



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IF184802

PENGEMBANGAN MODUL PENGOLAH DATA CITRA UNTUK APLIKASI PEMETAAN KERUSAKAN JALAN

IVAN AGUNG PANDAPOTAN
NRP 05111540000166

Dosen Pembimbing
Daniel O. Siahaan, S.Kom., M.Sc., Pd.Eng.
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR - IF184802

PENGEMBANGAN MODUL PENGOLAH DATA CITRA UNTUK APLIKASI PEMETAAN KERUSAKAN JALAN

**IVAN AGUNG PANDAPOTAN
NRP 0511154000166**

**Dosen Pembimbing
Daniel O. Siahaan, S.Kom., M.Sc., PD.Eng.
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.**

**DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019**

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - IF184802

**IMAGE DATA PROCESSING MODULE
DEVELOPMENT FOR ROAD DAMAGE MAPPING
APPLICATION**

**IVAN AGUNG PANDAPOTAN
NRP 05111540000166**

Supervisors

**Daniel O. Siahaan, S.Kom., M.Sc., PD.Eng.
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN
PENGEMBANGAN MODUL PENGOLAH DATA
CITRA UNTUK APLIKASI PEMETAAN
KERUSAKAN JALAN

TUGAS AKHIR

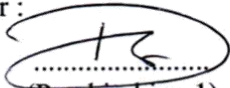
Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Teknik Informatika
Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:


IVAN AGUNG PANDAPOTAN
NRP : 05111540000166

Disetujui oleh Dosen Pembimbing Tugas Akhir :

1. Daniel O. Siahaan, S.Kom., M.Sc.,
PD.Eng.
NIP: 19741123 200604 1 001


.....
(Pembimbing 1)

2. Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.
NIP: 19751220 200112 2 002


.....
(Pembimbing 2)

SURABAYA
JUNI 2019

[Halaman ini sengaja dikosongkan]

PENGEMBANGAN MODUL PENGOLAH DATA CITRA UNTUK APLIKASI PEMETAAN KERUSAKAN JALAN

Nama Mahasiswa : Ivan Agung Pandapotan
NRP : 05111540000166
Jurusan : Departemen Informatika FTIK-ITS
Dosen Pembimbing 1 : Daniel O. Siahaan., S.Kom., M.Sc.,
PD.Eng.
Dosen Pembimbing 2 : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.

ABSTRAK

Jalan merupakan suatu prasarana transportasi darat yang merupakan kebutuhan sehari-hari bagi setiap manusia. Sampai saat ini kerusakan jalan masih merupakan salah satu masalah utama untuk transportasi darat. Untuk menangani kerusakan jalan tersebut masih dengan metode manual, dimana semua proses dari awal sampai akhir perbaikan jalan tersebut masih dengan cara yang belum efektif. Untuk memetakan kerusakan jalan tersebut surveyor harus langsung turun ke lokasi kerusakan yang memberikan resiko kecelakaan serta kurang efektif. Setelah pemetaan, surveyor harus memberikan data hasil pemetaan kepada perencana dengan berkas laporan yang merupakan cara yang masih belum efektif. Sampai saat ini belum ada pengembangan untuk membuat proses pemetaan pada perbaikan jalan tersebut lebih efektif.

Untuk mengotomatisasi proses pemetaan kerusakan jalan tersebut, maka dibangun Aplikasi Pemetaan Kerusakan Jalan, yang berbasis aplikasi desktop. Aplikasi yang dibangun dengan fitur-fitur yang dapat digunakan oleh pengguna untuk membuat proses pemetaan pada perbaikan jalan dari awal sampai akhir menjadi lebih efektif.

Aplikasi Pemetaan Kerusakan Jalan yang akan dibangun memiliki beberapa modul yang dapat membantu proses pemetaan. Namun, pada tugas akhir ini hanya akan dibahas dua modul, dimana dua modul tersebut adalah proses persiapan untuk melakukan pemetaan kerusakan jalan secara otomatis. Modul yang akan dibahas adalah tentang Image Manipulation, yaitu dengan metode Image Stretching dan metode Image Stitching. Keluaran dari modul yang dibangun dapat digunakan sebagai data masukan untuk modul atau proses selanjutnya pada aplikasi.

Kata kunci: aplikasi, pemetaan, image manipulation, image stitching, image stretching, kerusakan jalan

IMAGE DATA PROCESSING MODULE DEVELOPMENT FOR ROAD DAMAGE MAPPING APPLICATION

Name : Ivan Agung Pandapotan
NRP : 05111540000166
Major : Informatics Department FTIK-ITS
Supervisor I : Daniel O. Siahaan., S.Kom., M.Sc.,
PD.Eng
Supervisor II : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.

ABSTRACT

Road is a land transportation infrastructure that is a daily necessity for every human being. Until now road damage is still one of the main problems for land transportation. To deal with the damage to the road is still a manual method, where all processes from the beginning to the end of the road repairs are still in a way that has not been effective. To measure the road damage, the surveyor must go down to the location of the damage which gives an accident risk and also is less effective. After measurement, the surveyor must provide the measurement data to the planner with the report file which is a method that is still not effective. Until now there has been no development to make the process of mapping the road repairs more effective.

To automate the process of mapping road damage, a Road Damage Mapping Application was built, based on desktop application. The application is built with features that can be used by users to make the measurement process on road repairs from start to finish more effective.

The Road Damage Mapping application that will be built has several modules that can help the measurement process. However, in this final project only two modules will be discussed, where the two modules are the preparation process to automatically measure road damage. The module that will be discussed is about Image Manipulation, namely the Image Stretching method and the Image Stitching method. The output of the module built can be used as data entered for the module or subsequent process in the application.

Keyword: desktop application, image manipulation , road damage, stitching image, stretching image

KATA PENGANTAR

Puji syukur saya sampaikan kepada Tuhan yang Maha Esa karena berkat rahmat-Nya saya dapat melaksanakan Tugas Akhir yang berjudul:

Pengembangan Modul Pengolah Data Citra Untuk Aplikasi Pemetaan Kerusakan Jalan

Terselesaikannya Tugas Akhir ini tidak terlepas dari bantuan dan dukungan banyak pihak, oleh karena itu melalui lembar ini penulis ingin mengucapkan terima kasih dan penghormatan kepada:

1. Tuhan Yesus Kristus, karena limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir dan juga perkuliahan di Informatika ITS.
2. Kedua orangtua penulis, dan anggota keluarga lainnya yang telah memberikan dukungan doa, moral, dan material kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Bapak Daniel O. Siahaan., S.Kom., M.Sc., PD.Eng selaku dosen pembimbing I yang senantiasa memberikan masukan, arahan, dan bantuan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
4. Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. selaku pembimbing II yang telah membimbing dan memberikan nasihat dan bimbingan dalam menyelesaikan Tugas Akhir ini.
5. Admin-admin Laboratorium Komputasi Cerdas & Visi (KCV) yang memberikan kesempatan penulis untuk fokus mengerjakan Tugas Akhir ini dan menyediakan tempat di laboratorium tersebut, serta teman teman dari rumpun mata kuliah Rekayasa Perangkat Lunak (RPL).
6. Hendry Wiranto, Reinhart Caesar, dan Joshua Pardosi sebagai teman kontrakan penulis yang selalu memberi semangat dan dukungan moral kepada penulis.

7. Ronald Sumbayak, Rezky Alamsyah, dan Nuzul Ristyantika yang telah menemani dan membantu penulis selama pengerjaan Tugas Akhir ini.
8. Erio, Ditta, Emanuella, Julia, Gio yang telah membantu dan menyemangati penulis selama proses pengerjaan Tugas Akhir ini.
9. Seluruh mahasiswa Informatika ITS angkatan 2015 yang telah menjadi teman penulis selama menjalani masa kuliah di Informatika ITS.
10. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini. Serta pihak lain yang namanya tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini masih memiliki kekurangan. Oleh karena itu, dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan penulis kedepannya. Selain itu, penulis berharap laporan Tugas Akhir ini dapat berguna bagi pembaca secara umum.

Surabaya, Juni 2019

DAFTAR ISI

LEMBAR PENGESAHANKesalahan! Bookmark tidak ditentukan.

ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER.....	xxiii
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	5
1.6 Metodologi Pembuatan Tugas Akhir.....	5
1.7 Sistematika Penulisan	7
2 BAB II TINJAUAN PUSTAKA.....	9
2.1 Pengolahan Citra	9
2.2 Jalan.....	10
2.3 Citra Perspektif	14
2.4 HyperText Markup Language (HTML).....	14
2.5 JavaScript.....	14
2.6 Cascading Style Sheet (CSS).....	14
2.7 Electron.....	15
2.8 Node.js.....	15
2.9 Python.....	15
2.10 Python Imaging Library.....	15
2.11 Numpy	16
3 BAB III ANALISIS DAN PERANCANGAN SISTEM. 17	
3.1 Analisis Metode Secara Umum.....	17
3.1.1 Analisis Permasalahan	17

3.1.2	Deskripsi Umum Sistem.....	18
3.1.3	Spesifikasi Kebutuhan Perangkat Lunak.....	20
3.2	Perancangan	33
3.2.1	Perancangan Diagram Kelas.....	33
3.2.2	Perancangan Antarmuka Pengguna	33
3.2.3	Perancangan Diagram Keadaan	35
4	BAB IV IMPLEMENTASI.....	37
4.1	Lingkungan Implementasi Perangkat Lunak	37
4.2	Implementasi Kasus Penggunaan.....	38
4.2.1	Implementasi Kasus Penggunaan Memasukkan Data Jalan 38	
4.2.2	Implementasi Kasus Penggunaan Melihat Status Jalan....	42
4.2.3	Implementasi Kasus Penggunaan Memperbarui Data Jalan 43	
4.3	Implementasi Modul.....	43
4.3.1	Modul <i>Image Stretching</i>	44
4.3.2	Modul <i>Image Stitching</i>	49
4.4	Implementasi Tampilan Antarmuka	54
4.4.1	Tampilan Antarmuka Masukkan Data Jalan.....	54
4.4.2	Tampilan Antarmuka Lihat Status Jalan	55
4.4.3	Tampilan Antarmuka Perbarui Data Jalan.....	56
5	BAB V PENGUJIAN DAN EVALUASI.....	59
5.1	Lingkungan Pengujian.....	59
5.2	Data Citra Jalan	59
5.3	Pengujian Fungsionalitas	64
5.3.1	Pengujian Memasukkan Data Jalan	65
5.3.2	Pengujian Melihat Status Jalan	66
5.3.3	Pengujian Memperbarui Data Jalan	66
5.4	Pengujian Modul	67
5.4.1	Pengujian Modul Sebelum dan Sesudah <i>Image Stretching</i> dengan 2 Data Citra masukkan.....	68
5.4.2	Pengujian Modul Sebelum dan Sesudah <i>Image Stretching</i> dengan 3 Data Citra masukkan.....	69
5.4.3	Pengujian Modul Sebelum dan Sesudah <i>Image Stretching</i> dengan 4 Data Citra masukkan.....	70

5.4.4	Pengujian Modul dengan Mengubah Ukuran Data Citra ..	70
5.5	Evaluasi Pengujian Fungsionalitas	72
6	BAB VI KESIMPULAN DAN SARAN	73
6.1	Kesimpulan	73
6.2	Saran	74
	DAFTAR PUSTAKA	75
A.	LAMPIRAN A.....	77
1.	Use Case Name	77
1.1	Brief Description	77
2.	Flow of Events	77
2.1	Basic Flow	77
2.2	Alternative Flows	77
2.3	Exceptions.....	77
3.	Spesial Requirements	77
4.	Pre-Conditions	77
5.	Post-Conditions	78
6.	Extension Points.....	78
B.	LAMPIRAN B.....	79
1.	Use Case Name	79
1.1	Brief Description	79
2.	Flow of Events	79
2.1	Basic Flow	79
2.2	Alternative Flows	79
2.3	Exceptions.....	80
3.	Spesial Requirements	80
4.	Pre- Conditions	80
5.	Post- Conditions	80
6.	Extension Points.....	80
C.	LAMPIRAN C.....	81
1.	Introduction.....	81
1.1	Purpose	81
1.2	Scope	81
1.3	Definitions, Acronyms and abbreviations.....	81
1.4	References.....	81
1.5	Overview	81

2.	Introduction.....	82
3.	Specific Requirements.....	84
D.	LAMPIRAN D.....	85
E.	LAMPIRAN E.....	93
	BIODATA PENULIS.....	95

DAFTAR GAMBAR

Gambar 2.1 Retak refleksi sambungan	12
Gambar 2.2 Lubang	12
Gambar 2.3 Retak kulit buaya	13
Gambar 2.4 Retak selip	13
Gambar 3.1 Deskripsi Umum Sistem	19
Gambar 3.2 Diagram Kasus Penggunaan.....	22
Gambar 3.3 Diagram Alur Modul.....	23
Gambar 3.4 Diagram Alur <i>Image Stitching</i>	24
Gambar 3.5 Analisis Kelas UC-01	25
Gambar 3.6 Diagram Sekuens UC-01.....	26
Gambar 3.7 Diagram Aktivitas UC-01	26
Gambar 3.8 Analisis Kelas UC-02 (Surveyor).....	27
Gambar 3.9 Analisis Kelas UC-02 (Perencana)	28
Gambar 3.10 Diagram Sekuens UC-02 (Surveyor)	28
Gambar 3.11 Diagram Sekuens UC-02 (Perencana)	28
Gambar 3.12 Diagram Aktivitas UC-02 (Surveyor).....	29
Gambar 3.13 Diagram Aktivitas UC-02 (Perencana)	29
Gambar 3.14 Analisis Kelas UC-02a.....	30
Gambar 3.15 Diagram Sekuens UC-02a.....	30
Gambar 3.16 Diagram Aktivitas UC-02a.....	31
Gambar 3.17 Diagram keadaan (<i>State Diagram</i>)	36
Gambar 4.1 Tampilan Antarmuka Masukkan Data Jalan	54
Gambar 4.2 Tampilan Antarmuka Lihat Status Jalan (Surveyor).....	55
Gambar 4.3 Tampilan Antarmuka Lihat Status Jalan (Perencana)	56
Gambar 4.4 Tampilan Antarmuka Perbaruis Data Jalan	57
Gambar 5.1 Citra dengan pencahayaan yang tidak terkontrol.....	60
Gambar 5.2 Citra dengan gangguan objek	60
Gambar 5.3 Citra 1 tidak sesuai dengan jarak pada GPS	61
Gambar 5.4 Citra 2 tidak sesuai dengan jarak pada GPS	61
Gambar 5.5 Ketinggian 6 meter	62
Gambar 5.6 Ketinggian 7 meter	62
Gambar 5.7 Ketinggian 10 meter.....	63

Gambar 5.8 Citra 1 ideal	63
Gambar 5.9 Citra 2 ideal	64
Gambar 5.10 <i>Image Stitching</i> Tanpa <i>Image Stretching</i>	68
Gambar 5.11 <i>Image Stitching</i> Dengan <i>Image Stretching</i>	68
Gambar 5.12 <i>Image Stitching</i> Tanpa <i>Image Stretching</i>	69
Gambar 5.13 <i>Image Stitching</i> Dengan <i>Image Stretching</i>	69
Gambar D.1 Hasil Pengambilan Data Citra 1.....	85
Gambar D.2 Hasil Pengambilan Data Citra 2.....	85
Gambar D.3 Hasil Pengambilan Data Citra 3.....	86
Gambar D.4 Hasil Pengambilan Data Citra 4.....	86
Gambar D.5 Hasil Pengambilan Data Citra 5.....	87
Gambar D.6 Hasil Pengambilan Data Citra 6.....	87
Gambar D.7 Hasil Pengambilan Data Citra 7.....	88
Gambar D.8 Hasil Pengambilan Data Citra 8.....	88
Gambar D.9 Hasil Pengambilan Data Citra 9.....	89
Gambar D.10 Hasil Pengambilan Data Citra 10.....	89
Gambar D.11 Hasil Pengambilan Data Citra 11.....	90
Gambar D.12 Hasil Pengambilan Data Citra 12.....	90
Gambar D.13 Hasil Pengambilan Data Citra 13.....	91
Gambar D.14 Hasil Pengambilan Data Citra 14.....	91
Gambar D.15 Hasil Pengambilan Data Citra 15.....	92
Gambar D.16 Hasil Pengambilan Data Citra 16.....	92
Gambar E.1 Diagram Kelas.....	93

DAFTAR TABEL

Tabel 3.1 Kasus Penggunaan.....	21
Tabel 3.2 Kebutuhan Fungsional.....	32
Tabel 3.3 Kualitas Perangkat Lunak.....	32
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak	37
Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak	59
Tabel 5.2 Pengujian Memasukkan Data Jalan.....	65
Tabel 5.3 Pengujian Melihat Status Jalan	66
Tabel 5.4 Pengujian Memperbarui Data Jalan.....	67
Tabel 5.5 Ukuran Data Normal	70
Tabel 5.6 Ukuran Data 75% dari Data Normal	70
Tabel 5.7 Ukuran Data 50% dari Data Normal	71
Tabel 5.8 Evaluasi Pengujian Fungsionalitas.....	72

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 1 Implementasi Menampilkan Pratinjau Data	39
Kode Sumber 2 Implementasi fungsi sortFiles.....	40
Kode Sumber 3 Implementasi memasukkan data kedalam sistem	42
Kode Sumber 4 Implementasi Melihat Status Jalan	42
Kode Sumber 5 Implementasi mengambil data dari basis data ..	43
Kode Sumber 6 Membuat Bingkai Citra Keluaran.....	44
Kode Sumber 7 Membuat Citra Baru pada Bingkai	47
Kode Sumber 8 Proses Manipulasi Citra	49
Kode Sumber 9 Memindahkan Citra ke Bingkai dengan Ukuran Sama Seperti Citra Masukkan	49
Kode Sumber 10 Memulai proses <i>Image Stitching</i>	50
Kode Sumber 11 Fungsi <code>image_stitch</code>	51
Kode Sumber 12 Fungsi <code>Detect_Feature_And_Keypoints</code>	52
Kode Sumber 13 Fungsi <code>matchKeypoints</code>	53
Kode Sumber 14 Fungsi <code>get_Allpossible_Match</code>	53
Kode Sumber 15 Fungsi <code>All_validmatches</code>	53
Kode Sumber 16 Fungsi <code>Compute_Homography</code>	53
Kode Sumber 17 Fungsi <code>getwarp_perspective</code>	54

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1 Latar Belakang

Pemetaan dan perhitungan sudah menjadi bagian dari kebutuhan sehari-hari bagi banyak orang. Pemetaan ini memiliki tujuan untuk mempermudah atau memperjelas suatu kegiatan. Pemetaan dapat dilakukan dengan berbagai cara sesuai keadaan dan kebutuhan. *Input* (masukkan) dan *output* (keluaran) dari pemetaan berupa data.

Salah satu penggunaan pemetaan dapat dilakukan untuk memetakan tingkat kerusakan jalan, hasil pemetaan tersebut yang digunakan sebagai dasar perhitungan dalam pengklasifikasian kerusakan jalan yang ada. Berdasarkan data BPS (Badan Pusat Statistik) yang ada per tahun 2016 panjang jalan total di Indonesia adalah 537838 Km, dimana 326629 Km adalah jalan aspal dan 211209 Km adalah jalan non-aspal [1]. Untuk di daerah Jawa Timur sendiri memiliki total 1420.5 Km dengan keterangan 699.27 Km dikategorikan baik, 563.57 Km dikategorikan sedang, 141.33 Km dikategorikan rusak ringan, dan 16.33 Km dikategorikan rusak berat [2]. Untuk menangani kerusakan yang ada pemerintah mengalokasikan sekitar Rp 23.7 triliun untuk pemeliharaan jalan dan jembatan [3].

