



TUGAS AKHIR - KI141502

**EKSTRAKSI JALAN RAYA BERDASARKAN
PENDETEKSIAN *ZEBRA CROSS* DARI FOTO UDARA
BERESOLUSI SANGAT TINGGI DAN DATA DSM**

**NURUL SUPREMASINTA PUTRI
NRP 5112100197**

**Dosen Pembimbing I
Dr. Eng . Darlis Herumurti, S.Kom., M.Kom.**

**Dosen Pembimbing II
Bilqis Amaliah, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016**



TUGAS AKHIR - KI141502

**EKSTRAKSI JALAN RAYA BERDASARKAN
PENDETEKSIAN ZEBRA CROSS DARI FOTO UDARA
BERESOLUSI SANGAT TINGGI DAN DATA DSM**

**NURUL SUPREMASINTA PUTRI
NRP 5112100197**

**Dosen Pembimbing I
Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.**

**Dosen Pembimbing II
Bilqis Amaliah, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016**



UNDERGRADUATE THESIS - K1141502

**URBAN ROAD NETWORK EXTRACTION BASED ON
ZEBRA CROSSING DETECTION FROM A VERY HIGH
RESOLUTION RGB AERIAL IMAGE AND DSM DATA**

**NURUL SUPREMASINTA PUTRI
NRP 5112100214**

**Supervisor I
Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.**

**Supervisor II
Bilqis Amaliah, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016**

LEMBAR PENGESAHAN

**EKSTRAKSI JALAN RAYA BERDASARKAN
PENDETEKSIAN ZEBRA CROSS DARI FOTO
UDARA BERESOLUSI SANGAT TINGGI DAN
DATA DSM**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

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

Oleh

NURUL SUPREMASINTA PUTRI

NRP : 5112 100 197

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.
NIP: 197712172003121001 (Pembimbing 1)
2. Bilqis Amaliah, S.Kom., M.Kom.
NIP: 197509142001122002 (Pembimbing 2)



SURABAYA

Juni, 2016

EKSTRAKSI JALAN RAYA BERDASARKAN PENDETEKSIAN ZEBRA CROSS DARI FOTO UDARA BERESOLUSI SANGAT TINGGI DAN DATA DSM

Nama Mahasiswa : NURUL SUPREMASINTA PUTRI
NRP : 5112100197
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr. Eng. Darlis Herumurti, S.Kom.,
M.Kom.
Dosen Pembimbing 2 : Bilqis Amaliah, S.Kom., M.Kom.

Abstrak

Terdapat beberapa strategi pada proses ekstraksi jalan seperti segmentasi citra, deteksi garis, evolusi kurva, multi-resolution, kecerdasan buatan, deteksi tepi, road-tracking, operasi morfologi, pengenalan objek. Semua strategi dari ekstraksi jalan sangat tergantung pada karakteristik data.

Foto udara dan data ketinggian menjadi data pada tugas akhir ini. Mengintegrasikan foto udara dan data ketinggian akan menyelesaikan kelemahan masing-masing data dalam pengekstraksian jalan.

Pada Tugas Akhir ini digunakan pendeteksian zebra cross sebagai langkah awal untuk mengenali jalan, dilanjutkan dengan threshold, region growing, dan yang terakhir, road line filtering untuk mengekstraksi jalan.

Dari hasil uji coba didapatkan hasil quality pengekstraksian jalan yang dilakukan terbaik sebesar 42,2%, persentase jalan yang terdeteksi sebesar 86,5%, dan persentase objek bukan jalan yang terdeteksi sebesar 71,1% yang menggunakan metode threshold dan road line filtering.

Kata kunci : Foto Udara, Resolusi sangat tinggi, Data DSM, Ekstraksi Jalan, Pendeteksian

URBAN ROAD NETWORK EXTRACTION BASED ON ZEBRA CROSSING DETECTION FROM A VERY HIGH RESOLUTION RGB AERIAL IMAGE AND DSM DATA

Student's Name : NURUL SUPREMASINTA PUTRI
Student's ID : 5112100197
Department : Department of Informatics FTIF-ITS
First Advisor : Dr. Eng. Darlis Herumurti, S.Kom.,
M.Kom.
Second Advisor : Bilqis Amaliah, S.Kom., M.Kom.

Abstract

There are some strategies in term of road extraction process such as: Image segmentation, edge detection, curve evolution, multi-resolution, artificial intelligent, line detection, road tracking, morphology operation, object recognition. All strategies of road extraction are very depend on the characteristic of the data.

RGB aerial image and DSM data be the input data in this undergraduate thesis. Integrating the aerial image and elevation based data will overcome the shortcomings and weaknesses of each type of data in road extraction.

In this undergraduate thesis, to recognize the road used zebra cross detection as initial step, and then used threshold, region growing, and the final step is road line filtering to extract the road.

From experimental results, the quality of road extraction is 42,2%, percentage of detected road is 86,5%, and percentage of detected another object (non road) is 71,1% which used threshold and region growing method .

Keywords: Aerial image, Very high resolution, DSM data, Road extraction, Detection

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Ekstraksi Jalan Raya Berdasarkan Pendeteksian Zebra cross dari Foto Udara Beresolusi Sangat Tinggi dan Data DSM”**. Bagi penulis, pengerjaan Tugas Akhir ini merupakan sebuah pengalaman yang berharga. Selama pengerjaan Tugas Akhir, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menjalani perkuliahan di Teknik Informatika ITS dan Tugas Akhir ini adalah implementasi dari apa yang telah penulis pelajari.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Keluarga tercinta yakni Almarhum Papa, Mama, Adik Lukman, yang telah memberikan dukungan moral dan material serta do'a yang tak terhingga untuk penulis. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan Tugas Akhir ini.
3. Bapak Dr. Eng. Darlis Herumurti, S.Kom., M.Kom. selaku pembimbing I yang telah membimbing dan membantu penulis serta memberikan motivasi dalam menyelesaikan Tugas Akhir ini dengan sabar.
4. Ibu Bilqis Amaliah, S.Kom., M.Kom. selaku pembimbing II yang juga telah membantu, membimbing, dan memotivasi kepada penulis dalam mengerjakan Tugas Akhir ini.
5. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Radityo Anggoro,

S.Kom.,M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya serta staf karyawan Jurusan Teknik Informatika ITS yang telah memberikan bantuan demi kelancaran admisnistrasi penulis selama kuliah.

6. Sa'id Al Musayyab yang selalu menemani, memberi arahan dan motivasi, serta menjadi rekan berbagi untuk penulis.
7. Serta semua pihak yang yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga, penulis mengharapkan kritik dan saran yang membangun dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2016

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak.....	vii
Abstract.....	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
DAFTAR KODE SUMBER	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	4
1.4 Manfaat	4
1.5 Batasan Masalah.....	4
1.6 Metodologi.....	4
1.7 Sistematika Penulisan Laporan Tugas Akhir	6
BAB II TINJAUAN PUSTAKA	9
2.1 Fotografi Udara	9
2.2 Data DSM (Digital Surface Model).....	10
2.3 <i>Opening</i> (Morfologi)	11
2.4 <i>Closing</i> (Morfologi)	12
2.5 Segmentasi	12
2.6 <i>Thresholding</i>	13
2.7 <i>Region growing</i>	13
2.8 <i>Thinning</i> (Morfologi).....	14
2.9 <i>Regionprops</i>	14
BAB III PERANCANGAN SISTEM.....	15
3.1 Desain Metode Secara Umum	15
3.2 Perancangan Data	17
3.2.1 Data Masukan.....	17
3.2.2 Data Proses	18
3.2.3 Data Keluaran.....	23

BAB IV IMPLEMENTASI.....	25
4.1 Lingkungan Implementasi	25
4.1.1. Lingkungan Implementasi Perangkat Keras	25
4.1.2. Lingkungan Implementasi Perangkat Lunak.....	25
4.2 Implementasi Segmentasi Citra Foto Udara RGB	25
4.3 Implementasi Deteksi <i>Zebra cross</i> Pada Citra Udara RGB.	26
4.4 Implementasi Pengekstrasian Jalan.....	28
4.4.1 Implementasi <i>Threshold</i> Titik-titik <i>Zebra cross</i>	29
4.4.2 Implementasi <i>Region growing</i>	30
4.5 Implementasi <i>Road Line Filtering</i>	33
BAB V UJI COBA DAN EVALUASI.....	35
5.1 Lingkungan Uji Coba	35
5.2 Data Uji Coba.....	35
5.3 Skenario Uji Coba	36
5.3.1 Skenario 1.....	38
5.3.2 Skenario 2.....	39
5.3.3 Skenario 3.....	41
5.4 Analisa Hasil Uji Coba	42
BAB VI KESIMPULAN DAN SARAN	47
6.1 Kesimpulan.....	47
6.2 Saran.....	47
Daftar Pustaka	49
LAMPIRAN.....	51
BIODATA PENULIS	69

DAFTAR GAMBAR

Gambar 2. 1. Citra Foto Udara RGB	9
Gambar 2. 2. Citra Foto Udara RGB Diperbesar	10
Gambar 2. 3. Data DSM.....	11
Gambar 2. 4. Visualisasi Data DSM	11
Gambar 3. 1. Alur metode.....	16
Gambar 3. 2. Citra Foto Udara RGB	17
Gambar 3. 3. Data DSM.....	18
Gambar 3. 4. Alur pendeteksian zebra cross	19
Gambar 3. 5. Alur pengecekan pada <i>Region growing</i>	21
Gambar 3. 6. Tampilan Keluaran Aplikasi	23
Gambar 5. 1. <i>Ground Truth</i>	36
Gambar 5. 2. <i>Zebra cross</i> terdeteksi.....	38
Gambar 5. 3. Hasil Uji Coba Skenario 1 Dengan <i>Quality</i> Terbaik, 37,6%.....	43
Gambar 5. 4. Hasil Uji Coba Skenario 2 Dengan <i>Quality</i> Terbaik, 42,2%.....	44
Gambar 5. 5. Hasil Uji Coba Skenario 3 Dengan <i>Quality</i> Terbaik, 37,9%.....	44

DAFTAR TABEL

Tabel 5. 1. Hasil dari Uji Coba Skenario 1	38
Tabel 5. 2. Hasil dari Uji Coba Skenario 2.....	40
Tabel 5. 3. Hasil dari Uji Coba Skenario 3.....	41
Tabel 5. 4. Hasil dari Uji Coba Skenario 3 (Lanjutan)	42

DAFTAR KODE SUMBER

Kode Sumber 4. 1. Fungsi Segmentasi.....	26
Kode Sumber 4. 2. Fungsi Deteksi <i>Zebra cross</i>	27
Kode Sumber 4. 3. Fungsi <i>Threshold</i>	29
Kode Sumber 4. 4. Fungsi <i>Region growing</i>	32
Kode Sumber 4. 5 Fungsi <i>Road Line Filtering</i>	33

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sebagai Negara berkembang, Indonesia terus membangun berbagai macam infrastruktur yang berkelanjutan. Pembangunan tersebut umumnya dalam bentuk rencana pembangunan yang tentunya memerlukan alokasi APBN setiap tahunnya.

Untuk menjawab setiap tantangan sosial maupun ekonomi setiap tahunnya maka pembangunan yang menjadi perhatian serius pemerintah Indonesia saat ini adalah intensifikasi program pembangunan infrastruktur akses antar wilayah. Adanya kemudahan dalam hal akses akan membuat pertumbuhan ekonomi Indonesia meningkat dan berputar. Dengan begitu, akan ada transfer kesejahteraan dari elemen ekonomi tinggi ke elemen ekonomi menengah dan rendah. Komponen utama dalam kegiatan produksi, distribusi, dan konsumsi adalah akses. Oleh karena itu, Indonesia hingga saat ini telah banyak merealisasikan program pembangunan akses ekonomi, seperti jalan, tol, dan jembatan. Dengan adanya pembangunan akses ekonomi yang terus meningkat tersebut, diperlukan teknologi yang mampu menyediakan informasi mengenai jalan di suatu daerah.

