



TUGAS AKHIR - KI141502

**IMPLEMENTASI SEGMENTASI CITRA NATURAL
MENGUNAKAN TEKNIK BERBASIS
*THRESHOLDING***

**ISROVANA KHAMIDAH
NRP 5108 100 078**

**Dosen Pembimbing I
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**

**Dosen Pembimbing II
Anny Yuniarti, S.Kom., M.Comp.Sc.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015**



FINAL PROJECT - KI141502

IMPLEMENTATION OF NATURAL IMAGE SEGMENTATION USING THRESHOLDING- BASED TECHNIQUES

**ISROVANA KHAMIDAH
NRP 5108 100 078**

First Advisor

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

Second Advisor

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

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015**

**LEMBAR PENGESAHAN
IMPLEMENTASI SEGMENTASI CITRA NATURAL
MENGUNAKAN TEKNIK BERBASIS
THRESHOLDING**

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 :

**ISROVANA KHAMIDAH
NRP. 5108 100 078**

Disetujui oleh Pembimbing Tugas Akhir

**1. Dr. Eng. Nanik Suciati, S. Kom., M.Kom.
(NIP: 1971 04 28 1994 12 2001)**

(Pembimbing I)

**2. Anny Yuniarti, S. Kom., M.Comp.Sc.
(NIP: 1981 06 22 2005 01 2002)**

(Pembimbing II)

**SURABAYA
JULI, 2015**

IMPLEMENTASI SEGMENTASI CITRA NATURAL MENGUNAKAN TEKNIK BERBASIS *THRESHOLDING*

Nama Mahasiswa : Isrovana Khamidah
NRP : 5108 100 078
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing I : Dr. Eng. Nanik Suciati, S. Kom., M.
Kom.
Dosen Pembimbing II : Anny Yuniarti, S.Kom.,M.Comp.Sc.

ABSTRAK

Segmentasi citra merupakan salah satu masalah paling mendasar dan sulit dalam analisis citra karena merupakan bagian penting dalam pengolahan suatu citra. Segmentasi citra mengacu pada proses partisi gambar menjadi beberapa daerah atau set piksel dengan tujuan untuk mengambil obyek dari latar belakangnya. Namun proses segmentasi sangat menantang terutama pada percobaan yang dilakukan di lingkungan luar. Hal ini disebabkan sulit untuk menghasilkan nilai ambang signifikan yang diperlukan untuk segmentasi citra karena pencahayaan yang tidak seragam dan perbedaan refleksi pada obyek.

Tugas Akhir ini bertujuan untuk melakukan implementasi segmentasi berbasis teknik thresholding pada citra natural. Metode itu sendiri telah diusulkan pada penelitian sebelumnya [3]. Metode tersebut membentuk dan mengombinasikan teknik segmentasi berbasis thresholding dengan proses invers. Inisialisasi nilai threshold awal yang digunakan adalah metode Otsu. Metode segmentasi berbasis thresholding yang diimplementasi dalam Tugas Akhir ini disebut sebagai metode TsTN.

Uji coba dilakukan menggunakan set data dari penelitian sebelumnya [3]. Skenario uji coba berupa perbandingan segmentasi metode Otsu dan metode TsTN. Berdasarkan hasil uji coba, perangkat lunak yang dikembangkan dengan metode TsTN mampu melakukan segmentasi pada citra natural. Hal ini ditunjukkan oleh nilai parameter MSE dan PSNR yang lebih baik dari metode Otsu.

Kata kunci : segmentasi, citra natural, thresholding, Otsu.

IMPLEMENTATION OF NATURAL IMAGE SEGMENTATION USING THRESHOLDING-BASED TECHNIQUES

Student's Name : Isrovana Khamidah
Student's ID : 5108 100 078
Department : Informatics FTIf-ITS
First Advisor : Dr. Eng. Nanik Suciati, S. Kom., M.
Kom.
Second Advisor : Anny Yuniarti, S.Kom.,M.Comp.Sc.

ABSTRACT

Image segmentation is one of the fundamental problems in image processing and is a critical and essential component of image analysis system. Image segmentation refers to a process of partitioning an image into multiple meaningful regions or sets of pixels with the aim to extract object of interest from the background. However, the segmentation process is very challenging, especially for experiment which conducted in outdoor environment. It is difficult to produce a significant threshold value which required for segmenting images due to non uniform illumination and difference of reflection.

This final project aims to implement a segmentation thresholding-based techniques on image natural. The method itself has been proposed in previous studies [3]. The method of forming and combining thresholding-based segmentation techniques with an inverse process. Initialization threshold value first used is the Otsu's method. Thresholding-based segmentation techniques which is implemented in this final project is referred to as a TsTN method.

The segmentation techniques were implemented on a set of data from previous studies [3]. Otsu's method and TsTN method is used in the form of a comparison evaluation of segmentation.

Based on analysis results, the segmentation technique performed by TsTN method is capable of performing natural image segmentation. This is showed by the value of MSE and PSNR on TsTN is better than Otsu's method.

Keywords: segmentation, natural image, thresholding, Otsu.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah rabbil 'alamiin, puji syukur penulis ucapkan pada Allah SWT. yang telah memberi kesempatan dan kemampuan untuk dapat menyelesaikan Tugas Akhir dengan judul **“Implementasi Segmentasi Citra Natural Menggunakan Teknik Berbasis Thresholding”**. Sholawat serta salam senantiasa tercurahkan kepada Rasulullah SAW., pelita bagi umat manusia.

Harapan penulis adalah semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat menjadi bagian dari pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi nyata bagi kampus Teknik Informatika, ITS, dan bangsa Indonesia.

Terselesaikannya Tugas Akhir ini tidak lepas dari dukungan dan bantuan dari berbagai pihak. Oleh karena itu, dengan segala ketulusan dan kerendahan hati, penulis ingin mengucapkan rasa terima kasih yang tulus dan sedalam-dalamnya kepada :

1. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku dosen pembimbing pertama dan Kepala Jurusan Teknik Informatika ITS yang telah memberikan bimbingan, bantuan, dukungan, perhatian, arahan, dan masukan yang bermanfaat dalam proses pengerjaan Tugas Akhir, serta pelajaran hidup yang begitu berharga.
2. Ibu Anny Yuniarti, S.Kom., M.Comp.Sc. selaku dosen pembimbing kedua yang telah memberikan pengarahan dan bimbingan yang sangat bermanfaat kepada penulis.
3. Mama Wiwik Indayati, Papa Masjhari, Mas Miftakhul Khoiri, S.T., dan Adik Nizar El Karim tercinta serta segenap keluarga besar penulis, yang senantiasa memberikan dukungan moral dan material serta do'a tak

terhingga kepada penulis sehingga Tugas Akhir ini dapat terselesaikan tepat waktu.

4. Ibu Umi Laili Yuhana, S.Kom., M.Sc. selaku dosen wali penulis, Bapak Dr. Eng. Radityo Anggoro, S.Kom., M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika ITS yang telah memberikan segala ilmu, pengetahuan, bimbingan, bantuan, dan kemudahan kepada penulis selama menjalani kuliah.
5. Pak Yudi Mulyono dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama kuliah di Teknik Informatika ITS.
6. Sahabat-sahabat penulis: Cita, Nuru, Dea, Lena, Kharis, Hallaz, Mustofa, Pujiyanto, Echa, Maya, Venta, Siwi, Fachruddin, Erriek, Aguk, Mbak Zunni yang telah memberikan dukungan dan motivasi, serta menjadi teman diskusi untuk menyelesaikan Tugas Akhir ini.
7. Serta semua pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir, yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharap kritik dan saran dari pembaca untuk perbaikan nantinya. Pada akhirnya, penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan bagi semua pihak.

Surabaya, Juli 2015

Isrovana Khamidah

DAFTAR ISI

ABSTRAK	vii
KATA PENGANTAR	xi
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Metodologi Pembuatan Tugas Akhir.....	4
1.7 Sistematika Penulisan.....	6
BAB II DASAR TEORI	9
2.1 Citra Natural	9
2.2 Segmentasi Citra.....	10
2.2.1 Proses Segmentasi Citra	11
2.2.2 Contoh Aplikasi Segmentasi Citra	13
2.2.3 Algoritma Segmentasi Citra	14
2.3 Thresholding.....	15
2.4 Metode Otsu	18
2.5 TsTN.....	20
2.6 Fuzzy c-means	23

2.7	MSE (<i>Mean Square Error</i>) dan PSNR (<i>Peak Signal to Noise Ratio</i>).....	23
2.8	Matlab.....	25
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		27
3.1	Analisis.....	27
3.1.1	Cakupan Permasalahan	27
3.1.2	Lingkungan Perancangan Perangkat Lunak.....	28
3.2	Perancangan Data	29
3.2.1	Data Masukan.....	29
3.2.2	Data Keluaran.....	31
3.3	Perancangan Proses	31
3.3.1	Perancangan Proses Secara Umum	31
3.3.2	Otsu	33
3.3.3	TsTN	33
BAB IV IMPLEMENTASI SISTEM		37
4.1	Lingkungan Pengembangan Sistem.....	37
4.2	Implementasi Otsu.....	38
4.3	Implementasi TsTN.....	38
BAB V PENGUJIAN DAN EVALUASI		43
5.1	Lingkungan Pengujian.....	43
5.2	Data Uji Coba.....	44
5.3	Skenario Pengujian.....	44
5.3.1	Uji coba dengan perbandingan beberapa metode segmentasi	44
5.3.2	Hasil Uji coba beberapa metode segmentasi ..	45

5.3.3	Uji coba dengan perbandingan nilai parameter MSE dan PSNR.....	58
5.3.4	Hasil Ujicoba MSE dan PSNR.....	58
5.4	Analisis Hasil Ujicoba.....	63
BAB VI KESIMPULAN DAN SARAN		67
6.1	Kesimpulan.....	67
6.2	Saran.....	68
DAFTAR PUSTAKA.....		69
BIODATA PENULIS.....		71

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Analisis Segmentasi Citra	11
Gambar 2.2 Contoh Penggunaan <i>Threshold</i> pada Suatu Citra...	17
Gambar 3.1 Diagram Alir Proses TsTN.....	34
Gambar 4.1 Kode Program Proses Inisialisasi Otsu	38
Gambar 4.2 Kode Program Proses Modifikasi <i>Threshold</i> Pada TsTN.....	40
Gambar 4.3 Kode Program Fungsi Hitung Pixel Untuk Modifikasi <i>Threshold</i>	41
Gambar 4.4 Kode Program Fungsi Segmentasi TsTN.....	42
Gambar 5.1 Citra <i>Jatropha Fruit</i> Hijau Natural	47
Gambar 5.2 Citra <i>Jatropha Fruit</i> Hijau <i>Ground Truth</i>	47
Gambar 5.3 Citra <i>Jatropha Fruit</i> Hijau Segmentasi Otsu	48
Gambar 5.4 Citra <i>Jatropha Fruit</i> Hijau Segmentasi TsTN.....	48
Gambar 5.5 Citra <i>Jatropha Fruit</i> Kuning Natural	50
Gambar 5.6 Citra <i>Jatropha Fruit</i> Kuning <i>Ground Truth</i>	50
Gambar 5.7 Citra <i>Jatropha Fruit</i> Kuning Segmentasi Otsu	51
Gambar 5.8 Citra <i>Jatropha Fruit</i> Kuning Segmentasi TsTN	51
Gambar 5.9 Citra <i>Jatropha Fruit</i> Kuning Kecoklatan Natural ..	52
Gambar 5.10 Citra <i>Jatropha Fruit</i> Kuning Kecoklatan <i>Ground Truth</i>	52
Gambar 5.11 Citra <i>Jatropha Fruit</i> Kuning Kecoklatan Segmentasi Otsu	53
Gambar 5.12 Citra <i>Jatropha Fruit</i> Kuning Kecoklatan Segmentasi TsTN.....	53
Gambar 5.13 Citra <i>Jatropha Fruit</i> Hitam Natural	56
Gambar 5.14 Citra <i>Jatropha Fruit</i> Hitam <i>Ground Truth</i>	56
Gambar 5.15 Citra <i>Jatropha Fruit</i> Hitam Segmentasi Otsu	57
Gambar 5.16 Citra <i>Jatropha Fruit</i> Hitam Segmentasi TsTN	57

DAFTAR TABEL

Tabel 3.1 Lingkungan Perancangan Perangkat Lunak.....	28
Tabel 3.2 Set Data Citra <i>Jatropha Fruit</i>	30
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak	37
Tabel 5.1 Lingkungan Uji Coba Perangkat Lunak.....	43
Tabel 5.2 Performa Parameter MSE dan PSNR untuk Segmentasi Citra <i>Jatropha Fruit</i> Hijau	59
Tabel 5.3 Performa Parameter MSE dan PSNR untuk Segmentasi Citra <i>Jatropha Fruit</i> Kuning.....	60
Tabel 5.4 Performa Parameter MSE dan PSNR untuk Segmentasi Citra <i>Jatropha Fruit</i> Kuning Kecoklatan.....	61
Tabel 5.5 Performa Parameter MSE dan PSNR untuk Segmentasi Citra <i>Jatropha Fruit</i> Hitam.....	62

BAB I

PENDAHULUAN

Bab ini membahas garis besar penyusunan Tugas Akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan Tugas Akhir, dan sistematika penulisan.

1.1 Latar Belakang

Segmentasi citra adalah salah satu masalah paling mendasar dan sulit dalam analisis citra karena merupakan bagian penting dalam pengolahan suatu citra. Suatu citra digital monokrom biasanya merupakan suatu obyek *array* dua dimensi dari *gray pixels*. Proses isolasi daerah-daerah penting suatu citra agar menjadi sebuah komponen, umumnya disebut sebagai segmentasi citra. Segmentasi citra mengacu pada proses pembagian citra digital ke dalam beberapa daerah regional dengan tujuan memisahkan (ekstraksi) obyek yang dimaksud agar dapat terpisah dari obyek yang menjadi latar belakangnya (*background*). Dalam visi komputer, segmentasi citra merupakan proses partisi (pembagian) suatu citra ke dalam daerah atau obyek yang penting. Berbagai macam aplikasi segmentasi citra telah banyak diciptakan, seperti aplikasi untuk mencari tumor maupun patologi lainnya, mengukur volume jaringan, *computer-guide surgery*, perencanaan pengobatan, studi struktur anatomi, mencari obyek dalam citra satelit dan pengenalan sidik jari dan lain sebagainya.