Penanganan kerusakan dan pemeliharaan jalan tidak dapat dilakukan oleh sembarangan pihak, karena membutuhkan kontrol yang jelas dari pemerintahan. Kebijakan mengenai jalan di Indonesia diurus oleh Bina Marga, yang juga diatur oleh peraturan dari pemerintahan. Sesuai dengan Peraturan Menteri Pekerjaan Umum dan Perumahan Rakyat Nomor 15/PRT/M/2015

Tentang Organisasi dan Tata Kerja Kementerian Pekerjaan Umum dan Perumahan Rakyat Direktorat Jenderal Bina Marga mempunyai tugas menyelenggarakan perumusan dan pelaksanaan kebijakan di bidang penyelenggaraan jalan sesuai dengan ketentuan peraturan perundang-undangan. Dalam peraturan ini salah satu fungsi Bina Marga adalah untuk merumuskan dan melaksanakan kebijakan di bidang penyelenggaraan jalan.

Salah satu pengaplikasian penyelenggaraan jalan adalah dengan perbaikan jalan. Ketika jalan sudah tidak sesuai dengan standar maka jalan dapat dikondisikan tidak baik, yang kemudian dapat diklasifikasikan keadaan jalan tersebut untuk memperoleh tindakan yang spesifik. Selama ini, proses pemetaan kondisi jalan dilakukan secara manual, yaitu dengan pergi ke jalan untuk pengecekan, lalu jika menemukan bagian yang rusak, maka harus menunggu jalan kosong untuk sementara atau menutup jalan tersebut untuk melakukan pemetaan. Selain mengecek kondisi jalan dengan pemetaan manual di lokasi kerusakan, dapat juga dilakukan pemetaan melalui gambar yang diperoleh dari foto jalan.

Ada beberapa kelemahan dalam pemetaan kerusakan jalan secara manual. Pertama dari segi ruang dan waktu, pemetaan manual akan memakan waktu yang lebih panjang karena ketika dilakukan di saat terang maka akan banyak kendaraan yang melintas sehingga pemeta harus menunggu jalan sepi lalu menutup sekitaran bagian yang akan dipetakan, sedangkan jika dilakukan di saat gelap, jumlah kendaraan yang melintas memang akan berkurang namun dalam keadaan gelap maka membutuhkan alat bantu yang memadai (penerangan dan penutup jalan yang dapat dilihat jelas dari jauh) untuk melakukan pemetaan. Kedua dari segi keselamatan, dalam pemetaan manual pemeta akan memiliki resiko pekerjaan yang mungkin membahayakan pada saat pemetaan, karena harus benar benar memetakan ke bagian jalan yang rusak, meskipun bagian tersebut berada di tengah jalan. Ketiga dari segi biaya, karena membutuhkan ruang dan waktu yang relatif lebih lama, serta membutuhkan sumber daya manusia yang lebih banyak, maka biaya yang dikeluarkan pun akan semakin

besar, dimana akan sangat merugikan untuk proses yang berlangsung secara berkala. Keempat dari segi akurasi, dikarenakan pemetaan manual membutuhkan situasi yang kondusif, maka sangat memungkinkan terjadi situasi yang tidak sesuai perkiraan, yangmana akan mengakibatkan pemetaan tidak sesuai kenyataan, yang kemudian akan berdampak saat proses perbaikan, yang juga akan mengakibatkan peningkatan kebutuhan biaya.

Untuk mengatasi permasalahan tersebut, pengembangan sebuah sistem berbasis teknologi informasi untuk pemetaan kerusakan jalan perlu dibangun. Sejauh ini, sudah ada aplikasi yang dapat melaporkan jalan rusak, namun aplikasi tersebut hanya sejauh memberi laporan dan menerima laporan, tidak dapat membantu memetakan keadaan jalan yang rusak tersebut. Alasan tersebut yang membuat aplikasi pemetaan kerusakan jalan dengan citra diperlukan. Konsep aplikasi ini adalah melakukan pengambilan citra di sepanjang jalan yang akan diperiksa, lalu mengunggah ke server. Pada server data citra jalan tersebut akan diproses untuk menampilkan hasil akhir yang berupa kondisi jalan yang rusak, hasil akhir ini akan dikembalikan kepada orang yang membutuhkan. Sesuai deskripsi yang ada maka akan dibutuhkan beberapa komponen, yaitu drone dengan kamera untuk mengambil gambar dan server sebagai tempat terjadinya proses.

Tugas akhir ini merupakan bagian dari penelitian untuk pengembangan sistem pemetaan kerusakan jalan. Tugas akhir ini berfokus pada pengembangan salah satu modul dalam sistem tersebut. Modul yang dikembangkan adalah modul pengolahan data citra jalan masukan. Modul ini melakukan dua tahapan dari beberapa tahapan dalam sistem. Pertama modul ini mengolah data yang diterima dari posisi yang kurang lebih tegak lurus (90 derajat) yangmana citra tersebut memiliki perspektif karena diambil melalui sebuah media, data diolah menjadi semirip mungkin dengan kenyataan yang ada. Kedua, modul ini menyatukan banyak data yang sudah melalui proses pertama menjadi satu gambar utuh yang merupakan perpaduan dari data yang sudah diolah tersebut.

1.2 Rumusan Masalah

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

1. Bagaimana melakukan penyatuan citra jalan menjadi satu citra yang berkelanjutan dengan metode *Image Stitching* ?
2. Bagaimana arsitektur aplikasi untuk penerapan produk perhitungan kerusakan jalan?
3. Bagaimana cara mengolah data citra jalan yang diperoleh dengan metode *Image Stretching*?

1.3 Batasan Masalah

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

1. Data percobaan menyesuaikan dengan peraturan mengenai jalan yang berlaku.
2. Kondisi jalan dalam keadaan kering (tidak tergenang)
3. Kondisi jalan tidak tertutupi sesuatu dalam waktu lama (mobil parkir, pohon, sampah, dan sebagainya)
4. Jenis jalan yang ditangani adalah jenis jalan aspal.
5. Data yang dapat diproses adalah data yang sejenis, artinya diambil dengan alat yang sama, rentang waktu tertentu, maka akan diberikan standar pengambilan data gambar nantinya
6. Proses yang dibahas dalam modul adalah mengenai *stretching* data citra dan *stitching* data citra

1.4 Tujuan

Tujuan dari tugas akhir ini adalah untuk membangun sebuah modul penyatuan banyak data gambar menjadi satu gambar jalan yang utuh dari sistem pemetaan kerusakan jalan. Modul ini melakukan proses *stretching* (merubah bentuk gambar) dan penyatuan gambar.

1.5 Manfaat

Tugas akhir ini diharapkan dapat membantu membuat pengecekan kerusakan jalan menjadi lebih efektif dengan perhitungan melalui gambar, serta membuat sistem yang dapat melakukan perhitungan melalui gambar agar para pelaku pengecekan jalan dapat melakukan pengecekan jalan melalui sistem yang akan dibuat ini.

1.6 Metodologi Pembuatan Tugas Akhir

Adapun beberapa tahap dalam proses pengerjaan Tugas Akhir ini, yaitu:

1. Penyusunan proposal Tugas Akhir

Tahap awal yang dilakukan dalam proses pengerjaan tugas akhir ini adalah penyusunan proposal tugas akhir. Proposal tugas akhir ini berisi pendahuluan, tinjauan pustakan dan metodologi dari tugas akhir yang akan dibuat. Pendahuluan terdiri atas latar belakang usulan tugas akhir, rumusan masalah yang diangkat dan batasan masalah yang ditentukan. Tinjauan pustaka yang menjadi referensi pendukung pembuatan tugas akhir. Metodologi berisi penjelasan tahapan penyusunan tugas akhir. Selain itu, terdapat jadwal kegiatan pengerjaan tugas akhir. Pada tugas akhir ini dilakukan pengolahan data citra yang diharapkan dapat mempermudah pemetaan kerusakan jalan.

2. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang terkait untuk menyelesaikan studi kasus, diantara lain adalah buku, *scientific paper*, artikel di internet, dan materi kuliah yang terkait dengan *image processing*, *image affine*, *image stitching*, serta dokumen atau ketentuan mengenai jalan yang akan diuji.

3. Analisis dan desain perangkat lunak

Tahap ini dilakukan dengan analisis dan desain model yang sesuai dengan tujuan yang telah dijabarkan sebelumnya. Selain itu, pada tahap ini akan dilakukan eksplorasi terkait arsitektur perangkat lunak yang akan digunakan. Acuan eksplorasi arsitektur

dilakukan berdasarkan kebutuhan dan keefisienan perangkat lunak, untuk memperoleh hasil yang dituju, berupa gambar dan dimensi kerusakan jalan, dengan efektif.

4. Implementasi perangkat lunak

Tahap implementasi meliputi implementasi algoritma dan *library* pada perangkat lunak yang telah didukung oleh hasil analisis dan desain pada tahap sebelumnya. Implementasi algoritma ini dilakukan dengan menggunakan bahasa pemrograman *Python* serta beberapa *library* yang disediakan dalam *Python*, pada perangkat lunak menggunakan *Javascript*.

5. Uji coba dan evaluasi

Tahap pengujian dan evaluasi dilakukan menggunakan dataset yang diambil manual dengan skala yang mengikuti ketentuan yang sesuai dengan sumber yang sudah ditentukan sebelumnya. Untuk mengetahui hasil dan performa arsitektur yang telah dibangun. Selain arsitektur yang telah disebutkan di atas, juga akan dilakukan uji coba untuk mengubah parameter-parameter yang akan dipakai pada perangkat lunak. Beberapa parameter yang dapat diubah antara lain sudut pengambilan citra, dan jarak pengambilan citra. Data citra akan diambil menggunakan alat bantu, seperti drone, untuk mempermudah menyesuaikan dengan parameter yang dituju.

6. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini. Pada tahap ini juga disertakan hasil dari implementasi perangkat lunak yang telah dibuat. Sistematika penulisan buku Tugas Akhir ini secara garis besar antara lain:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Masalah
 - d. Tujuan

- e. Manfaat
 - f. Metodologi Pembuatan Tugas Akhir
 - g. Sistematika Penulisan
2. Tinjauan Pustaka
 3. Analisis dan Perancangan Sistem
 4. Implementasi
 5. Pengujian dan Evaluasi
 6. Kesimpulan dan Saran
 7. Daftar Pustaka

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Tinjauan Pustaka

Memaparkan dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Membahas tentang analisis permasalahan, deskripsi umum sistem, spesifikasi kebutuhan perangkat lunak, lingkungan perancangan, perancangan sistem, dan struktur data.

Bab IV Implementasi

Bab ini berisi implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode sumber yang digunakan untuk proses implementasi.

Bab V Pengujian dan Evaluasi

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

Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan pada Tugas Akhir ini. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi dokumen atau data yang mendukung Tugas Akhir.

BAB II

TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap perangkat lunak yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Pengolahan Citra

Ada beberapa teknik pengolahan citra yang terkait dengan tugas akhir ini, yaitu *Image manipulating* dan *image stitching*.

Image manipulating, atau manipulasi data citra, merupakan teknik yang dapat digunakan untuk meningkatkan atau mengurangi pesan yang ada pada data citra. Manipulasi data citra dapat dilakukan dengan mengubah proyeksi, memperluas, mengompresi, mengurangi, menghapus, memodifikasi, menambah, memindahkan, memutar, mengisolasi, atau menggabungkan berbagai bagian gambar. Tindakan manipulasi yang dilakukan akan berbeda beda mengikuti tujuan dari citra yang diinginkan.

Image Stitching adalah teknik penyelesaran dari beberapa gambar foto yang tumpang tindih (*overlapping*) menjadi sebuah komposisi besar yang merupakan bagian dari 3D *scene*. Metode menggabungkan beberapa gambar foto untuk menghasilkan satu gambar (*scene*) tersebut disebut panorama. Sebagai contoh, beberapa gambar langit diambil untuk menghasilkan panorama tunggal, maka menjadi panorama Astrophotography. Ada berbagai langkah *stitching* gambar seperti segmentasi (*segmentation*) gambar, ekstraksi (*extraction*) fitur dan pencocokan (*matching*) [4].

Image Stitching adalah proses dasar menggabungkan berbagai gambar fotografi dengan bidang yang tampilannya tumpang tindih untuk menghasilkan panorama tersegmentasi untuk mendapatkan gambar dengan resolusi tinggi. Proses *Image Stitching* pada dasarnya dapat dibagi menjadi tiga langkah utama atau komponen,

serta untuk pengembangan lebih lanjut akan membutuhkan satu langkah lagi, yaitu:

1. *Calibration*

Kalibrasi gambar berguna untuk meminimalisir perbedaan antara model lensa ideal dan kombinasi lensa kamera yang digunakan. Perbedaan ini disebabkan oleh cacat optik seperti perbedaan pencahayaan antara distorsi dan gambar. Parameter kamera ekstrinsik dan Intrinsik harus disesuaikan untuk tujuan merekonstruksi struktur 3D gambar foto dari titik koordinat piksel gambarnya.

2. *Registration*

Registrasi gambar adalah bagian penting dari prosedur *stitching*. Tujuan dasarnya adalah untuk membuat korespondensi geometris antar gambar. Karena itu, kita dapat membandingkan gambar yang diambil dan menerapkan langkah-langkah lain dengan tepat.

3. *Blending*

Proses pencampuran diterapkan di seluruh persimpangan sehingga *stitching* akan menghasilkan gambar yang mulus.

4. *Rotation matrix estimation*

Dengan memperhatikan hasil darifitur, kita dapat memperkirakan gerakan kamera pada bagian *input*. Gerakan kamera dapat dimodelkan sebagai rotasi 3D di antara gambar. Biasanya, pengguna perlu bergerak sedikit saat mengambil gambar gambar untuk menghasilkan foto panorama, sehingga tidak kehilangan keakuratan dan gerakan kamera dapat dimodelkan dengan rotasi 3D.

2.2 Jalan

Jalan adalah prasarana transportasi darat yang meliputi seluruh bagian jalan, termasuk bangunan pelengkap dan perlengkapannya yang diperuntukkan bagi lalu lintas, yang berada pada permukaan tanah, di atas permukaan tanah, di atas permukaan air serta di bawah permukaan tanah dan atau air, kecuali jalan kereta api, jalan lori dan jalan kabel (Peraturan Pemerintah Nomor 34 Tahun 2006).

Sistem jaringan jalan dijabarkan dalam UU Republik Indonesia No.38 tahun 2004 pasal 7, adalah sebagai berikut.

1. Sistem jaringan jalan primer, merupakan sistem jaringan jalan dengan peranan pelayanan distribusi barang dan jasa untuk pengembangan semua wilayah di tingkat nasional, dengan menghubungkan semua simpul jasa distribusi yang berwujud pusat-pusat kegiatan.

2. Sistem jaringan jalan sekunder, merupakan sistem jaringan jalan dengan peranan pelayanan distribusi barang dan jasa untuk masyarakat di dalam kawasan perkotaan.

Yang lalu dilanjutkan mengenai pengelompokkan jalan umum menurut fungsinya pada UU Republik Indonesia No.38 tahun 2004 pasal 8, yaitu sebagai berikut:

1. Jalan arteri, merupakan jalan umum yang berfungsi melayani angkutan utama dengan ciri perjalanan jarak jauh, kecepatan rata-rata tinggi, dan jumlah jalan masuk dibatasi secara berdaya guna.

2. Jalan kolektor, merupakan jalan umum yang berfungsi melayani angkutan pengumpul atau pembagi dengan ciri perjalanan jarak sedang, kecepatan rata-rata sedang, dan jumlah jalan masuk dibatasi.

3. Jalan local, merupakan jalan umum yang berfungsi melayani angkutan setempat dengan ciri perjalanan jarak dekat, kecepatan rata-rata rendah, dan jumlah jalan masuk tidak dibatasi.

4. Jalan lingkungan, merupakan jalan umum yang berfungsi melayani angkutan lingkungan dengan ciri perjalanan jarak dekat, dan kecepatan rata-rata rendah.

Khusus untuk keperluan dalam modul yang akan dikembangkan, jenis-jenis kerusakan pada jalan yang akan diperhitungkan adalah sebagai berikut [5].

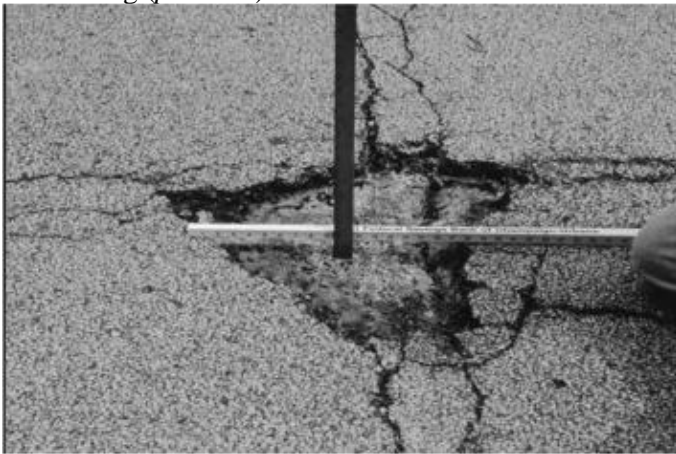
1. Retak refleksi sambungan (*joint reflection cracking*)

Kerusakan ini umumnya terjadi pada permukaan perkerasan aspal yang telah dihamparkan di atas perkerasan beton. Contoh retak refleksi sambungan dapat dilihat pada Gambar 2.1



Gambar 2.1 Retak refleksi sambungan

2. Lubang (*potholes*)



Gambar 2.2 Lubang

Lubang adalah lekukan permukaan perkerasan akibat hilangnya lapisan aus dan material lapis pondasi. Kerusakan berbentuk lubang kecil biasanya berdiameter kurang dari 0,9 m dan berbentuk mangkuk yang dapat berhubungan atau tidak berhubungan dengan permukaan lainnya. Lubang biasanya terjadi

akibat galian utilitas atau tambalan di area perkerasan yang telah ada. Contoh lubang dapat dilihat pada Gambar 2.2

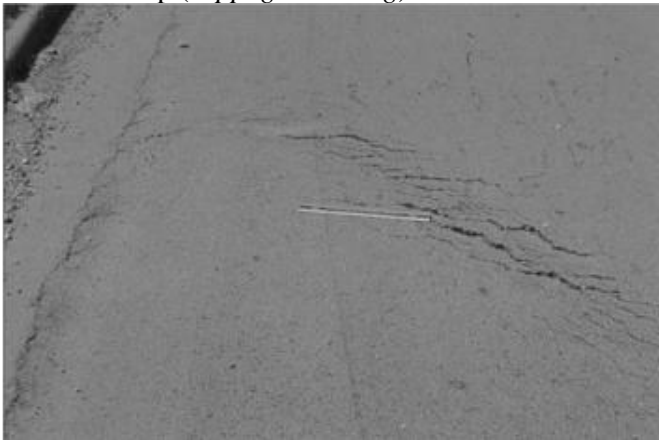
3. Retak kulit buaya (*alligator cracking*)

Retak kulit buaya adalah serangkaian retak memanjang paralel yang membentuk banyak sisi menyerupai kulit buaya. Contoh retak kulit buaya dapat dilihat pada Gambar 2.3



Gambar 2.3 Retak kulit buaya

4. Retak selip (*slippage cracking*)



Gambar 2.4 Retak selip

Retak selip atau retak yang berbentuk bulan sabit yang diakibatkan oleh gaya-gaya horizontal yang berasal dari kendaraan. Contoh retak selip dapat dilihat pada Gambar 2.4

2.3 Citra Perspektif

Perspektif adalah alat yang dapat digunakan untuk menggambar ulang suatu kejadian atau adegan tiga dimensi (3D) dengan memproyeksikannya ke permukaan dua dimensi (2D). Aturan dari perspektif mengikuti model proyeksi kamera lubang jarum, dimana titik titik pada kejadian diproyeksikan ke bidang gambar 2D melalui garis lurus yang menyatu di tengah proyeksi. [6]

2.4 HyperText Markup Language (HTML)

HyperText Markup Language lebih dikenal dengan singkatannya, yaitu HTML. HTML adalah Bahasa yang menggambarkan struktur dan isi semantik dari sebuah dokumen web, karena itu HTML adalah Bahasa inti yang digunakan sebagian besar *Web* yang ada. HTML adalah standar internasional yang spesifikasinya didasari dari World Wide Web Consortium dan WHATWG. Ini dianggap sebagai "standar dasar" dan secara teknik selalu dibenahi. Versi terkini dari HTML spesifikasinya menunjuk HTML5. [7]