Teknologi yang dapat menyajikan informasi yang selalu terbaru, sangat dibutuhkan pada era digital saat ini. Sebuah informasi peta atau jalan digital menjadi salah satu hal paling penting pada transportasi modern. Peta digital menjadi tempat untuk menjaga informasi agar selalu terbaru dan akurat untuk menyediakan sebuah jalan yang benar dan dapat dipercaya. Terlebih lagi pada daerah di mana peta digital tidak menyediakannya atau butuh diperbarui, ekstraksi jalan yang ter-otomatisasi merupakan cara yang efisien dan efektif untuk tujuan ini. Proses manual atau memperbaharui basis data jalan merupakan cara yang tidak efektif, memakan waktu dan sangat

mahal. Oleh karena itu, ekstraksi jalan menjadi salah satu hal penting untuk dapat digunakan pada navigasi berkendaraan, analisa jalan, melacak perbaruan jalan dan seterusnya. Sehubungan dengan isu ini, banyak penulis pada bidang *Intelligent Transportation Systems* (ITS) dan *remote sensing*, melakukan yang terbaik untuk masalah ini dan menghasilkan sebuah dampak yang bagus pada lingkungan sosial modern. Sejauh ini, banyak pencapaian yang telah dihasilkan, akan tetapi, sejalan dengan teknologi pada perkembangan dan metode-metode pendukung telah diperkenalkan, penulis pada topik ini akan melanjutkan untuk mencapai hasil yang lebih baik dari sebelumnya.

Kemajuan pada sensor citra menjadikan adanya citra dengan resolusi sangat tinggi, *very high resolution* (VHR). Terlebih lagi adanya data dengan resolusi sangat tinggi (0.2-0.5m/piksel pada resolusi spasial) *quality* dari informasi sangat ditingkatkan. Tidak hanya pada fotografi udara yang menghasilkan *quality* yang sangat bagus dari citra, tetapi juga teknologi dapat menyediakan data 3D yang memvisualisasikan informasi dari ketinggian permukaan termasuk gedung-gedung, pohon-pohon dan sebagainya, dikenal dengan data Digital Surface Model (DSM). Dengan mengombinasikan informasi-informasi tersebut akan membuat banyak pilihan pendekatan. Akan tetapi ekstraksi jalan di daerah perkotaan merupakan pekerjaan yang sangat menantang seperti ekstraksi jalan yang kompleks dengan objek yang banyak seperti gedung-gedung, taman-taman, mobil-mobil, pohon-pohon dan sebagainya. Ditambah lagi, bayangan dari gedung-gedung dapat membuat kesulitan tersendiri.

Terdapat beberapa strategi pada proses ekstraksi jalan seperti segmentasi citra, deteksi garis, evolusi kurva, *multi-resolution*, kecerdasan buatan, deteksi tepi, *road-tracking*, operasi morfologi, pengenalan objek [1]. Semua strategi dari ekstraksi jalan sangat tergantung pada karakteristik data. Selanjutnya, pandangan dari data masukan, utamanya hanya

terdapat dua kasus, yakni perkotaan dan perdesaan, biasanya, pendekatan tidak dapat digunakan untuk keduanya, karena keduanya memiliki karakteristik yang berbeda.

Mengintegrasikan foto udara dan data ketinggian akan menyelesaikan kelemahan masing-masing data [2]. Data DSM yang berdasarkan pada permukaan ketinggian tidak terdapat bayangan yang akan membantu ekstraksi jalan secara signifikan. Sedangkan untuk data VHR, informasi jalan seperti *zebra cross* memberikan informasi mengenai jalan tersebut.

Pada tugas akhir ini, penulis pertama-tama akan mendeteksi *zebra cross* sebagai penanda bahwa objek tersebut adalah jalan. Pendeteksian dilakukan dengan filter LoG, proses morfologi, dan disesuaikan dengan data DSM. Kemudian, setelah didapatkan titik ketinggian dari masing-masing titik *zebra cross*, dilakukan *thresholding* untuk mengambil titik yang tingkat kemungkinan jalannya tinggi. Dari titik *zebra cross* tersebut dilakukan *region growing* untuk membentuk area jalan. Setelah itu, dilakukan morfologi sederhana, yang dilanjutkan dengan *mem-filter* garis jalan menggunakan *regionprops* dan morfologi sederhana. Diharapkan dengan penggunaan foto udara dan data DSM, pengekstraksian jalan menjadi lebih akurat.

1.2 Rumusan Masalah

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

1. Bagaimana mendeteksi *zebra cross* yang sesuai dengan data yang digunakan untuk pengenalan jalan?
2. Bagaimana melakukan pendeteksian jalan yang akurat?
3. Bagaimana cara mengonstruksi segmentasi jalan raya agar terbentuk area jalan yang sebenarnya?
4. Bagaimana menggabungkan data foto udara dan data DSM?

1.3 Tujuan

Tujuan dari Tugas Akhir ini adalah mengekstraksi jalan perkotaan dari foto udara dan data DSM berdasarkan pendeteksian *zebra cross*.

1.4 Manfaat

Dengan dibuatnya Tugas Akhir ini maka manfaat yang diperoleh yaitu, mempermudah untuk menemukan jalan baru maupun jalan yang diperbaharui rutenya dan mempermudah untuk melakukan perancangan jalan pada wilayah kota.

1.5 Batasan Masalah

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

1. Data yang digunakan adalah foto udara beresolusi 75 mega piksel dari daerah Sapporo, Jepang tahun 2010 serta memiliki model warna RGB dan data DSM dengan resolusi 12 mega piksel.
2. Jalan yang akan diekstraksi merupakan jalan raya, perkotaan dengan objek sekitar, gedung, mobil, taman, dan pepohonan.
3. Objek penanda yang dijadikan acuan sebagai jalan adalah *zebra cross*.
4. Dalam data DSM, area selain jalan memiliki ketinggian yang sama dengan jalan.
5. Pembangunan perangkat lunak menggunakan Matlab R2013a.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.

Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Pada proposal Tugas Akhir tersebut berisi rencana Tugas Akhir yang akan dikerjakan sebagai syarat untuk menyelesaikan studi dan meraih gelar Strata-1 Teknik Informatika. Pada proposal tersebut dijelaskan secara garis besar tentang tahapan analisis citra yang akan dilakukan. Diantaranya tahapan preprocessing, ekstraksi fitur, morfologi, dan segmentasi, dan deskripsi.

2. Studi literatur

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk pengerjaan Tugas Akhir sekaligus mempelajarinya. Beberapa literatur yang perlu dipelajari lebih dalam lagi untuk proses pengolahan citra khususnya pendeteksian *zebra cross*, pemrosesan foto udara dan dengan data DSM, serta ekstraksi jalan.

3. Analisis dan desain perangkat lunak

Pada tahap ini akan dilakukan analisis dan design perancangan aplikasi sesuai dengan tujuan yang dijabarkan. Kemudian disesuaikan dengan metode yang tepat, hal ini dimaksudkan agar nantinya ketika diimplementasikan ke dalam aplikasi dapat berjalan sesuai yang diharapkan.

4. Implementasi perangkat lunak

Perangkat lunak ini akan dibangun dengan menggunakan bahasa pemrograman dan kaskas bantu Matlab R2013a dengan fungsi yang sudah tersedia di dalamnya.

5. Pengujian dan evaluasi

Aplikasi akan diuji setelah selesai diimplementasikan menggunakan skenario yang sudah dipersiapkan. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Dengan melakukan pengujian dan

evaluasi dimaksudkan juga untuk mengevaluasi jalannya program, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

6. Penyusunan buku Tugas Akhir.

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini merupakan laporan secara lengkap mengenai Tugas Akhir yang telah dikerjakan baik dari sisi teori, rancangan, maupun implementasi sehingga memudahkan bagi pembaca dan juga pihak yang ingin mengembangkan lebih lanjut. Sistematika penulisan buku Tugas Akhir secara garis besar antara lain:

Bab I Pendahuluan

Bab ini berisi penjelasan latar belakang, rumusan masalah, batasan masalah dan tujuan pembuatan Tugas Akhir. Selain itu, metodologi pengerjaan dan sistematika penulisan laporan Tugas Akhir juga terdapat di dalamnya.

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.

Bab III Perancangan Perangkat Lunak

Bab ini berisi penjelasan tentang rancangan dari sistem yang akan dibangun. Rancangan ini dituliskan dalam bentuk *pseudocode*.

Bab IV Implementasi

Bab ini berisi penjelasan implementasi dari rancangan yang telah dibuat pada bab sebelumnya. Implementasi disajikan dalam bentuk *code* secara keseluruhan disertai dengan penjelasannya.

Bab V Uji Coba Dan Evaluasi

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

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.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

Dalam bab ini akan dijelaskan mengenai teori-teori yang merupakan dasar dari pembangunan sistem. Selain itu terdapat penjelasan yang menunjang pengerjaan Tugas Akhir ini sehingga dapat memberikan gambaran secara umum sistem yang akan dibangun.

2.1 Fotografi Udara

Fotografi udara mengambil gambar dasar tanah dari sebuah ketinggian atau posisi tegak lurus ke bawah. Biasanya kamera yang digunakan tidak mendukung untuk digunakan di dasar tanah. *Platform* untuk fotografi udara terdiri dari pesawat dengan sayap tetap, helikopter, *multirotor Unmanned Aircraft Systems* (UAS), balon, *blimps* dan *dirigibles*, roket, *pigeons*, layang-layang, parasut, stand-alone telescoping dan vehicle-mounted poles. Kamera yang dipasang, digerakan dari jarak jauh atau secara otomatis, untuk fotografi dengan menggunakan tamgam dilakukan oleh seorang fotografer. Sebagai contoh citra foto udara, dapat dilihat pada Gambar 2. 1 dan untuk penampilan objek-objek pada citra, dapat dilihat pada Gambar 2. 2 .



Gambar 2. 1. Citra Foto Udara RGB



Gambar 2. 2. Citra Foto Udara RGB Diperbesar

Fotografi udara seharusnya tidak susah dengan *air-to-air photography* dimana satu atau lebih pesawat digunakan sebagai pesawat pengejar yang “mengejar” dan memfoto pesawat lainnya dalam penerbangan. [3]

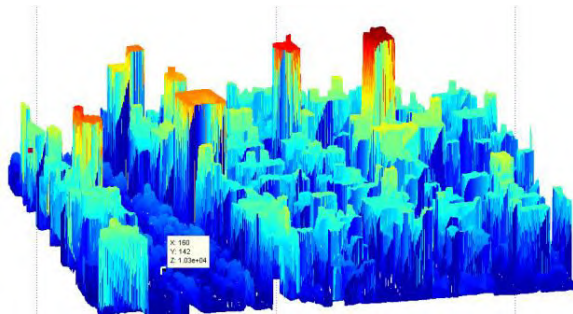
2.2 Data DSM (Digital Surface Model)

Pada kebanyakan kasus, data DSM merupakan model elevasi dengan referensi permukaan objek terhadap Mean Sea Level (MSL) 18,61 tahun, yang menampilkan ketinggian dari bentuk permukaan apapun di bumi beserta objek-objek apapun yang berada di atas tanah bumi, seperti gedung, pohon, rumah, dan sebagainya dalam bentuk citra. Gambar 2. 3. merupakan data DSM, namun agar dapat dimengerti secara visual oleh mata maka perlu dilakukan visualisasi dari data DSM menggunakan aplikasi seperti MATLAB sehingga didapatkan citra hasil visualisasi data DSM seperti pada Gambar 2. 4 yang menggambarkan ketinggian pada Gambar 2. 3. Semua data yang diambil menggunakan satelit, pesawat terbang, atau perangkat terbang lainnya merupakan data

DSM. Data DSM biasanya digunakan untuk pemodelan lanskap, pemodelan kota dan visualisasi aplikasi [4].



Gambar 2. 3. Data DSM



Gambar 2. 4. Visualisasi Data DSM

2.3 *Opening* (Morfologi)

Dalam matematika morfologi, *opening* adalah dilasi dari erosi dari sebuah set A (gambar biner) dengan *structuring element* B :

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

Di mana \ominus dan \oplus menunjukkan erosi dan dilasi.

Sama dengan *closing*, *opening* dipakai di visi komputer dan pengelolaan citra digital sebagai metode dasar morfologi untuk penghapusan objek kecil yang tidak diinginkan.

