



TUGAS AKHIR -KI141502

**SEGMENTASI CITRA PSORIASIS
MENGUNAKAN W-K-MEANS DAN OPERASI
MORFOLOGI**

**RIFKY MUHAMMAD RIDHO
NRP 5107 100 068**

Dosen Pembimbing I

Nanik Suciati, S.Kom., M.Kom., Dr.Eng.

Dosen Pembimbing II

Anny Yuniarti, S.Kom, M.Comp.Sc.

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2014**



UNDERGRADUATE THESES - KI141502

SEGMENTATION OF PSORIASIS IMAGE USING W-K-MEANS CLUSTERING AND MORPHOLOGICAL OPERATION

**RIFKY MUHAMMAD RIDHO
NRP 5107 100 068**

First Supervisor

Nanik Suciati, S.Kom., M.Kom., Dr.Eng.

Second Supervisor

Anny Yuniarti, S.Kom, M.Comp.Sc.

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2014**

LEMBAR PENGESAHAN

SEGMENTASI CITRA PSORIASIS MENGGUNAKAN W-K-MEANS DAN OPERASI MORFOLOGI

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Komputasi Cerdas dan Visualisasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

RIFKY MUHAMMAD RIDHO

NRP. 5107 100 068

Disetujui oleh Pembimbing Tugas Akhir:

1. Nanik Suciati, S.Kom., M.Kom., Dr. Eng.
NIP. 19650518 199203 1 003
(Pembimbing 1)
2. Anny Yuniarti, S.Kom, M.Comp.Sc.
NIP. 19810622 200501 2 002
(Pembimbing 2)



**SURABAYA
JANUARI, 2015**

SEGMENTASI CITRA PSORIASIS MENGGUNAKAN W-K-MEANS CLUSTERING DAN OPERASI MORFOLOGI

Nama Mahasiswa : Rifky Muhammad Ridho
NRP : 5107 100 068
Jurusan : Teknik Informatika – FTif ITS
Dosen Pembimbing I : Nanik Suciati, S.Kom., M.Kom.,
Dr.Eng.
Dosen Pembimbing II : Anny Yuniarti, S.Kom, M.Comp.Sc.

Abstrak

Psoriasis merupakan sebuah penyakit yang menyerang kulit, biasanya ditandai dengan sisik yang berlapis berwarna keperakan dan penabalan kulit berwarna kemerahan yang disertai dengan rasa gatal. Luka yang ditimbulkan oleh psoriasis ini dapat muncul secara bervariasi mulai berupa sisik lokal yang kecil maupun hingga meliputi seluruh anggota tubuh. Penyebab utama dari psoriasis masih belum diketahui. Penyakit ini umumnya dianggap penyakit genetik, yang dipicu atau dipengaruhi oleh faktor lingkungan. Sejauh ini masih belum ditemukan obat untuk menyembuhkan psoriasis, walaupun berbagai macam perawatan dapat dilakukan untuk mengontrol gejala psoriasis.

Pada Tugas Akhir ini dikembangkan perangkat lunak untuk melakukan segmentasi area dari psoriasis dengan menggunakan algoritma W-k-means clustering yang dipadukan dengan operasi morfologi. Metode ini memiliki tujuh tahap. Yang pertama adalah melakukan preprocessing pada citra. Kemudian, melakukan clustering pada citra psoriasis dengan menggunakan W-k-means clustering. Tahap ketiga adalah mengubah citra keluaran menjadi citra grayscale. Kemudian citra dihaluskan menggunakan median filtering. Tahap kelima adalah mengubah citra menjadi citra biner. Setelah itu dilakukan deteksi tepi dengan

menggunakan metode Sobel. Tahap terakhir adalah melakukan operasi morfologi pada citra.

Manfaat sistem ini adalah untuk mengetahui area dari penyakit psoriasis. Sehingga diharapkan proses penanganan penyakit ini akan menjadi lebih cepat.

Kata kunci: Psoriasis, W-k-means Clustering, Median Filtering, Deteksi Tepi Sobel, Operasi Morfologi.

SEGMENTATION OF PSORIASIS IMAGE USING W-K-MEANS CLUSTERING AND MORPHOLOGICAL OPERATION

Name : Rifky Muhammad Ridho
NRP : 5107 100 068
Department : Informatics – FTIf ITS
Supervisor I : Nanik Suciati, S.Kom., M.Kom., Dr.Eng.
Supervisor II : Anny Yuniarti, S.Kom, M.Comp.Sc.

Abstract

Psoriasis is a skin affecting disease, characterized by silvery layered scales and red skin lesions which usually itch. The skin lesions seen in psoriasis may vary in severity from minor localized patches to complete body coverage. The causes of psoriasis are not fully understood. It is generally considered a genetic disease, thought to be triggered or influenced by environmental factors. No cure is available for psoriasis, but various treatments can help to control the symptoms.

In this project, we proposed a system to identify psoriasis region using W-k-means algorithm with morphological operations. This method is consisted by seven steps. The first step is image preprocessing. Next, clustering. It is done using W-k-means clustering on psoriasis image. Third, the output image is changed into grayscale image. Then, the image is filtered using median filter. The fifth step is changing image into binary image. Next, edge detection is done by using Sobel method. Lastly, the image is processed using morphological operations

The benefits of this system is to find the area affected by this disease. Hopefully, the handling process of this disease can be quicker.

Keywords: *Psoriasis, W-k-means Clustering, Median Filtering, Sobel Edge Detection, Morphological Operations.*

KATA PENGANTAR

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis bisa menyelesaikan Tugas Akhir yang berjudul

“SEHMENTASI CITRA PSORIASIS MENGUNAKAN *W-K-MEANS CLUSTERING* DAN OPERASI MORFOLOGI”

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat berharga bagi penulis, karena dengan pengerjaan Tugas Akhir ini, penulis bisa memperdalam, meningkatkan, serta mengimplementasikan apa yang telah dipelajari penulis selama menempuh perkuliahan di Teknik Informatika ITS.

Terselesaikannya buku Tugas Akhir ini tidak terlepas dari bantuan dan dukungan semua pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih kepada:

1. Allah SWT atas limpahan rahmat-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.
2. Kedua orang tua, kakak serta segenap keluarga besar penulis yang selalu memberikan dukungan jasmani dan rohani, kasih sayang yang tiada terbatas, semangat, perhatian, selalu setia dan sabar dalam menghadapi keluh kesah penulis dari kecil sampai saat mengerjakan Tugas Akhir, serta doa yang luar biasa yang selalu dipanjatkan untuk penulis..
3. Ibu Dr. Eng. Nanik Suciati, S.Kom, M.Kom. selaku dosen pembimbing 1 dan selaku ketua jurusan Teknik Informatika ITS yang telah memberikan banyak nasehat, arahan, bantuan, ide-ide, perhatian, serta dukungan motivasi yang sangat berguna bagi penulis.
4. Ibu Anny Yuniarti, S.Kom, M.Comp.Sc. selaku Dosen Pembimbing 2, yang telah memberikan kepercayaan, motivasi, bimbingan, dukungan, nasehat, perhatian, serta bantuan yang besar yang telah diberikan kepada penulis.

5. Bapak Tohari Ahmad, S.Kom, MIT., Ph.D. selaku dosen wali. Terimakasih atas saran dan bimbingannya selama penulis menjadi mahasiswa Teknik Informatika ITS.
6. Segenap dosen Teknik Informatika ITS yang telah memberikan segala ilmu, pengetahuan, bimbingan, bantuan, dan kemudahan kepada penulis selama menjalani kuliah..
7. Pak Yudi, Pak Sugeng dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama menjalani kuliah di Teknik Informatika ITS.
8. Sahabat-sahabat penulis, Aem, Budi, Najip, Indro, Waktu, Ipang, Jaya, Oji, Amen, Pilot, Pani, Riwe, Elihu, Sasa, Acit atas nasehat, perhatian dan dukungannya untuk penulis sejak maba sampai penulis menyelesaikan Tugas Akhir ini.
9. Teman-teman angkatan 2007 yang sudah menjadi keluarga kedua bagi penulis selama kuliah di Teknik Informatika ITS.
10. Teman-teman kos dan kontrakan, Wildan, Rizwan, Imam, Nurdin, Mudi, Dimas, Mas Imron yang selalu menghibur penulis selama pembuatan Tugas Akhir.
11. Teman-teman ngopi di Giras 58, Samsul, Bekti, Bayu, Jali, Hari, Gogon, Johari, Budi, Solok, Gundul, Husen, Ian, Ciput, Ipang, Gadang yang selalu menemani penulis.
12. Teman-teman futsal, Nganjuk, Danang, Sunyi, Jamu, Boneng, Agung, Mas Fred, Boboi, Oyek, Stu, Ridan, Dito, Kentung, Radit, Cipeng yang selalu menemani hobi penulis.
13. Kakak-kakak dan adik-adik angkatan 2004, 2005, 2006, 2008, 2009, dan 2010 Teknik Informatika ITS.
14. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu disini yang telah membantu terselesaikannya Tugas Akhir ini.

Sebagai manusia biasa, penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan dan memiliki banyak kekurangan. Sehingga dengan segala kerendahan hati, penulis mengharapkan saran dan kritik yang membangun dari pembaca.

Surabaya, 8 Januari 2015

DAFTAR ISI

| | |
|--|------|
| LEMBAR PENGESAHAN | v |
| Abstrak..... | vii |
| Abstract..... | ix |
| KATA PENGANTAR | xi |
| DAFTAR ISI..... | xiii |
| DAFTAR GAMBAR | xvii |
| DAFTAR TABEL..... | xxi |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Permasalahan | 2 |
| 1.3 Batasan Permasalahan | 2 |
| 1.4 Tujuan..... | 3 |
| 1.5 Manfaat..... | 3 |
| 1.6 Metodologi | 3 |
| 1.7 Sistematika Penulisan | 4 |
| BAB II TINJAUAN PUSTAKA | 7 |
| 2.1 Psoriasis..... | 7 |
| 2.2 <i>K-means clustering</i> | 8 |
| 2.2.1 W-k-means clustering | 9 |
| 2.3 Citra | 11 |
| 2.3.1 Citra Biner | 12 |
| 2.3.2 Citra <i>Grayscale</i> | 13 |
| 2.3.3 <i>Noise</i> | 14 |
| 2.3.4 Mereduksi <i>Noise</i> pada Citra..... | 15 |
| 2.3.5 <i>Median filter</i> | 16 |
| 2.4 Deteksi Tepi (<i>Edge Detection</i>) | 17 |
| 2.4.1 Deteksi Tepi Robert | 19 |
| 2.4.2 Deteksi Tepi Prewitt | 20 |
| 2.4.3 Deteksi Tepi Sobel..... | 20 |

| | | |
|---|---|-----------|
| 2.5 | Operasi Morfologi..... | 20 |
| 2.5.1 | <i>Structuring element</i> | 21 |
| 2.5.2 | Dilasi | 22 |
| 2.5.3 | Erosi | 23 |
| 2.5.4 | <i>Opening dan Closing</i> | 24 |
| 2.5.5 | <i>Region Filling</i> | 25 |
| BAB III PERANCANGAN PERANGKAT LUNAK..... | | 27 |
| 3.1 | Perancangan Data..... | 27 |
| 3.1.1 | Data Masukan..... | 27 |
| 3.1.2 | Data Keluaran..... | 27 |
| 3.2 | Perancangan Aplikasi..... | 28 |
| 3.2.1 | <i>Preprocessing</i> | 30 |
| 3.2.2 | <i>W-k-means clustering</i> | 31 |
| 3.2.3 | <i>Gray-level Processing</i> | 34 |
| 3.2.4 | <i>Median Filtering</i> | 34 |
| 3.2.5 | <i>Binary Imaging</i> | 35 |
| 3.2.6 | Deteksi Tepi Sobel | 36 |
| 3.2.7 | Operasi Morfologi | 36 |
| BAB IV IMPLEMENTASI..... | | 39 |
| 4.1 | Lingkungan Implementasi..... | 39 |
| 4.2 | Implementasi Tahap <i>Preprocessing</i> | 39 |
| 4.3 | Implementasi Tahap <i>W-k-means Clustering</i> | 40 |
| 4.4 | Implementasi Tahap <i>Gray-level Processing</i> | 43 |
| 4.5 | Tahap <i>Median Filtering</i> | 43 |
| 4.6 | Implementasi Tahap <i>Binary Imaging</i> | 44 |
| 4.7 | Implementasi Tahap Deteksi Tepi Sobel | 45 |
| 4.8 | Implementasi Tahap Operasi Morfologi | 46 |
| BAB V UJI COBA DAN EVALUASI..... | | 47 |
| 5.1 | Lingkungan Pelaksanaan Uji Coba | 47 |
| 5.2 | Data Uji Coba..... | 47 |
| 5.3 | Skenario Uji Coba..... | 47 |
| 5.3.1 | Uji Coba Fungsionalitas | 48 |
| 5.3.2 | Uji Coba Perbandingan | 58 |
| 5.4 | Hasil Uji Coba..... | 60 |

| | | |
|--|-----------------|-----------|
| 5.5 | Evaluasi | 64 |
| BAB VI KESIMPULAN DAN SARAN | | 67 |
| 6.1 | Kesimpulan..... | 67 |
| 6.2 | Saran..... | 67 |
| DAFTAR PUSTAKA | | 69 |
| LAMPIRAN A..... | | 71 |
| LAMPIRAN B..... | | 77 |
| BIODATA PENULIS..... | | 83 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 4.1 Lingkungan Implementasi Perangkat Lunak | 39 |
| Tabel 5.1 Hasil Uji Coba Dengan <i>Preprocessing</i> dengan $\beta = 15$.. 60 | 60 |
| Tabel 5.2 Hasil Uji Coba Tanpa <i>Preprocessing</i> dengan $\beta = 15$... 61 | 61 |
| Tabel 5.3 Hasil Uji Coba dengan $\beta = 5$ | 62 |
| Tabel 5.4 Hasil Uji Coba dengan $\beta = 25$ | 63 |
| Tabel 5.5 Hasil Uji Coba Menggunakan <i>K-means Clustering</i> | 63 |
| Tabel A.8.1 Data Masukan Citra Psoriasis | 71 |
| Tabel B.9.1 Data Ground Truth | 77 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Aturan koordinat representasi citra digital | 11 |
| Gambar 2.2 Citra biner dan representasi biner dari derajat keabuannya | 12 |
| Gambar 2.3 Informasi pallet pada citra <i>grayscale</i> | 13 |
| Gambar 2.4 Contoh citra dengan kedalaman 128 piksel dan 256 piksel..... | 13 |
| Gambar 2.5 Macam- <i>macam noise</i> (a) <i>Gaussian noise</i> , (b) <i>Speckle noise</i> , dan (c) <i>Salt and pepper noise</i> | 15 |
| Gambar 2.6 <i>Filtering</i> pada domain spasial | 16 |
| Gambar 2.7 Cara kerja <i>median filter</i> | 17 |
| Gambar 2.8 Contoh <i>median filter</i> (a) Citra dengan <i>noise salt and pepper</i> (b) Citra setelah dilakukan <i>median filter 3x3</i> | 17 |
| Gambar 2.9 Contoh deteksi tepi..... | 18 |
| Gambar 2.10 Proses deteksi tepi citra..... | 19 |
| Gambar 2.11 Contoh <i>structuring element</i> | 21 |
| Gambar 2.12 Operasi dilasi (a) Himpunan A (b) <i>Structuring element</i> B (c) Dilasi A oleh B | 22 |
| Gambar 2.13 Dilasi pada citra (a) Contoh teks dengan resolusi rendah (b) <i>Structuring element</i> (c) Citra setelah dilakukan dilasi. 22 | |
| Gambar 2.14 Operasi erosi (a) Himpunan A (b) <i>Structuring element</i> B (c) Erosi A oleh B | 23 |
| Gambar 2.15 Erosi pada citra (a) Citra asli (b) Citra setelah erosi dengan <i>structuring element disk = 5</i> (c) Citra setelah erosi dengan <i>structuring element disk = 10</i> | 23 |
| Gambar 2.16 Operasi <i>opening</i> pada himpunan A oleh <i>structuring element</i> B | 24 |
| Gambar 2.17 Operasi <i>closing</i> pada himpunan A oleh <i>structuring element</i> B | 25 |
| Gambar 2.18 <i>Opening</i> dan <i>closing</i> pada citra (a) Citra asli (b) Hasil <i>opening</i> citra A oleh <i>structuring element disk = 9</i> (c) Hasil <i>closing</i> citra A oleh <i>structuring element disk = 9</i> | 25 |
| Gambar 2.19 Operasi <i>region filling</i> (a) Himpunan A (b) Komplemen dari A (c) <i>Structuring elemen</i> i B (d) Titik awal pada | |

| | |
|---|----|
| <i>boundary</i> (e)-(h) Langkah-langkah pada persamaan 2.12 (i) Hasil final yang merupakan union dari (a) dan (b) | 26 |
| Gambar 2.20 <i>Region filling</i> pada citra biner..... | 26 |
| Gambar 3.1 Diagram alur sistem..... | 29 |
| Gambar 3.2 Diagram alur tahap <i>preprocessing</i> | 30 |
| Gambar 3.3 Diagram alur <i>W-k-means clustering</i> | 31 |
| Gambar 3.4 Diagram alur <i>median filtering</i> | 35 |
| Gambar 4.1 Kode program tahap <i>preprocessing</i> pada enhance.m | 39 |
| Gambar 4.2 Kode program untuk menentukan centroid awal pada kamins.m | 40 |
| Gambar 4.3 Kode program untuk menghiung bobot tiap warna pada kamins.m..... | 41 |
| Gambar 4.4 Kode program untuk proses <i>clustering</i> pada kamins.m | 42 |
| Gambar 4.5 Kode program untuk proses <i>gray-level processing</i> .. | 43 |
| Gambar 4.6 Kode program untuk melakukan <i>median filtering</i> pada medianfilter.m | 44 |
| Gambar 4.7 Kode program untuk mengubah citra menjadi citra biner..... | 44 |
| Gambar 4.8 Kode program untuk deteksi tepi Sobel pada sobeledge.m..... | 45 |
| Gambar 4.9 Kode program untuk melakukan operasi morfologi. 46 | |
| Gambar 5.1 Antar muka hasil tahap <i>W-k-means clustering</i> dengan citra A | 48 |
| Gambar 5.2 Antar muka hasil tahap <i>W-k-means clustering</i> dengan citra B | 49 |
| Gambar 5.3 Antar muka hasil tahap <i>W-k-means clustering</i> dengan citra C | 49 |
| Gambar 5.4 Antar muka hasil tahap <i>gray-level processing</i> dengan citra A | 50 |
| Gambar 5.5 Antar muka hasil tahap <i>gray-level processing</i> dengan citra B | 51 |
| Gambar 5.6 Antar muka hasil tahap <i>gray-level processing</i> dengan citra C | 51 |
| Gambar 5.7 Antar muka hasil tahap <i>median filtering</i> pada citra A52 | |

| | |
|--|----|
| Gambar 5.8 Antar muka hasil tahap <i>median filtering</i> pada citra B | 53 |
| Gambar 5.9 Antar muka hasil tahap <i>median filtering</i> pada citra C | 53 |
| Gambar 5.10 Antar muka hasil tahap <i>binary imaging</i> pada citra A | 54 |
| Gambar 5.11 Antar muka hasil tahap <i>binary imaging</i> pada citra B | 54 |
| Gambar 5.12 Antar muka hasil tahap <i>binary imaging</i> pada citra C | 55 |
| Gambar 5.13 Antar muka hasil deteksi tepi Sobel pada citra A ... | 55 |
| Gambar 5.14 Antar muka hasil deteksi tepi Sobel pada citra B.... | 56 |
| Gambar 5.15 Antar muka hasil deteksi tepi Sobel pada citra C.... | 56 |
| Gambar 5.16 Antar muka hasil operasi morfologi pada citra A ... | 57 |
| Gambar 5.17 Antar muka hasil operasi morfologi pada citra B ... | 57 |
| Gambar 5.18 Antar muka hasil operasi morfologi pada citra C ... | 58 |
| Gambar 5.19 Antar muka hasil uji coba perbandingan pada citra A | 59 |
| Gambar 5.20 Antar muka hasil uji coba perbandingan pada citra B | 59 |
| Gambar 5.21 Antar muka hasil uji coba perbandingan pada citra C | 60 |

