

Aplikasi Pencarian Donor Darah Menggunakan Metode Personal Extreme Programming Berbasis Android

Blood Donor Search Application Using Personal Extreme Programming Method Based on Android

Muhammad Fajri Riki^{1*}
Irman Effendy²

^{1,2}Sistem Informasi, Universitas Bina Darma, Indonesia
¹181410190@student.binadarma.ac.id, ²irman.effendy@binadarma.ac.id

*Penulis Korespondensi:

Muhammad Fajri Riki
181410190@student.binadarma.ac.id

Riwayat Artikel:

Diterima : 2 Agustus 2022
Direview : 23 Agustus 2022
Disetujui : 26 September 2022
Terbit : 6 Desember 2022

Abstrak

Pada tahun 2019 lalu, dunia dikejutkan dengan adanya wabah COVID-19. COVID-19 menyebabkan berkurangnya stok darah dan pendonor karena khawatir terpapar COVID-19 saat melakukan donor. Pencarian donor darah oleh pasien cenderung mengalami kesulitan karena sulit ditemukannya pendonor. Masih banyak pasien yang kesulitan mendapatkan donor darah karena stok darah yang dibutuhkan tidak selalu ada. Banyak pasien yang menggunakan media sosial untuk mencari donor darah tetapi kurang efektif, informasi pendonor belum jelas, dan tidak selalu tepat sasaran. Berdasarkan masalah tersebut, maka solusinya adalah membuat sebuah aplikasi berbasis *android* sebagai *platform* yang dapat membantu pencarian donor darah dan menghubungkan pasien dan pendonor. Aplikasi ini dibangun dengan *framework Flutter, Firebase dan Google Cloud*. Metode yang digunakan dalam pembangunan aplikasi ini adalah *Personal Extreme Programming (PXP)* dengan pengujian *Black Box* didapatkan hasil yaitu aplikasi yang dibuat dapat bekerja dengan baik sebagaimana fungsinya. Hasil dari penelitian ini adalah aplikasi pencari donor darah berbasis android yang dapat mempermudah mencari pendonor.

Kata Kunci: Donor Darah, Android, Personal Extreme Programming, Black Box Testing

Abstract

In 2019, the world was shocked by the COVID-19 outbreak. COVID-19 causes a reduction in blood stocks and donors for fear of being exposed to COVID-19 when donating. The search for blood donors by patients tends to be difficult because it is difficult to find donors. There are still many patients who have difficulty getting blood donors because the blood stock needed is not always available. Many patients use social media to find blood donors but it is less effective, donor information is unclear, and not always on target. Based on these problems, the solution is to create an android-based application as a platform that can help search for blood donors and connect patients and donors. This app is built with Flutter, Firebase and Google Cloud frameworks. The method used in the development of this application is Personal Extreme Programming (PXP) with Black Box testing, the results are that the application is made to work properly as it functions. The results of this study are an android-based blood donor finder application that can make it easier to find donors.

Keywords: Blood Donor, Android, Personal Extreme Programming, Black Box Testing

1. Pendahuluan

Seluruh dunia dikejutkan oleh kemunculan wabah yang belum pernah terjadi sebelumnya pada tahun 2019 lalu di Wuhan, Cina. Sebuah virus corona baru 2019 novel coronavirus (2019-nCoV). Wabah infeksi tersebut disebut *Coronavirus Diseases 2019 (COVID-19)* [1].

COVID-19 berdampak buruk pada berkurangnya donor dan suplai darah. Di sisi lain, tingginya permintaan donor di masa pandemi menyebabkan tidak seimbangnya jumlah permintaan darah yang lebih besar dari suplai darah. Hal ini dikarenakan masyarakat yang takut tertular COVID-19 saat mendonorkan darahnya. Hingga saat ini, belum ada bukti penyebaran infeksi virus COVID-19 melalui donor darah, sehingga tidak perlu takut tertular COVID-19 ketika mendonorkan darah [2].

Menurut [3] donor darah merupakan proses pengambilan darah dari seorang pendonor dan menyimpannya di bank darah atau langsung digunakan dalam transfusi darah. Transfusi darah adalah proses dimana darah disalurkan dari pendonor ke pasien atau penerima donor. Kegiatan donor darah adalah kegiatan menyalurkan darah dari pendonor kepada yang membutuhkan, dan proses ini dilakukan oleh pihak yang berwenang seperti unit Transfusi Darah di bawah naungan Palang Merah Indonesia atau rumah sakit. Pendonoran darah dapat dilakukan jika pendonor sudah memenuhi syarat-syarat dan ketentuan mendonor. Salah satu syarat untuk mendonor adalah tipe golongan darah. Selain penggolongan darah ABO, golongan darah juga dibagi menjadi Rhesus(Rh).