Opening menghapus objek-objek kecil dari bagian depan gambar, biasanya dalam bentuk piksel gelap dari gambar, menempatkan piksel-piksel tersebut di bagian belakang gambar, selama *closing* menghapus lubang-lubang kecil di bagian depan gambar, merubah objek-objek kecil dari bagian belakang ke bagian depan. Teknik-teknik tersebut dapat juga dijadikan untuk menemukan bentuk-bentuk spesifik dari sebuah gambar. *Opening* dapat digunakan untuk mencari objek-objek yang cocok dengan *structuring element* seperti (*edges* dan *corners*). [5]

2.4 *Closing* (Morfologi)

Dalam matematika morfologi, *closing* dari sebuah set A dengan structuring element B adalah erosi dari dilasi set tersebut :

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

Di mana \oplus dan \ominus menunjukkan erosi dan dilasi.

Dalam pengelolaan citra digital, *closing* sama dengan *opening*, metode morfologi dasar untuk menghilangkan objek yang tidak diinginkan dari sebuah gambar. *Opening* menghapus objek-objek kecil, sedangkan *closing* menghapus lubang-lubang kecil. [6]

2.5 Segmentasi

Dalam visi komputer, segmentasi citra merupakan proses dari membagi-bagi sebuah citra digital menjadi beberapa bagian. Tujuan dari segmentasi adalah untuk menyederhanakan dan atau mengubah representasi dari sebuah citra menjadi sesuatu yang lebih berarti dan lebih mudah untuk dianalisa. Segmentasi citra digunakan untuk menempatkan objek-objek dan garis-garis batas dari citra-citra. Terlebih lagi, segmentasi citra adalah proses dari memberikan sebuah tanda untuk setiap piksel dalam sebuah citra,

piksel-piksel dengan tanda yang sama memberikan karakteristik yang sama. [7]

2.6 *Thresholding*

Thresholding merupakan metode paling sederhana dari segmentasi gambar atau citra, *thresholding* dapat digunakan untuk membuat citra biner.

Metode *thresholding* yang paling sederhana adalah mengganti setiap piksel dalam sebuah gambar dengan sebuah piksel hitam jika gambarnya intensitasnya $I_{i,j}$ kurang dari sebuah nilai konstan T (yakni $I_{i,j} < T$) atau sebuah piksel putih jika intensitas gambar lebih besar daripada nilai konstan. [8]

2.7 *Region growing*

Region growing merupakan metode segmentasi citra yang berdasarkan pada daerah atau *region*. *Region growing* juga diklasifikasikan sebagai segmentasi citra berdasarkan piksel yang mempengaruhi seleksi dari poin-poin *seed* awal.

Pendekatan ini untuk segmentasi, memeriksa piksel-piksel tetangganya dari *seed* awal dan menentukan apakah piksel tetangga tersebut seharusnya ditambahkan daerah yang akan dibentuk. Proses diulang terus, hingga tidak ada lagi piksel yang dapat diubah maupun diperiksa.

Tujuan utama dari segmentasi adalah untuk membagi sebuah gambar menjadi beberapa bagian. Beberapa metode segmentasi seperti *thresholding* mencapai tujuan ini dengan mencari garis-garis batas antara daerah-daerah berdasarkan piksel-piksel yang tidak berkelanjutan dari gambar *grayscale* atau warna lain. Segmentasi berdasarkan daerah merupakan sebuah teknik untuk menentukan sebuah daerah secara langsung.

Langkah awal dari *region growing* adalah untuk memilih poin-poin *seed*, yang dipilih berdasarkan kriteria yang diinginkan, sebagai contoh piksel-piksel dalam rentang *grayscale* tertentu, dan sebagainya. Daerah awal dimulai sesuai dengan lokasi tempat

seed-seed tersebut berada. Daerah-daerah tersebut kemudian diperluas dari *seed-seed* itu ke poin-poin sebelahnya tergantung dengan kriteria *region growing* yang diinginkan. Kriteria tersebut dapat berupa intensitas piksel, tekstur *grayscale*, atau warna.

Contoh sederhana dari *region growing*, digunakan 4 piksel tetangga yang terkoneksi untuk memperluas daerah dari *seed*. Dapat juga digunakan 8 piksel tetangga yang terhubung, dan untuk kriteria yakni nilai piksel yang sama. Piksel-piksel tetangga dibandingkan dengan *seed* yang akan diperiksa. Jika piksel-piksel tetangga tersebut memiliki nilai intensitas yang sama dengan *seed* yang akan diperiksa, piksel-piksel tersebut dilasifikasikan dengan *seed* tersebut. Proses ini dilakukan terus sampai tidak ada kemungkinan lagi. [9]

2.8 *Thinning* (Morfologi)

Thinning merupakan transformasi dari sebuah citra digital menjadi sebuah bentuk yang disederhanakan tetapi secara topologi merupakan gambar yang sama. Ini merupakan jenis dari topologi skeleton, tetapi dihitung menggunakan operator morfologi matematika. [10]

2.9 *Regionprops*

Tujuan dari penggunaan metode *regionprops* ini adalah untuk mengambil beberapa sifat atau fitur dari daerah yang dimiliki sebuah gambar atau citra.

Sebagai contoh dari beberapa sifat atau fitur yang diambil dari metode *regionprops* adalah, *orientation*, fitur ini memberikan sebuah skalar yang memiliki derajat sudut antara sudut x dan sumbu utama dari elips yang memiliki *second-moments* seperti daerah. Nilai-nilai yang ada dalam satuan derajat sudut, rentang anatar -90 sampai 90 derajat. [11]

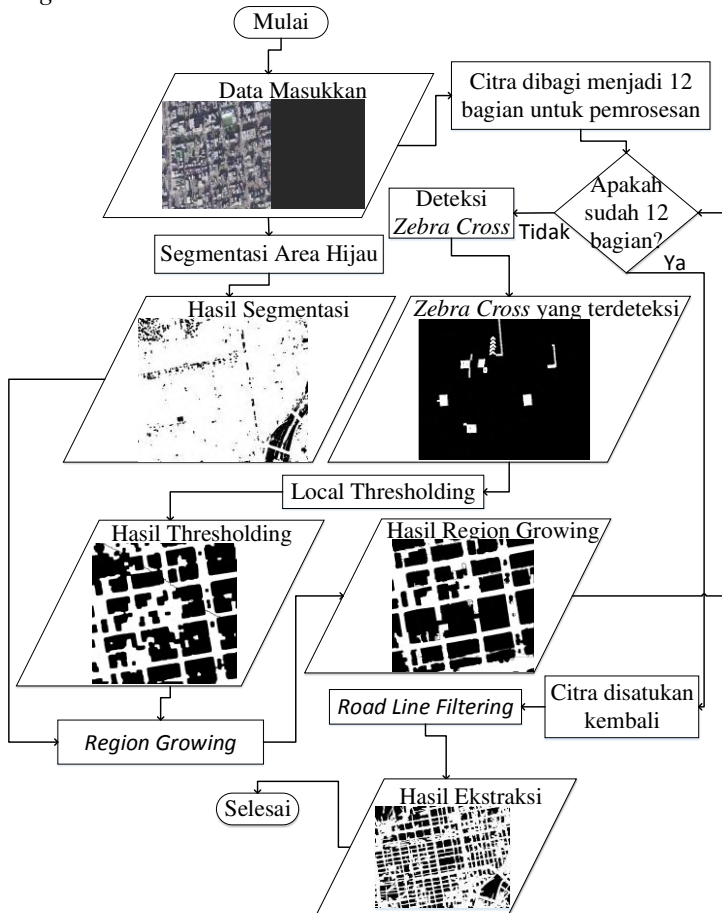
BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai perancangan sistem perangkat lunak yang akan dibuat. Perancangan yang dijelaskan meliputi data dan proses. Data yang dimaksud adalah data yang akan diolah dalam perangkat lunak baik digunakan sebagai pelatihan maupun pengujian sehingga tujuan Tugas Akhir ini bisa tercapai. Selain itu akan dijelaskan juga desain metode secara umum pada sistem.

3.1 Desain Metode Secara Umum

Pada tugas akhir ini akan dibangun suatu perangkat lunak yang dapat melakukan pengekstraksian jalan, dengan sumber data yang digunakan adalah citra RGB dan data DSM. Pada tahap ini dilakukan beberapa tahap meliputi *pre processing*, segmentasi, ekstraksi fitur, morfologi, dan deskripsi. Kemudian hasil akhirnya berupa citra yang menggambarkan garis jalan yang telah terekstrak, persentase banyaknya dari jalan dan bukan jalan, serta *quality* dari garis jalan yang didapatkan, dibandingkan dengan garis jalan yang sebenarnya. Diagram alur dari proses ini ditunjukkan oleh Gambar 3. 1. Gambar RGB pertama-tama dilakukan segmentasi untuk mendeteksi daerah yang berwarna hijau, dapat berupa pohon, taman, dan sejenisnya. Proses ini dilakukan untuk meminimalisir kesalahan pembentukan jalan pada proses *region growing*. *Zebra cross* dideteksi dari gambar RGB yang kemudian dicocokkan titik ketinggiannya dengan data DSM. Setelah dilakukan deteksi *zebra cross*, data DSM dilakukan *threshold* agar nilai ketinggian yang didapatkan mirip dengan ketinggian jalan. *Region growing* dilakukan kemudian untuk membentuk jalan berdasarkan hasil *threshold* data DSM dan gambar segmentasi. Dari hasil *region growing*, gambar ekstraksi jalan diperhalus menggunakan morfologi yang dilanjutkan

dengan membentuk garis-garis jalan menggunakan *road line filtering*.



Gambar 3. 1. Alur metode

3.2 Perancangan Data

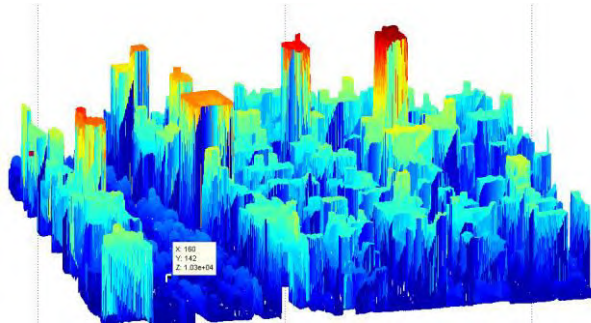
Dengan mempertimbangkan terbatasnya waktu pengerjaan Tugas Akhir, implementasi perangkat lunak haruslah efisien dan cepat. Karena memiliki fleksibilitas dan memiliki banyak kemudahan dalam mengimplementasikan filter-filter serta metode-metode citra lainnya maka MATLAB adalah solusi sebagai alat dalam implementasi.

3.2.1 Data Masukan

Data masukan adalah data yang akan diolah oleh sistem untuk mendapatkan hasil keluaran yang sudah ditentukan sistem. Pada perangkat lunak ini, *Dataset* yang digunakan merupakan citra foto udara dalam bentuk RGB dan data DSM yang menandakan nilai ketinggian dari masing-masing objek pada citra foto udara tersebut. Data citra RGB berukuran lebih besar dibandingkan dengan data DSM yakni 10.000 x 7.500 piksel, sedangkan data DSM berukuran 4000 x 3000 piksel. Berikut merupakan contoh inputan dalam sistem ini ditunjukkan pada Gambar 3. 2 dan Gambar 3. 3 di bawah.



Gambar 3. 2. Citra Foto Udara RGB



Gambar 3. 3. Data DSM

3.2.2 Data Proses

Data proses adalah proses pengolahan data dari data masukan. Sebelum keluarnya suatu data keluaran dari citra masukan maka data masukan tersebut harus di proses melalui tahap segmentasi, deteksi *zebra cross*, *threshold*, *region growing*, morfologi, dan *road line filtering* hingga akhirnya keluar sebuah data keluaran.

