

DESAIN SISTEM INFORMASI LAPORAN KEKERASAN TERHADAP ANAK DAN PEREMPUAN (KtPA) BERBASIS ANDROID, GOOGLE FIREBASE, DAN WEB

Sadr Lufti Mufreni^{*)}, dan Luluk Rosida

Jurusan Teknologi Informasi, Universitas 'Aisyiyah Yogyakarta
Jalan Siliwangi (Ring Road Barat) 63, Mlangi, Nogotirto, Gamping, Sleman, D.I. Yogyakarta, 55292, Indonesia

^{*)}E-mail: sadr@unisayogya.ac.id

Abstrak

Pada tahun 2019, Komnas Perempuan melaporkan telah terjadi 406.178 kekerasan terhadap perempuan dan anak. Paling tinggi adalah kekerasan dalam rumah tangga (71%). Penanganan korban yang terlambat dapat menimbulkan resiko kematian. Diperlukan aplikasi yang dapat mengirimkan laporan kekerasan secara cepat dan mudah diakses oleh semua pihak baik korban maupun masyarakat sekitar. Penelitian ini menawarkan solusi aplikasi yang memenuhi kedua syarat tersebut. Aplikasi dapat diakses secara *real time* dan dapat digunakan oleh masyarakat dan kader Puskesmas. Aplikasi yang dibuat berbasis Android dan Google Firebase untuk manajemen laporan dan web untuk mengelola kader Puskesmas.

Kata kunci: android, firebase, kekerasan, anak, perempuan

Abstract

In 2019, Komnas Perempuan reported that there had been 406,178 violence against women and children. The highest was domestic violence (71%). Late handling of victims can pose a risk of death. A fast respond and easy access application is needed to send reports of violence quickly and can be accessed by all parties, both victims and the surrounding community. This study offers an application solution that meets both requirements. Applications can be accessed in real time and can be used by the community and Puskesmas cadres. An application made based on Android and Google Firebase for report management and the web for managing Puskesmas cadres..

Keywords: android, firebase, violence, children, woman

1. Pendahuluan

Penelitian bermula ketika peneliti melakukan kunjungan ke salah satu Puskesmas di Bantul. Pihak Puskesmas mengidentifikasi maraknya kasus kekerasan terhadap perempuan dan anak. Sayangnya hanya sebagian kecil kasus dilaporkan ke kepolisian. Alasannya adalah Kekerasan terhadap Perempuan dan Anak (KtPA) merupakan aib di keluarga.

Menurut [1] jumlah kasus kekerasan terhadap perempuan pada tahun 2019 sebesar 406.178. Kekerasan dalam rumah tangga menempati urutan pertama (71%) yang terdiri dari kekerasan fisik 3.927 kasus, kekerasan seksual 2.988 kasus, psikis 1.658 kasus, dan ekonomi 1.064 kasus.

Data tersebut merupakan fenomena gunung es, banyak korban enggan untuk melaporkan [2]. Salah satu penyebabnya adalah sulit untuk melapor: jarak ke kantor polisi jauh, atau pusat layanan kesehatan terdekat tutup. Keengganan untuk melapor bisa mengakibatkan hal yang lebih fatal, misalnya resiko kematian karena penanganan

yang terlambat. Kemudahan untuk melapor merupakan kunci untuk mengurangi resiko kematian dan memberikan penanganan yang cepat untuk korban KtPA.

Solusi yang ditawarkan adalah rancangan sistem informasi yang memberikan kemudahan pelaporan dan membantu pelayanan kesehatan untuk mengambil tindakan berdasarkan laporan secara real time.

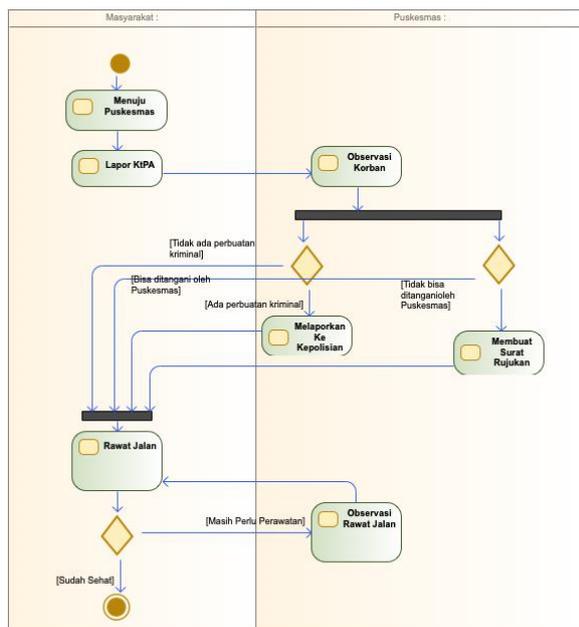
Penelitian sebelumnya [3] mempunyai 2 komponen utama yaitu pelapor dan puskesmas. Kelemahan dari [3], aplikasi bersifat pasif yaitu tidak ada notifikasi jika ada pelaporan baru, tidak adanya informasi lokasi kejadian menggunakan koordinat sehingga menyulitkan pencarian tempat kejadian, dan aplikasi hanya mendukung 2 jenis pengguna. Solusi ini menambahkan aplikasi untuk petugas pelayanan (kader) sehingga proses tindakan bisa lebih cepat. Penelitian lainnya [4][5][6] berfokus pada aspek selain teknologi, penelitian ini melengkapi penelitian tersebut. Aplikasi yang ada [7][8][9] tidak ditemukan aplikasi untuk kader dan puskesmas. Saat ini aplikasi [7] hanya digunakan untuk wilayah Sulawesi Utara. Rancangan aplikasi

menggunakan komputasi awan berbasis Google Firebase sehingga setiap puskesmas dapat mendaftar dan menggunakannya. Google Firebase digunakan untuk basis data dan notifikasi kepada pengguna secara *real time*. Penggunaan teknologi Google Firebase [10] yang membedakan solusi yang peneliti tawarkan dengan penelitian [4][5][6] atau solusi yang sudah ada [7][8][9]. Dengan teknologi ini diharapkan kekerasan terhadap anak dan perempuan dapat berkurang dan penanganan korban dapat dilakukan dengan cepat.

Perkembangan prosesor dan memori komputer yang semakin kecil dengan kapasitas dan kecepatan semakin cepat, ditambah harga komponen yang selalu turun mendorong munculnya gawai dengan harga yang terjangkau. Berdasarkan survei Statista, penggunaan gawai aktif di Indonesia melebihi 173 juta pada tahun 2017. Diperkirakan pengguna gawai yang memilih Android mencapai 80% pada bulan Juli 2019. Berdasarkan informasi tersebut, tulisan ini berfokus pada rancangan sistem informasi pelaporan KtPA di gawai Android [11][12]. Rancangan antar muka dan implementasi sistem informasi akan dibahas di artikel lain.

2. Analisa Kebutuhan

Pelaporan KtPA yang ada dimulai dengan korban atau rekan korban menghubungi pihak Puskesmas. Pihak Puskesmas akan melakukan turun ke lokasi jika korban tidak melapor sendiri. Setelah melakukan observasi diputuskan apakah korban dapat ditangani sendiri atau perlu diberi rujukan ke rumah sakit maupun laporan ke kepolisian. Penanganan berkala dilakukan jika korban memiliki trauma (Gambar 1).



