

Sistem Pemandu Pencarian Tambal Ban Terdekat Menggunakan Location-Based Services Pada Platform Android

Fahrizal¹, Ardiansyah², Maya Fitria³

^{1, 2, 3}Jurusan Teknik Elektro dan Komputer, Fakultas Teknik, Universitas Syiah Kuala
Jl. Tgk. Syech Abdurrauf No. 7 Darussalam, Banda Aceh 23111, Indonesia

¹fahri13zal@gmail.com

²ardiansyah@unsyiah.ac.id

³mayafitria@unsyiah.ac.id

Abstrak— Banyak kendala yang sering muncul ketika pengendara motor mengalami kebocoran ban diantaranya tidak mengetahui lokasi tempat tambal ban yang terdekat, lokasi tempat tambal ban yang sudah berpindah dan mungkin tempat tambal ban tersebut tidak buka. Dengan banyaknya pertimbangan dalam memilih tempat tambal ban seperti jarak tempuh, buka atau tidak tempat tersebut serta untuk mempercepat dalam proses pencarian, maka dibutuhkan sebuah aplikasi *smartphone* yang berperan untuk membantu dalam memecahkan masalah tersebut. Oleh karena itu, tugas akhir ini bertujuan untuk membuat aplikasi *mobile* yang dapat menunjukkan tempat tambal ban yang masih beroperasi dimana lokasi tersebut terdekat dengan posisi pengguna. Metode pembuatan *software* terdiri dari perencanaan, desain aplikasi, pembuatan aplikasi dan evaluasi aplikasi. Aplikasi tersebut dibuat menggunakan bahasa pemrograman *java* yang dijalankan pada aplikasi *android studio*. Hasil *unit testing* dan *usability testing* pada saat evaluasi aplikasi menunjukkan aplikasi layak untuk digunakan baik untuk pemilik tambal ban maupun untuk pengendara motor.

Kata Kunci— Bocor ban, *java*, *android studio*, *smartphone*, *unit testing*, *usability testing*.

I. PENDAHULUAN

Peningkatan penggunaan *smartphone* dari hasil riset lembaga digital marketing Emarketer, diperkirakan pada 2018 jumlah pengguna *smartphone* yang aktif di Indonesia lebih dari 100 juta orang [1]. Dengan adanya *smartphone* dan internet, kini para pengguna dapat semakin mudah dalam mencari berbagai informasi, seperti tempat makan, lokasi stasiun pengisian bahan bakar umum (SPBU), lokasi warung kopi, dan berbagai informasi lainnya. Banyak kendala yang sering muncul, misalnya seperti lokasi yang terlalu jauh untuk dijangkau, atau tidak adanya perubahan informasi.

Ada beberapa aplikasi pencarian lokasi, di antaranya adalah aplikasi pencarian tambal ban motor terdekat berbasis android (studi kasus tambal ban di kota Batam) [2], pada aplikasi ini terdapat beberapa permasalahan diantaranya untuk bisa merubah atau menambahkan titik lokasi tempat tambal ban terbaru hanya bisa dilakukan oleh pembuat aplikasi. Aplikasi pencarian lokasi lainnya yaitu penerapan

location-based services untuk pembuatan aplikasi pencarian tempat tambal ban berbasis android [3]. Pada aplikasi ini sudah banyak penambahan fitur dari pada aplikasi sebelumnya, antara lain aplikasi tersebut sudah mencakup area yang lebih luas, namun untuk penambahan lokasi yang sudah di *set* oleh pemilik tambal ban, harus menunggu persetujuan dari admin, sehingga memerlukan waktu untuk menunggu persetujuan apakah titik lokasi dari pemilik tambal ban diterima atau tidak.

Berdasarkan permasalahan pada aplikasi *mobile* yang telah ada tersebut, dilakukanlah penelitian untuk analisis kebutuhan dalam merancang perangkat lunak pada *smartphone* Android berbasis *location-based service (LBS)* dan *global positioning system (GPS)*. Aplikasi yang akan dibangun ini memiliki fitur yaitu dapat mengetahui tempat tambal ban yang masih buka, adanya fitur panggil bagi pengendara motor yang berguna untuk memudahkan interaksi dengan pemilik tambal ban, serta mengetahui lokasi tempat tambal ban terdekat.

II. DASAR TEORI

A. Global Positioning System (GPS)

Global Positioning System (GPS) merupakan sistem navigasi global yang dapat menerima informasi dari satelit. Informasi yang diterima yang berasal dari satelit *GPS* yang mengorbit di atas permukaan bumi. Satelit *GPS* memancarkan sinyal *GPS* yang memungkinkan penerima sinyal *GPS* untuk mendapatkan informasi berupa lokasi penerima, arah, dan kecepatan. Satelit *GPS* diuji coba dan pertama kali diluncurkan pada tahun 1978. Pada saat itu *GPS* sudah menjadi alat bantu navigasi diseluruh dunia dan menjadi alat yang penting untuk melakukan pembuatan peta dan *survey* wilayah [4].

GPS juga menyediakan acuan waktu yang tepat yang digunakan di banyak aplikasi termasuk studi ilmu gempa bumi dan sinkronisasi jaringan telekomunikasi. Sebuah penerima sinyal *GPS* mengkalkulasi posisinya dengan mengukur jarak dirinya dengan empat atau lebih satelit *GPS*. Dengan mengukur waktu tunda antara pengiriman dan penerimaan sinyal radio dari masing – masing *GPS* dan

mengetahui kecepatan sinyal maka didapat jarak ke masing-masing satelit tersebut. Sinyal tersebut juga berisi informasi mengenai posisi satelit. Dengan menentukan posisi dan jarak berdasarkan paling tidak empat satelit, penerima *GPS* dapat menghitung posisinya menggunakan mekanisme *trilateration* [5].

