

Aplikasi Pemilihan Rute Alternatif Akibat Kemacetan Lalu Lintas di Kota Makassar Menggunakan Google API dan ASP.Net

Alternatives Route Applications Due Traffic Congestion in Makassar Using Google API and ASP.Net

¹⁾Nur Alam, ²⁾Mukhlis Amin

^{1,2}Balai Besar Pengkajian dan Pengembangan Komunikasi dan Informatika Makassar
^{1,2}Jl. Prof. Dr. Abdurrahman Basalamah II No.25, Makassar, 90123, Telp/Fax: 0411-4460084

¹⁾nur.alam@kominfo.go.id, ²⁾mukhlis.amin@kominfo.go.id

Diterima: 4 Juli 2015 || Revisi: 14 Agustus 2015 || Disetujui: 18 Agustus 2015

Abstrak – Persoalan kemacetan yang semakin parah di Kota Makassar saat ini perlu segera diatasi. Penelitian ini berusaha memberikan sebuah solusi aplikasi pemilihan rute alternatif untuk mengatasi kemacetan. Aplikasi ini dapat menampilkan lokasi kemacetan yang sedang terjadi beserta jalur alternatif yang memungkinkan untuk dilalui agar terhindar dari kemacetan tersebut. Rute alternatif diperoleh berdasarkan layanan yang telah disediakan oleh Google API. Sistem ini menggunakan beberapa bahasa pemrograman sesuai jenis aplikasinya. Penentuan titik kemacetan dilakukan melalui telepon seluler dengan aplikasi berbasis android menggunakan bahasa pemrograman *eclipse*. Sementara itu, aplikasi untuk mendesain rute alternatif menggunakan *Active Server Pages* (ASP.Net). ASP.Net dikompilasi dan berbasis *.Net framework* sehingga dapat membuat aplikasi dalam berbagai macam bahasa yang mendukung *.Net*. Pengujian hasil implementasi sistem dilakukan dengan menggunakan metode *black box*. Hasil pengujian pada setiap halaman utama menunjukkan bahwa keluaran yang dihasilkan melalui beberapa skenario pengujian sudah sesuai dengan yang diharapkan yang berarti halaman-halaman tersebut dapat berjalan dengan baik.

Kata Kunci: ASP.Net, google API, kemacetan lalu lintas, rute alternatif

Abstract – *The issue of congestion is getting worse in Makassar today need to be addressed urgently. This study provides an Alternative Route Selection Application as a solutions to tackle congestion. This application can display the location of congestion that is happening along the alternative path that allows it to be passed in order to avoid the jam. Alternate route is obtained based on the services provided by the Google API. This system uses several programming languages according to the type of application. Determining the point of congestion is done via mobile phone with android based applications using the programming language eclipse. Meanwhile, applications for designing an alternative route using Active Server Pages (ASP.Net). Compiled ASP.NET and .NET Framework-based so that it can make an application in various languages that support .NET. Testing results of the implementation of the system is done by using black box method. The test results on each main page shows that the output produced by some test scenarios have been as expected which means that the pages can be run properly.*

Keywords: ASP.Net, google API, traffic congestion, alternative route

PENDAHULUAN

Perkembangan teknologi otomotif yang begitu pesat bukan hanya membawa dampak positif tetapi berbagai dampak negatif juga harus dibayar mahal oleh masyarakat. Salah satu teknologi otomotif yang paling pesat perkembangannya hingga saat ini adalah industri kendaraan bermotor (mobil dan sepeda motor).

Berdasarkan *Latest Vehicles in Operation (VIO) Market Analysis*, sampai tahun 2013 jumlah kendaraan berupa mobil dan truk di seluruh dunia mencapai 247.9 juta unit. Di Indonesia sendiri, menurut data terakhir Korps Lalu Lintas Kepolisian

Republik Indonesia (Korlantas Polri), jumlah kendaraan mobil sampai tahun 2012 adalah sebanyak 16.473.641 dan jumlah kendaraan sepeda motor mencapai 77.755.658 unit.

Banyaknya kendaraan yang diproduksi dan beredar menjadi penyebab utama terjadinya kemacetan di jalan raya karena hal ini menyebabkan pertumbuhan panjang dan lebar jalan yang tidak seimbang dengan laju pertumbuhan produksi kendaraan. Pertumbuhan produksi kendaraan setiap tahunnya sangatlah besar sementara panjang dan lebar jalan di tiap kota pertambahannya tidak cukup signifikan.

Besarnya jumlah kendaraan di Indonesia, Makassar menyumbang 2,4 juta unit kendaraan bermotor

dengan jumlah kendaraan bermotor roda 2 (sepeda motor) sebanyak 1,2 juta unit. Jumlah kendaraan yang demikian banyak beroperasi di Makassar membuat sebagian besar jalan di kota Makassar semakin hari semakin akrab dengan kemacetan dan nyaris tidak mengenal sistem waktu.

Kemacetan di Makassar yang kian parah juga diakui oleh ketua DPRD Kota Makassar sehingga meminta kepada pemerintah kota Makassar untuk segera mencari solusi akan hal tersebut termasuk dengan segera menyelesaikan proyek yang hingga kini belum terlaksana. Beberapa proyek yang dimaksud adalah pembangunan jalan lingkar (*middle ring road*) dan sistem transportasi massal (*busway*).

Potensi kerugian akibat kemacetan di Makassar tahun 2012 mencapai 400 miliar rupiah (Khan, 2013, hal. 3). Angka ini bukanlah angka yang kecil dan dapat dianggap remeh. Dari sisi lain, kerugian akibat kemacetan tidak bisa hanya diukur dari kerugian ekonomi. Waktu yang terbuang disertai dengan penderitaan fisik yang dialami masyarakat apabila terjebak dalam kemacetan pun merupakan sebuah bentuk kerugian masyarakat.

Solusi kemacetan berbasis teknologi informasi (TI) dapat diupayakan oleh pemerintah salah satunya dengan memberikan informasi kepada masyarakat tentang rute alternatif yang harus dilalui apabila terjadi kemacetan terhadap jalan-jalan tertentu. Hal ini bertujuan selain untuk mengurai kemacetan dengan mengurangi jumlah kendaraan yang melewati jalur/titik kemacetan, juga akan membuat para pengendara mengetahui arah dan jalan mana yang sebaiknya di tempuh agar tidak ikut terjebak dalam kemacetan yang terjadi.

Adapun sistem penanganan kemacetan di kota Makassar masih dengan cara yang sangat konvensional dimana ketika terjadi kemacetan, petugas dari kepolisian akan turun ke lokasi untuk mengatur lalu lintas agar kemacetan tersebut cepat terurai. Namun, hampir dalam segala kondisi, proses penguraian kemacetan berlangsung cukup lama. Hal ini dikarenakan oleh beberapa hal yaitu:

1. Jumlah kendaraan yang tidak sebanding dengan volume kendaraan di waktu-waktu tertentu. Biasanya kondisi ini terjadi di beberapa ruas jalan pada pagi dan sore hari.
2. Jumlah petugas dari kepolisian yang sangat terbatas.
3. Ketidaktahuan masyarakat akan adanya kemacetan tersebut sehingga meskipun macet, tetap saja

masyarakat melewati jalan tersebut sehingga kemacetan makin lama makin parah.

Solusi yang biasanya dipakai untuk mengurai kemacetan ini adalah dengan penempatan petugas atau lebih banyak petugas pada tiap titik kemacetan atau melakukan rekayasa terbatas bila diperlukan untuk melakukan penutupan/pengalihan arus lalu lintas serta melakukan beberapa teknik penanganan kemacetan yang dapat membuat arus lalu lintas berjalan dengan normal.

