

Pembangunan Aplikasi Pelaporan Informasi Jalan Rusak di Kota Malang

Abyan Naufal Amanullah Sugiharto¹, Faizatul Amalia², Djoko Pramono³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹abyan.sugiharto@gmail.com, ²faiz_amalia@ub.ac.id, ³djoko.jalin@ub.ac.id

Abstrak

Jalan sebagai prasarana transportasi dapat mengalami kerusakan yang disebabkan oleh berbagai hal. Dinas Pekerjaan Umum dan Penataan Ruang (DPUPR) merupakan otonomi daerah pada bidang infrastruktur yang meliputi jalan di kota Malang. Masyarakat dapat mengadukan jalan rusak kepada DPUPR melalui *website* pengaduan *online* milik pemkot karena DPUPR tidak memiliki sarana sendiri. Pada penelitian ini dikembangkan sebuah aplikasi pelaporan jalan rusak yang dapat menerima laporan masyarakat terkait informasi jalan rusak yang diterima oleh DPUPR. Aplikasi dikembangkan dengan metode *System Development Life Cycle Waterfall* yang terdiri dari enam tahap, yaitu studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian, kesimpulan dan saran. Pada penelitian ini dihasilkan 13 kebutuhan fungsional dan dua kebutuhan non-fungsional dengan tiga aktor pada analisis kebutuhan, dari kebutuhan tersebut dihasilkan perancangan berupa perancangan arsitektur, komponen, data dan antarmuka dengan hasil implementasi berupa implementasi kode, data dan antarmuka. Hasil dari implementasi diuji dengan *Blackbox testing* menggunakan pengujian validasi dengan tiga kasus uji 100% valid dan *Whitebox testing* menggunakan pengujian unit dengan tiga sampel 100% valid, pengujian usabilitas dilakukan dengan metode *System Usability Scale* dengan enam responden yang menghasilkan skor 75.25, pengujian kompatibilitas juga dilakukan dengan menggunakan *Firestore Test Lab* dengan hasil aplikasi dapat dijalankan pada lima perangkat berbeda dengan tingkat API yang berbeda.

Kata kunci: *android, dpupr, pelaporan, jalan, rusak*

Abstract

Road as an infrastructure for transportation can be damaged by various causes. Dinas Pekerjaan Umum dan Penataan Ruang (DPUPR) is a regional autonomy in infrastructure which covers road. Community complains damaged roads via government's online complaint website due to DPUPR's absence of own platform. This research develops an application which accomodates complains from community regarding damaged roads, the complaints are then received by DPUPR first hand. The application is developed using Waterfall System Development Life Cycle consisting of literature study, requirement analysis, design, implementation, testing, conclusions and suggestions. The results from this research are 13 functional requirements and two non-functional requirements involving three actors on requirement analysis, these requirements generate design and implementation i.e. architecture design, component design, data design and interface design, and implements code, data and interface. The results of implementation are tested with validation testing using Blackbox testing with three test cases resulted 100% validity and unit testing using Whitebox testing with three samples resulted 100% validity, usability testing is conducted with System Usability Scale involving six respondents which resulted score of 75.25, compatibility testing is conducted using Firestore Test Lab which resulted that the application can be run on five different devices with different API levels.

Keywords: *android, public works, reporting, road, damage*

1. PENDAHULUAN

Jalan beraspal merupakan prasarana yang dimiliki oleh peradaban modern di kota-kota besar maupun di pedesaan. Jalan beraspal memungkinkan sarana transportasi berjalan

lancar disebabkan permukaan aspal yang rata dan memiliki gaya gesek yang kuat sehingga kendaraan tidak mudah tergelincir. Jalan yang beraspal juga dapat rusak seiring waktu berjalan, penyebabnya pun beragam, dimulai dari erosi

yang disebabkan oleh air hujan maupun faktor eksternal lainnya. Jalan rusak dapat menyebabkan ketidaknyamanan bagi kendaraan yang melewatinya bahkan dapat menyebabkan kecelakaan.