B. Location Based Service (LBS)

Layanan Berbasis Lokasi atau *Location Based Service* adalah layanan informasi yang dapat diakses melalui perangkat *mobile* dengan menggunakan *mobile network*, yang berkemampuan untuk memanfaatkan lokasi dari perangkat *mobile* tersebut. *Location Based Service* menyajikan kemungkinan komunikasi dan interaksi dua arah. Oleh sebab itu pengguna memberitahu penyedia layanan untuk memperoleh informasi yang dibutuhkan dengan referensi lokasi pengguna tersebut. Layanan berbasis lokasi dapat diilustrasikan sebagai sebuah layanan yang berada pada pertemuan tiga teknologi yaitu *Geographic Information System*, *Internet Service*, dan *Mobile Devices* [6].

C. Bahasa Pemrograman Java

Java merupakan Bahasa pemrograman yang dikembangkan oleh sebuah tim dibawah pimpinan *James Gosling* pada perusahaan *Sun Microsystems*. Awalnya pada tahun 1991, bahasa pemrograman ini dikenal dengan nama *Oak*. Berfokus pada penanam *chip* untuk para konsumen peralatan elektronik. Pada tahun 1995, berganti nama menjadi *java* yang mengalami proses desain ulang serta pengembangan untuk aplikasi internet. Setelah mengalami perkembangan yang signifikan, *java* menjadi bahasa pemrograman yang sangat populer di kalangan *programmer* [7].

Java memiliki fitur yang lengkap sehingga dapat digunakan untuk mengembangkan aplikasi di berbagai jenis *platform*. Kini *Java* tidak hanya digunakan untuk pemrograman *web*, melainkan juga untuk mengembangkan aplikasi di seluruh *platform* pada *server*, *desktop*, dan perangkat berbasis *mobile*, contohnya *Android*. *Java* menyediakan satu *set library* khusus yang memungkinkan pengembang untuk mengembangkan aplikasi *Android* menggunakan bahasa pemrograman *java* [8].

D. Android

Android adalah sistem operasi yang diperuntukkan bagi *mobile* dan berbasis *Linux* yang telah mengalami proses modifikasi. Awalnya sistem operasi ini dikembangkan oleh perusahaan *startup* yang bernama *Android, Inc.* Seiring dengan perkembangan teknologi, *Google* membeli *Android, Inc.* pada tahun 2005 dan mengambil alih pekerjaan serta tim pengembangan yang telah dibentuk oleh *Android, Inc.*

Selanjutnya *Google* menerapkan sistem sebagian besar kode *Android* berada dibawah Lisensi *Apache open source*, hal ini bertujuan untuk menjadikan sistem operasi *Android* terbuka dan gratis bagi semua pihak, sehingga siapapun yang ingin menggunakan dan mempelajari *Android* bisa

mengunduh *source code Android* secara lengkap. Kebijakan *open source Android* ini telah dimanfaatkan oleh beberapa perusahaan yang menciptakan *smartphone*, diantaranya *Samsung* dan *Xiaomi*. Kedua perusahaan ini merancang perangkat keras mereka sendiri dan memanfaatkan *Android* sebagai sistem operasinya. Ditinjau berdasarkan perancangan aplikasi, pihak pengembang hanya perlu mengembangkan aplikasi *Android* secara umum, dengan catatan perangkat yang digunakan telah ter *install* sistem operasi *Android* [9].

E. Firebase

Firebase adalah sebuah layanan *DbaaS (Database as a Service)* dengan konsep *realtime* yang didirikan oleh *Adrew Lee* dan *James Tamplin*. *Firebase* menyediakan layanan penyimpanan data dengan *backend* sebagai *service* yang berlokasi di California, Amerika Serikat. *Firebase* terdiri dari fitur pelengkap yang bisa diubah sesuai dengan kebutuhan. *Firebase* memberikan anda perlengkapan dan infrastruktur untuk membuat sebuah aplikasi yang lebih baik dan meningkatkan keberhasilan pada aplikasi yang akan dibuat [10].

Di sediakannya API atau fungsi pada Layanan *Firebase* yaitu database memungkinkan pembuat aplikasi menyimpan dan mensinkronisasi data lewat *multiple client*.

F. Usability Testing

Usability testing adalah suatu metode yang digunakan dalam sebuah penelitian. Adapun aspek pengujian yang dilakukan oleh seseorang peneliti cukup banyak, mulai dari hal percobaan dengan ukuran sampel yang besar dan metode pengujian desain secara komplek. *Usability testing* bersifat sebagai penelitian yang tergolong kualitatif, metode penelitian ini sama sekali tidak bersifat formal, dengan kata lain *usability testing* dapat dilaksanakan hanya dengan memiliki 2 orang penguji. Tujuan utama dari *usability testing* adalah untuk meningkatkan kualitas dari suatu produk yang telah dirancang oleh pengembang berdasarkan dari hasil pengujian yang didapatkan dari pihak penguji [11]. Metode yang digunakan ialah *SUS "System Usability Scale"*, yang dibuat oleh *John Brooke* pada tahun 1986, metodenya terdiri dari 10 pertanyaan dengan 5 pilihan respons dari sangat tidak setuju sampai sangat setuju. Metode *SUS* digunakan karena biaya dan waktu yang dikeluarkan untuk pengujian ini tidak terlalu besar, serta dapat menggunakan sampel yang tidak banyak, jumlah sampel bisa hanya 2 user [12].

