

## APLIKASI ANDROID UNTUK REKOMENDASI TEMPAT FAVORIT BERDASARKAN PENJEJAKAN OBJEK

Annisa Rahmawati<sup>\*</sup>), Aghus Sofwan dan Yosua Alvin Adi Soetrisno

Departemen Teknik Elektro, Fakultas Teknik, Universitas Diponegoro  
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>\*</sup>E-mail: annisarahma06@students.undip.ac.id

### Abstrak

Banyak aplikasi *smartphone* yang memerlukan peranan GPS dalam menjalankan fungsinya seperti mengumpulkan data dengan cara melacak posisi pengguna untuk mencari tahu mengenai tempat-tempat yang paling banyak dikunjungi oleh pengguna aplikasi atau mencari lokasi pengguna untuk memberikan informasi mengenai tempat-tempat yang ada di sekitar pengguna. Tidak hanya lokasi tempat-tempat di sekitar pengguna, informasi mengenai berapa kali tempat tersebut dikunjungi juga disediakan. Informasi tersebut dapat membantu pengguna dalam menentukan tempat mana yang menarik untuk dikunjungi.

*Kata kunci:* Android, GPS (*Global Positioning System*), Java

### Abstract

*Many smartphone applications require the role of GPS in carrying out their functions such as collecting data by tracking the user's position to find out about the places most visited by application users or searching for the user's location to provide information about the places around the user. Not only the location of the places around the user, the information about how many times the place was visited is also provided. This information can help users determine which places are interesting to visit.*

*Keywords:* Android, GPS (*Global Positioning System*), Java

## 1. Pendahuluan

Aplikasi yang terdapat pada *smartphone* atau yang juga bisa disebut sebagai aplikasi *mobile* pada awal perkembangannya diciptakan dengan fungsionalitas terbatas[1]. Penggunaan aplikasi secara multitasking pada awalnya dihindari karena kemampuan *hardware smartphone* yang terbatas. Perkembangan aplikasi pada *smartphone* Android mengakibatkan perubahan pada cara berinteraksi antar manusia. Tidak hanya melalui suara, kini manusia juga dapat berinteraksi jarak jauh dengan menggunakan video dan suara yang ditransmisikan secara langsung dan biasa disebut dengan *video call* sehingga semua pihak yang melakukan *video call* ini dapat berinteraksi seakan-akan berada di tempat yang sama tanpa harus bertemu secara langsung. Teknologi pada *smartphone* yang semakin beragam membuat para pengembang aplikasi untuk membuat aplikasi-aplikasi baru dengan memanfaatkan teknologi terkini yang ada pada *smartphone*[2]. Berbagai aplikasi Android dikembangkan untuk mempermudah manusia dalam melakukan kegiatan sehari-hari seperti melakukan jual beli secara daring, berinteraksi dengan teman dan keluarga, sebagai sarana hiburan, dan kegiatan-kegiatan lainnya.

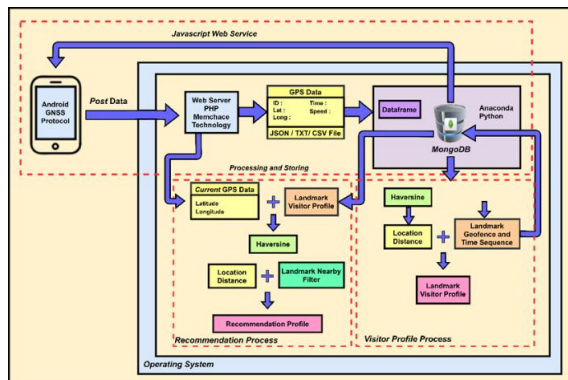
Salah satu aplikasi *mobile* yang sering digunakan yaitu *maps*[3]. *Maps* membantu manusia untuk menemukan letak sebuah tempat berdasarkan alamat atau data *latitude* dan *longitude* dari tempat tersebut[4]. *Maps* juga dapat membantu untuk menemukan rekomendasi berbagai tempat untuk dikunjungi. Terdapat 2 fitur penting pada *smartphone* yang diperlukan untuk menggunakan *maps*, yaitu GPS (*Global Positioning System*) dan internet[5][6]. GPS dan internet diperlukan untuk mengakses lokasi pengguna atau menemukan lokasi yang dicari oleh pengguna[7][8].

