

# Analisis Hasil Implementasi Konsep Context-Aware Pada Aplikasi Mobile Family Tracking Untuk Platform Android

Agustian  
Jurusan Teknologi Informasi  
Politeknik Caltex Riau  
agustian14ti@mahasiswa.pcr.ac.id

Anggy Trisnadoli  
Jurusan Teknologi Informasi  
Politeknik Caltex Riau  
anggy@pcr.ac.id

Indah Lestari  
Jurusan Teknologi Informasi  
Politeknik Caltex Riau  
indah@pcr.ac.id

**Abstrak** – Pada bulan Juni 2017, dilakukan survei terhadap 44 orang tua yang bekerja dan pengguna *smartphone* di Pekanbaru, diketahui bahwa orang tua membutuhkan aplikasi untuk memantau keberadaan anak, namun belum ada orang tua yang menggunakannya. Pada penelitian ini dikembangkan aplikasi yang dapat memantau keberadaan anggota keluarga yang menerapkan konsep *Context-Aware* dimana penalaran yang digunakan adalah *rules*. Ada 3 *rules* yang digunakan dimana ketika *rules* terpenuhi maka akan memunculkan notifikasi kepada pengguna. Aplikasi ini dikembangkan pada *platform Android* dengan menggunakan bahasa pemrograman Java, IDE Android Studio, dan Firebase sebagai *database*. Dilakukan 3 pengujian pada penelitian ini, pertama pengujian fungsionalitas yang dilakukan dengan metode *blackbox* dengan mengacu pada standar ISO 9126-2, dimana diperoleh bahwa aplikasi berfungsi dengan benar, diimplementasikan sesuai spesifikasi, dan fitur sesuai dengan tugasnya yang artinya secara fungsional aplikasi sudah berkualitas baik. Kedua adalah pengujian aplikasi kepada 5 keluarga untuk membuktikan *context-aware* berjalan semestinya. Dari hasil uji *context-aware* yang dilakukan diperoleh bahwa *context-aware* sudah berjalan dengan baik yang dibuktikan dengan notifikasi yang sesuai dengan *rules context-aware* dan setiap keluarga dapat memantau lokasi dan waktu kegiatan setiap anggota keluarga secara *realtime*. Ketiga adalah pengujian performansi, dimana diperoleh bahwa semakin banyak aktivitas dikerjakan semakin besar memori digunakan. Penggunaan CPU tidak bergantung pada aktivitas dan penggunaan data pada aktivitas yang sudah pernah berjalan akan lebih kecil dibanding pertama kali dijalankan.

**Kata Kunci:** Android, *Context-Aware*, Firebase, ISO 9126-2, *Performance Testing*.

## I. PENDAHULUAN

Menurut survei yang telah dilakukan di Pekanbaru terhadap 44 orang tua yang bekerja dan pengguna aktif *smartphone* pada bulan Juni 2017, didapatkan bahwa untuk

memantau keberadaan anak oleh orang tua dilakukan dengan mengawasi langsung, melakukan panggilan melalui telepon, memantau media sosial anak dan mendampingi anak belajar. Belum ada orang tua yang menggunakan aplikasi untuk melakukan pemantauan lokasi anaknya.

Selanjutnya dilakukan survei kembali pada bulan Juni 2017 berdasarkan hasil survei sebelumnya. Survei dilakukan terhadap 15 orang tua yang bekerja, pengguna *smartphone* Android, dan memiliki anak dengan usia minimal 17 tahun. Berdasarkan hasil survei didapatkan bahwa rata-rata orang tua berada dan berinteraksi di dalam rumah sekitar 8 jam sehari dan sisanya dilakukan di luar rumah. Selain itu, didapatkan juga bahwa mayoritas anak lebih banyak melakukan aktivitas di luar rumah. Dikarenakan waktu interaksi orang tua dan anak di luar rumah lebih besar dari pada di dalam rumah maka muncul kekhawatiran orang tua terhadap anaknya. Maka dari itu orang tua perlu memantau lokasi anaknya. Hal yang sama juga berlaku untuk orang tua, disaat anak ingin mengetahui keberadaan orang tuanya hal yang dilakukan pertama adalah menelepon dan mengirimkan pesan kepada orang tuanya.

Berdasarkan masalah di atas maka dibutuhkan sebuah aplikasi yang dapat melakukan pemantauan lokasi anggota keluarga satu sama lain. Aplikasi dapat berjalan pada *platform* Android dan aplikasi dapat mengirimkan notifikasi jika terjadi perubahan lokasi pada anak. Dimana tujuan aplikasi ini adalah untuk melakukan pemantauan lokasi anak secara *realtime*.

Penelitian ini menawarkan solusi untuk membangun aplikasi pada *platform* Android dengan *GPS* dan koneksi internet yang dapat melakukan *family tracking* atau deteksi keberadaan anggota keluarga. Aplikasi dirancang menggunakan konsep *aware* terhadap perubahan *context* atau yang disebut sebagai *context-aware*. *Context* yang digunakan adalah lokasi dan waktu serta *aware* yang diberikan berupa notifikasi kepada keluarga jika *rules* terpenuhi. Akuisisi *context* menggunakan sensor *GPS* pada *smartphone* untuk lokasi berupa *longitude* dan *latitude* serta waktu berdasarkan *smartphone*. Aplikasi pada penelitian ini menggunakan *Backend as a Service* (BaaS) dari Google yaitu Firebase yang dibangun pada IDE Android Studio menggunakan bahasa pemrograman Java.

