



TUGAS AKHIR - KI091391

**RANCANG BANGUN APLIKASI SIG SEBAGAI
SISTEM PERINGATAN DINI UNTUK MITIGASI
RISIKO BENCANA BANJIR DENGAN
PENGINGAT BERUPA SMS**

ANDREYAN RIZKY BASKARA
NRP 5109 100 036

Dosen Pembimbing
Umi Laili Yuhana, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI091391

DESIGN AND IMPLEMENTATION OF GIS APPLICATION AS EARLY WARNING SYSTEM FOR MITIGATION OF FLOOD DISASTER RISK USING SMS AS REMINDER

ANDREYAN RIZKY BASKARA
NRP 5109 100 036

Supervisor
Umi Laili Yuhana, S.Kom., M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2014

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN APLIKASI SIG SEBAGAI SISTEM PERINGATAN DINI UNTUK MITIGASI RISIKO BENCANA BANJIR DENGAN PENGINGAT BERUPA SMS

Nama Mahasiswa : Andreyan Rizky Baskara
NRP : 5109 100 036
Jurusan : Teknik Informatika FTIf – ITS
Dosen Pembimbing I : Umi Laili Yuhana, S.Kom., M.Sc.

ABSTRAK

Banjir merupakan bencana alam musiman yang kerap melanda Indonesia. Secara umum banjir adalah peristiwa dimana daratan yang biasanya kering (bukan daerah rawa) menjadi tergenang oleh air, hal ini disebabkan oleh curah hujan yang tinggi dan kondisi topografi wilayah yang rendah hingga cekungan. Salah satu penyebab terjadinya banjir adalah air sungai yang meluap. Biasanya, orang-orang langsung mendatangi lokasi profil sungai untuk mengetahui bagaimana kondisi aliran di profil sungai tersebut dan melakukan pengukuran. Pada saat ini, berbagai sensor pun dikembangkan agar bisa digunakan untuk memantau kondisi aliran sungai secara otomatis. Kemudian diperlukan juga sebuah perangkat lunak yang dapat mengintegrasikan sensor dan perangkat keras seperti PC (Personal Computer) untuk menampilkan data hasil pantauan sensor. Pada Tugas Akhir ini, dibangun sebuah aplikasi SIG (Sistem Informasi Geografis) yang digunakan untuk memantau data hasil tangkapan sensor yang dipasang pada suatu profil sungai.

Aplikasi SIG yang dibangun berguna untuk melihat lokasi pantauan profil sungai pada peta dan menampilkan informasi-informasi hasil tangkapan sensor yang berkaitan dengan profil sungai tersebut. Aplikasi SIG ini dapat memitigasi risiko sebelum terjadi bencana banjir dengan mengolah data sensor yang didapat sehingga mampu memberikan peringatan dini ke pengguna.

Kata kunci: *Banjir, Peringatan Dini, Sistem Informasi Geografis, SMS*

[Halaman ini sengaja dikosongkan]

DESIGN AND IMPLEMENTATION OF GIS APPLICATION AS EARLY WARNING SYSTEM FOR MITIGATION OF FLOOD DISASTER RISK USING SMS AS REMINDER

Student Name : Andreyan Rizky Baskara
NRP : 5109 100 036
Major : Informatics FTIf - ITS
Supervisor I : Umi Laili Yuhana, S.Kom., M.Sc.

ABSTRACT

Flooding is a seasonal natural disasters that often hit Indonesia. In general, flooding is an event where the land that is usually dry (not swampy areas) be inundated by water, this is caused by high rainfall and low topography areas to the basin. One of the causes of the flooding is the overflowing of water in the river. People usually come straight to river cross section location to find out about the flow conditions in there and take measurements. At this time, various sensors were developed to be used to monitor the river flow conditions automatically. And also needed a software that can integrate sensors and hardware such as a PC (Personal Computer) to display sensors data from monitoring. In this final project, will be built a GIS application (Geographic Information System) which is used to monitor the data catches from sensors mounted on a cross section of the river.

GIS applications that built are useful to see the river cross section monitoring locations on a map and displays the information of the caught sensor data associated with the river cross section. This GIS application can mitigate the risk before flood disaster happen with processing the sensor data obtained to provide early warning to the user.

Keywords: Early Warning, Flood, Geographic Information System, SMS

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**RANCANG BANGUN APLIKASI SIG SEBAGAI SISTEM
PERINGATAN DINI UNTUK MITIGASI RISIKO
BENCANA BANJIR DENGAN PENGINGAT BERUPA SMS**

TUGAS AKHIR

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

Oleh:

ANDREYAN RIZKY BASKARA
NRP 5109 100 036

Disetujui oleh Pembimbing Tugas Akhir:

Umi Laili Yuhana, S.Kom., M.Sc.
NIP197906262005012002



(Pembimbing I)

SURABAYA
JULI 2014

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas berkah, rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini. Shalawat serta salam tak lupa penulis haturkan kepada Rasulullah Muhammad SAW atas segala tuntunan dalam menjalani kehidupan ini.

Penulis ingin menyampaikan penghormatan dan rasa terima kasih sedalam-dalamnya kepada pihak-pihak yang telah membantu dalam menyelesaikan Tugas Akhir ini, terutama kepada:

1. Bapak H. Basuki MD dan Ibu Hj. Rahmi Widyanti, selaku orang tua penulis yang telah mendidik dan membesarkan penulis serta tak henti-hentinya memberikan doa, dukungan, dan semangat setiap saat.
2. Selviana Rizky Pramitha, saudara penulis yang selalu memberikan semangat dan doa.
3. Bebbly Idhiani Nikita, terima kasih telah menjadi penyebab dan alasan bagi penulis untuk terus berusaha menjadi yang terbaik. Terima kasih atas pengertian, kesabaran, dan doa yang senantiasa menjadi penyemangat penulis.
4. Ibu Umi Laili Yuhana, S.Kom, M.Sc. atas pengertian dan kesabaran saat menjadi Dosen Pembimbing.
5. Ibu Isye Arieshanti sebagai Dosen Wali penulis, yang telah membantu penulis selama menempuh kuliah S1.
6. Bapak Radityo Anggoro S.Kom., M.Sc. selaku koordinator Tugas Akhir, yang telah banyak membantu penulis dalam menyelesaikan Tugas Akhir.
7. Bapak dan Ibu Dosen Jurusan Teknik Informatika FTIf ITS, yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
8. Bapak dan Ibu pegawai dan staff jurusan Teknik Informatika ITS atas berbagai bantuan yang diberikan selama masa perkuliahan.

9. Muhammad Zaini Gaini, Wahyu Ika Aprilia, Adinegoro Choliq, Deka Agrapradhana, dan M. Misbachul Huda, teman-teman yang telah membantu penulis dalam pengerjaan Tugas Akhir (TA).
10. Teman-teman seperjuangan dalam mengerjakan Tugas Akhir (TA) 2014. Rheza Andana Memampo, Dwi Sulisty Nugroho, M. Haqqi W. dan Fandiasa Koruma Kostrada, Imam A. L, Fajar H. atas kerjasama dan suka duka yang dilalui bersama.
11. Keluarga Besar Angkatan 2009 yang telah menjadi keluarga kedua penulis selama menempuh kuliah.
12. Pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu per satu.

Penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan, maupun kelalaian yang telah dilakukan. Penulis berharap Tugas Akhir ini bisa bermanfaat bagi masyarakat, bangsa, dan negara.

Surabaya, Juli 2014

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan	2
1.3 Batasan Permasalahan.....	2
1.4 Tujuan Tugas Akhir	3
1.5 Manfaat Tugas Akhir	3
1.6 Metodologi Pengerjaan	3
1.6.1 Penyusunan Proposal Tugas Akhir	3
1.6.2 Studi Literatur	3
1.6.3 Analisis	4
1.6.4 Perancangan Sistem	4
1.6.5 Implementasi.....	4
1.6.6 Pengujian Dan Evaluasi	4
1.6.7 Penyusunan Buku TA	5
1.7 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Bencana Banjir.....	7
2.2 Mitigasi Risiko.....	7
2.3 Peringatan Dini	8
2.4 Sistem Informasi Geografis	8
2.5 Referensi Aplikasi Sejenis	10
2.6 Sistem Koordinat	10
2.7 Teknis Pengukuran Dan Pemetaan Kota.....	11
2.5.1. Metode Menghitung Luas Area Permukaan Tanah	12
2.5.2. Metode Menghitung Volume Tanah.....	13

2.8	Framework .NET	14
2.9	MySQL	14
2.10	Shapefile.....	15
2.11	Easy GIS .NET.....	16
2.12	SMS Gateway	16
BAB III ANALISIS DAN PERANCANGAN		17
3.1	Analisis Sistem.....	17
3.1.1	Gambaran Umum Perangkat Lunak.....	18
3.1.2	Arsitektur Perangkat Lunak	20
3.1.3	Spesifikasi Kebutuhan Perangkat Lunak	21
3.1.4	Analisis <i>Stakeholder</i>	22
3.1.5	Skenario Kasus Pengguna.....	23
3.2	Perancangan Sistem	32
3.2.1	Perancangan Basis Data	32
3.2.2	Perancangan Diagram Kelas	40
3.2.3	Perancangan Layanan <i>Web</i>	43
3.2.4	Perancangan Aplikasi Pembangkit Data Sensor	44
3.2.5	Perancangan Proses Aplikasi	45
3.2.6	Perancangan Antarmuka	49
BAB IV IMPLEMENTASI.....		57
4.1	Lingkungan Pembangunan.....	57
4.1.1	Lingkungan Pembangunan Perangkat Keras.....	57
4.1.2	Lingkungan Pembangunan Perangkat Lunak.....	57
4.2	Implementasi Layanan <i>Web</i> Pengirim Data Sensor	58
4.3	Implementasi Kelas.....	58
4.3.1	Kelas MapControl	59
4.3.2	Kelas DataSection	60
4.4	Implementasi Aplikasi Pembangkit Data Sensor	62
4.5	Implementasi Antarmuka Dan Proses Aplikasi	63
4.5.1	Implementasi Antarmuka Halaman Utama.....	64
4.5.2	Implementasi Antarmuka <i>Login</i>	69
4.5.3	Implementasi Antarmuka Mengelola Data Wilayah.....	71
4.5.4	Implementasi Antarmuka Mengelola Data Sungai	77
4.5.5	Implementasi Antarmuka Mengelola Data Aliran	84
4.5.6	Implementasi Antarmuka Pemantauan Kondisi Sungai...85	

4.5.7	Implementasi Antarmuka Melihat Laporan Data Sungai	90
4.5.8	Implementasi Antarmuka Detil Profil Sungai	91
4.5.9	Implementasi Antarmuka Aplikasi Pembangkit Data Sensor	91
BAB V PENGUJIAN DAN EVALUASI		93
5.1	Lingkungan Pengujian	93
5.1.1	Lingkungan Perangkat Keras	93
5.1.2	Lingkungan Perangkat Lunak	94
5.2	Pengujian Fungsionalitas	94
5.2.1	Skenario Pengujian Fungsionalitas	94
5.2.2	Hasil Pengujian Fungsionalitas	96
5.2.3	Evaluasi Hasil Pengujian Fungsionalitas	113
5.3	Pengujian Kegunaan	113
5.3.1	Kriteria Responden	113
5.3.2	Skenario Pengujian Kegunaan	114
5.3.3	Hasil Pengujian Kegunaan	114
5.3.4	Evaluasi Hasil Pengujian Kegunaan	116
BAB VI PENUTUP		117
6.1	Kesimpulan	117
6.2	Saran	118
DAFTAR PUSTAKA		119
LAMPIRAN A – FORMULIR PENILAIAN PENGUJIAN KEGUNAAN		121
LAMPIRAN B – HASIL PENGUJIAN KEGUNAAN		123
BIODATA PENULIS		127

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Tingkat Siaga Banjir Pada Sungai.....	8
Tabel 3.1 Deskripsi Kasus Penggunaan Sistem	23
Tabel 3.2 Spesifikasi Kasus Penggunaan: Mengelola Wilayah Pantauan	24
Tabel 3.3 Spesifikasi Kasus Penggunaan: Melihat Hasil Pantauan Sungai.....	27
Tabel 3.4 Spesifikasi Kasus Penggunaan: Menyimulasikan Data Sensor.....	30
Tabel 3.5 Deskripsi Tabel User.....	32
Tabel 3.6 Deskripsi Tabel Wilayah.....	34
Tabel 3.7 Deskripsi Tabel Section_Sungai	34
Tabel 3.8 Deskripsi Tabel Record.....	36
Tabel 3.9 Deskripsi Tabel Flow	37
Tabel 3.10 Deskripsi Tabel Shapeused	38
Tabel 3.11 Deskripsi Tabel Phonecontact.....	38
Tabel 3.12 Deskripsi Tabel Histori	39
Tabel 3.13 Deskripsi Tabel Histori_flood.....	40
Tabel 3.14 Struktur XML Pengiriman Data Sensor Buatan.....	44
Tabel 3.15 Status pesan peringatan	46
Tabel 5.1 Lingkungan Perangkat Keras Pengujian	93
Tabel 5.2 Lingkungan Perangkat Lunak	94
Tabel 5.3 Skenario Membuat Menambah Data Wilayah Pantauan	96
Tabel 5.4 Skenario Pengujian Menambahkan Peta	97
Tabel 5.5 Skenario Pengujian Membuat Data Profil Sungai.....	99
Tabel 5.6 Skenario Pengujian Menambah Data Penampang Sungai.....	100
Tabel 5.7 Skenario Pengujian Menambah Data Aliran Sungai.	102
Tabel 5.8 Skenario Pengujian Menghitung Luas Area Dan Volume Kontur.....	103
Tabel 5.9 Skenario Pengujian Melihat Pantauan Sungai.....	105
Tabel 5.10 Skenario Pengujian Melihat Laporan Pantauan	106
Tabel 5.11 Skenario Pengujian Menerima SMS Peringatan	108

Tabel 5.12 Skenario Pengujian Mengubah Tampilan Peta.....	108
Tabel 5.13 Skenario Pengujian Menampilkan Tinggi Genangan Air.....	111
Tabel 5.14 Skenario Pengujian Menghapus Wilayah Pantauan	112
Tabel 5.15 Daftar Responden	114
Tabel 5.16 Daftar Perubahan Penilaian pada Pengujian.....	115
Tabel 5.17 Penilaian Antarmuka Pengguna	115
Tabel 5.18 Penilaian Penyampaian Informasi Kondisi Sungai .	115
Tabel 5.19 Penilaian Penyampaian Peringatan SMS.....	115
Tabel 5.20 Rekapitulasi Akhir Pengujian Kegunaan.....	116

DAFTAR GAMBAR

Gambar 2.1 Halaman <i>Web</i> Tech4Water	10
Gambar 2.2 Peta Sistem Koordinat UTM	11
Gambar 2.3 Ilustrasi Metode Menghitung Luas Area Tanah	12
Gambar 3.1 Gambaran Umum Perangkat Lunak	20
Gambar 3.2 Arsitektur Perangkat Lunak Yang Dibangun	21
Gambar 3.3 Diagram Kasus Penggunaan Sistem	23
Gambar 3.4 Diagram Aktivitas Mengelola Wilayah Pantauan ...	26
Gambar 3.5 Diagram Aktivitas Melihat Hasil Pantauan Sungai	29
Gambar 3.6 Diagram Kasus Menyimulasikan Data Sensor	31
Gambar 3.7 Rancangan <i>Conceptual Data Model</i> Aplikasi	33
Gambar 3.8 Rancangan Diagram Kelas Aplikasi	40
Gambar 3.9 Rancangan Diagram Kelas MapControl	41
Gambar 3.10 Rancangan Diagram Kelas DataSection	41
Gambar 3.11 Rancangan <i>Physical Data Model</i> Aplikasi	42
Gambar 3.12 Arsitektur Layanan <i>Web</i> Pengirim Data Sensor Buatan	43
Gambar 3.13 Arsitektur Aplikasi Pembangkit Data Sensor	45
Gambar 3.14 Rancangan Antarmuka Halaman Utama	50
Gambar 3.15 Rancangan Antarmuka Halaman <i>Login</i>	50
Gambar 3.16 Rancangan Antarmuka Mengelola Data Wilayah	51
Gambar 3.17 Rancangan Antarmuka Mengelola Data Sungai	52
Gambar 3.18 Rancangan Antarmuka Mengelola Data Aliran	52
Gambar 3.19 Rancangan Antarmuka Pemantauan Kondisi Sungai	53
Gambar 3.20 Rancangan Antarmuka Melihat Laporan Data Sungai	54
Gambar 3.21 Rancangan Antarmuka Detil Profil Sungai	54
Gambar 3.22 Rancangan Antarmuka Aplikasi Pembangkit Data Sensor	55
Gambar 4.1 Implementasi Antarmuka Halaman Utama	64
Gambar 4.2 Implementasi Antarmuka <i>Login</i>	69
Gambar 4.3 Implementasi Antarmuka Mengelola Data Wilayah	71
Gambar 4.4 Implementasi Antarmuka Mengelola Data Sungai ..	79

Gambar 4.5 Implementasi Antarmuka Mengelola Data Aliran...	84
Gambar 4.6 Implementasi Antarmuka Pemantauan Kondisi Sungai.....	86
Gambar 4.7 Implementasi Antarmuka Melihat Laporan Data Sungai.....	91
Gambar 4.8 Implementasi Antarmuka Detil Profil Sungai	92
Gambar 4.9 Implementasi Antarmuka Aplikasi Pembangkit Data Sensor.....	92
Gambar 5.1 Skenario Layanan <i>Web</i> Pengiriman Data Sensor Buatan Pada Menit Kesatu	95
Gambar 5.2 Skenario Layanan <i>Web</i> Pengiriman Data Sensor Buatan Pada Menit Ketiga.....	95
Gambar 5.3 Skenario Layanan <i>Web</i> Pengiriman Data Sensor Buatan Pada Menit Keempat	96
Gambar 5.4 Pengujian Skenario Menambahkan Wilayah Pantauan	97
Gambar 5.5 Tampilan <i>GridView</i> Nama Wilayah Berhasil Dimasukkan.....	97
Gambar 5.6 Pengujian Skenario Menambahkan Peta.....	98
Gambar 5.7 Tampilan Panel Peta Ketika Peta Berhasil Ditambahkan.....	98
Gambar 5.8 Pengujian Skenario Menambahkan Data Profil Sungai.....	100
Gambar 5.9 Tampilan <i>GridView</i> Data Profil Sungai Berhasil Dimasukkan.....	100
Gambar 5.10 Pengujian Skenario Menambahkan Data Penampang Sungai.....	101
Gambar 5.11 Pengujian Skenario Menambah Data Aliran Sungai	102
Gambar 5.12 Tampilan <i>GridView</i> Data Aliran Berhasil Dimasukkan.....	103
Gambar 5.13 Pengujian Menggambar Area Yang Ingin Dihitung	104
Gambar 5.14 Pengujian Mengisikan Data Elevasi Kontur.....	104

Gambar 5.15 Tampilan <i>GridView</i> Data Area Berhasil Ditambahkan	105
Gambar 5.16 Skenario Pengujian Melihat Pantauan Sungai.....	106
Gambar 5.17 Tampilan Halaman Pantauan Sungai.....	106
Gambar 5.18 Tampilan Halaman Laporan Pantauan Profil Sungai	107
Gambar 5.19 Hasil Pengujian Menerima SMS Peringatan	109
Gambar 5.20 Skenario Pengujian Mengubah Tampilan Peta....	110
Gambar 5.21 Tampilan Peta Berhasil Diubah.....	110
Gambar 5.22 Hasil Pengujian Menampilkan Tinggi Genangan Air	111
Gambar 5.23 Skenario Pengujian Menghapus Wilayah Pantauan	112
Gambar 5.24 Tampilan <i>GridView</i> Wilayah Pantauan Setelah Penghapusan.....	112
Gambar 9.1 Bukti Pengujian Kegunaan (1)	123
Gambar 9.2 Bukti Pengujian Kegunaan (2)	124
Gambar 9.3 Bukti Pengujian Kegunaan (3)	125
Gambar 9.4 Bukti Pengujian Kegunaan (4)	126

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Layanan Web Pengirim Data Sensor Buatan.....	58
Kode Sumber 4.2 Implementasi Kelas MapControl.....	60
Kode Sumber 4.3 Implementasi Kelas DataSection.....	62
Kode Sumber 4.4 Implementasi Agen Pembangkit Data Sensor	63
Kode Sumber 4.5 Implementasi Proses Menampilkan Antarmuka Login.....	65
Kode Sumber 4.6 Implementasi Proses Menambah Data Wilayah Pantauan	66
Kode Sumber 4.7 Implementasi Proses Mengubah Data Wilayah Pantauan	67
Kode Sumber 4.8 Implementasi Proses Menghapus Data Wilayah Pantauan	68
Kode Sumber 4.9 Implementasi Proses Menampilkan Antarmuka Melihat Hasil Pantauan	68
Kode Sumber 4.10 Implementasi Proses Menampilkan Antarmuka Mengelola Data Langganan SMS Peringatan.....	69
Kode Sumber 4.11 Implementasi Proses Login	71
Kode Sumber 4.12 Implementasi Proses Menambah Peta	72
Kode Sumber 4.13 Implementasi Proses Menampilkan Peta.....	73
Kode Sumber 4.14 Implementasi Proses Menghapus Peta	74
Kode Sumber 4.15 Implementasi Proses Menampilkan Antarmuka Mengubah Lapisan Peta	75
Kode Sumber 4.16 Implementasi Proses Menambah Data Potongan Bagian Sungai	75
Kode Sumber 4.17 Implementasi Proses Menambah Data Aliran Sungai.....	76
Kode Sumber 4.18 Implementasi Proses Menghitung Luas Dan Volume Kontur.....	77
Kode Sumber 4.19 Implementasi Proses Menambah Data Potongan Bagian Sungai	79
Kode Sumber 4.20 Implementasi Proses Menambah Data Penampang Sungai Dari <i>File</i> Excel.....	81

Kode Sumber 4.21 Implementasi Proses Menggambar Data Penampang	83
Kode Sumber 4.22 Implementasi Proses Menambah Data Aliran	85
Kode Sumber 4.23 Implementasi Proses Mengambil Data Sensor	86
Kode Sumber 4.24 Implementasi Proses Mengirimkan SMS Peringatan.....	88
Kode Sumber 4.25 Implementasi Proses Mengirimkan SMS Peringatan (2)	89
Kode Sumber 4.26 Implementasi Proses Menghitung Tinggi Genangan Air.....	90

BAB I

PENDAHULUAN

Pada bagian ini akan dijelaskan hal-hal yang menjadi latar belakang pembuatan Tugas Akhir (TA), permasalahan yang dihadapi, batasan masalah, tujuan dan manfaat yang ingin dicapai dalam pengerjaan TA, metodologi pembuatan TA, dan sistematika penulisan yang digunakan dalam pembuatan TA.

1.1 Latar Belakang

Bencana banjir merupakan kejadian alam yang sulit diduga karena datang secara tiba-tiba dengan periodisitas yang tidak menentu, kecuali daerah-daerah yang sudah menjadi langganan terjadinya banjir tahunan. Secara umum banjir adalah peristiwa dimana daratan yang biasanya kering (bukan daerah rawa) menjadi tergenang oleh air, hal ini disebabkan oleh curah hujan yang tinggi dan kondisi topografi wilayah yang rendah hingga cekungan. Penyebab banjir salah satunya dikarenakan intensitas hujan yang besar sehingga air meluap dari sungai. Dampak dari adanya banjir salah satunya yaitu mendatangkan kerugian berupa harta, benda, bahkan korban jiwa. Salah satu cara untuk memitigasi risiko bencana banjir tersebut adalah dengan adanya peringatan dini.

Saat ini banyak teknologi sistem sensor yang dikembangkan untuk memantau keadaan aliran sungai secara *real-time*, hal ini dapat sangat membantu dalam mengetahui kondisi sungai. Dari data hasil pantauan sensor di setiap titik pantauan sungai dapat diolah menjadi informasi untuk memberikan peringatan dini kepada pegawai pengawas sungai mengenai banjir di daerah pantauan.

Sistem Informasi Geografis memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi menurut

lokasinya. Sehingga, SIG dapat membantu perencana untuk tanggap darurat saat akan terjadi bencana alam.

Pada TA ini dibuat sebuah aplikasi sistem informasi geografis yang dapat memberikan peringatan dini untuk mitigasi risiko bencana banjir. Aplikasi ini berbasis *desktop* dan berjalan di sistem operasi windows. Aplikasi ini menggunakan *framework* .NET 4.0 ke atas dalam proses pengembangannya.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam TA ini dapat dipaparkan sebagai berikut.

1. Bagaimana membangun aplikasi sistem informasi geografis yang mampu menampilkan data kondisi wilayah pantauan sungai?
2. Bagaimana membangun aplikasi sistem informasi geografis yang dapat mengolah data yang didapat sehingga dapat memberikan peringatan dini?

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam TA ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Lapisan peta yang digunakan berformat *shapefile* dan data-data sungai lengkap.
2. Data sensor yang digunakan adalah data sensor tinggi muka air dan kecepatan air.
3. Aplikasi yang dibangun berbasis *desktop*.
4. Bahasa pemrograman yang digunakan bahasa C#.
5. Pengujian menggunakan data sensor buatan.
6. Diasumsikan pengiriman data sensor melalui *web service*.
7. Tidak mengawasi faktor bendungan pada sungai.

1.4 Tujuan Tugas Akhir

TA ini memiliki tujuan yang rinciannya dapat ditulis sebagai berikut.

1. Terbentuknya sebuah aplikasi sistem informasi geografis yang mampu memberikan peringatan dini dengan mengirim SMS ke pengguna.
2. Terbentuknya sebuah aplikasi sistem informasi geografis yang mampu menampilkan data hasil pantauan.

1.5 Manfaat Tugas Akhir

Manfaat yang diharapkan dari TA ini adalah sebagai berikut.

1. Memudahkan proses pemantauan keadaan sungai di beberapa titik pantauan.
2. Memitigasi risiko bencana banjir dengan adanya peringatan dini.

1.6 Metodologi Pengerjaan

Tahapan yang akan dilakukan dalam TA ini di antaranya sebagai berikut:

1.6.1 Penyusunan Proposal Tugas Akhir

Tahap awal untuk memulai pengerjaan TA adalah penyusunan proposal TA. Pada proposal ini, penulis mengajukan gagasan perancangan dan pengembangan aplikasi sistem informasi geografis sebagai sistem peringatan dini untuk mitigasi risiko bencana banjir dengan pingingat SMS.

1.6.2 Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk pengerjaan TA sekaligus mempelajarinya.

Mulai dari pengumpulan literatur, diskusi, serta pemahaman topik TA di antaranya tentang:

- a. Metode untuk menampilkan *layer* peta pada aplikasi dengan menggunakan bahasa C#.
- b. Metode untuk mengolah data yang didapat sehingga dapat memberikan peringatan dini.

1.6.3 Analisis

Tahapan ini menganalisis permasalahan yang dihadapi dalam pembuatan aplikasi, deskripsi secara umum perangkat lunak, arsitektur sistem yang dibuat, spesifikasi kebutuhan perangkat lunak, menganalisis aktor yang dapat menggunakan perangkat lunak dan skenario kasus penggunaan perangkat lunak.

1.6.4 Perancangan Sistem

Tahapan ini digunakan untuk merancang sistem. Perancangan sistem terdiri dari perancangan terhadap data yang digunakan di dalam sistem, perancangan proses aplikasi, perancangan antarmuka grafis, perancangan hasil akhir yang diharapkan pada sistem dan perancangan diagram kelas sistem.

1.6.5 Implementasi

Implementasi merupakan tahap untuk membangun aplikasi sistem informasi geografis sebagai sistem peringatan dini untuk mitigasi risiko bencana banjir dengan menggunakan bahasa pemrograman C#. Sistem dibangun berpedoman pada konsep-konsep yang telah ditentukan sebelumnya pada analisis dan perancangan sistem.