Dinas Pekerjaan Umum dan Penataan Ruang (DPUPR) Kota Malang merupakan dinas yang melakukan otonomi daerah di sektor infrastruktur jalan, jembatan, gedung, drainase, air minum dan air limbah, pembinaan konstruksi, penataan daerah strategis dan tata ruang di Kota Malang. Seluruh jalan rusak yang terdapat di Kota Malang ditangani DPUPR di bidang infrastruktur jalan. DPUPR melakukan survey lapangan agar dapat menemukan lokasi dimana saja terdapat jalan rusak yang butuh penanganan. Selain survey lapangan, DPUPR dapat menangani laporan jalan rusak aduan dari masyarakat melalui pengaduan online di website milik DPUPR. Pengaduan yang diterima oleh DPUPR akan dipastikan kebenarannya dengan survey lapangan, setelah dikonfirmasi kebenaran pengaduan jalan rusak yang diterima, DPUPR akan memulai proses penanganan jalan rusak dengan menyewa jasa pihak ketiga dengan supervisi dari DPUPR.

Berdasarkan wawancara yang dilakukan, pihak DPUPR merasa bahwa secara keseluruhan proses administrasi pengaduan jalan rusak kurang efisien dalam jangka pelaksanaannya. Menurut pihak DPUPR, laporan jalan rusak dikirimkan oleh masyarakat melalui pengaduan online yaitu website milik Pemerintah Kota Malang, DPUPR sendiri tidak memiliki sarana untuk menerima laporan jalan rusak secara langsung. Masyarakat yang melakukan pengaduan online pun tidak dapat mengetahui laporan jalan rusak kepada pihak DPUPR sudah diterima atau belum, selain itu proses penanganan jalan rusak yang dilaporkan tidak dapat diketahui karena respon terkait aduan masyarakat pada website milik Pemerintah Kota Malang tidak dilakukan oleh DPUPR sendiri.

Berdasarkan permasalahan yang telah dipaparkan diatas dapat disimpulkan bahwa masih terdapat inefisiensi proses administrasi penanganan jalan rusak di Kota Malang. Inefisiensi proses administrasi penanganan jalan rusak yang menjadi permasalahan tersebut meliputi: Inefisiensi penanganan jalan rusak karena DPUPR tidak menerima laporan terkait jalan rusak secara langsung disebabkan oleh pihak DPUPR tidak memiliki platform sendiri untuk menerima laporan jalan rusak dari masyarakat, total laporan pada website sambat

online sampai tahun 2020 sudah mencapai lebih dari 900 laporan, tangkapan layar terkait jumlah laporan dapat dilihat pada lampiran B. Masyarakat yang mengadu hanya mendapat respon terkait aduannya dari Pemerintah Kota Malang bukan DPUPR itu sendiri, selain itu DPUPR tidak dapat menginformasikan proses penanganan jalan rusak di Kota Malang.

Solusi yang dapat menyelesaikan permasalahan-permasalahan yang telah dikemukakan adalah sebuah platform tersendiri bagi pihak DPUPR berupa sistem aplikasi pelaporan informasi berbasis android yang dapat menerima aduan jalan rusak secara langsung dari masyarakat. Mobilitas dan kapabilitas dari perangkat berbasis android yaitu kemampuan untuk mengunggah informasi berupa foto dan deskripsi jalan rusak di tempat yang dilaporkan oleh masyarakat. Selain itu, teknologi seperti *Google Maps* yang dapat menampilkan peta dapat diintegrasikan dengan sistem aplikasi melalui *Application Programming Interface* (API) untuk menandai lokasi jalan rusak yang dilaporkan. Informasi jalan rusak pada aplikasi akan diterima dan dikelola oleh admin pada sistem yang dipegang oleh Kepala Bidang Bina Marga DPUPR. Kepala Bidang Bina Marga DPUPR merupakan pegawai DPUPR yang bertugas untuk mencari laporan jalan rusak di website pemerintah Kota Malang dapat menerima dan mengolah laporan jalan rusak oleh masyarakat. Admin dapat memberi konfirmasi bahwa laporan jalan rusak sudah diterima melalui status laporan pada sistem, hingga informasi proses penanganan jalan rusak agar masyarakat dapat mengetahui jalan rusak mana yang sedang ditangani.