Penelitian yang telah dilakukan yaitu dengan merancang suatu aplikasi Android untuk mendapatkan data *latitude*, *longitude*, kecepatan, nama, nomor ID (Identity), dan waktu dari *smartphone* pengguna. Data-data tersebut kemudian dianalisa sehingga mendapatkan tempat-tempat yang dapat dijadikan rekomendasi bagi pengguna aplikasi yang telah dirancang. Hasil analisa yang didapat akan ditampilkan di aplikasi sebagai rekomendasi bagi pengguna. Keluaran dari penelitian ini adalah aplikasi Android untuk mengakses data pengguna secara *realtime* serta hasil dari analisa yang telah dilakukan.

## 2. Metode

### 2.1. Deskripsi Sistem

Konsep dari sistem yang dibuat yaitu bagaimana aplikasi dapat mengirimkan data *latitude*, *longitude*, kecepatan, nama, nomor ID, serta waktu dari perangkat pengguna ke *server* untuk dianalisa dan menampilkan hasil analisa tersebut ke perangkat pengguna sehingga dapat dilihat dan dimanfaatkan oleh pengguna. Alur dari sistem yang dibuat dapat dilihat pada gambar 1.



Gambar 1. Desain Sistem

Pada gambar 1 terlihat bahwa ketika pengguna menggunakan dan mengaktifkan aplikasi dari penulis, maka *smartphone* pengguna akan mencari lokasi terkini dari pengguna melalui fitur GPS yang terdapat pada *smartphone* pengguna. Aplikasi tersebut kemudian mengirimkan data *latitude*, *longitude*, nama, nomor ID, kecepatan, serta waktu dari *smartphone* pengguna ke *server* melalui jaringan internet. Data-data mengenai *latitude*, *longitude*, nama, nomor ID, kecepatan, serta waktu dari berbagai pengguna kemudian akan diakses dan dianalisa untuk mengetahui tempat-tempat yang banyak dan sering dikunjungi oleh para pengguna aplikasi. Setelah data-data tersebut dianalisa dan telah mendapatkan informasi mengenai nama tempat, *latitude* dan *longitude* tempat, alamat, serta jumlah kunjungan, maka informasi tersebut akan disimpan pada *database* MongoDB[9][10]. Informasi tersebut akan divisualisasikan pada peta dan ditampilkan melalui *webview* sehingga pengguna aplikasi dapat melihat lokasi terkini dan tempat-tempat rekomendasi di sekitarnya.

## 2.2. Analisa Kebutuhan

### 2.2.1. Kebutuhan Fungsional

Berikut merupakan kebutuhan fungsional yang terdapat pada sistem:

- 1) Aplikasi dapat mengirimkan data mengenai *latitude*, *longitude* yang telah didapatkan melalui fitur GPS[11][12] pada *smartphone* pengguna, waktu, serta nama dan nomor ID yang telah diisikan oleh pengguna.

- 2) Durasi pengiriman data ke *server* dapat diatur sesuai keinginan pengguna.
- 3) *Server* dapat menangkap data mengenai *latitude*, *longitude*, nama, nomor ID, waktu, dan kecepatan yang telah dikirimkan oleh *smartphone* pengguna.
- 4) *Web* penampil rekomendasi dapat menampilkan lokasi terkini pengguna serta mengambil data dari basis data MongoDB kemudian menampilkan rekomendasi tempat yang berada di sekitar pengguna berdasarkan data tersebut.
- 5) Aplikasi dapat menampilkan *webview* yang berisi informasi mengenai lokasi terkini pengguna dan rekomendasi tempat di sekitar pengguna.

### 2.2.2. Kebutuhan non Fungsional

Berikut merupakan kebutuhan non fungsional pada aplikasi Android untuk rekomendasi tempat favorit berdasarkan penjejak objek:

- 1) Aplikasi hanya dapat digunakan pada sistem operasi Android.
- 2) Agar dapat menggunakan aplikasi, perangkat pengguna harus menggunakan Android versi 6.0 (API level 23) ke atas.
- 3) Aplikasi memiliki tampilan yang sederhana sehingga mudah digunakan oleh pengguna.
- 4) Aplikasi memerlukan internet untuk melakukan pertukaran data dengan *server*.