1.6.6 Pengujian Dan Evaluasi

Pada tahap ini dilakukan uji coba terhadap aplikasi yang telah dibuat. Langkah-langkah pengujian yang akan dilakukan antara lain:

- a. Menguji apakah aplikasi yang dibuat sesuai dengan fungsionalitas bisnis prosesnya.
- b. Menguji apakah aplikasi telah sesuai secara non fungsionalitas dalam bisnis prosesnya.

1.6.7 Penyusunan Buku TA

Tahap terakhir merupakan penyusunan laporan yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perancangan yang telah dibuat. Secara garis besar, Buku TA yang nantinya akan dibuat terdiri dari beberapa bagian, yaitu :

1. Pendahuluan
 - 1.1. Latar Belakang
 - 1.2. Rumusan Permasalahan
 - 1.3. Batasan Permasalahan
 - 1.4. Tujuan Tugas Akhir
 - 1.5. Manfaat Tugas Akhir
 - 1.6. Metodologi
 - 1.7. Sistematika Penulisan
2. Kajian Pustaka
3. Analisis dan Perancangan
4. Implementasi
5. Pengujian dan Evaluasi
6. Penutup

Daftar Pustaka

1.7 Sistematika Penulisan

Buku TA ini terdiri dari beberapa bab yang dijelaskan sebagai berikut:

- Bab I Pendahuluan

Bab ini berisi latar belakang masalah, permasalahan, batasan masalah, tujuan TA, manfaat TA, metodologi yang digunakan dan sistematika penyusunan buku TA.

- Bab II Kajian Pustaka

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan TA ini.

- Bab III Analisis dan Perancangan

Bab ini membahas analisis kebutuhan bisnis dari proses bisnis yang ada dan rancangan dari aplikasi yang akan dibangun.

- Bab IV Implementasi

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan.

- Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat hasil keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

- Bab VI Penutup

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai dasar teori yang menjadi dasar pembuatan TA ini. Dasar teori yang dibahas di dalam bab ini adalah kajian mengenai Easy GIS .NET yang menjadi pustaka untuk pembuatan aplikasi, *framework* .NET, sebagai kerangka kerja yang digunakan dalam pembuatan aplikasi, penjelasan mengenai metode yang akan digunakan untuk melakukan penghitungan luas tanah dan volume kontur serta metode untuk menghitung debit air sungai yang meluap.

2.1 Bencana Banjir

Banjir adalah peristiwa terbenamnya daratan (yang biasanya kering) karena volume air yang meningkat. Banjir dapat terjadi karena peluapan air yang berlebihan di suatu tempat akibat hujan besar, peluapan air sungai, atau pecahnya bendungan sungai [1]. Banjir merupakan fenomena alam yang biasa terjadi di suatu kawasan yang banyak dialiri oleh aliran sungai.

Dampak adanya banjir, yaitu sebagai berikut:

- Mendatangkan kerugian yang berupa harta, benda, bahkan korban jiwa
- Merusak sarana dan prasarana umum, misalnya jalan raya yang rusak, jembatan hancur dan lain sebagainya
- Jika menerjang areal pertanian akan menyebabkan gagal panen
- Masyarakat akan kesulitan mendapatkan air bersih
- Sebagai media penyakit perut dan penyakit kulit.

2.2 Mitigasi Risiko

Mitigasi Risiko adalah suatu tindakan terencana dan berkelanjutan yang dilakukan oleh pemilik risiko agar bisa mengurangi dampak dari suatu kejadian yang berpotensi atau

telah merugikan atau membahayakan pemilik risiko tersebut [2]. Seperti sebelumnya yang telah dijelaskan pada subbab 2.1 mengenai dampak terjadinya banjir, dampak bencana banjir yang akan dimitigasi adalah mengurangi korban jiwa akibat bencana banjir dengan memberikan peringatan dini.

2.3 Peringatan Dini

Peringatan dini adalah serangkaian kegiatan pemberian peringatan sesegera mungkin kepada masyarakat tentang kemungkinan terjadinya bencana pada suatu tempat oleh lembaga yang berwenang (UU No. 24 Tahun 2007 Pasal 1 angka 8) [3]. Sistem peringatan dini digunakan sebisa mungkin untuk mencegah suatu hal buruk yang akan terjadi dengan memberikan peringatan sedini mungkin kepada yang bersangkutan agar bisa menghindari atau meminimalkan akibat yang ditimbulkan hal buruk tersebut. Aplikasi yang akan dibuat pada Tugas Akhir ini berfungsi untuk memberikan peringatan terjadinya bencana kepada pengguna dengan menggunakan data yang didapat. Tingkat siaga banjir dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tingkat Siaga Banjir Pada Sungai

No.	Tingkat Bahaya	Tingkat Siaga	Tinggi Jagangan Sungai	Selang Waktu Pengamatan
1.	Bahaya-I	Siaga-I	Ditetapkan sesuai kondisi sungai	Terus Menerus
2.	Bahaya-II	Siaga-II	Ditetapkan sesuai kondisi sungai	1 Jam
3.	Bahaya-III	Siaga-III	Ditetapkan sesuai kondisi sungai	2 Jam

2.4 Sistem Informasi Geografis

Sistem Informasi Geografis (*Geographic Information System* disingkat GIS) adalah sistem informasi khusus pengelola data yang memiliki informasi spasial (bereferensi

keruangan) yang dirancang untuk bekerja dengan data yang berkoordinat geografi [4]. Atau dalam arti yang lebih sempit, adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi menurut lokasinya dalam sebuah *database*.

Aplikasi SIG dapat digunakan untuk berbagai kepentingan selama data yang diolah memiliki referensi geografi, maksudnya data tersebut terdiri dari fenomena atau objek yang dapat disajikan dalam bentuk fisik serta memiliki lokasi keruangan. Tujuan pokok dari pemanfaatan Sistem Informasi Geografis adalah untuk mempermudah mendapatkan informasi yang telah diolah dan tersimpan sebagai atribut suatu lokasi atau obyek.

Data-data yang diolah dalam SIG pada dasarnya terdiri dari data spasial dan data atribut dalam bentuk digital, dengan demikian analisis yang dapat digunakan adalah analisis spasial dan analisis atribut. Data spasial merupakan data yang berkaitan dengan lokasi keruangan yang umumnya berbentuk peta. Sedangkan data atribut merupakan data tabel yang berfungsi menjelaskan keberadaan berbagai objek sebagai data spasial.

Penyajian data spasial mempunyai tiga cara dasar yaitu dalam bentuk titik, bentuk garis dan bentuk area (*polygon*). Titik merupakan kenampakan tunggal dari sepasang koordinat x,y yang menunjukkan lokasi suatu obyek berupa ketinggian, lokasi kota, lokasi pengambilan sampel dan lain-lain. Garis merupakan sekumpulan titik-titik yang membentuk suatu kenampakan memanjang seperti sungai, jalan, kontur dan lain-lain.

Sedangkan area adalah kenampakan yang dibatasi oleh suatu garis yang membentuk suatu ruang homogen, misalnya: batas daerah, batas penggunaan lahan, pulau dan lain sebagainya. Struktur data spasial dibagi dua yaitu model data raster dan model data vektor. Data raster adalah data yang disimpan dalam bentuk kotak segi empat (*grid*) sehingga

terbentuk suatu ruang yang teratur. Data vektor adalah data yang direkam dalam bentuk koordinat titik yang menampilkan, menempatkan dan menyimpan data spasial dengan menggunakan titik, garis atau area (*polygon*).

2.5 Referensi Aplikasi Sejenis

Tech4Water merupakan sebuah halaman *web* yang dikembangkan oleh pusat penelitian dan pengembangan sumber daya air Indonesia yang digunakan untuk memantau kondisi aliran sungai [5]. Pada halaman web Tech4Water, terdapat peta yang menunjukkan lokasi profil sungai yang dipantau. Halaman *web*-nya dapat dilihat pada Gambar 2.1.



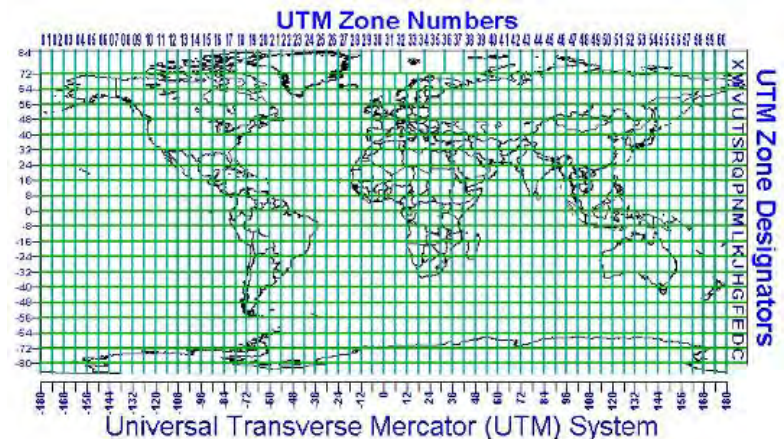
Gambar 2.1 Halaman *Web* Tech4Water

2.6 Sistem Koordinat

Dalam GIS, ada dua sistem koordinat yang biasa digunakan, yaitu koordinat geografi dan UTM (*Universal Transverse Mercator*). Sistem koordinat geografi digunakan

untuk menunjukkan suatu titik di Bumi berdasarkan garis lintang dan garis bujur [6].

Gambar 2.2 menunjukkan gambar peta sistem koordinat UTM pada permukaan bumi. Sistem Proyeksi Koordinat UTM (*Universal Transverse Mercator*) adalah rangkaian proyeksi *Transverse Mercator* untuk global dimana bumi dibagi menjadi 60 bagian zona. Setiap zona mencakup enam derajat bujur (*longitude*) dan memiliki meridian tengah tersendiri. Berbeda dengan koordinat geografi yang satuan unitnya adalah derajat, koordinat UTM menggunakan satuan unit meter. Setiap zona memiliki panjang x sebesar 500.000 meter dan panjang y sebesar 10.000.000 meter.



Gambar 2.2 Peta Sistem Koordinat UTM

2.7 Teknis Pengukuran Dan Pemetaan Kota

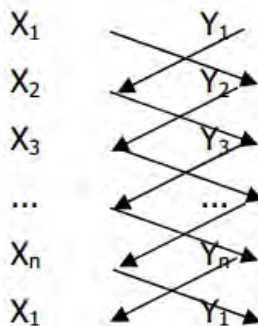
Dalam Tugas Akhir ini akan dibuat beberapa fitur yang mengimplementasikan metode yang ada pada ilmu ukur tanah. Adapun fitur-fitur tersebut adalah fitur untuk menghitung luas area permukaan tanah dan menghitung volume area kontur. Ilmu ukur tanah adalah cabang dari ilmu Geodesi yang khusus

mempelajari sebagian kecil dari permukaan bumi dengan cara melakukan pengukuran-pengukuran guna mendapatkan peta.

2.5.1. Metode Menghitung Luas Area Permukaan Tanah

Metode menghitung luas area permukaan tanah akan diimplementasikan untuk membuat fitur menghitung luas area permukaan tanah di peta [7]. Luas suatu objek (tanah, bangunan, dan lain-lain) di peta merupakan luas pada bidang datar (X,Y). Jadi perlu dijelaskan kepada pengguna peta yang ingin menghitung luas objek yang dimiliki di lapangan nanti hasilnya akan lebih kecil dibandingkan hasil hitungan dengan peta. Pada peta proyeksi yang digunakan adalah proyeksi tegak sehingga objek yang terletak pada bidang miring sebelum digambar di peta objek tersebut harus diproyeksikan terlebih dahulu pada bidang datar. Dengan demikian panjang objek di bidang miring tersebut setelah diproyeksikan akan menjadi lebih pendek dibandingkan kenyataan di lapangan. Konsekuensinya luas objek yang dihasilkan lebih kecil dari luas sebenarnya di lapangan.

Secara umum formula untuk menghitung poligon dengan n titik poligon yang diketahui koordinatnya dapat dituliskan pada Persamaan 2.1. Ilustrasi dari cara penghitungannya dapat dilihat pada Gambar 2.3.



Gambar 2.3 Ilustrasi Metode Menghitung Luas Area Tanah

$$A = \left| \frac{(X_2Y_1 - X_1Y_2) + (X_3Y_2 - X_2Y_3) + \dots + (X_nY_{n-1} - X_{n-1}Y_n)}{2} \right| \quad (2.1)$$

Keterangan: Tanda absolut untuk menghindari hasil luas negatif karena luas hasilnya selalu positif

2.5.2. Metode Menghitung Volume Tanah

Volume tanah yang dimaksud di sini adalah apabila ingin menggali atau menimbun tanah pada suatu tempat (*cut and fill*) atau untuk menghitung material (bahan) galian yang sifatnya padat. Suatu bidang tanah yang mempunyai ketinggian bervariasi, misalnya 10 m, 12 m, 15 m, 13m, 12 m dan seterusnya, jika ingin dibangun gedung di atasnya dengan level (ketinggian) tertentu, misalnya 16 m, maka bidang tanah tersebut harus ditimbun. Yang menjadi pertanyaan adalah berapa volume timbunannya? Volume timbunan ini yang akan dihitung besarnya. Kasus lain, apabila suatu daerah merupakan gundukan (tanah tinggi), sedangkan daerah tersebut akan dibangun dengan ketinggian tertentu yang mengharuskan memangkas (memotong) ketinggian daerah tersebut. Volume galian ini yang akan dihitung besarnya [8].

Perhitungan luas didasarkan pada daerah yang dibatasi masing-masing kontur, sedangkan perhitungan volume dihitung berdasarkan daerah yang dibatasi oleh dua garis kontur yang berurutan. Rumus yang digunakan dalam perhitungan lengkung kapasitas dapat dilihat pada Persamaan 2.2 :

$$V = \sum [(F_i + F_{i+1}) / 2 (h_i + h_{i+1})] \quad (2.2)$$

dimana:

V = volume (m³)

F_i = luas daerah ke-I (m²)

h_i = ketinggian (kontur) daerah ke-I (m)

2.8 Framework .NET

Framework .NET merupakan sebuah perangkat lunak kerangka kerja yang berjalan utamanya pada sistem operasi Microsoft Windows. Kerangka kerja ini menyediakan sejumlah besar *library* pemrograman komputer dan mendukung beberapa bahasa pemrograman serta interoperabilitas yang baik sehingga memungkinkan bahasa-bahasa tersebut berfungsi satu dengan lain dalam pengembangan system [9]. Program yang ditulis dengan memanfaatkan kerangka kerja .NET berjalan pada lingkungan perangkat lunak melalui *Common Language Runtime* dan bukan perangkat keras secara langsung. Hal ini memungkinkan aplikasi yang dibuat di atas kerangka kerja .NET secara teoritis dapat berjalan pada perangkat keras apapun yang didukung oleh kerangka kerja .NET. Program pembentuk *class library* dari kerangka kerja .NET mencakup area yang luas dari kebutuhan program pada bidang antarmuka pengguna, pengaksesan data, koneksi basis data, kriptografi, pembuatan aplikasi berbasis *web*, algoritma numerik dan komunikasi jaringan.

2.9 MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis [10].

Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja pengoptimasiannya dalam melakukan

proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basis data transaksional maupun operasi basis data non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basis data kompetitor lainnya.

Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis *web* (Wordpress), CMS dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basis data transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

2.10 Shapefile

Shapefile ESRI atau biasa disebut *shapefile* adalah format data geospasial yang umum untuk perangkat lunak sistem informasi geografis. Dikembangkan dan diatur oleh ESRI sebagai spesifikasi (hampir) terbuka untuk interoperabilitas data antara ESRI dan produk perangkat lunak lainnya [11].

Sebuah "shapefile" biasanya terdiri dari kumpulan *file* yang berekstensi ".shp", ".shx", ".dbf", dan ekstensi lainnya pada sebuah nama yang sama (misalkan "jalan.*"). Saat penggunaan, *shapefile* sebenarnya yang rujuk adalah yang berekstensi ".shp", namun *file* ini tidak lengkap dan membutuhkan *file* lainnya. Dimana *file* *.shp adalah *file* yang menyimpan data keruangan yang biasa disebut objek (*spatial data*), *.dbf adalah *file* yang menyimpan data tabular yang biasa disebut attribut. Sedangkan *.shx menyimpan data *index* yang digunakan oleh *shapefile* tersebut.

Shapefile biasanya berupa titik (*point*), garis (*polylines*) dan poligon (*polygons*). Dimana titik biasanya merepresentasikan lokasi suatu objek tertentu misal posisi BTS, posisi sekolah dan sebagainya. Sedangkan garis biasanya merepresentasikan objek yang memiliki titik awal dan titik akhir yang berbeda, misal jalan, sungai dan sebagainya. Sementara poligon biasanya digunakan untuk merepresentasikan objek yang memiliki titik awal dan titik akhir sama, misalnya danau.

2.11 Easy GIS .NET

Easy GIS .NET merupakan sederetan dari alat pemetaan GIS .NET dan kontrolnya agar pengembang dengan mudah dapat menggabungkan fungsionalitas GIS ke dalam aplikasinya. Edisi *desktop* yang disediakan Easy GIS .NET berisi aplikasi yang dapat digunakan untuk membuka dan melihat ESRI *shapefile*. *Library* Easy GIS .NET akan digunakan untuk membangun antarmuka aplikasi [12].

2.12 SMS Gateway

SMS Gateway adalah sebuah gerbang yang menghubungkan antara komputer dengan *client* melalui SMS. Jadi secara garis besar, SMS Gateway dapat digambarkan seperti gambar di atas. *Client* secara tidak langsung berinteraksi dengan aplikasi atau sistem melalui SMS Gateway. Saat melakukan SMS, maka informasi terpenting yang diperlukan adalah nomor tujuan dan pesan, maka itulah yang sebenarnya diolah oleh SMS Gateway [13]. Aplikasi akan menggunakan sistem SMS Gateway untuk mengirimkan SMS ke pengguna untuk mengingatkan terjadinya bencana.

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan membahas tahap analisis kebutuhan bisnis dari sistem yang akan dibangun dan perancangannya. Tahap analisis ini membahas mengenai analisis kebutuhan yang menjadi dasar pada tahap perancangan yang akan dipaparkan pada subbab selanjutnya.

3.1 Analisis Sistem

Bencana banjir merupakan kejadian alam yang sulit diduga karena datang secara tiba-tiba dengan periodisitas yang tidak menentu, kecuali daerah-daerah yang sudah menjadi langganan terjadinya banjir tahunan. Berbagai upaya dilakukan dalam aksi penanganan bencana banjir ini. Namun, hingga saat ini banjir masih saja menjadi ancaman bagi masyarakat di Indonesia khususnya yang tinggal di seputaran kali atau sungai. Di Indonesia, penanganan bencana tidak terkecuali bencana banjir, difokuskan pada upaya-upaya pencegahan, mitigasi dan kesiapan sebelum terjadi bencana serta upaya pemulihan yang dilakukan dengan mengerahkan semua potensi sumber daya secara efektif dan efisien. Mitigasi sebelum terjadinya kejadian bencana yang dapat dilakukan adalah dengan peringatan dini.

Pemantauan kondisi aliran sungai dapat dilakukan secara manual dengan melakukan pengukuran di setiap profil sungai (*cross section*) yang akan dipantau. Pengukuran tersebut berupa pengukuran debit, kecepatan dan luas penampang sungai. Untuk dapat mengukur debit, kecepatan dan luas penampang pada suatu profil sungai, kita perlu melakukan survey langsung ke profil sungai tersebut. Sehingga didapat data debit, kecepatan, dan luas penampang suatu profil sungai pada suatu waktu.

Dengan adanya banyak teknologi sistem sensor yang dikembangkan untuk memantau keadaan aliran sungai secara *real-time*, hal ini dapat sangat membantu dalam mengetahui

kondisi sungai. Dari data mentah hasil pantauan sensor di setiap titik pantauan sungai dapat diolah menjadi informasi untuk memberikan peringatan dini kepada masyarakat mengenai banjir di daerah pantauan.

Dari sekian banyaknya teknologi sistem sensor yang berkembang untuk memantau kondisi aliran sungai, maka diperlukan sebuah aplikasi yang dapat mengintegrasikan sensor dan perangkat keras seperti PC yang dapat menampilkan informasi dari hasil pantauan sensor.

3.1.1 Gambaran Umum Perangkat Lunak

Pada TA ini, akan dibuat sebuah aplikasi sistem informasi geografis untuk mitigasi risiko bencana banjir dengan peringatan dini berupa SMS. Aplikasi tersebut dapat menampilkan informasi-informasi seputar titik pantauan sungai secara geografis, dapat menampilkan kondisi sungai berdasarkan data sensor yang ada dan dapat mengirimkan pesan peringatan dini kepada penggunanya. Aplikasi berbasis *desktop* dan berjalan pada sistem operasi Windows.

Pada awal pemakaian aplikasi, pengguna menambahkan data wilayah pantauan sungai baru. Kemudian pengguna melengkapi peta-peta yang ingin ditampilkan pada aplikasi saat proses pemantauan kondisi sungai. Pengguna dapat merubah-ubah tampilan peta misalkan merubah warna dan teks yang akan ditampilkan. Setelah data wilayah baru ditambahkan, pengguna harus melengkapi data sungai pada wilayah tersebut. Pengguna menambahkan data bagian sungai yang akan dipantau. Setiap mengisi data bagian sungai, pengguna diharuskan mengisi url *web service* yang akan digunakan untuk menerima data dari sensor dimasing-masing area pantauan. Dari data sensor tersebut kemudian aplikasi mengolahnya sehingga dapat ditampilkan pada saat proses pemantauan dan diolah untuk memberikan peringatan dan mengirimkan SMS.

Pada aplikasi juga terdapat fitur untuk menghitung luas area. Dengan menggunakan metode koordinat pada ilmu ukur

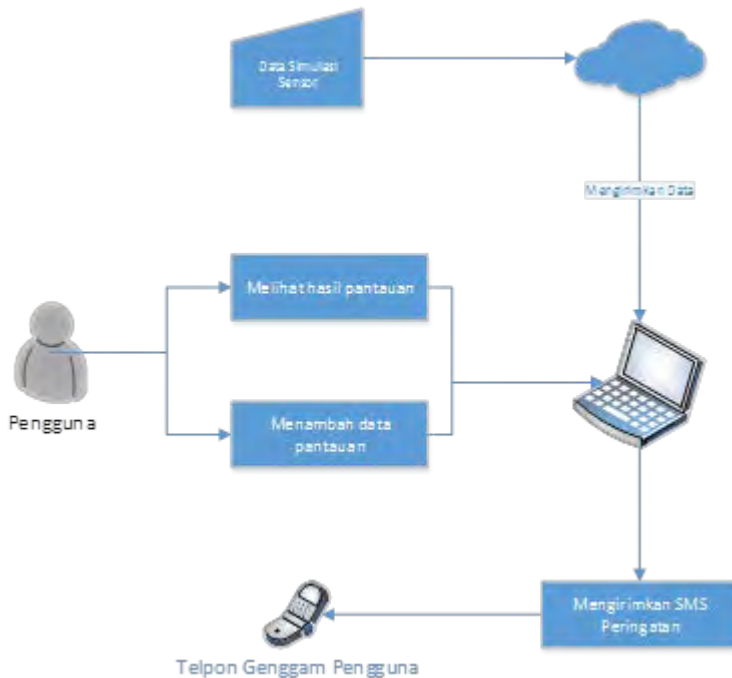
tanah, aplikasi menghitung luas area pada peta. Pengguna memasukkan titik-titik area yang ingin dihitung dengan mengeklik pada peta sehingga terkumpul titik yang membentuk poligon. Dari kumpulan titik tersebut dapat dihitung luas areanya. Untuk menghitung volume tanah, aplikasi mengimplementasikan metode kontur yang ada pada ilmu ukur tanah. Dari data luas area yang sebelumnya sudah dihitung, diberikan titik elevasi area tersebut. Untuk menghitung volume area tanah diperlukan luas dua daerah kontur pada elevasi berbeda, urut dari yang terkecil hingga terbesar elevasinya. Kemudian dijumlah semua volume area tanah yang telah dihitung untuk mendapatkan total volume area pada kontur tersebut.

Keluaran dari TA ini adalah sebuah aplikasi sistem informasi geografis yang digunakan untuk memantau aliran sungai serta dapat memberikan peringatan dengan cara mengirimkan SMS ke pengguna yang telah terdaftar.

Gambaran umum dari perangkat lunak ini dapat dilihat pada Gambar 3.1 yang memperlihatkan bahwa pengguna aplikasi ini dilakukan oleh admin yang bertugas untuk mengawasi bagian-bagian aliran sungai. Pengguna dapat melakukan beberapa kegiatan pada aplikasi seperti menambahkan data wilayah pantauan baru dan melihat hasil pantauan.

Pengguna menggunakan aplikasi di PC (*Personal Computer*) karena berbasis *desktop*. Pengguna mengisikan data-data wilayah pantauan. Setelah data wilayah pantauan lengkap dan url data sensor ada, aplikasi mengambil data sensor untuk kemudian diolah agar bisa memberikan peringatan yang akan dikirimkan ke telpon genggam pengguna.

Pada Tugas Akhir ini, dibuat sebuah aplikasi yang digunakan untuk menyimulasikan data kecepatan dan tinggi muka air sebagai pengganti data sensor asli yang digunakan. Aplikasi pembangkit data sensor ini dibuat ditujukan untuk pengujian fungsional aplikasi agar dapat dilihat apakah aplikasi bisa berjalan sesuai dengan yang diharapkan.



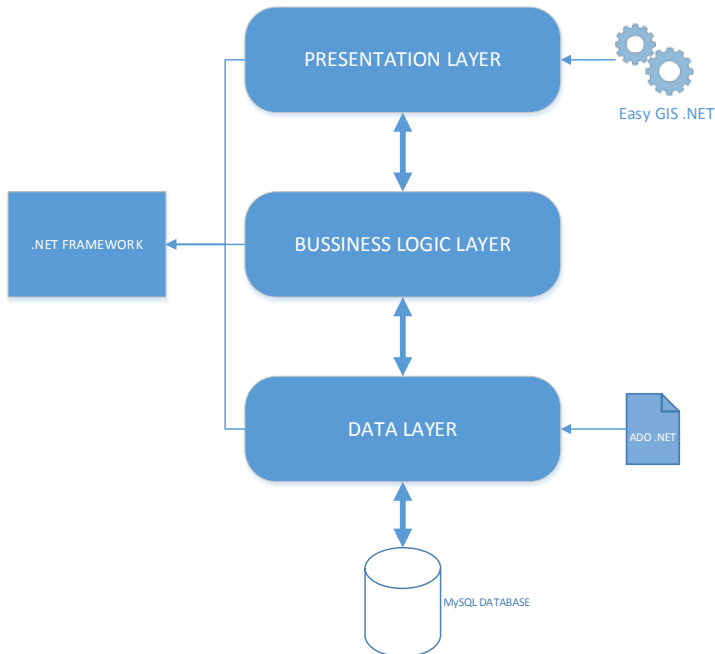
Gambar 3.1 Gambaran Umum Perangkat Lunak

3.1.2 Arsitektur Perangkat Lunak

Arsitektur dari perangkat lunak yang dibangun dapat dilihat pada Gambar 3.2. Perangkat lunak dibangun menggunakan kerangka kerja .NET. Arsitektur perangkat lunak terdiri dari tiga lapisan yaitu *presentation layer*, *business logic layer* dan *data layer*. *Presentation layer* yaitu lapisan pada arsitektur perangkat lunak yang menampilkan hasil dari perangkat lunak ke dalam antarmuka. Pada *layer* ini digunakan pustaka Easy GIS .NET untuk menampilkan dan mengolah peta.

Bussiness logic layer yaitu lapisan pada arsitektur perangkat lunak yang memproses data yang mengalir dari *presentation layer* ke *data layer* dan sebaliknya. Lapisan ini

mengontrol fungsionalitas yang ada pada perangkat lunak. *Data layer* yaitu lapisan pada arsitektur perangkat lunak yang memanajemen data yang mengalir dari basis data.



Gambar 3.2 Arsitektur Perangkat Lunak Yang Dibangun

3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Pada sistem yang akan dibuat, dibutuhkan beberapa fungsi yang dapat membantu proses bisnis dalam sistem. Fungsi-fungsi tersebut antara lain sebagai berikut.