2.5 JavaScript

JavaScript adalah Bahasa pemrograman tingkat tinggi dan dinamis, yang dapat bekerja di sebagian besar *Web Browser*. Kode JavaScript dapat disisipkan dalam halaman *Web* dengan menggunakan penanda (*tag*) *SCRIPT*. [8]

2.6 Cascading Style Sheet (CSS)

Cascading Style Sheet, yang dikenal dengan CSS merupakan bahasa *stylesheet* yang digunakan untuk merubah tampilan dokumen tertulis dalam HTML atau XML, perlu diingat CSS bukan merupakan sebuah bahasa pemrograman. CSS menjelaskan

bagaimana elemen terstruktur yang harus ditampilkan di layer, di kertas, dalam cara komunikasi, atau dalam media yang lain. [9]

2.7 Electron

Electron adalah *library* yang dikembangkan oleh GitHub untuk membangun aplikasi desktop lintas platform dengan HTML, CSS, dan JavaScript. Elektron dapat menyelesaikannya dengan menggabungkan Chromium dan Node.js yang dijalankan bersamaan dan aplikasi dikemas untuk Mac, Windows, dan Linux.js. [10]

2.8 Node.js

Node.js adalah *JavaScript run-time Environment* yang bersifat terbuka untuk umum, dengan menjalankan mesin JavaScript V8, inti dari Google Chrome, di luar *browser*. Hal ini membuat Node.js memiliki performa yang sangat baik. Aplikasi Node.js dijalankan dalam sebuah proses tunggal (*single process*), tanpa membuat *thread* baru untuk setiap permintaan. [11]

2.9 Python

Python adalah bahasa pemrograman yang dapat digunakan dalam banyak pengembangan *Website* dan perangkat lunak, digunakan juga dalam banyak penelitian dan *Workshop*, serta banyak dimanfaatkan sebagai *script* dimana digunakan sebagai pendukung dari suatu pengembangan atau penelitian dalam Bahasa yang berbeda. Python bebas digunakan dalam kepentingan apapun, karena Python bersifat *Open Source* dan tidak bertentangan dengan lisensi Python sendiri. Python merupakan Bahasa yang memiliki Bahasa yang dirancang untuk memudahkan mengerti Bahasa tersebut. Python juga bekerja di berbagai *platform* seperti Windows, Mac, Linux, dan lain-lain. [12]

2.10 Python Imaging Library

Python Imaging Library biasa dikenal dengan PIL, yang kemudian dikembangkan dengan nama Pillow. PIL adalah *library*

yang dapat digunakan bebas untuk Bahasa Python, yang dapat digunakan untuk membuka, memanipulasi, dan menyimpan banyak tipe data citra. [13]

2.11 Numpy

Numpy adalah *library* untuk Bahasa Python yang dapat mendukung untuk pengolahan data pada *array* dan matriks multidimensi yang berukuran besar, serta sekumpulan fungsi-fungsi matematika tingkat tinggi untuk mengoperasikan *array* tersebut. Numpy bersifat *open source* sehingga banyak digunakan dalam pengolahan data dalam penelitian. [14]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab 3 ini akan dijelaskan mengenai analisis dan perancangan perangkat lunak untuk mencapai tujuan dari Tugas Akhir. Perancangan ini meliputi analisis metode secara umum dan perancangan sistem.

3.1 Analisis Metode Secara Umum

Pada subbab berikut akan dijelaskan proses pemetaan kerusakan jalan pada aplikasi pemetaan kerusakan jalan. Analisis yang dilakukan meliputi analisis permasalahan, deskripsi umum sistem, dan kebutuhan fungsional sistem.

3.1.1 Analisis Permasalahan

Permasalahan utama yang diangkat pada pembuatan Tugas Akhir ini adalah bagaimana membuat modul yang dapat memproses citra yang dapat digunakan untuk memetakan dimensi kerusakan jalan pada Aplikasi Pemetaan Kerusakan Jalan.

Sampai saat ini masih banyak jalanan yang mengalami kerusakan, hal tersebut dapat ditangani, namun tindakan perbaikan jalan membutuhkan proses yang tidak singkat memakan banyak tenaga dan waktu, dimana harus dilakukan pemetaan, dimana pemetaan yang dilakukan harus langsung memetakan ke lokasi kerusakan dimana pemeta memiliki resiko terjadi kecelakaan saat pemetaan tersebut. Lalu hasil pemetaan diberikan kepada pelaku perencanaan, lalu pihak perencanaan mengumpulkan alat dan kebutuhan perbaikan serta sumber daya manusia (SDM) yang akan turun ke lokasi kerusakan jalan dan memperbaiki kerusakan tersebut.

Tugas Akhir ini bertujuan untuk mengatasi permasalahan tersebut, membuat proses untuk memperbaiki jalan yang rusak lebih singkat dan aman. Aplikasi Pemetaan kerusakan jalan ini memiliki empat modul utama dalam pemetaannya, yaitu membentuk ulang citra agar semirip mungkin dengan keadaan

aslinya (*Stretching Image*), menyatukan citra yang diambil terpisah agar menjadi satu citra yang mencakup kedua citra yang disatukan tersebut (*Stitching Image*), menghapus objek yang ada pada citra agar tidak mengganggu perhitungan dimensi jalan, dan mengukur dimensi kerusakan jalan yang ada.

Dari keempat fitur utama yang telah direncanakan, pada Tugas Akhir ini hanya menyelesaikan dua modul, yaitu modul *Stretching Image* dan modul *Stitching Image*. Dikarenakan dua modul yang belum diselesaikan pada Tugas Akhir ini, maka tidak menutup kemungkinan bahwa akan ada modul tambahan atau pergantian modul menyesuaikan dengan kebutuhan dan tujuan.

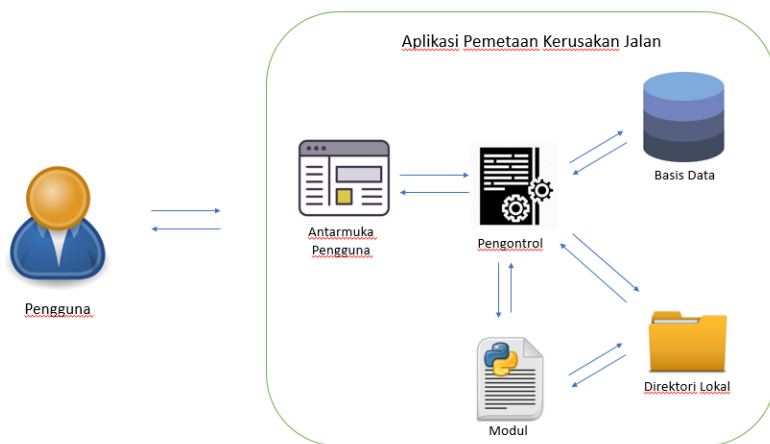
Dalam pembuatannya, aplikasi ini dikembangkan agar fitur fitur yang ada dapat sesesuai mungkin dengan kebutuhan pengguna dalam melakukan seluruh proses dari pemetaan jalan sampai jalan selesai diperbaiki, dimana pengguna yang dimaksud adalah surveyor dan perencana. Surveyor adalah orang yang melakukan pemetaan ke lokasi dan perencana adalah orang yang memanfaatkan data pemetaan untuk perbaikan jalan, masing masing pengguna aplikasi memiliki hak akses terhadap fitur tertentu yang sudah disesuaikan dengan kebutuhan.

3.1.2 Deskripsi Umum Sistem

Pada Tugas Akhir ini akan dibuat sebuah Aplikasi Pemetaan Kerusakan Jalan. Aplikasi yang akan dibuat akan dibuat sesesuai mungkin dengan kebutuhan kebutuhan dari para pengguna, maka akan dibuat fitur fitur yang dapat menyelesaikan ataupun mempermudah para pengguna tersebut. Fitur-fitur akan disediakan didalam sebuah aplikasi desktop, dimana pengguna akan diiberi hak akses yang berbeda sesuai dengan peranan pengguna tersebut didalam menjalankan tugasnya.

Aplikasi yang akan dibangun adalah aplikasi desktop yang diupayakan dapat menyelesaikan permasalahan seperti yang sudah dibahas pada subbab 3.1.1. Dikarenakan kurang efektifnya proses perbaikan jalan dari awal sampai akhir, maka aplikasi ini juga akan dapat membuat proses perbaikan lebih efektif. Namun, pada Tugas Akhir ini, pembuatan Aplikasi Pemetaan Kerusakan Jalan belum

dapat memperoleh tujuan akhir dari proses pemetaan kerusakan jalan itu sendiri, dikarenakan akan ada modul yang tidak diselesaikan pada Tugas Akhir ini. Pada bagian pemetaan kerusakan jalan akan ada 4 modul, dimana pada Tugas Akhir ini hanya akan diselesaikan 2 modul saja, yaitu modul *Image Stretching* dan *Image Stitching*, namun aplikasi sudah dapat dimanfaatkan oleh para pengguna dan sudah siap untuk menerima 2 modul yang belum diselesaikan tersebut.



Gambar 3.1 Deskripsi Umum Sistem

Pengguna akan berinteraksi dengan aplikasi dan menjalankan modul modul yang ada hanya dengan melakukan proses yang relatif sederhana. Gambar 3.1 menjelaskan bahwa pengguna berinteraksi dengan aplikasi, tanpa tahu detail proses yang terjadi didalam aplikasi. Informasi yang diberikan dan diterima pengguna merupakan hasil pengolahan data yang ada di dalam aplikasi, dimana aplikasi memiliki modul untuk membantu proses data citra, basis data untuk membantu proses penyimpanan data teks, direktori local sebagai tempat penyimpanan citra, yangmana semua proses dikontrol oleh pengontrol (*Controller*). Jadi pengguna aplikasi tidak perlu berinteraksi secara langsung dengan modul,

basis data, direktori local, dan pengontrol. Pengguna hanya cukup memberi perintah kepada aplikasi melalui tampilan antarmuka pengguna, lalu aplikasi akan menjalankan perintah-perintah yang memenuhi kebutuhan pengguna dengan pengontrol sebagai sebagai pusat proses untuk memproses basis data, direktori lokal, dan modul. Alur dibuat semudah mungkin untuk pengguna sehingga pekerjaan pengguna tersebut dapat terselesaikan dan akan menjadi lebih efektif tanpa perlu mempelajari hal yang kompleks.

3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Subbab ini membahas tentang analisis dari pengguna, kebutuhan-kebutuhan yang dapat diterapkan dalam perangkat lunak ini. Hasil dari analisis yang telah dilakukan tersebut adalah spesifikasi kebutuhan perangkat lunak. Bagian ini berisi kebutuhan perangkat lunak yang diperoleh dari analisa direpresentasikan dalam bentuk kebutuhan fungsional, kebutuhan non fungsional, diagram kasus penggunaan, dan diagram aktivitas.

3.1.3.1 Aktor

Pengertian aktor atau pengguna adalah pihak-pihak, baik manusia maupun sistem atau perangkat lain yang terlibat dan berinteraksi secara langsung dengan sistem. Pada perangkat lunak ini hanya terdapat dua pengguna, yaitu surveyor dan perencana. Disediakan dua peran karena pada perangkat lunak ini hanya berfokus kepada dua peran tersebut, sesuai dengan kebutuhan fungsional yang dibuat dengan tujuan membuat pekerjaan dua peran tersebut lebih efisien.

Pada Tugas Akhir ini, kedua peran aktor adalah surveyor dan perencana, dimana surveyor adalah seorang pekerja lapangan yang bertugas turun langsung ke lapangan untuk melakukan pengecekan terhadap kondisi lapangan, serta memasukkan hasil pengecekan yang dilakukannya kedalam sistem. Sedangkan perencana adalah orang yang bekerja sebagai orang yang mengurus data yang sudah dimasukkan oleh surveyor untuk diberikan tindakan.

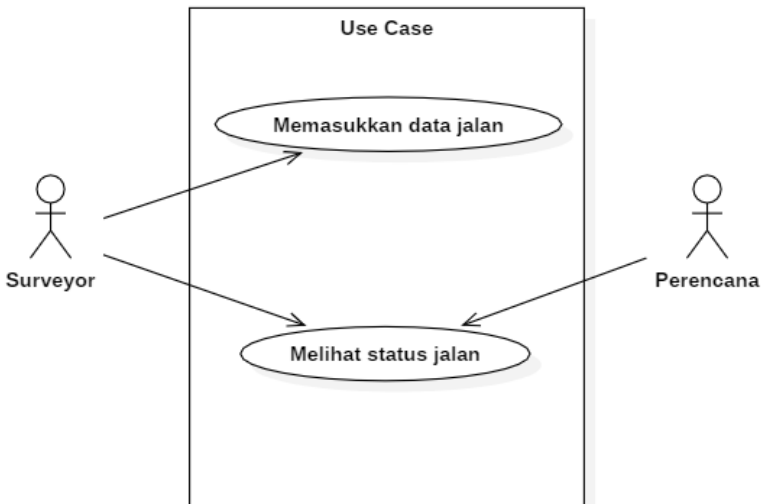
3.1.3.2 Definisi Kasus Penggunaan

Berdasarkan analisis kebutuhan aktor dari sistem, dibuat kasus penggunaan sistem. Kasus penggunaan sistem adalah hal-hal yang dapat dilakukan aktor untuk memperoleh suatu tujuan dengan perangkat lunak sebagai medianya. Kasus penggunaan harus dapat menyelesaikan semua permasalahan yang dimiliki aktor, dimana ketika permasalahan aktor tidak terselesaikan, maka kasus penggunaan harus dibenahi agar permasalahan aktor tersebut dapat terselesaikan. Kasus penggunaan dapat diperoleh dengan mengumpulkan kebutuhan-kebutuhan aktor lalu menyatukannya hingga menjadi kasus penggunaan yang utuh, ketika kasus penggunaan yang sudah dibuat sudah dapat memenuhi kebutuhan aktor, maka perancangan kasus penggunaan selesai.

Tabel 3.1 menjelaskan tentang kasus penggunaan (*Use Case*), dimana surveyor harus melakukan survey untuk memperoleh data jalan, lalu data jalan dimasukkan ke dalam sistem. Surveyor juga harus dapat melihat status jalan, dan juga perencana harus bisa melihat status jalan tersebut. Terkhusus untuk perencana harus dapat menggunakan data jalan untuk melakukan perbaikan, dan setelah perbaikan melakukan pembaruan data jalan. Gambar 3.2 menunjukkan hubungan antara aktor (surveyor dan perencana) dengan kasus penggunaan, dimana surveyor dapat melakukan kasus penggunaan memasukkan data jalan dan melihat status jalan, sedangkan perencana hanya dapat melihat status jalan. Namun perencana mendapat pilihan untuk melakukan pembaruan data jalan, setelah melihat status jalan.

Tabel 3.1 Kasus Penggunaan

Kode Kasus Penggunaan	Nama	Aktor
UC-01	Memasukkan data jalan	Surveyor
UC-02	Melihat status jalan	Surveyor dan Perencana



Gambar 3.2 Diagram Kasus Penggunaan

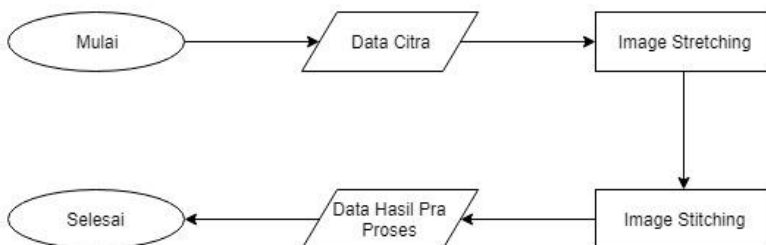
3.1.3.2.1 Merealisasikan Kasus Penggunaan Memasukkan Data Jalan (UC-01)

Kasus penggunaan nomor UC-01 yaitu memasukkan data jalan dibuat untuk mempermudah surveyor dalam penyampaian informasi mengenai data jalan kepada peran lain yang membutuhkan, dimana sebelumnya surveyor harus mengumpulkan data dan menyerahkan data tersebut kepada bagian arsip atau dokumen untuk diperiksa terlebih dahulu, lalu dicatat secara manual didalam arsip lainnya yang selanjutnya menggunakan arsip tersebut untuk mengambil keputusan terhadap jalan tersebut. Sehingga dibuat fitur yang dapat menyelesaikan masalah ini.

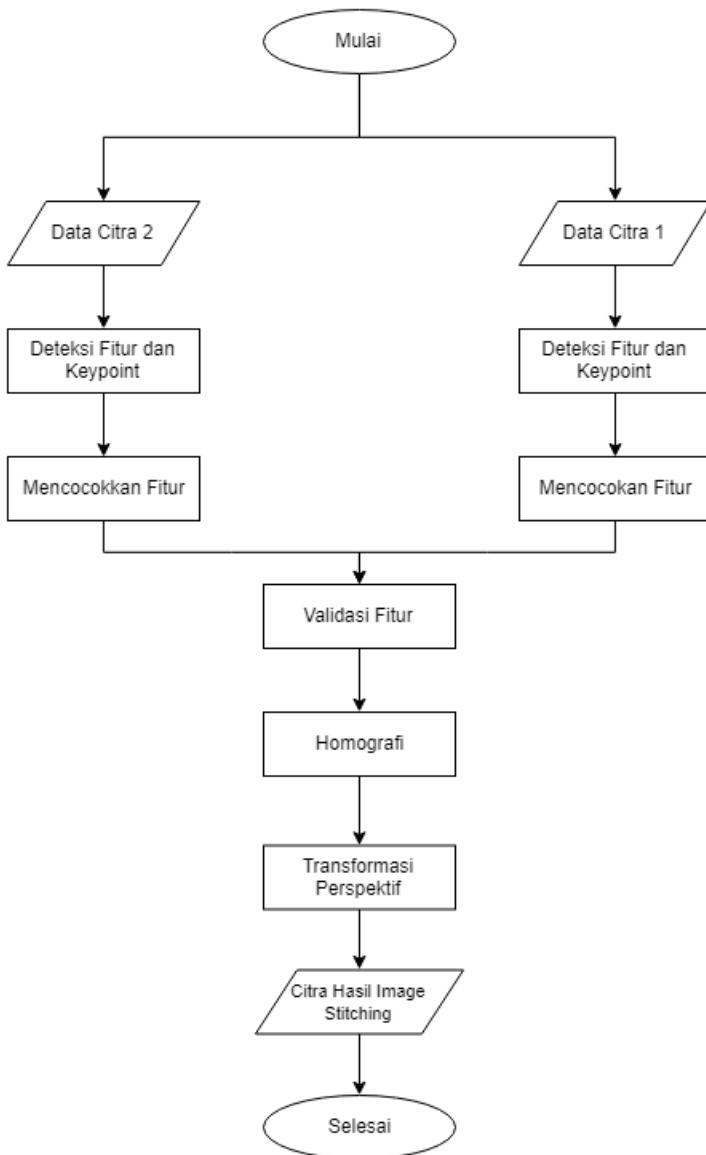
Kasus penggunaan nomor UC-01 diakses ketika surveyor (sebagai pengguna) hendak memasukkan data yang sudah diambil ke lapangan kedalam sistem. Data jalan terdiri dari status jalan dan sekelompok citra jalan. Status jalan terdiri dari nama jalan, keterangan citra, posisi, dan status jalan. Spesifikasi kasus

penggunaan Memasukkan Data Jalan dapat dilihat pada LAMPIRAN A, dan analisis kelas, diagram sekuens, dan diagram aktivitas kasus penggunaan ditunjukkan pada Gambar 3.5, Gambar 3.6, dan Gambar 3.7. Analisis kelas berguna untuk memeriksa kelas-kelas apa saja yang ada dalam kasus penggunaan ini. Dalam kasus ini ada 2 boundary yang berhubungan langsung dengan aktor, yaitu button dan form. Button merupakan bagian dari menu dan form. MasukkanDJWindow memiliki form dan controller yang menghubungkan dengan entity StatusJalan dan CitraJalan. Setelah semua kelas dijelaskan pada analisis kelas, kemudian dibuat proses yang jelas pada diagram sekuens. Awalnya surveyor berada pada menu, lalu surveyor klik tombol masukkanDJ, lalu halaman akan berpindah ke MasukkanDJWindow. Pada MasukkanDJWindow tersedia Form yang akan diisi data jalan oleh surveyor. Setelah selesai mengisi, surveyor mengklik tombol Submit lalu masukkanDJController memperoleh CitraJalan yang akan diproses dengan modul dan StatusJalan yang kemudian akan disimpan dalam basis data.

