

Development of a Mobile Application for Blood Donor Search

Pengembangan Aplikasi Mobile Pencarian Pendonor Darah

Joseph Julio Nicholas, Xaverius B. N. Najoan, Jimmy R. Robot

Dept. of Electrical Engineering, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia

e-mails : juliojnicolas1@gmail.com, xnajoan@unsrat.ac.id, jimmy.robot@unsrat.ac.id

Received: 22 July 2024; revised: 13 September 2024; accepted: 10 December 2024

Abstract — *Blood is one of the most important components of the human body. Limited blood supplies from health agencies sometimes become a serious problem, especially in emergency situations where someone needs a blood transfusion as soon as possible. One solution that people commonly use when they need blood is to use social media platforms to spread requests for blood. However, this method is still less effective because messages requesting blood often do not reach the right person at the right time. The aim of this research is how to build an application that can help bring together parties who need donors with parties who donate voluntarily. The development process was carried out using the waterfall development method with the programming languages used, namely Kotlin and JavaScript (Node.Js). The result of this research is an Android-based application that allows users to disseminate blood need information in a more targeted manner and makes it easier for voluntary donors to find blood donation requests that match their blood type.*

Key words—Android Based Application; Blood Donation; Blood Needs.

Abstrak — Darah merupakan salah satu komponen yang paling penting yang dimiliki oleh tubuh manusia. Keterbatasan persediaan darah dari pihak instansi kesehatan terkadang menjadi masalah yang serius terutama dalam situasi darurat di mana seseorang membutuhkan transfusi darah sesegera mungkin. Salah satu solusi yang umum digunakan masyarakat saat sedang membutuhkan darah adalah menggunakan platform media sosial untuk menyebarkan permintaan kebutuhan darah. Namun, cara ini masih kurang efektif dikarenakan pesan permintaan kebutuhan darah sering kali tidak mencapai orang yang tepat pada waktu yang tepat. Tujuan dari penelitian ini adalah tentang bagaimana membangun sebuah aplikasi yang dapat membantu mempertemukan pihak yang membutuhkan donor dengan pihak yang melakukan donor secara sukarela. Proses pengembangan dilakukan dengan menggunakan metode pengembangan waterfall dengan bahasa pemrograman yang digunakan yaitu Kotlin dan JavaScript (Node.Js). Hasil dari penelitian ini adalah sebuah aplikasi berbasis Android yang memungkinkan pengguna untuk menyebarkan informasi kebutuhan darah dengan lebih tepat sasaran serta memudahkan pemberi donor sukarela untuk menemukan permintaan donor darah yang sesuai dengan golongan darah mereka.

Kata kunci — Aplikasi Berbasis Android; Donor Darah; Kebutuhan darah.

I. PENDAHULUAN

Kesehatan adalah hak yang fundamental bagi setiap orang. Salah satu aspek kesehatan yang kritis adalah ketersediaan darah dalam situasi darurat. Darah merupakan komponen yang

sangat penting dalam pelayanan medis, terutama dalam penanganan keadaan darurat seperti kecelakaan, operasi, dan kondisi medis kritis lainnya. Secara umum, ketika seorang pasien memerlukan transfusi darah secara mendesak, maka pasien akan mendapatkan pasokan darah dari fasilitas Kesehatan tempat pasien berada atau dari Unit Transfusi Darah Palang Merah Indonesia (PMI) terdekat. Namun, sering kali golongan darah yang dibutuhkan tidak dapat dipenuhi dikarenakan kurangnya persediaan darah yang dimiliki unit donor darah di daerah tersebut. Berdasarkan badan Organisasi Kesehatan Dunia (WHO), dengan mengambil populasi Indonesia pada tahun 2023, Indonesia memerlukan stok darah kurang lebih 5,57 juta stok darah atau sebanyak 2% dari total penduduk, sedangkan menurut data dari Palang Merah Indonesia pada tahun 2023, Indonesia hanya memiliki stok kantong darah sebanyak 77.438 yang tersebar di seluruh unit donor darah (UDD) di Indonesia.

Dalam situasi di mana stok darah terbatas dan tidak mencukupi kebutuhan pasien, keluarga atau teman-teman pasien sering kali terpaksa mencari jalan lain yaitu dengan memanfaatkan sosial media atau aplikasi pengirim pesan seperti Twitter, Facebook, WhatsApp, dll, untuk mencari calon pendonor darah yang ideal. Hal ini memiliki beberapa kekurangan yakni informasi yang diunggah ke platform tersebut berpotensi terhalangi dengan informasi yang diunggah oleh pengguna lain. Selain itu, keefektifan penyebarluasan informasi melalui media sosial juga dipengaruhi dengan jumlah pengikut pada akun yang membagikan informasi. Semakin banyak jumlah pengikut maka semakin luas informasi dapat tersebar [1].

Berdasarkan permasalahan yang telah diuraikan, diperlukan suatu aplikasi yang mampu membantu pasien yang membutuhkan darah untuk menemukan pendonor darah dengan lebih cepat dalam situasi darurat. Teknologi push notification 2 dan Global Positioning System (GPS) akan digunakan untuk memberitahu pengguna lain yang menggunakan aplikasi mengenai kebutuhan darah yang mendesak di sekitar mereka serta memastikan bahwa notifikasi kebutuhan darah hanya akan tersebar ke pengguna yang berada dekat dengan lokasi pencari donor.