Peralatan komunikasi yang biasanya dipakai untuk berkomunikasi antar petugas adalah dengan menggunakan *handy talky* (HT) sementara untuk mengkomunikasikan titik-titik kemacetan ini dengan masyarakat, pihak kepolisian dalam hal ini Satlantas Polrestabes Makassar menggunakan Facebook dengan akun "Satlantas Polrestabes Makassar" atau melalui website <http://www.polantasmakassar.com>. Namun dengan kedua fasilitas tersebut dianggap belumlah efektif karena keduanya tidak di khususkan untuk informasi kemacetan semata. Dalam menentukan rute/jalur alternatif diterapkan skala prioritas dengan mengacu pada titik jalur jalan dimana kemacetan itu terjadi. Menurut hasil wawancara yang dilakukan, untuk wilayah Makassar terdapat beberapa titik yang hampir setiap harinya terjadi kemacetan diantaranya di sepanjang jalan AP. Pettarani, jalan Perintis Kemerdekaan, sepanjang jalan Urip Sumoharjo dan sepanjang jalan Sultan Alauddin.

Sebagai alternatif solusi untuk membantu mengurai kemacetan tersebut, maka perlu didesain sebuah sistem yang dapat menentukan rute alternatif untuk mengatasi kemacetan di kota Makassar. Salah satu usulan disampaikan oleh Yohan Lesmana dan kawan-kawan (2015) untuk mengatasi kemacetan dengan membuat sistem kendali lampu lalu lintas dengan pendekatan *Fuzzy Logic*.

Beberapa metode berusaha dimunculkan oleh pemerintah daerah maupun pusat untuk mengatasi masalah kemacetan, namun belum berhasil dengan baik. Salah satu metode yang efektif dalam menyelesaikan permasalahan kemacetan lalu lintas adalah "*Route Guidance System And Method*" atau yang biasa disebut dengan sistem *routing* (Muhtadin, Purnama, Zulfikar, & Hariadi, 2010). Sistem ini diperkenalkan pertama kali oleh Ryujiro Fujita dan kawan-kawan untuk mengatasi kepadatan lalu lintas yang terjadi di beberapa kota besar di Jepang (Ryujiro, Inoue, & Naohiko, 2005). Dong (2011) telah mengulas beberapa penelitian terkait sistem navigasi

rute untuk mengurai kemacetan. Hasil ulasannya menyatakan bahwa beberapa penelitian sebelumnya masih belum matang. Penelitian yang diperlukan saat ini adalah menciptakan sebuah sistem navigasi rute *real time*, yang dapat memberikan saran rute yang akurat, dapat dipercaya dan cerdas.

Route Guidance System (RGS) sudah menjadi sebuah teknologi penting yang perlu dipertimbangkan saat ini guna mengurai kemacetan. Namun, sistem ini hanya mengurai kemacetan saat kemacetan itu sudah terjadi. Dengan demikian, Liang dan Wakahara (2014) mengusulkan sebuah sistem yang dapat memprediksi kemacetan agar RGS dapat bekerja lebih proaktif.

Penelitian-penelitian tentang pencarian jalur alternatif sudah banyak dilakukan di Indonesia. Seperti yang dilakukan oleh Arna Fariza dan kawan-kawan (2009) yang melakukan penelitian tentang pencarian jalur alternatif pada daerah bencana lumpur Sidoarjo dengan metode *Analytic Hierarchy Process* (AHP) berbasis *Wireless Application Protocol* (WAP). Penelitian ini memberikan rekomendasi satu jalur alternatif terbaik yang dihasilkan berdasarkan metode AHP.

Penelitian lain dilakukan oleh Asti Ratnasari dan kawan-kawan (Ratnasari, Ardiani, & A., 2013) yang meneliti tentang penentuan jarak terpendek alternatif ketika terjadi hambatan atau pemblokiran jalan yang terjadi pada jarak terpendek utama dan juga dapat mengetahui estimasi waktu tempuhnya. Penentuan jalur terpendek pada penelitian tersebut menggunakan algoritma *Dijkstra*. Sementara Ivana Varita dan kawan-kawan (2013) meneliti tentang pencarian jalur tercepat dengan menggunakan algoritma *Tabu Search*.

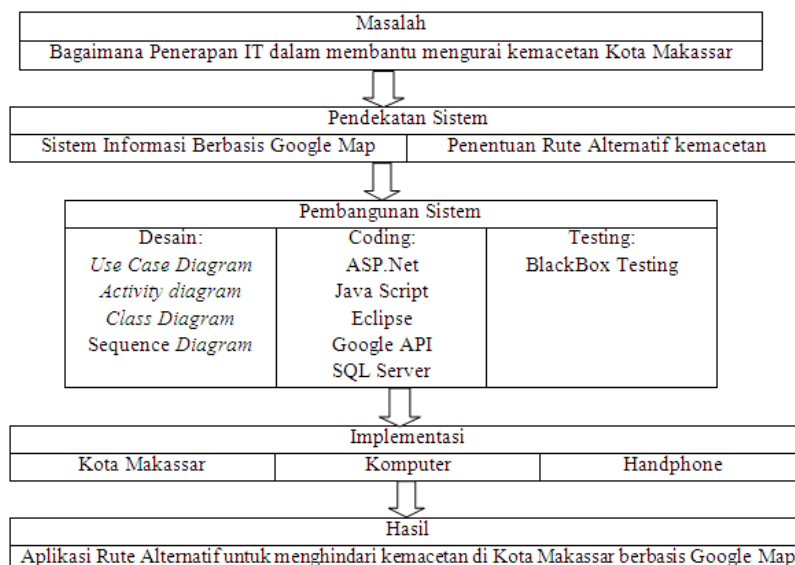
Algoritma ini dapat mengoptimalkan biaya dan menghasilkan jalur biaya minimum dengan cakupan maksimum (Sharma, Jadhaf, Srivastava, & Goyal, 2010).

Berbeda dengan penelitian-penelitian sebelumnya yang fokus pada pencarian jalur terpendek pada berbagai permasalahan lalu lintas, penelitian tentang jalur alternatif di Kota Makassar ini ditujukan untuk mengatasi kemacetan lalu lintas. Tujuan utama penelitian ini adalah membangun sebuah aplikasi yang dapat menampilkan lokasi kemacetan yang sedang terjadi beserta jalur alternatif yang memungkinkan untuk dilalui agar terhindar dari kemacetan tersebut. Rute alternatif diperoleh berdasarkan layanan yang telah disediakan oleh aplikasi Google API.

Bagian berikut akan dipaparkan metodologi penelitian berkaitan dengan langkah-langkah pembangunan aplikasi dan pengujian beserta aplikasi dan bahasa pemrograman yang digunakan. Selanjutnya ditampilkan hasil implementasi sistem beserta analisis tentang pengujian aplikasi. Kemudian ditutup dengan kesimpulan dan saran-saran untuk implementasi maupun pengembangan aplikasi kedepannya.

