusi 1599 x 1066, sehingga didapat sebanyak 1.704.534 perulangan saat proses penelusuran per piksel. Oleh karena itu, diterapkan proses *resize* citra untuk mengubah ukuran citra menjadi lebih kecil sehingga dapat mempercepat kinerja sistem. Perbandingan kecepatan sistem antara citra sebelum dan setelah *resize* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Perbandingan Kecepatan Sistem

Nama	Resolusi (BxK)	Proses (Detik)					Total (Detik)
		Resize	Segmentasi Kulit	Perbaikan Citra	Ekstraksi Ciri	Marking	
Data <i>Training</i> I	1599x1066		3,7863	0,1416	1,3973	0,7751	6,1003
	(Resize) 480x320	0,0433	0,4071	0,0322	0,6099	0,2501	1,3426
Data <i>Training</i> II	500x333		0,4011	0,0382	0,6103	0,2592	1,3088
	(Resize) 480x320	0,0276	0,3126	0,0360	0,6100	0,3133	1,2995

4.4. Segmentasi Kulit

Segmentasi kulit merupakan proses untuk memisahkan objek kulit (wajah) pada citra dengan *background* dan objek selain kulit. Proses ini bertujuan agar proses pencarian jerawat hanya berfokus pada daerah wajah saja, sehingga objek yang menyerupai jerawat pada *background* tidak terdeteksi oleh sistem.

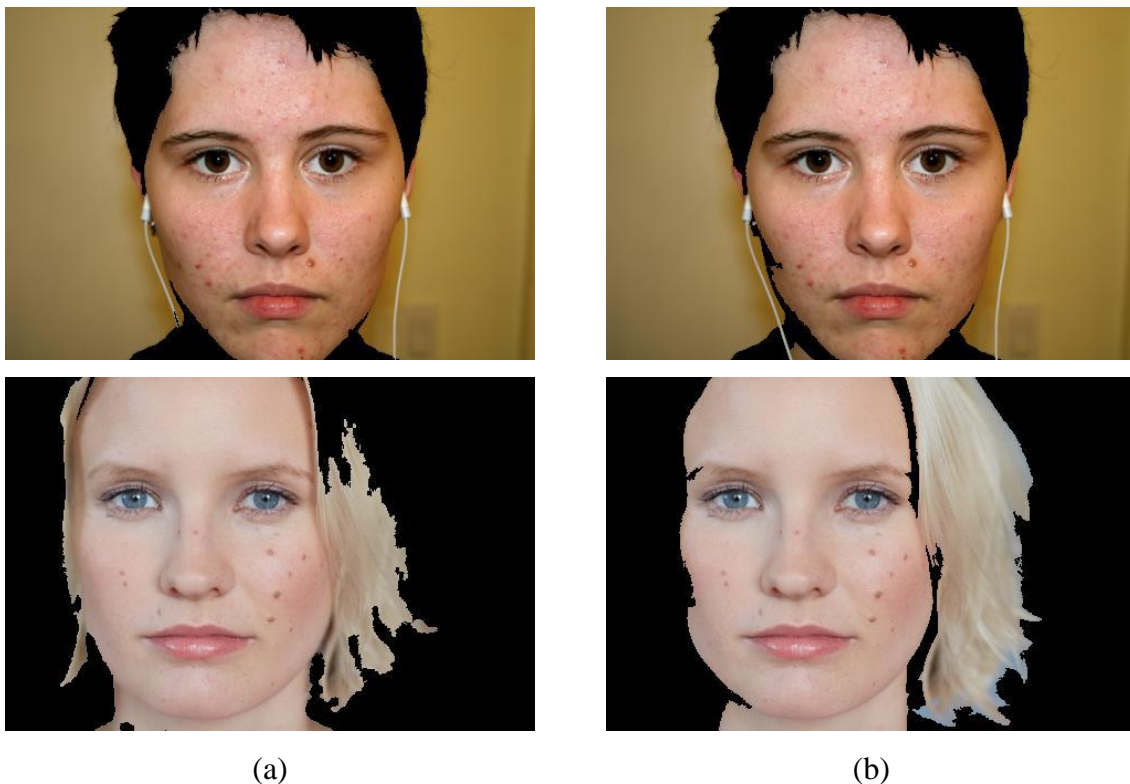
Dari hasil analisis, terdapat dua metode yang bisa digunakan untuk segmentasi kulit. Pertama, menurut Mujahidin (2012) proses segmentasi kulit dapat dilakukan dengan metode *thresholding* HSV. Untuk menghilangkan objek yang tidak terhubung dengan wajah, memanfaatkan *connected-component* untuk menentukan objek terluas sebagai wajah. Rentang HSV yang ditentukan Mujahidin (2012) adalah sebagai berikut:

$$0.0 \leq H \leq 0.25$$

$$0.15 \leq S \leq 0.9$$

$$0.2 \leq V \leq 0.95$$

Metode kedua adalah binerisasi dengan Otsu *threshold* yang umum digunakan pada MATLAB, dengan dasar bahwa syarat *input* berupa foto *close-up*. Sehingga wajah pada citra memiliki *area* yang luas, membuat pencarian wajah juga dapat menggunakan acuan objek terluas. Perbandingan antara citra segmentasi HSV dan Otsu dapat dilihat pada Gambar 4.3.



Gambar 4.3 Perbandingan Citra Hasil Segmentasi; (a) Metode HSV. (b) Metode Otsu

Secara kasat mata hasil segmentasi menggunakan HSV dan Otsu tidak memiliki perbedaan yang signifikan. Pada data *training* I hasil kedua segmentasi hampir sama, dengan hanya menghilangkan sedikit bagian bukan wajah pada citra dikarenakan warna *background* memiliki kesamaan dengan warna kulit. Sedangkan pada data *training* II hasil kedua segmentasi sudah baik, walaupun masih menyisakan rambut pada citra.

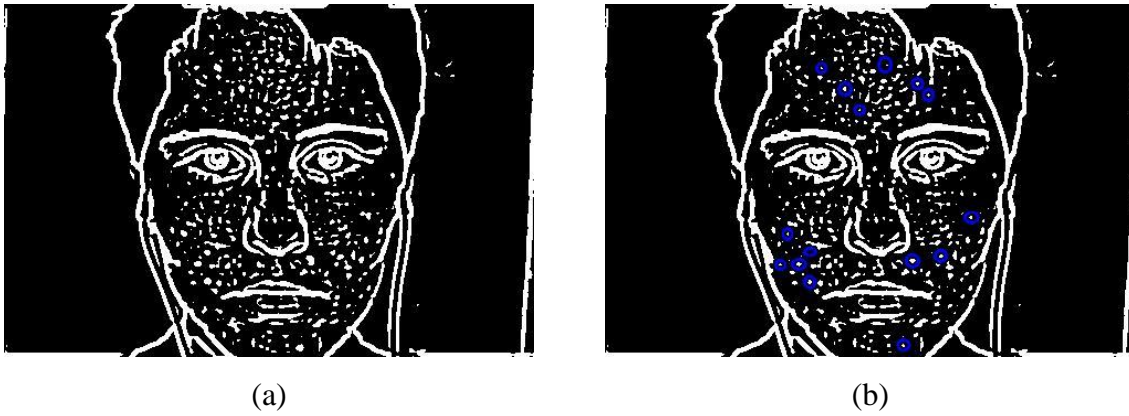
Walaupun secara visual terlihat mirip, tapi tidak dengan kecepatan. Pada data *training* I segmentasi menggunakan HSV dan Otsu masing-masing 0,848 dan 0,483 detik, sedangkan data *training* II masing-masing 0,573 dan 0,375 detik. Dari hasil uji kecepatan tersebut didapat metode Otsu memiliki kinerja lebih cepat. Dengan pertimbangan kinerja kecepatan, maka pada penelitian ini menggunakan **Otsu threshold** sebagai metode untuk segmentasi kulit.

Hasil segmentasi Otsu memungkinkan objek yang tersegmentasi memiliki lubang, sehingga diperlukan penambalan lubang dengan metode *fill image regions*.