Segmentasi merupakan salah satu fase kritis dalam permasalahan *image processing* karena kualitas gambar (citra) yang disegmentasi tersebut akan mempengaruhi hasil keseluruhan proses. Namun, proses segmentasi sangatlah menantang terutama pada beberapa percobaan yang dilakukan di lingkungan luar (*outdoor*), sehingga sulit untuk menghasilkan nilai *threshold* (ambang batas) yang signifikan yang diperlukan untuk segmentasi citra. Hal ini bisa saja terjadi karena pencahayaan

yang tidak seragam dan perbedaan refleksi pada obyeknya. Perbedaan pencahayaan dapat menghasilkan intensitas warna berbeda pada permukaan obyek tersebut sehingga menyebabkan segmentasi citra yang dihasilkan menjadi tidak akurat.

Beberapa teknik yang telah digunakan dalam proses segmentasi diantaranya adalah teknik *thresholding-based*, *edge-based*, *region-based*, dan teknik *clustering*. Diantara semua teknik segmentasi tersebut, teknik *thresholding* dan *clustering* merupakan metode yang paling sering digunakan. Salah satu teknik segmentasi yang banyak digunakan dalam *thresholding-based* ialah metode Otsu [1]. Sedangkan untuk teknik *clustering*, salah satu metode yang sering dipakai ialah metode Fuzzy c-means (FCM) [2]. Metode Otsu merupakan metode yang baik karena penerapannya cukup sederhana dan memiliki kemampuan untuk memproses *gray level image* secara langsung serta mampu memproses nilai *global threshold* yang disebabkan oleh sensitivitas rendah pada area gelap. Namun, metode ini tidak efisien dalam menentukan nilai *threshold* optimal karena membutuhkan waktu komputasi yang tinggi terutama untuk gambar yang diklasifikasikan ke dalam sejumlah besar kelas. Selain itu, penggunaan metode Otsu saja dalam suatu aplikasi tidak cukup menghasilkan segmentasi yang akurat, terutama untuk gambar pada kondisi pencahayaan yang tidak merata. Sedangkan metode FCM memungkinkan data yang diproses untuk satu atau lebih cluster dengan nilai keanggotaan yang berbeda, masih stabil meskipun terjadi *overlapping cluster*. Selain itu, teknik FCM ini efisien, sederhana dan mudah diimplementasikan. Namun sayangnya, metode ini tidak dapat menghasilkan area segmentasi dengan kualitas baik karena kompleksitas *background* dan pencahayaan yang tidak seragam pada citra yang diambil dari lingkungan natural, dan memerlukan waktu komputasi yang besar.

Untuk mengatasi permasalahan tersebut, dalam Tugas Akhir ini akan diimplementasikan sebuah pendekatan segmentasi citra

berbasis *thresholding* yang diintegrasikan dengan teknik invers, disebut algoritma TsTN (*an improved Thresholding-based segmentation Technique for Natural images*) [3], sehingga mampu mempartisi citra natural (*natural images*) secara tepat dan akurat. Metode TsTN yang diimplementasikan dalam Tugas Akhir ini menggunakan modifikasi nilai *threshold* yang didapatkan dari metode *thresholding-based* segmentasi citra yang sudah ada sebelumnya, yaitu Metode Otsu.

1.2 Rumusan Masalah

Permasalahan-permasalahan yang dikemukakan dan berusaha untuk diselesaikan pada implementasi metode *Thresholding-based segmentation Technique* (TsTN) ini adalah sebagai berikut:

- a) Bagaimana mengimplementasikan modul-modul dalam metode TsTN (*Thresholding-based segmentation Technique for Natural images*) yang meliputi konsep segmentasi *thresholding* dan diintegrasikan dengan teknik *inverse*.
- b) Bagaimana mengevaluasi model segmentasi citra dengan menggunakan metode TsTN yang diterapkan pada citra natural.

1.3 Batasan Masalah

Dalam pengerjaan Tugas Akhir ini, terdapat beberapa acuan yang menjadi batasan masalah sebagai berikut:

1. Program dibangun dengan menggunakan perangkat lunak MATLAB.
2. Metode segmentasi citra yang digunakan adalah pengembangan dari segmentasi berbasis *thresholding* yang diintegrasikan dengan teknik *inverse*.
3. Data yang dipakai adalah data citra bernama *jatropha fuit* seperti yang dipakai pada penelitian sebelumnya [3]. Data set tersebut dipilih berdasarkan warna permukaan

yang berbeda, disebabkan oleh pencahayaan yang tidak seragam, untuk menunjukkan level yang berbeda.

1.4 Tujuan

Tujuan yang akan dicapai pada Tugas Akhir ini adalah sebagai berikut:

1. Mengimplementasikan algoritma *thresholding-based* dan *inverse* untuk permasalahan segmentasi citra.
2. Mengetahui kinerja algoritma dalam melakukan segmentasi pada citra natural.

1.5 Manfaat

Keberhasilan dalam implementasi algoritma ini diharapkan dapat memberikan manfaat yang besar pada bidang *image processing* untuk proses segmentasi citra natural sehingga dapat memberikan hasil yang lebih baik dari algoritma sebelumnya.

1.6 Metodologi Pembuatan Tugas Akhir

Pembuatan Tugas Akhir ini dibagi menjadi beberapa tahap pengerjaan yang digunakan sebagai acuan dalam penyelesaian Tugas Akhir. Berikut ini adalah tahap-tahap pengerjaan Tugas Akhir yang dimulai sejak pengajuan proposal hingga buku ini disusun, yaitu 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 untuk mengimplementasikan *thresholding-based* dan *inverse* pada permasalahan segmentasi citra.
2. Studi Literatur
Pada tahap ini dilakukan pencarian, pengumpulan, penyaringan, pembelajaran dan pemahaman literatur yang berhubungan dengan proses segmentasi citra dan algoritma yang digunakan seperti *thresholding-based* (*Otsu method*),

clustering-based (FCM), dan *inversion technique*. Literatur yang digunakan dalam pengerjaan Tugas Akhir ini sebagian besar berasal dari internet berupa makalah ilmiah, artikel, materi kuliah, serta beberapa buku referensi.

3. Pengimplementasian Algoritma secara Analitis
Tahap ini merupakan tahap penerapan dasar teori yang telah dipahami dalam penggunaan algoritma dan rumus yang akan diterapkan.
4. Perancangan Perangkat Lunak dan Desain Sistem
Pada tahap ini dilakukan proses perancangan perangkat lunak, berdasarkan literatur yang telah dikaji kemudian dibuat disain model, diagram alir proses-proses yang ada, dan disain antarmuka aplikasi. Kemudian untuk selanjutnya dilakukan implementasi.
5. Implementasi Perancangan Perangkat Lunak
Pada tahap ini dilakukan proses pengimplementasian metode TsTN untuk proses segmentasi citra, dengan berdasar pada rancangan awal perangkat lunak dan literatur yang telah dikaji. Tahap ini terdiri atas perancangan dan implementasi metode Otsu untuk inisialisasi *threshold* awal, kemudian mengombinasikannya dengan invers untuk proses segmentasi citra.
6. Uji Coba dan Evaluasi
Pada tahap ini dilakukan uji coba terhadap implementasi yang telah dibuat, mengamati kinerjanya, dan mengidentifikasi kendala atau kesalahan yang mungkin timbul untuk selanjutnya dapat dilakukan perbaikan.
7. Penyusunan Laporan Tugas Akhir
Tahap terakhir ini merupakan tahap untuk menyusun laporan Tugas Akhir. Laporan Tugas Akhir merupakan dokumentasi dari keseluruhan tahap-tahap sebelumnya. Penyusunan laporan ini diharapkan untuk dapat berguna dalam penelitian atau pengembangan lebih lanjut.

1.7 Sistematika Penulisan

Pembahasan laporan Tugas Akhir ini dibagi menjadi beberapa bab sebagai berikut:

Bab I Pendahuluan

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

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini. Secara garis besar, bab ini berisi tentang teori segmentasi citra.

Bab III Perancangan Perangkat Lunak

Bab ini berisi pembahasan mengenai disain yang digunakan dalam implementasi *improved thresholding-based* yang diintegrasikan dengan teknik *inverse*, disebut dengan metode TsTN, pada permasalahan segmentasi citra. Pada bab ini juga dibahas mengenai perancangan data yang terdiri dari proses, data masukan, dan data keluaran.

Bab IV Implementasi

Bab ini akan membahas implementasi dari aplikasi yang telah dibuat, akan dilakukan pembuatan aplikasi yang dibangun dengan komponen-komponen yang telah ada yang sesuai dengan permasalahan dan batasannya yang telah dijabarkan pada bab pertama.

Bab V Uji Coba dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dalam melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II DASAR TEORI

Bab ini membahas dasar teori yang digunakan dalam penyelesaian masalah pada Tugas Akhir. Dasar teori yang akan dijelaskan terbagi dalam beberapa subbab, yaitu teori segmentasi, citra natural, *thresholding*, dan beberapa metode segmentasi seperti metode Otsu, TsTN dan FCM. Serta teori tentang parameter MSE dan PSNR sebagai pengujian *error*. Selain itu, pada bab dasar teori ini juga akan dijelaskan secara singkat tentang aplikasi Matlab yang berfungsi *platform* dalam membangun perangkat lunak segmentasi pada Tugas Akhir ini. Materi-materi tersebut masing-masing akan dijelaskan dalam subbab tersendiri.

2.1 Citra Natural

Menurut wikipedia, citra adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses *sampling*. Gambar analog dibagi menjadi N baris dan M kolom sehingga menjadi gambar diskrit. Persilangan antara baris dan kolom tertentu disebut dengan piksel, contohnya gambar atau titik diskrit pada baris n dan kolom m disebut dengan piksel $[n, m]$.

Citra atau *image* berasal dari kata “*imago*” yang merupakan bahasa Latin, yang berarti gambar. Sebuah gambar adalah *array* dua dimensi atau matriks piksel persegi yang diatur dalam kolom dan baris. Setiap piksel mewakili warna (RGB, - *red, green, blue*) atau abu-abu pada satu titik dalam gambar [7].

Citra Natural merupakan citra (*image*) yang didapat dari lingkungan natural tanpa melalui proses pengolahan digital, misalnya *filtering*. Citra natural terdiri dari sejumlah besar pola visual yang dihasilkan dari beragam distribusi probabilitas acak atau pola yang dapat dianalisis secara statistik tetapi tidak dapat

diprediksi secara tepat (*stochastic*). Citra natural biasanya sangat bising (*noisy*) karena lingkungan asal mereka. Citra natural yang berasal dari lingkungan luar (*outdoor*) biasanya mengalami pencahayaan alami yang berasal dari sinar matahari, atau sumber sinar lain pada lingkungan sekitarnya. Pencahayaan alami ini akan menyebabkan suatu citra natural memiliki kecenderungan warna permukaan yang tidak merata.

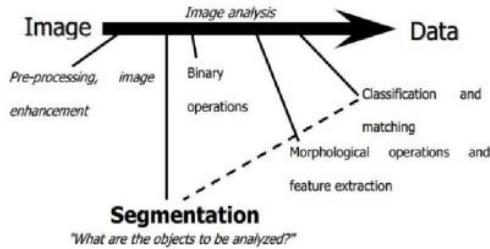
Sebuah citra digital merupakan representasi numerik (biner) dari citra dua dimensi. Setiap gambar dari *scanner*, kamera digital, atau komputer adalah contoh dari citra digital [7].

2.2 Segmentasi Citra

Segmentasi menurut Kamus Besar Bahasa Indonesia (KBBI) merupakan suatu proses pembagian ke dalam segmen-segmen (bagian) tertentu, dimana terdapat suatu obyek yang akan dibagi strukturnya ke dalam unit-unit tertentu yang memiliki karakteristik yang sama.

Dalam visi komputer juga dikenal istilah segmentasi yang merupakan bagian dari proses pengolahan citra, biasa disebut segmentasi citra. Pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasinya berwujud citra [1].

Segmentasi citra adalah proses membagi suatu citra menjadi beberapa wilayah homogen berdasarkan kriteria keserupaan tertentu antara tingkat keabuan suatu piksel dengan tingkat keabuan piksel-piksel lain di sekitarnya [4]. Tujuan dari segmentasi ialah untuk menyederhanakan atau mengubah representasi dari suatu gambar menjadi sesuatu yang lebih berarti dan lebih mudah untuk dianalisis, seperti yang ditunjukkan pada gambar 2.1. [5].



Gambar 2.1 Ilustrasi Analisis Segmentasi Citra

Tingkat detail pembagian citra yang dilakukan tergantung pada masalah yang sedang diselesaikan. Maksudnya ialah segmentasi akan berhenti ketika obyek atau daerah yang dimaksud di dalam aplikasi tersebut telah terdeteksi [4]. Sebagai contoh, misalnya aplikasi pendeteksi otomatis rangkaian elektronik, fokus segmentasi terletak pada analisa gambar produk dengan tujuan menentukan ada tidaknya anomali tertentu, seperti hilangnya suatu komponen atau rusaknya jalur koneksi. Bisa dikatakan, segmentasi citra termasuk salah satu masalah paling mendasar dan sulit dalam pengolahan citra karena merupakan bagian penting dalam pengolahan suatu citra.

2.2.1 Proses Segmentasi Citra

Proses segmentasi memiliki tujuan yang hampir sama dengan proses klasifikasi. Segmentasi sering dideskripsikan sebagai proses analogi terhadap proses pemisahan latar depan (*foreground*) dan latar belakang (*background*). Dengan proses segmentasi tersebut, masing-masing obyek pada citra dapat diambil secara individu sehingga dapat digunakan sebagai *input* bagi proses lain [5].

Secara matematis, proses segmentasi akan dijelaskan melalui persamaan 2.1. (a) sampai 2.1. (d) seperti berikut ini [4].

Misal, R merepresentasikan seluruh wilayah spasial yang dimiliki oleh suatu citra (gambar). Sehingga segmentasi citra yang dihasilkan merupakan satu proses yang mempartisi (membagi) R ke dalam n subregional (bagian-bagian wilayah yang lebih kecil), sebanyak R_1, R_2, \dots, R_n sehingga:

(Persamaan 2.1.)

- a) $\bigcup_{i=1}^n R_i = R$.
- b) R_i merupakan suatu *connected set*, dimana $i = 1, 2, \dots, n$.
- c) $R_i \cap R_j = \emptyset$ untuk semua i dan j , dimana $i \neq j$.
- d) $Q(R_i) = \text{TRUE}$, untuk $i = 1, 2, \dots, n$.
- e) $Q(R_i \cup R_j) = \text{FALSE}$ untuk setiap wilayah yang berdekatan (*adjacent regions*) dengan R_i dan R_j .