METODOLOGI PENELITIAN

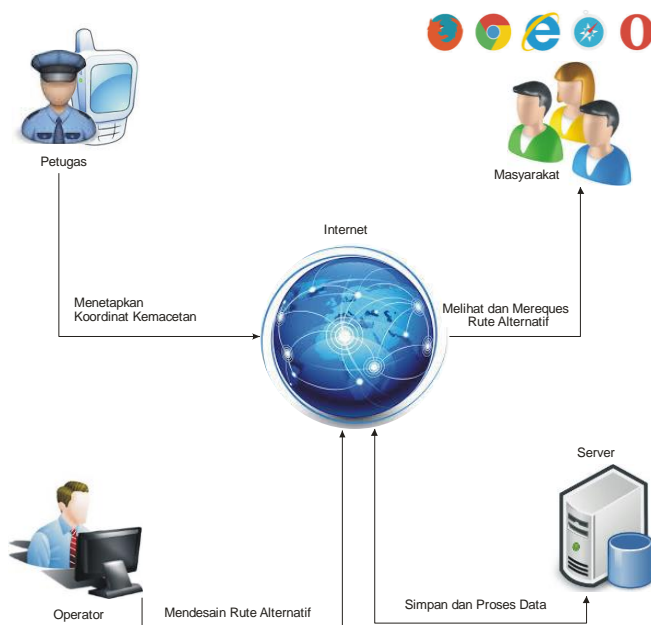
Kegiatan penelitian memerlukan perencanaan agar penelitian yang dilakukan dapat berjalan dengan baik, sistematis serta efektif. Ada tiga tahapan yang umum dilakukan dalam mendesain sistem. Tahapan-tahapan tersebut terdiri dari analisis sistem, desain sistem dan implementasi sistem. Kerangka pikir penelitian ini ditunjukkan pada Gambar 1.



Gambar 1 Kerangka Pikir Penelitian

Analisis sistem

Analisis sistem adalah sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian – bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan – permasalahan, kesempatan, hambatan – hambatan yang terjadi dan kebutuhan – kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan. Tujuan utamanya adalah untuk memahami sistem dan masalah yang ada, untuk menguraikan kebutuhan informasi dan untuk menetapkan prioritas pekerjaan sistem selanjutnya. Analisis kerja sistem yang diusulkan ditunjukkan pada Gambar 2.



Gambar 2 Analisis Kerja Sistem Usulan

Berdasarkan Gambar 2 dapat kita lihat alur dan gambaran sistem secara umum. Apabila terjadi sebuah kemacetan, petugas yang berada atau mengetahui kemacetan tersebut akan segera menginformasikan atau mengupdate data melalui *handphone* dengan menetapkan titik koordinat kemacetan yang terjadi agar segera diketahui oleh operator dan masyarakat. Koordinat kemacetan tersebut akan disimpan di server untuk kemudian oleh operator segera didesain rute alternatif untuk titik kemacetan tersebut apabila dirasa perlu dan memungkinkan. Seperti halnya titik kemacetan, rute alternatif yang telah di desain oleh operator kemudian juga akan disimpan dalam database yang telah dihubungkan dengan internet. Dengan demikian, titik kemacetan dan rute alternatif yang telah di desain akan sampai kemasyarakat dengan memanfaatkan fasilitas internet melalui *browser* yang mereka gunakan.

Jenis dan jumlah aplikasi yang akan didesain untuk memenuhi kebutuhan sistem diatas ada tiga yaitu aplikasi yang memungkinkan untuk petugas mengirim koordinat kemacetan ke server, aplikasi khusus bagi operator untuk mendesain rute alternatif yang disarankan serta aplikasi untuk menampilkan titik kemacetan dan rute alternatif yang telah di desain. Namun demikian, untuk aplikasi menampilkan hasil, aplikasi inipun dibagi menjadi dua jenis tergantung dari perangkat keras apa yang nantinya akan dipakai untuk menampilkannya. Apabila titik kemacetan dan rute alternatif tersebut ditampilkan melalui TV Tron, TV atau sejenisnya maka rute alternatif yang ditampilkan bersifat statis berdasarkan rute alternatif yang telah didesain. Apabila perangkat keras yang dipakai berupa PC, laptop atau *handphone* maka rute alternatif yang ditampilkan bersifat dinamis dimana user dapat menentukan lokasi awal dan lokasi tujuan. Apabila lokasi awal dan tujuan telah ditentukan maka dengan menggunakan *service* yang telah disediakan oleh *Google Map* melalui *Google API*, maka sistem akan memberikan beberapa rute yang mungkin dilalui dengan membedakan warna rute yang melewati titik kemacetan dan yang tidak terjadi kemacetan.

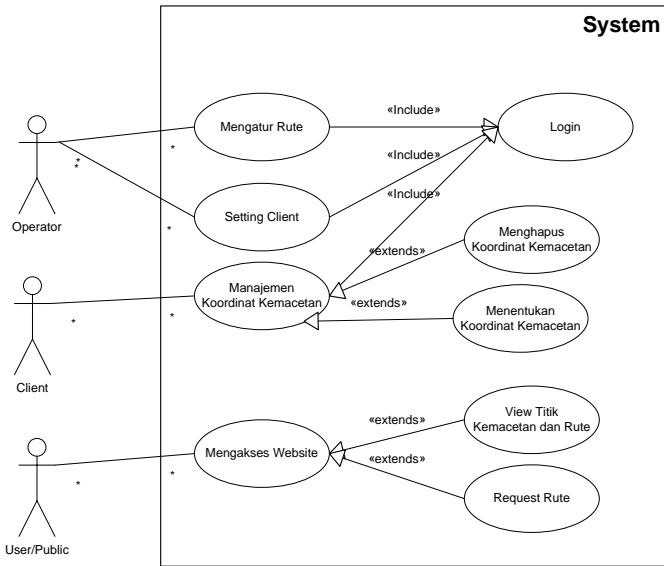
Desain Sistem

Sistem usulan yang telah dibuat perlu dilakukan desain untuk mempermudah penerjemahannya kedalam bahasa program yang akan digunakan untuk membangun sistem. Alat bantu pemodelan yang dipakai dalam perancangan sistem ini menggunakan *Unified Modeling Language* (UML) yang di dalamnya terdiri dari *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequential Diagram* (Fowler, 2004). UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan mendokumentasikan dari sebuah sistem pengembangan perangkat lunak berbasis *Object Oriented* (Duch, 2010).

Use Case Diagram

Berdasarkan Gambar 3 dapat kita lihat bagaimana interaksi yang terjadi antara aktor (operator, *client*, dan *user/public*) dengan sistem yang akan dibangun. Operator pada desain tersebut berfungsi untuk mengatur rute dan melakukan *setting* terhadap *Client*. Pengertian *client* adalah petugas yang nantinya berfungsi sebagai manajemen titik (koordinat) kemacetan. Aktor lain adalah *user/public*, yang dimaksud disini adalah pengguna sistem yaitu

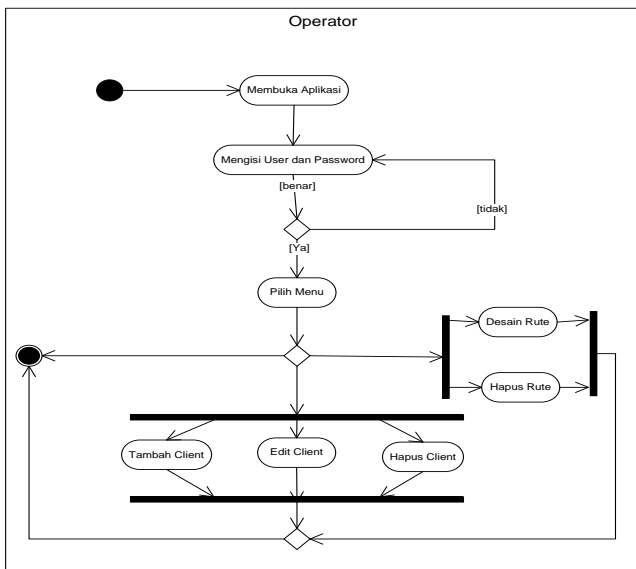
masyarakat yang dapat mengakses sistem yang dibangun melalui *view* titik kemacetan dan melakukan permintaan rute ke sistem.