Berdasarkan hasil kuesioner (Gambar 1) yang didapat dapat dihitung nilai *SUS*. Untuk melakukan perhitungan nilai *SUS* dengan cara memberikan bobot untuk setiap item akan berkisar dari 0 sampai 4. Perhitungan bobot untuk item mempunyai aturan sebagai berikut:

1. Untuk item no 1,3,5,7, dan 9 nilai yang didapat adalah posisi skala dikurangi 1.
2. Untuk item 2,4,6,8 dan 10, nilai yang didapat adalah 5 dikurangi posisi skala Kemudian jumlahkan nilai yang didapat dan lakukan proses perhitungan nilai jumlah dikalikan 2,5 untuk mendapatkan nilai keseluruhan skor *SUS* [12].

Nama : _____

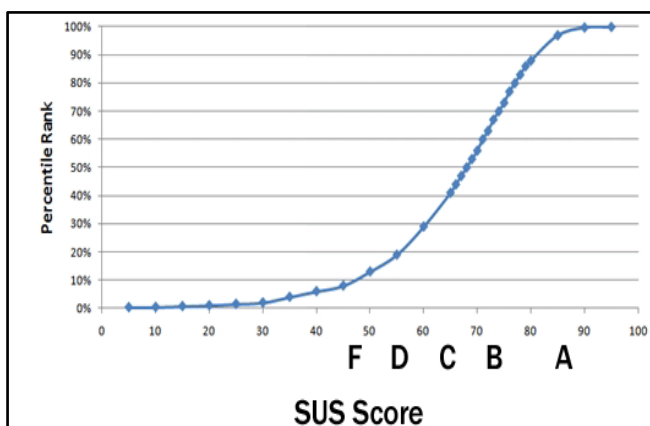
Anda Sebagai : Pemilik Kendaraan / Pemilik Tambal Ban *

Sangat Tidak Setuju **Sangat Setuju**

	1	2	3	4	5
1. Saya pikir bahwa saya akan ingin lebih sering menggunakan aplikasi ini.					
2. Saya menemukan bahwa aplikasi ini, tidak harus dibuat serumit ini.					
3. Saya pikir aplikasi mudah untuk digunakan.					
4. Saya pikir bahwa saya akan membutuhkan bantuan dari orang teknis untuk dapat menggunakan aplikasi ini.					
5. Saya menemukan berbagai fungsi di aplikasi ini diintegrasikan dengan baik.					
6. Saya pikir ada terlalu banyak ketidaksesuaian dalam aplikasi ini.					
7. Saya bayangkan bahwa kebanyakan orang akan mudah untuk mempelajari aplikasi ini dengan sangat cepat.					
8. Saya menemukan, aplikasi ini sangat rumit untuk digunakan.					
9. Saya merasa sangat percaya diri untuk menggunakan aplikasi ini.					
10. Saya perlu belajar banyak hal sebelum saya bisa memulai menggunakan aplikasi.					

*coret yang tidak perlu

Gambar 1 Kuesioner menggunakan metode SUS [12].



Gambar 2 Grafik rujukan [12].

TABEL I
SYSTEM USABILITY SCALE SCORE

Grade	SUS	Kelayakan	Kesesuaian
A	84,1 – 100	Sangat Layak	Sesuai
B	72,6 – 84,0	Layak	Sesuai
C	65,0 – 72,5	Cukup	Sesuai
D	55,0 – 64,9	Kurang	Kurang Sesuai
F	0 – 54,9	Tidak Layak	Tidak Sesuai

Untuk pemeringkatan nilai *SUS* dari jumlah nilai yang sudah didapatkan dapat dilihat pada Tabel 1 *System Usability Score*.

G. Unit Testing

Merupakan pengujian fungsi yang terdapat pada kode yang telah ditulis sesuai dengan spesifikasi. *Unit testing* terdiri dari sekumpulan kode yang ditulis kemudian

digunakan untuk menguji kode yang telah dibuat apakah sudah sesuai dengan spesifikasi yang ditentukan. Kode untuk pengujian dituliskan secara *manual* akan tetapi dijalankan secara otomatis [13].

Dengan metode pengujian yang sesuai dengan kebutuhan, juga dibutuhkan tools pendukung yang sesuai dan mudah digunakan untuk melakukan implementasi metode tersebut. Untuk menjawab permasalahan tersebut, *Expresso Testing Library* menjadi salah satu solusi untuk membantu proses pengujian. *Expresso Testing Library* adalah sebuah *Library* untuk melakukan pengujian *User Interface (UI)* aplikasi. *Expresso* sendiri sudah disediakan oleh *Android Testing Support Library* dan bisa digunakan pada perangkat dengan API Level 10 (Android 2.3.3) dan level di atasnya [14].

III. METODOLOGI PENELITIAN

A. Metodologi Penelitian

Metode penelitian yang dipakai yaitu (*Software Development Methodology*) atau metode pengembangan perangkat lunak yang merupakan panduan cara - cara dalam membuat aplikasi perangkat lunak. Cara yang digunakan pada proses pembuatan aplikasi ini adalah menggunakan metode *Extreme Programming (XP)* [15], ada beberapa bagian yang dilakukan terdiri dari perencanaan, desain aplikasi, pembuatan aplikasi, dan evaluasi aplikasi.

