



TUGAS AKHIR - SM141501

**SISTEM PELAPORAN DAN MONITORING
KERUSAKAN JALAN BERBASIS WEB
MENGGUNAKAN “MODEL VIEW
CONTROLLER (MVC)”**

**TIARA SURTIKANTI
NRP 06111340000107**

**Dosen Pembimbing
Dr. Budi Setiyono, S.Si, MT**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika, Komputasi dan Sains Data
Institut Teknologi Sepuluh Nopember
Surabaya 2018**



FINAL PROJECT - SM141501

**A WEB-BASED ROAD DAMAGE REPORTING
AND MONITORING SYSTEM USING “MODEL
VIEW CONTROLLER (MVC)”**

TIARA SURTIKANTI

NRP 06111340000107

Supervisor

Dr. Budi Setiyono, S.Si, MT

DEPARTMENT OF MATHEMATICS

Faculty of Mathematics, Computation and Data Sciences

Sepuluh Nopember Institute of Technology

Surabaya 2018

LEMBAR PENGESAHAN

SISTEM PELAPORAN DAN MONITORING KERUSAKAN JALAN BERBASIS WEB MENGGUNAKAN “MODEL VIEW CONTROLLER (MVC)”

*A WEB-BASED ROAD DAMAGE REPORTING AND
MONITORING SYSTEM USING “MODEL VIEW
CONTROLLER (MVC)”*

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat untuk
memperoleh gelar Sarjana Sains pada bidang Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika, Komputasi dan Sains Data
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :

TIARA SURTIKANTI
NRP. 06111340000107

Menyetujui,
Dosen Pembimbing

Dr. Budi Setyono, S.Si, MT
NIP. 19720207 199702 1 001



v

SISTEM PELAPORAN DAN MONITORING KERUSAKAN JALAN BERBASSIS WEB MENGGUNAKAN “MODEL VIEW CONTROLLER (MVC)”

Nama Mahasiswa : Tiara Surtikanti
NRP : 06111340000107
Departemen : Matematika FMKSD-ITS
Pembimbing : Dr. Budi Setiyono, S.Si, MT

Abstrak

Kondisi jalan raya yang kurang baik dapat menyebabkan kecelakaan terhadap pengguna jalan raya. Kerusakan jalan ini berupa retak-retak (*cracking*), gelombang (*corrugation*), juga kerusakan berupa alur atau cekungan arah memanjang jalan sekitar jejak roda kendaraan (*rutting*) ada juga berupa genangan aspal diperlukaan jalan (*bleeding*), dan ada juga berupa lobang-lobang (*pothole*). Kerusakan jalan seperti ini biasanya disebabkan oleh berbagai faktor misalnya, akibat beban roda kendaraan berat yang lalulalang dan juga kondisi muka air tanah yang tinggi. Setelah dilakukan analisis, maka akan dibuat suatu sistem pelaporan berbasis web dengan menggunakan MVC (Model View Controller). Sehingga dengan adanya sistem ini dapat menampilkan informasi dari pengguna jalan mengenai infrastuktur jalan yang mengalami kerusakan dan dapat dijadikan acuan dalam penanganan kerusakan jalan dengan tepat.

Kata kunci: jalan raya, MVC, monitoring, Google Map API

**A WEB-BASED ROAD DAMAGE REPORTING AND
MONITORING SYSTEM USING “MODEL VIEW
CONTROLLER (MVC)”**

Name	:	Tiara Surtikanti
NRP	:	06111340000107
Department	:	MathematicsFMKSD-ITS
Supervisor	:	Dr. Budi Setiyono, S.Si, MT

Abstract

Improper road conditions can cause accidents to road users. The damage of this road is cracking, corrugation, also damage of the form of groove or basin direction of road length around the vehicle wheels (rutting) there is also puddle of bleeding asphalt, and there are also in the form of holes (pothole). Road damage like this is usually caused by various factors, for example, due to heavy wheel loads of passing vehicles and high groundwater conditions. After analysis, it will be made a web-based reporting system using MVC (Model View Controller). So that with this system can display information from road users regarding road infrastructure that has been damaged and can be used as a reference in handling road damage properly.

Keywords: *highway, MVC, monitoring, Google Map API*

X

KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat Allah SWT, karena dengan ridlo-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul

“SISTEM PELAPORAN DAN MONITORING KERUSAKAN JALAN BERBASIS WEB MENGGUNAKAN “MODEL VIEW CONTROLLER (MVC)””

yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Sarjana Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan baik berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal tersebut, penulis ingin mengucapkan terima kasih kepada :

1. Bapak Dr. Imam Mukhlash, S.Si, MT selaku Kepala Departemen Matematika ITS.
2. Bapak DR. Mahmud Yunus, M.Siselaku Dosen Wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Departemen Matematika ITS.
3. Bapak Dr. Budi Setiyono, S.Si, MTselaku Dosen Pembimbing yang telah memberikan bimbingan dan motivasi kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
4. Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi S1 Departemen Matematika ITS.
5. Drs. Iis Herisman, M.Si selaku Sekretaris Program Studi S1 Departemen Matematika ITS.
6. Seluruh jajaran dosen dan staf Departemen Matematika ITS.
7. Keluarga tercinta yang senantiasa memberikan dukungan dan do'a yang tak terhingga.

8. Teman-teman angkatan 2013 yang saling mendukung dan memotivasi.
9. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juli 2018

Penulis

Special thanks to

Keberhasilan penulisan Tugas Akhir ini tidak lepas dari bantuan dan dukungan dari orang-orang terdekat penulis. Oleh sebab itu, penulis mengucapkan terimakasih dan apresiasi secara khusus kepada:

1. Bapak Tonny Hendi Suryanto dan Ibu Sri Wahyuni, kedua orang tua penulis yang selalu memberikan doa terbaik, kasih sayang, dukungan, motivasi, dan nasehat kepada penulis.
2. Harry Kurniawan dan Larasati Sekar Agustina, kakak penulis yang selalu memberikan semangat dan dukungan serta kepercayaan kepada penulis.
3. Kartika Ratna Dewi, sahabat penulis yang selalu mendengarkan keluh kesah penulis, memberikan dukungan, semangat, motivasi, waktu dan kecerian kepada penulis.
4. Wahyuni Islammiah, Muhammad Asna Faqih dan Muslimatun Nadhifah, teman teman bimbingan penulis yang selalu memberikan dukungan dan semangat kepada penulis ketika penulis *low* motivasi, memberikan bantuan kepada penulis ketika kesusahan dalam pengerjaan dan mengijinkan penulis menginap.
5. Azizah Widiasmara, Dinda Ulima, Lailatur Rosyidah, Dinan Fakhrana, dan teman – teman seperjuangan Matematika 2013 yang lain yang tidak dapat penulis sebutkan satu persatu yang selalu memberikan doa, dukungan, motivasi, serta bantuan kepada penulis selama ini.
6. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

DAFTAR ISI

Abstrak	vii
<i>Abstract</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan Tugas Akhir.....	4
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu.....	7
2.2 Google Maps	8
2.3 Sistem Informasi Geografis (SIG).....	9
2.4 <i>Model View Controller</i> (MVC)	11
2.5 PHP.....	14
2.6 Database	15
2.7 MySql	16
2.8 CodeIgniter.....	17
BAB III METODE PENELITIAN	21
3.1 Studi Literatur.....	21
3.2 Pengumpulan Data	21
3.3 Analisis dan Perancangan Sistem.....	21
3.4 Pengujian dan Evaluasi Sistem.....	23
3.5 Kesimpulan dan Saran.....	23
3.6 Penyusunan Tugas Akhir.....	23
3.7 Diagram Alir Sistem.....	24
BAB IV ANALISIS DAN PERANCANGAN SISTEM.....	27
4.1 Deskripsi Sistem.....	27

4.2	Analisis Kebutuhan Sistem.....	27
4.3	Perancangan Sistem.....	28
4.3.1	Desain Proses.....	29
4.3.2	Desain Arsitektur Sistem	33
4.3.3	Desain Data.....	36
4.3.4	Desain Interface.....	41
4.3.4	Desain Arsitektur MVC.....	44
BAB V IMPLEMENTASI DAN PENGUJIAN		
PERANGKAT LUNAK		49
5.1	Ruang Lingkup Implementasi	49
5.1.1.	Hardware.....	49
5.1.2.	Software	49
5.2	Implementasi Perangkat Lunak	49
5.2.1.	Tahap Persiapan.....	49
5.2.2.	Implementasi Database	50
5.2.3.	Implementasi Algoritma	53
5.2.3	Implementasi Antarmuka.....	69
5.3	Uji Coba Perangkat Lunak	74
5.3.1	Uji Coba Proses Data Input	74
5.3.2	Uji Coba Manajemen Data	78
5.3.3	Uji Coba Menampilkan Data Spasial dan Non Spasial	80
5.3.4	Uji Coba Pencarian Jalan.....	83
5.3.5	Hasil Uji Coba Perangkat Lunak	85
BAB VI PENUTUP		87
6.1	Kesimpulan.....	87
6.2	Saran.....	87
DAFTAR PUSTAKA		89
LAMPIRAN		91

DAFTAR GAMBAR

Gambar 2.1 Google Maps	9
Gambar 2. 2 Arsitektur Web GIS.....	10
Gambar 3.1Diagram Alir Sistem	25
Gambar 4. 1 DFD Level 0.....	29
Gambar 4. 2 DFD Level 1.....	29
Gambar 4. 3 DFD Level 2.....	30
Gambar 4. 4 DFD Level 2 Akusisi Data.....	30
Gambar 4. 5 DFD Level 2 Proses Spasial.....	31
Gambar 4. 6 Usecase Diagram.....	32
Gambar 4. 7 Desain Arsitektur Sistem.....	34
Gambar 4. 8 Pengambilan Meta Data	35
Gambar 4. 9 Conceptual Data Model.....	36
Gambar 4. 10 Desain Menu Utama.....	42
Gambar 4. 11 Desain Menu Pencarian Jalan	42
Gambar 4. 12 Desain Tabel Pencarian Jalan.....	43
Gambar 4. 13 Desain Menu Input Data	43
Gambar 4. 14 Desain Menu Login.....	44
Gambar 4. 15 Desain Tabel Data Kerusakan	44
Gambar 4. 16 Desain Arsitetur MVC	45
Gambar 5. 1 Relasi Database gis_perbaikan_jalan	50
Gambar 5. 2 Tampilan Awal Sistem.....	69
Gambar 5. 3 Tampilan Tabel Awal Sistem.....	70
Gambar 5. 4 Tampilan Infomasi Kerusakan Jalan	70
Gambar 5. 5 Tampilan Hasil Pencarian Jalan	71
Gambar 5. 6 Tabel Hasil Pencarian Jalan	72
Gambar 5. 7 Tampilan Hasil Pencarian Berdasarkan radius.	72
Gambar 5. 8 Tampilan Halaman Input Kerusakan Jalan	73
Gambar 5. 9 Tampilan Halaman Login	73
Gambar 5. 10 Tampilan Tabel Data Kerusakan.....	74
Gambar 5. 11 Input data spasial dan non spasial	76
Gambar 5. 12 Hasil proses meta data.....	76
Gambar 5. 13 Halaman home	77
Gambar 5. 14 Hasil input data	77

Gambar 5. 15 Login untuk halaman admin.....	79
Gambar 5. 16 Data kerusakan pada halaman admin	79
Gambar 5. 17 Proses update oleh admin.....	80
Gambar 5. 18 Proses delete oleh admin.....	80
Gambar 5. 19 Tampilan data dalam peta	82
Gambar 5. 20 Tampilan dengan informasi meta data	82
Gambar 5. 21 Tampilan data dalam tabel	83
Gambar 5. 22 Hasil pencarian jalan.....	84
Gambar 5. 23 Hasil pencarian jalan berdasarkan radius	85

DAFTAR TABEL

Tabel 4.1 Tabel Kerusakan	37
Tabel 4.2 Tabel konfigurasi	39
Tabel 4.3 Tabel user.....	40

BAB I

PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang yang mendasari penulisan Tugas Akhir ini. Di dalamnya mencakup identifikasi permasalahan pada topik Tugas Akhir kemudian dirumuskan menjadi permasalahan yang diberikan batasan-batasan dalam pembahasan pada Tugas Akhir ini.

1.1 Latar Belakang

Jalan raya adalah salah satu prasarana yang akan mempercepat pertumbuhan dan pengembangan suatu daerah serta akan membuka hubungan sosial, ekonomi dan budaya antardaerah. Seiring dengan bertambahnya jumlah kepemilikan kendaraan menyebabkan meningkatnya volume lalu lintas. Kepadatan jalan raya dan beban kendaraan yang berlebihan mengakibatkan kerusakan pada jalan raya. Beberapa penyebab kerusakan jalan tersebut antara lain adalah peningkatan dan repetisi beban pada jalan, iklim, kondisi tanah dasar yang tidak stabil, serta sistem drainase yang tidak baik.

Kerusakan jalan raya tersebut haruslah mendapatkan penanganan secepatnya karena kondisi jalan yang rusak dapat merugikan pengguna jalan dan menyebabkan kecelakaan lalu lintas. Menurut data dari Kepolisian Republik Indonesia, pada tahun 2017 30% kecelakaan lalu lintas disebabkan oleh faktor prasarana dan lingkungan [1]. Namun tidak semua kerusakan jalan raya mendapatkan penanganan, hal itu disebabkan karena kurangnya monitoring terhadap pemeliharaan jalan raya sehingga diperlukan kerjasama antara pemerintah dan masyarakat agar kerusakan jalan tersebut dapat segera mendapatkan perbaikan. Pentingnya pelaporan kerusakan jalan adalah agar jalan yang mengalami kerusakan mendapat penanganan dengan cepat dan tepat.

Saat ini pemerintah mempunyai suatu sistem bernama LAPOR! (Layanan Aspirasi dan Pengaduan Online Rakyat) untuk meningkatkan partisipasi masyarakat dalam rangka pengawasan program pembangunan dan pelayanan publik. Hingga April 2015, LAPOR! Telah digunakan oleh lebih dari 290.000 pengguna dan menerima rata-rata lebih dari 800 laporan masyarakat per harinya. Akan tetapi dalam sistem ini administrator harus memilah-milah laporan terlebih dahulu sesuai kategori laporan untuk selanjutnya diteruskan ke instansi K/L/D terkait. [2]

Dalam permasalahan tersebut diperlukan sistem yang dapat membantu untuk melaporkan dan monitoring kondisi jalan raya. Sistem tersebut terdapat informasi yang dapat diperbarui, dan mudah diakses, serta web mapping yang dapat menunjukkan lokasi dengan cepat, dan menampilkan koordinat jalan serta posisi jalan. Dengan adanya sistem ini dapat membantu memudahkan masyarakat atau pengguna jalan dalam melaporkan kerusakan jalan, sehingga jalan yang mengalami kerusakan dapat penanganan yang cepat dan tepat.

Oleh karena itu, dalam penelitian Tugas Akhir ini dirancang suatu sistem pelaporan dan monitoring kerusakan jalan berbasis web menggunakan Model View Controller (MVC). Dalam sistem pelaporan ini dibutuhkan data berupa data spasial dan non spasial. Kedua data tersebut kemudian akan diintegrasikan dengan Google Maps API sehingga sistem pelaporan dan monitoring kerusakan jalan lebih efektif dan akurat dalam memberikan visualisasi data mengenai kerusakan jalan. Selain itu, sistem pelaporan dan monitoring kerusakan jalan berbasis web menggunakan MVC sehingga akan memudahkan dalam maintenance dan pengembangan dari sistem ini. Kelebihan lain dari penggunaan MVC adalah penggunaan ulang komponen-komponen antarmuka dan kemampuan untuk mendefinisikan kelas-kelas pengaturan tampilan (control style) yang menyediakan fitur-fitur umum

secara terpisah dengan fitur-fitur yang akan ditampilkan oleh aplikasi yang akan dikembangkan.

Beberapa penelitian terkait yaitu penelitian yang dilakukan oleh Faya Mahdia dan Fiftin Noviyanto pada tahun 2013 dengan judul “Pemanfaatan Google Maps API untuk Pembangunan Sistem pelaporan Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis Mobile Web”. Dalam penelitian itu Faya Mahdia dan Fiftin Noviyanto berhasil membuat sistem informasi dengan memanfaatkan Google Maps API untuk mengatur keperluan logistik pada posko bencana alam [3]. Selain itu pada tahun 2016, Helmi Bahar Alim melakukan penelitian dengan judul “Sistem Informasi Geografis Jalan dan Jembatan di Kabupaten Wonogiri” yang membahas mengenai sistem informasi geografis jalan dan jembatan yang digunakan untuk memantau jalan dan jembatan di Kabupaten Wonogiri. Hasil penelitian ini membantu mempermudah pengelolaan jalan dan jembatan di Kabupaten Wonogiri [4]. Pada tahun 2017, Junar A. Landicho melakukan penelitian berjudul “A Web-based Geographical Project Monitoring and Information System For The Road and Highways” dengan bahasan pembuatan sistem informasi berbasis web untuk monitoring pembangunan jalan dan jalan tol. Hasil penelitian ini, informasi mengenai pembangunan jalan agar lebih terorganisir beserta maps yang menyajikan lokasi pembangunan lebih jelas [5].

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah yang akan dibahas dalam penelitian ini adalah bagaimana merancang dan membangun sebuah sistem pelaporan dan monitoring kerusakan jalan berbasis web menggunakan MVC

1.3 Batasan Masalah

Batasan masalah yang digunakan dalam penulisan Tugas Akhir ini adalah :

1. Data yang digunakan dalam penelitian ini adalah data-data kerusakan jalan di Surabaya
2. Pengambilan gambar menggunakan perangkat yang tersambung GPS

1.4 Tujuan

Berdasarkan permasalahan yang telah dirumuskan sebelumnya, tujuan penulisan Tugas Akhir ini adalah merancang dan membangun sistem pelaporan kerusakan jalan berbasis web menggunakan MVC, sehingga dapat membantu dalam memonitoring kerusakan jalan.

1.5 Manfaat

Manfaat yang akan diperoleh dari Tugas Akhir ini adalah:

1. Sebagai bahan informasi mengenai jalan yang mengalami kerusakan
2. Sebagai acuan dalam mengambil keputusan dari laporan mengenai kerusakan jalan.
3. Sebagai sarana untuk masyarakat melaporkan jalan yang mengalami kerusakan.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika dari penulisan Tugas Akhir ini adalah sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini menjelaskan tentang gambaran umum dari penulisan Tugas Akhir ini yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang materi-materi yang mendukung Tugas Akhir ini, antara lain penelitian terdahulu, Google Maps, Sistem Informasi Geografis (SIG), Model View Controller (MVC), PHP, Database, MySql, CodeIgniter.

3. BAB III METODE PENELITIAN

Pada bab ini dibahas tentang langkah – langkah dan metode yang digunakan untuk menyelesaikan Tugas Akhir ini.

4. BAB IV PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK

Pada bab ini akan menguraikan bagaimana tahapan tahapan dalam perancangan implementasi.