Modul yang digunakan pada kasus penggunaan ini adalah *Image Stretching* dan *Image Stitching*, dimana seluruh citra masukkan yang diterima akan melalui metode *Image Stretching* terlebih dahulu, lalu diproses dengan metode *Image Stitching*. Alur proses modul ditunjukkan pada Gambar 3.3.



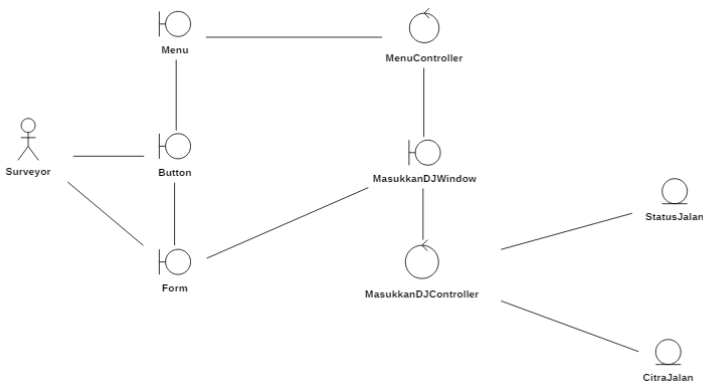
Gambar 3.3 Diagram Alur Modul



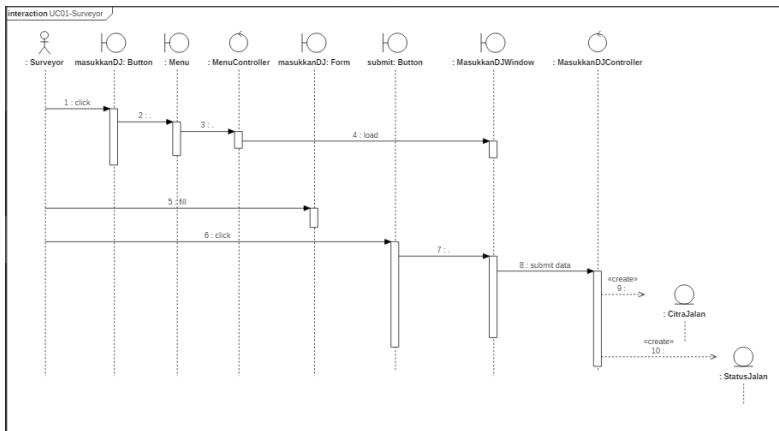
Gambar 3.4 Diagram Alur *Image Stitching*

Data citra yang dimasukkan pengguna dalam form difunakan dalam proses modul sebagai data masukkan. Data masukkan tersebut lalu diproses dengan *Image Stretching* yang akan menghasilkan data citra yang digunakan untuk data masukkan pada proses *Image Stitching*. Proses *Image Stitching* memiliki alur seperti yang ditunjukkan pada Gambar 3.4.

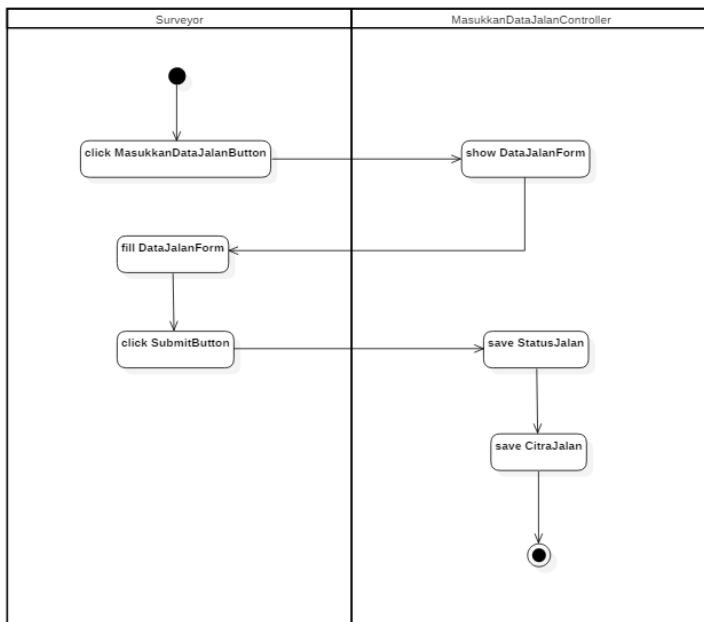
Pada proses *Image Stitching*, data citra yang diperoleh diproses setiap dua citra. Setiap citra dideteksi fitur dan *keypoint*-nya, lalu dicocokkan untuk masing-masing citra. Kemudian setelah fitru dan hasil fitur *keypoint*-nya dicocokkan, kedua citra tersebut divalidasi fiturnya. Setelah melalui proses validasi, citra melalui proses homografi untuk memeriksa pada posisi mana fitur terbaik diposisikan. Setelah melalui proses homografi maka citra digabungkan sesuai posisi hasil homografi dengan transformasi perspektif sehingga diperoleh citra hasil *Image Stitching* yang siap digunakan untuk data masukkan bagi modul-modul berikutnya. Untuk dapat digunakan ulang, maka citra hasil *Image Stitching* disimpan dalam direktori lokal.



Gambar 3.5 Analisis Kelas UC-01



Gambar 3.6 Diagram Sekuens UC-01

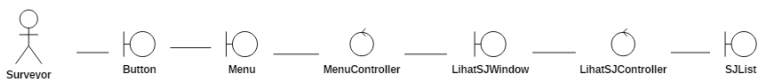


Gambar 3.7 Diagram Aktivitas UC-01

3.1.3.2.2 Merealisasikan Kasus Penggunaan Melihat Status Jalan (UC-02)

Kasus penggunaan nomor UC-02 yaitu melihat status jalan dibuat untuk mempermudah pengguna, baik surveyor maupun perencana untuk melihat status suatu jalan yang sudah pernah dimasukkan kedalam sistem, dimana sebelumnya pengguna harus datang ke lokasi arsip untuk melihat status jalan, lalu membuka arsip satu per satu untuk melihat banyak status jalan, sehingga dibuat fitur yang dapat mempermudah kasus ini.

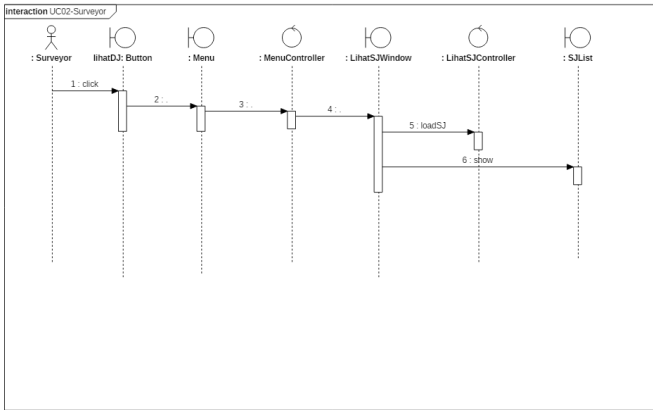
Kasus penggunaan nomor UC-02 diakses ketika surveyor atau perencana hendak melihat status jalan yang sudah pernah dimasukkan. Spesifikasi kasus penggunaan Melihat Status Jalan dapat dilihat pada LAMPIRAN B. Analisis kelas, diagram sekuens, dan Diagram aktivitas kasus penggunaan dengan aktor surveyor ditunjukkan pada Gambar 3.8, Gambar 3.10, dan Gambar 3.12 dan dengan aktor perencana ditunjukkan pada Gambar 3.9, Gambar 3.11, dan Gambar 3.13. Analisis kelas berguna untuk memeriksa kelas-kelas apa saja yang ada dalam kasus penggunaan ini. Pada kasus penggunaan ini, seluruh keadaan dan proses untuk kedua aktor yang berbeda adalah sama, yang berbeda hanyalah aktor tersebut sendiri. Dalam kasus ini hanya ada 1 boundary yaitu button yang berada pada menu, untuk membuka LihatSJWindow yang dengan LihatSJController menampilkan SJList kepada pengguna. Untuk diagram aktivitas, pada halaman awal aktor cukup mengklik tombol LihatStatusJalanButton, lalu sistem menampilkan StatusJalanList pada halaman baru.



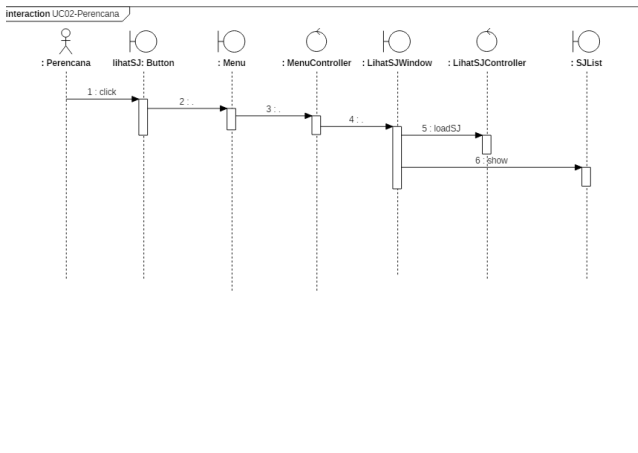
Gambar 3.8 Analisis Kelas UC-02 (Surveyor)



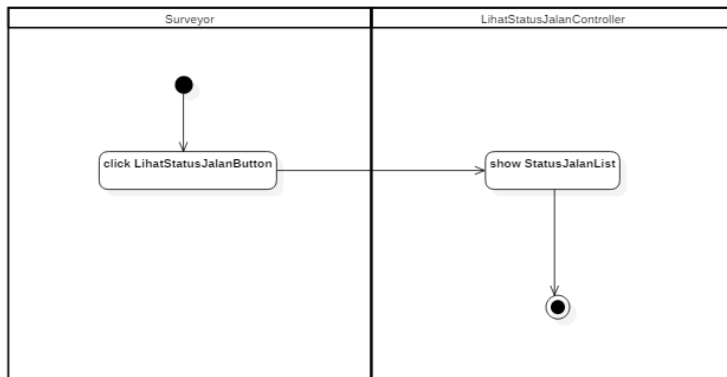
Gambar 3.9 Analisis Kelas UC-02 (Perencana)



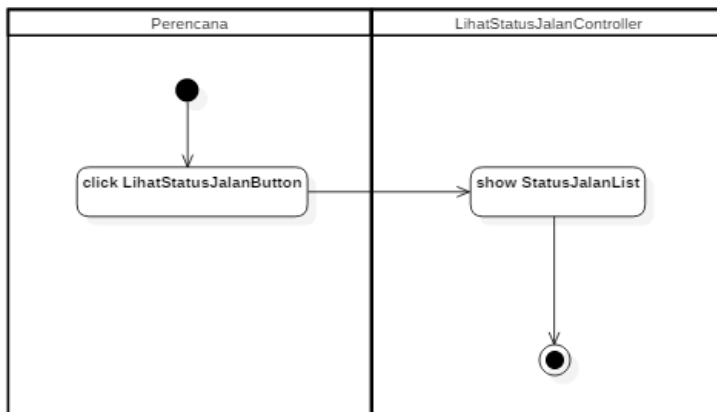
Gambar 3.10 Diagram Sekuens UC-02 (Surveyor)



Gambar 3.11 Diagram Sekuens UC-02 (Perencana)



Gambar 3.12 Diagram Aktivitas UC-02 (Surveyor)



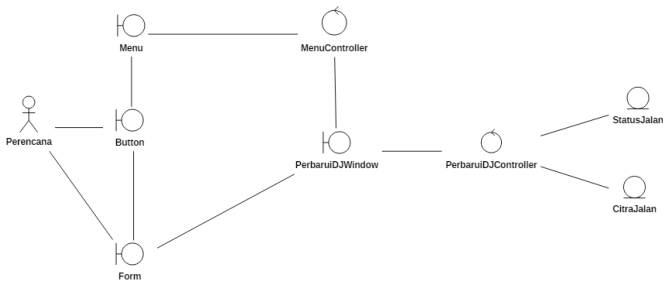
Gambar 3.13 Diagram Aktivitas UC-02 (Perencana)

3.1.3.2.2.1 Memperbarui Data Jalan (UC-02a)

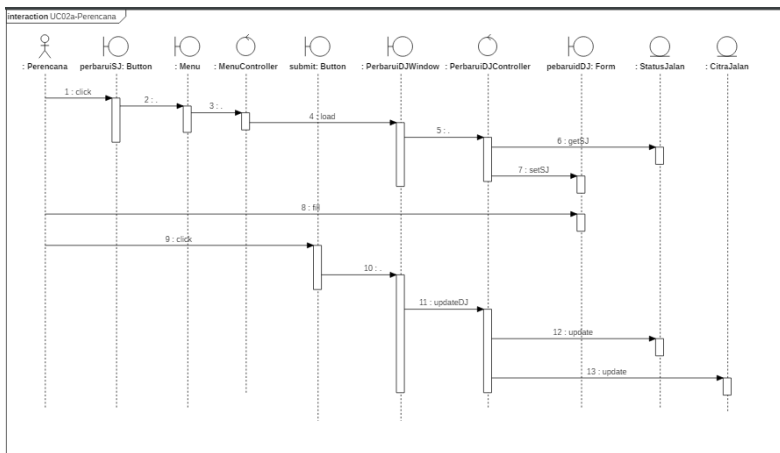
Kasus penggunaan nomor UC-02a yaitu memperbarui data jalan dibuat untuk mempermudah pengguna, yaitu perencana, memperbarui data dari data yang sudah pernah ada sebelumnya, dimana sebelumnya untuk memperbarui data pengguna harus mengubah arsip yang sudah ada secara manual ataupun membuat

arsip baru dengan nama yang menjelaskan bahwa data sebelumnya sudah diperbarui.

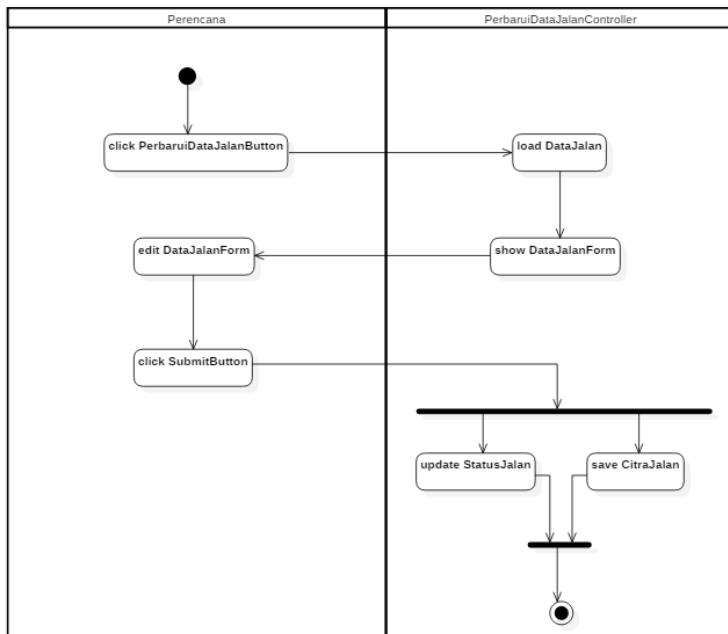
Kasus penggunaan nomor UC-02a diakses ketika perencana (sebagai pengguna) hendak memperbarui data yang sudah pernah dimasukkan kedalam sistem. Analisis kelas, diagram sekuens, dan Diagram aktivitas kasus penggunaan ditunjukkan pada Gambar 3.14, Gambar 3.15, dan Gambar 3.16. Analisis kelas berguna untuk memeriksa kelas kelas apa saja yang ada dalam kasus penggunaan ini.



Gambar 3.14 Analisis Kelas UC-02a



Gambar 3.15 Diagram Sekuens UC-02a



Gambar 3.16 Diagram Aktivitas UC-02a

3.1.1.3 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan layanan yang harus dimiliki oleh perangkat lunak, reaksi dari perangkat lunak terhadap suatu masukan, hasil yang dilakukan perangkat lunak pada situasi khusus. Pada tugas akhir ini kebutuhan fungsional dibuat berdasarkan kebutuhan aktor, untuk menyediakan segala kebutuhan di dalam aplikasi yang berfungsi sebagai media untuk mempermudah dan membuat pekerjaan aktor lebih efisien untuk memperoleh hasil yang berguna bagi pengguna. Kebutuhan fungsional dari perangkat lunak dijelaskan pada **Tabel 3.2**.

Tabel 3.2 Kebutuhan Fungsional

No	Kebutuhan Fungsional	Deskripsi
1	Memasukkan data jalan	Memasukkan data jalan, yaitu citra jalan serta status jalan yang sudah dilakukan pengambilan data oleh surveyor.
2	Melihat status jalan	Melihat kumpulan status jalan yang sudah dimasukkan oleh surveyor, status jalan melingkupi data jalan kecuali citra jalan itu sendiri. Status jalan terdiri dari nama jalan, keterangan gambar, posisi, dan status.
3	Melakukan pembaruan data jalan	Melakukan pembaruan data jalan dari data jalan yang sudah pernah dimasukkan.

3.1.3.4 Kebutuhan Non Fungsional

Tabel 3.3 Kualitas Perangkat Lunak

No	Parameter	Deskripsi
1	Bahasa	Bahasa yang digunakan pada antarmuka merupakan bahasa Inggris
2	Integritas	Hanya pengguna dengan peran tertentu yang mempunyai wewenang untuk melakukan fungsi tertentu sesuai fungsi yang sudah disediakan untuk peran pengguna
3	Portablilitas	Aplikasi disediakan dalam bentuk aplikasi Desktop
4	Kebenaran	Aplikasi menghasilkan keluaran yang benar untuk setiap kemungkinan masukan oleh pengguna

Kebutuhan non fungsional mendefinisikan batasan layanan yang ditawarkan oleh sistem. Kebutuhan non fungsional dibuat berdasarkan kebutuhan pengguna untuk mengakses kebutuhan fungsional yang ada. Kebutuhan non fungsional yang ada pada perangkat lunak ini disediakan dengan fokus kepada pengguna.

Kebutuhan non fungsional dari perangkat lunak dijelaskan pada Tabel 3.3.

3.2 Perancangan

Dalam subbab ini membahas perancangan dari aplikasi Tugas Akhir. Subbab ini terdiri dari lingkungan perancangan perangkat lunak, perancangan diagram kelas dan perancangan antarmuka pengguna.

3.2.1 Perancangan Diagram Kelas

Diagram kelas ditunjukkan pada Gambar E.1. Pada Tugas Akhir ini menjelaskan hubungan antar kelas yang ada dalam sistem dalam mencapai kebutuhan fungsional. Pada awal proses aktor akan berada didalam Menu, dari menu dapat mengklik Button untuk pergi ke window lain, MasukkanDJWindow atau LihatSJWindow. Window MasukkanDJWindow memiliki Form untuk mengisikan data jalan, yang dilengkapi dengan Button submit untuk memanggil MasukkanDJController untuk berinteraksi dengan CitraJalan dan StatusJalan. Serta Window LihatSJWindow memiliki LihatSJController untuk menunjukkan SJList kepada pengguna, serta dari window LihatSJWindow dapat mengklik Button untuk pergi ke PerbaruiDJWindow. PerbaruiDJWindow memiliki PerbaruiDJController untuk memanggil SJList untuk ditunjukkan pada pengguna. Dengan kondisi awal berada pada LihatSJWindow, perencana mengklik tombol untuk masuk ke PerbaruiDJWindow, lalu sistem membuka halaman dengan form didalamnya. Perencana mengisi form tersebut, lalu mengklik tombol submit, kemudian sistem menyimpan data yang sudah diisikan.