Berdasarkan permasalahan-permasalahan yang telah dikemukakan, penulis mempunyai gagasan untuk membuat judul penelitian "PEMBANGUNAN APLIKASI PELAPORAN INFORMASI JALAN RUSAK DI KOTA MALANG" yang akan dibangun dengan *Software Development Life Cycle* (SDLC) metode *Waterfall* dengan harapan hasil dari penelitian yang dilakukan dapat membantu DPUPR saat mengelola laporan terkait jalan rusak yang ada di Kota Malang.

2. LANDASAN KEPUSTAKAAN

2.1 Dinas Pekerjaan Umum Dan Penataan Ruang

Dinas Pekerjaan Umum dan Penataan Ruang (DPUPR) merupakan pelaksana otonomi

daerah pada sektor konstruksi infrastruktur jalan, jembatan, gedung, drainase, air minum dan air limbah, pembinaan jasa konstruksi, penataan kawasan strategis serta penataan ruang (Dinas Pekerjaan Umum dan Penataan Ruang, 2019).

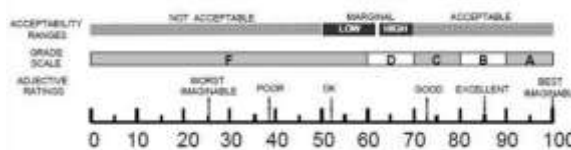
2.2 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak ialah pengetahuan terkait rekayasa yang berhubungan dengan seluruh aspek dari penerapan perangkat lunak (Sommerville, 2011). Dasar-dasar dari Rekayasa Perangkat Lunak adalah spesifikasi perangkat lunak, pengembangan perangkat lunak, validasi perangkat lunak dan perubahan perangkat lunak.

2.3 Pengujian Perangkat Lunak

Pengujian merupakan cara yang digunakan untuk melakukan verifikasi dan validasi dari perangkat lunak, dari hasil pengujian ini dapat dinilai kualitas perangkat lunak tersebut. Pengujian merupakan tindakan melatih perangkat lunak dengan sejumlah kasus uji yang bertujuan mencari kesalahan / error dan mendemonstrasikan bahwa perangkat lunak berjalan dengan benar (Jorgensen, 2013). Pengujian yang dilakukan berupa pengujian validasi menggunakan metode *Blackbox testing*, pengujian unit menggunakan *Whitebox testing*, pengujian usabilitas menggunakan *System Usability Scale (SUS)*, dan pengujian kompatibilitas menggunakan *Firebase Test Lab*.

Pengujian usabilitas yang diaplikasikan pada penelitian ini ialah *System Usability Scale (SUS)* 10 pernyataan dengan lima tingkat kesetujuan, mulai sangat tidak setuju sampai sangat setuju, masing-masing memiliki skor mulai dari 1 sampai 5. Seluruh pernyataan ganjil memiliki konotasi positif sedangkan pernyataan genap memiliki konotasi negatif (Brooke, 1996). SUS menggunakan skala Likert dengan bobot pernyataan ganjil dihitung dengan cara bobot = (skor - 1) sedangkan bobot pernyataan genap adalah bobot = (5 - skor). Hasil SUS didapat dengan menjumlahkan bobot total dikali dengan 2,5. Rata-rata dari seluruh bobot yang didapatkan dari responden yang terlibat merupakan skor SUS. Skala usabilitas sistem dapat dinilai dari *SUS Score* pada Gambar 1.



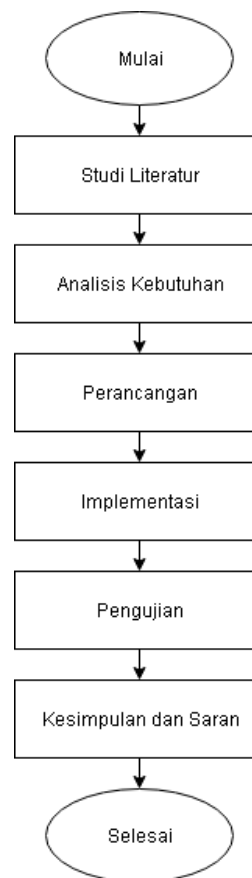
Gambar 1. SUS score

Sumber: (Bangor, 2009)

3. METODOLOGI

Pada penelitian ini digunakan tipe penelitian implementatif-pengembangan. Penelitian ini bertujuan untuk menghasilkan sebuah produk perangkat lunak.