Gambar 3 Use Case Diagram

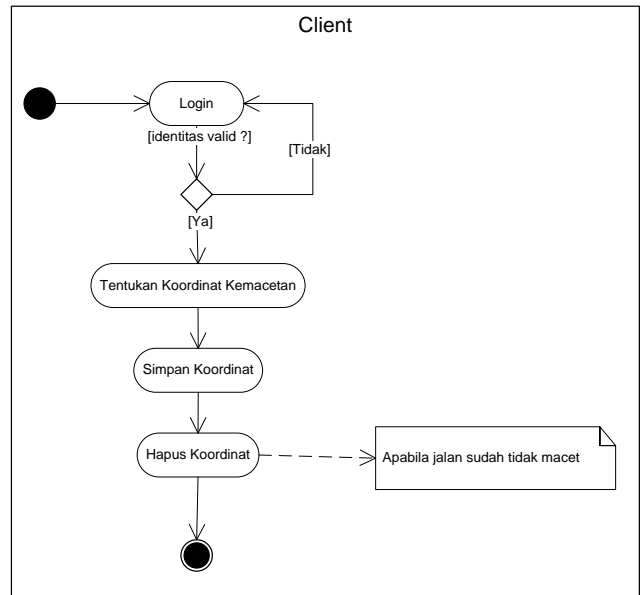
Activity Diagram

Activity diagram digunakan untuk menggambarkan alur bisnis dari sebuah program, dari *activity diagram* tersebut dapat kita lihat bagaimana program alur program ketika operator melakukan manajemen rute yang dibuat terhadap koordinat kemacetan yang ada. Dalam memanajemen rute alternatif, operator harus melakukan *login* terlebih dahulu dengan memasukkan *username* dan *password*. Apabila valid, maka operator memilih tindakan selanjutnya apakah mendesain rute alternatif atau melakukan manajemen (tambah, ubah, hapus) *client*.



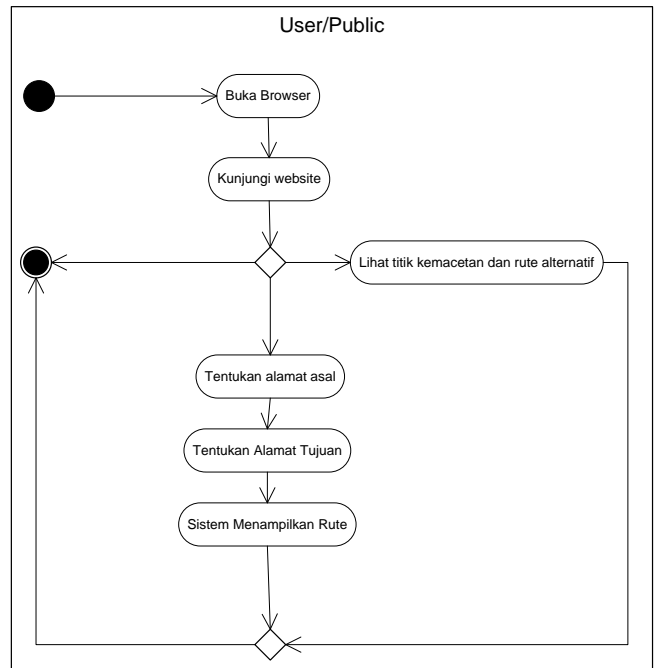
Gambar 4 Activity Diagram Operator

Alur kerja sistem dari sisi *client* (petugas) dimulai dari petugas melakukan login melalui aplikasi android untuk menentukan koordinat kemacetan.



Gambar 5 Activity Diagram Client

Koordinat kemacetan yang ditentukan oleh seorang petugas akan menjadi tanggung jawab petugas untuk menentukan apakah kemacetan ini masih berlangsung atau sudah terurai. Apabila kemacetan telah terurai, petugas kemudian menghapus koordinat yang akan mengakibatkan hilangnya koordinat tersebut dari peta secara visual.



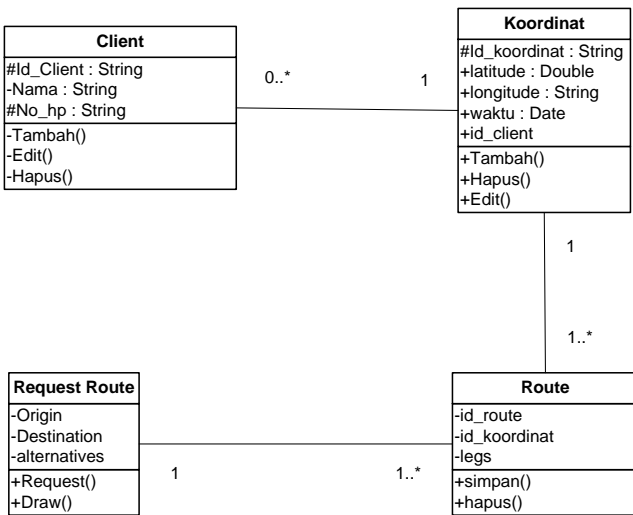
Gambar 6 Activity Diagram User/Public

Alur kerja sistem dari sisi *user* dapat dilihat pada Gambar 6. Untuk mengetahui titik-titik kemacetan

yang ada, *user/public* dalam hal ini yang dimaksud adalah masyarakat harus mengunjungi website yang dibangun untuk menentukan apakah hanya untuk melihat titik kemacetan yang ada atau melakukan *request* (permintaan) rute kepada sistem dengan menentukan titik awal keberangkatan dan titik yang ingin di jadikan tujuan. Setelah menentukan titik awal dan tujuan, sistem akan memberikan maksimal 3 jalur yang dapat di tempuh sebagai rute alternatif.

Class Diagram

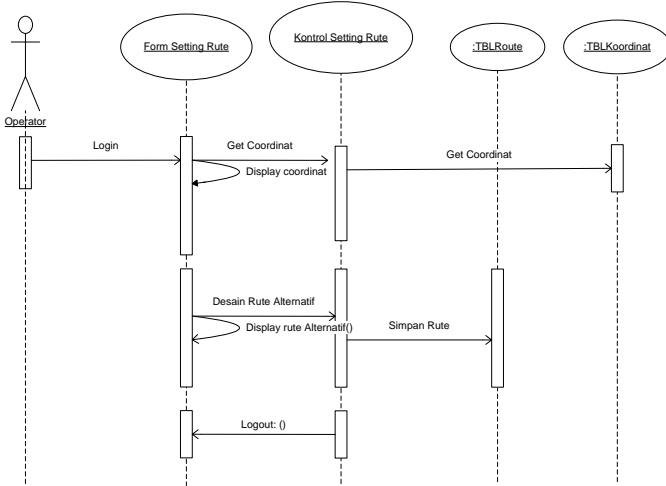
Diagram selanjutnya adalah diagram yang dipakai untuk menggambarkan sistem dengan relasi-relasi yang ada didalamnya melalui kelas dan paket-paket yang ada dalam sistem. *Class diagram* dari sistem yang dibangun dapat dilihat seperti Gambar 7.



Gambar 7 Class Diagram Sistem

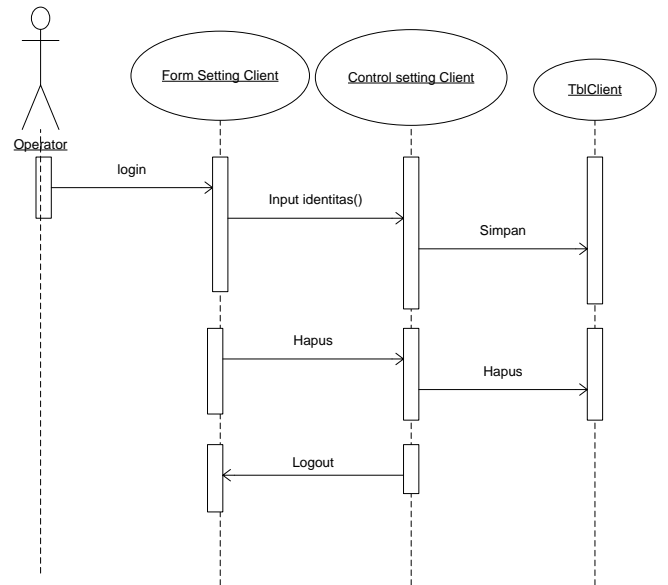
Class diagram diatas menggambarkan empat buah kelas yang terdiri dari *client*, *koordinat*, *request route* dan *route*.

