



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS141501

**RANCANG BANGUN APLIKASI
CROWDSOURCING PEMETAAN KERUSAKAN
JALAN DENGAN MENGGUNAKAN GOOGLE
MAPS API DAN SMS GATEWAY
(STUDI KASUS : BINA MARGA PROVINSI JAWA
TIMUR)**

YOHANDA
NRP 5212 100 113

Dosen Pembimbing I
Nur Aini Rakhmawati, S.Kom., M.Sc. Eng.

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - KS141501

**CROWDSOURCING APPLICATION FOR ROAD
DAMAGE MAPPING USING *GOOGLE MAPS API*
AND *SMS GATEWAY*
(CASE STUDY : BINA MARGA OF EAST JAVA)**

YOHANDA
NRP 5212 100 113

Academic Promotor I
Nur Aini Rakhmawati, S.Kom., M.Sc. Eng.

INFORMATION SYSTEM DEPARTMENT
Information Technology Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2016

KATA PENGANTAR

Puji syukur yang sebesar-besarnya Penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena penyertaan dan kasih-Nya Penulis dapat menyelesaikan Tugas Akhir yang merupakan salah satu syarat kelulusan pada Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Atas berbagai bantuan, Penulis ingin menghaturkan rasa terimakasih yang sebesar-besarnya kepada :

- Keluarga yang selalu memberikan dukungan dan semangat selama saya menjalani kuliah di Sistem Informasi ITS.
- Bapak Aris Tjahyanto, Bapak Febrilyan Samopa, dan Bapak Achmad Holil selaku Ketua dan Mantan Ketua Jurusan Sistem Informasi ITS. Terima kasih atas pengabdianya sehingga Jurusan Sistem Informasi ITS menjadi lebih baik dari waktu ke waktu.
- Ibu Nur Aini Rakhmawati selaku pembimbing I dalam proses pengerjaan Tugas Akhir ini yang selalu dengan sabar membimbing saya. Terima Kasih atas segala pengalaman, pengetahuan, dan waktu yang telah diberikan selama ini.
- Bapak Radityo Prasetyanto Wibowo, Nisfu Asrul Sani dan Bapak Arif Wibisono selaku penguji yang bersedia menguji Tugas Akhir saya, baik itu pada saat sidang progress maupun sidang akhir. Terima kasih atas segala kritik dan saran yang membangun sehingga Tugas Akhir ini dapat diselesaikan dengan baik.
- Ibu Wiwik A. selaku dosen wali selama 3,5 tahun menjalani kuliah di jurusan Sistem Informasi ITS.
- Mas Bambang Wijanarko yang telah meluangkan waktu untuk sharing dan mengatur waktu sidang di laboratorium E-Bisnis.

- Teman-teman laboratorium e-Bisnis yang selalu bersedia membantu dan berdiskusi untuk menyelesaikan permasalahan dalam pengerjaan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini mungkin masih memiliki kekurangan. Penulis terbuka terhadap kritik dan saran. Semoga Tugas Akhir ini dapat berguna bagi masyarakat secara luas.

Surabaya, Januari 2016

Penulis

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI *CROWDSOURCING* PEMETAAN KERUSAKAN JALAN DENGAN MENGGUNAKAN *GOOGLE MAPS* *API* DAN *SMS GATEWAY* (STUDI KASUS : BINA MARGA PROVINSI JAWA TIMUR)

TUGAS AKHIR

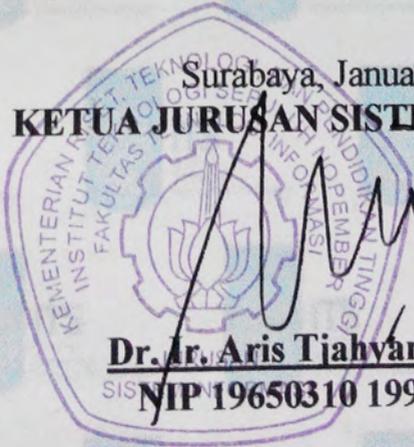
Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

YOHANDA
5212 100 113

Surabaya, Januari 2016

KETUA JURUSAN SISTEM INFORMASI



Dr. Ir. Aris Tjahyanto, M.Kom
NIP 19650310 199102 1 001

**RANCANG BANGUN APLIKASI *CROWDSOURCING*
PEMETAAN KERUSAKAN JALAN DENGAN
MENGUNAKAN *GOOGLE MAPS API* DAN *SMS
GATEWAY*
(STUDI KASUS : BINA MARGA PROVINSI JAWA
TIMUR)**

Nama Mahasiswa : Yohanda
NRP : 5212 100 113
Jurusan : Sistem Informasi FTIF-ITS
Dosen Pembimbing I : Nur Aini R., S.Kom., M.Sc. Eng.

ABSTRAK

Jalan merupakan aset yang penting bagi Indonesia yang harus dikelola dengan baik. Jalan berperan sebagai penunjang kegiatan ekonomi sehingga kondisi jalan harus terus diperhatikan. Jalan di Indonesia dikelola oleh Bina Marga. Bina Marga mempunyai tugas untuk melaksanakan perbaikan jalan. Mudah-mudahan kerusakan jalan terjadi membuat Bina Marga harus dengan tanggap menangani pelaporan kerusakan yang dilakukan oleh masyarakat. Belum adanya sebuah aplikasi yang terintegrasi dan memudahkan dalam pelaporan dan pengelolaan pelaporan kerusakan jalan menjadi salah satu hambatan Bina Marga dalam menangani kerusakan jalan Di Indonesia.

Adapun tujuan dari Tugas Akhir ini adalah merancang dan membangun aplikasi pengelola data kerusakan jalan yang dikelola oleh Bina Marga Provinsi Jawa Timur. Aplikasi ini berbasis Web dengan menggunakan Google Maps API dan SMS Gateway sebagai alat bantu untuk proses pelaporan kerusakan jalan. Dengan aplikasi ini masyarakat dapat melaporkan melalui dua cara. Yang pertama adalah dengan menandai sebuah titik kerusakan jalan pada sebuah peta digital Google. Dan yang kedua adalah dengan mengirim SMS dengan

pola pelaporan tertentu sehingga aplikasi dapat membacanya dengan mudah.

Melalui pengujian yang dilakukan pada aplikasi, didapatkan beberapa kesimpulan. Pertama, semua fitur aplikasi sudah berjalan dengan baik dan tampilan aplikasi sudah bisa menyesuaikan diri dengan perangkat bergerak. Kedua, pengguna sudah dapat merasakan manfaat dan kemudahan dalam menggunakan aplikasi. Ketiga, kekuatan server pada lingkungan implementasi bisa menangani sampai 50 pengguna yang mengakses aplikasi.

Kata Kunci : Crowdsourcing, pemetaan, SMS Gateway, Google Maps API , Bina Marga

**CROWDSOURCING APPLICATION FOR ROAD
DAMAGE MAPPING USING *GOOGLE MAPS* API AND
SMS GATEWAY
(CASE STUDY : BINA MARGA OF EAST JAVA)**

Student Name : Yohanda
NRP : 5212 100 113
Department : Information Systems FTIF-ITS
Supervisor I : Nur Aini R., S.Kom., M.Sc. Eng.

ABSTRACT

Road is an important asset for Indonesia, which must be managed properly. Road acts as a support for the economic activity so that the road should be in a good conditions all the time. Road in Indonesia are managed by Bina Marga. Road damage can be easily occurred everywhere. To the best of our knowledge, unavailability of an integrated applications that facilitate the report and road damage management becomes one of the obstacles for Bina Marga in dealing with roads damage in Indonesia.

The purpose of this final project is to design and build a road damage reporting application maintained by the Bina Marga of East Java. Application is a web-based application using the Google Maps API and SMS Gateway as a tool for reporting the road damage. This application could receive the report in two ways, namely a web form and SMS.

There are several conclusions obtained through the testing conducted on the application. First, all features of the application has been running well and applications display has been able to adapt to mobile devices. Second, users could feel application benefits and ease of use. Thirdly, the capability of the server can handle up to 50 users accessing the application.

Keywords: *Crowdsourcing, mapping, SMS Gateway, Google Maps API, Bina Marga*

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Relevansi	4
BAB II TINJAUAN PUSTAKA	5
2.1 Studi Sebelumnya	5
2.1.1 <i>Crowdsourcing geographic information for disaster response: a research frontier</i>	5
2.1.2 Peta Kondisi Jalan Bina Marga dan Smart city	6
2.2. Dasar Teori	8
2.2.1 Direktorat Jendral Bina Marga	8
2.2.2 <i>Google Maps API</i>	10
2.2.3 <i>Crowdsourcing</i>	12
2.2.4 Regular Expression	13
2.2.5 SMS Gateway	14
2.2.6 <i>System Performance Testing</i>	15
BAB III METODOLOGI	17
3.1 Urutan Pelaksanaan	17
3.1.1 Verifikasi Kebutuhan Aplikasi	18
3.1.2 Desain Sistem	18
3.1.3 Pengembangan Aplikasi	19
3.1.4 Pengujian Program	20
BAB IV PERANCANGAN	23

4.1 Analisis Kebutuhan	23
4.2 Desain Sistem.....	25
4.3 Desain <i>Database</i>	32
4.4 Desain <i>Interface</i>	38
BAB V IMPLEMENTASI	41
5.1 Lingkungan Implementasi.....	41
5.2 Direktori Aplikasi.....	42
5.3 Konfigurasi Aplikasi	43
5.4 Pembuatan Aplikasi	44
5.4.1 Fungsi <i>Login</i>	44
5.4.2 Fungsi Register	46
5.4.3 Fungsi Peta Google Maps	49
5.4.4 Fungsi Buat Laporan Baru.....	51
5.4.5 Fungsi <i>SMS Gateway</i>	55
5.4.6 Fungsi Pemberian Reputasi	62
5.4.7 Fungsi Grafik <i>Data</i>	63
5.4.8 Fungsi Upload Gambar	66
5.5 Pengujian Aplikasi	67
5.5.1 <i>System Testing</i>	67
5.5.2 <i>System Performance Testing</i>	67
5.5.3 <i>User Acceptance Testing</i>	68
5.5.4 <i>Responsive Layout Testing</i>	71
BAB VI HASIL DAN PEMBAHASAN.....	73
6.1 Hasil 73	
6.1.1 <i>System Testing</i>	73
6.1.2 <i>System Performance Test</i>	96
6.1.3 <i>User Acceptance Testing</i>	107
6.1.4 <i>Responsive Layout Testing</i>	112
6.2 Pembahasan.....	116
6.2.1 Pembahasan <i>System Testing</i>	116
6.2.2 Pembahasan <i>System Performance Test</i>	117
6.2.3 Pembahasan <i>User Acceptance Testing</i>	118
6.2.4 Pembahasan <i>Responsive Layout Test</i>	119
BAB VII KESIMPULAN DAN SARAN	121
7.1 Kesimpulan	121
7.2 Saran	121
DAFTAR PUSTAKA.....	123

Lampiran A Kuesioner User Acceptance Testing	125
Lampiran B Rekapitulasi Jawaban Responden terhadap Kemudahan Fitur (Uji Skenario).....	131
Lampiran C Rekapitulasi Jawaban Responden terhadap Kuesioner TAM	133
Lampiran D Saran Responden Mengenai Fitur Aplikasi.....	135
Lampiran E Dokumentasi <i>User Acceptance Testing</i>	137
BIODATA PENULIS	139

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 1.1 Road Map Lab E-Bisnis	4
Gambar 2.1 Tampilan Peta Kondisi Jalan Bina Marga	6
Gambar 2.2 Tampilan Smart City versi Web	7
Gambar 2.3 Tampilan Qlue	7
Gambar 2.4 Informasi kerusakan pada <i>marker</i>	11
Gambar 2.5 Pengelompokan laporan kerusakan jalan dengan <i>Marker Clusterer</i>	12
Gambar 2.6 Alur Kerja <i>SMS Gateway</i> [9]	14
Gambar 3.1 Alur pengerjaan Tugas Akhir	17
Gambar 3.2 Arsitektur Sistem	19
Gambar 3.3 Desain <i>dashboard</i>	20
Gambar 4.1 Alur pelaporan kerusakan jalan (<i>existing</i>)	24
Gambar 4.2 <i>Use Case Diagram</i>	26
Gambar 4.3 Arsitektur Sistem	27
Gambar 4.4 Pelaporan jalan rusak melalui komputer	28
Gambar 4.5 Alur proses bisnis (<i>to-be</i>)	31
Gambar 4.6 <i>Conceptual Data Model</i> aplikasi	32
Gambar 4.7 <i>Physical Data Model</i> aplikasi	35
Gambar 4.8 PDM yang telah diperbaharui	36
Gambar 4.9 Desain Halaman Beranda	38
Gambar 4.10 Informasi kerusakan jalan pada <i>marker</i>	39
Gambar 4.11 Desain Halaman Laporan Baru	39
Gambar 4.12 Desain Halaman <i>Dashboard</i>	40
Gambar 5.1 Direktori <i>Front-end</i> Aplikasi	42
Gambar 5.2 Direktori <i>Back-end</i> Aplikasi	43
Gambar 5.3 Potongan kode <i>conn.php</i>	44
Gambar 5.4 <i>Form login</i>	44
Gambar 5.5 Potongan kode validasi <i>captcha</i>	45
Gambar 5.6 Potongan kode pemanggilan <i>AJAX</i>	46
Gambar 5.7 <i>Form Register</i>	47
Gambar 5.8 Potongan kode input <i>form registrasi</i>	48
Gambar 5.9 Potongan kode pengiriman <i>email</i>	48
Gambar 5.10 Tampilan laporan pada peta	49
Gambar 5.11 Potongan kode <i>container</i> peta	49
Gambar 5.12 Potongan kode pengambil data laporan	50

Gambar 5.13 Struktur <i>array locations</i>	50
Gambar 5.14 Potongan kode <i>marker</i> dimasukkan pada peta ..	51
Gambar 5.15 Potongan kode fungsi <i>placeMarker()</i>	52
Gambar 5.16 <i>Event map click</i>	53
Gambar 5.17 <i>Event marker drag</i>	53
Gambar 5.18 Potongan kode fungsi <i>geocodeAddress()</i>	54
Gambar 5.19 Direktori Gammu	55
Gambar 5.20 Daftar tabel aplikasi	56
Gambar 5.21 Nomor <i>port modem</i>	57
Gambar 5.22 Pengaturan <i>file gammurc</i>	57
Gambar 5.23 Pengaturan <i>file smsdrc</i>	57
Gambar 5.24 Hasil perintah Gammu <i>identify</i>	58
Gambar 5.25 <i>Install</i> dan menjalankan <i>service Gammu</i>	58
Gambar 5.26 Potongan kode <i>inbox SMS Gateway</i>	59
Gambar 5.27 Tampilan <i>inbox SMS Gateway</i>	60
Gambar 5.28 <i>Flowchart</i> proses ekspor SMS	61
Gambar 5.29 Potongan kode pemberian reputasi <i>user</i>	63
Gambar 5.30 Potongan kode pembuatan tabel grafik	64
Gambar 5.31 Fungsi Highchart untuk menampilkan grafik ...	65
Gambar 5.32 Grafik Kondisi Jalan	65
Gambar 5.33 Potongan kode program <i>upload gambar</i>	67
Gambar 6.1 <i>Form register</i>	74
Gambar 6.2 Pesan <i>error</i> pada <i>form register</i>	75
Gambar 6.3 Error pada isian <i>username</i> dan <i>email</i>	76
Gambar 6.4 Error pada isian <i>password</i>	77
Gambar 6.5 Pesan berhasil <i>register</i>	77
Gambar 6.6 Email verifikasi akun	78
Gambar 6.7 Notifikasi akun berhasil diverifikasi	78
Gambar 6.8 Form Login	79
Gambar 6.9 Pesan error kombinasi <i>username</i> dan <i>password</i> tidak sesuai	79
Gambar 6.10 <i>Username</i> pada <i>menubar</i>	80
Gambar 6.11 <i>Sidebar user</i> dengan hak akses normal	80
Gambar 6.12 Halaman error menolak akses halaman	81
Gambar 6.13 Halaman Detil Laporan diakses oleh <i>user biasa</i>	81
Gambar 6.14 Tampilan Halaman Beranda	82

Gambar 6.15 <i>Marker Cluster</i> yang dinamis.....	82
Gambar 6.16 Informasi laporan pada peta	83
Gambar 6.17 Filter laporan pada peta	84
Gambar 6.18 Halaman Laporan Baru	84
Gambar 6.19 Halaman Detil Laporan diakses oleh admin.....	85
Gambar 6.20 Pengiriman SMS melalui telepon genggam	86
Gambar 6.21 SMS yang baru dikirim masuk ke dalam Halaman Inbox	86
Gambar 6.22 Memilih SMS yang ingin diekspor.....	87
Gambar 6.23 Informasi SMS yang tidak dapat diekspor	87
Gambar 6.24 Notifikasi semua SMS berhasil diekspor	88
Gambar 6.25 Detil Laporan SMS yang diekspor	88
Gambar 6.26 Reputasi laporan	89
Gambar 6.27 Pesan berhasil menambahkan reputasi laporan	89
Gambar 6.28 Pesan gagal menambahkan reputasi laporan	90
Gambar 6.29 Pesan berhasil mengubah reputasi laporan.....	90
Gambar 6.30 Menambah reputasi <i>user</i>	91
Gambar 6.31 Grafik pada Halaman <i>Dashboard</i>	91
Gambar 6.32 Halaman Daftar Laporan	92
Gambar 6.33 Filter daftar laporan.....	93
Gambar 6.34 Halaman Daftar Laporan yang difilter	93
Gambar 6.35 Halaman Edit Laporan.....	94
Gambar 6.36 Informasi laporan telah berhasil diubah	95
Gambar 6.37 Halaman <i>Upload</i> Gambar.....	95
Gambar 6.38 Gambar berhasil ditambahkan	96
Gambar 6.39 Contoh <i>Test Plan</i>	97
Gambar 6.40 Pengaturan <i>Cookie Manager</i>	97
Gambar 6.41 <i>Add-on Cookie Manager</i> pada Google Chrome	98
Gambar 6.42 HTTP Request Halaman Lihat Peta	99
Gambar 6.43 HTTP Request Halaman <i>Dashboard</i>	99
Gambar 6.44 HTTP Request Halaman <i>Search</i> Laporan	100
Gambar 6.45 Pengaturan <i>Thread Group</i> Skenario 50 <i>user</i> ..	101
Gambar 6.46 <i>Listener View Request in Table</i> skenario 50 <i>user</i>	101
Gambar 6.47 <i>Response Time Graph</i> skenario 50 <i>user</i>	102
Gambar 6.48 Pengaturan <i>Thread Group</i> skenario 100 <i>user</i> .	103

Gambar 6.49 <i>Listener View Results in Table</i> skenario 100 user	103
Gambar 6.50 <i>Response Time Graph</i> skenario 100 user	105
Gambar 6.51 Pengaturan Thread Group skenario 200 user..	105
Gambar 6.52 <i>Listener View Results in Table</i> skenario 200 user	106
Gambar 6.53 <i>Response Time Graph</i> skenario 200 user	107
Gambar 6.54 Profil Jenis kelamin responden	108
Gambar 6.55 Profil jurusan responden	108
Gambar 6.56 <i>Responsive Testing</i> pada Halaman Beranda ...	113
Gambar 6.57 <i>Responsive Testing filter</i> pada Halaman Beranda	114
Gambar 6.58 <i>Responsive Testing</i> pada Halaman <i>Dashboard</i>	115
Gambar 6.59 <i>Responsive Testing</i> pada Halaman Laporan Baru	116

DAFTAR TABEL

Tabel 2.1 Tabel perbandingan antar aplikasi pelaporan kerusakan jalan.....	8
Tabel 2.2 Kategori Tingkat Kerusakan Jalan [5]	10
Tabel 2.3 Pola Regex Umum	13
Tabel 2.4 Batas <i>response time</i>	16
Tabel 4.1 Penjelasan entitas pada CDM.....	33
Tabel 4.2 Relasi antar entitas CDM	33
Tabel 4.3 Penjelasan atribut PDM yang ditambah	36
Tabel 5.1 Spesifikasi Komputer (<i>Server</i>).....	41
Tabel 5.2 Teknologi yang Digunakan	41
Tabel 5.3 Spesifikasi modem	42
Tabel 5.4 Pemetaan <i>Test Scenario</i> pada Fitur Aplikasi.....	69
Tabel 5.5 Fitur tambahan Uji Skenario untuk <i>administrator</i> .	70
Tabel 6.1 Fitur yang diuji.....	73
Tabel 6.2 Tabel <i>Aggregate Graph</i> skenario 50 user	102
Tabel 6.3 Tabel <i>Aggregation Graph</i> skenario 100 user.....	104
Tabel 6.4 <i>Listener Aggregation Graph</i> skenario 200 user...	106
Tabel 6.5 Rata-rata nilai <i>Test Scenario</i>	109
Tabel 6.6 Pengujian Realibilitas dan Validitas pada pelapor	110
Tabel 6.7 Rata-rata bagian kedua kuisioner pelapor	110
Tabel 6.8 Rekapitulasi jawaban Uji Skenario Bpk. Atok.....	111
Tabel 6.9 Rekapitulasi kuisioner TAM Bpk. Atok	112
Tabel 6.10 Hasil <i>System Performance Test</i>	117

Halaman ini sengaja dikosongkan

DAFTAR LAMPIRAN

Lampiran A Kuesioner User Acceptance Testing	A - 1 -
Lampiran B Rekapitulasi Jawaban Responden terhadap Kemudahan Fitur (Uji Skenario)	B - 1 -
Lampiran C Rekapitulasi Jawaban Responden terhadap Kuesioner TAM.....	C - 1 -
Lampiran D Saran Responden Mengenai Fitur Aplikasi. D - 1 -	
Lampiran E Dokumentasi User Acceptance Testing.....	E - 1 -

BAB I

PENDAHULUAN

Pada bab ini, akan dijelaskan tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan Tugas Akhir, dan relevansi atau manfaat kegiatan Tugas Akhir.

1.1 Latar Belakang Masalah

Jalan merupakan aset yang penting bagi Indonesia yang harus dikelola dengan baik. Jalan berperan sebagai penunjang kegiatan ekonomi sehingga kondisi jalan harus terus diperhatikan. Dengan jalan yang baik maka proses distribusi akan secara luas dapat dilakukan ke daerah-daerah yang sulit diakses pada awalnya. Selain itu, jalan yang baik akan mempercepat proses distribusi sehingga dapat menimbulkan keuntungan seperti efisiensi biaya, peningkatan kapasitas produksi, dan peningkatan kepuasan pelanggan. Terakhir, jalan yang baik akan menghindarkan para pengendara dari kecelakaan.

Jalan dikelola oleh Bina Marga. Bina Marga merupakan lembaga pemerintah yang berada di bawah kementerian Kementerian Pekerjaan Umum Dan Perumahan Rakyat. Bina Marga melaksanakan pengelolaan prasarana dan sarana bidang pekerjaan umum seperti jalan, tata air, dan jembatan. Pengelolaan tersebut termasuk pengecekan kondisi jalan, pembuatan jalan baru, dan perbaikan jalan.

Jalan dapat dengan mudah rusak oleh banyak faktor. Dengan demikian, banyak kerusakan jalan yang ada di Indonesia. DKI Jakarta saja terdapat 2.998 titik kerusakan jalan yang diterima oleh Bina Marga sampai Mei 2015. Informasi kerusakan ini didapat melalui laporan dari masyarakat yang dikirim melalui Twitter, *email*, dan call center pengaduan. [1]

Jumlah laporan kerusakan yang banyak diterima tersebut harus dikelola dengan baik sehingga laporan masyarakat dapat ditangani dengan optimal. Pengelolaan informasi mengenai kerusakan Jalan dihadapkan oleh beberapa kendala. Pertama, proses pelaporan yang masih sangat terbatas karna belum terdapat banyak saluran informasi dari masyarakat kepada masyarakat. Kedua, pengecekan status dari kerusakan jalan yang belum optimal. Setelah kerusakan dilaporkan, maka Bina Marga harus segera memperbaiki kerusakan jalan tersebut. Masih belum adanya alat pembantu pengecekan apakah laporan kerusakan jalan itu berada dalam status berjalan, belum ditangani, maupun sudah selesai menimbulkan adanya jalan yang tidak jelas penanganannya.

Oleh karena itu, pada Tugas Akhir ini dibuat sebuah aplikasi yang bernama Bina Marga RRS (*Road Reporting System*). Bina Marga RRS membantu menghubungkan masyarakat dengan Bina Marga Provinsi Jawa Timur. Masyarakat dapat melaporkan kerusakan jalan, sehingga proses identifikasi kerusakan jalan dan penanganannya dapat dilakukan lebih cepat.

1.2 Perumusan Masalah

Berdasarkan penjelasan latar belakang di atas, rumusan masalah yang menjadi fokus utama dalam Tugas Akhir ini adalah :

1. Bagaimana merancang dan membangun aplikasi Pengelola Informasi Kerusakan Jalan?
2. Bagaimana memanfaatkan *Google Maps API* dan *SMS Gateway* sebagai media *Crowdsourcing* untuk melaporkan kerusakan jalan?

1.3 Batasan Masalah

Dari permasalahan yang disebutkan di atas, batasan masalah dalam Tugas Akhir ini adalah:

1. Aplikasi digunakan untuk Provinsi Jawa Timur.
2. Aplikasi dibuat berbasis *web* dengan menggunakan alat pengembangan *PHP, MySQL, Google Maps API*.
3. Aplikasi hanya mengelola informasi jalan saja tidak termasuk perlengkapan jalan ataupun bangunan pelengkap jalan, seperti lampu jalan, terowongan, rambu lalu lintas, dan lain sebagainya.
4. Aplikasi tidak membedakan Jalan Kota, Provinsi, dan sebagainya.
5. Tidak ada pengujian kapasitas *SMS Gateway*.

1.4 Tujuan Penelitian

Tujuan dari Tugas Akhir ini adalah merancang dan membangun aplikasi pengelola *data* kerusakan jalan yang dikelola oleh Bina Marga Provinsi Jawa Timur. Aplikasi ini berbasis *web* dengan menggunakan *Google Maps API* dan *SMS Gateway* sebagai alat bantu untuk proses pelaporan kerusakan jalan.

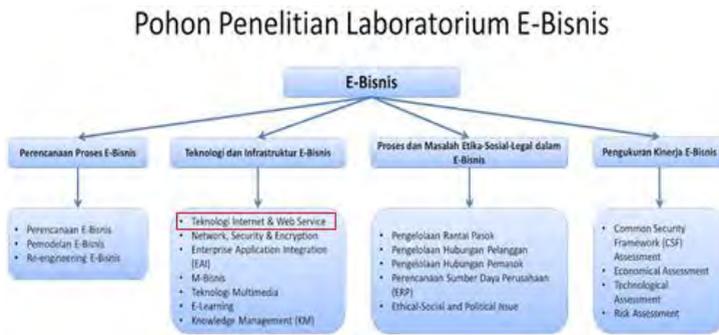
1.5 Manfaat Penelitian

Melalui Tugas Akhir ini diharapkan dapat memberi manfaat, yaitu:

1. Membantu Bina Marga Provinsi Jawa Timur untuk memantau kondisi jalan dan proyek perbaikan jalan.
2. Membantu masyarakat dalam melaporkan kerusakan jalan.

1.6 Relevansi

Topik pada Tugas Akhir ini adalah mengenai rancang bangun Aplikasi Pemetaan Kerusakan Jalan dengan *SMS Gateway* dan *Google Maps API*, dimana penelitian ini masih terkait pada pohon penelitian yang terletak di bagian Teknologi dan Infrastruktur E-Bisnis yaitu pada bidang Teknologi Internet dan *Web Service* seperti yang ditampilkan pada Gambar 1.1.