3.2.2 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan perangkat lunak. Antarmuka pengguna yang berhubungan dengan aktor harus memberikan kemudahan serta tampilan yang menarik bagi penggunanya. Pada aplikasi ini, pengguna yaitu surveyor dan

perencana dapat mengakses antarmuka yang sesuai dengan peran masing masing dengan antarmuka yang ditampilkan pada aplikasi.

3.2.2.1 Rancangan Halaman Antarmuka Memasukkan Data Jalan

Halaman ini digunakan oleh surveyor untuk memasukkan data jalan kedalam sistem seperti yang dijelaskan pada UC-01 yang berupa status jalan, yang terdiri dari nama jalan, keterangan gambar, posisi, dan status jalan, dan citra jalan. Rancangan halaman diperoleh dari boundary yang ada pada pada Gambar 3.5. Ada dua boundary yang berhubungan dengan aktor, yaitu Button dan Form, Button (untuk pergi ke MasukkanDJWindow) terletak pada Menu, ketika diklik maka window akan berubah menjadi MasukkanDJWindow yang memiliki Form didalamnya, serta Button (untuk submit) didalam Form tersebut. Form untuk mengisi data memiliki beberapa kolom untuk diisi seperti nama jalan, keterangan citra, posisi, status jalan, serta citra jalan. Untuk memasukkan citra jalan aktor harus membuka *Window Explorer* untuk memilih citra yang akan dimasukkan kedalam sistem, setelah memasukkan data citra

3.2.2.2 Rancangan Halaman Antarmuka Melihat Status Jalan

Halaman ini digunakan oleh surveyor atau perencana untuk melihat daftar status jalan yang didapatkan dari tiap memasukkan data jalan seperti yang dijelaskan pada UC-02. Rancangan halaman diperoleh dari boundary yang ada pada pada Gambar 3.8 dan Gambar 3.9. Aktor akan memilih tombol untuk pergi ke halaman LihatSJWindow pada Menu. Halaman ini menyediakan status jalan dalam bentuk tabel, tabel akan berisi nama jalan, keterangan citra, posisi, status jalan, serta dilengkapi dengan nomor unik setiap status jalan yang berada ditunjukkan dengan SJList. Halaman ini akan membedakan aktornya, karena pada halaman ini untuk aktor perencana dapat melanjutkan ke halaman perbarui data jalan, yaitu dengan menekan tombol untuk memperbarui pada status jalan yang

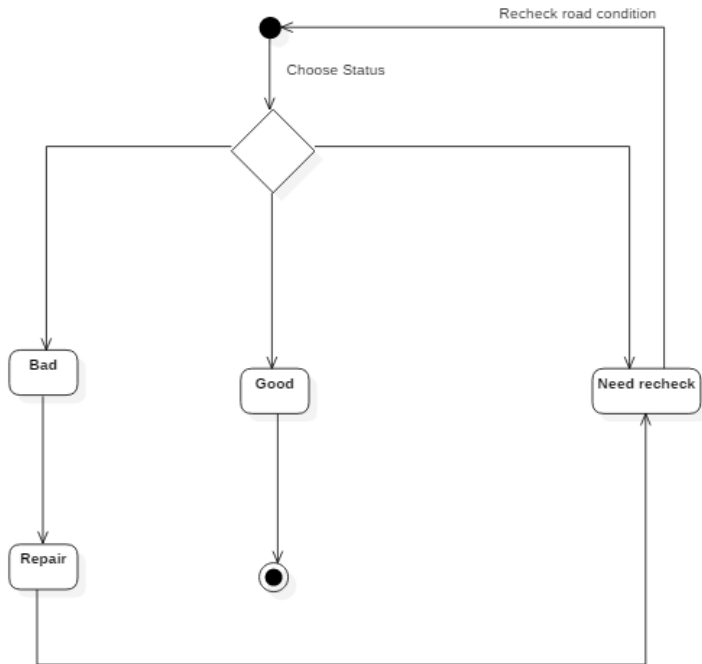
akan diperbarui, sehingga tombol perbarui tersebut hanya tersedia pada aktor perencana.

3.2.2.3 Rancangan Halaman Antarmuka Memperbarui Data Jalan

Halaman ini digunakan oleh perencana untuk melihat memperbarui data jalan yang sudah pernah dimasukkan, yang merupakan ekstensi dari kasus penggunaan melihat status jalan. Untuk masuk ke halaman ini, aktor yang dapat mengakses hanyalah perencana. Pada halaman melihat data jalan akan tersedia tombol untuk masuk ke halaman memperbarui data jalan, seperti yang ditunjukkan pada Gambar 3.14. Rancangan pada halaman ini serupa dengan rancangan pada subbab 3.2.2.1, hanya saja pada halaman ini data jalan sudah tersedia sesuai dengan yang ada pada halaman lihat status jalan.

3.2.3 Perancangan Diagram Keadaan

Diagram keadaan atau *state diagram* ditunjukkan pada Gambar 3.17. Diagram keadaan perangkat lunak ini menjelaskan proses mencapai tujuan akhir perangkat lunak ini. Pada diagram terdapat tiga keadaan yang menjelaskan mengenai keadaan jalan, yaitu *Bad*, *Good*, *Need Recheck*. *Bad* adalah kondisi jalan yang rusak dimana harus diperbaiki, setelah diperbaiki maka akan membutuhkan pengecekan ulang yang dijelaskan keadaan *Need Recheck*. *Need Recheck* juga dapat dilakukan ketika citra yang diperoleh tidak memenuhi kebutuhan atau tidak dapat diproses, yang dapat disebabkan oleh gangguan lingkungan sekitar. Untuk keadaan *Good* merupakan keadaan yang diharapkan untuk jalan tersebut, sehingga jika jalan sudah berada dalam keadaan *good*, maka proses perbaikan atau pemeriksaan jalan dianggap selesai.



Gambar 3.17 Diagram keadaan (*State Diagram*)

BAB IV IMPLEMENTASI

Bab ini membahas implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program Tugas Akhir ini dan disertai dengan kode sumber masing-masing fungsi utama.

4.1 Lingkungan Implementasi Perangkat Lunak

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

Perangkat Keras	Komputer	Prosesor: Intel® Core™ i5-4460 CPU @ 3.20GHz Memori: 8192MB RAM
	Drone	Phantom 3 Professional
Perangkat Lunak	Sistem Operasi	Windows Home 64-bit
	Perangkat Lunak	Star UML 3.1, Visual Studio Code, Anaconda, Spyder 3.3 (Python 3.6), Microsoft Word 2015
	<i>Library</i>	OpenCV, Numpy, PIL

Lingkungan implementasi sistem yang digunakan untuk mengembangkan Tugas Akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti ditampilkan pada Tabel 4.1.

4.2 Implementasi Kasus Penggunaan

Implementasi kasus penggunaan menggunakan *Framework* Electron yang menggunakan HTML, CSS, dan JavaScript sebagai dasarnya serta menggunakan Node.js untuk menempurnakan kasus penggunaan. Pada subbab ini akan menjelaskan dan menampilkan kode yang digunakan dalam pembuatan Aplikasi Pemetaan Kerusakan Jalan pada beberapa kasus penggunaan.

4.2.1 Implementasi Kasus Penggunaan Memasukkan Data Jalan

Kode sumber dari implementasi kasus penggunaan memasukkan data jalan ditunjukkan pada subbab ini.

Untuk memasukkan data jalan kedalam sistem, aplikasi akan menyediakan formulir untuk diisi pada tampilan halaman. pengguna dapat memasukkan data jalan kedalam formulir, termasuk data citra. Pada saat pengguna menekan tombol untuk memasukkan data citra, maka muncul *file explorer* untuk memilih data citra, ketika selesai memilih citra yang akan dimasukkan kedalam sistem, pengguna akan menekan tombol untuk menyelesaikan memasukkan data citra, setelah tombol ditekan, maka citra akan ditampilkan sebagai pratinjau.

```

1. $("#fileupload1").change(function () {
2.     if (typeof (FileReader) !== "undefined") {
3.         var dvPreview = $("#dvPreview1");
4.         dvPreview.html(" ");
5.         var regex = /^[a-zA-Z0-9\s_\\.\-
:]+(\.jpg|.jpeg|.gif|.png|.bmp)$/;
6.         let sortedFiles = sortFiles($(this)[0].files
s);
7.         $(sortedFiles).each(function () {
8.             var file = $(this);

```



```

9.         if (regex.test(file[0].name.toLowerCase
10.         ())) {
11.             var reader = new FileReader();
12.             reader.onload = function (e) {
13.                 var img = $("

```

Kode Sumber 1 Implementasi Menampilkan Pratinjau Data

Kode Sumber 1 menjelaskan tentang bagaimana aplikasi menampilkan citra pratinjau, pada awalnya ketika citra dimasukkan kedalam sistem oleh pengguna, sistem membaca data tersebut lalu memasukkan data citra tersebut ke dalam variabel `dvPreview`, namun sebelum sistem memasukkan data citra ke dalam variabel, citra harus terlebih dulu diurutkan dengan fungsi `sortFiles` yang ditunjukkan pada Kode Sumber 2, karena untuk memperoleh citra jalan yang merupakan gabungan dari semua masukkan citra diperlukan citra yang sudah urut. Untuk mengurutkan data fungsi `sortFiles` melakukan perulangan untuk memeriksa nama dari kumpulan data citra yang ada. Fungsi akan memeriksa dua per dua nama citra dari awal sampai akhir sehingga semua citra terperiksa. Ketika nama citra ke dua dimulai dari abjad yang lebih besar dari abjad nama citra pertama, maka urutan citra tetap, sebaliknya urutan citra ditukar, begitu seterusnya (baris 3-14 Kode Sumber 2). Setelah citra diurutkan, maka citra dibuat dalam bentuk pratinjau secara acak lalu dimasukkan kedalam variabel

dvPreview untuk ditampilkan (baris 7-21) dan tidak lupa data citra dimasukkan kedalam variabel `is` untuk dimasukkan ke dalam sistem nantinya.

```

1. function sortFiles(listFiles) {
2.     var temp = [...listFiles];
3.     for(var x = 0; x < temp.length-1; x++)
4.     {
5.         for(var y = x+1; y < temp.length; y++)
6.         {
7.             if(temp[x].name > temp[y].name)
8.             {
9.                 var temp1 = temp[y];
10.                temp[y] = temp[x];
11.                temp[x] = temp1;
12.            }
13.        }
14.    }
15.    return temp;
16. }

```

Kode Sumber 2 Implementasi fungsi `sortFiles`

Untuk memasukkan data yang sudah dimasukkan pengguna kedalam sistem, maka pengguna harus mengklik tombol submit. Ketika tombol ditekan maka sistem akan menyimpan nama jalan dalam variabel `roadname`, keterangan gambar dalam variabel `image_note`, posisi dalam variabel `position`, dan status dalam variabel `status` (baris 2-5 Kode Sumber 3). Lalu data tersebut dimasukkan kedalam basis data dengan fungsi `query`. Karena untuk menyimpan data citra diperlukan tempat, maka dipersiapkan 3 folder, yaitu `input`, `script1`, dan `script2` yang masing masing dilengkapi kode unik, yaitu kode data terakhir yang dimasukkan kedalam sistem (baris 10-13 Kode Sumber 3). Lalu dibuat arsip untuk dengan nama unik pada folder `input` dengan variabel `os`, lalu dimasukkan data citra dari variabel `is` kedalam variabel `os`.

Setelah masukkan data citra disimpan pada folder `input`, maka dipanggil modul *Image Stitching* dengan keluaran data citra yang

sudah dimanipulasi untuk semirip mungkin dengan kenyataan didalam folder script1. Setelah proses *Image Stitching* selesai maka sistem menjalankan modul *Image Stretching* untuk menyatukan citra yang ada di folder script1 lalu memberikan keluaran data citra pada folder script2.

```

1. document.getElementById('submitButton').addEventListener("click", function(){
2.     var roadname = document.getElementById("roadname").value;
3.     var image_note = document.getElementById("image_note").value;
4.     var position = document.getElementById("position").value;
5.     var status = document.getElementById("status").value;
6.     var lastid = null;
7.
8.     var sql = insertIntoQuery(roadname, image_note, position, status);
9.     query(sql, function(err, rows, fields) {
10.         lastid = rows.insertId;
11.         mkdirp('./image/input/'+lastid+');
12.         mkdirp('./image/script1/'+lastid+');
13.         mkdirp('./image/script2/'+lastid+');
14.
15.         lastid = rows.insertId;
16.         var flag = 0;
17.
18.         for (var i = 0; i < is.length; i++) {
19.
20.             os[i] = fs.createWriteStream(path.join(__dirname, './image/input/'+lastid+'/' +i+'.jpg'));
21.             is[i].pipe(os[i]);
22.
23.             // process = call image stitching
24.
25.             process.stdout.on('data', function(data) {
26.                 flag = flag + 1;

```

```

27.
28.         if(flag == is.length){
29.             // process2 = call image stretc
    hing
30.         }
31.     });
32. }
33. });
34. });

```

Kode Sumber 3 Implementasi memasukkan data kedalam sistem

4.2.2 Implementasi Kasus Penggunaan Melihat Status Jalan

Kode sumber dari implementasi kasus melihat status jalan ditunjukkan pada subbab ini.

Pada halaman melihat status jalan, pengguna dapat melihat daftar status jalan yang ada didalam sistem. Kode sumber dari implementasi melihat status jalan ditunjukkan pada Kode Sumber 4. Pada saat halaman dibuka, maka sistem akan memanggil fungsi query untuk mengambil status jalan dari basis data, lalu menampilkan kumpulan status jalan pada tabel.

```

1. //sql = status jalan
2. query(sql, function(err, rows, fields) {
3.     rows.forEach(element => {
4.         let table_rows = `
5.             <tr>
6.                 //memasukkan data ke dalam tabel
7.             </tr>
8.         `;
9.         $("#table_show").append(table_rows);
10.     });
11. });

```

Kode Sumber 4 Implementasi Melihat Status Jalan

4.2.3 Implementasi Kasus Penggunaan Memperbarui Data Jalan

Untuk kasus penggunaan memperbarui data jalan, sebagian besar kode sumber sama dengan subbab 4.2.1 karena fungsi yang digunakan adalah fungsi yang sama, seperti fungsi untuk menampilkan pratinjau data citra pada Kode Sumber 1 dan Kode Sumber 2. Namun, untuk memasukkan data ke dalam basis data, cara yang digunakan sedikit berbeda, karena data sudah pernah dimasukkan ke dalam basis data, maka hanya perlu melakukan pembaruan data dimana dapat menambahkan kode sumber seperti yang ditunjukkan pada Kode Sumber 5, untuk mengambil data yang sudah pernah diisi untuk ditampilkan agar pengguna tahu data apa yang sudah pernah diisi. Lalu memasukkan data seperti pada Kode Sumber 3. Namun pada implementasi kasus penggunaan memperbarui data jalan, tidaklah diperlukan untuk menjalankan modul *Image Stitching* dan *Image Stretching* karena citra yang diambil bukan oleh surveyor, melainkan petugas yang melakukan perbaikan jalan.

```

1. // sql = selectAllQuery()
2. query(sql, function(err, rows, fields) {
3.     $("#roadname").val(rows[0].status_roadname)
4.     $("#image_note").val(rows[0].status_image_note)
5.     $("#position").val(rows[0].status_position)
6.     $("#status").val(rows[0].status_condition)
7. });

```

Kode Sumber 5 Implementasi mengambil data dari basis data

4.3 Implementasi Modul

Pada perangkat lunak dalam Tugas Akhir ini, sistem akan mengolah data dengan salah satu data yang berupa citra. Maka perangkat lunak menyediakan modul yang akan mengolah data citra tersebut.

4.3.1 Modul *Image Stretching*

Image Stretching yang dimaksud adalah memanipulasi data citra untuk mengembalikan dimensi data citra agar sesuai dengan dimensi asal citra tersebut. Serupa dengan kasus ketika kita melihat gedung yang tinggi, maka seolah olah gedung yang tinggi itu mengecil sampai ujung atas gedung tersebut tidak kelihatan, padahal pada kenyataannya gedung tersebut memiliki ukuran yang sama dari bawah sampai atas, tetapi mata kita melihat gedung tersebut seperti mengecil, hal ini disebut persepsi. Tujuan utama dari modul ini adalah untuk mengembalikan data citra yang merupakan persepsi kamera menjadi data yang diusahakan sesesuai mungkin dengan kenyataan.

Proses pertama dalam *Image Stretching* adalah mengambil citra masukkan dan membuat bingkai citra (baris 2-4) kosong untuk meletakkan citra hasil proses pada bingkai tersebut yang ditunjukkan pada Kode Sumber 6, bingkai memiliki lebar yang sama, namun tinggi yang berbeda, tinggi bingkai lebih dari tinggi citra masukkan. Pada modul ini citra yang digunakan adalah citra vertikal (tinggi citra lebih dari lebar citra), ketika citra yang diterima horizontal (tinggi citra kurang dari lebar citra) maka citra harus diputar agar menjadi vertikal. Serta untuk melakukan proses selanjutnya citra sebelum proses dan bingkai dimasukkan kedalam variabel a dan b.

```

1. im = Image.open(path)
2. frame_width = im.width
3. frame_height = im.height + int(im.height * 0.02)
4. new_size = (frame_width, frame_height)
5. new_im = Image.new("RGB", new_size, (0,0,0))
6. empty = new_im.height - im.height
7. fill = int((new_im.height/2) / (empty/2))
8. diff = int((empty)/2)
9. a = new_im.load()
10. b = im.load()

```

Kode Sumber 6 Membuat Bingkai Citra Keluaran

Saat sudah tersedia bingkai kosong untuk meletakkan citra keluaran maka citra masukan harus diproses dan diletakkan dengan pola pada bingkai tersebut. Proses peletakan citra ditunjukkan pada Kode Sumber 7 hasil proses pada bingkai dibagi menjadi empat bagian sama rata, yaitu kiri atas (baris 1-18), kiri bawah (baris 20-37), kanan atas (baris 39-56), kanan bawah (baris 58-74), dengan masing-masing pola peletakan yang berbeda-beda.

Pola peletakan citra adalah dengan mengambil nilai pixel dari citra masukan lalu memasukkan pixel tersebut kedalam bingkai. Pada bagian kiri atas citra diletakkan dari kiri ke kanan dan dari atas ke bawah dengan pixel yang berada di sumbu x citra masukan dan citra yang dibuat sama, namun pixel pada sumbu y berbeda, mengikuti pola yang didasari dengan persamaan garis lurus. Citra bagian kiri bawah pixel diletakkan dari kiri ke kanan dan dari bawah ke atas. Citra bagian kanan atas diletakkan dari kiri ke kanan dan dari atas ke bawah. Citra bagian kanan bawah diletakkan dari kiri ke kanan dan bawah ke atas ke bawah.

```

1. for i in range (new_im.width//2):
2.     k = 0
3.     print0 = 0
4.     total_fill = int(((diff / (new_im.width / 2)) *
5.         i))
6.     for j in range (int((diff - (diff / (new_im.wid
7.         th / 2)) * i)), new_im.height//2):
8.         if print0 < fill:
9.             a[i,j] = b[i,k]
10.            k += 1
11.            print0 += 1
12.        else:
13.            if total_fill > 0:
14.                a[i,j] = (0,0,0)
15.                print0 = 0
16.                total_fill -= 1
17.            else:
18.                a[i,j] = b[i,k]
19.                k += 1
20.                print0 += 1

```

```

20. for i in range (new_im.width//2):
21.     k = im.height-1
22.     print0 = 0
23.     total_fill = int(((diff / (new_im.width / 2)) *
    i))
24.     for j in range (new_im.height - 1 - int(diff -
    ((diff / (new_im.width // 2)) * i)), new_im.height/
    /2 - 1 , -1):
25.         if print0 < fill:
26.             a[i,j] = b[i,k]
27.             k -= 1
28.             print0 += 1
29.         else:
30.             if total_fill > 0:
31.                 a[i,j] = (0,0,0)
32.                 print0 = 0
33.                 total_fill -= 1
34.             else:
35.                 a[i,j] = b[i,k]
36.                 k-= 1
37.                 print0 += 1
38.
39. for i in range (new_im.width//2 , new_im.width ):
40.     k = 0
41.     print0 = 0
42.     total_fill = int(diff - ((diff / (new_im.width
    / 2)) * (i - new_im.width //2)))
43.     for j in range (int(- diff + ((2 * diff / (new_
    im.width)) * i)) , new_im.height//2):
44.         if print0 < fill:
45.             a[i,j] = b[i,k]
46.             k += 1
47.             print0 += 1
48.         else:
49.             if total_fill > 0:
50.                 a[i,j] = (0,0,0)
51.                 print0 = 0
52.                 total_fill -= 1
53.             else:
54.                 a[i,j] = b[i,k]
55.                 k += 1
56.                 print0 += 1
57.