5. BAB V HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai hasil pengujian Sistem Pelaporan dan Monitoring Kerusakan Jalan Menggunakan Model Vuew Controller (MVC). Setelah itu dilakukan analisis terhadap hasil implementasi.

6. BAB VI PENUTUP

Bab ini berisi kesimpulan yang diperoleh dari pembahasan masalah sebelumnya serta saran yang diberikan untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori yang digunakan dalam penyusunan Tugas Akhir ini. Dasar teori yang dijelaskan dibagi menjadi beberapa subbab yaitu penelitian terdahulu, Google Maps, GIS (*geographical Information System*), dan MVC (Model View Controller).

2.1 Penelitian Terdahulu

Pada tahun 2013 telah dilakukan suatu penelitian oleh Faya Mahdia dan Fiftin Noviyanto yang ditulis dengan judul “Pemanfaatan Google Maps API untuk Pembangunan Sistem pelaporan Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis Mobile Web”. Penelitian tersebut menggunakan Google Maps API untuk memberikan informasi kepada masyarakat dan instansi yang akan memberikan bantuan mengenai jarak, rute jalan, penunjuk arah, dan daftar kebutuhan logistik posko bencana alam. Sistem pelaporan logistik yang dihasilkan dapat digunakan untuk mengelola data logistik yang dimiliki oleh posko. Pengelolaan mencangkup manajemen data logistik yang masuk, logistik yang tersedia, an data penggunaan logistik. [3]

Selanjutnya pada tahun 2016, penelitian yang dilakukan Helmi Bahar Alim dengan judul “Sistem Informasi Geografis Jalan dan Jembatan di Kabupaten Wonogiri”. Penelitian ini dimaksudkan untuk memberikan informasi tentang kondisi jalan dan jembatan di Kabupaten Wonogiri dengan tersedianya sistem informasi geografis jalan dan jembatan yang dapat menjadi acuan dan dasar penetapan penanganan jalan dan jembatan secara cepat sehingga dapat mendukung sistem penanganan jaringan jalan dan jembatan yang ada. [4]

Pada tahun 2017, Junar A. Landicho melakukan penelitian dengan judul “A Web-based Geographical Project

Monitoring and Information System For The Road and Highways". Penelitian ini dimaksudkan untuk membuat sistem informasi mengenai proyek pembangunan jalan dan jalan tol. Dengan adanya maps dalam sistem informasi ini, user dapat menandai lokasi, melihat informasi dan detail dari proyek pembangunan jalan. [5]

Setelah mengetahui penelitian-penelitian yang sudah dilakukan, hal tersebut dapat membantu penulis untuk memperoleh beberapa pengetahuan dalam menyelesaikan tugas akhir ini.

2.2 Google Maps

Google Maps adalah jasa peta gratis dan online disediakan oleh Google yang dapat ditemukan di <https://maps.google.com>. Pada layanan Google Maps peta dapat digeser sesuai keinginan pengguna, mengubah tingkat zoom, serta mengubah tampilan peta. Google maps juga menawarkan peta yang dapat diseret dan gambar setelit untuk seluruh dunia, serta menawarkan rute perjalanan.

Google Maps dibuat dengan menggunakan kombinasi dari gambar peta, database, serta objek-objek interaktif yang dibuat dengan bahasa pemrograman HTML, Javascript, dan AJAX, serta beberapa bahasa pemrograman lainnya. Gambar-gambar peta yang muncul pada layar merupakan hasil komunikasi dari pengguna dengan database pada web server google untuk menampilkan gabungan dari potongan-potongan yang nantinya akan dipanggil sesuai kebutuhan permintaan.



Gambar 2.1Google Maps

Google Maps Application Programming Interface (API) merupakan suatu fitur aplikasi yang dikeluarkan oleh google untuk memfasilitasi pengguna yang ingin mengintegrasikan Google Maps ke dalam website masing-masing dengan menampilkan data point milik sendiri. Dengan menggunakan Google Maps API, Google Maps dapat di-embed pada website eksternal. Agar aplikasi Google Maps dapat muncul di website tertentu, diperlukan adanya API key. API key merupakan kode unik yang digenerasikan oleh google untuk suatu website tertentu, agar server Google Maps dapat mengenali. [6]

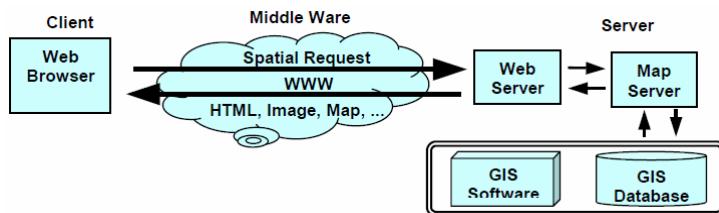
2.3 Sistem Informasi Geografis (SIG)

Pada dasarnya, istilah sistem informasi geografis merupakan gabungan dari tiga unsur pokok: sistem, informasi, dan geografis. Dengan melihat unsur-unsur pokoknya, maka sistem informasi geografis merupakan suatu sistem yang menekankan pada unsur informasi geografis. Istilah “geografis” merupakan bagian dari spasial (keruangan). Kedua istilah ini sering digunakan secara bergantian atau

tertukar hingga timbul istilah yang ketiga, geospasial. Ketiga istilah ini mengandung pengertian yang sama di dalam konteks sistem informasi geografis (SIG). [7]

Penggunaan kata “geografis” mengandung pengertian suatu persoalan mengenai bumi: permukaan dua atau tiga dimensi. Istilah “informasi geografis” mengandung pengertian informasi mengenai tempat-tempat yang terletak di permukaan bumi, pengetahuan mengenai posisi dimana suatu objek terletak di permukaan bumi, dan informasi mengenai keterangan-keterangan (atribut) yang terdapat di permukaan bumi yang posisinya diberikan atau diketahui. Dengan memperhatikan pengertian sistem informasi, maka SIG merupakan suatu kesatuan formal yang terdiri dari berbagai sumberdaya fisik dan logika yang berkenaan dengan objek-objek yang terdapat di permukaan bumi. [7]

Jadi SIG juga merupakan sejenis perangkat lunak yang dapat digunakan untuk pemasukan, penyimpanan, manipulasi, menampilkan, dan keluaran informasi geografis berikut atribut-atributnya. Menurut Demers definisi SIG adalah sistem komputer yang digunakan untuk mengumpulkan, memeriksa, mengintegrasikan, dan menganalisa informasi-informasi yang berhubungan dengan permukaan bumi. [7]



Gambar 2. 2Arsitektur Web GIS

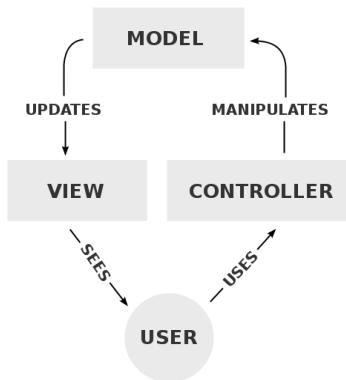
Aplikasi SIG saat ini tumbuh tidak hanya secara jumlah aplikasi namun juga bertambah dari jenis keragaman aplikasinya. Pengembangan aplikasi SIG kedepannya

mengarah kepada aplikasi berbasis Web yang dikenal dengan Web SIG. Hal ini disebabkan karena pengembangan aplikasi di lingkungan jaringan telah menunjukkan potensi yang besar dalam kaitannya dengan geo informasi. Sebagai contoh adalah adanya peta online sebuah kota dimana pengguna dapat dengan mudah mencari lokasi yang diinginkan secara online melalui jaringan intranet/internet tanpa mengenal batas geografi penggunanya. Secara umum sistem pelaporan Geografis dikembangkan berdasarkan pada prinsip input/masukan data, managemen, analisis dan representasi data. [8]

2.4 *Model View Controller (MVC)*

Model View Controller (MVC) adalah sebuah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan antarmuka/tampilan (user interface), data, dan proses sehingga memungkinkan untuk melakukan pengembangan atau pemeliharaan aplikasi secara lebih efektif dan efisien. Model View Controller pertama sekali dipublikasikan oleh penelitiXEROX PARC yang bekerja dalam pembuatan bahasa pemrogramanSmalltalk sekitar tahun1970-1980.

Dalam implementasinya kebanyakan framework dalam aplikasiwebsite adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web.Arsitektur metode MVC akan ditunjukkan pada Gambar 2.3.



Gambar 2. 3Arsitektur MVC

Metode MVC membagi aplikasi menjadi 3 komponen, yaitu :

a. Model

Model berfungsi sebagai pengelola perilaku dan data pada domain aplikasi, melakukan tanggapan terhadap permintaan informasi dan merespons instruksi untuk merubah suatu kondisi (state). Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.

b. View

View adalah representasi visual dari suatu aplikasi. Pada bagian ini memiliki fungsi untuk menyajikan data yang diterima oleh Controller dari Model. Atau dalam kata lain View adalah bagian dari sistem di mana HTML dihasilkan dan kemudian ditampilkan. Bisa dikatakan berupa halaman web.

c. Controller

Controller merupakan bagian yang menjembatani model dan view. Controller berisi perintah-perintah

berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web. [9]

Dengan menggunakan metode MVC maka aplikasi akan lebih mudah untuk dirawat dan dikembangkan. Untuk memahami metode pengembangan aplikasi menggunakan MVC diperlukan pengetahuan tentang pemrograman berorientasi objek (Object Oriented Programming).

Adapun jenis-jenis MVC diantaranya :

a. Server Side MVC

Server Side MVC biasa terjadi pada aplikasi web tradisional, yang tidak melibatkan client side seperti Javascript, Java applet, Flash, dan lain-lain. Server Side MVC menyerahkan keseluruhan proses bisnis pada server, aplikasi pada sisi pengguna hanya dapat menerima. MVC jenis ini kadang-kadang disebut juga dengan nama Thin Client.

b. Mixed Client Side and Server Side MVC

Pada Mixed Client Side and Server Side MVC 1 client tidak menggunakan model sebagai jembatan untuk melakukan komunikasi pada server, dibandingkan dengan Server Side MVC, arsitektur ini memiliki tingkat kompleksitas yang lebih tinggi karena lebih banyak komponen yang terlibat. Untuk selanjutnya arsitektur ini disebut, dengan Mixed MVC 1. Pada Mixed Client Side and Server Side MVC 2, client menggunakan model sebagai jembatan untuk melakukan komunikasi pada server, dibandingkan dengan arsitektur MVC yang lain, arsitektur ini memiliki tingkat kompleksitas yang paling tinggi karena lebih banyak komponen yang terlibat, sehingga membutuhkan sumber daya yang lebih besar pula. Untuk selanjutnya arsitektur ini disebut dengan Mixed MVC 2

c. Rich Internet Application MVC

Application MVC Rich Internet Application (RIA) disebut juga dengan nama Fat Client, merupakan aplikasi web yang memiliki kemampuan dan fungsi hampir seperti aplikasi desktop. RIA pada sisi client, memiliki mesin untuk mengambil data yang berada pada server, sehingga pada client terdapat bagian MVC sendiri dan hanya membutuhkan bagian model pada sisi server.

2.5 PHP

PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group.

PHP disebut bahasa pemrograman **server side** karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).

Pada awalnya PHP merupakan singkatan dari Personal Home Page. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dll.

Saat ini PHP adalah singkatan dari **PHP: Hypertext Preprocessor**, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: **PHP: Hypertext Preprocessor**.

PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi PHP License, sedikit

berbeda dengan lisensi GNU *General Public License* (GPL) yang biasa digunakan untuk proyek *Open Source*.

Untuk membuat halaman web, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bisa saja membuat website hanya menggunakan HTML saja. Web yang dihasilkan dengan HTML (dan CSS) ini dikenal dengan website statis, dimana konten dan halaman web bersifat tetap.

Sebagai perbandingan, website dinamis yang bisa dibuat menggunakan PHP adalah situs web yang bisa menyesuaikan tampilan konten tergantung situasi. Website dinamis juga bisa menyimpan data ke dalam database, membuat halaman yang berubah-ubah sesuai input dari *user*, memproses form, dll.

Untuk pembuatan web, kode PHP biasanya di sisipkan kedalam dokumen HTML. Karena fitur inilah PHP disebut juga sebagai Scripting Language atau bahasa pemrograman script. [10]

2.6 Database

Basis data (database) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang akan disimpan.

Basis data merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasi data, menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga update yang rumit.

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (database management system | DBMS). DBMS merupakan sistem perangkat lunak yang memungkinkan pengguna basis data (database user) untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien.

Ada beberapa fungsi yang harus ditangani DBMS seperti pendefinisian data, menangani permintaan pengguna untuk mengakses data, memeriksa sekuriti dan integriti data yang didefinisikan oleh DBA (Database Administrator), menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun media penyimpanan (disk) dan juga menangani unjuk kerja semua fungsi secara efisien.

Tujuan utama DBMS adalah untuk memberikan tinjauan abstrak data kepada pengguna. Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara dan juga bisa diakses secara efisien. Pertimbangan efisien di sini adalah rancangan struktur data yang kompleks tetapi masih bisa digunakan oleh pengguna awam tanpa mengetahui kompleksitas strukturnya.[11].

2.7 MySql

MySQL adalah sebuah perangkat lunak sistem manajemen basis dataSQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia

Dari pengertian di atas MySQL merupakan perangkat lunak database yang mempunyai fungsi untuk menyimpan informasi berupa data secara sistematis dan saling terintegrasi. Database terdiri dari tabel-tabel yang dapat diolah dan diintegrasikan antara tabel yang satu dengan tabel yang

lainnya dan tabel sendiri terdiri field-field yang diatur dan disesuaikan dengan kebutuhan untuk menampung data.[12]

MySQL mempunyai banyak keistimewaan, salah satunya adalah **Open Souce** artinya digunakan untuk umum dan gratis dalam penggunaannya. Jika ingin mendapatkan fitur lebih dapat menghubungi langsung pihak MySQL development. Keistimewaan yang lainnya adalah **Multi Platform** artinya bisa dijalankan di berbagai sistem operasi seperti Windows, Linux, Max OS, Solaris dan lain sebagainya. Kemudian **Multi User** artinya dapat digunakan oleh beberapa user (pengguna) dalam waktu bersamaan tanpa takut terganggu performanya. Selanjutnya Mudah dalam koneksi, disini banyak konektor-konektor yang dapat menghubungkan antara My SQL dengan aplikasi pemrograman. Kemudian GUI untuk pengolahan database My SQL pun lumayan banyak seperti : PHP MyAdmin, Navicat, My SQL Workbench dan lain sebagainya.[12]

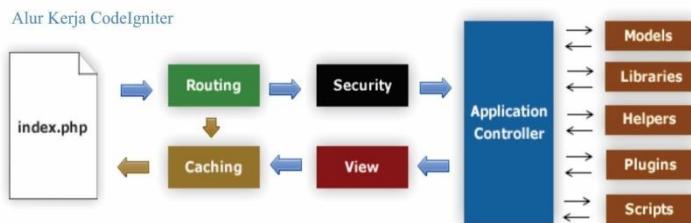
2.8 CodeIgniter

CodeIgniter adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis[10].

CodeIgniter menjadi sebuah framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, CodeIgniter juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodennya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih CodeIgniter sebagai framework pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh CodeIgniter, pembuat PHP Rasmus Lerdorf memuji CodeIgniter di frOSCon (Agustus 2008) dengan mengatakan bahwa dia menyukai

CodeIgniter karena “it is faster, lighter and the least like a framework.”

CodeIgniter pertama kali dikembangkan pada tahun 2006 oleh Rick Ellis. Dengan logo api yang menyala, CodeIgniter dengan cepat “membakar” semangat para web developer untuk mengembangkan web dinamis dengan cepat dan mudah menggunakan framework PHP yang satu ini.[13]



Gambar 2. 4 Alur Kerja CodeIgniter

- Index.php: Index.php disini berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
- The Router: Router akan memeriksa HTTP request untuk menentukan hal apa yang harus dilakukan oleh program.
- Cache File: Apabila dalam program sudah terdapat “cache file” maka file tersebut akan langsung dikirim ke browser. File cache inilah yang dapat membuat sebuah website dapat di buka dengan lebih cepat. Cache file dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter.
- Security: Sebelum file controller di load keseluruhan, HTTP request dan data yang disubmit oleh user akan disaring terlebih dahulu melalui fasilitas security yang dimiliki oleh codeigniter.

- Controller: Controller akan membuka file model, core libraries, helper dan semua resources yang dibutuhkan dalam program tersebut.
- View: Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam view file dan mengirimkannya ke browser supaya dapat dilihat. Apabila file view sudah ada yang di “cache” maka file view baru yang belum ter-cache akan mengupdate file view yang sudah ada.

BAB III

METODE PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang digunakan dalam penyusunan Tugas Akhir. Disamping itu, dijelaskan pula prosedur dan proses pelaksanaan tiap-tiap langkah yang dilakukan dalam menyelesaikan Tugas Akhir.

3.1 Studi Literatur

Studi literatur dilakukan melalui penelusuran buku-buku dan jurnal yang berkaitan dengan SIG, Google Map, serta MVC. Hal ini bertujuan untuk memahami konsep dari Tugas Akhir ini.

3.2 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang dibutuhkan dalam penelitian. Data yang dimaksud adalah data spasial dan non spasial. Data spasial yaitu berupa citra dari kerusakan jalan, sedangkan data non spasial adalah informasi mengenai identitas pelapor. Data tersebut yang akan digunakan sebagai input dari sistem.

3.3 Analisis dan Perancangan Sistem

Pada tahap ini dilakukan pembuatan sistem dengan menggunakan bahasa pemrograman PHP. Dalam sistem ini Google Maps API berfungsi sebagai layanan untuk menampilkan informasi lokasi kerusakan jalan dan sebagai penunjuk jalan berdasarkan data yang telah disimpan.

Tahap-tahap dalam perancangan sistem adalah sebagai berikut :

a. Desain Sistem

Pembuatan desain sistem dari sistem pelaporan dan monitoring kerusakan jalan raya berbasis web menggunakan MVC meliputi beberapa langkah, diantaranya :

- *Data Flow Diagram*

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data pada suatu sistem atau menjelaskan proses kerja suatu sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersuktur dan jelas. Secara singkatnya, DFD adalah alat pemodelan untuk memodelkan alur kerja sistem.

- *Conceptual Data Model*

CDM dipakai untuk menggambarkan secara detail struktur basis data dalam bentuk logik. Struktur ini independen terhadap semua software maupun struktur data storage tertentu yang digunakan dalam aplikasi ini. CDM terdiri dari objek yang tidak diimplementasikan secara langsung kedalam basis data yang sesungguhnya.