Gambar 1.1 Road Map Lab E-Bisnis

BAB II

TINJAUAN PUSTAKA

Untuk memudahkan pemahaman tentang apa yang akan dilakukan pada Tugas Akhir ini, berikut ini akan di paparkan tentang konsep dan teknologi apa saja yang akan digunakan atau diterapkan untuk membangun aplikasi pada Tugas Akhir ini. Berikut konsep teknologi yang digunakan pada Tugas Akhir ini.

2.1 Studi Sebelumnya

Berikut ini adalah daftar penelitian terdahulu yang terkait dengan penelitian kali ini.

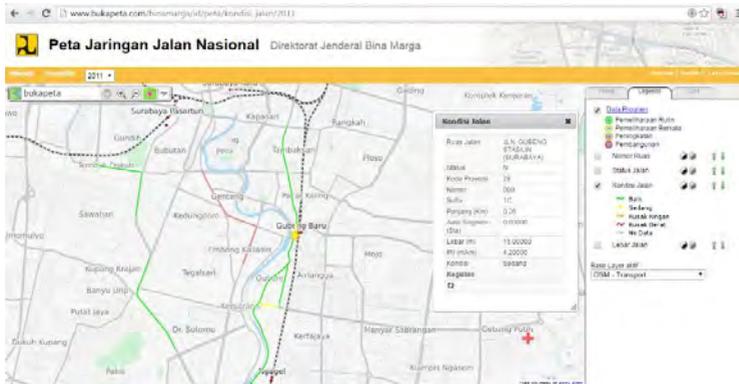
2.1.1 *Crowdsourcing geographic information for disaster response: a research frontier*

Crowdsourcing digunakan untuk pengumpulan informasi bencana. Masyarakat yang ada di daerah bencana dapat menyebarkan informasi melalui aplikasi sehingga semua orang yang berada disekitarnya dapat waspada dan melakukan tindakan yang tepat untuk menghadapi bencana tersebut.

Informasi yang didapatkan melalui *Crowdsourcing* memiliki kelemahan dan kelebihan. Kelemahannya adalah informasi *Crowdsourcing* sulit untuk dipastikan kebenarannya. Sedangkan, kelebihannya adalah informasi *Crowdsourcing* dapat lebih cepat sampai dan dalam jumlah yang banyak. [2]

2.1.2 Peta Kondisi Jalan Bina Marga dan Smart city

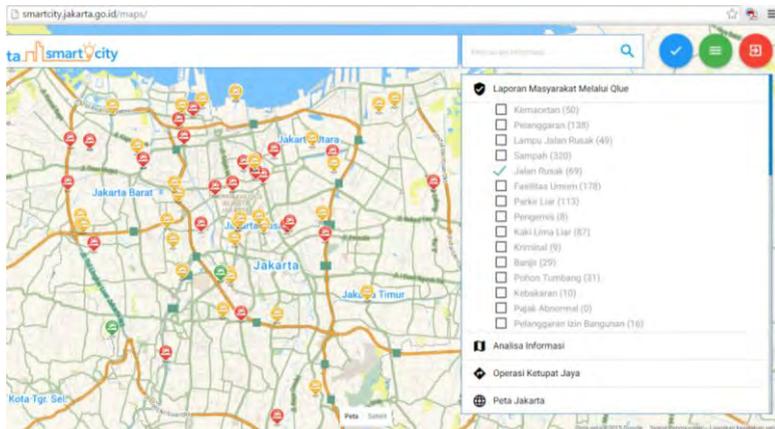
Peta Kondisi Jalan Bina Marga dapat diakses melalui URL http://www.bukapeta.com/binamarga/id/peta/kondisi_jalan/. Aplikasi ini dapat menunjukkan kondisi jalanan di seluruh Indonesia mulai dari tahun 2011 sampai 2012. Berikut adalah tampilan aplikasinya pada Gambar 2.1.



Gambar 2.1 Tampilan Peta Kondisi Jalan Bina Marga

Pengguna dapat mengetahui kondisi jalan melalui warna yang berada pada jalan, yaitu hijau untuk jalanan dengan kondisi baik, kuning untuk jalanan dengan kondisi sedang, dan merah untuk jalan dengan kondisi rusak.

Selain aplikasi pemetaan Bina Marga sebelumnya dibahas, terdapat juga aplikasi *Smart City* (Qlue) yang dimiliki oleh DKI Jakarta. Aplikasi ini mempunyai banyak kegunaan, yaitu sebagai sarana pelaporan masyarakat, forum, dan ulasan tempat. Masyarakat dapat melaporkan berbagai macam hal termasuk kerusakan jalan. Laporan tersebut akan muncul sebagai titik-titik *marker* di *map*. Berikut adalah tampilan *Smart City* pada Gambar 2.2.



Gambar 2.2 Tampilan Smart City versi Web

Aplikasi *Smart City* ini bisa diakses melalui Android dengan *download* aplikasi bernama Qlue di Play Store. Berikut adalah tampilan aplikasi Qlue pada gambar 2.3.



Gambar 2.3 Tampilan Qlue

Aplikasi-aplikasi yang sebelumnya dibahas memiliki perbedaan dengan Aplikasi Bina Marga RRS pada Tugas Akhir ini. Perbedaan itu dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel perbandingan antar aplikasi pelaporan kerusakan jalan

Aplikasi	Lingkup	Sumber Informasi	Jalur Masukan	Gamification
Smart City	Jakarta	Masyarakat	Qlue	Ya
Peta Kondisi Jalan Bina Marga	Indonesia	Bina Marga	Aplikasi <i>web</i>	Tidak
Aplikasi <i>Crowdsourcing</i> Pemetaan Kerusakan Jalan	Jawa Timur	Masyarakat & Bina Marga Provinsi Jawa Timur	Aplikasi <i>web & SMS Gateway</i>	Tidak

2.2. Dasar Teori

2.2.1 Direktorat Jendral Bina Marga

Direktorat Jendral merupakan suatu organisasi yang memiliki tugas untuk menyelenggarakan perumusan dan pelaksanaan kebijakan dibidang yang telah ditentukan. Bina Marga merupakan sebuah Direktorat Jendral yang berada di bawah Kementerian Pekerjaan Umum yang mengelola marga/jalan. Menurut Pasal 395 Peraturan Menteri Pekerjaan Umum nomor 08/PRT/M/2010 [3], Direktorat Jendral Bina Marga memiliki tugas sebagai berikut :

- Perumusan kebijakan di bidang Bina Marga yang meliputi penyelenggaraan Jalan Nasional, Provinsi, Kabupaten, Kota dan Desa.
- Pelaksanaan kebijakan dibidang Bina Marga meliputi pengaturan, pembinaan, pembangunan dan pengawasan Jalan Nasional, Provinsi, Kabupaten, Kota dan Desa, serta penanggulangan darurat dan rehabilitasi kerusakan jalan akibat bencana alam.

- Penyusunan norma, standar, prosedur dan kriteria dibidang Bina Marga.
- Pemberian bimbingan teknis dan evaluasi dibidang Bina Marga meliputi pengaturan, pembinaan, pembangunan dan pengawasan Jalan Nasional, Provinsi, Kabupaten, Kota dan Desa.

Terdapat beberapa definisi mengenai jalan menurut Peraturan Menteri Pekerjaan Umum nomor 04/PRT/M/2012 [4], yaitu :

- Jalan adalah prasarana transportasi darat yang meliputi segala bagian jalan, termasuk bangunan pelengkap dan perlengkapannya yang diperuntukan bagi lalu-lintas, yang berada pada permukaan tanah, di atas permukaan tanah, dibawah permukaan tanah dan/atau air, serta di atas permukaan air, kecuali kereta api, jalan lori, dan jalan kabel.
- Bangunan pelengkap adalah bangunan untuk mendukung fungsi dan keamanan konstruksi jalan yang meliputi jembatan, terowongan, ponton, lintas atas (*flyover, elevated road*), lintas bawah (*underpass*), tempat parkir, gorong-gorong, tembok penahan, dan saluran tepi jalan dibangun sesuai dengan persyaratan teknis.
- Perlengkapan Jalan adalah sarana yang dimaksudkan untuk keselamatan, keamanan, ketertiban, dan kelancaran lalu-lintas serta kemudahan bagi pengguna jalan dalam berlalu-lintas yang meliputi marka jalan, rambu lalu lintas, alat pemberi isyarat lalu-lintas, lampu penerangan jalan, rel pengaman (*guardrail*), dan penghalang lalu-lintas (*traffic barrier*).

Aplikasi Bina Marga RRS pada Tugas Akhir ini tidak akan mengolah informasi yang berkaitan dengan kerusakan bangunan pelengkap ataupun perlengkapan jalan. Jalan yang rusak akan dilaporkan masyarakat dengan empat kategori tingkat kerusakan seperti yang ada dalam Peraturan Menteri

Pekerjaan Umum nomor 13 /PRT/M/2011. Tabel 2.2 menunjukkan empat kategori tingkat kerusakan jalan berserta penanganannya.

Tabel 2.2 Kategori Tingkat Kerusakan Jalan [5]

Kondisi	Prosentase Batasan Kerusakan Jalan		Program Penanganan
	Berpenutup Aspal	Tidak Berpenutup Aspal	
Baik	< 6 %	< 11 %	Pemeliharaan rutin
Sedang	6 - 11%	11 - 16%	Pemeliharaan rutin/berkala Pemeliharaan rehabilitasi
Rusak Ringan	11 - 15%	16 - 23%	
Rusak Berat	> 15%	> 23 %	Rekonstruksi/peningkatan struktur

Masyarakat tidak perlu menghitung secara detil persentase kerusakan jalan. Masyarakat hanya menimbang secara kasar seberapa parah jalan tersebut rusak. Pihak Bina Marga yang akan menentukan status terakhir kerusakan jalan yang telah dilaporkan dan akan menentukan program penanganannya.

2.2.2 Google Maps API

API adalah “Sekumpulan fungsi, perintah dan protokol yang dapat digunakan untuk menghubungkan satu aplikasi dengan aplikasi yang lain agar dapat berinteraksi”. [6] *API* bekerja sebagai jembatan antara satu *software* dengan *software* lain.

Google Maps API adalah sebuah *API* gratis yang berguna untuk menampilkan informasi-informasi geografis dalam bentuk *map*. *Google Maps API* ini akan membantu untuk menerjemahkan informasi geografis ke dalam titik *data* yang dapat terlihat melalui *Google Maps*.

Dalam pembuatan program *Google Maps API* menggunakan urutan sebagai berikut :

1. Memasukkan *Maps API Javascript* ke dalam *HTML*.
2. Membuat *element div* dengan nama tertentu untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan properti-properti pada peta.
4. Menuliskan fungsi *Javascript* untuk membuat objek peta.
5. Melakukan inisiasi peta dalam *tag body HTML* dengan *event onload*.

Terdapat sebuah elemen dari *Google Maps* yang banyak dipakai, yaitu *marker*. *Marker* adalah sebuah penanda pada *Google Maps* yang berbentuk seperti jarum. *Marker* ini digunakan untuk menandakan titik jalan yang dilaporkan rusak. *Marker* muncul ketika pelapor melakukan klik pada salah satu titik pada peta ataupun dengan memasukkan nama jalan. Ketika *marker* diklik maka akan muncul informasi tentang titik kerusakan seperti contoh pada Gambar 2.4.



Gambar 2.4 Informasi kerusakan pada *marker*

Marker yang terdapat pada peta akan bertambah banyak seiring dengan banyaknya laporan kerusakan jalan yang masuk sehingga *marker-marker* tersebut dapat terlihat saling bertumpukan. *Google Maps API* menyediakan sebuah fungsi untuk mengelompokkan *marker-marker* dengan memakai *Marker Clusterer*. *Marker* dapat dikelompokkan berdasarkan kedekatan jarak antar *marker*. Gambar 2.5 menunjukkan fungsi dari *Marker Clusterer*.



Gambar 2.5 Pengelompokan laporan kerusakan jalan dengan *Marker Clusterer*

Kumpulan *marker* memiliki bentuk yang berbeda dengan *marker* biasa. Ketika pengguna melakukan klik pada kumpulan *marker* ini maka *marker-marker* yang merupakan bagian dari kelompok *marker* itu akan muncul sebagai *marker* biasa.

2.2.3 Crowdsourcing

Crowdsourcing adalah sebuah tindakan yang membuat sebuah tugas yang biasanya dilakukan oleh seseorang yang berwenang menjadi sebuah tugas yang dapat dilakukan oleh sekumpulan orang yang belum diketahui. [7]

Crowdsourcing dalam aplikasi pelaporan ini digunakan agar informasi mengenai kerusakan jalan dapat dengan cepat diketahui oleh pihak Bina Marga. Masyarakat dapat

melaporkan kerusakan jalan melalui *website* ataupun melalui *SMS*.

2.2.4 Regular Expression

Regular Expression adalah pola yang menggambarkan sejumlah teks. Pola dalam *regex* ini akan digunakan untuk menangkap informasi penting dalam sebuah *SMS*. Jadi, nantinya kebenaran pola dalam *regex* ini yang memisahkan informasi yang dikirimkan melalui *SMS*. Misalnya, masukan kondisi jalan, alamat, deskripsi, dan nama pelapor di dalam sebuah *SMS*. Berikut adalah beberapa pola pengenalan dalam *regex* pada tabel 2.3. [8]

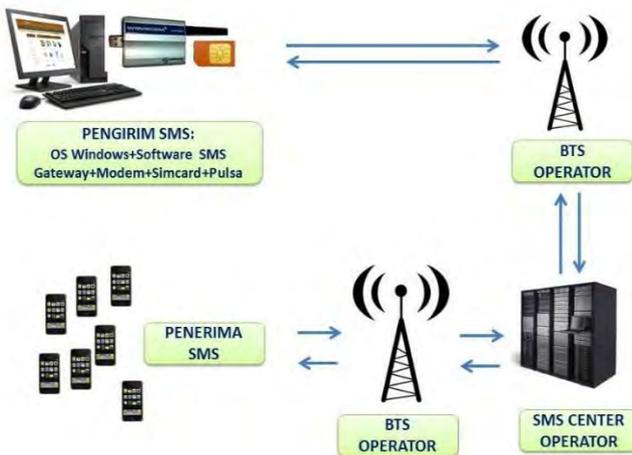
Tabel 2.3 Pola Regex Umum

<i>Regex</i>	Deskripsi
.	Seluruh karakter
^ <i>regex</i>	Kata “ <i>regex</i> ” harus berada di awal <i>string</i>
<i>regex</i> \$	Kata “ <i>regex</i> ” harus berada di akhir <i>string</i>
[abc]	Definisi himpunan karakter yang diperbolehkan, a, b atau c
[abc[vz]]	Definisi himpunan karakter a, b, atau c yang diikuti oleh v atau z
[^abc]	Bila tanda “^” berada dalam “[]”, adalah negasi dari himpunan karakter, yaitu karakter selain a, b, dan c
[a-d1-7]	Tanda “-“ menunjukkan urutan karakter, yaitu a sampai d atau 1 sampai 7
X Z	Tanda “ ” berarti “atau”, artinya X atau Z

XZ	Bila berurutan biasa tanpa tanda apa-apa, artinya X harus diikuti oleh Z
----	--

2.2.5 SMS Gateway

SMS Gateway adalah sebuah sistem yang menjadi penengah antara *internet* dengan sebuah fitur dalam telepon genggam yaitu *SMS (Short Message Service)*. *SMS Gateway* memungkinkan komputer dengan bantuan sebuah *modem* untuk mengirimkan dan menerima *SMS* dari telepon genggam biasa. *SMS gateway* memerlukan sebuah *server*. Gambar 2.6 menunjukkan alur kerja dari *SMS Gateway*. [9]



Gambar 2.6 Alur Kerja *SMS Gateway* [9]

Jadi, aplikasi yang terdapat di *server* akan mengirimkan sebuah *SMS*. *SMS* ini dikirim melalui *modem* yang didalamnya terdapat *Simcard* yang digunakan telpon genggam pada umumnya. *SMS* tersebut akan dikirimkan layaknya *SMS* biasa yang pada akhirnya akan diterima oleh penerima *SMS* tujuan.

SMS Gateway yang dipakai dalam Tugas Akhir ini adalah Gammu. *SMS* akan dipakai sebagai salah satu jalur masukan pelaporan kerusakan jalan. Pelapor dapat mengirimkan *SMS* dari telepon genggam mereka. *SMS* tersebut akan ditentukan polanya. Isi utamanya adalah nama jalan lengkap, kondisi kerusakan, nama pelapor, dan deskripsi.

SMS dipilih menjadi salah satu jalur masuknya laporan dikarenakan *SMS* merupakan jalur masukan yang dapat diakses disebagian besar tempat di Indonesia. *SMS* berbeda dengan pelaporan melalui *Google Maps API* karena *SMS* tidak membutuhkan sinyal *GPS* ataupun *internet* sehingga lebih besar jangkauannya. Selain itu, setiap orang sekarang pasti memiliki sebuah telepon genggam minimal dengan fitur untuk telepon dan *SMS*. Menurut Menkominfo terdapat 270.000.000 pengguna telepon genggam di Indonesia. [10] Jumlah tersebut merupakan jumlah yang besar sehingga setiap masyarakat di Indonesia berkesempatan untuk bisa melaporkan jalan rusak meskipun di pelosok desa sekalipun.

2.2.6 System Performance Testing

System Performance Testing adalah pengujian yang dilakukan untuk melihat bagaimana hasil kerja sebuah sistem dalam hal ketanggapan dan keandalan dalam suatu beban kerja. Dalam aplikasi berbasis *web*, beban kerja adalah banyaknya permintaan dari *user* kepada *server*. Beban kerja yang tinggi biasanya akan semakin memperburuk kinerja sistem. [11]

Terdapat beberapa kriteria analisis kualitas kinerja sebuah sistem yaitu :

- *Response time*. Waktu yang diperlukan permintaan *data* yang dikirimkan oleh *client* ke *server* sampai *data* yang diminta tersebut sampai ke *client*. Misalnya, untuk membuka halaman <http://google.com> diperlukan waktu 3 detik sampai halaman terbuka di *browser*

- *Throughput.*
Banyaknya transaksi permintaan *data* pada *server* dalam satuan waktu.
- *Resource utilization.*
Banyaknya sumber daya *hardware* yang diperlukan aplikasi untuk bekerja, seperti pemakaian prosesor, memori, dan baca/tulis *hard disk*.
- *Maximum user load.*
Berapa banyak *user* yang bisa mengakses aplikasi.

Hasil *System Performance Testing* dapat dijadikan landasan untuk mengetahui konfigurasi beserta *hardware* yang terbaik untuk mencapai suatu batas yang ditentukan. Misalnya, aplikasi dapat diakses 100 *user* pada saat yang bersamaan.

Salah satu hasil *System Performance Testing* adalah *response time*. Berdasarkan [12], *response time* dapat dikelompokkan menjadi empat batas-batas sebuah aplikasi masih mendapatkan perhatian penuh dari *user*. Batas-batas tersebut disajikan pada Tabel 2.4.

Tabel 2.4 Batas *response time*

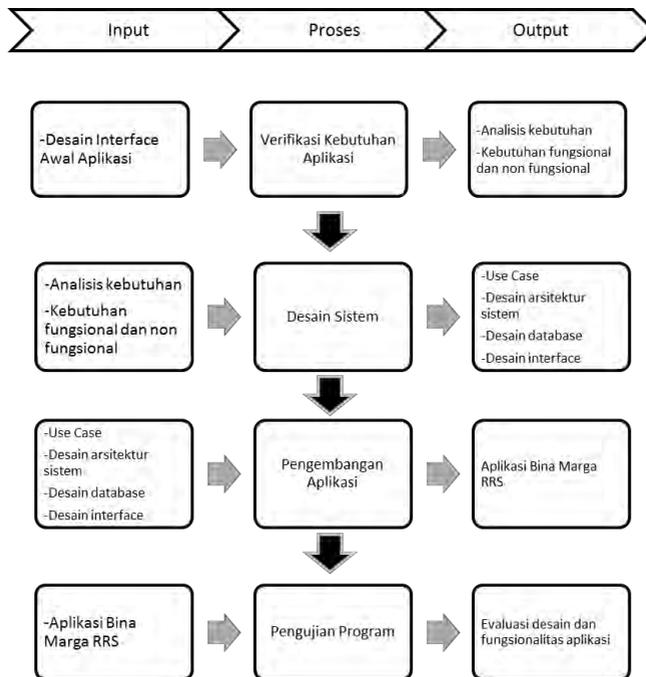
Response time	Efek
< 0,1 detik	Membuat <i>user</i> merasa sistem memberikan tanggapan yang instan dan sangat cepat
0,1 - 1 detik	Pikiran <i>user</i> tidak teralihkan dengan hal lain
1 - 10 detik	<i>User</i> masih bisa untuk fokus dengan halaman
> 10 detik	<i>User</i> akan teralihkan untuk mengerjakan tugas lain

BAB III METODOLOGI

Bab ini berisi metodologi dalam pengerjaan Tugas Akhir. Urutan pengerjaan, definisi dari setiap proses, alat dan bahan.

3.1 Urutan Pelaksanaan

Bagian ini menjelaskan metodologi yang digunakan dalam pengerjaan Tugas Akhir ini. Gambar 3.1 menunjukkan metodologi pengerjaan Tugas Akhir.



Gambar 3.1 Alur pengerjaan Tugas Akhir

Metodologi yang dipakai mengadopsi model *Software Development Lifecycle* (SDLC) *waterfall*. Model ini dipilih karena mudah diimplementasikan, *scope* aplikasi yang tidak besar, kebutuhan yang tidak banyak berubah, dan sumber daya manusia yang sedikit. [13] Tahap pada metodologi ini tidak sampai pada tahap *deployment* karena aplikasi tidak akan dipasang pada Bina Marga.

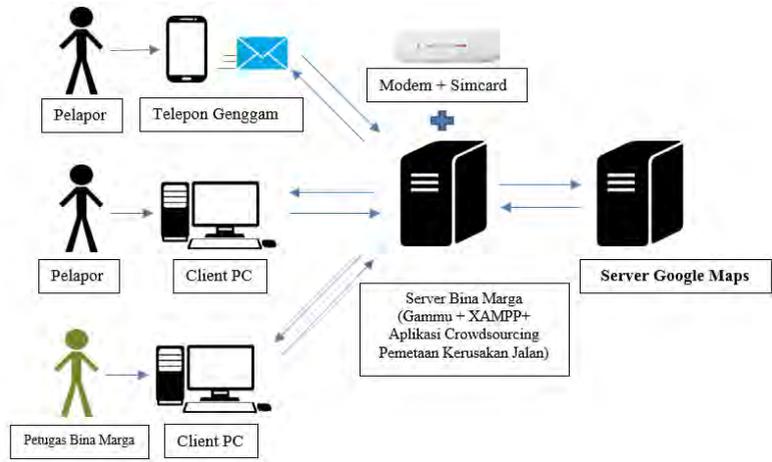
3.1.1 Verifikasi Kebutuhan Aplikasi

Verifikasi kebutuhan aplikasi dilakukan dengan melakukan survey kepada Bina Marga Provinsi Jawa Timur. Verifikasi diperlukan agar aplikasi dapat menunjang proses bisnis yang sudah ada. Verifikasi juga berguna untuk memastikan tentang kebutuhan apa saja yang diperlukan oleh Bina Marga Provinsi Jawa Timur sebagai pengelola jalan di Provinsi Jawa Timur.

Sebelum melakukan verifikasi, dibuat sebuah rancangan desain aplikasi awal. Desain tersebut didasarkan pada desain-desain aplikasi yang memiliki fungsi yang mirip dengan Bina Marga RRS, yaitu aplikasi Peta Kondisi Jalan milik Bina Marga dan *Smart City* DKI Jakarta. Desain ini dapat menjadi alat *brainstorming* kebutuhan dan fitur apa saja yang harus dimiliki oleh aplikasi.

3.1.2 Desain Sistem

Pada proses ini rancangan sistem akan dibuat dan didokumentasikan. Rancangan sistem meliputi desain sistem, desain *database*, dan desain *interface*. Gambar 3.2 menunjukkan arsitektur sistem yang didalamnya tergambar alur kerja sistem dan elemen-elemen dalam sistem secara umum.

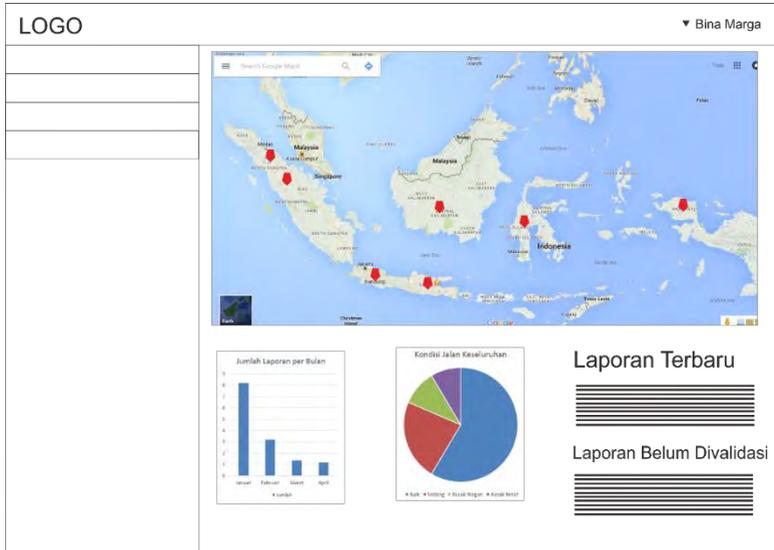


Gambar 3.2 Arsitektur Sistem

3.1.3 Pengembangan Aplikasi

Pada proses ini aplikasi mulai dikembangkan dengan memakai teknologi yang telah disebutkan sebelumnya. Teknologi yang dipakai adalah *PHP*, *MySQL*, dan *Google Maps API*. Hal yang pertama dilakukan adalah pembuatan bagian *front-end* dari aplikasi yaitu pelaporan kerusakan jalan. Pelaporan jalan melalui komputer akan dibuat terlebih dahulu, lalu pelaporan melalui *SMS Gateway*.

Selanjutnya, Aplikasi akan disesuaikan sehingga ketika dibuka pada *smartphone*, aplikasi dapat memberikan fungsionalitas dan tampilan yang lengkap bagi pengguna. Terakhir, *back-end* aplikasi untuk petugas Bina Marga akan dibuat. *Back-end* aplikasi ini berguna untuk Bina Marga dalam mengelola informasi yang telah dilaporkan. Misalnya, menghapus laporan yang tidak valid, memperbaharui status kondisi jalan, dan melihat statistik laporan kerusakan jalan. Berikut pada gambar 3.3 adalah desain *dashboard* yang menjadi bagian *back-end* aplikasi.



Gambar 3.3 Desain *dashboard*

3.1.4 Pengujian Program

Aplikasi diuji untuk mengetahui apakah aplikasi sudah bekerja dengan benar dan sesuai dengan yang diharapkan. Terdapat empat tes yang akan dilakukan, yaitu *System Testing*, *System Performance Test*, *User Acceptance Testing*, dan *Responsive Layout Testing*. Empat tes tersebut menguji komponen aplikasi yang berbeda.

System Testing menguji apakah semua fungsionalitas aplikasi sudah berjalan dengan benar. Setiap fitur-fitur aplikasi akan dicoba. Jika terdapat kesalahan program, maka akan langsung diperbaiki.