```



```

58. for i in range (new_im.width // 2 , new_im.width )
   :
59.     k = im.height - 1
60.     total_fill = int(diff - ((diff / (new_im.width
   / 2)) * (i - new_im.width //2)))
61.     for j in range (new_im.height -
   1 - int(- diff + ((2 * diff / (new_im.width)) * i))
   , new_im.height//2 - 1, -1):
62.         if print0 < fill:
63.             a[i,j] = b[i,k]
64.             k -= 1
65.             print0 += 1
66.         else:
67.             if total_fill > 0:
68.                 a[i,j] = (0,0,0)
69.                 print0 = 0
70.                 total_fill -= 1
71.             else:
72.                 a[i,j] = b[i,k]
73.                 k -= 1
74.                 print0 += 1

```

Kode Sumber 7 Membuat Citra Baru pada Bingkai

Setelah menyelesaikan Kode Sumber 7 maka bingkai sudah terisi oleh pixel pixel yang diterima dari citra masukkan.

Setelah bingkai kosong diberi isian data citra, citra kosong tadi sudah memiliki isi, namun masih memiliki bagian kosong. Maka bagian kosong harus diisi dengan nilai (dalam bentuk warna Red Green Blue) rata-rata dari pixel-pixel disekitar bagian tersebut, yang ditunjukkan pada Kode Sumber 8, dimana pengambilan nilai rata rata ini adalah untuk mengisi pixel kosong agar menyerupai pixel-pixel disekitarnya (bias disebut proses manipulasi citra). Pada proses ini gambar tetap dibagi menjadi empat bagian samarata seperti proses sebelumnya, namun tidak ada pola pada proses ini, hanya mengisi bagian kosong dari pola yang diciptakan pada proses sebelumnya.

```

1. for i in range (1, new_im.width//2):

```

```

2.     for j in range (1, new_im.height//2):
3.         if a[i,j] == (0,0,0):
4.             neighbors = [a[i-1,j-1], a[i,j-
5.                 1], a[i+1,j-1], a[i-1,j], a[i,j+1], a[i-1,j+1]]
6.             sum_of_neighbors = reduce(lambda x, y:
7.                 np.add(np.array(x), np.array(y)), neighbors)
8.             avg = np.array(sum_of_neighbors) // 6
9.             a[i,j] = tuple(avg)
10.
11.    for i in range (1, new_im.width//2):
12.        for j in range (new_im.height//2, new_im.height
13.            - 1):
14.            if a[i,j] == (0,0,0):
15.                neighbors = [a[i-1,j-1], a[i,j-1], a[i-
16.                    1,j], a[i-1,j+1], a[i,j+1], a[i+1,j+1]]
17.                sum_of_neighbors = reduce(lambda x, y:
18.                    np.add(np.array(x), np.array(y)), neighbors, [0, 0,
19.                        0])
20.                avg = np.array(sum_of_neighbors) // 6
21.                a[i,j] = tuple(avg)
22.
23.    for i in range (new_im.width//2, new_im.width -
24.        1):
25.        for j in range (1, new_im.height//2 ):
26.            if a[i,j] == (0,0,0):
27.                neighbors = [a[i-1,j-1], a[i,j-1], a[i-
28.                    1,j], a[i-1,j+1], a[i,j+1], a[i+1,j+1]]
29.                sum_of_neighbors = reduce(lambda x, y:
30.                    np.add(np.array(x), np.array(y)), neighbors, [0, 0,
31.                        0])

```

```

30.         avg = np.array(sum_of_neighbors) // 6
31.         a[i,j] = tuple(avg)

```

Kode Sumber 8 Proses Manipulasi Citra

Setelah citra yang baru sudah melalui proses, maka citra yang baru harus dikembalikan ke ukuran awal, yaitu dengan ukuran yang sama dengan citra masukan, agar tidak ada pixel yang kosong. Lalu dibuat bingkai dengan ukuran sama seperti citra masukan seperti yang ditunjukkan pada Kode Sumber 9. Lalu hasil manipulasi citra diletakkan pada bingkai tersebut dengan posisi titik pusat yang sama, sehingga bagian yang terbuang hanya merupakan bagian tepi yang tidak terlalu berguna untuk proses.

```

1. crop_size = im.size
2. crop_image = Image.new("RGB", crop_size, (0,0,0))
3. crop_image.paste(new_im, ((crop_size[0] - new_im.size[0]) // 2, (crop_size[1] - new_im.size[1]) // 2))

```

Kode Sumber 9 Memindahkan Citra ke Bingkai dengan Ukuran Sama Seperti Citra Masukan

4.3.2 Modul *Image Stitching*

Stitching Image yang dimaksud adalah menyatukan beberapa citra menjadi satu citra yang mengandung beberapa citra masukan tersebut. Data citra yang akan disatukan harus memiliki bagian yang berhubungan atau timpang tindih satu sama lain, sebagai contoh akan menyatukan dua citra, maka citra satu dan citra dua harus memiliki irisan, dimana irisan tersebut adalah bagian citra yang ada di citra satu maupun citra dua, semakin banyak dan jelas irisan citra tersebut maka akan semakin baik hasil dari proses penyatuan citra tersebut. Pada Tugas Akhir ini tujuan dari *Stitching Image* adalah menyatukan beberapa citra jalan menjadi satu citra panjang yang mempresentasikan jalan, untuk diukur dimensinya. Karena akan mengukur dimensi, maka citra yang digunakan sebagai masukan harus memiliki kondisi bagus, yang artinya jelas

dan tidak memiliki gangguan. Kumpulan data citra yang akan digunakan sebagai masukan modul *Stitching Image* harus melalui modul *Stretching Image* terlebih dahulu, karena citra yang disatukan akan digunakan untuk mengukur dimensi aslinya nantinya.

Untuk prosesnya citra akan diletakkan berurutan pada suatu variabel, karena proses penyatuan citra hanya dapat dilakukan kepada citra yang mempunyai irisan sehingga citra harus dibuat berurut untuk diproses, seperti yang ditunjukkan pada Kode Sumber 10, lalu memanggil fungsi Panorama untuk menjalankan proses *Image Stitching*.

```

1. images = []
2.
3. for i in range(no_of_images):
4.     images.append(cv2.imread(filename[i]))
5.
6. for i in range(no_of_images):
7.     images[i] = imutils.resize(images[i], width=400
8.     )
9. for i in range(no_of_images):
10.    images[i] = imutils.resize(images[i], height=40
11.    0)
12. panorama = Panorama()

```

Kode Sumber 10 Memulai proses *Image Stitching*

Panorama adalah sebuah kelas yang memiliki beberapa fungsi untuk menjalankan proses *Image Stitching*. Fungsi pertama dalam Panorama yaitu `image_stitch` yang ditunjukkan pada Kode Sumber 11, yang melakukan pra proses dengan memasukkan citra ke dalam variabel `imageB` dan `imageA` lalu menjalankan fungsi `Detect_Feature_and_Keypoints` kepada masing-masing citra untuk mencari poin-poin inti dari citra (baris 3-5). Setelah memperoleh poin inti maka dicari titik persamaan inti tersebut dengan fungsi `matchKeypoints` (baris 6). Untuk mengambil berbagai perspektif

dari citra maka perlu dihitung homografi citra dengan `getwarp_perspective` (baris 12)

```

1. def image_stitch(self, images, Lowe_ratio=0.75, max
   _Threshold=4.0, match_status=False):
2.
3.     (imageB, imageA) = images
4.     (KeypointsA, features_of_A) = self.Detect_Featu
   re_And_KeyPoints(imageA)
5.     (KeypointsB, features_of_B) = self.Detect_Featu
   re_And_KeyPoints(imageB)
6.     Values = self.matchKeypoints(KeypointsA, Keypoi
   ntsB, features_of_A, features_of_B, Lowe_ratio, max_
   Threshold)
7.
8.     if Values is None:
9.         return None
10.
11.    (matches, Homography, status) = Values
12.    result_image = self.getwarp_perspective(imageA,
   imageB, Homography)
13.    result_image[0:imageB.shape[0], 0:imageB.shape[
   1]] = imageB
14.
15.    return result_image

```

Kode Sumber 11 Fungsi `image_stitch`

Untuk memulai proses penyatuan citra, dimulai dengan mencari poin-poin inti citra dengan fungsi `Detect_Feature_And_KeyPoints`, yang ditunjukkan Kode Sumber 12. Untuk memeriksa poin tersebut citra harus dalam bentuk abu-abu, sehingga citra yang berwarna diubah menjadi abu-abu (baris 2), lalu poin dicari dengan *library* yang disediakan `xfeatures2d`, dengan fungsi `detectAndCompute` (baris 3-5).

```

1. def Detect_Feature_And_KeyPoints(self, image):

```

```

2.     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
3.     descriptors = cv2.xfeatures2d.SIFT_create()
4.     (Keypoints, features) = descriptors.detectAndCo
mpute(image, None)
5.     Keypoints = np.float32([i.pt for i in Keypoints
])

```

Kode Sumber 12 Fungsi Detect_Feature_And_Keypoints

Dengan poin-poin inti citra yang telah diperoleh, maka poin-poin antar citra tersebut dikomputasikan dan dicocokkan dengan fungsi `get_Allpossible_Match` (baris 2 Kode Sumber 13) yang diperoleh dari fungsi yang ada pada library `cv2` dengan teknik *k-Nearest Neighbors* (Kode Sumber 14), lalu diambil poin-poin yang sesuai dengan fungsi `All_validmatches`, dimana untuk setiap poin poin yang ditemukan dikalkulasikan dengan konsep `lowe_ratio`, ketika sesuai maka poin dianggap valid (Kode Sumber 15). Pada kasus ini, ketika ada lebih dari 4 poin yang sesuai, maka kedua citra tersebut didefinisikan dapat disatukan (baris 4-6 Kode Sumber 13). Poin poin dari citra disimpan dalam variable `pointsA` dan `pointsB` yang kemudian diperiksa posisi mana yang paling banyak mendapat poin-poin yang sama antar kedua citra agar posisi tersebut disimpan sebagai posisi paling sesuai antar kedua citra dengan fungsi `Compute_Homography` (Kode Sumber 16).

```

1. def matchKeypoints(self, KeypointsA, KeypointsB, fe
aturesA, featuresB,lowe_ratio, max_Threshold):
2.     AllMatches = self.get_Allpossible_Match(feature
sA,featuresB);
3.     valid_matches = self.All_validmatches(AllMatche
s,lowe_ratio)
4.     if len(valid_matches) > 4:
5.         pointsA = np.float32([KeypointsA[i] for (_,
i) in valid_matches])
6.         pointsB = np.float32([KeypointsB[i] for (i,
_) in valid_matches])
7.         (Homograppy, status) = self.Compute_Homogra
phy(pointsA, pointsB, max_Threshold)

```

Kode Sumber 13 Fungsi matchKeypoints

```

1. def get_Allpossible_Match(self, featuresA, featuresB)
   :
2.     match_instance = cv2.DescriptorMatcher_create("
   BruteForce")
3.     All_Matches = match_instance.knnMatch(featuresA
   , featuresB, 2)

```

Kode Sumber 14 Fungsi get_Allpossible_Match

```

1. def All_validmatches(self, AllMatches, Lowe_ratio):
2.     valid_matches = []
3.     for val in AllMatches:
4.         if len(val) == 2 and val[0].distance < val[
5. 1].distance * Lowe_ratio:
6.             valid_matches.append((val[0].trainIdx,
7. val[0].queryIdx))

```

Kode Sumber 15 Fungsi All_validmatches

```

1. def Compute_Homography(self, pointsA, pointsB, max_Thr
   eshold):
2.     (H, status) = cv2.findHomography(pointsA, point
   sB, cv2.RANSAC, max_Threshold)

```

Kode Sumber 16 Fungsi Compute_Homography

Setelah melalui proses perhitungan homografi tersebut, maka posisi citra diubah menjadi sesuai dengan posisi dengan poin-poin sama terbanyak yang telah diperoleh dari perhitungan homografi (Kode Sumber 17), dimana posisi ini akan membentuk dua citra yang saling timpa, dimana poin-poin yang sama saling berdekatan atau bersentuhan. Sehingga diperoleh satu citra yang merupakan pergabungan dari dua citra masukkan.

```

1. def getwarp_perspective(self, imageA, imageB, Homograp
   hy):
2.     val = imageA.shape[1] + imageB.shape[1]

```

```
3.     result_image = cv2.warpPerspective(imageA, Homo
      graphy, (val , imageA.shape[0]))
```

Kode Sumber 17 Fungsi `getwarp_perspective`

4.4 Implementasi Tampilan Antarmuka

Pada subbab ini akan dijelaskan mengenai implementasi dari perancangan antarmuka pengguna yang sudah dijelaskan pada subbab 3.2.2.

4.4.1 Tampilan Antarmuka Masukkan Data Jalan

The screenshot shows a web application window titled "Road Damage Mapping App". The interface includes a menu bar with "File", "Edit", "View", "Window", and "Help". The main content area features a form titled "Road Status" with the following fields:

- Road Name ***: A text input field.
- Image Note***: A text input field.
- Position ***: A text input field.
- Status**: A dropdown menu currently set to "Good".
- Choose Files**: A button next to the text "No file chosen".
- Submit**: A button at the bottom of the form.

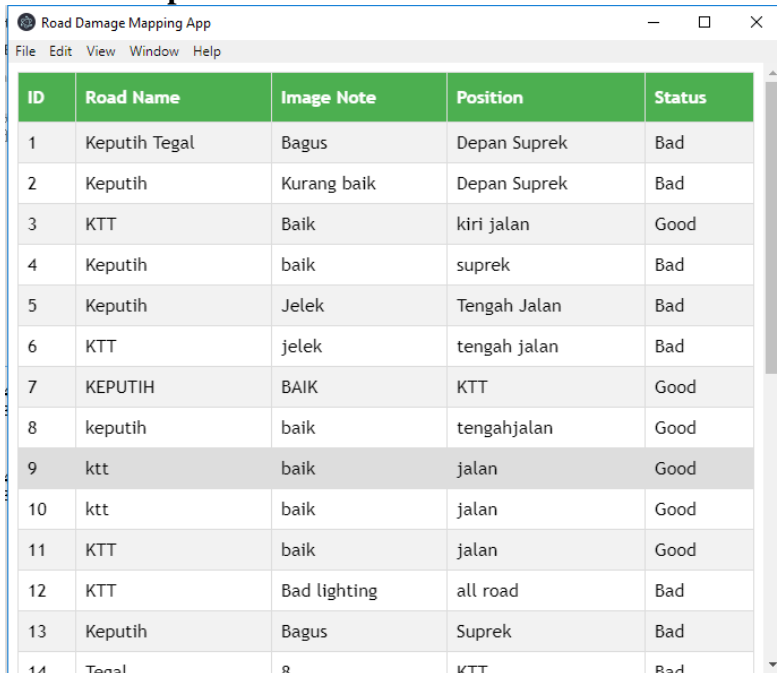
Below the main form, there are two buttons labeled "index" and "status".

Gambar 4.1 Tampilan Antarmuka Masukkan Data Jalan

Tampilan antarmuka masukkan data jalan menyesuaikan dengan rancangan yang telah dibuat pada subbab 3.2.2.1. Halaman masukkan data jalan memiliki sebuah formulir untuk mengisi data jalan, data form yang harus diisi adalah nama jalan (*Road name*), Keterangan citra (*Image note*), status jalan (*Status*), serta data citra. Untuk memasukkan data citra pengguna harus mengklik

tombol *Choose Files* yang akan membuka *File Explorer* untuk memasukkan data citra. Setelah memilih data citra dan memasukkan data jalan, maka pengguna dapat memasukkan data jalan ke dalam sistem dengan mengklik tombol *Submit*. Tampilan antarmuka dapat dilihat pada Gambar 4.1

4.4.2 Tampilan Antarmuka Lihat Status Jalan



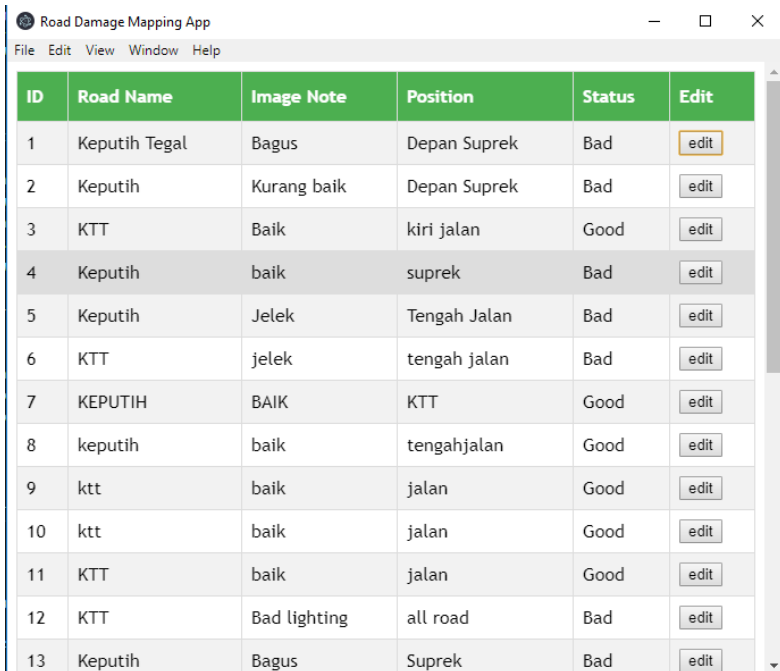
The screenshot shows a window titled "Road Damage Mapping App" with a menu bar (File, Edit, View, Window, Help) and a table with the following data:

ID	Road Name	Image Note	Position	Status
1	Keputih Tegal	Bagus	Depan Suprek	Bad
2	Keputih	Kurang baik	Depan Suprek	Bad
3	KTT	Baik	kiri jalan	Good
4	Keputih	baik	suprek	Bad
5	Keputih	Jelek	Tengah Jalan	Bad
6	KTT	jelek	tengah jalan	Bad
7	KEPUTIH	BAIK	KTT	Good
8	keputih	baik	tengahjalan	Good
9	ktt	baik	jalan	Good
10	ktt	baik	jalan	Good
11	KTT	baik	jalan	Good
12	KTT	Bad lighting	all road	Bad
13	Keputih	Bagus	Suprek	Bad
14	Tegal	8	KTT	Bad

Gambar 4.2 Tampilan Antarmuka Lihat Status Jalan (Surveyor)

Tampilan antarmuka masukkan data jalan menyesuaikan dengan rancangan yang telah dibuat pada subbab 3.2.2.2. Halaman lihat status jalan menampilkan sebuah tabel dengan data ID yang merupakan nomor unik untuk status jalan yang ada pada baris tersebut, nama jalan (*Road name*), Keterangan citra (*Image note*), status jalan (*Status*). Tampilan antarmuka untuk surveyor dapat dilihat pada Gambar 4.2

Karena kebutuhan untuk surveyor dan perencana sedikit berbeda, maka tampilannya pun sedikit berbeda. Untuk tampilan dengan pengguna perencana ditambahkan sebuah kolom untuk memasukkan tombol edit. Tampilan antarmuka untuk perencana dapat dilihat pada Gambar 4.3.