II. METODOLOGI PENELITIAN

A. Penelitian Terdahulu Tentang Aplikasi Context-Aware

Penelitian tentang aplikasi tracking pernah dilakukan untuk Friend Tracker dimana pengguna aplikasi dapat mengetahui lokasi teman dan mendapatkan jarak terbaik antar pengguna dan temannya [1]. Selain itu, penelitian juga pernah dilakukan untuk mengetahui lokasi dari sebuah smartphone, dimana aplikasi memanfaatkan GPS pada perangkat smartphone untuk mengetahui lokasi sebuah smartphone tersebut [2].

Selanjutnya penelitian yang digunakan untuk melacak lokasi sales dimana supervisor pada aplikasi ini dapat melihat lokasi sales melalui website dengan tampilan Google Maps [3]. Dan penelitian aplikasi tracking pernah dilakukan untuk mengetahui lokasi anak dimana orang tua juga dapat memberikan batasan area bagi anaknya [4].

Perbandingan penelitian sekarang dengan penelitian terdahulu dapat dilihat pada Tabel 1.

Tabel 1. Perbandingan Fitur Penelitian Terdahulu Dengan Penelitian Sekarang.

Peneliti	Fitur Penelitian
[1]	Pengguna melihat lokasi teman dan jarak terbaik antar pengguna dan teman
[2]	Pengguna melacak lokasi smartphone
[3]	Supervisor melacak lokasi sales melalui website
[4]	Orang tua melihat lokasi anak dan menetapkan batasan area untuk anak
Penelitian saat ini	Anggota keluarga melacak lokasi antar anggota keluarga, melihat jadwal anggota keluarga, dan menetapkan batas area untuk anggota keluarga

B. Mobile Application Context-Aware

Aplikasi mobile adalah aplikasi yang berjalan pada perangkat mobile. Dalam pengembangan sebuah aplikasi terdapat konsep yang disebut sebagai context-aware. Sebuah sistem dapat dikatakan context-aware jika sistem memiliki kemampuan untuk peka terhadap perubahan context pada suatu lingkungan berdasarkan kondisi tertentu yang ditentukan sistem [5]. Context sendiri adalah informasi yang digunakan menggambarkan atau mengklasifikasi lingkungan suatu entitas [6]. Salah satu penerapan context-aware adalah geofence. Geofence adalah sebuah pembatas virtual dalam hal pergerakan objek pada suatu area, sehingga ketika objek melewati pembatas akan diberikan suatu aware kepada pemantau [4].

Dalam pengembangan aplikasi mobile terkadang dibutuhkan sebuah halaman administrasi tertentu yang hanya dapat diakses oleh user tertentu yang disebut sebagai backend. Salah satu backend yang ditawarkan Google adalah Firebase. Terdapat beberapa fitur yang ditawarkan oleh Firebase yaitu seperti, Analytics, Cloud Messaging, Cloud Function, Authentication, Realtime Database, Storage, Hosting, Test Lab, dan Crash Reporting [7]. Pada penelitian ini menggunakan beberapa fitur dari Firebase berupa

Firestore, Firebase Authentication, Firebase Realtime Database dan Firebase Storage. Firebase Authentication digunakan untuk melakukan otentikasi akun email yang dimasukkan saat melakukan pendaftaran pengguna pada aplikasi penelitian. Database Realtime pada Firebase menggunakan database NoSQL. NoSQL berbeda dengan SQL yang menyimpan data dalam bentuk record-record di dalam table pada database relational, NoSQL menyimpan data tidak dalam bentuk table serta NoSQL tidak memiliki relasi. Database Realtime Firebase menggunakan struktur data JSON Tree. Firebase Realtime Database menyimpan data-data teks sedangkan untuk Firebase Storage digunakan untuk menyimpan gambar-gambar yang dimasukkan pengguna pada aplikasi.

III. HASIL DAN PEMBAHASAN

A. Perancangan Context-Aware

Konsep context pada penelitian ini menggunakan dua data yaitu data lokasi dan data waktu. Context diakuisisi menggunakan sensor GPS pada smartphone dan waktu. Setelah context didapatkan maka context akan dimasukkan kedalam sebuah context reasoning dengan teknik rules dimana terdapat 3 buah rules. Sebelumnya, pengguna akan melakukan registrasi aplikasi lalu setelah itu pengguna dapat membuat grup keluarga atau bergabung dengan grup keluarga menggunakan kode grup. Kemudian pengguna memasukkan jadwal kegiatan sehari-hari baik itu di dalam ataupun di luar rumah. Data kegiatan yang dimasukkan mencakup waktu kegiatan dan lokasi kegiatan dalam bentuk longitude dan latitude. Kemudian anggota keluarga lain dapat melakukan geofence atau pembatasan area terhadap jadwal pengguna yang sudah dimasukkan. Setelah dilakukan geofence maka ketika terjadi perubahan lokasi atau waktu akan dilakukan pengecekan waktu dan lokasi sehingga menghasilkan aware. Adapun rules yang diterapkan saat pengecekan adalah sebagai berikut dengan merujuk pada hasil survei dan user requirement [8][9]:

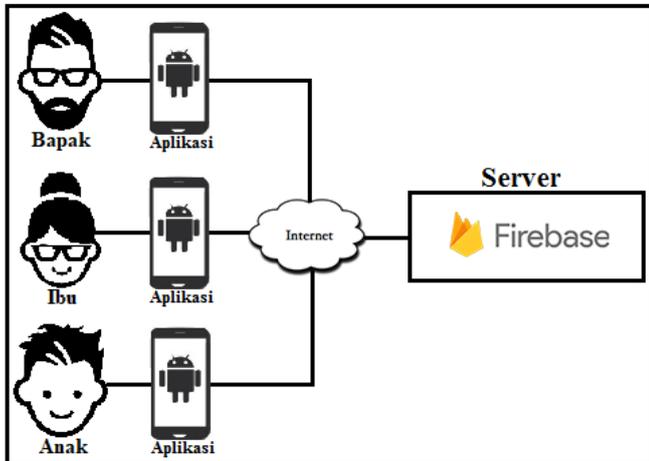
Tabel 2. Rule Context-Aware.