- *Use case Diagram*

Use case adalah kegiatan atau urutan interaksi yang saling berkaitan antara sistem dan aktor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Use case juga digunakan untuk membentuk perilaku (behaviour) sistem yang akan dibuat. Sebuah use case menggambarkan sebuah interaksi antara pengguna (aktor) dengan sistem yang ada.

b. Pembuatan Database

Pembuatan database diperlukan untuk mengidentifikasi data setelah dilakukan proses meta data dan setelah data diintergrasikan pada Google Map API. Dari hasil tersebut data disimpan dalam database, kemudian didapat hasil query dari data spasial dan non spasial. Dalam sistem ini akan menggunakan database MySQL.

c. Framework

Dalam pengembangan sistem, framework digunakan untuk menambah fungsional sistem serta memperbaiki design. Dalam sistem ini, berdasarkan pengembangan MVC. Model sebagai data aplikasi, View sebagai user interface, dan Controller sebagai penghubung antara model dan view dengan menangani permintaan dan menyampaikan data dari model ke view.

d. Akusisi Data

Tahap ini berfungsi berfungsi untuk mengambil, mengumpulkan dan menyiapkan data, hingga memprosesnya untuk menghasilkan data yang dikehendaki.

3.4 Pengujian dan Evaluasi Sistem

Tahap ini menganalisis hasil sistem yang telah dibuat. Pengujian dikatakan berhasil jika sistem dan fungsinya dapat berjalan dengan baik. Jika terdapat fungsi dari sistem yang tidak dapat berjalan sebagaimana mestinya maka dari analisis ini dapat membantu kekurangan sistem tersebut.

3.5 Kesimpulan dan Saran

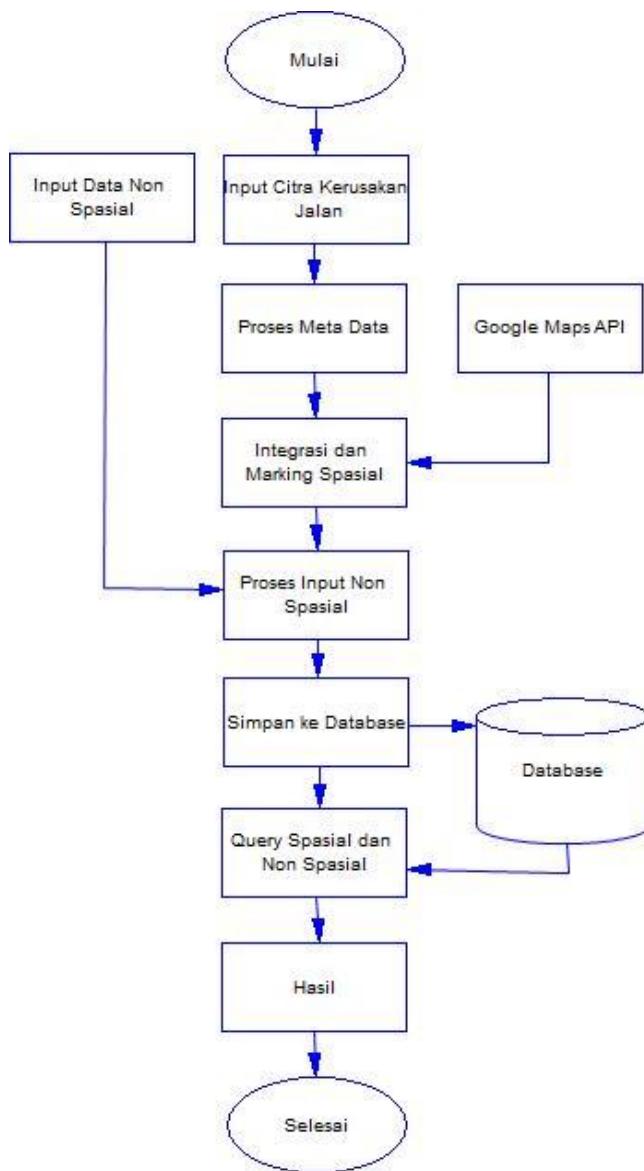
Setelah dilakukan pengujian sistem maka dapat dilakukan penarikan kesimpulan dari hasil pengujian yang telah dilakukan. Selain itu, diberikan saran terhadap kekurangan pada sistem yang dapat digunakan untuk pengembangan penelitian di masa yang akan datang.

3.6 Penyusunan Tugas Akhir

Pada tahap terakhir penelitian ini, dilakukan penyusunan laporan dari proses atau tahapan yang sudah dilakukan.

3.7 Diagram Alir Sistem

Berikut adalah diagram alir dari sistem yang akan ditunjukkan pada Gambar 3.1 berikut :



Gambar 3.1 Diagram Alir Sistem

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan rancangan yang digunakan sebagai acuan untuk implementasi sistem. Perancangan implementasi menggambarkan proses rancang bangun secara terperinci dari awal tahap pengumpulan data hingga implementasi aplikasi

4.1 Deskripsi Sistem

Aplikasi *Geographic Information System* berbasis web dibuat untuk menampilkan titik-titik lokasi kerusakan jalan di Kota Surabaya yang sesuai dengan input user. Aplikasi ini dapat melakukan proses spasial untuk menghasilkan data spasial. Aplikasi ini membuat peletakkan lokasi kerusakan jalan di Surabaya menjadi lebih jelas dan tepat sasaran. Aplikasi ini juga dapat menampilkan data spasial untuk pencarian lokasi-lokasi kerusakan jalan.

4.2 Analisis Kebutuhan Sistem

Saat ini Bina Marga Surabaya menyimpan berbagai data kerusakan jalan dalam bentuk Ms. Excel. Data-data tersebut banyak digunakan untuk pencacatan data biasa sebagai data historis sehingga data yang ada kurang dimanfaatkan dengan baik. Kelemahan data dari penyimpanan data kerusakan jalan di Bina Marga Surabaya saat ini yaitu *report* yang dihasilkan dari data tersebut hanya berisi lokasi yang berupa nama jalan, volume lubang, kategori, dan rayon sehingga dari data yang tersimpan di Bina Marga Surabaya maka sulit untuk menemukan lokasi kerusakan jalan. Selain itu untuk pelaporan kerusakan jalan yang didapat oleh Bina Marga berdasarkan dari laporan masyarakat melalui telepon ataupun laporan masyarakat dari radio Suara Surabaya.

Dari hasil diskusi dengan pihak terkait yaitu Bina Marga Surabaya, maka dibutuhkan sebuah sistem yang dapat mengatasi permasalahan tersebut, yaitu:

1. Sistem mampu menerima input berupa data-data foto yang terdiri dari data beserta meta datanya dan data kerusakan jalan.
2. Sistem mampu digunakan untuk manajemen data-data master seperti *update*, *deleted* dan juga verifikasi data.
3. Sistem mampu menyajikan data kerusakan jalan dalam bentuk spasial dan non spasial. Dengan menggunakan Google Maps API dapat mengetahui titik-titik lokasi kerusakan jalan yang ada di Surabaya.
4. Sistem mampu digunakan untuk pencarian berdasar data kerusakan jalan dan radius di persekitaran landmark yang dipilih.
5. Mampu melakukan pelaporan kerusakan jalan dengan memanfaatkan data-data spasial dan non spasial dari kerusakan jalan. Data spasial yang digunakan berupa citra dari jalan yang mengalami kerusakan, sedangkan untuk data non spasial berupa identitas dari pelapor berupa nama, alamat, dan no.hp.

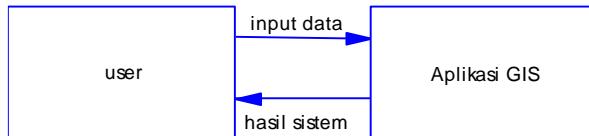
4.3 Perancangan Sistem

Perancangan sistem merupakan persiapan rancang dari implementasi yang bertujuan untuk memberikan gambaran sistem dan rancang bangun yang lengkap kepada user. Desain sistem berupa gambaran perencanaan dan pembuatan sketsa pembuatan sistem. Perancangan sistem yang akan dibahas pada laporan Tugas Akhir ini adalah desain proses, desain arsitektur sistem, desain data, dan desain interface. Berikut akan dibahas lebih lanjut mengenai desain sistem ini.

4.3.1 Desain Proses

Untuk lebih memahami tahapan dan implementasi sistem maka akan dijelaskan alur proses dan data dengan menggunakan pemodelan Data Flow Diagram (DFD) dan *Usecase Diagram*. Berikut penjelasan masing-masing proses.

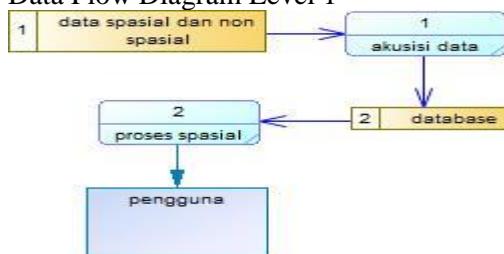
1. Data Flow Diagram Level 0



Gambar 4. 1 DFD Level 0

Proses utama dari sistem ini digambarkan pada Gambar 4.1 yang menunjukkan DFD Level 0. Sebagai input, pengguna memasukkan data yang akan diproses dalam aplikasi dan sebagai output dari aplikasi pengguna akan mendapatkan informasi berupa sistem informasi geografis. Data yang digunakan dalam sistem ini berupa data spasial dan non spasial.

2. Data Flow Diagram Level 1

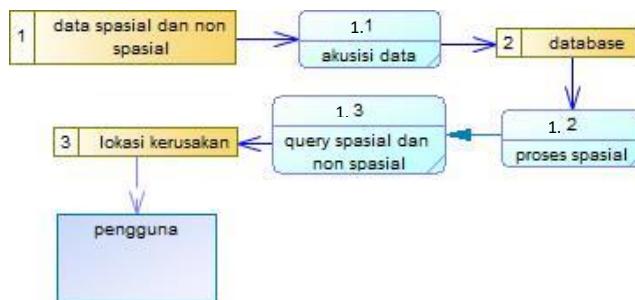


Gambar 4. 2DFD Level 1

Proses dimulai dengan penginputan data spasial dan non spasial oleh pengguna/useryang kemudian data

diakusisi dengan bantuan Google Maps. Lalu data akan disimpan dalam database. Setelah data disimpan pada database akan dilakukan proses query spasial dan non spasial untuk mengetahui lokasi kerusakan jalan sebagai hasil dari sistem ini.

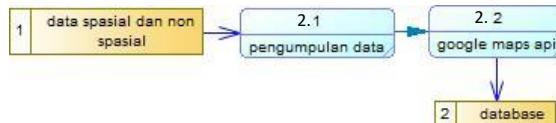
3. Data Flow Diagram Level 2



Gambar 4. 3 DFD Level 2

Pada DFD Level 2 ini terdapat 3 proses yaitu proses akusisi data, proses spasial, dan query spasial dan non spasial. Pada proses query spasial dan non spasial data yang didapatkan adalah titik longitude, titik latitude serta alamat yang nantinya akan diketahui lokasi dari kerusakan jalan. Untuk proses spasial dan akusisi data akan dijelaskan sebagai berikut :

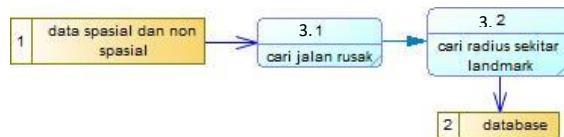
- a) DFD Level 2 untuk akusisi data



Gambar 4. 4 DFD Level 2 Akusisi Data

Pada proses ini data spasial dan non spasial yang telah didapat akan diintegrasikan ke google maps yang kemudian disimpan kedalam database. Data spasial yang digunakan pada sistem ini adalah citra kerusakan jalan, sedangkan untuk data non spasial berupa informasi-informasi mengenai jalan yang mengalami kerusakan.

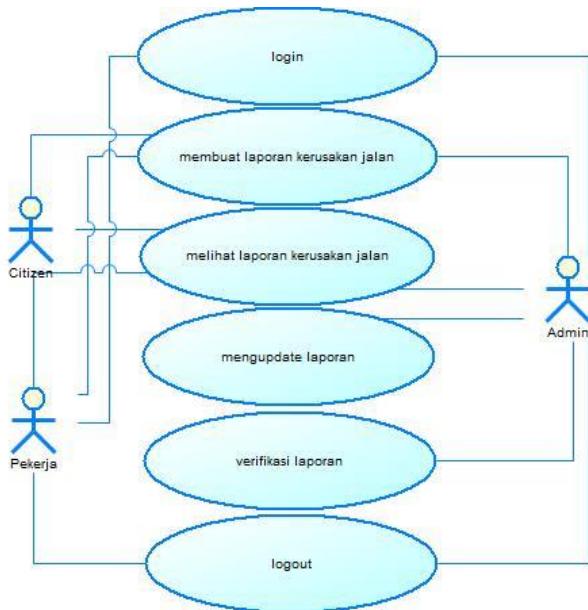
b) DFD Level 2 untuk proses spasial



Gambar 4. 5 DFD Level 2 Proses Spasial

Pada proses ini data akan diproses secara spasial. Dari data spasial akan dicari lokasi kerusakan jalan berdasarkan nama jalannya. Lalu akan dicari jarak radius di sekitaran landmark yang dipilih. Setelah itu didapat lokasi kerusakan jalan didekat landmark beserta jaraknya. Setelah proses tersebut data akan disimpan dalam database.

4. Usecase Diagram



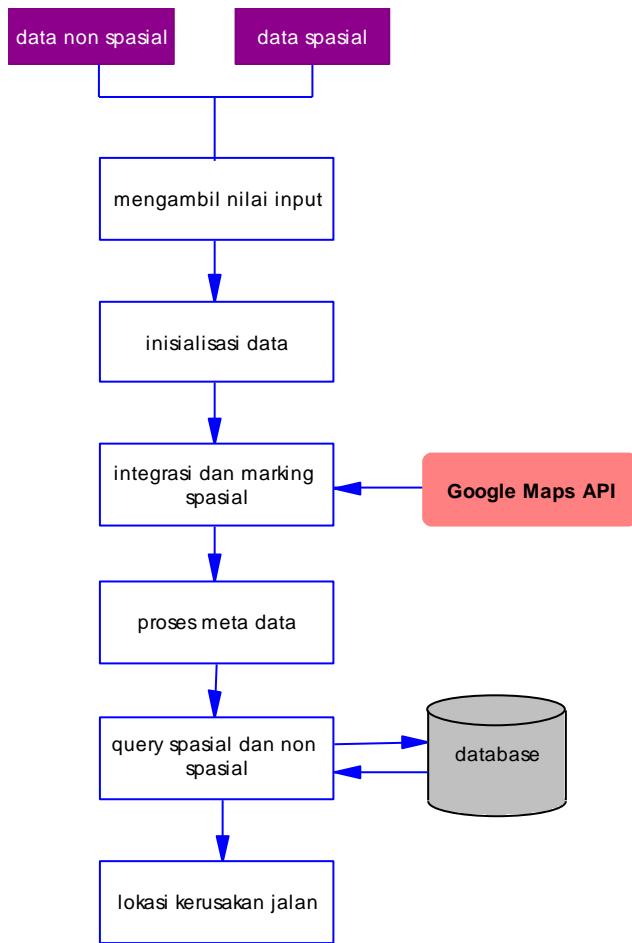
Gambar 4. 6Usecase Diagram

Dalam sistem ini nantinya terdapat tiga *actor* yang akan berperan sebagai *citizen*, admin dan pekerja. Untuk *citizen* dapat mengakses peta yang menampilkan titik-titik lokasi kerusakan jalan dan informasi kerusakan jalan dalam bentuk tabel. Selain itu juga dapat mencari titik-titik lokasi kerusakan berdasarkan nama jalan dan juga berdasarkan radius yang berada disekitaran landmark yang dipilih. Selain itu juga *citizen* dapat melakukan input kerusakan jalan dan harus menyertakan citra/gambar yang diambil menggunakan perangkat yang terhubung dengan GPS. Sedangkan untuk administrator dapat melakukan hal-hal yang dilakukan *citizen* dan juga melakukan

pembaharuan atau *update* pada data yang telah diinput. Selain itu, sebagai admin juga dapat menghapus laporan dari data yang telah diinput sebelumnya dan melakukan verifikasi terhadap laporan yang diinput oleh *citizen* yang kemudian laporan tersebut dapat ditampilkan di peta. Sedangkan untuk pekerja dapat masuk ke dalam sistem dan melakukan kegiatan yang dilakukan oleh *citizen*.

4.3.2 Desain Arsitektur Sistem

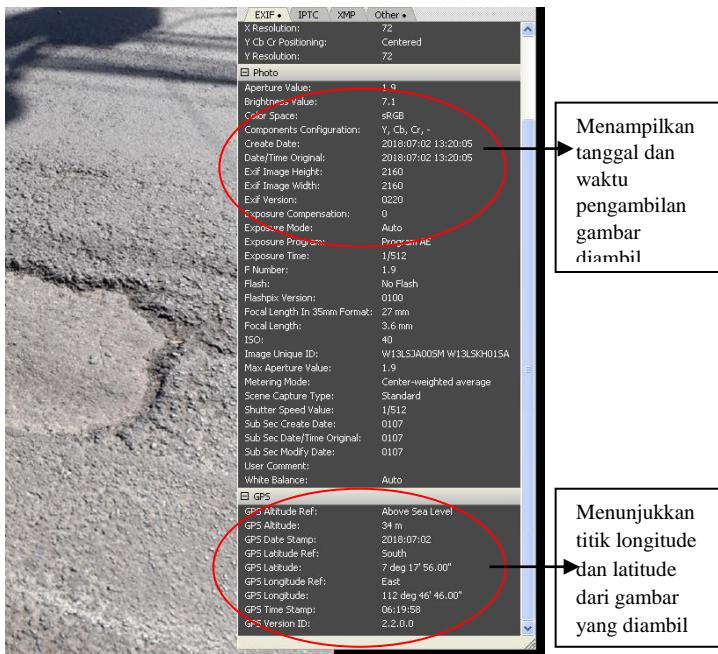
Desain arsitektur sistem adalah desain keseluruhan sistem (termasuk sistem input dan output) yang merupakan bagian dari keseluruhan jalannya sistem. Berikut ini adalah gambaran proses berjalananya sistem yang akan dibuat :



Gambar 4. 7 Desain Arsitektur Sistem

Data input berupa data spasial dan non spasial yang diinput oleh user. Data dari user akan dimasukkan ke dalam sistem untuk dilakukan inisialisasi data. Data spasial yang akan diinput oleh user merupakan citra dari jalan yang mengalami

kerusakan dengan menggunakan perangkat yang terhubung dengan GPS yang kemudian akan diintegrasikan ke Google Maps API. Setelah proses integrasi dan marking spasial akan dilakukan proses meta data. Dari proses meta data akan didapatkan informasi *geolocation* dari jalan yang mengalami kerusakan. Setelah melalui proses meta data dihasilkan query data spasial dan data non spasial yang kemudian disimpan dalam database. Dari hasil query data spasial dan non spasial akan didapatkan lokasi kerusakan jalan yang akan ditampilkan di peta dengan informasi yang meliputi titik longitude, titik latitude, alamat, waktu dan tanggal pengambilan gambar.