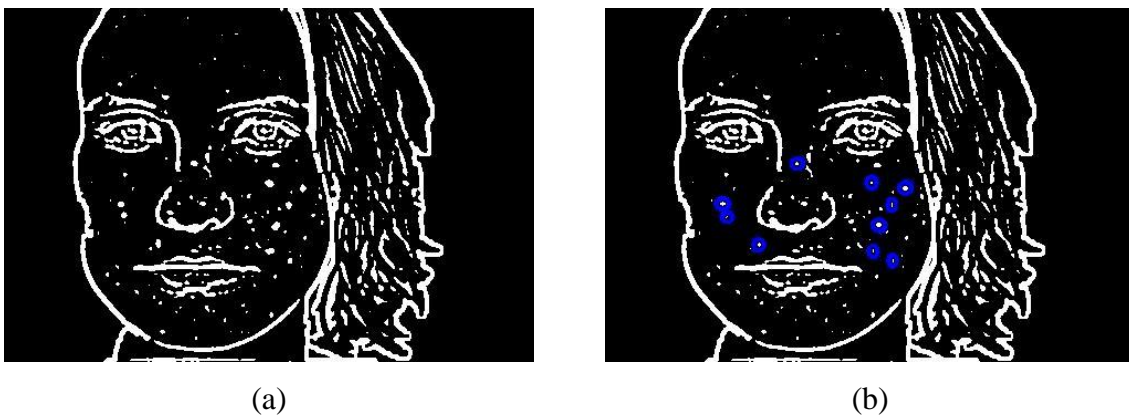
4.5. Perbaikan Citra

Perbaikan citra merupakan proses eksperimental untuk menonjolkan ciri tertentu dari sebuah citra. Proses perbaikan citra dalam kasus ini adalah *sharpening*. *Sharpening* merupakan metode untuk mempertajam kualitas citra, sehingga membuat objek jerawat lebih menonjol dan dapat dikenali pada daerah kulit.

Objek yang ditajamkan pada penelitian ini adalah jerawat. Jerawat pada citra berbentuk *blob* (gumpalan). Menurut Kong (2013), *blob* pada citra dapat dideteksi dengan menggunakan *filter* Laplacian of Gaussian (Log). *Filter* Laplacian of Gaussian memiliki dua parameter yaitu ukuran matriks dan ukuran sigma. Setelah dilakukan beberapa kali percobaan dari skala 5 (ukuran standar) hingga 10, didapat ukuran **matriks** adalah **9 x 9**. Ukuran tersebut dirasa sudah baik karena jika terlalu kecil maka ukuran *blob* yang terdeteksi semakin kecil. Sedangkan ukuran **sigma** adalah **2** karena semakin kecil ukuran maka *blob* akan saling bersentuhan dan jika terlalu besar maka *blob* akan terkikis. Adapun hasil perbaikan citra dengan LoG dapat dilihat pada Gambar 4.4 dan Gambar 4.5.



Gambar 4.4 Perbaikan Citra Data *Training I*; (a) Citra LoG. (b) Citra Penandaan Jerawat



Gambar 4.5 Perbaikan Citra Data *Training II*; (a) Citra LoG. (b) Citra Penandaan Jerawat

Hasil perbaikan citra dengan LoG pada data *training I* terdapat 393 objek yang terdeteksi, di mana 14 dari 15 jerawat yang ditandai AVO berhasil terdeteksi. Sedangkan pada data *training II* terdapat 267 objek, dimana 10 dari 10 jerawat berhasil terdeteksi. Sehingga hasil deteksi tersebut mendukung perbaikan citra pada penelitian ini menggunakan **Laplacian of Gaussian**.

4.6. Ekstraksi Ciri

Ekstraksi ciri merupakan proses untuk mendapatkan informasi ciri dari setiap objek yang terdapat pada citra (biner). Setelah itu, dilakukan analisis informasi yang didapat untuk menentukan parameter *threshold* (batas ambang) yang digunakan untuk mengenali objek jerawat. Proses ini bertujuan untuk memisahkan antara objek jerawat dengan objek selain jerawat. Berdasarkan analisis ciri jerawat yang didapat, ekstraksi ciri menggunakan tiga ciri fitur yaitu luas, bentuk dan warna.

Hasil perbaikan citra dengan LoG berhasil mendeteksi jerawat. Akan tetapi, lebih dari 90% objek pada citra bukanlah jerawat. Pada data *training I*, dari 393 objek yang terdeteksi hanya 14 objek yang merupakan jerawat. Sehingga dilakukan ekstraksi ciri untuk mendapatkan informasi dalam mengenali jerawat, sehingga objek-objek selain jerawat dapat tereliminasi. Berdasarkan ciri klinis jerawat pada Tabel 2.1, terdapat tiga informasi diantaranya:

- a. Luas dan Bentuk
 - *Area*, untuk mengukur luas objek
 - *Eccentricity*, untuk mengukur nilai bundar objek
- b. Warna
 - *Mean Intensity*, untuk menghitung rata-rata intensitas warna objek di setiap *layer* warna RGB

Informasi jerawat didapat dengan menghilangkan garis tepi wajah, hidung dan mata yang terdeteksi oleh *filter* Laplacian of Gaussian dengan cara mengeliminasi objek dengan ukuran lebih dari 100 piksel, dikarenakan objek jerawat yang terdeteksi LoG memiliki ukuran kurang dari 100 piksel. Hasil eliminasi tersebut mengurangi jumlah objek yang terdeteksi pada data *training I* menjadi 213 objek dan data *training II* menjadi 121 objek. Jumlah tersebut dirasa masih banyak, sehingga perlu direduksi menggunakan ekstraksi ciri dengan komputasi ciri fitur *area*, *eccentricity* dan *mean intensity* objek. Setelah itu dilakukan pengumpulan informasi jerawat dengan mencari indeks objek yang ditandai sebagai jerawat. Informasi jerawat yang didapat dari kedua data *training* dapat dilihat pada Tabel 4.3 dan Tabel 4.4.

Tabel 4.3 Ciri Jerawat Data *Training I*

No Jerawat	Area	Eccentricity	Mean Intensity		
			Red	Green	Blue
1	28	0,707	179	112	90
2	30	0,702	164	86	66
3	24	0,575	191	123	102
4	35	0,416	209	86	63
5	37	0,743	135	144	124
6	31	0,628	226	157	150
7	31	0,582	213	152	145
8	45	0,751	179	84	146
9	36	0,536	207	139	60
10	52	0,747	191	107	69
11	26	0,734	223	157	132
12	27	0,719	219	147	116

13	25	0,741	196	123	96
14	40	0,868	214	133	109
Min	24	0,416	135	84	60
Max	52	0,868	226	157	150

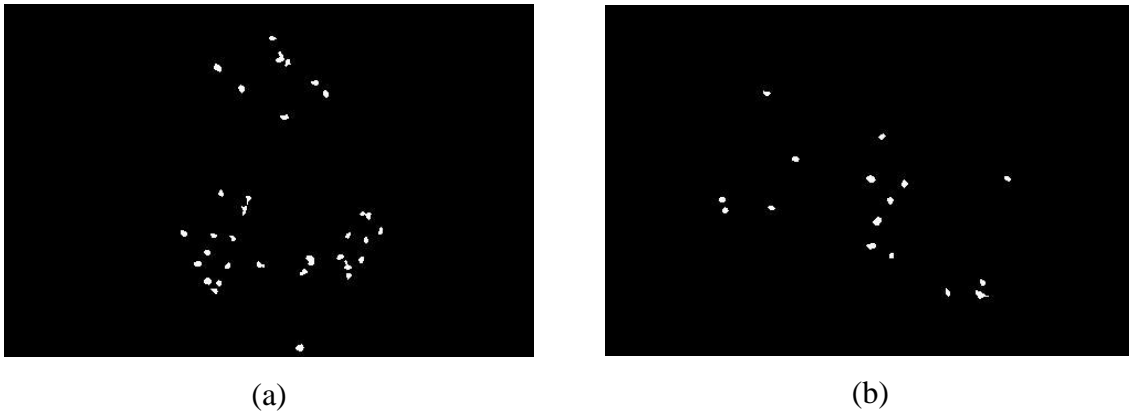
Tabel 4.4 Ciri Jerawat Data *Training* II

No Jerawat	Area	Eccentricity	Mean Intensity		
			Red	Green	Blue
1	26	0,535	200	158	148
2	24	0,341	198	156	143
3	31	0,807	189	162	155
4	27	0,671	202	162	153
5	42	0,660	203	156	144
6	37	0,772	185	134	120
7	44	0,681	184	131	118
8	29	0,555	205	156	142
9	29	0,586	200	147	134
10	19	0,619	189	143	128
Min	19	0,341	184	131	118
Max	44	0,807	205	162	155

Tabel 4.5 Parameter Percobaan Ciri Jerawat

Skala	Area	Eccentricity	Mean Intensity		
			Red	Green	Blue
Min	19	0,34	135	84	60
Max	52	0,87	226	162	155

Informasi jerawat pada Tabel 4.5 merupakan skala yang dibentuk dari kedua data *training* dan dijadikan sebagai parameter *threshold* untuk mengenali jerawat. Hasil dan pengujian ekstraksi ciri pada kedua data *training* dapat dilihat pada Gambar 4.6 dan Tabel 4.6 berikut.