System Performance Test menguji bagaimana kinerja *server* dalam melayani setiap pengguna aplikasi. Pengujian dilakukan dengan menggunakan sebuah aplikasi bernama JMeter. Terdapat tiga skenario yang akan dicoba, yaitu jika Bina Marga

RRS diakses oleh 50, 100, dan 200 pengguna. *System Performance Test* dapat menunjukkan berapa jumlah pengguna aplikasi optimal yang dapat dilayani oleh Bina Marga RRS berdasarkan hasil yang didapat ketika skenario dijalankan.

User Acceptance Testing dilakukan untuk menguji apakah pengguna merasakan kemudahan dan manfaat ketika menggunakan aplikasi. Dua hal tersebut akan mempengaruhi keinginan pengguna untuk terus menggunakan aplikasi. Dengan demikian, aplikasi dapat digunakan secara optimal.

Aplikasi Bina Marga RRS dirancang untuk dapat menyesuaikan tampilan ketika diakses pada perangkat dengan layar kecil seperti *smartphone* dan *tablet*. Tes dilakukan dengan mencoba mengakses aplikasi melalui *Developer Tools* yang ada pada Google Chrome dan mengatur perangkat yang mengaksesnya menjadi sebuah *smartphone* dan *tablet*.

Halaman ini sengaja dikosongkan

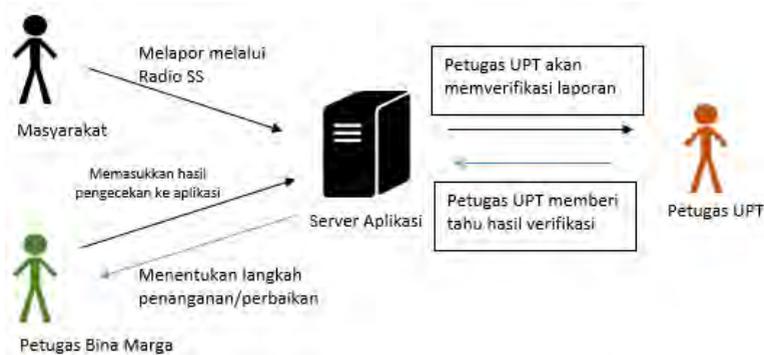
BAB IV PERANCANGAN

Pada bab ini, akan dijelaskan perancangan yang akan dilakukan untuk membangun aplikasi pada Tugas Akhir ini. Pada perancangan aplikasi Bina Marga RRS ini dilakukan beberapa langkah perancangan meliputi analisis kebutuhan, desain sistem, desain *database* dan desain *interface* aplikasi.

4.1 Analisis Kebutuhan

Kebutuhan Aplikasi didapatkan melalui wawancara dengan pihak Bina Marga Provinsi Jawa Timur. Narasumbernya adalah Bapak Ir. Atok Nardyawanto, MMT sebagai Kepala Seksi pengaturan dan pengendalian dan Ibu Krisna sebagai bagian dari bagian pengaturan dan pengendalian. Wawancara dilakukan pada Kamis, 22 Oktober 2015.

Pihak Bina Marga sendiri memiliki sebuah aplikasi untuk mengelola informasi pengelola kerusakan secara internal. Aplikasi tersebut hanya menerima masukan dari pihak Bina Marga. Selain melalui aplikasi itu, Bina Marga Jatim juga menjalin kerjasama dengan Radio Suara Surabaya (SS). Radio SS akan memberikan laporan tentang kerusakan jalan yang dilaporkan masyarakat melalui jejaring sosial milik Radio SS kepada Bina Marga. Alur pelaporan kerusakan digambarkan pada Gambar 4.1.



Gambar 4.1 Alur pelaporan kerusakan jalan (*existing*)

Laporan-laporan kerusakan jalan yang diterima Bina Marga diproses dengan melakukan pengecekan kebenaran dari laporan tersebut. Kemudian, jalan dinilai tingkat kerusakannya dan akan diberikan penanganan yang sesuai.

Bina Marga Provinsi Jawa Timur hanya mengelola jalan-jalan yang termasuk jalan provinsi. Bina Marga Jatim memiliki 12 Unit Pelaksana Teknis (UPT) yang tersebar di Jawa Timur. Jika ada laporan jalan yang masuk dan jauh jaraknya dari Bina Marga Jatim, maka UPT yang bertugas untuk mengecek kerusakan jalan tersebut. Sehingga pada aplikasi ini ditambahkan *user* UPT untuk melakukan penambahan dan pembaharuan informasi jalan. Namun, bila jalan yang dilaporkan bukan termasuk Jalan Provinsi maka Bina Marga Jatim tidak berhak untuk mengurus jalan tersebut. Wewenangnya ada berada di pemerintah Kota/Kabupaten jika jalan tersebut termasuk Jalan Kota/Kabupaten. Jadi, *user*nya bertambah lagi bukan hanya pihak Bina Marga, *user* UPT perlu ditambahkan juga.

Laporan yang masuk akan bertambah banyak seiring waktu. Bina Marga harus menentukan prioritas untuk pemilihan laporan yang akan ditangani karena tidak semua laporan dapat ditangani mengingat adanya keterbatasan sumber daya

manusia. Maka dari itu, diperlukan sebuah fitur untuk memberikan *rating* kepada *user* yang kemungkinan besar dapat dipercaya. *Rating* itu berbentuk sebuah poin yang akan ditambahkan kepada akun dengan laporan yang sudah diverifikasi. Semakin banyak poin yang dimiliki akun, semakin terpercaya kebenaran laporannya.

Berikut adalah kebutuhan fungsional aplikasi :

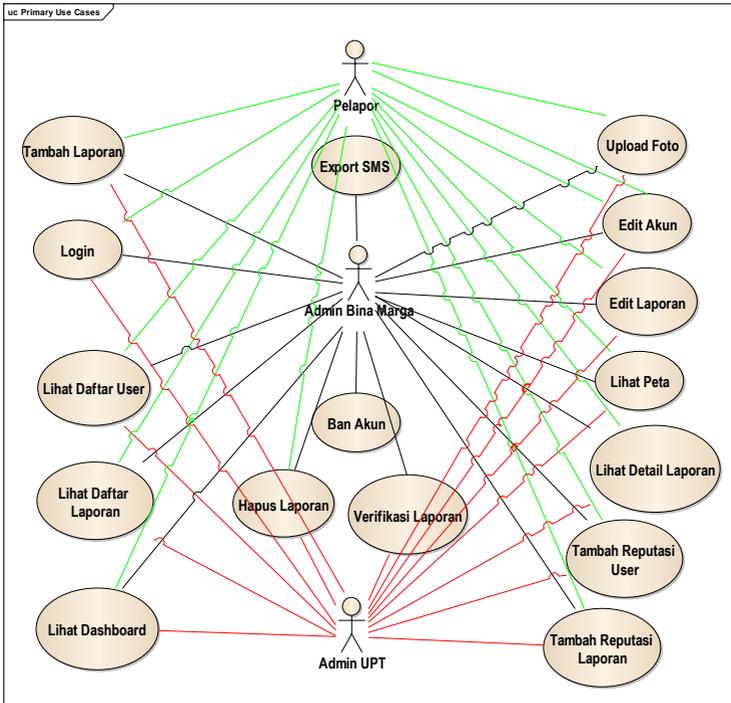
- Aplikasi dapat menangani proses *login* dan registrasi user
- Aplikasi dapat mengelola *data* kerusakan jalan, meliputi *input*, *edit*, dan *delete*
- Aplikasi dapat menampilkan laporan kerusakan jalan pada sebuah peta dan daftar
- Aplikasi dapat melakukan *filter* atribut terhadap *data-data* laporan
- Aplikasi dapat menerima laporan melalui *SMS* maupun melalui *web*
- Laporan dan *user* bisa diberikan reputasi
- Aplikasi dapat menampilkan *data* dan grafik statistik mengenai laporan kerusakan jalan

Berikut adalah kebutuhan non fungsional aplikasi :

- Laporan yang selama tiga bulan tidak diverifikasi akan berada pada status *expired* sehingga tidak dapat ditampilkan pada peta
- Layout aplikasi harus responsif
- Aplikasi dapat diakses oleh 50 *user* bersamaan

4.2 Desain Sistem

Fitur-fitur dan hak akses untuk setiap *user* pada aplikasi digambarkan melalui *Use Case Diagram*. *Use Case Diagram* bisa dilihat pada Gambar 4.2.



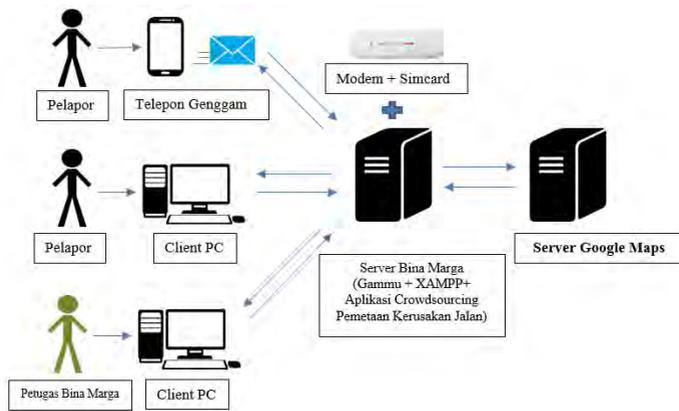
Gambar 4.2 Use Case Diagram

Terdapat perbedaan hak akses antar user. Pertama, pelapor tidak bisa mengakses beberapa fitur, seperti *ban* akun, verifikasi laporan, dan *export SMS*. Kedua, *admin* UPT hanya diberikan beberapa fitur tambahan dari pelapor, yaitu verifikasi, *edit* laporan, dan melihat *SMS Gateway*. Terakhir, *admin* Bina Marga dapat mengakses semua fitur aplikasi.

Aplikasi Bina Marga RRS menggunakan *Google Maps API* dan Gammu sebagai jalur masuknya laporan. *Google Maps API* akan menangani pelaporan melalui *web*. Pelapor dapat menentukan titik kerusakan jalan melalui *map* yang disediakan

oleh *Google Maps API*. Pelapor dapat mencari alamat ataupun dengan melakukan navigasi berupa klik ataupun *drag* pada *map* untuk menentukan titik kerusakan. Gammu akan menangani pelaporan melalui *SMS*. *SMS* yang dikirimkan oleh pelapor harus memiliki *format* yang tepat dan alamat yang valid. Jika *SMS* tidak valid, maka *SMS* tidak dapat dikonversi menjadi laporan yang dapat tertampil pada aplikasi Bina Marga RRS.

Gambaran arsitektur dan alur sistem dapat dilihat pada Gambar 4.3 berikut.



Gambar 4.3 Arsitektur Sistem

Terdapat dua jalur pelapor memasukkan laporannya ke dalam aplikasi Bina Marga RRS, yaitu :

1. Melalui komputer/PC (*Personal Computer*)

Pelapor memiliki dua alat bantu untuk menunjuk titik kerusakan jalan, yaitu :

- a. Melakukan *drag and drop marker* ataupun dengan melakukan klik pada salah satu area *map*. Aplikasi akan

mencatat berapa *latitude* dan *longitude* dari *marker* yang merepresentasikan dimana *marker* itu berada. Melalui koordinat tersebut, aplikasi akan melakukan permintaan nama alamat dan kota kepada *Google Maps API*.

- b. Memasukkan alamat pada isian yang disediakan. Setelah pelapor menekan tombol pencarian maka aplikasi akan berkomunikasi dengan *Google Maps API* dan menampilkan posisi *marker* yang telah sesuai dengan alamat yang dimasukkan seperti yang ditunjukkan pada Gambar 4.4 di bawah. Alamat harus valid, jika tidak maka *Google Maps API* tidak akan mengenali alamat tersebut.



Gambar 4.4 Pelaporan jalan rusak melalui komputer

2. Melalui telepon genggam

Terdapat dua pilihan pelaporan melalui telepon genggam. Jika telepon genggam bisa melakukan koneksi internet, maka pelapor bisa mengakses aplikasi *Crowdsourcing* Pemetaan Kerusakan Jalan melalui *browser* telepon genggam atau *smartphone* yang mereka miliki. Hal tersebut dimungkinkan

karena aplikasi aplikasi Bina Marga RRS Jalan didesain memakai konsep *responsive design*. Tampilan aplikasi aplikasi Bina Marga RRS akan mengikuti ukuran layar dimana aplikasi tersebut dibuka, sehingga tampilan dan *user experience* akan semaksimal mungkin didapatkan sama seperti saat aplikasi dibuka dikomputer.

Cara lainnya adalah melalui *SMS* dengan memasukkan informasi dengan *format* tertentu ke nomor telepon Bina Marga. Berikut adalah *format SMS* yang harus dipatuhi oleh pelapor.

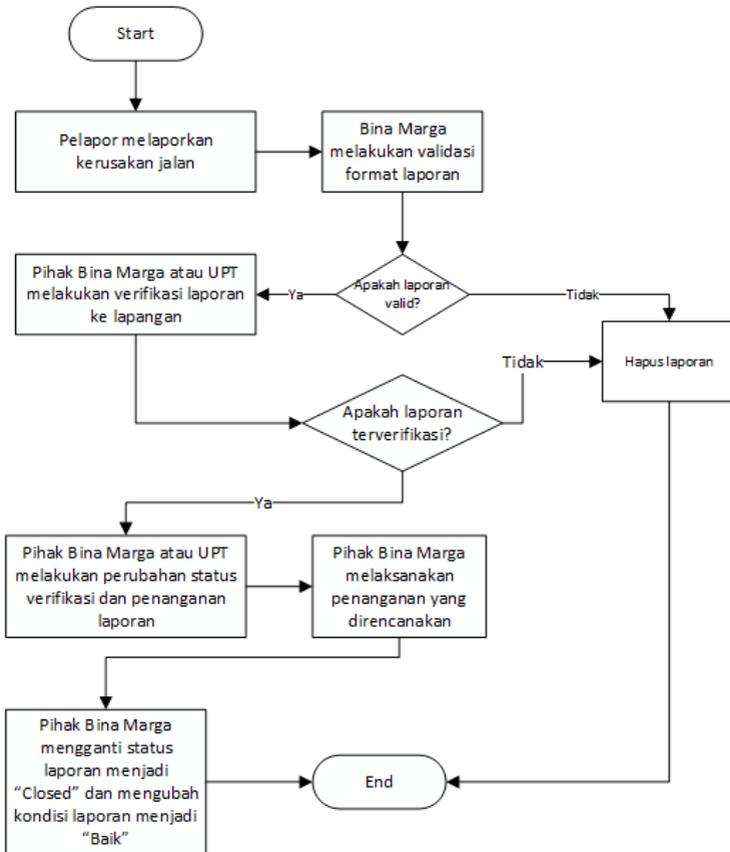
- *Format* : *username* * tingkat kerusakan * alamat lengkap * deskripsi
- Contoh : Usman123 * 3 * Gebang Wetan No.22a, Sukolilo, Surabaya * Sudah lebih dari 2 tahun jalan tidak diperbaiki

Keterangan :

- Jika pelapor belum memiliki akun, mereka bisa memasukkan nama mereka. Aplikasi akan membuatkan akun khusus untuk mereka jika nomor mereka belum terdaftar pada sistem. Ketika petugas Bina Marga mengeksport *SMS* mereka yang valid untuk menjadi sebuah laporan *web*, aplikasi akan mencatat juga isi *SMS* yang berisi *username* dan *password* pelapor yang telah diacak pada suatu tabel yang digunakan Gammu. Tabel tersebut menampung semua *SMS* yang akan dikirimkan ketika Gammu diaktifkan.
- Terdapat 4 kode kondisi kerusakan jalan, yaitu :
 - ✓ 1 : Baik
 - ✓ 2 : Sedang
 - ✓ 3 : Rusak Ringan
 - ✓ 4 : Rusak Berat

Setelah *SMS* terkirim ke nomor telepon Bina Marga, *Modem* yang telah terisi *SIM card* dengan nomor telepon Bina Marga akan dibaca oleh Gammu. Gammu kemudian akan mengambil informasi jalan rusak tersebut dan meneruskannya ke dalam Aplikasi *Crowdsourcing* Pemetaan Kerusakan Jalan. Informasi tersebut akan tercatat ke dalam *database*. Laporan melalui *SMS* tersebut belum tertampil jika belum di*export* ke dalam laporan *web*. Petugas Bina Marga yang bertanggung jawab melakukan *export*. Pada saat petugas memilih dan menekan tombol *export* laporan, nama jalan akan dikirim ke *Server Google Maps* untuk diubah menjadi koordinat, alamat dan kota.

Aplikasi Bina Marga dapat mengubah sedikit proses bisnis yang ada pada Bina Marga. Mulai dari proses penentuan jalan yang akan diverifikasi sampai kepada perubahan kondisi jalan yang telah diberikan penanganan. Gambar 4.5 menunjukkan alur proses bisnis Bina Marga jika mengimplementasikan aplikasi Bina Marga RRS.



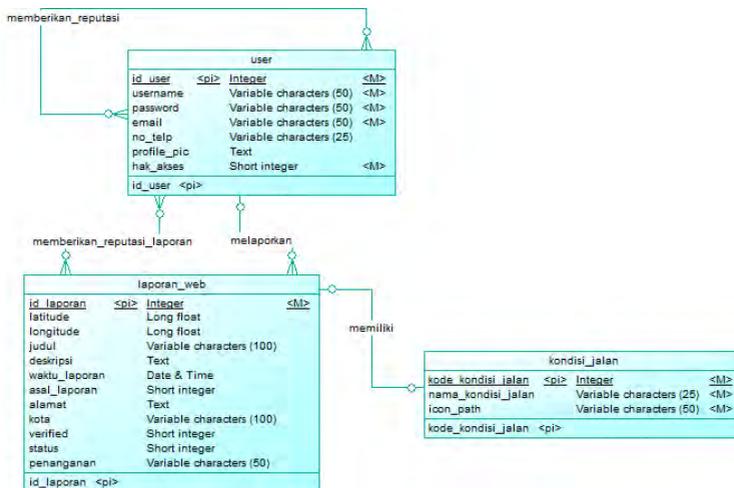
Gambar 4.5 Alur proses bisnis (*to-be*)

Proses bisnis dimulai ketika laporan-laporan dari masyarakat masuk. Kemudian, alamat dan lokasi laporan tersebut akan divalidasi. Lokasi jalan yang valid adalah lokasi yang memiliki titik marker yang tepat pada sebuah jalan tertentu. Lokasi jalan tersebut juga harus memiliki alamat yang dikelola Bina Marga. Selanjutnya, laporan tersebut akan diverifikasi dengan pengecekan langsung ke lokasi laporan. Jika laporan benar, Bina Marga akan mengubah status verifikasi laporan menjadi “terverifikasi” dan memberikan informasi penanganan apa yang

diberikan. Pihak Bina Marga akan melakukan penanganan dan jika telah selesai maka Admin Bina Marga akan mengubah status laporan menjadi “Closed” dan mengubah kondisi laporan menjadi “Baik”. Jika laporan tidak valid dan tidak terverifikasi, Bina Marga akan menghapus laporan tersebut.

4.3 Desain Database

Database yang digunakan pada aplikasi ini terdiri dari beberapa table untuk menyimpan data pada *MySQL*. Terdapat dua kategori tabel yang digunakan, yaitu tabel-tabel untuk pencatatan *SMS* oleh Gammu dan tabel-tabel yang dipakai Bina Marga RRS. Pada bagian ini akan dibahas mengenai tabel-tabel yang dipakai Bina Marga RRS. Berikut pada Gambar 4.6 adalah *Conceptual Data Model* (CDM) dari aplikasi *Crowdsourcing Pemetaan Kerusakan Jalan*. CDM dibuat dengan menggunakan *software Power Designer*.



Gambar 4.6 *Conceptual Data Model* aplikasi

Tabel 4.1 menjelaskan mengenai entitas yang terdapat pada ERD.

Tabel 4.1 Penjelasan entitas pada CDM

No	Entitas	Atribut	Penjelasan
1	laporan_web	id_laporan latitude longitude judul deskripsi waktu_laporan alamat kota verified status penanganan	Tabel ini digunakan untuk menyimpan informasi laporan yang dilaporkan, baik melalui <i>Website</i> maupun melalui <i>SMS</i>
2	<i>user</i>	id_user username password email no_telp profile_pic hak_akses	Tabel ini digunakan untuk menyimpan informasi tentang <i>user</i>
3	kondisi_jalan	kode_kondisi_jalan nama_kondisi_jalan icon_path	Tabel ini digunakan untuk menyimpan keterangan tentang kondisi jalan dan ikon yang akan digunakan pada <i>map</i>

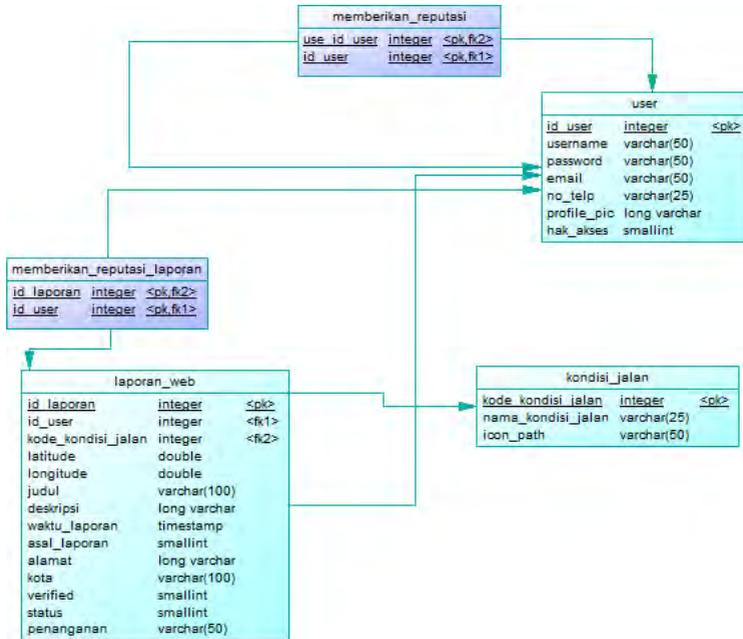
Tabel 4.2 menjelaskan relasi antar entitas pada CDM.

Tabel 4.2 Relasi antar entitas CDM

Relationship Type	Constraint	Penjelasan
MELAPORKAN on USER, LAPORAN_WEB	(1,N)	Setiap <i>user</i> dapat melaporkan lebih dari satu laporan

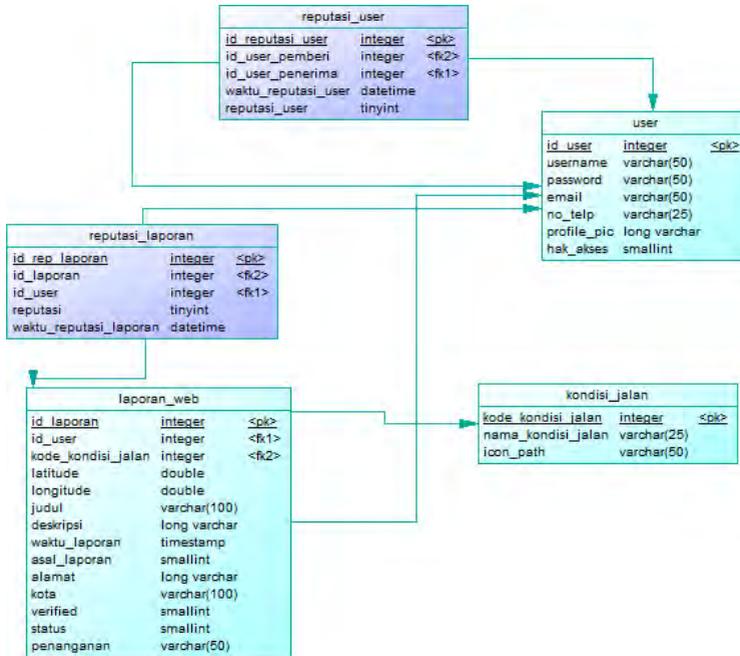
		dan satu laporan dilaporkan oleh satu <i>user</i>
MEMILIKI on LAPORAN_WEB, KONDISI_JALAN	(N,1)	Setiap laporan memiliki satu kondisi jalan dan satu kondisi jalan bias dimiliki oleh banyak laporan
MEMBERIKAN_REPUTASI on USER, USER	(N,N)	Setiap <i>user</i> bisa memberikan reputasi kepada lebih dari satu <i>user</i> dan satu <i>user</i> bisa diberikan reputasi oleh banyak <i>user</i> .
MEMBERIKAN_REPUTASI_LAPORAN on USER, LAPORAN_WEB	(N,N)	Satu <i>user</i> bisa memberikan reputasi pada banyak laporan dan satu laporan bisa diberikan reputasi oleh banyak <i>user</i>

Terdapat dua hubungan *many-to-many*. Dua hubungan tersebut itu akan menghasilkan masing-masing satu tabel baru jika *Conceptual Relationship Diagram* diubah menjadi *Physical Relationship Diagram*. Gambar 4.7 menunjukkan *Physical Relationship Diagram* aplikasi.



Gambar 4.7 Physical Data Model aplikasi

Dua tabel yang terbentuk adalah tabel `memberikan_reputasi_laporan` dan `memberikan_reputasi` sesuai dengan nama *relationship* antar entitas yang sebelumnya ada pada *Conceptual Relationship Diagram*. Dua tabel tersebut akan menyimpan catatan transaksi mengenai reputasi laporan dan reputasi *user*. Lalu, tambahkan beberapa atribut untuk memperjelas informasinya. *Physical Relationship Diagram* akan berubah menjadi seperti yang ditampilkan pada Gambar 4.8.



Gambar 4.8 PDM yang telah diperbaharui

Tabel 4.3 menjelaskan atribut yang telah ditambahkan pada PDM.

Tabel 4.3 Penjelasan atribut PDM yang ditambah

Tabel	Atribut	Penjelasan
reputasi_laporan	id_rep_laporan	Atribut yang digunakan sebagai <i>primary key</i> dan untuk keperluan <i>indexing</i>
	id_laporan	<i>Foreign key</i> yang menunjuk ke laporan yang diberikan reputasi
	id_user	<i>Foreign key</i> yang menunjuk pada <i>user</i>

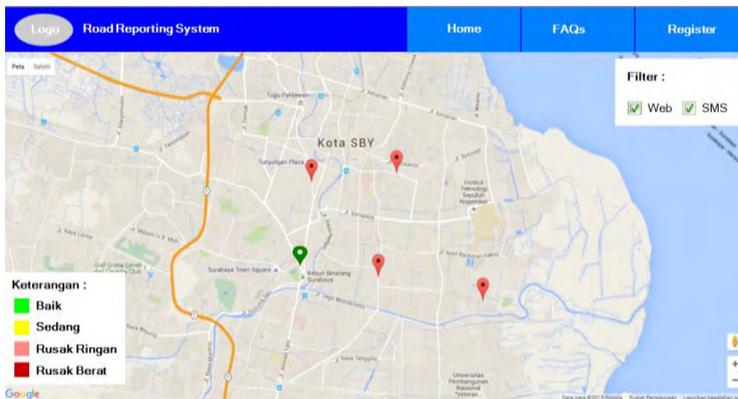
		yang memberikan reputasi pada laporan
	reputasi	Nilai reputasi yang diberikan, terdapat dua kemungkinan yaitu 1 (baik) dan -1 (buruk)
	waktu_reputasi_laporan	Waktu reputasi diberikan
reputasi_user	id_reputasi_user	Atribut yang digunakan sebagai <i>primary key</i> dan untuk keperluan indexing
	id_user_pemberi	<i>Foreign key</i> yang menunjuk kepada <i>user</i> yang memberikan reputasi
	id_user_penerima	<i>Foreign key</i> yang menunjuk kepada <i>user</i> yang menerima reputasi
	waktu_reputasi_user	Waktu reputasi diberikan
	reputasi_user	Nilai reputasi yang diberikan, terdapat dua kemungkinan yaitu 1 (baik) dan -1 (buruk). Untuk admin yang melakukan verifikasi dapat menambahkan 20 (terverifikasi) atau -20 (tidak terverifikasi) pada reputasi <i>user</i> .