Gambar 1. Activity Diagram dari Pelaporan KtPA sebelumnya.

Berdasarkan wawancara dan diskusi dengan pihak Puskesmas disimpulkan perlunya penanganan reaksi cepat dari pelaporan sampai observasi. Penangan reaksi cepat diwujudkan dalam bentuk sistem informasi yang interaktif dan *real time*. Laporan yang diterima akan langsung ditangani oleh kader yang paling dekat dengan lokasi. Kader melakukan pertolongan pertama terhadap korban, kemudian korban akan dibawa ke Puskesmas untuk penanganan lebih lanjut (Gambar 2). Sistem informasi terdiri dari 4 komponen yaitu aplikasi untuk warga digunakan untuk melapor jika terjadi kejadian KtPA, aplikasi untuk kader puskesmas untuk mendapatkan laporan secara real time dari warga sehingga memudahkan kader untuk menangani korban secara cepat, dan aplikasi untuk puskesmas untuk mengkoordinir penanganan KtPA (Gambar 3). Aplikasi admin dan server diperlukan untuk memajemen ketiga aplikasi lainnya. Server akan ditempatkan di komputasi awan sehingga dapat diakses 24 jam dimanapun.

Pengembangan rancang bangun menggunakan model *waterfall* [13] dan menggunakan pendekatan berorientasi objek [14]. Berdasarkan analisa yang di dapat maka sistem informasi mempunyai spesifikasi sebagai berikut

1. Server berbasis awan.
2. Aplikasi terdiri dari 2 aplikasi: aplikasi berbasis Android dan Web.
3. Aplikasi Android terdiri dari admin, kader, dan masyarakat.
4. Aplikasi Web untuk admin.

Aplikasi masyarakat mempunyai fasilitas sebagai berikut:

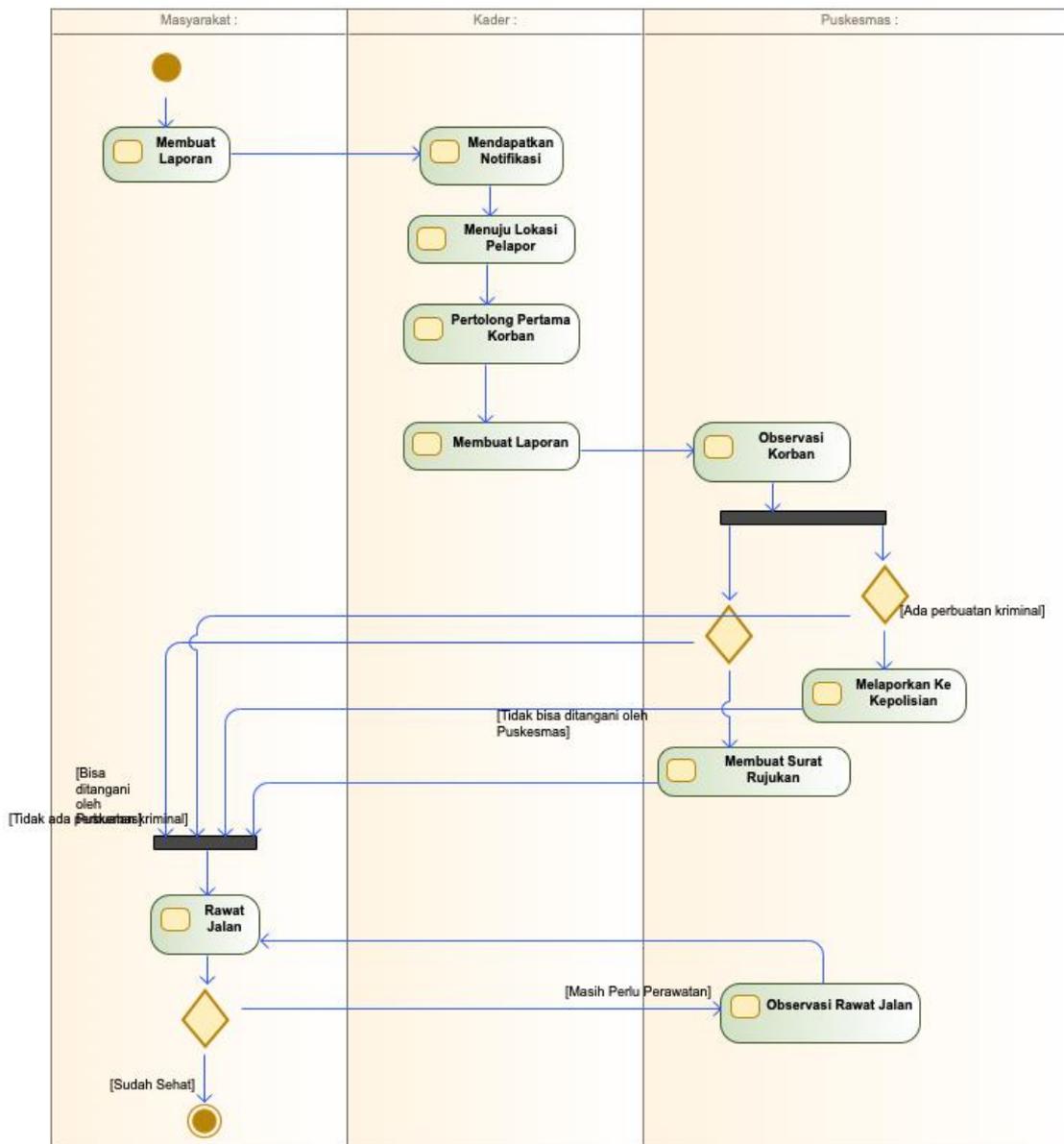
1. Melaporkan kasus.
2. Melihat status kasus.
- 3.

Aplikasi kader mempunyai fasilitas sebagai berikut:

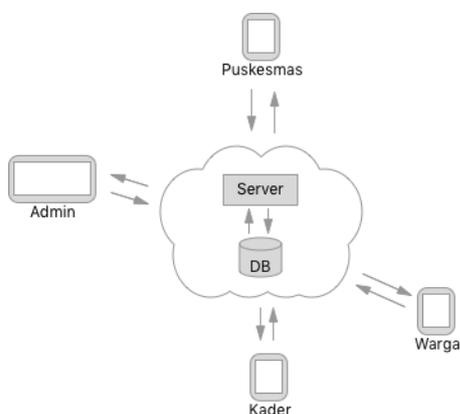
1. Daftar kasus terbaru di wilayah yang menjadi tanggung jawab kader.
2. Mengambil kasus untuk dikerjakan.
3. Daftar kasus yang sedang ditangani.
4. Mengajukan kasus yang telah selesai.
5. Daftar kasus yang sudah ditangani.
6. Mengajukan kasus yang akan dianulir.
7. Daftar kasus yang dianulir.