### 2.2.3. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras dalam melakukan pengembangan aplikasi Android untuk rekomendasi tempat favorit berdasarkan penjejak objek ditunjukkan pada tabel 1

Tabel 1. Kebutuhan perangkat keras

Spesifikasi	Keterangan
Processor	Intel(R) Core(TM) i5-5200U
RAM	4096 Megabyte
VGA	AMD Radeon R5 M330
Hard drive	500 Gigabyte

Sedangkan untuk spesifikasi kebutuhan pada *server* dalam melakukan pengembangan aplikasi Android untuk rekomendasi tempat favorit berdasarkan penjejak objek, ditunjukkan pada tabel 2.

Tabel 2. Kebutuhan server

Spesifikasi	Keterangan
Jumlah CPU	2
Memori Penyimpanan	30 Gigabyte
Memori Virtual	1024 Megabyte
Bandwidth	Unmetered

### 2.2.4. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dalam melakukan pengembangan aplikasi Android untuk rekomendasi

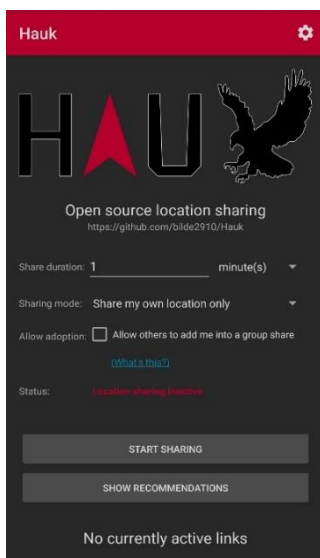
tempat favorit berdasarkan penjejukan objek ditunjukkan pada tabel 3.

Tabel 3. Kebutuhan server

Spesifikasi	Keterangan
Sistem Operasi	Windows 10 64-bit, Build 18362
IDE Android	Android Studio 4.0
Database Tools	MongoDBCompass
Text Editor	VSCode 1.51.1

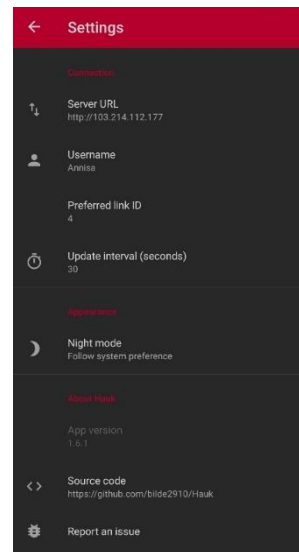
### 2.3. Desain Aplikasi

Desain Aplikasi dilakukan untuk membuat tampilan atau layout yang nantinya akan ditampilkan pada aplikasi sehingga pengguna dapat menggunakan fungsi-fungsi yang terdapat pada aplikasi.



Gambar 2. Tampilan MainActivity

Dapat dilihat pada gambar 2 merupakan tampilan pada MainActivity. MainActivity merupakan activity yang pertama kali muncul ketika pengguna membuka aplikasi. SettingsActivity merupakan activity dimana pengguna dapat mengatur server URL, username, preferred link ID, dan update interval. Tampilan SettingsActivity diatur pada file root\_preferences.xml. Tampilan SettingsActivity ditunjukkan pada gambar 3.



Gambar 3. Tampilan SettingsActivity

Dapat dilihat pada gambar 3 merupakan tampilan pada SettingsActivity. SettingsActivity merupakan activity dimana pengguna dapat mengatur server URL, username, preferred link ID, dan update interval.