B. Perencanaan

Pada bagian perencanaan ini peneliti melakukan serangkaian kegiatan dan perumusan masalah dan pengumpulan data. Untuk tahapannya terdiri dari kepustakaan, dilanjutkan dengan analisis yang terdiri dari analisis kebutuhan sistem yang akan dibangun dan kebutuhan perangkat serta tahapan penelitian. Luaran dari tahapan ini adalah rancangan aplikasi *mobile* untuk sistem alur pencarian tambal ban berupa *Use case*, *Activity Diagram*, *Object Oriented Database Model*, *Deployment Diagram* dan *Database Firebase*. Indikator keberhasilan adalah menghasilkan rancangan aplikasi *mobile* untuk pencarian tambal ban terdekat.

C. Desain Aplikasi

Pada bagian desain aplikasi ini peneliti secara langsung melakukan proses pembuatan aplikasi menggunakan media visual berupa *mockup* yang berfungsi menampilkan hasil dari contoh aplikasi agar terlihat nyata.

D. Pembuatan Aplikasi

Proses pembuatan aplikasi pencarian tempat tambal terdekat menggunakan *software Android Studio Versi 3* dengan menggunakan bahasa pemrograman Java dalam proses kodenya. Pada tahap ini dilakukan juga pengujian aplikasi dengan menggunakan *unit test*, dengan cara pengetesan permodul tanpa menunggu perangkat lunak selesai dibuat dengan cara mencari *widget* yang akan diuji yang kemungkinan akan terjadinya kesalahan sewaktu memasukkan data (misalnya *TextView* atau *Button*).

Pengujian tersebut dilakukan bertujuan untuk menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi – fungsi, masukkan, dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pada pengujian aplikasi ini terdapat 3 *layout* yang diujikan, yaitu halaman login, halaman registrasi dan halaman *edit profil*. Pada pengujian unit test ini, penulis melakukan pengetesan dengan menggunakan *framework* pengujian *espresso* dengan menambahkan depedensi berikut dari android *testing support library* di *build.gradle*. Gambar 3 merupakan potongan kode *framework* pengujian *espresso*.

Adapun luaran dari tahapan ini adalah setiap fitur yang terdapat pada aplikasi dapat diakses dan berjalan sesuai dengan fungsinya masing-masing.

```
androidTestImplementation
'com.android.support.test:runner:1.0.1'
androidTestCompile('com.android.support.
test.espresso.espresso-core:3.0.1', {
exclude group: 'com.android.support',
module: 'support-annotations'})
```

Gambar 3 Kode *framework* pengujian *espresso*

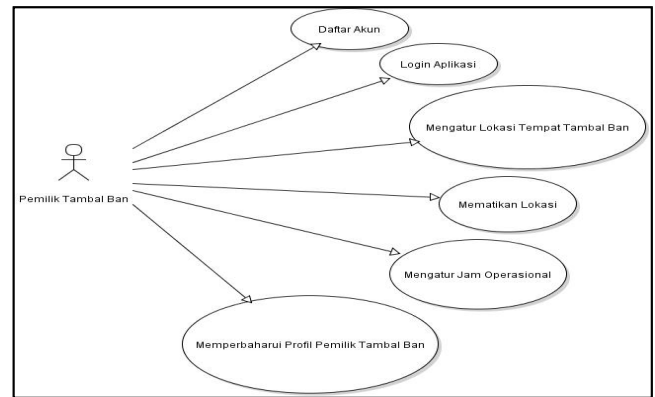
E. Evaluasi Aplikasi

Pada evaluasi aplikasi ini, sistem yang telah menjadi sebuah aplikasi dilakukannya evaluasi kelayakan aplikasi menggunakan *usability testing* dengan menggunakan metode *SUS* “*System Usability Scale*”. Adapun pihak penguji aplikasi antara lain adalah pemilik tambal ban, dan pengendara sepeda motor. Aplikasi akan diuji dengan cara diuji coba penggunaan aplikasi ini bagi pemilik tambal ban dan pengendara motor, setelah aplikasi tersebut diuji coba, kemudian diberikan pertanyaan kuesioner kepada pemilik tambal ban dan pengendara motor untuk penilaian kelayakan aplikasi yang sudah diuji tersebut. Tahapan pengujian *usability testing* yang dilakukan terhadap dua aktor ini bertujuan untuk mengetahui seberapa besar manfaat aplikasi pencarian tambal ban berbasis android dibandingkan dengan aplikasi yang sudah ada.

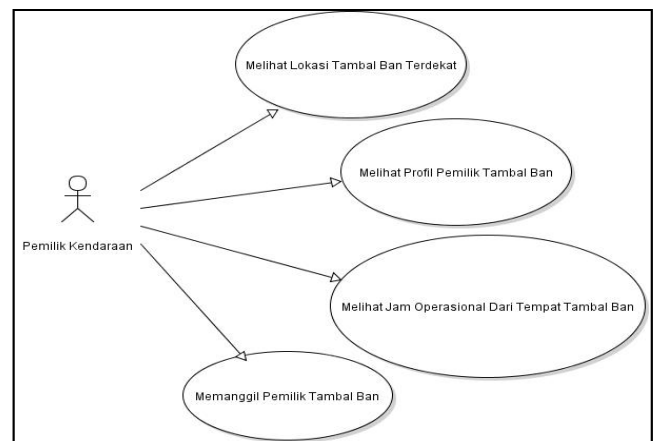
IV. HASIL DAN PEMBAHASAN

A. Use Case

Use case pada aplikasi ini menggambarkan interaksi antara user dengan aplikasi pencarian tambal ban tersebut. Pada use case untuk aplikasi pemilik tambal ban pada Gambar 4 tersebut, pemilik diharuskan melakukan daftar akun terlebih dahulu, kemudian login, setelah itu dapat mengatur lokasi tambal ban, mengatur jam operasional, memperbaharui profil akun dari pemilik tambal ban, serta dapat mematikan lokasi tambal ban tanpa harus menunggu jam operasional dari tempat tambal ban tersebut tutup.