Aplikasi admin mempunyai fasilitas sebagai berikut:

1. Daftar semua kasus.
2. Melimpahkan kasus baru kepada kader.
3. Memindahkan kasus yang dikerjakan kepada kader lain.
4. Menyetujui penyelesaian kasus.
5. Menyetujui peng anuliran kasus.
6. Mengaktifkan/menonaktifkan kader.
7. Memindahkan wilayah tanggung jawab kader.



Gambar 2. Activity Diagram Pelaporan KtPA menggunakan aplikasi



Gambar 3. Sistem Informasi KtPA

3. Pembahasan

Berdasarkan analisa kebutuhan, rancangan sistem informasi akan menggunakan Unified Modelling Language [15][16][17][18]. Diagram yang dipakai adalah *activity diagram* untuk menggambarkan proses aktivitas pelaporan KtPA, *class diagram* untuk menggambarkan elemen dari sistem dan hubungan antar elemen, *use case diagram* untuk mendeskripsikan fitur yang ada di sistem informasi, dan *sequence diagram* untuk menjelaskan secara detail setiap fitur yang ada.

Sistem Informasi KtPA mempunyai 3 Aktor:

1. Masyarakat
2. Kader dari puskesmas
3. Admin dari puskesmas

Berikut adalah fitur pada sistem informasi yang digambarkan dalam *use case diagram* (Gambar 4):

1. Melaporkan Kasus: Masyarakat melaporkan kasus melalui sistem.
2. Melihat Status Laporan Per Masyarakat: Masyarakat memantau perkembangan kasus yang dilaporkan.
3. Daftar Laporan Baru Per Wilayah: Kader melihat laporan yang baru berada di wilayah kader bekerja.
4. Daftar Laporan Per Kader: Kader melihat laporan aktif yang sedang dikerjakan.
5. Ajukan Laporan Selesai: Kader mengajukan laporan sudah diselesaikan.
6. Ajukan Anulir Kasus: Kader mengajukan laporan dianulir karena tidak sesuai fakta di lapangan.
7. Persetujuan Anulir Laporan: Admin menyetujui atau menolak laporan yang diajukan Kader untuk dianulir.
8. Persetujuan Laporan Selesai: Admin menyetujui atau menolak laporan yang diklaim oleh Kader sudah selesai.
9. Limpahkan Laporan Ke Kader: Admin dapat memindahkan penyelesaian laporan ke Kader lainnya. Pengalihan terbatas pada kader yang bertanggung jawab di wilayah laporan terjadi.

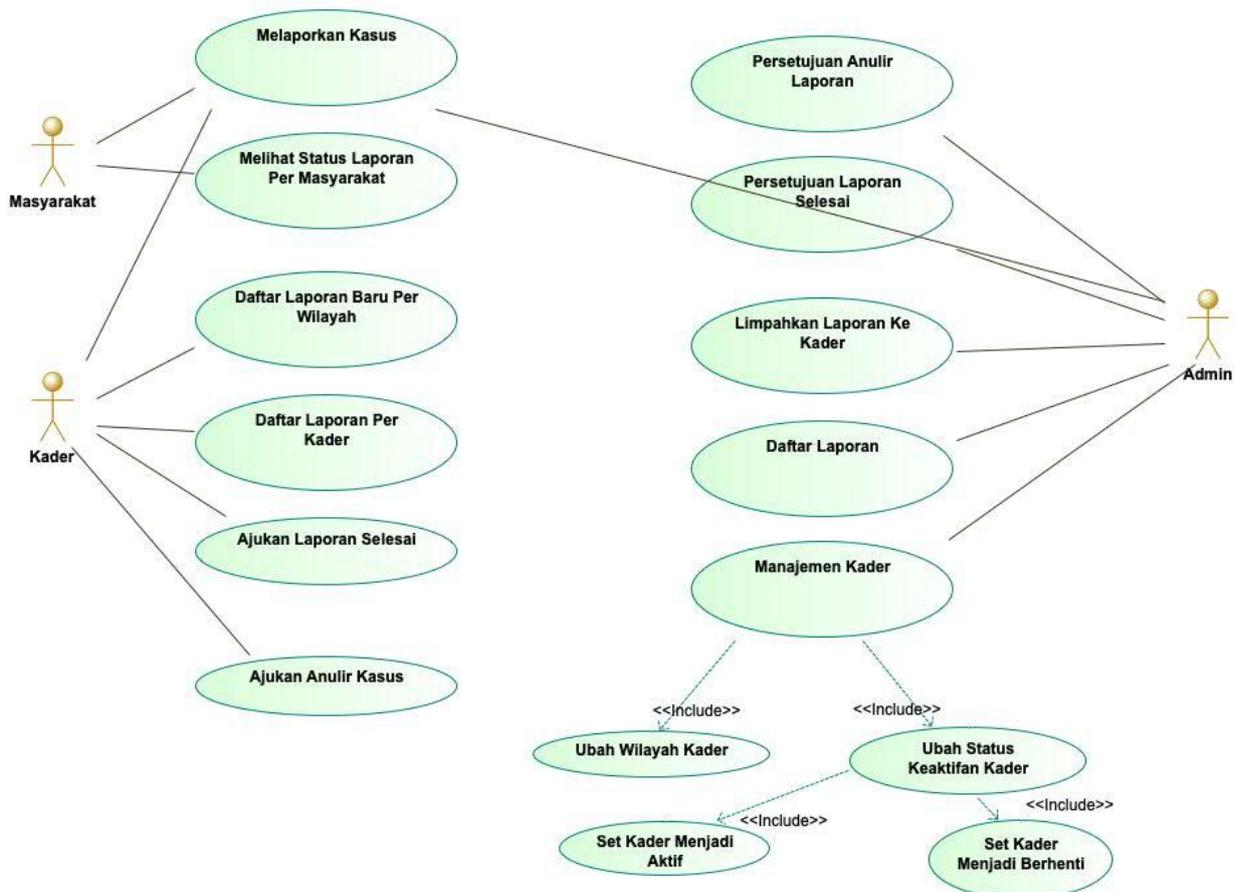
10. Daftar Laporan: Kader dapat melihat semua laporan ataupun berdasarkan status dari laporan.

11. Manajemen Kader: Admin dapat mengubah wilayah tempat Kader beroperasi dan mengubah hak akses Kader untuk menangani laporan.

Penjabaran dari setiap *use case* dijelaskan pada *sequence diagram*. *Sequence Diagram* untuk Admin dan Kader menggunakan 3 asumsi: bahwa pengguna sudah melakukan otentifikasi dan valid, setiap wilayah mempunyai kader yang bertanggung jawab, dan kriteria yang diberikan sudah terdaftar di basis data (contoh: ketika Kader meminta laporan aktif dengan id = 2, maka dipastikan laporan dengan id = 2 ada di basis data). Berikut skenario dan *sequence diagram* yang digunakan:

I. Sequence Diagram dari Masyarakat Melaporkan Kasus (Gambar 5)

Tujuan	Mendaftarkan laporan baru
Aktor	Masyarakat (Pengguna)
Kondisi Awal	Laporan belum tersimpan di basis data (Cloud Firestore)



Gambar 4. Use Case Diagram dari Sistem Informasi KtPA

Kondisi Akhir Laporan baru sudah tersimpan dan notifikasi telah diterima oleh aplikasi kader.