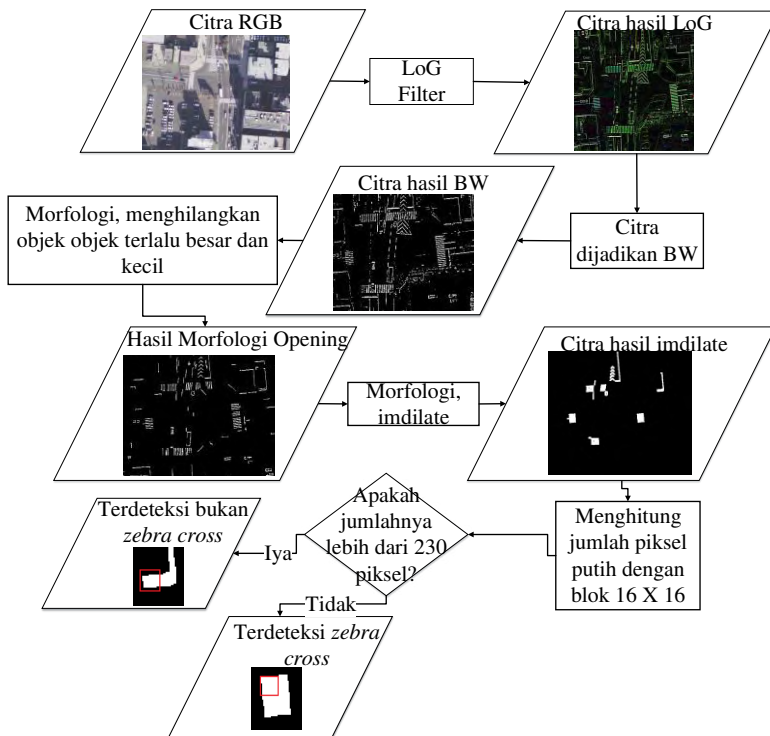
3.2.2.1 Segmentasi citra foto udara RGB

Proses segmentasi bertujuan membagi-bagi citra foto udara RGB menjadi beberapa bagian-bagian berdasarkan warna dari masing-masing piksel. Segmentasi yang dilakukan, menggunakan L^*a^*b *color space* dan clustering dengan k-means.

Piksel yang ingin didapatkan dari proses ini yakni piksel berwarna hijau yang ada pada citra berupa taman-taman maupun pohon-pohon karena selama proses, sering kali taman-taman maupun pepohonan masih dianggap sebagai jalan. Maka, segmentasi digunakan untuk meminimalkan kesalahan pegekstrasian jalan pada proses *region growing*.

3.2.2.2 Deteksi *zebra cross* pada citra foto udara RGB

Untuk mendeteksi *zebra cross*, terlebih dahulu citra foto udara RGB di-*filter* menggunakan LoG untuk mendeteksi garis tepi pada masing-masing objek di citra. Kemudian, dilakukan morfologi untuk menghilangkan objek-objek yang tidak ingin dideteksi yakni objek yang berukuran terlalu besar atau kecil dari ukuran *zebra cross*. Morfologi ini bertujuan agar objek-objek garis pada citra seukuran dengan garis-garis pada *zebra cross* saja.



Gambar 3. 4. Alur pendeteksian *zebra cross*

Kemudian, untuk mengetahui manakah *zebra cross*, citra di-morfologi dengan *imdilate*. Dengan menggunakan morfologi ini objek-objek citra akan berukuran lebih besar dari sebelumnya. Objek *zebra cross* yang sebelumnya garis-garisnya memiliki jarak satu sama lain maka akan terlihat menyatu sehingga objek *zebra cross* nantinya akan terlihat sebagai satu objek persegi berwarna putih seperti pada Gambar 3. 4.

Untuk pemrosesan pendeteksian, dilakukan dengan cara pembagian citra menjadi beberapa blok. Pada masing-masing blok dilakukan penjumlahan piksel yang warnanya putih. Jumlah piksel putih tersebut nantinya yang menentukan pada blok tersebut terdeteksi *zebra cross* atau tidak.

3.2.2.3 Pengekstrasian Jalan

Terdapat dua metode utama untuk pengekstrasian jalan yakni *threshold* dan *region growing*. Pada dasarnya, keduanya memiliki fungsi yang sama untuk ekstraksi jalan. Masing-masing metode memiliki keunggulan dan kelemahan, maka kedua metode ini digabungkan untuk hasil ekstraksi jalan yang optimal.

3.2.2.3.1 *Threshold* titik-titik *zebra cross*

Metode *threshold* digunakan untuk mempercepat pengekstrasian jalan. Dibandingkan dengan proses *region growing*, proses *threshold* lebih cepat, karena proses yang dilakukan metode *threshold* lebih sederhana dibandingkan dengan *region growing*.

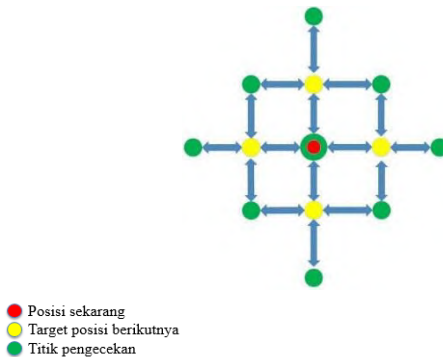
Pada metode *threshold*, semua piksel yang berada dalam rentang yang telah ditentukan maka akan dianggap sebagai jalan. Rentang tersebut berisi nilai titik-titik ketinggian dari titik-titik *zebra cross* yang telah didapatkan, dengan nilai ketinggian yang cukup rendah sehingga dapat langsung dipastikan sebagai jalan.

3.2.2.3.2 *Region growing data hasil threshold*

Metode *region growing* digunakan untuk membuat pengekstrasian jalan lebih akurat. *Region growing* memiliki proses yang lebih kompleks dibandingkan dengan *threshold*.

Data DSM memiliki ketinggian jalan yang beragam nilainya. Maka, tidak cukup hanya dengan menggunakan metode *threshold* saja.

Metode *region growing* dilakukan dengan *seed* yakni koordinat titik-titik piksel pada citra hasil *threshold* dan data DSM. *Seed* untuk *region growing* didapatkan dari koordinat titik-titik *zebra cross* yang tidak terpenuhi pada metode *threshold*. *Seed* tersebut akan berjalan menyebar untuk membentuk jalan dengan adanya pengecekan terlebih dahulu pada titik-titik sekitarnya, untuk lebih jelasnya dapat dilihat pada Gambar 3. 5. Setiap kali pengecekan pada titik-titik sekitar dari *seed*, dilakukan pengecekan selisih nilai ketinggian. Posisi *seed* sekarang akan mengecek pada empat target posisi *seed* berikutnya. Namun pada masing-masing posisi target *seed* berikutnya dilakukan pengecekan terlebih dahulu pada empat titik pengecekan. Jika titik-titik pengecekan tersebut, selisih nilai ketinggiannya kurang dari nilai *threshold* yang telah ditentukan maka titik target posisi *seed* berikutnya akan disimpan dan dijadikan *seed* selanjutnya.



Gambar 3. 5. Alur pengecekan pada *Region growing*

Dengan penggunaan metode *region growing*, dapat meminimalkan kesalahan ekstraksi jalan pada objek-objek jalan pada citra yang memiliki perbedaan ketinggian yang berbeda-beda. Seperti antara jalan dengan gedung-gedung atau bangunan-bangunan lain, jalan di daerah yang satu dengan daerah lainnya, dan juga antara jalan dengan jembatan yang sering kali menjadi objek yang salah untuk dideteksi.

3.2.2.4 Road line filtering

Hasil dari *region growing* dilakukan morfologi untuk memperbaiki citra tersebut kemudian dilakukan *road line filtering*. Metode ini bertujuan untuk membentuk garis jalan dari seluruh proses-proses pengolahan citra yang telah dilakukan sebelumnya.

Pada metode ini, citra hasil *Region growing* pertama-tama dilakukan *thinning*, untuk mendapatkan garis jalan. Setelah itu, digunakan *region props* untuk mendapatkan nilai sudut dari masing-masing garis jalan. Beberapa nilai sudut yang dominan yang akan merepresentasikan bahwa pada nilai sudut tersebut dideteksi adanya jalan. Dari nilai sudut tersebut, dibentuk garis menggunakan operasi *opening* dengan *structuring element* garis. Hasil dari operasi *opening* ini yang menjadi garis-garis jalan pada citra keluaran.

3.2.3 Data Keluaran

Untuk tampilan dari keluaran aplikasi tergambar pada Gambar 3. 6 di bawah.



Gambar 3. 6. Tampilan Keluaran Aplikasi

Data keluaran adalah data yang dihasilkan dari proses yang berjalan pada aplikasi. Pada aplikasi ini, data yang dihasilkan adalah citra yang menggambarkan ruas jalan yang telah terekstraksi menggunakan citra foto udara RGB dan data DSM. Dan juga terdapat hasil *quality* dari pengekstrasian jalan yang dihasilkan, persentase kesempurnaan jalan yang terdeteksi, serta persentase kesempurnaan bukan jalan yang terdeteksi.

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Setelah dijelaskan mengenai proses yang dijalankan untuk melakukan pengekstrasian jalan pada Bab 3, pada Bab 4 ini akan dijelaskan mengenai implementasi yang dilakukan pada proses percobaan.

4.1 Lingkungan Implementasi

Dalam mengimplementasikan pengekstrasian jalan diperlukan beberapa perangkat pendukung sebagai berikut.

4.1.1. Lingkungan Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan pada lingkungan pengembangan aplikasi adalah sebagai sebuah komputer PC atau laptop sebagai komputer pemroses dengan spesifikasi:

- Prosesor : Intel® Core™ i5 CPU 3317U @ 1.70 GHz
- Memori : 4.00 GB
- Tipe Sistem : 64-bit *Operating System*

4.1.2. Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan pada lingkungan pengembangan aplikasi adalah sebagai berikut:

- ❖ Windows 7 sebagai sistem operasi
- ❖ MATLAB R2013a sebagai IDE untuk mengimplementasikan aplikasi

4.2 Implementasi Segmentasi Citra Foto Udara RGB

Pada tahap implementasi ini menjelaskan tentang pembangunan perangkat lunak secara detail termasuk menampilkan kode program bila diperlukan. Segmentasi ini

memiliki tujuan untuk mendeteksi objek-objek yang berwarna hijau pada citra foto udara RGB. Implementasi dari proses segmentasi citra foto udara RGB dapat dilihat pada Kode Sumber 4. 1 di bawah ini.

```

1  function kemungkinanjalan = segmentasi(rgb)
2  rgb = imresize(rgb, 0.4);
3
4  cform = makecform('srgb2lab');
5  lab_he = applycform(rgb,cform);
6  ab = double(lab_he(:, :, 2:3));
7  nrows = size(ab,1);
8  ncols = size(ab,2);
9  ab = reshape(ab,nrows*ncols,2);
10
11 nColors = 4;
12 [cluster_idx, cluster_center] =
13 kmeans(ab,nColors,'distance','sqEuclidean',
14 'Replicates',3);
15 pixel_labels = reshape(cluster_idx,nrows,ncols);

```

Kode Sumber 4. 1. Fungsi Segmentasi

Pada baris empat hingga enam, dilakukan konversi citra ke dalam bentuk $L*a*b$ *color space*. Kemudian, untuk baris ke tujuh hingga 15 merupakan implementasi k-means *clustering*. Pada fungsi segmentasi ini, citra dikelompokkan ke dalam empat *cluster*, seperti pada baris 11, untuk mempermudah.