Aplikasi yang akan dikembangkan akan menerapkan pola arsitektur ModelView-ViewModel (MVVM). Pola arsitektur

ini dipilih karena kemampuannya untuk memisahkan logika bisnis, tampilan, dan data sehingga komponen aplikasi dapat bekerja secara terpisah dan lebih terorganisir. Implementasi MVVM membantu dalam meningkatkan kualitas kode, memudahkan proses pengembangan dan pemeliharaan aplikasi. Dengan pendekatan ini, diharapkan aplikasi pencarian donor darah dapat berfungsi dengan lebih optimal, efisien dan memberikan pengalaman pengguna yang lebih baik.

Dengan demikian, pengembangan aplikasi mobile pencarian donor darah dapat menjadi solusi dalam meningkatkan efisiensi dan kecepatan dalam menyediakan darah dalam situasi mendesak.

A. Penelitian terkait

Dalam penelitian yang dilakukan oleh Ma Xin Ying dan Hanyanti Hafit yang berjudul "Emergency Notification And Healthcare Mobile Application for Elderly," membahas mengenai implementasi pemberitahuan darurat berbasis *Push Notification* pada platform *Mobile*. Hasil dari penelitian ini yaitu sebuah aplikasi untuk membantu lansia dan wali mereka bertemu dalam keadaan darurat [2].

Dalam penelitian yang dilakukan Natalie Jomini Stroud bersama rekan-rekannya dengan judul "The Effects of Mobile Push Notifications on News Consumption and Learning," ditemukan bukti empiris peningkatan penggunaan aplikasi yang menerapkan mekanisme *Push Notification* [3].

Dalam penelitian yang dilakukan oleh Muh. Iqbal, bersama rekan-rekannya, yang berjudul "Penerapan Metode Haversine Pada Sistem Informasi Geografis Pencarian Lokasi Laundry Terdekat di Kota Makassar," para peneliti mengembangkan suatu sistem informasi geografis untuk melakukan pemetaan lokasi tempat cucian baju terdekat. Penelitian ini menerapkan metode *haversine* untuk memetakan lokasi cucian baju berdasarkan lokasi pengguna aplikasi. Dalam menerapkan metode *haversine*, penelitian ini menemukan bahwa kecepatan pemetaan lokasi cucian terdekat dipengaruhi oleh kondisi lingkungan dan cuaca [4].

Dalam penelitian yang dilakukan oleh Wiwik Wiharti dan rekan-rekannya dengan judul "Rancangan dan Implementasi Aplikasi Mobile Tanggap Darurat COVID-19 Berbasis Android dan GPS," peneliti bersama rekan-rekannya mengembangkan aplikasi tanggap COVID-19 pada platform Android yang memanfaatkan teknologi *Global Positioning System* (GPS). Penelitian ini menghasilkan aplikasi yang dapat memetakan fasilitas Kesehatan terdekat serta dapat menunjukkan lokasi keberadaan OTG (Orang Tanpa Gejala) secara real-time, dengan tingkat akurasi jarak rata-rata sebesar 12,9 meter dari jarak sebenarnya [5].

Dalam penelitian yang dilakukan oleh Tri Ferga Prasetyo bersama rekan-rekannya dengan judul "Implementasi Sistem Tanggap Darurat Bencana Dengan Menggunakan Firebase Cloud Messaging Berbasis Android Dan Sms Gateway". Peneliti bersama rekan-rekannya mengembangkan aplikasi tanggap darurat bencana dengan memanfaatkan *Firebase Cloud Messaging* dan *SMS Gateway* untuk memberikan notifikasi secara *real-time* kepada pengguna untuk meningkatkan kecepatan dalam penyampaian informasi kebencanaan kepada masyarakat [6].

Dalam penelitian yang dilakukan oleh Ruth Simamora beserta rekan-rekannya dengan judul "Rancang Bangun Aplikasi Ambulance Online Berbasis Android," bertujuan mengembangkan aplikasi Ambulans online berbasis Android, penelitian ini berhasil mengintegrasikan layanan *Firebase Cloud Messaging* dan *Google Maps API*, untuk memberikan informasi status pesanan dengan cepat serta menampilkan informasi lokasi ambulans melalui aplikasi [7].

Dalam penelitian yang dilakukan oleh Mohamad Eko Saifudin dan Hadi Zakaria dengan judul "Rancang Bangun Aplikasi Antrian Secara Realtime Dengan Fitur Push Notification Menggunakan Smartphone Berbasis Android (Studi Kasus: Klinik Sandiana)", para peneliti mengimplementasikan *Firebase Cloud Messaging* (FCM) pada aplikasi antrian yang akan diterapkan di klinik Sandiana. Penelitian ini menerapkan model *waterfall* sebagai metode pengembangan. Penerapan FCM pada aplikasi ini memungkinkan pasien untuk mendapatkan informasi *real-time* dari nomor antrian yang sedang berjalan [8].

Dalam penelitian yang dilakukan oleh Mohammad Imron bersama rekan-rekannya dengan judul "Implementasi Push Notification Pada Sistem Peminjaman Sarana dan Prasarana Berbasis Website", para peneliti mengembangkan aplikasi peminjaman sarana prasarana perkuliahan di Universitas Amikom Purwokerto. *Website* tersebut dikembangkan dengan menerapkan model *waterfall* sebagai metode pengembangan. Penelitian ini berhasil mengimplementasikan mekanisme *push notification* berbasis SMS sebagai pengingat akan status pinjaman barang yang dilakukan oleh pengguna tersebut, sehingga dapat mengurangi resiko keterlambatan dalam pengembalian [9].