4.4 Desain *Interface*

Interface aplikasi memakai kerangka dasar tampilan yang disediakan oleh *HTML5up* yang bisa diakses pada URL *HTML5up.com*. Berikut adalah beberapa desain *interface* yang akan digunakan dalam aplikasi ini :

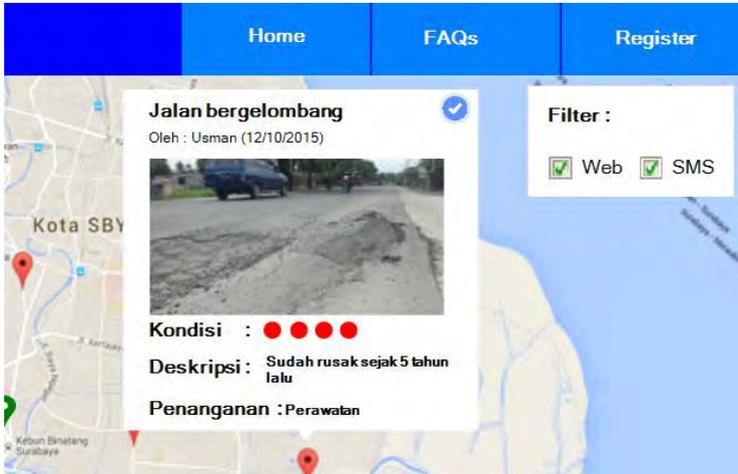
a. Beranda

Beranda dari aplikasi ini akan memuat sebuah peta dengan *marker-marker* dari 100 laporan terbaru yang masuk baik melalui *web* maupun melalui *SMS*. Beranda akan menampilkan keterangan dari warna-warna *marker* yang menunjukkan kondisi jalan. Selain itu, beranda juga memuat *filter* untuk menyaring laporan yang ingin diketahui, seperti *filter* asal laporan, kondisi, dan status verifikasi seperti yang ditunjukkan pada Gambar 4.9.



Gambar 4.9 Desain Halaman Beranda

Pengguna dapat melakukan klik pada salah satu *marker* untuk melihat informasi kerusakan jalan seperti yang tergambar pada Gambar 4.10.



Gambar 4.10 Informasi kerusakan jalan pada marker

b. Laporan Baru

Pelapor dapat memasukkan laporan dengan mencari melalui kotak pencarian yang tersedia maupun dengan mengklik dan *mendrag* suatu daerah pada peta. Jika *marker* diklik, maka akan muncul sebuah *window* yang berisi *form* yang harus diisi seperti yang digambarkan pada Gambar 4.11.

Gambar 4.11 Desain Halaman Laporan Baru

c. Dashboard

Pengguna dapat melihat statistik-statistik penting terkait laporan kerusakan pada halaman *dashboard*. Pada halaman *dashboard* terdapat grafik-grafik untuk mempermudah pengguna untuk mengambil informasi dari *data* yang tersedia seperti yang digambarkan pada Gambar 4.12.



Gambar 4.12 Desain Halaman *Dashboard*

BAB V IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai implementasi dari perancangan yang telah dilakukan pada bab sebelumnya. Pada bagian implementasi akan dijelaskan mengenai lingkungan implementasi, pembuat fitur-fitur aplikasi dalam bentuk kode, serta proses pengujian aplikasi.

5.1 Lingkungan Implementasi

Pengembangan aplikasi ini menggunakan komputer dengan spesifikasi seperti pada Tabel 5.1 berikut.

Tabel 5.1 Spesifikasi Komputer (Server)

Prosesor	Intel® Core™ i3 CPU M330 @ 2.13GHz
Memory	5000 MB RAM
Sistem Operasi	Windows 7 (64-bit)

Selain itu, dalam pengembangan aplikasi ini juga memanfaatkan beberapa teknologi lain seperti *editor*, *database*, *server*, bahasa pemrograman dan *library* yang disajikan dalam Tabel 5.2 berikut.

Tabel 5.2 Teknologi yang Digunakan

Websserver	Apache 2.4.12
Bahasa Pemrograman	PHP 5.6.8, Javascript
Database	MySql 5.6.24
Editor	Netbeans 8.0.1
Browser	Google Chrome 43.0.2357.124 m
Library	- Gammu v1.33 (SMS Gateway) - jQuery v 1.11.3 (Javascript library) - Font Awesome v 4.4.0 (Icon) - HighCharts v 4.1.9 (Chart)

Performance Testing	jMeter v 2.13
----------------------------	---------------

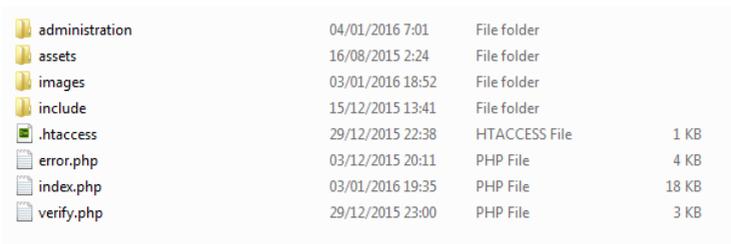
Untuk dapat menerima dan mengirim *SMS*, pada pengembangan aplikasi ini memakai *modem* vodafone K3565. Tabel 5.3 menampilkan spesifikasi *modem* vodafone K3565.

Tabel 5.3 Spesifikasi modem

Atribut	Spesifikasi
Sistem Operasi	Windows 2000, Windows XP, Windows Vista, Windows 7 dan Mac OS
Frekuensi radio	HSDPA UMTS : 900, 2100MHz
Dukungan koneksi	HSDPA : 3,6 MBit/s EDGE : 236,8 KBit/s GPRS : 57,6 KBit/s
Micro SD	Ya, sampai 8GB

5.2 Direktori Aplikasi

Aplikasi RRS Bina Marga memiliki dua direktori utama, yaitu direktori untuk *front-end* seperti pada Gambar 5.1 dan direktori untuk *back-end* seperti pada Gambar 5.2.



administration	04/01/2016 7:01	File folder	
assets	16/08/2015 2:24	File folder	
images	03/01/2016 18:52	File folder	
include	15/12/2015 13:41	File folder	
.htaccess	29/12/2015 22:38	HTACCESS File	1 KB
error.php	03/12/2015 20:11	PHP File	4 KB
index.php	03/01/2016 19:35	PHP File	18 KB
verify.php	29/12/2015 23:00	PHP File	3 KB

Gambar 5.1 Direktori *Front-end* Aplikasi

 assets	16/08/2015 2:24	File folder	
 images	03/01/2016 11:46	File folder	
 include	24/12/2015 21:09	File folder	
 .htaccess	21/12/2015 23:18	HTACCESS File	1 KB
 index.php	03/01/2016 19:47	PHP File	28 KB

Gambar 5.2 Direktori Back-end Aplikasi

Aplikasi ini memiliki *file* `index.php` yang berfungsi untuk mengarahkan *file* apa yang akan dimasukkan ke dalam halaman. Pada masing-masing direktori terdapat *file-file*, yaitu :

- Assets : memuat *file-file* berkaitan dengan *interface* dari kerangka tampilan *HTML5up*.
- Images : menampung *file-file* gambar.
- Include : memuat *file-file* inti dari program yang berisi fungsi-fungsi dan fitur yang telah direncanakan.

Pada kedua direktori tersebut terdapat *file* `.htaccess` yang berfungsi untuk melakukan *redirect URL* halaman sehingga lebih sederhana dan mudah diingat. Contoh perubahannya adalah sebagai berikut :

`RRS.com/index.php?menu=dashboard` → `RRS.com/dashboard`

5.3 Konfigurasi Aplikasi

Pertama-tama, koneksi *database* harus diatur terlebih dahulu. Direktori *file* `conn.PHP` terdapat di direktori `include/php/conn.php`. Atribut yang harus diatur adalah nama *database*, *username*, dan *password database* seperti yang terlihat pada Gambar 5.3.

```

1. <?php
2. if(!isset($conn_i)) {
3.     die('Direct access not permitted');
4. }
5. global $db;
6. $dbHost = "localhost";
7. $dbUser = "root";

```

```

8. $dbPass = "";
9. $dbDatabase = "bmr";
10. //connect to the database
11. $db = mysql_connect ($dbHost, $dbUser, $dbPass) or die ("I can
not connect to the database because: " . mysql_error());
12. mysql_select_db($dbDatabase, $db) or die ("I cannot select the
database '$dbname' because: " . mysql_error());
13. ?>

```

Gambar 5.3 Potongan kode conn.php

5.4 Pembuatan Aplikasi

Pada bagian ini akan dijelaskan penerapan desain aplikasi dalam bentuk kode program, yaitu bahasa pemrograman *Javascript* dan *PHP*.

5.4.1 Fungsi Login

Login dalam aplikasi RRS Bina Marga menggunakan *AJAX* untuk melakukan validasi *username* dan *password* yang pengguna masukkan. *Form login* muncul dalam bentuk sebuah *pop-up window* seperti pada Gambar 5.4.

Gambar 5.4 Form login

Form login memiliki sebuah *captcha* untuk mencegah terjadinya *spamming* yang dapat membebani *database*. Bila *username* dan *password* tidak cocok maka *captcha* tersebut akan berganti nilainya. Nilai *captcha* didapat dari hasil penjumlahan dua angka acak. Gambar 5.5 menampilkan potongan gambar untuk melakukan validasi *captcha*.

```

1. $("#confirm-Login").click( function() {
2.     username=$("#username").val();
3.     password=$("#password").val();
4.     isi_cap = document.getElementById('captcha_login').value;
5.     cap = document.getElementById('captcha_val_login').value;
6.
7.     if(username == '' && password == '' && isi_cap == ''){
8.         toastr.error('Semua field harus diisi');
9.         return false;
10.    }
11.
12.    if(isi_cap != cap){
13.        toastr.error('Captcha salah');
14.        return false;
15.    }

```

Gambar 5.5 Potongan kode validasi *captcha*

Proses *login* diinisiasi ketika tombol *confirm-login* diklik. Program akan mengecek apakah isian *form login* kosong dan mengecek apakah hasil masukan *captcha* sudah benar. Jika sudah benar, maka program akan melanjutkan untuk memanggil fungsi *AJAX* seperti pada Gambar 5.6.

```

1. $.ajax({
2.     type: "POST",
3.     url: "include/PHP/Login.PHP",
4.     data: "name="+username+"&pwd="+password+"&conn_i=true",
5.     success: function(HTML){
6.         if(HTML=='true') {
7.
8.             toastr.success('Login berhasil, Anda akan diredire
ct dalam 5 detik');

```

```

9.         setTimeout(function() {
10.             window.location.href = "./"
11.             }, 5000);
12.         } else {
13.             toastr.error(HTML);
14.             document.getElementById('captcha_login').value = '
';
15.             var c1 = Math.floor((Math.random() * 20) + 1);
16.             var c2 = Math.floor((Math.random() * 20) + 1);
17.             var c3 = c1+c2;
18.             document.getElementById('captcha_val_login').value
= c3;
19.             document.getElementById('captcha_login').setAttrib
ute('placeholder', 'Masukkan hasil ' + c1 + " + " + c2);
20.         }
21.     }
22. });

```

Gambar 5.6 Potongan kode pemanggilan AJAX

Program akan memanggil *file PHP* yang akan mengecek kecocokan *username* dan *password*. Jika berhasil halaman akan diredirect ke Halaman Beranda.

5.4.2 Fungsi Register

Pengguna yang ingin membuat akun baru harus mengisi *form* registrasi yang tampil sebagai *pop-up*. Adapun *data-data* yang harus diisi adalah seperti *username*, *password*, *email*, nomor telepon, dan *profile picture*. Berikut pada Gambar 5.7 adalah tampilan *form* untuk *register*.

The image shows a registration form with the following elements:

- A close button (x) in the top left corner.
- The title "REGISTER" centered at the top.
- Four input fields stacked vertically: "Username", "Password", "Confirm Password", and "Email".
- A "Choose File" button with the text "No file chosen" next to it.

Gambar 5.7 Form Register

Form tersebut akan divalidasi masuknya dengan menggunakan plugin jQuery Validator yang bisa didownload pada URL <http://www.formvalidator.net/>. Plugin ini akan memvalidasi beberapa isian, yaitu :

- Mengecek apakah nama *user* dan *email* yang diisi sudah terdaftar
- Mengecek kekuatan *password* (8 karakter kombinasi angka dan huruf)
- Mengecek kesamaan *password* yang dimasukkan
- Mengecek *file profile picture* yang diupload (*file* gambar dengan *format* .jpg, .jpeg, .png max. 10 MB)
- Mengecek nomor telepon (angka)

Untuk memakai plugin Form Validator tersebut, atribut pada input harus ditambahkan sesuai dengan ketentuan pengecekan. Berikut pada Gambar 5.8 adalah salah satu potongan kode input *form* yang divalidasi.

```

1. <input data-validation="length alphanumeric server"
2.   data-validation-length="min4" data-validation-
   url= "http://<?PHP echo $domain; ?>/RRS/include/PHP/check_use
   rname.PHP?check_username_i=true"

```

```

3. id="username_reg" name="username_reg" placeholder=
4. "Username" type="text">
5.
6. <input data-validation="strength" data-validation-help=
7. "Password minimal 8 karakter, kombinasi angka dan huruf"
8. data-validation-strength="2" id="password_reg" name=
9. "pass_confirmation" placeholder="Password" type="password">
10.

```

Gambar 5.8 Potongan kode input form registrasi

Selanjutnya, aplikasi akan melakukan penyimpanan *data* ke *database* lalu mengirimkan *email* untuk alamat verifikasi. *Email* dikirim menggunakan sebuah kelas dan fungsi untuk pengiriman *email* yaitu Mail.php. Kelas Mail dapat diinstal menggunakan sebuah *framework* PEAR dan mengaktifkan *support* koneksi SSL ke dalam *PHP*. Berikut pada Gambar 5.9 adalah potongan kode yang digunakan untuk memasukkan kelas Mail.php dan mengirimkan *email*.

```

1. require_once "Mail.PHP";
2. $headers = array('From' => $from, 'To' => $to,
3. 'Subject' => $subject);
4. $smtp = Mail::factory('smtp', array('host' => $host,
5. 'port' => $port, 'auth' => true,
6. 'username' => $username, 'password' => $password));
7.
8. $mail = $smtp -> send($to, $headers, $body);
9.
10. if (PEAR::isError($mail)) {
11. echo("Email aktivasi gagal dikirim");
12. } else {
13. echo "Silahkan ikuti tautan yang telah kami kirim pada
14. email Anda untuk mengaktifasi akun";

```

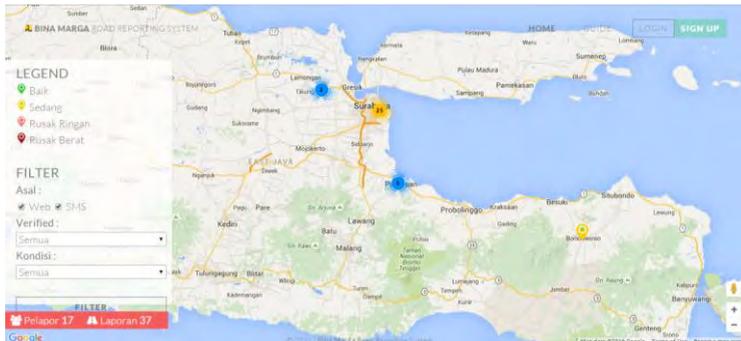
Gambar 5.9 Potongan kode pengiriman *email*

Email dikirimkan dengan *format* sesuai dengan *email* pada umumnya, yaitu adanya *subject* dan *body*. *Email* dikirimkan dengan menggunakan akun Gmail, jadi protokol SMTP untuk

pengiriman *email* harus diatur juga sesuai dengan ketentuan yang Google tetapkan.

5.4.3 Fungsi Peta Google Maps

Laporan-laporan yang masuk akan muncul sebagai *marker* dan *marker cluster* pada peta Google. Gambar 5.10 menampilkan peta dengan laporan-laporan yang telah masuk.



Gambar 5.10 Tampilan laporan pada peta

Pertama-tama, buat sebuah tag div yang akan menampung Google Maps. Berikut pada gambar 5.11 adalah potongan kodenya.

```

1. <div class="google-map wrap" itemscope
2.   itemprop="hasMap" itemType="http://schema.org/Map">
3.   <div id="google-map" class="google-map"></div>
4. </div>

```

Gambar 5.11 Potongan kode *container* peta

Selanjutnya, ambil *data-data* laporan dari *database* dan *filter* berdasarkan *filter* yang diinginkan pengguna. Laporan yang ditampilkan adalah 100 laporan terbaru yang masuk. Banyaknya laporan dibatasi untuk mempercepat proses

pemuatan halaman. Gambar 5.12 merupakan potongan kode program untuk mengambil *data* laporan pada *database*.

```

1. $locations = array();
2.
3. $sql = 'SELECT * FROM laporan_web inner join kondisi_jalan on
kondisi_jalan.kode_kondisi_jalan =laporan_web.kode_kondisi_jal
an join user on user.id_user = laporan_web.id_user';
4. $pre_qry = " WHERE ";
5.
6. if(!empty($safe_POST) AND (!isset($safe_POST['asl_web'])
AND !isset($safe_POST['asl_SMS']))) {
7. $sql .= $pre_qry." laporan_web.asal_laporan = -1 ";
8. $pre_qry = " AND ";
9. } else if(isset($_POST['asl_web']) XOR isset($_POST['asl_SMS']
)){
10.     if(isset($_POST['asl_web'])){
11.         $sql .= $pre_qry." laporan_web.asal_laporan = 0 ";
12.         $pre_qry = " AND ";
13.     }

```

Gambar 5.12 Potongan kode pengambila data laporan

Data array yang dihasilkan dari *database* akan disimpan dalam *array* \$locations. *Array* tersebut akan menampung atribut-atribut dari *marker* yang akan ditampilkan. Gambar 5.13 menampilkan potongan kode yang berisi struktur *array* locations.

```

1. $locations[] = array('google_map' =>
2. array('lat' => $row['latitude'],
3. 'lng' => $row['longitude'], ),
4. 'content' => $content,
5. 'location_name' => judul,
6. 'icon' => $row['icon_path'],
7. 'id_laporan' => $row['id_laporan']
8. );

```

Gambar 5.13 Struktur *array* locations

Selanjutnya, *map* Google perlu diinisiasikan, dimana titik tengah, berapa level *zoom*, dan properti lain dari *map*. Kemudian, setiap *data* pada *array* *locations* akan dimasukkan dalam sebuah objek *marker*. *Marker-marker* itu akan dimasukkan ke dalam *map*. Terakhir, plugin *Marker Clusterer* dipanggil agar fungsi *clustering marker* dapat berjalan. Gambar 5.14 menampilkan potongan kode untuk memasukkan data *marker* yang sudah diambil ke dalam *map*.

```

1.  jQuery(document).ready(function($) {
2.  var is_touch_device = 'ontouchstart' in document.documentElement;
3.  map = new google.maps.Map(document.getElementById('google-
4.  map'), {
5.  zoom: 9,
6.  center: new google.maps.LatLng('-7.604423', '112.772419'),
7.  mapTypeId: google.maps.MapTypeId.ROADMAP,
8.  mapTypeControl: false
9.  });
10.   var bounds = [];
11.   var markers = []; <? PHP
12.   foreach($locations as $location) {
13.       var latlng = new google.maps.LatLng( <? PHP echo $
14.       map_lat; ?> , <?
15.       PHP echo $map_lng; ?> );
16.       bounds.push(latlng);
17.       var marker_cur = new google.maps.Marker({
18.           position: latlng,
19.           icon: '<?PHP echo $icn; ?>'
20.       });
21.       addInfoWindow(marker_cur, '<?PHP echo $cnt; ?>',
22.           '<?PHP echo $id_lap; ?>');
23.       markers.push(marker_cur); <? PHP
24.   }

```

Gambar 5.14 Potongan kode *marker* dimasukkan pada peta

5.4.4 Fungsi Buat Laporan Baru

Untuk membuat sebuah laporan baru dari *web*, Google Maps perlu dimasukkan kembali ke dalam halaman seperti yang

sudah dijelaskan sebelumnya. Kemudian, satu *marker* dibuat dengan posisi *default* yang berada pada Kampus Institut Teknologi Sepuluh Nopember (ITS) menggunakan fungsi `placeMarker()`. Berikut pada Gambar 5.15 adalah potongan kode program untuk menempatkan sebuah *marker*.

```

1. function placeMarker(location) {
2.   var contentString = document.getElementById('form_laporan').innerHT
   ML;
3.     var infowindow = new google.maps.InfoWindow({
4.       content: contentString
5.     });
6.     if (marker) {
7.       marker.setPosition(location);
8.     } else {
9.       marker = new google.maps.Marker({
10.        position: location,
11.        map: map,
12.        draggable: true
13.      });
14.      infowindow.open(map, marker);
15.    }
16.    marker.addListener('click', function() {
17.      infowindow.open(map, marker);
18.    });
19.    google.maps.event.addListener(marker, 'dragend', function(a) {
20.      document.getElementById('koordinat_input').value = a.latLng;
21.      geocodeLatLng(geocoder, a.latLng);
22.    });
23.  }

```

Gambar 5.15 Potongan kode fungsi `placeMarker()`

Pelapor harus memposisikan *marker* pada titik kerusakan jalan yang ingin dilaporkan. Terdapat tiga cara pelapor memposisikan *marker*, yaitu :

- a. Melakukan klik pada *map*
Jika ada klik yang dilakukan pada *map*, maka sebuah listener akan menangkap *event* tersebut dan mengubah

posisi *marker* keposisi dimana klik dilakukan. Kemudian, program memanggil fungsi `geocodeLatLng()` untuk mengambil nama kota dan alamat lengkap dari koordinat *marker* sekarang. Gambar 5.16 merupakan potongan kode program untuk membuat sebuah *event map click*.

```
1. google.maps.event.addListener(map, 'click', function(event) {
2.     placeMarker(event.latLng);
3.     geocodeLatLng(geocoder, event.latLng);
4. });
```

Gambar 5.16 *Event map click*

b. Melakukan *drag* pada *marker*

Listener *event drag* pada *marker* akan dibuat pada saat fungsi `placeMarker()` dipanggil. Koordinat yang menjadi tempat *marker* sekarang akan disimpan pada sebuah input rahasia. Gambar 5.17 merupakan potongan kode untuk membuat *event marker drag*,

```
1. google.maps.event.addListener(marker, 'dragend', function(a) {
2.     document.getElementById('koordinat_input').value = a.latLng;
3.     geocodeLatLng(geocoder, a.latLng);
4. });
```

Gambar 5.17 *Event marker drag*

c. Mencari dengan alamat

Mencari alamat dilakukan dengan menggunakan fungsi *Javascript* `geocodeAddress()`. Aplikasi akan menggunakan kelas `geocoder` untuk melakukan koneksi dengan *server* Google Maps untuk meminta informasi jalan yang dimasukkan oleh pelapor. Banyak informasi yang akan dikirimkan oleh *server* Google Maps tetapi aplikasi hanya memakai koordinat, kota, dan alamat lengkap. Jika berhasil, program akan mengubah posisi *marker* pada koordinat yang ditemukan jika tidak maka akan muncul *error*.

Gambar 5.18 menampilkan potongan kode program fungsi `geocodeAddress()`.

```

1. function geocodeAddress(geocoder, resultsMap, marker) {
2.     var address = document.getElementById('address').value;
3.     geocoder.geocode({
4.         'address': address
5.     }, function(results, status) {
6.         if (status === google.maps.GeocoderStatus.OK) {
7.             var add = results[0].formatted_address;
8.             var value = add.split(",");
9.             count = value.length;
10.            country = value[count - 1];
11.            state = value[count - 2];
12.            city = value[count - 3];
13.            map_recenter(resultsMap, results[0].geometry.location,
14.                0, -220);
15.            marker.setMap(resultsMap);
16.            marker.setPosition(results[0].geometry.location);
17.            var koordinat = results[0].geometry.location;
18.            document.getElementById('koordinat_input').value =
19.                results[0].geometry.location;
20.            document.getElementById('alamat_input').value = add;
21.            document.getElementById('kota_input').value = city;
22.        } else {
23.            alert('Geocode was not successful for the following reason: '
24.                + status);
25.        }
26.    });
27. }

```

Gambar 5.18 Potongan kode fungsi `geocodeAddress()`

Setelah posisi *marker* sudah ditentukan, pelapor harus memasukkan beberapa informasi dasar, seperti judul, kondisi jalan, deskripsi, dan *captcha*. Judul dan *captcha* harus diisi sedangkan sisanya optional. Jika sudah mengisi *form* yang muncul pada *marker*, pelapor melakukan klik pada tombol Save dan program akan menyimpannya ke dalam *database*.

5.4.5 Fungsi SMS Gateway

a. Konfigurasi Awal Gammu

Download Gammu sesuai dengan versi Sistem Operasi *Server*. Pertama-tama, tabel-tabel *database default* yang diperlukan Gammu perlu *diimpor* pada *database* aplikasi. Tabel-tabel ini berguna untuk mencatat *data-data SMS* yang masuk pada *SIM Card*. Buka *file .zip* yang telah didownload dan ekstrak pada suatu direktori. Berikut pada Gambar 5.19 adalah direktori-direktori yang ada di dalamnya setelah diekstrak.

 bin	04/12/2015 20:16	File folder
 include	04/12/2015 19:19	File folder
 lib	04/12/2015 19:19	File folder
 share	04/12/2015 19:19	File folder

Gambar 5.19 Direktori Gammu

Import tabel-tabel yang diperlukan dengan menggunakan *script mysql.sql* yang berada pada direktori `gammu\share\doc\gammu\examples\sql`. Setelah proses *impor* selesai maka akan terbentuk tabel-tabel baru, yaitu :

- Daemons
- Gammu
- Inbox
- Outbox
- Outbox_multipart
- Pbk
- Pbk_groups
- Phones
- Sentitems

Selain tabel-tabel tersebut, terdapat satu tabel tambahan yaitu tabel Archive yang berguna untuk menyimpan SMS yang telah diekspor. *Script* yang dieksekusi sebelumnya akan membuat

beberapa *trigger*. Berikut adalah daftar tabel *database* aplikasi setelah melakukan *impor* tabel-tabel tersebut, seperti pada gambar berikut.

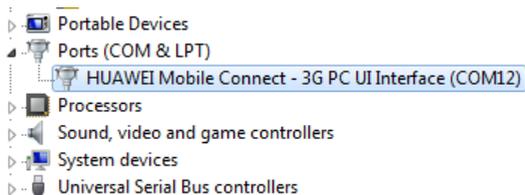
<input type="checkbox"/>	archive	Browse	Structure
<input type="checkbox"/>	daemons	Browse	Structure
<input type="checkbox"/>	gammu	Browse	Structure
<input type="checkbox"/>	inbox	Browse	Structure
<input type="checkbox"/>	kondisi_jalan	Browse	Structure
<input type="checkbox"/>	laporan_web	Browse	Structure
<input type="checkbox"/>	outbox	Browse	Structure
<input type="checkbox"/>	outbox_multipart	Browse	Structure
<input type="checkbox"/>	pbk	Browse	Structure
<input type="checkbox"/>	pbk_groups	Browse	Structure
<input type="checkbox"/>	phones	Browse	Structure
<input type="checkbox"/>	reputasi_laporan	Browse	Structure
<input type="checkbox"/>	reputasi_user	Browse	Structure
<input type="checkbox"/>	sentitems	Browse	Structure
<input type="checkbox"/>	user	Browse	Structure

Gambar 5.20 Daftar tabel aplikasi

Selanjutnya, *modem* yang dipakai untuk menerima *SMS* harus dipastikan telah didukung oleh Gammu. Untuk mengeceknya, buka alamat <http://wammu.eu/phones/> dan cari tipe *modem* yang digunakan. Catat kode koneksi sesuai dengan tipe *modem*. Dalam Tugas Akhir ini, *modem* yang dipakai adalah Vodafone K3565 dengan koneksi at.