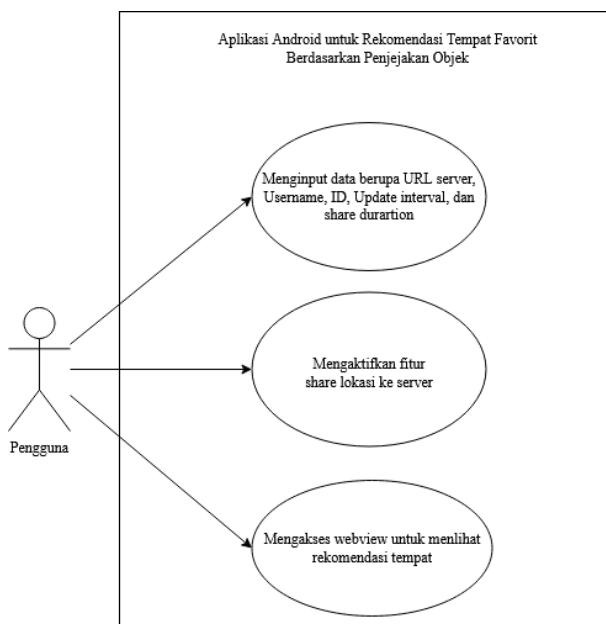


Gambar 4. Tampilan WebviewActivity

Dapat dilihat pada gambar 2 merupakan tampilan pada *WebViewActivity*. Tampilan digunakan untuk memuat tampilan web dari *localhost*[13][14][15].

#### 2.4. Diagram Use Case Aplikasi

Diagram *use case* merupakan diagram yang digunakan untuk menggambarkan segala fungsi yang terdapat pada sistem yang dibuat pengembang. Diagram ini menunjukkan berbagai interaksi yang dapat dilakukan oleh pengguna terhadap aplikasi. Pada perancangan aplikasi Android untuk rekomendasi tempat favorit berdasarkan penjejakan objek, diagram *use case* bermanfaat untuk menspesifikasikan fungsi pada aplikasi serta isi yang terdapat pada aplikasi.



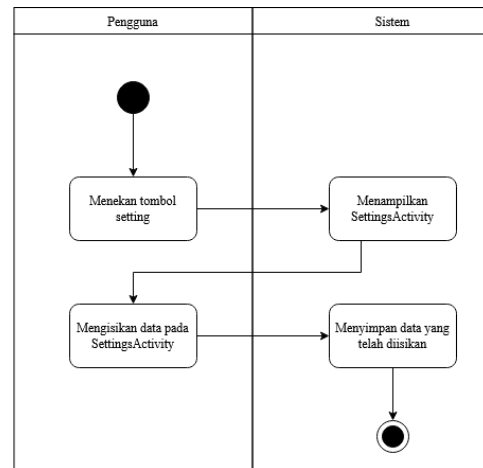
Gambar 5. Diagram use case

#### 2.5. Diagram Aktivitas

Diagram aktivitas digunakan untuk mengilustrasikan alur dari sebuah sistem dan menampilkan berbagai jalur keputusan yang ada ketika *activity* sedang dijalankan. Diagram aktivitas di aplikasi Android untuk rekomendasi tempat favorit berdasarkan penjejakan objek dibagi menjadi Diagram *SettingsActivity*, Diagram *MainActivity*, dan Diagram *WebViewActivity*.

##### 2.5.1. Diagram Aktivitas SettingsActivity

Diagram aktivitas *SettingsActivity* menggambarkan aktivitas pengguna mengatur atau mengisi nilai pada bagian *setting* aplikasi. Pengguna dapat masuk ke *settings* dengan menekan tombol *setting* yang berada di bagian kanan atas tampilan utama aplikasi. Diagram aktivitas *SettingsActivity* ditunjukkan pada gambar 6.

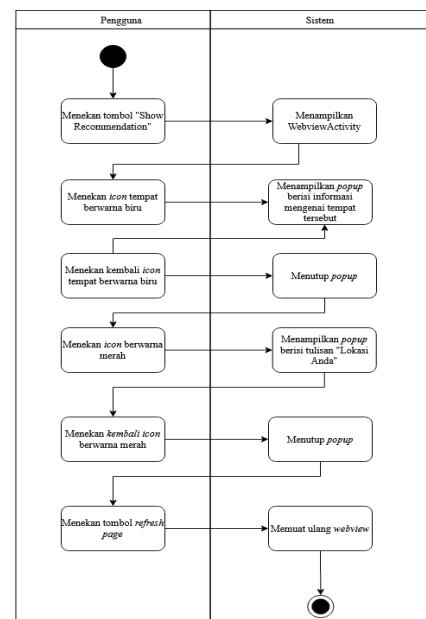


Gambar 6. Diagram aktivitas SettingsActivity

Setelah pengguna menekan tombol *setting*, maka akan muncul tampilan *SettingsActivity* yang berisikan kolom *server URL*, *username*, *preferred link id*, dan *update interval*. Pengguna perlu mengisi semua kolom tersebut. Pada poyek tugas akhir ini, bagian *server URL* diisikan <http://103.214.112.177>. Untuk *username* diisikan nama pengguna, pada *preferred link ID* diisikan ID setiap pengguna yang telah ditentukan oleh peneliti.