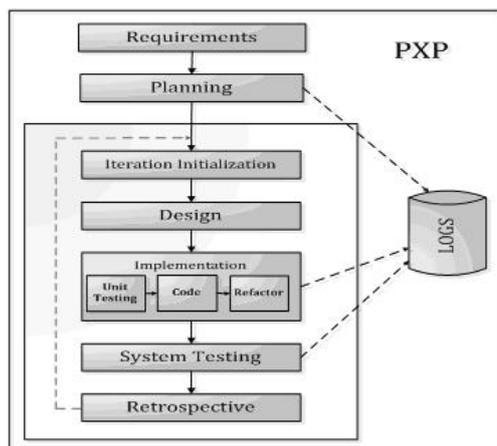
Pemilik golongan darah dengan Rh negatif juga membuat pencarian donor menjadi sulit. Secara keseluruhan, kurang lebih 94% penduduk dunia memiliki Rh positif dan hanya kurang lebih 6% saja penduduk dunia yang memiliki Rh negatif. Ras yang paling banyak memiliki Rh negatif adalah ras caucasian, sementara yang paling sedikit memiliki Rh negatif adalah ras asia (termasuk di dalamnya penduduk Indonesia) [4].

Pencarian dapat ditingkatkan dengan adanya sebuah aplikasi mobile yang dapat mempermudah mencari dan mempertemukan pencari donor dan calon pendonor. Aplikasi mobile adalah perangkat lunak yang dikembangkan untuk perangkat genggam seperti smartphone dan tablet yang memberikan akses cepat ke informasi serta dapat diakses dimana saja dan kapan saja selama memiliki jaringan internet [5]. Berdasarkan survei APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) pada tahun 2020, Indonesia memiliki 196,71 juta jiwa pengguna internet. 95,4% dari jumlah tersebut terhubung melalui smartphone [6].

Pada penelitian sebelumnya, yaitu 'Sistem Informasi Penjualan dan Pelelangan Berbasis Web pada Ricardo Corner MLG Menggunakan Metode Personal Extreme Programming (PXP)' sistem yang dikembangkan dengan metode PXP terbukti memberikan kemudahan menghasilkan perangkat lunak dalam memperkirakan fungsional sistem yang dikembangkan [7]. Pada penelitian lain oleh Suharto dan Mardiana dalam 'Aplikasi Eresha Mobile Berbasis Android dengan Metode Personal Extreme Programming di Era Industri 4.0' dengan menggunakan metode PXP sistem yang dibangun berjalan sesuai kebutuhan dengan hasil yang memuaskan [8]. Pada penelitian ini, metode yang digunakan untuk pembangunan aplikasi adalah metode Personal Extreme Programming (PXP). Metode PXP digunakan karena dapat mengatasi masalah dalam pengembangan sistem skala kecil yang membutuhkan kecepatan dalam pembuatan, memperkecil resiko dan biaya sistem. Aplikasi dapat mengambil dan menampilkan titik koordinat pencari donor, pendonor dan lokasi donor serta menampilkan acara dan stok darah.

2. Metode Penelitian

Metode pengembangan yang digunakan pada penelitian ini adalah *Personal Extreme Programming* (PXP). Adapun tahapan pada penelitian ini diilustrasikan pada gambar 1. Metode PXP adalah metode pembangunan perangkat lunak yang dirancang untuk pemrogram tunggal. Proses pengembangan PXP bersifat iteratif dan menerapkan praktiknya yang memungkinkan pemrogram menjadi lebih fleksibel dan responsif terhadap perubahan [9]. Metode PXP adalah pengembangan dari metode *Extreme Programming* (XP) dengan menggunakan praktik metode XP. Praktik XP dimodifikasi sehingga dapat masuk ke dalam situasi pemrogram tunggal dan proses pengembangan perangkat lunak dibuat [10].



Gambar 1. Tahapan Personal Extreme Programming

Requirements (Kebutuhan)

Requirements atau keperluan adalah tahap identifikasi pengguna sistem dan pengumpulan data, setelah itu dilanjutkan dengan pembentukan arsitektur program.

Planning (Perencanaan)

Planning atau perencanaan adalah tahap menentukan fitur-fitur yang diimplementasikan ke dalam sistem.

Iteration Initialization (Inisialisasi Iterasi)

Iteration initialization adalah fungsionalitas sistem yang telah ditentukan, digambarkan kedalam bentuk *Unified Modelling Language*(UML). UML adalah salah satu dari bahasa yang digunakan untuk mendefinisikan *requirements*, menggambarkan arsitektur pada *object oriented programming*, serta membuat analisis dan desain. UML dapat merepresentasikan rancangan sistem yang hendak diimplementasikan sehingga menjadi sebuah sistem yang siap pakai [11]. *Use case diagram*, *activity diagram*, dan *class diagram* adalah diagram yang digunakan dalam pembangunan aplikasi. *Class diagram* akan digunakan pada tahap *design*.

Design (Desain)

Design adalah tahap mulainya sistem didesain, mulai dari desain antarmuka pengguna dan desain database. Desain yang dibuat adalah berdasarkan *requirements* saat ini.

Implementation (Implementasi)

Implementation adalah tahap dilakukannya pengkodean sistem menggunakan framework *Flutter*. *Flutter* adalah kerangka kerja lintas platform yang ditujukan untuk mengembangkan aplikasi seluler yang kuat. *Flutter* dirilis oleh *Google* secara publik pada tahun 2016. *Flutter* menggunakan mesin rendering berperforma tinggi untuk merender setiap komponen tampilan. Fitur *Hot Reload* pada *Flutter* adalah salah satu yang mendorong pengembangan sistem [12]. Ketika pengkodean selesai, aplikasi akan diuji dengan *unit testing*. Akan dilakukan koreksi ulang (*refactor*) jika ada kesalahan, bila tidak ada kesalahan maka dilanjutkan ke tahap selanjutnya.