Gambar 4 Use case aplikasi untuk pemilik tambal ban



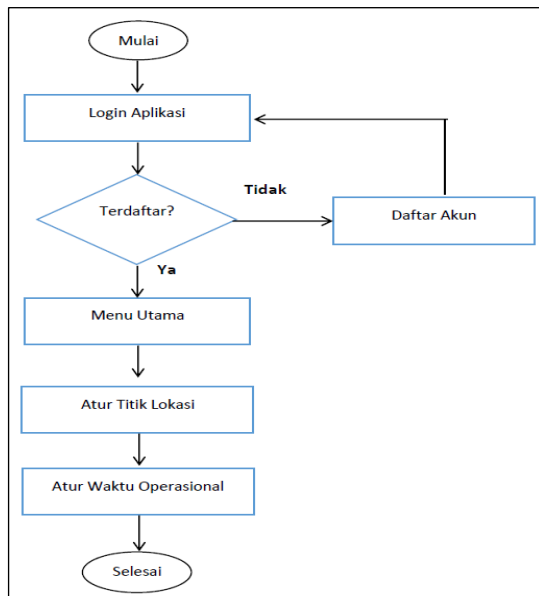
Gambar 5 Use case aplikasi untuk pemilik kendaraan

Pada *use case* pemilik kendaraan pada Gambar 5, pemilik dapat melihat lokasi terdekat dari tempat tambal ban tersebut, melihat profil dari pemilik tempat tambal ban, melihat jam operasional dari tempat tambal ban, serta dapat memanggil pemilik tambal ban tersebut.

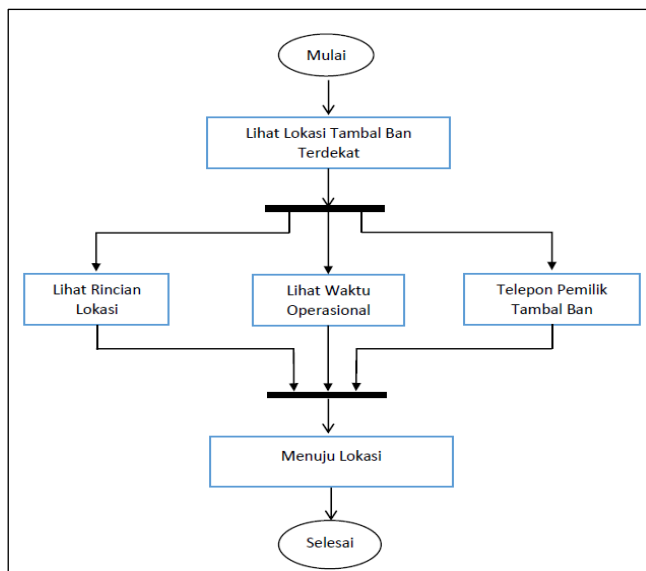
B. Diagram Aktivitas

Diagram aktivitas pada aplikasi ini menggambarkan kegiatan yang dilakukan antara *user* dengan aplikasi pencarian tambal ban untuk pemilik tambal ban dan juga pemilik kendaraan.

Pada Gambar 6 pemilik tambal ban pada daftar aktivitas ini diharuskan *login* terlebih dahulu, jika belum mempunyai akun dapat menekan menu belum daftar dan mengisi biodata yang ada pada halaman tersebut. Setelah masuk pada halaman menu utama, pemilik dapat mengatur titik lokasi tempat tambal ban serta mengatur waktu operasional dari tempat tambal ban. Dan pemilik juga dapat memperbaharui profil dari akun tambal ban tersebut.



Gambar 6 Diagram aktivitas untuk aplikasi pemilik tambal ban



Gambar 7 Diagram aktivitas untuk aplikasi pemilik kendaraan

Pada Gambar 7 aktivitas yang dapat dilakukan oleh pemilik kendaraan di antaranya dapat melihat rincian lokasi dari pemilik tambal ban, pada rincian tersebut terdapat informasi dari jadwal operasional dari tempat tambal ban serta adanya fitur panggilan yang memudahkan kita dalam menghubungi pemilik tambal ban tersebut.

C. Pembuatan Aplikasi

Dalam proses pembuatan aplikasi pencarian tambal ban terdekat, peneliti membuat dua aplikasi yang ditunjukkan untuk pemilik kendaraan dan untuk pemilik tempat tambal ban. Kegunaan dari dibuatkannya aplikasi pencarian tambal ban ini menjadi dua aplikasi yaitu aplikasi untuk pemilik tambal ban dan aplikasi untuk pengendara adalah untuk

membuat aplikasi ini lebih terstruktur dalam proses pengkodean dan proses pengembangannya nanti.