Dalam hal ini, $Q(R_k)$ didefinisikan sebagai poin-poin pada set R_k , dan \emptyset adalah himpunan nol. Simbol \bigcup dan \cap merepresentasikan gabungan (*union*) dan irisan (*intersection*). Dua daerah R_i dan R_j dikatakan berdekatan (*adjacent*) jika gabungan dari dua daerah tersebut membentuk suatu set yang terhubung (*connected set*) [4].

Persamaan 2.1.(a) menunjukkan bahwa segmentasi harus lengkap, yaitu setiap piksel harus berada di suatu wilayah. Persamaan 2.1.(b) mensyaratkan bahwa titik-titik pada suatu daerah (*region*) harus terhubung sebagai suatu *connected set*, (misal, titik 4 atau 8-*connected*). Persamaan 2.1.(c) menunjukkan bahwa daerah-daerah tersebut harus *disjoint* (tidak memiliki elemen yang sama). Persamaan 2.1.(d) sesuai dengan sifat yang harus dipenuhi oleh piksel pada daerah tersegmentasi,

misalnya $Q(R_i) = \text{TRUE}$ jika semua piksel pada R_i memiliki tingkat intensitas (*intensity level*) yang sama. Sedangkan persamaan 2.1.(e) menunjukkan bahwa dua daerah yang berdekatan (*adjacent*), yaitu R_i dan R_j harus berbeda dalam arti predikat Q. Secara umum, Q dapat memenuhi beberapa pernyataan, seperti misalnya, $Q(R_i) = \text{TRUE}$ jika intensitas rata-rata piksel pada R_i kurang dari m_i , dan jika standar deviasi intensitasnya lebih dari σ_i , di mana m_i dan σ_i memiliki nilai konstanta yang spesifik [4].

2.2.2 Contoh Aplikasi Segmentasi Citra

Berbagai macam aplikasi segmentasi citra telah banyak diciptakan. Menurut wikipedia, beberapa aplikasi yang menerapkan segmentasi citra di dalamnya, terdiri dari aplikasi-aplikasi berikut ini:

- a. *Content-based Image Retrieval*.
- b. *Machine Vision*.
- c. Pencitraan Medis (*medical imaging*).
Beberapa aplikasi *medical imaging* yang umum digunakan dalam kegiatan medis sehari-hari diantaranya ialah aplikasi untuk mendeteksi tumor maupun patologi lainnya, aplikasi untuk mengukur volume jaringan, aplikasi diagnosis dan studi struktur anatomi, aplikasi perencanaan operasi (*surgery planning*), aplikasi simulasi operasi virtual, serta aplikasi *intra-surgery navigation*.
- d. Deteksi Obyek.
Beberapa aplikasi deteksi obyek (*object detection*) yang umum digunakan dalam kegiatan sehari-hari diantaranya ialah aplikasi pendeteksi pejalan kaki (*pedestrian detection*), aplikasi pendeteksi wajah, aplikasi pendeteksi lampu rem, serta aplikasi pencari obyek dalam citra satelit (jalan, hutan, tanaman, dan lain sebagainya).
- e. *Recognition Tasks*.
Beberapa aplikasi pengenalan obyek (*recognition tasks*) yang umum digunakan dalam kegiatan sehari-hari

diantaranya ialah aplikasi pengenalan wajah (*face recognition*), aplikasi pengenalan sidik jari, serta aplikasi pengenalan iris mata.

- f. Sistem Pengontrol Lalu Lintas.
- g. *Video Surveillance*.

Akurasi segmentasi akan menentukan tingkat keberhasilan maupun kegagalan suatu prosedur analisis komputerisasi. Seorang *designer* (perancang) sistem pengolahan citra yang berpengalaman selalu mempunyai perhatian yang besar terhadap hal tersebut. Dalam aplikasi akuisisi sasaran otonom, perancang sistem tidak memiliki kontrol atas lingkungan operasi, sehingga pendekatan umumnya ialah fokus pada pemilihan jenis sensor yang paling mungkin untuk meningkatkan obyek yang dimaksud, sementara mengurangi kontribusi detail gambar yang tidak relevan. Sebuah contoh aplikasi pengolahan citra yang baik salah satunya adalah penggunaan pencitraan inframerah oleh satuan militer untuk mendeteksi obyek dengan menggunakan penanda temperatur panas yang kuat, seperti pada pendeteksian peralatan perang maupun pasukan yang sedang bergerak [4].

2.2.3 Algoritma Segmentasi Citra

Segmentasi merupakan proses mempartisi citra menjadi beberapa daerah atau obyek. Dengan demikian, kita melihat bahwa masalah mendasar dalam segmentasi adalah kebutuhan untuk partisi sebuah citra ke daerah yang memenuhi kondisi sebelumnya. Sebagian besar algoritma segmentasi didasarkan pada salah satu dari dua sifat dasar nilai intensitas piksel, yaitu diskontinuitas (*discontinuity*) dan kesamaan (*similarity*)[1, 4, 6].

- a. Pendekatan *discontinuity*.

Pada kategori pertama, yaitu *discontinuity*, pendekatan ini bertujuan untuk mempartisi sebuah citra (gambar) berdasarkan perubahan dalam intensitas yang terjadi secara tiba-tiba, misalnya perubahan tepi (*edge*). Pendekatan ini biasa disebut sebagai *edge-based segmentation*.

Metode *edge-based segmentation* melakukan segmentasi citra dengan mendeteksi tepi (*edge*) atau piksel antara daerah yang berbeda yang memiliki transisi cepat dalam intensitas yang diekstraksi dan dihubungkan untuk membentuk batas-batas obyek tertutup (*closed object boundaries*). Hasil dari metode ini adalah citra biner.

Berdasarkan teori *discontinuity*, ada dua metode utama dari segmentasi berdasarkan tepi (*edge-based segmentation*), yaitu:

- metode *gray histogram based*.
- metode *gradient based*.

b. Pendekatan *similarity*.

Pendekatan utama dalam kategori kedua ini dilakukan berdasarkan pada proses mempartisi citra ke dalam daerah-daerah yang memiliki kesamaan sifat tertentu. Pendekatan ini biasa disebut sebagai *region-based segmentation*.

Metode *region-based segmentation* melakukan segmentasi citra dengan mempartisi suatu citra (gambar) ke dalam area-area yang sama berdasarkan piksel-piksel yang menghubungkannya. Setiap piksel dalam suatu wilayah dihitung kemiripannya dengan memperhatikan beberapa properti karakteristik seperti warna, intensitas, dan teksturnya.

Contoh metode segmentasi yang termasuk dalam *region-based segmentation* ialah:

- *Thresholding*.
- *region growing*.
- *region splitting and merging*.

2.3 Thresholding

Menurut wikipedia, *thresholding* adalah metode paling sederhana dalam segmentasi citra. Dari sebuah citra (gambar) *grayscale*, *thresholding* dapat digunakan untuk membuat citra

(gambar) biner. Metode ini didasarkan pada tingkat *clip* (nilai ambang batas, - *threshold*) untuk mengubah suatu citra *grayscale* menjadi citra biner. Dalam pembahasan *thresholding* juga dikenal istilah *histogram thresholding*. Metode *histogram thresholding* ini digunakan untuk memilih nilai *threshold*, atau nilai ketika *multiple-level* dipilih. Beberapa metode yang populer digunakan dalam *thresholding-based segmentation* diantaranya ialah metode entropi maksimum, metode Otsu (*maximum variance*), dan *k-means clustering*.

Thresholding adalah teknik penting dalam aplikasi segmentasi citra. Ide dasar dari *thresholding* adalah memilih nilai optimal ambang batas *gray-level* (*optimal gray-level threshold*) untuk memisahkan obyek yang menarik pada gambar (*object of interest*) dari latar belakangnya (*background*), berdasarkan distribusi *gray-level*nya. Karena manusia dengan mata telanjang dapat dengan mudah terdiferensiasi oleh obyek dan latar belakang yang kompleks, maka *image thresholding* merupakan tugas yang sulit untuk dilakukan, dengan tujuan memisahkan kedua hal tersebut. Histogram tingkat keabuan dari suatu gambar biasanya dianggap sebagai alat yang efisien untuk pengembangan algoritma *image thresholding* [1].

Metode *thresholding* secara sederhana mengganti setiap piksel dalam sebuah citra (gambar) menjadi piksel hitam jika intensitas citra $I_{i,j}$ kurang dari sama dengan nilai T , (yaitu, $I_{i,j} \leq T$), dan akan mengganti setiap piksel dalam sebuah citra (gambar) menjadi piksel putih jika intensitas citra lebih besar dari nilai konstanta T , seperti yang ditunjukkan pada Persamaan 2.2 [4],

$$g(i, j) = \begin{cases} 1 \text{ (putih)} & \text{jika } I(i, j) > T \\ 0 \text{ (hitam)} & \text{jika } I(i, j) \leq T \end{cases} \quad \dots (2.2)$$

dimana $g(i, j)$ merupakan nilai piksel dari suatu citra biner dan $I(i, j)$ merupakan nilai intensitas dari citra *grayscale*.

Operasi *thresholding* didefinisikan seperti pada persamaan 2.3 berikut ini [1].

$$T = M [x, y, p(x, y), f(x, y)] \quad \dots (2.3)$$

Dalam persamaan 2.3, T merupakan *threshold*, sedangkan $f(x, y)$ adalah nilai *gray level* suatu titik (x, y) , dan $p(x, y)$ menunjukkan beberapa properti lokal dari suatu titik seperti nilai rata-rata *gray value* dari lingkungan pusat pada titik (x, y) .

Dalam gambar. 2.2(b) ditunjukkan bahwa hasil pada citra pohon yang berwarna gelap menjadi benar-benar hitam, sedangkan citra salju menjadi berwarna putih.



(a) Citra Awal (*original*) (b) Citra hasil *threshold*

Gambar 2.2 Contoh Penggunaan *Threshold* pada Suatu Citra

Ada dua jenis metode *thresholding*, yaitu:

- a) *Global thresholding*
Suatu nilai *threshold* dikatakan sebagai *global thresholding* ketika nilai T hanya tergantung pada $f(x, y)$. Dengan kata lain, hanya pada nilai-nilai *gray-level*. Dan nilai T semata-mata berkaitan dengan dengan karakter piksel.
- b) *Lokal thresholding*
Suatu nilai *threshold* dikatakan sebagai *lokal thresholding* ketika nilai T hanya tergantung pada $f(x, y)$ dan $p(x, y)$.

Metode ini membagi citra asli menjadi beberapa sub wilayah (*region*), dan memilih berbagai nilai batas ambang T untuk setiap sub *region* yang cukup [8].

2.4 Metode Otsu

Metode Otsu biasanya digunakan untuk menghitung nilai *threshold* secara otomatis, dengan tujuan untuk memisahkan obyek yang dimaksud (*object of interest*) dari latar belakangnya. Metode berbasis histogram ini membagi tingkat abu-abu (*gray level*) dari suatu citra menjadi dua kelas. *Gray level* tersebut terdiri dari berbagai tingkat abu-abu yang bervariasi dari mulai hitam pada intensitas terlemah, hingga putih pada intensitas terkuat, yang berkisar dari 0 sampai 255. Dua kelas yang dimaksud adalah *area of interest* (daerah obyek) dan latar belakang. Dalam hal ini, daerah dengan tingkat abu-abu yang berbeda (puncak pertama) dianggap sebagai *area of interest*, sedangkan daerah dengan tingkat abu-abu yang sama (puncak residual) dianggap sebagai latar belakang. Nilai optimal *threshold* pada suatu citra (gambar) dipilih ketika varians antara dua kelas bernilai maksimum, yang berarti hal tersebut memberikan pemisahan terbaik dari kelas-kelas pada *gray level*nya [3].

Nilai *threshold* (ambang batas) kemudian digunakan untuk mengisolasi *area of interest* (daerah obyek) dari latar belakang dengan mengubah citra (gambar) *grayscale* menjadi citra biner. Citra biner terdiri dari piksel hitam dan putih, dimana piksel dengan tingkat keabuan (*gray level*) yang lebih besar dari nilai *threshold* dianggap sebagai putih dan semua piksel yang lain dianggap sebagai hitam. Citra biner yang dihasilkan adalah citra tersegmentasi, dimana “nilai 1” (putih) merepresentasikan *object of interest* dan “nilai 0” (hitam) merepresentasikan latar belakang. Namun, pada beberapa kasus bisa saja ditemukan bahwa gambar tidak memiliki bentuk yang tepat untuk merepresentasikan wilayah yang dimaksud, hal ini disebabkan karena ketidaksesuaian nilai *threshold* yang dihitung secara otomatis. Keadaan tersebut menunjukkan bahwa metode Otsu yang sudah

ada, tidak dapat melakukan segmentasi pada citra natural secara benar. Oleh karena itu, modifikasi nilai-nilai *threshold* diperlukan untuk mengekstrak daerah tertentu saja (*area of interest*) [3].

Dalam metode Otsu, dilakukan pencarian nilai *threshold* yang meminimalisasi *intra-class variance* (varians dalam kelas), yang didefinisikan sebagai jumlah tertimbang varians dari dua kelas, seperti pada persamaan 2.4 di bawah ini [10].

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \quad \dots (2.4)$$

dimana berat ω_i merupakan probabilitas dari dua kelas yang dipisahkan oleh ambang t dan σ_i^2 merupakan varians dari kelas-kelas tersebut.

Otsu menunjukkan bahwa meminimalkan varians dalam kelas sama dengan memaksimalkan *inter-class variance* (varians antar kelas), seperti yang ditunjukkan pada persamaan 2.5. [10].

(persamaan 2.5)

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t) [\mu_1(t) - \mu_2(t)]^2$$

yang dinyatakan dalam hal kelas probabilitas ω_i dan kelas rata-rata μ_i .

probabilitas kelas $\omega_1(t)$ dihitung dari histogram sebagai t , seperti yang ditunjukkan oleh persamaan 2.6 [10].

$$\omega_1(t) = \sum_0^t p(i) \quad \dots (2.6)$$

Sedangkan kelas rata-rata (*class mean*) $\mu_1(t)$ dihitung seperti pada persamaan 2.7 [10].

$$\mu_1(t) = [\sum_0^t p(i) x(i)]/\omega_1 \quad \dots (2.7)$$

dimana $x(i)$ adalah nilai pusat histogram ke- i . Sehingga dengan demikian kita dapat menghitung $\omega_2(t)$ dan μ_2 pada sisi kanan histogram untuk area (*bin*) yang lebih besar dari t . kelas probabilitas dan kelas rata-rata dapat dihitung secara *iterative* sehingga menghasilkan suatu algoritma yang efektif.