Sequence Diagram



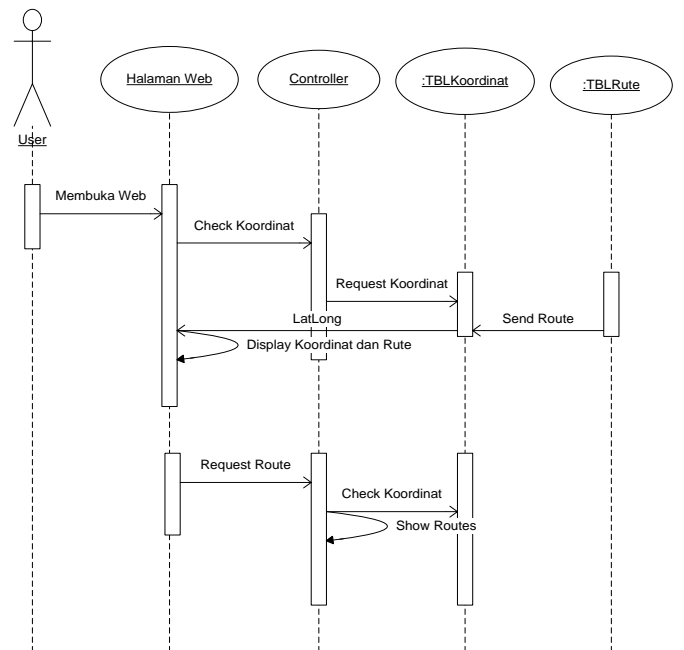
Gambar 8 Sequence diagram pembuatan rute alternatif

Untuk menggambarkan interaksi antar objek dalam sistem, didesainlah *sequence diagram*. *Sequence diagram* yang menggambarkan interaksi antar objek ketika terjadi proses pembuatan rute alternatif dilakukan (Gambar 8). *Sequence diagram* untuk proses manajemen *client* dapat kita lihat seperti Gambar 9.



Gambar 9 Sequence Diagram Manajemen Client

Sequence diagram lainnya adalah yang menggambarkan mengenai bagaimana proses ketika *user* mengunjungi dan melakukan *view* serta *request* rute alternatif ke tempat yang ingin dituju. *Sequence diagram* yang dimaksud dapat dilihat berdasarkan Gambar 10.



Gambar 10 Sequence diagram Request Rute

Implementasi Sistem

Tahapan selanjutnya adalah implementasi sistem, tahap dimana modul-modul desain diterjemahkan dalam bentuk program dengan menggunakan bahasa pemrograman untuk menghasilkan sebuah aplikasi. Sistem ini akan menggunakan beberapa bahasa pemrograman sesuai jenis aplikasinya. Untuk aplikasi penentuan titik kemacetan, karena ini akan dilakukan melalui telepon seluler dengan aplikasi berbasis android, maka bahasa pemrograman yang akan dipakai adalah *eclipse*. *Eclipse* adalah *Integrated Development Environment* (IDE) atau aplikasi yang dipakai untuk pengembangan java/android yang gratis dan dapat didownload melalui <http://www.eclipse.org/downloads/>. *Eclipse* saat ini telah melalui perkembangan dengan beberapa versi (Syafaat, 2012). Sementara itu, aplikasi untuk mendesain rute alternatif (pencarian rute) akan digunakan *Active Server Pages* (ASP.Net). Menurut Erick kurniawan (2010) ASP.NET adalah pengembangan terbaru dari *Microsoft Active Server Pages* (ASP) dan merupakan suatu teknologi *server side* yang baru dan handal untuk membuat halaman web dinamis. ASP.NET merupakan *platform* pembuatan aplikasi web yang menyatu dengan *.NET Framework* serta menyediakan fasilitas-fasilitas bagi developer untuk membangun aplikasi web untuk level *enterprise*. ASP.NET dikompilasi dan berbasis *.NET Framework* sehingga dapat membuat aplikasi dalam berbagai macam bahasa yang mendukung *.NET* (kompatibel dengan *.NET*), termasuk Visual Basic *.NET*, Jscript *.NET*.

ASP.NET didesain supaya dapat bekerja dengan WYSIWYG HTML editor dan tool pemrograman lain termasuk *Microsoft Visual Studio .NET*. Hal ini dapat membuat pembuatan aplikasi web lebih mudah, tetapi juga membuat berbagai fasilitas yang ada pada tool tersebut dapat digunakan, termasuk GUI (*Graphical User Interface*) dimana *developer* dapat *men-drag* and *drop* *server* kontrol ke halaman web dan menggunakan fasilitas *debugging* yang ada (Kurniawan, 2010).

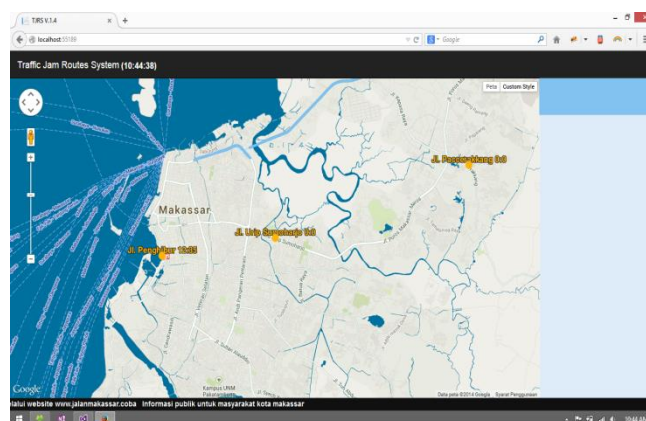
Pengujian

Tahap akhir dari desain penelitian ini adalah dengan menentukan jenis pengujian yang akan diterapkan pada sistem yang dibuat. Untuk aplikasi ini, akan diterapkan sistem pengujian *black box*, yaitu teknik pengujian yang berfokus pada sisi fungsionalitas. Pengujian terhadap input dan output aplikasi apakah sudah sesuai dengan yang diharapkan

atau belum. Pengujian *black box* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jika dianalogikan seperti melihat suatu kotak hitam, hanya bisa melihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. Sama seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya (*interface-nya*), fungsionalitasnya tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detailnya. Keunggulan pengujian ini adalah bisa memilih *subset test* secara efektif dan efisien, dapat menemukan *error*, memaksimalkan *testing investment*. Sementara itu pengujian ini juga memiliki kelemahan yaitu *tester* tidak pernah yakin apakah perangkat lunak tersebut benar (Taslim, 2012).

HASIL DAN PEMBAHASAN

Implementasi rancangan sistem ini membutuhkan beberapa komponen yang terdiri dari *hardware*, *software* dan *user interface*. *Hardware* dan *software* yang dipakai disesuaikan dengan alat dan bahan penelitian yang telah dijelaskan sebelumnya. Implementasi desain yang telah dirancang dituangkan dalam bahasa program yang dipakai dengan menggunakan *form-form* (*User Interface*).