4.3 Implementasi Deteksi *Zebra cross* Pada Citra Udara RGB

Untuk pendeteksian *zebra cross* sebagai ciri khusus untuk mengenali suatu objek adalah jalan, digunakan fungsi deteksi *zebra cross*. Pendeteksian *zebra cross* menggunakan data RGB dan data DSM sebagai data masukannya. Data RGB digunakan sebagai tempat data di mana objek *zebra cross* ingin dideteksi, sedangkan data DSM digunakan untuk menyesuaikan titik koordinat di mana *zebra cross* yang sudah berhasil dideteksi pada

citra RGB serta mendapatkan nilai ketinggiannya. Implementasi dari pendeteksian *zebra cross* di citra foto udara RGB dapat dilihat pada Kode Sumber 4. 2.

```

1 function zc_dsm = findzc(rgb_local,dsm_local)
2     [height, width, z] = size(rgb_local);
3     [dsm_h,dsm_w,dsm_z] = size(dsm_local);
4     zc_rgb = [];
5
6     H = fspecial('log',[5 4] ,0.3);
7     rgb_log = imfilter(rgb_local,H,'replicate');
8     bw = im2bw(rgb_log);
9
10    bwopen = xor(bwareaopen(bw,20),
11 bwareaopen(bw,80));
12    se = strel('square',4);
13    BW2 = imdilate(bwopen,se);
14    BW2 = bwareaopen(BW2,500);
15
16    block_size = 16;
17    bwd = double(BW2);
18    for h=1:8:height - block_size
19    for w=1:8:width - block_size
20        imtarget = BW2(h:h+block_size-
21 1,w:w+block_size-1,:);
22        imtargetsum = sum(imtarget(:));
23    if imtargetsum > 230
24        zc_rgb =
25 [zc_rgb; [w+block_size/2,h+block_size/2]];
26        end
27    end
28    end

```

Kode Sumber 4. 2. Fungsi Deteksi *Zebra cross*

Sebelum dilakukan pendeteksian *zebra cross*, pada baris enam hingga tujuh pada kode sumber di atas, data RGB di-*filter* menggunakan LoG untuk menemukan garis tepi masing-masing objek di citra. Setelah itu, citra dijadikan dalam bentuk hitam dan putih seperti pada baris delapan. Di mana piksel berwarna putih sebagai foreground, menandakan objek hasil dari filter LoG dan piksel hitam sebagai background. Kemudian, dilakukan morfologi

untuk menghilangkan objek berukuran besar serta objek lain yang kemungkinan bukan *zebra cross* pada baris sepuluh hingga 12.

Agar dapat membedakan objek *zebra cross* dengan objek lainnya, dilakukan morfologi dengan *imdilate*. Pada bagian ini, baris ke 13, objek digemukkan dengan *imdilate*. Baris ke 14, dilakukan morfologi *opening* untuk menghapus objek yang memiliki ukuran terlalu besar setelah dilakukan morfologi *imdilate*, dalam hal ini objek yang dimaksud bukan merupakan objek *zebra cross*.

Blok pada baris 16 merupakan media untuk mendeteksi *zebra cross*. Blok akan berjalan mengikuti perulangan, *looping*, sesuai ukuran citra foto udara RGB, untuk mendeteksi *zebra cross*, seperti pada baris 18 hingga 28. Ukuran blok yang digunakan berukuran 16 x 16 piksel karena ukuran tersebut cukup sesuai dengan ukuran satu objek *zebra cross* pada citra RGB. Pada setiap perulangan, panjang blok diperbaharui dengan penambahan 8 piksel begitu pun nanti dengan lebarnya, hal ini diimplementasikan pada baris 18 hingga 19. Pada baris 20 hingga 21, definisi dari blok yang akan dideteksi.

Untuk menentukan apakah objek yang telah dideteksi pada blok merupakan *zebra cross* atau bukan, dihitung jumlah nilai piksel putih pada blok tersebut. Jika jumlahnya melebihi 230 piksel maka objek tersebut merupakan *zebra cross*, seperti pada baris 22 hingga 23.

Semua *zebra cross* yang telah terdeteksi di citra RGB kemudian disimpan titik koordinatnya untuk dilakukan proses selanjutnya.

4.4 Implementasi Pengekstrasian Jalan

Untuk mengimplementasikan pengekstrasian jalan, digunakan dua metode yang digambarkan pengimplementasiannya pada Kode Sumber 4. 3 dan Kode Sumber 4. 4 di sub sub bab – sub sub bab bawah ini. Dalam melakukan kedua metode tersebut, proses dilakukan secara lokal,

di mana citra hasil segmentasi yang menjadi data masukan, dibagi menjadi 12 bagian untuk mempermudah pemrosesan.

4.4.1 Implementasi *Threshold Titik-titik Zebra cross*

Setelah mendapatkan titik-titik *zebra cross* yang sudah terdeteksi kemudian didapatkan pula nilai ketinggian dari masing-masing titik tersebut. Fungsi *threshold* digunakan untuk mengekstrak jalan dari titik-titik tersebut.

Kode Sumber 4. 3 di bawah merupakan pengimplementasian *threshold* pada titik-titik *zebra cross* yang telah didapatkan dari penggunaan fungsi deteksi *zebra cross* pada sub bab sebelum ini.

```

1  function dsmroad_t = threshold(zc_dsm,dsm_local)
2
3      zc_dsm = sortrows(zc_dsm);
4
5      iq2 = int16((size(zc_dsm,1)+1)/4);
6      q2 = zc_dsm(iq2,1);
7
8      dsmroad_t = (dsm_local < q2 & dsm_local >
9      zc_dsm(1,1));

```

Kode Sumber 4. 3. Fungsi *Threshold*

Nilai-nilai ketinggian pada titik-titik *zebra cross* diurutkan berdasarkan nilai ketinggiannya dari yang paling rendah hingga paling tinggi, seperti pada baris tiga. Kemudian pada baris ke lima, nilai-nilai tersebut dibagi menjadi empat bagian. Nilai-nilai yang diambil untuk dilakukan *threshold* adalah nilai yang berada dalam rentang paling rendah hingga seperempatnya karena dapat dipastikan bahwa nilai-nilai pada bagian atau rentang tersebut merupakan jalan, hal tersebut diimplementasikan pada baris delapan hingga sembilan. Nilai-nilai ketinggian pada rentang tersebut kemudian titik-titik koordinatnya dijadikan titik-titik jalan, dan kemudian diproses

lebih lanjut untuk pengesktrasian jalan pada proses *region growing*.

4.4.2 Implementasi *Region growing*

Tahap ini merupakan tahap terakhir dari pengekstrasian jalan. Tahap *region growing* melanjutkan dari hasil pengekstrasian jalan yang sudah dilakukan fungsi *threshold*. Untuk pengimplementasian *region growing* ditunjukkan oleh Kode Sumber 4. 4 di bawah ini.

```

1  function dsmroad_t =
2  region_growing(dsmroad_t,dsm_local,kemungkinanj
3  alan_local)
4
5  boundaries = bwboundaries(dsmroad_t);
6  queue = [];
7
8  for k=1:size(boundaries,1)
9      b = boundaries{k};
10     for i=1:size(b,1)
11         queue = [queue;[b(i,1), b(i,2)]];
12     end
13 end
14
15 threshold = 2;
16 check =
17 false(size(dsm_local,1),size(dsm_local,2));
18 while 1
19     x = queue(1,1);
20     y = queue(1,2);
21
22     queue = queue(2:end,:);
23
24     x1 = x+1;
25     y1 = y;
26     if size(dsmroad_t,1) >= x+2 && ...
27         x-2 > 0 && ...
28         size(dsmroad_t,2) >= y+2 && ...
29         y-2 > 0 && ...
30         dsmroad_t(x1,y1) == false && ...

```

```

31         check(x1,y1)==false && ...
32         dsmroad_t(x1,y1)== false && ...
33         kemungkinanjalan_local(x1,y1) == true
34
35         counter = 0;
36         selisih = dsm_local(x1,y1) -
37 dsm_local(x1+1,y1);
38         selisih = abs(selisih);
39         if selisih <= threshold
40             counter = counter+1;
41         end
42         selisih = dsm_local(x1,y1) -
43 dsm_local(x1-1,y1);
44         selisih = abs(selisih);
45         if selisih <= threshold
46             counter = counter+1;
47         end
48         selisih = dsm_local(x1,y1) -
49 dsm_local(x1,y1+1);
50         selisih = abs(selisih);
51         if selisih <= threshold
52             counter = counter+1;
53         end
54         selisih = dsm_local(x1,y1) -
55 dsm_local(x1,y1-1);
56         selisih = abs(selisih);
57         if selisih <= threshold
58             counter = counter+1;
59         end
60
61         if counter == 4
62             queue = [queue;[x1, y1]];
63             dsmroad_t(x1,y1) = true;
64         end
65         check(x+1,y) = true;
66     end
67
68     x1 = x-1;
69     y1 = y;
70     ...
71
72     x1 = x;
73     y1 = y-1;
74     ...

```

```

75
76     x1 = x;
77     y1 = y+1;
78     ...
79
80     if size(queue,1) == 0
81         break;
82     end
83 end

```

Kode Sumber 4. 4. Fungsi *Region growing*

Dari fungsi deteksi *zebra cross* didapatkan titik-titik *zebra cross*. Kemudian titik-titik tersebut pada fungsi *threshold* diproses untuk pengekstrasian jalan, sisanya dilanjutkan pada proses *region growing* ini. Titik atau *seed* yang akan dicek pada proses *region growing*, didapatkan dari titik-titik pinggir dari objek hasil *threshold*. Pada baris ke lima hingga 13, *bwboundaries* digunakan untuk mendapatkan titik-titik pinggir tersebut.

Baris ke 24 dan 25 merupakan salah satu definisi dari titik yang akan menjadi target posisi seed berikutnya, begitu untuk ke empat titik lainnya yakni pada baris 68 hingga 77.

Seed akan memeriksa empat titik atau piksel tetangganya, dan masing-masing piksel tetangga tersebut akan mengecek empat piksel tetangga yang dimilikinya juga.

Pada baris 26 hingga 66, *Statement if* pada kode sumber di atas merupakan contoh pengecekan pada satu titik dari target posisi *seed* berikutnya, dan di dalamnya terdapat empat *statement if* lagi untuk mengecek empat titik pengecekan tetangga dari titik tersebut. Pengecekan dilakukan untuk mengecek selisih nilai ketinggian. Pada baris 36 hingga 38 sebagai contoh untuk menghitung selisih nilai ketinggian. Jika selisih dari semua titik target posisi seed berikutnya dengan titik-titik pengecekannya tersebut nilai ketinggiannya kurang dari sama dengan *threshold* maka titik target posisi seed berikutnya akan dijadikan seed selanjutnya, seperti pada baris 61 hingga 65. Dengan melakukan pengecekan tersebut, pengekstrasian jalan dapat lebih akurat

karena pada jalan yang naik, atau pada jembatan, proses *Region growing* dapat mendeteksi atau mengestraksi jalan tersebut.

4.5 Implementasi *Road Line Filtering*

Pada metode *road line filtering*, jalan yang sudah berhasil diekstraksi sampai tahap *region growing*, dibuat garis jalannya menggunakan metode ini. Pengimplementasian dari *road line filtering* ada pada Kode Sumber 4. 5 di bawah.

```

1  function [hasil_thinning] =
2  thinning(hasil_morfologi)
3      morfologi_thin =
4  bwmorph(hasil_morfologi, 'thin', Inf);
5
6      for i=1:50:size(morfologi_thin,1)
7          for j=1:3
8              morfologi_thin(:,i+j) = false;
9              morfologi_thin(i+j,:) = false;
10         end
11     end
12
13     BW3_rp =
14     regionprops(morfologi_thin, 'orientation');
15     sudut_by_index = zeros(200,1);
16
17     ...
18
19     for i=1:size(sudut,1)
20         if sudut(i,2)>=1
21             se = strel('line', 400, sudut(i,1));
22             hasil_thinning =
23             or(hasil_thinning, imopen(hasil_morfologi, se));
24         end
25     end

```

Kode Sumber 4. 5 Fungsi *Road Line Filtering*

Pada fungsi *road line filtering*, citra hasil dari *Region growing* dilakukan *thinning*, seperti pada baris tiga hingga empat. Kemudian, hasil dari *thinning* berupa garis panjang, dipotong-

dipotong agar mudah untuk pengambilan fitur sudut, hal tersebut diimplementasikan pada baris enam hingga 11. Pada baris 13 hingga 15, hasil potong-potongan garis tersebut dilakukan ekstraksi fitur sudut dengan menggunakan metode *region props*. Dari sudut-sudut yang sudah didapatkan, dilakukan morfologi opening garis, seperti pada baris 19 hingga 25.

BAB V

UJI COBA DAN EVALUASI

Pada bab kelima ini akan dijelaskan mengenai skenario dan uji coba perangkat lunak yang telah dibangun. Selain itu, hasil uji coba akan dievaluasi kinerjanya sehingga dapat diputuskan apakah perangkat lunak ini mampu menyelesaikan permasalahan yang telah dirumuskan di awal.

5.1 Lingkungan Uji Coba

Sebelumnya, perlu diketahui lingkungan uji coba, baik perangkat keras maupun perangkat lunak, yang digunakan pada uji coba Tugas Akhir ini. Lingkungan tersebut ditunjukkan pada Tabel 5.1 berikut ini.