Algoritma metode Otsu ditunjukkan seperti penjelasan di bawah ini [10].

1. Hitung histogram dan probabilitas untuk tiap level intensitas
2. Set up inialisasi $\omega_i(0)$ dan $\mu_i(0)$
3. Lakukan iterasi untuk semua nilai kemungkinan *threshold* dari $t = 1$ hingga intensitas maksimum
 - i. Update ω_i dan μ_i
 - ii. Hitung $\sigma_b^2(t)$
4. Ambang batas (*threshold*) yang diinginkan sesuai dengan maksimal $\sigma_b^2(t)$
5. Hitung dua maksima (dan dua ambang batas yang sesuai). $\sigma_{b1}^2(t)$ adalah maksima yang lebih besar (*the greater max*) sedangkan $\sigma_{b2}^2(t)$ lebih dari sama dengan maksimum.
6. Hitung nilai *threshold* dengan rumus:

$$\frac{\mathit{threshold}_1 + \mathit{threshold}_2}{2}$$

2.5 TsTN

Bagian ini akan menjelaskan tentang metode TsTN, yaitu teknik segmentasi yang sudah ditingkatkan (*improved segmentation technique*) untuk citra natural. Perbaikan untuk teknik segmentasinya dilakukan dengan memodifikasi algoritma

segmentasi berbasis *threshold* yang digabungkan dengan teknik invers. Metode ini disebut sebagai metode TsTN. Metode dari *improved thresholding-based segmentation technique* ini terbagi menjadi tiga langkah utama, yaitu inialisasi nilai *threshold*, konversi citra *grayscale* menjadi citra biner, dan modifikasi nilai *threshold* yang kemudian digunakan dalam proses segmentasi citra [3].

Proses inialisasi nilai optimal global *threshold* (T) dilakukan dengan menggunakan metode Otsu. Metode Otsu ini diterapkan pada citra untuk mendapatkan nilai awal secara otomatis dan cepat. Nilai awal dihitung untuk memperoleh set area *disjoint* yang sesuai dengan obyek yang dimaksud (*object of interest*) dan latar belakangnya (*background*). Proses ini dilakukan untuk memastikan bahwa setidaknya sebagian dari objek yang diteliti dapat diekstraksi [3].

Konversi dari citra *grayscale* menjadi citra biner dilakukan berdasarkan nilai T , yaitu nilai optimal global *threshold*. Citra biner merepresentasikan pemisahan daerah antara daerah obyek (*object of interest*) dan latar belakang (*background*). Proses transformasi dari citra *grayscale* menjadi citra biner didefinisikan pada Persamaan 2.8 [3],

$$g(x,y) = \begin{cases} 1 \text{ (putih)} & \text{jika } f(x,y) > T \\ 0 \text{ (hitam)} & \text{jika } f(x,y) \leq T \end{cases} \quad \dots (2.8)$$

dimana $g(x, y)$ merupakan nilai piksel dari suatu citra biner dan $f(x, y)$ merupakan nilai piksel dari citra *grayscale*.

Namun, nilai global *threshold* (T) yang dihasilkan oleh Otsu tidak cukup untuk menghasilkan citra tersegmentasi yang baik untuk beberapa obyek. Oleh karena itu, dalam algoritma *improved thresholding-based* (TsTN) ini nilai global *threshold* akan dimodifikasi (T_2). Dalam rangka untuk menghasilkan nilai terbaik dari T_2 , beberapa perbaikan perlu dilakukan pada algoritma segmentasi ini. Pertama, elemen tengah (*middle*

element) dari gambar biner yang dihasilkan harus diperiksa untuk mengidentifikasi nilai pikselnya. Jika nilai piksel tidak '1', maka proses invers diperlukan untuk meng-invers setiap piksel dalam citra, dari nilai '1' ke '0' dan begitu pula sebaliknya, seperti pada persamaan 2.9 [3],

$$g(x,y) = \begin{cases} 1 \text{ (putih)} & \text{jika } g(x,y) = 0 \\ 0 \text{ (hitam)} & \text{jika } g(x,y) = 1 \end{cases} \quad \dots (2.9)$$

dimana $g(x, y)$ merupakan nilai piksel dari citra biner.

Proses invers sering diperlukan untuk beberapa kasus, dimana obyek yang diteliti (*object of interest*) lebih gelap dari latar belakang. Karena citra gelap yang menyembunyikan bentuk obyek tersebut, maka operasi invers diperlukan sehingga latar belakang akan muncul sebagai hitam dan obyek yang diteliti (*object of interest*) muncul sebagai putih.

Selanjutnya, setelah proses invers dilakukan, peningkatan kedua adalah dengan melakukan perluasan daerah tersegmentasi menggunakan modifikasi nilai T untuk memperoleh T_2 yang terbaik. Modifikasi nilai *threshold* dilakukan seperti pada persamaan 2.10 [3],

$$T_2 = \begin{cases} T_2 - 0.01 \text{ (daerah terang)} \\ T_2 + 0.01 \text{ (daerah gelap)} \end{cases} \text{ jika } invers = 0 \\ \begin{cases} T_2 + 0.01 \text{ (daerah terang)} \\ T_2 - 0.01 \text{ (daerah gelap)} \end{cases} \text{ jika } invers = 1 \quad \dots (2.10)$$

dimana T_2 merupakan nilai *threshold* yang sudah dimodifikasi.

Nilai T_2 yang sudah diperbarui ini akan digunakan untuk proses segmentasi selanjutnya. Dan proses segmentasi diiterasi

(dilakukan secara berulang-ulang) hingga menghasilkan citra tersegmentasi yang terbaik [3].

2.6 Fuzzy c-means

Secara tradisional, FCM (Fuzzy c-means) merupakan teknik pengelompokan sepenuhnya tanpa pengawasan (*unsupervised clustering technique*) yang tidak memerlukan data berlabel untuk mengidentifikasi struktur kumpulan data (data set). Namun, teknik ini memiliki beberapa kelemahan. Kelemahan pertama adalah teknik ini gagal memberikan hasil yang berarti pada kasus yang melibatkan data set dengan dimensi yang tinggi (*high dimensional data set*). Kelemahan yang kedua yaitu, teknik FCM sensitif terhadap pusat *cluster* awal (*initial cluster centers*) dimana pusat awal yang tidak akurat dapat menghasilkan (memberikan) hasil klasifikasi yang buruk. Oleh karena itu, teknik FCM dieksekusi dengan pusat *cluster* awal yang diperoleh dari tahap sebelumnya [3].

2.7 MSE (*Mean Square Error*) dan PSNR (*Peak Signal to Noise Ratio*)

Dalam citra digital terdapat suatu standar pengukuran *error* (galat) kualitas citra, yaitu besar PSNR dan MSE.

Tingkat keberhasilan dan performa dari suatu metode *filtering* pada citra dihitung dengan menggunakan *Peak Signal to Noise Ratio* atau biasa disingkat dengan PSNR. Meskipun performa metode *filtering* juga dapat diukur dengan teknik visual (hanya melihat pada citra hasil dan membandingkannya dengan citra yang terdapat *noise*). Namun hasil pengukuran teknik visual setiap orang berbeda-beda. Sehingga MSE dan PSNR merupakan solusi pengukuran performa yang baik [9].

Peak Signal to Noise Ratio (PSNR) adalah sebuah perhitungan yang menentukan nilai dari sebuah citra yang dihasilkan. Nilai PSNR ditentukan oleh besar atau kecilnya nilai MSE yang terjadi pada citra. Semakin besar nilai PSNR, semakin baik pula hasil yang diperoleh pada tampilan citra hasil.

Sebaliknya, semakin kecil nilai PSNR, maka akan semakin buruk pula hasil yang diperoleh pada tampilan citra hasil. Satuan nilai dari PSNR sama seperti MSE, yaitu *decibel* (dB). Jadi hubungan antara nilai PSNR dengan nilai MSE adalah semakin besar nilai PSNR, maka akan semakin kecil nilai MSE-nya. PSNR secara umum digunakan untuk mengukur kualitas pada penyusunan ulang citra. Hal ini lebih mudah didefinisikan dengan *Mean Square Error* (MSE) [9].

Mean Square Error (MSE) adalah kesalahan kuadrat rata-rata. Nilai MSE didapat dengan membandingkan nilai selisih piksel-piksel citra asal dengan citra hasil pada posisi piksel yang sama. Semakin besar nilai MSE, maka tampilan pada citra hasil akan semakin buruk. Sebaliknya, semakin kecil nilai MSE, maka tampilan pada citra hasil akan semakin baik [9].

Misal $I(x,y)$ adalah citra masukan dan $I'(x,y)$ adalah citra keluaran, keduanya memiliki M baris dan N kolom, maka MSE dapat didefinisikan seperti pada persamaan 2.11 [9].

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N [I(x,y) - I'(x,y)]^2 \quad \dots (2.11)$$

Sedangkan rumus menghitung PSNR adalah seperti yang ditunjukkan pada persamaan 2.12 [9].

$$PSNR = 20 \times \log_{10} (255/\sqrt{MSE}) \quad \dots (2.12)$$

dimana,

x = ukuran baris dari citra

y = ukuran kolom dari citra

I = matriks citra awal

I' = matriks citra hasil

2.8 Matlab

Nama MATLAB merupakan singkatan dari *Matrix Laboratory*. MATLAB adalah aplikasi perangkat lunak yang memungkinkan kita untuk melakukan manipulasi data dan visualisasi, analisis citra, perhitungan, matematika dan pemrograman. Aplikasi ini dapat digunakan untuk melakukan tugas-tugas secara sangat sederhana serta canggih [7].

MATLAB adalah bahasa kinerja tinggi untuk komputasi teknis. MATLAB merupakan suatu sistem interaktif dimana data basis elemennya berupa *array* yang tidak memerlukan dimensi. MATLAB dapat mengimpor maupun mengekspor beberapa format gambar seperti BMP (*Microsoft Windows Bitmap*), GIF (*Graphics Interchange File*), HDF (*Hierarchical data Format*), JPEG (*Joint Photographic Experts Group*), PCX (*Paintbrush*), PNG (*Portable Network Graphics*), TIFF (*Tagged Image file Format*), XWD (*X Window Dump*). MATLAB juga dapat memuat data mentah (*raw data*) atau jenis data citra yang lain [7].

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini menguraikan perancangan proses dan perancangan perangkat lunak implementasi metode *thresholding* untuk segmentasi citra natural. Perancangan yang diuraikan meliputi arsitektur sistem secara umum, perancangan data, perancangan proses, dan perancangan antarmuka yang diimplementasikan pada perangkat lunak.

3.1 Analisis

Subbab ini menguraikan secara umum arsitektur sistem perangkat lunak implementasi metode *thresholding* untuk segmentasi citra natural. Arsitektur sistem yang diuraikan meliputi deskripsi umum perangkat lunak dan lingkungan perancangan perangkat lunak.

3.1.1 Cakupan Permasalahan

Perangkat lunak ini harus mampu mengimplementasikan metode *thresholding* untuk segmentasi citra natural, menggunakan algoritma *Otsu method* yang diintegrasikan dengan teknik *inverse*, disebut sebagai algoritma TsTN (*an improved Thresholding-based segmentation Technique for Natural images*). Selain itu, perangkat lunak harus mampu menampilkan hasil segmentasi citra secara mudah untuk pengguna. Demi memenuhi kebutuhan tersebut, maka perangkat lunak dibagi menjadi dua proses utama, yaitu:

1. Metode segmentasi *thresholding-based*
Proses ini mengimplementasikan metode *thresholding-based* untuk proses segmentasi citra natural.
Proses yang dilakukan adalah preproses input citra natural menjadi citra *grayscale*, pengolahan segmentasi menggunakan metode *thresholding* (metode Otsu), dan

mengubah citra *grayscale* menjadi citra biner (*binary image*).

2. TsTN

Proses ini mengimplementasikan metode TsTN untuk proses segmentasi citra natural. Metode ini melakukan segmentasi dengan pendekatan secara *thresholding-based* (menggunakan metode Otsu) sebagai langkah awal, kemudian menggabinkannya dengan *inverse* untuk melakukan proses segmentasi selanjutnya. Proses yang dilakukan adalah proses segmentasi pada citra natural.

3.1.2 Lingkungan Perancangan Perangkat Lunak

Proses perancangan perangkat lunak Tugas Akhir ini membutuhkan lingkungan perancangan yang sesuai. Tabel 3.1 menjelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam perancangan perangkat lunak segmentasi citra natural menggunakan teknik berbasis *thresholding*.

Tabel 3.1 Lingkungan Perancangan Perangkat Lunak

Perangkat	Spesifikasi
Perangkat Keras	1. Prosesor : Intel(R) Celeron(R) M CPU 440 @1.86GHz 2. Memori (RAM) : 1.50 GB
Perangkat Lunak	1. Sistem Operasi : Microsoft Windows Vista Ultimate 32-bit 2. Perangkat Pengembang : Matlab R2010b

3.2 Perancangan Data

Perancangan data merupakan hal penting untuk diperhatikan. Perangkat lunak memerlukan data yang tepat agar beroperasi dengan baik dan benar. Data yang dibutuhkan dalam membangun perangkat lunak untuk proses implementasi terdiri atas data masukan dan data keluaran. Data masukan berupa dataset yang berisi citra natural. Data keluaran berupa citra yang sudah tersegmentasi.

3.2.1 Data Masukan

Data masukan adalah data-data yang digunakan sebagai masukan dari aplikasi. Data-data tersebut kemudian akan diolah oleh aplikasi melalui langkah-langkah tertentu hingga menghasilkan keluaran yang diinginkan. Data masukan yang digunakan dalam aplikasi adalah data *real* berupa empat citra *jatropha fruit* yang diambil secara langsung pada lingkungan natural. Data masukan ini dipilih karena citra memiliki perbedaan warna permukaan yang menandakan tingkat kematangan pada level-level yang berbeda. Tabel 3.2 menjelaskan empat kategori citra *jatropha fruit* yang digunakan sebagai data masukan. Data-data tersebut dipilih dengan kecenderungan warna permukaan yang berbeda-beda, terdiri dari warna hijau, kuning, kuning kecoklatan dan hitam.