Terdapat dua *file* konfigurasi yang harus diatur untuk dapat menjalankan Gammu, yaitu *gammurc* dan *smsdrc* yang terdapat pada direktori *gammu/bin*. Pertama, atur konfigurasi yang ada pada *file* *gammurc*. Ada dua komponen yang perlu diatur, yaitu nomor *port* yang digunakan *modem* dan tipe koneksi *modem* yang sudah dijelaskan pada bagian sebelumnya. Untuk mengecek nomor *port* buka *device manager* pada windows.

Nomor *port* akan tertera pada bagian *ports*, seperti pada Gambar 5.21.



Gambar 5.21 Nomor *port* modem

Buka *file* gammurc menggunakan *editor text* yang digunakan untuk *programming*, seperti notepad++. Ubah baris kode yang mengatur nomor *port* dan tipe koneksi *modem* yang sesuai. Gambar 5.22 menunjukkan pengaturan untuk *file* gammurc.

```
1. device = com12:
2. connection = at
```

Gambar 5.22 Pengaturan *file* gammurc

Selanjutnya, ubah pengaturan pada *file* smsdrc. Berikut pada Gambar 5.23 adalah komponen-komponen yang harus diubah.

```
1. device = com12
2. connection = at
3. service = sql
4. logfile = SMSdlog
5. debuglevel = 0
6. commtimeout = 30
7. sendtimeout = 30
8. user = root
9. password =
10. pc = localhost
11. database = bmrr
12. driver = native_mysql
```

Gambar 5.23 Pengaturan *file* smsdrc

Terdapat konfigurasi *database* yang dipakai, nilainya menyesuaikan dengan konfigurasi pada *database* yang dipakai tersebut. Setelah semua telah terkonfigurasi dengan benar, *service* Gammu perlu diinstall dan dijalankan pada *server*. Buka *command prompt* dan ubah direktori sesuai dengan tempat Gammu diekstrak. Lalu, ketik perintah “gammu --identify” dan tekan tombol enter. Gambar 5.24 menampilkan hasil dari perintah tersebut. Perintah ini digunakan untuk mengecek apakah semua *modem* sudah terhubung dan terkonfigurasi dengan baik.

```
c:\gammu\bin>gammu --identify
Device           : com12:
Manufacturer     : Huawei
Model            : unknown (K3565)
Firmware         : 11.608.13.00.37
IMEI             : 359574036208735
SIM IMSI        : 510891636727763
```

Gambar 5.24 Hasil perintah Gammu *identify*

Install dan jalankan *service* Gammu, jalankan perintah “gammu-smsd -c smsdrc -i” untuk memasang *service* gammu dan “gammu-smsd -c smsdrc -s” untuk menjalankan *service* gammu seperti pada Gambar 5.25. Jika *command prompt* menampilkan pesan “Service GammuSMSD started successfully” maka *service* Gammu sudah berjalan dan siap digunakan.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation

C:\Windows\system32>cd c:/gammu/bin

c:\gammu\bin>gammu-smsd -c smsdrc -i
Service GammuSMSD installed successfully

c:\gammu\bin>gammu-smsd -c smsdrc -s
Service GammuSMSD started successfully
```

Gambar 5.25 *Install* dan menjalankan *service* Gammu

b. Menampilkan *SMS* Inbox

Gammu menyimpan semua *data SMS*nya pada tabel-tabel yang sebelumnya diimpor. *SMS* yang masuk akan ditambahkan *datanya* langsung pada tabel *inbox*, *SMS* yang terkirim pada *sentitems*, dan *SMS* yang akan dikirim pada *outbox*.

Untuk menampilkan *SMS* yang ada pada suatu nomor telepon, aplikasi hanya perlu melakukan proses *query* seperti biasa. Berikut pada Gambar 5.26 adalah potongan kode untuk menampilkan *SMS* yang masuk.

```

1. $query = "SELECT ID,ReceivingDateTime,SenderNumber,TextDecoded
    FROM inbox";
2. $pre_qry = " WHERE ";
3.
4. if(isset($safe_GET['nomor'])){
5.     if($safe_GET['nomor'] != ""){
6.         $query .= $pre_qry." SenderNumber LIKE '%".$safe_GET['nomor'].
            "%";
7.         $pre_qry = " AND ";
8.     }
9. }

```

Gambar 5.26 Potongan kode *inbox SMS Gateway*

c. Mengekspor *SMS*

SMS yang masuk pada *inbox* bisa diekspor sehingga isi *SMS* dapat diparsing dan masuk sebagai laporan yang bisa tertampil pada peta. Untuk mengekspor, *user* memberukan tanda centang pada baris *SMS* yang akan diekspor lalu melakukan klik pada tombol *Export* seperti pada Gambar 5.27.

Inbox (11) Sent (3) Outbox (0) Archive (8)

Inbox

Filter

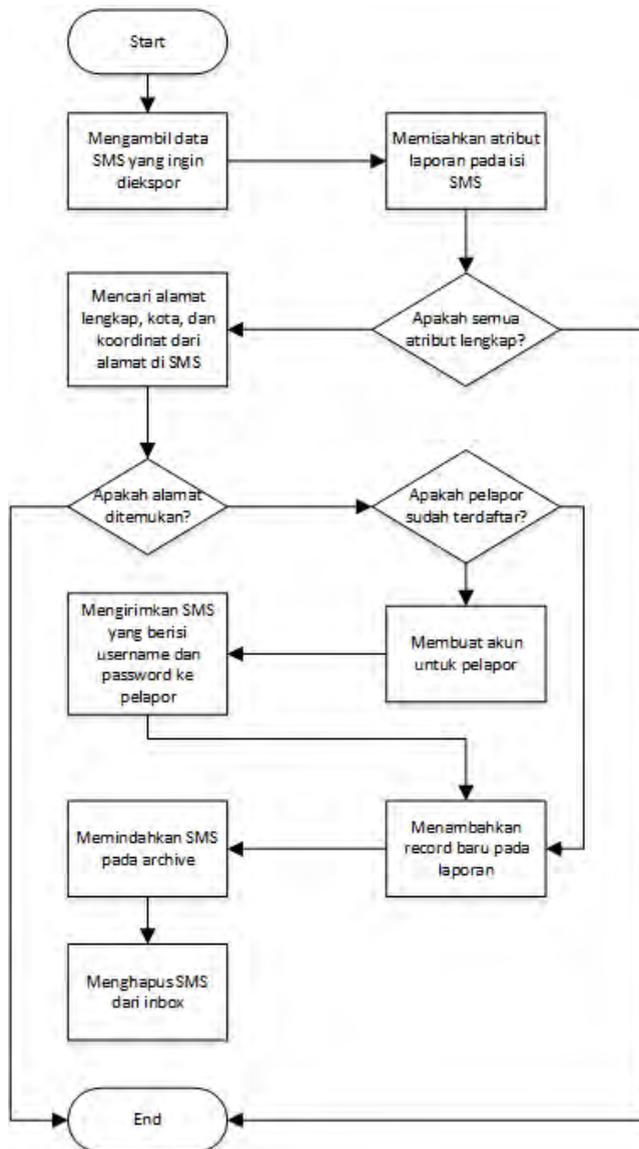
Sort

Export Delete

	KODE	WAKTU	PENGIRIM	ISI
<input checked="" type="checkbox"/>	13	2016-01-05 12:57:51	3	Yohan * 4* Jl. Bratang gede no.13 * Lubang didepan terminal bratang
<input type="checkbox"/>	14	2016-01-05 12:58:02	3	Pelanggan Yth, pengiriman SMS ke no 0885719363149 gagal. Silahkan coba bbrp saat lagi, pastikan pulsa/limit Anda cukup & masih dlm masa aktif. Info kartu Tri te
		2015-		

Gambar 5.27 Tampilan inbox SMS Gateway

Setiap SMS akan memanggil sebuah fungsi AJAX untuk memanggil *script* PHP yang akan mengekspor SMS tersebut. Berikut pada Gambar 5.28 adalah *flowchart* proses ekspor SMS. Isi SMS akan dibagi dan dimasukkan ke dalam array. SMS dipisahkan dengan tanda “*”. Isi SMS akan dicek apakah valid atau tidak. SMS tidak valid jika terdapat salah satu dari 4 index array yang belum diset. Kemudian, alamat dari SMS tersebut akan dicari *longitude*, *laltitude*, dan kotanya menggunakan fungsi *lookup*.



Gambar 5.28 Flowchart proses ekspor SMS

Fungsi lookup akan meminta JSON yang berisi informasi koordinat dan lainnya dari url yang mengarah ke *Server* Google Maps. Untuk menggunakan fungsi ini, *cURL* pada *PHP* sudah harus terinstal dan aktif. JSON yang dikirimkan oleh Google Maps kemudian didecode menjadi sebuah *array*. Kemudian, informasi koordinat, kota, dan alamat lengkap diekstrak dengan mengakses index *array* yang tepat.

Jika *array* yang dikembalikan tidak kosong, aplikasi akan mengecek apakah ada nama pelapor dengan nomor teleponnya tercatat pada tabel *user*. Jika belum, maka pelapor tersebut akan dibuatkan sebuah akun dengan *password* acak. Aplikasi akan membuat *record* pada tabel *outbox* untuk mengirimkan *SMS* kembali pada pelapor bahwa laporannya sudah bisa dilihat pada Bina Marga RRS dan pelapor tersebut sudah dibuatkan akun dengan *password* tertentu. Laporan dari *SMS* beserta kelengkapannya yang telah dicari akan dimasukkan pada tabel *laporan_web* dan *SMS* akan dimasukkan pada tabel *archive*.

5.4.6 Fungsi Pemberian Reputasi

Terdapat dua reputasi yang terdapat di Bina Marga RRS, yaitu reputasi untuk *user* dan laporan. Satu *user* hanya bisa sekali memberikan reputasi pada laporan atau *user* yang sama. Cara kerja pemberian reputasi *user* dan laporan hampir sama. Reputasi diberikan dengan melakukan klik pada tombol setuju (*agree/good*) ataupun tidak setuju (*disagree/bad*). Setelah tombol diklik, maka akan dipanggil fungsi AJAX dan kemudian ada *file PHP* yang dipanggil untuk menambahkan reputasi pada *database*. Gambar 5.29 merupakan potongan kode program *PHP* yang dipanggil.

```

1. $rep = $_POST['rep'];
2. $id_user = $_POST['id_user'];
3. $id_user_pemberi = $_POST['id_pemberi'];
4. $qry = "SELECT * FROM reputasi_user WHERE id_user_penerima=".$
    id_user." AND id_user_pemberi = ".$id_user_pemberi;
5. $res = MySQL_query($qry) or die('ERROR!!!');

```

```

6. $num_row = MySql_num_rows($res);
7. if( $num_row > 0 ) {
8.     $dukung = MySql_fetch_array($res);
9.     if($dukung['reputasi_user'] != $rep){
10.        $qry = "UPDATE `reputasi_user` SET `reputasi_user`='".
        $rep.'" WHERE id_user_penerima=".$id_user." AND id_user_pemberi
        i = ".$id_user_pemberi;
11.        $res = MySql_query($qry) or die(MySql_error());
12.        echo 'Anda berhasil mengubah reputasi';
13.    } else {
14.        echo 'Anda sudah memberikan reputasi pada user ini';
15.    }
16.
17. } else {
18. $qry = "INSERT INTO `reputasi_user` VALUES ('',".$id_user
19.        .", ".$id_user_pemberi.",NOW(),".$rep.")";
20. $res = MySql_query($qry) or die('ERROR!!');
21. echo 'Anda berhasil menambah reputasi';
22. }

```

Gambar 5.29 Potongan kode pemberian reputasi *user*

Program akan mengecek apakah *user* yang memberikan reputasi sudah memberikan reputasi kepada *user* yang diberikan sekarang. Jika sudah, maka program akan mengecek apakah reputasi yang diberikan sama atau tidak. Jika berbeda maka program akan mengubah reputasinya sesuai dengan reputasi yang diberikan sekarang. Terakhir, jika *user* belum pernah memberikan reputasi pada *user* tersebut, maka reputasi akan langsung ditambahkan ke *database*.

5.4.7 Fungsi Grafik *Data*

Pada Halaman Dashboard ditampilkan *data-data* statistik yang ada pada aplikasi. *Data* statistik itu ditampilkan dengan visualisasi melalui grafik. Untuk menampilkan *data* tersebut menjadi sebuah grafik dibutuhkan *plugin Javascript Highchart*.

Data yang ditampilkan pada grafik bisa dimasukkan melalui sebuah tabel atau melalui *array Javascript*. Pada Bina Marga

RRS, *data* dimasukkan pada sebuah tabel. *Data* yang diambil dari *database* perlu diagregasikan terlebih dahulu. Berikut adalah contoh kode tabel untuk menampilkan grafik jumlah laporan kerusakan jalan per jenis kondisi jalan. Struktur tabelnya harus seperti pada Gambar 5.30.

```

1. $qry = 'SELECT nama_kondisi_jalan, count(*) as "jumlah_kondisi
   " FROM laporan_web inner join kondisi_jalan '
2. . 'on kondisi_jalan.kode_kondisi_jalan = laporan_web.kode_kond
   isi_jalan '
3. . 'group by laporan_web.kode_kondisi_jalan order by laporan_we
   b.kode_kondisi_jalan ';
4. $res = MySQL_query($qry);
5. echo '<table id="kondisi_jalan_tbl" hidden><thead><tr><th>Kond
   isi Jalan</th><th>Jumlah</th>';
6.
7.     echo '</tr></thead><tbody>';
8.
9. while($row = MySQL_fetch_array($res)){
10.     echo '<tr>';
11.     echo '<th>'.$row['nama_kondisi_jalan'].'</th>';
12.     echo '<td>'.$row['jumlah_kondisi'].'</td>';
13.     echo '</tr>';
14. }
15. echo '</tbody></table>';

```

Gambar 5.30 Potongan kode pembuatan tabel grafik

Setelah tabel terbuat, masukkan fungsi *Javascript* yang akan memanggil *plugin* Highchart untuk mengubah tabel tersebut. Atribut yang perlu diatur adalah *data* dan *chart*. Data menunjukkan ID tabel yang akan diubah. Chart mengatur tipe grafik yang dipakai, seperti *pie*, *bar*, dan *line*. Gambar 5.31 menunjukkan fungsi Highchart untuk menampilkan grafik.

```

1. $(function() {
2.     $('#kondisi_jalan').highcharts({
3.         data: {
4.             table: 'kondisi_jalan_tbl'
5.         },
6.         chart: {

```

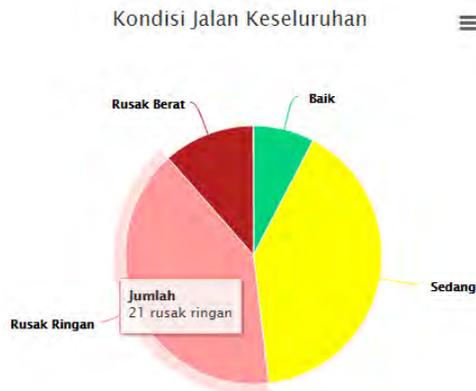
```

7.         type: 'pie'
8.     },
9.     title: {
10.        text: 'Kondisi Jalan Keseluruhan'
11.    },
12.    yAxis: {
13.        allowDecimals: false,
14.        title: {
15.            text: 'Laporan'
16.        }
17.    },
18.    tooltip: {
19.        formatter: function() {
20.            return '<b>' + this.series.name +
21.                '</b><br/>' + this.point.y + ' ' + this
22.                .point.name.toLowerCase();
23.        }
24.    }
25. });
26. });

```

Gambar 5.31 Fungsi Highchart untuk menampilkan grafik

Jika program dijalankan, maka halaman akan memuat grafik *pie* seperti pada Gambar 5.32.



Gambar 5.32 Grafik Kondisi Jalan

5.4.8 Fungsi Upload Gambar

Path direktori gambar pendukung laporan tidak disimpan di *database*. Direktori gambar pendukung laporan berada pada direktori `administration/images/{id_laporan}`. Misalnya, laporan dengan id 50, direktori gambarnya berada pada direktori `administration/images/50`.

Pertama-tama, program akan mengecek apakah ada *file* yang ingin *diupload*. Program akan mengecek apakah jumlah gambar yang sudah *diupload* dan akan *diupload* melebihi 4. Jika belum, akan dicek apakah direktori gambar laporan sudah ada atau belum. Jika belum, maka akan dibuat direktorinya. Setelah direktori ada, program akan mengecek apakah *file* yang *diupload* merupakan *file* yang valid. Pengecekan itu adalah pengecekan tipe gambar dan besar gambar. Jika memenuhi, program akan melakukan *upload* ke direktori gambar laporan dan menampilkan pesan berhasil. Berikut pada Gambar 5.33 adalah potongan kodenya.

```

1.  if(isset($_POST) and $_SERVER['REQUEST_METHOD'] == "POST"){
2.      // Loop $_FILES to execute all files
3.      foreach ($_FILES['files']['name'] as $f => $name) {
4.          if ($_FILES['files']['error'][$f] == 4) {
5.              continue; // Skip file if any error found
6.          }
7.          if ($_FILES['files']['error'][$f] == 0) {
8.
9.              if ($_FILES['files']['size'][$f] > $max_file_size)
10.             {
11.                 $message[] = "$name is too large!.";
12.                 continue; // Skip large files
13.             }
14.             elseif( ! in_array(pathinfo($name, PATHINFO_EXTENS
15.             ION), $valid_formats) ){
16.                 $message[] = "$name is not a valid format";
17.                 continue; // Skip invalid file formats
18.             }
19.             Else { // No error found! Move uploaded files

```