System Testing (Pengujian Sistem)

System testing adalah tahap pengujian apakah suatu sistem berfungsi penuh atau masih cacat. Pengujian dilakukan menggunakan *Black Box Testing*. *Black Box Testing* adalah metode yang pengujiannya berdasarkan kebutuhan dan spesifikasi tanpa perlu memeriksa kode program. Pengujian yang dilakukan berdasarkan sudut pandang pengguna dan cukup meninjau *input* dan *output*nya [13].

Retrospective (Retrospeksi)

Retrospective adalah sesi pengambilan kesimpulan terhadap sistem, bila terdapat kesalahan maka akan dilakukan revisi dimulai dari *iteration initialization*.

3. Hasil dan Pembahasan

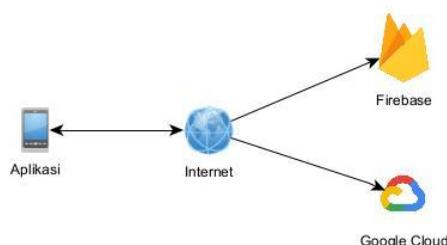
Berikut adalah hasil berdasarkan metode penelitian di atas.

Requirements (Kebutuhan)

Pada sesi *requirements*, pengumpulan data dan identifikasi pengguna sistem dilakukan dengan cara observasi dan wawancara bersama Ketua Unit Donor Darah Palang Merah Indonesia Kota Palembang. Berdasarkan hasil diskusi, sistem yang dibuat berupa aplikasi *mobile* berbasis *android*. Sistem operasi *android* diakuisisi oleh Google pada tahun 2005 yang pada awalnya dikembangkan oleh *Android Inc.* *Android* adalah sistem operasi yang dikembangkan untuk perangkat seluler berbasis *Linux*. Aplikasi *mobile* berbasis *android* dapat memberikan efisiensi dan efektifitas kepada pengguna [14]. Terdapat tiga pengguna dalam aplikasi ini, yaitu pencari donor, pendonor, dan admin.

Planning (Perencanaan)

Secara keseluruhan pengguna selain admin dapat melihat acara donor darah dan melihat stok darah. Pencari donor dapat mencari pendonor dan mengkonfirmasi ketika donor darah telah selesai. Pendonor dapat menanggapi permintaan donor dan membatalkan donor apabila saat pemeriksaan tidak dapat melakukan donor. Admin dapat mengelola acara donor darah dan stok darah.



Gambar 2. Arsitektur Aplikasi

Gambar 2 adalah arsitektur aplikasi yang akan dibangun. Sistem akan terhubung ke internet untuk berkomunikasi dengan server. Firebase database API dapat digunakan untuk transaksi data seperti *insert*, *update*, dan *delete*. *Firebase* adalah layanan yang dikembangkan oleh *Google* untuk membuat aplikasi *mobile* dan *web*. *Firebase database* menyediakan solusi untuk pengembang yang dirancang untuk mempercepat integrasi fitur berbasis *cloud*. *Firebase database* digunakan sebagai media *database* aplikasi [15]. Dengan memanfaatkan *firebase cloud messaging*, aplikasi dapat mengirimkan *push notification* ke pengguna secara *realtime*. Penggunaan *firebase cloud messaging* akan membuat kinerja aplikasi menjadi lebih efektif [16]. *Google Cloud* digunakan untuk implementasi *google maps API* dan *geocoding*. *Google maps API* memungkinkan penyesuaian titik lokasi asal ke destinasi dan perkiraan waktu berdasarkan rute yang diambil [17]. Titik koordinat yang didapat berupa *latitude* dan *longitude*. Koordinat ini akan diubah menjadi alamat menggunakan *geocoding*.

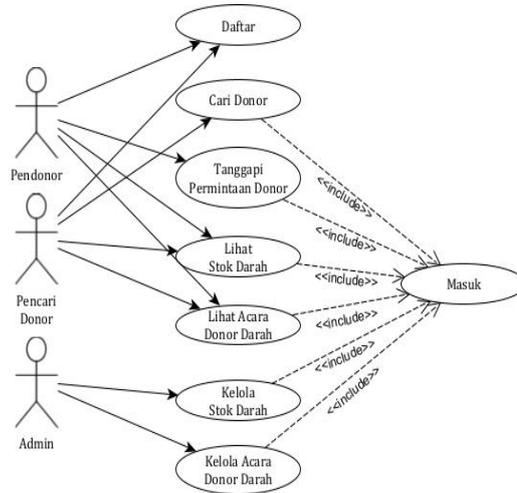
Iteration Initialization (Inisialisasi Iterasi)

Iteration initialization adalah fungsionalitas yang sudah ditentukan, lalu digambarkan dalam bentuk *Unified Modelling Language (UML)*.