Gambar 4.6 Hasil Percobaan Ekstraksi Ciri; (a) Data *Training* I. (b) Data *Training* II

Tabel 4.6 Hasil Uji Percobaan Ekstraksi Ciri

Nama	True Positive	True Negatif	False Positive	False Negative	Sensitivity	Specificity
Data <i>Training</i> I	13	181	19	2	0,867	0,905
Data <i>Training</i> II	10	105	6	0	1,000	0,946

Tingkat kebenaran deteksi dalam mengenali jerawat sudah baik, dengan nilai *sensitivity* mencapai angka lebih dari 0,8. Walaupun memiliki nilai *specificity* yang tinggi, tingkat kesalahan sistem dalam mendeteksi jerawat masih tinggi. Hal itu terbukti dari 32 objek yang terdeteksi pada data *training* I terdapat 19 objek bukan jerawat, dengan kata lain lebih dari 50 % hasil deteksi pada data *training* I adalah bukan jerawat. Sehingga tingkat kesalahan deteksi masih perlu diperbaiki. Parameter informasi diubah guna memperbaiki kesalahan deteksi sistem. Parameter *area* dan *eccentricity* diubah menjadi seperti pada Tabel 4.7.

Tabel 4.7 Parameter Area dan *Eccentricity*

Skala	Area	Eccentricity
Min	24	0
Max	381*	0,81

* mean area + standar deviasi area

Nilai parameter minimal area diubah menjadi 24 yang merupakan nilai area terkecil kedua dari data *training* untuk mempersempit skala luas area, sehingga dapat menghindari deteksi objek bukan jerawat. Luas area wajah pada citra akan berpengaruh terhadap ukuran jerawat yang terdeteksi dengan *filter* LoG, sehingga ukuran terbesar jerawat bervariasi. Oleh karena itu, dibutuhkan otomatisasi nilai parameter maksimal area dengan memanfaatkan mean dan standar deviasi seluruh objek seperti pada Persamaan (3.1).

$$\text{max area} = \text{mean area} - \text{standar deviasi area} \quad (3.1)$$

Parameter minimal *eccentricity* dihilangkan karena semakin kecil nilainya maka semakin bundar, dengan dasar jerawat berbentuk kubah (kebundaran). Sedangkan nilai maksimal *eccentricity* diubah menjadi 0,81 yang merupakan nilai *eccentricity* terbesar kedua dari data *training*, sehingga terdapat lima objek jerawat pada data *training* I yang tereliminasi. Hasil pengujian pengubahan parameter area dan *eccentricity* dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil Uji Perubahan Parameter Area dan *Eccentricity*

Nama	True Positive	True Negative	False Positive	False Negative	Sensitivity	Specificity
Data <i>Training</i> I	12	197	4	3	0,800	0,980
Data <i>Training</i> II	8	106	0	2	0,800	1,000

Nilai *specificity* menjadi lebih baik setelah parameter *area* dan *eccentricity* diubah, begitu juga dengan tingkat kesalahan deteksi sistem. Tetapi nilai *sensitivity* turun dikarenakan tidak terdeteksinya satu jerawat pada data *training* I dan dua pada data *training* II yang terdeteksi pada parameter sebelumnya.

Tabel 4.9 Ciri Jerawat Yang Tidak Terdeteksi

Data Training	No Jerawat	Area	Eccentricity	Mean Intensity		
				Red	Green	Blue
I	6	31	0,582	226,2	156,5	145,0
II	3	31	0,807	189,2	162,2	155,4
	10	19	0,619	189,4	143,3	128,4

Tabel 4.9 menunjukkan bahwa kegagalan deteksi diakibatkan oleh nilai *mean intensity* merupakan bilangan desimal. Sehingga parameter *mean intensity* perlu diubah. Pada penelitian ini warna wajah memiliki rentang putih hingga kuning langsung, sehingga memungkinkan parameter intensitas warna jerawat berubah-ubah. Oleh karena itu, perlu dilakukan otomatisasi parameter intensitas warna untuk menyesuaikan intensitas warna jerawat pada warna wajah yang berbeda. Otomatisasi warna memanfaatkan metode mean dan standar deviasi dan diformulasikan pada persamaan berikut.

$$\text{min} = \text{mean}(\text{layer}) - \text{standar deviasi}(\text{layer}) \times 1,75 \quad (3.2)$$

$$max = mean(layer) + standar\ deviasi(layer) \times 1,75 \quad (3.3)$$

Standar deviasi dikalikan dengan 1,75 untuk memperbesar skala intensitas warna, agar tidak terjadi *miss detection* dikarenakan perbedaan nilai desimal. Nilai tersebut didapat dari percobaan perkalian standar deviasi dengan 1,05 hingga 2,00, dimana didapat hasil yang maksimal pada 1,75. Jika terlalu kecil maka skala parameter semakin tipis, sehingga terdapat jerawat yang tereliminasi. Sedangkan jika terlalu besar maka skala parameter terlalu tebal, sehingga akan banyak objek bukan jerawat yang terdeteksi. Hasil otomatisasi parameter intensitas warna dapat dilihat pada Tabel 4.10.

Tabel 4.10 Parameter Intensitas Warna

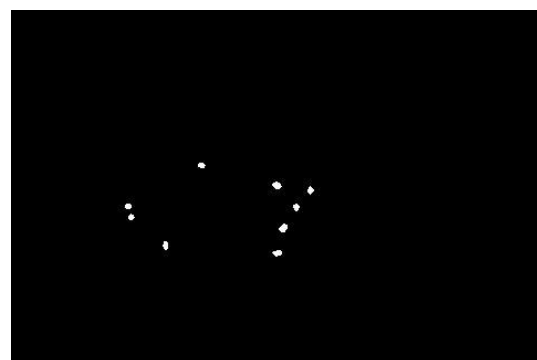
Data Training	No Jerawat	Mean Intensity		
		Red*	Green*	Blue*
I	Min	150	77	48
	Max	244	179	167
II	Min	130	116	114
	Max	242	179	163

* mean + std. deviasi x 1,75

Parameter *mean intensity* pada kedua data *training* menjadi berbeda dengan otomatisasi. Semua parameter yang telah diukur ulang kemudian digabungkan. Hasil perubahan parameter dapat dilihat pada Gambar 4.7 dan Tabel 4.11.



(a)



(b)

Gambar 4.7 Hasil Ekstraksi Ciri; (a) Data *Training* I dan (b) Data *Training* II

Tabel 4.11 Hasil Pengujian Ekstraksi Ciri

Nama	True Positive	True Negative	False Positive	False Negative	Sensitivity	Specificity
Data <i>Training</i> I	13	200	3	2	0,867	0,985
Data <i>Training</i> II	9	112	0	1	0,900	1,000

Hasil perubahan parameter didapat nilai *sensitivity* dan *specificity* tetap tinggi, dengan melebihi angka 0,8. Kesalahan deteksi juga rendah, yaitu hanya 3 kesalahan dari 16 deteksi pada data *training* I dan tidak ada kesalahan deteksi pada data *training* II. Sehingga parameter ekstraksi ciri yang digunakan pada penelitian ini ditetapkan seperti pada Tabel 4.12.

Tabel 4.12 Parameter Ekstraksi Ciri

Skala	Area	Eccentricity	Mean Intensity		
			Red	Green	Blue
Min	21	0	**	**	**
Max	*	0,81	**	**	**

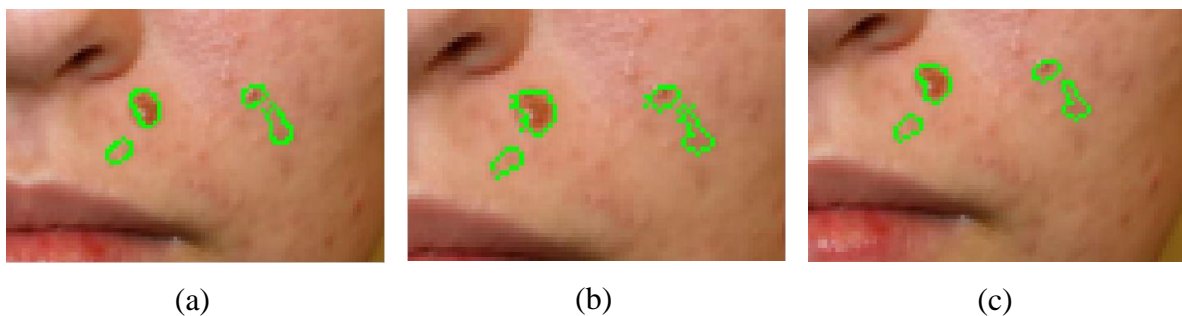
* mean + std. deviasi

** mean + std. deviasi x 1,75

4.7. *Marking*

Marking merupakan proses untuk menandai jerawat yang telah tersegmentasi pada proses sebelumnya. Penandaan berupa pemberian warna *outline* (garis tepi) jerawat yang telah tersegmentasi. Warna yang dipilih adalah warna yang berbeda dengan warna kulit dan jerawat, sehingga mudah dikenali secara kasat mata.