1) *Aplikasi untuk Pemilik Tambal Ban*: Pada Gambar 8 ini pemilik tambal ban yang belum memiliki akun diharuskan mendaftar terlebih dahulu dengan menekan text “Belum Daftar?” dibawah tombol “MASUK” pada halaman *Login Pemilik Tambal Ban*, dibawah ini merupakan bentuk halaman login bagi pemilik tempat tambal ban. Untuk bisa masuk ke halaman utama, pemilik tambal ban diharuskan memasukkan email dan password yang sudah terdaftar. Jika pemilik tambal ban belum mendaftar, maka bisa mengklik text “Belum Daftar?”. Dibawah ini merupakan kode dari proses login bagi pemilik tambal ban. Pada halaman ini pemilik tambal ban yang belum memiliki akun akan diarahkan ke halaman registrasi dengan cara mengklik tombol “Belum Daftar?”, Gambar 9 ini merupakan halaman dari proses pengimputan data pemilik tambal ban tersebut. Pada Gambar 8 pengguna diminta memasukkan data seperti yang sudah tersedia, pada halaman ini *user* dapat menambahkan foto profil dan juga dapat menambahkan foto dari tempat tambal ban tersebut, supaya memudahkan para pemilik kendaraan mengetahui secara pasti informasi tempat tambal ban tersebut.

Gambar 8 Halaman Login

Gambar 9 Halaman Daftar

Setelah berhasil menambahkan data maka akan keluar pesan dalam bentuk toast “Berhasil Menambah Data”. Dan jika proses memasukkan data terjadi error maka akan keluar pesan berupa toast “Gagal Menambah Data”. Data dari halaman tersebut kemudian dikirim ke firebase database dengan menggunakan method `ref = rootRef.child("Profil").push();`, kemudian pemilik tambahan diarahkan ke halaman menu utama.

Pada Gambar 10 halaman menu utama ini pemilik tambal ban dapat mengeset titik lokasi dari tempat tambal ban tersebut lalu mengeset waktu operasional dari tempat tambal ban tersebut, yang terdiri dari tombol set buka, dan tombol untuk set waktu tutup.

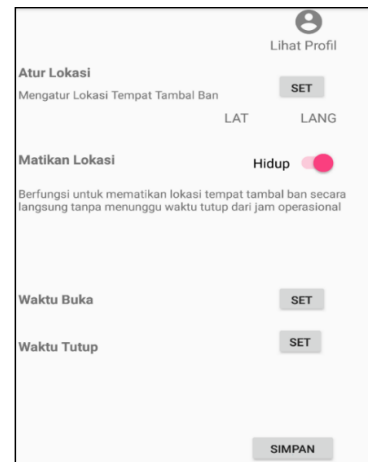
Pada Gambar 10, mengeset waktu buka dan tutup diperlukan untuk menghidupkan titik lokasi dan menghilangkan secara otomatis titik lokasi jika waktu yang sudah di *set* sudah dalam keadaan waktu tutup. Fungsi lainnya yaitu sebagai bentuk informasi jam operasional yang akan ditampilkan di aplikasi untuk pengendara, serta adanya tombol untuk menghidup matikan lokasi secara otomatis, yang berfungsi bagi pemilik tambal ban yang ingin mematikan titik lokasi tempat tambal ban secara langsung dan cepat tanpa menunggu waktu tutup yang sudah diset pada aplikasi yang masih aktif.

2) *Aplikasi untuk Pemilik Kendaraan:* Gambar 11 halaman ini pemilik kendaraan akan diarahkan langsung oleh sistem pada titik tempat tambal ban terdekat. Pemilihan tempat tambal ban terdekat dilakukan dengan cara menghitung jarak pemilik kendaraan dengan jarak tambal ban yang masih aktif. Selanjutnya semua jarak tambal ban yang masih aktif dibandingkan dan tambal ban dengan jarak terkecil dipilih sebagai tambal ban terdekat. Pada Gambar 11 terdapat tombol pada bagian paling bawah pada halaman aplikasi untuk pemilik kendaraan yang berguna untuk memudahkan pengendara yang ingin menuju lokasi agar dapat terhubung dengan aplikasi *google map*.

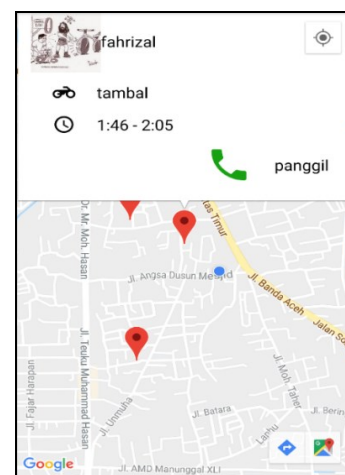
D. Pengujian Unit Testing

Setelah itu dilakukan pengujian unit testing dengan menggunakan testing tools espresso. Unit test dilakukan pada beberapa halaman diantaranya untuk halaman login ini, yang bertujuan untuk menguji kesesuaian kode yang telah dibangun dengan rancangan spesifikasi. Dengan jenis anotasi yang diterapkan dalam ke metode pengujian *class EmailValidatorTest* adalah *@Test*, yang menunjukkan bahwa metode anotasi ini dilampirkan untuk dijalankan sebagai kasus pengujian.