Use Case Diagram

Gambar 3 adalah gambar use case dari aplikasi yang dibuat. Pengguna harus login terlebih dahulu sebelum menggunakan aplikasi. Pendonor dapat menanggapi permintaan donor saat pencari donor

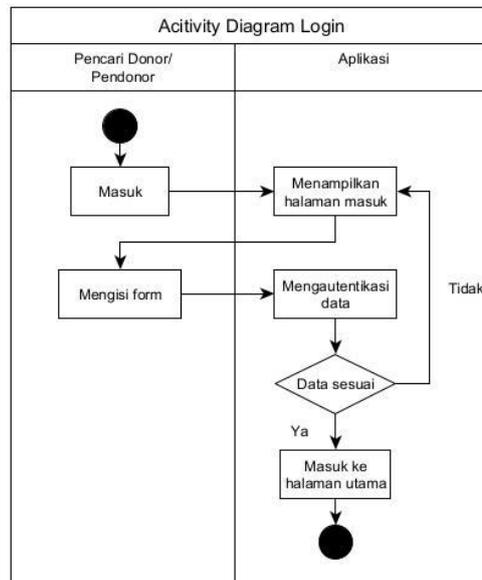
melakukan pencarian donor. Pendoron dan pencari donor dapat melihat menu acara dan stok darah. Admin dapat mengelola acara dan stok darah.



Gambar 3. Use Case Diagram

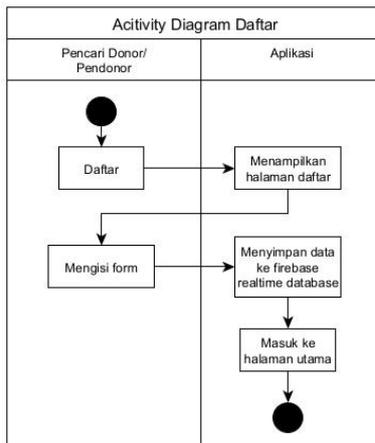
Activity Diagram

Gambar 4 adalah gambar activity diagram login. Pengguna harus mengisi form dan ketika aplikasi berhasil mengautentikasi data pengguna menggunakan firebase, maka akan masuk ke halaman utama.



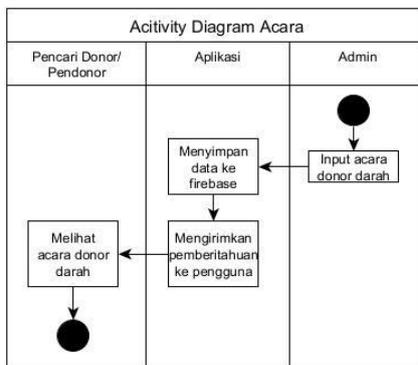
Gambar 4. Activity Diagram Login

Gambar 5 merupakan activity diagram daftar. Pengguna yang belum mempunyai akun harus mendaftarkan terlebih dahulu. Data pengguna akan tersimpan di *firebase realtime database*. Setelah terdaftar pengguna akan diarahkan ke halaman utama.



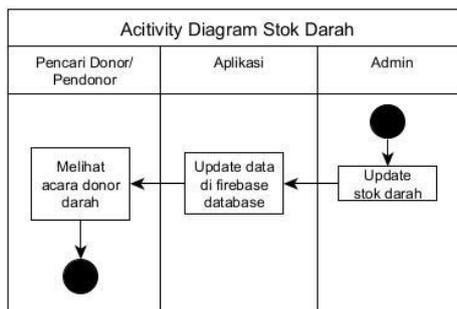
Gambar 5. Activity Diagram Daftar

Gambar 6 adalah *activity diagram* acara. Admin dapat menginput acara dan pengguna akan mendapatkan pemberitahuan serta melihat acara.



Gambar 6. Activity Diagram Acara

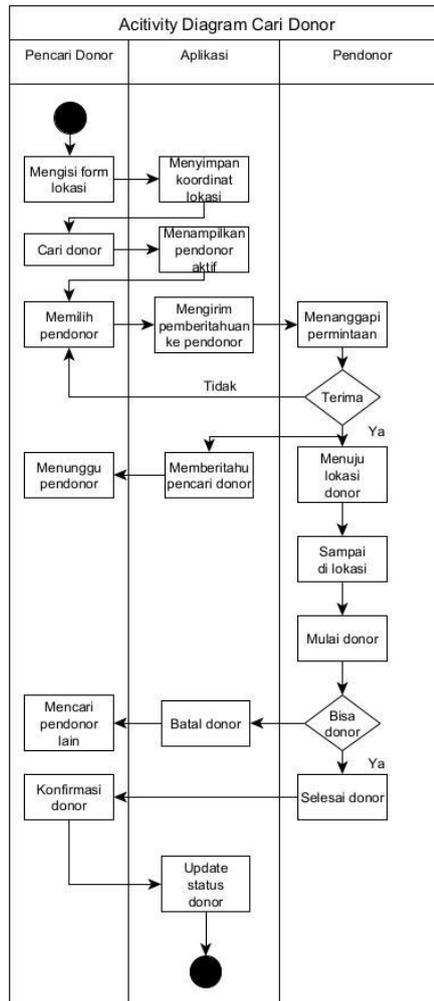
Gambar 7 adalah *activity diagram* stok darah. Admin dapat mengupdate stok darah dan pengguna dapat melihat stok darah yang tersedia.