##### 2.5.2. Diagram Aktivitas WebViewActivity

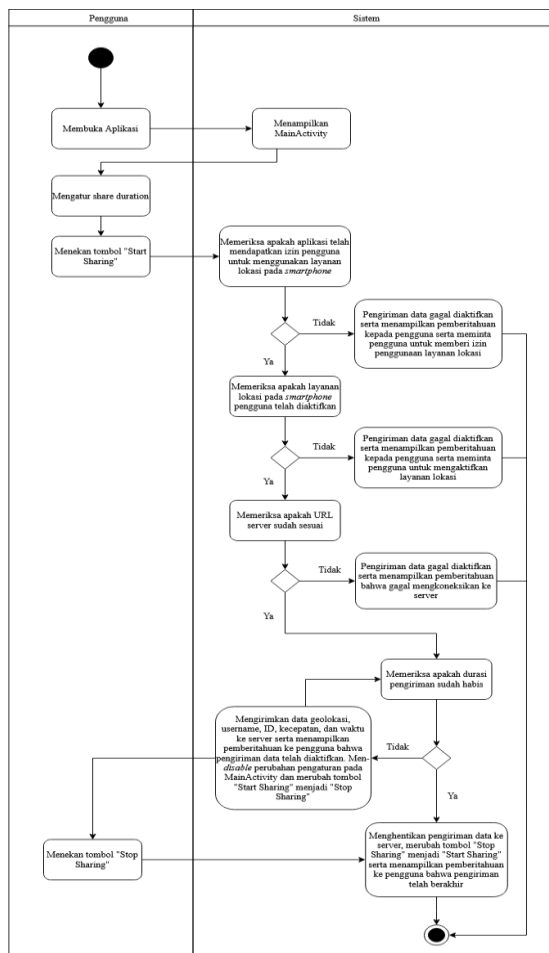
Diagram aktivitas *WebViewActivity* menggambarkan aktivitas pengguna ketika mengakses *WebViewActivity.java*. *WebViewActivity.java* digunakan pengguna untuk mengakses rekomendasi tempat. Diagram aktivitas *WebViewActivity* ditunjukkan pada gambar 7.



Gambar 7. Diagram Aktivitas WebViewActivity

2.5.3. Diagram Aktivitas *MainActivity*

Diagram aktivitas *MainActivity* menggambarkan aktivitas pengguna pada tampilan utama aplikasi. *MainActivity* merupakan *activity* Ketika pengguna membuka aplikasi, tampilan inilah yang pertama kali dilihat oleh pengguna. Tampilan ini juga memuat beberapa tombol yang dapat mengantarkan pengguna ke *activity* lain. Pada *MainActivity*, juga menggunakan fitur-fitur utama pada aplikasi yaitu mengirim data geolokasi, *username*, ID, kecepatan, dan waktu pengguna ke *server*. Untuk mengakses *SettingsActivity* dan *WebviewActivity* pengguna dapat melakukannya hanya melalui *MainActivity*. *MainActivity* ditunjukkan pada gambar 8 berikut.



Gambar 8. Diagram aktivitas *MainActivity*

3. Hasil dan Analisis

Pada penelitian tugas akhir ini, dilakukan dua macam pengujian. Untuk pengujian fungsi pada aplikasi dilakukan dengan menggunakan metode *Black Box*. Sedangkan untuk pengujian yang kedua, aplikasi akan diuji dengan menggunakan metode *White Box*.