Tabel 5. 1. Spesifikasi lingkungan uji coba

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i5-3317U CPU @ 1.70 GHz Memori: 4.00 GB
Perangkat lunak	Sistem Operasi: Microsoft Windows 7 64-bit Perangkat Pengembang: Matlab R2013a

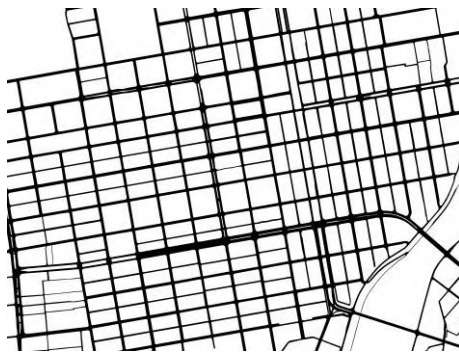
5.2 Data Uji Coba

Data uji coba yang digunakan pada Tugas Akhir ini adalah citra foto udara dalam bentuk RGB dan data DSM yang menandakan ketinggian dari masing-masing objek pada citra foto udara tersebut. Citra foto udara RGB yang digunakan merupakan citra dari kota Sapporo di Jepang yang diambil pada tahun 2010. Citra RGB yang digunakan, beresolusi sangat tinggi yakni 10.000 x 7.500 piksel, sedangkan untuk data DSM berukuran 4000 x 3000 piksel.

5.3 Skenario Uji Coba

Sebelum melakukan uji coba, perlu ditentukan skenario yang akan digunakan dalam uji coba. Melalui skenario ini, perangkat lunak diuji apakah sudah berjalan dengan benar dan memiliki performa yang baik sesuai dengan kondisi yang ditentukan. Terdapat tiga macam skenario dari uji coba dengan menggunakan kombinasi nilai pada beberapa metode untuk mengekstraksi jalan. Hasil data keluaran dari kombinasi metode-metode tersebut dapat dilihat pada LAMPIRAN.

Dari uji coba yang dilakukan menggabungkan atau penggunaan beberapa metode pengekstraksian jalan dapat mempengaruhi hasil *quality*. Dalam skenario uji coba dihitung perhitungan *quality*, persentase kesempurnaan jalan yang berhasil diekstrak, persentase kesempurnaan objek bukan jalan yang berhasil diekstrak, dan waktu pemrosesannya (*running time*). Hasil ekstraksi jalan akan dibandingkan dengan *ground truth* yakni model jalan yang asli dan sesuai dengan citra foto udara RGB serta data DSM. Model jalan tersebut dibuat secara manual dari peta digital. Untuk lebih jelas, model jalan *ground truth* dapat dilihat pada Gambar 5. 1 di bawah.



Gambar 5. 1. *Ground Truth*

Jalan yang terekstraksi dan sesuai dengan jalan pada *ground truth* dinotasikan dengan *True Positives* (TP) yakni

suatu kondisi di mana hasil ekstraksi jalan sama dengan *ground truth*. *True Negatives* (TN) adalah kondisi di mana hasil ekstraksi objek bukan jalan sama dengan *ground truth*. *False Positives* (FP) merupakan kondisi di mana jalan yang terekstraksi, tidak sama dengan *ground truth*. Dan, *False Negatives* (FN) adalah kondisi di mana objek bukan jalan yang terdeteksi, tidak sama dengan *ground truth* [12].

Persamaan-persamaan untuk pengukuran evaluasi dari pengekstraksian jalan sebagai berikut :

E_{road} : seberapa besar persentase kesempurnaan dari jalan yang berhasil diekstraksi. Kesempurnaan adalah presentase dari terekstraksinya jalan yang seharusnya diekstraksi. Nilai E_{road} dihitung dengan

$$E_{road} = \frac{TP}{TP + FN}$$

$E_{non\ road}$: seberapa besar persentase kesempurnaan objek bukan jalan yang berhasil diekstraksi. Kesempurnaan adalah presentase dari terekstraksinya objek bukan jalan yang seharusnya diekstraksi. Nilai $E_{non\ road}$ dihitung dengan

$$E_{non\ road} = \frac{TN}{TN + FN}$$

Quality: sebuah ukuran dari kebagusan hasil akhir pengekstrasian jalan. Hasil *quality* merupakan kombinasi antara perhitungan kesempurnaan dari jalan yang berhasil diekstrak dan ketepatannya. Nilai *quality* dihitung dengan rumus

$$Quality = \frac{TP}{TP + FP + FN}$$

Perhitungan untuk menentukan nilai-nilai dari TP, TN, FP, dan FN dilakukan dengan membandingkan masing-masing piksel antara piksel-piksel hasil ekstraksi jalan dengan piksel-piksel dari gambar *ground truth*. Kemudian, nilai-nilai yang didapatkan dijumlahkan untuk mendapatkan total masing-masing TP, TN, FP, dan FN.

5.3.1 Skenario 1

Skenario 1 yakni uji coba yang membandingkan hasil perhitungan *quality*, persentase kesempurnaan dari jalan yang berhasil diekstraksi, persentase kesempurnaan dari objek bukan jalan yang berhasil diekstraksi, dan waktu untuk pemrosesannya (*running time*) pada metode *threshold* dengan nilai persentase penggunaan *zebra cross* yang berbeda-beda. Untuk hasil uji coba dan evaluasi yang didapatkan pada skenario ini dapat dilihat pada tabel Tabel 5. 1 di bawah.

Tabel 5. 1. Hasil dari Uji Coba Skenario 1

Persentase <i>Zebra cross</i> yang digunakan	<i>Quality</i>	Eroad	Enon road	<i>Running time</i> (detik)
25%	34,8%	51,8%	86,5%	169,44
33%	36,5%	63,9%	79,2%	241,421
50%	37,6%	83,9%	66,0%	331,016
75%	29,9%	97,7%	37,6%	206,676
80%	27,8%	98,5%	29,9%	173,197



Gambar 5. 2. Zebra cross terdeteksi

Dengan menggunakan metode *threshold* pada skenario 1, hasil *quality* terbaik yang didapatkan yakni 37,6%. Berdasarkan data hasil uji coba pada skenario 1, semakin banyak jumlah titik *zebra cross* yang digunakan maka semakin banyak jalan yang berhasil diekstrak, hal tersebut dapat dilihat pada peningkatan nilai *Eroad*. Namun, semakin banyak jalan yang ingin diekstraksi, hasil *quality* jalan yang dihasilkan berkurang, hal tersebut digambarkan dari penurunan nilai *E non road*. Hal ini dikarenakan, penggunaan jumlah titik-titik *zebra cross* yang semakin banyak yakni titik-titik *zebra cross* yang berada pada posisi tinggi, akan menyebabkan titik-titik ketinggian yang di bawah titik-titik *zebra cross* tersebut akan dianggap sebagai jalan, padahal tidak semuanya objek di bawah titik tersebut adalah jalan.

Tidak semua *zebra cross* pada citra foto udara RGB berhasil dideteksi. Pada Gambar 5. 2, terdapat contoh dari *zebra cross* yang berhasil dideteksi yang ditandai dengan kotak merah, dan *zebra cross* yang tidak berhasil dideteksi. Beberapa faktor yang menyebabkan adanya *zebra cross* yang tidak terdeteksi yakni, pada saat penghitungan jumlah piksel putih dengan menggunakan blok, posisi dari blok tidak tepat karena pergeseran posisi blok setiap delapan piksel, serta penggunaan morfologi *opening* yang menyebabkan elemen *zebra cross* yang terlalu besar dan terlalu kecil menjadi hilang.

Kemudian, *running time* pada skenario 1 rata-rata terhitung cepat yakni sekitar 220 detik.

5.3.2 Skenario 2

Skenario 2 yakni uji coba yang membandingkan hasil perhitungan *quality*, persentase kesempurnaan dari jalan yang berhasil diekstraksi, persentase kesempurnaan dari objek bukan jalan yang berhasil diekstraksi, dan waktu untuk pemrosesannya (*running time*) pada metode *threshold* dengan *region growing*. Untuk hasil uji coba dan evaluasi yang

didapatkan pada skenario ini dapat dilihat pada tabel Tabel 5. 2.

Tabel 5. 2. Hasil dari Uji Coba Skenario 2

Persentase <i>Zebra cross</i> yang digunakan	Selisih nilai ketinggian <i>Region growing</i>	<i>Quality</i>	Eroad	Enomroad	<i>Running time</i>
25%	2	42,2%	86,5%	71,1%	657,094
	4	41,0%	87,9%	68,5%	623,693
	7	39,6%	88,5%	65,9%	429,262
30%	2	39,9%	92,8%	63,5%	189,448
	4	38,6%	94,0%	60,4%	370,773
	7	37,2%	94,5%	57,4%	296,669
50%	2	36,1%	94,9%	55,0%	245,77
	4	34,9%	95,8%	51,9%	259,323
	7	33,7%	96,2%	48,8%	282,234
75%	2	27,7%	99,0%	28,8%	250,606
	4	27,2%	99,2%	27,1%	157,134
	7	26,8%	99,3%	25,4%	268,143
80%	2	26,0%	99,5%	22,0%	299,286
	4	25,7%	99,6%	20,8%	173
	7	25,4%	99,6%	19,3%	157,645

Dengan menggunakan metode *threshold* dan *region growing* pada skenario 2, hasil *quality* terbaik yang didapatkan yakni 42,2%. Berdasarkan data hasil uji coba pada skenario 2, *zebra cross* yang digunakan pada metode *threshold* lebih sedikit akan lebih akurat karena penggunaan *zebra cross* yang sedikit dapat mengurangi kesalahan ekstraksi jalan. Selisih ketinggian pada metode *region growing* berpengaruh juga

pada hasil *quality* skenario uji coba ini. Berdasarkan tabel hasil uji coba di atas, nilai selisih ketinggian pada *Region growing* yang bagus selalu pada nilai dua, berapa pun nilai *zebra cross* yang digunakan pada metode *threshold*-nya. Hal ini dikarenakan, selisih yang lebih kecil akan lebih aman, tidak menghasilkan banyak kesalahan ekstraksi pada saat melakukan *region growing* dibandingkan menggunakan nilai selisih yang lebih besar. Kemudian, rata-rata *running time* pada skenario 2 cenderung lebih lama karena penambahan metode *region growing* yakni 310 detik.

5.3.3 Skenario 3

Skenario 3 yakni uji coba yang membandingkan skenario 2, dengan menggunakan metode tambahan *road line filtering*. Untuk hasil uji coba dan evaluasi yang didapatkan pada skenario ini dapat dilihat pada Tabel 5. 3 di bawah.

Tabel 5. 3. Hasil dari Uji Coba Skenario 3

Persentase <i>Zebra cross</i> yang digunakan	Selisih Nilai Ketinggian <i>Region growing</i>	<i>Quality</i>	<i>Eroad</i>	<i>Enonroad</i>	<i>Running time</i>
25%	2	37,9%	90,1%	61,8%	456,016
	4	36,1%	90,9%	58,2%	565,264
	7	34,5%	91,5%	54,5%	404,572
30%	2	35,0%	96,3%	51,6%	260,317
	4	33,5%	96,9%	47,7%	312,672
	7	32,2%	97,6%	43,9%	373,516
50%	2	31,4%	92,3%	42,3%	391,468
	4	30,2%	97,8%	38,4%	277,418
	7	29,0%	98,2%	34,3%	431,283

Tabel 5. 4. Hasil dari Uji Coba Skenario 3 (Lanjutan)

Persentase <i>Zebra cross</i> yang digunakan	Selisih Nilai Ketinggian <i>Region growing</i>	<i>Quality</i>	Eroad	Enonroad	<i>Running time</i>
75%	2	24,7%	99,7%	16,3%	264,288
	4	24,5%	99,8%	15,0%	266,411
	7	24,1%	99,8%	13,3%	401,312
80%	2	23,8%	99,8%	12,0%	426,282
	4	23,6%	99,8%	11,2%	370,125
	7	23,4%	99,9%	10,0%	269,827

Dengan menggunakan metode *threshold* dan *region growing* pada skenario 2, hasil *quality* terbaik yang didapatkan yakni 37,9%. Secara garis besar, hasil skenario 3 tidak jauh berbeda dengan skenario 2 karena skenario ini merupakan lanjutan dari skenario 2. Dengan menambahkan metode *road line filtering*, hasil *quality* pada skenario 3 cenderung berkurang dibandingkan dengan skenario 2, dikarenakan metode *road line filtering* belum menggunakan nilai parameter-parameter yang cocok yakni sudut, panjang garis, dan dilasi garis. Kemudian, *running time* rata-rata pada skenario ini terhitung paling lama yakni 364 detik.