ID	Road Name	Image Note	Position	Status	Edit
1	Keputih Tegal	Bagus	Depan Suprek	Bad	<input type="button" value="edit"/>
2	Keputih	Kurang baik	Depan Suprek	Bad	<input type="button" value="edit"/>
3	KTT	Baik	kiri jalan	Good	<input type="button" value="edit"/>
4	Keputih	baik	suprek	Bad	<input type="button" value="edit"/>
5	Keputih	Jelek	Tengah Jalan	Bad	<input type="button" value="edit"/>
6	KTT	jelek	tengah jalan	Bad	<input type="button" value="edit"/>
7	KEPUTIH	BAIK	KTT	Good	<input type="button" value="edit"/>
8	keputih	baik	tengahjalan	Good	<input type="button" value="edit"/>
9	ktt	baik	jalan	Good	<input type="button" value="edit"/>
10	ktt	baik	jalan	Good	<input type="button" value="edit"/>
11	KTT	baik	jalan	Good	<input type="button" value="edit"/>
12	KTT	Bad lighting	all road	Bad	<input type="button" value="edit"/>
13	Keputih	Bagus	Suprek	Bad	<input type="button" value="edit"/>

Gambar 4.3 Tampilan Antarmuka Lihat Status Jalan (Perencana)

4.4.3 Tampilan Antarmuka Perbarui Data Jalan

Tampilan antarmuka perbarui data jalan menyesuaikan dengan rancangan yang telah dibuat pada subbab 3.2.2.3, Dimana tampilan untuk halaman perbarui data jalan serupa dengan halaman masukkan data jalan. Hanya saja pada halaman ini data jalan sudah tersedia, karena data sudah diambil pada halaman lihat status jalan. Tampilan antarmuka dapat dilihat pada Gambar 4.4.

The screenshot displays a web application window titled "Road Damage Mapping App". The window has a standard menu bar with "File", "Edit", "View", "Window", and "Help". The main content area features a light pink form titled "Road Status". The form contains the following fields:

- Road Name ***: A text input field containing "Keputih".
- Image Note***: A text input field containing "Kurang baik".
- Position ***: A text input field containing "Depan Suprek".
- Status**: A dropdown menu with "Bad" selected.
- Choose Files**: A button next to the text "No file chosen".
- Submit**: A button at the bottom of the form.

Below the main form, there are two additional buttons: "index" and "status".

Gambar 4.4 Tampilan Antarmuka Perbaruis Data Jalan

[Halaman ini sengaja dikosongkan]

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas tentang pengujian dan evaluasi pada perangkat lunak yang dibangun untuk Tugas Akhir ini. Pengujian dilakukan pada kasus penggunaan dari sistem perangkat lunak.

5.1 Lingkungan Pengujian

Pada proses pengujian perangkat lunak, dibutuhkan suatu lingkungan pengujian yang sesuai dengan standar kebutuhan. Lingkungan pengujian dalam Tugas Akhir ini dilakukan pada setiap kasus penggunaan. Spesifikasi masing-masing lingkungan pengujian dijabarkan pada Tabel 5.1.

Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak

Perangkat	Spesifikasi
Drone	Phantom 3 Professional
Komputer	Prosesor: Intel® Core™ i5-4460 CPU @ 3.20GHz Memori: 8192MB RAM
Perangkat Lunak	Sistem Operasi: Windows 10 Home 64-bit Perangkat Pengembang: Node.js, Electron, Python Basis Data: MySQL

5.2 Data Citra Jalan

Pada proses pengujian perangkat lunak, pada bagian menjalankan modul untuk mengolah data citra dibutuhkan suatu Data Citra Jalan yang sesuai standar untuk dapat diproses dengan baik dan menghasilkan keluaran yang sesuai dengan sebagaimana mestinya.

Pada Tugas Akhir ini proses pengambilan data citra menggunakan *drone* dengan spesifikasi yang dijelaskan pada Tabel 5.1 dengan target yang berupa jalan aspal. Citra jalan yang diperoleh akan dipergunakan dalam modul *Stretching Image* dan

Stitching Image sehingga untuk setiap percobaan diperlukan lebih dari satu citra ideal, yang artinya dapat diproses dengan baik seperti pada Gambar 5.8 dan Gambar 5.9.



Gambar 5.1 Citra dengan pencahayaan yang tidak terkontrol



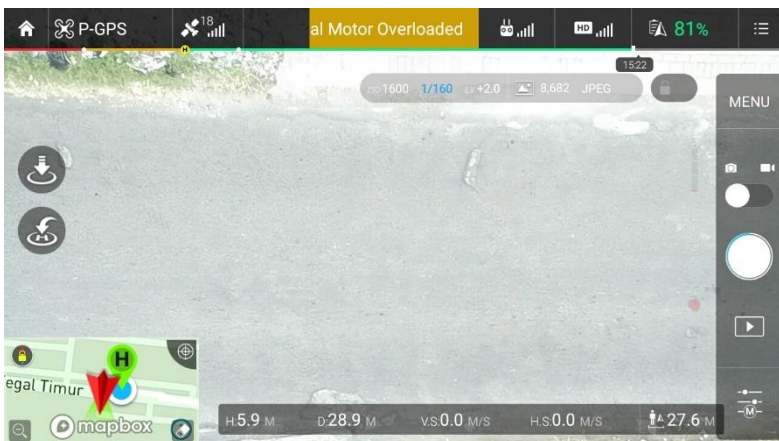
Gambar 5.2 Citra dengan gangguan objek

Saat pengambilan data berlangsung di lapangan, ternyata ditemukan beberapa kendala yang tidak diperhitungkan dari awal, seperti adanya objek pengganggu seperti pada Gambar 5.2, pencahayaan yang tidak dapat dikontrol seperti pada Gambar 5.1,

posisi drone yang tidak dapat tetap sejajar dengan jalan, ketinggian dan jarak drone dari pengguna yang menggunakan GPS (*Global Positioning System*) tidak menunjukkan posisi yang akurat seperti pada Gambar 5.3 dan Gambar 5.4. Berikut beberapa contoh citra yang diperoleh:



Gambar 5.3 Citra 1 tidak sesuai dengan jarak pada GPS



Gambar 5.4 Citra 2 tidak sesuai dengan jarak pada GPS



Gambar 5.5 Ketinggian 6 meter



Gambar 5.6 Ketinggian 7 meter

Dalam pengambilan data, salah satu parameter yang cukup penting adalah ketinggian pengambilan citra. Ketinggian akan berdampak terhadap hasil citra yang diperoleh, dalam pengujian dicoba dengan 3 ketinggian, yaitu 6m, 7m, dan 10m. Pada ketinggian 6m, tidak seluruh bagian jalan dapat terlihat dalam citra. Pada ketinggian 7m seluruh bagian jalan terlihat dalam citra. Pada ketinggian 10m seluruh bagian jalan terlihat dalam citra, tetapi

terlalu banyak tepian jalan yang ikut terlihat dalam citra, sehingga dianggap kurang ideal.



Gambar 5.7 Ketinggian 10 meter



Gambar 5.8 Citra 1 ideal

Pada saat pengambilan data, cukup sulit untuk memperoleh data ideal yang beruntun agar dapat digunakan untuk data uji agar dapat menghasilkan keluaran yang sesuai dengan harapan, seperti

pada Gambar 5.8 dan Gambar 5.9 yang diperoleh dengan gangguan seminimal mungkin secara beruntun.



Gambar 5.9 Citra 2 ideal

Sehingga diperoleh kesimpulan untuk data yang digunakan memiliki standar minimal untuk dapat digunakan sebagai data citra masukkan aplikasi, yaitu:

1. Ketinggian pada saat pengambilan citra adalah 7 meter.
2. Posisi kamera tegak lurus (90 derajat) dari jalan
3. Pencahayaan normal dan konsisten untuk setiap citra (tidak terlalu gelap dan tidak terlalu terang)
4. Citra yang akan digabungkan harus beruntun (memiliki irisan 50%), serta untuk setiap data memiliki jarak yang sama untuk antar pengambilan data citra
5. Tidak ada objek besar pada jalan

5.3 Pengujian Fungsionalitas

Pengujian akan dilakukan menggunakan sekumpulan data jalan, dengan menguji fungsionalitas dari aplikasi yang ada. Data jalan yang digunakan berbeda beda, sesuai dengan ketersediaan data dan sesesuai mungkin dengan standar data yang ada

5.3.1 Pengujian Memasukkan Data Jalan

Tabel 5.2 Pengujian Memasukkan Data Jalan

No. Pengujian	UC-01
Referensi Kasus Penggunaan	UC-01
Nama	Memasukkan data jalan
Tujuan Pengujian	Menguji fungsionalitas memasukkan data jalan kedalam sistem serta menjalankan modul yang ada untuk mengeluarkan keluaran yang diharapkan
Kondisi Awal	Surveyor pada menu utama
Data Uji	Status jalan dan data citra
Langkah Pengujian	<ol style="list-style-type: none"> 1. Surveyor mengklik tombol untuk memasukkan data jalan 2. Surveyor mengisi status jalan dan memasukkan data citra 3. Surveyor mengklik “Submit”
Hasil yang diharapkan	Sistem dapat menyimpan status jalan dan menjalankan modul dengan data citra serta menyimpan keluaran dari modul tersebut
Hasil yang didapat	Status jalan yang dimasukkan tersimpan di basis data dan keluaran dari modul tersimpan pada folder yang ditentukan pada aplikasi.
Hasil Pengujian	Berhasil
Kondisi Akhir	Status jalan tersimpan, keluaran dari modul tersimpan

Pengujian memasukkan data jalan merupakan pengujian terhadap kemampuan sistem (aplikasi) untuk mengunggah data, serta menjalankan modul *Stretching Image* dan *Stitching Image* dengan citra jalan yang diperoleh. Pengujian ini dilakukan ketika pengajar menekan tombol “Submit” pada *form* dari halaman masukkan data jalan pada Aplikasi Pemetaan Kerusakan Jalan.

Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.2.

5.3.2 Pengujian Melihat Status Jalan

Pengujian melihat status jalan merupakan pengujian terhadap kemampuan sistem (aplikasi) untuk menampilkan kumpulan status jalan yang ada dalam sistem. Pengujian ini dilakukan ketika surveyor atau perencana memilih menu lihat status jalan. Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.3

Tabel 5.3 Pengujian Melihat Status Jalan

No. Pengujian	UC-02
Referensi Kasus Penggunaan	UC-02
Nama	Pengujian melihat status jalan
Tujuan Pengujian	Menguji fungsionalitas untuk melihat status jalan
Kondisi Awal	Surveyor atau Perencana berada pada menu utama
Data Uji	Status jalan
Langkah Pengujian	Surveyor atau Perencana mengklik tombol untuk melihat status jalan
Hasil yang diharapkan	Sistem dapat menampilkan kumpulan data status jalan
Hasil yang didapat	Sistem menampilkan kumpulan data status jalan.
Hasil Pengujian	Berhasil
Kondisi Akhir	Kumpulan status jalan berhasil ditampilkan

5.3.3 Pengujian Memperbarui Data Jalan

Pengujian memperbarui data jalan merupakan pengujian terhadap kemampuan sistem (aplikasi) memperbarui data yang sudah pernah ada didalam sistem. Pengujian ini dilakukan ketika perencana menekan tombol “Edit” pada halaman lihat status jalan

di baris yang statusnya akan diperbarui. Rincian skenario pengujian pada kasus penggunaan ini ditunjukkan pada Tabel 5.4.

Tabel 5.4 Pengujian Memperbarui Data Jalan

No. Pengujian	UC-02a
Referensi Kasus Penggunaan	UC-02a
Nama	Pengujian memperbarui data jalan
Tujuan Pengujian	Menguji fungsionalitas untuk memperbarui data jalan
Kondisi Awal	Perencana berada pada halaman melihat status jalan
Data Uji	Status jalan dan data citra
Langkah Pengujian	<ol style="list-style-type: none"> 1. Perencana mengklik tombol “Edit” pada baris yang akan diperbarui 2. Perencana mengisi form dan memasukkan data citra 3. Perencana mengklik tombol “Submit”
Hasil yang diharapkan	Sistem dapat memperbarui data jalan yang sudah pernah ada
Hasil yang didapat	Data jalan yang sudah pernah ada diperbarui
Hasil Pengujian	Berhasil
Kondisi Akhir	Berhasil memperbarui data jalan yang sudah dipilih

5.4 Pengujian Modul

Pengujian modul, yaitu modul *Image Stretching* dan *Image Stitching* dilakukan dengan spesifikasi komputer yang tertera pada subbab 4.1.

5.4.1 Pengujian Modul Sebelum dan Sesudah *Image Stretching* dengan 2 Data Citra masukkan

Pada subbab ini dilakukan pengujian modul dengan dua data masukkan yang ditunjukkan pada LAMPIRAN D, Gambar D.1 sebagai citra 1, dan Gambar D.2 sebagai citra 2.



Gambar 5.10 *Image Stitching* Tanpa *Image Stretching*



Gambar 5.11 *Image Stitching* Dengan *Image Stretching*

Gambar 5.10 merupakan hasil dari *Image Stitching* Tanpa *Image Stretching* dan Gambar 5.11 menunjukkan hasil *Image Stitching* Dengan *Image Stretching*. Proses penyatuan citra berhasil dilakukan dengan baik, namun hasil dari *Image Stretching* belum menunjukkan hasil yang cukup bermanfaat.

5.4.2 Pengujian Modul Sebelum dan Sesudah *Image Stretching* dengan 3 Data Citra masukkan

Pada subbab ini dilakukan pengujian modul dengan tiga data masukkan yang ditunjukkan pada LAMPIRAN D, Gambar D.5 sebagai citra 1, Gambar D.6 sebagai citra 2, dan Gambar D.7 sebagai citra 3.



Gambar 5.12 *Image Stitching* Tanpa *Image Stretching*



Gambar 5.13 *Image Stitching* Dengan *Image Stretching*

Gambar 5.12 merupakan hasil dari *Image Stitching* Tanpa *Image Stretching* dan Gambar 5.13 menunjukkan hasil *Image Stitching* Dengan *Image Stretching*. Proses penyatuan citra berhasil dilakukan dengan baik, serta hasil dari *Image Stretching* sudah mulai menunjukkan hasil yang cukup terlihat, namun belum sepenuhnya bermanfaat.

5.4.3 Pengujian Modul Sebelum dan Sesudah *Image Stretching* dengan 4 Data Citra masukkan

Pada subbab ini dilakukan pengujian modul dengan empat data masukkan yang ditunjukkan pada LAMPIRAN D, Gambar D.9 sebagai citra 1, Gambar D.10 sebagai citra 2, Gambar D.11 sebagai citra 3, dan Gambar D.12 sebagai citra 4.

Namun pada saat pengujian modul, proses Image Stitching tidak berhasil, diakarenakan subproses modul yang melakukan proses Image Stitching terhenti. Hal ini disebabkan oleh terbatasnya jumlah pemakaian CPU dan memori pada subproses yang dijalankan dari dalam aplikasi. Dari kegagalan pengujian ini, dengan asumsi data citra masukkan berukuran terlalu besar, maka dilakukan pengujian dengan mengubah ukuran data citra

5.4.4 Pengujian Modul dengan Mengubah Ukuran Data Citra

Pada subbab ini dilakukan pengujian modul dengan berbagai ukuran data citra, dikarenakan terjadi kegagalan pada pengujian data citra dengan ukuran data normal. Jumlah citra menunjukkan jumlah data citra masukkan untuk pengujian modul. Data masukkan menunjukkan citra yang digunakan untuk data masukkan yang ditunjukkan pada LAMPIRAN D. Status menunjukkan apakah pengujian berhasil atau tidak dilakukan, dilengkapi dengan keterangan tambahan. Waktu menunjukkan berapa lama proses berlangsung.

Tabel 5.5 Ukuran Data Normal

No	Jumlah Citra	Data Masukkan	Status	Waktu
1.	2	1, 2	Berhasil	45.89s
2.	3	5, 6, 7	Berhasil	144.55s
3.	4	13, 14, 15, 16	Gagal	-

Tabel 5.6 Ukuran Data 75% dari Data Normal

No	Jumlah Citra	Data Masukkan	Status	Waktu
1.	2	1, 2	Berhasil	10.85s
2.	3	5, 6, 7	Berhasil, kurang baik	31.53s
3.	4	13, 14, 15, 16	Berhasil	217.77s
4.	5	3, 4, 5, 6, 7	Berhasil, kurang baik	97.43s
5.	5	9, 10, 11, 12, 13	Berhasil, tidak dapat digunakan	194.07s
6.	5	1, 2, 3, 4, 5	Berhasil	97.178s
7.	5	12, 13, 14, 15, 16	Berhasil, tidak dapat digunakan	292.79s
8.	6	1, 2, 3, 4, 5, 6	Berhasil, kurang baik	129.56s
9.	6	7, 8, 9, 10, 11, 12	Berhasil, tidak dapat digunakan	161.17s

Tabel 5.7 Ukuran Data 50% dari Data Normal

No	Jumlah Citra	Data Masukkan	Status	Waktu
1.	2	1, 2	Berhasil	10.85s
2.	3	5, 6, 7	Berhasil, kurang baik	31.53s
3.	4	13, 14, 15, 16	Berhasil, kurang baik	217.77s
4.	5	3, 4, 5, 6, 7	Berhasil, kurang baik	97.43s
5.	5	9, 10, 11, 12, 13	Berhasil, tidak dapat digunakan	194.07s
6.	5	1, 2, 3, 4, 5	Berhasil	97.178s
7.	5	12, 13, 14, 15, 16	Berhasil, kurang baik	292.79s
8.	6	1, 2, 3, 4, 5, 6	Berhasil	129.56s
9.	6	7, 8, 9, 10, 11, 12	Berhasil, tidak dapat digunakan	161.17s

Dari pengujian yang dilakukan, diperoleh data yang ditunjukkan pada Tabel 5.5, Tabel 5.6, Tabel 5.7. Diperoleh kesimpulan bahwa pengujian data normal gagal karena ukuran data yang terlalu besar, karena pada saat ukuran data diperkecil, pengujian berhasil dilakukan. Namun ketika data diperkecil, diperoleh hasil yang cenderung semakin tidak baik.

5.5 Evaluasi Pengujian Fungsionalitas

Berdasarkan pengujian yang telah dilakukan, semua pengujian fungsionalitas memberikan hasil yang sesuai dengan skenario yang direncanakan. Rangkuman mengenai hasil pengujian fungsionalitas yang diujicobakan ditunjukkan pada Tabel 5.8.

Tabel 5.8 Evaluasi Pengujian Fungsionalitas

No	Fungsionalitas	Terpenuhi
1.	Memasukkan data jalan	✓
2.	Melihat Status Jalan	✓
3.	Memperbarui Status Jalan	✓

Berdasarkan data pada Tabel 5.8, seluruh skenario pengujian berhasil dilakukan. Sehingga dapat disimpulkan bahwa fungsionalitas dari sistem (aplikasi) bekerja sesuai dengan yang diharapkan.