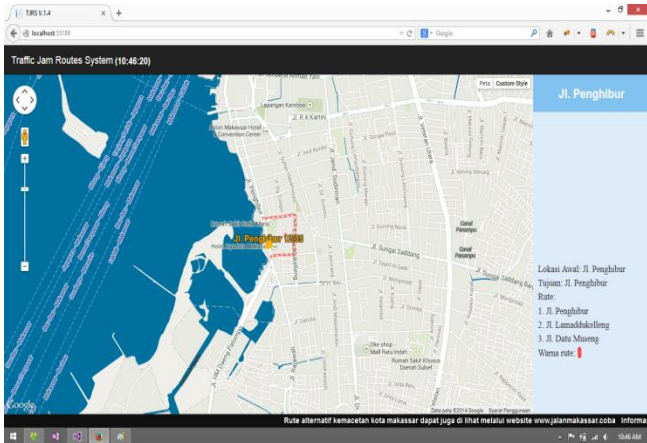


Gambar 11 Halaman View Kemacetan

Gambar 11 adalah tampilan pada saat aplikasi di akses untuk keperluan penyiaran umum dengan menggunakan media seperti LCD TV, Video Tron yang akan ditempatkan pada lokasi strategis agar informasi kemacetan dapat di sebarakan secara efektif untuk masyarakat atau bisa juga di siarkan umum melalui siaran TV lokal daerah.

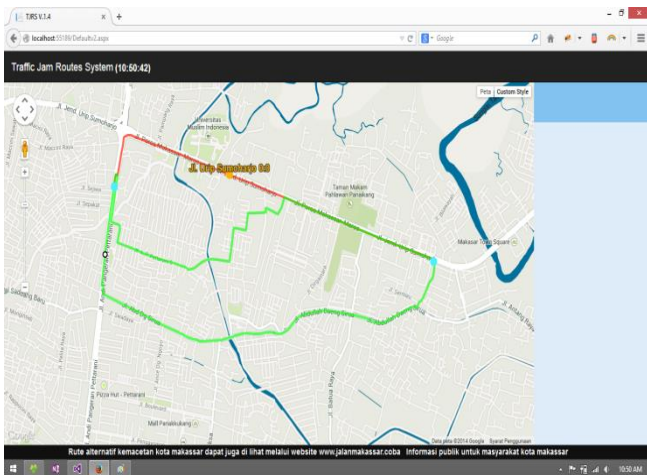
Saat pertama kali *load*, halaman tersebut akan menampilkan semua data kemacetan, selanjutnya setiap beberapa menit halaman tersebut akan *zoom-in*

ke titik-titik kemacetan secara bergantian dan menampilkan penjelasan rute alternatif yang berada di sebelah kanan peta jika sudah ada, seperti ditampilkan pada Gambar 12.

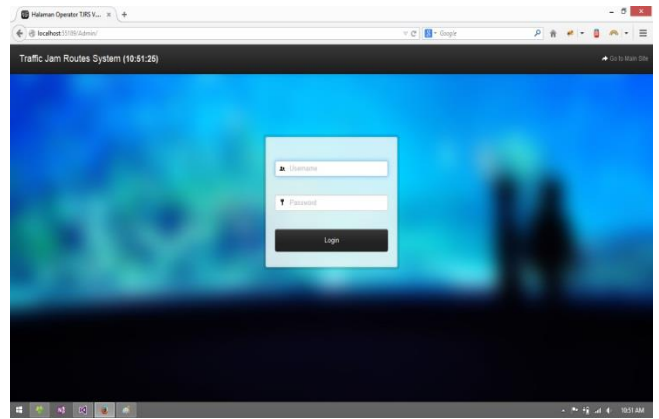


Gambar 12 Halaman Desain Rute

Pilihan lain bagi pengguna yang membutuhkan informasi kemacetan yaitu dapat mengakses aplikasi *Traffic Jam Routes System* dengan menggunakan media komputerisasi secara langsung, seperti menggunakan perangkat komputer PC, laptop, tablet atau *smartphone*. Cara kerja sistem yang diakses langsung oleh pengguna agak berbeda sedikit dengan cara kerja sistem yang dijelaskan sebelumnya, karena aplikasi pada saat ini akan bersifat spesifik kepada pengguna yang mengaksesnya. Setelah aplikasi berhasil di-load oleh *web browser*, pengguna dapat membuat *point* (*point* daerah asal dan tujuan), selanjutnya sistem akan memberikan sedikitnya satu rute dan maksimal tiga rute. Jika salah satu rute melewati daerah macet maka rute tersebut akan berwarna merah. Untuk lebih jelasnya Gambar 13 menunjukkan tampilan dari aplikasi yang dimaksud.

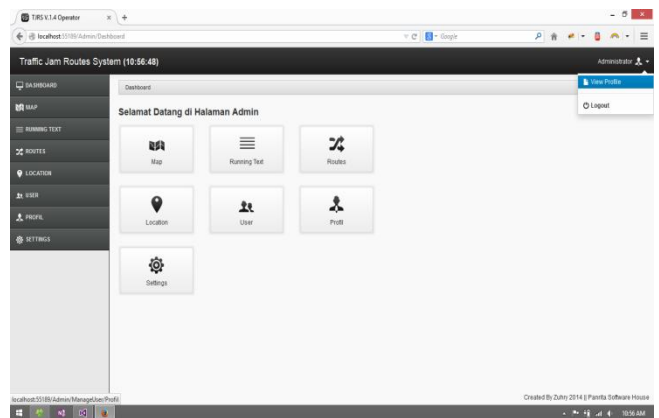


Gambar 13 Request Rute



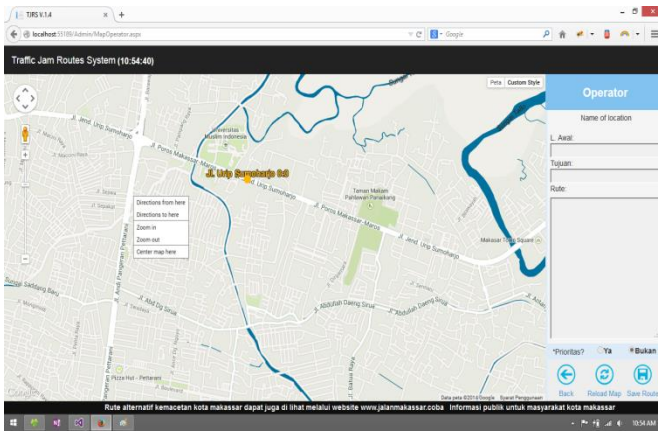
Gambar 14 Halaman Login

Gambar 14 adalah halaman *login* bagi pengguna/*user* yang bertindak sebagai operator sistem. Setelah sistem berhasil memvalidasi *username* dan *password* operator yang dimasukkan ke halaman *login* sistem, maka sistem akan menampilkan menu-menu yang akan dikelola oleh operator sistem, seperti ditunjukkan pada Gambar 15.



Gambar 15 Halaman Operator

Apabila operator melihat ada data baru berupa titik kemacetan berwarna merah yang dikirimkan oleh *client* maka operator bisa langsung membuat rute alternatif dengan melakukan klik kanan pada peta, pilih *direction from here* dan operator juga harus membuat *point* tujuan rute alternatif dengan klik kanan pada lokasi tujuan dan pilih *direction to here* maka sistem akan membuat rute dasar berupa garis yang selanjutnya operator dapat melakukan pengalihan rute dengan melakukan *drag* pada rute ke jalan yang ingin dituju, selanjutnya sistem akan mengambil dan menampilkan informasi yang akan di simpan ke database sistem, jika operator selesai melakukan validasi data maka operator menyimpan data ke database sistem dengan mengklik tombol *save* seperti yang ditunjukkan pada Gambar 16.