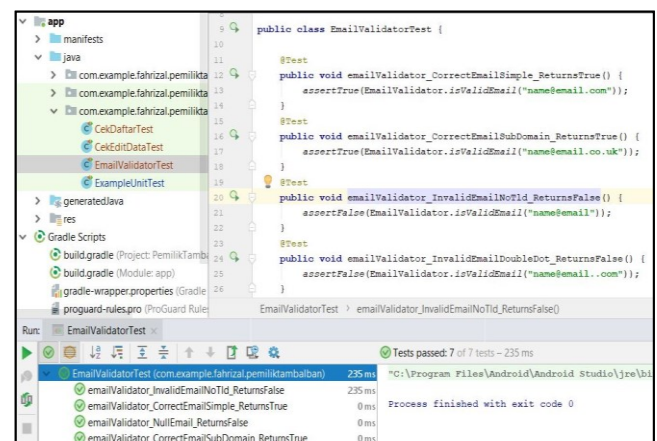
Pada Gambar 12 memperlihatkan tidak adanya error yang terdapat pada pengujian unit test ini, sehingga kode yang telah dibangun ini sesuai dengan spesifikasi yang dirancang sebelumnya. Pengujian ini berisi rangkaian *assertion statement* (kalimat panuntunan) seperti *assertTrue* dan juga *assertFalse* yang berfungsi sebagai cek poin dalam validasi perilaku kelas yang akan diuji.



Gambar 10 Halaman Menu Utama



Gambar 11 Halaman menu utama Aplikasi Untuk pemilik kendaraan



Gambar 12 Unit Test Halaman Login

E. Pengujian Usability Testing

Usability testing yang digunakan ini menggunakan metode *System Usability Score (SUS)* yang bertujuan untuk

mengukur usability dari aplikasi pencarian tambal ban terdekat ini. Evaluasi ini dilakukan pada masing – masing aplikasi, yaitu 5 orang aplikasi pencarian tambal ban bagi pemilik tambal ban dan 5 orang pencarian tambal ban terdekat bagi pemilik kendaraan.

1) *Evaluasi untuk Pemilik Tambal Ban*: Berikut ini data evaluasi untuk pemilik tambal ban, yang dilakukan pada 5 responden. Data tersebut diambil pada saat proses evaluasi aplikasi ditempat tambal ban dan dari data tersebut dapat diketahui seberapa lama pemilik tambal ban dapat menggunakan aplikasi pencarian tempat tambal ban tersebut. Pada Tabel 1 terdapat hasil dari kumpulan waktu aktivitas dari pemilik tambal ban. Dari Data waktu tersebut dapat dilihat mudah atau tidak-nya aplikasi pencarian tempat tambal ban tersebut untuk digunakan. Dari data pada tabel diatas, waktu yang diperlukan paling lama adalah pada saat aktifitas pendaftaran akun dengan waktu rata – rata 122.8 detik. Hal ini dikarenakan kurangnya intensitas pengguna dalam menggunakan aplikasi yang mengharuskan proses pendaftaran akun. Setelah dilakukan pengumpulan data evaluasi untuk pemilik tambal ban, kemudian data tersebut dilakukan perhitungan sesuai dengan aturan metode SUS untuk menghasilkan skor SUS. Nilai akhir *SUS* yang dihasilkan dari pertanyaan kuesioner kepada 5 orang responden adalah 77.5. Hasil tersebut diambil dari total *SUS score* dari setiap responden lalu dikalikan 2,5 kemudian hasil dari total *SUS* dijumlahkan dan dibagi dengan banyaknya responden yang diuji. Hasil dari nilai akhir *SUS* menunjukkan bahwa aplikasi pencarian tambal ban terdekat bagi pemilik tambal ban memiliki tingkat usability diatas angka 70. Berdasarkan nilai total *SUS* sebesar 77.5, yang berarti bernilai B jika dikonversikan kedalam *SUS Score* seperti pada Tabel 1 *System Usability Score* [12] untuk kelayakan aplikasi, maka aplikasi tersebut sudah masuk dalam kategori “Layak”.

2) *Evaluasi untuk Pemilik Kendaraan*: Berikut ini data evaluasi untuk pemilik kendaraan, yang dilakukan pada 5 responden. Data tersebut diambil pada saat proses evaluasi aplikasi dan dari data tersebut dapat diketahui seberapa lama pemilik kendaraan dapat menggunakan aplikasi pencarian tempat tambal ban tersebut. Pada Tabel 2 terdapat hasil dari kumpulan waktu aktivitas dari pemilik kendaraan. Dari Data waktu tersebut dapat dilihat mudah atau tidak-nya aplikasi pencarian tempat tambal ban tersebut untuk digunakan. Dari data pada tabel di atas, tidak perlu waktu yang lama untuk bisa menemukan lokasi tempat tambal ban terdekat, pada saat proses pemanggilan pemilik tambal ban tersebut yaitu rata – rata 3.4 detik. Untuk proses melihat lokasi tempat tambal ban rata – rata waktu yang diperlukan 3.4 detik. Setelah dilakukan pengumpulan data evaluasi untuk pemilik tambal ban, kemudian data tersebut dilakukan perhitungan sesuai dengan aturan metode *SUS* untuk menghasilkan skor *SUS*. Nilai akhir *SUS* yang dihasilkan dari pertanyaan kuesioner kepada 5 orang responden adalah 91.5. Hasil tersebut diambil dari total *SUS score* dari setiap responden lalu dikalikan 2,5 kemudian hasil dari total *SUS* dijumlahkan dan dibagi dengan banyaknya responden yang diuji. Hasil dari nilai akhir *SUS*

menunjukkan bahwa aplikasi pencarian tambal ban terdekat bagi pemilik kendaraan memiliki tingkat usability diatas rata-rata karena berada di atas angka 90. Berdasarkan nilai total *SUS* sebesar 91.5, yang berarti bernilai A jika dikonversikan kedalam *SUS Score* seperti pada Tabel 1 *System Usability Score* [12] untuk kelayakan aplikasi, maka aplikasi tersebut sudah masuk dalam kategori “Sangat Layak”.