1. Menambahkan wilayah pantauan.
Fungsi ini digunakan untuk membuat data wilayah pantauan baru. Pengguna memasukkan nama wilayah

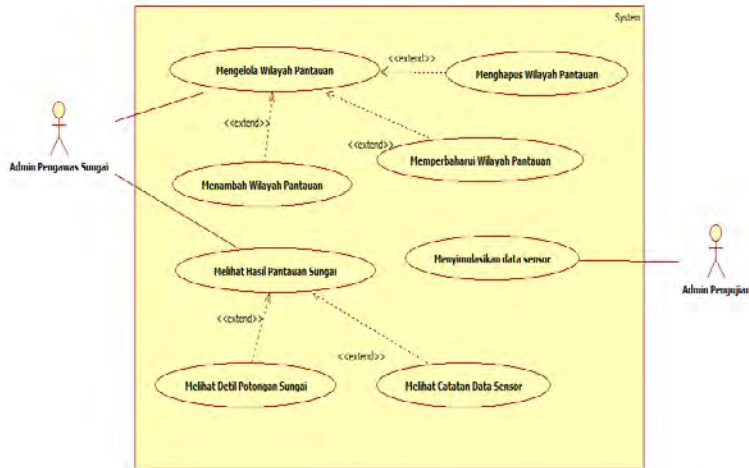
pantauan dan data-data bagian sungai yang berkaitan dengan wilayah tersebut.

2. Menampilkan data pada peta.
Fungsi ini digunakan untuk menampilkan data-data sensor yang berkaitan dalam proses pemantauan kondisi aliran sungai pada lapisan peta yang digunakan.
3. Memanajemen lapisan peta yang digunakan.
Fungsi ini digunakan untuk menambah, menghapus dan mengatur posisi lapisan peta. Pengguna juga dapat mengubah tampilan peta dan teks yang ingin ditampilkan pada peta.
4. Mengirimkan SMS peringatan.
Fungsi ini digunakan untuk mengolah data yang didapat dari sensor yang digunakan untuk mengawasi kondisi aliran sungai. Data sensor yang didapat kemudian diolah sedemikian rupa sehingga aplikasi mampu memberikan peringatan yang nantinya akan dikirimkan melalui SMS.
5. Menghitung luas dan volume area kontur.
Fungsi ini digunakan untuk menghitung luas area dan volume area kontur menggunakan peta kontur yang tersedia untuk mengetahui lengkung kapasitas pada suatu daerah seperti bendungan.

3.1.4 Analisis Stakeholder

Stakeholder didefinisikan sebagai aktor yang berperan di dalam aplikasi ini. Dalam aplikasi sistem informasi geografis ini terdapat beberapa aktor yaitu admin pengawas sungai dan admin testing. Admin pengawas sungai disini merupakan pengguna yang bertugas untuk memantau kondisi sungai dan pengguna yang mempunyai dan memahami data yang digunakan pada aplikasi. Admin pengawas sungai mengisi data-data yang bersangkutan dengan wilayah dan kondisi sungai yang diawasi. Admin pengujian merupakan pengguna yang menjalankan aplikasi pembangkit data sensor untuk menyimulasikan data sensor untuk

tujuan pengujian fungsionalitas aplikasi SIG yang dibangun. Data sensor dari bagian-bagian sungai ditampilkan pada peta.



Gambar 3.3 Diagram Kasus Penggunaan Sistem

3.1.5 Skenario Kasus Pengguna

Secara umum, kasus penggunaan sistem digambarkan pada kasus penggunaan diagram pada Gambar 3.3. Dari diagram tersebut, maka dapat diketahui bahwa terdapat enam kasus penggunaan yang ditunjukkan pada Tabel 3.1 mengenai penjelasan dari masing-masing kasus penggunaan beserta kodenya.

Selain spesifikasi kasus penggunaan, disertakan pula diagram aktivitas yang merupakan ilustrasi alur normal dari kasus penggunaan yang bersangkutan.

Tabel 3.1 Deskripsi Kasus Penggunaan Sistem

No.	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1.	UC-001	Mengelola	Pengguna dapat menambah,

		Wilayah Pantauan	mengubah, menghapus data wilayah pantauan.
2.	UC-002	Melihat Hasil Pantauan Sungai	Pengguna dapat melihat data kondisi aliran sungai dan posisinya pada peta.
3.	UC-003	Menyimulasikan data sensor	Admin pengujian dapat menambah dan mengubah data sensor untuk tujuan pengujian.

3.1.5.1 Kasus Penggunaan Mengelola Wilayah Pantauan

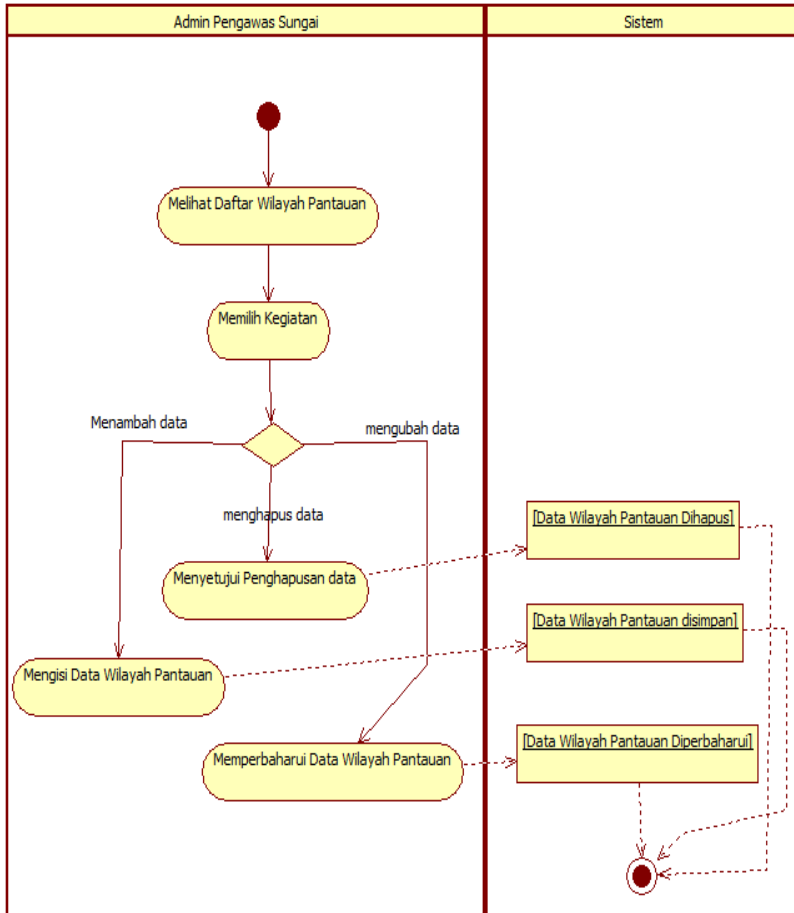
Spesifikasi kasus penggunaan mengelola wilayah pantauan pada aplikasi dijelaskan pada Tabel 3.2. Aktivitas yang dilakukan untuk memenuhi kasus penggunaan ini tergambar pada diagram aktivitas pada Gambar 3.4. Pada kasus penggunaan ini, pengguna dapat membuat data wilayah baru, mengubah data wilayah pantauan dan menghapus data wilayah pantauan yang tidak digunakan.

Tabel 3.2 Spesifikasi Kasus Penggunaan: Mengelola Wilayah Pantauan

Nama Kasus Penggunaan	Mengelola Wilayah Pantauan
Nomor	UC-001
Deskripsi	Pengguna dapat memanajemen data wilayah pantauan
Aktor	Admin pengawas sungai.
Kondisi Awal	Pengguna sudah login ke sistem dan berada pada halaman beranda.
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar wilayah pantauan. 2. Pengguna memilih kegiatan yang dapat dilakukan. <ol style="list-style-type: none"> A1. Pengguna memilih membuat wilayah pantauan baru. A2. Pengguna memilih mengubah data

	<p>wilayah pantauan.</p> <p>A3. Pengguna memilih menghapus wilayah pantauan.</p> <p>3. Kasus penggunaan berakhir</p>
<p>Alur Alternatif</p>	<p>A1. Pengguna memilih membuat wilayah pantauan baru.</p> <ol style="list-style-type: none"> 1. Pada halaman beranda, pengguna mengisi nama wilayah pantauan. 2. Sistem menampilkan halaman mengelola data wilayah pantauan. 3. Pengguna menambahkan peta yang digunakan dan mengatur tampilan peta. 4. Sistem menyimpan peta dan pengaturan peta. 5. Pengguna mengisikan data potongan profil sungai. 6. Sistem menyimpan data potongan profil sungai. 7. Pengguna mengisikan data aliran sungai. 8. Sistem menyimpan data aliran sungai. 9. Pengguna mengisikan data luas dan volume kontur. 10. Sistem menyimpan data luas dan volume kontur. 11. Kembali ke alur normal langkah 3. <p>A2. Pengguna memilih mengubah data wilayah pantauan.</p> <ol style="list-style-type: none"> 1. Sistem menampilkan halaman mengelola data wilayah pantauan 2. Pengguna mengubah data wilayah pantauan 3. Sistem memperbaharui data wilayah pantauan. 4. Kembali ke alur normal langkah 3. <p>A3. Pengguna memilih menghapus data wilayah pantauan.</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pemberitahuan penghapusan wilayah pantauan 2. Pengguna menyetujui penghapusan.

	3. Sistem menghapus data. 4. Kembali ke alur normal langkah 3.
Kondisi Akhir	Sistem menampilkan daftar wilayah pantauan.



Gambar 3.4 Diagram Aktivitas Mengelola Wilayah Pantauan

3.1.5.2 Kasus Penggunaan Melihat Hasil Pemantauan Sungai

Spesifikasi kasus penggunaan melihat hasil pantauan sungai pada aplikasi dijelaskan pada Tabel 3.3. Aktivitas yang dilakukan untuk memenuhi kasus penggunaan ini tergambar pada diagram aktivitas Gambar 3.5. Pada kasus penggunaan ini, pengguna dapat melihat hasil pengawasan oleh sensor yang dipasang di bagian potongan sungai dan posisi dari bagian sungai tersebut di peta.

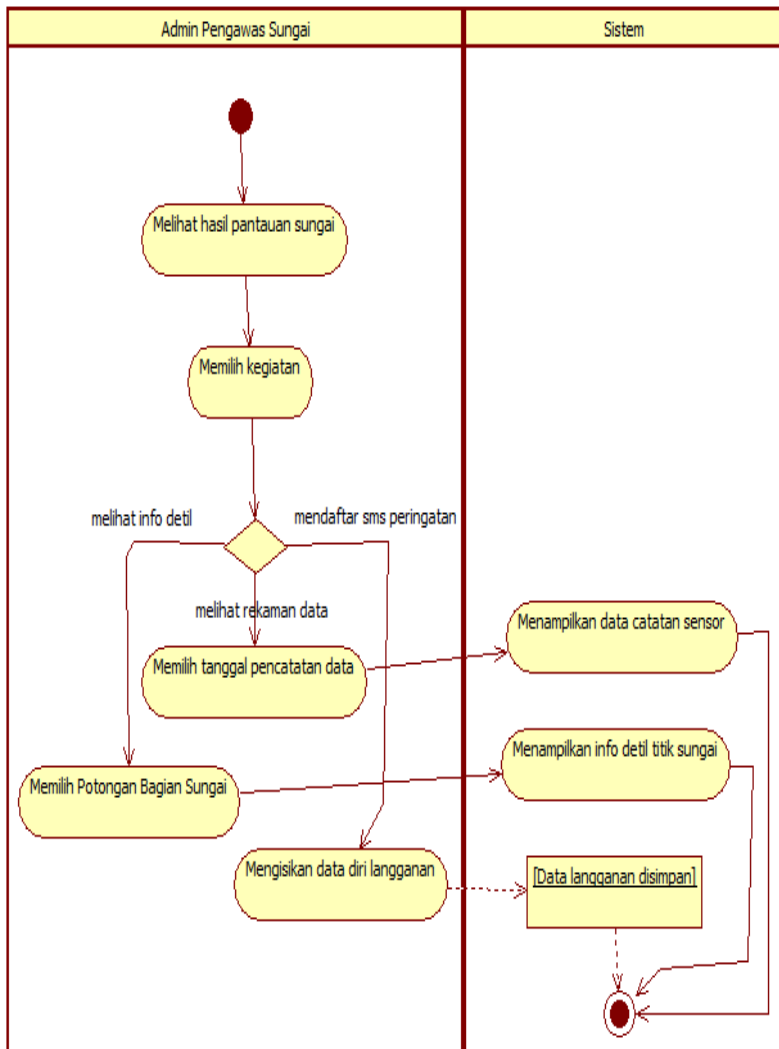
Tabel 3.3 Spesifikasi Kasus Penggunaan: Melihat Hasil Pantauan Sungai

Nama Kasus Penggunaan	Melihat Hasil Pantauan Sungai
Nomor	UC-002
Deskripsi	Pengguna dapat melihat hasil data sensor yang mengawasi kondisi sungai.
Aktor	Admin pengawas sungai.
Kondisi Awal	Pengguna berada pada halaman awal aplikasi.
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan pantauan. 2. Sistem menampilkan antarmuka melihat hasil pemantauan kondisi aliran sungai. 3. Pengguna memilih kegiatan yang dapat dilakukan. <ol style="list-style-type: none"> A1. Pengguna memilih melihat informasi detail bagian potongan sungai. A2. Pengguna memilih melihat laporan data sensor. A3. Pengguna memilih mendaftar langganan SMS peringatan. 4. Kasus penggunaan berakhir.
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih melihat informasi detail bagian potongan sungai. <ol style="list-style-type: none"> 1. Pada antarmuka pantauan sungai, pengguna mengklik titik bagian potongan sungai yang muncul pada panel peta. 2. Sistem menampilkan informasi detail

	<p>mengenai titik potongan bagian sungai tersebut.</p> <ol style="list-style-type: none"> 3. Kembali ke alur normal langkah 4. <p>A2. Pengguna memilih melihat laporan data sungai.</p> <ol style="list-style-type: none"> 1. Pada antarmuka pantauan sungai, pengguna memilih lihat laporan data sensor. 2. Sistem menampilkan antarmuka melihat laporan data sensor. 3. Pengguna memilih potongan bagian sungai dan tanggal rekaman data sensor. 4. Sistem menampilkan data serta grafik rekaman. 5. Kembali ke alur normal langkah 4. <p>A3. Pengguna memilih mendaftar langganan SMS peringatan.</p> <ol style="list-style-type: none"> 1. Pada antarmuka pantauan sungai, pengguna memilih daftar langganan SMS. 2. Sistem menampilkan isian data diri untuk langganan. 3. Pengguna mengisi data diri. 4. Sistem menyimpan data diri pengguna. 5. Kembali ke alur normal langkah 4.
Kondisi Akhir	Sistem menampilkan hasil pantauan sungai.

3.1.5.3 Kasus Penggunaan Menyimulasikan Data Sensor

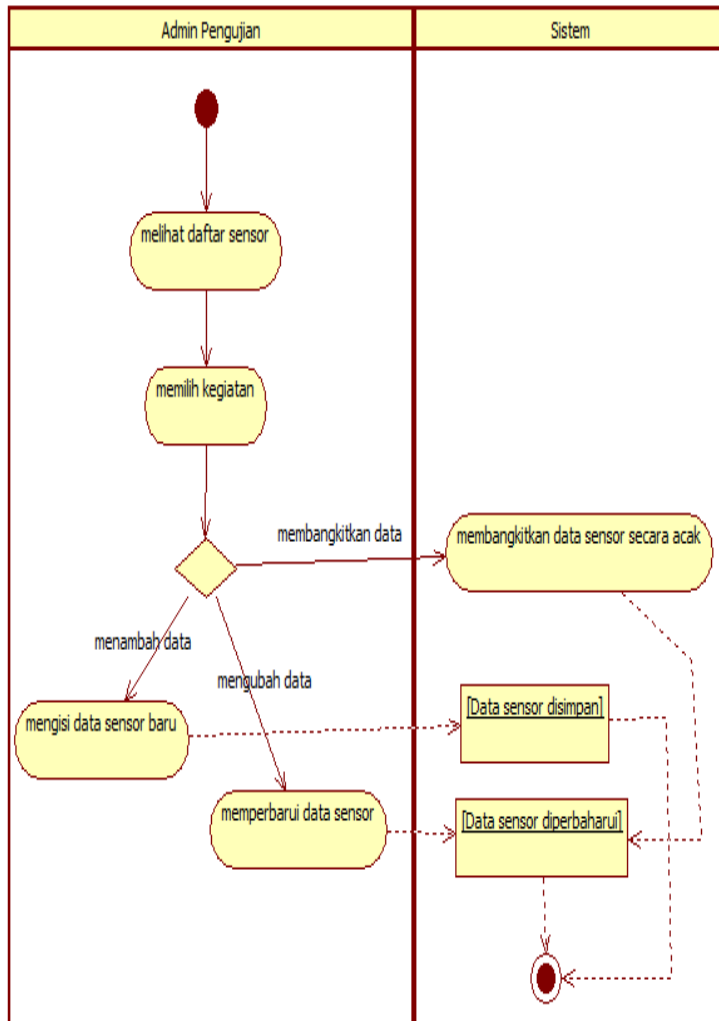
Spesifikasi kasus penggunaan menyimulasikan data sensor pada aplikasi pembangkit data sensor dijelaskan pada Tabel 3.4. Aktivitas yang dilakukan untuk memenuhi kasus penggunaan ini tergambar pada diagram aktivitas Gambar 3.6. Pada kasus penggunaan ini, pengguna dapat menambah dan mengubah data sensor untuk keperluan pengujian.



Gambar 3.5 Diagram Aktivitas Melihat Hasil Pantauan Sungai

Tabel 3.4 Spesifikasi Kasus Penggunaan: Mensimulasikan Data Sensor

Nama Kasus Penggunaan	Mensimulasikan data sensor.
Nomor	UC-003
Deskripsi	Pengguna dapat memanipulasi data masukan sebagai pengganti data sensor.
Aktor	Admin pengujian.
Kondisi Awal	Pengguna berada pada halaman awal aplikasi pembangkit data sensor.
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar sensor. 2. Pengguna memilih kegiatan yang dapat dilakukan. <ol style="list-style-type: none"> A1. Pengguna memilih menambahkan sensor baru. A2. Pengguna memilih mengubah data sensor. A3. Pengguna memilih membangkitkan data sensor secara acak. 3. Kasus penggunaan berakhir.
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih menambahkan sensor baru. <ol style="list-style-type: none"> 1. Pengguna memasukkan data sensor baru. 2. Sistem menyimpan data sensor baru 3. Kembali ke alur normal langkah 3. A2. Pengguna memilih mengubah data sensor. <ol style="list-style-type: none"> 1. Sistem menampilkan isian mengubah data sensor. 2. Pengguna mengubah data sensor. 3. Sistem memperbaharui data sensor. 4. Kembali ke alur normal langkah 3. A3. Pengguna memilih membangkitkan data sensor secara acak. <ol style="list-style-type: none"> 1. Sistem mengacak perubahan data sensor. 2. Sistem memperbaharui data sensor. 3. Kembali ke alur normal langkah 4.
Kondisi Akhir	Sistem menampilkan daftar sensor.



Gambar 3.6 Diagram Kasus Menyimulasikan Data Sensor

3.2 Perancangan Sistem

Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu perancangan data, perancangan kelas, perancangan proses aplikasi dan perancangan antarmuka aplikasi yang akan dibuat.

3.2.1 Perancangan Basis Data

Pada subbab ini dijelaskan tentang rancangan basis data yang akan digunakan pada aplikasi. Basis data pada sistem yang akan dibangun pada Tugas Akhir ini menggunakan *Relational Data Base Management System* (RDBMS) MySQL.

3.2.1.1 Conceptual Data Model

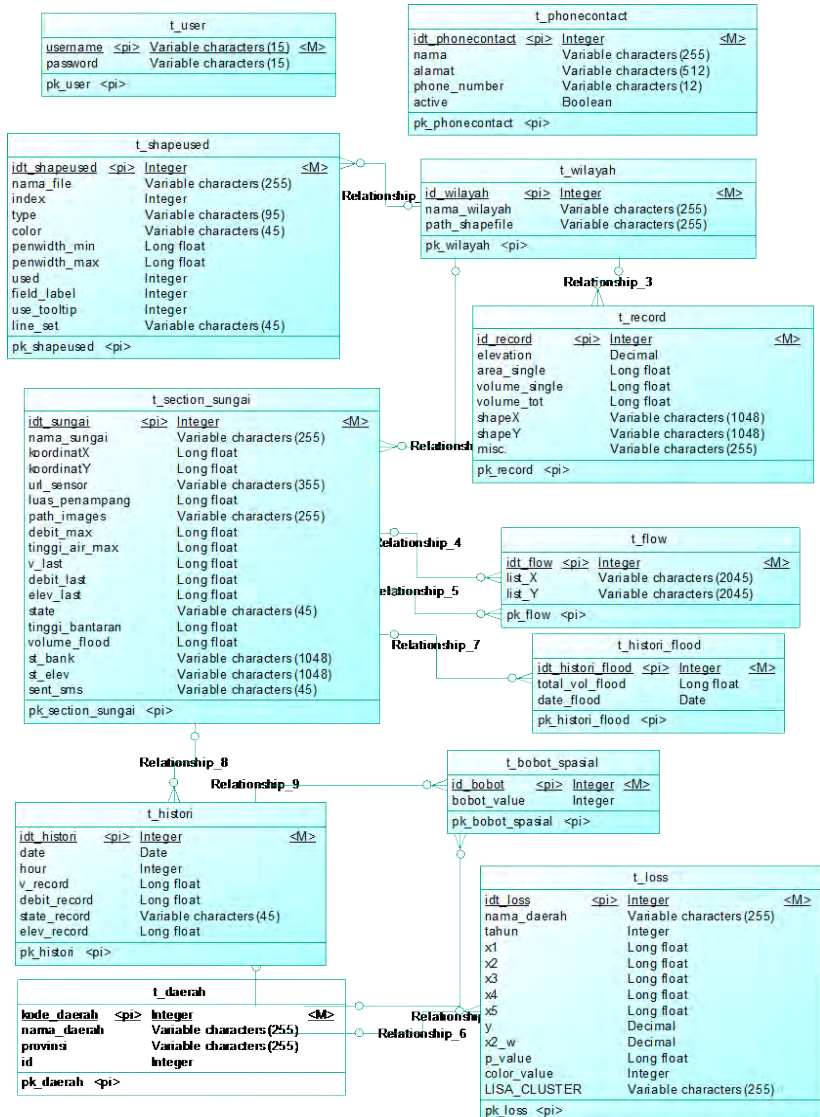
Conceptual Data Model (CDM) merupakan rancangan basis data yang menggambarkan tabel beserta relasinya. CDM pada aplikasi yang dibangun mempunyai beberapa tabel, yaitu: tabel *user*, tabel wilayah, tabel *section_sungai*, tabel *record*, tabel *flow*, tabel *shapeused*, tabel *phonecontact*, tabel *loss*, tabel *histori_flood*, tabel *histori*, tabel daerah, tabel *bobot_spasial*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.7.

3.2.1.1.1 Tabel user

Tabel user digunakan untuk menyimpan data user yang digunakan untuk mengakses aplikasi. Setiap baris pada tabel ini mewakili data *user* yang mempunyai akses penuh pada aplikasi. Atribut dari tabel ini dijelaskan pada Tabel 3.5.

Tabel 3.5 Deskripsi Tabel User

Atribut	Tipe	Null	Default	Keterangan
username	varchar (15)	Tidak		<i>Primary Key</i> (PK)
password	varchar (15)	Ya		Password pengguna



Gambar 3.7 Rancangan Conceptual Data Model Aplikasi

3.2.1.1.2 Tabel wilayah

Tabel wilayah digunakan untuk menyimpan nama-nama wilayah yang dipantau. Setiap baris pada tabel ini mewakili data wilayah yang dipantau. Atribut dari tabel ini dijelaskan pada Tabel 3.6.

Tabel 3.6 Deskripsi Tabel Wilayah

Atribut	Tipe	Null	Default	Keterangan
Id_wilayah	int(11)	Tidak	<i>Auto Increment</i>	<i>Primary Key (PK)</i>
Nama_wilayah	varchar (255)	Tidak		Nama wilayah pantauan.
Path_shapefile	Varchar(255)	Tidak		Direktori yang menyimpan file-file peta.

3.2.1.1.3 Tabel section_sungai

Tabel section_sungai digunakan untuk menyimpan informasi-informasi yang berkaitan dengan bagian sungai yang dipantau pada wilayah yang ada. Setiap baris mewakili data bagian sungai pada wilayah tertentu. Atribut dari tabel ini dijelaskan pada Tabel 3.7.

Tabel 3.7 Deskripsi Tabel Section_Sungai

Atribut	Tipe	Null	Default	Keterangan
Idt_sungai	int(11)	Tidak	<i>Auto Increment</i>	<i>Primary Key (PK)</i>
Id_wilayah	int(11)	Tidak		<i>Foreign Key (FK) tabel wilayah</i>
Nama_sungai	Varchar(255)	Tidak		Nama bagian sungai
KoordinatX	Double	Ya		Koordinat x

				bagian sungai
KoordinatY	Double	Ya		Koordinat y bagian sungai
url_sensor	Varchar(355)	Ya		url yang digunakan untuk mendapatkan data sensor
Luas_penampang	Double	Ya		Luas penampang
Path_images	Varchar(255)	Ya		Direktori penyimpanan gambar
Debit_max	Double	Ya		Debit maksimal
Tinggi_air_max	Double	Ya		Tinggi air maksimal
V_last	Double	Ya		Kecepatan terakhir aliran sungai
Debit_last	Double	Ya		Debit terakhir sungai
Elev_last	Double	Ya		Tinggi muka air sungai terakhir
State	Varchar(45)	Ya		Status bagian sungai
Tinggi_bantaran	Double	Ya		Tinggi bantaran sungai
Volume_flood	Double	Ya		Volume luapan air
St_bank	Varchar(1048)	Ya		Penampang x sungai

St_elev	Varchar(1048)	Ya		Penampang y sungai
Sent_sms	Varchar(45)	Ya		Status pengiriman peringatan

3.2.1.1.4 Tabel record

Tabel record digunakan untuk menyimpan data hasil penghitungan luas area kontur dan volume lengkung kapasitas. Setiap baris data pada tabel mewakili data hasil penghitungan dari masing-masing wilayah. Atribut tabel dijelaskan pada Tabel 3.8.

Tabel 3.8 Deskripsi Tabel Record

Atribut	Tipe	Null	Default	Keterangan
Id_record	int(11)	Tidak	<i>Auto Increment</i>	<i>Primary Key (PK)</i>
Id_wilayah	int(11)	Tidak		<i>Foreign Key (FK) tabel wilayah</i>
Elevation	Decimal(10,2)	Ya		Tinggi elevasi kontur
Area_single	Double	Ya		Luas area satu kontur
Area_tot	Double	Ya		Total luas area kontur suatu wilayah
Volume_single	Double	Ya		Volume pada dua kontur
Volume_tot	Double	Ya		Total volume pada kontur suatu wilayah
shapeX	Varchar(1048)	Ya		List

				koordinat X kontur
shapeY	Varchar(1048)	Ya		List koordinat Y kontur
Misc.	Varchar(255)	Ya		Keterangan

3.2.1.1.5 Tabel flow

Tabel flow digunakan untuk menyimpan data aliran sungai dari satu bagian sungai ke bagian lainnya. Setiap baris pada tabel ini mewakili data aliran sungai pada bagian sungai. Atribut tabel dijelaskan pada Tabel 3.9.

Tabel 3.9 Deskripsi Tabel Flow

Atribut	Tipe	Null	Default	Keterangan
Idt_flow	int(11)	Tidak	<i>Auto Increment</i>	<i>Primary Key (PK)</i>
Flow_from	int(11)	Tidak		<i>Foreign Key (FK) tabel section_sungai</i>
Flow_to	Int(11)	Ya		<i>Foreign Key (FK) tabel section_sungai</i>
List_X	Varchar(2045)	Ya		List koordinat x aliran sungai
List_Y	Varchar(2045)	Ya		List koordinat y aliran sungai

3.2.1.1.6 Tabel shapeused

Tabel shapeused digunakan untuk menyimpan data peta yang digunakan beserta pengaturan tampilannya. Tiap baris pada tabel shapeused mewakili data peta yang digunakan tiap wilayah dan pengaturan tampilannya. Atribut tabel shapeused dijelaskan pada Tabel 3.10.