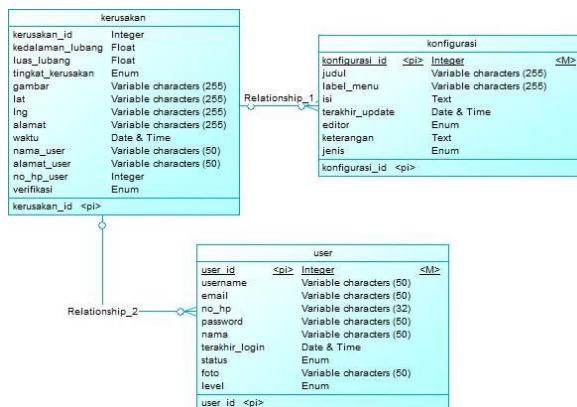


Gambar 4. 8 Pengambilan Meta Data

4.3.3 Desain Data

Data yang digunakan adalah citra dari jalan yang mengalami kerusakan kemudian diintegrasikan ke google maps sebagai data spasial, dan data non spasial yang berisi identitas pelapor seperti nama, alamat, dan no.hp. Semua sumber data dikumpulkan dan disiapkan terlebih dahulu dalam proses akuisisi data. Lalu data yang telah siap akan disimpan dalam database. Data pada database diambil setelah melakukan input oleh *user*. Setelah itu data akan melalui proses metadata dan menghasilkan titik-titik lokasi kerusakan jalan.

Desain database dalam aplikasi ini akan dijelaskan dalam bentuk *Conceptual Data Model* (CDM). CDM digunakan untuk menggambarkan secara detail struktur basis data dalam bentuk logic. Pada CDM belum tergambar jelas bentuk tabel-tabel penyusun basis data beserta field-field yang terdapat pada setiap tabel. Gambaran CDM (*Conceptual Data Model*) yang dipakai dalam sistem ini ditunjukkan sebagai berikut:



Gambar 4. 9 Conceptual Data Model

Akuisisi data adalah proses persiapan sebelum data dimasukkan ke dalam database. Pada akuisisi data dilakukan proses persiapan data sehingga menghasilkan data yang siap untuk dimasukkan ke dalam database. Tabel-tabel yang digunakan dalam sistem ini akan dijelaskan sebagai berikut :

1. Tabel kerusakan adalah tabel yang berisi field-field yang diantaranya yaitu kerusakan_id, kedalaman_lubang, luas_lubang, tingkat_kerusakan, gambar, lat, lng, alamat, waktu, nama_user, alamat_user, no_hp_user, verifikasi.
2. Tabel konfigurasi merupakan tabel yang menampung data mengenai informasi tempat yang diintegrasikan dengan google maps. Kolom-kolom nya adalah konfigurasi_id, judul, label_menu, isi, terakhir_update, editor, keterangan, jenis
3. Tabel user merupakan tabel yang menampung data admin. Field-field pada tabel user yaitu user_id, username, email, password, nama, terakhir_login, status, foto, level.

Keterangan dari struktur masing-masing tabel akan dijelaskan sebagai berikut :

1. Tabel kerusakan

Tabel kerusakan adalah tabel yang memuat data kerusakan jalan

Tabel 4.1Tabel Kerusakan

No	Atribut	Tipe Data	Keterangan
1	kerusakan_id	Int(11)	kode

			kerusakan
2	kedalaman_lubang	Float	Kedalaman lubang
3	luas_lubang	Float	Luas kerusakan jalan
4	tingkat_kerusakan	Enum('Ringan', 'Sedang', 'Parah')	Tingkat kerusakan jalan
5	gambar	Varchar(255)	Citra kerusakan jalan
6	latitude	Varchar(255)	Titik latitude data spasial
7	longitude	Varchar(255)	Titik longitude data spasial
8	alamat	Varchar(255)	Lokasi kerusakan jalan
9	waktu	Datetime	Waktu pengambilan gambar
10	nama_user	Varchar(50)	Nama

			pelapor
11	Alamat_user	Varchar(50)	Alamat pelapor
12	No_hp_user	Int(13)	No. HP pelapor
13	Verifikasi	Enum('Y','N')	Untuk verifikasi laporan oleh admin

2. Tabel konfigurasi

Tabel konfigurasi adalah tabel yang berisi mengenai informasi tempat-tempat yang ada pada map

Tabel 4.2Tabel konfigurasi

No	Atribut	Tipe Data	Keterangan
1	konfigurasi_id	Int(11)	kode konfigurasi
2	judul	Varchar(255)	Kedalaman lubang
3	label_menu	Varchar(255)	Luas kerusakan jalan
4	isi	Text	Tingkat

			kerusakan jalan
5	terakhir_update	Datetime	Update terakhir yang dilakukan
6	editor	Enum('Y', 'N')	Titik latitude data spasial
7	keterangan	Text	Titik longitude data spasial
8	jenis	Enum('textarea', 'text', 'file')	Lokasi kerusakan jalan

3. Tabel user

Tabel user adalah tabel yang berisi data mengenai admin

Tabel 4.3 Tabel user

No	Atribut	Tipe Data	Keterangan
1	user_id	int(11)	kode user
2	username	Varchar(50)	Nama user untuk login ke sistem

3	email	Varchar(50)	Email user
4	No_hp	Int(13)	No. hp user
5	password	Varchar(32)	Password user
6	nama	Varchar(50)	Nama user
7	terakhir_login	Datetime	Login terakhir yang dilakukan
8	status	Enum('blokir', 'aktif', 'temporary')	Status user
9	foto	Varchar(100)	Foto user
10	Level	Enum('Administrator', 'Pekerja')	Bagian kerja dalam sistem

4.3.4 Desain Interface

Desain Interface aplikasi GIS berbasis web ini terdiri dari beberapa form, yaitu form utama yang berisi menu utama program beserta tampilan peta untuk menampilkan informasi titik-titik kerusakan jalan, form untuk input data kerusakan jalan, dan form lainnya. Untuk menu utama akan ditampilkan sebagai berikut :

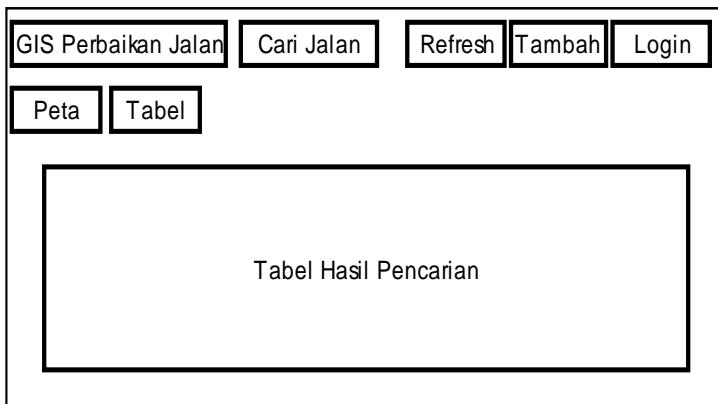


Gambar 4. 10 Desain Menu Utama

Pada menu aplikasi GIS Perbaikan Jalan terdiri dari perintah-perintah untuk menuju form lainnya. Menu tersebut antara lain pencarian titik lokasi kerusakan jalan, tambah untuk input data kerusakan jalan, refresh untuk memperbarui tampilan peta, dan login untuk admin mengelola informasi dari data kerusakan data yang telah diinput oleh user.

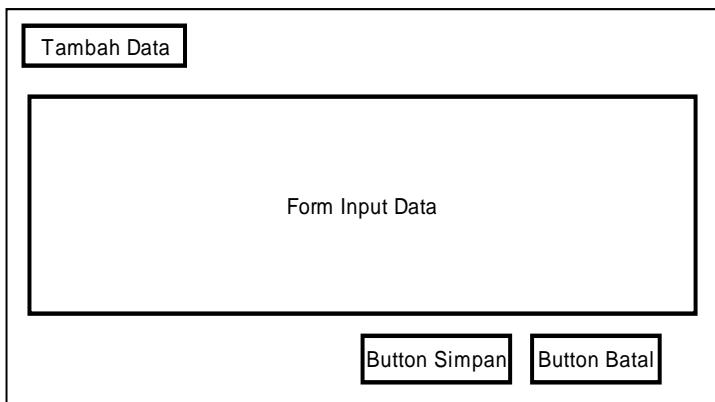


Gambar 4. 11 Desain Menu Pencarian Jalan

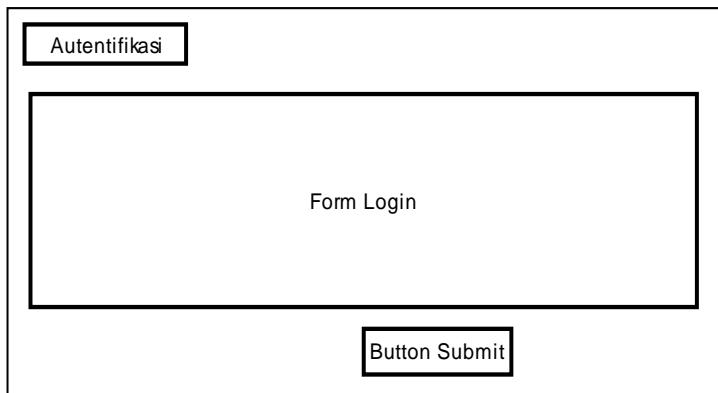


Gambar 4. 12 Desain Tabel Pencarian Jalan

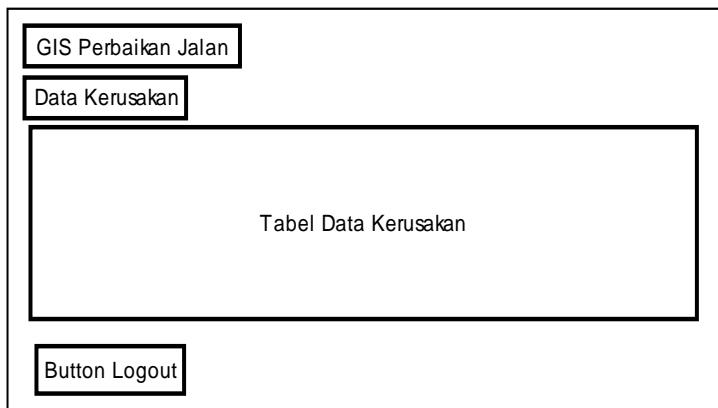
Untuk menu input data digambarkan sebagai berikut :



Gambar 4. 13 Desain Menu Input Data



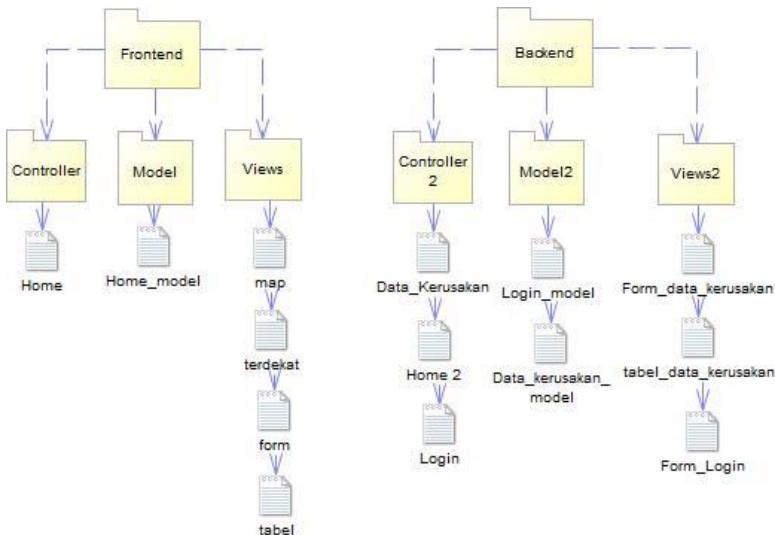
Gambar 4. 14 Desain Menu Login



Gambar 4. 15 Desain Tabel Data Kerusakan

4.3.4 Desain Arsitektur MVC

Desain MVC aplikasi GIS berbasis web ini terdiri dari beberapa file yang dibagi menjadi bagian frontend dan backend. Berikut adalah gambaran desain arsitektur MVC pada aplikasi ini :



Gambar 4. 16 Desain Arsitektur MVC

Pembuatan sistem ini dengan menggunakan framework CodeIgniter 3.1.7. Bagian pada aplikasi ini terdiri dari 2 bagian yaitu bagian frontend dan backend. Bagian frontend untuk menampilkan sedangkan bagian backend untuk bagian manajemen. Pada bagian frontend terdapat controller, model, dan view. Bagian controller mengatur bagian Home pada aplikasi, Model untuk merespon dalam pengelolaan data pada Home, sedangkan untuk view terdapat file untuk tampilan halaman map, radius, form, dan tabel. Pada bagian backend juga terdapat controller, model, dan views. Untuk bagian controller memproses data kerusakan, Home2, dan login untuk admin. Sedangkan bagian model untuk mengelola data pada login admin dan data kerusakan. Lalu bagian views terdapat file untuk tampilan form data kerusakan, tabel data kerusakan, dan form login.

Bagian	File	Kegunaan
Controller (frontend)	home	Untuk menampilkan halaman home pada aplikasi
Model (frontend)	home_model	Untuk merespon pengelolaan data kerusakan pada halaman home
Views (frontend)	Main	Untuk tampilan halaman utama
	Form	Tampilan untuk table data kerusakan jalan pada home
	table	tampilan untuk table data kerusakan pada halaman home
	map	tampilan peta pada halaman home
	terdekat	untuk tampilan halaman yang menunjukkan lokasi di persekitaran landmark
Controller (backend)	Data_kerusakan	Untuk menampilkan data kerusakan pada halaman admin
	Login	Untuk menampilkan halaman login untuk admin
	Logout	Logout untuk admin dan kembali ke halaman untuk login
Model (backend)	Data_kerusakan_mode 1	Untuk merespon dan mengelola data

Model (backend)		kerusakan pada halaman admin
	Login_model	Untuk merespon dan menangkap username dan password ketika admin login ke dalam system
Views (backend)	Form	Untuk tampilan form data kerusakan pada halaman admin
	Table	Untuk tampilan data kerusakan dalam bentuk table pada halaman admin
	Form	Untuk tampilan form login

BAB V

IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK

Pada bab ini akan dibahas mengenai implementasi dan uji coba perangkat lunak aplikasi GIS berbasis web untuk sistem pelaporan dan monitoring keruakan jalan yang telah didesain sebelumnya. Pembahasan terdiri dari ruang lingkup implementasi, implementasi perangkat lunak, uji coba perangkat lunak.

5.1 Ruang Lingkup Implementasi

5.1.1. Hardware

Perangkat keras yang digunakan dalam implementasi perangkat lunak adalah :

1. Processor Core i3
2. Memory RAM 2 Gb
3. Harddisk 500 Gb

5.1.2. Software

Perangkat lunak yang digunakan dalam implementasi perangkat lunak adalah :

1. PHP sebagai bahasa pemrograman untuk antarmuka aplikasi
2. MySql sebagai pengelolaan DBMS
3. Google Maps API untuk kebutuhan pemetaan dan data spasial
4. CodeIgniter sebagai framework

5.2 Implementasi Perangkat Lunak

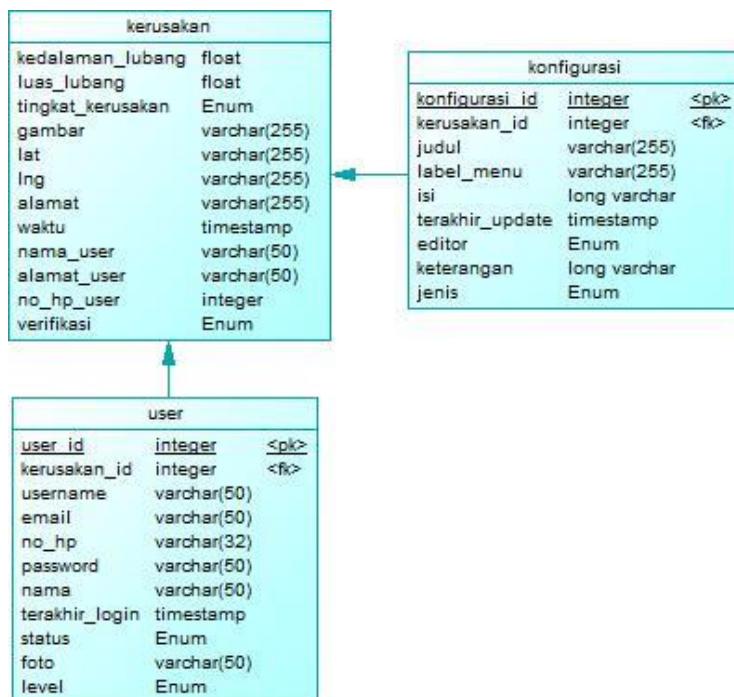
5.2.1. Tahap Persiapan

Pada tahap ini akan dilakukan persiapan data yang terdiri dari data spasial dan non spasial. Sebelum melakukan

implementasi database dan program, pada tahap persiapan ini akan dilakukan pemindaian datake dalam DBMS mysql yang dipakai sebagai implementasi.

5.2.2. Implementasi Database

Implementasi database adalah proses pembuatan tabel-tabel yang diperlukan pada database.