Penandaan jerawat pada citra memanfaatkan metode *edge detection*. Dari sekian banyak metode *edge detection*, didapat tiga metode yang dipertimbangkan untuk deteksi tepi objek yaitu prewitt, sobel dan canny. Perbandingan deteksi tepi dapat dilihat pada Gambar 4.8.



Gambar 4.8 Perbandingan *Edge Detection*; (a) Canny, (b) Prewitt dan (c) Sobel

Hasil deteksi tepi canny dan sobel lebih rapi dibanding prewitt. Pengujian kecepatan proses dilakukan lima kali percobaan pada setiap metode untuk mencari waktu tercepat dengan hasil perbandingan pada Tabel 4.13.

Tabel 4.13 Perbandingan Kecepatan Proses *Edge Detection*

Metode	Waktu (Detik)
Canny	0,241496
Prewitt	0,271020
Sobel	0,257839

Dari ketiga metode tersebut, canny merupakan metode dengan proses tercepat. Dengan hasil deteksi yang rapi dan waktu proses tercepat, metode ***edge detection canny*** digunakan untuk menandai jerawat yang terdeteksi.

BAB V

HASIL DAN PEMBAHASAN

5.1. Implementasi Perangkat Lunak

Semua metode dari perancangan perangkat lunak yang telah dirancang kemudian dilakukan implementasi dengan menggunakan MATLAB R2014a.

5.1.1. *Resize* Citra

Resize citra merupakan proses untuk mengubah ukuran citra menjadi lebih kecil jika ukuran dimensi citra terlalu besar. Kode program *resize* citra dapat dilihat pada Gambar 5.1.

```
citra = imread('2q1f5oj.jpg');  
  
% Ukuran Citra Semula  
[bar, kol, dlm] = size(citra);  
  
if (bar > kol)  
    maxLength = bar;  
    if (maxLength >= 480);  
        citra = imresize(citra, [480 NaN]);  
    end;  
else  
    maxLength = kol;  
    if (maxLength >= 480);  
        citra = imresize(citra, [NaN 480]);  
    end;  
end;
```

Gambar 5.1 Kode Program *Resize* Citra

Citra *input* akan di-*resize* jika memiliki ukuran baris atau kolom lebih dari 480 piksel. Setelah memenuhi kondisi tersebut, ukuran baris dan kolom akan dibandingkan. Jika baris lebih besar dari kolom, maka ukuran baris akan diubah menjadi 480 piksel dan ukuran kolom akan disesuaikan dengan perbandingan ukuran baris agar bentuk citra tidak berubah. Sedangkan jika kolom lebih besar dari baris, maka ukuran kolom akan diubah menjadi 480 piksel dan ukuran baris akan disesuaikan dengan perbandingan ukuran kolom agar bentuk citra tidak berubah.

5.1.2. Segmentasi Kulit

Segmentasi kulit merupakan proses untuk memisahkan objek kulit dengan *background* dan objek selain kulit. Proses pemisahan kulit dalam penelitian ini memanfaatkan fungsi *connected-component* yang sudah disediakan MATLAB. Kode program segmentasi kulit dapat dilihat pada Gambar 5.2.

```

citraG = rgb2gray(citra);
level = graythresh(citraG);

citraB = im2bw(citraG, level);
citraB = imfill(citraB, 'holes');

citraCC = bwconncomp(citraB);
citraL = labelmatrix(citraCC);

% Ukuran Citra Setelah Resize
[bar, kol, dlm] = size(citra);

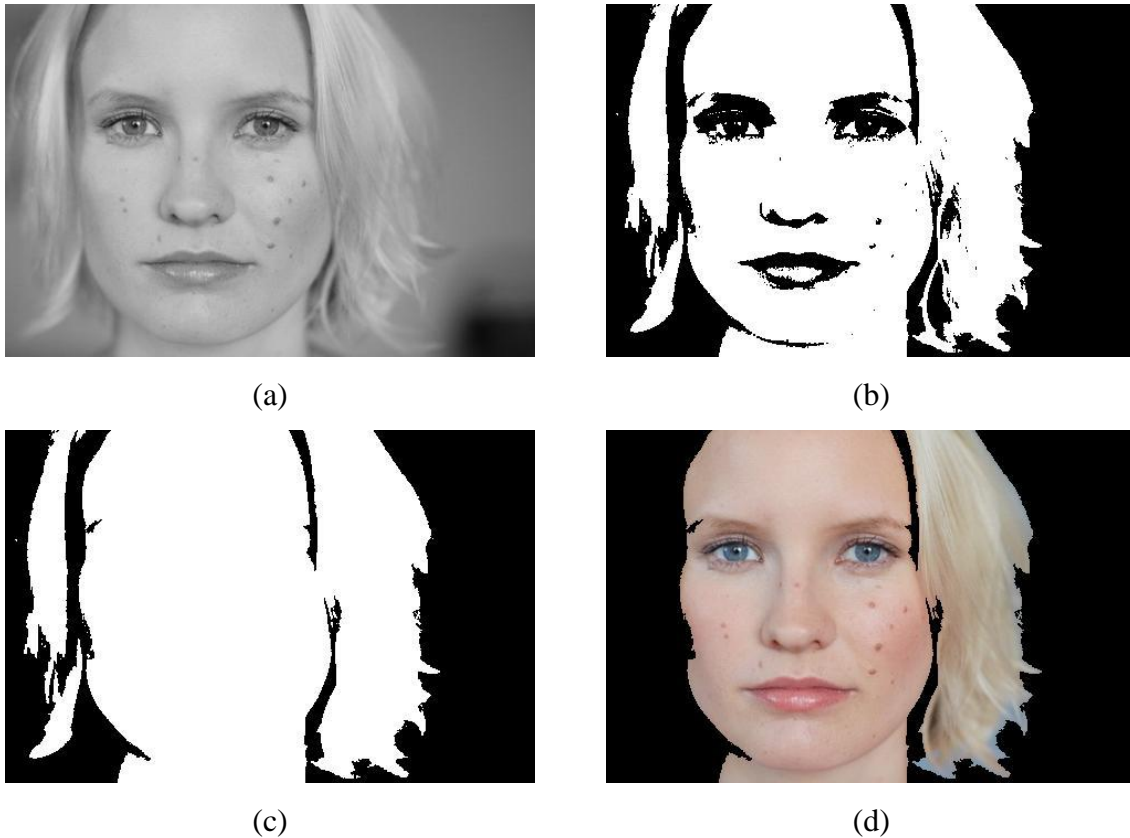
% Cari Connected Component Terbanyak
terluas = 0;
for i = 1 : length(citraCC.PixelIdxList)
    if (length(citraCC.PixelIdxList{i})) > terluas
        terluas = length(citraCC.PixelIdxList{i});
        index = i;
    end;
end;

% Ambil Label
wajah = uint8(zeros(bar, kol, dlm));
for i = 1 : bar
    for j = 1 : kol
        if citraL(i, j) == index;
            wajah(i, j, :) = citra(i, j, :);
        end;
    end;
end;

```

Gambar 5.2 Kode Program Segmentasi Kulit

Citra yang sudah melalui proses *resize* citra dikonversi menjadi citra biner dengan metode Otsu. Objek pada citra biner dikelompokkan menggunakan *connected-component* dan diberi label untuk tiap objek yang terpisah. Setiap objek akan dihitung luasnya dan dicari objek dengan luas terbesar untuk menentukan objek kulit. Objek tersebut diambil nilai indeksinya dan dikembalikan nilai warnanya menjadi RGB. Tahap perubahan citra dari proses segmentasi kulit dapat dilihat pada Gambar 5.3 dibawah ini.



Gambar 5.3 Tahap Segmentasi Kulit; (a) *Grayscale*. (b) Biner. (c) Imfill. (d) Hasil

5.1.3. Perbaikan Citra

Proses perbaikan citra merupakan proses untuk mencari kandidat jerawat pada citra. Proses perbaikan citra menggunakan metode *sharpening* Laplacian of Gaussian dengan memanfaatkan fungsi **fspecial** dan **imfilter** yang disediakan MATLAB. Kode program perbaikan citra dapat dilihat pada Gambar 5.4.

```
% Sharpening
wajahG = rgb2gray(wajah);
h = fspecial('log', [9 9], 2);
m = imfilter(wajahG, h, 'circular', 'same', 'conv');

% Labeling
candidate = logical(m);
[labeledCandidate, numberOfCandidates] = bwlabel(candidate, 8);
```

Gambar 5.4 Kode Program Perbaikan Citra

Citra yang telah disegmentasi dikonversi menjadi citra *grayscale*. Kemudian dilakukan *sharpening* pada citra tersebut dengan menggunakan metode Laplacian of Gaussian, sehingga menghasilkan sebuah keluaran citra LoG dalam bentuk citra biner. Citra yang dihasilkan dari *sharpening* menggunakan *filtering* Laplacian of Gaussian ini sudah terdapat jerawat di

dalamnya. Sehingga citra Laplacian of Gaussian ini disebut sebagai kandidat jerawat. Tahap perubahan citra dari proses perbaikan citra dapat dilihat pada Gambar 5.5 dibawah ini.