5.4 Analisa Hasil Uji Coba

Setelah dilakukan uji coba menggunakan tiga skenario, didapatkan hasil terbaik untuk pengekstrasian jalan pada tugas akhir ini adalah dengan menggunakan metode *threshold* yang digabung dengan *region growing* yakni dengan skenario 2.

Hasil masing-masing metode yang digunakan pada masing-masing skenario sangat bergantung pada banyaknya jumlah *zebra cross* yang digunakan, serta selisih nilai

ketinggian *Region growing* untuk skenario 2 dan 3, karena nilai-nilai tersebut sebagai parameter penting dalam melakukan pengekstrasian jalan.

Berikut tampilan hasil uji coba dengan nilai *quality* terbaik pada masing-masing skenario.



Gambar 5. 3. Hasil Uji Coba Skenario 1 Dengan *Quality* Terbaik, 37,6%



Gambar 5. 4. Hasil Uji Coba Skenario 2 Dengan *Quality* Terbaik, 42,2%



Gambar 5. 5. Hasil Uji Coba Skenario 3 Dengan *Quality* Terbaik, 37,9%

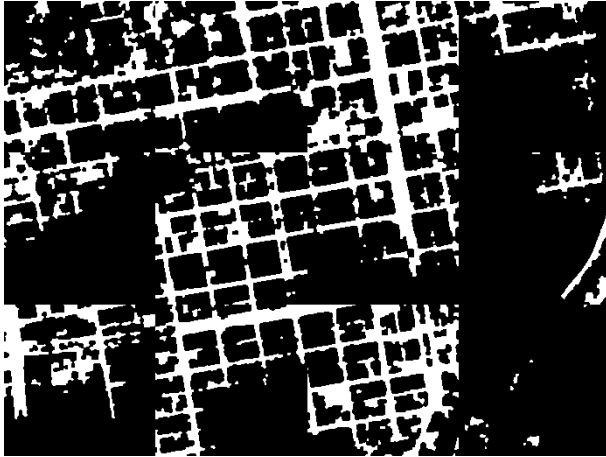
Pada Gambar 5. 3. persentase banyaknya *zebra cross* terdeteksi yang digunakan adalah 50% dari jumlah titik-titik *zebra cross* yang terdeteksi. Terdapat beberapa jalan yang tidak terdeteksi, berwarna hitam dominan, tertutup, disebabkan

titik-titik *zebra cross* yang digunakan berada pada posisi yang rendah, sehingga jalan di posisi yang tinggi tidak dapat terekstraksi.

Kemudian, untuk Gambar 5. 4, digunakan 25% dari jumlah *zebra cross* yang terdeteksi dan nilai selisih ketinggian pada *Region growing* bernilai 2. Pada sudut kanan bawah gambar tersebut, seharusnya terdapat objek jembatan dan sungai namun tidak dapat terdeteksi. Hal tersebut disebabkan *threshold* tidak dapat berjalan dengan baik karena *zebra cross* yang digunakan terlalu sedikit. Namun nilai *quality* yang dihasilkan oleh Gambar 5. 4. lebih baik dibandingkan dengan Gambar 5. 3. karena pada sudut kanan gambar tersebut lebih besar persentase objek yang bukan jalan sehingga hasil hitam dominan, menandakan sebagai objek bukan jalan, pada sudut tersebut tidak menyebabkan penurunan *quality* pengestrasian jalan.

Dan yang terakhir, Gambar 5. 5. Penggunaan *road line filtering* menghasilkan nilai *quality* yang berkurang dibandingkan dengan metode pada skenario 2. Hal tersebut disebabkan, nilai persentase $E_{non-road}$ yang berkurang yang berarti persentase objek bukan jalan berkurang. Objek yang seharusnya bukan jalan dideteksi sebagai jalan. Ukuran dari objek bukan jalan menjadi lebih kecil dari seharusnya, menyebabkan jalan yang terekstraksi semakin besar.

LAMPIRAN



Gambar Lampiran 1. 1. Hasil Skenario 1, Nilai *threshold* 1 sampai 0,25 dari jumlah *zebra cross* yang terdeteksi



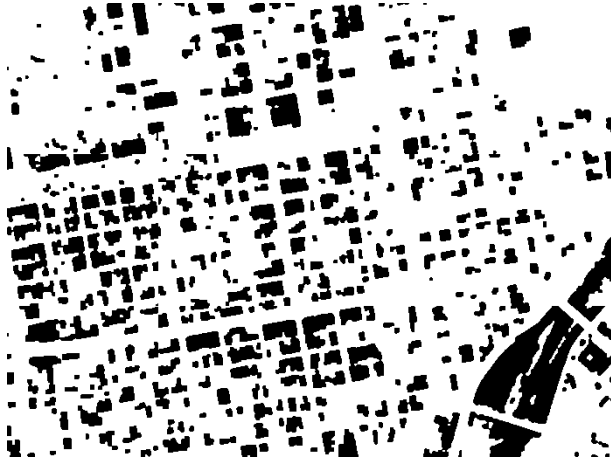
Gambar Lampiran 1. 2. Hasil Skenario 1, Nilai *threshold* 1 sampai 0,3 dari jumlah *zebra cross* yang terdeteksi



Gambar Lampiran 1. 3. Hasil Skenario 1, Nilai *threshold* 1 sampai 0,5 dari jumlah *zebra cross* yang terdeteksi



Gambar Lampiran 1. 4. Hasil Skenario 1, Nilai *threshold* 1 sampai 0,75 dari jumlah *zebra cross* yang terdeteksi



Gambar Lampiran 1. 5. Hasil Skenario 1, Nilai *threshold* 1 sampai 0,8 dari jumlah *zebra cross* yang terdeteksi



Gambar Lampiran 2. 1. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,25 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 2 pada metode *Region growing*



Gambar Lampiran 2. 2. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,25 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 4 pada metode *Region growing*



Gambar Lampiran 2. 3. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,25 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 7 pada metode *Region growing*



Gambar Lampiran 2. 4. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,3 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 2 pada metode *Region growing*



Gambar Lampiran 2. 5. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,3 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 4 pada metode *Region growing*



Gambar Lampiran 2. 6. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,3 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 7 pada metode *Region growing*



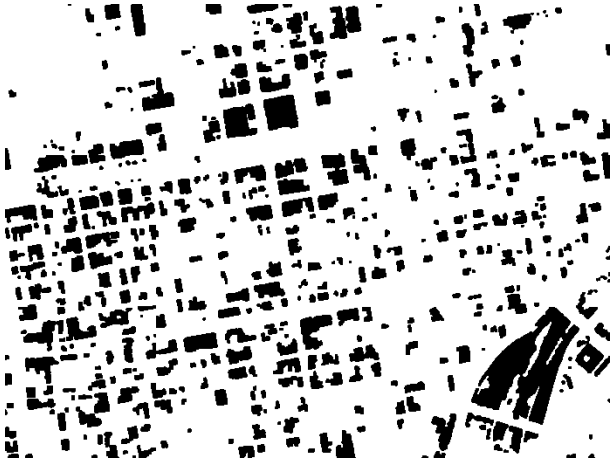
Gambar Lampiran 2. 7. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,5 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 2 pada metode *Region growing*



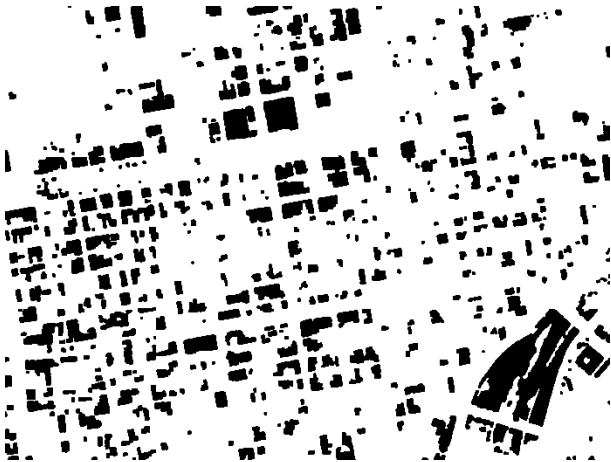
Gambar Lampiran 2. 8. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,5 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 4 pada metode *Region growing*



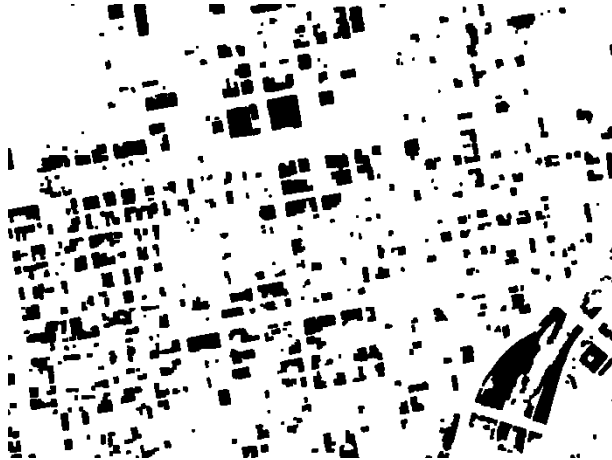
Gambar Lampiran 2. 9. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,5 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 7 pada metode *Region growing*



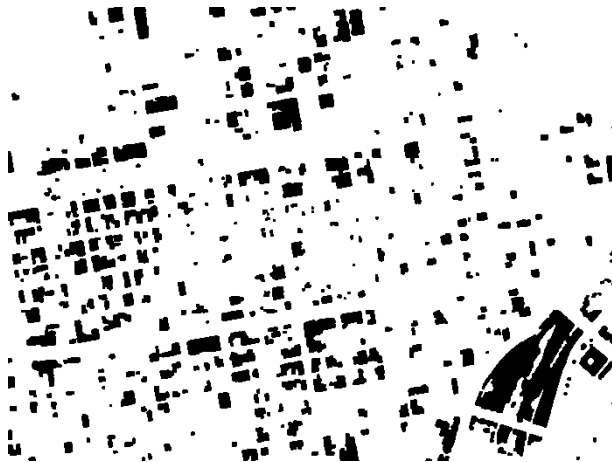
Gambar Lampiran 2. 10. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,75 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 2 pada metode *Region growing*



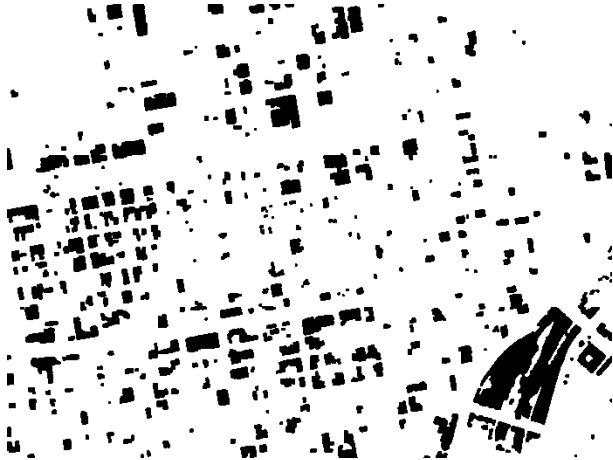
Gambar Lampiran 2. 11. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,75 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 4 pada metode *Region growing*



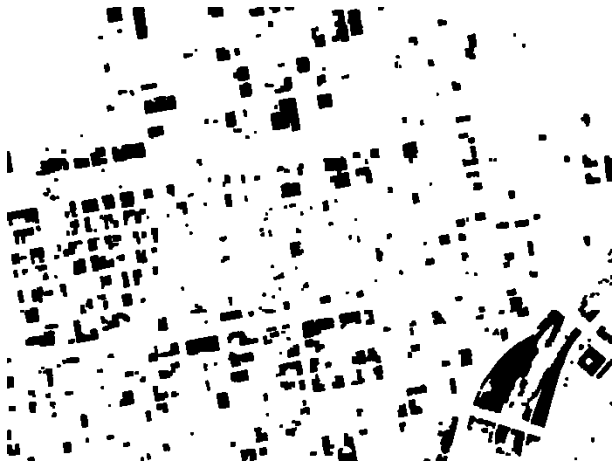
Gambar Lampiran 2. 12. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,75 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 7 pada metode *Region growing*