Gambar 5. 1Relasi Database gis_perbaikan_jalan

1. Configurasi database pada framework CodeIgniter

```

$db['default'] = array(
    'dsn'           => '',
    'hostname'     =>
    'localhost',
)

```

```

        'username'          =>
'root',
        'password'         => '',
        'database'         =>
'gis_perbaikan_jalan',
        'dbdriver'         =>
'mysqli',
        'dbprefix'         => '',
        'pconnect'         => FALSE,
        'db_debug'         =>
(ENVIRONMENT !== 'production'),
        'cache_on'         => FALSE,
        'cachedir'         => '',
        'char_set'         =>
'utf8',
        'dbcollat'         =>
'utf8_general_ci',
        'swap_pre'          => '',
        'encrypt'           => FALSE,
        'compress'          => FALSE,
        'stricton'          => FALSE,
        'failover'          =>
array(),
        'save_queries'      => TRUE
);

```

2. Pembuatan tabel kerusakan pada database gis perbaikan jalan

```

CREATE TABLE `kerusakan` (
    `kerusakan_id` int(11) NOT
NULL,
    `kedalaman_lubang` float NOT
NULL,
    `luas_lubang` float NOT NULL,
    `tingkat_kerusakan` 

```

```

enum('Ringan','Sedang','Parah')
NOT NULL,
`gambar` varchar(255) NOT NULL,
`lat` varchar(255) NOT NULL,
`lng` varchar(255) NOT NULL,
`alamat` varchar(255) NOT NULL,
`waktu` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

3. Pembuatan tabel konfigurasi pada database gis perbaikan jalan

```

CREATE TABLE `konfigurasi` (
`konfigurasi_id`      varchar(100)
NOT NULL,
`judul`    varchar(255)     DEFAULT
NULL,
`label_menu`        varchar(255)
DEFAULT NULL,
`isi`      text,
`terakhir_update`   datetime
DEFAULT NULL,
`editor`   enum('Y','N')    DEFAULT
'N',
`keterangan`      text,
`jenis`    enum('textarea','text','file')
DEFAULT 'text'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

4. Pembuatan tabel user pada database gis perbaikan jalan

```

CREATE TABLE `user` (
`user_id` int(11) NOT NULL,
`username` varchar(50) DEFAULT

```

```

        NULL,
        `email`      varchar(50)      DEFAULT
NULL,
        `password`   varchar(32)      DEFAULT
NULL,
        `nama`       varchar(50)      DEFAULT
NULL,
        `terakhir_login`           datetime
DEFAULT NULL,
        `status`      enum('blokir','aktif','temporary'
)      DEFAULT 'aktif',
        `foto`        varchar(100)     DEFAULT
NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

5.2.3. Implementasi Algoritma

1. Menampilkan peta pada halaman antarmuka

```

<script
src="https://maps.googleapis.com/m
aps/api/js?key=<?=$this->config-
>item('map_api')?>&language=id&lib
raries=drawing,geometry,places&lan
guage=id"></script>
<script>

    var map, infowindow, bounds =
new google.maps.LatLngBounds();

<?php foreach($data as $key =>
$c): ?>
        var marker_<?=$c-
>kerusakan_id?>;
<?php endforeach; ?>

$(document).ready(function() {
    map = new

```

```

google.maps.Map(document.getElemen
tById('map'), {
  center : <?=$this->config-
>item('map_center_coordinate')?>,
  zoom : <?=$this-
>config->item('map_zoom')?>,
  zoomControl : true,
  mapTypeControl : true,
  mapTypeControlOptions: {
    style : google.maps.MapTypeControlStyle.HO
RIZONTAL_BAR,
    position : google.maps.ControlPosition.TOP_RI
GHT
  },
  mapTypeId : google.maps.MapTypeId.ROADMAP,
  scaleControl : true,
  streetViewControl : true,
  rotateControl : true,
});

```

2. Menampilkan meta data (titik longitude, titik latitude, alamat, waktu dan tanggal pengambilan gambar)

```

<?php foreach($data as $key =>
$c): ?>
  marker_<?=$c->kerusakan_id?> =
  new google.maps.Marker({
    position : {lat : <?=$c->lat?>,
      lng : <?=$c->lng?>},
    map : map,
    title : '<?=$c->alamat?>'
  });
  bounds.extend({lat : <?=$c->lat?>,
    lng : <?=$c->lng?>});

```

```

var content_<?=$c->kerusakan_id?>
=
<div style="width: 200px; float: left; text-align: center"><br>
<h5 style="margin: 5px 0px;">Latitude : <b><?=$c-
>lat?></b></h5>
<h5 style="margin: 0px;">Longitude :
<b><?=$c->lng?></b></h5>

<h5 style="margin: 0px;">Waktu :
<b><?=format_tanggal_indonesia($c-
>waktu, true)?></b></h5>
<h5><?=$c->alamat?>
</h5>

<button type="button"
onclick="lihat_sekitar('<?=$c-
>kerusakan_id?>')" class="btn btn-
default btn-xs btn-default btn-
block">
<i class="fa fa-map-marker"></i>
Lihat Sekitar
</button>
</div>
`;
marker_<?=$c-
>kerusakan_id?>.addListener('click
', function() {
if(infowindow)
{
infowindow.close();
}
infowindow = new
google.maps.InfoWindow({content :
content_<?=$c->kerusakan_id?>});
infowindow.open(map, marker_<?=$c-
```

```
>kerusakan_id?>);
    });

<?php endforeach; ?>
```

3. Menampilkan data kerusakan jalan dalam bentuk tabel

```
function data_table()
{
    $param = array(
        'offset'=>      $this->input-
>get('start'),
        'limit'=>       $this->input-
>get('length'),
        'keyword'=>     $this->input-
>get('q')
    );
    $data =           $this->home_model-
>get_data_kerusakan($param)-
>result();

unset($param['offset']);
unset($param['limit']);
$num_rows=         $this->home_model-
>get_data_kerusakan($param)-
>num_rows();

$result = array();
foreach($data as $key => $c)
{
$c->waktu =
format_tanggal_indonesia($c-
>waktu, true);
$c->kedalaman_lubang=           $c-
>kedalaman_lubang . ' cm';
$c->luas_lubang= $c->luas_lubang .
' cm';
```

```

$c->gambar = '<img src="" .  

base_url($c->gambar) .  

class="img-responsive">';  
  

$c->action ='<button  

type="button" data-dismiss="modal"  

class="btn btn-success btn-xs"  

onclick="lihat_kerusakan_'. $c-  

>kerusakan_id . '();">  

<i class="fa fa-map-marker"></i>  

Tunjukkan lokasi  

</button>';  

$result[] = $c;  

}  

$respon = array(  

'recordsTotal' => $num_rows,  

'recordsFiltered' => $num_rows,  

'data'=> $result  

);  

echo json_encode($respon);
}

```

4. Menampilkan pencarian berdasarkan radius

```

function terdekat($id = '')  

{  

if(empty($id)) {  

    show_404();  

}  

$param['info'] = $this-  

>home_model-  

>get_data_kerusakan(array('id' =>  

$id))->row();  

if(empty($param['info'])) {  

    show_404();  

}  
  

$param['terdekat'] = $this-  

>home_model->get_terdekat(@$id,

```

```

@$param['info']->lat,
@$param['info']->lng)->result();
        $this->load-
>view('terdekat', $param);
    }
}

```

5. Implementasi pencarian berdasarkan jalan

```

public function
get_data_kerusakan($param      =
array())
{
    if (!empty($param))
    {
        if (!empty($param['limit']))
        {
            if (!empty($param['offset']))
            {
                $this->db->limit($param['limit'],
$param['offset']);
            }else
            {
                $this->db->limit($param['limit']);
            }
        }
        if (!empty($param['keyword']))
        {
            $this->db->like('alamat',
$param['keyword']);
        }
        if (!empty($param['id'])) {
            $this->db->where('kerusakan_id',
$param['id']);
        }
    }
    $this->db-
>order_by('kerusakan_id', 'DESC');
    $this->db->from('kerusakan');
    $query = $this->db->get();
    return $query;
}

```

```
}
```

6. Implementasi untuk input data

```
function post_kerusakan()
{
    $data_post = $this->input->post();
    $this->form_validation->set_rules('kedalaman_lubang',
        'Kedalaman Lubang', 'required');
    $this->form_validation->set_rules('luas_lubang', 'Luas Lubang', 'required');
    $this->form_validation->set_rules('tingkat_kerusakan',
        'Tingkat Kerusakan', 'required');

    $respon = array();
    if($this->form_validation->run() == false)
    {
        $respon = array(
            'status'      => '201',
            'msg'         =>
            war_msg(validation_errors()),
            'data'        => array()
        );
    }
    else
    if(empty($_FILES['userfile']['tmp_name']))
    {
        $respon = array(
            'status'      => '201',
            'msg'         =>
```

```
'msg'          =>      war_msg('Gambar  
harus disertakan'), => array()  
    );  
}  
else{  
$config['upload_path']           =  
'./uploads/kerusakan_jalan/';  
  
$config['allowed_types']         =  
'jpg|png|jpeg';  
  
// $config['max_size'] = '2048';  
  
if  
(!is_dir($config['upload_path']))  
{  
  
mkdir($config['upload_path']);  
}  
  
$this->load->library('upload',  
$config);  
if  
(!(!$this->upload-  
>do_upload('userfile'))  
{  
$respon = array(  
    'status'=> '201',  
    'msg'  =>      war_msg($this-  
>upload->display_errors()),  
    'data'=> array()  
);  
}  
else{  
$data_upload     =      $this->upload-  
>data();  
$data_post['gambar']           =  
$config['upload_path']         =  
$data_upload['file_name'];  
$exif_gambar        =
```

```
exif_read_data($data_post['gambar']
], 0, true);

        if(empty($exif_gambar['GPS'])
) {
    $respon = array(
        'status'=> '201',
        'msg' => war_msg('Gambar
tidak valid, tidak ditemukan
lokasi gambar (latitude /
longitude kosong).'),
        'data'=> array()
    );
    unlink($data_post['gambar'])
;
}
else {
    $getLocation =
readGPSImage($exif_gambar['GPS']);

    if(empty($getLocation['lat'])
) || empty($getLocation['lng']))
$respon = array(
        'status' => '201',
        'msg' =>
war_msg('Lokasi gambar tidak
valid.'),
        'data' => array()

    );
    unlink($data_post['gambar'])
;
}
else{
    if(empty($exif_gambar['EXIF'
]['DateTimeOriginal']))
    {
        $data_post['waktu'] = date('Y-m-d
H:i:s');
    }
}
```

```
        }
    else{
        $data_post['waktu'] = date('Y-m-d
H:i:s',
        strtotime($sexif_gambar['EXIF']['Da
teTimeOriginal']));
    }
    $data_post['lat'] =
    $getLocation['lat'];
    $data_post['lng'] =
    $getLocation['lng'];
    $proses = $this->home_model-
>insert_data_kerusakan($data_post)
;
    if($proses)
    {
        $data_post['waktu'] =
        format_tanggal_indonesia($data_pos
t['waktu'], true);
        $data_post['gambar']=base_url($data_pos
t['gambar']);
        $data_post['id'] = $this->db-
>insert_id();
        $respon = array(
            'status' => '200',
            'msg' => suc_msg('Data
kerusakan berhasil disimpan.'),
            'data'=> $data_post
        );
    }
    else
    {
        unlink($data_post['gambar']);
        $respon = array(
            'status' => '201',
            'msg' => war_msg('Data
kerusakan gagal disimpan.'),
            'data' => array()
        );
    }
}
```

```
        } }  
    } }  
echo json_encode($respon); }
```

7. Implementasi untuk login admin

```
function submit()  
{  
    $this->form_validation->set_rules('email', 'Email / Username', 'required');  
    $this->form_validation->set_rules('password', 'Password', 'required');  
    if($this->form_validation->run() == false) {  
        $respon = array('status' => '201', 'data' => validation_errors('', ''), 'msg' => 'Login gagal.'); }  
    else  
    {  
        $email = $this->input->post('email');  
        $get_data_login = $this->login_model->get_login($email)->row();  
        if(!empty($get_data_login))  
        {  
            $password = $this->input->post('password');  
            if(md5($password) == $get_data_login->password) {  
                if($get_data_login->status == 'blokir') {
```

```

$respon = array('status' => '201',
    'data' => 'Login gagal, Akun diblokir.', 'msg' => 'Login gagal,
Akun diblokir.');
}
elseif($get_data_login->status == 'aktif')
$respon = array('status' => '200',
    'data'=> 'Login berhasil.',
    'msg' => 'Login berhasil.');

```

8. Menampilkan data kerusakan pada halaman admin

```

function __construct()
{
parent::__construct();
$this->login_status      =      $this-
>session-
>userdata('login_status');
$this->login_uid          =      $this-
>session->userdata('login_uid');
if($this->login_status != 'ok')
{
$this->session-
>set_flashdata('msg',
err_msg('Silahkan login untuk
melanjutkan.'));
redirect(site_url('login'));
}
$this->load-
>model('data_kerusakan_model');
$this->page_active
= 'data_kerusakan';
$this->sub_page_active      =
'data_kerusakan';
}
public function index()
{

```

```
$param['keyword'] = $this->input->get('q');
$limit = 10;
$uri_segment= 3;
$filter = array('limit' => limit,
'offset' => $this->uri->segment($uri_segment), 'keyword'
=> $param['keyword']
);
$param['data'] = $this->data_kerusakan_model->get_data($filter)->result();
unset($filter['limit']);
unset($filter['offset']);
$total_rows = $this->data_kerusakan_model->get_data($filter)->num_rows();
$param['pagination'] =
paging('data_kerusakan/index',
$total_rows, $limit,
$uri_segment);

$param['main_content'] =
'data_kerusakan/table';
$param['page_active'] = $this->page_active;
$param['sub_page_active'] = $this->sub_page_active;
$this->templates->load('main_templates', $param);
}
public function form($id = '')
{
    $param['msg'] = $this->session->flashdata('msg');
    $param['id'] = $id;
    $last_data = $this->session->flashdata('last_data');
    if(!empty($last_data))
```

```

        {
            $param['data'] = (object)
            $last_data;
        }
        else
        {
            if (!empty($id)) {
                $param['data'] = $this-
>data_kerusakan_model-
>get_data_row($id);
            }
        }
        $param['opt_tingkat_kerusakan'] =
array(
    'Ringan' => 'Ringan',
    'Sedang' => 'Sedang',
    => 'Parah'
);

$param['main_content'] =
'data_kerusakan/form';
$param['page_active'] = $this-
>page_active;
$param['sub_page_active'] =
$this->sub_page_active;
    $this->templates-
>load('main_templates', $param);
}

public function submit($id = '')
{
    $data_post = $this->input-
>post();
    $this->form_validation-
>set_rules('kedalaman_lubang',
    'Kedalaman Lubang', 'required');
    $this->form_validation-
>set_rules('luas_lubang', 'Luas
Lubang', 'required');
}

```

```
$this->form_validation-
>set_rules('tingkat_kerusakan',
'Tingkat Kerusakan', 'required');
    if($this->form_validation-
>run() == false) {
$this->session-
>set_flashdata('msg',
err_msg(validation_errors()));
$this->session-
>set_flashdata('last_data',
$data_post);

    redirect('data_kerusakan/for
m/' . $id);
}
else{
if(empty($id)) {
$proses = $this-
>data_kerusakan_model-
>insert($data_post);
if($proses) {
$this->session-
>set_flashdata('msg',
suc_msg('Data berhasil
disimpan.'));
}
else{
$this->session-
>set_flashdata('msg',
err_msg('Data gagal disimpan,
silahkan ulangi lagi.'));
}

    redirect('data_kerusakan/for
m/' . $id);
}
}
else{
$proses = $this-
>data_kerusakan_model-
```

```
>update($data_post, $id);
if($proses) {
    $this->session-
    >set_flashdata('msg',
        suc_msg('Data berhasil
diperbaharui.'));
    }
else{
    $this->session-
    >set_flashdata('msg',
        err_msg('Data gagal diperbaharui,
tidak ada yang berubah.'));
    }
}
redirect('data_kerusakan');
}
}

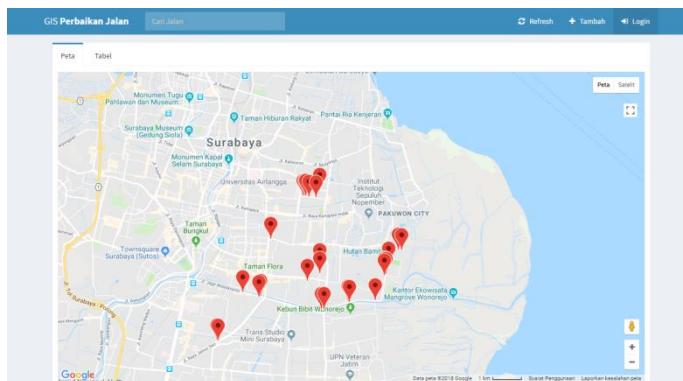
public function hapus($id)
{
    $data = $this-
>data_kerusakan_model-
>get_data_row($id);
    if(empty($data)) {
        show_404();
    }
unlink($data->gambar);