Pengembangan perangkat lunak tersebut menggunakan prinsip rekayasa perangkat lunak yaitu analisis, perancangan, implementasi, dan pengujian. Metodologi dalam penelitian ini terdiri dari enam tahap, yaitu studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian, kesimpulan dan saran. Diagram alir atau *flowchart* mengenai penelitian ini akan ditampilkan pada Gambar 2.



Gambar 2. Diagram alir penelitian

3.1. Studi Literatur

Dalam tahapan ini dilakukan pengambilan literatur sebagai acuan dalam penelitian ini.

Literatur yang digunakan sebagai sumber didapatkan dari buku, jurnal, dan artikel yang ada di internet. Berikut merupakan teori yang digunakan dalam penelitian antara lain:

1. Tinjauan Pustaka
2. Dinas Pekerjaan Umum Dan Penataan Ruang
3. Rekayasa Perangkat Lunak
 - a. Pengembangan Perangkat Lunak
 - b. Pendekatan Berorientasi Objek
 - c. Pemodelan Berorientasi Objek
4. Android
5. Android Studio
6. Java
7. *Application Programming Interface*
8. *Database*
9. Pengujian Perangkat Lunak
 - a. Pengujian Validasi
 - b. Pengujian Unit
 - c. Pengujian Usabilitas
 - d. Pengujian Kompatibilitas

3.2. Analisis Kebutuhan

Analisis kebutuhan pada penelitian ini dilakukan dengan elisitasi menggunakan teknik wawancara secara langsung dengan pihak DPUPR. Wawancara dilakukan kepada Kepala Bidang Bina Marga DPUPR yang mengelola laporan jalan rusak di Kota Malang, Bapak Slamet Santosa. Wawancara dengan Kepala Bidang Bina Marga DPUPR dilakukan untuk mencari tahu alur pengaduan jalan rusak, serta proses penanganan jalan rusak di Kota Malang.

Tahapan penting ini digunakan sebagai dasar untuk mendeskripsikan kebutuhan mana yang akan dibutuhkan bagi sistem nantinya. Analisis kebutuhan berisi kebutuhan fungsional dari sistem untuk digunakan oleh aktor (pengguna sistem) yang nantinya dirumuskan dengan wujud *use case diagram* juga *use case scenario*.

3.3. Perancangan

Perancangan ialah fase yang dikerjakan saat kebutuhan terdefiniskan di fase analisis kebutuhan sudah didapatkan. Perancangan dilakukan dengan menggunakan pendekatan *Object-oriented*. Di tahapan ini dibuat perancangan arsitektur berupa *sequence diagram* dan *class diagram*, perancangan komponen berupa algoritma, perancangan basis data, juga perancangan antarmuka berupa *wireframe*.

3.4. Implementasi

Fase ini menjelaskan prosedur implementasi sistem berdasar dari perancangan yang sebelumnya dilakukan. Implementasi yang dilakukan berupa implementasi arsitektur dan komponen berupa kode program menggunakan bahasa pemrograman Java dengan bantuan Android Studio, implementasi basis data menggunakan *Firebase*, dan implementasi antarmuka menggunakan XML.

3.5. Pengujian

Dalam tahap ini terdapat penjelasan terkait pengujian sistem. Pengujian sistem mencakup kinerja dan performa dari sistem yang sudah dibangun. Pengujian dilaksanakan dengan tujuan membandingkan apa sudah sesuai kebutuhan yang didapat dengan implementasinya.

3.6. Kesimpulan dan Saran

Kesimpulan ialah jawaban tentang rumusan masalah yang sudah dirumuskan pada pendahuluan. Saran diberikan sebagai penyempurnaan bagi penelitian selanjutnya jika dilakukan pengembangan lanjutan sistem.