Tabel 3.2 Set Data Citra *Jatropha Fruit*

Citra Natural	Warna Permukaan
	Hijau
	Kuning
	Kuning Kecoklatan
	Hitam

Semua citra yang menjadi data masukan diubah ukurannya menjadi 250 x 250 piksel. Pengaturan ukuran piksel ini dilakukan untuk mengurangi waktu pemrosesan komputasi dan agar data masukan memiliki nilai intensitas piksel standar untuk semua citra.

3.2.2 Data Keluaran

Data keluaran merupakan data yang dihasilkan oleh aplikasi setelah proses-proses tertentu selesai dijalankan. Data keluaran dari proses uji menggunakan algoritma segmentasi *improved thresholding-based* (TsTN) ini adalah citra yang telah tersegmentasi.

3.3 Perancangan Proses

Perancangan proses dilakukan untuk mengetahui alur dalam penerapan algoritma yang nantinya akan dipakai dalam tahap implementasi.

3.3.1 Perancangan Proses Secara Umum

Tugas Akhir ini dirancang untuk dapat menyelesaikan problem segmentasi citra natural dengan dua metode segmentasi yang berbeda, yaitu metode segmentasi *thresholding-based* (metode Otsu) dan TsTN (*improved thresholding-based* yang diintegrasikan dengan teknik *inverse*). Selain *improved thresholding-based* (TsTN), metode segmentasi *thresholding-based* Otsu juga perlu menjadi *feature* dalam perangkat lunak Tugas Akhir ini. Selain untuk memudahkan pengguna membandingkan secara langsung hasil kinerjanya (berupa akurasi) dengan metode *improved thresholding-based* (TsTN), metode segmentasi *thresholding-based* Otsu merupakan metode yang terlibat secara langsung dalam proses segmentasi sebagai model segmentasi dasar dalam metode *improved thresholding-based* (TsTN) ini.

a. Akuisisi Citra

Secara umum, masukan awal dari sistem segmentasi ini adalah sejumlah data citra natural. Data masukan awal ini berupa citra *jatropha fruit* seperti yang ditampilkan pada Tabel. 3.2. Citra *jatropha fruit* ini diambil dari lingkungan natural dengan tujuan untuk mendapatkan data *real*. Data masukan ini dipilih karena citra memiliki perbedaan warna permukaan yang menandakan tingkat kematangan pada level-level yang berbeda. Data-data citra tersebut dipilih dengan kecenderungan warna permukaan yang berbeda-beda pada tiap citra, yang terdiri dari warna hijau, kuning, kuning kecoklatan dan hitam.

b. Preproses Citra

Semua citra yang menjadi data masukan diubah ukurannya menjadi 250 x 250 piksel. Pengaturan ukuran piksel ini dilakukan untuk mengurangi waktu pemrosesan komputasi dan agar data masukan memiliki nilai intensitas piksel standar untuk semua citra.

c. Segmentasi Citra

Segmentasi citra mengacu pada proses partisi citra (gambar) digital menjadi beberapa segmen (bagian) atau daerah, yang bertujuan untuk menyederhanakan representasi dari suatu citra (gambar). Segmentasi wilayah yang dipilih memastikan bahwa hanya obyek yang dimaksud (*object of interest*) yang akan diproses selama tahap analisis warna. Dalam Tugas Akhir ini, pemisahan antara citra *jatropha fruit* dan latar belakangnya dilakukan dengan menggunakan dua metode segmentasi, yaitu metode Otsu yang merupakan metode segmentasi *thresholding-based* dan metode TsTN yang merupakan pengembangan dari metode segmentasi *thresholding-based (improved thresholding-based)*.

Secara umum, masukan awal dari sistem segmentasi ini adalah sejumlah data citra natural. Model untuk *improved thresholding-based* (TsTN) ini harus memenuhi persamaan yang telah dibahas pada Bab II.

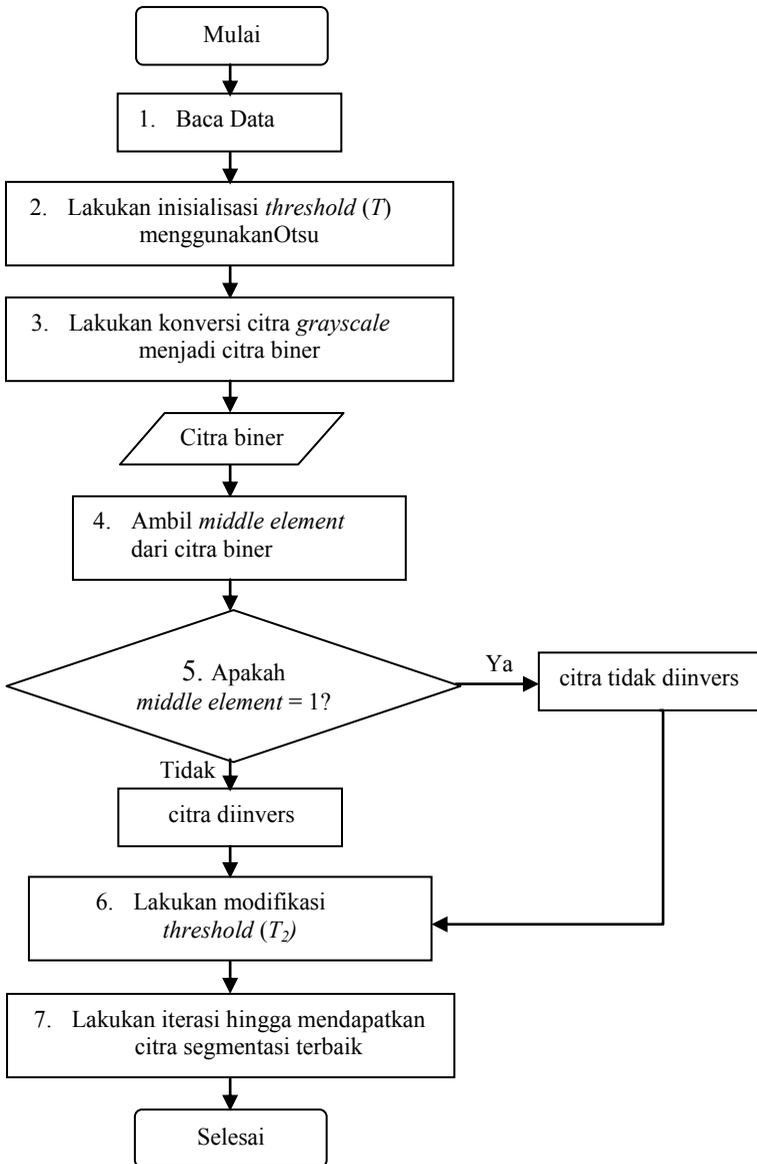
Metode TsTN yang digunakan pada Tugas Akhir ini mengacu kepada persamaan 2.8 sampai persamaan 2.10. Proses segmentasi dengan menggunakan metode *thresholding-based* Otsu mengacu kepada persamaan 2.2, dan juga persamaan 2.4 sampai dengan persamaan 2.7. Proses segmentasi metode *improved thresholding-based* yang diintegrasikan dengan teknik *inverse* (TsTN) mengacu kepada persamaan 2.8 sampai dengan persamaan 2.10.

3.3.2 Otsu

Metode segmentasi Otsu akan digunakan pada metode *improved thresholding-based* ini sebagai inisialisasi nilai global *threshold* (T). Dalam implementasi Otsu, diterapkan sejumlah persamaan yang telah disebutkan pada Bab II, mulai dari persamaan 2.4 sampai dengan persamaan 2.7.

3.3.3 TsTN

Bagian ini akan menjelaskan tentang metode TsTN, yaitu teknik segmentasi yang sudah ditingkatkan (*improved segmentation technique*) untuk citra natural. Perbaikan untuk teknik segmentasinya dilakukan dengan memodifikasi algoritma segmentasi berbasis *threshold* yang digabungkan dengan teknik *invers*. Metode ini disebut sebagai metode TsTN, seperti yang ditunjukkan pada Gambar 3.1



Gambar 3.1 Diagram Alir Proses TsTN

Metode dari *improved thresholding-based segmentation technique* ini terbagi menjadi tiga langkah utama, yaitu inialisasi nilai *threshold*, konversi citra *grayscale* menjadi citra biner, dan modifikasi nilai *threshold* yang kemudian digunakan dalam proses segmentasi citra.

Pada *step* 1, dilakukan proses membaca data input yang berupa citra (gambar) natural RGB 250 x 250 piksel. Selanjutnya pada *step* 2 dilakukan proses inialisasi nilai optimal global *threshold* (T). Proses inialisasi nilai *threshold* ini dilakukan dengan menggunakan metode Otsu. Metode Otsu yang diterapkan pada citra dilakukan untuk mendapatkan nilai awal secara otomatis dan cepat. Nilai awal dihitung untuk memperoleh set area *disjoint* yang sesuai dengan obyek yang dimaksud (*object of interest*) dan latar belakangnya (*background*). Proses ini dilakukan untuk memastikan bahwa setidaknya sebagian dari obyek yang diteliti dapat diekstraksi.

Step 3 menjelaskan tentang konversi citra input dari citra *grayscale* menjadi citra biner yang dilakukan berdasarkan nilai T , yaitu nilai optimal global *threshold*. Citra biner merepresentasikan pemisahan daerah antara daerah obyek (*object of interest*) dan latar belakang (*background*). Proses transformasi dari citra *grayscale* menjadi citra biner didefinisikan pada Persamaan 2.8.

Namun, nilai global *threshold* (T) yang dihasilkan oleh Otsu tidak cukup untuk menghasilkan citra tersegmentasi yang baik untuk beberapa obyek. Oleh karena itu, dalam algoritma *improved thresholding-based* (TsTN) ini nilai global *threshold* akan dimodifikasi (T_2). Dalam rangka untuk menghasilkan nilai terbaik dari T_2 , beberapa perbaikan perlu dilakukan pada algoritma segmentasi ini seperti yang ditunjukkan pada *step* 4 dan 5. Pertama, elemen tengah (*middle element*) dari gambar biner yang dihasilkan harus diperiksa untuk mengidentifikasi nilai

pikselnya. Jika nilai piksel tidak '1', maka proses invers diperlukan untuk meng-invers setiap piksel dalam citra, dari nilai '1' ke '0' dan begitu pula sebaliknya, seperti pada persamaan 2.9.

Proses invers sering diperlukan untuk beberapa kasus, dimana obyek yang diteliti (*object of interest*) lebih gelap dari latar belakang. Karena citra gelap yang menyembunyikan bentuk obyek tersebut, maka operasi invers diperlukan sehingga latar belakang akan muncul sebagai hitam dan obyek yang diteliti (*object of interest*) muncul sebagai putih.

Selanjutnya, setelah proses invers dilakukan, peningkatan kedua adalah dengan melakukan perluasan daerah tersegmentasi menggunakan modifikasi nilai T untuk memperoleh T_2 yang terbaik. Modifikasi nilai *threshold* dilakukan seperti pada persamaan 2.10. Nilai T_2 yang sudah diperbarui ini akan digunakan untuk proses segmentasi selanjutnya. Dan proses segmentasi diiterasi (dilakukan secara berulang-ulang) hingga menghasilkan citra tersegmentasi yang terbaik.

BAB IV IMPLEMENTASI SISTEM

Bab ini akan menguraikan mengenai implementasi perangkat lunak yang meliputi lingkungan implementasi, potongan kode program untuk tiap proses atau fungsi, dan implementasi antarmuka. Tahap implementasi dari tiap fungsi akan menjelaskan mengenai parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan kode program dan teori.

4.1 Lingkungan Pengembangan Sistem

Proses implementasi perangkat lunak Tugas Akhir ini membutuhkan lingkungan implementasi yang sesuai. Tabel 4.1 menjelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam implementasi perangkat lunak.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

Perangkat	Spesifikasi
Perangkat Keras	<ol style="list-style-type: none">1. Prosesor : Intel(R) Celeron(R) M CPU 440 @1.86GHz2. Memori (RAM) : 1.50 GB
Perangkat Lunak	<ol style="list-style-type: none">1. SistemOperasi : Microsoft Windows Vista Ultimate 32-bit2. Perangkat Pengembang : Matlab R2010b

4.2 Implementasi Otsu

Fungsi ini merupakan inialisasi nilai *threshold* awal menggunakan metode Otsu. Fungsi ini digunakan untuk menentukan nilai *threshold* awal yang akan dimodifikasi pada metode TsTN.

Potongan kode program pada gambar 4.1 digunakan pada inialisasi nilai *threshold* awal.

- | | |
|----------------------|--------------------------------------|
| 1. Nama berkas | : Tstn.m |
| Nama fungsi | : Tstn |
| Data masukan | |
| img | : citra natural 250 x 250 piksel RGB |
| Data keluaran | |
| T | : nilai <i>threshold</i> awal |

```
function [simg] = Tstn(img)
    grey = rgb2gray(img); %mengubah gambar
    berwarna jadi grayscale
    T = graythresh(grey); %memanggil fungsi otsu
    simg = im2bw(grey,T); %mengubah gambar
    grayscale jadi binary
```

Gambar 4.1 Kode Program Proses Inialisasi Otsu

4.3 Implementasi TsTN

Selanjutnya adalah membuat modul untuk TsTN. Modul TsTN ini mengimplementasikan diagram alir pada Bab 3. Modul ini dicantumkan dalam tiga potongan kode program. Potongan kode program pada gambar 4.2 digunakan pada proses modifikasi nilai *threshold* (T_2) pada metode TsTN. Potongan kode program pada gambar 4.3 digunakan pada proses membangun fungsi hitung piksel untuk modifikasi *threshold*. Potongan kode program pada gambar 4.4 digunakan pada proses segmentasi TsTN.