```

17.         if(move_uploaded_file($_FILES["files"]["tmp_na
me"][$f], $path.$name))
18.             $count++; // Number of successfully uploaded f
ile
19.         }
20.     }

```

Gambar 5.33 Potongan kode program *upload gambar*

5.5 Pengujian Aplikasi

5.5.1 *System Testing*

System Testing yaitu melakukan pengujian terhadap aplikasi secara keseluruhan terkait spesifikasi aplikasi apakah telah memenuhi spesifikasi yang telah ditentukan di awal, memastikan integrasi antar aplikasi pada *web*, yaitu *Google Maps API* dan *Gammu* serta setiap fitur yang telah dibuat dapat berfungsi dengan baik.

5.5.2 *System Performance Testing*

System Performance Testing dilakukan dengan menggunakan Apache JMeter. JMeter dapat didownload pada URL <http://JMeter.apache.org/download/JMeter.cgi>, pilih sesuai dengan Sistem Operasi yang dipakai. JMeter dapat melakukan tes terhadap kinerja *hardware* yang menjadi *server* dan keefisienan kode program. Jmeter dapat memperhatikan kecepatan koneksi antara *server* dan *client* secara *real time*. Jadi, hasil *response time* aplikasi yang ditunjukkan akan lebih besar pada praktiknya.

System performance test dilakukan dengan mensimulasikan sekelompok *user* yang melakukan permintaan ke *server* pada saat yang berdekatan. Permintaan itu dilakukan pada tiga halaman berbeda, yaitu halaman Beranda, halaman Dashboard, dan halaman Pencarian laporan. Tiga halaman tersebut akan

diuji dengan jumlah *user* yang yang berbeda-beda, yaitu 50, 100, dan 200 *user*.

Indikator utama kinerja yang dipilih adalah *response time*. *Response time* akan dihitung untuk setiap skenario *user* yang berbeda-beda dengan sebuah *software*. Ekspektasi yang diharapkan adalah dibawah 10 detik. Jika diatas 10 detik *user* biasanya akan teralihkan perhatiannya dan itu akan menghambat tujuannya mengakses Bina Marga RRS. Selain itu, waktu tunggu yang lama membuat *user* dapat membuat *user* kesal sehingga dapat memberikan kesan yang buruk pada Bina Marga RRS. Selain itu, lamanya *response time* harus stabil tidak boleh adanya peningkatan terus menerus. Hal itu menandakan *server* tidak dapat dengan cepat menyelesaikan *request* sebelum *request* lain datang.

5.5.3 *User Acceptance Testing*

Aplikasi yang dibuat tentunya tidak akan berguna jika *user* tidak menggunakannya. Untuk itu, *User Acceptance Testing* dilakukan untuk memastikan aplikasi Bina Marga RRS dapat membuat *user* untuk terus memakainya. Dalam *Technology Acceptance Model* (TAM) [14], kemauan *user* untuk memakai suatu aplikasi ditentukan oleh kegunaan dan kemudahan yang dirasakan (*perceived usefulness & perceived ease of use*) *user* ketika memakai suatu aplikasi.

User-Acceptance Testing dilakukan dengan beberapa responden yang diberikan 25 pertanyaan dan terdiri dari dua bagian, yaitu Test Scenario dan *User Acceptance Test*. Pernyataan-pernyataan kuisioner dapat dilihat pada Lampiran A. Kuisioner tersebut didasarkan pada *Technology Acceptance Model* (TAM). Bagian (Uji skenario) memberikan pertanyaan tentang kemudahan dalam penggunaan fitur-fitur yang ada pada Bina Marga RRS. Responden menjawab setiap pertanyaan dengan memilih nilai dari 1 – 5, yaitu :

- 1 : Sangat sulit dilakukan
- 2 : Sulit dilakukan
- 3 : Netral
- 4 : Mudah dilakukan
- 5 : Sangat mudah dilakukan

Bagian *Test Scenario* ada untuk membantu responden menjawab Bagian *User Acceptance Test*. Hal itu disebabkan karena pertanyaan pada *User Acceptance Test* bersifat umum dan merupakan keseluruhan tanggapan *user* dari penggunaan aplikasi, sehingga *user* harus memakai terlebih dahulu aplikasi Bina Marga RRS. Berikut pada Tabel 5.4 adalah pemetaan setiap skenario pada fitur aplikasi.

Tabel 5.4 Pemetaan *Test Scenario* pada Fitur Aplikasi

Skenario	Fitur
Q1	<i>General</i>
Q2	Registrasi akun baru
Q3	Verifikasi akun
Q4	<i>Login</i>
Q5	<i>Filtering</i> laporan pada peta
Q6	Navigasi pada peta
Q7	Pembuatan laporan baru
Q8	Reputasi laporan
Q9	Reputasi <i>user</i>
Q10	<i>Dashboard</i>
Q11	Daftar <i>user</i>
Q12	Daftar laporan
Q13	<i>Edit</i> laporan
Q14	<i>Edit profile</i>
Q15	Pelaporan melalui <i>SMS</i>

Pada bagian *User Acceptance Test* terdapat 10 pertanyaan, yaitu 5 pertanyaan untuk menilai *perceived usefulness* dan 5

pertanyaan untuk menilai *perceived ease of use*. Responden menjawab dengan memberikan nilai dari skala 1 – 5, berikut adalah arti dari nilai tersebut :

- 1 : Sangat tidak setuju
- 2 : Tidak setuju
- 3 : Netral
- 4 : Setuju
- 5 : Sangat setuju

User-Acceptance Testing diberikan juga pada Bapak Ir. Atok Nardyawanto, MMT selaku Kepala Seksi pengaturan dan pengendalian dan Ibu Krisna sebagai bagian dari bagian pengaturan dan pengendalian pada tanggal 5 Januari 2016. Terdapat penambahan 5 pernyataan pada bagian Uji Skenario yang berisi pernyataan tentang fitur-fitur yang digunakan oleh pengelola aplikasi. Berikut pada Tabel 5.5 adalah lima fitur tambahan yang diberikan.

Tabel 5.5 Fitur tambahan Uji Skenario untuk administrator

Skenario	Fitur
Q16	Ekspor <i>SMS</i>
Q17	Verifikasi laporan
Q18	Isi penanganan & penutupan laporan
Q19	Banning <i>user</i>
Q20	Hapus laporan

Hasil kuisioner ini kemudian akan ditentukan Validitas maupun Reliabilitasnya sebelum diolah dengan menghitung nilai *Cronbach's Alpha* dan *Kaiser-Meyer-Olkin* (KMO). Nilai itu dihitung dengan bantuan *software* SPSS. Masing-masing nilai memiliki batasan nilai minimum untuk digunakan. Menurut [15], data masih dikatakan reliable jika nilai nilai *Cronbach's Alpha* melebihi 0,5. Menurut [16], data masih tergolong valid jika nilai KMO lebih dari 0,5.

5.5.4 *Responsive Layout Testing*

Aplikasi Bina Marga RRS didesain untuk bisa dibuka dengan perangkat-perangkat *mobile* melalui *internet* atau *browser* masing-masing perangkat tersebut. Selain bisa dibuka, desain aplikasi dibuat agar dapat memberikan *user experience* yang baik seperti halnya ketika aplikasi dibuka didesktop PC. Pengujian akan dilakukan dengan bantuan *Developer Tools* pada Google Chrome.

Halaman ini sengaja dikosongkan

BAB VI HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil dari pengujian aplikasi serta pembasan terhadap hasil pengujian yang dilakukan. Berikut adalah penjelasan untuk hasil dan pembahasan pengujian yang dilakukan.

6.1 Hasil

Pada sub bab ini akan diuraikan tentang hasil dari pengujian aplikasi yang telah dibangun.

6.1.1 *System Testing*

System Testing menguji apakah suatu fitur pada aplikasi sudah berjalan dengan benar. Testing dilakukan semua fitur aplikasi, *tetapi* yang didokumentasikan pada buku ini hanya fitur-fitur utama aplikasi saja. Tabel 6.1 memuat daftar fitur yang diuji.

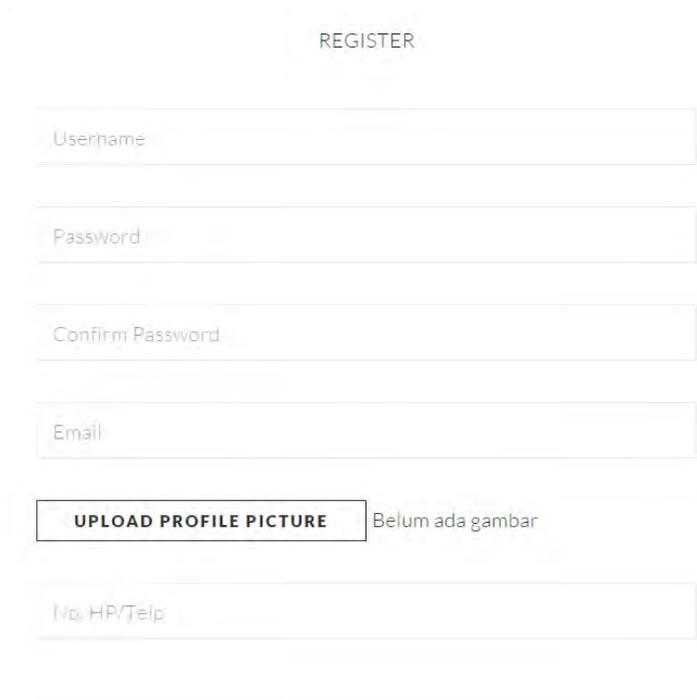
Tabel 6.1 Fitur yang diuji

No.	Fitur
1.	<i>Register</i>
2.	<i>Login</i>
3.	Lihat Peta
4.	Pelaporan melalui <i>Website</i>
5.	Pelaporan melalui <i>SMS</i>
6.	Reputasi
7.	<i>Dashboard</i>
8.	Daftar laporan dan <i>user</i>
9.	Ubah laporan

a. Register

Pengguna yang belum mendaftar hanya bisa melihat Halaman Beranda. Untuk membuat akun baru tekan tombol “Sign Up” yang berada pada Halaman Beranda. Aplikasi

akan menampilkan sebuah pop-up seperti pada Gambar 6.1 yang berisi *form* untuk melakukan registrasi.



The image shows a registration form with the following fields and elements:

- REGISTER** (Title)
-
-
-
-
- Belum ada gambar
-

Gambar 6.1 *Form register*

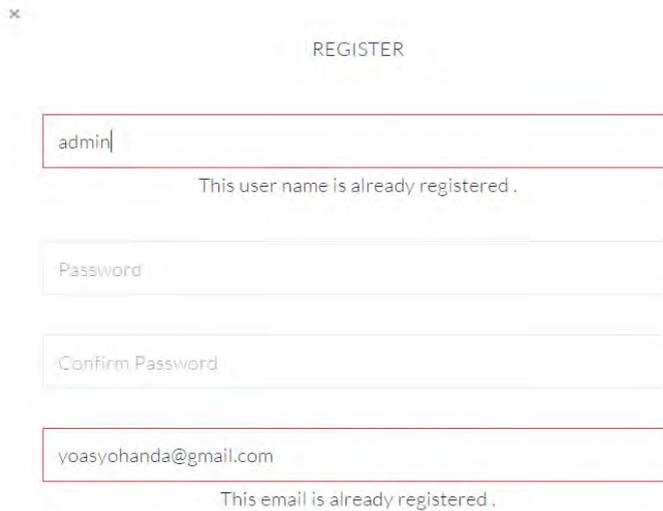
User harus mengisi semua isian, kecuali mengisi *profile picture*. Jika tidak diisi, maka *form* akan menampilkan pesan *error* seperti pada Gambar 6.2.

The image shows a registration form titled "REGISTER" with a close button (x) in the top left corner. The form contains four input fields, each with a red border and a corresponding error message below it:

- Username:** The input value is shorter than 4 characters
- Password:** Password minimal 8 karakter, kombinasi angka dan huruf
- Confirm Password:** You have not answered all required fields
- Email:** You have not given a correct e-mail address

Gambar 6.2 Pesan *error* pada *form register*

Pada *form* ini terdapat komponen beberapa validasi. Pertama validasi untuk kesamaan nama *username* dan *username* yang telah terdaftar. Jika *user* memasukkan *username* dan *email* yang telah terdaftar maka akan muncul sebuah *error* seperti pada Gambar 6.3.



The image shows a web form titled "REGISTER" with a close button (x) in the top left corner. The form contains four input fields: "Username", "Password", "Confirm Password", and "Email". The "Username" field contains the text "admin" and has a red border with the error message "This user name is already registered ." below it. The "Email" field contains the text "yoasyohanda@gmail.com" and has a red border with the error message "This email is already registered ." below it. The "Password" and "Confirm Password" fields are empty.

Gambar 6.3 Error pada isian *username* dan *email*

Pada isian *password*, *user* harus memasukkan *password* dengan panjang minimal delapan karakter dan merupakan kombinasi angka dan huruf. *Password* harus dikonfirmasi kesamaanya dengan yang telah diisikan sebelumnya. Jika kedua kondisi itu tidak terpenuhi, maka aplikasi akan memunculkan *error* seperti pada Gambar 6.4.

REGISTER

admin3

.....|

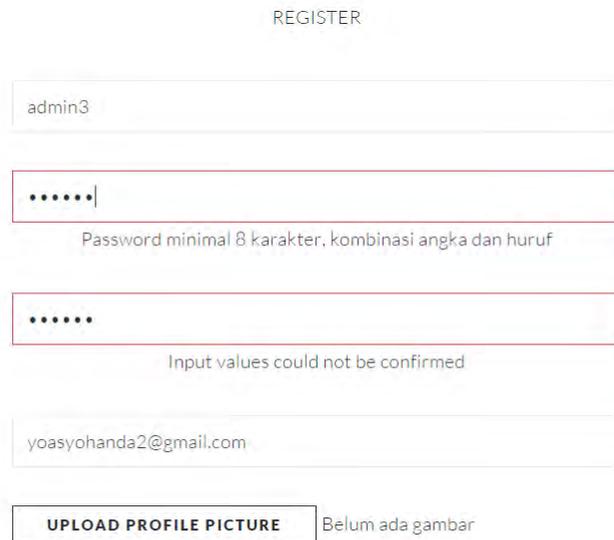
Password minimal 8 karakter, kombinasi angka dan huruf

.....

Input values could not be confirmed

yoasyohanda2@gmail.com

UPLOAD PROFILE PICTURE Belum ada gambar

A screenshot of a web registration form. At the top, the word "REGISTER" is centered. Below it are four input fields: a username field containing "admin3", a password field with a red border and a red error message below it stating "Password minimal 8 karakter, kombinasi angka dan huruf", a confirmation password field with a red border and a red error message below it stating "Input values could not be confirmed", and an email field containing "yoasyohanda2@gmail.com". At the bottom, there is a button labeled "UPLOAD PROFILE PICTURE" and a text label "Belum ada gambar".

Gambar 6.4 Error pada isian *password*

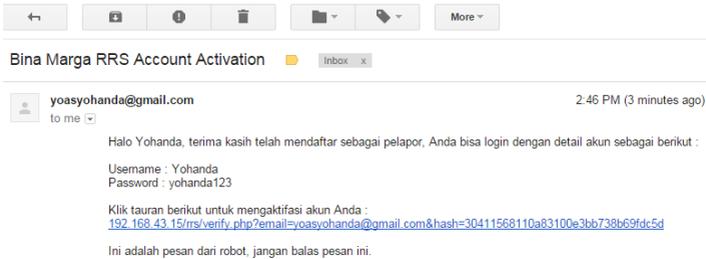
Jika *user* sudah mengisi *form* dengan benar, *user* dapat menekan submit untuk melakukan pendaftaran. Jika berhasil, sistem akan menampilkan pesan berhasil seperti pada Gambar 6.5.



Gambar 6.5 Pesan berhasil *register*

Akun yang baru dibuat belum dapat digunakan. Jika *user login* dengan akun tersebut, maka sistem akan

menampilkan pesan *error*. *User* harus melakukan verifikasi *email*. Dengan adanya pesan seperti gambar di atas sistem sudah mengirimkan *email* yang berisi *URL* untuk memverifikasi akun *user* seperti pada Gambar 6.6.



Gambar 6.6 Email verifikasi akun

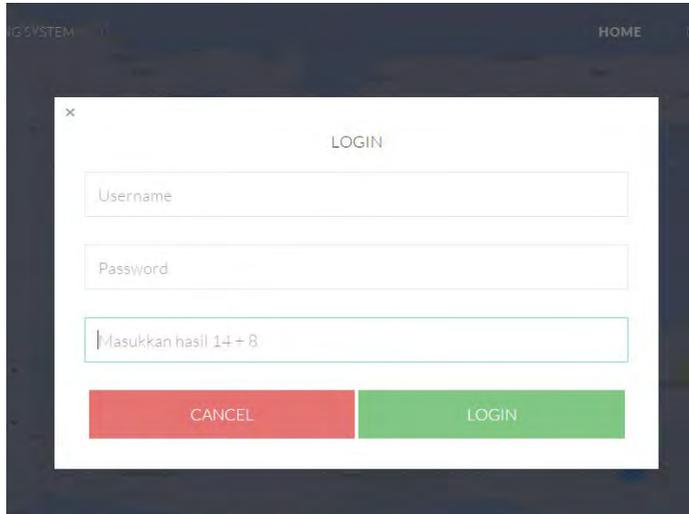
Jika *user* menekan link tersebut, maka sistem akan melakukan verifikasi akun. Jika berhasil, sistem akan menampilkan pesan seperti pada Gambar 6.7.



Gambar 6.7 Notifikasi akun berhasil diverifikasi

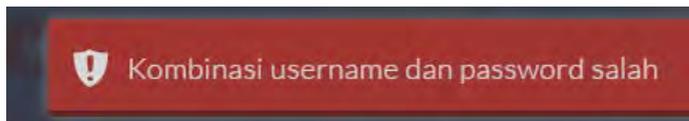
b. *Login*

Login dapat dilakukan dengan melakukan klik pada tombol *Login* di Halaman Beranda. Setelah melakukan klik, sistem akan menampilkan *pop-up* yang berisi *form* untuk melakukan *login* seperti pada Gambar 6.8.

The image shows a login form titled "LOGIN" within a dark-themed application window. The form has three input fields: "Username", "Password", and a captcha field with the text "Masukkan hasil 14 + 8". Below the fields are two buttons: a red "CANCEL" button and a green "LOGIN" button. The application window has "LOG SYSTEM" on the top left and "HOME" on the top right.

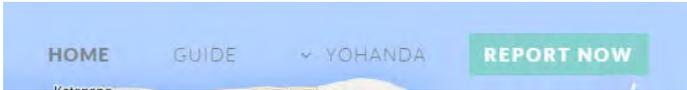
Gambar 6.8 Form Login

Jika *user* memasukkan *username* dan *password* yang tidak sesuai, sistem akan menampilkan pesan *error* seperti pada Gambar 6.9.



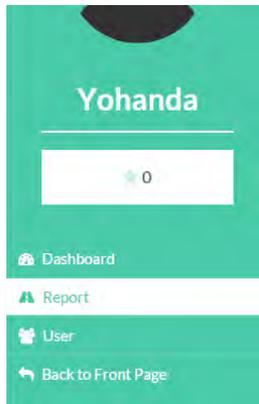
Gambar 6.9 Pesan error kombinasi *username* dan *password* tidak sesuai

Jika *user* memasukkan *username*, *password*, dan hasil *captcha* yang tepat, maka sistem akan mengatur *session user* dan melakukan *redirect* halaman ke Halaman Beranda dengan nama *user* yang sudah tercatat pada menu bar seperti pada Gambar 6.10.



Gambar 6.10 Username pada menubar

Fitur *login* diperlukan untuk membedakan hak akses. *User* yang memiliki hak akses yang terbatas sebagai pelapor tidak dapat melihat Halaman *SMS Gateway*. Contohnya, pada *user* dengan *username* “Yohanda” yang memiliki hak akses terbatas. Pada bagian *back-end* aplikasi *user* ini tidak memiliki pilihan untuk memasuki Halaman *SMS Gateway* seperti yang ditunjukkan pada Gambar 6.11.



Gambar 6.11 Sidebar user dengan hak akses normal

Jika *user* memaksa untuk masuk pada Halaman *SMS Gateway* dengan menuliskan *URL* pada *browser*, maka sistem akan menolak aksesnya dan menampilkan Halaman *error* seperti pada Gambar 6.12.



Gambar 6.12 Halaman error menolak akses halaman

User biasa tidak bisa menghapus ataupun mengubah laporan *user* lain. Selain itu, *user* biasa juga tidak bisa melakukan memverifikasi laporan. Hal ini ditunjukkan pada Gambar 6.13.



Gambar 6.13 Halaman Detil Laporan diakses oleh *user* biasa

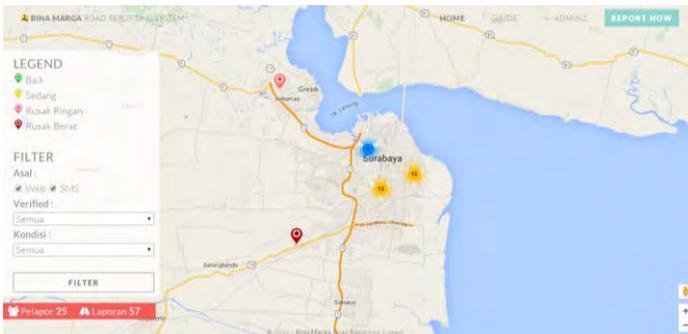
c. Lihat Peta

Halaman Beranda menampilkan peta berisi *marker-marker* yang menandakan laporan kerusakan jalan. *Marker-marker* tersebut memiliki warna yang berbeda sesuai dengan kondisi kerusakan jalan. Kumpulan *marker-marker* yang berdekatan akan membentuk sebuah *marker* cluster yang memiliki simbol bulat dengan warna yang menunjukkan banyaknya *marker* yang berdekatan pada daerah tersebut. Tingkatan banyaknya *marker* mulai dari yang sedikit sampai yang paling banyak berdasarkan warnanya adalah biru, kuning, dan merah. Gambar 6.14 menunjukkan tampilan Halaman Beranda.



Gambar 6.14 Tampilan Halaman Beranda

Jika *marker cluster* diklik maka peta akan melakukan perbesaran skala dan memperlihatkan *marker-marker* yang ada didalamnya seperti pada Gambar 6.15. Jika skala peta masih terlalu besar, maka akan terbentuk *marker cluster* yang lebih kecil lagi seperti pada Gambar 6.15.



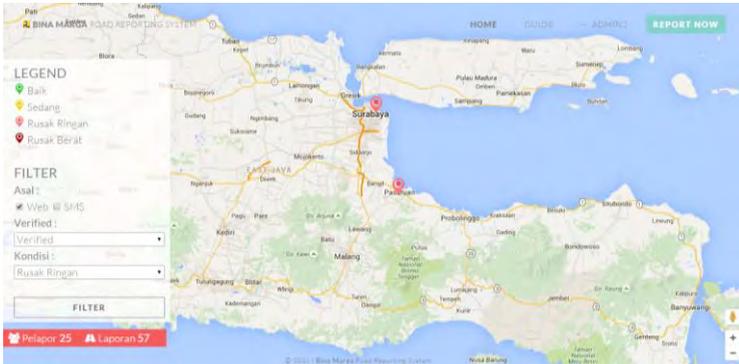
Gambar 6.15 Marker Cluster yang dinamis

Jika *marker* diklik, maka akan muncul sebuah *panel* yang menampilkan sebagian informasi laporan seperti pada Gambar 6.16.



Gambar 6.16 Informasi laporan pada peta

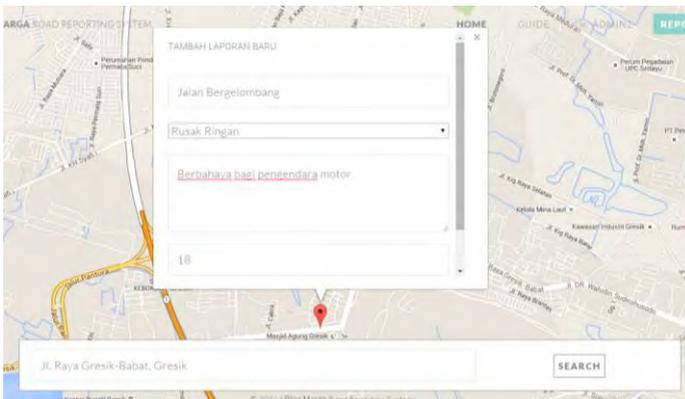
Marker yang tampil pada peta dapat disaring sesuai kebutuhan. Terdapat tiga komponen penyaringan, yaitu asal laporan, status verifikasi, dan kondisi kerusakan jalan. Misalnya, pengguna ingin mengetahui laporan dari *web* yang sudah terverifikasi dan kondisinya rusak ringan. Setelah melakukan klik pada tombol *filter* maka program akan menyaring *marker* yang memenuhi kriteria. Berikut tampilannya pada Gambar 6.17.



Gambar 6.17 Filter laporan pada peta

d. Pelaporan melalui *web*

Untuk membuat laporan baru, dari Halaman Beranda klik tombol “*Report Now*”, maka akan muncul sebuah *form* seperti pada Gambar 6.18, isi *form* tersebut dan arahkan *marker* ke tempat yang menjadi titik kerusakan jalan. Pelapor harus menentukan titik kerusakan dengan memasukkan alamat, melakukan klik pada peta, ataupun menggeser *marker*.



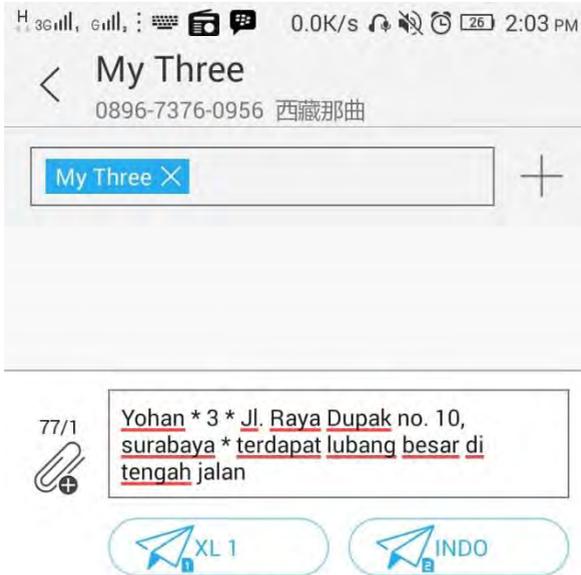
Gambar 6.18 Halaman Laporan Baru

Setelah mengisi *form* laporan, pelapor harus melakukan klik pada tombol Save. Jika berhasil maka akan muncul sebuah notifikasi dan pelapor akan *redirect* ke halaman Detil Laporan. Pada halaman ini terdapat informasi-informasi lengkap tentang laporan. Pelapor bisa menambahkan gambar ataupun menambahkan informasi lainnya. Jika admin UPT atau admin Bina Marga yang membuka halaman ini, mereka berhak penuh atas setiap laporan. Mereka bisa melakukan verifikasi, mengubah informasi laporan, bahkan menghapusnya seperti pada Gambar 6.19.



Gambar 6.19 Halaman Detil Laporan diakses oleh admin

- e. Pelaporan melalui *SMS*
 Kirim *SMS* sesuai dengan *format* yang telah ditentukan ke nomor yang telah ditentukan. Nomor telepon bisa diganti kapanpun. Berikut pada Gambar 6.20 adalah contoh *SMS* yang valid.



Gambar 6.20 Pengiriman SMS melalui telepon genggam

Setelah *SMS* dikirimkan, *service* Gammu yang aktif akan memasukkan *SMS* tersebut ke dalam tabel *inbox* aplikasi seperti pada Gambar 6.21.

Inbox

Filter

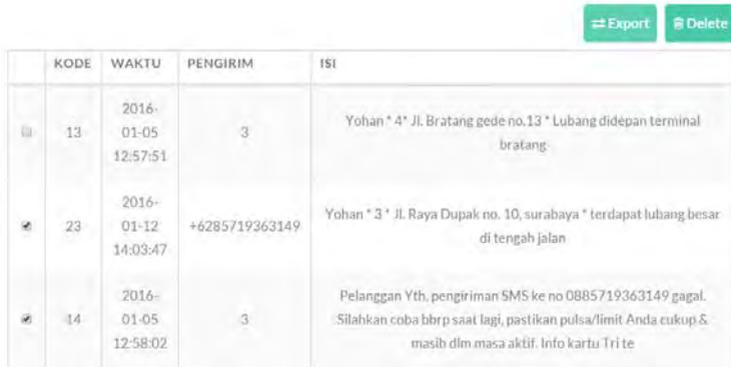
Sort

Export Delete

KODE	WAKTU	PENKIRIM	ISI
13	2016-01-05 12:57:51	3	Yohan * 4 * Jl. Bratang gede no.13 * Lubang didepan terminal bratang
	2016-		

Gambar 6.21 SMS yang baru dikirim masuk ke dalam Halaman Inbox

Laporan melalui *SMS* tidak langsung tampil pada *map*. Laporan tersebut perlu diekspor. Pilih *SMS* yang ingin diekspor dengan mencentang *checkbox* yang ada pada setiap baris laporan, lalu klik ekspor seperti pada Gambar 6.22.



	KODE	WAKTU	PENGRIM	ISI
<input type="checkbox"/>	13	2016-01-05 12:57:51	3	Yohan * 4* Jl. Bratang gede no.13 * Lubang didepan terminal bratang
<input checked="" type="checkbox"/>	23	2016-01-12 14:03:47	+6285719363149	Yohan * 3 * Jl. Raya Dupak no. 10, surabaya * terdapat lubang besar di tengah jalan
<input checked="" type="checkbox"/>	14	2016-01-05 12:58:02	3	Pelanggan Yth. pengiriman SMS ke no 0885719363149 gagal. Silahkan coba bbrp saat lagi, pastikan pulsa/limit Anda cukup & masih dim masa aktif. Info kartu Tri te

Gambar 6.22 Memilih SMS yang ingin diekspor

Proses ekspor akan berjalan dan jika selesai akan muncul notifikasi. Notifikasi ini juga akan memberikan informasi mengenai laporan yang tidak dapat diekspor seperti pada Gambar 6.23. Laporan yang tidak berhasil diekspor akan tetap berada di *inbox*, sedangkan laporan yang berhasil akan diubah menjadi seperti laporan yang dibuat pada *Website* dan dimasukkan dalam tabel archive.



Gambar 6.23 Informasi SMS yang tidak dapat diekspor

Jika tidak adanya kesalahan maka akan muncul *pop-up* seperti pada Gambar 6.24.



Gambar 6.24 Notifikasi semua SMS berhasil diekspor

Laporan sudah masuk dan bisa dilihat dengan detail seperti pada Gambar 6.25.



Gambar 6.25 Detil Laporan SMS yang diekspor

f. Reputasi

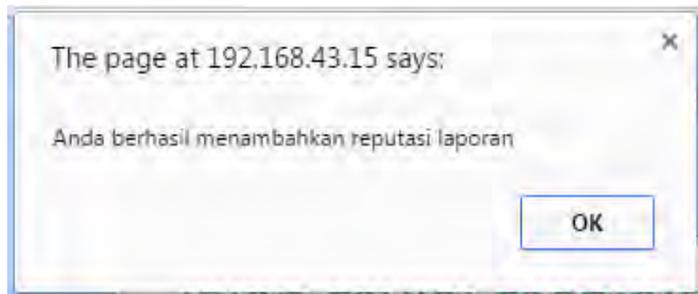
Terdapat dua reputasi yang bisa diberikan, yaitu reputasi laporan dan reputasi *user*. Reputasi laporan bisa diberikan pada Halaman Beranda ataupun melalui Halaman Detil

Laporan dengan melakukan klik pada tombol “agree” atau “disagree” seperti pada Gambar 6.26.



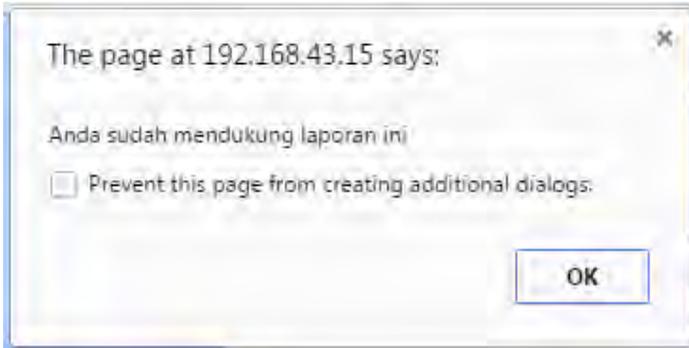
Gambar 6.26 Reputasi laporan

Jika berhasil memberikan reputasi, maka sistem akan menampilkan pesan berhasil seperti pada gambar di bawah.



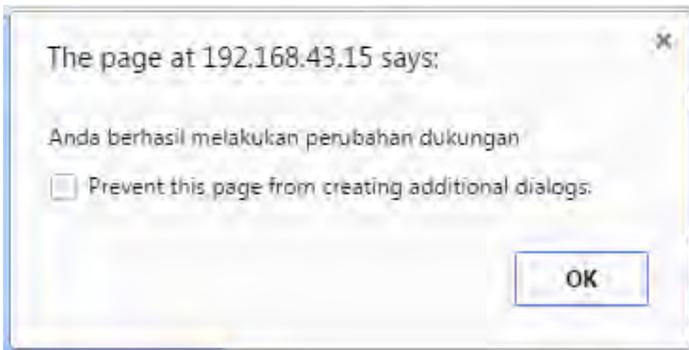
Gambar 6.27 Pesan berhasil menambahkan reputasi laporan

User hanya bisa memberikan satu reputasi pada satu laporan. Jika *user* menekan tombol “Agree” lagi pada laporan yang sama, maka sistem tidak akan menambah reputasi laporan. Sebuah pesan kegagalan akan muncul seperti pada Gambar 6.28.



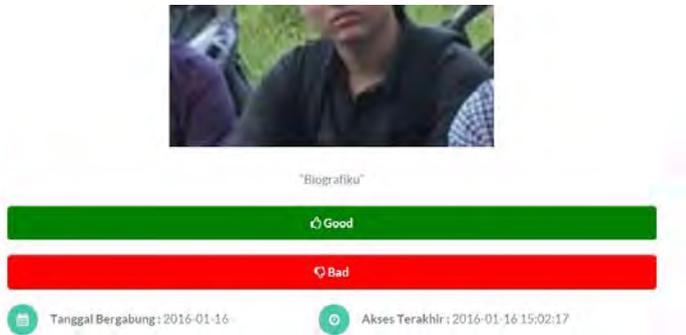
Gambar 6.28 Pesan gagal menambahkan reputasi laporan