4. ANALISIS KEBUTUHAN

Di tahap ini dilaksanakan analisis kebutuhan pada pengembangan sistem dengan hasil yang didapatkan dari hasil wawancara dengan Dinas Pekerjaan Umum Kota Malang. Dari hasil wawancara didapatkan permasalahan-permasalahan yang menjadi dasar untuk spesifikasi dan pemodelan kebutuhan sistem pada tahapan ini.

4.1. Identifikasi Aktor

Identifikasi aktor dilaksanakan agar ditemukan apa atau siapa nanti yang berinteraksi kepada sistem. Detail terkait aktor di penelitian ini dideskripsikan pada Tabel 1.

Tabel 1. Identifikasi Aktor

Nama Aktor	Deskripsi
Pengguna	Pengguna adalah semua yang memanfaatkan sistem, pengguna dapat melihat informasi jalan rusak pada aplikasi
Anggota	Anggota adalah pengguna yang sudah terdaftar, anggota dapat melihat dan membuat informasi jalan rusak

	pada aplikasi
Admin	Merupakan pengguna yang dapat mengelola informasi yang dilaporkan oleh anggota seperti mengkonfirmasi, menandai bahwa jalan rusak telah tidak ada dan menghapus informasi jalan rusak pada aplikasi

4.2. Spesifikasi Kebutuhan

Spesifikasi kebutuhan dilaksanakan agar dapat diketahui apa saja kebutuhan dari aktor-aktor yang telah diidentifikasi. Spesifikasi kebutuhan terdiri atas spesifikasi kebutuhan fungsional juga kebutuhan non fungsional. Pada spesifikasi kebutuhan terdapat kode kebutuhan yaitu format IJR-F-XX bagi kebutuhan fungsional serta IJR-NF-XX bagi kebutuhan non fungsional, IJR memiliki kepanjangan yaitu Informasi Jalan Rusak, F juga NF adalah kepanjangan dari Fungsional juga Non Fungsional, sedangkan XX ialah nomor kebutuhan. Deskripsi kebutuhan dijelaskan pada Tabel 2 dan Tabel 3.

Tabel 2. Kebutuhan Fungsional

No.	Nama	Deskripsi
1	Melihat informasi jalan rusak	Sistem dapat menampilkan informasi jalan rusak
2	Melihat lokasi	Sistem dapat menampilkan lokasi aktor pada <i>map</i>
3	<i>Login</i>	Sistem menyediakan fungsi <i>login</i>
4	<i>Register</i>	Sistem menyediakan fungsi <i>register</i>
5	Membuat informasi jalan rusak	Sistem menyediakan fungsi untuk membuat informasi jalan rusak
6	Menilai informasi jalan rusak	Sistem menyediakan fungsi untuk menilai informasi jalan rusak
7	Menyunting profil	Sistem menyediakan fungsi menyunting profil
8	<i>Logout</i>	Sistem menyediakan fungsi <i>logout</i>
9	Melihat navigasi ke lokasi informasi jalan rusak	Sistem dapat menampilkan navigasi ke lokasi informasi jalan rusak
10	Mengubah status informasi jalan rusak	Sistem menyediakan fungsi untuk mengubah status informasi jalan rusak
11	Menghapus informasi jalan rusak	Sistem menyediakan fungsi untuk menghapus informasi jalan rusak
12	Melihat profil	Sistem menyediakan