2. Nama berkas : find_treshold.m

Nama fungsi : find_treshold

Data masukan

img : citra natural 250 x 250 piksel RGB

margin : piksel yang menjadi middle element

T : nilai *threshold* awal

Data keluaran

T2 : modifikasi *threshold* baru

```
function [T2] = find_treshold(img,margin,T)
%margin utk menentukan middle element (?)
    inverse = 0;

    for i = (margin+1):(250-margin)
        for j = (margin+1):(250-margin)
            if img(i,j)==0
                inverse = 1;
            end
        end
    end

    if inverse == 1
        for i=1:250
            for j=1:250
                if img(i,j)==0
                    img(i,j)=1;
                else
                    img(i,j)=0;
                end
            end
        end
    end

    area = hitung_piksel(img);

    if inverse == 0
        if area == 1
```

```

        T2 = T - 0.01;
    else
        T2 = T + 0.01;
    end
else
    if area == 1
        T2 = T+0.01;
    else
        T2 = T-0.01;
    end
end
end
end

```

Gambar 4.2 Kode Program Proses Modifikasi *Threshold* Pada TsTN

Potongan kode program pada gambar 4.3 digunakan pada proses membangun fungsi hitung piksel untuk modifikasi *threshold*.

3. Nama berkas : hitung_piksel.m

Nama fungsi : hitung_piksel

Data masukan

img : citra natural 250 x 250 piksel RGB

atas : banyaknya daerah terang pada *middle element*

bawah : banyaknya daerah gelap pada *middle element*

Data keluaran

area : daerah piksel

```

function [area] = hitung_piksel(img)
    atas = 0;
    bawah = 0;
    area = 0;
    for i=1:250
        for j=1:250
            if img(i,j)==1
                atas = atas + 1;
            end
        end
    end
end

```

```

        else
            bawah = bawah + 1;
        end
    end
end
if atas > bawah
    area = 1;
end
end

```

Gambar 4.3 Kode Program Fungsi Hitung Pixel Untuk Modifikasi *Threshold*

Potongan kode program pada gambar 4.4 digunakan pada proses Segmentasi TsTN.

4. Nama berkas : segmentasi.m

Nama fungsi : segmentasi

Data masukan

img : citra natural 250 x 250 piksel RGB

margin : piksel yang menjadi middle element

T : nilai *threshold* awal

gtruth : citra ground truth

n : banyaknya jumlah iterasi yang dilakukan

Data keluaran

T2 : modifikasi *threshold* baru

Mse : nilai MSE

PSNR : nilai PSNR

```

function [T2,mse, psnr, citra] =
segmentasi(gtruth,margin,img,n)

    T2 = zeros(n,1); %matriks untuk menyimpan T2
    di setiap iterasi
    mse = zeros(n,1); %matriks untuk menyimpan
    mse di setiap iterasi
    psnr = zeros(n,1); %matriks untuk menyimpan
    psnr di setiap iterasi
    gtruth = im2bw(gtruth,0.5); %mengubah gambar
    groundtruth dari grey menjadi biner
    grey = rgb2gray(img); %mengubah gambar
    berwarna jadi grayscale
    T = graythresh(grey); %memanggil fungsi otsu
    citra = im2bw(grey,T); %mengubah gambar
    grayscale jadi binary
    T2(1)=T;
    [mse(1), psnr(1)]= eval_msepsnr(gtruth,
    citra);
    for i=2:n
        T2(i) = find_treshold(citra,margin,T2(i-
    1));%fungsi untuk update treshold
        citra = im2bw(grey,T2(i)); %update citra
        sesuai treshold yang dihasilkan
        [mse(i), psnr(i)]= eval_msepsnr(gtruth,
    citra);%fungsi untuk evaluasi
    end
end

```

Gambar 4.4 Kode Program Fungsi Segmentasi TsTN

BAB V

PENGUJIAN DAN EVALUASI

Bab ini akan menjelaskan rangkaian uji coba dan analisis yang dilakukan. Pembahasan yang dikemukakan meliputi lingkungan uji coba, data uji coba, skenario uji coba, hasil uji coba, dan analisis hasil uji coba.

5.1 Lingkungan Pengujian

Lingkungan uji coba yang digunakan dalam pembuatan Tugas Akhir ini meliputi perangkat keras dan perangkat lunak yang digunakan untuk proses segmentasi citra natural dengan metode *improved thresholding-based segmentation*. Tabel 5.1 menjelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam uji coba perangkat lunak.

Tabel 5.1 Lingkungan Uji Coba Perangkat Lunak

Perangkat	Spesifikasi
Perangkat Keras	<ol style="list-style-type: none">1. Prosesor : Intel(R) Celeron(R) M CPU 440 @1.86GHz2. Memori (RAM) : 1.50 GB
Perangkat Lunak	<ol style="list-style-type: none">1. Sistem Operasi : Microsoft Windows Vista Ultimate 32-bit2. Perangkat Pengembang : Matlab R2010b

5.2 Data Uji Coba

Uji coba ini menggunakan 4 macam citra *Jatropha Fruit*. Bab 3 telah menjelaskan kategori citra yang dipilih dengan kecenderungan warna permukaan yang berbeda-beda untuk set data tersebut.

5.3 Skenario Pengujian

Bagian ini menjelaskan skenario uji coba yang telah dilakukan. Skenario uji coba yang dilakukan adalah dengan melakukan segmentasi citra menggunakan dua metode yang berbeda, yaitu metode Otsu dan metode TsTN. Metode Otsu yang digunakan berfungsi sebagai inisialisasi nilai *threshold* awal yang akan digunakan dalam proses iterasi selanjutnya untuk menentukan nilai modifikasi *threshold* (T_2). Proses iterasi dilakukan sebanyak 10 kali, lalu dilakukan pengecekan nilai *error* menggunakan MSE (*Mean Squared Error*) dan PSNR (*Peak Signal to Noise Ratio*) untuk mengetahui hasil iterasi ke berapakah yang memiliki tingkat keberhasilan segmentasi paling tinggi dari metode TsTN ini.

Data latih yang digunakan merupakan sebuah set data yang telah didefinisikan sebelumnya. Data tersebut dipilih dengan kecenderungan warna permukaan yang berbeda-beda yang menandakan tingkat kematangan pada level-level yang berbeda. Uji coba dan analisis hasil uji coba akan dijelaskan lebih lanjut.

5.3.1 Uji Coba dengan Perbandingan Beberapa Metode Segmentasi

Uji coba dilakukan dengan menyegmentasi citra menggunakan dua metode yang berbeda, yaitu metode Otsu dan metode TsTN.

Data yang diuji adalah empat macam citra *jatropha fruit* dengan intensitas warna yang berbeda-beda. Uji coba

menggunakan nilai margin (*middle element*) sebesar 10. Uji coba dilakukan sebanyak sepuluh kali iterasi untuk metode TsTN. Hasil segmentasi terbaik yang akan diambil sebagai hasil untuk metode TsTN adalah hasil segmentasi yang tidak berbeda jauh dari citra *ground truth*.

5.3.2 Hasil Uji coba beberapa metode sesgmentasi

Uji coba pada skenario ini digunakan untuk mengetahui metode terbaik untuk segmentasi citra natural.

a. Citra berwarna hijau

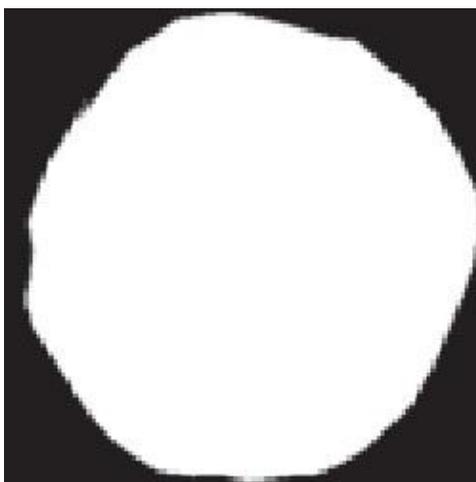
Gambar 5.1. merupakan gambar citra natural dari *jatropha fruit* dengan kecenderungan warna permukaan hijau. Citra natural yang berfungsi sebagai data input ini berukuran 250 x 250 piksel. Gambar 5.2. merupakan gambar citra *ground truth* dari *jatropha fruit*. Citra *ground truth* pada gambar 5.2. ini berasal dari citra natural dengan kecenderungan warna hijau. Data citra *ground truth* ini didapat dari proses manual segmentasi dengan menebalkan tepi obyek sehingga memudahkan dalam pemberian warna pembedanya, dimana obyek (*object of interest*) diberi warna putih sedangkan latar belakangnya (*background*) diberi warna hitam. Proses manual segmentasi seperti pada gambar 5.2 dilakukan dengan bantuan aplikasi *Paint*.

Gambar 5.3. merupakan gambar dari citra *jatropha fruit* dengan kecenderungan warna permukaan hijau, yang telah mengalami proses segmentasi dengan menggunakan metode Otsu. Citra pada gambar 5.3. ini telah mengalami proses konversi menjadi citra biner berukuran 250 x 250 piksel. Gambar 5.4. merupakan gambar dari citra *jatropha fruit* dengan kecenderungan warna permukaan hijau, yang telah mengalami proses segmentasi dengan menggunakan metode TsTN. Citra pada gambar 5.4. ini telah mengalami proses konversi menjadi citra biner berukuran 250 x 250 piksel. Berdasarkan perbandingan gambar secara kasat mata, citra *output* menggunakan metode TsTN seperti yang ditunjukkan pada gambar 5.4. memberikan hasil segmentasi yang lebih baik

daripada citra *output* segmentasi menggunakan metode Otsu seperti yang ditunjukkan pada gambar 5.3. Pada gambar 5.3., citra obyek (*object of interest*) hasil segmentasi menggunakan metode Otsu berbentuk seperti kacang mede. Hasil ini berbeda jauh dengan citra natural yang menjadi *inputnya*, maupun citra *ground truth* segmentasi manual yang berbentuk lingkaran. Perbedaan hasil ini disebabkan sebagian wilayah gelap pada obyek (*object of interest*) dikenali oleh Otsu sebagai area latar belakang (*background*). Perbaikan yang dilakukan oleh metode TsTN dapat mengatasi masalah ini. Hal ini ditunjukkan oleh gambar 5.4. dimana hasil segmentasi memberikan *output* citra obyek (*object of interest*) berbentuk lingkaran menyerupai citra *ground truth*. Proses iterasi untuk menghasilkan citra seperti pada gambar 5.4. dilakukan sebanyak sepuluh kali.



Gambar 5.1 Citra *Jatropha Fruit Hijau* Natural



Gambar 5.2 Citra *Jatropha Fruit Hijau* Ground Truth



Gambar 5.3 Citra *Jatropha Fruit* Hijau Segmentasi Otsu



Gambar 5.4 Citra *Jatropha Fruit* Hijau Segmentasi TsTN

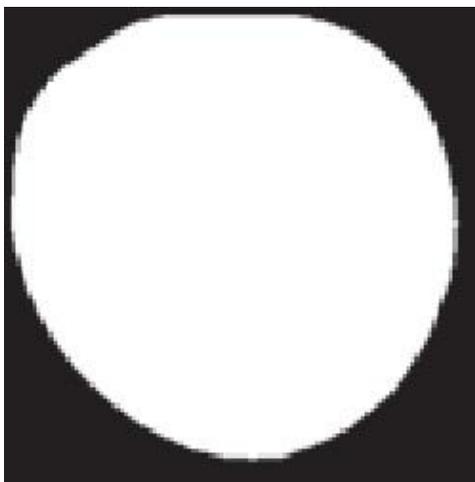
b. Citra berwarna kuning

Gambar 5.5. merupakan gambar citra natural dari *jatropha fruit* dengan kecenderungan warna permukaan kuning. Citra natural yang berfungsi sebagai data input ini berukuran 250 x 250 piksel. Gambar 5.6. merupakan gambar citra *ground truth* dari *jatropha fruit*. Citra *ground truth* pada gambar 5.6. ini berasal dari citra natural dengan kecenderungan warna kuning. Data citra *ground truth* ini didapat dari proses manual segmentasi dengan menebalkan tepi obyek sehingga memudahkan dalam pemberian warna pembedanya, dimana obyek (*object of interest*) diberi warna putih sedangkan latar belakangnya (*background*) diberi warna hitam. Proses manual segmentasi seperti pada gambar 5.6 dilakukan dengan bantuan aplikasi *Paint*.

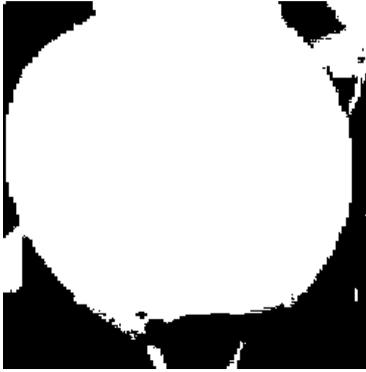
Gambar 5.7. merupakan gambar dari citra *jatropha fruit* kuning, yang telah mengalami proses segmentasi menggunakan metode Otsu. Citra pada gambar 5.7. ini telah mengalami proses konversi menjadi citra biner berukuran 250 x 250 piksel. Gambar 5.8. merupakan gambar dari citra *jatropha fruit* dengan segmentasi TsTN. Berdasarkan perbandingan gambar secara kasat mata, citra *output* dengan metode TsTN seperti pada gambar 5.8. memberikan hasil segmentasi yang lebih baik daripada citra *output* segmentasi menggunakan metode Otsu seperti yang ditunjukkan pada gambar 5.7. Pada gambar 5.7., citra hasil segmentasi Otsu masih mengalami beberapa kerancuan piksel latar belakang yang dianggap sebagai obyek (*object of interest*), sehingga menyebabkan latar belakang dengan intensitas cahaya yang terang ikut tersegmentasi sebagai obyek. Hal ini dapat dilihat dari banyaknya warna putih yang terdapat di luar obyek berbentuk lingkaran. Hal berbeda dapat dilihat pada gambar 5.8. dimana perbaikan telah dilakukan oleh metode TsTN sehingga latar belakang lebih banyak didominasi oleh warna hitam, seperti citra *ground truth*. Proses iterasi untuk menghasilkan citra seperti pada gambar 5.4. dilakukan sebanyak sepuluh kali.