No	Rule
1	Jika waktu dan lokasi anggota keluarga tidak ada pada jadwal atau jadwal anggota keluarga kosong maka mengirimkan notifikasi
2	Jika waktu jadwal pribadi berada dalam range 10 menit sebelum waktu berakhir maka mengirimkan notifikasi
3	Jika lokasi anggota keluarga tidak sesuai dengan lokasi dijadwal yang dimasukkan maka mengirimkan notifikasi

Dalam mendistribusikan context maka akan digunakan teknik subscription sehingga akan mengirimkan aware berupa notifikasi secara berkala jika salah satu rule pada context reasoning terpenuhi sehingga pemrosesan dilakukan secara realtime. Ketiga rules ini diterapkan didalam service pada aplikasi.

B. Desain Arsitektur

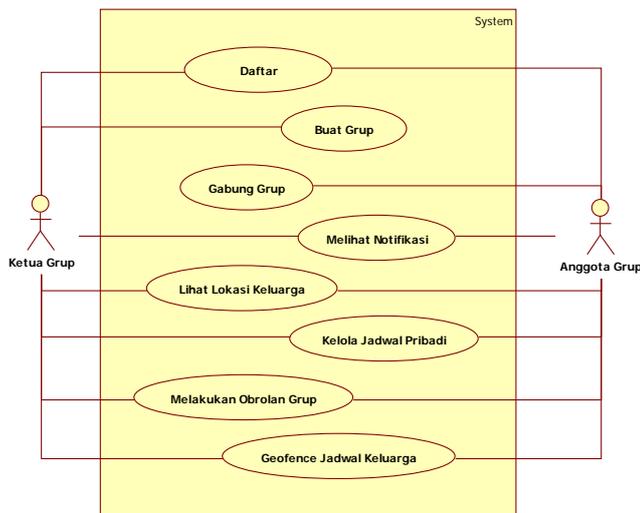
Perancangan arsitektur aplikasi pada penelitian ini dalam dilihat pada Gambar 1. Dimana pengguna akan berinteraksi dengan aplikasi melalui *smartphone* Android. *Smartphone* harus terhubung pada internet untuk mengakses *server* Firebase yang menyediakan *database* untuk aplikasi ini.



Gambar 1. Arsitektur Aplikasi.

C. Use Case Diagram

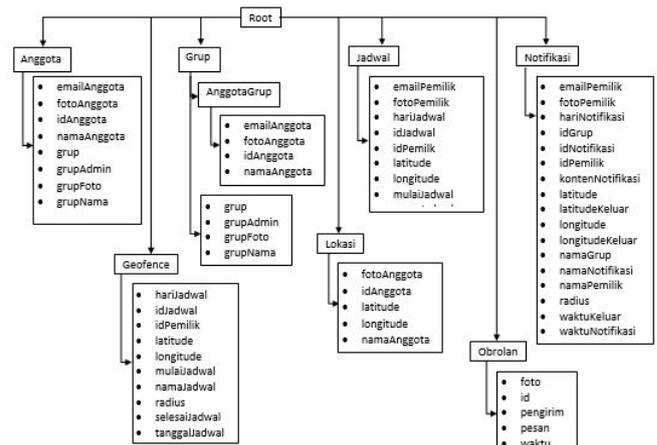
*Use Case Diagram* memuat kegiatan yang dapat dilakukan pengguna terhadap sistem. Perancangan *Use Case Diagram* dapat dilihat pada Gambar 2 dimana terdapat 2 buah aktor utama yaitu Ketua Grup dan Anggota Grup. Terdapat 8 buah *Use Case* yang dapat dilakukan oleh pengguna terhadap sistem.



Gambar 2. Use Case Diagram.

D. Database Diagram

Penyimpanan data yang disediakan Firebase adalah *Database Realtime* yang berbentuk *database NoSQL* dengan bentuk *JSON Tree*. Perancangan *database* berdasarkan kebutuhan aplikasi dapat dilihat pada Gambar 3.

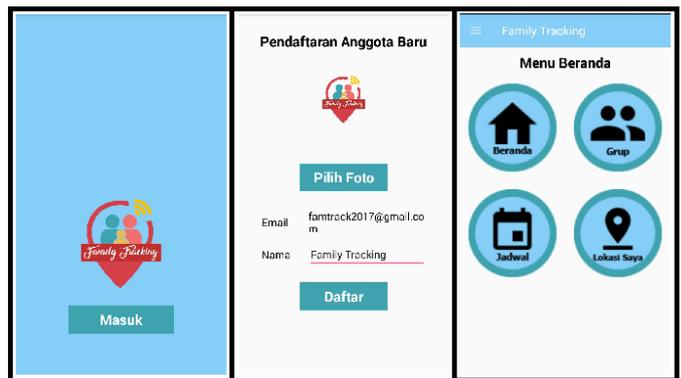


Gambar 3. Database Diagram.