DAFTAR PUSTAKA

- [1] Li Hong Juang and Ming Ni Wu, "Psoriasis image identification using k-means clustering with morphological processing," 2011.
- [2] Wikipedia. [Online]. <http://en.wikipedia.org/wiki/Psoriasis>
- [3] M. Emre Celebi, "Improving the performance of k-means for color quantization," 2010.
- [4] Wen Liang Hung, Yen Chang Chang, and E. Stanley Lee, "Weight selection in W-K-means algorithm with an application in color image segmentation," 2011.
- [5] R.C. Gonzalez and R.E. Woods, *Digital Image Processing, second ed.*: Prentice Hall, 2002.
- [6] Bertalya, "Representasi Citra," 2005.
- [7] Apriyana, Delta Sri Maharani, Shinta Puspasari, and Reni Angreni, "Perbandingan Metode Sobel, Metode Prewitt dan Metode Robert untuk Deteksi Tepi Objek pada Aplikasi Pengenalan Bentuk Berbasis Citra Digital".
- [8] Batra Yudha Pratama, "Operasi Morfologi pada Citra Biner".

BIODATA PENULIS



Rifky Muhammad Ridho, lahir di Solo pada tanggal 24 Maret 1990. dan dibesarkan di Jember, merupakan anak kedua dari dua bersaudara. Penulis telah menempuh pendidikan mulai SDN Jember Lor I (1995-2001), SMPN 2 Jember (2001-2004), SMAN 1 Jember (2004-2007), dan terakhir sebagai mahasiswa Teknik Informatika ITS (2007-2015). Selama kuliah, penulis pernah menjabat sebagai ketua Organizing Committee 5th ICTS. Dalam menyelesaikan kuliahnya, penulis mengambil bidang minat Komputer Cerdas dan Visualisasi (KCV). Penulis memiliki hobi menonton film, mendengarkan musik, dan berolahraga, terutama futsal. Penulis pernah mewakili Teknik Informatika pada turnamen futsal Rektor Cup pada tahun 2010. Penulis dapat dihubungi melalui email: rmridho@gmail.com.

BAB I

PENDAHULUAN

Pada bagian ini akan dijelaskan beberapa hal dasar mengenai Tugas Akhir ini meliputi latar belakang, rumusan permasalahan, batasan permasalahan, tujuan, manfaat dari Tugas Akhir, metodologi serta sistematika penulisan Tugas Akhir.

1.1 Latar Belakang

Psoriasis merupakan sejenis penyakit dimana terdapat tanda berupa sisik pada kulit, dan penyakit ini biasanya diturunkan secara genetik, walaupun sumbernya sampai sekarang masih belum diketahui. Luka dari psoriasis ini sendiri dapat terlihat dengan jelas, biasanya berupa sisik-sisik yang berlapis berwarna keperakan, disertai dengan penebalan warna kemerahan dan rasa gatal dan perih. Apabila sisik ini dilepaskan maka akan timbul bintik pendarahan pada kulit di bawahnya [1].

Psoriasis merupakan inflamasi kronis pada kulit yang sering terjadi. Psoriasis sering timbul di kuku, dimulai dari bintik putih pada kuku sampai penebalan kuku, juga mengenai kulit kepala ditandai dengan sisik besar dan penebalan dengan warna kemerahan yang akan melewati batas rambut. Selain itu, penyakit ini sering mengenai siku dan lutut, walaupun dapat juga mengenai wajah, lipatan lutut, siku, genitalia, telapak tangan dan kaki. Sesuai dengan tingkat keparahannya, penyakit ini bisa meluas ke seluruh tubuh yang dapat mengancam jiwa penderitanya.

Pada tugas akhir ini akan difokuskan pada penyakit psoriasis kronis dengan beberapa karakteristik, biasanya berupa plak yang cukup jelas yang terdapat pada permukaan kulit dengan bintik berwarna coklat, putih atau merah apabila plak tersebut sudah mereda. Selanjutnya, mereka akan bertambah besar menjadi ukuran tertentu dan kemudian cenderung untuk menjadi stabil selama berbulan-bulan atau bertahun-tahun. Presentase area

psoriasis ini merupakan poin utama untuk mengevaluasi kondisi penyebaran penyakit. Karena psoriasis merupakan penyakit kronis, maka sangatlah penting untuk mengerti kondisi penderita dan kemudian menentukan penanganan yang tepat. Pada tugas akhir ini, akan dikembangkan sebuah proses pengolahan citra yang akan membagi citra menjadi bagian kulit normal dan abnormal, sehingga area dari penyakit psoriasis dapat diketahui..

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dipaparkan sebagai berikut:

1. Melakukan klasifikasi citra menggunakan *W-k-means clustering*.
2. Menghilangkan noise pada citra dengan menggunakan median filter.
3. Melakukan deteksi tepi dengan menggunakan sobel edge process.
4. Melakukan rekonstruksi morfologi sebagai tahap akhir dalam proses segmentasi area psoriasis.

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Data uji coba yang digunakan adalah 20 buah citra psoriasis berwarna dengan ukuran *height* atau *width* 100 piksel.
2. Algoritma *clustering* yang digunakan adalah W-k-means clustering.
3. Implementasi tugas akhir menggunakan perangkat lunak Matlab..

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah melakukan segmentasi pada citra psoriasis menggunakan *W-k-means clustering* dan operasi morfologi. Penggunaan *W-k-means clustering* adalah untuk membagi citra antara kulit normal dan area yang terkena psoriasis. Penambahan operasi morfologi digunakan untuk meningkatkan kinerja dari *W-k-means clustering* tersebut.

1.5 Manfaat

Manfaat dari Tugas Akhir ini adalah untuk mengetahui area kulit yang terkena psoriasis. Sehingga diharapkan dapat mempermudah penanganan terhadap penyakit tersebut.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan melalui beberapa tahapan. Adapun tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir
Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan Proposal Tugas Akhir. Pada proposal ini, penulis mengajukan gagasan pembuatan sistem untuk melakukan segmentasi terhadap citra psoriasis dengan menggunakan *W-k-means clustering* dan operasi morfologi.
2. Studi Literatur
Studi literatur bertujuan untuk pencarian, pengumpulan, pembelajaran, dan pemahaman informasi dan literatur yang diperlukan untuk pembuatan aplikasi Tugas Akhir. Pada tahap ini akan dipelajari literatur mengenai *W-k-means clustering*, *gray-level processing*, *median filtering*, *binary image processing*, *sobel edge detection* serta operasi morfologi.
3. Implementasi

Implementasi merupakan tahap membangun rancangan aplikasi yang telah dibuat. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan perancangan, sehingga menjadi sebuah aplikasi yang sesuai dengan apa yang direncanakan. Implementasi dilakukan dengan bantuan perangkat lunak MATLAB.

4. Pengujian dan Evaluasi

Pada tahapan ini dilakukan uji coba terhadap aplikasi yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perancangan. Tahapan ini dimaksudkan juga untuk mengevaluasi jalannya aplikasi, mencari masalah timbul, dan melakukan perbaikan untuk lebih menyempurnakan hasil implementasi yang dibuat.

5. Tahap Penyusunan Buku Tugas Akhir

Tahap ini merupakan penyusunan buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan

Buku tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

1. Bab I. Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang masalah, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu rumusan permasalahan, batasan permasalahan, dan sistematika penulisan juga merupakan bagian dari bab ini.

2. Bab II. Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang untuk mendukung pembuatan Tugas Akhir ini.

3. Bab III. Perancangan Perangkat Lunak

Bab ini berisi penjelasan mengenai desain dan perancangan yang digunakan dalam Tugas Akhir, bahan dan peralatan yang digunakan untuk memenuhi Tugas Akhir, serta urutan pelaksanaan percobaan.

4. Bab IV. Implementasi

Bab ini akan dilakukan pembuatan aplikasi yang dibangun dengan MATLAB 7.6.0 sesuai dengan permasalahan dan batasannya yang telah dijabarkan pada bab pertama.

5. Bab VI Uji Coba dan Evaluasi

Bab ini berisi penjelasan mengenai data hasil percobaan atau pengukuran, dan pembahasan mengenai hasil percobaan yang telah dilakukan.

6. Bab VI. Kesimpulan dan Saran

Bab ini berupa hasil penelitian yang menjawab permasalahan atau yang berupa konsep, program, dan karya rancangan. Selain itu, pada bab ini diberikan saran-saran yang berisi hal-hal yang masih dapat dikerjakan dengan lebih baik dan dapat dikembangkan lebih lanjut, atau berisi masalah-masalah yang dialami pada proses pengerjaan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas tentang teori dasar yang menunjang penyusunan Tugas Akhir. Bab ini diawali dengan penjelasan mengenai psoriasis, algoritma W-k-means, citra *grayscale* dan biner, *median filtering*, deteksi tepi dan operasi morfologi.

2.1 Psoriasis

Psoriasis merupakan sebuah penyakit yang menyerang kulit, biasanya ditandai dengan sisik yang berlapis berwarna keperakan dan penabalan kulit berwarna kemerahan yang disertai dengan rasa gatal. Luka yang ditimbulkan oleh psoriasis ini dapat muncul secara bervariasi mulai berupa sisik lokal yang kecil maupun hingga meliputi seluruh anggota tubuh. Penyakit ini mengenai sekitar 2-4 % populasi dunia [2].

Psoriasis memiliki lima tipe utama, yaitu *psoriasis vulgaris (plaque)*, *psoriasis guttate*, *inverse psoriasis*, *pustular psoriasis*, dan *erythrodermic psoriasis*. *Plaque psoriasis* merupakan tipe yang paling sering ditemukan pada manusia. Biasanya berupa sisik-sisik berwarna merah dan putih pada bagian atas kulit. Sel-sel kulit akan dengan cepat terkumpul pada daerah yang terserang ini dan menyebabkan tampilan berwarna putih keperakan pada kulit. *Plaque psoriasis* biasanya menyerang daerah siku atau lutut tetapi dapat menyebar ke berbagai anggota tubuh lainnya, seperti kulit kepala, telapak tangan, telapak kaki dan alat kelamin. Tidak seperti penyakit eksim, psoriasis lebih sering ditemukan pada sisi luar persendian. Peradangan sendi atau lebih dikenal dengan *psoriatic arthritis* menyerang 30% penderita psoriasis.

Penyebab utama dari psoriasis masih belum diketahui. Ini bukan semata penyimpangan kulit dan dapat menyebabkan dampak yang buruk pada organ yang lain. Psoriasis telah dikaitkan dengan meningkatnya resiko penyakit kanker, jantung dan

beberapa penyakit yang menyerang kekebalan tubuh lainnya. Penyakit ini umumnya dianggap penyakit genetik, yang dipicu atau dipengaruhi oleh faktor lingkungan. Psoriasis berkembang ketika sistem kekebalan tubuh salah mengira sebuah sel kulit normal sebagai patogen sehingga memicu sinyal untuk memproduksi sel kulit baru secara berlebihan.

Sejauh ini masih belum ditemukan obat untuk menyembuhkan psoriasis, walaupun berbagai macam perawatan dapat dilakukan untuk mengontrol gejala psoriasis. Pada psoriasis ringan, metode yang banyak digunakan adalah *topical agents*. Sedangkan pada psoriasis menengah menggunakan *phototherapy* dan psoriasis akut menggunakan *systemic agents*.

2.2 K-means clustering

Clustering merupakan sebuah proses untuk membagi sejumlah obyek menjadi beberapa *cluster* dimana masing-masing obyek pada *cluster* yang sama akan memiliki tingkat kesamaan yang lebih tinggi dibandingkan dengan obyek pada cluster yang lain berdasarkan kriteria tertentu. Algoritma k-means merupakan salah satu metode yang paling banyak digunakan dalam pengelompokan data. Tujuan utama dari k-means clustering adalah untuk membagi data set $X = \{X_1, \dots, X_n\}$ ke dalam k cluster sehingga didapatkan fungsi obyektif k-means clustering P seperti Persamaan 2.1

$$P(U, Z) = \sum_{l=1}^k \sum_{i=1}^n u_{il} \sum_{j=1}^m d(x_{ij}, z_{ij}) \quad (2.1)$$

Langkah awal dalam k-means clustering adalah menentukan centroid dan jumlah cluster yang akan digunakan. Kemudian, menghitung jarak antara tiap titik dengan tiap-tiap centroid dan menetapkan titik tersebut pada centroid terdekat. Centroid tersebut kemudian dihitung ulang sebagai nilai rata-rata dari tiap titik yang ditetapkan pada centroid tersebut. Langkah-langkah tersebut dilakukan berulang kali hingga kriteria penghentian tercapai.

Dari sudut pandang clustering, k-means memiliki beberapa keuntungan, antara lain [3] :

1. Secara konseptual, hal ini sederhana, fleksibel dan mudah diimplementasikan.
2. Memiliki kompleksitas waktu yang linier dengan k dan n. Selain itu, banyak teknik percepatan yang tersedia di berbagai literatur.
3. Memiliki kepastian untuk berhenti dengan tingkat konvergensi kuadrat.

Kerugian utama dari *k-means clustering* adalah seringkali fungsi ini berhenti pada *local minimum* dan pemilihan titik awal centroid sangat mempengaruhi hasil output.

2.2.1 *W-k-means clustering*

W-k-means clustering merupakan modifikasi dari *k-means clustering* dengan menambahkan bobot pada variabel berdasarkan kepentingan dari bobot tersebut pada *clustering*. Bobot yang kecil akan mengurangi efek dari perbedaan variabel. Algoritma *W-k-means clustering* akan dijelaskan sebagai berikut.

Terdapat data set $X = \{X_1, \dots, X_n\}$ dimana $X_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$, $i = 1, \dots, n$ merupakan *feature vectors* pada m -dimensional R^m . *K-means clustering* akan membagi data set X menjadi k kelompok dengan $2 \leq k \leq n - 1$. Fungsi $P(U, Z, W)$ dengan bobot $W = \{w_1, \dots, w_m\}$ untuk m variabel dan β merupakan parameter untuk bobot atribut w_j dirumuskan pada Persamaan 2.2 [4].

$$P(U, Z, W) = \sum_{l=1}^k \sum_{i=1}^n u_{il} \sum_{j=1}^m w_j^\beta d(x_{ij}, z_{ij}) \quad (2.2)$$

$$\begin{cases} \sum_{l=1}^k u_{il} = 1, 1 \leq i \leq n \\ u_{il} \in \{0,1\}, 1 \leq i \leq n, 1 \leq l \leq k \\ \sum_{j=1}^m w_j = 1, 0 \leq w_j \leq 1 \end{cases}$$

Dengan menggunakan metode *Lagrange's multipliers*, didapatkan kondisi yang dibutuhkan untuk meminimalkan $P(U,Z,W)$.

$$z_{ij} = \frac{\sum_{i=1}^n u_{il} x_{ij}}{\sum_{i=1}^n u_{il}}, \text{ untuk } 1 \leq l \leq k \text{ dan } 1 \leq j \leq m \quad (2.3)$$

$$\begin{cases} u_{il} = 1, \text{ jika } \sum_{j=1}^m w_j^\beta d(x_{ij}, z_{ij}) \leq \sum_{j=1}^m w_j^\beta d(x_{ij}, z_{tj}) \\ \text{ untuk } 1 \leq t \leq k \\ u_{ij} = 0, \text{ untuk } t \neq 1 \end{cases} \quad (2.4)$$

ketika $\beta > 1$ atau $\beta \leq 0$, $w_j =$

$$\begin{cases} 0, \text{ jika } D_j = 0 \\ \frac{(1/D_j)^{1/(\beta-1)}}{\sum_{t=1}^h (1/D_t)^{1/(\beta-1)}}, \text{ jika } D_j \neq 0 \end{cases} \quad (2.5)$$

$$D_j = \sum_{l=1}^k \sum_{i=1}^n u_{il} d(x_{ij}, z_{ij}) \quad (2.6)$$

Sehingga algoritma W-k-means clustering dapat digambarkan sebagai berikut :

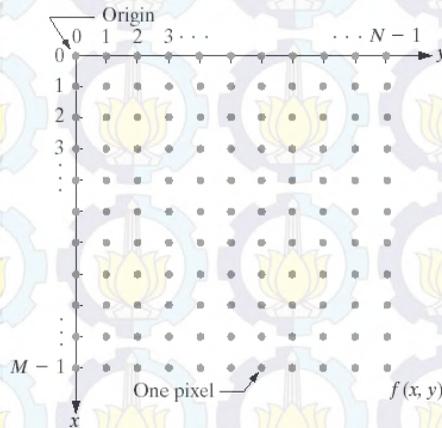
1. Tentukan β dan $2 \leq k \leq n - 1$ serta $r = 0$.
2. Berikan setiap $\epsilon > 0$ dan inisial cluster center $Z_l^{(0)} = (z_{l1}^{(0)}, \dots, z_{lm}^{(0)}) (1 \leq l \leq k)$ dan bobot atribut $W^{(0)} = (w_1^{(0)}, \dots, w_m^{(0)}) (\sum_{j=1}^m w_j^{(0)} = 1)$.
3. Tentukan $u_{il}^{(r)}$ menggunakan Persamaan 3.
4. Tentukan $z_{ij}^{(r)}$ menggunakan Persamaan 2 dengan $u_{il}^{(r)}$.