Gambar 7. Activity Diagram Stok Darah

Gambar 8 merupakan *activity diagram* cari donor. Pencari donor harus memilih lokasi donor kemudian setelah mengklik cari donor aplikasi akan menampilkan daftar pendonor aktif. Pencari donor memilih donor dengan golongan darah yang diinginkan. Aplikasi akan mengirimkan pemberitahuan kepada pendonor. Pendonor yang mendapat permintaan menanggapi dengan menerima atau menolak. Jika pendonor menerima permintaan maka aplikasi akan menampilkan lokasi donor. Pendonor yang sampai di lokasi bisa memulai donor. Jika pendonor dapat melakukan

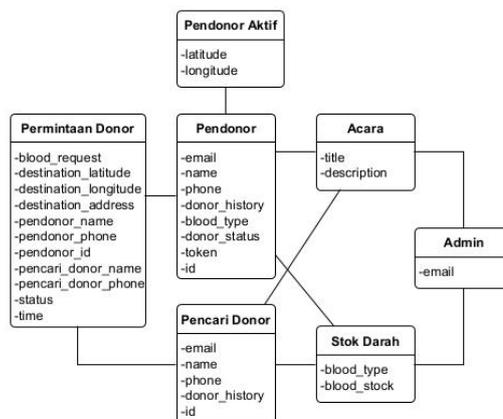
donor maka dilanjutkan sampai selesai, jika tidak maka pencari donor mencari pendonor lain. Donor yang telah selesai dikonfirmasi oleh pencari donor.



Gambar 8. Activity Diagram Cari Donor

Design (Desain)

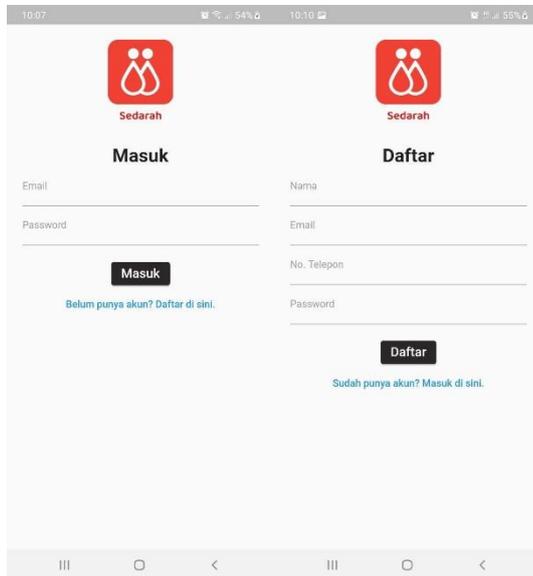
Design adalah tahap sistem mulai didesain, mulai dari desain antarmuka pengguna dan rancangan database. Gambar 9 adalah class diagram dari firebase realtime database.



Gambar 9. Class Diagram

Implementation (Implementasi)

Implementation merupakan tahapan dilakukannya pengkodean sistem menggunakan *framework Flutter*. Ketika pengkodean selesai, aplikasi akan diuji dengan *unit testing*. Akan dilakukan koreksi ulang (*refactor*) jika ada kesalahan, bila tidak ada kesalahan maka lanjut ke tahap berikutnya. Hasil implementasi ditunjukkan pada gambar dalam bentuk tangkapan layar aplikasi.