B. Donor Darah

Donor darah merupakan tindakan seseorang menyumbangkan darahnya secara sukarela untuk keperluan transfusi atau pengobatan medis lainnya [10]. Donor darah terbagi dalam 4 jenis, yaitu darah utuh, trombosit, plasma, dan sel darah merah. Tindakan donor darah penting dalam mendukung pasien yang membutuhkan transfusi darah atau perawatan medis yang memerlukan persediaan darah yang memadai. Dikutip dari Palang Merah Indonesia (PMI), donor darah dapat dilakukan dengan syarat pendonor berusia 17 hingga 65 tahun dengan minimal berat badan 45 Kg. Pendonor harus memiliki tekanan darah dalam kisaran normal yaitu Sistole 100-180 dan Diastole 70-100 dengan kadar hemoglobin yang diperlukan berkisar antar 12,5-17,0 gr/dL%. Sesuai dengan PERMENKES 91 Tahun 2015, pendonor diberikan jeda waktu minimal 2 bulan sejak donor darah terakhir untuk dapat melakukan donor darah kembali.

C. Android

Android merupakan sebuah sistem operasi mobile berbasis 32-bit dan 64 bit yang berasal dari modifikasi kernel Linux dan perangkat lunak *open-source* lainnya. Android bersifat *open source*, sehingga memungkinkan pengembang untuk mengakses serta melakukan modifikasi pada *source code* yang sudah ada. Aplikasi Android umumnya ditulis dalam bahasa pemrograman Java [11] namun pada tahun 2017, Google mengumumkan bahasa pemrograman Kotlin sebagai bahasa resmi Android.

D. Global Positioning System (GPS)

Global Positioning System atau yang biasa disebut GPS merupakan sebuah sistem navigasi yang menggunakan sinyal dari sejumlah satelit untuk menentukan lokasi dengan akurasi tinggi dengan koordinat yang ditandai dengan latitude (garis lintang), longitude (garis bujur), dan elevation (ketinggian) [12]. Saat ini, GPS telah menjadi fitur standar yang terintegrasi dalam sebagian besar perangkat terutama perangkat mobile tak terkecuali Android [13].

E. Push Notification

Push Notification merupakan sebuah pesan yang secara otomatis dikirimkan oleh aplikasi atau ke server perangkat pengguna tanpa perlu diminta oleh pengguna. Tujuan utama dari push notification adalah memberitahu orang tentang konten yang disediakan melalui aplikasi [14]. Push notification bekerja melalui koneksi internet, sehingga untuk menerima pesan dari push notification, pengguna tidak perlu membuka aplikasi atau situs web untuk menerima pesan tersebut.

F. Firebase

Firebase adalah sebuah platform yang dikembangkan oleh perusahaan Google dengan tujuan mendukung pengembangan aplikasi web maupun mobile. Firebase menyediakan layanan serverless yang memudahkan pengembang untuk mengembangkan dan mengelola aplikasi mereka. Hal ini karena, Firebase dirancang dengan fleksibilitas tinggi untuk memenuhi kebutuhan beragam aplikasi, sehingga pengguna tidak perlu lagi memikirkan pengaturan server, manajemen data, atau tugas-tugas administratif lainnya yang biasa terjadi dalam penggunaan server konvensional [15].

G. Haversine Formula

Formula Haversine adalah rumus matematika yang digunakan untuk menentukan jarak antara dua koordinat lintang dan bujur (latitude dan longitude) pada sebuah permukaan bola [16] dengan mempertimbangkan kelengkungannya. Dalam studi kasus aplikasi pencarian pendonor darah, rumus ini akan diterapkan pada fitur notifikasi untuk memastikan bahwa hanya pengguna yang berada dalam radius tertentu dari alamat pasien yang akan menerima notifikasi. Berikut adalah persamaan dari formula Haversine:

$$\begin{aligned} a &= \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (1) \\ c &= 2 \cdot \text{atan} 2(\sqrt{a}, \sqrt{1-a}) \\ d &= R \cdot c \end{aligned}$$

Di mana ϕ_1 dan ϕ_2 adalah garis lintang (*latitude*) dari dua titik dalam radian., λ_1 dan λ_2 adalah garis bujur (*longitude*) dari dua titik dalam radian, $\Delta\phi$ adalah perbedaan garis lintang: $\Delta\phi = \phi_2 - \phi_1$, $\Delta\lambda$ adalah perbedaan garis bujur: $\Delta\lambda = \lambda_2 - \lambda_1$, R adalah jari-jari Bumi (~ 6371 km), dan d adalah jarak antar dua titik.

H. Model-View-ViewModel (MVVM)

MVVM adalah sebuah model arsitektur dalam pengembangan perangkat lunak yang mana model ini bertujuan untuk menghilangkan ketergantungan antar komponen [17]. Dalam MVVM, terdapat 3 komponen utama sesuai dengan

namanya, *Model*, *View*, dan *ViewModel*. *Model* merujuk pada representasi data dan logika bisnis, *View* merujuk pada tampilan antarmuka pengguna, dan *ViewModel* bertindak sebagai penghubung antara *Model* dan *View*.

I. Waterfall Model

Waterfall Model adalah salah satu pendekatan SDLC dengan model pengembangan secara linear dan sekuensial. Model ini merupakan model yang paling awal dan salah satu model yang paling terkenal dalam SDLC. Dalam pengembangan perangkat lunak menggunakan model waterfall, setiap fase harus diselesaikan sepenuhnya sebelum lanjut ke fase berikutnya. Hal inilah yang membuat model waterfall sebagai model pengembangan secara sekuensial. Setiap fase memiliki tujuannya masing-masing yang nantinya akan menunjang pengerjaan fase berikutnya [18].