Tabel 3.10 Deskripsi Tabel Shapeused

Atribut	Tipe	Null	Default	Keterangan
Idt_shapeused	int(11)	Tidak	<i>Auto Increment</i>	<i>Primary Key (PK)</i>
Nama_file	varchar(255)	Ya		Nama file peta
Index	int(11)	Ya		Index layer peta
type	Varchar(95)	Ya		Tipe peta
Color	Varchar(45)	Ya		Warna tampilan peta
Penwidth_min	Double	Ya		Pengaturan garis peta
Penwidth_max	Double	Ya		Pengaturan garis peta
Used	Int(11)	Ya		Status peta digunakan
Field_label	Int(11)	Ya		Teks yang ditampilkan peta
Use_tooltip	Int(11)	Ya		Tampilan peta
Line_set	Varchar(45)	Ya		Pengaturan garis

3.2.1.1.7 Tabel phonecontact

Tabel phonecontact digunakan untuk menyimpan data orang yang berlangganan peringatan SMS. Atribut tabel ini dijelaskan pada Tabel 3.11.

Tabel 3.11 Deskripsi Tabel Phonecontact

Atribut	Tipe	Null	Default	Keterangan
Idt_phonecontact	int(11)	Tidak	<i>Auto Increment</i>	<i>Primary Key (PK)</i>
Nama	varchar(255)	Ya		Nama Pengguna

Alamat	int(11)	Ya		Alamat pengguna
Phone_number	Varchar(95)	Ya		Nomor telepon pengguna
Active	Varchar(45)	Ya		Status langganan

3.2.1.1.8 Tabel histori

Tabel histori digunakan untuk menyimpan rekaman data yang didapat dari sensor per jam selama aplikasi aktif. Atribut tabel dijelaskan pada Tabel 3.12.

Tabel 3.12 Deskripsi Tabel Histori

Atribut	Tipe	Null	Default	Keterangan
Idt_histori	int(11)	Tidak	<i>Auto Increment</i>	<i>Primary Key (PK)</i>
date	date	Ya		Tanggal perekaman
hour	int(11)	Ya		Jam perekaman
V_record	double	Ya		Kecepatan aliran terekam
Debit_record	double	Ya		Debit terekam
State_record	double	Ya		Status bagian sungai
Elev_record	double	Ya		Tinggi muka air terekam

3.2.1.1.9 Tabel histori_flood

Tabel histori_flood digunakan untuk menyimpan data volume luapan air pada bagian sungai yang meluap pada suatu waktu. Atribut tabel dijelaskan pada Tabel 3.13.

Tabel 3.13 Deskripsi Tabel Histori_flood

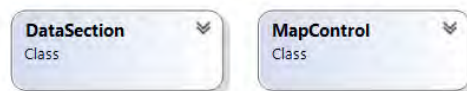
Atribut	Tipe	Null	Default	Keterangan
Idt_histori_flood	int(11)	Tidak	<i>Auto Increment</i>	<i>Primary Key (PK)</i>
date	Datetime	Ya		Tanggal terjadinya luapan
Total_vol_flood	Double	Ya		Total volume luapan air

3.2.1.2 Physical Data Model

Gambar 3.11 merupakan *Physical Data Model* (PDM) dari aplikasi sistem informasi geografis peringatan dini yang akan dibangun. PDM ini juga menggambarkan tabel yang digunakan beserta relasinya.

3.2.2 Perancangan Diagram Kelas

Pada subbab ini akan dijelaskan mengenai perancangan diagram kelas yang disertai operasi yang akan digunakan dalam logika bisnis aplikasi. Perancangannya dapat dilihat pada Gambar 3.8.



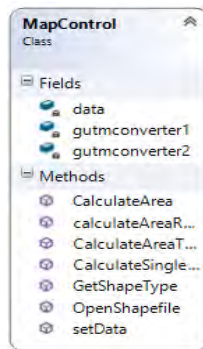
Gambar 3.8 Rancangan Diagram Kelas Aplikasi

3.2.2.1 Kelas MapControl

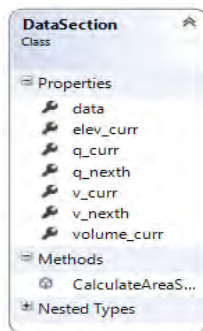
Kelas MapControl merupakan sebuah kelas yang digunakan untuk menampilkan peta pada aplikasi. Kelas MapControl dapat dilihat pada Gambar 3.9.

3.2.2.2 Kelas DataSection

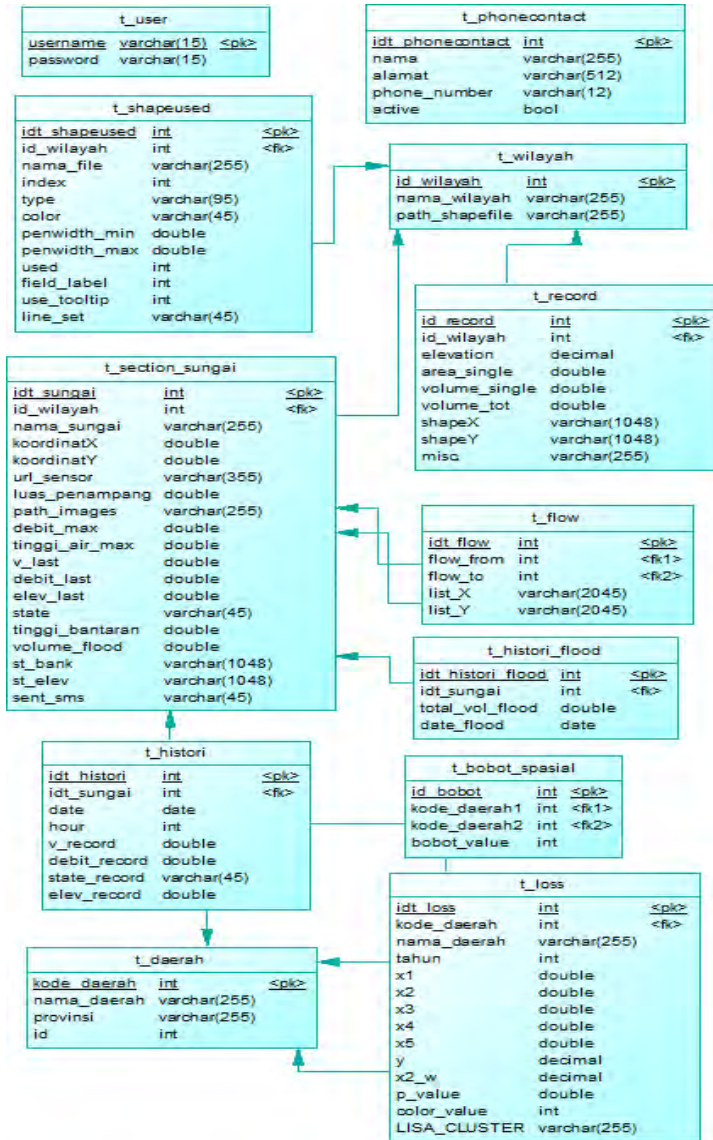
Kelas DataSection merupakan kelas yang digunakan untuk menyimpan data debit, kecepatan, dan ketinggian air dari suatu bagian sungai. Kelas DataSection dapat dilihat pada Gambar 3.10.



Gambar 3.9 Rancangan Diagram Kelas MapControl



Gambar 3.10 Rancangan Diagram Kelas DataSection



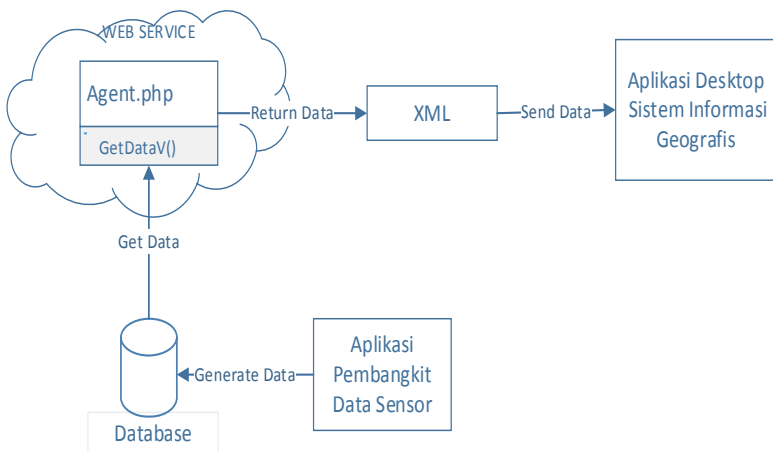
Gambar 3.11 Rancangan *Physical Data Model* Aplikasi

3.2.3 Perancangan Layanan Web

Pada Tugas Akhir ini akan dibuat layanan *web* yang digunakan untuk mengirimkan data sensor buatan. Arsitektur dari layanan *web* yang dibuat dapat dilihat pada Gambar 3.12.

Layanan *web* dibangun menggunakan pendekatan arsitektur *1-tier* dimana semua kode (SQL dan PHP) diletakkan pada satu tempat, yaitu di bagian *user interface*. Data yang dikirimkan berformat XML. Layanan *web* yang dibangun memiliki sebuah fungsi bernama `GetDataV()` yang digunakan untuk mengambil data sensor buatan dari basis data dan mengolahnya menjadi format XML.

Layanan *web* ini menggunakan parameter `HTTP_GET`. Layanan *web* menggunakan dua parameter untuk dapat diakses yaitu parameter mode dan id sensor. Parameter mode adalah aksi yang dilakukan oleh layanan *web* yaitu mengambil data sensor buatan. Parameter id sensor adalah id dari sensor buatan yang akan diambil datanya.



Gambar 3.12 Arsitektur Layanan Web Pengirim Data Sensor Buatan

Data sensor buatan yang dikirimkan layanan *web* berformat XML. Struktur dari XML yang dikirimkan oleh layanan web dapat dilihat pada Tabel 3.14. Data yang dikirimkan adalah data kecepatan, data tinggi muka air dan tanggal pengiriman pada data sensor buatan.

Tabel 3.14 Struktur XML Pengiriman Data Sensor Buatan

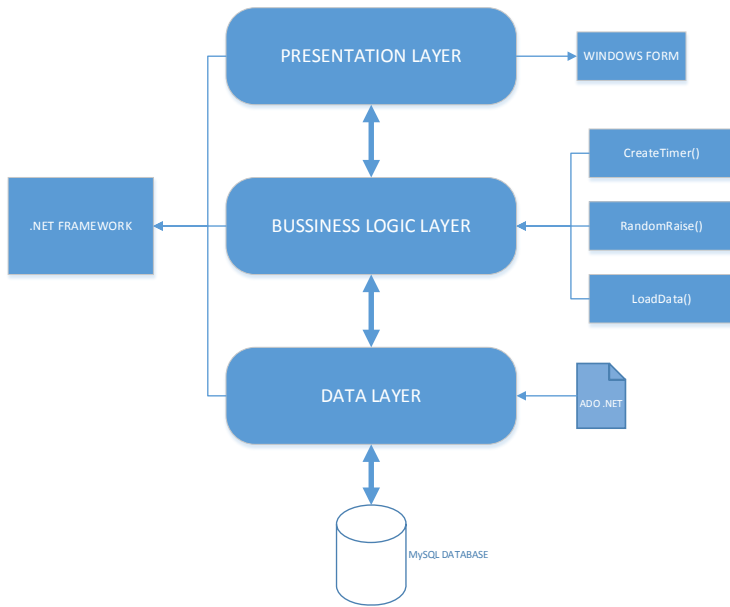
```
<data>
<velocity> data_velocity </velocity>
<elevation> data_elevation </elevation>
<date> date_now </date>
</data>
```

3.2.4 Perancangan Aplikasi Pembangkit Data Sensor

Pada Tugas Akhir ini akan dibuat sebuah aplikasi pembangkit data sensor yang akan digunakan pada tahap pengujian untuk menyimulasikan data sensor. Arsitektur dari aplikasi pembangkit data sensor yang dibangun dapat dilihat pada Gambar 3.13.

Aplikasi pembangkit data sensor dibangun menggunakan pendekatan arsitektur *3-tier* dimana arsitektur aplikasi dibagi menjadi tiga *layer*. *Presentation layer* yaitu lapisan pada aplikasi yang digunakan untuk menampilkan data. Data ditampilkan pada antarmuka *windows form*. *Bussiness logic layer* yaitu lapisan pada aplikasi yang digunakan untuk memproses data dan menjalankan fungsionalitas aplikasi. Pada *layer* ini terdapat fungsi-fungsi yaitu *CreateTimer()*, *RandomRaise()* dan *LoadData()*. Fungsi *CreateTimer()* digunakan untuk membuat alat penghitung waktu acakan data sensor. Fungsi *RandomRaise()* digunakan untuk mengacak data kenaikan sensor buatan. Fungsi *LoadData()* digunakan untuk menampilkan data pada antarmuka.

Data layer yaitu lapisan pada aplikasi yang digunakan untuk manajemen data yang mengalir dari atau ke basis data. ADO.NET *Entity Framework* digunakan pada *data layer* untuk manajemen data yang mengalir dari aplikasi ke basis data dan sebaliknya.



Gambar 3.13 Arsitektur Aplikasi Pembangkit Data Sensor

3.2.5 Perancangan Proses Aplikasi

Pada subbab ini akan dijelaskan mengenai rancangan proses aplikasi yang digunakan untuk pencapaian suatu fungsi pada aplikasi.

3.2.5.1 Rancangan Proses Mengelola Wilayah Pantauan

Proses mengelola wilayah pantauan merupakan proses awal dimana admin pengawas sungai mengisikan data-data yang diperlukan agar aplikasi dapat menjalankan fungsinya secara benar. Urutan dari proses ini adalah sebagai berikut.

1. Pengguna berada pada antarmuka beranda.
2. Pengguna *login* agar mendapatkan akses penuh terhadap aplikasi.
3. Pengguna menambahkan wilayah baru.

4. Sistem menampilkan antar muka mengelola data wilayah.
5. Pengguna menambahkan peta yang bertipe shapefile, data sungai, data aliran dan data luas dan volume kontur yang diperlukan.
6. Sistem menyimpan semua data yang telah dimasukkan.

3.2.5.2 Rancangan Proses Melihat Hasil Pantauan

Proses melihat hasil pantauan merupakan proses yang dilakukan untuk mendapatkan informasi yang didapatkan dari sensor. Proses ini dilakukan setelah data wilayah pantauan lengkap. Langkah-langkah dari proses ini adalah sebagai berikut.

1. Pengguna berada pada antarmuka beranda.
2. Pengguna memilih pilihan monitoring.
3. Sistem menampilkan antarmuka melihat hasil pantauan.
4. Pengguna memilih potongan bagian sungai.
5. Sistem menampilkan informasi detail potongan sungai.

3.2.5.3 Rancangan Proses Mengirimkan SMS Peringatan

Proses mengirimkan SMS peringatan merupakan proses yang dilakukan aplikasi untuk mengolah data masukan dari sensor untuk memberikan peringatan ke pengguna. Status peringatan dapat dilihat pada Tabel 3.15.

Tabel 3.15 Status pesan peringatan

Status	Keterangan Kondisi Sungai
Siaga I	Tinggi muka air di atas 66% dari tinggi muka air maksimal dan mendekati tinggi bantaran
Siaga II	Tinggi muka air sekitar 34%-66% antara tinggi muka air maksimal dan tinggi bantaran
Siaga III	Tinggi muka air sekitar 0%-33% antara tinggi muka air maksimal dan tinggi bantaran

Langkah-langkah dari proses ini adalah sebagai berikut.

1. Sistem menerima data sensor.
2. Sistem mengambil data potongan bagian sungai dari basis data.
3. Sistem membandingkan data tinggi muka air yang didapat dari sensor sekarang dengan data tinggi muka air maksimal yang sudah disimpan sebelumnya.
4. Jika data tinggi muka air sekarang melebihi data tinggi muka air maksimal, maka sistem mengirimkan pesan peringatan.
5. Jika tidak, data tinggi muka air disimpan ke penyimpanan sementara.
6. Jika data tinggi muka air telah tersimpan selama 10 menit, maka dihitung tinggi rata-ratanya. Kemudian data rata-rata kenaikan tinggi diakumulasi selama 2 jam dan ditambahkan ke data tinggi air sekarang.
7. Sistem membandingkan data tinggi muka air 2 jam dengan tinggi muka air maksimal. Jika tinggi air 2 jam melebihi tinggi muka air maksimal, maka sistem akan mengirimkan pesan peringatan.

3.2.5.4 Rancangan Proses Menghitung Luas Area Kontur

Proses menghitung luas area kontur merupakan proses yang digunakan oleh aplikasi untuk mengolah data kontur wilayah agar dapat memperkirakan tinggi genangan luapan air. Langkah-langkah dari proses ini adalah sebagai berikut.

1. Sistem mengambil data kontur pada suatu elevasi.
2. Sistem menggunakan fungsi metode koordinat untuk menghitung luas area kontur
3. Sistem menyimpan data luas area ke basis data.

3.2.5.5 Rancangan Proses Menghitung Volume Area Kontur

Proses menghitung volume area kontur merupakan proses yang digunakan oleh aplikasi untuk mengolah data luas area kontur sehingga didapat volume pada dua area kontur agar dapat memperkirakan tinggi genangan air. Langkah-langkah dari proses ini adalah sebagai berikut.

1. Sistem mengambil data luas area kontur i dan $i+1$ pada basis data.
2. Sistem menggunakan fungsi metode kontur untuk mendapatkan volume area dari kontur i dan $i+1$.
3. Sistem menyimpan data volume kontur ke basis data.

3.2.5.6 Rancangan Proses Memperkirakan Tinggi Genangan Air

Proses menghitung tinggi genangan air merupakan proses yang dilakukan aplikasi untuk mengolah data luas dan volume kontur untuk mengetahui tinggi genangan air yang meluap pada sungai. Langkah-langkah dari proses ini adalah sebagai berikut.

1. Sistem menerima data sensor.
2. Jika data tinggi muka air suatu potongan sungai melebihi data tinggi muka air maksimal, maka sistem menghitung volume luapan air dari potongan bagian sungai tersebut.
3. Sistem mengambil data volume kontur wilayah pantauan berdasarkan besar volume luapan air.
4. Sistem menampilkan tinggi genangan air.

3.2.5.7 Rancangan Proses Menyimulasikan Data Sensor

Proses menyimulasikan data sensor merupakan proses yang dilakukan untuk membangkitkan data sensor agar dapat

digunakan untuk pengujian fungsionalitas aplikasi SIG. Langkah-langkah dari proses ini adalah sebagai berikut.

1. Pengguna berada pada antarmuka awal.
2. Pengguna mengisi data sensor baru.
3. Sistem menyimpan data sensor baru.
4. Pengguna menjalankan fungsi pembangkitan data sensor.
5. Sistem membangkitkan data sensor secara acak.

3.2.6 Perancangan Antarmuka

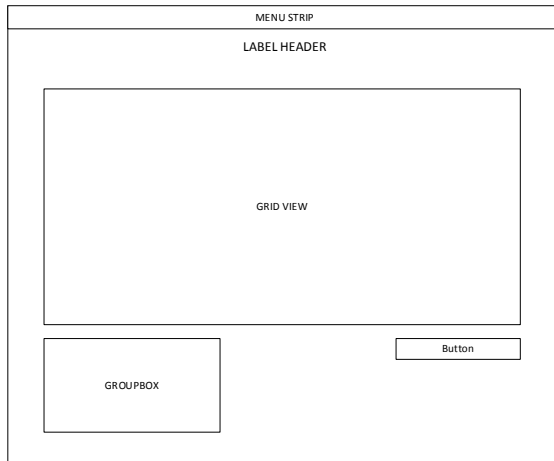
Pada subbab ini dibahas mengenai rancangan antarmuka yang digunakan untuk memfasilitasi pengguna dalam berinteraksi dengan fitur-fitur perangkat lunak yang ada. Berikut ini dijabarkan beberapa rancangan antarmuka yang ada.

3.2.6.1 Rancangan Antarmuka Halaman Utama

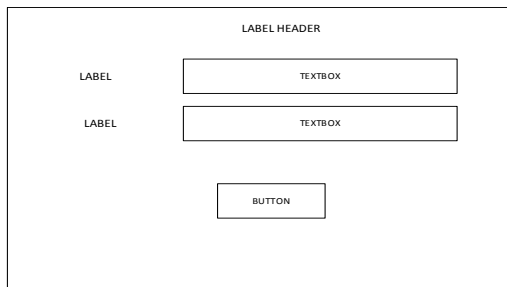
Gambar 3.14 merupakan gambar rancangan antarmuka untuk halaman utama. Pada antarmuka ini digunakan untuk mengelola data wilayah. Pengguna dapat menambah, mengubah dan menghapus data wilayah. Antarmuka halaman utama juga terdapat *menu* untuk menampilkan antarmuka *login* pengguna dan mengubah *username* dan *password* pengguna.

3.2.6.2 Rancangan Antarmuka *Login*

Gambar 3.15 merupakan gambar rancangan antarmuka untuk *login*. Pengguna melakukan *login* agar dapat mendapatkan akses penuh terhadap aplikasi, sehingga pengguna dapat mengelola data wilayah. Pengguna nantinya mengisi *username* dan *password* yang valid agar dapat *login* ke dalam aplikasi. Secara *default*, *username* dan *password* pengguna untuk mengakses aplikasi adalah “admin”.



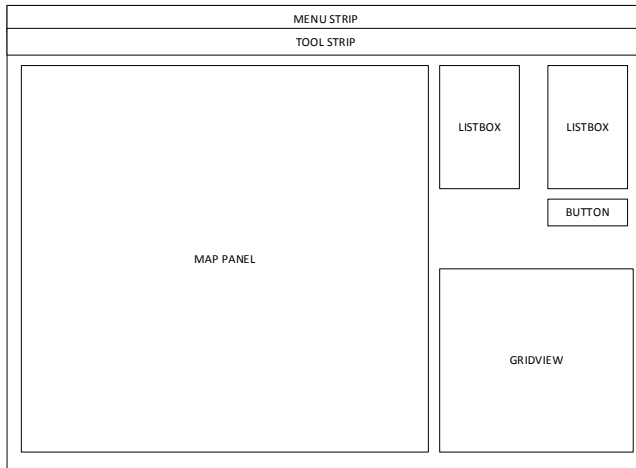
Gambar 3.14 Rancangan Antarmuka Halaman Utama



Gambar 3.15 Rancangan Antarmuka Halaman *Login*

3.2.6.3 Rancangan Antarmuka Mengelola Data Wilayah

Gambar 3.16 merupakan gambar rancangan antarmuka untuk mengelola data-data yang berkaitan dengan data wilayah seperti data sungai, peta, data luas area dan volume dan lain-lain. Pada antarmuka ini terdapat *toolstrip* yang digunakan untuk menambah data sungai, data aliran, serta untuk menghitung luas area.



Gambar 3.16 Rancangan Antarmuka Mengelola Data Wilayah

3.2.6.4 Rancangan Antarmuka Mengelola Data Sungai

Gambar 3.17 merupakan gambar rancangan antarmuka untuk mengelola data sungai. Pada antarmuka ini terdapat kolom-kolom teks yang diisi untuk melengkapi data profil sungai yang digunakan dalam aplikasi. Serta terdapat panel yang digunakan untuk menggambar data penampang.

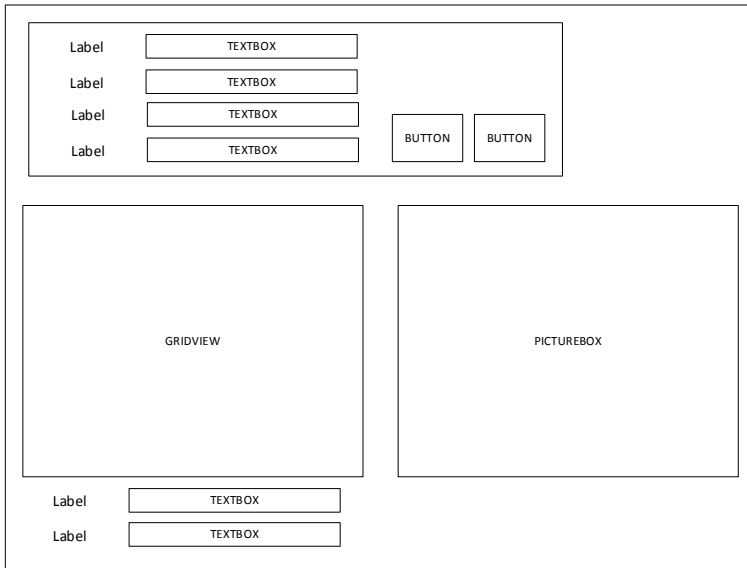
3.2.6.5 Rancangan Antarmuka Mengelola Data Aliran

Gambar 3.18 merupakan gambar rancangan antarmuka untuk mengelola data aliran. Pada antarmuka ini terdapat *combobox* yang berisi daftar profil sungai. Pengguna memilih aliran sungai dari satu profil sungai ke profil sungai lain.

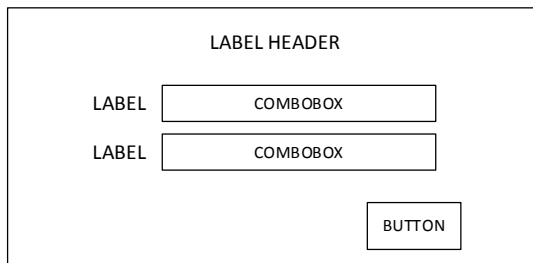
3.2.6.6 Rancangan Antarmuka Pemantauan Kondisi Sungai

Gambar 3.19 merupakan gambar rancangan antarmuka untuk mengelola data aliran. Pada antarmuka ini, pengguna dapat

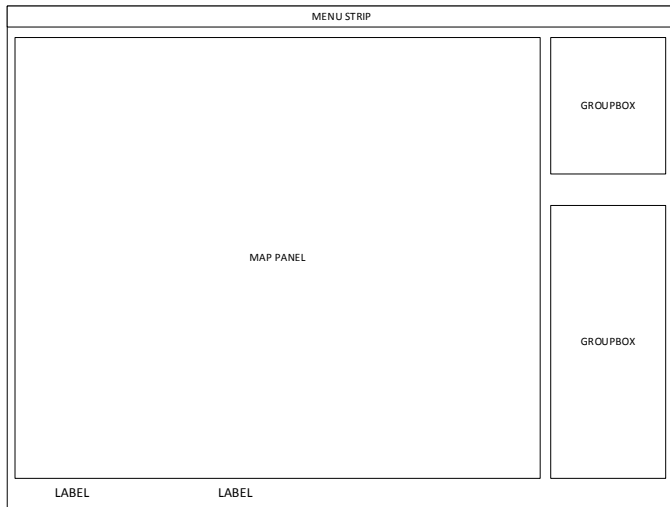
melihat hasil yang didapat dari sensor dan ditampilkan. Pada antarmuka ini ditunjukkan posisi atau lokasi dari profil sungai yang dipantau pada panel peta.



Gambar 3.17 Rancangan Antarmuka Mengelola Data Sungai



Gambar 3.18 Rancangan Antarmuka Mengelola Data Aliran



Gambar 3.19 Rancangan Antarmuka Pemantauan Kondisi Sungai

3.2.6.7 Rancangan Antarmuka Melihat Laporan Data Sungai

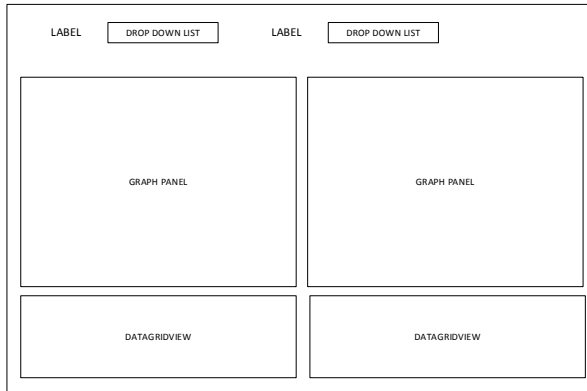
Gambar 3.20 merupakan gambar rancangan antarmuka untuk melihat rekaman dari data kecepatan aliran dan ketinggian muka air pada suatu bagian sungai. Data dikategorikan berdasarkan tanggal yang terekam. Data direkam per satu jam oleh aplikasi.