Jadi dalam perancangan aplikasi ini akan berhubungan dengan data anggota, grup, jadwal, lokasi, obrolan, *geofence* dan notifikasi. Dimana data akan disimpan dalam Firebase dan untuk mengaksesnya dibutuhkan koneksi internet.

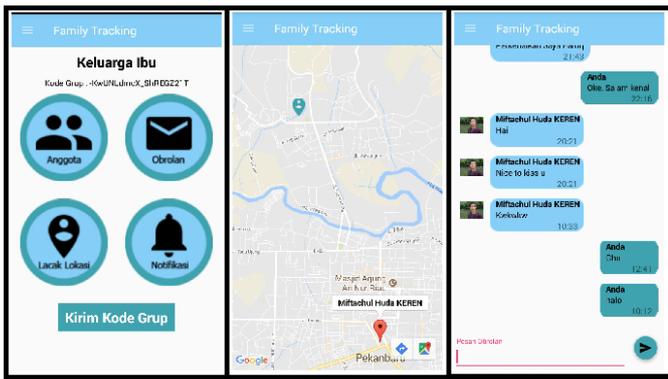
E. Analisis Hasil Aplikasi Mobile Family Tracking

Hasil pada penelitian ini adalah sebuah aplikasi *Family Tracking* yang dapat menampilkan lokasi keluarga secara *realtime* dan dapat memberikan *geofence* kepada jadwal anggota keluarga. Dalam penerapannya dapat dilihat pada Gambar 4 sampai Gambar 6.



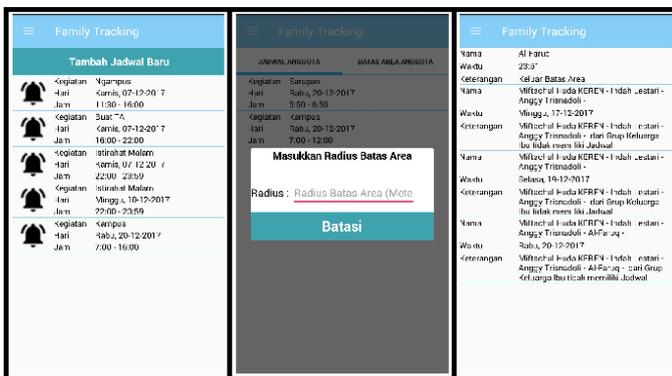
Gambar 4. Tampilan Registrasi.

Pada Gambar 4, ditampilkan 3 buah tampilan. Gambar kiri merupakan tampilan *login* aplikasi menggunakan akun *email* yang sudah terdaftar baik itu akun *Google*, *Yahoo!*, *Outlook*, dan sebagainya. Gambar tengah tampilan registrasi yang meminta pengguna memasukkan foto serta mengubah nama. Gambar kanan merupakan tampilan beranda setelah pengguna sudah melakukan registrasi.



Gambar 5. Tampilan Fitur Grup.

Pada Gambar 5 bagian kiri merupakan tampilan grup dimana pengguna dapat melihat kode grup, mengirim kode grup, *button* anggota, obrolan, lacak dan notifikasi. Gambar tengah merupakan tampilan lacak lokasi keluarga. Gambar kanan merupakan tampilan obrolan grup.



Gambar 6. Tampilan Terkait Jadwal.

Gambar 6 menampilkan tampilan yang berkaitan dengan jadwal. Gambar kiri merupakan tampilan kelola jadwal pribadi pengguna. Gambar tengah merupakan tampilan *geofence* jadwal anggota keluarga. *Geofence* dapat dilakukan setelah anggota keluarga memasukkan jadwal kegiatannya. *Geofence* hanya diketahui dan diubah oleh anggota keluarga yang melakukan *geofencing* pada jadwal tersebut, misalkan Ibu melakukan *geofencing* jadwal anak maka hanya Ibu yang bisa merubah dan melihat *geofence* tersebut. Gambar kanan merupakan tampilan untuk melihat notifikasi.

Bedasarkan hasil aplikasi diatas, dilakukan pengujian fungsionalitas dengan metode pengujian *black box* yang mengacu pada standar ISO 9126-2 dengan faktor kualitas *functionality* dengan sub faktor kualitas *suitability*. Pengujian dilakukan untuk mengetahui kualitas dari aplikasi yang dibangun dalam hal fungsionalitas.

Pengujian *functionality* dilakukan terhadap 3 (tiga) metrik pada sub faktor kualitas *suitability*, yang digunakan pada penelitian ini yaitu *Functional Adequacy* (FA), *Functional Implementation Completeness* (FICM) dan *Functional Implementation Coverage* (FIC). Pengujian dilakukan oleh pengembang dan 5 pengguna yang dipilih secara acak.

Pengujian dilakukan terhadap 13 *test case* yang dilakukan berdasarkan fitur dan *service* yang digunakan pada aplikasi. Nilai dari hasil *test case* berada pada *range* 0 dan 1. Dengan nilai 0 (nol) dikatakan tidak baik dan nilai mendekati atau sama dengan 1 (satu) dikatakan baik. Salah satu contoh *test case* yang dilakukan pada fitur *Login* dapat dilihat pada Tabel 3, pengujian fungsionalitas keseluruhan dapat dilihat pada Tabel 4.