II. METODE

A. Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan mulai dari bulan Februari 2024 hingga selesai. Lokasi penelitian bertempat di Program Studi Teknik Informatika, Jurusan Elektro, Fakultas Teknik, Universitas Sam Ratulangi..

B. Perangkat Keras dan Perangkat Lunak Penelitian

Perangkat keras dan perangkat lunak yang akan digunakan dalam penelitian ini dapat dilihat pada tabel I:

TABEL I
PERANGKAT KERAS DAN PERANGKAT LUNAK PENELITIAN

No.	Perangkat Lunak/Perangkat Keras	Keterangan
1	Smartphone Android	Xiaomi Redmi Note 9
2	Laptop	HP Victus Gaming 15-fa0xxx, Intel Core i7, 8 GB RAM, 500GB Internal Memory, Windows 11, 64-bit.
3	Android Studio	IDE akan digunakan untuk mengembangkan aplikasi Android.
4	Visual Studio Code	IDE yang akan digunakan untuk pengembangan <i>backend</i> .
5	Kotlin	Bahasa pemrograman yang akan digunakan selama proses pengembangan aplikasi.
6	JavaScript (Node.js)	Bahasa pemrograman yang akan digunakan untuk pengembangan <i>backend</i> .
7	Figma	Digunakan untuk merancang antarmuka aplikasi.
8	Firebase (Realtime Database, Authentication, Firebase Cloud Messaging)	Digunakan untuk keperluan penyimpanan data, autentikasi, dan layanan <i>push notification</i> .
9	Google Maps API	Digunakan untuk menampilkan alamat pasien yang membutuhkan darah.

C. Metode Pengembangan Perangkat Lunak

Proses pengembangan perangkat lunak mengikuti pendekatan sekuensial dari Software Development Lifecycle (SDLC) dengan mengikuti model Waterfall. Metode ini dipilih karena menyediakan kerangka kerja yang terstruktur dan terorganisir dengan fase-fase yang terdefinisi dengan baik, sehingga memungkinkan perencanaan 13 dan dokumentasi yang lebih komprehensif. Metode Waterfall meliputi beberapa fase antara lain:

1) Analisis Kebutuhan

Pada tahap ini, identifikasi tujuan pengembangan aplikasi akan dilakukan. Hal ini meliputi identifikasi permasalahan yang ada serta pengumpulan data dari calon pengguna. Dalam penelitian ini, data akan diambil dari hasil responden kuesioner yang mana data ini nantinya akan dijadikan pertimbangan untuk pengembangan fitur dan rancangan antarmuka nantinya.

2) Desain Sistem

Pada tahap ini akan dilakukan perancangan arsitektur sistem, antarmuka pengguna, serta rancangan basis data dari perangkat lunak yang akan dikembangkan. Untuk pengembangan sistem akan menerapkan pola arsitektur Model-View-Viewmodel (MVVM) untuk mencegah ketergantungan antar komponen, sehingga dapat mempermudah proses pengembangan lebih lanjut.

3) Implementasi

Pada tahap ini, akan dilakukan pengembangan aplikasi berdasarkan rancangan dan desain dari tahapan-tahapan sebelumnya. Tahapan ini meliputi proses penyusunan antarmuka serta penulisan kode program sesuai dengan bahasa pemrograman yang sudah ditentukan.

4) Pengujian

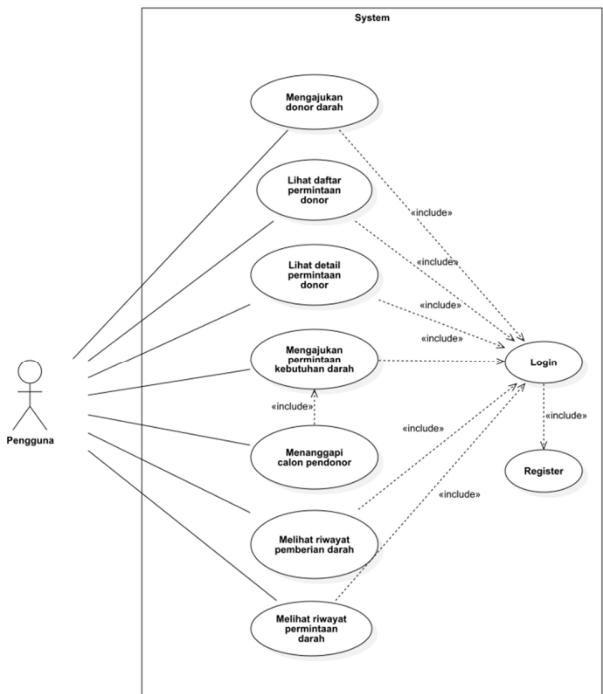
Pada tahap ini, akan dilakukan pengujian pada perangkat lunak yang telah bangun. Pengujian akan dilakukan dengan metode Blackbox Testing yang mana hasil akhirnya akan dievaluasi untuk dilakukan penilaian fungsionalitas aplikasi apakah sudah sesuai dengan yang diharapkan.

D. Unified Modeling Language (UML)

UML adalah sebuah standar yang digunakan untuk melakukan pemodelan sistem secara visual. Terdapat banyak jenis diagram UML yang tersedia, namun dalam studi ini fokus utama adalah pada dua jenis diagram UML, yaitu Use Case Diagram, dan Activity Diagram.