3.2.6.8 Rancangan Antarmuka Detil Profil Sungai

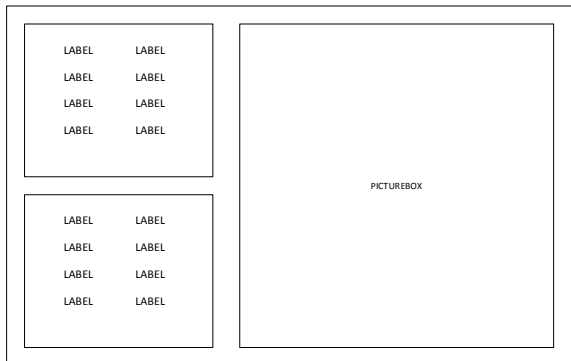
Gambar 3.21 merupakan gambar rancangan antarmuka yang menampilkan detil suatu profil sungai. Pada antarmuka ini, diperlihatkan informasi-informasi yang berhubungan dengan suatu profil sungai. Pada antarmuka ini juga digambarkan bentuk penampang dari profil sungai tersebut.

3.2.6.9 Rancangan Antarmuka Aplikasi Pembangkit Data Sensor

Gambar 3.22 merupakan gambar rancangan antarmuka pada aplikasi pembangkit data sensor. Pada antarmuka ini pengguna dapat membuat data sensor baru, mengubah dan menghapus data sensor lama. Antarmuka ini juga mempunyai tombol yang dapat membangkitkan data sensor.



Gambar 3.20 Rancangan Antarmuka Melihat Laporan Data Sungai



Gambar 3.21 Rancangan Antarmuka Detil Profil Sungai

The diagram illustrates a user interface for a data entry application. It consists of a main container with a header section and a data display section. The header section contains four labels, each followed by a text box, and two buttons. The data display section is a large rectangular area labeled 'GRIDVIEW'.

LABEL	TEXTBOX	LABEL	TEXTBOX	BUTTON	BUTTON
LABEL	TEXTBOX	LABEL	TEXTBOX	GRIDVIEW	

Gambar 3.22 Rancangan Antarmuka Aplikasi Pembangkit Data Sensor

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Di dalamnya mencakup proses penerapan dan pengimplementasian data, proses, antarmuka dan kelas yang mengacu pada rancangan yang dibahas sebelumnya.

4.1 Lingkungan Pembangunan

Dalam membangun aplikasi ini digunakan beberapa perangkat pendukung baik perangkat keras maupun perangkat lunak. Lingkungan pembangunan dijelaskan sebagai berikut.

4.1.1 Lingkungan Pembangunan Perangkat Keras

Perangkat keras yang dipakai dalam pembuatan aplikasi ini memiliki spesifikasi sebagai berikut.

- Prosesor Intel(R) Core™ i5 CPU 3317U @1,70GHz.
- Memori (RAM) 4,00 GB.

4.1.2 Lingkungan Pembangunan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk membuat aplikasi ini yakni sebagai berikut.

- Sistem operasi Windows 7 Ultimate (64 bit).
- Microsoft Visual Studio 2012.
- Pustaka Easy GIS .NET
- MySQL Server 5.0.
- Gammu SMS Gateway
- Bahasa pemrograman C#

4.2 Implementasi Layanan *Web* Pengirim Data Sensor

Pada tugas akhir ini, data sensor buatan dikirimkan melalui layanan *web*. Implementasi dari layanan *web* pengiriman data sensor dapat dilihat pada Kode Sumber 4.1.

```

<?php
// koneksi ke database di sistem
$conn = mysql_connect("localhost", "andre", "awerty");
mysql_select_db("agentdb");
if(! $conn ) { die('Could not connect: ' . mysql_error()); }
else {
$mode = $_GET['mode'];
if($mode == "getDataV")
{$name = $_GET['name'];
$query = "SELECT * FROM t_agent WHERE agent_name = '$name'";
$hasil = mysql_query($query);
$data = mysql_fetch_array($hasil);
$velocity = $data['velocity'];
$elevation = $data['elevation'];
// membuat header dokumen XML
header('Content-Type: text/xml');
echo "<?xml version='1.0'?>";
// membuat tag data respon pada dokumen XML
echo "<data>";
echo "<velocity>".$velocity."</velocity>";
echo "<elevation>".$elevation."</elevation>";
echo "<date>".date("Y-m-d H:i:s")."</date>";
echo "</data>"; } }
$xml .= "</root>";
header ("Content-Type:text/xml");
echo $xml; } else { echo "Error"; }

```

Kode Sumber 4.1 Implementasi Layanan *Web* Pengirim Data Sensor Buatan

4.3 Implementasi Kelas

Pada subbab ini akan dibahas mengenai implementasi seluruh diagram kelas yang sebelumnya telah dilakukan analisis dan perancangan di subbab 3.2.2.

4.3.1 Kelas MapControl

Implementasi dari kelas MapControl yang merupakan kelas yang berisi fungsi untuk menampilkan peta dan fungsi untuk menghitung luas area. Implementasi Kelas MapControl dapat dilihat pada Kode Sumber 4.2.

```

class MapControl
{
    GeoUTMConverter gutmconverter1 = new GeoUTMConverter();
    GeoUTMConverter gutmconverter2 = new GeoUTMConverter();
    List<LOSS> data;

    public void OpenShapefile(string path, SFMap sfMap1, int
index, string control)
    {
        try
        {
            sfMap1.AddShapeFile(path, "ShapeFile", "");
            EGIS.ShapeFileLib.ShapeFile sf = sfMap1[index];
            sf.RenderSettings.Font = new Font(sfMap1.Font.FontFamily,
12);
            sf.RenderSettings.MaxPixelPenWidth = 1;

            if (Path.GetFileName(path) == "Java_District.shp" ||
Path.GetFileName(path) == "jawa2.shp")
                {
                    sf.RenderSettings.FieldName =
sf.RenderSettings.DbfReader.GetFieldNames()[0];
                    sf.RenderSettings.MinPixelPenWidth = 3;
                    sf.RenderSettings.FillColor = Color.LawnGreen;
                }
        }
        catch { MessageBox.Show("file used by another
process"); }
    }

    public double CalculateArea(ReadOnlyCollection<PointD[]>
_vertices)
    {
        double totalArea = 0;
        for (int ish = 0; ish < _vertices.Count; ish++)
        {
            double sumArea = 0;
            for (int i = 0, j = i + 1; i <

```

```

_vertices[ish].Length; i++, j++)
    {
        if (i == _vertices[ish].Length - 1)
        {
            gutmconverter1.ToUTM(_vertices[ish][i].Y, _vertices[ish][i].X);
            gutmconverter2.ToUTM(_vertices[ish][0].Y, _vertices[ish][0].X);

            sumArea = sumArea + ((gutmconverter2.X *
            gutmconverter1.Y) - (gutmconverter1.X * gutmconverter2.Y));
        }
        else
        {
            gutmconverter1.ToUTM(_vertices[ish][i].Y, _vertices[ish][i].X);
            gutmconverter2.ToUTM(_vertices[ish][j].Y, _vertices[ish][j].X);

            sumArea = sumArea +
            ((gutmconverter2.X * gutmconverter1.Y) - (gutmconverter1.X *
            gutmconverter2.Y));
        }
        sumArea = sumArea / 2;
        totalArea = totalArea + sumArea;
    }
    if (totalArea < 0)
    {
        totalArea = totalArea * (-1);
    }
    return totalArea;
}

```

Kode Sumber 4.2 Implementasi Kelas MapControl

4.3.2 Kelas DataSection

Kelas DataSection merupakan kelas yang digunakan sebagai penyimpan sementara data keterangan bagian sungai seperti data kecepatan dan tinggi muka air. Implementasi dari kelas DataSection dapat dilihat pada Kode Sumber 4.3.

```

using System;
using System.Collections.Generic;
using System.Linq;

```

```
using System.Text;
using System.Threading.Tasks;

namespace TAV1._0._0.Class
{
    public class DataSection
    {
        public double v_curr { get; set; }
        public t_section_sungai data { get; set; }
        public double v_nexth { get; set; }
        public double q_nexth { get; set; }
        public double q_curr { get; set; }
        public double elev_curr { get; set; }
        public double volume_curr { get; set; }

        public class crosspoint
        {
            public double X { get; set; }
            public double Y { get; set; }
        }

        public class DataPenampang
        {
            public int id { get; set; }
            public List<crosspoint> dp_all { get; set; }
            public List<crosspoint> dp_basah { get; set; }
        }

        public double CalculateAreaSection(List<crosspoint>
testcs)
        {
            double tot = 0;

            for (int i = 0, j = i + 1; i < testcs.Count; i++, j++)
            {
                if (i == testcs.Count - 1)
                {
                    tot = tot + ((testcs[0].X * testcs[i].Y) -
(testcs[i].X * testcs[0].Y));
                }
                else
                {
                    tot = tot + ((testcs[j].X * testcs[i].Y) -
(testcs[i].X * testcs[j].Y));
                }
            }

            tot = tot / 2;
        }
    }
}
```

```

        tot = Math.Abs(tot);
        return tot;
    }
}

```

Kode Sumber 4.3 Implementasi Kelas DataSection

4.4 Implementasi Aplikasi Pembangkit Data Sensor

Dalam subbab ini akan dibahas tentang implementasi aplikasi yang digunakan untuk membangkit data sensor. Aplikasi mengacak status dan data kenaikan untuk data sensor yang dibuat. Status yaitu naik atau turunnya kondisi tinggi muka air dan kecepatan. Data kenaikan yaitu data acak nantinya ditambahkan ke data tinggi muka air dan kecepatan. Waktu pengacakan data kenaikan secara *default* adalah satu detik. Data sensor yang didapat dari agen ini akan digunakan untuk pengujian. Implementasi agen pembangkit data sensor dapat dilihat pada Kode Sumber 4.4.

```

_timer.Stop();
    try
    {
        if (params1 == "High")
        {
            _raise = GetRandomRaise(0.0001,0.01);
            _raise_velo = GetRandomRaise(0.000001, 0.0009);
        }
        else if (params1 == "Low")
        {
            _raise = GetRandomRaise(0.00001, 0.0009);
            _raise_velo = GetRandomRaise(0.0000001, 0.000001);
        }

        agentdbEntities agentdb = new agentdbEntities();
        t_agent data = agentdb.t_agent.Single(x => x.id == _id);
        if (modes == 1)
        {
            data.velocity = data.velocity + _raise_velo;
            data.elevation = data.elevation + _raise;
            data.state = "Raise";
            data.raise = _raise;
        }
    }
}

```

```

    }
    else if (modes == 2)
    {
        if (data.velocity - _raise_velo > 0)
        {
            data.velocity = data.velocity - _raise_velo;
            data.state = "Down";
            data.raise = _raise;
        }

        if (data.elevation - _raise > 0)
        {
            data.elevation = data.elevation - _raise;
        }
    }
    else
    {
        data.state = "None";
        data.raise = 0;
    }
    agentdb.SaveChanges();

    agf.dataGridView1.Rows[rowindex].Cells[2].Value = data.velocity;
    agf.dataGridView1.Rows[rowindex].Cells[6].Value = data.elevation;
    agf.dataGridView1.Rows[rowindex].Cells[5].Value = data.state;
    agf.dataGridView1.Rows[rowindex].Cells[3].Value = data.raise;
    }
    catch (Exception ex)
    { MessageBox.Show(ex.ToString()); }
    finally
    {
        _timer.Start();
    }
}

```

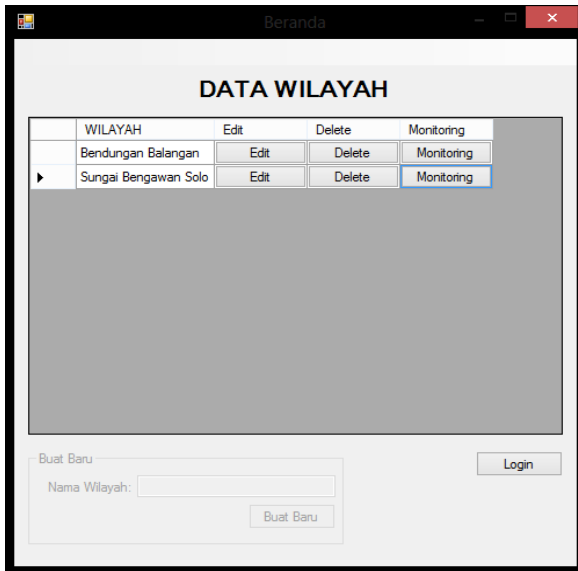
Kode Sumber 4.4 Implementasi Agen Pembangkit Data Sensor

4.5 Implementasi Antarmuka Dan Proses Aplikasi

Dalam subbab ini akan dibahas tentang implementasi antarmuka dan juga proses yang ada pada antarmuka tersebut dari analisis dan perancangan perangkat lunak dalam bentuk potongan gambar antarmuka dan kode sumber implementasi fungsi pada aplikasi, seperti yang telah dibahas pada subbab 3.2.4.

4.5.1 Implementasi Antarmuka Halaman Utama

Pada antarmuka ini, pengguna dapat mengelola data wilayah pantauan. Antarmuka mempunyai beberapa tombol yaitu, tombol *edit*, *delete*, *monitoring*, dan *login*. Antarmuka halaman utama digambarkan pada Gambar 4.1. Pada antarmuka ini terdapat beberapa proses.



Gambar 4.1 Implementasi Antarmuka Halaman Utama

4.5.1.1 Proses Menampilkan Antarmuka *Login*

Proses ini digunakan untuk menampilkan antarmuka *login* yang digunakan untuk memasukkan *username* dan *password*. Setelah *username* dan *password* dimasukkan, sistem mengambil data tersebut dan mencocokkannya ke dalam basis data. Implementasi proses ini dapat dilihat pada Kode Sumber 4.5.


```

private void button2_Click(object sender, EventArgs e)
{
    if (!logged)
    {
        LoginForm lf = new LoginForm();
        lf.sf = this;
        lf.mode = "login";
        lf.Show();
    }
    else
    {
        logged = false;
        changeState();
    }
}

public void changeState()
{
    if (logged)
    {
        button2.Text = "Log Out";
        phoneAccountToolStripMenuItem1.Visible = true;
        groupBox1.Enabled = true;
        ubahPasswordToolStripMenuItem.Visible = true;
    }
    else
    {
        button2.Text = "Login";
        phoneAccountToolStripMenuItem1.Visible = false;
        groupBox1.Enabled = false;
        ubahPasswordToolStripMenuItem.Visible = false;
    }
}

```

**Kode Sumber 4.5 Implementasi Proses Menampilkan Antarmuka
*Login***

4.5.1.2 Proses Menambah Data Wilayah Pantauan

Proses ini digunakan untuk menambah data wilayah pantauan ke basis data. Pengguna kemudian memasuki proses selanjutnya untuk melengkapi data wilayah pantauan. Implementasi menambah data wilayah pantauan dapat dilihat pada Kode Sumber 4.6.

```

private void button1_Click(object sender, EventArgs e)
{
    if (logged)
    {
        if (textBox1.Text != "")
        {
            try
            {
                string namaWilayah = textBox1.Text.Trim();
                var data = (from i in
tadbentities.t_wilayah
                where i.nama_wilayah == namaWilayah
                select i).ToList();

                if (data.Count == 0 || data == null)
                {
                    wilayahCRUD.insertWilayah(textBox1.Text);
                    SungaiCRUD sdc = new SungaiCRUD();
                    t_wilayah NewDataWilayah =
tadbentities.t_wilayah.Single(x => x.nama_wilayah ==
textBox1.Text);

                    loadDGV();
                    sdc.InsertDataSungai(NewDataWilayah.id_wilayah, "None", 0, 0, 0,
"None", "None", "", "", 0, 0);
                    CalculateAreaForm cform = new CalculateAreaForm();
                    cform.setMode("new", NewDataWilayah.id_wilayah);
                    cform.form_before = this;
                    textBox1.Text = "";
                    this.Hide();
                    cform.Show();
                }
            }
            else
            {
                MessageBox.Show("Nama Wilayah Sudah Tendaftar!");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
    }
    else { MessageBox.Show("Login First !!"); }
}

```

Kode Sumber 4.6 Implementasi Proses Menambah Data Wilayah Pantauan

4.5.1.3 Proses Mengubah Data Wilayah Pantauan

Proses ini digunakan untuk menampilkan antarmuka mengelola data wilayah pantauan yang sudah terdaftar sehingga pengguna dapat mengubah data wilayah pantauan tersebut. Implementasi proses ini dapat dilihat pada Kode Sumber 4.7.

```

string tes = dgvWilayah.CurrentRow.Cells[0].Value.ToString();
int currentRow = int.Parse(e.RowIndex.ToString());
idWilayah = int.Parse(tes);
//MessageBox.Show(idWilayah.ToString());

//edit button
if (dgvWilayah.Columns[e.ColumnIndex] ==
editButton && currentRow >= 0)
{
    if (logged) {
        CalculateAreaForm cform = new CalculateAreaForm();
        cform.form_before = this;
        cform.setMode("edit", idWilayah);
        cform.Show();
        this.Hide();
    }
    Else {
        MessageBox.Show("Login First !!");
        LoginForm lf = new LoginForm();
        lf.sf = this;
        lf.mode = "login";
        lf.Show();
    }
}
}

```

Kode Sumber 4.7 Implementasi Proses Mengubah Data Wilayah Pantauan

4.5.1.4 Proses Menghapus Data Wilayah Pantauan

Proses ini digunakan untuk menghapus data wilayah pantauan yang tidak digunakan. Pengguna memilih wilayah yang akan dihapus. Sistem menampilkan pesan untuk menyetujui penghapusan. Kemudian pengguna menyetujui menghapus data. Implementasi proses ini dapat dilihat pada Kode Sumber 4.8.

```

else if (dgvWilayah.Columns[e.ColumnIndex] == deleteButton &&
currentRow >= 0)
    {
        if (logged)
            {
                if (MessageBox.Show("Really delete " +
dgvWilayah.CurrentRow.Cells[1].Value.ToString() + " ?",
"Confirm delete", MessageBoxButtons.YesNo) ==
DialogResult.Yes) {
                    try {
wilayahCRUD.deletewilayah(idWilayah);
loadDGV();
MessageBox.Show("Deleted");
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show(ex.ToString());
                    }
                }
            }
        else { MessageBox.Show("Login First !!"); }
    }

```

Kode Sumber 4.8 Implementasi Proses Menghapus Data Wilayah Pantauan

4.5.1.5 Proses Menampilkan Antarmuka Melihat Hasil Pantauan Sungai

Proses ini digunakan untuk menampilkan antarmuka melihat hasil pantauan dari suatu wilayah pantauan yang sudah terdaftar dan memiliki data yang lengkap. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.9.

```

else if (dgvWilayah.Columns[e.ColumnIndex] ==
startMonitorButton && currentRow >= 0)
    {
        MonitoringForm mf = new MonitoringForm();
        mf.form_before = this;
        mf.idWilayah = idWilayah;
        mf.Show();
        this.Hide();
    }

```

Kode Sumber 4.9 Implementasi Proses Menampilkan Antarmuka Melihat Hasil Pantauan

4.5.1.6 Proses Menampilkan Antarmuka Mengelola Data Langganan SMS Peringatan

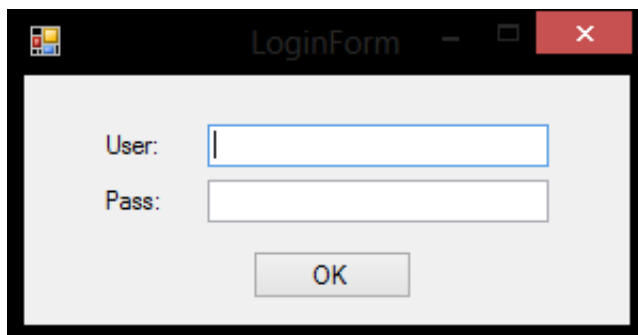
Proses ini digunakan untuk menampilkan antarmuka untuk manajemen data langganan SMS peringatan. Implementasi proses ini dapat dilihat pada Kode Sumber 4.10

```
private void phoneAccountToolStripMenuItem1_Click(object sender, EventArgs e)
{
    ManagePhoneForm mpf = new ManagePhoneForm();
    mpf.Show();
}
```

Kode Sumber 4.10 Implementasi Proses Menampilkan Antarmuka Mengelola Data Langganan SMS Peringatan

4.5.2 Implementasi Antarmuka *Login*

Pada antarmuka *login*, Pengguna melakukan login agar dapat mendapatkan akses penuh terhadap aplikasi, sehingga pengguna dapat mengelola data wilayah. Antarmuka ini digambarkan pada Gambar 4.2. Pada antarmuka ini terdapat beberapa proses aplikasi.



Gambar 4.2 Implementasi Antarmuka *Login*

4.5.2.1 Proses *Login* Ke Dalam Aplikasi

Proses ini digunakan untuk memberikan hak akses penuh kepada pengguna yang berstatus admin sehingga pengguna dapat mengelola data wilayah pantauan. Pengguna mengisi *username* dan *password* dan menekan tombol “OK”. Kemudian aplikasi membandingkan data masukan dengan data di basis data. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.11.

```
private void button1_Click(object sender, EventArgs e)
{
    if (mode == "login")
    {
        try
        {
            string user = textBox1.Text.Trim();

            tadbEntities tadb = new tadbEntities();

            var que = (from i in tadb.t_user
                      where i.username == user
                      select i).ToList();
            if (que[0].password == textBox2.Text.Trim())
            {
                sf.logged = true;
                sf.changeState();
                sf.username = user;
                this.Close();
            }
            else { MessageBox.Show("Username atau
Password Salah!"); }
        }
        catch { MessageBox.Show("Username atau
Password Salah!"); }
    }
    else if (mode == "edit")
    {
        tadbEntities tadb = new tadbEntities();
        t_user data_user = tadb.t_user.Single(x =>
x.username == sf.username);

        if (textBox2.Text.Trim() ==
data_user.password)
        {
```

```

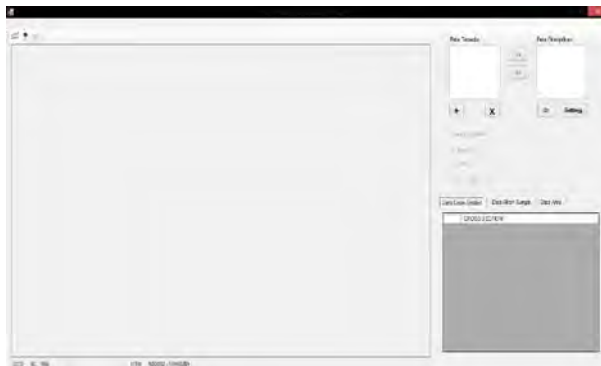
        data_user.username = textBox1.Text.Trim();
        data_user.password = textBox3.Text.Trim();
        tadb.SaveChanges();
        MessageBox.Show("Password Changed");
        this.Close();
    }
    else
    {
        MessageBox.Show("Old Pass MisMatch");
    }
}
}

```

Kode Sumber 4.11 Implementasi Proses Login

4.5.3 Implementasi Antarmuka Mengelola Data Wilayah

Pada antarmuka ini, pengguna dapat menambah data-data yang berhubungan dengan bagian sungai yang dipantau, seperti menambahkan peta yang digunakan, menambah data sungai dan penampang sungai, menambah data aliran dan lain-lain. Pengguna juga dapat menghitung luas area. Antarmuka mengelola data bagian sungai digambarkan pada Gambar 4.3.



Gambar 4.3 Implementasi Antarmuka Mengelola Data Wilayah

Pada antarmuka mengelola data wilayah terdapat beberapa proses sebagai berikut.