Alur

1. Masyarakat(pengguna) mendaftarkan laporan baru via aplikasi.
2. Sebelum mengirimkan laporan ke basis data, pengguna memvalidasi bahwa yang mengirim adalah manusia bukan bot.
3. Setelah validasi berhasil, laporan akan disimpan di basis data (Cloud Firestore).
4. Penambahan data akan men-trigger Cloud Function yang akan mengirimkan notifikasi kepada aplikasi Kader menggunakan Cloud Messaging.
5. Notifikasi diterima oleh aplikasi Kader.

Alur Alternatif

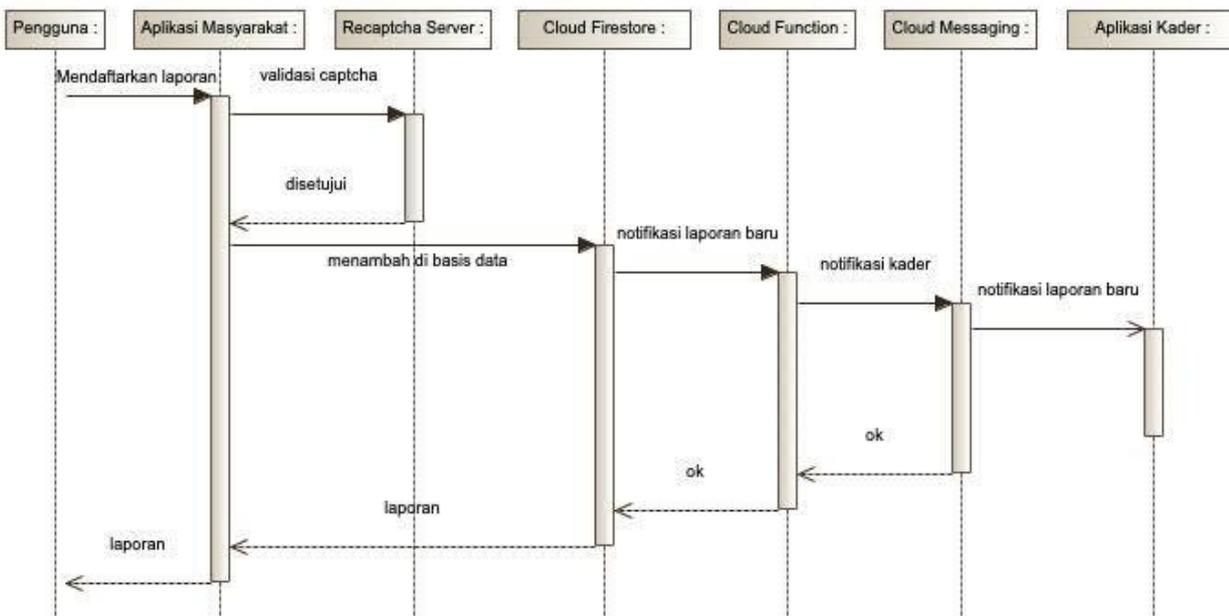
Pengguna gagal melakukan validasi. Laporan baru tidak tersimpan

II. Sequence Diagram dari Melihat Status Laporan per Masyarakat (Gambar 6)

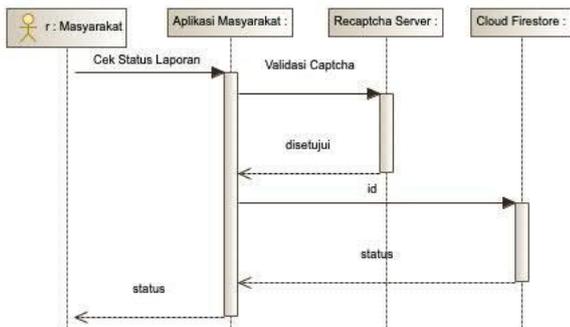
Tujuan Melihat status laporan
 Aktor Masyarakat (Pengguna)
 Kondisi Awal Laporan sudah tersimpan di basis data (Cloud Firestore)
 Kondisi Akhir Detail Laporan ditampilkan kepada pengguna.

Alur

1. Masyarakat(pengguna) memasukkan id dari laporan.
2. Sebelum menerima laporan dari basis data, pengguna memvalidasi bahwa yang menggunakan adalah manusia bukan bot.
3. Setelah validasi berhasil, laporan akan dikirimkan dari basis data (Cloud Firestore).



Gambar 5. Sequence Diagram dari Masyarakat Melaporkan kasus



Gambar 6. Sequence Diagram dari Melihat Status Laporan per Masyarakat

Alur Alternatif

1. Pengguna gagal melakukan validasi. Laporan tidak dapat ditampilkan.
2. Id tidak ditemukan di basis data. Laporan tidak dapat ditampilkan.

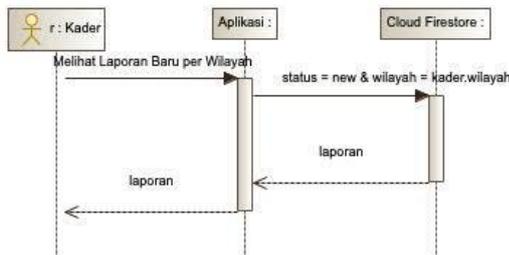
III. Sequence Diagram dari Daftar Laporan Baru Per Wilayah (Gambar 7)

Tujuan Melihat daftar laporan baru yang berada di wilayah kader beroperasi
 Aktor Kader

Kondisi Awal Daftar laporan baru belum ditampilkan di aplikasi.
 Kondisi Akhir Daftar laporan baru yang sesuai dengan wilayah kader beroperasi ditampilkan.

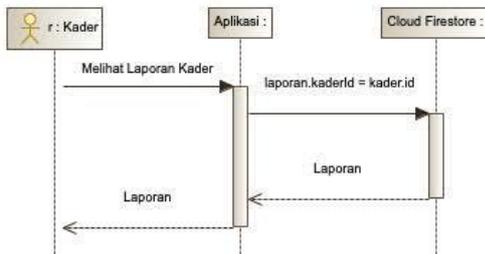
Alur

1. Kader membuka menu daftar laporan baru.
2. Aplikasi meminta data ke Cloud Firestore yang sesuai dengan kriteria yang diberikan.
3. Cloud Firestore akan mengirimkan data terkait.
4. Aplikasi akan menampilkan daftar laporan baru.