Tabel 3. Contoh *Test Case* ISO 9126-2 Fitur *Login*.

<b>Tester Name</b>	<b>Agustian</b>
<b>Test Type</b>	Tes Fungsional
<b>Test Case Name</b>	<i>Login</i>
<b>Test Case Number</b>	FT – 019
<b>Test Case Description</b>	<p><i>Test case</i> ini memastikan bahwa <i>Login activity</i>:</p> <ul style="list-style-type: none"> <li>- Diimplementasikan sesuai dengan fungsinya</li> <li>- Sesuai dengan spesifikasi kebutuhan.</li> <li>- Berfungsi dengan benar/dapat digunakan</li> </ul>
<b>Item to be used</b>	

- Menampilkan akun terdaftar pada *smartphone*
- Mengarahkan ke *Loading Acitivity*
- Mengarahkan ke *Daftar Activity*

<b>Specifications</b>			
No.	Input	Expected Output	Actual Output
1.	Jalankan Aplikasi pertama kali	Mengarahkan ke fitur <i>Login</i> dan menampilkan akun <i>email</i>	<ul style="list-style-type: none"> <li>• Sesuai</li> <li>• Benar</li> </ul>
2.	Pilih akun atau masukkan akun	Cek akun pada <i>database</i> . Jika terdaftar maka ke <i>menyarahkan</i> fitur Beranda	<ul style="list-style-type: none"> <li>• Sesuai</li> <li>• Benar</li> </ul>
3.	Pilih akun atau masukkan akun baru	Cek akun pada <i>database</i> . Jika tidak ada maka <i>menyarahkan</i> ke fitur <i>Daftar</i>	<ul style="list-style-type: none"> <li>• Sesuai</li> <li>• Benar</li> </ul>
<b>Metrik Functional Adequacy (FA)</b>		$FA = 1 - (\text{fitur tidak sesuai}) / (\text{fitur sesuai})$ $= 1 - 0/3 = 1$	
<b>Metrik Functional Implementation Coverage (FIC)</b>		$FIC = 1 - (\text{fitur tidak berfungsi}) / (\text{fitur berfungsi})$ $= 1 - 0/3 = 1$	
<b>Metrik Functional Implementation Completeness (FICM)</b>		$FICM = 1 - (\text{fitur tidak diimplementasikan}) / (\text{fitur diimplementasikan})$ $= 1 - 0/3 = 1$	

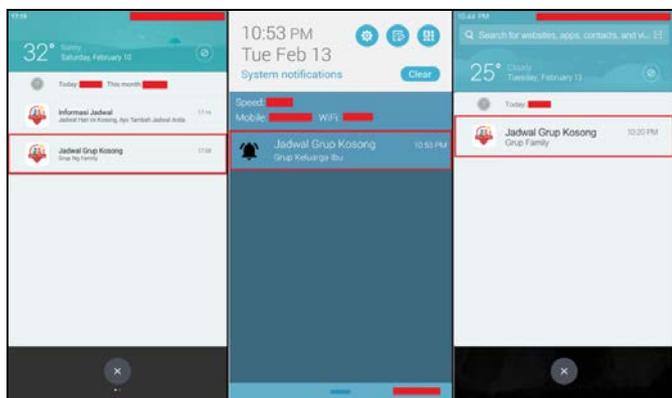
Tabel 4. Pengujian Fungsionalitas.

No	Fitur	FA	FIC	FICM
1	Login	1	1	1
2	Daftar	1	1	1
3	Beranda	1	1	1
4	Menu Grup	1	1	1
5	Buat Grup	1	1	1
6	Gabung Grup	1	1	1
7	Detail Grup	1	1	1
8	Lacak Lokasi Keluarga	1	1	1
9	Obrolan Grup	1	1	1
10	Kelola Jadwal Pribadi	1	1	1
11	Geofence Jadwal	1	1	1
12	Melihat Notifikasi	1	1	1
13	Service Rules	1	1	1
<b>Rata - rata</b>		1	1	1

Berdasarkan pengujian Tabel 4, dihasilkan bahwa aplikasi berfungsi dengan benar bernilai 1, diimplementasikan sesuai spesifikasi bernilai 1, dan fitur sesuai dengan tugasnya bernilai 1 yang artinya secara fungsional aplikasi sudah berkualitas baik serta *context-aware* berhasil diterapkan dan berjalan dengan baik.

Pengujian kedua merupakan pengujian *context-aware* yang dilakukan terhadap 5 keluarga menggunakan media kuisioner setelah pemakaian aplikasi selama 3 hari. Hasil pengujian *context-aware* menampilkan hasil notifikasi dari *rules* dan perubahan lokasi secara *realtime* yang dapat dilihat pada Gambar 7 sampai Gambar 10.

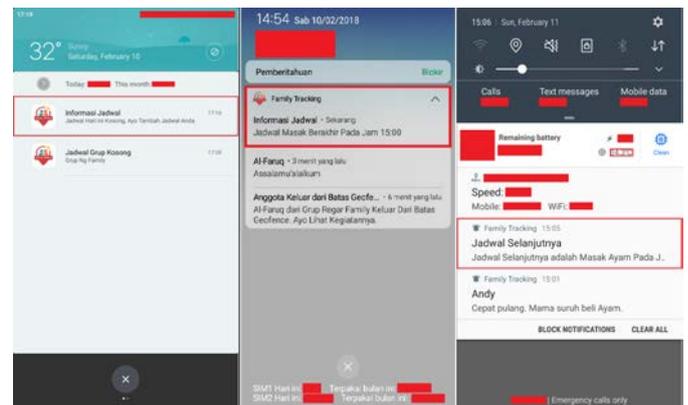
Hasil pengujian *Rule 1* dibagi menjadi 3 skenario dimana masing-masing skenario menghasilkan notifikasi jadwal grup kosong. Notifikasi ini hanya muncul kepada anggota keluarga yang membuat grup tersebut. Notifikasi ini akan muncul ketika salah satu atau seluruh anggota keluarga yang tergabung didalam grup tidak memasukkan jadwal harian mereka.