Gambar 5.5 Tahap Perbaikan Citra; (a) Citra *Grayscale* (b) Citra LoG

5.1.4. Ekstraksi Ciri

Proses ekstraksi ciri bertujuan untuk mengenali objek jerawat dan memisahkannya berdasarkan kondisi yang mematuhi. Adapun dalam penelitian ini menggunakan metode ekstraksi ciri berdasarkan luas, bentuk dan warna.

Luas dan Bentuk

```
% Eliminasi Berdasarkan Luas Dan Bentuk
blobMeasurements = regionprops(labeledCandidate, 'Area', 'Eccentricity');

allArea = [blobMeasurements.Area];
allEccentricity = [blobMeasurements.Eccentricity];

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

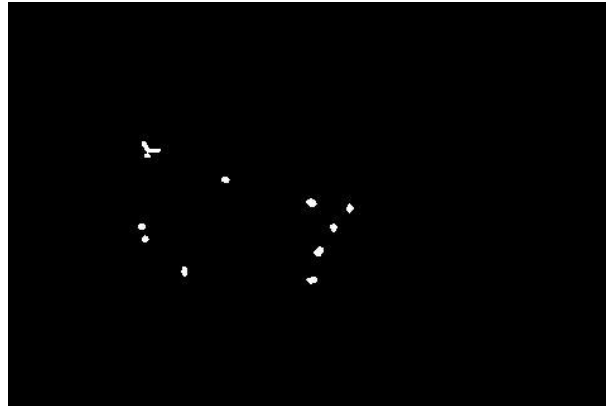
indexBlob = find(allArea >= 24 & allArea <= (meanArea + stdArea) &
allEccentricity < 0.81);
ambilBlob = ismember(labeledCandidate, indexBlob);

blobBW = ambilBlob > 0;
[labeledBlob, numberOfBlobs] = bwlablel(blobBW);
```

Gambar 5.6 Kode Program Ekstraksi Ciri Luas dan Bentuk

Pada Gambar 5.6, *input* proses yaitu citra LoG yang kemudian diberi label setiap objeknya. Setiap objek yang telah diberi label dihitung nilai *eccentricity* dan luasnya. Kemudian objek-objek tersebut diseleksi dengan kondisi luas area lebih dari sama dengan 24 piksel dan *eccentricity* kurang dari sama dengan 0,81. Selanjutnya objek yang tidak memenuhi

kondisi akan dieliminasi dari citra tersebut, sehingga hanya tersisa objek yang memenuhi kondisi tersebut. Hasil perubahan citra dari proses ekstraksi ciri luas dan bentuk dapat dilihat pada Gambar 5.7 dibawah ini.



Gambar 5.7 Hasil Ekstraksi Ciri Luas dan Bentuk

Warna

```
% Eliminasi Berdasarkan Warna
red = citra(:, :, 1);
green = citra(:, :, 2);
blue = citra(:, :, 3);

r = regionprops(labeledBlob, red, 'MeanIntensity');
g = regionprops(labeledBlob, green, 'MeanIntensity');
b = regionprops(labeledBlob, blue, 'MeanIntensity');

fiturR = [r.MeanIntensity]';
fiturG = [g.MeanIntensity]';
fiturB = [b.MeanIntensity]';
fitur = [fiturR fiturG fiturB];

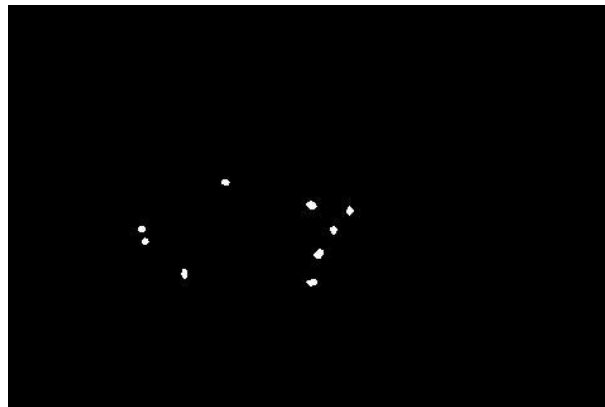
meanR = mean(fiturR);
meanG = mean(fiturG);
meanB = mean(fiturB);
stdR = std(fiturR);
stdG = std(fiturG);
stdB = std(fiturB);

indexJerawat = [];
for i = 1 : numberOfBlobs
    if(fiturR(i) >= (meanR-stdR*1.75) && fiturR(i) <= (meanR+stdR*1.75) &&
fiturG(i) >= (meanG-stdG*1.75) && fiturG(i) <= (meanG+stdG*1.75) &&
fiturB(i) >= (meanB-stdB*1.75) && fiturB(i) <= (meanB+stdB*1.75))
        indexJerawat = [indexJerawat i];
    end;
end;

jumlahJerawat = length(index);
jerawatBW = ismember(labeledBlob, indexJerawat);
```

Gambar 5.8 Kode Program Ekstraksi Ciri Warna

Pada Gambar 5.8, *input* berupa citra eliminasi yang telah disegmentasi berdasarkan luas dan bentuk. Setiap objek pada citra tersebut diberi label dan dihitung nilai *mean intensity* dari komponen warna RGB per objek. Kemudian objek tersebut diseleksi nilai *mean intensity* komponen warna RGB tiap objek dengan rentang warna berupa komputasi nilai *mean* dan standar deviasi dari komponen warna RGB objek dari citra tersebut, sehingga tersisa objek-objek yang dikenali sebagai jerawat. Keluaran dari proses ini adalah citra jerawat berupa citra biner. Hasil perubahan citra dari proses ekstraksi ciri warna dapat dilihat pada Gambar 5.9 dibawah ini.



Gambar 5.9 Hasil Ekstraksi Ciri Warna

5.1.5. *Marking*

Proses *marking* merupakan proses untuk menandai setiap objek jerawat yang telah dikenali dengan memberi garis tepi pada jerawat yang telah dikenali. Kode program *marking* dapat pada Gambar 5.10.

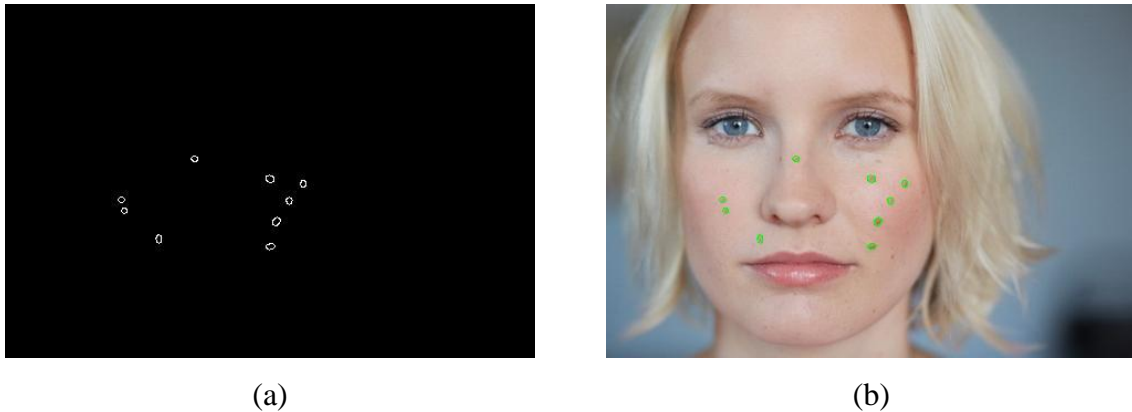
```
jerawatEdge = edge(jerawatBW, 'canny');
hasil = citra;

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

hasil = uint8(hasil);
```