Perangkat-perangkat Android yang digunakan untuk melakukan pengujian diantaranya:

- 1) Samsung Galaxy A50, OS Android versi 10, Chipset Exynos 9610 (10nm), CPU Octa-core (4x2.3 GHz Cortex-A73 & 4x1.7 GHz Cortex-A53).
- 2) Xiaomi Redmi Note 8, OS Android versi 10, Chipset Snapdragon 665 (11 nm), CPU Octa-core (4x2.0 GHz Kryo 260 Gold & 4x1.8 GHz Kryo 260 Silver).
- 3) Asus Zenfone Max Pro M1, OS Android versi 8.1, Chipset Snapdragon 636 (14 nm), CPU Octa-core (4x1.8 GHz Kryo 260 Gold & 4x1.6 GHz Kryo 260 Silver).

3.1. Pengujian Aplikasi dengan Metode *Black Box*

Hasil pengujian aplikasi pada tiga perangkat tersebut ditampilkan pada tabel 4 berikut. Pada pengujian dengan metode *Black Box* diperlukan indikator berdasarkan bentuk pengujiannya masing-masing yang menentukan apakah hasil uji berhasil atau tidak.

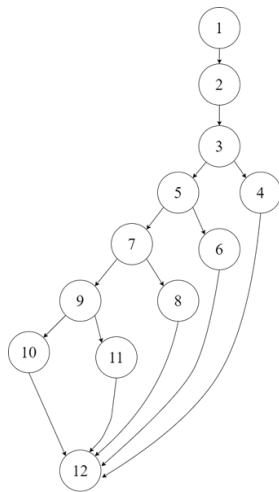
Tabel 4. Hasil pengujian *Black Box*

No	Bentuk Pengujian	Hasil Pengujian		
		Samsung A50	Xiaomi Redmi Note 8	Asus Zenfone Max Pro M1
1	Memasang aplikasi pada perangkat Android	Berhasil	Berhasil	Berhasil
2	Menampilkan menu <i>MainActivity</i>	Berhasil	Berhasil	Berhasil
3	Menampilkan <i>settingsactivity</i>	Berhasil	Berhasil	Berhasil
4	Mengaktifkan <i>Start Sharing</i> dengan kondisi aplikasi sudah memiliki izin akses ke layanan lokasi <i>smartphone</i> , fitur gps dalam keadaan aktif, <i>server URL</i> yang dimasukkan pada <i>setting</i> sesuai	Berhasil	Berhasil	Berhasil
5	Menekan tombol stop sharing	Berhasil	Berhasil	Berhasil
6	Menunggu share duration hingga habis	Berhasil	Berhasil	Berhasil
7	Mengakses <i>webview</i>	Berhasil	Berhasil	Berhasil
8	Menekan <i>icon landmark</i> pada <i>webview</i>	Berhasil	Berhasil	Berhasil
9	Menekan <i>icon</i> lokasi pengguna pada <i>webview</i>	Berhasil	Berhasil	Berhasil
10	Menekan tombol <i>refresh page</i> pada <i>webview</i>	Berhasil	Berhasil	Berhasil

Berdasarkan Tabel 4, dapat dilihat bahwa bahwa pengujian terhadap fitur-fitur yang ada pada aplikasi dapat berjalan dengan baik dan berhasil pada semua perangkat yang digunakan untuk pengujian.

### 3.2. Pengujian Aplikasi dengan Metode White Box

Pengujian dengan menggunakan metode white box dilakukan dengan cara basis path testing. Basis path testing dilakukan dengan mengeksekusi semua jalur independen minimal satu kali. pengujian ini dilakukan pada logika tampilan yang seharusnya ada pada aplikasi ketika melakukan pengiriman data dari aplikasi ke server. Jumlah jalur independen yang dieksekusi, ditentukan dengan menghitung *cyclomatic complexity* dan menentukan notasi diagram alir. Notasi diagram alir ditunjukkan pada gambar 9.



Gambar 9. Notasi diagram alir aktivitas Start Sharing

Sedangkan untuk hasil pengujian ditunjukkan pada tabel 5.

Tabel 5. Hasil pengujian white box

No	Jalur									Ketercapaian		
	1	2	3	4	5	6	7	8	9	10	Ya	Tidak
1	1	1	2	3	5	7	9	10			✓	
2	2	1	2	3	4	12					✓	
3	1	2	3	5	6	12					✓	
4	1	2	3	5	7	8	12				✓	
5	1	2	3	5	7	9	11	12			✓	

Berdasarkan hasil pengujian *white box* pada tabel 4.3 dengan menggunakan *basis path testing*, ditunjukkan bahwa seluruh *test case* berhasil dieksekusi minimal satu kali sehingga dapat disimpulkan bahwa aplikasi lolos pengujian *white box*.