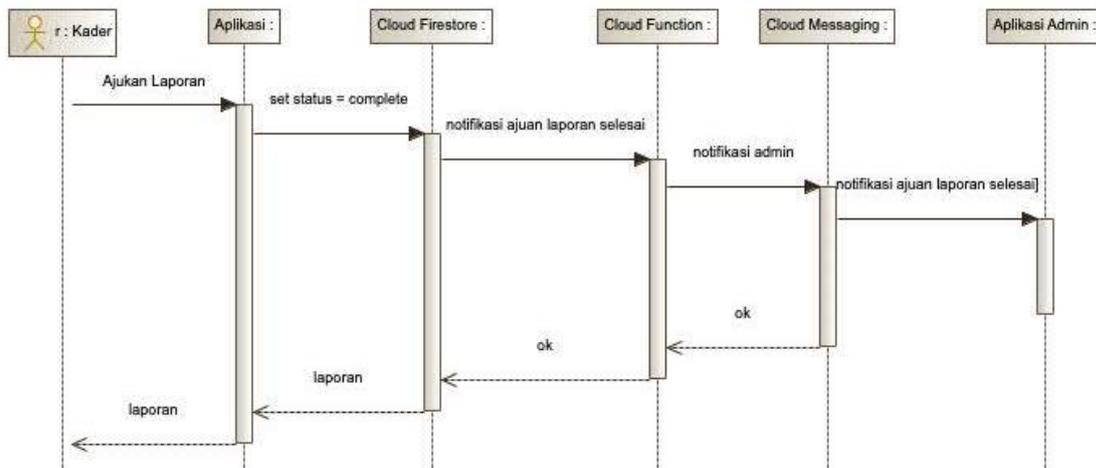


Gambar 7. Sequence Diagram dari Daftar Laporan Baru Per Wilayah

IV. Sequence Diagram dari Daftar Laporan Per Kader (Gambar 8)



Gambar 8. Sequence Diagram dari Daftar Laporan Per Kader.



Gambar 9. Sequence Diagram dari Ajukan Laporan Selesai

Tujuan Melihat daftar laporan yang sedang dikerjakan oleh Kader.

Aktor Kader

Kondisi Awal Daftar laporan aktif belum ditampilkan di aplikasi.

Kondisi Akhir Daftar laporan aktif yang menjadi tanggung jawab Kader ditampilkan.

Alur:

1. Kader membuka menu daftar laporan aktif.
2. Aplikasi meminta data ke Cloud Firestore yang sesuai dengan kriteria yang diberikan.
3. Cloud Firestore akan mengirimkan data terkait.
4. Aplikasi akan menampilkan daftar laporan aktif yang menjadi tanggung jawab Kader.

V. Sequence Diagram dari Ajukan Laporan Selesai (Gambar 9)

Tujuan Mengubah status laporan menjadi ajuan selesai (complete).

Aktor Kader

Kondisi Awal Laporan aktif di aplikasi Kader.

Kondisi Akhir Laporan tersimpan di daftar laporan ajuan selesai.

Alur

1. Kader memilih laporan yang ingin diajukan selesai di daftar laporan aktif.
2. Aplikasi memasukkan laporan aktif ke daftar laporan ajuan selesai di Cloud Firestore.
3. Perubahan data akan *trigger* Cloud Function yang akan mengirimkan notifikasi kepada aplikasi Admin menggunakan Cloud Messaging.
4. Notifikasi diterima oleh aplikasi Admin.

VI. Sequence Diagram dari Ajukan Anulir Laporan (Gambar 10)

Tujuan Mengubah status laporan menjadi ajuan dianulir (cancel).
 Aktor Kader
 Kondisi Awal Laporan aktif di aplikasi Kader.
 Kondisi Akhir Laporan tersimpan di daftar laporan ajuan anulir.

- Alur:
1. Kader memilih laporan yang ingin diajukan anulir di daftar laporan aktif.
 2. Aplikasi memasukkan laporan aktif ke daftar laporan ajuan anulir di Cloud Firebase.
 3. Perubahan data akan men-trigger Cloud Function yang akan mengirimkan notifikasi kepada aplikasi Admin menggunakan Cloud Messaging.
 4. Notifikasi diterima oleh aplikasi Admin.

Kondisi Awal Daftar laporan belum ditampilkan di aplikasi.

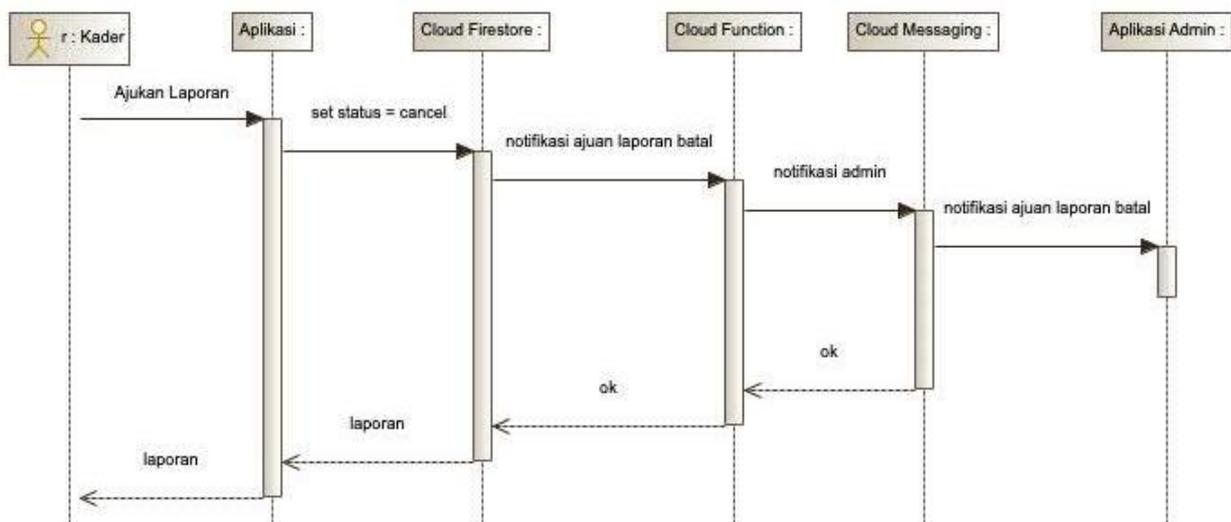
Kondisi Akhir Daftar laporan ditampilkan.

- Alur:
1. Kader membuka menu daftar laporan.
 2. Aplikasi meminta data ke Cloud Firebase.
 3. Cloud Firestore akan mengirimkan data terkait.
 4. Aplikasi akan menampilkan daftar laporan baru.

VIII. Sequence Diagram dari Persetujuan Laporan Selesai (Gambar 12)

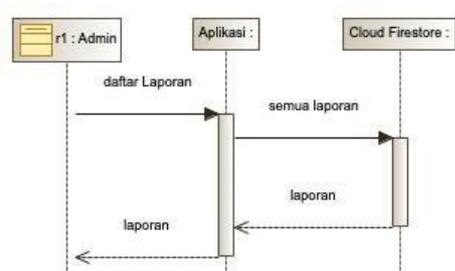
Tujuan Menyetujui laporan yang berstatus ajuan selesai (complete) menjadi selesai (done).

Aktor Admin
 Kondisi Awal Laporan status ajuan selesai di aplikasi Admin.
 Kondisi Akhir Laporan sudah selesai



Gambar 10. Sequence Diagram dari Ajukan Anulir Laporan

VII. Sequence Diagram dari Daftar Laporan (Gambar 11)



Gambar 11. Sequence Diagram dari Daftar Laporan

Tujuan Melihat daftar laporan
 Aktor Admin

Alur :

1. Admin memilih laporan yang ingin diselesaikan di daftar laporan ajuan selesai.
2. Aplikasi mengubah status laporan menjadi selesai (done) di Cloud Firebase.
3. Perubahan data akan men-trigger Cloud Function yang akan mengirimkan notifikasi kepada aplikasi Kader menggunakan Cloud Messaging.
4. Notifikasi diterima oleh aplikasi Kader.