    $proses = $this-
>data_kerusakan_model-
>delete($id);
    $this->session-
    >set_flashdata('msg',
        suc_msg('Data berhasil
dihapus.'));
    redirect('data_kerusakan');
}
}
```

5.2.3 Implementasi Antarmuka

Seperti yang telah dijelaskan pada desain antarmuka, form yang digunakan terdiri dari form halaman utama aplikasi, form pencarian, form input data, dan form login.

Pada bagian halaman utama berfungsi untuk mengakses aplikasi GIS berbasis web. Halaman utama sistem akan langsung mengarah pada sistem informasi untuk lokasi kerusakan jalan. Berikut adalah tampilan form halaman utama :



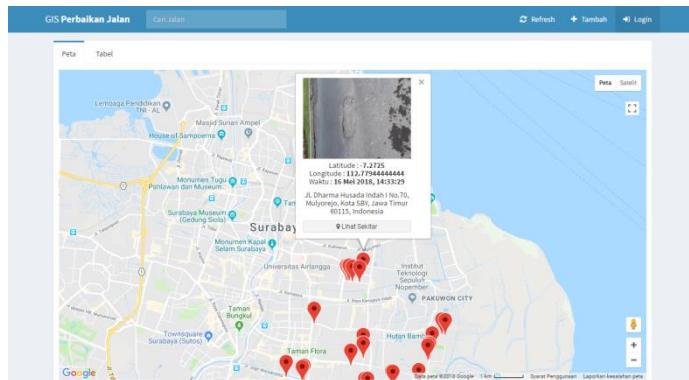
Gambar 5. 2 Tampilan Awal Sistem

Pada halaman utama menampilkan peta yang berisi titik-titik lokasi kerusakan jalan yang ada di Surabaya.

Kedalaman	Lekar	Tingkat	Alamat	Waktu	
6 cm	42 cm	Sedang	Jl. Semolowaru Utara No.31, Semolowaru, Sukolilo, Kota SBK, Jawa Timur 60115, Indonesia	02 Juli 2018, 13:20:05	<button>Unduh Gambar</button>
5 cm	29 cm	Sedang	Jl. Semolowaru Elok No.11, RT.004/RW.03, Semolowaru, Sukolilo, Kota SBK, Jawa Timur 60115, Indonesia	02 Juli 2018, 13:25:46	<button>Unduh Gambar</button>
8 cm	31 cm	Sedang	Jl. Semolowaru Elok No.11, RT.004/RW.03, Semolowaru, Sukolilo, Kota SBK, Jawa Timur 60115, Indonesia	02 Juli 2018, 13:25:58	<button>Unduh Gambar</button>

Gambar 5. 3 Tampilan Tabel Awal Sistem

Pada halaman ini juga terdapat tabel yang berisikan atribut-atribut dari lokasi kerusakan jalan

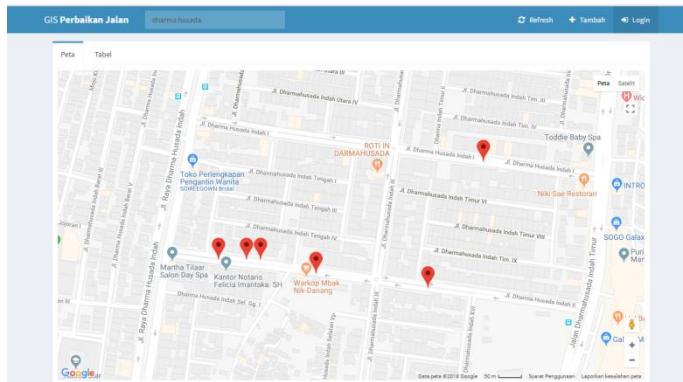


Gambar 5. 4 Tampilan Infomasi Kerusakan Jalan

Selanjutnya adalah halaman yang memuat informasi-informasi kerusakan jalan yang ditampilkan pada halaman utama. Informasi-informasi yang diberikan adalah titik koordinat longitude dan latitude, alamat, tanggal dan waktu pengambilan gambar. Selain itu terdapat *button* untuk melihat

lokasi sekitar yang memungkinkan pengguna dapat mengetahui lokasi jalan yang mengalami kerusakan disekitar landmark dalam radius tertentu.

Pada halaman ini terdapat pencarian lokasi kerusakan jalan. Berikut adalah hasil dari pencarian lokasi dari sistem



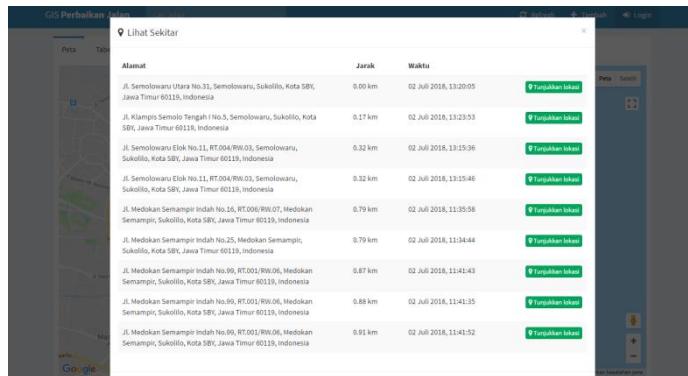
Gambar 5. 5 Tampilan Hasil Pencarian Jalan

Pada halaman ini juga terdapat tabel hasil pencarian lokasi kerusakan jalan yang berisikan atribut-atribut seperti kedalaman, luas, tingkat kerusakan, dan waktu pengambilan gambar.

Peta	Tabel			
Show 5 entries				
Kedalaman	Lebar	Tingkat	Alamat	Waktu
	13 cm	37 cm	Parah Jl. Dharma Husada Indah II Blok D-43 No.12, RT.003/RW.09, Mulyorejo, Kota SBV, Jawa Timur 60115, Indonesia	02 Juli 2018, 13:39:23
	13 cm	30 cm	Parah Jl. Dharma Husada Indah II No.9, Mulyorejo, Kota SBV, Jawa Timur 60115, Indonesia	02 Juli 2018, 13:38:15
	6 cm	47 cm	Sedang Jl. Dharma Husada Indah II Blok C17B No.22, RT.002/RW.09, Mulyorejo, Kota SBV, Jawa Timur 60115, Indonesia	02 Juli 2018, 13:37:01

Gambar 5. 6 Tabel Hasil Pencarian Jalan

Pada halaman berikutnya yaitu halaman pencarian berdasarkan radius. Form ini berfungsi untuk mencari titik lokasi kerusakan jalan yang berada disekitar landmark yang dipilih user dengan jarak dalam satuan km.



Gambar 5. 7 Tampilan Hasil Pencarian Berdasarkan radius

Selanjutnya adalah halaman input data kerusakan jalan. Pada halaman ini user melakukan input informasi jalan yang

rusak dan input citra kerusakan jalan yang nantinya menghasilkan titik lokasi jalan yang rusak. Berikut adalah halaman yang menampilkan form input data kerusakan jalan yang akan diinput oleh user.

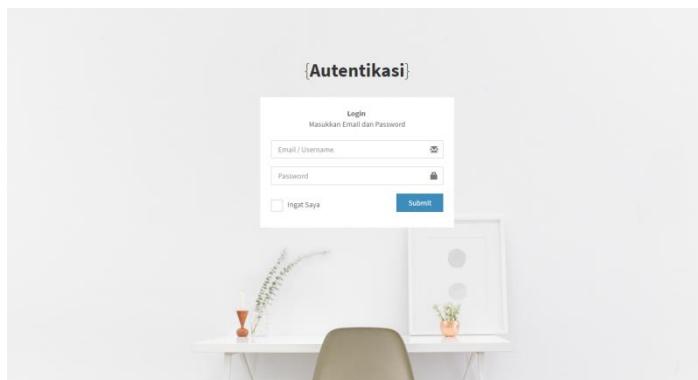
The screenshot shows a web-based application for reporting road damage. On the left, there is a map of Surabaya with a red marker indicating a specific location. To the right of the map is a form with the following fields:

- Gambar *: A file input field with the placeholder "Choose File | No file chosen".
- Alamat: An input field for the address.
- Identitas Pengirim: A section for sender identity.
- Nama *: An input field for the name, marked with a required asterisk.
- Nomor Handphone *: An input field for the phone number, marked with a required asterisk.
- Alamat *: Another input field for the address, marked with a required asterisk.

At the bottom of the form are two buttons: "Simpan" (Save) with a checkmark icon and "Batal" (Cancel) with a cross icon. Below the buttons is a link "Lain kesalahan pada".

Gambar 5. 8 Tampilan Halaman Input Kerusakan Jalan

Pada halaman berikutnya adalah halaman login yang dilakukan oleh admin untuk mengakses data-data kerusakan jalan.



Gambar 5. 9 Tampilan Halaman Login

Selanjutnya adalah halaman yang berisi data-data kerusakan jalan yang dapat diakses oleh admin. Pada halaman ini admin dapat melakukan edit dan menghapus data yang telah ada.

Alamat	Keterangan	Koordinat
Jl. Semolowaru Utara No.31, Semolowaru, Sukolilo, Kota SBY, Jawa Timur 60119, Indonesia	Kedalaman: 6 cm Luas Lubang: 42 cm Tgrt Kerusakan: Sedang	-7.29888888888889, 112.77944444444444
Jl. Semolowaru Elok No.11, RT.004/RW.03, Semolowaru, Sukolilo, Kota SBY, Jawa Timur 60119, Indonesia	Kedalaman: 5 cm Luas Lubang: 29 cm Tgrt Kerusakan: Sedang	-7.30138888888889, 112.77355555555556
Jl. Semolowaru Elok No.11, RT.004/RW.03, Semolowaru, Sukolilo, Kota SBY, Jawa Timur 60119, Indonesia	Kedalaman: 8 cm Luas Lubang: 31 cm Tgrt Kerusakan: Sedang	-7.30138888888889, 112.77355555555556

Gambar 5. 10 Tampilan Tabel Data Kerusakan

5.3 Uji Coba Perangkat Lunak

Pada sub bab ini akan dibahas mengenai uji coba perangkat lunak yang telah diimplementasikan sebelumnya. Perangkat keras yang digunakan dalam uji coba adalah computer dengan processor i3 2,3 GHz, memory RAM 2 Gb, harddisk 500 Mb. Sedangkan perangkat lunak yang digunakan adalah sistem operasi windows 8.1, bahasa pemrograman PGP, mysql sebagai DBMS, google maps API untuk mengintegrasikan peta spasial, dan framework CodeIgniter 3.1.7.

5.3.1 Uji Coba Proses Data Input

Berikut akan dijelaskan tentang uji coba proses data input. Data yang digunakan untuk proses ini adalah data-data kerusakan jalan di Surabaya sebagai data spasial dan data spasial dan non spasial:

a) Skenario uji coba proses

Proses yang diuji coba	Menguji input foto dan data kerusakan jalan
Input	Citra kerusakan jalan, nama pelapor, alamat pelapor, dan no.hp pelapor
Skenario uji coba	Pada uji coba ini user melakukan input foto yang sebelumnya telah diambil menggunakan perangkat yang tersambung GPS dan identitas pelapor berupa nama, alamat, dan no.hp
Hasil yang diharapkan	Diperolehnya titik lokasi kerusakan jalan beserta meta data yang terdapat pada foto yaitu titik longitude, titik latitude, alamat, tanggal dan waktu pengambilan foto.

b) Proses uji coba proses data input

1. Uji coba proses data input diawali dengan proses upload data spasial berupa foto dan data non spasial.

Gambar *
 Choose File No file chosen

Alamat

Identitas Pengirim
 Nama *

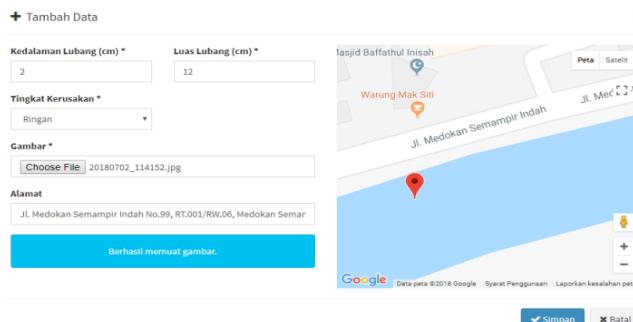
Nomor Handphone *

Alamat *

Simpan Batal

Gambar 5. 11 Input data spasial dan non spasial

2. Uji coba proses data input dilanjutkan dengan didapatkannya titik lokasi kerusakan jalan beserta alamatnya.



Gambar 5. 12 Hasil proses meta data

3. Selanjutnya user dapat melihat data yang telah diinput pada halaman home yang menampilkan titik-titik lokasi kerusakan jalan



Gambar 5. 13 Halaman home

c) Hasil uji coba proses data input

Hasil uji coba proses data input ini dapat dilihat dari didapatkannya titik lokasi kerusakan jalan. Dengan diperolehnya titik tersebut maka meta data yang terdapat dalam foto yang telah diinput oleh user berhasil ditangkap oleh sistem.



Gambar 5. 14 Hasil input data

5.3.2 Uji Coba Manajemen Data

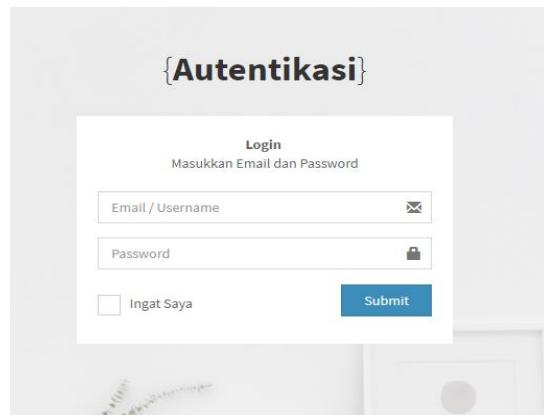
Uji coba selanjutnya adalah manajemen data-data yang telah diinput user oleh admin

- a) Skenario uji coba manajemen data

Proses yang diuji coba	Manajemen data-data yang telah diinput oleh user
Input	Citra kerusakan jalan, nama pelapor, alamat pelapor, dan no.hp pelapor
Skenario uji coba	Pada uji coba ini dilakukan oleh admin dengan login terlebih dahulu kedalam sistem. Setelah login, admin dapat melihat data-data kerusakan jalan yang telah diinput oleh user. Admin dapat memanajemen data dengan mengedit atau menghapus data tersebut. Selain itu juga admin dapat memverifikasi laporan dari user yang kemudian laporan tersebut dapat ditampilkan di peta.
Hasil yang diharapkan	Admin dapat melakukan update, menghapus, dan memverifikasi laporan yang ada

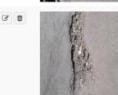
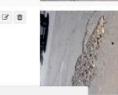
- b) Proses uji coba manajemen data

1. Uji coba diawali dengan login oleh admin terlebih dahulu.



Gambar 5. 15 Login untuk halaman admin

2. Pada tabel yang menampilkan data-data kerusakan yang telah diinput sebelumnya, admin dapat melakukan update atau menghapus data.
3. Hasil dari uji coba ini adalah admin dapat memanajemen data yang telah diinput oleh user.

Alamat	Keterangan	Koordinat
 Jl. Semolowaru Utara No.31, Semolowaru, Sukolilo, Kota SBV, Jawa Timur 60119, Indonesia	Kedalaman : 6 cm Luas Lubang : 42 cm Tgl Kerusakan : Sedang	-7.29888888888888, 112.779444444444
	Kedalaman : 5 cm Luas Lubang : 29 cm Tgl Kerusakan : Sedang	-7.30138888888888, 112.775555555555
	Kedalaman : 8 cm Luas Lubang : 31 cm Tgl Kerusakan : Sedang	-7.30138888888888, 112.775555555555
	Kedalaman : 6 cm Luas Lubang : 57 cm Tgl Kerusakan : Sedang	-7.30111111111111, 112.760555555556

Gambar 5. 16 Data kerusakan pada halaman admin

c) Hasil uji coba manajemen data

Hasil uji coba ini dapat dilihat dengan dapat dilakukannya proses update dan delete oleh admin.

Kedalaman Lubang (cm) *

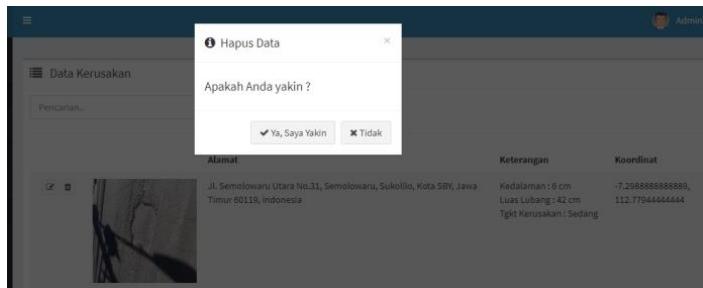
Luas Lubang (cm) *

Tingkat Kerusakan *

Alamat

Gambar

Gambar 5. 17 Proses update oleh admin



Gambar 5. 18 Proses delete oleh admin

5.3.3 Uji Coba Menampilkan Data Spasial dan Non Spasial

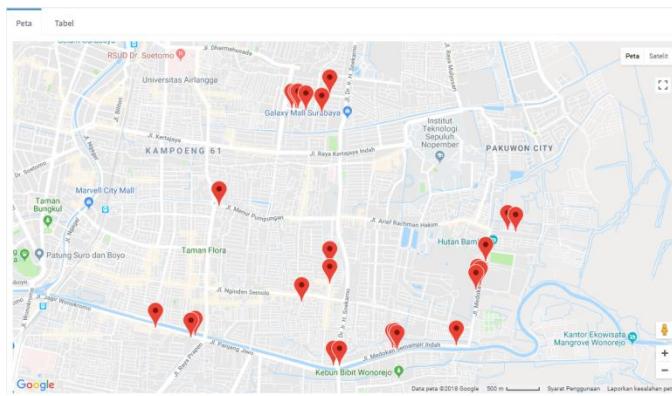
Uji coba selanjutnya adalah menampilkan data spasial dan non spasial

a) Skenario uji coba manajemen data

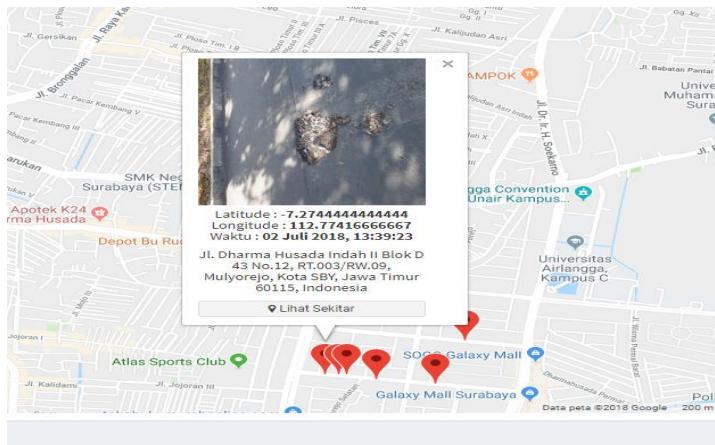
Proses	yang	Menampilkan data spasial dan non
--------	------	----------------------------------

diuji coba	spasial yang telah diinput user
Input	Citra kerusakan jalan, nama pelapor, alamat pelapor, dan no.hp pelapor
Skenario uji coba	Pada uji coba ini data spasial dan non spasial yang telah diinput user akan ditampilkan. Data yang akan ditampilkan dalam bentuk peta yang diintegrasikan dengan google maps API.
Hasil yang diharapkan	Data akan ditampilkan dalam bentuk peta dengan titik-titik lokasi kerusakan jalan beserta informasi mengenai jalan tersebut.

- b) Hasil uji coba menampilkan data spasial dan non spasial
- Hasil uji coba ini dapat dilihat dengan dapat ditampilkannya data spasial dan non spasial dalam bentuk peta beserta informasi mengenai kerusakan jalan. Informasi tersebut berupa titik longitude, titik latitude, alamat, tanggal dan waktu pengambilan foto.



Gambar 5. 19 Tampilan data dalam peta



Gambar 5. 20 Tampilan dengan informasi meta data

Kedalaman	Lebar	Tingkat	Alamat	Waktu	
6 cm	42 cm	Sedang	Jl. Semolowaru Utara No.31, Semolowaru, Sukolilo, Kota SBY, Jawa Timur 60119, Indonesia	02 Juli 2018, 13:20:05	
5 cm	29 cm	Sedang	Jl. Semolowaru Elok No.11, RT.004/RW.03, Semolowaru, Sukolilo, Kota SBY, Jawa Timur 60119, Indonesia	02 Juli 2018, 13:15:46	
8 cm	31 cm	Sedang	Jl. Semolowaru Elok No.11, RT.004/RW.03, Semolowaru, Sukolilo, Kota SBY, Jawa Timur 60119, Indonesia	02 Juli 2018, 13:15:36	
6 cm	57 cm	Sedang	Jl. Jagir Wonokromo No.303, RT.011/RW.01, Panjang Jivo, Tenggils Mejoyo, Kota SBY, Jawa Timur 60299, Indonesia	02 Juli 2018, 12:55:50	

Gambar 5. 21 Tampilan data dalam tabel

5.3.4 Uji Coba Pencarian Jalan

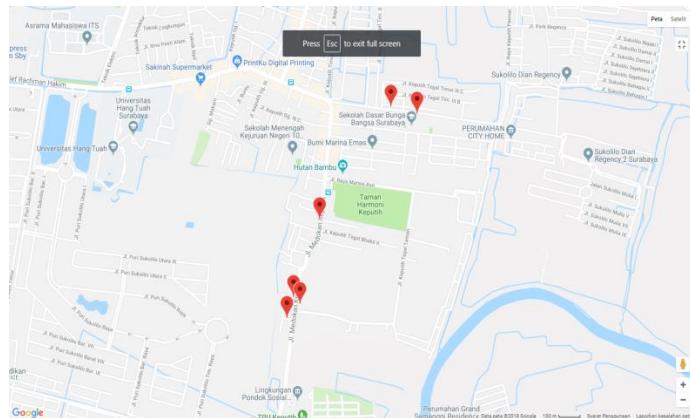
Uji coba selanjutnya adalah pencarian jalan dari hasil input yang telah dilakukan user

- a) Skenario uji coba manajemen data

Proses yang diuji coba	Pencarian lokasi kerusakan jalan berdasarkan data kerusakan dan daerah persekitaran
Input	Citra kerusakan jalan, nama pelapor, alamat pelapor, dan no.hp pelapor
Skenario uji coba	Proses ini dilakukan dengan mencari lokasi kerusakan jalan berdasarkan data kerusakan jalan dan daerah persekitaran. Kemudian user dapat memilih titik-titik tempat yang ada pada google maps (landmark) sesuai keinginan user.
Hasil yang diharapkan	Diperolehnya informasi mengenai lokasi kerusakan dengan lebih rinci.

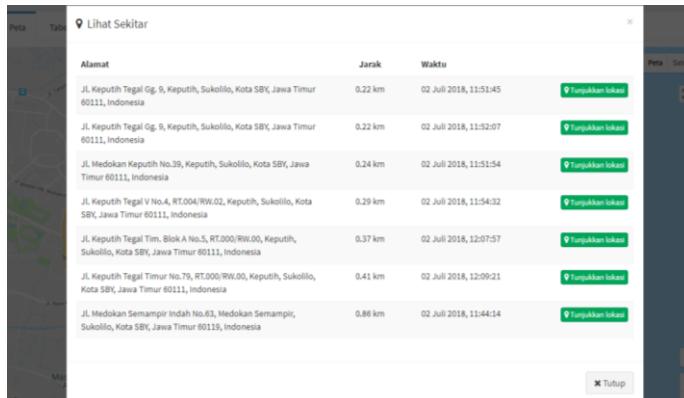
	Contohnya adalah didapat informasi lokasi kerusakan jalan pada suatu daerah atau dikawasan tertentu. Kemudian juga didapat lokasi kerusakan jalan berdasarkan landmark yang dipilih oleh user dalam radius tertentu.
--	--

- b) Hasil uji coba menampilkan data spasial dan non spasial
- Uji coba pertama adalah dengan melakukan pencarian berdasarkan nama jalan. Tampak titik-titik lokasi kerusakan jalan.



Gambar 5. 22 Hasil pencarian jalan

Uji coba kedua dilakukan pencarian berdasarkan daerah persekitaran. Dalam uji coba ini dilakukan pembuatan daerah persekitaran dengan radius 1 kilometer dari salah satu titik di lokasi kerusakan jalan.



Gambar 5. 23 Hasil pencarian jalan berdasarkan radius

5.3.5 Hasil Uji Coba Perangkat Lunak

Setelah melakukan beberapa uji coba terhadap perangkat lunak yang telah dibuat, mulai dari proses data input sampai pencarian jalan, hasil dari beberapa uji coba dapat ditabelkan sebagai berikut :

No.	Proses yang diujikan	Hasil yang diharapkan	Status
1	Proses data input	Diperolehnya titik lokasi kerusakan jalan beserta meta data yang terdapat pada foto yaitu titik longitude, titik latitude, alamat, tanggal dan waktu pengambilan foto.	OK
2	Manajemen data	Admin dapat melakukan update, menghapus, dan memverifikasi laporan yang ada	OK
3	Menampilkan data spasial	Data akan ditampilkan dalam bentuk peta dengan	OK

	dan non spasial	titik-titik lokasi kerusakan jalan beserta informasi mengenai jalan tersebut	
4	Pencarian jalan	Diperolehnya informasi mengenai lokasi kerusakan dengan lebih rinci. Contohnya adalah didapat informasi lokasi kerusakan jalan pada suatu daerah atau dikawasan tertentu. Kemudian juga didapat lokasi kerusakan jalan berdasarkan landmark yang dipilih oleh user dalam radius tertentu.	OK

BAB VI

PENUTUP

Pada bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan dan saran yang dapat digunakan jika penelitian ini dikembangkan.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian program, maka dapat diambil kesimpulan sebagai berikut :

1. Sistem pelaporan dan monitoring kerusakan jalan berbasis web menggunakan MVC berhasil diwujudkan untuk mengetahui lokasi-lokasi kerusakan jalan yang terdapat di Surabaya dan tingkat kerusakannya.
2. Pada sistem ini juga terdapat fitur pencarian untuk memudahkan mencari lokasi kerusakan jalan berdasarkan nama jalan dan berdasarkan radius disekitaran landmark yang dipilih.

6.2 Saran

Ada beberapa hal yang penulis sarankan untuk pengembangan penelitian selanjutnya :

1. Data yang digunakan pada aplikasi ini citra kerusakan jalan. Sehingga pada penelitian selanjutnya dapat menambah data yang dikelola pada aplikasi ini.
2. Pada tugas akhir ini hanya terdapat fitur pencarian berdasarkan nama jalan dan berdasarkan radius. Oleh karena itu pada penelitian selanjutnya dapat ditingkatkan lagi dengan menambah fitur-fitur tambahan lainnya.

DAFTAR PUSTAKA

- [1] Kominfo. https://kominfo.go.id/index.php/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel_gpr. Diakses pada 9 Januari 2018
- [2] Lapor Layanan Aspirasi dan Pengaduan Online Rakyat. <http://opengovindonesia.org/practice/2/lapor-layanan-aspirasi-dan-pengaduan-online-rakyat>. Diakses 5 Februari 2018.
- [3] Faya Mahdia & Fiftin Noviyanto. 2013. **Pemanfaatan Google Maps API untuk Pembangunan Sistem pelaporan Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis Mobile Web**. Jurusan Teknik informatika, Universitas Ahmad Dahlan, Yogyakarta.
- [4] Helmi Bahar Alim. 2016. **Sistem Informasi Jalan dan Jembatan di Kabupaten Wonogiri**. Jurusan Teknik Informatika, Universitas Islam Negeri Sunan Kalijaga, Yogyakarta
- [5] Junar A.Landicho. 2016. *A Web-based Geographical Project Monitoring and Information System For The Road and Highways*. Journal of Electrical Systems and Information Technology 5 (2018) 252-261.
- [6] Artikel Teknologi. <http://www.artikelteknologi.com/2016/09/apa-itu-google-maps-dan-cara-kerjanya.html>. Diakses pada 4 Februari 2018

- [7] Prahasta, E. 2014. Sistem pelaporan Geografis: Konsep-konsep Dasar(Perspektif Geodesi & Geomatika). Edisi Revisi. Bandung: Informatika.
- [8] Charter, Denny. 2004. **Desain dan Applikasi GIS**. PT. Elex Media Komputindo.
- [9] Yohanli. <http://www.yohanli.com/konsep-mvc-model-view-controller.html>. Diakses pada 14 Februari 2018
- [10] Dunia ilkom. <https://www.duniailkom.com/pengertian-dan-fungsi-php-dalam-pemograman-web/>. Diakses pada 19 Juli 2018.
- [11] Termas
Media.<https://www.termasmedia.com/lainnya/software/69-pengertian-database.html>. diakses pada 19 Juli 2018.
- [12] Kolektor ilmu. <https://kolektorilmu.com/tentang-mysql/>. Diakses pada 19 Juli 2018.
- [13] Cloud Hosting Indonesia.
<https://idcloudhost.com/panduan/mengenal-apa-itu-framework-codeigniter/>. Diakses pada 19 Juli 2018.

LAMPIRAN

Source Code

a. Controller pada Home (Frontend)