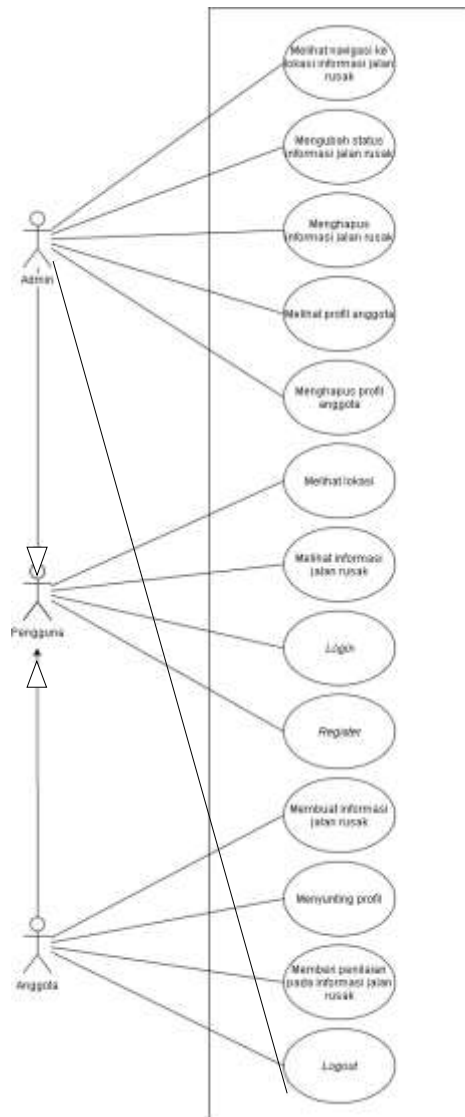
	anggota	fungsi untuk melihat profil anggota
13	Menghapus profil anggota	Sistem menyediakan fungsi untuk menghapus profil anggota

Tabel 3. Kebutuhan Non Fungsional

No.	Nama	Deskripsi
1	<i>Compatibility</i>	Aplikasi dapat digunakan pada API 25-29
2	<i>Usability</i>	Aplikasi memungkinkan untuk digunakan secara mudah bagi pengguna juga memiliki usabilitas melebihi 60%

4.3. Pemodelan Kebutuhan

Pemodelan kebutuhan dilakukan untuk memperdalam pengertian terhadap analisis kebutuhan. Pada pemodelan kebutuhan terdapat *use case diagram* dan *use case scenario*. *Use case diagram* aplikasi pelaporan informasi jalan rusak dapat dilihat pada Gambar 3.



Gambar 3. Use case diagram

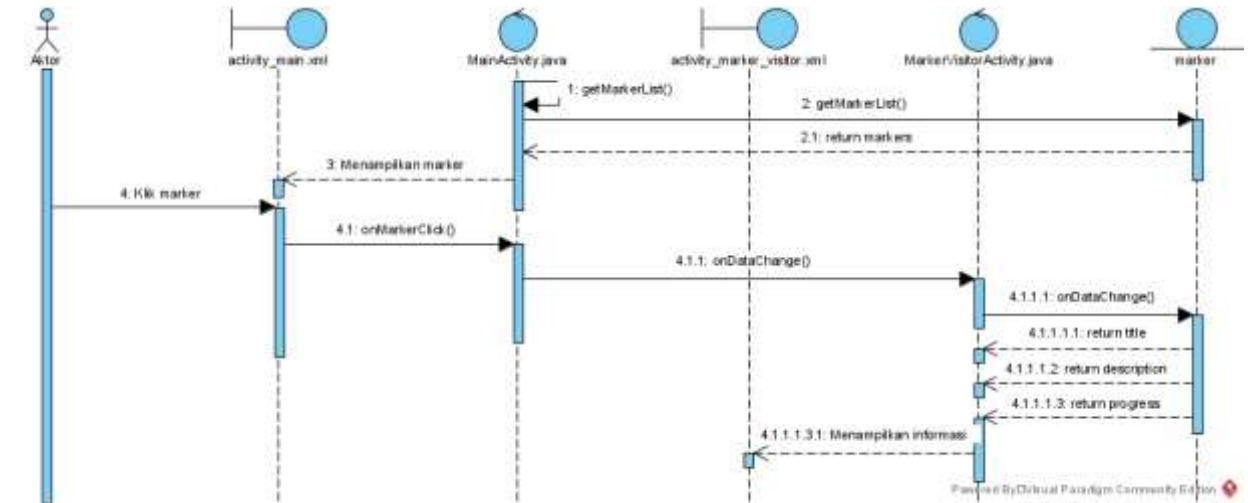
5. PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

Perancangan sistem dilakukan berdasarkan hasil analisis kebutuhan, dalam perancangan system terdapat perancangan arsitektur, perancangan komponen, perancangan data, dan perancangan antarmuka sebagai dasar dari implementasi sistem. Dalam perancangan sistem

komponen dan perancangan antarmuka dari sistem.