IX. Sequence Diagram dari Persetujuan Anulir Laporan (Gambar 13)

Tujuan Menyetujui anulir laporan yang berstatus ajuan anulir (cancel) menjadi setuju anulir (cancelled).

Aktor Admin
 Kondisi Awal Laporan status ajuan anulir di aplikasi Admin.
 Kondisi Akhir Laporan sudah dianulir

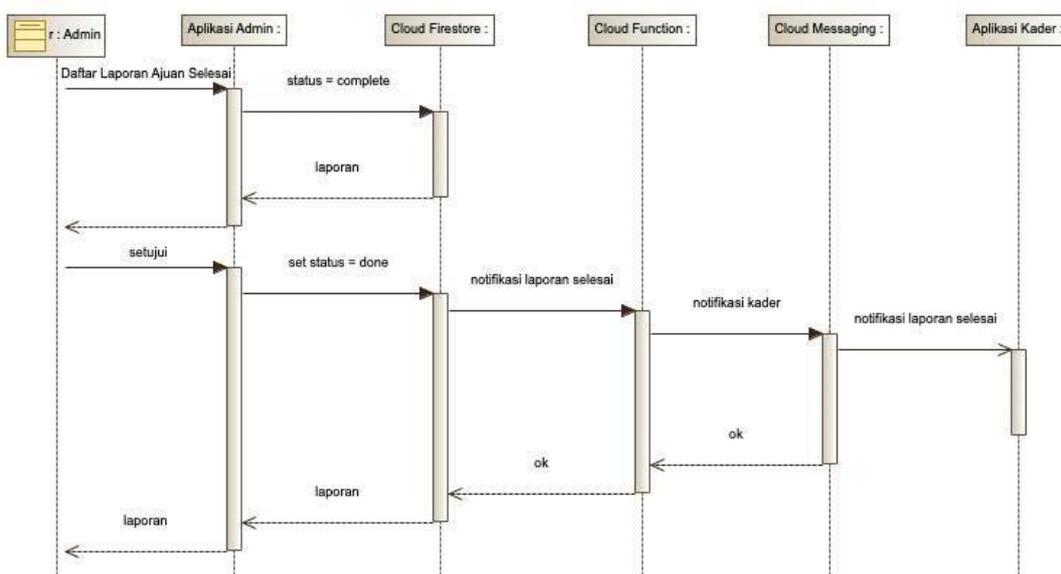
- Alur :
1. Admin memilih laporan yang ingin disetujui di daftar laporan ajuan anulir.
 2. Aplikasi mengubah status laporan menjadi setuju anulir (canceled) di Cloud Firebase.
 3. Perubahan data akan men-trigger Cloud Function yang akan mengirimkan notifikasi kepada aplikasi Kader menggunakan Cloud Messaging.
 4. Notifikasi diterima oleh aplikasi Kader.

X. Sequence Diagram dari Limpahkan Laporan Ke Kader (Gambar 14)

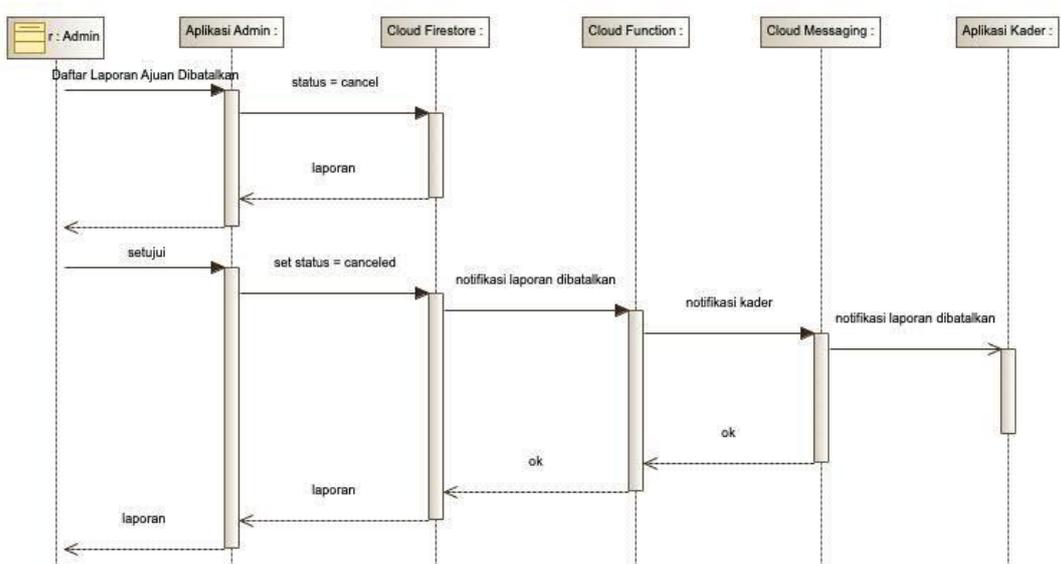
Tujuan Mengubah status Kader menjadi aktif = false.

Aktor Admin
 Kondisi Awal Status Kader aktif = true
 Kondisi Akhir Status Kader aktif = false

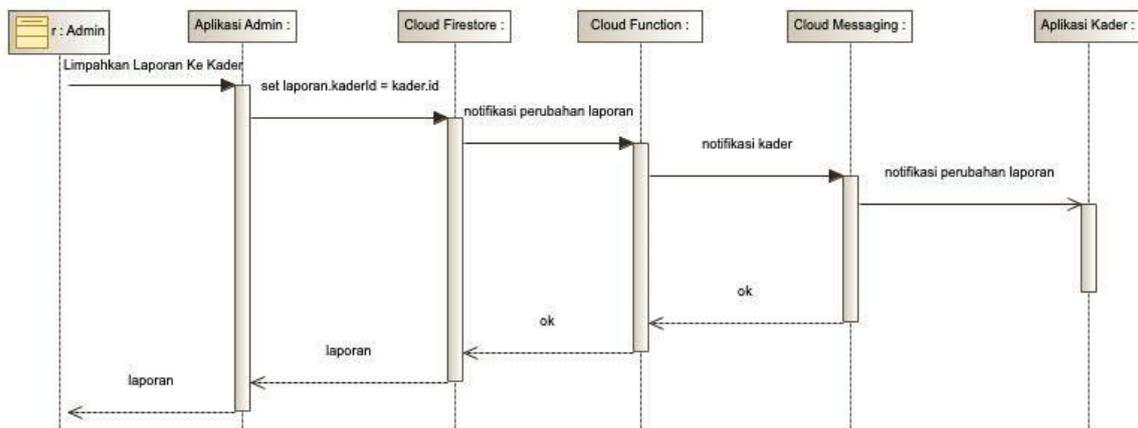
- Alur :
1. Admin memilih kader yang ingin diubah statusnya.
 2. Aplikasi mengubah status laporan menjadi aktif = false di Cloud Firebase.
 3. Perubahan data akan men-trigger Cloud Function yang akan mengirimkan notifikasi kepada aplikasi Kader menggunakan Cloud Messaging.
 4. Notifikasi diterima oleh aplikasi Kader.