### 4. Kesimpulan

Ketika pengguna memasukkan *server URL* yang tidak sesuai pada bagian *setting*, atau *smartphone* pengguna tidak memiliki akses internet, maka aplikasi akan menampilkan pemberitahuan bahwa koneksi ke *server* gagal. Aplikasi akan menampilkan pemberitahuan bahwa aplikasi membutuhkan izin akses ke layanan lokasi

*smartphone* jika pengguna belum memberikan izin kepada aplikasi dan langsung menekan tombol *Start Sharing*. Pemberitahuan juga akan muncul ketika pengguna ingin melakukan pengiriman data ke server tetapi layanan GPS pada *smartphone* pengguna belum diaktifkan. Selain itu, Aplikasi juga akan menampilkan pemberitahuan bahwa pengiriman data telah berakhir ketika *Share Duration* sudah habis atau pengguna menghentikan proses pengiriman data ke *server*. Berdasarkan pengujian aplikasi dengan menggunakan metode *Black Box*, didapatkan hasil bahwa semua fungsi aplikasi bekerja dengan baik pada tiga perangkat uji yang digunakan sedangkan pada pengujian *White Box* didapatkan hasil bahwa semua *test case* berhasil dilakukan.

### Referensi

- [1]. N. Sifaat, *Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet Berbasis Android*, Bandung, Indonesia: Informatika, 2012.
- [2]. N. Gandhewar dan R. Sheikh, "Google Android: An emerging software platform for mobile devices," *Int. Journal on Computer Science and Engineering*, vol. 1, no. 1, hal. 12-17, Nov, 2009.
- [3]. A. Küpper, *Location-Based Services: Fundamentals and Operation*, Chichester, Inggris: John Wiley & Sons, Ltd, 2005.
- [4]. A. Kushwaha dan V. Kushwaha, "Location Based Services using Android Mobile Operating System", *International Journal of Advances in Engineering & Technology* vol. 1, no. 1, hal. 14-20, Mar. 2011.
- [5]. B. R. Rompas, A. A. E. Sinsuw, S. R. U. A. Sompie, A. S. M. Lumenta, "Aplikasi Location-Based Service Pencarian Tempat Di Kota Manado Berbasis Android," *Jurnal Teknik Elektro Dan Komputer Universitas Sam Ratulangi*, vol. 1, no. 2. [Online] Tersedia: <https://ejournal.unsrat.ac.id/index.php/elekdankom/article/view/600>. Diakses: Juni, 14, 2020.
- [6]. S. Steiniger, M. Neun, A. Edwardes, *Foundations of Location Based Services*, Zürich, Swiss: University of Zurich, ETH Zurich, 2006.
- [7]. A. J. Brimicombe, "GIS - Where are the frontiers now?," dalam *Proceedings GIS 2002*, Bahrain, 2002, hal. 33-45.
- [8]. G. F. Bonham-Carter, *Geographic Information Systems for Geoscientists: Modelling with GIS*, New York, Amerika Serikat: Pergamon, 1994.
- [9]. S. G. Edward, N. Sabharwal, *Practical MongoDB*, Berkeley, United States of America: Apress, 2015.
- [10]. A. Giamas, *Mastering MongoDB 3.x*, Birmingham, Inggris: Packt Publishing, 2017.
- [11]. P. Bolstad, *GIS Fundamentals: A First Text on Geographic Information Systems*, Minnesota, United States of America: Eider Press, 2016.
- [12]. A.S. Rosa, Shalahuddin M, *Rekayasa Perangkat Lunak*, Bandung, Indonesia: Penerbit Informatika, 2014.
- [13]. B. Smith, *Beginning JSON*, California, United States of America: Apress, 2015.
- [14]. M. Lewin, *Leaflet.js Succinctly*, Morrisville, United States of America: Syncfusion, 2016.
- [15]. D. Goodman, *JavaScript Bible*, 4<sup>th</sup> ed, New York, United States of America: Hungry Minds, 2001.