```
<?php
defined('BASEPATH') OR exit('No direct
script access allowed');

class Home extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('home_model');
        $this->page_active = 'home';
    }

    function index()
    {
        $param['keyword'] = $this->input-
>get('q');
        $param['data'] = $this-
>home_model->get_data_kerusakan($param)-
>result();
        $param['opt_tingkat_kerusakan'] =
array(
            'Ringan' => 'Ringan',
            'Sedang' => 'Sedang',
            'Parah' => 'Parah'
        );
        $param['main_content'] = 'main';
        $param['page_active'] = $this-
>page_active;
        $this->templates-
>load('main_templates', $param);
    }
}
```

```

function preview_kerusakan()
{
    $respon = array(
        'status'      => '201',
        'msg'         => '',
        'data'         => array()
    );
    if(!empty($_FILES['userfile']['tmp_name']))
    {
        $exif_gambar =
exif_read_data($_FILES['userfile']['tmp_name'], 0, true);
        if(empty($exif_gambar['GPS']))
        {
            $respon = array(
                'status'=> '201',
                'msg'=> war_msg('Gambar tidak valid,
tidak ditemukan lokasi gambar (latitude /
longitude kosong).'),
                'data'=> array()
            );
        }
    }
    else
    {
        $getLocation =
readGPSImage($exif_gambar['GPS']);
        if(!empty($getLocation['lat']) &&
!empty($getLocation['lng'])) {
            $data = array(
                'lat'=> $getLocation['lat'],
                'lng'=> $getLocation['lng'],
                'alamat'=> '',
            );
        }
        $key = $this->config->item('map_api');
        $url=
"https://maps.googleapis.com/maps/api/geocode/json?latlng=$getLocation[lat],$getLocation[lon]"
    }
}

```

```

tion[lng]&key=$key";
$get_address = get_curl($url);
if(!empty($get_address))
{
    $address = json_decode($get_address);
    $data['alamat']=@$address->results[0]-
>formatted_address;
}

$respon = array(
    'status'      => '200',
    'msg'  =>info_msg('Berhasil memuat gambar.'),
    'data'=> $data
);
}

else
{
    $respon = array(
        'status'=> '201',
        'msg'=>err_msg('Gagal memuat gambar.'),
        'data'=> array()
    );
}
}

echo json_encode($respon);
}

function post_kerusakan()
{
    $data_post = $this->input->post();
    $this->form_validation-
>set_rules('kedalaman_lubang', 'Kedalaman Lubang', 'required');
    $this->form_validation-
>set_rules('luas_lubang', 'Luas Lubang',
'required');
}

```

```
    $this->form_validation->set_rules('tingkat_kerusakan',      'Tingkat Kerusakan', 'required');
    $respon = array();
    if($this->form_validation->run ()==false)
    {
        $respon = array(
            'status' => '201',
            'msg'      =>
        war_msg(validation_errors()),
            'data'      => array()
        );
    }
    else
    {
        if(empty($_FILES['userfile']['tmp_name']))
        {
            $respon = array(
                'status'=> '201',
                'msg'  =>war_msg('Gambarharus disertakan'),
                'data'=> array()
            );
        }
        else{
            $config['upload_path']= './uploads/kerusakan_jalan/';
            $config['allowed_types'] = 'jpg|png|jpeg';
            // $config['max_size']= '2048';

            if (!is_dir($config['upload_path']))
            {

                mkdir($config['upload_path']);
            }
            $this->load->library('upload', $config);
            if (!$this->upload->do_upload('userfile'))
```

```
        {
$respon = array(
    'status' => '201',
    'msg'=>war_msg($this->upload-
>display_errors()),
    'data'=> array()
);
}
else{
    $data_upload = $this->upload->data();
    $data_post['gambar']=$config['upload_p
ath'] . $data_upload['file_name'];

    $exif_gambar=
exif_read_data($data_post['gambar'],      0,
true);

    if(empty($exif_gambar['GPS']))
    {
        $respon = array(
        'status' => '201',
        'msg'=> war_msg('Gambar tidak valid,
tidak ditemukan lokasi gambar (latitude /
longitude kosong.'),
        'data'=> array()
    );
    unlink($data_post['gambar']);
    }
else{
    $getLocation=
readGPSImage($exif_gambar['GPS']);
    if(empty($getLocation['lat']))           ||
empty($getLocation['lng']))
    {
        $respon = array(
        'status' => '201',
        'msg'=>war_msg('Lokasi gambar tidak
valid.'),
        'data'=> array())
    }
}
```

```

    );
    unlink($data_post['gambar']);
}
else{
    if(empty($exif_gambar['EXIF']['DateTimeOriginal'])) {
        $data_post['waktu']=date('Y-m-d H:i:s');
    }
    else{
        $data_post['waktu'] = date('Y-m-d H:i:s',
        strtotime($exif_gambar['EXIF']['DateTimeOriginal']));
    }
}
$data_post['lat']=$getLocation['lat'];
$data_post['lng']=$getLocation['lng'];
$proses=$this->home_model->insert_data_kerusakan($data_post);
if($proses)
{
    $data_post['waktu'] =
format_tanggal_indonesia($data_post['waktu'],
true);
    $data_post['gambar'] =
base_url($data_post['gambar']);
    $data_post['id']=$this->db->insert_id();
    $respon = array(
        'status' => '200',
        'msg' =>suc_msg('Data kerusakan berhasil disimpan.'),
        'data' => $data_post
    );
}
else
{
    unlink($data_post['gambar']);
    $respon = array(
        'status' => '201',
        'msg' => war_msg('Data kerusakan gagal disimpan.'),
}

```

```

        'data'          => array()
    );
}
}
}
}
echo json_encode($respon);
}

function data_table()
{
    $param = array(
        'offset'=>$this->input-
>get('start'),
        'limit'=>$this->input-
>get('length'),
        'keyword'=>$this->input-
>get('q')
    );
    $data = $this->home_model-
>get_data_kerusakan($param)->result();

    unset($param['offset']);
    unset($param['limit']);
    $num_rows = $this->home_model-
>get_data_kerusakan($param)->num_rows();

    $result = array();
    foreach($data as $key => $c)
    {
        $c->waktu =
format_tanggal_indonesia($c->waktu, true);
        $c->kedalaman_lubang = $c-
>kedalaman_lubang . ' cm';
        $c->luas_lubang = $c->luas_lubang
. ' cm';
        $c->gambar = '';
        $c->action='<button type="button" '
    }
}

```

```
data-dismiss="modal"      class="btn      btn-
success btn-xs" onclick="lihat_kerusakan_'
. $c->kerusakan_id . '();">
<i class="fa fa-map-marker"></i> Tunjukkan
lokasi</button>';

    $result[] = $c;
}

$respon = array(
    'recordsTotal'=> $num_rows,
    'recordsFiltered'=> $num_rows,
    'data'=> $result
);
echo json_encode($respon);
}

function terdekat($id = '')
{
    if(empty($id)) {
        show_404();
    }

    $param['info'] = $this->home_model-
>get_data_kerusakan(array('id' => $id))->
row();
    if(empty($param['info'])) {
        show_404();
    }

    $param['terdekat'] = $this-
>home_model->get_terdekat(@$id,
@$param['info']->lat,      @$param['info']-
>lng)->result();
    $this->load->view('terdekat',
$param);
}
}
```

b. Model pada Home (Frontend)

```

<?php
defined('BASEPATH') OR exit('No direct
script access allowed');

class Home_model extends CI_Model
{
    public function __construct()
    {
        parent::__construct();
    }
    public function get_data_kerusakan($param
= array())
    {
        if(!empty($param))
        {
            if(!empty($param['limit']))
            {
                if(!empty($param['offset']))
                {
                    $this->db-
>limit($param['limit'], $param['offset']);
                }
                else
                {
                    $this->db->limit($param['limit']);
                }
            }
            if(!empty($param['keyword']))
            {
                $this->db->like('alamat',
$param['keyword']);
            }
            if(!empty($param['id'])) {
                $this->db->where('kerusakan_id',
$param['id']);
            }
        }
    }
}

```

```

        $this->db-
>order_by('kerusakan_id', 'DESC');
        $this->db->from('kerusakan');
        $query = $this->db->get();
        return $query;
    }

    Publicfunction
insert_data_kerusakan($param = array())
{
    $this->db->insert('kerusakan',
$param);
    if($this->db->affected_rows() > 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}

public      function      get_terdekat($id,
$lat, $lng, $distance = 1)
{
    $query_str      =
"SELECTkerusakan_id,
alamat, waktu,
(
3959 * acos(
cos( radians($lat) ) * cos( radians( lat
) ) *cos( radians( lng ) - radians($lng) )
+ sin( radians($lat) ) *sin(radians(lat)
)
)      ASdistanceFROM      kerusakanWHERE
kerusakan_id      !=      $idHAVING      distance      <
$distanceORDER BY distance";
    $query=$this->db->query($query_str);
    return $query;
}

```

{}

c. Views pada Form (Frontend)

```
<div class="modal fade" tabindex="-1"
role="dialog" id="modal-tambah-kerusakan">
<div class="modal-dialog modal-lg">
<div class="modal-content">
<form id="tambahKerusakan"
enctype="multipart/form-data"
onsubmit="return false;">
<div class="modal-header">
<button type="button" class="close" data-
dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
<h4 class="modal-title">
<i class="fa fa- plus"></i>&ampnbsp&ampnbspTambah Data</h4>
</div>
<div class="modal-body">
<div class="row">
<div class="col-md-6">
<div class="row">
<div class="col-md-6">
<div class="form-group">
<label class="control-label">Kedalaman Lubang (cm) *</label>
<input type="text" name="kedalaman_lubang"
class="form-control">
</div>
</div>
<div class="col-md-6">
<div class="form-group">
<label class="control-label">Luas Lubang (cm) *</label>
<input type="text" name="luas_lubang"
class="form-control">
```

```
</div>
</div>
</div>
<div class="row">
<div class="col-md-6">
<div class="form-group">
<label class="control-label">Tingkat Kerusakan *</label>
<?=form_dropdown('tingkat_kerusakan',
$opt_tingkat_kerusakan, '',
'>', 'class="form-control")?>
</div>
</div>
</div>
<div class="form-group">
<label class="control-label">Gambar *</label>
<input type="file" class="form-control" name="userfile"
onchange="previewKerusakan()"/>
</div>
<div class="form-group">
<label class="control-label">Alamat</label>
<input type="text" name="alamat" class="form-control">
</div>
<div class="postResult text-center"></div>
</div>
<div class="col-md-6">
<div id="map_preview"></div>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-primary" onclick="postKerusakan()">
    <i class="fa fa-check"></i> Simpan
</button>
```

```
<button type="button" class="btn btn-default" data-dismiss="modal">
    <i class="fa fa-times"></i> Batal
</button>
</div>
</form>
</div>
</div>
</div>

<style>
    #map_preview
    {
        width: 100%;
        height: 380px;
        background-color: #eaeaea;
    }
</style>
<script>
    function tambahKerusakan()
    {
        $('#modal-tambah-
kerusakan').modal();
    }

    function previewKerusakan()
    {
        var formData = new
FormData($('form#tambahKerusakan')[0]);

        $('form#tambahKerusakan
.postResult').html('<img
src=<?=base_url('assets/img/loader.gif')?>
>" style="width:40px;">Loading ...');

        $.ajax({
            url
            : '<?=site_url('home/preview_kerusakan')?>',
            type
            : 'POST',
            data
            : formData,
```

```
        dataType      : 'json',
        cache        : false,
        contentType   : false,
        processData   : false,
        success       : function (result)
{
    $('form#tambahKerusakan
.postResult').html(result.msg);
    if(result.status == '200')
    {
        $('form#tambahKerusakan
input[name="alamat"]').val(result.data.alamat);
        var myLatLng = {lat : result.data.lat,
            lng : result.data.lng};
        var map_preview = new
            google.maps.Map(document.getElementById('map_preview'), {
                center : myLatLng,
                zoom : 5,
                zoomControl : true,
                mapTypeControl : true,

                mapTypeControlOptions: {
                    style:
                    google.maps.MapTypeControlStyle.HORIZONTAL
                    _BAR,
                    position:
                    google.maps.ControlPosition.TOP_RIGHT
                },
                mapTypeId
                    :
                    google.maps.MapTypeId.ROADMAP,
                rotateControl : true, });