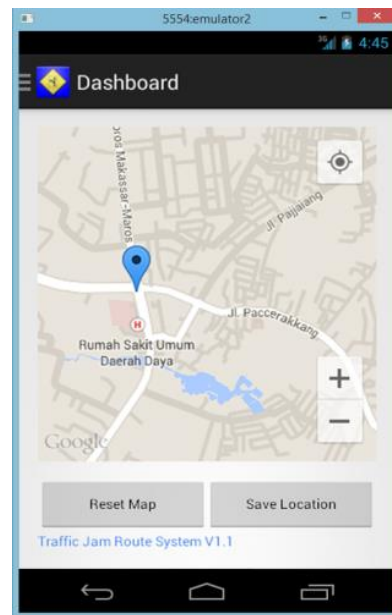
Gambar 16 Desain Rute Alternatif

Untuk melakukan manajemen data rute yang telah dibuat oleh operator, maka operator dapat mengaksesnya melalui menu rute yang ada di samping kiri aplikasi atau di halaman depan (dashboard) aplikasi *Traffic Jam Routes System* ini.

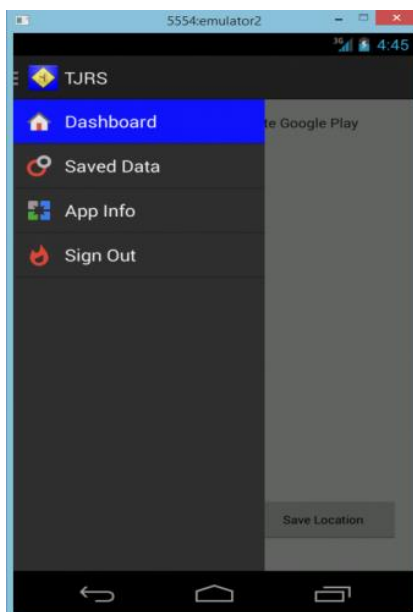
Gambar 17 adalah tampilan aplikasi khusus yang dibuat untuk *client* yang akan di *install* pada *smartphone client* berbasis android untuk mengirimkan data-data kemacetan dan memanajemen data kemacetan yang telah dikirimkan ke aplikasi pusat/operator.



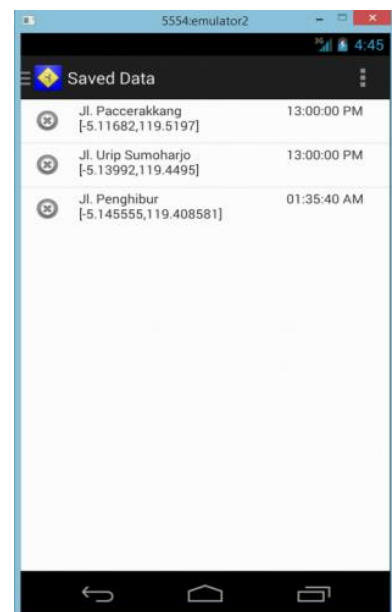
(a)



(b)



(c)



(d)

Gambar 17 Tampilan Halaman: a) Login Aplikasi di Android; b) Penentuan Koordinat Kemacetan di Android; c) Menu Aplikasi; d) Daftar Koordinat Kemacetan

Hasil Pengujian

Pengujian adalah proses untuk menemukan *error* pada perangkat lunak sebelum dikirim kepada pengguna. Pengujian *software* adalah kegiatan yang ditujukan untuk mengevaluasi atribut atau kemampuan program dan memastikan bahwa itu memenuhi hasil yang dicari, atau suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji (*under test*), pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya. Pengujian perangkat lunak yang dipakai untuk menguji kualitas

dari sistem yang dibangun adalah dengan menggunakan pengujian *black box*.

Tabel 1 memperlihatkan bahwa *output* yang diinginkan melalui beberapa skenario *testing* seperti mengosongkan *username* dan *password*, mengosongkan salah satunya, memasukkan *username* atau *password* yang salah dan memasukkan *username* dan *password* yang benar. Hasil pengujiannya menunjukkan hasil yang sesuai dengan yang diharapkan yang berarti halaman *login* dapat berjalan dengan baik. Berdasarkan Tabel 2, setelah dilakukan beberapa skenario pengujian dengan variasi input yang beragam menunjukkan hasil pengujian bahwa semua *tab* pada menu operator dapat berfungsi dengan baik sesuai dengan yang diharapkan.

Tabel 1 Hasil Pengujian Halaman *Login*.

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengosongkan <i>User Name</i> dan <i>Password</i> lalu mengklik <i>Login</i>	<i>User Name=""</i> <i>Password=""</i>	Sistem akan menolak akses dan memberikan pesan " <i>User Name</i> dan <i>Password</i> Tidak Sesuai"	Sesuai harapan	<i>Valid</i>
2	Hanya mengisi <i>User Name</i> dan mengosongkan <i>Password</i>	Nama=" <i>admin</i> " <i>Password=""</i>	Sistem akan menolak akses dan memberikan pesan " <i>Password</i> tidak boleh kosong"	Sesuai harapan	<i>Valid</i>
3	Hanya mengisi <i>Password</i> dan Mengosongkan <i>User Name</i>	<i>Password=""123</i> " <i>User name=""</i>	Sistem akan menolak akses dan memberikan pesan " <i>User Name</i> tidak boleh kosong"	Sesuai harapan	<i>Valid</i>
4	Menginput kondisi salah satu salah dan satu benar	<i>User name</i> = <i>admin</i> " <i>Password=""1234</i> "	Sistem akan menolak akses dan memberikan pesan " <i>User Name</i> dan <i>Password</i> Tidak Sesuai"	Sesuai harapan	<i>Valid</i>
5	Menginputkan data benar lalu mengklik	<i>User name</i> = <i>admin</i> " <i>Password=""123</i> "	Sistem menerima akses dan menampilkan <i>dashboard</i>	Sesuai harapan	<i>Valid</i>

Tabel 2 Hasil Pengujian Halaman Menu Operator

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengklik <i>tab Dashboard</i>	Klik <i>Dashboard</i>	Sistem akan tetap <i>te-refresh</i> dan tetap pada halaman operator	Sesuai harapan	<i>Valid</i>
2	Menguji <i>tab Routes</i>	Klik <i>Routes</i>	Sistem akan menampilkan data rute yang telah di desain	Sesuai harapan	<i>Valid</i>
3	Menguji <i>tab Location</i>	Klik <i>Location</i>	Sistem akan menampilkan data lokasi kemacetan dari <i>database</i>	Sesuai harapan	<i>Valid</i>
4	Menguji <i>tab User</i>	Klik <i>User</i>	Sistem akan menampilkan data <i>user</i> dari <i>database</i>	Sesuai harapan	<i>Valid</i>
5	Menguji <i>tab Setting</i>	Klik tombol <i>Setting</i>	Sistem akan menampilkan <i>page setting</i>	Sesuai harapan	<i>Valid</i>
6	Menguji <i>tab Map</i>	Klik tombol <i>Map</i>	Sistem akan menampilkan peta kemacetan yang telah ditentukan	Sesuai harapan	<i>Valid</i>

Pengujian halaman *request route* (Tabel 3) dengan skenario pengujian yang diberikan menunjukkan bahwa halaman tersebut telah terbebas dari *error* maupun *bug* sehingga halaman tersebut dapat berfungsi sesuai dengan yang diharapkan.

Halaman selanjutnya yang akan diuji adalah halaman yang dibuat pada aplikasi *mobile* berbasis android yaitu halaman untuk menentukan koordinat kemacetan yang akan memberikan nilai dan mengirim

posisi (*latitude, longitude*) ke server sebagai titik kemacetan. Skenario pengujian yang dilakukan sesuai Tabel 4.