Gambar 7. Notifikasi Pengujian Rule 1.

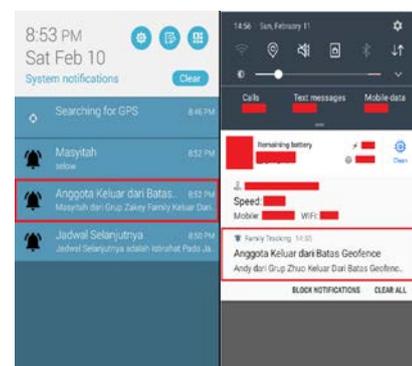
Hasil pengujian *Rule 2* dibagi menjadi 3 skenario pengujian dimana setiap skenario menghasilkan notifikasi yang berbeda. Pada skenario 1 untuk gambar pertama memunculkan notifikasi jadwal pribadi kosong, hal ini terjadi ketika pengguna tidak memasukkan jadwal untuk hari

tersebut. Skenario 2 untuk gambar ditengah menghasilkan notifikasi jadwal akan berakhir pada waktu atau jam 15.00. Pengujian dilakukan ketika Ibu memasukkan jadwal dengan nama “Masak” dengan rentang waktu 14.00 sampai 15.00. Ketika notifikasi muncul, waktu *smartphone* menunjukkan waktu 14.54 dimana jadwal akan berakhir 6 menit kemudian. Skenario 3 memunculkan notifikasi jadwal selanjutnya, dimana pengujian dilakukan ketika Ibu memasukkan jadwal dengan nama “Masak Ayam” dengan rentang waktu 15.10-16.00. Notifikasi muncul ketika waktu *smartphone* menunjukkan waktu 15.06 dimana jadwal akan dimulai 4 menit kemudian.



Gambar 8. Notifikasi Pengujian Rule 2.

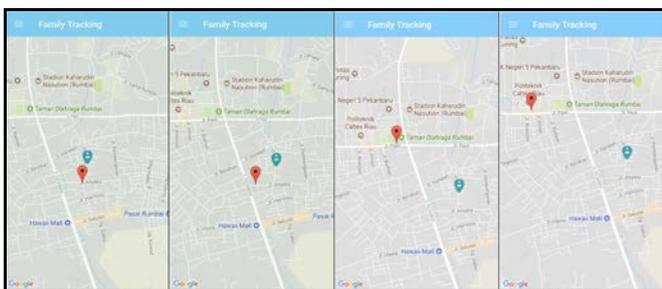
Hasil pengujian *Rule 3* dibagi menjadi 2 skenario dimana menghasilkan notifikasi bahwa anggota keluar dari batas *geofence*. Hasil ini terjadi ketika anggota keluarga melakukan *geofencing* pada jadwal anggota keluarga lain, dan anggota keluarga tersebut berjalan keluar dari batas area *geofence* yg telah dimasukkan. Sebagai contoh skenario 1 terjadi ketika “Zaki” dari grup “Zakey Family” melakukan *geofence* pada jadwal “Marsyiah” dari grup yang sama. Zakey memberikan batas *geofence* sejauh 300m. Lalu Marsyiah bergerak sejauh 400m sehingga keluar dari batas *geofence* yang diberikan Zakey maka Zakey akan menerima notifikasi seperti pada Gambar 9 sebelah kiri.



Gambar 9. Notifikasi Pengujian Rule 3.

Pengujian melacak lokasi secara *realtime* pada Gambar 10 terjadi ketika saat anggota keluarga melakukan perpindahan lokasi. Skenario dibawah terjadi ketika anggota

keluarga bergerak dari Jalan Ampera, Rumbai menuju Jalan Umban Sari, Rumbai. Gambar 10 dibagi menjadi 3 untuk melihat garis besar pergerakan dan hal ini terjadi secara *realtime*.



Gambar 10. Pengujian Lacak Lokasi Secara *Realtime*.

Hasil notifikasi pengujian *rule* 1, 2, dan 3 menunjukkan bahwa aplikasi dan *context-aware* sudah berjalan dengan sesuai dengan *rule* yang telah ditetapkan. Pengujian untuk lacak lokasi dilakukan secara *realtime* dimana saat salah satu anggota berjalan maka titik lokasi yang ditampilkan pada aplikasi berubah sesuai lokasi serta didapatkan hasil bahwa pelacakan lokasi berjalan secara *realtime*.

Selain pengujian secara langsung terhadap 5 keluarga, dilakukan juga kuisisioner setelah penggunaan aplikasi ini selama 3 hari. Kuisisioner dilakukan terhadap 11 anggota keluarga dari 5 keluarga. Kuisisioner berisi 5 pernyataan seperti Tabel 6 dengan skala Likert, dimana setiap anggota keluarga memilih apakah Sangat Setuju, Setuju, Netral, Tidak Setuju dan Sangat Tidak Setuju pada setiap pernyataan yang diberikan. Setiap pilihan untuk pernyataan memiliki nilai seperti Tabel 5.