Dalam perancangan arsitektur terdapat *sequence diagram*, ditunjukkan alur sistem yang dibagi berdasarkan fungsinya serta objek-objek yang terlibat pada fungsi tersebut. *Sequence diagram* dibuat berdasarkan *use case scenario*.



ini terdapat perancangan arsitektur, perancangan

Berikut adalah *sequence diagram* melihat informasi jalan rusak pada Gambar 4.

Gambar 4. *Sequence diagram* melihat jalan rusak

Perancangan antarmuka melihat informasi jalan rusak dapat dilihat pada Gambar 5.

Implementasi antarmuka melihat informasi jalan rusak dapat dilihat pada Gambar 6.



Gambar 5. Perancangan antarmuka melihat informasi jalan rusak



Gambar 6. Implementasi antarmuka melihat informasi jalan rusak

5.2 Implementasi Sistem

Dalam spesifikasi sistem dijelaskan mengenai spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk pembangunan aplikasi pelaporan informasi jalan rusak.

6. PENGUJIAN

6.1 Pengujian Validasi

Pengujian validasi dilakukan dengan menggunakan *Blackbox testing* yaitu menguji fungsionalitas sebuah sistem sesuai prosedur skenario yang sudah ditetapkan dan memvalidasi kesesuaian hasil dengan ekspekstasi dari fungsionalitas tersebut. Pengujian validasi ditunjukkan pada Tabel 4.

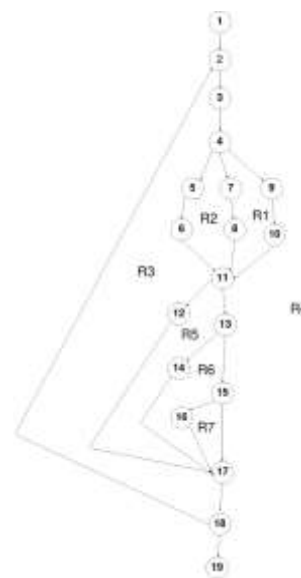
Tabel 4. Pengujian Validasi Melihat Informasi Jalan Rusak

Kode Kebutuhan	IJR-F-1.1
Nama Kasus Uji	Melihat Informasi Jalan Rusak
Prosedur	<ul style="list-style-type: none"> • Sistem berada pada halaman awal berupa map • Aktor mengklik salah satu marker yang ditampilkan • Sistem berada pada informasi jalan rusak
Expected Result	Sistem berada pada informasi jalan rusak
Result	Sistem berada pada informasi jalan rusak
Status	Valid

Blackbox testing yang dilakukan pada tiga kasus uji kebutuhan fungsional melihat informasi jalan rusak, membuat informasi jalan rusak, dan mengubah status informasi jalan rusak dalam penelitian ini dengan hasil valid pada semua kasus uji.

6.2 Pengujian Unit

Pengujian unit dilakukan dengan menggunakan *Whitebox testing*. *Whitebox testing* dilakukan dengan membuat grafik aliran dari *pseudocode* kode program sebuah fungsionalitas dan mencari jalur independen untuk prosedur ujinya. Gambar 8 menjelaskan grafik aliran dari *pseudocode* kode program melihat informasi jalan rusak.



Gambar 7. Grafis aliran melihat informasi jalan rusak

Whitebox testing dilakukan pada tiga fungsi, yaitu melihat informasi jalan rusak, membuat informasi jalan rusak, dan mengubah status informasi jalan rusak. Pada setiap kasus uji menghasilkan nilai valid.

6.3 Pengujian Usabilitas

Pengujian usabilitas dilakukan dengan *System Usability Testing* terhadap lima responden dari aktor anggota dengan hasil 70,5 dan satu responden dari aktor admin dengan hasil 80. Rata-rata dari hasil kedua pengujian tersebut adalah 75.25 yang berarti usabilitas masuk kedalam kategori *acceptable* dengan *rating good*.