1) Use case diagram

Use case diagram adalah salah satu bentuk diagram Unified Modelling Language (UML) yang digunakan untuk menggambarkan interaksi antara pengguna atau yang biasa disebut aktor, dengan sistem. Dalam studi kasus aplikasi pencarian donor, Use case diagram ditampilkan pada gambar 1.



Gambar 1. Use Case Diagram aplikasi pencarian donor

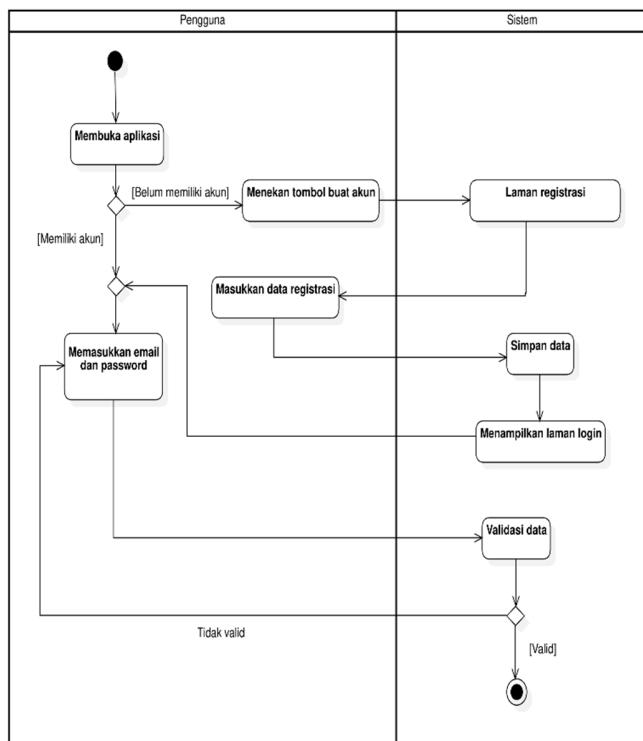
Dalam studi kasus aplikasi pencarian donor yang dikembangkan, satu akun dapat berperan sebagai pendonor sekaligus pencari donor. Penjelasan setiap *use case* disajikan dalam tabel II berikut.

TABEL II
DESKRIPSI USE CASE DIAGRAM APLIKASI PENCARIAN DONOR

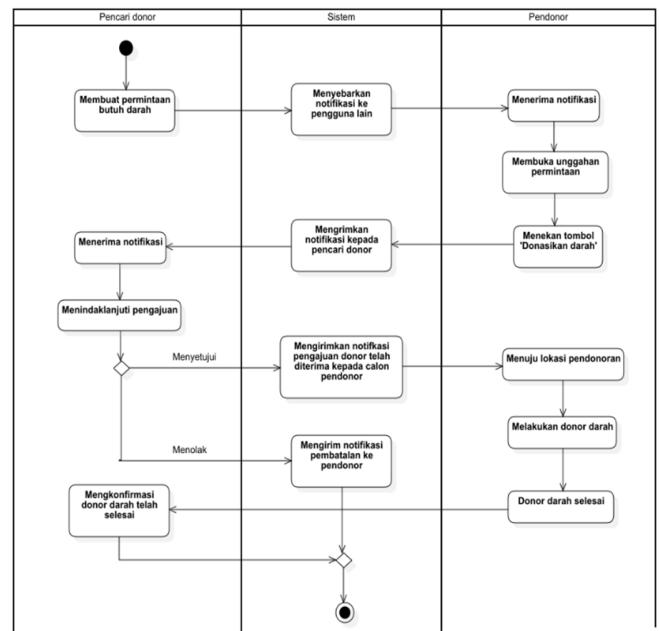
Nama Use Case	Deskripsi
Mengajukan donor darah	Fitur ini memungkinkan pengguna untuk dapat menawarkan diri pendonor pada salah satu permintaan kebutuhan darah. Pengguna dapat melihat daftar permintaan kebutuhan darah berdasarkan golongan darah, memungkinkan pengguna untuk melihat siapa yang membutuhkan donor darah.
Lihat daftar permintaan donor	Pengguna dapat melihat informasi lebih lanjut tentang kebutuhan dan siapa yang memerlukan donor tersebut.
Lihat detail permintaan donor	Pengguna dapat mengajukan permintaan kebutuhan darah untuk disebarluaskan ke pengguna lain.
Mengajukan permintaan kebutuhan darah	Pengguna yang sedang mencari donor dapat menanggapi calon pendonor yang telah mengajukan diri untuk mendonorkan darah. Untuk melakukan ini, maka pengguna harus mengajukan permintaan kebutuhan darah terlebih dahulu.
Menanggapi calon pendonor	Pengguna dapat melihat data riwayat pemberian darah yang pernah dilakukan. Pengguna dapat melihat data riwayat permintaan darah yang pernah dilakukan. Fungsi ini memungkinkan pengguna untuk dapat masuk ke dalam sistem.
Melihat riwayat pemberian darah	Fungsi ini memungkinkan pengguna baru untuk mendaftar ke dalam sistem.
Melihat riwayat permintaan darah	
Login	
Register	