Tabel 5. Skor Skala Likert.

No	Nilai	Skor
1	Sangat Setuju (SS)	5
2	Setuju (S)	4
3	Netral (N)	3
4	Tidak Setuju (TS)	2
5	Sangat Tidak Setuju (STS)	1

Perhitungan analisa dilakukan dengan mengalikan jumlah skor responden dengan nilai tiap pilihan pada pernyataan. Kemudian menambahkan setiap hasil perkalian pada setiap pilihan dan dibagi dengan jumlah maksimal skor dan dikalikan 100%. Berikut contoh perhitungan untuk pernyataan 1.

- Jumlah skor untuk 8 responden sangat setuju =  $5 * 8 = 40$
- Jumlah skor untuk 3 responden netral =  $3 * 3 = 9$
- Total Skor =  $0 + 0 + 9 + 0 + 40 = 49$
- Persentase =  $(49 / 55) * 100 \% = 89\%$

0 ----- STS ----- TS ----- N ----- S ----- SS

↑  
89%

Hasil perhitungan analisa setiap pernyataan kuisisioner dapat dilihat pada Tabel 6.

Tabel 6. Hasil Analisa Kuisisioner.

No	Pernyataan	Persentase
1	Aplikasi Memiliki Tampilan yang Mudah Digunakan	89%
2	Aplikasi Memudahkan Memantau Keberadaan Keluarga	93%
3	Lokasi yang Ditampilkan Merupakan Lokasi Terkini	91%
4	Notifikasi yang Dimunculkan Sesuai Dengan Kondisi <i>Geofence</i>	87%
5	Notifikasi yang Dimunculkan Sesuai Dengan Jadwal Pengguna	87%

Hasil analisa kuisisioner didapatkan bahwa responden memberikan respon positif dengan persentase  $\geq 85\%$  dengan nilai setuju. Hal ini membuktikan bahwa responden setuju aplikasi memiliki tampilan yang mudah digunakan dan aplikasi ini memberikan kemudahan dalam memantau keberadaan keluarga. Selain itu lokasi yang ditampilkan pada aplikasi merupakan lokasi terkini atau *realtime* dan juga notifikasi yang dimunculkan sesuai dengan kondisi *geofence* serta sesuai dengan jadwal pengguna.

Pengujian ketiga merupakan pengujian performansi yang dilakukan untuk mengetahui kualitas dari aplikasi dalam hal *performance* atau kinerja. Pengujian performansi dilakukan terhadap penggunaan CPU, *memory*, dan *data amount*. Untuk mendukung pelaksanaan *performance testing* akan digunakan *tools* yang dapat membantu pengumpulan data dan visualisasi data. Android Studio menyediakan *tools* pembuatan profil untuk merekam dan memvisualisasikan kinerja aplikasi yaitu Android *Profiler*. Dalam melakukan pengujian maka digunakan sebuah perangkat *smartphone* dengan spesifikasi seperti pada Tabel 7.

Tabel 7. Spesifikasi Perangkat Pengujian.

No	Fitur	Spesifikasi
1	Nomor Seri Perangkat	Nexus 4
2	Sistem Operasi	Android Jelly Bean 4.2
3	Memori	2 GB
4	CPU	1.512 GHz quad-core Krait
5	Penyimpanan Internal	8 Gb
6	Baterai	2100 mAH

Dilakukan percobaan dimana sebuah *activity* akan dilakukan sebanyak 5 kali secara berulang untuk melihat penggunaan memori. Penggunaan memori dibagi menjadi 2 yaitu *Peak* saat memori mencapai titik puncak proses dan Stabil saat memori tidak melakukan proses lagi.

Tabel 8. Pengujian Performa Memori.

No	Activity	Peak	Stabil
1	Beranda	40,25 MB	39,15 MB
2	Lokasi	55,31 MB	51,35 MB
3	Lokasi	56,83 MB	53,95 MB
4	Lokasi	60,46 MB	58,79 MB
5	Lokasi	61,79 MB	60,97 MB
6	Lokasi	63,66 MB	63,36 MB

Berdasarkan pengujian pada memori di atas dapat diketahui sebuah pola dimana saat menjalankan aktivitas, memori yang dipakai akan lebih besar kemudian memori akan berada dalam kondisi stabil. Maka dari pengujian diatas dihasilkan bahwa semakin banyak aktivitas dikerjakan, semakin besar memori digunakan.

Selanjutnya dilakukan pengujian pada CPU dengan menjalankan 2 percobaan dimana *activity* yang dijalankan sama tetapi berbeda urutan.

Tabel 9. Pengujian Performa CPU Percobaan 1.

No	Activity	CPU
1	Beranda	17,76 %
2	Grup	12,37 %
3	Jadwal	10,92 %
4	Lokasi	17,95 %

Tabel 10. Pengujian Performa CPU Percobaan 2.

No	Activity	CPU
1	Beranda	24 %
2	Lokasi	21,84 %
3	Grup	9,73 %
4	Jadwal	9,82 %

Dari percobaan diatas didapatkan bahwa penggunaan CPU tidak memiliki ketergantungan urutan pada aktivitas. Penggunaan CPU terbesar pada *activity* yang menjalankan *map* yaitu pada Lokasi dan Beranda.