4.5.3.1 Proses Menambahkan Peta

Proses ini digunakan untuk menambahkan peta pada daftar peta yang tersedia. Pengguna mengklik tombol “+” pada antarmuka mengelola data wilayah. Kemudian sistem menampilkan *file dialog* untuk memilih *file* peta. Implementasi proses ini dapat dilihat pada Kode Sumber 4.12.

```
private void button1_Click(object sender, EventArgs e)
{
    OpenFileDialog fdg = new OpenFileDialog();
    fdg.Filter = "Shapefiles (*.shp)|*.shp";
    if (fdg.ShowDialog() == DialogResult.OK)
    {
        listBox1.Items.Add(Path.GetFileNameWithoutExtension(fdg.FileName));

        shp.AddSHPFile(Path.GetDirectoryName(fdg.FileName),
            Path.GetFileNameWithoutExtension(fdg.FileName), rootdir + "\\\"
            + datawilayah.path_shapefile);
        ShapeUsedCRUD sdr = new ShapeUsedCRUD();
        sdr.InsertShapeUsed(idWil,
            Path.GetFileNameWithoutExtension(fdg.FileName), 0, "", 2, 1,
            "0");
    }
}
```

Kode Sumber 4.12 Implementasi Proses Menambah Peta

4.5.3.2 Proses Menampilkan Peta Ke Panel Peta

Proses ini digunakan untuk menampilkan peta yang ada pada daftar peta tersedia ke panel peta. Pengguna mengklik *file* peta yang ada pada daftar peta tersedia dan kemudian mengklik tombol “>>”. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.13.

```
private void button6_Click(object sender, EventArgs e)
{
    if (listBox1.SelectedIndex != -1)
    {
        tadbEntities tadbentities = new tadbEntities();
        string filename = listBox1.SelectedItem.ToString();
    }
}
```



```

        var idshape = (from i in tadbentities.t_shapeused
            where i.nama_file == filename && i.id_wilayah == idwil
            select i).ToList();

        int idshp = idshape[0].idt_shapeused;
        ShapeUsedCRUD sdr = new ShapeUsedCRUD();
        sdr.UpdateUsed(idshp, 1);

        listBox2.Items.Add(listBox1.SelectedItem.ToString());
        int index = listBox2.Items.Count - 1;
        sdr.UpdateIndex(idshp, index);

        mapcontrol.OpenShapefile(datawilayah.path_shapefile + "\\\" +
            listBox1.SelectedItem + ".shp", this.sfMap1, index, "area");
        listBox1.Items.RemoveAt(listBox1.SelectedIndex);
        ShapeFile sf = sfMap1[index];
        sf.RenderSettings.MinPixelPenWidth =
        (int)idshape[0].penwidth_min;
        sf.RenderSettings.MaxPixelPenWidth =
        (int)idshape[0].penwidth_max;
        if (idshape[0].field_label != -1)
        {
            sf.RenderSettings.FieldName =
            sf.RenderSettings.DbFReader.GetFieldNames()[(int)idshape[0].fi
            eld_label];
            sf.RenderSettings.ToolTipFieldName =
            sf.RenderSettings.FieldName;
        }
        try
        {
            sf.RenderSettings.FillColor =
            Color.FromArgb(int.Parse(idshape[0].color));
        }
        catch { }
    }
}

```

Kode Sumber 4.13 Implementasi Proses Menampilkan Peta

4.5.3.3 Proses Menghapus Peta

Proses ini digunakan untuk menghapus *file* peta yang terdaftar pada daftar peta tersedia. Pengguna mengklik *file* peta yang akan dihapus dan kemudian mengklik tombol “X”. Sistem menampilkan pesan menyetujui penghapusan. Pengguna setuju

menghapus peta. Implementasi proses ini dapat dilihat pada Kode Sumber 4.14.

```
private void button2_Click(object sender, EventArgs e)
{
    tadbEntities tadbentities = new tadbEntities();
    if (listBox1.SelectedIndex != -1)
    {
        string name =
listBox1.SelectedItem.ToString();
        var q = (from i in tadbentities.t_shapeused
                where i.nama_file == name &&
i.id_wilayah == idWil
                select i.idt_shapeused).ToList();
        int idshp = q[0];
        ShapeUsedCRUD sdr = new ShapeUsedCRUD();
        sdr.DeleteShapeUsed(idshp);
        shp.DeleteSHPFile(rootdir + "\\\" +
datawilayah.path_shapefile, listBox1.SelectedItem.ToString());

listBox1.Items.RemoveAt(listBox1.SelectedIndex);
    }
    else
    {
        MessageBox.Show("Choose Layer to be Removed");
    }
}
}
```

Kode Sumber 4.14 Implementasi Proses Menghapus Peta

4.5.3.4 Proses Menampilkan Antarmuka Mengubah Lapisan Peta

Proses ini digunakan untuk menampilkan antarmuka untuk mengubah tampilan peta. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.15.

```
private void button4_Click(object sender, EventArgs e)
{
    ManageShapeFileForm mgs = new
ManageShapeFileForm();
    mgs.calculateAreaForm = this;
    mgs.idWil = idWil;
}
```

```

        mgs.path_maps = rootdir + "\\\" +
datawilayah.path_shapefile;
        mgs.Show();
    }

```

Kode Sumber 4.15 Implementasi Proses Menampilkan Antarmuka Mengubah Lapisan Peta

4.5.3.5 Proses Menambah Data Potongan Bagian Sungai

Proses ini digunakan untuk mengelola data potongan bagian sungai. Pengguna mengklik tombol *toolstrip* “Add Cross Section”. Kemudian pengguna mengklik lokasi titik pada peta. Sistem kemudian menampilkan antarmuka mengelola data potongan bagian sungai. Implementasi proses ini dapat dilihat pada Kode Sumber 4.16.

```

private void sfMap1_MouseUp(object sender, MouseEventArgs e)
{
    if (addSungai)
    {
        InsertDataSungaiForm idsf = new InsertDataSungaiForm();
        idsf.kood = sfMap1.PixelCoordToGisPoint(e.Location);
        idsf.idWilayah = idWil;
        idsf.caf = this;
        idsf.Show();
        return;
    }
}

```

Kode Sumber 4.16 Implementasi Proses Menambah Data Potongan Bagian Sungai

4.5.3.6 Proses Menambah Data Aliran Sungai

Proses ini digunakan untuk menambah data aliran sungai dari satu potongan bagian sungai ke bagian sungai lain. Pengguna menggambar aliran di panel peta. Sistem menampilkan antarmuka isian aliran potongan bagian sungai. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.17.

```

private void sfMap1_MouseUp(object sender, MouseEventArgs e)
{
    if (!mouseMove)

```

```

        {
    if ((drawToCalculate || addFlow) && (!getVerticesToDraw)){
        initialMousePos = e.Location;
        initialpos =
sfMap1.PixelCoordToGisPoint(e.Location);
        points.Add(initialpos);
        sfMap1.Invalidate();
        //mouseMove = true;
    }
}

private void sfMap1_KeyDown(object sender, KeyEventArgs e)
{
    if (addFlow)
    {
        if (e.KeyCode == Keys.S)
        {
            sfMap1.Invalidate();
            drawMove = false;
            stopDraw = true;
            addFlow = false;

            InsertFlowForm iff = new InsertFlowForm();
            iff.idWilayah = idWil;
            iff.points = this.points;
            iff.caf = this;
            iff.Show();
        }
        if (e.KeyCode == Keys.D)
        {
            points.RemoveAt(points.Count - 1);
            initialpos = points[points.Count - 1];
        }
    }
}

```

Kode Sumber 4.17 Implementasi Proses Menambah Data Aliran Sungai

4.5.3.7 Proses Menghitung Luas Dan Volume Area Kontur

Proses ini digunakan untuk mengetahui luas dan volume area kontur suatu wilayah pantauan. Pengguna mengklik tombol *toolbar* “Draw to Calculate”. Kemudian pengguna menggambar area kontur yang akan dihitung. Setelah selesai digambar, sistem menghitung luas dengan memanggil kelas yang

mengimplementasikan metode koordinat yang telah dijelaskan pada bab sebelumnya. Implementasi dari kelas ini dapat dilihat pada Kode Sumber 4.18.

```
private void sfMap1_KeyDown(object sender, KeyEventArgs e)
{
    if (drawToCalculate)
    {
        if (e.KeyCode == Keys.S)
        {
            try
            {
                getVerticesToDraw = false;
                points.Add(points[0]);
                drawToCalculate = false;
                drawMove = false;
                stopDraw = true;
                sfMap1.Invalidate();
                textBox2.Text =
                mapcontrol.CalculateSingleArea(points).ToString();
                groupBox1.Enabled = true;
                button5.Visible = true;
                sfMap1.Cursor = Cursors.Default;
            }
            catch { }
        }
    }
}
```

Kode Sumber 4.18 Implementasi Proses Menghitung Luas Dan Volume Kontur

4.5.4 Implementasi Antarmuka Mengelola Data Sungai

Gambar 4.4 menunjukkan implementasi dari antarmuka mengelola data sungai. Pada antarmuka ini pengguna dapat mengelola data-data yang berhubungan dengan profil sungai seperti nama profil sungai, lokasi koordinat lintang dan bujur, id sensor buatan yang digunakan, lokasi profil sungai, serta data penampang sungai. Antarmuka ini juga memiliki fungsi untuk meng-*export* data penampang sungai dari *file* excel. Pada antarmuka ini terdapat beberapa proses.

4.5.4.1 Proses Menambah Data Potongan Bagian Sungai

Proses ini digunakan untuk menambah data potongan bagian sungai ke dalam basis data. Pengguna memasukkan data potongan bagian sungai dan data penampang sungai. Data penampang sungai dibuat menjadi kumpulan titik x dan y. Kemudian sistem menyimpan data masukan ke dalam basis data. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.19.

```
private void button1_Click(object sender, EventArgs e)
{
    dataChanged = false;
    if (mode == "edit")
    {
        string[] dataP = BuildDataPenampang(testcs);
        string NamaSensor = textBox6.Text.Trim();
        //MessageBox.Show("X: "+ dataP[0] + "Y: " +
dataP[1]);
        tadbEntities tadb = new tadbEntities();
        t_section_sungai data =
tadb.t_section_sungai.Single(x => x.id_wilayah == idWilayah &&
x.idt_sungai == idrecord);
        data.nama_sungai = textBox1.Text;
        data.koordinatX = double.Parse(textBox2.Text);
        data.koordinatY = double.Parse(textBox3.Text);
        data.debit_max = double.Parse(textBox4.Text);
        //data.luas_penampang =
double.Parse(textBox5.Text);
        data.lokasi = textBox5.Text;
        data.url_sensor = NamaSensor;
        data.st_bank = dataP[0];
        data.st_elev = dataP[1];
        data.tinggi_air_max =
double.Parse(textBox7.Text);
        data.tinggi_bantaran =
double.Parse(textBox8.Text);
        tadb.SaveChanges();
        add = true;
        MessageBox.Show("Data Edited");
        this.Close();
    }
    else
    {
        string[] data = BuildDataPenampang(testcs);
```

```

        string NamaSensor = textBox6.Text.Trim();
        SungaiCRUD scrud = new SungaiCRUD();
        scrud.InsertDataSungai(idWilayah,
        textBox1.Text, double.Parse(textBox2.Text),
        double.Parse(textBox3.Text), double.Parse(textBox4.Text),
        textBox5.Text, NamaSensor, data[0], data[1],
        double.Parse(textBox7.Text), double.Parse(textBox8.Text));
        add = true;
        MessageBox.Show("Data Added");
        this.Close();
    }
}

```

Kode Sumber 4.19 Implementasi Proses Menambah Data Potongan Bagian Sungai

Gambar 4.4 Implementasi Antarmuka Mengelola Data Sungai

4.5.4.2 Proses Menambah Data Penampang Sungai Dari File Excel

Proses ini digunakan untuk menambah data penampang sungai secara otomatis dari *file* excel yang menyimpan data penampang. Pengguna mengklik tombol “export” dan kemudian memilih *file* excel. Sistem memproses masukan dan kemudian menampilkan gambar bentuk penampangnya di panel gambar penampang sungai. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.20.

```
private void button6_Click(object sender, EventArgs e)
{
    string path;
    OpenFileDialog fdg = new OpenFileDialog();
    fdg.Filter = "Excel Files (*.xls)|*.xls;*.xlsx";
    if (fdg.ShowDialog() == DialogResult.OK)
    {
        try
        {
            path = fdg.FileName;
            OleDbConnection oledbConn = new OleDbConnection();

            if (Path.GetExtension(path) == ".xls")
            {
                oledbConn = new
                OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0; Data
                Source=" + path + ";Extended Properties=\\"Excel
                8.0;HDR=Yes;IMEX=2\\"");
            }
            else if (Path.GetExtension(path) == ".xlsx")
            {
                oledbConn = new
                OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0; Data
                Source=" + path + "; Extended Properties='Excel
                12.0;HDR=YES;IMEX=1;');");
            }

            oledbConn.Open();
            OleDbDataAdapter da = new
            OleDbDataAdapter("select * from [Sheet1$]", oledbConn);
            da.TableMappings.Add("Table", "TestTable");
            DataSet ds = new DataSet();
            da.Fill(ds);
        }
    }
}
```



```

        int count = ds.Tables[0].Rows.Count;

        for (int i = 0; i < count; i++)
        {
            seqnum++;
            testcs.Add(new crosspoint { X =
double.Parse(ds.Tables[0].Rows[i][0].ToString()), Y =
double.Parse(ds.Tables[0].Rows[i][1].ToString()) });
        }
        LoadDGVSec();

        oledbConn.Close();
        crosspoint cslowest = testcs.OrderBy(x => x.Y).First();
        crosspoint cshighest = testcs.OrderByDescending(x =>
x.Y).First();
        crosspoint cshighest2 = testcs.OrderByDescending(x =>
x.X).First();

        if (starty > cslowest.Y)
        {
            starty = (int)Math.Floor(cslowest.Y);
        }
        if (endy < cshighest.Y)
        {
            endy = (int)Math.Ceiling(cshighest.Y);
        }
        if (endx < cshighest2.X)
        {
            endx = (int)Math.Ceiling(cshighest2.X);
        }
        if (endy < height) {
            endy = (int)Math.Ceiling(height);
        }
        tot_x = endx / 10;
        tot_y = endy - starty;
        Scaling(tot_x, tot_y);
        dataChanged = true;
        pictureBox1.Invalidate();
    }
    catch (Exception ex)
    {

        MessageBox.Show(ex.ToString()); } } }

```

**Kode Sumber 4.20 Implementasi Proses Menambah Data
Penampang Sungai Dari File Excel**

4.5.4.3 Proses Menggambar Data Penampang

Proses ini merupakan proses yang digunakan untuk merepresentasikan data penampang yang ada menjadi bentuk gambar yang dapat dilihat. Data penampang sungai dibuat menjadi sekumpulan koordinat titik x dan y. Kemudian kumpulan titik tersebut diiterasi dan digambarkan di panel gambar penampang sungai. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.21.

```
//Menggambar Grafik x,y
for (int i = starty; i <= endy; i++)
    {
        using (Font myFont = new Font("Arial", 7))
        {
            if (i % 2 == 0)
            {
e.Graphics.DrawString(i.ToString(), myFont, Brushes.Black, new
Point(15, ((max_height_pic - 5) - county)));
            }
        }
        county += scale_coord_y;
    }
    e.Graphics.DrawString("Tinggi", new
Font("Arial", 7), Brushes.Black, new Point(15,
((max_height_pic - 15) - (county - scale_coord_y))));

    for (int i = 0; i <= endx; i += 10)
    {
        using (Font myFont = new Font("Arial", 7))
        {
            if (i % 20 == 0)
            {
e.Graphics.DrawString(i.ToString(), myFont, Brushes.Black, new
Point((25 + countx), max_height_pic + 5));
            }
        }
        countx += scale_coord_x;
    }
    e.Graphics.DrawString("Sta. Bank", new
Font("Arial", 7), Brushes.Black, new Point((25 + (countx -
scale_coord_x)), max_height_pic - 10));
```

```

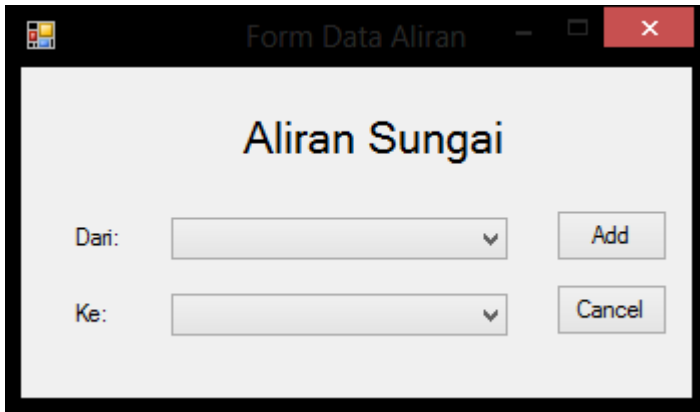
//Menggambar Penampang
for (int i = 0; i < listheight.Count - 1; i++)
    {
        if (i % 2 == 0)
            {
                e.Graphics.DrawLine(new
                Pen(Color.Aqua), new Point((int)listheight[i].X,
                (int)listheight[i].Y), new Point((int)listheight[i + 1].X,
                (int)listheight[i + 1].Y));
            }
        }
    e.Graphics.DrawString("Tinggi air max", new
    Font("Arial", 7), Brushes.Black, new
    Point((int)listheight[listheight.Count-1].X,
    (int)listheight[listheight.Count-1].Y));
    e.Graphics.DrawString("Bantaran", new
    Font("Arial", 7), Brushes.Black, new
    Point((int)listheight_ban[listheight_ban.Count - 1].X,
    (int)listheight_ban[listheight_ban.Count - 1].Y));
    for (int i = 0; i < listheight_ban.Count - 1;
    i++)
        {
            if (i % 2 == 0)
                {
                    e.Graphics.DrawLine(new
                    Pen(Color.Brown), new Point((int)listheight_ban[i].X,
                    (int)listheight_ban[i].Y), new Point((int)listheight_ban[i +
                    1].X, (int)listheight_ban[i + 1].Y));
                }
            }
        using (Pen p = new Pen(Color.Black))
            {
                e.Graphics.DrawLine(p, new Point(30,
                max_height_pic), new Point(30, max_height_pic - (county -
                scale_coord_y)));
                e.Graphics.DrawLine(p, new Point(30,
                max_height_pic), new Point(30 + (countx - scale_coord_x),
                max_height_pic));
                //e.Graphics.DrawString("o",new
                Font("Arial",7),Brushes.Black,new Point(30+pixx,275-pixy));
            }

```

Kode Sumber 4.21 Implementasi Proses Menggambar Data Penampang

4.5.5 Implementasi Antarmuka Mengelola Data Aliran

Gambar 4.5 menunjukkan implementasi dari antarmuka mengelola data aliran yang sudah dirancang. Pada antarmuka ini, pengguna dapat menambahkan data aliran dari suatu bagian sungai ke bagian sungai lain yang dipantau. Pada antarmuka ini terdapat satu proses yaitu menambah data aliran.



Gambar 4.5 Implementasi Antarmuka Mengelola Data Aliran

4.5.5.1 Proses Menambah Data Aliran

Proses ini digunakan untuk menambah data aliran sungai ke basis data. Pengguna mengisi data aliran. Kemudian sistem menyimpan data ke basis data. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.22.

```
private void button1_Click(object sender, EventArgs e)
{
    tadbEntities tadb = new tadbEntities();

    if (comboBox1.SelectedItem != null &&
        comboBox2.SelectedItem != null)
    {
        string sfrom = comboBox1.SelectedItem.ToString();
        string sto = comboBox2.SelectedItem.ToString();
    }
}
```

```

        var qform = (from i in tadb.t_section_sungai
                    where i.id_wilayah == idWilayah &&
i.nama_sungai == sfrom
                    select i).ToList();

        var qto = (from i in tadb.t_section_sungai
                  where i.id_wilayah == idWilayah &&
i.nama_sungai == sto
                  select i).ToList();

        PointD pfrom = new
PointD((double)qform[0].koordinatX, (double)qform[0].koordinatY);
        PointD pto = new PointD((double)qto[0].koordinatX,
(double)qto[0].koordinatY);
        if (sfrom != "None")
        {
            points.Insert(0, pfrom);
        }

        if (sto != "None")
        {
            points.Add(pto);
        }

        SungaiCRUD scd = new SungaiCRUD();
        scd.InsertDataFlow(qform[0].idt_sungai,
qto[0].idt_sungai, points);
        MessageBox.Show("Flow Added");
        this.Close();
    }
    else
    {
        MessageBox.Show("Data Invalid!");
    }
}

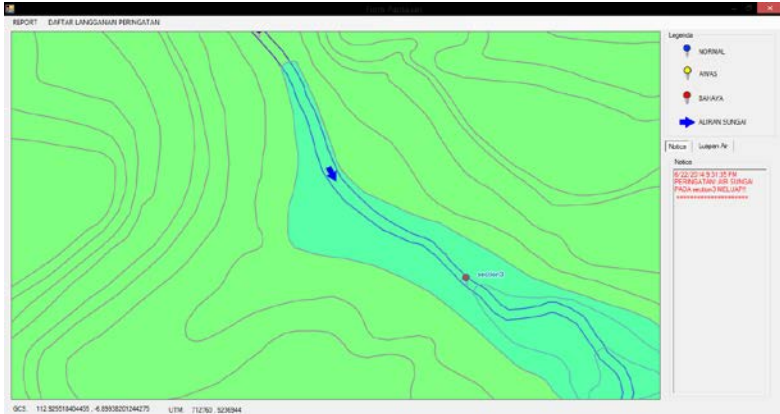
```

Kode Sumber 4.22 Implementasi Proses Menambah Data Aliran

4.5.6 Implementasi Antarmuka Pemantauan Kondisi Sungai

Gambar 4.6 menunjukkan implementasi dari antarmuka pemantauan kondisi sungai. Pengguna dapat melihat hasil dari pemantauan sensor pada tiap bagian sungai. Pengguna juga dapat

melihat pesan peringatan kejadian pada wilayah pantauan Pada antarmuka ini terdapat beberapa proses.



Gambar 4.6 Implementasi Antarmuka Pemantauan Kondisi Sungai

4.5.6.1 Proses Mengambil Data Sensor

Proses ini digunakan untuk membaca data sensor yang dikirimkan melalui *web service*. Pengambilan data dilakukan oleh sistem setiap satu menit sekali. Data yang dikirimkan oleh *web service* berformat XML. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.23.

```
void updateDataVelocity() {
    getDataVTimer.Interval = 60000;
    getDataVTimer.Elapsed += getDataVTimer_Elapsed;
    getDataVTimer.Start(); }
Void getDataVelocity {
url = "http://localhost/agent/agent.php?mode=getDataV&name=" +
data[i].url_sensor;
    XMLParser parser = new XMLParser();
    XDocument ds = parser.MakeRequest(url);
    velo =
double.Parse(ds.Descendants("velocity").ElementAt(0).Value);
    elev=
double.Parse(ds.Descendants("elevation").ElementAt(0).Value); }
```

Kode Sumber 4.23 Implementasi Proses Mengambil Data Sensor

4.5.6.2 Proses Mengirimkan Pesan Peringatan

Proses ini digunakan aplikasi untuk mengirimkan SMS peringatan ke pengguna. Sistem membandingkan data tinggi muka air sekarang dengan data tinggi muka air maksimal. Bila data sekarang melebihi data maksimal, maka sistem akan mengirimkan SMS peringatan. Implementasinya dapat dilihat pada Kode Sumber 4.24.

```
#region CEK TINGGI AIR MELEBIHI MAKSIMAL
if (elev < ((double)data[i].tinggi_bantaran + (double)
data[i].tinggi_air_max)) {
if ((double)data[i].tinggi_air_max < elev) {
if (!notice_warn[i] && !notice_emerg[i]) {
if (notice_earlywarn[i]) {
data[i].sent_sms = "none"; }
if (data[i].sent_sms == "none") {
string message = DateTime.Now.ToString() + "\nPERINGATAN! AIR
SUNGAI PADA " + data[i].nama_sungai + " TELAH MELEBIHI MAKSIMAL\n"
+ "TINGGI AIR MAKS: " + data[i].tinggi_air_max + "m \n" + "TINGGI
AIR SKG: " + Math.Round(elev, 2) + "m";
var q_phone = (from p in tadb.t_phonecontact
where p.active == "active"
select p.phone_number).ToList();
for (int co = 0; co < q_phone.Count; co++) {
tadb.Database.ExecuteNonQuery("INSERT INTO outbox
(DestinationNumber, TextDecoded, CreatorID) VALUES ({0},
{1}, 'Gammu')", q_phone[co], message); }
data[i].sent_sms = "done"; }
richTextBox1.Invoke((MethodInvoker)(() =>
richTextBox1.SelectionColor = Color.Orange));
richTextBox1.Invoke((MethodInvoker)(() =>
richTextBox1.AppendText(DateTime.Now.ToString() + "\nPERINGATAN!
AIR SUNGAI PADA " + data[i].nama_sungai + " TELAH MELEBIHI
MAKSIMAL\n" + "TINGGI AIR MAKS: " + data[i].tinggi_air_max + "m
\n" + "TINGGI AIR SKG: " + Math.Round(elev,2) + "m" +
"\n=====\n\n")); }
notice_warn[i] = true;
if (datanya.state.Trim() != "Awas") {
datanya.state = "Awas";
using (var bmpTemp = new Bitmap(rootdir + "\\Images\\warn.png"))
{ img = new Bitmap(bmpTemp); }
images.Remove(datanya.nama_sungai.ToString());
images.Add(datanya.nama_sungai.ToString(), img);
warnon = true; } }
```

```

else { if (datanya.state != "Normal" && datanya.state != "Bahaya")
{ notice_warn[i] = false;
datanya.state = "Normal";
using (var bmpTemp = new Bitmap(rootdir + "\\Images\\normal.png"))
{ img = new Bitmap(bmpTemp); }
images.Remove(datanya.nama_sungai.ToString());
images.Add(datanya.nama_sungai.ToString(), img);
warnon = true;
if (data[i].sent_sms == "done") {
data[i].sent_sms = "none"; } }
else if(warnonemerg && datanya.state == "Normal") {
datanya.state = "Normal";
using (var bmpTemp = new Bitmap(rootdir + "\\Images\\normal.png"))
{ img = new Bitmap(bmpTemp); }
images.Remove(datanya.nama_sungai.ToString());
images.Add(datanya.nama_sungai.ToString(), img);
warnon = true; } } }
#endregion

```

**Kode Sumber 4.24 Implementasi Proses Mengirimkan SMS
Peringatan**

Jika data tinggi muka air sekarang tidak melebihi maksimal, maka sistem akan menghitung rata-rata kenaikan tinggi muka air selama 10 menit. Kemudian data rata-rata tersebut diakumulasi sebanyak dua jam ke depan. Bila data total tinggi muka air sekarang ditambah tinggi muka air rata-rata melebihi maksimal maka sistem akan mengirimkan pesan peringatan kepada pengguna. Implementasi dari proses ini dapat dilihat pada Kode Sumber 4.25

```

#region CEK VELOCITY DAN WS ELEVATION AKUMULASI DUA JAM KEDEPAN
if (!firstLoad) {
datasec.v_nexth = v2h[i];
datasec.elev_nexth = elev2h[i];
if (rata_diff[i].Count == 10) {
diff = diff * 120;
diff_elev = diff_elev * 120;
velo_acc = velo + diff;
elev_acc = elev + diff_elev;
v2h[i] = velo_acc;
elev2h[i] = elev_acc;
datasec.v_nexth = velo_acc;
datasec.elev_nexth = elev_acc;
if (elev_acc > data[i].tinggi_air_max) {

```



```

if (data[i].sent_sms == "none") {
if (!notice_earlywarn[i] && !notice_emerg[i] && !notice_warn[i]) {
string message = "PERINGATAN! TINGGI MUKA AIR PADA " +
data[i].nama_sungai +
"\nTINGGI SKG: " + Math.Round(elev,2) + "m" +
"\nRATA2 KENAIKAN: " + Math.Round(diff_elev / 120,2) + "m" +
"\nDALAM 2 JAM: " + Math.Round(elev_acc,2)+"m" +
"\nTINGGI AIR MAX: " + data[i].tinggi_air_max+"m";
var q_phone = (from p in tadb.t_phonecontact
where p.active == "active"
select p.phone_number).ToList();
for (int co = 0; co < q_phone.Count; co++) {
tadb.Database.ExecuteNonQuery("INSERT INTO outbox
(DestinationNumber, TextDecoded, CreatorID) VALUES ({0},
{1},'Gammu')", q_phone[co], message); }
data[i].sent_sms = "done";

richTextBox1.Invoke((MethodInvoker)(() =>
richTextBox1.SelectionColor = Color.Brown));
richTextBox1.Invoke((MethodInvoker)(() => richTextBox1.Text +=
DateTime.Now.ToString() + "\nPERINGATAN! TINGGI MUKA AIR PADA " +
data[i].nama_sungai +

"\nTINGGI SKG: " + Math.Round(elev, 2) + "m" +
"\nRATA2 KENAIKAN: " + Math.Round(diff_elev / 120,2) + "m" +
"\nDALAM 2 JAM: " + Math.Round(elev_acc,2)+"m" +
"\nTINGGI AIR MAX: " + data[i].tinggi_air_max +
"\n=====\\n\\n")); }
notice_earlywarn[i] = true; } }
else { notice_earlywarn[i] = false;
if (data[i].sent_sms == "done" && !notice_warn[i]) {
data[i].sent_sms = "none"; } } } }
#endregion

```

Kode Sumber 4.25 Implementasi Proses Mengirimkan SMS Peringatan (2)

4.5.6.3 Proses Menghitung Tinggi Genangan Air

Proses ini digunakan aplikasi untuk mengetahui tinggi genangan air pada kontur wilayah berdasarkan volume luapan air. Sistem mengakumulasi volume luapan air. Kemudian membandingkan volume luapan air dengan volume kontur dimana volume kontur lebih besar dari volume luapan air. Sehingga diketahui tinggi genangan berada pada elevasi kontur

berapa. Implementasi proses menghitung tinggi genangan air dapat dilihat pada Kode Sumber 4.26.

```

if (vol_flood_total > 0) {
try {
    poly_kontur.Clear();
    tadbEntities tadb = new tadbEntities();
    kontur = (from i in tadb.t_record
    where i.id_wilayah == idWilayah && i.volume_tot < vol_flood_total
    select i).Distinct().OrderByDescending(x => x.elevation).ToList();

    if (kontur.Count > 0) {
    string[] konturx = kontur[0].shapeX.Split(';');
    string[] kontury = kontur[0].shapeY.Split(';');

    for (int j = 0; j < konturx.Length - 1; j++){
        Point p =
    sfMap1.GisPointToPixelCoord(double.Parse(konturx[j]),
    double.Parse(kontury[j]));
        poly_kontur.Add(p);
    } }

    System.Drawing.Drawing2D.FillMode ds =
    System.Drawing.Drawing2D.FillMode.Alternate;
    Point[] sdsd = poly_kontur.ToArray();
    e.Graphics.FillPolygon(new SolidBrush(Color.FromArgb(80,
    Color.Aqua)), sdsd, ds);
    }
    catch { }
}
}

```

Kode Sumber 4.26 Implementasi Proses Menghitung Tinggi Genangan Air

4.5.7 Implementasi Antarmuka Melihat Laporan Data Sungai

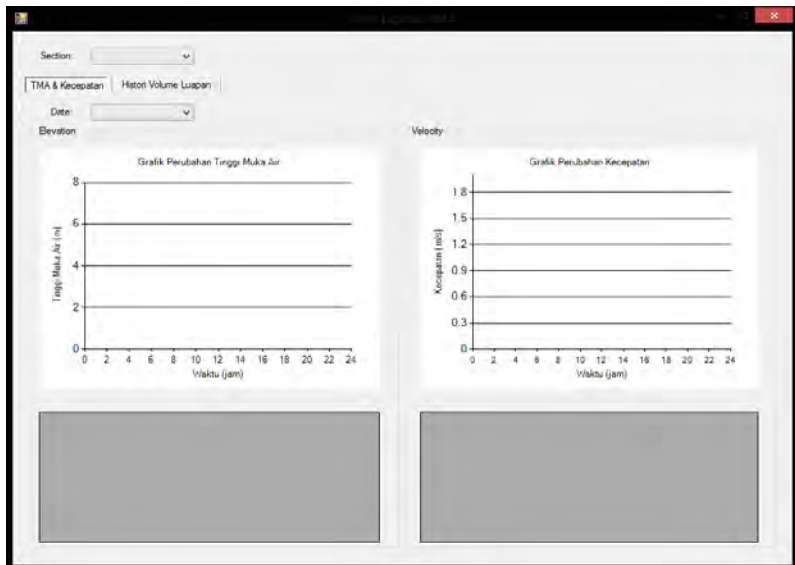
Gambar 4.7 menunjukkan implementasi dari antarmuka melihat laporan data sungai. Pada antarmuka ini, pengguna dapat melihat data kecepatan dan tinggi muka air yang terekam oleh sensor berdasarkan tanggal perekaman per jamnya. Antarmuka ini juga memiliki panel yang dapat menggambarkan grafik perubahan tinggi muka air dan kecepatan pada suatu profil sungai berdasarkan waktu yang tersimpan di basis data.