6.4 Pengujian Kompatibilitas

Pengujian kompatibilitas dilakukan dengan memanfaatkan *Firebase Test Lab*, aplikasi yang dikembangkan diunggah dan dijalankan oleh *Firebase Test Lab* untuk dijalankan pada perangkat keras secara nyata. Hasilnya perangkat yang menjalankan aplikasi pada pengujian ini adalah Razer Phone dengan *level API 25*, Pixel dengan *level API 26*, Huwaei Honor Play dengan *level API 27*, Moto G7 Plus XT19652 dengan *level API 28*, dan Pixel 4 dengan *level API 29*.

7. PENUTUP

7.1 Kesimpulan

Dalam penelitian yang dilakukan, telah dilalui tahapan-tahapan pembangunan perangkat lunak dengan metode SDLC *waterfall*. Tahapan-

tahapan tersebut antara lain: analisis kebutuhan, perancangan dan implementasi sistem, dan pengujian sistem. Tahapan-tahapan tersebut merupakan rumusan masalah dari penelitian ini, dari penelitian yang telah dilakukan didapatkan kesimpulan sebagai berikut:

Analisis kebutuhan dilakukan dengan elisitasi menggunakan metode wawancara, dari hasil analisis kebutuhan diidentifikasi tiga aktor dan fungsionalitas berupa 13 kebutuhan fungsional. Berdasarkan hasil analisis kebutuhan dilakukan perancangan yang menghasilkan perancangan arsitektur berupa *sequence diagram* dan *class diagram*, perancangan komponen berupa *pseudocode*, perancangan data berupa struktur basis data, dan perancangan antarmuka berupa *wireframe*. Implementasi dilakukan berdasarkan perancangan yang telah dilakukan, hasil dari implementasi perancangan adalah implementasi kode program menggunakan bahasa pemrograman Java dengan Android Studio, implementasi basis data menggunakan *Firebase* dari Google, dan implementasi antarmuka menggunakan XML. Pengujian dilakukan terhadap hasil implementasi yang sudah dilakukan berupa pengujian validasi dengan metode *Blackbox testing* dengan hasil valid pada tiga kasus uji, pengujian unit dengan metode *Whitebox testing* dengan hasil valid pada tiga kasus uji, pengujian usability menggunakan *System Usability Testing* dengan skor 75.25 dengan kategori *acceptable* dan *rating good*, pengujian kompatibilitas dilakukan dengan *Firebase Test Lab* dengan hasil aplikasi dapat dijalankan pada lima perangkat berbeda dengan tingkat API yang berbeda.

7.2 Saran

Setelah penelitian Pembangunan Aplikasi Pelaporan Informasi Jalan Rusak di Kota Malang dilakukan, didapatkan beberapa saran untuk penelitian selanjutnya dalam pengembangan lebih lanjut, yaitu:

1. Sistem dapat dikembangkan dengan penambahan *web interface* yang tersedia pada *browser* agar dapat diakses melalui *desktop*.
2. Validasi jalan rusak dapat dikembangkan dengan memanfaatkan pencocokkan gambar dalam informasi jalan rusak dengan kondisi nyata jalan rusak via citra *digital*.

8. DAFTAR REFERENSI

- Brooke, J., 1996. SUS - A quick and dirty usability scale. In: B. T. I. L. M. B. W. Patrick W. Jordan, ed. *Usability Evaluation In Industry*. s.l.:CRC Press, pp. 189-196.
- Dinas Pekerjaan Umum dan Penataan Ruang, 2019. *Profil Dinas Pekerjaan Umum Dan Penataan Ruang*. [Online] Available at: <http://www.dpuprpkp.malangkota.go.id/profil> [Accessed 9 May 2020].
- Jorgensen, P. C., 2013. *Software Testing: A Craftsman's Approach, Fourth Edition*. 4th ed. s.l.:CRC Press.
- Sommerville, I., 2011. *Software Engineering*. 9th ed. s.l.:Pearson.