5. Tentukan $w_j^{(r+1)}$ menggunakan Persamaan 4 dengan $u_i^{(r)}$ dan $w_j^{(r+1)}$. Jika $\max_j \|w_j^{(r+1)} - w_j^{(r)}\| < \epsilon$ maka berhenti. Jika tidak maka $r = r + 1$ dan ulangi ke langkah ke-3.

2.3 Citra

Citra atau *image* merupakan istilah lain dari gambar, yang merupakan informasi berbentuk visual. Suatu citra adalah fungsi intensitas dua dimensi $f(x,y)$ dimana x dan y adalah koordinat spasial dan f pada titik (x,y) merupakan tingkat kecerahan (*brightness*) suatu citra pada suatu titik [5]. Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh obyek.

Citra digital adalah citra $f(x,y)$ yang telah dilakukan digitalisasi baik koordinat area maupun *brightness* level. Nilai f di koordinat (x,y) menunjukkan *brightness* atau *grayness* level dari citra pada titik tersebut. Citra digital adalah representasi dari sebuah citra dua dimensi sebagai sebuah kumpulan nilai digital yang disebut elemen gambar atau piksel. Piksel adalah satuan terkecil dari citra yang mengandung nilai terkuantisasi yang mewakili *brightness* dari sebuah warna pada sebuah titik tertentu,

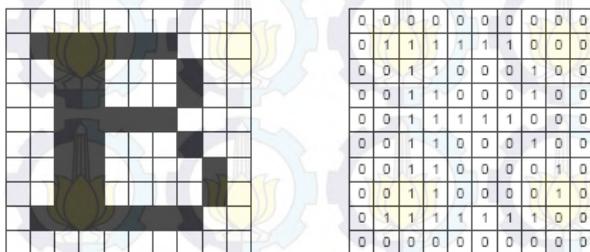


Gambar 2.1 Aturan koordinat representasi citra digital

Berdasarkan warna penyusunnya, citra digital dibagi menjadi tiga macam yaitu citra biner, citra *grayscale* dan citra berwarna (RGB).

2.3.1 Citra Biner

Citra biner (*binary image*) adalah citra yang hanya memiliki dua derajat keabuan, yaitu hitam dan putih. Piksel-piksel pada citra biner hanya memiliki dua nilai, yaitu 0 untuk putih dan 1 untuk hitam.



Gambar 2.2 Citra biner dan representasi biner dari derajat keabuannya

Meskipun komputer saat ini sudah mampu memproses citra *grayscale* maupun citra berwarna, namun citra biner masih dipertahankan keberadaannya. Hal ini dikarenakan citra biner memiliki beberapa keuntungan sebagai berikut :

1. Kebutuhan memori kecil karena nilai derajat keabuan hanya membutuhkan representasi 1 bit.
2. Waktu pemrosesan yang dibutuhkan lebih singkat dibandingkan citra *grayscale* karena banyak operasi pada citra biner yang dilakukan sebagai operasi logika daripada operasi aritmatika bilangan bulat.

Proses peminoran suatu citra dapat dilakukan dengan membulatkan ke atas atau ke bawah untuk setiap nilai keabuan dari piksel yang berada di atas atau di bawah nilai batas ambang. Metode untuk menentukan nilai batas ambang ini disebut *thresholding* [6].

2.3.2 Citra *Grayscale*

Citra *grayscale* merupakan citra yang nilai pikselnya merepresentasikan derajat keabuan atau intensitas warna putih. Nilai intensitas paling rendah merepresentasikan warna hitam, sedangkan nilai intensitas paling tinggi merepresentasikan warna putih.



Gambar 2.3 Informasi pallet pada citra *grayscale*

Banyaknya warna yang ada tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Pada umumnya citra *grayscale* memiliki kedalaman piksel 8-bit (256 derajat keabuan), tetapi ada juga citra *grayscale* yang kedalaman pikselnya bukan 8-bit, misalnya 16-bit. Ini biasanya digunakan untuk penggunaan yang memerlukan ketelitian tinggi.



Gambar 2.4 Contoh citra dengan kedalaman 128 piksel dan 256 piksel

Konversi dari citra RGB menjadi citra *grayscale* dapat dilakukan dengan memberi beban pada setiap elemen warna seperti Persamaan 2.7

$$Y = 0.2989R + 0.5870G + 0.1140B \quad (2.7)$$

Pada Persamaan 2.7, nilai Y merupakan nilai keabuan dari piksel, R merupakan nilai warna merah, G merupakan nilai warna hijau dan B merupakan nilai warna biru.

2.3.3 *Noise*

Pada saat pengambilan gambar ada beberapa gangguan yang mungkin terjadi seperti kamera tidak fokus atau munculnya bintik-bintik yang bisa jadi disebabkan oleh proses pengambilan gambar yang tidak sempurna. Setiap gangguan pada citra tersebut disebut *noise*. *Noise* pada citra tidak hanya terjadi karena ketidaksempurnaan dalam proses pengambilan gambar, tapi bisa juga disebabkan oleh kotoran-kotoran yang terjadi pada citra.

Noise pada umumnya berupa variasi intensitas suatu piksel yang tidak berkorelasi dengan piksel-piksel tetangganya. Secara visual, *noise* mudah dilihat oleh mata karena tampak berbeda dengan piksel tetangganya. berdasarkan bentuk dan karakteristiknya, *noise* pada citra dibedakan menjadi tiga macam, yaitu :

1. *Gaussian noise*
Noise ini merupakan model *noise* yang mengikuti distribusi normal standar dengan rata-rata 0 dan standar deviasi 1. Efek dari *Gaussian noise* pada gambar adalah munculnya titik-titik berwarna yang jumlahnya sama dengan persentase *noise*.
2. *Speckle noise*
Noise ini merupakan model *noise* yang memberikan warna hitam pada titik yang terkena *noise*.
3. *Salt and pepper noise*

Noise ini merupakan bentuk *noise* yang biasanya terlihat titik-titik hitam dan putih pada citra seperti tebaran garam dan merica.



Gambar 2.5 Macam- macam *noise* (a) *Gaussian noise*, (b) *Speckle noise*, dan (c) *Salt and pepper noise*

2.3.4 Mereduksi *Noise* pada Citra

Reduksi *noise* merupakan suatu proses untuk menghilangkan atau mengurangi *noise* dari suatu sinyal. Metode reduksi *noise* secara konsep hampir sama penerapannya pada setiap jenis sinyal, tetapi untuk implementasinya reduksi *noise* tergantung dari jenis sinyal yang akan diproses. Metode *filtering* merupakan salah satu metode untuk mengurangi *noise*.

Filtering adalah suatu cara untuk melakukan ekstraksi bagian data tertentu dari suatu himpunan data dengan menghilangkan bagian-bagian data yang tidak diinginkan. Metode *filtering* pada domain spasial merupakan suatu proses manipulasi kumpulan piksel dari sebuah citra untuk menghasilkan citra baru.

Filtering dilakukan dengan menggunakan *mask/window* yang berukuran 3x3, 5x5 dan seterusnya. *Mask* ini kemudian digeser hingga semua piksel pada citra telah terlewati. Sehingga nilai piksel yang dihasilkan sangat bergantung pada nilai piksel tetangganya.

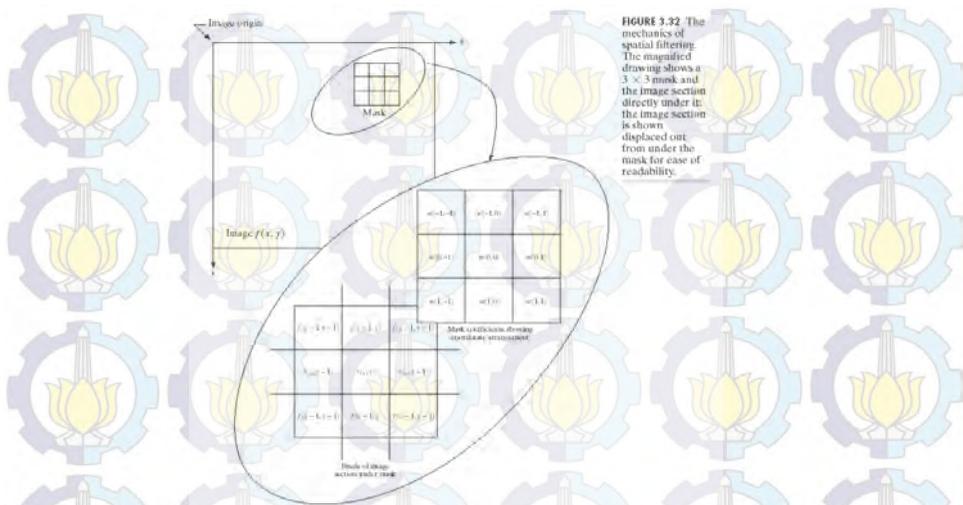


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

Gambar 2.6 Filtering pada domain spasial

2.3.5 Median filter

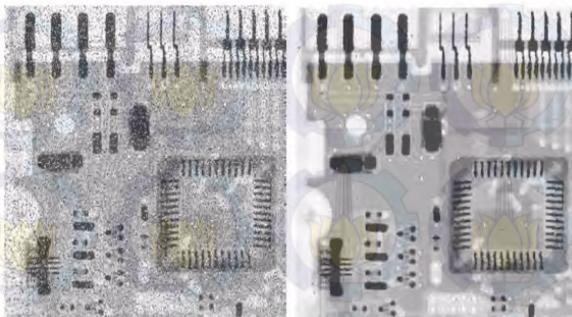
Median filter merupakan *non-linear filter* yang dikembangkan oleh Tukey, yang berfungsi untuk mengurangi *noise* pada citra. *Median filter* termasuk dalam *order-statistic filter*, yang merupakan *filter* spasial yang bekerja berdasarkan urutan piksel-piksel yang ada pada area yang dicakup oleh *filter*.

Pada *median filter*, suatu *window/mask* yang memuat sejumlah piksel ganjil digeser titik per titik pada seluruh daerah citra. Nilai-nilai yang berada pada *window* diurutkan untuk kemudian dihitung nilai mediannya. Nilai tersebut akan menggantikan nilai piksel yang berada pada bagian pusat *window* jika ukuran *window* adalah 3×3 , maka nilai mediannya adalah nilai terbesar kelima.



Gambar 2.7 Cara kerja *median filter*

Salah satu kelebihan dari *median filter* adalah memberikan hasil yang bagus pada citra yang terkena *salt and pepper noise* dimana salah satu piksel dapat memiliki nilai yang ekstrim. Selain itu, *median filter* juga memiliki kemampuan reduksi *noise* yang lebih baik daripada *linear filter* untuk ukuran citra yang sama.



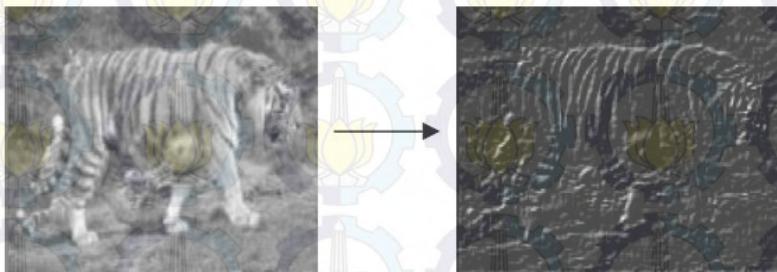
Gambar 2.8 Contoh *median filter* (a) Citra dengan *noise salt and pepper* (b) Citra setelah dilakukan *median filter 3x3*

2.4 Deteksi Tepi (*Edge Detection*)

Deteksi tepi pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra. Tujuan dari deteksi tepi ini antara lain :

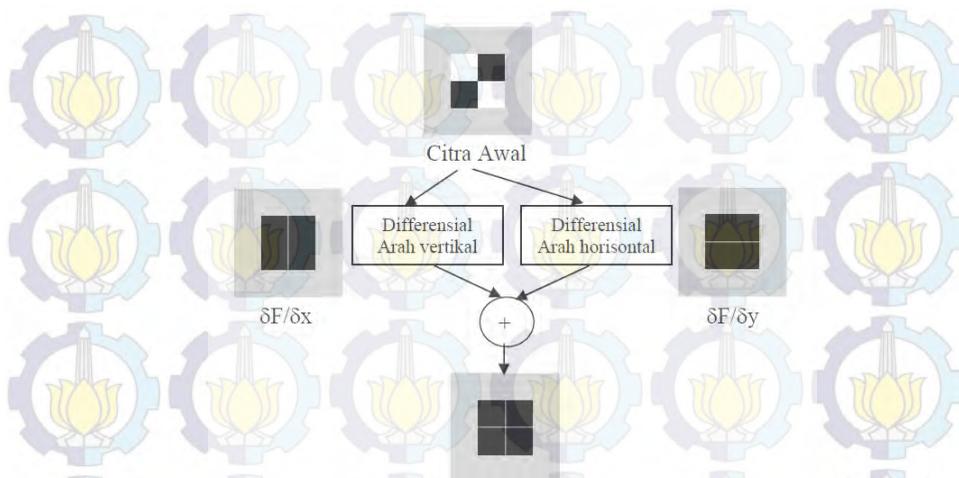
1. Untuk menandai bagian yang menjadi detail citra.
2. Untuk memperbaiki detail dari citra yang kabur. Hal ini biasanya terjadi dikarenakan *error* atau adanya efek dari proses akuisisi citra.

Suatu titik dikatakan sebagai tepi (*edge*) dari sebuah citra apabila titik tersebut memiliki perbedaan yang tinggi dengan tetangganya.



Gambar 2.9 Contoh deteksi tepi

Deteksi tepi sangat penting dalam pengolahan citra karena tepi mencirikan batas-batas objek sehingga tepi berguna untuk proses segmentasi dan identifikasi objek dalam citra. Tujuan utama deteksi tepi, yaitu meningkatkan penampakan garis batas pada suatu citra, memungkinkan untuk mengkombinasikan tingkat kehalusan dan pendeteksian tepi ke dalam suatu konvolusi dalam satu dimensi dengan dua arah yang berbeda (vertikal dan horizontal) [7]. Gambar 2.10 menjelaskan bagaimana tepi suatu gambar dapat diperoleh.



Gambar 2.10 Proses deteksi tepi citra

Ada tiga metode yang paling banyak digunakan untuk proses deteksi tepi, yaitu metode Robert, Prewitt, dan Sobel.

2.4.1 Deteksi Tepi Robert

Metode Robert merupakan nama lain dari teknik diferensial yang dikembangkan pada Gambar 2.10, yaitu diferensial pada arah vertikal dan horizontal dengan ditambahkan proses konversi biner setelah dilakukan diferensial. Teknik konversi biner yang disarankan adalah konversi biner dengan meratakan distribusi warna hitam dan putih. Metode Robert ini juga disamakan dengan teknik DPCM (*Differential Pulse Code Modulation*).

Kernel *filter* yang digunakan dalam metode Robert ini adalah :

$$H = \begin{bmatrix} -1 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

2.4.2 Deteksi Tepi Prewitt

Metode Prewitt merupakan pengembangan dari metode metode Robert dengan menggunakan filter HPF (*High Pass Filter*) yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi Laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF.

Kernel *filter* yang digunakan dalam metode Prewitt ini adalah :

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 & -1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

2.4.3 Deteksi Tepi Sobel

Metode Sobel merupakan pengembangan dari metode metode Robert dengan menggunakan filter HPF (*High Pass Filter*) yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi Laplacian dan Gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode Sobel ini adalah kemampuannya untuk mngurangi noise sebelum melakukan perhitungan deteksi tepi.

Kernel *filter* yang digunakan dalam metode Sobel adalah :

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

2.5 Operasi Morfologi

Morfologi matematika merupakan alat untuk mengekstrak komponen citra yang berguna untuk representasi dan deskripsi bentuk region, seperti *boundaries*, *skeletons*, dan *convex hull*. Morfologi juga digunakan untuk pra-pemrosesan dan paska-pemrosesan, seperti *filtering*, *thinning*, dan *pruning*.