2) Activity diagram

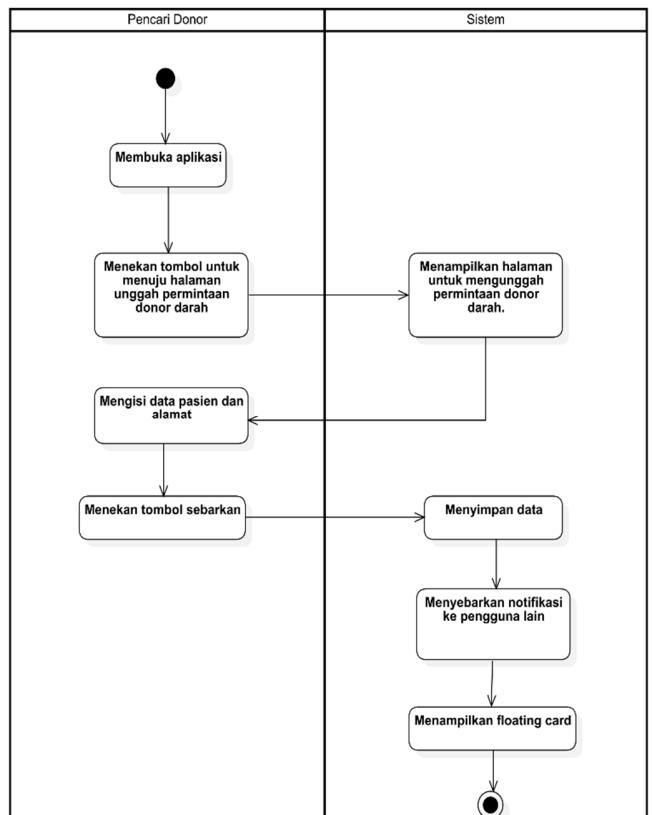
Activity diagram adalah salah satu jenis diagram Unified Modelling Language (UML) yang bertujuan untuk memodelkan alur aktivitas dari sebuah sistem. Dalam studi kasus aplikasi pencarian pendonor darah, Activity Diagram digunakan untuk memodelkan alur aktivitas pengguna saat menggunakan sistem sesuai dengan fitur-fitur yang ada



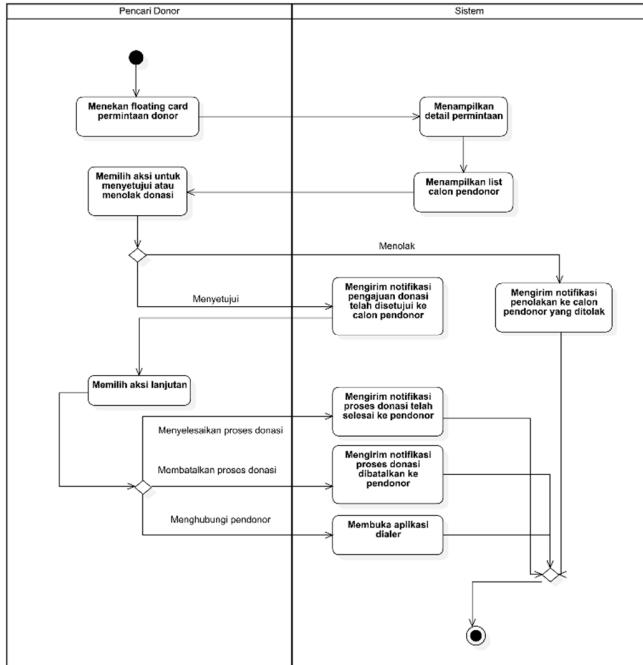
Gambar 2. Activity Diagram proses login dan registrasi



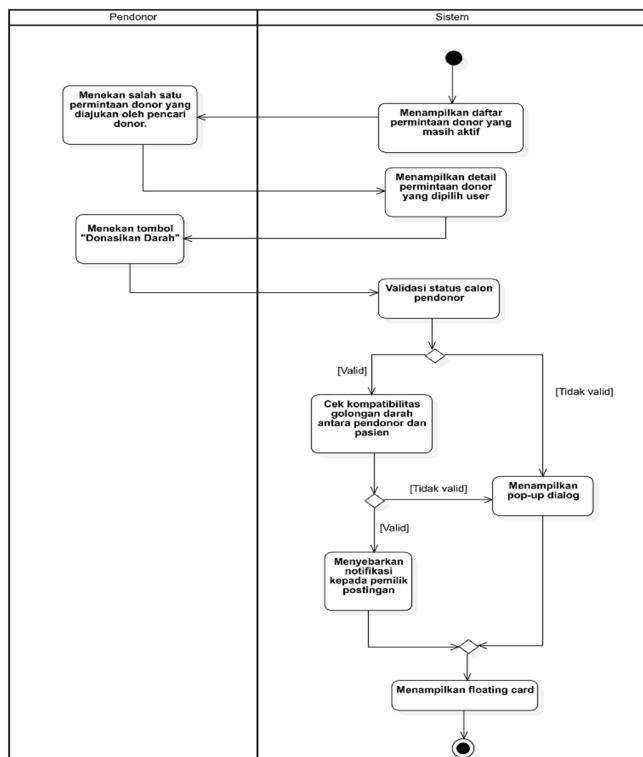
Gambar 3. Activity Diagram aplikasi pencarian donor darah