4.5.8 Implementasi Antarmuka Detil Profil Sungai

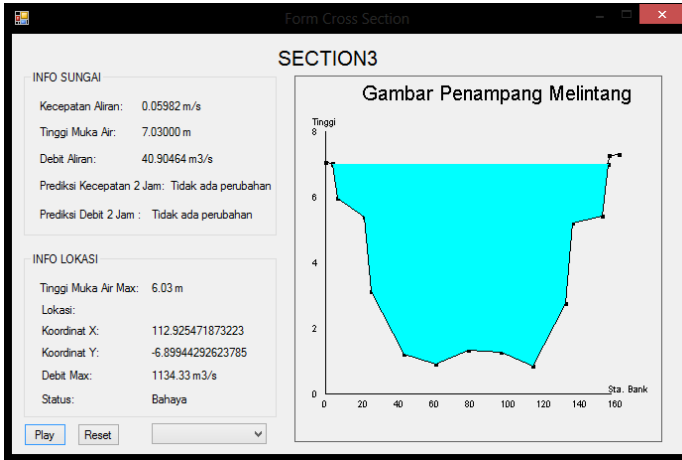
Gambar 4.8 menunjukkan implementasi dari antarmuka detil profil sungai. Pada antarmuka ini, pengguna dapat melihat informasi yang berkaitan dengan suatu profil sungai.

4.5.9 Implementasi Antarmuka Aplikasi Pembangkit Data Sensor

Gambar 4.9 menunjukkan implementasi dari antarmuka aplikasi pembangkit data sensor. Antarmuka ini digunakan untuk membangkitkan data sensor yang akan digunakan dalam tahap pengujian aplikasi sistem informasi geografis.



Gambar 4.7 Implementasi Antarmuka Melihat Laporan Data Sungai



Gambar 4.8 Implementasi Antarmuka Detil Profil Sungai

The screenshot shows a data management application interface. At the top, there are input fields for "Name:", "Velocity:", and "Elevation:", along with "Raise:" and "Interval:" fields. There are also buttons for "Simpan", "Rate: Low", "Rate: High", "None", and "Auto". Below these fields is a table with the following data:

id	agent_name	velocity	raise	interval_time	state	elevation	Edit	Delete	Start
1	s1	0.526128692367...	0	1000	None	3.002657015092	Edit	Delete	Start
3	s2	0.667617394016...	0	1000	None	3.022681049607	Edit	Delete	Start
7	s3	0.319447069622...	0	1000	None	2.688898204374	Edit	Delete	Start
8	s4	0.576765257788...	0	1000	None	3.250091510680	Edit	Delete	Start
9	w1a1	0.513811830569...	0.000029724784	1000	Raise	4.741487118463	Edit	Delete	Start
5	w1a2	0.571821158957...	0.000113439202	1000	Raise	7.372091776578	Edit	Delete	Start
6	w1a3	0.098821294237...	0.000152781722	1000	Raise	8.094390781389	Edit	Delete	Start

Gambar 4.9 Implementasi Antarmuka Aplikasi Pembangkit Data Sensor

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas tentang rangkaian pengujian dan evaluasi perangkat lunak yang dilakukan dari hasil implementasi. Pengujian dilakukan untuk menguji secara keseluruhan apakah semua fungsionalitas berjalan sesuai keinginan. Perangkat lunak dilakukan pengujian dengan menggunakan beberapa macam skenario. Pembahasan pada bab ini meliputi lingkungan pengujian, dasar pengujian, skenario pengujian, hasil pengujian dan evaluasi.

5.1 Lingkungan Pengujian

Lingkungan pengujian merupakan komputer tempat pengujian aplikasi dilakukan. Lingkungan pengujian ini menggunakan 1 unit komputer. Spesifikasi lingkungan pengujian terbagi menjadi 2, yaitu lingkungan perangkat keras dan lingkungan perangkat lunak.

5.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengujian memiliki spesifikasi seperti yang disebutkan dalam Tabel 5.1.

Tabel 5.1 Lingkungan Perangkat Keras Pengujian

No	Perangkat Keras	Spesifikasi
1	Prosesor	Prosesor Intel(R) Core™ i5 CPU 3317U @1,70GHz.
2	RAM	4.00 GB
3	Modem	PROLINK 3.75G HSUPA

5.1.2 Lingkungan Perangkat Lunak

Beberapa perangkat lunak yang digunakan dalam pengujian disebutkan dalam Tabel 5.2.

Tabel 5.2 Lingkungan Perangkat Lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows 8
2	<i>Tools</i>	Microsoft Visual Studio 2012
3	<i>Framework</i>	.NET 4.0 ke atas
4	Basis Data	MySQL 5.0

5.2 Pengujian Fungsionalitas

Pengujian pada perangkat lunak sistem yang dibangun dalam Tugas Akhir ini dilakukan dengan menggunakan metode pengujian *black box* yang menitikberatkan pada kebutuhan fungsional sistem. Pengujian ini bertujuan untuk menguji apakah sistem telah diimplementasi dengan fitur-fitur yang mengakomodir kebutuhan-kebutuhan fungsional seperti yang telah diidentifikasi pada tahap analisa sistem dengan semestinya.

5.2.1 Skenario Pengujian Fungsionalitas

Pada bagian ini akan dilakukan sejumlah pengujian perangkat lunak untuk menguji kebenaran dari perangkat lunak yang telah dibangun. Pengujian ini meliputi seluruh kasus penggunaan atau bagian-bagian dari kasus penggunaan yang telah dijelaskan pada tahap perancangan sistem. Pengujian fungsionalitas ini meliputi proses yang dijabarkan sebagai berikut:

1. Membuat wilayah pantauan.
2. Melihat pantauan sungai.
3. Melihat laporan pantauan sungai.
4. Menerima SMS peringatan.
5. Mengubah tampilan peta
6. Menghapus wilayah pantauan

Layanan *web* yang digunakan untuk mengirimkan data sensor buatan diakses melalui alamat *http://localhost/agent/agent.php*. Layanan *web* ini mempunyai parameter-parameter yang harus diisi untuk mengirimkan data sensor buatan. Data parameter yang digunakan adalah mode dan id sensor. Parameter mode yaitu tipe aksi yang dilakukan layanan *web*. Aksi yang digunakan adalah mengambil data sensor buatan. Parameter id sensor yaitu id sensor yang akan diambil datanya.

Data sensor buatan diambil oleh aplikasi SIG setiap satu menit sekali. Skenario pengiriman data sensor buatan oleh layanan *web* pada menit pertama pengiriman dapat dilihat pada Gambar 5.1. Aplikasi pembangkit data sensor membangkitkan data sensor baru pada menit berikutnya. Skenario pengiriman data sensor pada menit berikutnya dapat dilihat pada Gambar 5.2. Selanjutnya pada menit keempat pada Gambar 5.3 terjadi perubahan kenaikan data kecepatan dan tinggi muka air.

This XML file does not appear to have any style information associated with it. Th

```
▼<data>
  <velocity>0.657137066968512</velocity>
  <elevation>6.03197683949867</elevation>
  <date>2014-07-17 15:13:37</date>
</data>
```

Gambar 5.1 Skenario Layanan *Web* Pengiriman Data Sensor Buatan Pada Menit Kesatu

This XML file does not appear to have any style information associated with it. The

```
▼<data>
  <velocity>0.6592320374421696</velocity>
  <elevation>6.052926544235244</elevation>
  <date>2014-07-17 15:15:01</date>
</data>
```

Gambar 5.2 Skenario Layanan *Web* Pengiriman Data Sensor Buatan Pada Menit Ketiga

This XML file does not appear to have any style information associated with it. The

```

▼<data>
  <velocity>0.661818985439792</velocity>
  <elevation>6.078796024211463</elevation>
  <date>2014-07-17 15:16:01</date>
</data>

```

Gambar 5.3 Skenario Layanan Web Pengiriman Data Sensor Buatan Pada Menit Keempat

5.2.2 Hasil Pengujian Fungsionalitas

Hasil pengujian dari poin-poin pada skenario pada subbab sebelumnya dilampirkan pada bagian subbab ini. Berikut ini adalah hasil pengujian fungsionalitas fitur yang telah diimplementasikan pada tahap pengembangan perangkat lunak.

5.2.2.1 Pengujian Membuat Data Wilayah Pantauan Baru

Pengujian ini dilakukan terhadap subbagian membuat data wilayah pantauan pada fungsionalitas menambah wilayah pantauan. Pengujian ini dimulai ketika pengguna mengakses halaman awal dan berakhir di halaman mengelola data wilayah pantauan. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.3, sedangkan hasilnya ditunjukkan pada Gambar 5.4 dan Gambar 5.5.

Tabel 5.3 Skenario Membuat Menambah Data Wilayah Pantauan

Kode	PF-001
Tujuan Pengujian	Menguji fungsional membuat data wilayah pantauan.
Kondisi Awal	Pengguna sudah <i>login</i> dan berada di dalam halaman utama.
Data Masukan	1. Kolom nama wilayah diisi dengan “Desa Bulung”.
Prosedur Pengujian	1. Memasukkan nama wilayah yang ingin dibuat 2. Menekan tombol “Buat Baru”
Hasil yang Diharapkan	Aplikasi mengeksekusi masukkan dan nama wilayah yang dimasukkan tersimpan di basis data

Hasil yang Diperoleh	Nama wilayah yang dimasukkan telah tersimpan di basis data.
Kesimpulan	Proses menambahkan wilayah pantauan baru berhasil.
Kondisi Akhir	Menampilkan halaman mengelola data wilayah.

Gambar 5.4 Pengujian Skenario Menambahkan Wilayah Pantauan

	WILAYAH	Edit	Delete	Monitoring
	Bendungan Balangan	Edit	Delete	Monitoring
	Sungai Bengawan Solo	Edit	Delete	Monitoring
	Desa Bulung	Edit	Delete	Monitoring

Gambar 5.5 Tampilan *GridView* Nama Wilayah Berhasil Dimasukkan

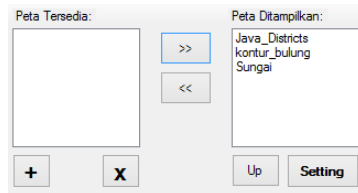
5.2.2.2 Pengujian Menambahkan Peta

Pengujian ini dilakukan terhadap subbagian menambahkan peta pada fungsionalitas menambah wilayah pantauan. Pengujian ini dimulai ketika pengguna mengakses halaman mengelola data wilayah pantauan dan berakhir di halaman yang sama. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.4 dan hasil pengujian dapat dilihat pada Gambar 5.6 dan Gambar 5.7.

Tabel 5.4 Skenario Pengujian Menambahkan Peta

Kode	PF-002
Tujuan Pengujian	Menguji fungsional menambahkan peta.
Kondisi Awal	Pengguna sudah <i>login</i> dan masuk ke dalam halaman mengelola data wilayah pantauan.

Data Masukan	<ol style="list-style-type: none"> 1. <i>File</i> peta “Java_District.shp”. 2. <i>File</i> peta “kontur_bulung.shp” 3. <i>File</i> peta “sungai.shp”
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol “+”. 2. Memilih <i>file</i> peta yang ingin dimasukkan 3. Menekan tombol “>>”.
Hasil yang Diharapkan	Aplikasi menambahkan <i>file</i> peta dan menampilkannya di aplikasi.
Hasil yang Diperoleh	Proses berhasil dieksekusi dan peta berhasil ditambahkan dan ditampilkan di aplikasi.
Kesimpulan	Proses menambahkan peta berhasil.
Kondisi Akhir	Peta ditampilkan di panel tampilan peta.



Gambar 5.6 Pengujian Skenario Menambahkan Peta



Gambar 5.7 Tampilan Panel Peta Ketika Peta Berhasil Ditambahkan

5.2.2.3 Pengujian Membuat Data Profil Sungai

Pengujian ini dilakukan terhadap subbagian membuat data profil sungai pada fungsionalitas menambah wilayah pantauan. Pengujian ini dimulai ketika pengguna mengakses halaman mengelola data wilayah pantauan dan berakhir di halaman yang sama. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.5 dan hasilnya ditunjukkan pada Gambar 5.8 dan Gambar 5.9.

Tabel 5.5 Skenario Pengujian Membuat Data Profil Sungai

Kode	PF-003
Tujuan Pengujian	Menguji fungsional membuat data profil sungai..
Kondisi Awal	Pengguna sudah <i>login</i> dan masuk ke dalam halaman mengelola data wilayah pantauan.
Data Masukan	<ol style="list-style-type: none"> 1. Kolom Section Name diisi “Section 1”. 2. Kolom Koordinat X diisi “112.925258330319”. 3. Kolom Koordinat Y diisi “-6.89921174429821”. 4. Kolom ID Sensor diisi “w1s1” 5. Kolom Debit Max diisi “649.12”. 6. Kolom Lokasi diisi “Section 1”.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol “Add Cross Section”. 2. Mekluk lokasi profil sungai di peta. 3. Mengisi kolom-kolom yang ada dengan data yang disediakan. 4. Menekan tombol “Save”.
Hasil yang Diharapkan	Aplikasi menambahkan data profil sungai baru ke basis data.
Hasil yang Diperoleh	Proses berhasil dieksekusi dan data berhasil ditambahkan ke dalam basis data.
Kesimpulan	Proses menambahkan data profil sungai baru berhasil.
Kondisi Akhir	Data ditampilkan pada <i>GridView</i> data profil sungai.

Informasi Sungai

Section Name:

Koordinat X:

Koordinat Y:

ID Sensor:

Debit Max (Q): (m³/s)

Lokasi:

Gambar 5.8 Pengujian Skenario Menambahkan Data Profil Sungai

	CROSS SECTION		
▶	Section 1	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
	Section 2	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

Gambar 5.9 Tampilan *GridView* Data Profil Sungai Berhasil Dimasukkan

5.2.2.4 Pengujian Menambah Data Penampang Sungai

Pengujian ini dilakukan terhadap subbagian menambah data penampang pada fungsionalitas menambah wilayah pantauan. Pengujian ini dimulai ketika pengguna mengakses halaman mengelola data sungai dan berakhir di halaman mengelola data wilayah pantauan. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.6, sedangkan hasilnya ditunjukkan pada Gambar 5.10.

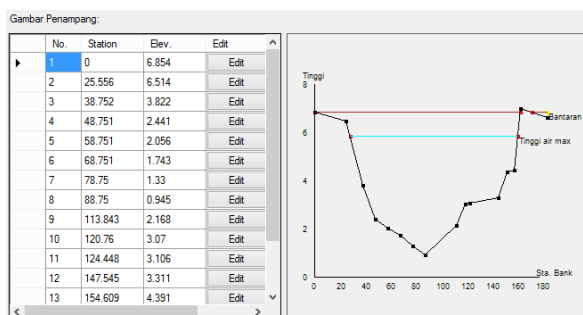
Tabel 5.6 Skenario Pengujian Menambah Data Penampang Sungai

Kode	PF-004
Tujuan Pengujian	Menguji fungsional menambah data penampang sungai.
Kondisi Awal	Pengguna sudah <i>login</i> dan masuk ke dalam halaman

	mengelola data sungai.
Data Masukan	1. Kolom Sta bank dan elevation diisi dengan data yang disediakan.
Prosedur Pengujian	1. Mengisi kolom Sta bank dan elevation. 2. Menekan tombol “Add”.
Hasil yang Diharapkan	Data penampang sungai dimasukkan ke basis data dan digambarkan pada panel gambar penampang sungai.
Hasil yang Diperoleh	Data berhasil masuk ke dalam basis data dan gambar penampang sungai dapat ditampilkan.
Kesimpulan	Proses menambah data penampang sungai berhasil.
Kondisi Akhir	Pengguna kembali ke halaman mengelola data wilayah pantauan.

5.2.2.5 Pengujian Menambah Data Aliran Sungai

Pengujian ini dilakukan terhadap subbagian menambah data aliran sungai pada fungsionalitas menambah wilayah pantauan. Pengujian ini dimulai ketika pengguna mengakses halaman mengelola data wilayah pantauan dan berakhir di halaman yang sama. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.7, sedangkan hasilnya ditunjukkan pada Gambar 5.11 dan Gambar 5.12.



Gambar 5.10 Pengujian Skenario Menambahkan Data Penampang Sungai

Tabel 5.7 Skenario Pengujian Menambah Data Aliran Sungai

Kode	PF-005
Tujuan Pengujian	Menguji fungsional menambah data aliran sungai.
Kondisi Awal	Pengguna sudah <i>login</i> dan masuk ke dalam halaman mengelola data wilayah pantauan.
Data Masukan	<ol style="list-style-type: none"> 1. Kolom <i>From</i> diisi “Section 1”. 2. Kolom <i>To</i> diisi “Section 2”.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol “<i>draw River Flow</i>”. 2. Menggambar aliran pada peta. 3. Mengisi kolom-kolom yang ada dengan data yang disediakan. 4. Menekan tombol “<i>add</i>”.
Hasil yang Diharapkan	Data aliran sungai berhasil dimasukkan ke dalam basis data.
Hasil yang Diperoleh	Proses berhasil dieksekusi dan data berhasil dimasukkan ke dalam basis data.
Kesimpulan	Proses menambahkan data aliran sungai berhasil.
Kondisi Akhir	Data aliran sungai ditampilkan pada <i>GridView</i> data aliran sungai.

The image shows a screenshot of a web application window titled "Form Data Aliran". The main heading is "Aliran Sungai". Below the heading, there are two rows of input fields. The first row is labeled "Dari:" and contains a dropdown menu with "Section 1" selected, followed by an "Add" button. The second row is labeled "Ke:" and contains a dropdown menu with "Section 2" selected, followed by a "Cancel" button. The window has a standard Windows-style title bar with a close button (red X) on the right.

Gambar 5.11 Pengujian Skenario Menambah Data Aliran Sungai

	ALIRAN DARI	ALIRAN KE		
	None	Section 1	View	Delete
	Section 1	Section 2	View	Delete

Gambar 5.12 Tampilan *GridView* Data Aliran Berhasil Dimasukkan

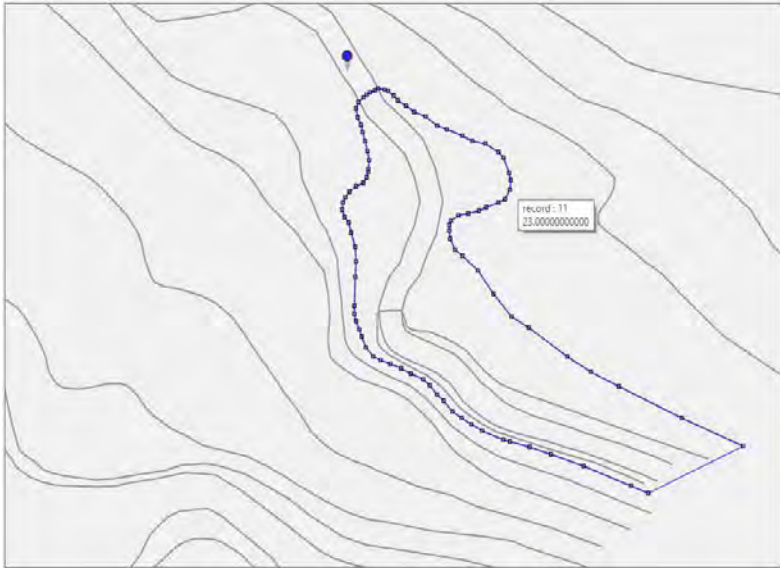
5.2.2.6 Pengujian Menghitung Luas Area Dan Volume Kontur

Pengujian ini dilakukan terhadap subbagian menghitung luas area dan volume kontur pada fungsionalitas menambah wilayah pantauan. Pengujian ini dimulai ketika pengguna mengakses halaman mengelola data wilayah pantauan dan berakhir di halaman yang sama. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.8, sedangkan hasilnya ditunjukkan pada Gambar 5.13, Gambar 5.14 dan Gambar 5.15.

Tabel 5.8 Skenario Pengujian Menghitung Luas Area Dan Volume Kontur

Kode	PF-006
Tujuan Pengujian	Menguji fungsional menghitung luas area dan volume kontur.
Kondisi Awal	Pengguna sudah <i>login</i> dan masuk ke dalam halaman mengelola data wilayah pantauan.
Data Masukan	A1. Kolom elevasi diisi “23”.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Menekan tombol “<i>Draw to calculate area</i>”. 2. Menggambar luas area yang ingin dihitung. 3. Menekan tombol “s”. 4. Mengisi kolom “elevation” 5. Menekan tombol “<i>add</i>”.
Hasil yang Diharapkan	Data area kontur berhasil dimasukkan ke dalam basis data.
Hasil yang Diperoleh	Proses berhasil dieksekusi dan data berhasil dimasukkan ke dalam basis data.

Kesimpulan	Proses menghitung luas area dan volume kontur berhasil.
Kondisi Akhir	Data area kontur ditampilkan pada <i>GridView</i> data area.



Gambar 5.13 Pengujian Menggambar Area Yang Ingin Dihitung

Area Calculation

Elevasi:

Area :

Gambar 5.14 Pengujian Mengisikan Data Elevasi Kontur

			ELEVASI	LUAS AREA	L T
	<input type="button" value="View"/>	<input type="button" value="Delete"/>	22.00	55.5	0
	<input type="button" value="View"/>	<input type="button" value="Delete"/>	23.00	224	27

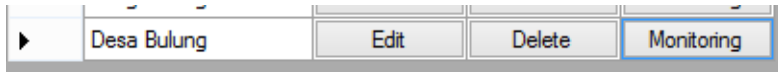
Gambar 5.15 Tampilan *GridView* Data Area Berhasil Ditambahkan

5.2.2.7 Pengujian Melihat Pantauan Sungai

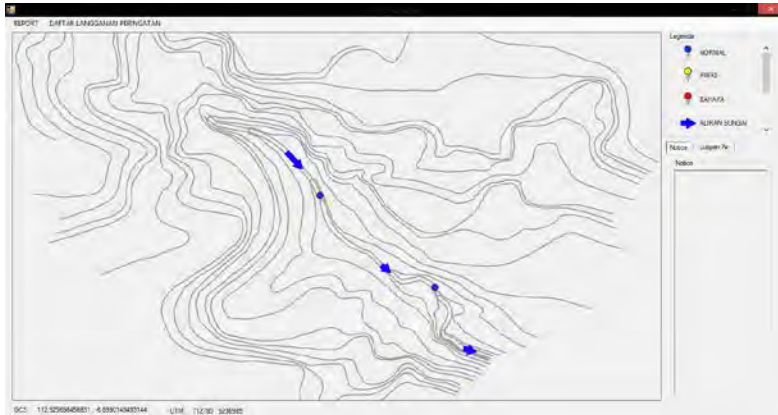
Pengujian ini dilakukan terhadap subbagian melihat pantauan sungai pada fungsionalitas melihat hasil pantauan sungai. Pengujian ini dimulai ketika pengguna mengakses halaman melihat hasil pantauan sungai. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.9, sedangkan hasilnya ditunjukkan pada Gambar 5.16 dan Gambar 5.17.

Tabel 5.9 Skenario Pengujian Melihat Pantauan Sungai

Kode	PF-007
Tujuan Pengujian	Menguji fungsional melihat pantauan sungai
Kondisi Awal	Pengguna masuk ke dalam halaman utama.
Data Masukan	-
Prosedur Pengujian	1. Menekan tombol “monitoring”.
Hasil yang Diharapkan	Halaman melihat hasil pantauan sungai ditampilkan.
Hasil yang Diperoleh	Halaman melihat hasil pantauan sungai berhasil ditampilkan
Kesimpulan	Proses melihat hasil pantauan sungai berhasil.
Kondisi Akhir	Halaman melihat hasil pantauan sungai ditampilkan.



Gambar 5.16 Skenario Pengujian Melihat Pantauan Sungai



Gambar 5.17 Tampilan Halaman Pantauan Sungai

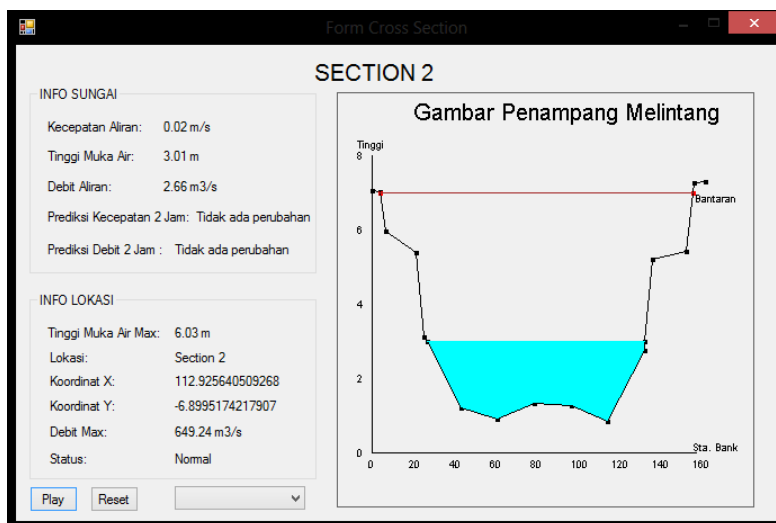
5.2.2.8 Pengujian Melihat Laporan Pantauan Sungai

Pengujian ini dilakukan terhadap subbagian melihat laporan pantauan sungai pada fungsionalitas melihat laporan pantauan sungai. Pengujian ini dimulai ketika pengguna berada pada halaman melihat hasil pantauan. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.10, sedangkan hasilnya ditunjukkan pada Gambar 5.18.

Tabel 5.10 Skenario Pengujian Melihat Laporan Pantauan

Kode	PF-008
Tujuan Pengujian	Menguji fungsional melihat laporan pantauan.
Kondisi Awal	Pengguna berada pada antarmuka melihat hasil pantauan
Data Masukan	-
Prosedur	1. Menekan pin “Section 2” yang muncul dipeta

Pengujian	pantauan.
Hasil yang Diharapkan	Antarmuka laporan pantauan profil sungai ditampilkan.
Hasil yang Diperoleh	Proses berhasil dieksekusi dan halaman laporan pantauan profil sungai ditampilkan.
Kesimpulan	Proses melihat laporan pantauan sungai berhasil.
Kondisi Akhir	Antarmuka laporan pantauan profil sungai ditampilkan.



Gambar 5.18 Tampilan Halaman Laporan Pantauan Profil Sungai

5.2.2.9 Pengujian Menerima SMS Peringatan

Pengujian ini dilakukan terhadap subbagian menerima SMS peringatan pada fungsionalitas menerima SMS peringatan. Pengujian ini dimulai ketika halaman melihat hasil pantauan sungai aktif. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.11, sedangkan hasilnya ditunjukkan pada Gambar 5.19.