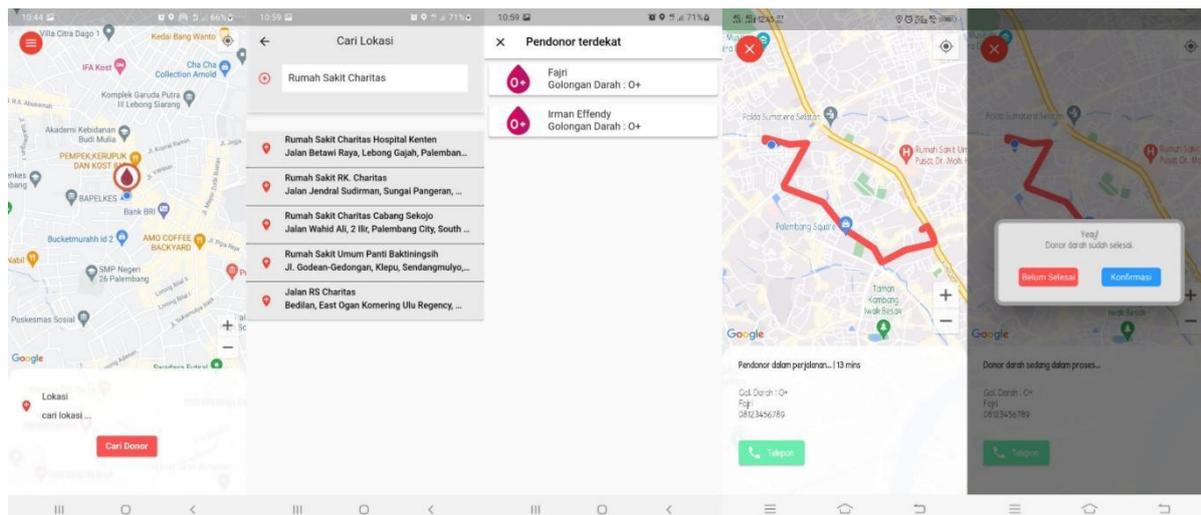


Gambar 10. Tampilan Login dan Daftar

Identifler	Providers	Created ↓	Signed In	User UID
nawaitesibor2000@gmail...	📧	Aug 30, 2022	Aug 30, 2022	W9s3c1HsF8T0aHLPKMSJfru4eTg1
imaneffendy@gmail.com	📧	Jul 16, 2022	Jul 16, 2022	PGXWlOm3chbNdKWy34WJXmp...
mfajririki@gmail.com	📧	Jul 16, 2022	Jul 16, 2022	L7EqinBztIQSmFF4nn5obccu9tKS2
diki@gmail.com	📧	Jul 6, 2022	Jul 6, 2022	DVVBjCPFNIPueZx4dLKZHLaXJts1
robi@gmail.com	📧	Jun 2, 2022	Jul 17, 2022	9z2Jplw1SrWofGGRLvYwwUNHu...
riki@gmail.com	📧	Jun 2, 2022	Jun 2, 2022	Y7QsJ6RdahEK5CPmmZrvOM8Pxc...
fajri@gmail.com	📧	Jun 1, 2022	Sep 14, 2022	lFDfEEmnKshduo2TrKmmUcp3xIz2

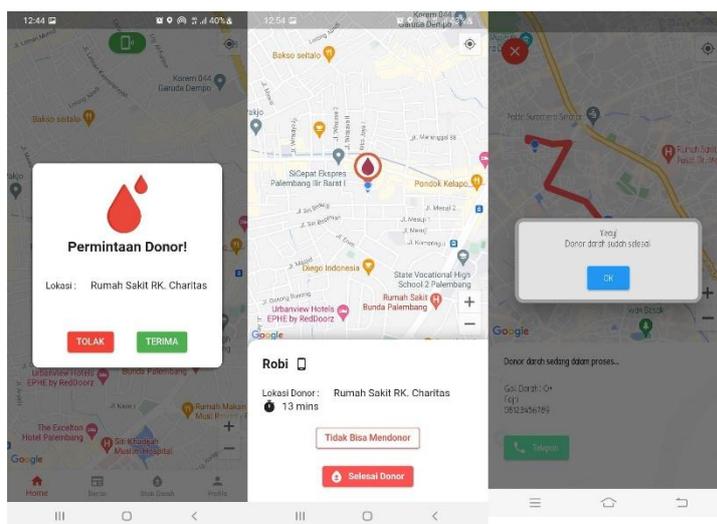
Gambar 11. Data pengguna di firebase

Gambar 10. adalah tampilan *Login* dan *Daftar*. Setelah pengguna mendaftar, aplikasi akan menyimpan data pengguna ke dalam *firebase realtime database* seperti yang ditunjukkan pada Gambar 11. Aplikasi akan mengautentikasi pengguna saat *login* menggunakan *firebase authentication*.



Gambar 12. Tampilan Pencari Donor

Gambar 12. adalah tampilan pencari donor. Pencari donor dapat mengklik cari lokasi, lalu memilih lokasi untuk melakukan pendonoran. Setelah pencari donor mengklik tombol Cari Donor maka akan muncul daftar pendonor aktif. Pencari donor memilih pendonor lalu akan muncul garis polyline pada peta yang menandakan jarak pendonor dengan lokasi donor. Pencari donor dapat mengkonfirmasi donor apabila donor sudah selesai.



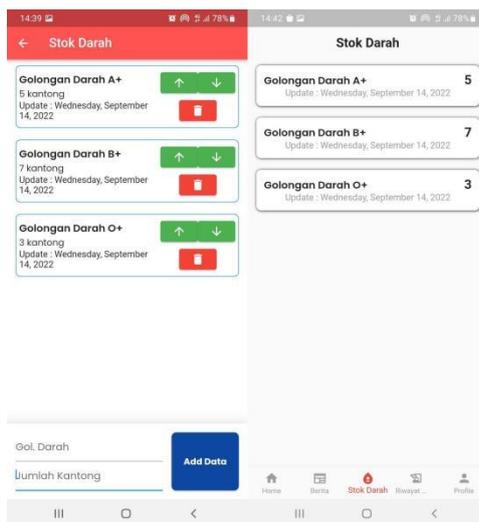
Gambar 13. Tampilan Pendonor

Gambar 13. adalah tampilan pendonor. Pendonor akan mendapatkan notifikasi jika menerima permintaan donor. Pendonor dapat menerima atau menolak permintaan. Jika pendonor menerima permintaan maka akan muncul garis polyline yang menandakan jarak pendonor dan lokasi donor. Pendonor dapat mengupdate proses donor melalui aplikasi. Donor yang selesai dilakukan dikonfirmasi oleh pencari donor.