Gambar 4. Activity Diagram proses membuat unggahan permintaan kebutuhan darah

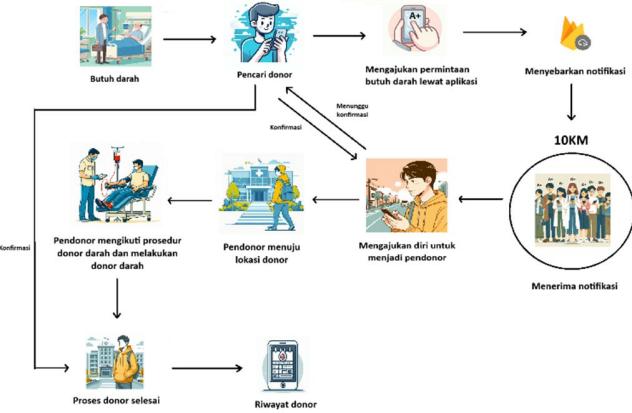


Gambar 5. *Activity Diagram* menanggapi pengajuan donor darah dari calon pendonor



Gambar 6. *Activity Diagram* menanggapi pengajuan donor darah dari calon pendonor

E. Rich Picture Diagram



Gambar 7. Diagram *Rich Picture*

III. HASIL DAN PEMBAHASAN.

A. Analisis Kebutuhan

Pada tahap ini, dilakukan identifikasi kebutuhan dari pengguna yang didasarkan pada hasil observasi dan kuesioner yang telah dilakukan. Analisis kebutuhan dibagi menjadi dua, yakni kebutuhan fungsional dan kebutuhan non-fungsional. Berikut adalah kebutuhan fungsional dan non-fungsional yang berhasil diidentifikasi:

1) Kebutuhan fungsional

Kebutuhan fungsional meliputi kebutuhan-kebutuhan yang harus ada dalam sistem. Kebutuhan fungsional dari aplikasi pencarian donor, antara lain:

1. Sistem memungkinkan pengguna untuk mendaftar agar dapat menggunakan aplikasi.
2. Sistem memungkinkan pengguna untuk *login* dengan memasukkan kata sandi dan email sebelum mengakses fitur dalam aplikasi.
3. Sistem memungkinkan pengguna untuk mengunggah permintaan butuh darah.
4. Sistem memungkinkan pengguna untuk mengajukan permintaan donasi darah ke salah satu permintaan butuh darah yang masih aktif, dengan syarat hanya jika golongan darahnya kompatibel dengan yang tertera pada permintaan tersebut.
5. Sistem memungkinkan pengguna yang mengunggah permintaan butuh darah untuk dapat menanggapi pengajuan donor dari pengguna lain.
6. Sistem dapat mengirimkan notifikasi kepada pengguna lain yang sesuai saat ada permintaan butuh donor.
7. Sistem memungkinkan pengguna untuk mengakses riwayat pemberian darah/permintaan darah.

2) Kebutuhan non-fungsional

Kebutuhan non-fungsional meliputi kebutuhan-kebutuhan yang tidak berkaitan langsung dengan fitur utama dari sistem yang akan dikembangkan. Kebutuhan ini umumnya berkaitan dengan kinerja, kemudahan, dan keamanan aplikasi. Beberapa

kebutuhan non-fungsional dari aplikasi pencarian donor yang dipertimbangkan, antara lain:

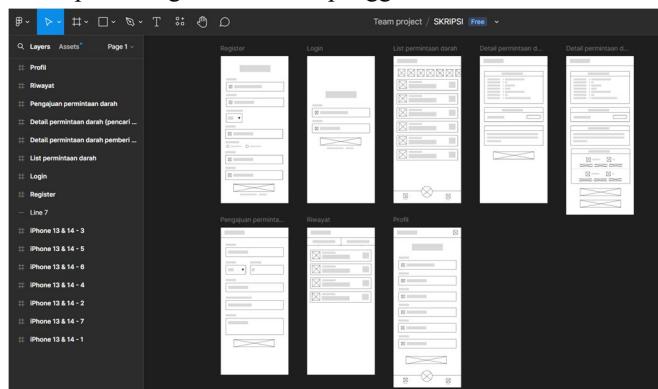
1. Antarmuka sistem harus mudah digunakan oleh pengguna.
2. Sistem harus menggunakan mekanisme autentifikasi yang handal.
3. Notifikasi harus disampaikan secara instan dan konsisten.
4. Sistem harus mengoptimalkan penggunaan sumber daya, seperti memori dan CPU.
5. Ukuran aplikasi yang dihasilkan harus minimal.

B. Desain Sistem

Tahap ini akan menjelaskan rancangan antarmuka pengguna (UI) sistem pada perangkat mobile, arsitektur sistem, dan rancangan basis data untuk memenuhi kebutuhan yang telah diidentifikasi sebelumnya.

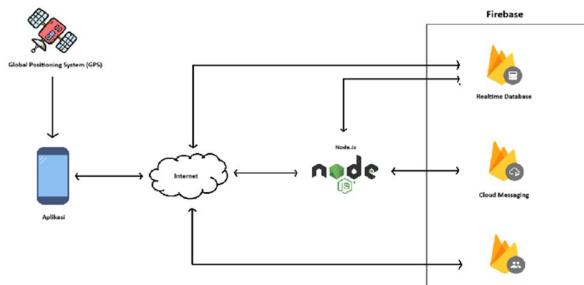
1) Desain antarmuka pengguna

Desain awal antarmuka pengguna aplikasi pencarian pendonor darah dibuat menggunakan teknik *wireframing*. *Wireframe* ini akan berperan sebagai cetak biru yang memberikan gambaran kasar dari tampilan aplikasi sesungguhnya. Untuk proses perancangan *wireframe* dilakukan menggunakan aplikasi Figma, sebuah *tool* yang digunakan untuk perancangan antarmuka pengguna.