Gambar 5.10 Kode Program *Marking*

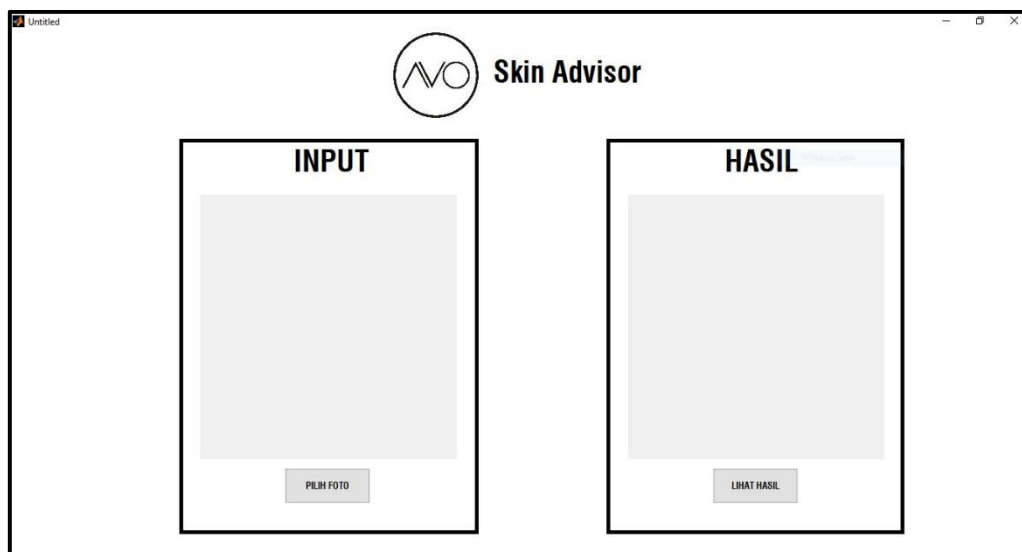
Input dari proses *marking* adalah citra jerawat biner yang sudah disegmentasi. Citra tersebut akan dideteksi tepi dengan metode Canny, sehingga citra hanya terdapat garis tepi dari objek jerawat saja. Koordinat garis tepi dari objek jerawat tersebut akan dijadikan acuan untuk garis tanda objek jerawat pada citra awal. Garis tepi objek jerawat pada citra awal akan diberi warna hijau untuk menandakan bahwa objek yang diberi garis tersebut adalah jerawat. Tahap perubahan citra dari proses *marking* dapat dilihat pada Gambar 5.11 dibawah ini.



Gambar 5.11 Tahap *Marking*; (a) Canny. (b) Hasil

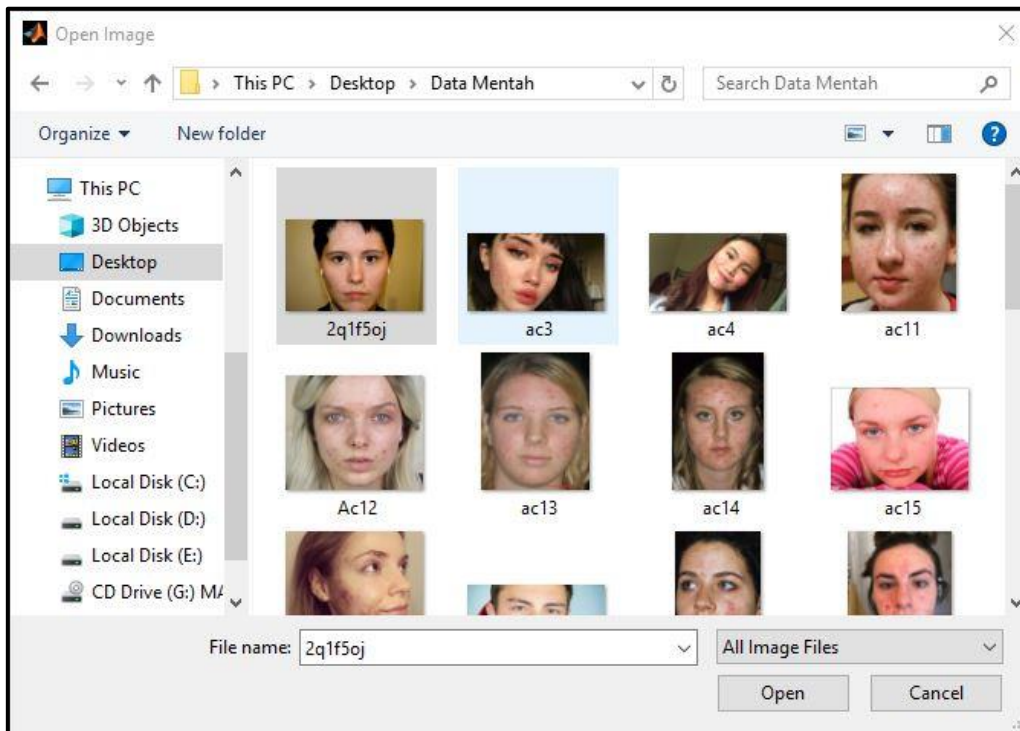
5.2. Implementasi Antarmuka

Rancangan dari antarmuka diimplementasikan dengan tools GUIDE pada MATLAB. Berikut hasil implementasi antarmuka dapat dilihat pada Gambar 5.12.



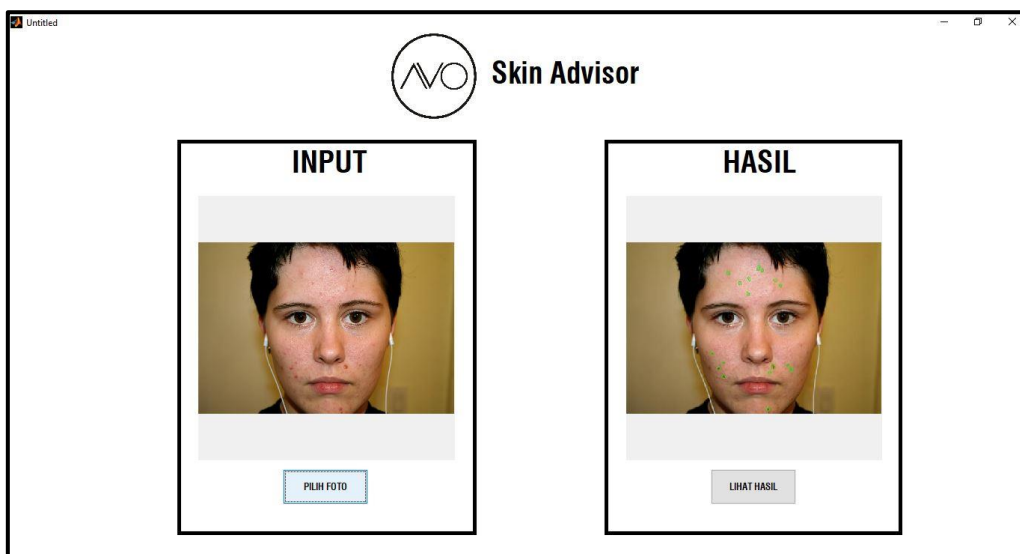
Gambar 5.12 Antarmuka Kosong

Saat menekan tombol PILIH FOTO, maka sistem akan mengeluarkan *pop-up* window *file input* untuk memilih *file* foto seperti terlihat pada Gambar 5.12 berikut.



Gambar 5.13 File Input

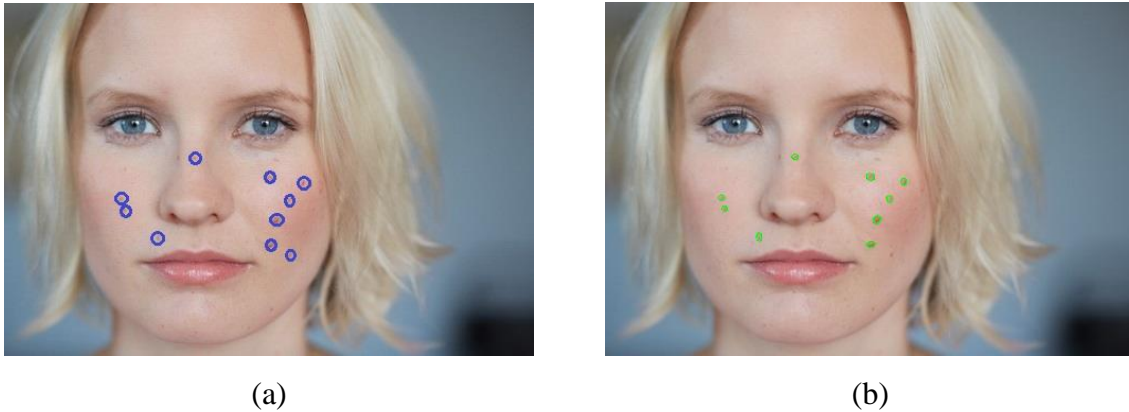
Setelah *file* berhasil dimasukkan maka foto akan ter-*load* dan dengan menekan tombol LIHAT hasil, maka sistem akan menampilkan hasil dari citra dengan jerawat yang telah ditandai seperti pada Gambar 5.14.