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1 Kesimpulan

Kesimpulan dari Tugas Akhir ini adalah tujuan dari Aplikasi Pemetaan Kerusakan jalan untuk memetakan jalan yang dapat digunakan untuk memetakan kerusakan jalan telah berhasil dibangun dengan solusi dari beberapa rumusan masalah sebagai berikut:

1. Implementasi modul *Image Stretching* pada Aplikasi Kerusakan Jalan berhasil dilakukan. Modul berhasil mengubah dimensi citra yang diperoleh menjadi citra yang lebih menyerupai dimensi kenyataannya.
2. Implementasi modul *Image Stitching* pada Aplikasi Pemetaan Kerusakan Jalan berhasil dilakukan. Modul berhasil menyatukan beberapa citra yang sudah diproses melalui modul *Image Stretching* menjadi satu image yang utuh.
3. Data yang dapat digunakan untuk Aplikasi Pemetaan Kerusakan Jalan adalah ketinggian drone pada saat pengambilan citra adalah 7 meter. Posisi kamera tegak lurus (90 derajat) dari jalan. Pencahayaan normal dan konsisten untuk setiap citra (tidak terlalu gelap dan tidak terlalu terang). Citra yang akan digabungkan harus beruntun (memiliki irisan 50%), serta untuk setiap data memiliki jarak yang sama untuk antar pengambilan data citra serta tidak ada objek besar pada jalan.
4. Semakin besar ukuran data citra maka hasil penggabungan citra akan semakin baik, walaupun akan memakan waktu lebih banyak dan membutuhkan spesifikasi komputer yang lebih

tinggi, sebaliknya semakin kecil ukuran data citra maka hasil penggabungan akan semakin buruk, namun akan memakan waktu lebih singkat dan membutuhkan spesifikasi komputer yang lebih rendah.

6.2 Saran

Saran untuk pengembangan Aplikasi Pemetaan Kerusakan Jalan di masa yang akan datang adalah sebagai berikut:

1. Perlu ditentukan standar terhadap metode pengambilan data citra, untuk mengurangi jumlah kegagalan penggunaan data citra
2. Diperlukan spesifikasi computer yang lebih memadai untuk menjalankan aplikasi serta modulnya.

DAFTAR PUSTAKA

- [1] “Badan Pusat Statistik,” Badan Pusat Statistik, [Online]. Available: <https://www.bps.go.id/linkTableDinamis/view/id/820>. [Diakses 9 January 2019].
- [2] “Badan Pusat Statistik,” Badan Pusat Statistik Provinsi Jawa Timur, 18 Oktober 2017. [Online]. Available: <https://jatim.bps.go.id/statictable/2017/10/18/690/panjang-jalan-menurut-kondisi-permukaan-jalan-km-2013---2016.html>. [Diakses 9 Januari 2019].
- [3] “Kompas.com,” PT Kompas Cyber Media, 26 April 2018. [Online]. Available: <https://properti.kompas.com/read/2018/04/26/220351821/anggaran-pemeliharaan-jalan-dan-jembatan-rp-237-triliun>. [Diakses 9 Januari 2019].
- [4] V. P. Patil dan U. B. Gohatre, “Performance Comparison of Image Stitching Methods under Different Illumination Conditions,” *International Journal of Engineering Technology Science and Research*, vol. 4, no. 9, pp. 1-11, 2017.
- [5] V. A. Putri, “Identifikasi Jenis Kerusakan Pada Perkerasan Lentur,” *Tugas Akhir Fakultas Teknik Universitas Bandar Lampung*, pp. 1 - 73, 2016.
- [6] V. B. C. M. Romana OANCEA, “PERSPECTIVE CORRECTION FOR IMAGE PROCESSING,” no. XXI, 2015.
- [7] W3C, “HTML,” 28 May 2019. [Online]. Available: <https://www.w3.org/html/>. [Diakses 7 Juli 2019].
- [8] D. Flanagan, *JavaScript: The Definitive Guide*, Sixth Edition, O’Reilly Media, Inc., 2011.
- [9] W3C, “Cascading Style Sheets,” [Online]. Available: <https://www.w3.org/Style/CSS/>. [Diakses 10 Juli 2019].

- [10] Github, “Electron,” [Online]. Available: <https://electronjs.org>. [Diakses 10 Juli 2019].
- [11] NODE.JS FOUNDATION, “Node.js,” [Online]. Available: <https://nodejs.org>. [Diakses 10 Juli 2019].
- [12] G. Van Rossum dan F. L. Drake Jr, Python tutorial, Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.
- [13] Secret Labs AB, “The Python Imaging Library (PIL),” [Online]. Available: <https://pillow.readthedocs.io/en/stable/>. [Diakses 10 Juli 2019].
- [14] T. E. Oliphant, A guide to NumPy, vol. 1, Trelgol Publishing USA, 2006.

LAMPIRAN A

Use Case Specification: Memasukkan Data Jalan

1. Use Case Name

1.1 Brief Description

Use Case ini mendeskripsikan bagaimana Surveyor memasukkan data jalan pada Aplikasi Pemetaan Kerusakan Jalan

2. Flow of Events

2.1 Basic Flow

1. Surveyor mengklik tombol untuk memasukkan data jalan
2. Sistem menampilkan halaman berisi form data jalan
3. Surveyor mengisi status jalan dan memasukkan data citra
4. Surveyor mengklik *submit*
5. Sistem menyimpan status jalan dan data citra

2.2 Alternative Flows

n.a.

2.3 Exceptions

2b. Menutup halaman yang ada pada form sebelum proses memasukkan data jalan selesai sepenuhnya. Kejadian ini berlangsung saat form data jalan telah terisi, kemudian Surveyor menutup halaman sebelum proses memasukkan data jalan selesai.

3. Spesial Requirements

1. Pengguna harus terotorisasi dan terotentikasi sebagai Surveyor

4. Pre-Conditions

n.a

5. Post-Conditions

n.a

6. Extension Points

n.a

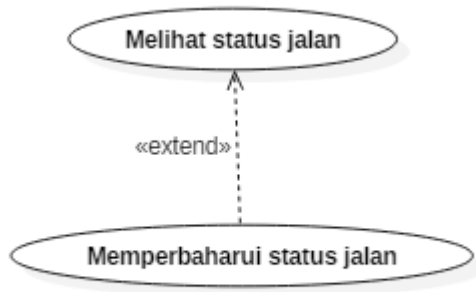
LAMPIRAN B

Use Case Specification: Memasukkan Data Jalan

1. Use Case Name

1.1 Brief Description

Use Case ini mendeskripsikan bagaimana Surveyor memasukkan status jalan pada Aplikasi Pemetaan Kerusakan Jalan



2. Flow of Events

2.1 Basic Flow

1. Surveyor mengklik tombol untuk melihat status jalan
2. Sistem menampilkan halaman berisi kumpulan data status jalan

2.2 Alternative Flows

n.a.

2.3 Exceptions

2b. 2a. Menutup halaman yang ada pada form sebelum proses memperbarui data jalan selesai sepenuhnya. Kejadian ini berlangsung saat form data jalan telah terisi, kemudian Surveyor menutup halaman sebelum proses pembaruan data jalan selesai.

3. Spesial Requirements

1. Pengguna harus terotorisasi dan terotentikasi sebagai Surveyor atau Perencana
2. a. Pengguna harus terotorisasi dan terotentikasi sebagai Perencana

4. Pre- Conditions

n.a

5. Post- Conditions

n.a

6. Extension Points

2a Perencana dapat **memperbaharui status jalan**

2a. 1. Perencana mengklik tombol pada data jalan yang akan diperbarui dari daftar data jalan

2a. 2. Sistem menampilkan data jalan pada form data jalan

2a. 3. Perencana memperbarui form data jalan, kemudian perencana mengklik tombol *submit*

2a. 4. Sistem memperbarui data jalan

LAMPIRAN C

Software Requirements Specification

1. Introduction

1.1 Purpose

Tujuan dari penulisan dokumen ini adalah memberikan penjelasan detail dari Aplikasi Pemetaan Kerusakan Jalan. Penjelasan yang akan dimaksudkan dalam dokumen ini berisi tujuan pembuatan dan fitur-fitur sistem, antarmuka sistem, apa yang dilakukan oleh sistem, batasan saat sistem harus beroperasi dan bagaimana sistem akan bereaksi terhadap stimulus dari luar sistem. Dokumen ini dimaksudkan untuk stakeholder maupun pengembang dari sistem.

1.2 Scope

Sistem perangkat lunak yang akan dibangun adalah Aplikasi Pemetaan Kerusakan Jalan. Sistem ini dirancang untuk memaksimalkan produktivitas dengan menyediakan alat untuk mempermudah proses pengambilan data citra jalan serta memetakan kerusakan jalan jika kerusakan ditemukan yang biasa harus dilakukan manual.

1.3 Definitions, Acronyms and abbreviations

n.a.

1.4 References

UC01 Use Case Specification Document “Memasukkan data barang”

UC02 Use Case Specification Document “Melihat status jalan”

1.5 Overview

Bagian selanjutnya dari dokumen ini diatur sebagai berikut. Bab kedua dari dokumen ini memberikan gambaran

mengenai fungsionalitas dari produk. Bab ini juga menjelaskan kebutuhan informal yang digunakan untuk menyusun konteks dari spesifikasi kebutuhan teknis bab selanjutnya

Bab ketiga, bagian Specific Requirement dari dokumen ini ditulis untuk pengembang dan dideskripsikan dengan bahasa teknis mengenai detail fungsionalitas dari produk.

2. Introduction

2.1 Scenario

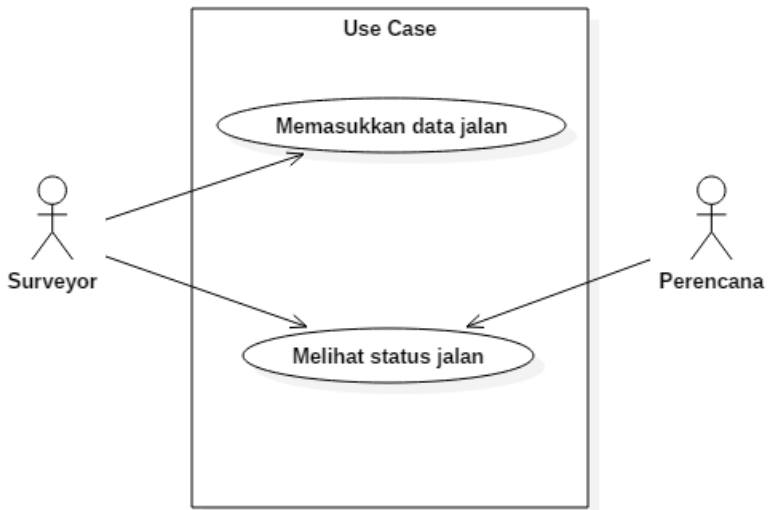
Udin adalah seorang surveyor dari Badan Perencanaan Pembangunan Kota Surabaya (BAPPEKO Surabaya) yang memiliki tugas melakukan survey mengenai keadaan jalan ke jalanan di Surabaya. Hari ini Udin mendapat panggilan tugas dari Andi, seorang perencana yang juga bekerja di BAPPEKO Surabaya, untuk melakukan survey ke Jalan Keputih, karena kemarin Andi melewati Jalan Keputih dan menyadari bahwa jalan tersebut cukup rusak dan membutuhkan perbaikan.

Pada siang hari, Udin berangkat ke Keputih dan menemukan banyak kerusakan pada jalan seperti kata Andi. Lalu Udin mengeluarkan dronanya untuk mengambil gambar jalan yang rusak. Setelah Udin mengambil gambar pada bagian jalan yang rusak dan sekitarnya menggunakan drone, Udin mampir ke Café di Keputih untuk istirahat dan juga mengunggah gambar tersebut ke Aplikasi Pemetaan Kerusakan Jalan. Udin membuka aplikasi, lalu mengklik menu “Masukkan data jalan”, lalu Udin mengisi form dan gambar, setelah selesai Udin mengklik tombol selesai. Karena sudah menyelesaikan tugasnya, Udin memberitahu Andi bahwa Ia sudah menyelesaikan tugasnya. Andi lalu membuka Aplikasi Kerusakan Pemetaan Jalan, dan membuka menu “Melihat status jalan”. Karena sudah mendapat data mengenai kerusakan jalan Andi lalu memberikan tugas pada pihak

perbaikan dari BAPEKKO untuk melaksanakan perbaikan sesuai data yang sudah ada di Aplikasi.

Setelah pihak perbaikan menyelesaikan perbaikan, mereka melaporkan bahwa tugas mereka sudah selesai kepada Andi, dengan mengirimkan gambar jalan yang sudah diperbaiki kepada Andi. Lalu andi ingin memperbarui status jalan tersebut, Andi memilih menu “Melihat Status Jalan” lalu memilih menu edit pada baris jalan yang akan diperbarui. Lalu andi mengisi form yang ada dan memasukkan gambar jalan yang baru, setelah selesai Andi mengklik tombol selesai. Esok harinya Andi mendapat apresiasi karena pekerjaan BAPPEKO yang relative cepat oleh salah satu keluarga Andi di daerah Keputih.

2.2 Context Diagram



ID	UC Name	UC Description
UC01	Memasukkan data jalan	Surveyor dapat mengunggah data citra dan status jalan ke dalam sistem

UC02	Melihat status jalan	Surveyor dan perencana dapat melihat status jalan di dalam sistem
------	----------------------	---

3. Specific Requirements

3.1 Functionality

ID	Statement	UC	Priority
F01	Surveyor dapat memasukkan data jalan	UC01	Must
F02	Surveyor dapat melihat status jalan	UC02	Must
F03	Perencana dapat melihat status jalan	UC02	Must
F04	Perencana dapat	UC02	Optional

3.2 Non-Functionality

ID	Statement	Quality	Priority
NF01	Bahasa yang digunakan pada antarmuka merupakan Bahasa Inggris	Operability	Must
NF02	Hanya user yang terotentikasi dan terotorisasi yang bisa menggunakan aplikasi, dengan hak akses sesuai dengan peran user	Integrity	Must
NF03	Data jalan terdiri atas beberapa informasi yaitu nomor pengenal data, nama jalan, keterangan data citra, posisi dan juga status kondisi jalan	Operability	Must
NF04	Sistem menyediakan pilihan minimize, maximize untuk window yang ada di sistem	Operability	Optional
NF05	Sistem akan disebarakan dalam bentuk aplikasi desktop	Portability	Must

LAMPIRAN D

Lampiran ini berisi data-data citra yang merupakan hasil pengambilan data, yang akan digunakan dalam proses pengujian modul *Image Stretching* dan *Image Stitching*.



Gambar D.1 Hasil Pengambilan Data Citra 1



Gambar D.2 Hasil Pengambilan Data Citra 2



Gambar D.3 Hasil Pengambilan Data Citra 3



Gambar D.4 Hasil Pengambilan Data Citra 4



Gambar D.5 Hasil Pengambilan Data Citra 5



Gambar D.6 Hasil Pengambilan Data Citra 6



Gambar D.7 Hasil Pengambilan Data Citra 7



Gambar D.8 Hasil Pengambilan Data Citra 8



Gambar D.9 Hasil Pengambilan Data Citra 9



Gambar D.10 Hasil Pengambilan Data Citra 10



Gambar D.11 Hasil Pengambilan Data Citra 11



Gambar D.12 Hasil Pengambilan Data Citra 12



Gambar D.13 Hasil Pengambilan Data Citra 13



Gambar D.14 Hasil Pengambilan Data Citra 14

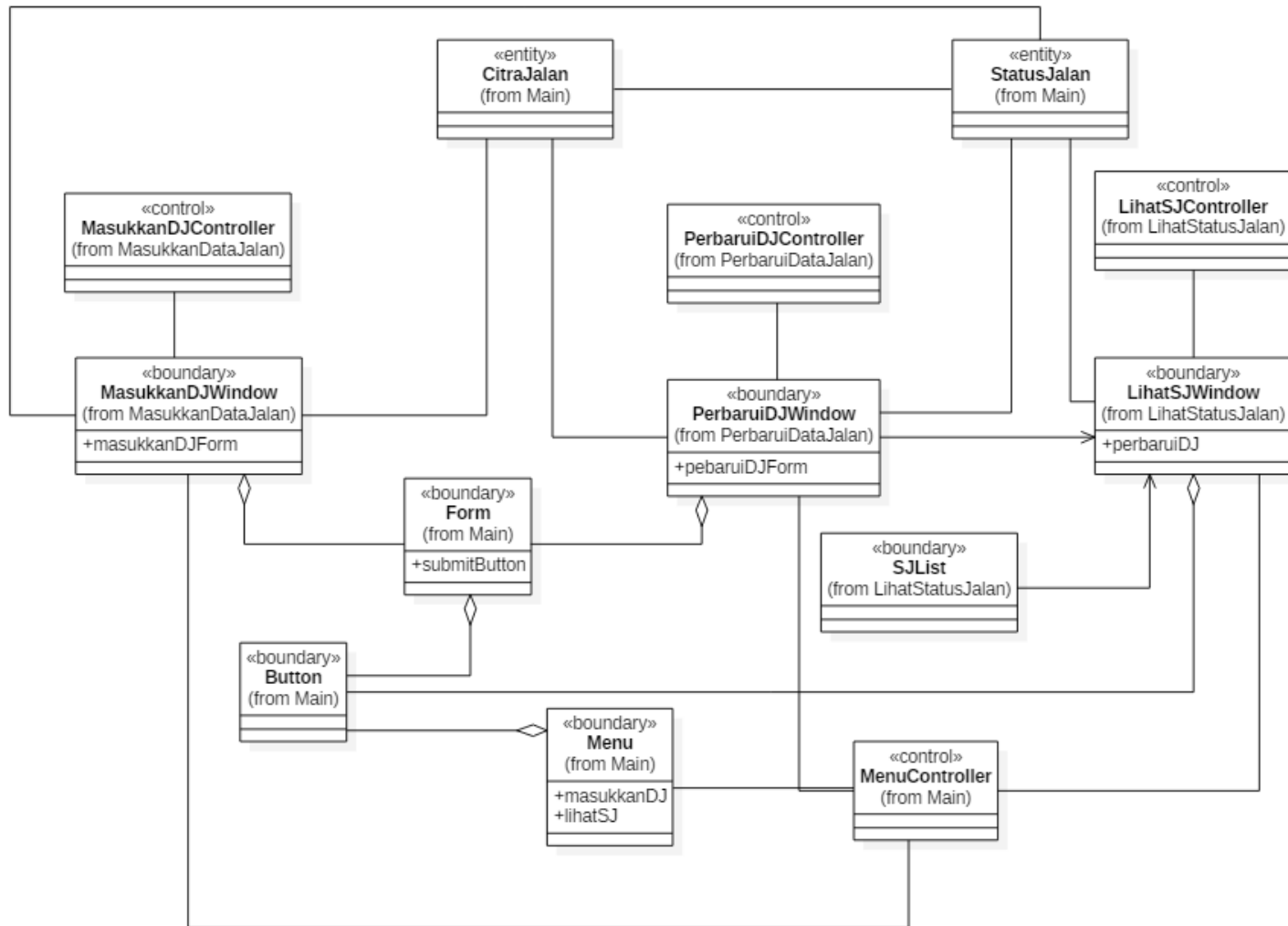


Gambar D.15 Hasil Pengambilan Data Citra 15



Gambar D.16 Hasil Pengambilan Data Citra 16

LAMPIRAN E



Gambar E.1 Diagram Kelas

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Ivan Agung Pandapotan, anak kedua dari tiga bersaudara yang lahir di Bandung pada tanggal 28 Maret 1997. Penulis telah menempuh pendidikan formal mulai dari SD ST. Antonius Medan, SMP Sutomo 1 Medan, SMA ST. Thomas 1 Medan dan terakhir sebagai mahasiswa Departemen Informatika Institut Teknologi Sepuluh Nopember dengan rumpun mata kuliah

Rekayasa Perangkat Lunak (2015-2019). Lulus dari SMA, penulis melanjutkan pendidikan di Departemen Informatika.

Selama perkuliahan, penulis aktif dalam organisasi kemahasiswaan, antara lain sebagai Pengurus PMK ITS divisi Natal dan Paskah 2017 – 2019, Staff Departemen Pengembangan Profesi Himpunan Mahasiswa Teknik Computer-Informatika ITS 2016-2018, dan Staff Humas Schematics 2016 dan 2017.

Selama kuliah di Departemen Informatika ITS, penulis mengambil bidang minat Rekayasa Perangkat Lunak (RPL) dengan ketertarikan penulis terdapat pada rancang bangun perangkat lunak. Penulis dapat dihubungi melalui surel ivanagungpasaribu@gmail.com.