Pada pengujian untuk melihat penggunaan data dilakukan dengan menjalankan 7 *activity* yang kemudian diulang sesuai urutan sebelumnya untuk melihat *Data Amount* dari sisi pengiriman data dan penerimaan data.

Tabel 11. Pengujian Performa Data.

No	Activity	Data Receive	Data Sent
1	Grup	61,68 KB	37,78 KB
2	Keluarga	3,18 KB	4,13 KB
3	Lacak	6,45 MB	1,18 MB
4	Notifikasi	47,51 KB	1,05 KB
5	Obrolan	7,55 MB	0,141 MB
6	Anggota	9,72 MB	0,15 MB
7	Anggota Detail	5,37 MB	0,11 MB
8	Grup	0 KB	0 KB
9	Keluarga	0 KB	0 KB
10	Lacak	40,56 KB	51,27 KB

11	Notifikasi	0 KB	0 KB
12	Obrolan	0 KB	0 KB
13	Anggota	0 KB	0 KB
14	Anggota Detail	4,1 MB	0,01 MB

Berdasarkan pengujian performansi yang dilakukan pada percobaan ini didapatkan bahwa penggunaan data terbesar pada *activity* Anggota disusul oleh Obrolan, Lacak dan Anggota Detail. Selain itu, jika *activity* sudah pernah dijalankan dan akan dijalankan kembali maka penggunaan data selanjutnya lebih kecil dari pertama kali dijalankan karena Firebase pada Android akan menyimpan data sementara yang sudah dijalankan pada *disk* sehingga membuat penggunaan memori besar.

#### IV. KESIMPULAN

Berdasarkan pengujian dan analisa ISO 9126-2 dengan faktor kualitas *functionality* dengan sub faktor kualitas *suitability* pada penelitian ini dan pengujian yang dilakukan terhadap 5 keluarga di Pekanbaru serta pengujian performansi, maka dapat diambil kesimpulan sebagai berikut.

- Berdasarkan pengujian dan analisa ISO 9126-2 dengan faktor kualitas *functionality* dengan sub faktor kualitas *suitability* didapatkan bahwa aplikasi berfungsi dengan benar, diimplementasikan sesuai spesifikasi, dan fitur sesuai dengan tugasnya yang bernilai 1 yang berarti secara fungsional aplikasi sudah berkualitas baik.
- Pengujian *context-aware* yang dilakukan terhadap 5 keluarga diperoleh hasil bahwa *context-aware* sudah berjalan dengan baik yang dibuktikan dengan notifikasi yang dimunculkan dan setiap keluarga dapat memantau lokasi setiap anggota keluarga secara *realtime*.
- Hasil analisa kuisioner didapatkan bahwa responden setuju aplikasi memiliki tampilan yang mudah digunakan, aplikasi memberikan kemudahan dalam memantau keberadaan keluarga, lokasi yang dimunculkan merupakan lokasi terkini dengan notifikasi yang dimunculkan sesuai dengan kondisi *geofence* dan jadwal responden.
- Berdasarkan pengujian dan analisa performansi dapat diambil kesimpulan bahwa semakin banyak aktivitas dikerjakan semakin besar memori digunakan. Penggunaan CPU tidak bergantung aktivitas dan penggunaan data pada aktivitas yang sudah pernah berjalan akan lebih kecil dibanding pertama kali dijalankan. [1]

#### REFERENSI

- [1] Kusuma, W. & Septiani, T. (2013). Aplikasi Friend Tracker Berbasis Android Smartphone Menggunakan GPS Tracking. *Seminar Nasional Teknologi Informasi dan Multimedia*, Vol. 20, pp. 13-18. Yogyakarta.

- [2] Wahyudi, A, Andjarwirawan, J. & Handojo, A. (2013). *Perancangan Dan Pembuatan Aplikasi Tracker Position Dan Alert Pada Mobile Device Berbasis Android Yang Dapat Dipantau Via Website*. Surabaya : Universitas Kristen Petra.
- [3] Kurniawan, M.I. (2016). *Rancang Bangun Perangkat Lunak GPS Based Location Tracker Pada Platform Android Untuk Pelacakan Lokasi Sales PT . Meliana Perkasa*. Semarang : Universitas Dian Nuswantoro.
- [4] Segara, R. & Subari. (2017). Sistem Pemantauan Lokasi Anak Menggunakan Metode Geofencing pada Platform Android. *Teknologi & Manajemen Informatika*, Vol. 3, pp. 17-35.
- [5] Schilit, B.N., Adams, N. & Want, R. (1994). *Context-Aware Computing Applications*. *International Workshop on Mobile Computing Systems and Applications*, pp. 85-90.
- [6] Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M. & Steggles, P. (2000). Towards a Better Understanding of Context and Context-awareness. *Handheld and Ubiquitous Computing*. Berlin: Springer Berlin Heidelberg.
- [7] Firebase. (2017). *Firestore 2017*. Diakses dari: <https://firebase.google.com/> pada tanggal 19 Mei 2017.
- [8] Trisnadoli, A. & Lestari, I (2017). Software requirement analysis of “family tracking mobile application” on cross platform with hybrid approach. *International Conference on Data and Software Engineering (ICoDSE)*.
- [9] Lestari, I., & Tisnadoli, A. (2017). Usulan Model Kualitas Aplikasi Context Aware Mobile: Family Tracking pada Hybrid-Cross Platform. *Jurnal Komputer Terapan*, Vol. 3, pp. 261-270. Pekanbaru.