Gambar 5.14 Antarmuka Aktif

5.3. Pengujian Sistem

Pengujian sistem dilakukan dengan cara membandingkan hasil deteksi sistem dengan citra yang telah ditandai oleh pakar, seperti pada Gambar 5.15.



Gambar 5.15 Perbandingan Penandaan; (a) Pakar. (b) Sistem

Pengujian sistem pada penelitian ini dengan menggunakan *single decision threshold* dengan menggunakan 2 parameter yaitu *sensitivity* dan *specificity*. Berikut tabel hasil uji sistem dengan membandingkan hasil penandaan oleh pakar (AVO).

Tabel 5.1 Hasil Pengujian Sistem

No	Nama File	True Positive	True Negative	False Positive	False Negative	Sensitivity	Specificity
1	01.jpg	13	200	3	2	0,867	0,985
2	02.jpg	2	95	3	12	0,143	0,969
3	03.jpg	9	112	0	1	0,9	1
4	04.jpg	10	111	4	6	0,625	0,965
5	05.jpg	9	51	1	15	0,375	0,981
6	06.jpg	9	149	5	13	0,409	0,968
7	07.jpg	8	156	6	16	0,333	0,963
8	08.jpg	5	201	2	8	0,385	0,99
9	9.jpg	5	459	8	7	0,417	0,983
10	10.jpg	8	295	11	12	0,4	0,964
11	11.jpg	24	346	9	23	0,511	0,975
12	12.jpg	22	275	6	22	0,5	0,979
13	13.jpg	3	310	4	4	0,429	0,987
14	14.jpg	12	390	15	16	0,429	0,963
15	15.jpg	9	102	3	15	0,375	0,971
16	16.jpg	33	508	20	26	0,559	0,962
17	17.jpg	12	170	12	17	0,414	0,934

18	18.jpg	16	261	6	8	0,667	0,978
19	19.jpg	13	331	11	16	0,448	0,968
20	20.jpg	9	69	1	11	0,450	0,986
21	21.jpg	8	256	3	3	0,727	0,988
22	22.jpg	9	54	2	12	0,429	0,964
23	23.jpg	13	158	2	16	0,448	0,988
24	24.jpg	7	297	13	7	0,5	0,958
25	25.jpg	23	385	16	41	0,359	0,96
26	26.jpg	11	195	2	18	0,379	0,99
27	27.jpg	15	148	1	11	0,577	0,993
28	28.jpg	23	238	1	32	0,418	0,996
29	29.jpg	19	133	2	12	0,613	0,985
30	30.jpg	19	288	25	17	0,528	0,92
31	31.jpg	11	123	2	16	0,407	0,984
32	32.jpg	6	46	7	8	0,429	0,868
33	33.jpg	6	94	2	12	0,333	0,979
34	34.jpg	12	109	0	12	0,5	1
35	35.jpg	8	60	1	17	0,32	0,984
36	36.jpg	12	125	6	16	0,429	0,954
Total		433	7300	215	500		

Keterangan :

1. **True Positive** (TP) : Jumlah jerawat yang terdeteksi di sistem dan ditandai pada citra pakar.
2. **True Negative** (TN) : Jumlah kandidat jerawat yang tidak terdeteksi di sistem.
3. **False Positive** (FP) : Jumlah jerawat yang terdeteksi di sistem, tetapi tidak ditandai pada citra pakar.
4. **False Negative** (FN) : Jumlah jerawat yang tidak terdeteksi di sistem, tetapi ditandai pada citra pakar.

Terdapat dua kolom tambahan pada Tabel 5.1, yaitu *sensitivity* untuk mengukur tingkat keberhasilan sistem dalam mengenali objek jerawat dan *specificity* untuk mengukur tingkat kesalahan sistem dalam mengenali objek bukan jerawat. Setiap baris data diukur nilai *sensitivity* dan *specificity* dengan mengaplikasikan Persamaan (2.11) dan Persamaan (2.12).

Dari data yang telah diolah, didapat nilai *sensitivity* tertinggi sebesar 0,867 dan terendah sebesar 0,143. Sedangkan untuk nilai *specificity*, nilai tertinggi sebesar 1 dan terendah sebesar 0,868. Untuk mengukur sistem secara keseluruhan, dilakukan perhitungan nilai *sensitivity* dan *specificity* dengan mengaplikasikan total nilai TP, TN, FP, dan FN, sehingga didapat hasil seperti pada Persamaan (5.1).

$$\begin{aligned}
 \text{Sensitivity} &= \frac{TP}{TP + FN}, & \text{Specificity} &= \frac{TN}{TN + FP} \\
 &= \frac{433}{433 + 500} = 0,464 & &= \frac{7300}{7300 + 215} = 0,971
 \end{aligned}
 \tag{5.1}$$

Tabel 5.2 Tabel *Contingency* Pengujian

		Pakar	
		Jerawat	Non-Jerawat
Sistem	Jerawat	433	215
	Non-Jerawat	500	7300
		Sensitivity	Specificity
		0,464	0,971

Dari hasil pengolahan data pada Tabel 5.2, sistem mendapatkan nilai *sensitivity* sebesar 0,464 atau dalam satuan persen yaitu 46,4%. Berdasarkan nilai tersebut, dapat diartikan sistem belum maksimal dalam mengenali jerawat karena nilai kebenaran deteksi jerawat masih belum sampai pada angka 50%. Sedangkan untuk *specificity*, sistem mendapatkan nilai sebesar 0,971 atau 97,1%. Hasil tersebut sangatlah baik karena kesalahan sistem untuk mendeteksi objek bukan jerawat sangat kecil, dengan melihat nilai *specificity* yang mencapai angka 90%.

5.4. Pengujian Kecepatan Sistem

Tahap pengujian selanjutnya adalah pengujian kinerja sistem. Tujuan pengujian kinerja sistem adalah untuk mengetahui seberapa cepat sistem dapat memproses *input* hingga menjadi sebuah *output*. Pengujian dilakukan dengan menguji 5 data sampel dengan ukuran dimensi yang beragam.

Tabel 5.3 Hasil Pengujian Kinerja Sistem

No	Nama File	Dimensi (BxK)	Proses (Detik)					Total
			Resize	Segmentasi Kulit	Perbaikan Citra	Ekstraksi Ciri	Marking	
1	03.jpg	500x333	0,0687	0,4473	0,0538	0,9906	0,3058	1,8661
2	19.jpg	959x1280	0,0559	0,7288	0,1134	0,9606	0,4862	2,3448
3	02.jpg	1200x1600	0,0595	2,1011	0,0434	0,6869	0,3736	3,2644
4	11.jpg	800x600	0,0317	0,5586	0,0376	0,6561	0,3027	1,5866
5	25.jpg	960x1280	0,0541	0,7058	0,0491	0,7927	0,4063	2,0080
Rata-Rata			0,0540	0,9083	0,0594	0,8174	0,3749	2,2140

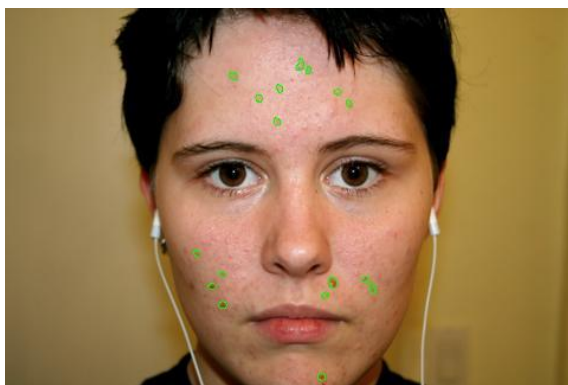
Berdasarkan Tabel 5.3, didapat performa keseluruhan proses dari sistem adalah 2,214. Dari 5 data sampel, kinerja tercepat yaitu 1,5866 pada data ke-4. Sedangkan kinerja terlambat yaitu 3,2644 pada data ke-3. Walaupun kinerja terlambat terdapat pada citra dengan dimensi terbesar, tetapi tidak dengan kinerja tercepat yang bukan pada data dengan dimensi terkecil. Sehingga kecepatan kinerja sistem tidak dapat ditentukan oleh ukuran dimensi citra.

5.5. Perbandingan Dengan Sistem Sejenis

Pada latar belakang dikatakan sudah terdapat sistem sejenis yang dimiliki oleh Ponds, pesaing AVO. Oleh karena itu, dilakukan perbandingan hasil deteksi antara sistem yang dikembangkan dengan sistem Ponds. Perbandingan menggunakan *F1 Score* untuk mengukur sistem mana yang mendekati hasil penandaan AVO. *F1 Score* memiliki skala 0 sampai 1, dimana semakin besar nilai maka semakin baik. Formulasi *F1 Score* dapat dilihat pada Persamaan (5.2).