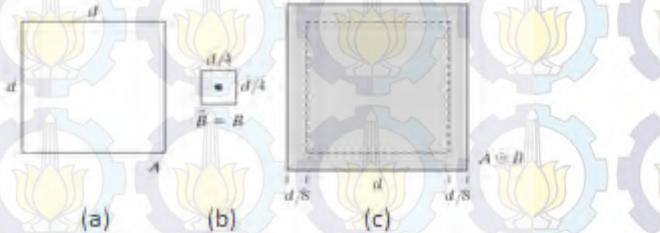
Operasi morfologi adalah teknik pengolahan citra yang didasarkan pada bentuk segmen dan region dalam citra [8]. Operasi morfologi biasa dilakukan pada citra biner. Segmentasi

2.5.2 Dilasi

Operasi dilasi dilakukan untuk memperbesar ukuran segmen objek dengan menambah lapisan di sekeliling objek. Dilasi A dengan B didefinisikan dengan :

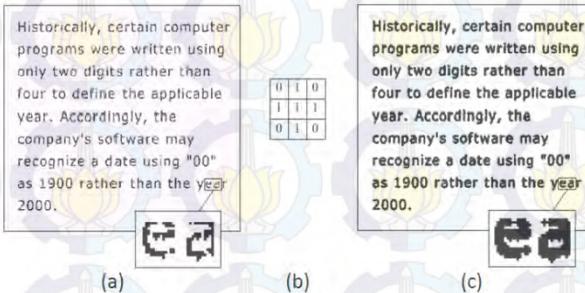
$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \tag{2.8}$$

Terdapat dua cara untuk melakukan operasi ini, yaitu dengan cara mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik objek dan dengan cara mengubah semua titik di sekeliling titik bata menjadi titik objek.



Gambar 2.12 Operasi dilasi (a) Himpunan A (b) Structuring element B (c) Dilasi A oleh B

Dilasi ini sangat berguna ketika diterapkan dalam objek-objek yang terputus dikarenakan hasil pengambilan citra yang terganggu oleh *noise*, kerusakan objek fisik yang dijadikan citra digital, atau disebabkan oleh resolusi yang jelek.



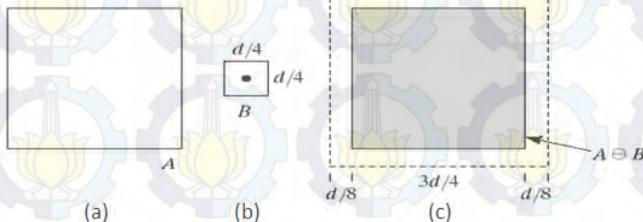
Gambar 2.13 Dilasi pada citra (a) Contoh teks dengan resolusi rendah (b) Structuring element (c) Citra setelah dilakukan dilasi

2.5.3 Erosi

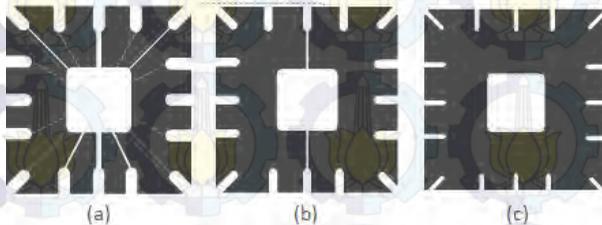
Erosi merupakan kebalikan dari operasi dilasi. Erosi merupakan proses penghapusan titik-titik objek menjadi bagian dari latar berdasarkan *structuring element* yang digunakan. Pada operasi ini ukuran obyek diperkecil dengan mengikis sekeliling objek. Erosi A oleh B didefinisikan dengan :

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \quad (2.9)$$

Erosi dapat dilakukan dengan dua cara, yaitu dengan mengubah semua titik batas menjadi titik latar dan dengan cara mengubah semua titik di sekeliling titik latar menjadi titik latar.



Gambar 2.14 Operasi erosi (a) Himpunan A (b) *Structuring element* B (c) Erosi A oleh B



Gambar 2.15 Erosi pada citra (a) Citra asli (b) Citra setelah erosi dengan *structuring element* $disk = 5$ (c) Citra setelah erosi dengan *structuring element* $disk = 10$

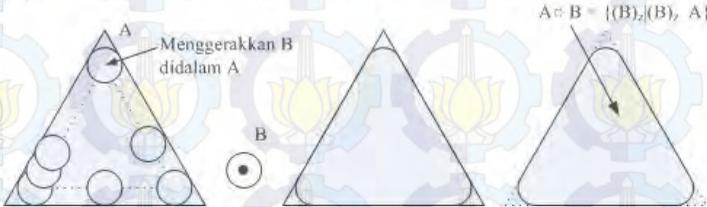
2.5.4 *Opening dan Closing*

Opening dan *closing* merupakan operasi morfologi yang dapat dikategorikan sebagai operasi level kedua dimana *opening* dan *closing* dibangun berdasarkan operasi dilasi dan erosi.

Operasi *opening* merupakan kombinasi antara erosi dan dilasi dimana citra dierosi terlebih dahulu baru kemudian hasilnya didilasi. *Opening* digunakan untuk menghaluskan batas (*contour*) objek, mematahkan jembatan yang sempit, dan menghilangkan tonjolan yang tipis.

Opening dari himpunan A oleh *structuring element* B didefinisikan sebagai berikut :

$$A \circ B = (A \ominus B) \oplus B \quad (2.10)$$

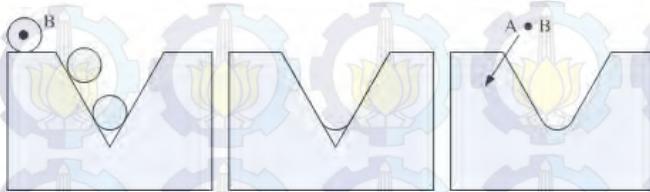


Gambar 2.16 Operasi *opening* pada himpunan A oleh *structuring element* B

Operasi *closing* merupakan kombinasi antara operasi dilasi dan erosi dimana citra didilasi terlebih dahulu baru kemudian dierosi. *Closing* juga digunakan untuk menghaluskan batas objek, namun kebalikan dari *opening*, *closing* menggabungkan jembatan yang sempit dan jurang sempit yang panjang, menghilangkan lubang-lubang yang kecil, dan mengisi celah di dalam batas objek.

Closing dari himpunan A oleh *structuring element* B didefinisikan sebagai berikut :

$$A \bullet B = (A \oplus B) \ominus B \quad (2.11)$$



Gambar 2.17 Operasi *closing* pada himpunan A oleh *structuring element* B



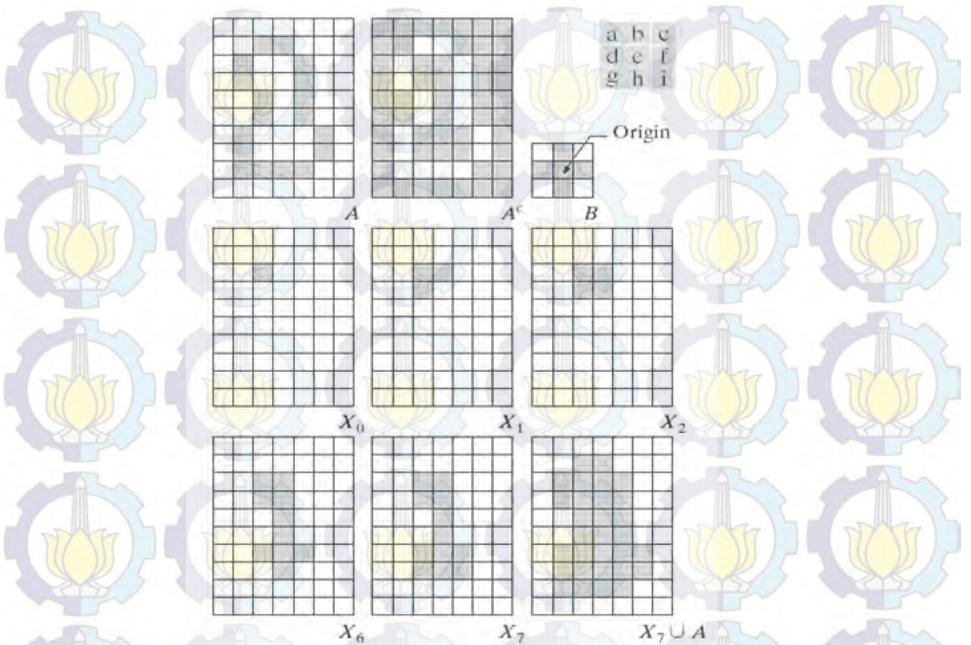
Gambar 2.18 *Opening* dan *closing* pada citra (a) Citra asli (b) Hasil *opening* citra A oleh *structuring element disk = 9* (c) Hasil *closing* citra A oleh *structuring element disk = 9*

2.5.5 Region Filling

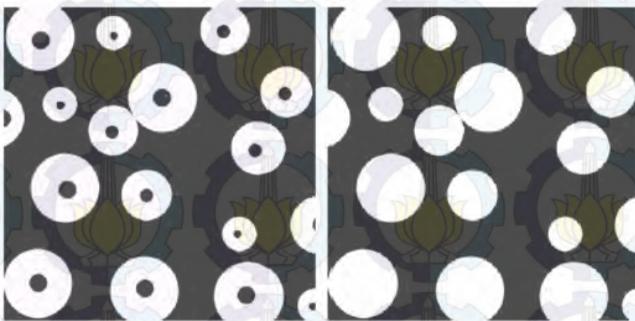
Operasi *region filling* didasarkan pada sejumlah operasi dilasi, komplementasi, dan interseksi. *Region filling* dimulai dari satu titik p di dalam *boundary*, dengan tujuan untuk mengisi semua *region* dengan nilai 1 (satu). Jika mengikuti konvensi bahwa semua titik *non-boundary* (*background*) diberi nilai 0, maka p harus diberi nilai 1 untuk memulai. Persamaan 2.12 digunakan untuk mengisi *region* dengan nilai 1 :

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots \quad (2.12)$$

Pada persamaan 2.12, $X_0 = p$ dan B adalah *symmetric structuring element*. Algoritma akan berhenti pada iterasi ke- k apabila $X_k = X_{k-1}$. Union X_k dan A adalah himpunan isi *region* dan *boundary*-nya.



Gambar 2.19 Operasi *region filling* (a) Himpunan A (b) Komplemen dari A (c) *Structuring element* B (d) Titik awal pada *boundary* (e)-(h) Langkah-langkah pada persamaan 2.12 (i) Hasil final yang merupakan union dari (a) dan (b)



Gambar 2.20 *Region filling* pada citra biner

BAB III

PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan dijelaskan mengenai perancangan sistem perangkat lunak (*software*) agar dapat mencapai tujuan dari tugas akhir. Perancangan akan dibagi menjadi dua bagian yaitu perancangan data dan perancangan aplikasi.

3.1 Perancangan Data

Perancangan data merupakan bagian yang penting untuk diperhatikan dalam pengoperasian perangkat lunak karena dengan data yang tepat maka perangkat lunak dapat beroperasi dengan baik. Data yang diperlukan dalam pengoperasian perangkat lunak yaitu data masukan (*input*), yaitu data proses yang dibutuhkan dan dihasilkan selama proses eksekusi perangkat lunak, dan data keluaran (*output*) yang memberikan hasil proses pengoperasian perangkat lunak untuk pengguna yang menggunakannya.

3.1.1 Data Masukan

Data masukan berupa data yang dimasukkan oleh pengguna perangkat lunak sebagai citra yang akan diproses pada aplikasi. Data masukan berupa citra psoriasis berwarna (RGB). Sebisa mungkin menggunakan citra dengan format 24 bits BMP atau JPG dan memiliki pencahayaan yang baik. Citra masukan berupa citra berwarna dengan ukuran height atau width 100 piksel.

3.1.2 Data Keluaran

Data keluaran yang dihasilkan dari perangkat lunak ini adalah citra daerah yang terkena psoriasis. Data keluaran ini nantinya akan dibandingkan citra *ground truth* untuk mendapatkan akurasi dari metode yang digunakan.

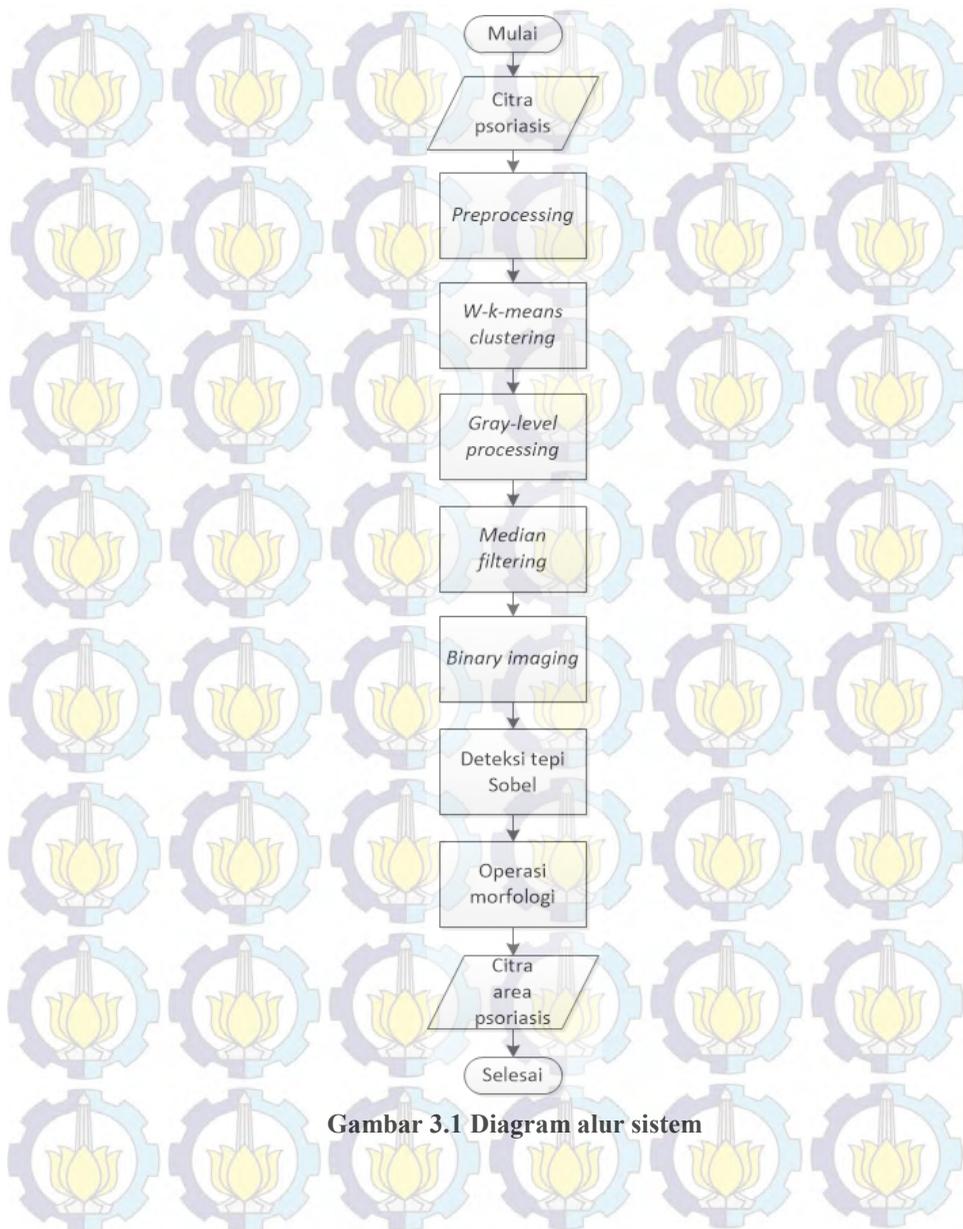
3.2 Perancangan Aplikasi

Pada bagian ini akan dijelaskan tentang alur dalam penerapan algoritma yang akan dipakai dalam tahap implementasi. Proses utama yang diterapkan pada perangkat lunak dibagi menjadi tujuh tahap, yaitu *preprocessing*, *W-k-means clustering*, *gray-level processing*, *median filtering*, *binary imaging*, deteksi tepi Sobel, dan operasi morfologi.

Secara umum, *input* awal dari proses segmentasi citra psoriasis ini adalah citra psoriasis berwarna. *Preprocessing* dilakukan untuk meningkatkan kontras pada citra. Kemudian citra ini dibagi menjadi dua bagian, yaitu daerah yang terkena psoriasis dan daerah yang normal, dengan menggunakan algoritma *W-k-means clustering*. Setelah proses *clustering*, citra akan diubah menjadi citra *grayscale* dengan menggunakan *gray-level processing*.

Citra *grayscale* yang didapat selanjutnya akan diperhalus dengan menggunakan *median filtering*. *Median filtering* ini bertujuan untuk menghilangkan *noise* yang terdapat pada citra. Kemudian citra diubah ke dalam bentuk biner pada proses *binary imaging*.

Setelah menjadi bentuk biner, citra tersebut akan dicari *boundary*-nya dengan menggunakan metode deteksi tepi Sobel. Tahap yang terakhir adalah dilakukan operasi morfologi pada citra untuk mendapatkan region dari daerah yang terkena psoriasis, kemudian citra akan dibandingkan dengan *ground truth*-nya untuk mendapatkan akurasi dari region yang terkena psoriasis. Diagram alur untuk keseluruhan proses sistem ditunjukkan oleh Gambar 3.1.



Gambar 3.1 Diagram alur sistem

3.2.1 *Preprocessing*

Pada tahap pertama ini, citra akan ditingkatkan nilai kontrasnya. Hal ini dilakukan untuk mengoptimalkan proses *W-k-means clustering*. *Preprocessing* dilakukan dengan cara mengubah citra RGB menjadi citra HSV (*Hue, Saturation, Value*). Kemudian nilai saturasi S akan dikalikan dengan sebuah konstanta untuk meningkatkan nilai kontrasnya. Setelah itu citra dikembalikan lagi menjadi citra RGB yang akan dipergunakan pada proses selanjutnya. Diagram alur proses ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram alur tahap *preprocessing*

3.2.2 *W-k-means clustering*

Tahap *clustering* ini dilakukan untuk membagi citra menjadi dua bagian, yaitu daerah yang terkena psoriasis dan daerah yang normal. Algoritma keseluruhan pada proses *W-k-means clustering* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram alur *W-k-means clustering*

Data masukan yang digunakan adalah citra psoriasis berwarna. Pada tahap ini, jumlah k yang digunakan adalah 2 (dua), karena citra akan dibagi menjadi dua bagian saja, yaitu daerah yang terserang psoriasis dan daerah yang normal. Tahap pertama dalam *W-k-means clustering* ini adalah menentukan centroid dari tiap-tiap *cluster*. Hal ini dilakukan dengan menggunakan histogram tiap-tiap warna. Kemudian dicari bobot tiap-tiap warna. Bobot ini akan digunakan pada perhitungan jarak antara tiap piksel dengan masing-masing centroid.