Jika *user* menekan tombol “Disagree”, sistem akan mengubah reputasi laporan dan menampilkan pesan seperti pada Gambar 6.29.



Gambar 6.29 Pesan berhasil mengubah reputasi laporan

Pemberian reputasi untuk *user* mempunyai logika yang sama dengan reputasi laporan. *User* bisa memberikan reputasi *user* pada halaman profil *user* lain dengan melakukan klik pada tombol “Good” atau “Bad” seperti pada Gambar 6.30.



Gambar 6.30 Menambah reputasi user

- g. Dashboard
 Dashboard sudah menampilkan visualisasi dan informasi penting dengan baik. Berikut pada Gambar 6.31 adalah potongan grafik pada *dashboard*.



Gambar 6.31 Grafik pada Halaman Dashboard

h. Daftar laporan dan *user*

Daftar laporan dan daftar laporan *user* memiliki cara kerja yang sama. Kedua halaman tersebut memuat tabel berisi *data* yang ada di *database* beserta komponen untuk penyaringan dan pengurutan *data*. Berikut adalah tampilan Halaman Daftar Laporan pada Gambar 6.32.

Report

▼ Filter

▼ Sort

	KODE LAPORAN	JUDUL	PELAPOR	WAKTU LAPORAN	VERIFIKASI	LOKASI	REPUTASI
☐	138	Jalan retak	admin2	2016-01-05 23:40:25	☑		0
☐	136	jalan berlubang	admin2	2016-01-05 14:42:36	☑		0
☐	131	Jalan bergelombang	admin2	2016-01-03	☑		-1

Gambar 6.32 Halaman Daftar Laporan

Misalnya, dicari laporan dengan judul yang mengandung kata “jalan” yang berada di Kota Surabaya seperti pada Gambar 6.33.

Report

▲ Filter

Judul :

Pelapor :

Kota :

Gambar 6.33 Filter daftar laporan

Sistem akan menampilkan laporan-laporan dengan kriteria seperti yang telah ditentukan seperti pada Gambar 6.34. Jika laporan hasil penyaringan ada banyak akan dibuat halaman-halaman. Fungsi tersebut ketika dicoba sudah berjalan dengan baik.

Report

▼ Filter

▼ Sort

	KODE LAPORAN	JUDUL	PELAPOR	WAKTU LAPORAN	VERIFIKASI	LOKASI	REPUTASI
<input type="checkbox"/>	133	Jalan rusak karena banjir	admin2	2016-01-03 18:58:57			1
<input type="checkbox"/>	155	jalan lubang lubang	abas060	2016-01-08 10:26:40			1
		jalan		2016-01-			

Gambar 6.34 Halaman Daftar Laporan yang difilter

i. Ubah Laporan

Form untuk mengubah informasi pada suatu laporan terdapat pada Halaman Detil Laporan. Untuk mengubahnya, klik tombol “Edit Report”. Sistem akan menampilkan *form* seperti pada Gambar 6.35.



The image shows a web form titled "Edit Laporan" in green text. Below the title are four input fields, each with a label and a text box containing a value:

- Judul :** Jalan mulai meripis
- Status laporan :** Open
- Status verifikasi :** Not Verified
- Penanganan :** Belum ada

Gambar 6.35 Halaman Edit Laporan

Semua informasi diubah untuk mengetahui apakah semuanya telah berjalan dengan benar. Setelah diubah, klik tombol “Save Changes”, lalu sistem akan menampilkan notifikasi bahwa perubahan berhasil dan meredirect *user* pada Halaman Detil Laporan. Pada Gambar 6.36 dapat dilihat bahwa semua informasi telah berubah.



Gambar 6.36 Informasi laporan telah berhasil diubah

Pelapor dapat menambahkan sampai empat gambar pendukung laporan. Pelapor dapat menambahnya dengan melakukan klik pada tombol “Add Images” pada Halaman Detil Laporan. Sistem akan menampilkan Halaman *Upload Images* seperti pada Gambar 6.37.



Gambar 6.37 Halaman *Upload Gambar*

Pelapor dapat melakukan *upload* dengan memilih *file* pada *form* input yang berada di bawah, lalu melakukan klik pada tombol “*Upload*”. Jika Pelapor melakukan *upload* 4 gambar maka akan muncul notifikasi gagal. Hal itu dikarenakan pelapor sudah menambahkan satu gambar sehingga pelapor hanya bisa menambahkan 3 gambar lagi. Jika pelapor menambahkan tiga gambar dan berhasil, maka gambar itu akan muncul pada Halaman Detil Laporan seperti pada Gambar 6.38.



Gambar 6.38 Gambar berhasil ditambahkan

6.1.2 *System Performance Test*

System Performance Test dimulai dengan membuat sebuah *test plan*. *Test plan* ini memuat komponen-komponen yang digunakan untuk melakukan test. Berikut pada Gambar 6.39 adalah contoh sebuah test plan.

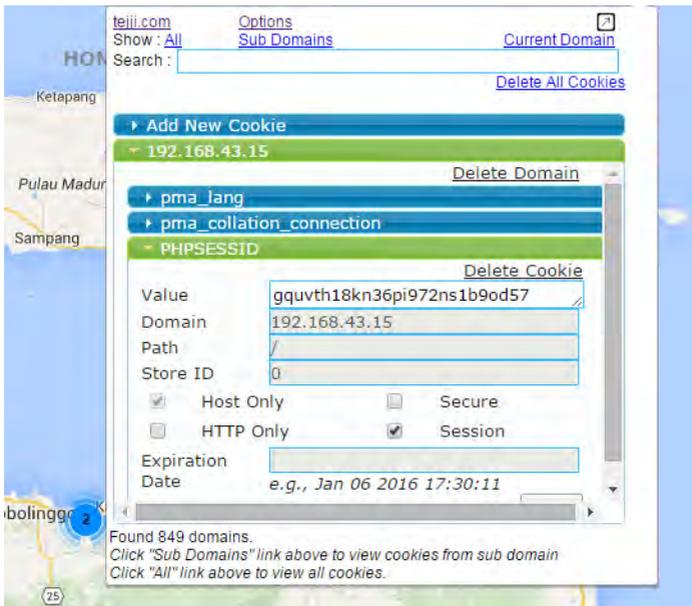
Gambar 6.39 Contoh Test Plan

JMeter mendukung pengujian untuk halaman yang menggunakan *login* untuk mengaksesnya. Terdapat dua halaman yang memerlukan *user* untuk *login* yaitu Halaman *Dashboard* dan Pencarian Laporan. Untuk itu, buat sebuah *Cookie Manager* dan isi nama variable beserta nilainya, seperti pada Gambar 6.40.

Name	Value	Domain
PHPSESSID	2g3nvbanmet9g449qutb83140	192.168.43.15

Gambar 6.40 Pengaturan Cookie Manager

Aplikasi RRS Bina Marga memakai variable `$_SESSION PHP` untuk menampung *data user* yang melakukan *login*. `$_SESSION` tersebut akan menjadi sebuah *cookie* yang tersimpan pada komputer *client*. Informasi cookie dan valuenya yang dimasukkan pada *Cookie Manager* di JMeter didapatkan dengan memakai *add-on Cookie Manager* dari browser Google Chrome. Berikut adalah tampilan *add-on Cookie Manage* pada Gambar 6.41.



Gambar 6.41 Add-on Cookie Manager pada Google Chrome

Pengujian ini akan menguji tiga halaman, pertama adalah Halaman Lihat Peta yang menampilkan peta beserta *marker-marker* dan informasi kerusakan jalan di dalamnya. Pada saat pengujian, terdapat 50 laporan yang tercatat pada *database*. Banyaknya laporan akan mempengaruhi *response time server*. Berikut adalah tampilan HTTP Request Halaman Lihat Peta pada Gambar 6.42.

HTTP Request

Name:

Comments:

Web Server

Server Name or IP: Port Num

HTTP Request

Implementation: Protocol [http]: Method:

Path:

Redirect Automatically Follow Redirects Use KeepAlive Use m

Parameters Body Data

Send Parame

Gambar 6.42 HTTP Request Halaman Lihat Peta

Halaman kedua adalah Halaman *Dashboard*. *Dashboard* memuat informasi-informasi penting dan sebagian besar disajikan dalam bentuk grafik. Berikut adalah pengaturan HTTP Request Halaman *Dashboard* pada Gambar 6.43.

HTTP Request

Name:

Comments:

Web Server

Server Name or IP:

HTTP Request

Implementation: Protocol [http]:

Path:

Redirect Automatically Follow Redirects Use KeepAlive

Gambar 6.43 HTTP Request Halaman *Dashboard*

Halaman yang ketiga adalah Halaman Daftar Laporan. Halaman Daftar Laporan memuat daftar-daftar laporan kerusakan jalan. Pada halaman tersebut terdapat *filter-filter* yang dapat membantu untuk menyaring laporan yang diinginkan. Pencarian yang dilakukan adalah mencari laporan yang berisikan “jalan”, berikut adalah pengaturan HTTP Request Search Laporan pada Gambar 6.44.

Send Parameters With the Request:	
Name:	Value
judul	jalan
kota	all

Gambar 6.44 HTTP Request Halaman *Search Laporan*

Hasil pengujian ditampilkan didalam sebuah *Listener* di dalam JMeter. Dalam pengujian ini, terdapat tiga *listener* yang dipakai yaitu *View Results in Table*, *Aggregate Graph*, dan *Response Time Graph*. Tambahkan ketiganya dengan melakukan klik kanan pada thread group, lalu klik listener dan pilih jenisnya.

Hasil Skenario 50 User

Skenario 50 *user* dilakukan dengan menjalankan tiga halaman yang telah dibuat dengan 50 *user*. Setiap halaman akan dieksekusi satu kali untuk setiap *user*. Jadi, nanti akan terdapat 150 *informasi*. Waktu pengujian adalah selama 10 detik, artinya dalam 10 detik 150 *request* tersebut harus dieksekusi. Berikut pada Gambar 6.45 adalah pengaturan *Thread Group* sesuai dengan skenario 50 *user*.

Thread Group

Name: Thread Group

Comments:

Action to be taken after a Sampler error

Continue

Thread Properties

Number of Threads (users): 50

Ramp-Up Period (in seconds): 10

Loop Count: Forever 1

Delay Thread creation until needed

Gambar 6.45 Pengaturan *Thread Group* Skenario 50 user

Klik start untuk memulai pengujian dan tunggu sampai pengujian selesai. Setiap *informasi* yang dikirimkan dapat dilihat melalui *listener View Results in Table*. Semua *informasi* mempunyai status dengan tanda centang hijau menandakan *request* telah berjalan dengan benar.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status
1	15:46:54.712	Thread Group 1-1	Dashboard	170	
2	15:46:54.885	Thread Group 1-1	Lihat Peta	50	
3	15:46:54.912	Thread Group 1-2	Dashboard	73	
4	15:46:54.938	Thread Group 1-1	Search Laporan	63	
5	15:46:54.987	Thread Group 1-2	Lihat Peta	56	
6	15:46:55.045	Thread Group 1-2	Search Laporan	15	
7	15:46:55.117	Thread Group 1-3	Dashboard	87	
8	15:46:55.206	Thread Group 1-3	Lihat Peta	44	
9	15:46:55.252	Thread Group 1-3	Search Laporan	14	
10	15:46:55.314	Thread Group 1-4	Dashboard	58	
11	15:46:55.374	Thread Group 1-4	Lihat Peta	34	
12	15:46:55.409	Thread Group 1-4	Search Laporan	16	
13	15:46:55.521	Thread Group 1-5	Dashboard	57	
14	15:46:55.579	Thread Group 1-5	Lihat Peta	31	
15	15:46:55.611	Thread Group 1-5	Search Laporan	15	
16	15:46:55.723	Thread Group 1-6	Dashboard	103	
17	15:46:55.827	Thread Group 1-6	Lihat Peta	42	
18	15:46:55.871	Thread Group 1-6	Search Laporan	24	
19	15:46:55.913	Thread Group 1-7	Dashboard	60	
20	15:46:55.975	Thread Group 1-7	Lihat Peta	36	
21	15:46:56.013	Thread Group 1-7	Search Laporan	24	
22	15:46:56.121	Thread Group 1-8	Dashboard	94	

Gambar 6.46 *Listener View Request in Table* skenario 50 user

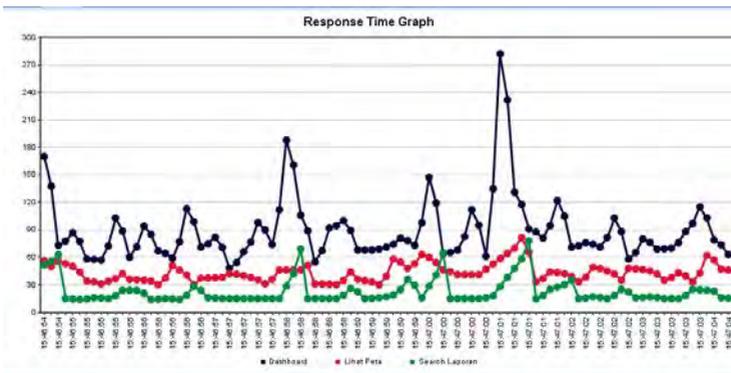
Berikut adalah tabel, berisi *data* yang berada pada *Listener Aggregate Graph*. Tabel di bawah menunjukkan *response time server* masih cepat, yaitu 0,51 detik (100 ms = 1 detik).

Dashboard menjadi halaman yang paling lambat kecepatan aksesnya, tetapi secara umum tidak berbeda jauh. Tabel dibawah menunjukkan tidak ada *error* yang terjadi saat pengujian.

Tabel 6.2 Tabel *Aggregate Graph* skenario 50 user

Label	# Samples	Average	Min	Max	Error %	Throughput
Dashboard	50	88 ms	48	282	0,00%	5,1
Lihat Peta	50	43	30	93	0,00%	5,1
Search Laporan	50	23	14	96	0,00%	5,1
TOTAL	150	51	14	282	0,00%	15,1

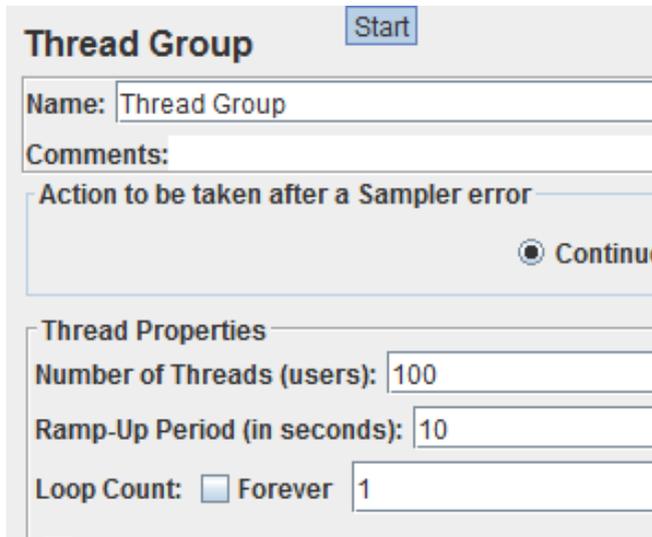
Pada Gambar 6.47 terdapat *Response Time Graph* yang menunjukkan tidak adanya peningkatan *response time* secara terus menerus. Grafik menunjukkan *response time* pada setiap *informasi* cenderung stabil. Hal ini menunjukkan kinerja *server* masih baik dan dapat menangani permintaan yang masuk dengan optimal.



Gambar 6.47 *Response Time Graph* skenario 50 user

Hasil Skenario 100 User

Skenario 100 *user* dilakukan dengan mengubah nilai *Number of Threads* pada pengaturan *Thread Group*. Berikut pada Gambar 6.48 adalah pengaturan untuk skenario ini.



Thread Group

Name:

Comments:

Action to be taken after a Sampler error: Continue

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

Loop Count: Forever

Gambar 6.48 Pengaturan *Thread Group* skenario 100 *user*

Gambar 6.49 menunjukkan hasil pada *Listener View Results in Table*. Dari tabel tersebut, dapat dilihat adanya peningkatan jumlah *informasi*. Status *informasi* masih menunjukkan hasil yang baik dengan adanya tanda centang hijau.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status
1	15:56:15.169	Thread Group 1-1	Dashboard	246	
2	15:56:15.278	Thread Group 1-2	Dashboard	194	
3	15:56:15.385	Thread Group 1-3	Dashboard	153	
4	15:56:15.417	Thread Group 1-1	Lihat Peta	163	
5	15:56:15.479	Thread Group 1-2	Lihat Peta	132	
6	15:56:15.484	Thread Group 1-4	Dashboard	173	
7	15:56:15.539	Thread Group 1-3	Lihat Peta	151	
8	15:56:15.582	Thread Group 1-1	Search Laporan	120	

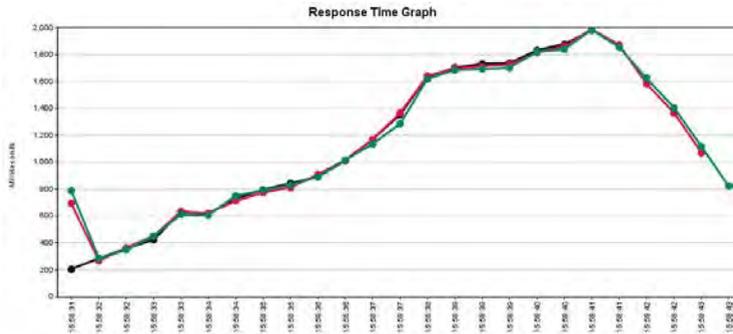
Gambar 6.49 *Listener View Results in Table* skenario 100 *user*

Tabel 6.3 menunjukkan hasil *Listener Aggregation Graph*. Terjadi peningkatan kurang lebih sebesar 7 kali dari nilai *response time* pada skenario 50 *user*. Hal ini menunjukkan bahwa untuk mendapatkan permintaan *data* seorang *client* rata-rata minimal harus menunggu 7,31 detik. Kecepatan akses tersebut sudah masuk ke dalam kategori yang lambat. Karena, pengujian hanya dilakukan untuk HTTP *informasi*. Selain melakukan permintaan ke *server* Bina Marga RRS, *client* juga akan melakukan permintaan *data* ke *server* Google Maps.

Tabel 6.3 Tabel Aggregation Graph skenario 100 user

Label	# Samples	Average	Min	Max	Error %	Throughput
Dashboard	100	723	153	1383	0,00%	8,8
Lihat Peta	100	753	132	1373	0,00%	8,5
Search Laporan	100	716	120	1353	0,00%	8,5
TOTAL	300	731	120	1383	0,00%	24,6

Gambar 6.50 menunjukkan *Response Time Graph* skenario 100 *user*. Pada grafik tersebut dapat dilihat adanya peningkatan *response time*. Hal ini disebabkan karena adanya permintaan yang secara terus menerus masuk, sedangkan *server* masih berusaha untuk memenuhi permintaan sebelumnya. Grafik seperti ini menunjukkan bahwa *server* sudah tidak mampu menangani permintaan yang ada. Meskipun, semua permintaan dapat dipenuhi sistem menjadi tidak responsif.



Gambar 6.50 Response Time Graph skenario 100 user

Hasil Skenario 200 User

Skenario 200 user dilakukan dengan mengubah nilai *Number of Threads* pada pengaturan *Thread Group*. Berikut pada Gambar 6.51 adalah pengaturan untuk skenario ini.

Thread Group

Name:

Comments:

Action to be taken after a Sampler error

Continue

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

Loop Count: Forever

Delay Thread creation until needed

Scheduler

Gambar 6.51 Pengaturan Thread Group skenario 200 user

Gambar 6.52 menunjukkan hasil pada *Listener View Results in Table*. Dari tabel tersebut, dapat dilihat adanya peningkatan jumlah *informasi*. Status *informasi* masih menunjukkan hasil yang baik dengan adanya tanda centang hijau.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status
1	15:59:28.226	Thread Group 1-1	Dashboard	89	
2	15:59:28.274	Thread Group 1-2	Dashboard	110	
3	15:59:28.317	Thread Group 1-1	Lihat Peta	115	
4	15:59:28.326	Thread Group 1-3	Dashboard	159	
5	15:59:28.374	Thread Group 1-4	Dashboard	177	
6	15:59:28.386	Thread Group 1-2	Lihat Peta	203	
7	15:59:28.427	Thread Group 1-5	Dashboard	214	
8	15:59:28.435	Thread Group 1-1	Search Laporan	226	
9	15:59:28.478	Thread Group 1-6	Dashboard	237	
10	15:59:28.487	Thread Group 1-3	Lihat Peta	259	
11	15:59:28.529	Thread Group 1-7	Dashboard	264	
12	15:59:28.553	Thread Group 1-4	Lihat Peta	269	
13	15:59:28.578	Thread Group 1-8	Dashboard	298	
14	15:59:28.591	Thread Group 1-2	Search Laporan	303	
15	15:59:28.635	Thread Group 1-9	Dashboard	325	

Gambar 6.52 *Listener View Results in Table* skenario 200 user

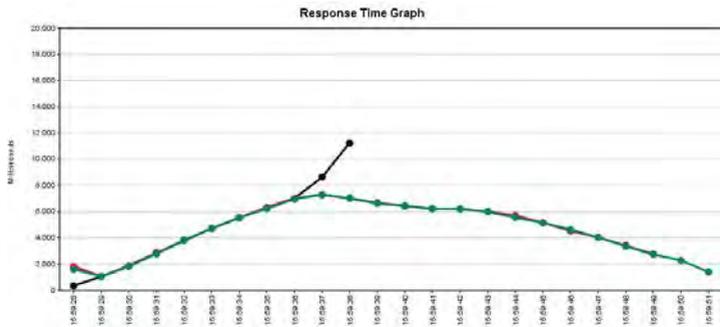
Tabel 6.4 yang menunjukkan hasil *Listener Aggregation Graph*. Terjadi peningkatan kurang lebih sebesar 6-7 kali dari nilai *response time* pada skenario 1000 user. Hal ini menunjukkan bahwa untuk mendapatkan permintaan *data* seorang *client* rata-rata minimal harus menunggu 45,27 detik. Kecepatan akses tersebut sudah masuk ke dalam kategori yang lambat. Karena, pengujian hanya dilakukan untuk HTTP *informasi*. Selain melakukan permintaan ke *server* Bina Marga RRS, *client* juga akan melakukan permintaan *data* ke *server* Google Maps.

Tabel 6.4 *Listener Aggregation Graph* skenario 200 user

Label	# Samples	Average	Min	Max	Error %	Throughput
Dashboard	200	4450	89	11533	0,00%	9,3
Lihat Peta	200	4936	115	7425	0,00%	8,3
Search Laporan	200	4195	226	7401	0,00%	8,1
TOTAL	600	4527	89	11533	0,00%	24,1

Gambar 6.53 menunjukkan *Response Time Graph* skenario 200 user. Pada grafik tersebut dapat dilihat adanya peningkatan

response time. Jika diperhatikan terdapat *informasi* yang selesai mendahului *informasi* yang lain. Beban kerja yang berat membuat *informasi* tidak dengan stabil ditangani.



Gambar 6.53 *Response Time Graph* skenario 200 user

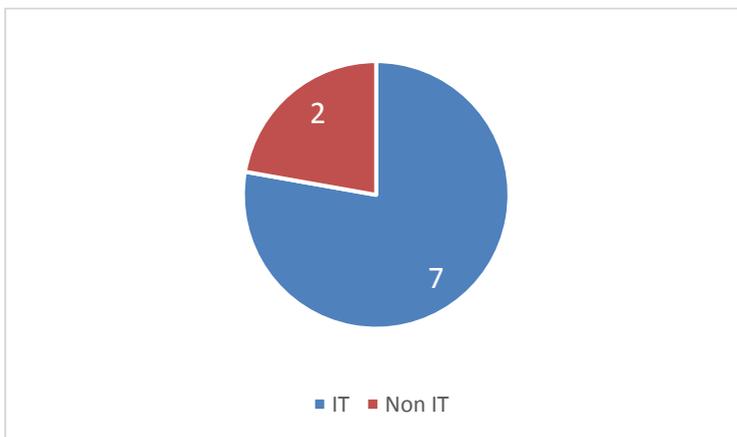
6.1.3 *User Acceptance Testing*

User Acceptance Testing dilakukan pada dua tipe peran yang ada pada Bina Marga RRS, yaitu sebagai pelapor dan penanggap (pihak Bina Marga). Responden yang memberikan tanggapan sebagai pelapor pada *User Acceptance Testing* adalah 10 mahasiswa. Responden terdiri dari dua perempuan dan 8 laki-laki. Grafik *pie* mengenai jenis kelamin ditampilkan pada Gambar 6.54.



Gambar 6.54 Profil Jenis kelamin responden

Terdapat tiga mahasiswa yang berasal dari jurusan selain Teknologi Informasi dan sisanya berasal dari Jurusan Sistem Informasi Pengguna dari Bina Marga RRS diharapkan sudah terbiasa mengoperasikan komputer. Grafik *pie* mengenai asal jurusan ditampilkan pada Gambar 6.55.



Gambar 6.55 Profil jurusan responden

Rekapitulasi jawaban responden terkait *Test Scenario* ini dapat dilihat pada Lampiran B. Tabel 6.5 menampilkan ringkasan nilai rata-rata dari setiap fitur berdasarkan *Test Scenario* yang dilakukan.

Tabel 6.5 Rata-rata nilai *Test Scenario*

Skenario	Fitur	Rata-rata
Q1	<i>General</i>	4.3
Q2	Registrasi akun baru	4.4
Q3	Verifikasi akun	4.6
Q4	<i>Login</i>	4.4
Q5	<i>Filtering</i> laporan pada peta	4.3
Q6	Navigasi pada peta	4.2
Q7	Pembuatan laporan baru	4.2
Q8	Reputasi laporan	4.6
Q9	Reputasi <i>user</i>	4.4
Q10	<i>Dashboard</i>	4.3
Q11	Daftar <i>user</i>	4.2
Q12	Daftar laporan	3.9
Q13	<i>Edit</i> laporan	4.6
Q14	<i>Edit</i> profil	4.6
Q15	Pelaporan melalui <i>SMS</i>	3.5
<u>Rata-rata keseluruhan</u>		4.3

Sebelum melakukan penarikan kesimpulan secara umum seberapa baik aplikasi dilihat dari dua dimensi TAM yaitu *perceived usefulness* dan *perceived ease of user*, perlu dilakukan analisis reliabilitas dan validitas. Berikut pada tabel dibawah adalah hasil uji reliabilitas dan validitas menggunakan *software SPSS*.

Tabel 6.6 Pengujian Realibilitas dan Validitas pada pelapor

	Reliabilitas	Validitas
	(Cronbach's Alpha > 0.5)	(Kaiser-Meyer-Olkin > 0.5)
Usefulness	0.66	0.523
Ease of Use	0.707	0.536

Rekapitulasi jawaban responden terkait *perceived usefulness* dan *perceived ease of use* ini dapat dilihat pada Lampiran C. Penarikan kesimpulan nilai *perceived usefulness* dan *perceived ease of use* dilakukan dengan menghitung hasil rata-rata nilai dari pertanyaan seperti pada Tabel 6.7.

Tabel 6.7 Rata-rata bagian kedua kuisioner pelapor

Pernyataan	Rata-rata
Usefulness	
Q1	4.3
Q2	4.9
Q3	4.4
Q4	3.9
Q5	4.4
Rata-rata Keseluruhan	4.38
Ease of Use	
Q1	4.1
Q2	4
Q3	4.2
Q4	3.9
Q5	4.2
Rata-rata Keseluruhan	4.08

Selanjutnya, analisis dilakukan pada tanggapan dari Bapak Ir. Atok Nardyawanto, MMT selaku perwakilan Bina Marga Provinsi Jawa Timur. Pada Tabel 6.8 disajikan rekapitulasi

jawaban kuisioner yang dijawab oleh Bapak Ir. Atok Nardyawanto, MMT.

Tabel 6.8 Rekapitulasi jawaban Uji Skenario Bpk. Atok

Skenario	Fitur	Nilai
Q1	General	3
Q2	Registrasi akun baru	3
Q3	Verifikasi akun	3
Q4	<i>Login</i>	5
Q5	Filtering laporan pada peta	3
Q6	Navigasi pada peta	4
Q7	Pembuatan laporan baru	4
Q8	Reputasi laporan	4
Q9	Reputasi <i>user</i>	4
Q10	Dashboard	4
Q11	Daftar <i>user</i>	4
Q12	Daftar laporan	4
Q13	Edit laporan	4
Q14	Edit <i>pprofile</i>	4
Q15	Pelaporan melalui <i>SMS</i>	4
Q16	Ekspor <i>SMS</i>	4
Q17	Verifikasi laporan	4
Q18	Isi penanganan & penutupan laporan	4
Q19	Banning <i>user</i>	4
Q20	Hapus laporan	5
<i>Rata-rata nilai :</i>		3.9

Berikut pada Tabel 6.9 adalah rekapitulasi tanggapan untuk bagian kedua yaitu *perceived usefulness* dan *perceived ease of use*.

Tabel 6.9 Rekapitulasi kuisisioner TAM Bpk. Atok

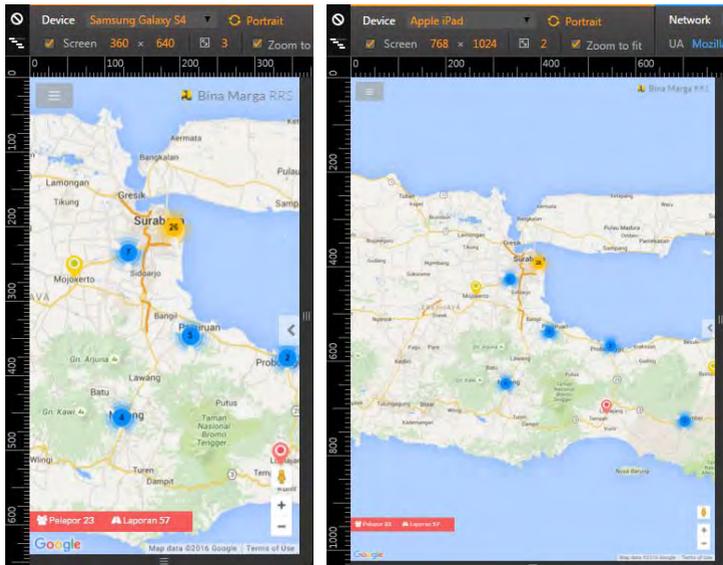
Pernyataan	Nilai
Usefulness	
Q1	4
Q2	4
Q3	4
Q4	4
Q5	4
Rata-rata nilai	4