TABLE I
KUMPULAN WAKTU AKTIFITAS PADA APLIKASI UNTUK PEMILIK TAMBAL BAN

No	Aktifitas	Waktu Yang Diharapkan (Detik)	Rata – Rata Waktu Pengujian Dari 5 Orang Pemilik Tambal Ban	Kesimpulan
1	Mendaftar Akun	90	122.8	Berhasil
2	Masuk Aplikasi (Login)	20	20.2	Berhasil
3	Mengatur Lokasi Tempat Tambal Ban	20	13	Berhasil
4	Mengatur Jam Operasional	15	15.6	Berhasil
5	Mematikan Lokasi Tempat Tambal Ban	5	4.8	Berhasil
6	Memperbaharui Profil Pemilik Tambal Ban	60	52.8	Berhasil

TABEL II
KUMPULAN WAKTU AKTIFITAS PADA APLIKASI UNTUK PEMILIK KENDARAAN

No.	Aktifitas	Waktu Yang Diharapkan	Rata – Rata Waktu Pengujian Dari 5 Orang Pemilik Kendaraan	Kesimpulan
1	Melihat Lokasi Tambal Ban Terdekat	10 Detik	8.4 Detik	Berhasil
2	Memanggil Pemilik Tambal Ban	5 Detik	3.4 Detik	Berhasil

V. KESIMPULAN

Aplikasi untuk Sistem Pemandu Pencarian Tambal Ban Terdekat Menggunakan Location-Based Services Pada Platform Android telah berhasil dibangun dengan memakai bahasa pemrograman *Java* dalam proses pembuatan aplikasi dan menggunakan *firebase* sebagai *database manajemen*.

Hasil pengujian aktivitas aplikasi untuk pemilik tambal ban terdapat kendala dalam proses pendaftaran akun yang dikarenakan kurang pengalaman-nya para pemilik tambal ban dalam proses pendaftaran akun tersebut. Untuk pengujian aplikasi bagi pemilik kendaraan menunjukkan tidak ada-nya kendala dalam proses pemakaian-nya.

Hasil *usability testing* dianalisa dengan menggunakan metode SUS menunjukkan skor SUS sebesar 77.5 untuk aplikasi pemilik tambal ban yang dikonversikan bernilai B. atau masuk dalam kategori “Layak”. Selain itu untuk aplikasi pemilik kendaraan bernilai 91.5 yang dikonversikan bernilai A, atau masuk dalam kategori “Sangat Layak” untuk digunakan.

REFERENSI

- [1] R. Indah, "kominfo.go.id," kominfo, 2 10 2015. [Online]. Available: https://www.kominfo.go.id/content/detail/6095/indonesia-raksasa-teknologi-digital-asia/0/sorotan_media. [Accessed 7 2 2018].
- [2] S. Irawan and G. H. Putra, "TEKNOMATIKA," *Aplikasi Pencarian Tambal Ban Motor Terdekat Berbasis Android (Studi Kasus Tambal Ban Di Kota Batam)*, vol. 09, p. 16, 2016.
- [3] A. A. Slameto, "DASI," *Penerapan Location Based Service Untuk Pembuatan Aplikasi Pencarian Tambal Ban Berbasis Android*, vol. 16, p. 02, 2015.
- [4] B. Yulianto and R. Layona, "Aplikasi Pencarian Tempat Wisata Berbasis GPS dengan Metode Radius dan Rating" Comtech, vol. 6, no. 1, pp. 109-120, 2015.
- [5] Sunomo. 2004. Pengantar Sistem Komunikasi Nirkabel. Jakarta :Grasindo.
- [6] R. R. Siahaan, K. I. Satoto, and K. T. Martono, "Implementasi Sistem Informasi Geografis Daerah Pariwisata Kota Semarang Berbasis Android dengan Global Positioning System (GPS)," *Jurnal Teknologi dan Sistem Komputer*, vol. 2, no. 1, pp. 96-109, Jan. 2014.
- [7] Imaniar, Juwita. 2013. Jurnal Tugas Akhir "Aplikasi Location Based Service untuk Sistem Informasi Publikasi Acara pada Platform Android".
- [8] Y. D. Liang, *Introduction To Java Programming Brief Version*, 8th Ed. New Jersey, NJ: Prentice Hall, 2011.
- [9] G. Company, "Android Developer Fundamentals Course," Google, 2016. [Online]. Available: <https://developers.google.com/training/courses/android-fundamentals>.
- [10] M. W. Desta, *Implementasi Teknologi Firebase Pada Aplikasi Pencarian Lokasi Service Kamera Berdasarkan Rating Berbasis Android*, Yogyakarta: Stmik Akakom, 2017.
- [11] Jeff Sauro, "Measuring Usability With The System Usability Scale (SUS)", <http://www.measuringu.com/sus.php>, 2011.
- [12] John Brooke, "SUS - A quick and dirty usability scale", Redhatch Consulting Ltd., 12 Beaconsfield Way, Earley, Reading RG6 2UX, United Kingdom.
- [13] P. Runeson, "A survey of unit testing practices," *IEEE software*, vol. 23, no. 4, pp. 22-29, (2006).
- [14] developer. (2018). Retrieved from android.com: <https://developer.android.com/training/testing/ui-testing/espresso-testing.html>
- [15] Roger, S. Pressman, Ph.D., 2012, *Rekayasa Perangkat Lunak (Pendekatan Praktisi) Edisi 7: Buku 1*, Yogyakarta: Andi.