Setelah didapat centroid dan bobot tiap warna, jarak antara tiap-tiap piksel dengan masing-masing centroid dihitung dengan menggunakan rumus *Euclidean distance* yang dikalikan dengan bobot tiap warna. Setelah itu, dicari nilai keanggotaan tiap-tiap piksel terhadap masing-masing centroid. Apabila suatu piksel memiliki jarak yang lebih dekat terhadap centroid pertama, maka piksel tersebut akan dimasukkan ke dalam *cluster* pertama, dan begitu pula sebaliknya.

Setelah semua piksel selesai dicari nilai keanggotaannya, centroid akan dihitung kembali. Kemudian proses *clustering* ini akan dimulai lagi seperti langkah pertama. Proses akan berhenti apabila sudah tidak terjadi pergeseran nilai centroid (centroid lama dan centroid yang baru memiliki nilai yang sama).

3.2.2.1 Menghitung Centroid Awal

Penentuan centroid awal dilakukan dengan menggunakan histogram tiap-tiap warna. Centroid dibagi berdasarkan lokasi titik puncak dari histogram. Hal ini dilakukan pada ketiga warna R , G dan B .

$$c_1 = \frac{\text{nilai titik puncak}}{2}$$

$$c_2 = \frac{255 - \text{nilai titik puncak}}{2} + \text{nilai titik puncak} \quad (3.1)$$

3.2.2.2 Menghitung Bobot tiap Warna

Bobot tiap warna didapatkan dengan menghitung *coefficient of variation* (cv). *Coefficient of variation* sendiri didapatkan dengan membagi standar deviasi dengan nilai rata-rata tiap warna (R,G,B). Secara umum, algoritma untuk menghitung bobot tiap warna adalah sebagai berikut :

1. Menghitung nilai standar deviasi s tiap-tiap warna dengan Persamaan 3.2

$$s = \sqrt{\frac{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}{n - 1}}, \quad (3.2)$$

dimana n merupakan jumlah piksel, x adalah nilai dari piksel, dan \bar{x}_j merupakan nilai rata-rata dari tiap warna.

2. Menentukan *coefficient of variation* cv dengan Persamaan 3.3

$$cv_j = \frac{s}{\bar{x}_j}. \quad (3.3)$$

3. Menghitung nilai bobot tiap warna dengan Persamaan 3.4

$$w_j = \frac{1/cv_j}{\sum_{j=1}^m 1/cv_j}. \quad (3.4)$$

3.2.2.3 Menentukan Jarak antara tiap Piksel dengan Centroid

Penghitungan jarak dilakukan dengan menggunakan rumus *Euclidean distance* yang dikalikan dengan bobot tiap warna yang telah dipangkatkan dengan konstanta β . Sehingga, rumus untuk menghitung jarak akan tampak seperti pada Persamaan 3.5

$$d_{ij} = \sqrt{w_{red}^{\beta} (red_{ij} - z_{red})^2 + w_{blue}^{\beta} (blue_{ij} - z_{blue})^2 + w_{green}^{\beta} (green_{ij} - z_{green})^2}, \quad (3.5)$$

dimana w_{red} merupakan bobot nilai merah, red_{ij} adalah nilai warna merah pada piksel, z_{red} adalah nilai warna merah pada centroid, dan begitu pula seterusnya. Penghitungan jarak ini dilakukan dua kali, yaitu terhadap masing-masing centroid.

3.2.2.4 Menghitung Ulang Centroid

Setelah nilai keanggotaan tiap piksel ditetapkan, dilakukan penghitungan ulang centroid. Penghitungan ulang centroid dilakukan seperti pada Persamaan 3.6

$$z_{lj} = \frac{\sum_{i=1}^n u_{il} x_{ij}}{\sum_{i=1}^n u_{il}}, \quad (3.6)$$

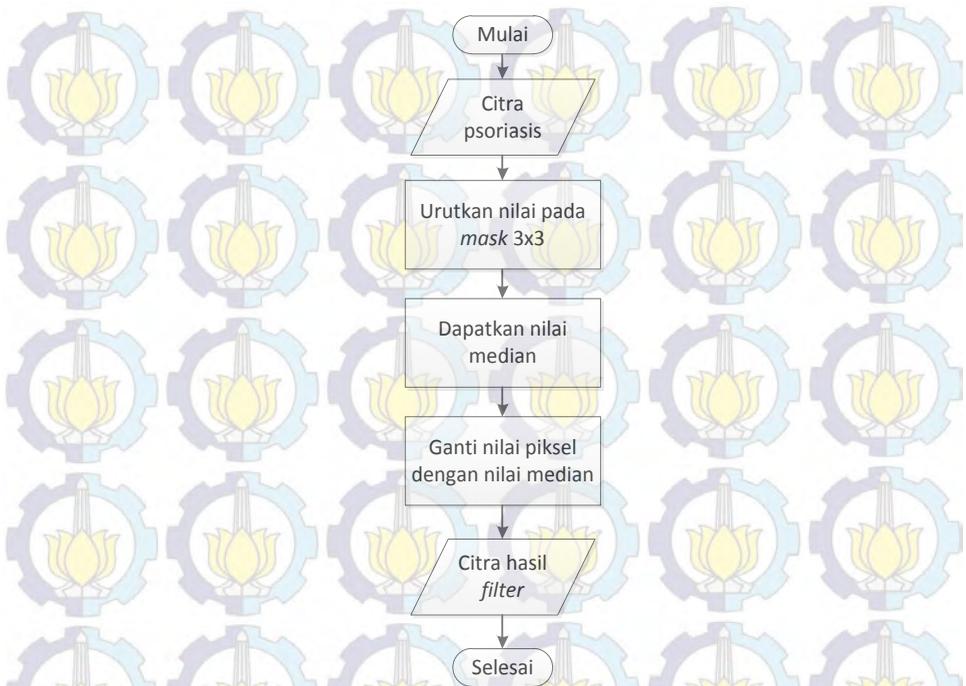
3.2.3 Gray-level Processing

Tahap *gray-level processing* ini bertujuan untuk mengubah citra RGB menjadi citra *grayscale*. Citra RGB diubah menjadi citra *grayscale* dengan cara mengalikan tiap warna dengan koefisien tertentu. Algoritma pada tahap *gray-level processing* ini dapat dilihat pada Persamaan 3.7.

$$Y = 0.2989R + 0.5870G + 0.1140B \quad (3.7)$$

3.2.4 Median Filtering

Setelah dilakukan proses *clustering* dan *gray-level processing*, maka akan tampak *noise* pada citra. Tahap *median filtering* ini berguna untuk menghilangkan *noise* yang terjadi akibat kedua proses tersebut. Algoritma *median filtering* akan dijelaskan dalam diagram alur seperti pada Gambar 3.4.



Gambar 3.4 Digram alur *median filtering*

Median filtering dilakukan dengan menggunakan *mask* 3x3. *Median filtering* diawali dengan mengambil nilai ketetanggaan 3x3 dari piksel, kemudian nilai tersebut diurutkan. Setelah pengurutan tersebut, maka akan didapat nilai median dari *mask*. Nilai pada piksel kemudian akan diganti dengan nilai median tersebut. Diawali pada piksel (1,1), *median filtering* kemudian dilakukan ke seluruh piksel pada citra.

3.2.5 *Binary Imaging*

Tahap *binary imaging* bertujuan untuk mengubah citra menjadi citra biner. Hal ini dikarenakan citra biner akan diperlukan pada pengolahan tahap berikutnya. Pengubahan citra *grayscale* menjadi citra biner dilakukan dengan menggunakan *thresholding*.

Thresholding dilakukan dengan cara menentukan nilai batas ambang (*threshold*) T . Kemudian nilai pada piksel akan dibandingkan dengan nilai *threshold*. Apabila nilai piksel lebih besar daripada nilai *threshold*, maka piksel akan berwarna hitam. Sebaliknya, apabila nilai piksel lebih kecil daripada nilai *threshold*, maka piksel akan berwarna putih. Proses *thresholding* dijelaskan pada Persamaan 3.8.

$$g(x, y) = \begin{cases} 1 & \text{jika } f(x, y) > T \\ 0 & \text{jika } f(x, y) \leq T \end{cases} \quad (3.8)$$

3.2.6 Deteksi Tepi Sobel

Pada tahap ini dilakukan pendeteksian tepi dengan metode Sobel. Deteksi tepi dilakukan untuk menandai tepi daerah yang terkena psoriasis. Deteksi tepi sobel dilakukan dengan menggunakan operator turunan pertama berupa *mask(kernel) filter* berukuran 3x3. Kernel *filter* yang digunakan pada deteksi tepi Sobel adalah :

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } S_y = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.9)$$

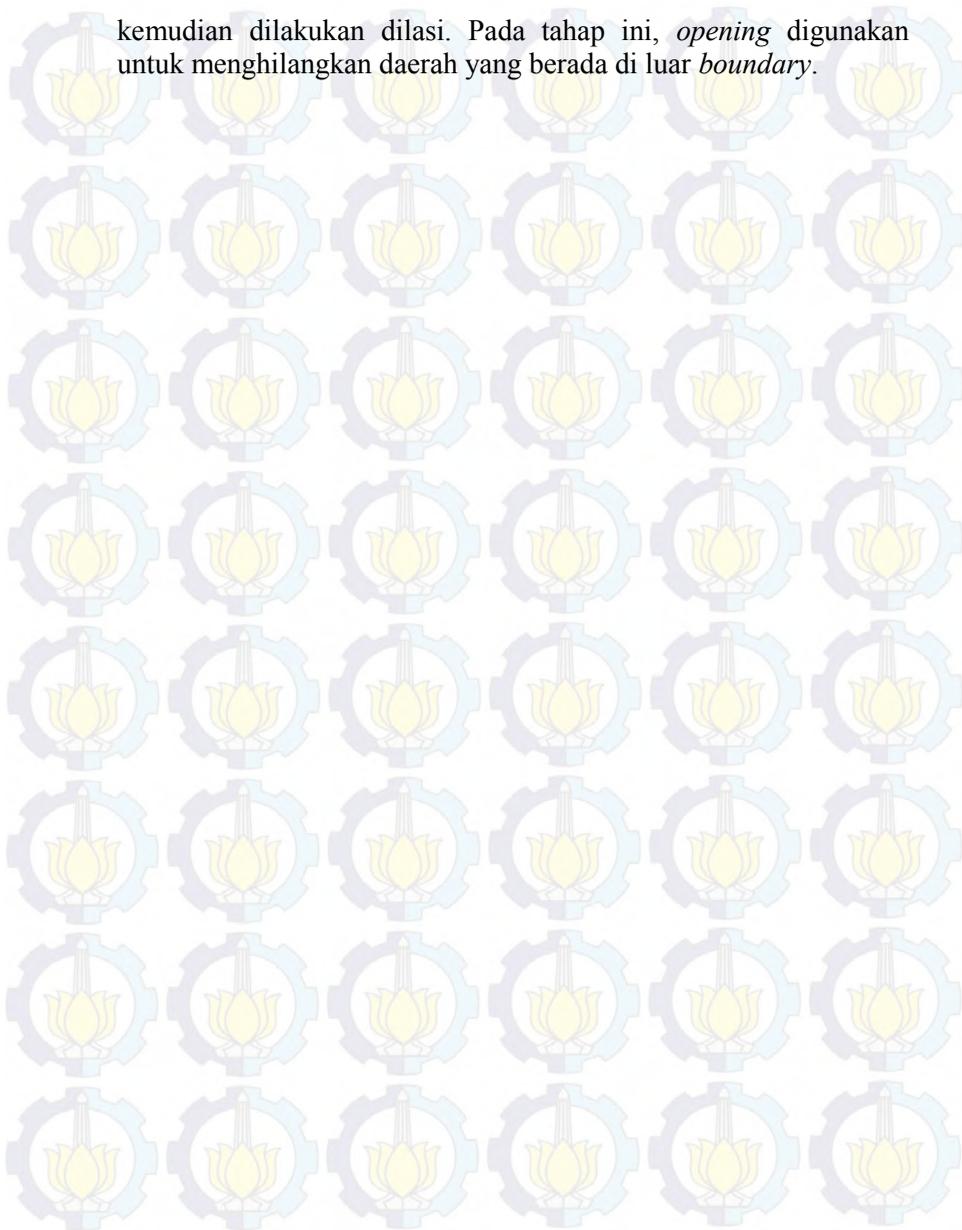
3.2.7 Operasi Morfologi

Setelah tepi dari area yang terkena psoriasis diketahui, tahap selanjutnya yang akan dilakukan adalah melakukan operasi morfologi. Jenis operasi morfologi yang digunakan pada tugas akhir ini adalah *region filling* dan *opening*.

Region filling berguna untuk mengisi area yang terkena psoriasis sehingga daerah yang terkena psoriasis dapat diketahui. *Region filling* merupakan operasi morfologi yang didasarkan pada operasi dilasi, komplementasi dan interseksi. *Region filling* bekerja dengan cara mengisi nilai di dalam *boundary* dengan nilai 1.

Opening merupakan operasi morfologi yang didasarkan pada erosi dan dilasi. Citra akan dierosi terlebih dahulu baru

kemudian dilakukan dilasi. Pada tahap ini, *opening* digunakan untuk menghilangkan daerah yang berada di luar *boundary*.



BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi yang meliputi algoritma dan kode program yang terdapat pada perangkat lunak. Pada tahap implementasi, dari tiap fungsi akan dijelaskan mengenai parameter *input*, *output*, dan beberapa keterangan yang berhubungan dengan program dan teori.

4.1 Lingkungan Implementasi

Implementasi pada proses segmentasi citra psoriasis menggunakan *W-k-means clustering* dan proses morfologi dilakukan dengan menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

| | |
|-----------------|--|
| Perangkat Keras | Prosesor : Intel(R) Core(TM) i3-3120M CPU @2,50GHz |
| | RAM : 2 GB |
| Perangkat Lunak | OS : Windows 8 pro 32-bit |
| | Perangkat Pengembang : MATLAB 7.6.0 (R2008a) |

4.2 Implementasi Tahap *Preprocessing*

Pada tahap pertama ini, citra psoriasis yang merupakan citra RGB akan diubah ke dalam bentuk HSV. Kemudian nilai saturasinya akan ditingkatkan. Data masukan yang digunakan berupa citra psoriasis berwarna.

| | |
|---|--|
| 1 | <code>HSV = rgb2hsv(image);</code> |
| 2 | <code>HSV(:, :, 2) = HSV(:, :, 2) * 10;</code> |
| 3 | <code>HSV(HSV > 1) = 1; % Limit values</code> |
| 4 | <code>RGB = hsv2rgb(HSV);</code> |
| 5 | <code>RGB = round(RGB * 256);</code> |

Gambar 4.1 Kode program tahap *preprocessing* pada *enhance.m*

4.3 Implementasi Tahap *W-k-means Clustering*

Pada tahap ini akan dilakukan proses *clustering* dengan menggunakan algoritma *W-k-means clustering*. Data masukan pada tahap ini adalah sebuah citra psoriasis yang telah ditingkatkan kontrasnya. Citra akan diolah menjadi dua bagian, yaitu bagian yang terserang psoriasis dan bagian yang normal. Proses awal pada *W-k-means clustering* adalah penentuan centroid.

| | |
|----|---|
| 1 | <code>% menentukan centroid awal</code> |
| 2 | <code>for i = 1:256</code> |
| 3 | <code> if histR(i) == max(histR)</code> |
| 4 | <code> Z(1,1) = floor(i/2);</code> |
| 5 | <code> Z(2,1) = i + floor((256-i)/2);</code> |
| 6 | <code> elseif histG(i) == max(histG)</code> |
| 7 | <code> Z(1,2) = floor(i/2);</code> |
| 8 | <code> Z(2,2) = i + floor((256-i)/2);</code> |
| 9 | <code> elseif histB(i) == max(histB)</code> |
| 10 | <code> Z(1,3) = floor(i/2);</code> |
| 11 | <code> Z(2,3) = i + floor((256-i)/2);</code> |
| 12 | <code> end</code> |
| 13 | <code>end</code> |

Gambar 4.2 Kode program untuk menentukan centroid awal pada `kamins.m`

Setelah didapatkan centroid awal, maka bobot tiap warna akan dihitung.

| | |
|----|---|
| 1 | <code>% menentukan bobot tiap warna</code> |
| 2 | <code>for i=1:kali</code> |
| 3 | <code> r1(i) = r(i) - meanC(1);</code> |
| 4 | <code> g1(i) = g(i) - meanC(2);</code> |
| 5 | <code> b1(i) = b(i) - meanC(3);</code> |
| 6 | <code>end</code> |
| 7 | <code>s(1) = sqrt(sum(r1)/(kali-1));</code> |
| 8 | <code>s(2) = sqrt(sum(g1)/(kali-1));</code> |
| 9 | <code>s(3) = sqrt(sum(b1)/(kali-1));</code> |
| 10 | <code>for j=1:C</code> |
| 11 | <code> cv(j) = s(j)/mean(j);</code> |

| | |
|----|---|
| 12 | <code>cv_total = cv_total + (1/cv(j));</code> |
| 13 | <code>end</code> |
| 14 | <code>for k=1:C</code> |
| 15 | <code>w(k) = (1/cv(k))/cv_total;</code> |
| 16 | <code>end</code> |

Gambar 4.3 Kode program untuk menghiung bobot tiap warna pada kamins.m

Setelah kedua proses tersebut, maka proses iterasi untuk membagi citra ke dalam *cluster* dimulai. Proses akan berhenti apabila centroid tidak lagi bergeser (centroid awal dan centroid baru sama).