Gambar 5.5 Citra *Jatropha Fruit* Kuning Natural



Gambar 5.6 Citra *Jatropha Fruit* Kuning Ground Truth



Gambar 5.7 Citra *Jatropha Fruit* Kuning Segmentasi Otsu

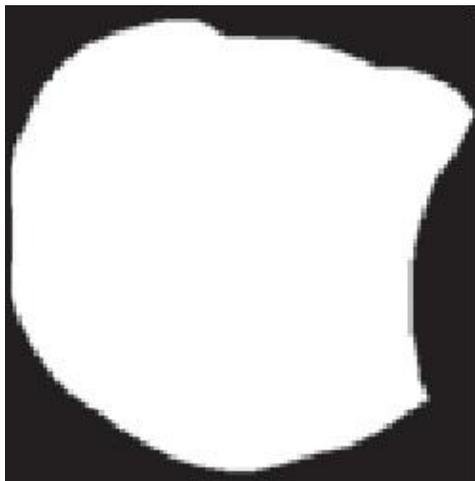


Gambar 5.8 Citra *Jatropha Fruit* Kuning Segmentasi TsTN

c. Citra berwarna kuning kecoklatan



Gambar 5.9 Citra *Jatropha Fruit* Kuning Kecoklatan Natural



**Gambar 5.10 Citra *Jatropha Fruit* Kuning Kecoklatan
*Ground Truth***



Gambar 5.11 Citra *Jatropha Fruit* Kuning Kecoklatan
Segmentasi Otsu



Gambar 5.12 Citra *Jatropha Fruit* Kuning Kecoklatan
Segmentasi TsTN

Gambar 5.9. merupakan gambar citra natural dari *jatropha fruit* dengan kecenderungan warna permukaan kuning kecoklatan. Citra natural yang berfungsi sebagai data input ini berukuran 250 x 250 piksel. Gambar 5.10. merupakan gambar citra *ground truth* dari *jatropha fruit*. Citra *ground truth* pada gambar 5.10. ini berasal dari citra natural dengan kecenderungan warna kuning kecoklatan. Data citra *ground truth* ini didapat dari proses manual segmentasi dengan menebalkan tepi obyek sehingga memudahkan dalam pemberian warna pembedanya, dimana obyek (*object of interest*) diberi warna putih sedangkan latar belakangnya (*background*) diberi warna hitam. Proses manual segmentasi seperti pada gambar 5.10. dilakukan dengan bantuan aplikasi *Paint*.

Gambar 5.11. merupakan gambar dari citra *jatropha fruit* kuning kecoklatan, dengan segmentasi Otsu. Citra pada gambar 5.11. ini telah mengalami proses konversi menjadi citra biner berukuran 250 x 250 piksel. Gambar 5.12. merupakan gambar dari citra *jatropha fruit* dengan segmentasi TsTN. Berdasarkan perbandingan gambar secara kasat mata, citra *output* dengan metode TsTN seperti pada gambar 5.12 memberikan hasil segmentasi yang lebih baik daripada citra *output* segmentasi menggunakan metode Otsu seperti yang ditunjukkan pada gambar 5.11. Pada gambar 5.11., citra hasil segmentasi Otsu masih mengalami beberapa kerancuan piksel obyek (*object of interest*), untuk intensitas cahaya yang terang ikut tersegmentasi sebagai latar belakang (*background*). Hal ini dapat dilihat dari banyaknya warna putih yang terdapat di dalam obyek berbentuk lingkaran. Hal berbeda dapat dilihat pada gambar 5.12. dimana perbaikan telah dilakukan oleh metode TsTN sehingga obyek (*object of interest*) lebih banyak didominasi oleh warna hitam. Proses iterasi untuk menghasilkan citra seperti pada gambar 5.4. dilakukan sebanyak sepuluh kali.

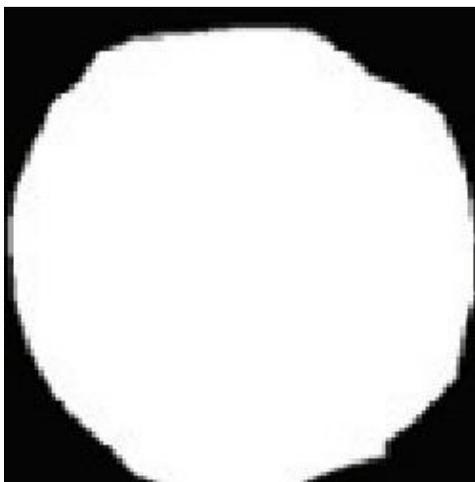
d. Citra berwarna hitam

Gambar 5.13. merupakan gambar citra natural dari *jatropha fruit* dengan kecenderungan warna permukaan hitam. Citra natural yang berfungsi sebagai data input ini berukuran 250 x 250 piksel. Gambar 5.14. merupakan gambar citra *ground truth* dari *jatropha fruit*. Citra *ground truth* pada gambar 5.14. ini berasal dari citra natural dengan kecenderungan warna hitam. Data citra *ground truth* ini didapat dari proses manual segmentasi dengan menebalkan tepi obyek sehingga memudahkan dalam pemberian warna pembedanya, dimana obyek (*object of interest*) diberi warna putih sedangkan latar belakangnya (*background*) diberi warna hitam. Proses manual segmentasi pada gambar 5.14. dilakukan dengan bantuan aplikasi *Paint*.

Gambar 5.15. merupakan gambar dari citra *jatropha fruit* dengan kecenderungan permukaan berwarna hitam, yang telah mengalami proses segmentasi menggunakan metode Otsu. Citra pada gambar 5.15. ini telah mengalami proses konversi menjadi citra biner berukuran 250 x 250 piksel. Gambar 5.16. merupakan gambar dari citra *jatropha fruit* dengan segmentasi TsTN. Berdasarkan perbandingan gambar secara kasat mata, citra *output* dengan metode TsTN seperti pada gambar 5.16 memberikan hasil segmentasi yang lebih baik daripada citra *output* segmentasi menggunakan metode Otsu seperti yang ditunjukkan pada gambar 5.15. Pada gambar 5.15., citra hasil segmentasi Otsu masih mengalami beberapa kerancuan piksel obyek (*object of interest*), untuk intensitas cahaya yang terang ikut tersegmentasi sebagai latar belakang (*background*). Hal ini dapat dilihat dari masih adanya warna putih yang terdapat di dalam obyek berbentuk lingkaran. Hal berbeda dapat dilihat pada gambar 5.16. dimana perbaikan telah dilakukan oleh metode TsTN sehingga obyek (*object of interest*) lebih banyak didominasi oleh warna hitam. Proses iterasi untuk menghasilkan citra seperti pada gambar 5.16. dilakukan sebanyak sepuluh kali.



Gambar 5.13 Citra *Jatropha Fruit Hitam* Natural



Gambar 5.14 Citra *Jatropha Fruit Hitam* Ground Truth



Gambar 5.15 Citra *Jatropha Fruit* Hitam Segmentasi Otsu



Gambar 5.16 Citra *Jatropha Fruit* Hitam Segmentasi TsTN

5.3.3 Uji coba dengan perbandingan nilai parameter MSE dan PSNR

Uji coba dilakukan dengan menghitung nilai *error* data tiap iterasi. Nilai MSE merupakan jumlah minimum *error* yang terdapat pada metode TsTN. Semakin rendah nilai MSE suatu citra tersegmentasi, menunjukkan bahwa hasil segmentasi tersebut semakin baik, sedangkan semakin tinggi nilai PSNR suatu citra tersegmentasi, maka hal tersebut menunjukkan bahwa hasil segmentasi yang dilakukan semakin baik. Pada Tugas Akhir ini, proses segmentasi awal yang digunakan adalah segmentasi dengan menggunakan metode Otsu. Parameter awal yang dijadikan acuan ialah nilai MSE dan PSNR yang dihasilkan oleh metode Otsu. Metode yang menjadi pembandingnya ialah metode segmentasi TsTN. Uji coba menggunakan metode TsTN ini dilakukan sebanyak sepuluh kali.

5.3.4 Hasil Ujicoba MSE dan PSNR

Uji coba pada skenario ini digunakan untuk mengetahui nilai MSE terbaik untuk metode TsTN. Nilai MSE terbaik menunjukkan jumlah minimal *error* yang dihasilkan dari proses segmentasi. Nilai MSE terkecil dari keseluruhan iterasi yang dilakukan menunjukkan banyaknya jumlah iterasi yang diperlukan untuk menghasilkan *output* terbaik.

Hasil uji coba pada skenario ini ditunjukkan pada tabel 5.2 sampai dengan tabel 5.5. Tabel 5.2 menunjukkan performa parameter MSE dan PSNR untuk segmentasi citra *jatropha fruit* warna hijau. Tabel 5.3 menunjukkan performa parameter MSE dan PSNR untuk segmentasi citra *jatropha fruit* warna kuning. Tabel 5.4 menunjukkan performa parameter MSE dan PSNR untuk segmentasi citra *jatropha fruit* warna kuning kecoklatan. Tabel 5.5 menunjukkan performa parameter MSE dan PSNR untuk segmentasi citra *jatropha fruit* warna hitam. Uji coba dilakukan sebanyak sepuluh kali untuk tiap citra. Performa terbaik yang akan diambil sebagai hasil untuk tiap citra adalah proses iterasi

yang memberikan nilai MSE paling kecil dari sepuluh hasil uji coba tersebut.

Tabel 5.2 Performa Parameter MSE dan PSNR untuk Segmentasi Citra *Jatropha Fruit* Hijau

Iterasi	<i>Threshold</i> (T_2)	MSE	PSNR
	<i>T = 0.5373 (Otsu)</i>	0.2404	54.3563
1	0.5273	0.2301	54.5457
2	0.5173	0.2191	54.7583
3	0.5073	0.2115	54.9119
4	0.4973	0.2011	55.1313
5	0.4873	0.1935	55.2983
6	0.4773	0.1881	55.4210
7	0.4673	0.1864	55.4603
8	0.4573	0.1856	55.4794
9	0.4473	0.1868	55.4518

Tabel 5.2 memperlihatkan bahwa proses iterasi ke delapan memberikan *output* hasil segmentasi TsTN yang terbaik untuk citra *jatropha fruit* warna hijau. Hal ini disebabkan karena pada iterasi ke delapan memberikan nilai MSE yang terkecil dan nilai PSNR yang dihasilkan merupakan nilai tertinggi dari seluruh uji coba. Tabel 5.2 juga memperlihatkan bahwa perbandingan nilai MSE dan PSNR pada metode TsTN lebih baik daripada nilai MSE dan PSNR yang dihasilkan oleh segmentasi metode Otsu. Sehingga bisa dikatakan bahwa perbaikan yang dilakukan oleh metode TsTN pada proses segmentasi adalah berhasil, dengan nilai *error* yang hampir mendekati nol, yaitu 0,1856.

Tabel 5.3 Performa Parameter MSE dan PSNR untuk Segmentasi Citra *Jatropha Fruit* Kuning

Iterasi	<i>Threshold</i> (T_2)	MSE	PSNR
	<i>T = 0.4980 (Otsu)</i>	0.1705	55.8481
1	0.4880	0.1582	56.1730
2	0.4780	0.1506	56.3868
3	0.4680	0.1406	56.6837
4	0.4580	0.1272	57.1183
5	0.4480	0.1164	57.5065
6	0.4380	0.1059	57.9170
7	0.4280	0.0995	58.1864
8	0.4180	0.0937	58.4476
9	0.4080	0.0908	58.5855

Tabel 5.3 memperlihatkan bahwa proses iterasi ke sembilan memberikan *output* hasil segmentasi TsTN yang terbaik untuk citra *jatropha fruit* warna kuning. Hal ini disebabkan karena pada iterasi ke sembilan memberikan nilai MSE yang terkecil dan nilai PSNR yang dihasilkan merupakan nilai tertinggi dari seluruh uji coba. Tabel 5.3 juga memperlihatkan bahwa perbandingan nilai MSE dan PSNR pada metode TsTN lebih baik daripada nilai MSE dan PSNR yang dihasilkan oleh segmentasi metode Otsu. Sehingga bisa dikatakan bahwa perbaikan yang dilakukan oleh metode TsTN pada proses segmentasi adalah berhasil, dengan nilai *error* yang hampir mendekati nol, yaitu 0,0908.

Tabel 5.4 Performa Parameter MSE dan PSNR untuk Segmentasi Citra *Jatropha Fruit* Kuning Kecoklatan

Iterasi	<i>Threshold</i> (T_2)	MSE	PSNR
	$T = 0.2941$ (Otsu)	0.6536	50.0117
1	0.3041	0.6704	49.9011
2	0.3141	0.6944	49.7487
3	0.3241	0.7134	49.6313
4	0.3341	0.7460	49.4377
5	0.3441	0.7662	49.3215
6	0.3541	0.7915	49.1803
7	0.3641	0.8004	49.1318
8	0.3741	0.8079	49.0913
9	0.3841	0.8085	49.0882

Tabel 5.4 memperlihatkan bahwa proses iterasi pertama memberikan *output* hasil segmentasi TsTN yang terbaik untuk citra *jatropha fruit* warna kuning kecoklatan. Hal ini disebabkan karena pada iterasi pertama memberikan nilai MSE yang terkecil dan nilai PSNR yang dihasilkan merupakan nilai tertinggi dari seluruh uji coba. Tabel 5.4 juga memperlihatkan bahwa perbandingan nilai MSE dan PSNR pada metode TsTN lebih baik daripada nilai MSE dan PSNR yang dihasilkan oleh segmentasi metode Otsu. Sehingga bisa dikatakan bahwa perbaikan yang dilakukan oleh metode TsTN pada proses segmentasi adalah berhasil, dengan nilai *error* yang hampir mendekati nol, yaitu 0,6704.

Tabel 5.5 Performa Parameter MSE dan PSNR untuk Segmentasi Citra *Jatropha Fruit* Hitam

Iterasi	<i>Threshold</i> (T_2)	MSE	PSNR
	$T = 0.3137$ (Otsu)	0.9517	48.3800
1	0.3237	0.9499	48.3881
2	0.3337	0.9460	48.4058
3	0.3437	0.9437	48.4166
4	0.3537	0.9410	48.4291
5	0.3637	0.9389	48.4385
6	0.3737	0.9352	48.4559
7	0.3837	0.9321	48.4703
8	0.3937	0.9236	48.5101
9	0.4037	0.9180	48.5365

Tabel 5.5 memperlihatkan bahwa proses iterasi ke sembilan memberikan *output* hasil segmentasi TsTN yang terbaik untuk citra *jatropha fruit* warna hitam. Hal ini disebabkan karena pada iterasi ke sembilan memberikan nilai MSE yang terkecil dan nilai PSNR yang dihasilkan merupakan nilai tertinggi dari seluruh uji coba. Tabel 5.5 juga memperlihatkan bahwa perbandingan nilai MSE dan PSNR pada metode TsTN lebih baik daripada nilai MSE dan PSNR yang dihasilkan oleh segmentasi metode Otsu. Sehingga bisa dikatakan bahwa perbaikan yang dilakukan oleh metode TsTN pada proses segmentasi adalah berhasil, dengan nilai *error* yang hampir mendekati nol, yaitu 0,9180.