        Varmarker_preview =
            new
            google.maps.Marker({
                position : myLatLng,
                map : map_preview,
                animation : google.maps.Animation.DROP,
```

```
title      : 'Hello World!'
    });
Varbounds_preview = new
google.maps.LatLngBounds();
bounds_preview.extend(myLatLng);
map_preview.fitBounds(bounds_preview);
},
});
return false;
}
var marker_add = [], content_add = [];
function postKerusakan(){
varformData = new
FormData($('form#tambahKerusakan')[0]);

$('form#tambahKerusakan
.postResult').html('<img
src=<?=base_url('assets/img/loader.gif')?
>" style="width:40px;">Loading ...');
$.ajax({
url:
'<?=site_url('home/post_kerusakan')?>',
    type      : 'POST',
    data      : formData,
    dataType  : 'json',
cache      : false,
    contentType : false,
processData : false,
    success   : function (result) {
$('form#tambahKerusakan
.postResult').html(result.msg);
if(result.status == '200')
{
marker_add[result.data.id] = new
google.maps.Marker({
position: {lat : result.data.lat, lng :
result.data.lng},map: map,animation:
google.maps.Animation.DROP, title :
result.data.alamat
```

```
});  
bounds.extend({lat : result.data.lat, lng  
: result.data.lng});  
content_add[result.data.id] = `  
style="width: 200px;"><br>  
<h5 style="margin: 5px 0px 0px;">Latitude  
: <b>${result.data.lat}</b></h5>  
<h5 style="margin: 0px;">Longitude :  
<b>${result.data.lng}</b></h5>  
<h5 style="margin: 0px;">Waktu :  
<b>${result.data.waktu}</b></h5>  
<h5>${result.data.alamat}</h5>  
<button type="button"  
onclick="lihat_sekitar(${result.data.id})"  
class="btn btn-default btn-xs btn-default  
btn-block">  
<i class="fafa-map-marker"></i>Lihat  
Sekitar</button>  
</div>`;  
marker_add[result.data.id].addListener('cl  
ick', function() {  
if(infowindow) infowindow.close();  
infowindow=new  
google.maps.InfoWindow({content:  
content_add[result.data.id]});  
  
infowindow.open(map,  
marker_add[result.data.id]);  
});  
map.fitBounds(bounds);  
$('#modal-tambah-  
kerusakan').modal('hide');  
  
$('form#tambahKerusakan')[0].reset();  
  
$('.table-jalan-
```

```

    rusak') .DataTable() .ajax.reload();
}
})
});
return false;
}
</script>

```

d. Views pada lokasi Terdekat (Frontend)

```

<table class="table table-striped tabel-hover">
<thead>
<tr>
<th>Alamat</th>=<th class="col-md-2 text-center">Jarak</th>
<th class="col-md-3">Waktu</th>
<th></th>
</tr>
</thead>
<tbody>
<?php if(!empty($terdekat)) { ?>
<?php foreach($terdekat as $key => $c): ?>
<tr>
<td><?=$c->alamat?></td>
<td class="text-center"><?=number_format($c->distance, 2, '.', ',')?> km</td>
<td><?=format_tanggal_indonesia($c->waktu, true)?></td>
<td>
<button type="button" data-dismiss="modal" class="btn btn-success btn-xs" onclick="lihat_kerusakan_<?=$c->kerusakan_id?>();">
<i class="fa fa-map-marker"></i> Tunjukkan lokasi
</button>
</td>

```

```

</tr>
<?php endforeach; ?>
<?php } else { ?>
<tr>
<td colspan="4">
<?=info_msg('Tidak ada data kerusakan terdekat.')?>
</td>
</tr>
<?php } ?>
</tbody>
</table>

```

e. Controller pada Home bagian backend

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Home extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->login_status = $this->session->userdata('login_status');
        $this->login_uid = $this->session->userdata('login_uid');
        if($this->login_status == 'ok')
        {
        }
        else
        {
            $this->session->set_flashdata('msg', err_msg('Silahkan login untuk melanjutkan.'));
            redirect(site_url('login'));
        }
    }
}

```

```

        $this->page_active = 'home';
    }

    function index()
    {
        $param['main_content'] = 'main';
        $param['page_active'] = $this-
>page_active;
        $this->templates-
>load('main_templates', $param);
    }
}

```

f. Controller pada Data kerusakan

```

<?php if ( ! defined('BASEPATH')) exit('No
direct script access allowed');

class Data_kerusakan extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->login_status = $this-
>session->userdata('login_status');
        $this->login_uid = $this-
>session->userdata('login_uid');
        if($this->login_status != 'ok')
        {
            $this->session-
>set_flashdata('msg', err_msg('Silahkan
login untuk melanjutkan.'));
            redirect(site_url('login'));
        }

        $this->load-
>model('data_kerusakan_model');
        $this->page_active =
'data_kerusakan';
        $this->sub_page_active =

```

```

'data_kerusakan';
}

public function index(){
    $param['keyword'] = $this->input-
>get('q');
    $limit = 10;
    $uri_segment= 3;
    $filter = array(
        'limit'      => $limit,
        'offset'     =>$this->uri-
>segment($uri_segment),
        'keyword'   => $param['keyword']
    );
    $param['data'] =
$this->data_kerusakan_model-
>get_data($filter)->result();

    unset($filter['limit']);
    unset($filter['offset']);
    $total_rows = $this-
>data_kerusakan_model->get_data($filter)-
>num_rows();
    $param['pagination'] =
paging('data_kerusakan/index',
$total_rows, $limit, $uri_segment);
    $param['main_content'] =
'data_kerusakan/table';
    $param['page_active'] =
$this->page_active;
    $param['sub_page_active'] =
$this->sub_page_active;
    $this->templates-
>load('main_templates', $param);
}

public function form($id = '') {
    $param['msg'] = $this->session-

```

```

>flashdata('msg');
    $param['id'] = $id;

    $last_data = $this->session-
>flashdata('last_data');
    if(!empty($last_data)) {
        $param['data'] = (object)
$last_data;
    }
    else{
        if(!empty($id)) {
            $param['data'] = $this-
>data_kerusakan_model->get_data_row($id);
        }
    }
    $param['opt_tingkat_kerusakan'] = array(
        'Ringan' => 'Ringan',
        'Sedang' => 'Sedang',
        'Parah' => 'Parah'
    );
    $param['main_content'] =
'data_kerusakan/form';
    $param['page_active'] =
$this->page_active;
    $param['sub_page_active'] =
$this->sub_page_active;
    $this->templates-
>load('main_templates', $param);
}
public function submit($id = '') {
    $data_post = $this->input->post();
    $this->form_validation-
>set_rules('kedalaman_lubang', 'Kedalaman
Lubang', 'required');
    $this->form_validation-
>set_rules('luas_lubang', 'Luas Lubang',
'required');
    $this->form_validation-
>set_rules('tingkat_kerusakan', 'Tingkat

```

```

Kerusakan', 'required');
        if($this->form_validation->run() == false) {
            $this->session->set_flashdata('msg',
err_msg(validation_errors()));
            $this->session-
>set_flashdata('last_data', $data_post);
            redirect('data_kerusakan/form/' .
$id);
        }
        else{
            if(empty($id)) {
                $proses = $this-
>data_kerusakan_model->insert($data_post);
                if($proses) {
                    $this->session->set_flashdata('msg',
suc_msg('Data berhasil disimpan.'));
                }
            }
            else{
                $this->session->set_flashdata('msg',
err_msg('Data gagal disimpan, silahkan ulangi lagi.'));
            }
            redirect('data_kerusakan/form/' .
$id);
        }
    }
    else{
        $proses = $this-
>data_kerusakan_model->update($data_post,
$id);
        if($proses) {
            $this->session->set_flashdata('msg',
suc_msg('Data berhasil diperbarui.'));
        }
        else{
            $this->session->set_flashdata('msg',
err_msg('Data gagal diperbarui, tidak ada yang berubah.'));
        }
    }
}

```

```
        }
    }
    redirect('data_kerusakan');
}
}

public function hapus($id) {
    $data = $this->data_kerusakan_model-
>get_data_row($id);
    if(empty($data)) {
        show_404();
    }
unlink($data->gambar);
$proses = $this->data_kerusakan_model-
>delete($id);
    $this->session->set_flashdata('msg',
suc_msg('Data berhasil dihapus.'));
    redirect('data_kerusakan');
}
}
```

g. Model dari Data kerusakan

```
<?php
class Data_kerusakan_model extends CI_Model
{
    function get_data($param = array()){
        if(!empty($param)) {
            if(!empty($param['limit'])) {
                if(!empty($param['offset'])) {
                    $this->db->limit($param['limit'],
$param['offset']);
                }
            }
            else{
                $this->db->limit($param['limit']);
            }
        }
        $this->db->select('a.*');
        $this->db-
```

```
>order_by('a.kerusakan_id', 'DESC');
    $this->db->from('kerusakan a');
    $get = $this->db->get();
    return $get;
}

function get_data_row($id) {
    $this->db->where('kerusakan_id',
$id);
    $get = $this->db->get('kerusakan');
    return $get->row();
}

function insert($data) {
    $this->db->insert('kerusakan',
$data);
    if($this->db->affected_rows() > 0) {
        return true;
    }
    else{
        return false;
    }
}

function update($data, $id) {
    $this->db->where('kerusakan_id',
$id);
    $this->db->update('kerusakan',
$data);
    if($this->db->affected_rows() > 0) {
        return true;
    }
    else{
        return false;
    }
}
function delete($id) {
    $this->db->where('kerusakan_id',
$id);
```

```
        $this->db->delete('kerusakan');
        return true;
    }
}
```

h. Views pada Data Kerusakan

```
<div class="row">
    <div class="col-md-12">
<div class="box box-default">
<div class="box-header with-border">
<h3 class="box-title">
<i class="fa fa-list" style="font-size: 1.5em; color: #337ab7; margin-right: 10px;"></i> Data Kerusakan
</h3>
</div>
<div class="box-body">
<div class="row">
    <div class="col-md-12">
        <a href="=site_url('data_kerusakan')?&gt;" class="btn btn-default"&gt;
            &lt;i class="fa fa-history" style="font-size: 1.5em; color: #337ab7; margin-right: 10px;"&gt;&lt;/i&gt; Kembali
        &lt;/a&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;hr/&gt;
&lt;form method="POST" class="form-horizontal" action="<?=site_url('data_kerusakan/submit/?'. $id)?&gt;" enctype="multipart/form-data"&gt;
    &lt;div class="form-group"&gt;
        &lt;label class="col-md-3 control-label"&gt;Kedalaman Lubang (cm) * &lt;/label&gt;
        &lt;div class="col-md-2"&gt;
            &lt;input type="text" class="form-control" name="kedalaman_lubang" value="<?=@$data-&gt;kedalaman_lubang?&gt;"&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/form&gt;</pre
```

```
>kedalaman_lubang?>">
        </div>
        </div>
<div class="form-group">
label class="col-md-3 control-label">Luas Lubang (cm) *</label>
<div class="col-md-2">
<input type="text" class="form-control" name="luas_lubang" value="<?=@$data->luas_lubang?>">
        </div>
        </div>
<div class="form-group">
<label class="col-md-3 control-label">Tingkat Kerusakan *</label>
<div class="col-md-3">
        <?=form_dropdown('tingkat_kerusakan', $opt_tingkat_kerusakan, '', 'class="form-control')?>
        </div>
        </div>
<div class="form-group">
<label class="col-md-3 control-label">Alamat</label>
<div class="col-md-8">
<input type="text" class="form-control" name="alamat" value="<?=@$data->alamat?>">
        </div>
        </div>
<div class="form-group">
<label class="col-md-3 control-label">Gambar</label>
<div class="col-md-3">

        </div>
        </div>
<hr/>
```

```
<div class="form-group"><div class="col-md-10 col-md-offset-1">
<div id="map"></div>
</div>
</div>
<hr/>
<div class="form-group">
<div class="col-md-12 text-right">
<button type="submit" class="btn btn-default">
<i class="fa fa-check">&ampnbsp&ampnbspSimpan</button>
<a href="=$site_url('data_kerusakan') ?" class="btn btn-default">
<iclass="fafa-times"></i>&ampnbsp&ampnbspBatal
</a>
</div>
</div>
</form>
</div>
</div>
</div>
</div>

<style>
#map{
    width: 100%;
    height: 380px;
    background-color: #eaeaea;
}
</style>
<script
src="https://maps.googleapis.com/maps/api/js?key=<?=$this->config->item('map_api') ?>&language=id&libraries=drawing,geometry,places&language=id"></script>
<script>
var myLatLng      = {lat : <?=@$data-
```

```

>lat?>, lng : <?=@$data->lng?>};
    var map_preview = new
google.maps.Map(document.getElementById('m
ap'), {
    center : myLatLng,
    zoom : 5,
    zoomControl : true,
    mapTypeControl : true,
    mapTypeControlOptions: {
        style : google.maps.MapTypeControlStyle.HORIZONTAL
_BAR,
        position : google.maps.ControlPosition.TOP_RIGHT
    },
    mapTypeId : google.maps.MapTypeId.ROADMAP,
    rotateControl : true,
});

var marker_preview = new
google.maps.Marker({
    position : myLatLng,
    map : map_preview,
    animation : google.maps.Animation.DROP,
    title : 'Hello World!'
});

var bounds_preview = new
google.maps.LatLngBounds();
bounds_preview.extend(myLatLng);
map_preview.fitBounds(bounds_preview);
</script>

```

i. Login pada bagian Controller

```

<?php if ( ! defined('BASEPATH')) exit('No
direct script access allowed');

```

```

class Login extends CI_Controller{
    function __construct(){
        parent::__construct();
        $this->load->model('login_model');
        $this->page_active      = 'login';
        $this->load->helper('cookie');
        $app_name      =      $this->config-
>item('app_name');
        $this->cookie_name      =  'cookie_'
. format_uri($app_name, '_');
        $login_status      =  $this->session-
>userdata('login_status');
        if($login_status == 'ok') {
            redirect('home');
        }
        else{
            $cek_login = get_cookie($this-
>cookie_name);
            if(!empty($cek_login)) {
                $this->auto_login($cek_login);
            }
        }
    }
    function index(){
        $param['main_content'] = 'form';
        $param['page_active'] =      $this-
>page_active;
        $this->templates-
>load('login_templates', $param);
    }
    function auto_login($data_cookie) {
        $get_data_login      =      $this-
>login_model->get_login($data_cookie)-
>row();
        $this->session->set_flashdata('msg',
suc_msg('Selamat Datang.'));
        $this->session-
>set_userdata('login_status', 'ok');
        $this->session-

```

```

>set_userdata('login_uid',
$get_data_login->user_id);
    redirect('home');
}
function submit(){
    $this->form_validation-
>set_rules('email', 'Email / Username',
'required');
    $this->form_validation-
>set_rules('password', 'Password',
'required');
    if($this->form_validation->run() == false) {
$respon = array('status' => '201', 'data' => validation_errors('', ''), 'msg' => 'Login gagal.');
    }
    else{
$email = $this->input->post('email');
$get_data_login = $this->login_model->get_login($email)->row();
(!empty($get_data_login)) {
$password = $this->input->post('password');
if(md5($password) == $get_data_login->password)
{
    if($get_data_login->status == 'blokir') {
$respon = array('status' => '201',
'data' => 'Login gagal,
Akun diblokir.',
'msg' => 'Login gagal,
Akun diblokir.');
    }
    elseif($get_data_login->status == 'aktif') {
        $respon =

```

```

array('status'      => '200',
      'data'        => 'Login
berhasil.',
      'msg'         => 'Login
berhasil.');
$ingat_saya      = $this->input-
>post('ingat_saya');

if(!empty($ingat_saya)) {
    set_cookie($this->cookie_name,
    $get_data_login->email, '1209600');
}

$this->login_model-
>update_last_login($get_data_login-
>user_id);
$this->session->set_flashdata('msg',
suc_msg('Login Berhasil, Selamat
Datang.'));
$this->session-
>set_userdata('login_status', 'ok');
$this->session->set_userdata('login_uid',
$get_data_login->user_id);
if(empty($get_data_login->terakhir_login))
{
    $terakhir_login = date('H:i:s d-m-Y');
}
else{
    $terakhir_login = date('H:i:s d-m-Y',
strtotime($get_data_login-
>terakhir_login));
}
$this->session-
>set_userdata('login_terakhir',
$terakhir_login);
}
else{
    $respon = array('status' => '201',
'data' => 'Login Gagal.', 'msg' => 'Login
gagal, silahkan ulangi lagi.');
}

```

```

        }
    }

    else{
        $respon = array('status' => '201',
'data' => 'Password tidak valid', 'msg' =>
'Login gagal.');
    }
    else{
        $respon = array('status' => '201',
'data' => 'Email tidak valid', 'msg' =>
'Login gagal.');
    }
    echo json_encode($respon);
}
}

```

j. Model pada Login

```

<?php if ( ! defined('BASEPATH')) exit('No
direct script access allowed');

class Login_model extends CI_Model {

    public function __construct(){
        parent::__construct();
    }
    function get_login($email) {
        $this->db->or_where('username',
$email);
        $this->db->or_where('email',
$email);
        $this->db->from('user');
        $query = $this->db->get();
        return $query;
    }
    function update_last_login($user_id) {

```

```

    $this->db->where('user_id',
$user_id);
    $this->db->update('user',
array('terakhir_login'      =>      date('Y-m-d
H:i:s')));
}
}
/* End of file Login_model.php */

```

k. Views pada Login

```

<p class="login-box-
msg"><strong>Login</strong><br/>Masukkan
Email dan Password</p>
<form action="javascript:;" class="login-
form" id="form-login">
<div class="form-group has-feedback">
<input type="text" class="form-control"
placeholder="Email / Username"
name="email">
<span class="glyphicon glyphicon-envelope
form-control-feedback"></span>
</div>
<div class="form-group has-feedback">
<input type="password" class="form-
control" placeholder="Password"
name="password">
<span class="glyphicon glyphicon-lock
form-control-feedback"></span>
</div>
<div class="row">
<div class="col-md-12">
<div id="form-login-result"></div>
</div>
</div>
<div class="row">
<div class="col-xs-8">
<div class="checkbox icheck">
<label>
<input type="checkbox" name="ingat saya">

```

```
value="1">&nbsp;&nbsp;&nbsp;Ingat Saya
</label>
</div>
</div>
<div class="col-xs-4">
<button type="submit" class="btn btn-primary btn-block btn-flat">Submit</button>
</div>
</div>
</form>
function(result) {
{
$('#form-login-result').html('<div
class="alert alert-danger"
role="alert"><strong>Kesalahan
!</strong><br/>' + result.data + '</div>');
}
}
else{
$('#form-login-result').html('<div
class="alert alert-success" role="alert">' + result.data + '</div>');
window.location = "<?=site_url()?>";
}
},
error : function(result){
$('#form-login-
result').html('<?=err_msg('Gagal melakukan
login.')?>');
}
})
return false;
});
})
</script>
```