| | |
|----|--|
| 1 | <code>while 1</code> |
| 2 | <code> for i=1:kali</code> |
| 3 | <code> for j=1:C</code> |
| 4 | <code> % menghitung jarak antara</code> <code> piksel dengan centroid</code> |
| 5 | <code> d(i,j) =</code> <code> sqrt((w(1)^beta)*((r(1,i)-Z(j,1))^2) +</code> <code> (w(2)^beta)*((g(1,i)-Z(j,2))^2) +</code> <code> (w(3)^beta)*((b(1,i)-Z(j,3))^2));</code> |
| 6 | <code> % menentukan nilai keanggotaan</code> |
| 7 | <code> if d(i,1) > d(i,2)</code> |
| 8 | <code> U(i,1) = 1;</code> |
| 9 | <code> U(i,2) = 0;</code> |
| 10 | <code> else</code> |
| 11 | <code> U(i,1) = 0;</code> |
| 12 | <code> U(i,2) = 1;</code> |
| 13 | <code> end</code> |
| 14 | <code> end</code> |
| 15 | <code> end</code> |
| 16 | <code> atas = zeros(C,o);</code> |
| 17 | <code> total bawah = zeros(1,C);</code> |
| 18 | <code> Z_temp = zeros(C,o);</code> |
| 19 | |
| 20 | <code> % menghitung nilai centroid baru</code> |
| 21 | <code> for i=1:C</code> |
| 22 | <code> for j=1:kali</code> |

| | |
|----|--|
| 23 | <code>atas(i,1) = atas(i,1) + (r(1,j)*U(j,i));</code> |
| 24 | <code>atas(i,2) = atas(i,2) + (g(1,j)*U(j,i));</code> |
| 25 | <code>atas(i,3) = atas(i,3) + (b(1,j)*U(j,i));</code> |
| 26 | <code>total_bawah(1,i) = total_bawah(1,i) + U(j,i);</code> |
| 27 | <code>end</code> |
| 28 | <code>for k=1:o</code> |
| 29 | <code> Z_temp(i,k) = atas(i,k) / total_bawah(1,i);</code> |
| 30 | <code>end</code> |
| 31 | <code>end</code> |
| 32 | |
| 33 | <code>for i=1:C</code> |
| 34 | <code> for j=1:o</code> |
| 3 | <code> Z_selisih = abs(Z(i,j)- Z_temp(i,j));</code> |
| 5 | <code> end</code> |
| 36 | <code>end</code> |
| 37 | |
| 38 | <code>if Z_selisih == Z_selisih2</code> |
| 39 | <code> break;</code> |
| 40 | <code>else</code> |
| 41 | <code> for i=1:C</code> |
| 42 | <code> for j=1:o</code> |
| 43 | <code> Z(i,j) = Z_temp(i,j);</code> |
| 44 | <code> Z_selisih2 = Z_selisih;</code> |
| 45 | <code> end</code> |
| 46 | <code> end</code> |
| 47 | <code>end</code> |
| 48 | <code>end</code> |

Gambar 4.4 Kode program untuk proses *clustering* pada *kamins.m*

Data keluaran dari tahap W-k-means clustering ini adalah citra berwarna yang telah ter-*cluster*.

4.4 Implementasi Tahap *Gray-level Processing*

Gray-level processing merupakan tahap yang dilakukan setelah tahap *clustering*. Pada tahap ini, citra akan diubah menjadi citra *grayscale*. Proses ini dilakukan dengan menggunakan Persamaan 3.6 seperti yang telah dibahas pada subbab 3.2.2.

| | |
|---|--|
| 1 | <code>% proses mengubah citra menjadi citra grayscale</code> |
| 2 | <code>gray=uint8(zeros(size(clustering,1),size(clustering,2)));</code> |
| 3 | <code>for i=1:size(clustering,1)</code> |
| 4 | <code> for j=1:size(clustering,2)</code> |
| 5 | <code> gray(i,j)=0.2989*clustering(i,j,1)+0.5870*clustering(i,j,2)+0.1140*clustering(i,j,3);</code> |
| 6 | <code> end</code> |
| 7 | <code>end</code> |

Gambar 4.5 Kode program untuk proses *gray-level processing*

4.5 Tahap *Median Filtering*

Pada tahap ini, citra *grayscale* yang menjadi data keluaran dari *gray-level processing* akan dihilangkan *noise*-nya. Proses penghilangan *noise* dilakukan dengan menggunakan *median filter* dengan mask 3x3.

| | |
|----|--|
| 1 | <code>% menyalin matriks asli ke dalam matriks baru</code> |
| 2 | <code>for x=1:size(A,1)</code> |
| 3 | <code> for y=1:size(A,2)</code> |
| 4 | <code> modifyA(x+1,y+1)=A(x,y);</code> |
| 5 | <code> end</code> |
| 6 | <code>end</code> |
| 7 | |
| 8 | <code>for i= 1:size(modifyA,1)-2</code> |
| 9 | <code> for j=1:size(modifyA,2)-2</code> |
| 10 | <code> window=zeros(9,1);</code> |
| 11 | <code> inc=1;</code> |

| | |
|----|---|
| 12 | <code>% memasukkan nilai ketetanggaan ke dalam mask</code> |
| 13 | <code>for x=1:3</code> |
| 14 | <code>for y=1:3</code> |
| 15 | <code> window(inc)=modifyA(i+x-</code> <code>1,j+y-1);</code> |
| 16 | <code> inc=inc+1;</code> |
| 17 | <code> end</code> |
| 18 | <code>end</code> |
| 19 | <code>% mengurutkan nilai di dalam mask</code> |
| 20 | <code>med=sort(window);</code> |
| 21 | <code>% mengganti nilai piksel dengan nilai median</code> |
| 22 | <code>B(i,j)=med(5);</code> |
| 23 | |
| 24 | <code>end</code> |
| 25 | <code>end</code> |

Gambar 4.6 Kode program untuk melakukan *median filtering* pada `medianfilter.m`

Data keluaran dari tahap *median filtering* berupa citra *grayscale* yang telah diperhalus.

4.6 Implementasi Tahap *Binary Imaging*

Tahap *binary imaging* ini digunakan untuk mengubah citra *grayscale* menjadi citra biner. *Binary imaging* dilakukan karena citra biner akan diperlukan pada pemrosesan tahap selanjutnya.

| | |
|---|--|
| 1 | <code>% mengubah citra grayscale ke biner</code> |
| 2 | <code>level = graythresh(median);</code> |
| 3 | <code>biner = im2bw(median,level);</code> |

Gambar 4.7 Kode program untuk mengubah citra menjadi citra biner

Pada implementasi tahap *binary imaging* ini, digunakan fungsi matlab, yaitu `graythresh` dan `im2bw`. `Graythresh` berfungsi untuk mendapatkan nilai *threshold*, sedangkan `im2bw` digunakan

untuk konversi citra *grayscale* ke citra biner dengan menggunakan nilai *threshold* yang didapatkan pada fungsi *graythresh*.

4.7 Implementasi Tahap Deteksi Tepi Sobel

Pada tahap ini akan dilakukan proses deteksi tepi dengan menggunakan metode Sobel. Data masukan yang digunakan berupa citra biner yang telah didapatkan pada tahap sebelumnya. Proses deteksi tepi Sobel dilakukan dengan menggunakan kernel filter berukuran 3x3. Kernel filter yang digunakan berupa matriks seperti pada Persamaan 3.8 yang telah dibahas pada subbab 3.2.5.

| | |
|----|---|
| 1 | <code>for i=1:size(C,1)-2</code> |
| 2 | <code> for j=1:size(C,2)-2</code> |
| 3 | <code> %Sobel mask untuk arah x</code> |
| 4 | <code>Gx=((2*C(i+2,j+1)+C(i+2,j)+C(i+2,j+2))-</code> <code>(2*C(i,j+1)+C(i,j)+C(i,j+2)));</code> |
| 5 | <code> %Sobel mask untuk arah y</code> |
| 6 | <code>Gy=((2*C(i+1,j+2)+C(i,j+2)+C(i+2,j+2))-</code> <code>(2*C(i+1,j)+C(i,j)+C(i+2,j)));</code> |
| 7 | |
| 8 | <code> %nilai gradien dari citra</code> |
| 9 | <code> B(i,j)=sqrt(Gx.^2+Gy.^2);</code> |
| 10 | |
| 11 | <code> end</code> |
| 12 | <code>end</code> |
| 13 | |
| 14 | <code>Thresh=0;</code> |
| 15 | <code>B=max(B,Thresh);</code> |
| 16 | <code>B(B==round(Thresh))=0;</code> |

Gambar 4.8 Kode program untuk deteksi tepi Sobel pada `sobeledge.m`

Data keluaran yang dihasilkan pada tahap ini berupa citra biner berisi tepian dari area yang terserang psoriasis.

4.8 Implementasi Tahap Operasi Morfologi

Tahap terakhir yang dilakukan pada proses adalah operasi morfologi. Teknik operasi morfologi yang digunakan adalah *region filling* dan *opening*. *Region filling* digunakan untuk mengisi daerah yang terkena psoriasis, sedangkan *opening* digunakan untuk menghilangkan titik-titik di luar *boundary* yang terjadi karena operasi *region filling*.

| | |
|---|---------------------------------------|
| 1 | <code>% operasi morfologi</code> |
| 2 | <code>se = strel('disk', 3);</code> |
| 3 | <code>Be = imfill(B, 'holes');</code> |
| 4 | <code>final = imopen(Be, se);</code> |

Gambar 4.9 Kode program untuk melakukan operasi morfologi

Pada implementasi tahap operasi morfologi ini digunakan fungsi operasi morfologi yang telah disediakan oleh matlab, yaitu *imfill* dan *imopen*. *Imfill* digunakan melakukan *region filling*, sedangkan *imopen* digunakan untuk melakukan *opening* pada citra. Variabel *se* pada Gambar 4.8 merupakan *structuring element* yang digunakan pada tahap ini. *Structuring element* yang digunakan berupa *disk* dengan ukuran 3x3.

BAB V

UJI COBA DAN EVALUASI

Bab ini merupakan bahasan mengenai hasil uji coba dan evaluasi yang dilakukan pada proses segmentasi citra psoriasis menggunakan *W-k-means clustering* dan proses morfologi. Pembahasan pada bab ini meliputi lingkungan uji coba, data uji coba, skenario uji coba, hasil uji coba, dan evaluasi.

5.1 Lingkungan Pelaksanaan Uji Coba

Lingkungan pelaksanaan uji coba yang digunakan dalam pembuatan Tugas Akhir ini meliputi perangkat lunak dan perangkat keras yang digunakan pada proses pembuatan aplikasi.

Seperti yang telah dibahas dalam tabel 4.1 pada subbab 4.1, sistem ini dibangun dengan menggunakan komputer dengan prosesor Intel(R) Core(TM) i3-3120M CPU @2,50GHz dan memori sebesar 2 GB. Sistem operasi yang digunakan adalah Windows 8 Pro 32-bit dengan perangkat pengembang yang digunakan adalah Matlab 7.6.0 (R2008a), Microsoft Visio 2010 dan Microsoft Word 2010.

5.2 Data Uji Coba

Data masukan yang digunakan pada uji coba ini adalah 20 buah citra psoriasis. Data ini berupa citra berwarna dengan *height* atau *width* berukuran 100 piksel. Semua data pada uji coba ini akan dilampirkan pada Lampiran A.

5.3 Skenario Uji Coba

Pada bagian ini akan dijelaskan mengenai skenario uji coba yang akan dilakukan untuk mengetahui kesesuaian dan kinerja aplikasi pada proses segmentasi citra psoriasis. Terdapat dua skenario dalam pelaksanaan uji coba, yaitu :

1. Uji coba fungsionalitas.
2. Uji coba perbandingan.

5.3.1 Uji Coba Fungsionalitas

Uji coba ini dilakukan untuk mengetahui kesesuaian antara proses yang berjalan pada program dengan proses yang diharapkan. Terdapat tujuh tahap pada proses segmentasi citra psoriasis menggunakan *W-k-means clustering* dan proses morfologi seperti yang telah dijelaskan oleh Gambar 3.1 pada subbab 3.2.

5.3.1.1 *W-k-means Clustering*

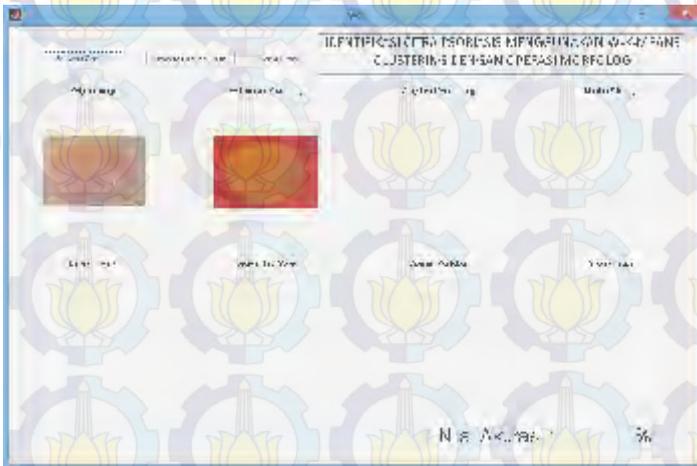
Data masukan untuk tahap ini berupa citra psoriasis berwarna. Setelah dilakukan proses, maka citra keluaran dari tahap ini adalah citra berwarna yang telah ter-*cluster*. Contoh antar muka pada proses ini dapat dilihat pada Gambar 5.1-5.3.



Gambar 5.1 Antar muka hasil tahap *W-k-means clustering* dengan citra A



Gambar 5.2 Antar muka hasil tahap *W-k-means clustering* dengan citra B



Gambar 5.3 Antar muka hasil tahap *W-k-means clustering* dengan citra C

Warna yang digunakan pada tiap cluster didapat dari nilai RGB centroid terakhir. Hasil keluaran dari tahap ini akan digunakan sebagai data masukan pada tahap *gray-level processing*.

5.3.1.2 *Gray-level Processing*

Pada tahap ini, citra yang telah ter-*cluster* kemudian diubah menjadi citra *grayscale*. Data keluaran akan digunakan pada pemrosesan selanjutnya, yaitu *median filtering*. Contoh antar muka pada tahap ini dapat dilihat pada Gambar 5.4-5.6.



Gambar 5.4 Antar muka hasil tahap *gray-level processing* dengan citra A



Gambar 5.5 Antar muka hasil tahap *gray-level processing* dengan citra B



Gambar 5.6 Antar muka hasil tahap *gray-level processing* dengan citra C

5.3.1.3 Median Filtering

Tahap selanjutnya adalah melakukan *median filtering*. Pada tahap ini, citra *grayscale* yang telah didapat akan dihilangkan *noise*-nya dengan cara melakukan *filtering*. Metode yang digunakan adalah *median filtering*. Data keluaran yang dihasilkan pada tahap ini adalah citra *grayscale* yang telah dihaluskan. Contoh antar muka tahap *median filtering* dapat dilihat pada Gambar 5.7-5.9.

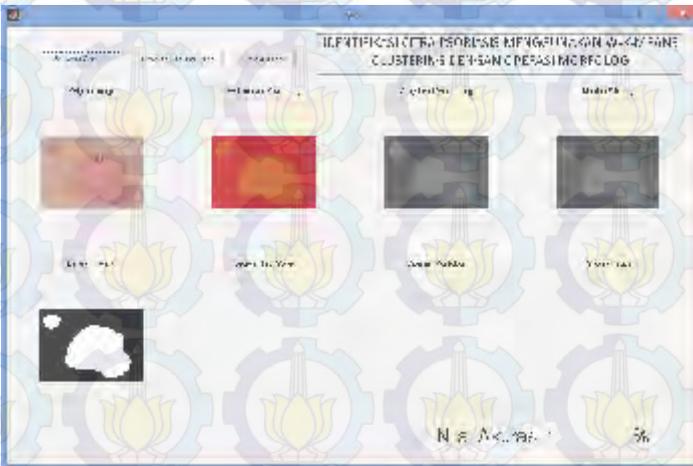


Gambar 5.7 Antar muka hasil tahap *median filtering* pada citra A

biner sebagai hasil keluarannya. Contoh antar muka hasil pada tahap *binary imaging* ini dapat dilihat pada Gambar 5.10-5.12.



Gambar 5.10 Antar muka hasil tahap *binary imaging* pada citra A



Gambar 5.11 Antar muka hasil tahap *binary imaging* pada citra B



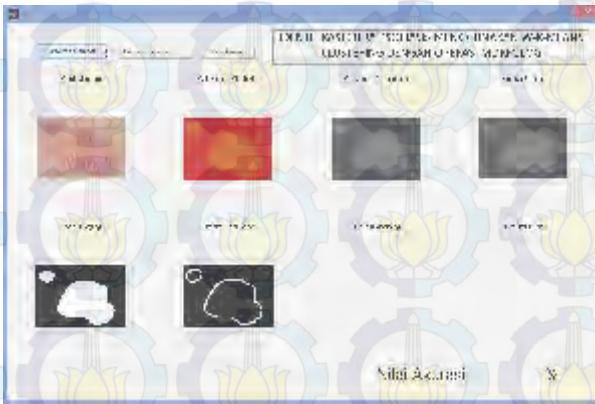
Gambar 5.12 Antar muka hasil tahap *binary imaging* pada citra C

5.3.1.5 Deteksi Tepi Sobel

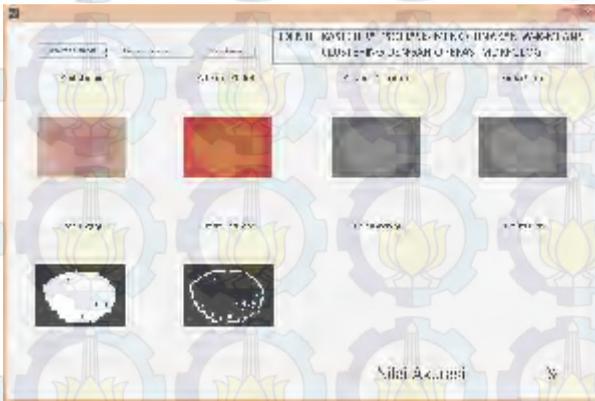
Tahap selanjutnya adalah melakukan deteksi tepi pada citra dengan menggunakan metode Sobel. Data masukan yang digunakan adalah citra biner. Deteksi tepi dilakukan untuk menandai daerah yang terkena psoriasis. Contoh antar muka hasil deteksi tepi dapat dilihat pada Gambar 5.13-5.15.