Tabel tersebut memperlihatkan skenario pengujian dimana mulai pada penentuan koordinat dengan klik pada peta hingga *save* dan *reset* tombol dapat berfungsi dengan baik dan memberikan nilai yang sesuai dengan yang diharapkan.

Tabel 3 Hasil Pengujian Halaman *Request Route*

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengunjungi <i>Website</i>	<i>Browser:</i> “http://localhost:8080/Defaultv2”	Muncul koordinat seluruh titik-titik kemacetan	Sesuai harapan	<i>Valid</i>
2	Mengklik peta untuk menentukan lokasi awal rute yang diinginkan	Klik pada peta	Muncul tanda pada koordinat yang diklik sebagai patokan awal rute	Sesuai harapan	<i>Valid</i>
3	Mengklik peta untuk menentukan tujuan dari rute yang diinginkan	Klik pada peta	Muncul tanda pada koordinat yang diklik sebagai patokan akhir rute dan sistem akan menggambar rute dari titik awal ke titik tujuan	Sesuai harapan	<i>Valid</i>
4	Menggeser koordinat hingga salah satu rutenya melewati koordinat kemacetan	Geser koordinat tujuan	Rute yang melewati koordinat kemacetan akan berwarna merah dan yang tidak akan berwarna hijau	Sesuai harapan	<i>Valid</i>

Tabel 4 Hasil Pengujian Halaman Penentuan Koordinat Kemacetan

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengklik pada peta	Klik peta	Muncul pin pada peta sebagai penanda koordinat	Sesuai harapan	<i>Valid</i>
2	Menggeser pin ke koordinat yang diinginkan untuk menandai kemacetan	Geser pin	Pin akan berpindah ke koordinat di koordinat kemana pin itu digeser	Sesuai harapan	<i>Valid</i>
3	Klik tombol <i>save</i> untuk menyimpan koordinat ke <i>database</i>	Klik <i>save</i>	Koordinat kemacetan akan tersimpan ke database dan pada <i>view</i> peta muncul koordinat kemacetan baru	Sesuai harapan	<i>Valid</i>
4	Mengklik tombol reset map untuk mengembalikan map tanpa pin	Klik <i>reset map</i>	Peta kembali ke posisi seperti pertama kali aplikasi dibuka tanpa adanya koordinat (pin) kemacetan	Sesuai harapan	<i>Valid</i>

Halaman yang dilakukan skenario pengujian dengan menggunakan *black box* hanya pada *form-form* utama mengingat dalam aplikasi ini terdapat beberapa *form* pendukung terutama untuk melakukan setting oleh operator dan beberapa halaman pada aplikasi android yang dibuat yang pengujiannya kami tidak sertakan mengingat fungsi utama dari aplikasi yang dibangun telah terwakili oleh beberapa halaman yang telah dilakukan pengujian.

KESIMPULAN

Penelitian ini telah menghasilkan sebuah aplikasi berupa sistem informasi kemacetan lalu lintas yang dapat menampilkan solusi rute alternatif sehingga pengguna jalan dapat menghindari titik-titik kemacetan tersebut. Hasil pengujian menunjukkan bahwa sistem aplikasi dapat berfungsi dengan baik dan memberikan nilai yang sesuai dengan yang diharapkan. Implementasi aplikasi ini harapannya dapat mengurai kemacetan yang terjadi di Kota Makassar. Kedepannya, aplikasi ini diharapkan dapat diimplementasikan secara menyeluruh dan dimanfaatkan oleh pihak terkait untuk memberikan informasi kemacetan dan rute alternatif untuk menghindarinya kepada masyarakat. Aplikasi ini masih memiliki kekurangan dan masih sangat memungkinkan untuk dikembangkan dan disempurnakan sebelum diimplementasikan. Penelitian-penelitian selanjutnya diharapkan mampu menyempurnakan aplikasi ini dan dapat diintegrasikan dengan sistem ini diantaranya monitoring jalan raya dengan CCTV, Penentuan laju kendaraan di jalan raya Makassar, Perancangan rute alternatif kemacetan berbasis android serta integrasi monitoring posisi BRT Makassar. Efektifitas pengimplementasian sistem memungkinkan penambahan media distribusi informasi dengan bekerjasama dengan armada taksi dan angkutan darat yang ada, kerjasama dengan lembaga penyiaran radio dan televisi lokal serta pihak-pihak lain yang ingin berkontribusi dalam hal mengatasi kemacetan di kota Makassar.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu terlaksananya penelitian ini, khususnya kepada BBPPKI Makassar yang telah mendanai penelitian ini. penulis juga mengucapkan

terima kasih kepada Poltabes Makassar yang telah membantu memberikan informasi yang dibutuhkan dalam penelitian ini. Serta kepada Muhammad Zuhrri yang telah membantu dalam pembangunan sistem ini.

DAFTAR PUSTAKA

- Dong, W. (2011). *An overview of in-vehicle route guidance system. Australasian Transport Research Forum 2011 Proceedings* (pp. 1-12). Adelaide: ATRF.
- Duch, M. B. (2010). *Real Time Object Uniform Design Methodology With UML*. Springer Verlag.
- Fariza, A., Helen, A., & Mahyuzar, A. (2009, Agustus). Pencarian Jalur Alternatif Pada Daerah Bencana Lumpur Sidoarjo dengan Metode Analytic Hierarchy Process Berbasis WAP. *TELKOMNIKA*, 7(2), 137-144.
- Fowler, M. (2004). *UML Distilled: A Brief Guide to Standar Object Modeling*. Addison Wesley.
- Khan, M. A. (2013). Tinjauan Yuridis Terhadap Pelaksanaan Perparkiran di Kota Makassar. Makassar: Universitas Hasanuddin.
- Kurniawan, E. (2010). *Cepat Mahir ASP.Net 3.5*. Yogyakarta: Andi.
- Lesmana, Y., Indrabayu, & Ilham, A. A. (2015). Sistem Kendali Lampu Lalu Lintas dengan Pendekatan Fuzzy Logic. *Seminar nasional Komunikasi dan Informatika* (pp. 121 - 126). Makassar: BBPPKI Makassar.
- Liang, Z., & Wakahara, Y. (2014). *Real-time urban traffic amount prediction models for dynamic route guidance systems. EURASIP Journal on Wireless Communications and Networking*, 1 - 13.
- Muhtadin, Purnama, I. E., Zulfikar, & Hariadi, M. (2010, Oktober). Sistem Pencarian Rute Alternatif Dinamis Dengan Smart Routing System Menggunakan A* Algorithm Berbasis Web. *Journal of Electrical and Electronics Engineering*, 8(2), 6-10.
- Ratnasari, A., Ardiani, F., & A., N. F. (2013). Penentuan Jarak Terpendek dan Jarak Terpendek Alternatif Menggunakan Algoritma Dijkstra Serta Estimasi Waktu Tempuh. *Seminar Nasional Komunikasi dan Informatika Terapan*, (pp. 29 - 34). Semarang.
- Ryujiro, F., Inoue, H., & Naohiko. (2005). *Route Guidance System and Method*.
- Sharma, A., Jadhaf, A., Srivastava, R. P., & Goyal, R. (2010). *Test Cost Optimization Using Tabu Search. Software Engineering & Applications* 3, 477-486.
- Syafaat, N. (2012). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Taslim, A. (2012). *Black Box*. Retrieved 10 12, 2014, from Academia: <http://www.academia.edu/5574402/Blackbox>
- Varita, I., Setyawati, O., & Rahadi, D. (2013). Pencarian Jalur Tercepat Rute Perjalanan Wisata Dengan Algoritma Tabu Search. *Jurnal EECCIS*, 185-190.