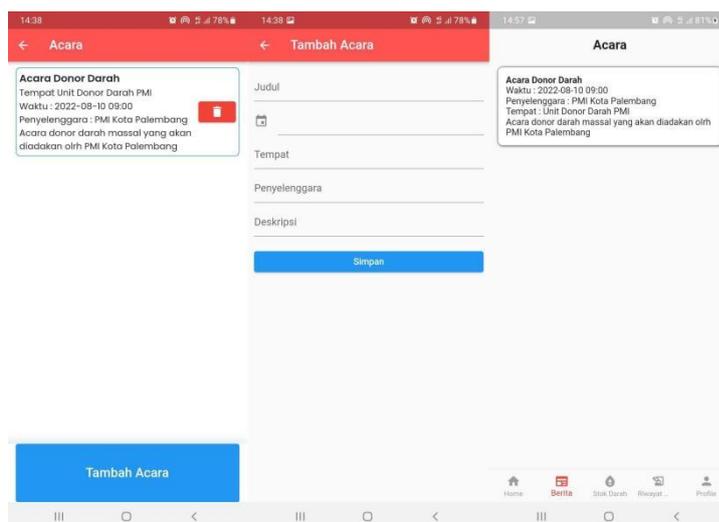
Gambar 14. Riwayat Donor

Gambar 14. adalah tampilan riwayat donor. Aplikasi akan menyimpan donor yang telah selesai dan dikonfirmasi oleh pencari donor. Riwayat donor menampilkan nama pencari donor, golongan darah, lokasi donor dan waktu mendonor.



Gambar 15. Tampilan Stok Darah

Gambar 14. adalah tampilan donor darah. Admin dapat melakukan input dan edit stok darah melalui aplikasi admin. Data yang ada juga akan terupdate secara realtime di aplikasi pengguna. Stok darah menampilkan jenis golongan darah, jumlah stok dan waktu update terakhir.



Gambar 16. Tampilan Acara

Gambar 16 adalah tampilan acara. Admin dapat menambahkan acara baru melalui aplikasi admin. Data yang ada juga akan terupdate secara realtime pada aplikasi pengguna. Acara menampilkan judul acara, waktu acara, pelaksana acara, tempat, dan deskripsi acara.

System Testing (Pengujian Sistem)

Uji aplikasi menggunakan metode pengujian *Black Box*. Hasil uji aplikasi diperlihatkan pada Tabel 1.

Tabel 1. Pengujian Aplikasi

Uji Tes	Harapan	Hasil
Masuk	Aplikasi akan mevalidasi data. Jika berhasil, maka pengguna akan dinavigasikan ke menu utama. Jika gagal, coba lagi	Berhasil
Daftar	Aplikasi akan mevalidasi data. Jika berhasil, maka data akan disimpan ke dalam database dan pengguna akan dinavigasikan ke menu utama. Jika gagal, coba lagi	Berhasil
Mencari Donor Darah	Aplikasi akan menampilkan sugesti lokasi donor. Aplikasi akan menampilkan daftar pendonor aktif. Aplikasi akan menyimpan permintaan donor ke database	Berhasil
Push Notification	Aplikasi akan mengirimkan pemberitahuan permintaan donor dengan data pencari donor	Berhasil
Terima atau tolak permintaan donor	Jika diterima, aplikasi akan mengupdate status donor menjadi diterima di database dan memberitahu pencari donor. Jika ditolak, aplikasi akan menghapus permintaan donor di database dan memberitahu pencari donor untuk mencoba mencari pendonor lain	Berhasil
Menampilkan jarak lokasi	Aplikasi akan menggambar polyline dari lokasi pendonor ke lokasi donor darah	Berhasil
Update lokasi pendonor	Aplikasi akan mengupdate lokasi pendonor di database dan mengupdate marker pada aplikasi	Berhasil
Mulai donor	Aplikasi akan mengupdate status donor di database. Jika pendonor tidak dapat melakukan donor, aplikasi akan menghapus permintaan donor di database dan memberitahu pencari donor untuk mencari pendonor lain	Berhasil
Selesai donor	Aplikasi akan mengupdate status donor di database	Berhasil
Menu acara	Aplikasi akan menampilkan acara donor darah	Berhasil
Menu stok darah	Aplikasi akan menampilkan stok darah	Berhasil
Menu acara admin	Aplikasi akan menyimpan informasi ke dalam database dan mengirimkan pemberitahuan ke pengguna jika menambah acara baru	Berhasil
Menu stok darah admin	Aplikasi akan menyimpan informasi ke dalam database	Berhasil

Tabel I merupakan tabel pengujian aplikasi menggunakan metode *Black Box Testing*. Terdapat 13 pengujian yang dilakukan terhadap aplikasi. Dari hasil pengujian didapat aplikasi dapat berfungsi sepenuhnya.

Retrospective (Retrospeksi)

Karena aplikasi sudah berjalan dengan baik maka Penulis hanya perlu mengkaji hasil yang telah didapat bersumber pada pengujian aplikasi. Sebab itu, Penulis tidak butuh mengulang lagi ke tahapan sebelumnya. Namun, jika masih terdapat kesalahan pada aplikasi maka dilakukan revisi mulai dari sesi *Iteration Initialization*.