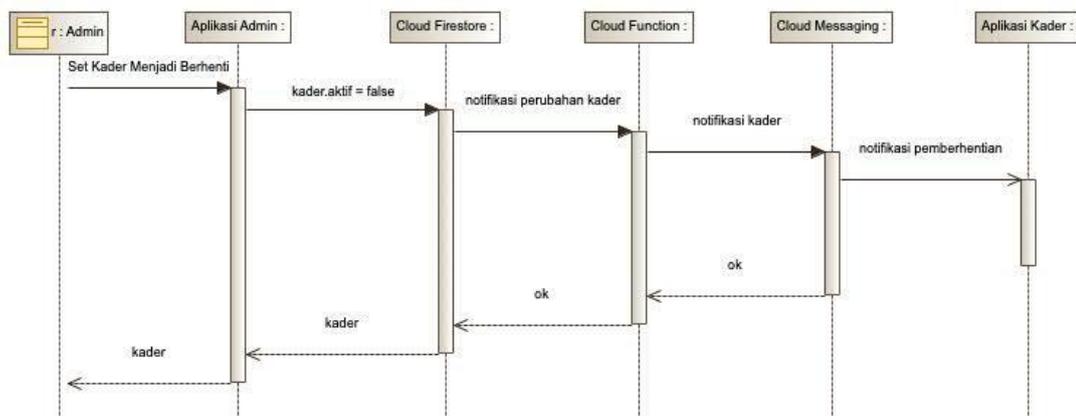
Gambar 12. Sequence Diagram dari Persetujuan Laporan Selesai



Gambar 13. Sequence Diagram dari Persetujuan Anulir Laporan



Gambar 14. Sequence Diagram dari Limpahkan Laporan Ke Kader



Gambar 15. Sequence Diagram dari Set Kader Menjadi Berhenti

XI. Sequence Diagram dari Set Kader Menjadi Berhenti (Gambar 15)

Tujuan Mengubah status Kader menjadi aktif = false.

Aktor Admin

Kondisi Awal Status Kader aktif = true

Kondisi Akhir Status Kader aktif = false

Alur :

1. Admin memilih kader yang ingin diubah statusnya.
2. Aplikasi mengubah status laporan menjadi aktif = false di Cloud Firestore.
3. Perubahan data akan men-trigger Cloud Function yang akan mengirimkan notifikasi kepada aplikasi Kader menggunakan Cloud Messaging.
4. Notifikasi diterima oleh aplikasi Kader.

Alur :

1. Admin memilih kader yang ingin diubah statusnya.
2. Aplikasi mengubah status laporan menjadi aktif = true di Cloud Firestore.
3. Perubahan data akan men-trigger Cloud Function yang akan mengirimkan notifikasi kepada aplikasi Kader menggunakan Cloud Messaging.
4. Notifikasi diterima oleh aplikasi Kader.

XII. Sequence Diagram dari Set Kader Menjadi Aktif (Gambar 16)

Tujuan Mengubah status Kader menjadi aktif = true.

Aktor Admin

Kondisi Awal Status Kader aktif = false

Kondisi Akhir Status Kader aktif = true

XIII. Sequence Diagram dari Ubah Wilayah dari Kader (Gambar 17)

Tujuan Mengubah wilayah kader beroperasi.

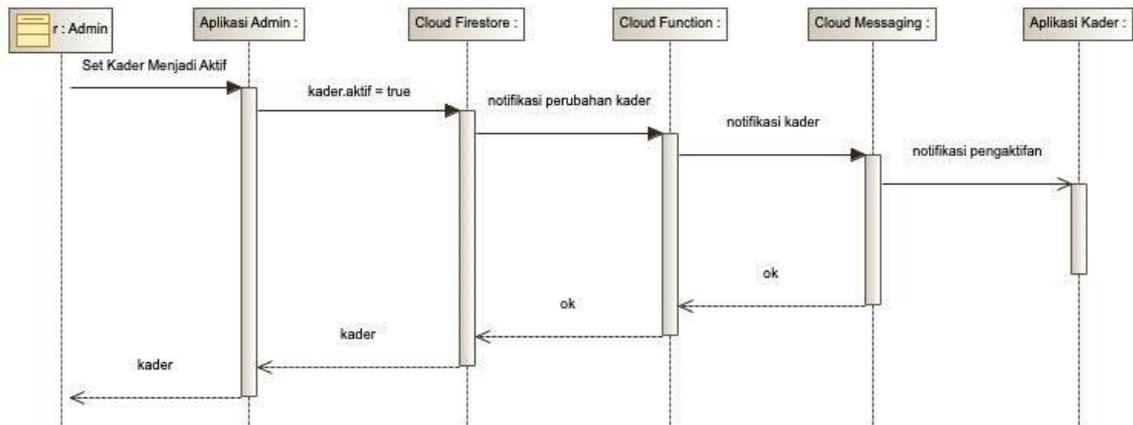
Aktor Admin

Kondisi Awal Wilayah kader sudah ditentukan atau belum di set.

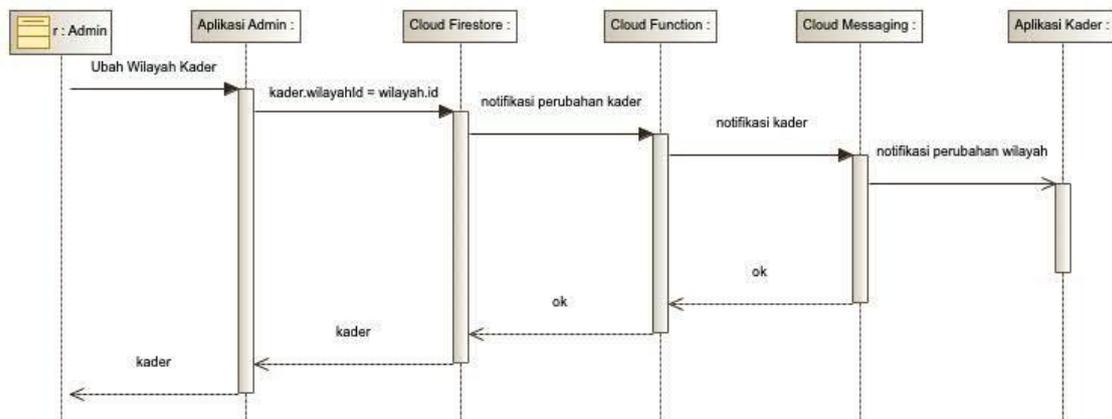
Kondisi Akhir Wilayah kader sudah diganti.