Ease of Use	
Q1	4
Q2	4
Q3	4
Q4	4
Q5	4
Rata-rata nilai	4

6.1.4 Responsive Layout Testing

Responsive Layout Testing dilakukan menggunakan bantuan *Developer Tools* Google Chrome. Terdapat pilihan perangkat yang akan mensimulasikan rendering halaman sesuai dengan spesifikasi layar yang mereka miliki. Dalam pengujian ini, perangkat yang dipakai adalah Samsung Galaxy S4 dan Apple iPad. Dua perangkat ini dipilih karena populer dan merupakan perwakilan dari dua kategori perangkat mobile yang beredar saat ini, yaitu *smartphone* dan *tablet*. Samsung memiliki ukuran layar sebesar 5 inch dengan resolusi 360 x 640 *pixel*, sedangkan apple 7,9 inch dengan resolusi 768 x 1024 *pixel*.

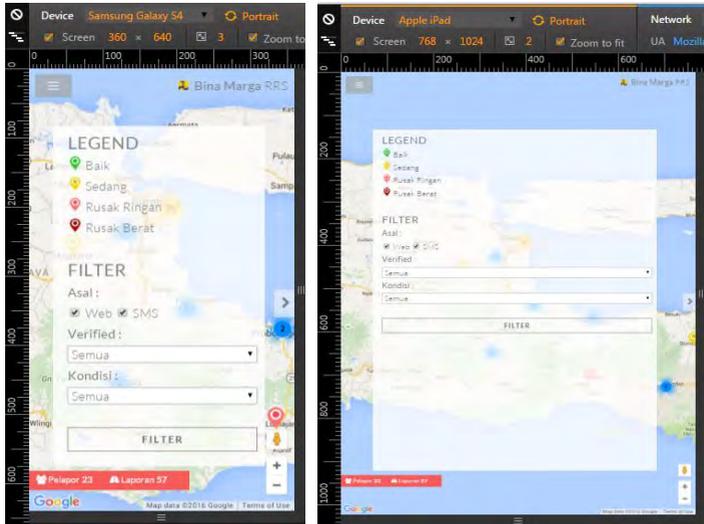
Pengujian akan dilakukan pada beberapa halaman yang menjadi komponen utama aplikasi dan kemungkinan akan sering dibuka oleh pengguna. Halaman pertama adalah Halaman Beranda yang menampilkan laporan-laporan pada sebuah peta. Berikut

pada Gambar 6.56 ditampilkan hasil tampilan yang dihasilkan pada masing-masing perangkat.



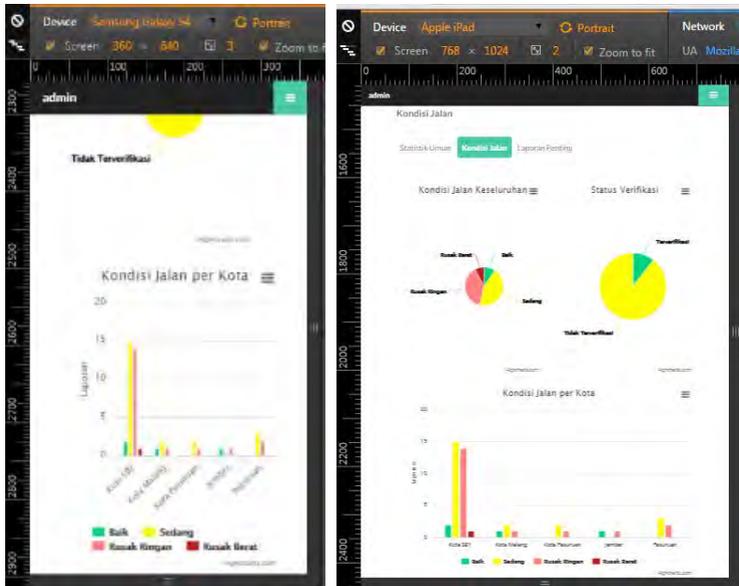
Gambar 6.56 Responsive Testing pada Halaman Beranda

Halaman beranda memiliki *filter*, *filter* tersebut bisa disembunyikan jika dibuka pada perangkat dengan layar yang kecil. Berikut pada Gambar 6.57 ditampilkan *filter* pada beranda dimasing-masing perangkat.



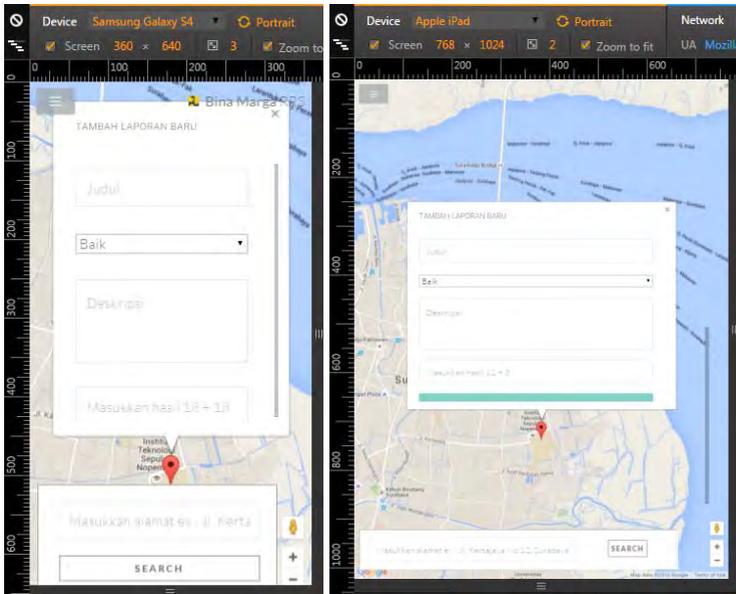
Gambar 6.57 *Responsive Testing filter* pada Halaman Beranda

Selanjutnya, halaman Dashboard diuji. Pada halaman Dashboard terdapat grafik-grafik, pada Gambar 6.58 dapat dilihat bahwa grafik dapat menyesuaikan ukuran dan tata letaknya pada setiap perangkat.



Gambar 6.58 Responsive Testing pada Halaman Dashboard

Terakhir, pengujian dilakukan pada halaman pembuatan laporan baru. Berikut pada Gambar 6.59 ditampilkan hasil pengujian pada masing-masing perangkat.



Gambar 6.59 Responsive Testing pada Halaman Laporan Baru

6.2 Pembahasan

Pada subbab ini akan diuraikan tentang pembahasan dari hasil kuesioner yang telah dilakukan sebelumnya.

6.2.1 Pembahasan *System Testing*

Pada *System Testing*, setiap fitur aplikasi telah berjalan dengan baik dan mampu dioperasikan sesuai dengan output yang diharapkan. Aplikasi juga bisa mencegah *user* untuk melebihi batasan sistem dan bisa mendeteksi adanya kesalahan *user*. Dari hasil *System Testing* yang telah dilakukan maka aplikasi dapat disimpulkan telah berjalan dengan baik dan benar.

6.2.2 Pembahasan *System Performance Test*

Dari 3 skenario *System Performance Test* yang telah dilakukan dihasilkan beberapa poin penting yang disajikan pada Tabel 6.10.

Tabel 6.10 Hasil *System Performance Test*

Skenario	Average Response Time	Peningkatan response time terus menerus	Ketercapaian Target
50 <i>User</i>	0,51 detik	Tidak	Tercapai
100 <i>User</i>	7,31 detik	Ya	Tidak Tercapai
200 <i>User</i>	45, 27 detik	Ya	Tidak Tercapai

Dari Tabel 6.10 didapatkan bahwa skenario 200 *user* tidak mencapai target *response time* yang ditentukan. Jika Bina Marga ingin agar Bina Marga RSS dapat diakses oleh sekitar 200 orang maka Bina Marga harus melakukan upgrade *hardware server*. Upgrade dapat membantu untuk meningkatkan kapasitas *hardware* sehingga dapat menangani lebih banyak lagi permintaan.

Skenario 100 *user* tidak bisa diterima, meskipun response time-nya masih dibawah 10 detik. Pada skenario 100 *user* terjadi peningkatan response time secara terus menerus. Untuk itu, penggunaan optimal aplikasi Bina Marga RRS dengan spesifikasi *hardware* yang telah dipaparkan sebelumnya adalah

untuk 50 *user* karena response timenya yang cepat bahkan dibawah 1 detik.

6.2.3 Pembahasan *User Acceptance Testing*

Dari hasil tanggapan responden yang bertindak sebagai pelapor pada *User Acceptance Testing*, dapat ditarik beberapa kesimpulan :

- Secara keseluruhan pelapor merasa mudah untuk menggunakan fitur-fitur aplikasi yang dicoba pada Test Scenario. Hal tersebut bisa dilihat pada hasil rata-rata keseluruhan yang bernilai 4.3. Namun, terdapat beberapa fitur aplikasi yang memiliki nilai rata-rata per fitur yang berada dibawah 4, yaitu daftar laporan dan pelaporan melalui *SMS* masing-masing memiliki nilai 3.9 dan 3.5. Meskipun tidak berada pada rentang sulit dilakukan, ini menunjukkan bahwa pelapor masih belum terbiasa dengan fitur tersebut terutama pelaporan melalui *SMS*. Pelaporan melalui *SMS* cukup sulit karena memerlukan *format* yang telah ditentukan.
- Rata-rata keseluruhan nilai pada bagian kedua, yaitu penilaian *perceived usefulness* dan *perceived ease of use* adalah masing-masing 4.38 dan 4.08. Hal ini menunjukkan bahwa aplikasi Bina Marga RRS berguna dan mudah untuk digunakan bagi pelapor.

Sedangkan, dari hasil tanggapan responden yang bertindak sebagai penanggap (pihak Bina Marga) pada *User Acceptance Testing*, dapat ditarik beberapa kesimpulan :

- Secara keseluruhan penanggap merasa mudah untuk menggunakan fitur-fitur aplikasi yang dicoba pada uji skenario. Hal tersebut bisa dilihat pada hasil rata-rata keseluruhan yang bernilai 3.9. Namun, terdapat beberapa fitur aplikasi yang memiliki nilai rata-rata per fitur yang diberi nilai 3, yaitu membuka aplikasi, registrasi, dan verifikasi. Penanggap merasa sulit untuk

membuka aplikasi melalui *ip address* karena aplikasi belum berada pada *server* yang *online*. Sedangkan, untuk registrasi penanggap merasa tidak cukup diberikan arahan untuk mengisi *form*. Pada *form* registrasi tidak ada informasi batasan *password* dan label pada *form upload profile picture*.

- Rata-rata keseluruhan nilai pada bagian kedua, yaitu penilaian *perceived usefulness* dan *perceived ease of use* adalah masing-masing 4 dan 4. Hal ini menunjukkan bahwa aplikasi Bina Marga RRS berguna dan mudah untuk digunakan bagi pelapor.

6.2.4 Pembahasan *Responsive Layout Test*

Gambar-gambar hasil simulasi tampilan dengan *Developer Tools* pada Google Chrome menunjukkan bahwa Bina Marga RRS sudah dapat menyesuaikan tampilan halamannya sesuai dengan ukuran layar pada perangkat-perangkat *mobile*. Tidak ada elemen halaman yang terpotong dan semua dapat diakses dengan baik.

Halaman ini sengaja dikosongkan

BAB VII

KESIMPULAN DAN SARAN

Bab ini berisi mengenai kesimpulan yang didapatkan dari hasil penelitian yang telah dilakukan. Kesimpulan ini diharapkan dapat menjawab tujuan yang telah ditetapkan di awal penelitian. Saran diberikan untuk digunakan dalam penelitian selanjutnya.

7.1 Kesimpulan

Dari pelaksanaan penelitian Tugas Akhir ini di dapatkan kesimpulan sebagai berikut.

1. Proses *Crowdsourcing data* untuk pelaporan kerusakan jalan dilakukan dengan memanfaatkan teknologi *Google Maps API*, *Gammu*, dan *PHP*.
2. Melalui *System Testing* diketahui bahwa fungsionalitas aplikasi sudah berjalan baik.
3. Dari hasil uji *user acceptance* pelapor lebih mudah untuk melapor melalui *web* daripada *SMS*. Karena, pelaporan melalui *SMS* menggunakan *format* pelaporan yang cenderung sulit bagi pengguna.
4. Kinerja *server* dengan sistem saat ini mampu menangani hingga 50 *user* yang mengakses aplikasi tersebut.
5. Tampilan aplikasi sudah dapat berjalan baik melalui perangkat *desktop* maupun perangkat bergerak.

7.2 Saran

Untuk pengembangan Tugas Akhir ini agar kedepannya lebih baik dan lebih bermanfaat, maka terdapat beberapa saran yang dapat dipertimbangkan yaitu:

1. Aplikasi dapat membedakan jenis jalan, seperti jalan kota dan jalan provinsi.
2. Jalur masukan laporan diperbanyak, misalnya melalui jejaring sosial.

3. Membuat versi *native* Android aplikasi sehingga kinerjanya akan lebih cepat.
4. Spesifikasi *server* perlu ditingkatkan jika perkiraan user yang mengakses aplikasi pada saat yang bersamaan lebih dari 50 *user*.
5. Perlu dibuat sebuah *user guide* untuk pelapor dan *administrator* agar aplikasi dapat dipakai dan dikelola dengan baik.
6. Tambahkan fitur sugesti pada pencarian alamat, sehingga dapat meminimalisir alamat yang tidak valid.
7. Tambahkan informasi daftar Jalan Provinsi yang dikelola Bina Marga pada Bina Marga RRS.

Lampiran A

Kuesioner User Acceptance Testing

Informasi Umum

Nama :
Jenis Kelamin : Laki-laki / Perempuan
Jurusan :

Test Scenario

1. Silahkan buka *Website Bina Marga Road Reporting System* pada alamat <http://192.168.43.15/RRS>

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan
Saran:

2. Silahkan lakukan registrasi pada Halaman Beranda.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan
Saran:

3. Silahkan buka *email* Anda dan lakukan verifikasi akun.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan
Saran:

4. Silahkan *login* dengan akun yang sudah dibuat sebelumnya.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan

Saran: _____

5. Silahkan buka halaman Beranda dan lakukan *filter* kondisi dan asal laporan.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan

Saran:

6. Silahkan buka halaman Beranda dan buka salah satu informasi laporan yang tersedia.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan

Saran:

7. Silahkan buka halaman Laporan Baru dan posisikan *marker* pada jalan yang ingin dilaporkan dengan memasukkan alamat pada kotak pencarian atau dengan melakukan klik pada peta, lalu isi informasi laporan dan simpan.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan

Saran:

8. Silahkan buka halaman Beranda dukung laporan yang Anda inginkan.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan

Saran:

9. Silahkan buka halaman profil *user* lain dan berikan reputasi kepadanya.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan

Saran:

10. Silahkan buka halaman *Dashboard* dan lakukan navigasi antar bagian *Dashboard*.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan

Saran:

11. Silahkan buka halaman Daftar *User* dan cari *user* dengan *username* “admin”.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan

Saran:

12. Silahkan buka halaman Daftar Laporan dan cari laporan yang Anda telah buat sebelumnya dengan bantuan *filter* yang ada.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan

Saran:

13. Silahkan buka halaman Detil Laporan yang Anda buat sebelumnya, lalu klik tombol *edit report*. Ubah judul dan gambar pendukung laporan.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan

Saran:

14. Silahkan buka halaman Beranda lalu arahkan *pointer* pada nama *user* yang ada di *menubar* lalu klik *edit profile* dan ubah *password* Anda.

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan
Saran:

15. Silahkan melakukan pelaporan kerusakan jalan melalui *SMS* dengan *format* berikut:

nama pelapor * tingkat kerusakan * alamat lengkap * deskripsi

Contoh:

Jono*3* Jl. Kertajaya Indah No. 1, surabaya * Terdapat lubang berdiameter 1 meter dan berkedalaman 70cm tepat didepan minimarket

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan
Saran:

16. Silahkan buka halaman *SMS Gateway* dan ekspor *SMS* yang Anda kirimkan

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan
Saran:

17. Silahkan buka menu laporan dan verifikasi laporan yang Anda kirimkan

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan
Saran:

18. Silahkan buka detil laporan dan isi penanganan yang diberikan pada laporan kerusakan jalan, lalu tutup laporan tersebut

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan
Saran:

19. Silahkan buka profil salah satu *user* dan lakukan banning

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan
Saran:

20. Silahkan buka menu laporan dan hapus beberapa laporan

Sangat Sulit Dilakukan 1 2 3 4 5 Sangat Mudah Dilakukan
Saran:

Kuesioner Technology Acceptance Model (TAM)

Perceived Usefulness

1. Menggunakan Bina Marga RRS mempercepat saya dalam melaporkan jalan rusak di Jawa Timur.

Sangat Tidak Setuju 1 2 3 4 5 Sangat Setuju

2. Menggunakan Bina Marga RRS membantu saya dalam melaporkan jalan rusak di Jawa Timur.

Sangat Tidak Setuju 1 2 3 4 5 Sangat Setuju

3. Menggunakan Bina Marga RRS mempermudah saya dalam mengetahui kondisi jalan di Jawa Timur.

Sangat Tidak Setuju 1 2 3 4 5 Sangat Setuju

4. Menggunakan Bina Marga RRS mampu membantu mempercepat proses penanganan jalan rusak.

Sangat Tidak Setuju 1 2 3 4 5 Sangat Setuju

5. Saya merasa Bina Marga RRS berguna bagi saya, masyarakat maupun pihak Bina Marga sebagai pengelola jalan.
Sangat Tidak Setuju 1 2 3 4 5 Sangat Setuju

Perceived Ease-of-Use

1. Menggunakan Bina Marga RRS mudah bagi saya
Sangat Tidak Setuju 1 2 3 4 5 Sangat Setuju
2. Interaksi saya dan Bina Marga RRS jelas dan dapat dipahami
Sangat Tidak Setuju 1 2 3 4 5 Sangat Setuju
3. Mudah bagi saya untuk memahami Bina Marga RRS dan menjelaskannya kepada orang lain
Sangat Tidak Setuju 1 2 3 4 5 Sangat Setuju
4. Saya merasa Bina Marga RRS fleksibel untuk digunakan
Sangat Tidak Setuju 1 2 3 4 5 Sangat Setuju
5. Saya merasa mudah dalam menggunakan Bina Marga RRS
Sangat Tidak Setuju 1 2 3 4 5 Sangat Setuju

Lampiran B
Rekapitulasi Jawaban Responden terhadap Kemudahan Fitur (Uji Skenario)

Fitur	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Rata-rata
General	3	4	5	4	4	4	5	5	5	4	4.3
Registrasi akun baru	4	4	5	5	5	5	5	3	4	4	4.4
Verifikasi akun	4	5	5	4	5	5	4	5	5	4	4.6
<i>Login</i>	5	5	5	4	4	4	5	4	4	4	4.4
Filtering laporan pada peta	4	4	4	5	5	4	5	4	4	4	4.3
Navigasi pada peta	5	5	5	4	2	3	5	5	5	3	4.2
Pembuatan laporan baru	4	3	4	3	4	5	5	5	5	4	4.2
Reputasi laporan	5	4	5	5	4	4	5	5	5	4	4.6
Reputasi <i>user</i>	4	5	5	4	5	4	5	4	4	4	4.4
Dashboard	4	5	5	4	4	4	5	4	4	4	4.3
Daftar <i>user</i>	4	5	4	4	3	4	5	4	5	4	4.2

B - 2 -

Daftar laporan	3	5	4	4	3	4	5	4	4	3	3.9
Edit laporan	3	5	5	4	5	5	5	5	5	4	4.6
Edit profil	4	5	5	5	5	4	5	4	5	4	4.6
Pelaporan melalui SMS	4	3	3	3	4	4	4	3	4	3	3.5
<u>Rata-rata</u>	4	4.47	4.6	4.13	4.13	4.2	4.87	4.27	4.53	3.8	4.3

Keterangan :

R : Responden

Lampiran C
Rekapitulasi Jawaban Responden terhadap Kuesioner TAM

Pernyataan	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Rata-rata
Usefulness											
Q1	4	4	4	5	5	4	5	4	4	4	4.3
Q2	4	5	5	5	5	5	5	5	5	5	4.9
Q3	4	4	5	5	5	4	4	4	5	4	4.4
Q4	4	4	5	4	4	4	5	3	3	3	3.9
Q5	4	4	5	4	5	5	5	4	4	4	4.4
Rata-rata Keseluruhan											4.38
Ease of Use											
Q1	4	4	5	4	4	3	4	4	5	4	4.1
Q2	4	4	5	4	4	3	5	3	5	3	4
Q3	4	3	5	4	4	5	5	3	4	5	4.2
Q4	4	4	5	4	4	4	4	4	3	3	3.9
Q5	4	4	5	4	4	4	4	4	5	4	4.2
Rata-rata Keseluruhan											4.08

Halaman ini sengaja dikosongkan

Lampiran D

Saran Responden Mengenai Fitur Aplikasi

Berikut adalah saran-saran dari responden yang bertindak sebagai pelapor pada Bina Marga RRS :

- Penggunaan kata-kata dalam aplikasi masih banyak yang dicampur dengan bahasa Inggris.
- Kurang adanya penjelasan pada *form* Registrasi
- Belum ada fitur forgot *password*
- Pada tampilan window laporan dipeta cara untuk menutup windownya masih kurang. Seharusnya ditambahkan lagi cara-cara lain untuk menutup window tersebut.
- Seharusnya link ke Halaman Dashboard ditambahkan ke menu bar, tidak pada dropdown list pada nama *user*.
- Pencarian nama pada Halaman Daftar *User/Report* harusnya dipisah diatas.
- Seharusnya ada batasan berapa kali pelapor boleh melakukan perubahan pada laporan, misalnya setelah diverifikasi tidak boleh lagi laporan diubah. Ini dilakukan untuk mencegah terjadinya laporan yang tidak konsisten.

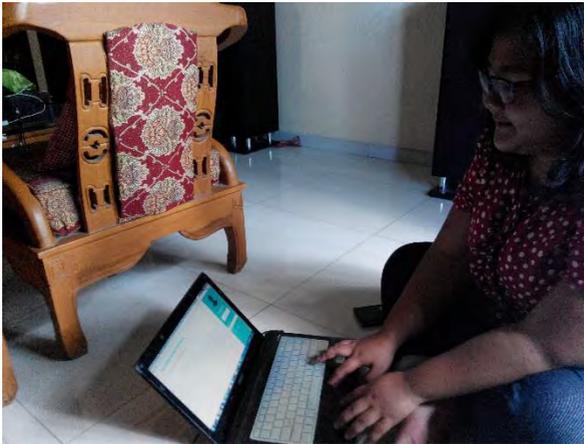
Berikut adalah saran-saran dari pihak Bina Marga Provinsi Jawa Timur pada Bina Marga RRS :

- Buat *User Guide* agar pengguna dapat mempelajari fitur-fitur aplikasi
- Tambahkan fitur untuk mengupload gambar kondisi jalan sebelum dan sesudah penanganan.

Halaman ini sengaja dikosongkan

Lampiran E

Dokumentasi *User Acceptance Testing*



E - 2 -



DAFTAR PUSTAKA

- [1] JPPN, "Hingga Mei 2015, Bina Marga DKI Sudah Perbaiki 1.964 Jalan Rusak," 2015. [Online]. Available:
<http://www.jpnn.com/read/2015/05/10/303148/Hingga-Mei-2015,-Bina-Marga-DKI-Sudah-Perbaiki-1.964-Jalan-Rusak>. [Accessed 16 September 2015].
- [2] M. F. Goodchild and J. Alan Glennon, "Crowdsourcing geographic information for disaster response: a research," 2010.
- [3] M. P. U. RI, "Peraturan Menteri Pekerjaan Umum nomor 08/PRT/M/2010," 08 07 2010. [Online]. Available:
<http://binamarga.pu.go.id/bm/index.php/produk/2014-11-28-03-12-26/peraturan-menteri?task=weblink.go&id=29>. [Accessed 01 10 2015].
- [4] M. P. U. RI, "Peraturan Menteri Pekerjaan Umum nomor 04/PRT/M/2012," 01 02 2012. [Online]. Available:
<http://binamarga.pu.go.id/bm/index.php/produk/2014-11-28-03-12-26/peraturan-menteri?task=weblink.go&id=33>. [Accessed 01 10 2015].
- [5] M. P. U. R. Indonesia, "Peraturan Menteri Pekerjaan Umum nomor 13 /PRT/M/2011," 2011. [Online]. Available:
<http://binamarga.pu.go.id/bm/index.php/produk/2014-11-28-03-12-26/peraturan-menteri?task=weblink.go&id=47>. [Accessed 01 10 2015].
- [6] A. Muttaqin and R. Arief Setyawan, "Perancangan dan pembuatan Application Programming Interface Server untuk Arduino," 2013.

- [7] J. Howe, "crowdsourcing.com," 2008. [Online].
- [8] J. Goyvaerts, Regular Expressions: The Complete Tutorial, 2007.
- [9] I. Irwansyah, "Rancang Bangun SMS Gateway Berbasis Web dengan Sistem Operasi Linux Ubuntu," p. 81, 2008.
- [10] Gusti, "Menkominfo: 270 Juta Pengguna Ponsel di Indonesia," UGM, 10 03 2014. [Online]. Available: <http://ugm.ac.id/id/berita/8776-menkominfo%3A.270.juta.pengguna.ponsel.di.indonesia>. [Accessed 01 10 2015].
- [11] J. Meier, Performance Testing Guidance for Web Applications, Microsoft, 2007.
- [12] J. Nielsen, "Website Response Times," Nielsen Norman Group, 21 06 2010. [Online]. Available: <https://www.nngroup.com/articles/website-response-times/>. [Accessed 11 01 2016].
- [13] S.Balaji, "Waterfall vs V-Model vs Agile : A Comparative Study On SDLC," *International Journal of Information Technology and Business Management*, vol. 2, no. 1, p. 27, 2012.
- [14] F. D. Davis, "User acceptance of information technology : system characteristics, user perceptions, and behavioural impacts," *Machine Studies*, pp. 475 - 476, 1991.
- [15] G. D., SPSS for Windows step by step: A simple guide and Reference Fourth Edition, SPSS for Windows step by step: A simple guide and, 2003.
- [16] A. S. Beavers, "Practical Considerations for Using Exploratory Factor Analysis in Educational Research," *Practical Assesment, Research & Evaluation*, vol. 18, no. 6, p. 4, 2013.

BIODATA PENULIS



Penulis dilahirkan di Tangerang, Banten, pada tanggal 26 Januari 1995. Penulis merupakan anak kedua dari dua bersaudara. Penulis menempuh pendidikan di TK Tunas Bangsa, SDN 5 Kedaung Wetan, SMPN 1 Tangerang, dan SMAN 2 Tangerang. Pada tahun 2012 penulis diterima di jurusan Sistem Informasi – Institut Teknologi Sepuluh Nopember (ITS) sebagai mahasiswa melalui jalur SNMPTN.

Selain kesibukan akademik, penulis aktif dalam organisasi PMK (Persekutuan Mahasiswa Kristen) ITS. Selain itu juga mengikuti berbagai kegiatan kemahasiswaan dan kepanitian. Tugas Akhir yang dipilih penulis di Jurusan Sistem Informasi ini masuk ke dalam bidang minat E-bisnis. Penulis dapat menghubungi melalui *email* yoasyohanda@gmail.com.