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (5.2)$$

Dari 36 data yang diujikan pada sistem, terdapat 10 data yang memenuhi kebutuhan *input* sistem Ponds. Hasil perbandingan antara sistem yang dikembangkan dengan sistem Ponds dapat dilihat pada Tabel 5.4 dan grafik pada Gambar 5.17.



(a)

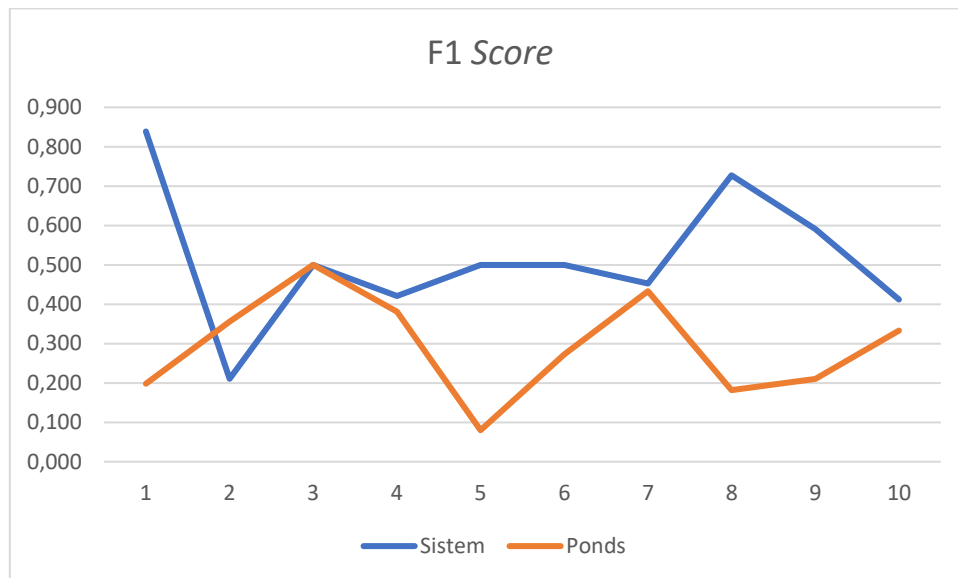


(b)

Gambar 5.16 Perbandingan Hasil Deteksi Data 1; (a) Sistem. (b) Ponds

Tabel 5.4 Perbandingan Hasil Sistem Dengan Ponds

No	Nama File	Sistem				Ponds			
		TP	FP	FN	F1 Score	TP	FP	FN	F1 Score
1	01.jpg	13	3	2	0,839	11	85	4	0,198
2	02.jpg	2	3	12	0,211	13	46	1	0,356
3	06.jpg	9	5	13	0,500	6	6	6	0,500
4	07.jpg	8	6	16	0,421	8	10	16	0,381
5	08.jpg	5	2	8	0,500	3	53	16	0,080
6	15.jpg	9	3	15	0,500	3	6	10	0,273
7	17.jpg	12	12	17	0,453	21	34	21	0,433
8	21.jpg	8	3	3	0,727	7	55	8	0,182
9	23.jpg	13	2	16	0,591	4	26	4	0,211
10	24.jpg	7	13	7	0,412	3	7	5	0,333
Total		86	52	109	0,517	79	328	91	0,274



Gambar 5.17 Grafik Perbandingan Hasil Deteksi Sistem Dengan Ponds

* x: nomer data

** y: skala f1 score

Hasil perbandingan kedua sistem dengan *F1 Score* menunjukkan sistem yang dikembangkan lebih baik dari Ponds. Hal itu dapat dilihat dari nilai keseluruhan sistem yang dikembangkan dengan sistem Ponds masing-masing 0,517 dan 0,274. Walaupun memiliki nilai keseluruhan yang lebih baik, tetapi sistem yang dikembangkan masih memiliki kekurangan dari sistem Ponds.



(a)

(b)

Gambar 5.18 Perbandingan Hasil Deteksi Data 2; (a) Sistem. (b) Ponds

Hasil deteksi data nomer dua memiliki nilai yang lebih rendah dari Ponds. Hal itu terjadi karena pencahayaan pada wajah cenderung gelap dan tidak merata seperti pada Gambar 5.18. Pada data tersebut, Ponds dapat mengenali 13 jerawat dari 59 objek yang terdeteksi. Sedangkan sistem yang dikembangkan hanya mengenali 2 jerawat dari 5 objek yang terdeteksi. Angka tersebut cukup jauh jika dibandingkan. Sehingga dapat disimpulkan bahwa secara keseluruhan sistem lebih baik dari Ponds. Tetapi, sistem yang dikembangkan masih belum cukup baik mendeteksi citra dengan pencahayaan tidak merata, yang mana hal tersebut menjadi keunggulan Ponds.

5.6. Kelebihan dan Kekurangan Sistem

a. Kelebihan Sistem

- Deteksi jerawat dengan jumlah sedikit sudah bagus.
- Sistem dapat memproses citra dengan ukuran dimensi yang besar.
- Sistem dapat menghindari kesalahan pendeteksian objek yang baik dengan specificity lebih dari 0,9.

b. Kekurangan Sistem

- Identifikasi sistem terhadap jerawat masih rendah.
- Jika pencahayaan tidak merata, hasil kurang baik.

- Sistem masih kurang akurat mendeteksi jerawat dengan jumlah banyak.
- Segmentasi wajah yang kurang sempurna membuat sebagian jerawat tidak terdeteksi.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan seluruh proses penelitian yang dilakukan pada tugas akhir ini, Berikut ini kesimpulan yang dapat diambil :

1. Sistem dapat mengenali jerawat pada citra wajah.
2. Nilai *sensitivity* dan *specificity* yang didapat masing-masing 0,464 dan 0,971.
3. Nilai *sensitivity* yang di bawah 0,5 sehingga masih banyak jerawat yang belum terdeteksi.
4. Sistem dapat mengenali objek bukan jerawat berdasarkan nilai *specificity* yang mencapai angka 0,9. Sehingga kesalahan deteksi jerawat kecil.
5. Dari 36 data, hanya 9 data yang mencapai 0,5 nilai *sensitivity* yang menandakan proses deteksi masih perlu ditingkatkan.

6.2. Saran

Dalam penelitian tugas akhir ini, penulis mengetahui masih memiliki banyak kekurangan yang perlu diperbaiki. Untuk penelitian lebih lanjut, berikut saran-saran yang dapat dipertimbangkan :

1. Untuk penelitian selanjutnya, lebih baik konsultasi dengan pakar secara langsung. Konsultasi dengan pakar dapat memperdalam pemahaman penelitian terhadap jerawat, sehingga identifikasi jerawat lebih rinci.
2. Data yang digunakan pada penelitian ini didapat dari internet, sehingga sedikit sulit untuk mendapat citra yang memenuhi. Untuk penelitian selanjutnya lebih baik mencari data langsung di lapangan atau bekerjasama dengan dokter kulit sehingga data yang diperoleh bisa memenuhi standar kebutuhan.

DAFTAR PUSTAKA

- Ahmad, U. (2005). *Pengolahan Citra Digital & Teknik Pemrogramannya*. Yogyakarta: Graha Ilmu.
- Dethira, A. (2016, Agustus 24). *Anugerah Pakerti dkk. Dari Yogya Melejitkan Kosmetik*. Diambil kembali dari SWA: <https://swa.co.id/youngster-inc/youngsterinc-startup/anugerah-pakerti-dkk-dari-yogya-melejitkan-kosmetik>
- Djuanda, A., Hamzah, M., & Aisah, S. (1999). *Ilmu Penyakit Kulit dan Kelamin*. Jakarta: Fakultas Kedokteran Universitas Indonesia.
- Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing Second Edition*. New Jersey: Prentice Hall.
- Kong, H., Akakin, H. C., & Sarma, S. E. (2013). A Generalized Laplacian of Gaussian Filter. *IEEE TRANSACTIONS ON CYBERNETICS*, 43(6), 1719-1732.
- Mujahidin, S. (2012). *Aplikasi Perhitungan Jumlah Orang Dalam Satu Foto*. Yogyakarta: Universitas Islam Indonesia.
- Munir, R. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.
- Murtiastutik, D., dkk. (2007). *Atlas Penyakit Kulit & Kelamin Edisi 2*. Surabaya: Airlangga University Press.
- Shortliffe, E. H., & Cimino, J. J. (2010). *Biomedical Informatics: Computer Applications in Health Care and Biomedicine Third Edition*. New York: Springer.
- Supangat, A. (2007). *Statistika Dalam Kajian Deskriptif, Inferensi dan Nonparametik*. Jakarta: Kencana.
- Wijaya, M. C., & Prijono, A. (2007). *Pengolahan Citra Digital Menggunakan MATLAB*. Bandung: Informatika.

LAMPIRAN