Gambar 5.13 Antar muka hasil deteksi tepi Sobel pada citra A



Gambar 5.14 Antar muka hasil deteksi tepi Sobel pada citra B



Gambar 5.15 Antar muka hasil deteksi tepi Sobel pada citra C

Data keluaran dari tahap deteksi tepi Sobel adalah citra biner yang telah terdeteksi tepinya,

5.3.1.6 Operasi Morfologi

Tahap terakhir yang dilakukan pada uji coba fungsionalitas adalah operasi morfologi. Data masukan yang digunakan pada operasi morfologi adalah citra tepian biner. Operasi yang dilakukan pertama adalah *region filling*, yaitu melakukan

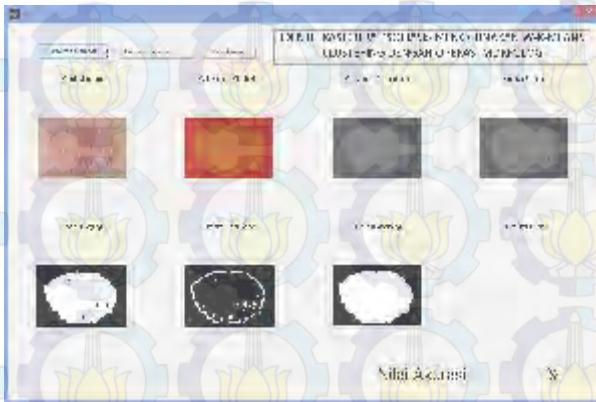
pengisian pada tepian citra. Sehingga akan didapat *region* yang terkena psoriasis. Operasi yang kedua adalah melakukan *opening* untuk menghilangkan titik-titik yang tercipta akibat operasi *region filling*. Contoh antar muka hasil operasi morfologi dapat dilihat pada Gambar 5.16-5.18.



Gambar 5.16 Antar muka hasil operasi morfologi pada citra A



Gambar 5.17 Antar muka hasil operasi morfologi pada citra B



Gambar 5.18 Antar muka hasil operasi morfologi pada citra C

Data keluaran yang dihasilkan pada tahap operasi morfologi adalah citra area hasil segmentasi. Citra ini akan dibandingkan dengan citra *ground truth* untuk mendapatkan akurasi dari teknik yang diajukan.

5.3.2 Uji Coba Perbandingan

Pada uji coba ini, citra hasil segmentasi akan dibandingkan dengan citra *ground truth* sehingga didapatkan nilai akurasinya. Citra *ground truth* berupa citra *hand-drawing region* yang didapatkan dengan cara melakukan segmentasi secara manual. Citra *ground truth* yang digunakan pada uji coba ini akan dilampirkan pada Lampiran B.

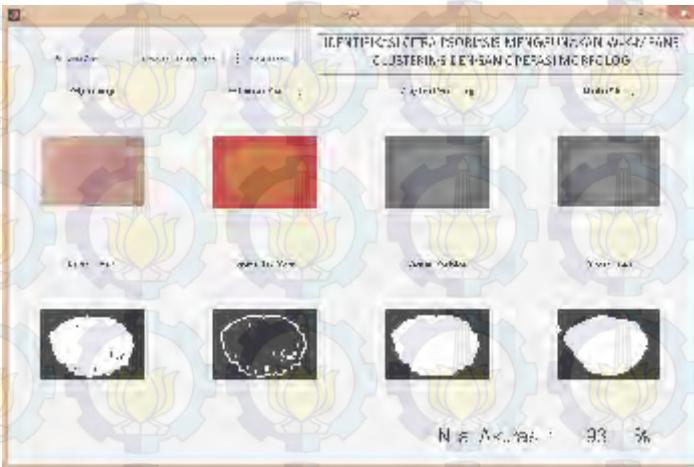
Nilai akurasi dari teknik yang diajukan didapatkan dengan menghitung kesamaan nilai per piksel antara citra hasil segmentasi dengan citra *ground truth*. Contoh antar muka hasil uji coba perbandingan dapat dilihat pada Gambar 5.19-5.21.



Gambar 5.19 Antar muka hasil uji coba perbandingan pada citra A



Gambar 5.20 Antar muka hasil uji coba perbandingan pada citra B



Gambar 5.21 Antar muka hasil uji coba perbandingan pada citra C

Selain itu, akan dibandingkan pula nilai akurasi sistem yang menggunakan preprocessing dengan sistem tanpa preprocessing. hal ini dilakukan untuk mengetahui seberapa besar pengaruh preprocessing pada sistem.

5.4 Hasil Uji Coba

Dari uji coba perbandingan yang dilakukan pada seluruh data masukan, maka akan didapatkan nilai akurasi dari seluruh citra. Hasil dari uji coba tersebut dipaparkan dalam Tabel 5.1-5.5.

Tabel 5.1 Hasil Uji Coba Dengan *Preprocessing* dengan $\beta = 15$

| Citra Psoriasis | Citra Ground Truth | Akurasi |
|-------------------|---------------------|---------|
| Psoriasis (1).jpg | groundtruth (1).jpg | 95% |
| Psoriasis (2).jpg | groundtruth (2).jpg | 79% |
| Psoriasis (3).jpg | groundtruth (3).jpg | 98% |
| Psoriasis (4).jpg | groundtruth (4).jpg | 80% |
| Psoriasis (5).jpg | groundtruth (5).jpg | 81% |
| Psoriasis (6).jpg | groundtruth (6).jpg | 96% |

| | | |
|-------------------------|----------------------|-----|
| Psoriasis (7).jpg | groundtruth (7).jpg | 98% |
| Psoriasis (8).jpg | groundtruth (8).jpg | 86% |
| Psoriasis (9).jpg | groundtruth (9).jpg | 77% |
| Psoriasis (10).jpg | groundtruth (10).jpg | 86% |
| Psoriasis (11).jpg | groundtruth (11).jpg | 77% |
| Psoriasis (12).jpg | groundtruth (12).jpg | 95% |
| Psoriasis (13).jpg | groundtruth (13).jpg | 93% |
| Psoriasis (14).jpg | groundtruth (14).jpg | 85% |
| Psoriasis (15).jpg | groundtruth (15).jpg | 81% |
| Psoriasis (16).jpg | groundtruth (16).jpg | 77% |
| Psoriasis (17).jpg | groundtruth (17).jpg | 94% |
| Psoriasis (18).jpg | groundtruth (18).jpg | 84% |
| Psoriasis (19).jpg | groundtruth (19).jpg | 73% |
| Psoriasis (20).jpg | groundtruth (20).jpg | 98% |
| Nilai Akurasi Rata-Rata | | 87% |

Tabel 5.2 Hasil Uji Coba Tanpa *Preprocessing* dengan $\beta = 15$

| Citra Psoriasis | Citra Ground Truth | Akurasi |
|------------------------|---------------------------|----------------|
| Psoriasis (1).jpg | groundtruth (1).jpg | 86% |
| Psoriasis (2).jpg | groundtruth (2).jpg | 58% |
| Psoriasis (3).jpg | groundtruth (3).jpg | 76% |
| Psoriasis (4).jpg | groundtruth (4).jpg | 48% |
| Psoriasis (5).jpg | groundtruth (5).jpg | 80% |
| Psoriasis (6).jpg | groundtruth (6).jpg | 96% |
| Psoriasis (7).jpg | groundtruth (7).jpg | 95% |
| Psoriasis (8).jpg | groundtruth (8).jpg | 73% |
| Psoriasis (9).jpg | groundtruth (9).jpg | 61% |
| Psoriasis (10).jpg | groundtruth (10).jpg | 85% |
| Psoriasis (11).jpg | groundtruth (11).jpg | 58% |
| Psoriasis (12).jpg | groundtruth (12).jpg | 62% |
| Psoriasis (13).jpg | groundtruth (13).jpg | 93% |
| Psoriasis (14).jpg | groundtruth (14).jpg | 54% |
| Psoriasis (15).jpg | groundtruth (15).jpg | 56% |

| | | |
|-------------------------|----------------------|-----|
| Psoriasis (16).jpg | groundtruth (16).jpg | 81% |
| Psoriasis (17).jpg | groundtruth (17).jpg | 66% |
| Psoriasis (18).jpg | groundtruth (18).jpg | 89% |
| Psoriasis (19).jpg | groundtruth (19).jpg | 92% |
| Psoriasis (20).jpg | groundtruth (20).jpg | 97% |
| Nilai Akurasi Rata-Rata | | 75% |

Tabel 5.3 Hasil Uji Coba dengan $\beta = 5$

| Citra Psoriasis | Citra Ground Truth | Akurasi |
|-------------------------|---------------------------|----------------|
| Psoriasis (1).jpg | groundtruth (1).jpg | 94% |
| Psoriasis (2).jpg | groundtruth (2).jpg | 79% |
| Psoriasis (3).jpg | groundtruth (3).jpg | 59% |
| Psoriasis (4).jpg | groundtruth (4).jpg | 79% |
| Psoriasis (5).jpg | groundtruth (5).jpg | 81% |
| Psoriasis (6).jpg | groundtruth (6).jpg | 96% |
| Psoriasis (7).jpg | groundtruth (7).jpg | 98% |
| Psoriasis (8).jpg | groundtruth (8).jpg | 86% |
| Psoriasis (9).jpg | groundtruth (9).jpg | 77% |
| Psoriasis (10).jpg | groundtruth (10).jpg | 84% |
| Psoriasis (11).jpg | groundtruth (11).jpg | 76% |
| Psoriasis (12).jpg | groundtruth (12).jpg | 95% |
| Psoriasis (13).jpg | groundtruth (13).jpg | 93% |
| Psoriasis (14).jpg | groundtruth (14).jpg | 77% |
| Psoriasis (15).jpg | groundtruth (15).jpg | 54% |
| Psoriasis (16).jpg | groundtruth (16).jpg | 77% |
| Psoriasis (17).jpg | groundtruth (17).jpg | 94% |
| Psoriasis (18).jpg | groundtruth (18).jpg | 84% |
| Psoriasis (19).jpg | groundtruth (19).jpg | 73% |
| Psoriasis (20).jpg | groundtruth (20).jpg | 72% |
| Nilai Akurasi Rata-Rata | | 81% |

Tabel 5.4 Hasil Uji Coba dengan $\beta = 25$

| Citra Psoriasis | Citra Ground Truth | Akurasi |
|-------------------------|----------------------|---------|
| Psoriasis (1).jpg | groundtruth (1).jpg | 95% |
| Psoriasis (2).jpg | groundtruth (2).jpg | 79% |
| Psoriasis (3).jpg | groundtruth (3).jpg | 98% |
| Psoriasis (4).jpg | groundtruth (4).jpg | 84% |
| Psoriasis (5).jpg | groundtruth (5).jpg | 28% |
| Psoriasis (6).jpg | groundtruth (6).jpg | 96% |
| Psoriasis (7).jpg | groundtruth (7).jpg | 98% |
| Psoriasis (8).jpg | groundtruth (8).jpg | 86% |
| Psoriasis (9).jpg | groundtruth (9).jpg | 77% |
| Psoriasis (10).jpg | groundtruth (10).jpg | 86% |
| Psoriasis (11).jpg | groundtruth (11).jpg | 77% |
| Psoriasis (12).jpg | groundtruth (12).jpg | 96% |
| Psoriasis (13).jpg | groundtruth (13).jpg | 50% |
| Psoriasis (14).jpg | groundtruth (14).jpg | 87% |
| Psoriasis (15).jpg | groundtruth (15).jpg | 89% |
| Psoriasis (16).jpg | groundtruth (16).jpg | 77% |
| Psoriasis (17).jpg | groundtruth (17).jpg | 94% |
| Psoriasis (18).jpg | groundtruth (18).jpg | 84% |
| Psoriasis (19).jpg | groundtruth (19).jpg | 73% |
| Psoriasis (20).jpg | groundtruth (20).jpg | 98% |
| Nilai Akurasi Rata-Rata | | 83% |

Tabel 5.5 Hasil Uji Coba Menggunakan *K-means Clustering*

| Citra Psoriasis | Citra Ground Truth | Akurasi |
|-------------------|---------------------|---------|
| Psoriasis (1).jpg | groundtruth (1).jpg | 74% |
| Psoriasis (2).jpg | groundtruth (2).jpg | 52% |
| Psoriasis (3).jpg | groundtruth (3).jpg | 74% |
| Psoriasis (4).jpg | groundtruth (4).jpg | 52% |
| Psoriasis (5).jpg | groundtruth (5).jpg | 74% |

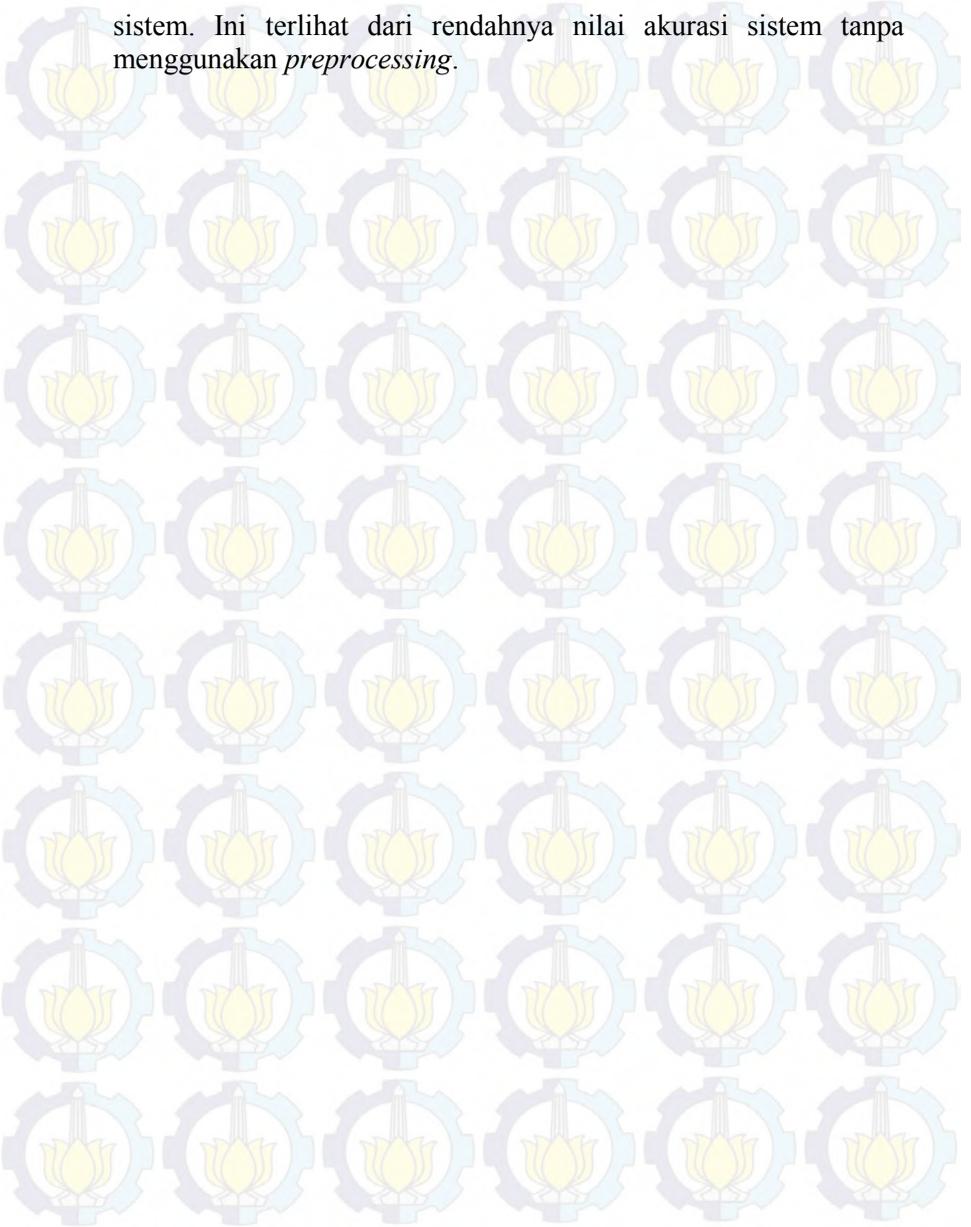
| | | |
|-------------------------|----------------------|-----|
| Psoriasis (6).jpg | groundtruth (6).jpg | 95% |
| Psoriasis (7).jpg | groundtruth (7).jpg | 78% |
| Psoriasis (8).jpg | groundtruth (8).jpg | 62% |
| Psoriasis (9).jpg | groundtruth (9).jpg | 42% |
| Psoriasis (10).jpg | groundtruth (10).jpg | 72% |
| Psoriasis (11).jpg | groundtruth (11).jpg | 51% |
| Psoriasis (12).jpg | groundtruth (12).jpg | 65% |
| Psoriasis (13).jpg | groundtruth (13).jpg | 80% |
| Psoriasis (14).jpg | groundtruth (14).jpg | 60% |
| Psoriasis (15).jpg | groundtruth (15).jpg | 58% |
| Psoriasis (16).jpg | groundtruth (16).jpg | 79% |
| Psoriasis (17).jpg | groundtruth (17).jpg | 59% |
| Psoriasis (18).jpg | groundtruth (18).jpg | 87% |
| Psoriasis (19).jpg | groundtruth (19).jpg | 87% |
| Psoriasis (20).jpg | groundtruth (20).jpg | 97% |
| Nilai Akurasi Rata-Rata | | 70% |

5.5 Evaluasi

Dari uji coba fungsionalitas yang telah dilakukan, terlihat bahwa tahap pertama dalam segmentasi citra psoriasis, yaitu *W-k-means clustering*, sangat berpengaruh pada proses segmentasi. Penentuan konstanta β yang tepat pada *W-k-means clustering* sangat penting karena konstanta β dapat memiliki efek yang berbeda-beda pada tiap data masukan. Ini terlihat pada citra Psoriasis (14) dan (15) di mana penggunaan $\beta = 25$ memiliki nilai akurasi yang lebih tinggi.