5.4 Analisis Hasil Ujicoba

Bagian ini menganalisis hasil uji coba yang telah dilakukan. Hasil yang dianalisis pada bagian ini adalah nilai akurasi. Nilai akurasi merupakan nilai performa dengan melihat jumlah data yang diprediksi secara tepat. Uji coba tersebut menggunakan dua skenario yang berbeda.

Skenario pertama adalah uji coba dengan menggunakan dua metode yang berbeda, yaitu segmentasi dengan menggunakan metode Otsu dan segmentasi dengan menggunakan metode TsTN. Metode Otsu juga perlu menjadi fitur dalam perangkat lunak Tugas Akhir ini. Selain untuk memudahkan pengguna membandingkan secara langsung hasil kinerjanya (berupa *output* citra biner) dengan metode *improved thresholding-based* (TsTN), Otsu merupakan metode yang terlibat secara langsung sebagai inisialisasi *threshold* awal dalam metode *improved thresholding-based* (TsTN).

Skenario kedua adalah uji coba dengan menghitung nilai parameter MSE dan PSNR tiap iterasi segmentasi, untuk masing-masing citra natural yang menjadi *inputnya*. Nilai MSE merupakan parameter untuk menentukan tingkat keberhasilan suatu segmentasi citra. Semakin rendah nilai MSE suatu citra dibandingkan iterasi sebelumnya, maka semakin baik hasil segmentasi citra tersebut. Sedangkan semakin tinggi nilai PSNR maka semakin baik hasil segmentasi citra tersebut. Pada skenario kedua, bisa diketahui proses iterasi ke berapa yang akan menghasilkan segmentasi terbaik, karena banyaknya proses iterasi yang diperlukan untuk menghasilkan citra tersegmentasi terbaik bisa berbeda-beda untuk tiap intensitas pencahayaan citra natural.

Gambar 5.1 sampai dengan gambar 5.16 menunjukkan hasil uji coba skenario pertama. Masing-masing metode diuji sebanyak empat kali pada citra natural dengan kecenderungan warna permukaan yang berbeda-beda. Masing-masing inisialisasi nilai *threshold* awal segmentasi pada metode TsTN menggunakan inisialisasi nilai *threshold* dari metode Otsu. Uji coba tiap citra

dilakukan sebanyak sepuluh kali. Dari sepuluh percobaan tersebut, metode *improved thresholding-based* (TsTN) secara konsisten menghasilkan segmentasi citra yang lebih baik dibandingkan dengan metode Otsu.

Hasil uji coba skenario pertama memperlihatkan bahwa metode *improved thresholding-based* (TsTN) mampu melakukan proses segmentasi citra pada citra natural. Hasil uji coba juga memperlihatkan bahwa metode TsTN menghasilkan segmentasi citra yang mendekati citra *ground truth*nya.

Pengujian pada skenario kedua membantu menentukan banyaknya jumlah iterasi yang diperlukan tiap citra natural untuk menghasilkan segmentasi terbaik dari metode TsTN. Penentuan jumlah iterasi ini dipengaruhi oleh perbandingan nilai MSE dan PSNR yang dihasilkan oleh tiap iterasi. Hasil uji coba pada skenario ini ditunjukkan oleh tabel 5.2 sampai tabel 5.5.

Tabel 5.2 memperlihatkan bahwa proses iterasi ke delapan memberikan *output* hasil segmentasi TsTN yang terbaik untuk citra *jatropha fruit* warna hijau. Hal ini disebabkan karena pada iterasi ke delapan memberikan nilai MSE yang terkecil dan nilai PSNR yang dihasilkan merupakan nilai tertinggi dari seluruh uji coba. Tabel 5.2 juga memperlihatkan bahwa perbandingan nilai MSE dan PSNR pada metode TsTN lebih baik daripada nilai MSE dan PSNR yang dihasilkan oleh segmentasi metode Otsu. Sehingga bisa dikatakan bahwa perbaikan yang dilakukan oleh metode TsTN pada proses segmentasi adalah berhasil, dengan nilai *error* yang hampir mendekati nol, yaitu 0,1856.

Hasil uji coba skenario memperlihatkan bahwa diperlukan delapan kali iterasi pada citra *jatropha fruit* dengan kecenderungan warna hijau untuk dapat menghasilkan citra tersegmentasi terbaik dengan menggunakan metode TsTN.

Tabel 5.3 memperlihatkan bahwa proses iterasi ke sembilan memberikan *output* hasil segmentasi TsTN yang terbaik untuk citra *jatropha fruit* warna kuning. Hal ini disebabkan karena pada iterasi ke sembilan memberikan nilai MSE yang terkecil dan nilai PSNR yang dihasilkan merupakan nilai tertinggi dari seluruh uji

coba. Tabel 5.3 juga memperlihatkan bahwa perbandingan nilai MSE dan PSNR pada metode TsTN lebih baik daripada nilai MSE dan PSNR yang dihasilkan oleh segmentasi metode Otsu. Sehingga bisa dikatakan bahwa perbaikan yang dilakukan oleh metode TsTN pada proses segmentasi adalah berhasil, dengan nilai *error* yang hampir mendekati nol, yaitu 0,0908.

Hasil uji coba skenario memperlihatkan bahwa diperlukan hingga sembilan kali iterasi pada citra *jatropa fruit* dengan kecenderungan warna kuning untuk dapat menghasilkan citra tersegmentasi terbaik dengan menggunakan metode TsTN.

Tabel 5.4 memperlihatkan bahwa proses iterasi pertama memberikan *output* hasil segmentasi TsTN yang terbaik untuk citra *jatropa fruit* warna kuning kecoklatan. Hal ini disebabkan karena pada iterasi pertama memberikan nilai MSE yang terkecil dan nilai PSNR yang dihasilkan merupakan nilai tertinggi dari seluruh uji coba. Tabel 5.4 juga memperlihatkan bahwa perbandingan nilai MSE dan PSNR pada metode TsTN lebih baik daripada nilai MSE dan PSNR yang dihasilkan oleh segmentasi metode Otsu. Sehingga bisa dikatakan bahwa perbaikan yang dilakukan oleh metode TsTN pada proses segmentasi adalah berhasil, dengan nilai *error* yang hampir mendekati nol, yaitu 0,6704.

Hasil uji coba skenario memperlihatkan bahwa diperlukan satu kali iterasi pada citra *jatropa fruit* dengan kecenderungan warna kuning kecoklatan untuk dapat menghasilkan citra tersegmentasi terbaik dengan menggunakan metode TsTN.

Tabel 5.5 memperlihatkan bahwa proses iterasi ke sembilan memberikan *output* hasil segmentasi TsTN yang terbaik untuk citra *jatropa fruit* warna hitam. Hal ini disebabkan karena pada iterasi ke sembilan memberikan nilai MSE yang terkecil dan nilai PSNR yang dihasilkan merupakan nilai tertinggi dari seluruh uji coba. Tabel 5.5 juga memperlihatkan bahwa perbandingan nilai MSE dan PSNR pada metode TsTN lebih baik daripada nilai MSE dan PSNR yang dihasilkan oleh segmentasi metode Otsu.

Sehingga bisa dikatakan bahwa perbaikan yang dilakukan oleh metode TsTN pada proses segmentasi adalah berhasil, dengan nilai *error* yang hampir mendekati nol, yaitu 0,9180.

Hasil uji coba skenario memperlihatkan bahwa diperlukan delapan kali iterasi pada citra *jatropha fruit* dengan kecenderungan warna hitam untuk dapat menghasilkan citra tersegmentasi terbaik dengan menggunakan metode TsTN.

Dari dua skenario uji coba yang telah dilakukan, metode *improved thresholding-based* (TsTN) dapat menghasilkan segmentasi citra pada lingkungan natural dengan baik. Pemilihan banyaknya jumlah iterasi yang tepat dapat mempengaruhi hasil *output* citra natural yang tersegmentasi menggunakan metode tersebut. Dari hasil uji coba, nilai MSE menunjukkan nilai yang kecenderungan mendekati nol, yaitu antara 0.9180 hingga menurun ke angka 0.0908 yang berarti semakin kecil kesalahan yang dilakukan oleh metode TsTN dalam melakukan segmentasi citra natural. Peningkatan hasil segmentasi dapat dilihat seperti pada gambar *output* yang ditunjukkan oleh gambar 5.3 dan 5.4 untuk citra *jatropha fruit* dengan kecenderungan warna hijau, gambar 5.7 dan 5.8 untuk citra *jatropha fruit* dengan kecenderungan warna kuning, gambar 5.11 dan 5.12 untuk citra *jatropha fruit* dengan kecenderungan warna kuning kecoklatan, dan pada gambar 5.15 dan 5.16 untuk citra *jatropha fruit* dengan kecenderungan warna hitam.

BAB VI KESIMPULAN DAN SARAN

Bab terakhir ini menjelaskan kesimpulan yang didapat dari pengerjaan Tugas Akhir ini, beserta saran yang dapat digunakan sebagai bahan pertimbangan untuk pengembangan atau riset selanjutnya.

6.1 Kesimpulan

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

- a. Implementasi modul-modul dalam metode TsTN (*Thresholding-based segmentation Technique for Natural images*) terbagi menjadi dua fungsi utama, yaitu fungsi Otsu sebagai inisialisasi nilai *threshold* awal dan fungsi TsTN yang terdiri dari beberapa fungsi seperti fungsi untuk menghitung piksel pada *middle element*, fungsi untuk melakukan invers, dan fungsi untuk memodifikasi nilai baru *threshold* (T_2).
- b. Evaluasi segmentasi citra menggunakan metode TsTN menghasilkan citra *output* yang jauh lebih baik dari metode Otsu untuk ketiga data *jatropha fruit* dengan kecenderungan warna permukaan hijau, kuning dan kuning kecoklatan. Namun pada citra berwarna hitam, *output* TsTN tidak berbeda jauh dengan *output* segmentasi Otsu, namun tetap lebih baik.
- c. Perangkat lunak yang dikembangkan dengan metode *improved thresholding-based* (TsTN) terbukti mampu untuk melakukan proses segmentasi citra natural. Hal ini ditunjukkan oleh nilai *error* pada MSE yang berada diantara 0.9180 - 0.0908 atau bisa dikatakan mendekati nol.
- d. Pada uji coba yang dilakukan, perangkat lunak segmentasi citra natural yang dikembangkan dengan metode *improved thresholding-based* (TsTN) memiliki akurasi yang lebih baik

dibandingkan dengan perangkat lunak yang hanya menggunakan metode *thresholding* Otsu untuk data citra *jatropha fruit*.

- e. Pada uji coba yang dilakukan, penentuan jumlah iterasi tiap citra ialah berbeda-beda agar dapat menghasilkan *output* citra tersegmentasi yang baik. Banyaknya iterasi yang diperlukan berhubungan dengan intensitas pencahayaan citra dan kecenderungan warna permukaan suatu citra natural.

6.2 Saran

Adapun saran yang disampaikan adalah uji coba menggunakan metode *thresholding-based* yang lain, seperti *watershed algorithm* [4] sehingga proses segmentasi diharapkan akan lebih baik dan dapat menghasilkan citra tersegmentasi yang lebih akurat.

DAFTAR PUSTAKA

- [1] H. J. Vala and A. Baxi, “A Review on Otsu Image Segmentation Algorithm”, *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, pp. 387 – 389, 2013.
- [2] A. Konar, “Computational Intelligence: Principles, Techniques, and Applications”, Springer-Verlag Berlin Heidelberg, Berlin, 2005.
- [3] S. L. S. Abdullah, H. A. Hambali, and N. Jamil, “Segmentation of Natural Images Using an Improved Thresholding-based Technique”, *Procedia Engineering*, vol. 41, pp. 938 - 944, 2012.
- [4] Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 3rd ed., Prentice Hall, New Jersey, 2008.
- [5] Pramuda A. Cahyan, dkk., “Segmentasi Citra Digital Menggunakan Algoritma *Watershed* dan *Lowpass Filter* Sebagai Proses Awal”, Brawijaya, Malang, 2013.
- [6] W. X. Kang, Q. Q. Yang, R. R. Liang, “The Comparative Research on Image Segmentation Algorithms”, IEEE Conference on ETCS, pp. 703 – 707, 2009.
- [7] Saiful Islam, Majidul Ahmed, “Implementation of Image Segmentation for Natural Images using Clustering Methods”,

International Journal of Emerging Technology and Advanced Engineering (IJETAE), vol. 3, pp. 175 - 180, 2013.

- [8] Er. Nirpjeet kaur and Er Rajpreet kaur, "A review on various method of image thresholding", IJCSE, 2011.
- [9] C. S. Varnan, et al., "Image Quality Assessment Techniques pn Spatial Domain", IJCST, vol. 2, pp. 177– 184, 2011.
- [10] N. Otsu, "A Threshold Selection Method From Gray-Level Histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 62-66, 1979.

BIODATA PENULIS



Isrovana Khamidah, yang akrab dipanggil Nova, lahir di Gresik pada tanggal 3 Maret 1990, merupakan anak kedua dari tiga bersaudara, pasangan Masjhari dan Wiwik Indayati. Penulis telah menempuh pendidikan formal mulai TK YWSG Tuban (1994-1996), SDN Latsari 1 Tuban (1996-2002), SMPN 1 Tuban (2002-2005), SMAN 1 Tuban (2005-2008), hingga melanjutkan pendidikan tinggi S1 Jurusan Teknik Informatika ITS Surabaya pada tahun 2008 melalui jalur PMDK.

Selama masa kuliah, penulis aktif berorganisasi dalam Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) dan Forum Komunikasi Mahasiswa Ronggolawe Tuban (FKMRT). Pada akhir masa pendidikannya, penulis terpilih untuk mengerjakan Tugas Akhir di Laboratorium Komputasi Cerdas dan Visualisasi dibawah bimbingan Dr.Eng. Nanik Suciati, S.Kom, M.Kom. dan Anny Yuniarti, S.Kom., M.Comp.Sc. Dengan ketekunan, motivasi tinggi untuk terus belajar dan berusaha, penulis berharap semoga Tugas Akhir ini mampu memberikan kontribusi positif bagi dunia pendidikan serta permulaan bagi penelitian lanjutan di Indonesia.

Data Pribadi Penulis:

Nama : Isrovana Khamidah
Alamat : Jl. Delima Gg. Delima Sejahtera no. 16 Perbon, Tuban
Telp. : 085655287132
Surel : isrovana.khamidah@gmail.com