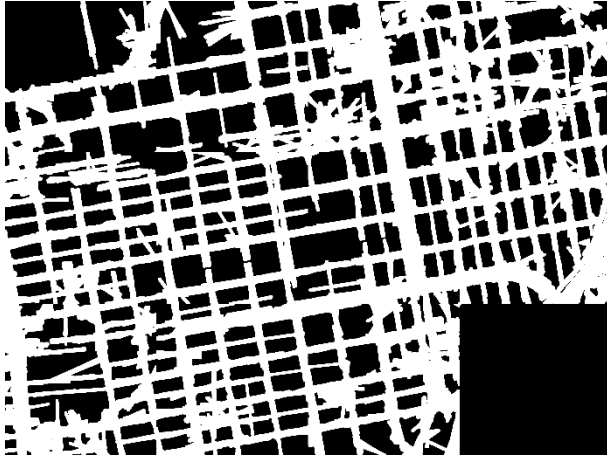
Gambar Lampiran 2. 13. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,8 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 2 pada metode *Region growing*



Gambar Lampiran 2. 14. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,8 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 4 pada metode *Region growing*



Gambar Lampiran 2. 15. Hasil Skenario 2, Nilai *threshold* 1 sampai 0,8 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 7 pada metode *Region growing*



Gambar Lampiran 3. 1. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,25 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 2 pada metode *Region growing*



Gambar Lampiran 3. 2. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,25 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 4 pada metode *Region growing*



Gambar Lampiran 3. 3. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,25 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 7 pada metode *Region growing*



Gambar Lampiran 3. 4. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,3 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 2 pada metode *Region growing*



Gambar Lampiran 3. 5. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,3 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 4 pada metode *Region growing*



Gambar Lampiran 3. 6. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,3 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 7 pada metode *Region growing*



Gambar Lampiran 3. 7. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,5 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 2 pada metode *Region growing*



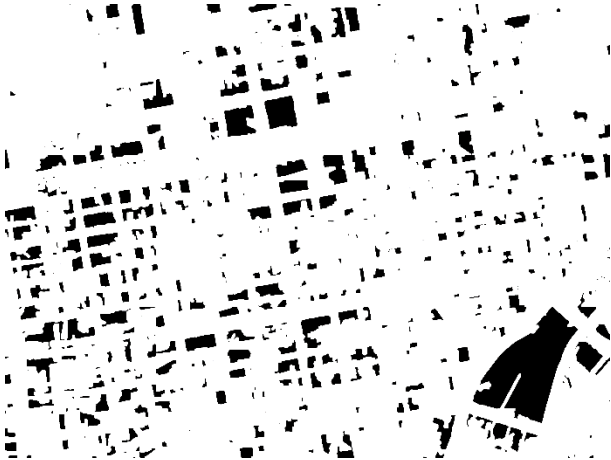
Gambar Lampiran 3. 8. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,5 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 4 pada metode *Region growing*



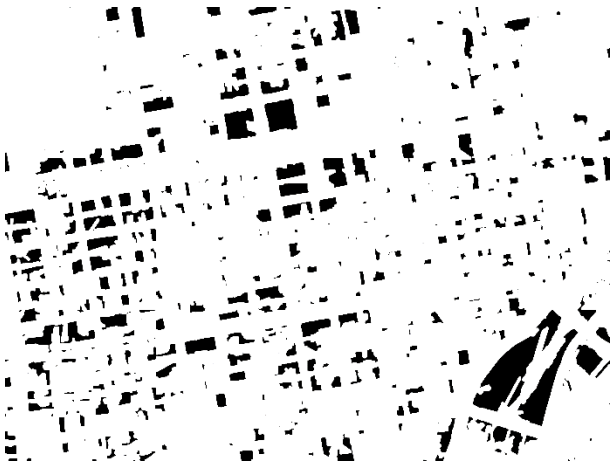
Gambar Lampiran 3. 9. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,5 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 7 pada metode *Region growing*



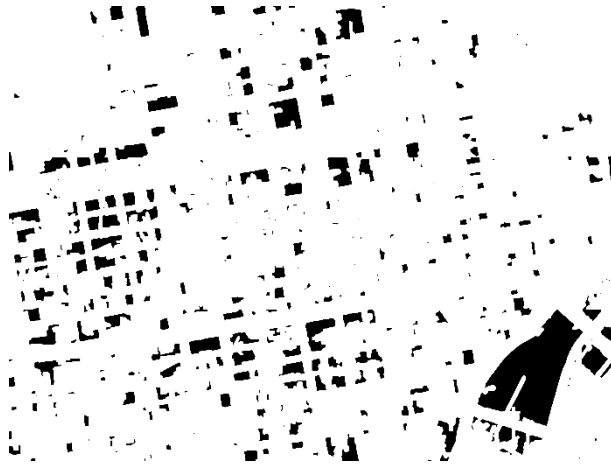
Gambar Lampiran 3. 10. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,75 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 2 pada metode *Region growing*



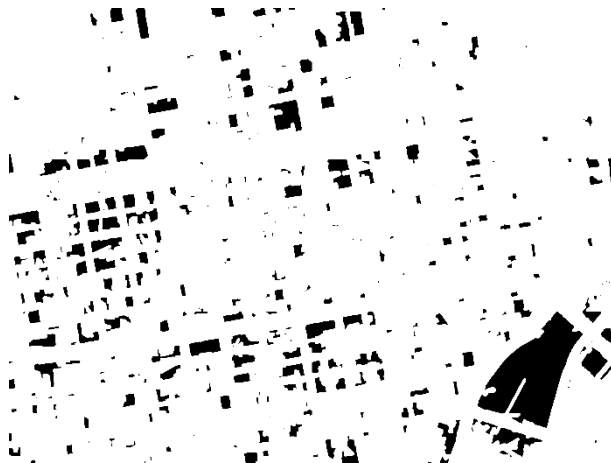
Gambar Lampiran 3. 11. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,75 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 4 pada metode *Region growing*



Gambar Lampiran 3. 12. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,75 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 7 pada metode *Region growing*

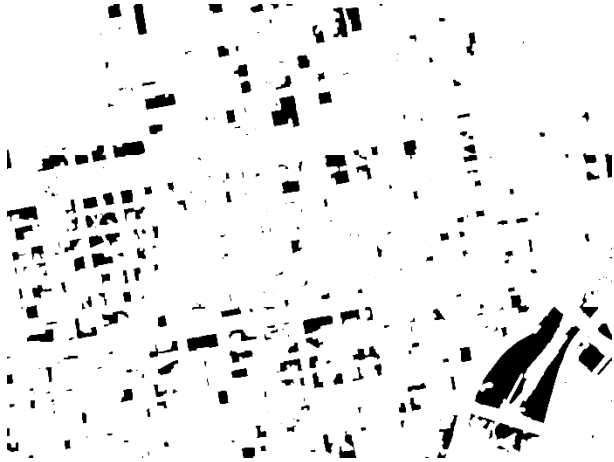


Gambar Lampiran 3. 13. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,8 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 2 pada metode *Region growing*



Gambar Lampiran 3. 14. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,25 dari jumlah *zebra cross* yang terdeteksi pada

metode *Threshold*, Nilai *threshold* 4 pada metode *Region growing*



Gambar Lampiran 3. 15. Hasil Skenario 3, Nilai *threshold* 1 sampai 0,8 dari jumlah *zebra cross* yang terdeteksi pada metode *Threshold*, Nilai *threshold* 7 pada metode *Region growing*

BAB VI

KESIMPULAN DAN SARAN

Bab VI ini membahas tentang kesimpulan yang didasari oleh hasil uji coba pada bab sebelumnya. Kesimpulan tersebut nantinya menjawab rumusan masalah yang telah ada pada pendahuluan. Selain itu, juga terdapat saran sebagai acuan untuk mengembangkan topik Tugas Akhir ini lebih lanjut di masa depan.

6.1 Kesimpulan

Dari hasil uji coba yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Dari skenario uji coba yang telah dilakukan, hasil *quality* dari pengestrasian jalan menggunakan metode *threshold* ditambah *Region growing* merupakan metode terbaik.
2. Dari skenario uji coba yang telah dilakukan, hasil *quality* terbaik yakni 42, 2%, dengan nilai Eroad 86,5% dan nilai Enon road 71,1%.
3. Metode *road line filtering* belum dapat membuat pengestrasian jalan menjadi lebih baik.

6.2 Saran

Saran yang diberikan untuk pengembangan perangkat lunak ini adalah:

1. Penggunaan metode *road line filtering* dapat menggunakan pendeteksian marka jalan sebagai acuan untuk mendapatkan garis sudut dari jalan.
2. Analisa pengambilan sudut pada saat ekstraksi fitur sudut di metode *road line filtering* dapat dilakukan, panjang garis yang cocok untuk digunakan untuk opening garis, serta dilasi garis jalan.

Daftar Pustaka

- [1] N. Sun, J. X. Zhang, G. M. Huang, Z. Zhao, and L. J. Lu, "Review of Road Extraction Methods from SAR Image," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 17, p. 12245, Mar. 2014.
- [2] D. Herumurti, K. Uchimura, G. Koutaki, and T. Uemura, "Urban Road Network Extraction Based on Zebra Crossing Detection from a Very High Resolution RGB Aerial Image and DSM Data," 2013, pp. 79–84.
- [3] "Aerial photography," *Wikipedia, the free encyclopedia*. 23-Mar-2016.
- [4] "Digital elevation model," *Wikipedia, the free encyclopedia*. 19-Apr-2016.
- [5] "Opening (morphology)," *Wikipedia, the free encyclopedia*. 16-Feb-2016.
- [6] "Closing (morphology)," *Wikipedia, the free encyclopedia*. 18-Mar-2014.
- [7] "Image segmentation," *Wikipedia, the free encyclopedia*. 07-May-2016.
- [8] "Thresholding (image processing)," *Wikipedia, the free encyclopedia*. 12-Jan-2016.
- [9] "Region growing," *Wikipedia, the free encyclopedia*. 24-Jan-2016.
- [10] "Thinning (morphology)," *Wikipedia, the free encyclopedia*. 14-Apr-2013.
- [11] "Measure properties of image regions - MATLAB regionprops." [Online]. Available: <http://www.mathworks.com/help/images/ref/regionprops.html>. [Accessed: 23-May-2016].
- [12] C. Heipke, H. Mayer, and C. Wiedemann, "Evaluation of Automatic Road Extraction," in *3D Reconstruction and Modelling of Topographic Objects*, 3–4W2 ed., vol. 32, Technische Universitat M'unchen, D–80290 Munich,

Germany: *International Archives of Photogrammetry and Remote Sensing*, 1997, pp. 151–157.

BIODATA PENULIS



Nurul Supremasinta Putri, lahir di Jakarta pada tanggal 9 Maret 1993. Penuliss menempuh pendidikan mulai dari TK Al Huda Bekasi (1999), SDIT Thariq Bin Ziyad Bekasi (1999-2005), SMPIT Thariq Bin Ziyad Bekasi (2005-2008), SMAIT Thariq Bin Ziyad Bekasi (2008-2011), Biologi, Universitas Airlangga (2011-2012), dan sekarang sedang menjalani pendidikan S1 Teknik Informatika di ITS. Penulis aktif dalam organisasi

Himpunan Mahasiswa Teknik Computer (HMTC). Diantaranya adalah menjadi staff departemen dalam negeri himpunan mahasiswa teknik computer ITS 2013-2014, anggota komisi pemilihan umum ITS 2014, dan staff kementerian kebijakan kampus badan eksekutif mahasiswa ITS 2014-2015. Penulis juga aktif dalam kegiatan kepanitiaan Schematics. Diantaranya penulis pernah menjadi staff national logic competition Schematics 2013 dan 2014. Penulis mengambil bidang minat Komputasi Cerdas Visual (KCV). Komunikasi dengan penulis dapat melalui email: **supremasinta@gmail.com**.