Dari uji coba perbandingan, terlihat bahwa citra masukan yang mendapatkan pencahayaan yang baik seperti pada citra Psoriasis(6), (7), dan (20) akan memiliki nilai akurasi yang lebih tinggi. Sedangkan citra masukan yang mendapatkan pencahayaan yang tidak rata akan memiliki nilai akurasi yang rendah seperti terlihat pada citra Psoriasis(9), (11), (15), dan (16). Selain itu, terlihat bahwa tahap *preprocessing* memiliki peran vital pada

sistem. Ini terlihat dari rendahnya nilai akurasi sistem tanpa menggunakan *preprocessing*.



BAB VI KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1 Kesimpulan

Dari hasil uji coba yang telah dilakukan, terdapat beberapa kesimpulan yang dapat diambil, yaitu :

1. Data masukan dengan pencahayaan yang lebih rata akan memiliki nilai akurasi yang lebih tinggi.
2. Penggunaan *preprocessing* sangat membantu pada proses segmentasi citra, dengan perbedaan nilai akurasi sebesar 12%.
3. Konstanta β yang memiliki nilai akurasi tertinggi pada sistem adalah 15.
4. Nilai akurasi rata-rata yang didapatkan oleh sistem ini sebesar 87%.

6.2 Saran

Saran yang hendak disampaikan penulis terkait dengan pengerjaan Tugas Akhir ini adalah sebaiknya data masukan yang digunakan dalam proses segmentasi citra psoriasis adalah citra yang memiliki tingkat pencahayaan yang rata. Karena citra dengan tingkat pencahayaan yang tidak rata akan memiliki nilai akurasi yang rendah atau tidak dapat melakukan proses segmentasi dengan baik. Selain itu, metode *preprocessing* yang digunakan tidak harus terbatas pada peningkatan kontras. Dapat pula digunakan metode *preprocessing* yang lain.

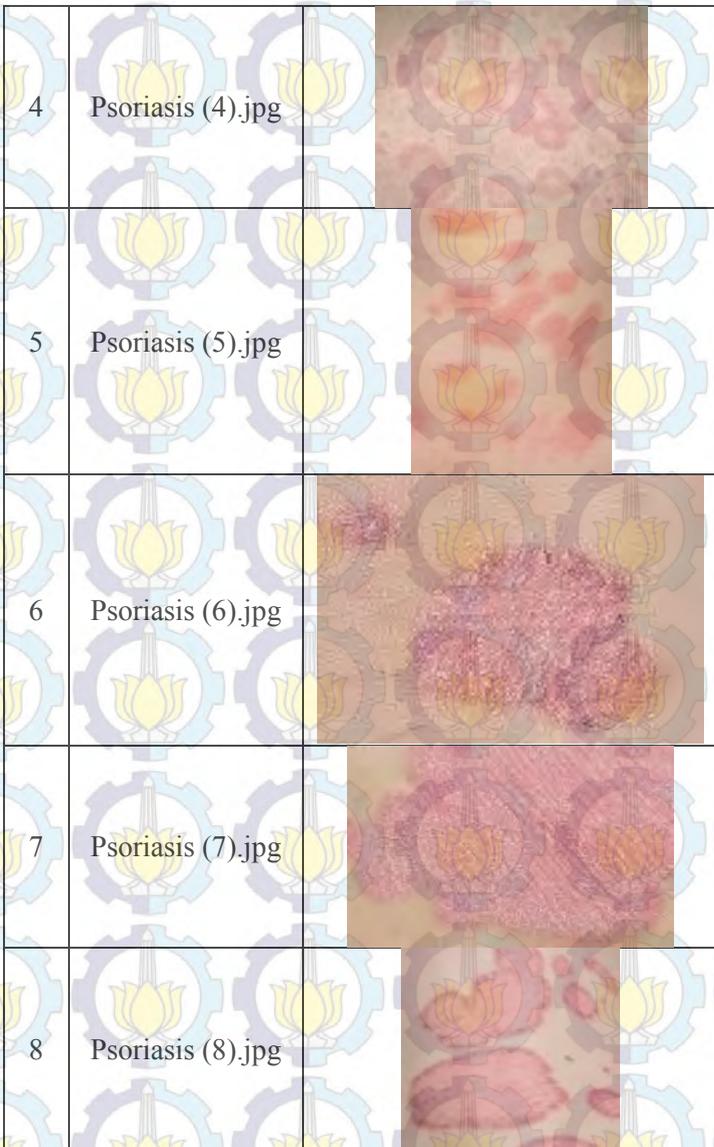
LAMPIRAN A

DATA MASUKAN CITRA PSORIASIS

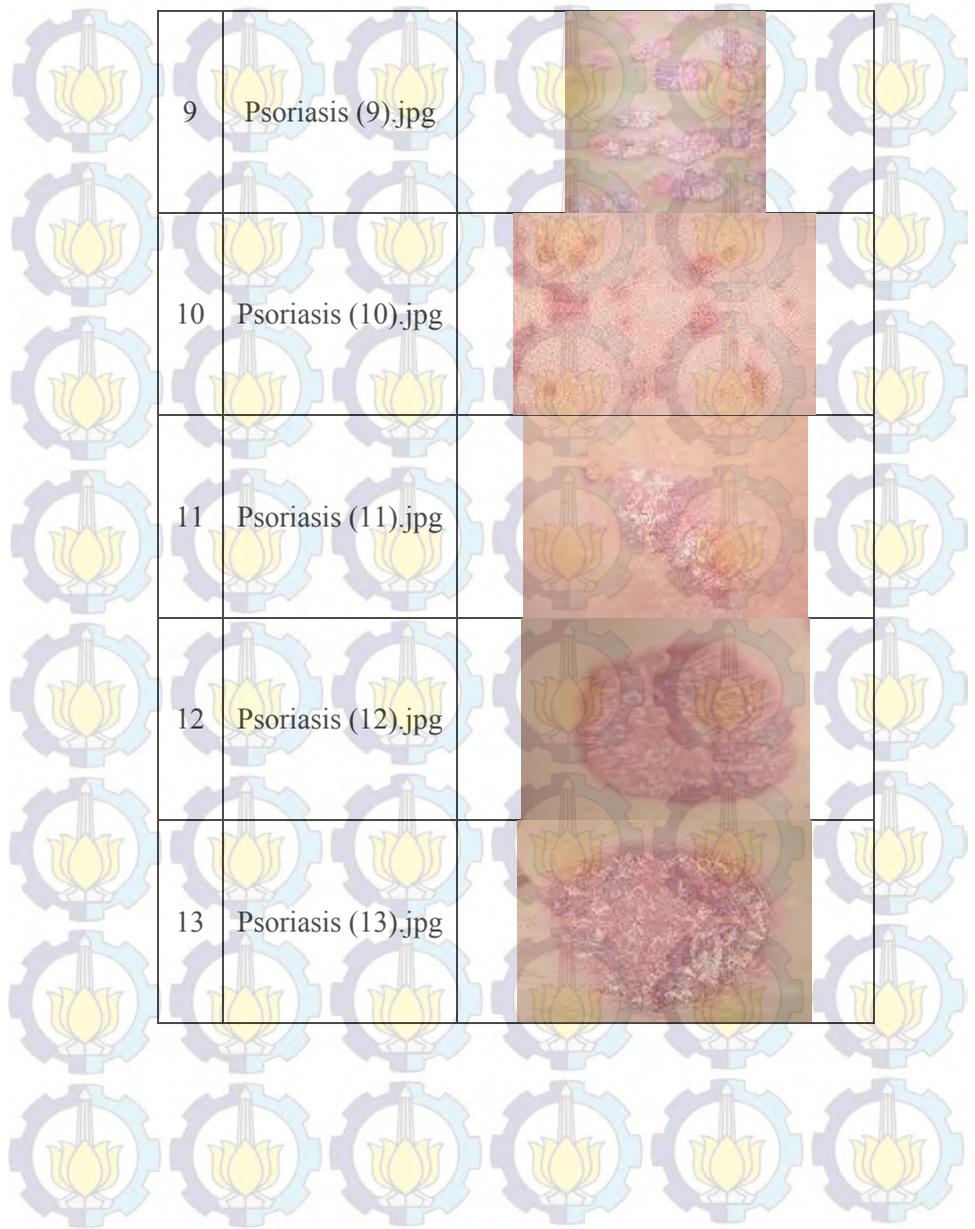
Pada lampiran ini, akan ditampilkan citra psoriasis yang menjadi data masukan pada aplikasi. Citra psoriasis yang digunakan dapat dilihat pada Tabel A.1.

Tabel A.8.1 Data Masukan Citra Psoriasis

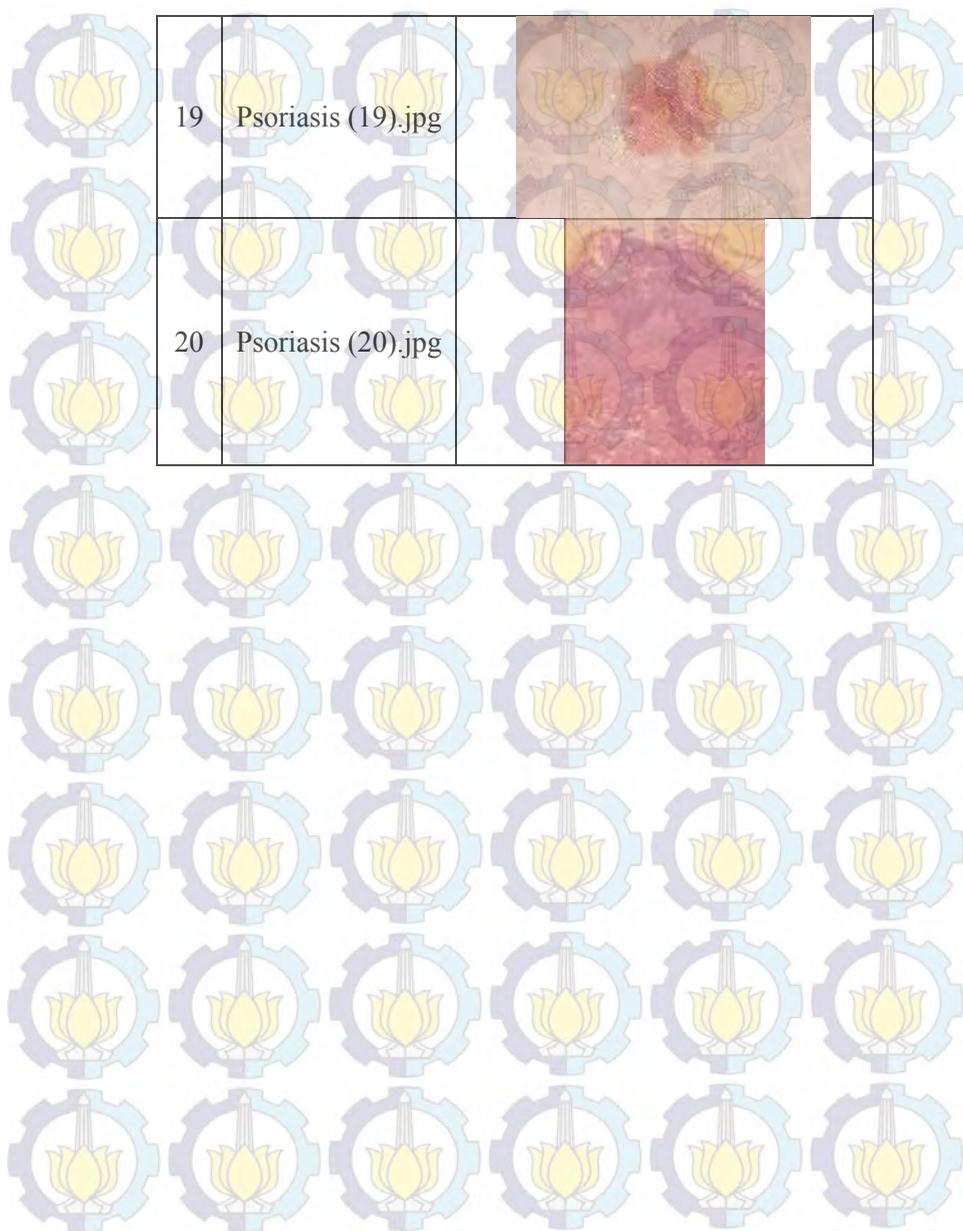
| No | Nama Citra | Gambar |
|----|-------------------|---|
| 1 | Psoriasis (1).jpg |  |
| 2 | Psoriasis (2).jpg |  |
| 3 | Psoriasis (3).jpg |  |



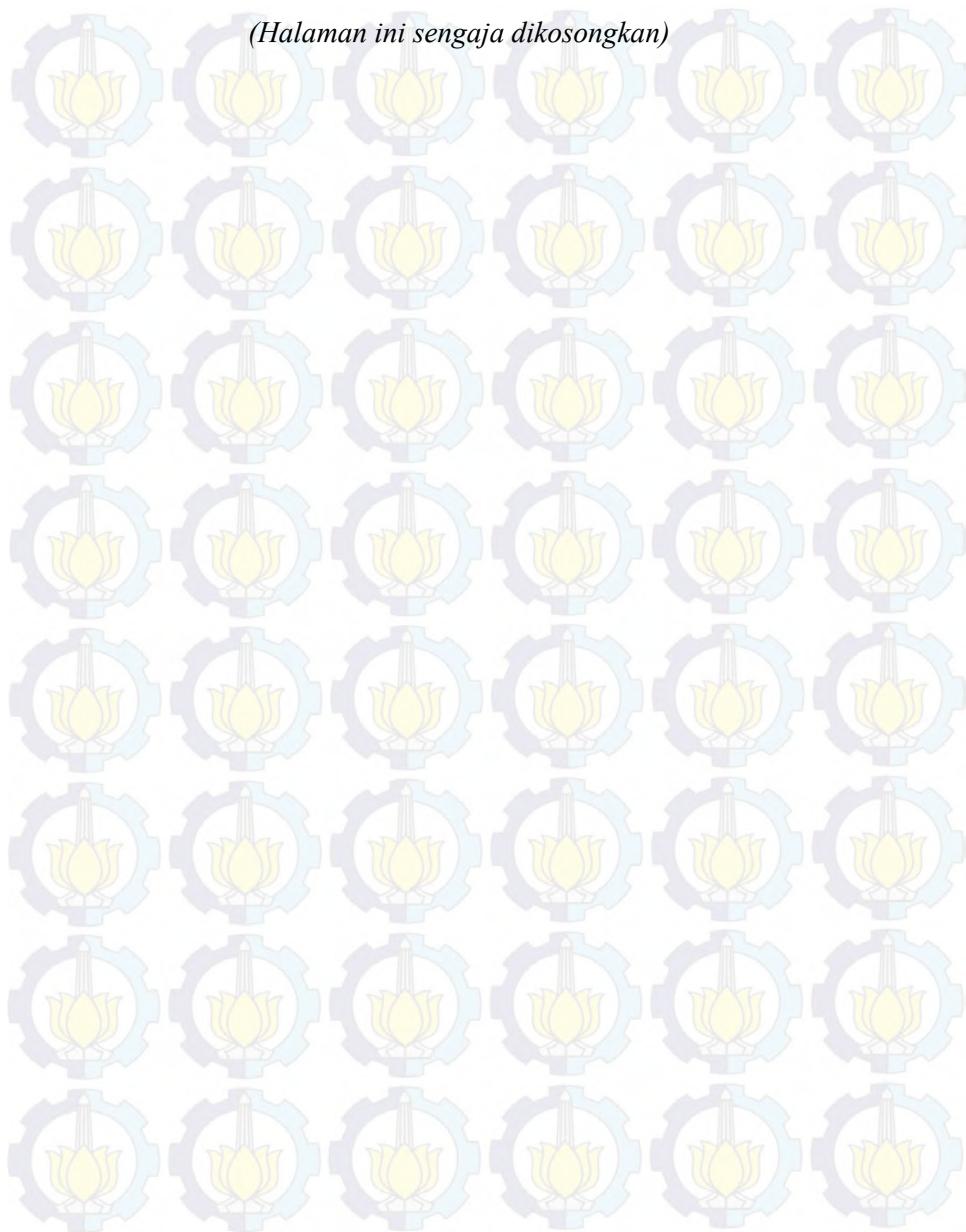
| | | |
|----|--------------------|--|
| 9 | Psoriasis (9).jpg |  |
| 10 | Psoriasis (10).jpg |  |
| 11 | Psoriasis (11).jpg |  |
| 12 | Psoriasis (12).jpg |  |
| 13 | Psoriasis (13).jpg |  |



| | | |
|----|--------------------|---|
| 14 | Psoriasis (14).jpg |  |
| 15 | Psoriasis (15).jpg |  |
| 16 | Psoriasis (16).jpg |  |
| 17 | Psoriasis (17).jpg |  |
| 18 | Psoriasis (18).jpg |  |



(Halaman ini sengaja dikosongkan)

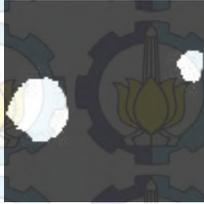
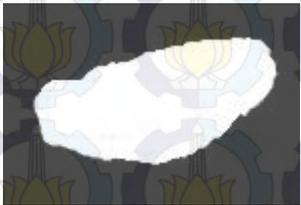


LAMPIRAN B

CITRA GROUND TRUTH

Pada lampiran ini, akan ditampilkan data *ground truth* yang akan dijadikan bahan dalam uji perbandingan. Citra *ground truth* dapat dilihat pada Tabel B.1..

Tabel B.9.1 Data Ground Truth

| No | Nama Citra | Gambar |
|----|---------------------|---|
| 1 | groundtruth (1).jpg |  |
| 2 | groundtruth (2).jpg |  |
| 3 | groundtruth (3).jpg |  |