Alur :

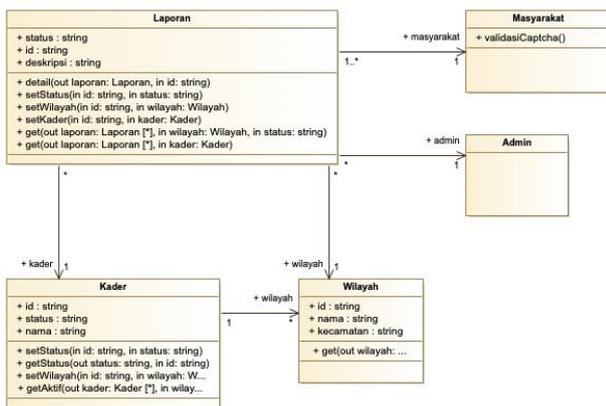
1. Admin memilih kader yang ingin diubah wilayahnya.
2. Aplikasi mengubah wilayahId di Cloud Firestore.
3. Perubahan data akan men-trigger Cloud Function yang akan mengirimkan notifikasi kepada aplikasi Kader menggunakan Cloud Messaging.
4. Notifikasi diterima oleh aplikasi Kader.



Gambar 16. Sequence Diagram dari Set Kader Menjadi Aktif



Gambar 17. Sequence Diagram dari Ubah Wilayah dari Kader



Gambar 18. Class Diagram dari Sistem Informasi KtPA

- Class Diagram (Gambar 18) terdiri dari
1. Masyarakat: yang melaporkan kasus.
 2. Laporan: info lengkap tentang kasus.
 3. Kader: identitas kader beserta wilayah beroperasi.
 4. Admin: identitas pengatur aplikasi.
 5. Wilayah: area dimana laporan terjadi dan kader yang bertanggung jawab di wilayah tersebut.

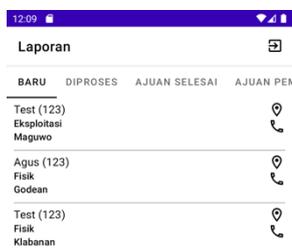
4. Hasil



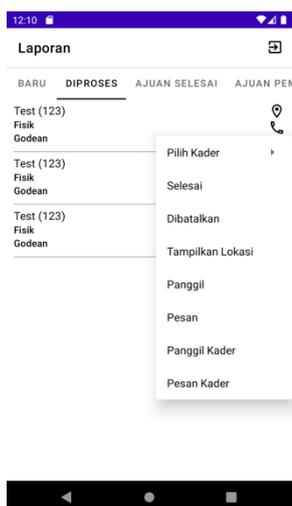
Gambar 19. Tampilan aplikasi untuk masyarakat

Bermula dari kebutuhan puskesmas yang dijabarkan ke dalam analisa kebutuhan kemudian menghasilkan desain sistem informasi laporan kekerasan terhadap perempuan dan anak. Hasil desain sudah beberapa kali mengalami revisi dengan adanya masukan dari puskesmas. Desain

sistem informasi sudah disetujui oleh pihak puskesmas dan siap untuk diimplementasikan. Berikut beberapa tampilan dari aplikasi yang sudah jadi (Gambar 19, 20, 21)



Gambar 20. Tampilan aplikasi untuk Kader



Gambar 21. Tampilan aplikasi untuk Admin

5. Kesimpulan

Desain sistem informasi yang dihasilkan sudah mendapat persetujuan dari pihak puskesmas dan siap untuk diimplementasikan. Pengembangan lebih lanjut dapat dilakukan dengan implementasi aplikasi pada gawai berbasis Android dan Web. Tidak lupa kami haturkan terima kasih kepada Kementerian Riset dan Teknologi / Badan Riset dan Inovasi Nasional atas kesempatannya melaksanakan Penelitian Dosen Pemula.

Referensi

- [1]. K. Perempuan, *Catatan Kekerasan Terhadap Perempuan Tahun 2018*. Jakarta: Komnas Perempuan, 2019.
- [2]. J. Ahyar and S. Safrida, "Optimalisasi Pelayanan Kepolisian dalam Menangani Tindak Pidana Kekerasan Terhadap Perempuan dan Anak," *HUMANIS J. Ilmu Adm. Negara*, vol. 6, no. 2, pp. 69–89, 2020, doi: 10.52137/humanis.v6i1.20.
- [3]. Hermawati, "Rancang Bangun Sistem Pelaporan Tindak Kekerasan bagi Perempuan dan Anak pada Pusat Pelayanan Terpadu Pemberdayaan Perempuan dan Anak Kota Makassar Berbasis Android," Universitas Islam Negeri Alauddin Makassar, 2018.
- [4]. T. M. Andini, T. Sulistyowati, A. Alifatin, R. P. Sudibyo, J. Raya, and M. Email, "Identifikasi Kejadian Kekerasan pada Anak di Kota Malang Identification of Violence in Children in Malang City," vol. 2, no. 1, pp. 13–28, 2019.
- [5]. A. N. I. Surtinah, P. Studi, M. Ilmu, P. Pascasarjana, and U. M. Surakarta, "Studi Tentang Efektifitas Hukum Penanganan Kasus Kekerasan Terhadap Perempuan Di Pelayanan Terpadu Perempuan dan Anak Surakarta (PTPAS)," 2017.
- [6]. H. Siregar, "Bentuk-Bentuk Kekerasan yang Dialami Perempuan Warga Komplek Dinas Peternakan Provinsi Sumatera Utara Form Of Violence Experienced by Women Citizens Complex Dinas Peternakan North Sumatera Province," *J. Pemberdaya. Komunitas*, vol. 14, no. 1, pp. 10–19, 2016.
- [7]. F. N. Adi, "Menteri Puji Aplikasi Laker DP3A," *Suara Merdeka News*, 2018.
- [8]. D. R. Ardinta, "Analisis Efektivitas Penanganan Pengaduan Tindak Kekerasan Pada Perempuan dan Anak Di Kota Semarang Tahun 2017-2019 Melalui Fitur Geber Pandanaran Pada Aplikasi Laporan Hendi," Universitas Bakrie, 2021.
- [9]. A. Q. Munir, Z. B. S. Nuswantoro, and Yusriadi, "Pendampingan Penggunaan Aplikasi Pengelolaan Kegiatan Satuan Tugas Perlindungan Perempuan dan Anak," *Wikrama Parahita J. Pengabd. Masy.*, vol. 5, no. 1, 2021.
- [10]. L. Moroney, *The Definitive Guide to Firebase*. Seattle: Apress, 2017.
- [11]. M. Wickham, *Practical Android*. 2018.
- [12]. J. F. DiMarzio, *Beginning Android Programming with Android Studio*. 2017.
- [13]. I. Sommerville, *Software Engineering: Global Edition, Tenth Edition*. 2016.
- [14]. A. Dennis, B. H. Wixom, and D. Tegarden, *Systems Analysis and Design : An Object-Oriented Approach with UML*, 5th ed. Wiley Publishing, Inc, 2015.
- [15]. J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, 2nd ed. Pearson Education, Inc., 2005.
- [16]. B. Rumpe, *Modeling with UML - Language, concept, methods*. 2016.
- [17]. B. Unhelkar, *Software Engineering with UML*. CRC Press, 2018.
- [18]. M. Seidl, M. Scholz, C. Huemer, and G. Kappel, *UML @ Classroom: An Introduction to Object-Oriented Modeling*. Springer, 2015