Tabel 5.11 Skenario Pengujian Menerima SMS Peringatan

Kode	PF-009
Tujuan Pengujian	Menguji fungsional menerima SMS Peringatan.
Kondisi Awal	Antarmuka pantauan sungai aktif dan nomor pengguna telah terdaftar di aplikasi.
Data Masukan	-
Prosedur Pengujian	1. Kondisi Section 2 dimanipulasi agar memenuhi kondisi pengiriman peringatan.
Hasil yang Diharapkan	SMS peringatan diterima pengguna.
Hasil yang Diperoleh	Aplikasi mengirimkan SMS peringatan ke pengguna.
Kesimpulan	Proses menerima SMS peringatan berhasil.
Kondisi Akhir	SMS peringatan diterima pengguna.

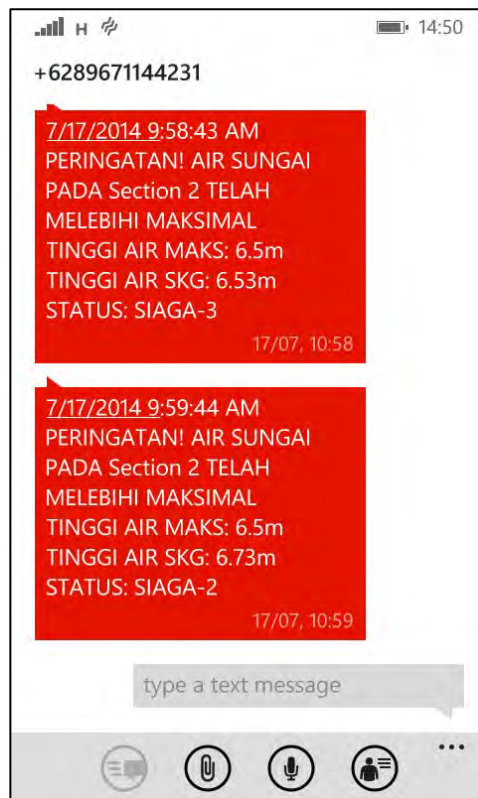
5.2.2.10 Pengujian Mengubah Tampilan Peta

Pengujian ini dilakukan terhadap subbagian mengubah tampilan peta pada fungsionalitas mengubah tampilan peta. Pengujian ini dimulai ketika pengguna berada pada halaman mengelola data pantauan sungai dan berakhir pada halaman yang sama. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.12, sedangkan hasilnya ditunjukkan pada Gambar 5.20 dan Gambar 5.21.

Tabel 5.12 Skenario Pengujian Mengubah Tampilan Peta

Kode	PF-010
Tujuan Pengujian	Menguji fungsional mengubah tampilan peta.
Kondisi Awal	Pengguna berada di antarmuka mengubah setelan peta.
Data Masukan	1. Warna peta diubah menjadi kuning.
Prosedur Pengujian	1. Menekan tombol “edit” pada peta “Java_District.shp” 2. Mengubah warna menjadi kuning.

	3. Menekan tombol “simpan”. 4. Menutup halaman ubah setelan peta.
Hasil yang Diharapkan	Tampilan peta berubah menjadi warna kuning.
Hasil yang Diperoleh	Proses berhasil dieksekusi dan peta ditampilkan dengan warna kuning.
Kesimpulan	Proses mengubah tampilan peta berhasil.
Kondisi Akhir	Warna peta berubah.



Gambar 5.19 Hasil Pengujian Menerima SMS Peringatan

Index:	<input type="text" value="0"/>	Warna:	<input type="text" value="Yellow"/>	Use Tooltip:	<input type="text" value="0"/>	<input type="button" value="Simpan"/>
PenWidthMin:	<input type="text" value="1"/>	Type:	<input type="text" value="Dataran"/>	Line Set:	<input type="text" value="manual"/>	<input type="button" value="Batal"/>
PenWidthMax:	<input type="text" value="2"/>	Field Label:	<input type="text" value="0"/>	<input type="button" value="View Data"/>		

Gambar 5.20 Skenario Pengujian Mengubah Tampilan Peta



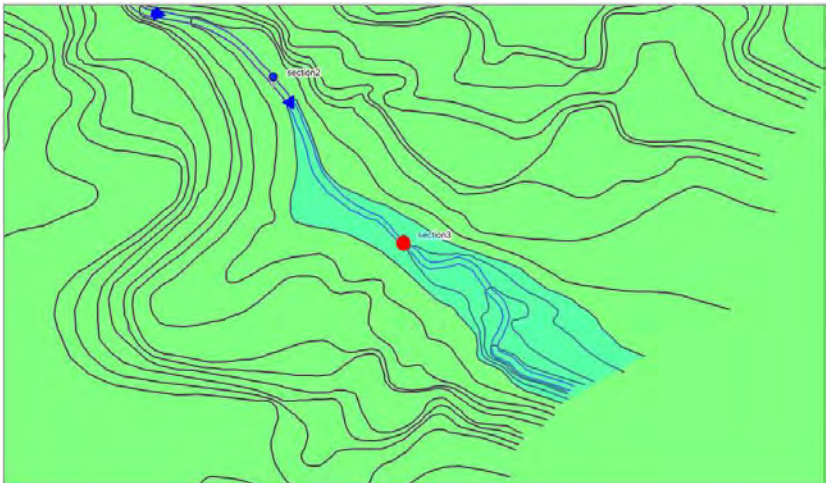
Gambar 5.21 Tampilan Peta Berhasil Diubah

5.2.2.11 Pengujian Menampilkan Tinggi Genangan Air

Pengujian ini dilakukan untuk melihat hasil keluaran fungsionalitas menampilkan tinggi genangan air. Pengujian ini dimulai ketika admin pengujian berada pada antarmuka awal aplikasi simulasi data sensor. Admin pengujian mengubah data sensor pada section 3 menjadi melebihi maksimal. Penjelasan terperinci mengenai skenario pengujian ini dapat dilihat pada Tabel 5.13 dan hasil pengujian dapat dilihat pada Gambar 5.22.

Tabel 5.13 Skenario Pengujian Menampilkan Tinggi Genangan Air

Kode	PF-011
Tujuan Pengujian	Menguji fungsional menampilkan tinggi genangan air.
Kondisi Awal	Pengguna masuk ke antarmuka melihat hasil pantauan
Data Masukan	1. Data sensor dari potongan sungai yang meluap.
Prosedur Pengujian	1. Admin pengawas sungai berada di antarmuka melihat hasil pantauan. 2. Admin pengujian mengubah data sensor menjadi melebihi maksimal.
Hasil yang Diharapkan	Tinggi genangan air ditampilkan pada antarmuka melihat hasil pantauan.
Hasil yang Diperoleh	Proses berhasil dieksekusi dan tinggi genangan air ditampilkan.
Kesimpulan	Proses menampilkan tinggi genangan air berhasil.
Kondisi Akhir	Gambar tinggi genangan air ditampilkan.

**Gambar 5.22 Hasil Pengujian Menampilkan Tinggi Genangan Air**

5.2.2.12 Pengujian Menghapus Wilayah Pantauan

Pengujian ini dilakukan terhadap subbagian menghapus wilayah pantauan pada fungsionalitas menghapus wilayah pantauan. Pengujian ini dimulai ketika pengguna berada pada halaman utama dan berakhir pada halaman yang sama. Penjelasan terperinci mengenai skenario pengujian fungsionalitas ini dapat dilihat pada Tabel 5.14, sedangkan hasilnya ditunjukkan pada Gambar 5.23 dan Gambar 5.24.

Tabel 5.14 Skenario Pengujian Menghapus Wilayah Pantauan

Kode	PF-012
Tujuan Pengujian	Menguji fungsional menghapus wilayah pantauan
Kondisi Awal	Pengguna masuk ke antarmuka beranda.
Data Masukan	1. Data wilayah pantauan “Sungai A”.
Prosedur Pengujian	1. Menekan tombol “delete” pada data wilayah pantauan “Sungai A”.
Hasil yang Diharapkan	Data wilayah pantauan terhapus dari basis data.
Hasil yang Diperoleh	Proses berhasil dieksekusi dan data wilayah pantauan dihapus dari basis data.
Kesimpulan	Proses menghapus data wilayah pantauan berhasil.
Kondisi Akhir	Data wilayah pantauan dihapus.

	3	Sungai Bengawa...	Edit	Delete	Monitoring
▶	4	Sungai A	Edit	Delete	Monitoring

Gambar 5.23 Skenario Pengujian Menghapus Wilayah Pantauan

	ID	WILAYAH	Edit	Delete	Monitoring
▶	1	Bendungan Bala...	Edit	Delete	Monitoring
	3	Sungai Bengawa...	Edit	Delete	Monitoring

Gambar 5.24 Tampilan *GridView* Wilayah Pantauan Setelah Penghapusan

5.2.3 Evaluasi Hasil Pengujian Fungsionalitas

Berdasarkan hasil pengujian fungsionalitas aplikasi, semua skenario berhasil dilakukan. Evaluasi terhadap pengujian yang telah dilaksanakan dijelaskan sebagai berikut.

1. Fungsionalitas membuat wilayah pantauan berjalan sesuai dengan yang diharapkan.
2. Fungsionalitas melihat pantauan sungai berjalan sesuai dengan yang diharapkan.
3. Fungsionalitas laporan pantauan sungai berjalan sesuai dengan yang diharapkan.
4. Fungsionalitas menerima SMS peringatan berjalan sesuai dengan yang diharapkan.
5. Fungsionalitas menampilkan tinggi genangan air berjalan sesuai dengan yang diharapkan.
6. Fungsionalitas mengubah tampilan peta berjalan sesuai dengan yang diharapkan.
7. Fungsionalitas menghapus wilayah pantauan berjalan sesuai dengan yang diharapkan.

Dari keseluruhan hasil pengujian fungsionalitas terhadap aplikasi yang dibangun dalam Tugas Akhir ini, semua skenario telah berhasil dilakukan. Aplikasi telah berjalan sesuai dengan yang diharapkan.

5.3 Pengujian Kegunaan

Selain melakukan pengujian fungsionalitas dengan metode *black box*, juga dilakukan pengujian kegunaan untuk menilai kegunaan aplikasi secara langsung kepada pengguna. Pengujian kegunaan dilakukan untuk mengetahui penilaian dan tanggapan dari pengguna terhadap sejumlah aspek dari aplikasi ini.

5.3.1 Kriteria Responden

Pengujian kegunaan dilakukan dengan melibatkan beberapa pengguna dari aplikasi ini. Para pengguna yang terlibat dalam

pengujian ini selanjutnya disebut sebagai responden. Para responden diberikan keleluasaan untuk memakai aplikasi pada *laptop* yang telah disediakan. Yang ditunjuk sebagai responden adalah pengguna yang mengerti mengenai data-data yang digunakan pada aplikasi sehingga penulis mengambil sampel 4 orang mahasiswa teknik sipil ITS. Daftar responden dapat dilihat pada Tabel 5.15.

Tabel 5.15 Daftar Responden

Nama	Profesi
Bebby Idhiani Nikita	Mahasiswa S1 Teknik Sipil ITS
Deka Agrapradana	Mahasiswa S1 Teknik Sipil ITS
Muhammad Zaini Gani	Mahasiswa S1 Teknik Sipil ITS
Hary Puji A.	Mahasiswa S2 Teknik Sipil ITS

5.3.2 Skenario Pengujian Kegunaan

Dalam melakukan pengujian kegunaan aplikasi, responden diminta untuk memakai aplikasi sesuai dengan prosedur pengujian yang telah disediakan. Dalam memberikan penilaian terhadap aplikasi, responden mengisi formulir penilaian yang telah disediakan untuk pengujian ini. Formulir penilaian memiliki beberapa aspek penilaian secara kualitatif. Formulir penilaian pengujian kegunaan aplikasi yang digunakan adalah sebagaimana pada LAMPIRAN A KUESIONER PENGUJIAN KEGUNAAN.

5.3.3 Hasil Pengujian Kegunaan

Pada penilaian pengujian kegunaan, terdapat empat pilihan nilai untuk setiap pertanyaan yaitu tidak baik, kurang baik, cukup baik dan sangat baik. Untuk menghitung hasil pengujian, maka dilakukan pengubahan nilai tersebut menjadi nilai-nilai angka dan persentase seperti pada Tabel 5.16. Beberapa hasil pengujian pada responden dapat dilihat pada LAMPIRAN B HASIL PENGUJIAN KEGUNAAN.

Tabel 5.16 Daftar Perubahan Penilaian pada Pengujian

No	Nilai pada Kuesioner	Nilai Angka	Persentase
1	Tidak	1	25%
2	Kurang	2	50%
3	Cukup	3	75%
4	Sangat	4	100%

Pengujian dibagi menjadi tiga aspek yaitu antarmuka pengguna, penyampaian informasi kondisi sungai dan penyampaian peringatan SMS. Berikut ini adalah rekapitulasi penilaian dari hasil pengujian kegunaan. Rekapitulasi penilaian antarmuka pengguna dapat dilihat pada Tabel 5.17. Rekapitulasi penilaian penyampaian informasi kondisi sungai dapat dilihat pada Tabel 5.18. Rekapitulasi penilaian penyampaian peringatan SMS dapat dilihat pada Tabel 5.19.

Tabel 5.17 Penilaian Antarmuka Pengguna

No	Penilaian				Rata-rata
	Tidak	Kurang	Cukup	Sangat	
1	0	1	3	0	2,75
2	0	0	3	1	3,25
3	0	0	1	3	3,75
Nilai Akhir					3,25

Tabel 5.18 Penilaian Penyampaian Informasi Kondisi Sungai

No	Penilaian				Rata-rata
	Tidak	Kurang	Cukup	Sangat	
1	0	1	0	3	3,5
2	0	0	3	1	3,25
3	0	0	2	2	3,5
Nilai Akhir					3,41

Tabel 5.19 Penilaian Penyampaian Peringatan SMS

No	Penilaian				Rata-rata
	Tidak	Kurang	Cukup	Sangat	

1	0	0	1	3	3,75
2	0	0	0	4	4,0
Nilai Akhir					3,87

5.3.4 Evaluasi Hasil Pengujian Kegunaan

Untuk menganalisis pengujian kegunaan aplikasi, maka dilakukan rekapitulasi akhir. Rekapitulasi akhir menghasilkan nilai persentase terhadap aspek-aspek yang dinilai dalam pengujian kegunaan. Rekapitulasi akhir ini ditunjukkan dalam Tabel 5.20.

Tabel 5.20 Rekapitulasi Akhir Pengujian Kegunaan

No	Aspek Pengujian	Nilai Akhir	Persentase
1	Antarmuka Pengguna	3,25	81,2%
2	Penyampaian Informasi Kondisi Sungai	3,41	85,25%
3	Penyampaian Peringatan SMS	3,87	96,75%

Dari Tabel 5.20, dapat dilihat bahwa aspek-aspek yang dinilai menghasilkan nilai-nilai persentase. Penilaian terhadap antarmuka pengguna menghasilkan persentase sebesar 81,2%. Penilaian terhadap penyampaian informasi kondisi sungai menghasilkan persentase sebesar 85,25%. Sedangkan penilaian terhadap penyampaian peringatan SMS menghasilkan persentase sebesar 96,75%. Ketiga aspek menunjukkan persentase nilai yang lebih dari 75% sehingga dapat disimpulkan bahwa dari segi kegunaan, perangkat lunak yang dibuat pada Tugas Akhir ini memberikan hasil yang lebih dari cukup baik.

BAB VI PENUTUP

Dalam bab ini dibahas kesimpulan yang diambil dari tujuan pembuatan aplikasi, serta hasil pengujian yang telah dilakukan. Selain itu, terdapat beberapa saran untuk pengembangan lebih lanjut.

6.1 Kesimpulan

Dari proses pengerjaan selama perancangan, implementasi, dan proses pengujian aplikasi yang dilakukan, dapat diambil kesimpulan sebagai berikut.

1. Aplikasi berhasil menampilkan lapisan peta dan data hasil pantauan dibantu dengan pustaka Easy GIS .NET. Selanjutnya data spasial wilayah seperti kontur diolah dengan menggunakan teknik pemetaan yaitu metode koordinat dan metode kontur untuk mendapatkan volume kontur.
2. Data sensor yang dikirimkan berformat XML sehingga menggunakan XMLParser agar aplikasi bisa membaca data yang dikirimkan. Kemudian aplikasi mengolah data yang didapat sehingga dapat memberikan peringatan. Peringatan yang diberikan ada dua kondisi:
 1. Peringatan diberikan apabila tinggi muka air melebihi maksimal dan mendekati tinggi bantaran sungai.
 2. Peringatan diberikan apabila rata-rata kenaikan tinggi muka air dan kecepatan dua jam kedepan melebihi batas maksimal.
3. Aplikasi dapat mengolah data kontur wilayah untuk memperkirakan tinggi genangan air yang meluap.

6.2 Saran

Dalam pembuatan Tugas Akhir ini, terdapat beberapa saran untuk perbaikan serta pengembangan dari aplikasi yang telah dikerjakan untuk kedepannya, yakni sebagai berikut.

1. Sistem peringatan dini dapat lebih dikembangkan dengan menggunakan parameter masukan yang lebih banyak sebagai informasi untuk pengambilan keputusan sesuai kebutuhan.
2. Menambahkan lapisan peta yang lebih banyak, seperti peta jalan lengkap dan peta kawasan perumahan untuk dapat memperkirakan kerugian materil.

DAFTAR PUSTAKA

- [1] M. Middlemann, B. Harper dan R. Lacey, "Flood Risks," Geoscience Australia, Australia, 2008.
- [2] R. Junction, "Risk Junction," [Online]. Available: <http://riskjunction.blogspot.com/2009/06/mitigasi-risiko-apaan-sih.html>. [Diakses 13 Juni 2014].
- [3] P. I. B. Aceh, "PIBA," [Online]. Available: <http://piba.tdmrc.org/content/istilah-dalam-peringatan-dini>. [Diakses 11 Oktober 2013].
- [4] D. M. Darmawan, "Sistem Informasi Geografis (SIG) dan Standarisasi Pemetaan Tematik," BAKOSURTANAL, Jakarta, 2011.
- [5] P. SDA, "Tech4Water," [Online]. Available: <http://www.tech4water.com/id/>. [Diakses 13 Juni 2014].
- [6] B. P. NTB, "Tutorial Dasar ArcGIS10," Bappeda Provinsi NTB, NTB, 2006.
- [7] S. Khomsin, "Pendidikan dan Pelatihan (Diklat) Teknis Pengukuran dan Pemetaan Kota," FTSP, Surabaya, 2004.
- [8] M. Ir. Yuwono, "Pendidikan dan Pelatihan (Diklat) Teknis Pengukuran dan Pemetaan Kota," FTSP, Surabaya, 2004.
- [9] Microsoft, "Overview of .NET framework," Microsoft, [Online]. Available: <http://msdn.microsoft.com/en-us/library/zw4w595w.aspx>. [Diakses 27 September 2013].
- [10] Oracle, "Why MySQL," Oracle Corporation, [Online]. Available: <http://www.mysql.com/why-mysql/>. [Diakses 27 September 2013].
- [11] E. S. R. Institute, "ESRI Shapefile Technical Description," ESRI, United States of America, 1998.
- [12] W. Fletcher, "Easy GIS .NET," [Online]. Available: <http://www.easygisdotnet.com/About.aspx>. [Diakses 27 September 2013].

- [13] Marcinello dan Nijel, "Gammu SMS gateway," [Online]. Available: <http://sourceforge.net/projects/gammu/>. [Diakses 27 September 2013].

BIODATA PENULIS



Penulis, Andreyan Rizky Baskara, lahir di Banjarmasin, Kalimantan Selatan, pada tanggal 3 Juli 1993. Penulis menyelesaikan pendidikan Sekolah Dasar di SD Pasar Lama 2 Banjarmasin, SMP Negeri 1 Banjarmasin, dan SMAN 1 Banjarmasin. Setelah tamat sekolah menengah atas, penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya melalui jalur Mandiri sebagai pilihan pertamanya. Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Rekayasa Perangkat Lunak (*Software Engineering*).

Selama berada di dunia akademi kampus, Penulis sempat aktif di beberapa lomba pengembangan aplikasi tingkat regional dan Institut. Beberapa prestasi yang pernah diraih oleh penulis adalah juara pertama pada lomba pengembangan aplikasi tingkat institut di ITS. Penulis juga aktif dalam bidang nonakademik dengan menjadi menjadi panitia maupun peserta di berbagai kegiatan jurusan maupun institut.

Penulis dapat dihubungi via e-mail ke andreyanrb@gmail.com.

LAMPIRAN A – FORMULIR PENILAIAN PENGUJIAN KEGUNAAN

Kuesioner Pengujian Kegunaan Aplikasi GIS Sebagai Sistem Peringatan Dini Untuk Mitigasi Risiko Bencana Banjir Dengan Peningat Berupa Sms

Data Responden

Tanggal :

Nama :

Profesi :

A. Antarmuka Pengguna

1. Apakah aplikasi ini memiliki antarmuka pengguna yang menarik?
a. Tidak b. Kurang c. Cukup d. Sangat
2. Apakah antarmuka aplikasi ini mudah dipahami?
a. Tidak b. Kurang c. Cukup d. Sangat
3. Apakah aplikasi ini sudah memiliki tata letak yang baik?
a. Tidak b. Kurang c. Cukup d. Sangat

B. Penyampaian Informasi Kondisi Sungai

1. Apakah informasi mengenai kondisi sungai yang ditampilkan aplikasi mudah dipahami?
a. Tidak b. Kurang c. Cukup d. Sangat
2. Apakah informasi kondisi sungai yang ditampilkan aplikasi sudah cukup informatif?
a. Tidak b. Kurang c. Cukup d. Sangat
3. Apakah informasi yang ditampilkan sudah sesuai dengan data simulasi yang digunakan?
a. Tidak b. Kurang c. Cukup d. Sangat

C. Penyampaian Peringatan Melalui SMS

1. Apakah SMS peringatan yang dikirimkan sudah sesuai dengan kondisi simulasi yang dilakukan?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
2. Apakah SMS peringatan yang dikirimkan mudah dipahami?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat

LAMPIRAN B – HASIL PENGUJIAN KEGUNAAN

Kuesioner Pengujian Kegunaan Aplikasi GIS Sebagai Sistem Peringatan Dini Untuk Mitigasi Risiko Bencana Banjir Dengan Peningkat Berupa SMS

Data Responden
Tanggal : 16 Juni 2014
Nama : REBBY IDHANI NIKITR
Profesi : Mahasiswa

A. Antarmuka Pengguna

1. Apakah aplikasi ini memiliki antarmuka pengguna yang menarik?
a. Tidak b. Kurang c. Cukup d. Sangat
2. Apakah antarmuka aplikasi ini mudah dipahami?
a. Tidak b. Kurang c. Cukup d. Sangat
3. Apakah aplikasi ini sudah memiliki tata letak yang baik?
a. Tidak b. Kurang c. Cukup d. Sangat

B. Penyampaian Informasi Kondisi Sungai

1. Apakah informasi mengenai kondisi sungai yang ditampilkan aplikasi mudah dipahami?
a. Tidak b. Kurang c. Cukup d. Sangat
2. Apakah informasi kondisi sungai yang ditampilkan aplikasi sudah informatif?
a. Tidak b. Kurang c. Cukup d. Sangat
3. Apakah informasi yang ditampilkan sudah sesuai dengan data simulasi yang digunakan?
a. Tidak b. Kurang c. Cukup d. Sangat

C. Penyampaian Peringatan Melalui SMS

1. Apakah SMS peringatan yang dikirimkan sudah sesuai dengan kondisi simulasi yang dilakukan?
a. Tidak b. Kurang c. Cukup d. Sangat
2. Apakah isi SMS peringatan yang dikirimkan mudah dipahami?
a. Tidak b. Kurang c. Cukup d. Sangat

Saran :

- Perunjuk penggunaan lebih diperjelas ya
- Interface-nya kaku kesannya

Gambar 9.1 Bukti Pengujian Kegunaan (1)

**Kuesioner Pengujian Kegunaan
Aplikasi GIS Sebagai Sistem Peringatan Dini Untuk Mitigasi Risiko Bencana Banjir
Dengan Peningkat Berupa SMS**

Data Responden

Tanggal : 12 Juni 2014
 Nama : Hary Puji - A
 Profesi : Mahasiswa

A. Antarmuka Pengguna

1. Apakah aplikasi ini memiliki antarmuka pengguna yang menarik?
 a. Tidak b. Kurang Cukup d. Sangat
2. Apakah antarmuka aplikasi ini mudah dipahami?
 a. Tidak b. Kurang Cukup d. Sangat
3. Apakah aplikasi ini sudah memiliki tata letak yang baik?
 a. Tidak b. Kurang Cukup d. Sangat

B. Penyampaian Informasi Kondisi Sungai

1. Apakah informasi mengenai kondisi sungai yang ditampilkan aplikasi mudah dipahami?
 a. Tidak Kurang c. Cukup d. Sangat
2. Apakah informasi kondisi sungai yang ditampilkan aplikasi sudah informatif?
 a. Tidak b. Kurang Cukup d. Sangat
3. Apakah informasi yang ditampilkan sudah sesuai dengan data simulasi yang digunakan?
 a. Tidak b. Kurang Cukup d. Sangat

C. Penyampaian Peringatan Melalui SMS

1. Apakah SMS peringatan yang dikirimkan sudah sesuai dengan kondisi simulasi yang dilakukan?
 a. Tidak b. Kurang Cukup d. Sangat
2. Apakah isi SMS peringatan yang dikirimkan mudah dipahami?
 a. Tidak b. Kurang c. Cukup Sangat

Gambar 9.2 Bukti Pengujian Kegunaan (2)

**Kuesioner Pengujian Kegunaan
Aplikasi GIS Sebagai Sistem Peringatan Dini Untuk Mitigasi Risiko Bencana Banjir
Dengan Peningkat Berupa SMS**

Data Responden

Tanggal : 22 Juni 2014
 Nama : DEKA AGRA PRADILAKA
 Profesi : Mahasiswa

A. Antarmuka Pengguna

1. Apakah aplikasi ini memiliki antarmuka pengguna yang menarik?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
2. Apakah antarmuka aplikasi ini mudah dipahami?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
3. Apakah aplikasi ini sudah memiliki tata letak yang baik?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat

B. Penyampaian Informasi Kondisi Sungai

1. Apakah informasi mengenai kondisi sungai yang ditampilkan aplikasi mudah dipahami?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
2. Apakah informasi kondisi sungai yang ditampilkan aplikasi sudah cukup informatif?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
3. Apakah informasi yang ditampilkan sudah sesuai dengan data simulasi yang digunakan?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat

C. Penyampaian Peringatan Melalui SMS

1. Apakah SMS peringatan yang dikirimkan sudah sesuai dengan kondisi simulasi yang dilakukan?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
2. Apakah SMS peringatan yang dikirimkan mudah dipahami?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat

Saran/Komentar:

Gambar 9.3 Bukti Pengujian Kegunaan (3)

**Kuesioner Pengujian Kegunaan
Aplikasi GIS Sebagai Sistem Peringatan Dini Untuk Mitigasi Risiko Bencana Banjir
Dengan Pengingat Berupa SMS**

Data Responden

Tanggal : 20.06.2014
 Nama : MUHAMMAD ZAIM GANI
 Profesi : Mahasiswa Teknik Sipil

A. Antarmuka Pengguna

1. Apakah aplikasi ini memiliki antarmuka pengguna yang menarik?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
2. Apakah antarmuka aplikasi ini mudah dipahami?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
3. Apakah aplikasi ini sudah memiliki tata letak yang baik?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat

B. Penyampaian Informasi Kondisi Sungai

1. Apakah informasi mengenai kondisi sungai yang ditampilkan aplikasi mudah dipahami?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
2. Apakah informasi kondisi sungai yang ditampilkan aplikasi sudah cukup informatif?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
3. Apakah informasi yang ditampilkan sudah sesuai dengan data simulasi yang digunakan?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat

C. Penyampaian Peringatan Melalui SMS

1. Apakah SMS peringatan yang dikirimkan sudah sesuai dengan kondisi simulasi yang dilakukan?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat
2. Apakah SMS peringatan yang dikirimkan mudah dipahami?
 - a. Tidak
 - b. Kurang
 - c. Cukup
 - d. Sangat

Saran/Komentar:

Gambar 9.4 Bukti Pengujian Kegunaan (4)