4. Penutup

Dari hasil penelitian ini dapat disimpulkan bahwa aplikasi pencarian donor darah berbasis *android* yang dibangun dengan metode PXP berjalan sesuai kebutuhan. Layanan *Google Cloud* yang diimplementasikan ke dalam aplikasi ini adalah *google maps API*, *cloud messaging*, *firebase API*. *Firebase database* digunakan sebagai database dan autentikasi. Pemanfaatan *google maps API* untuk implementasi peta dan mengambil titik koordinat lokasi sangat berguna untuk mengetahui lokasi pengguna. Pengambilan titik koordinat lokasi pencari donor, pendonor dan lokasi donor berjalan dengan baik. *Cloud messaging* digunakan untuk mengirim pemberitahuan kepada pendonor. Pencari donor dapat mencari pendonor aktif. Pengguna dapat mengetahui acara donor darah dan stok darah yang tersedia yang sudah diinput oleh admin. Saran untuk pengembangan selanjutnya adalah pencarian donor darah dapat menampung lebih dari satu pendonor.

5. Referensi

- [1] J. Ou-Yang dkk., "Blood donor recruitment in Guangzhou, China, during the 2019 novel coronavirus (COVID -19) epidemic," *Transfusion (Paris)*, vol. 60, no. 11, hlm. 2597-2610, Nov 2020, doi: 10.1111/trf.15971.
- [2] A. M. P. Djuardi, "Blood Donors During The Covid-19 Pandemic", *JMH*, vol. 2, no. 01 Oktober, pp. 298-303, Oct. 2020.
- [3] Sari, E., & Gultom, D. M. (2022). Penyuluhan Kesehatan Tentang Donor Darah di Institut Teknologi dan Kesehatan Sumatera Utara : Padangsidempuan. Pengabdian Deli Sumatera, 1(1), 40-46..
- [4] "golongan darah rhesus negatif," *Alodokter*, 7 Maret 2019. <https://www.alodokter.com/komunitas/topic/golongan-darah-rhesus-negatif> (diakses 18 Juli 2022).
- [5] O. Jonathan, S. Misra, E. Ibanga, R. Maskeliunas, R. Damasevicius, dan R. Ahuja, "Design and implementation of a mobile webcast application with google analytics and cloud messaging functionality," *J. Phys. Conf. Ser.*, vol. 1235, no. 1, hlm. 012023, Jun 2019, doi: 10.1088/1742-6596/1235/1/012023.
- [6] APJII, "Buletin APJII," Asosiasi Penyelenggara Jasa Internet Indonesia. p. 1, 2020, [Online]. Available: <https://apjii.or.id/content/read/104/503/BULETIN-APJII-EDISI-74---November-2020>.
- [7] Suprpto, Fajrur & Marthasari, Gita & Nuryasin, Ilyas. (2020). Sistem Informasi Penjualan dan Pelelangan Berbasis Web pada Ricardo Corner MLG Menggunakan Metode Personal eXtreme Programming (PXP). *Jurnal Repositor*. 2. 1535. 10.22219/repositor.v2i11.926.
- [8] Mardiana Mardiana; Agus Suharto, "Aplikasi Eresha Mobile Berbasis Android dengan Metode Personal Extreme Programming di Era Industri 4.0," *JurTI J. Teknol. Inf.*, no. Vol 4, No 2 (2020): DESEMBER 2020, hlm. 335-344, 2020.
- [9] Dzhurov, Y., Krasteva, I., & Ilieva, S. (2009). Personal Extreme Programming-An Agile Process for Autonomous Developers.
- [10] R. Agarwal dan D. Umphress, "Extreme programming for a single person team," dalam *Proceedings of the 46th Annual Southeast Regional Conference on XX - ACM-SE 46*, Auburn, Alabama, 2008, hlm. 82. doi: 10.1145/1593105.1593127.

- [11]D. W. T. Putra dan R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi *Permohonan* Pembayaran Restitusi SPPD," *J. Teknof*, vol. 7, no. 1, hlm. 32, Apr 2019, doi: 10.21063/jtif.2019.V7.1.32-39.
- [12]Tashildar, A., Shah, N., Gala, R., Giri, T., & Chavhan, P. (2020). Application development using flutter. *International Research Journal of Modernization in Engineering Technology and Science*, 2(8), 1262-1266.
- [13]S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, hlm. 29-50, Jun 2012, doi: 10.5121/ijesa.2012.2204.
- [14]E. Maiyana, "Pemanfaatan Android Dalam Perancangan Aplikasi Kumpulan Doa," *J. Sains Dan Inform.*, vol. 4, no. 1, hlm. 54-65, Apr 2018, doi: 10.22216/jsi.v4i1.3409.
- [15]Smyth, N. (2017). *Firestore Essentials-Android Edition*. Payload Media, Inc..
- [16]A. Faisal dan F. Rahmadianto, "Realtime Notification Pada Aplikasi Berbasis Web Menggunakan Firebase Cloud Messaging (FCM)," *J. Mnemon.*, vol. 1, no. 2, hlm. 14-17, Des 2019, doi: 10.36040/mnemonic.v1i2.32.
- [17]H. Wu, "Comparing Google Maps and Uber Movement Travel Time Data," *Transp. Find.*, Nov 2018, doi: 10.32866/5115.