Gambar 8. Tampilan perancangan *wireframe* aplikasi

2) Arsitektur sistem

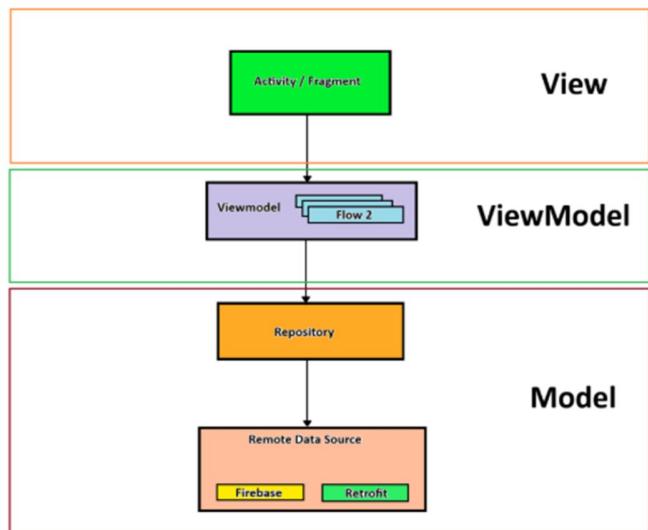


Gambar 9. Arsitektur sistem aplikasi pencarian donor darah

Diagram arsitektur sistem pada Gambar 9 di atas menunjukkan gambaran interaksi antar komponen pada sistem aplikasi pencarian donor darah. Sistem ini memiliki empat komponen utama yaitu aplikasi klien, *Global Positioning*

System (GPS), *Node.js* dan *Firebase*. *GPS* mengirimkan data lokasi pengguna ke aplikasi klien untuk menghitung jarak antara pasien dan pengguna aplikasi. Aplikasi klien menerima data lokasi dari *GPS* dan berkomunikasi dengan layanan *Firebase* melalui Internet. *Firebase* menawarkan tiga layanan utama: *Realtime Database*, *Authentication*, dan *Cloud Messaging*. Aplikasi klien dapat berinteraksi langsung dengan semua layanan ini, kecuali *Cloud Messaging*, yang memerlukan proksi layanan backend *Node.js*. *Node.js* bertindak sebagai *backend* yang menghubungkan aplikasi klien dengan *Cloud Messaging*, meneruskan pesan notifikasi dari klien ke *Cloud Messaging*, dan berkomunikasi dengan *Realtime Database* untuk mendapatkan token perangkat, yang dikirim ke *Cloud Messaging* sebagai identitas notifikasi *Messaging* perangkat sasaran.

Adapun pola arsitektur perangkat lunak yang digunakan selama pengembangan aplikasi pencarian donor darah dapat dilihat pada Gambar 10



Gambar 10. Diagram pola arsitektur MVVM

Pengembangan aplikasi pencarian pendonor darah menerapkan pola arsitektur *Model-View-ViewModel* (MVVM) untuk memisahkan logika presentasi dan logika bisnis dari aplikasi. *View* merupakan lapisan yang bertanggung jawab menangani interaksi pengguna dan menampilkan data kepada pengguna, dengan komponen berupa kelas *Activity* atau kelas *Fragment*. *ViewModel* bertindak sebagai penghubung antara *View* dan *Model*, menyimpan dan mengelola data yang dibutuhkan oleh *View*, serta berisi objek-objek *Flow* untuk mengalirkan data secara asinkron dari *Model* ke *View* dan menyajikannya secara reaktif dengan mekanisme observasi. *Model* mengelola kebutuhan data yang dibutuhkan oleh *View*, berisi kelas-kelas *model* sebagai representasi data, dan kelas *Repository* sebagai penyedia data kepada *ViewModel* dengan mengabstraksikan sumber data seperti *Firebase* dan *Web Service*, sehingga *ViewModel* tidak perlu mengetahui sumber data tersebut.

3) Struktur basis data

Terdapat tiga objek utama dalam struktur basis data aplikasi pencarian donor darah, yakni *donorRequest*, *history*, dan *users*. Objek *donorRequest* menyimpan informasi tentang permintaan darah yang disebarluaskan oleh para pencari donor, termasuk objek *potentialDonor* yang menyimpan daftar calon pondonor dari permintaan tersebut. Objek *history* menyimpan informasi mengenai riwayat permintaan dan pemberian darah dari para pengguna. Objek *users* menyimpan informasi data diri pengguna, termasuk objek *deviceToken* yang menyimpan token perangkat pengguna untuk digunakan dalam *push notification* bersama waktu pembuatan token tersebut.

Donor Request	History	Users
<pre>{ "donorRequest": { "donorRequestId": "string", "address": "string", "bloodType": "string", "contact": "string", "dateRequested": "string", "id": "string", "latitude": "number", "longitude": "number", "name": "string", "notes": "string", "potentialDonors": [{ "potentialDonorId": "string", "bloodType": "string", "donation": "string", "donorName": "string", "id": "string", "phoneNumber": "string", "status": "string" }], "quantity": "number", "requested": "string", "status": "string" } }</pre>	<pre>{ "history": [{ "historyId": "string", "bloodRequesterId": "string", "bloodType": "string", "bloodVolume": "string", "id": "string", "location": "string", "pacientName": "string", "type": "string", "userId": "string" }] }</pre>	<pre>[{ "users": [{ "age": "number", "bloodType": "string", "deviceToken": "string", "email": "string", "id": "string", "isAvailable": "boolean", "lastOperationTime": "number", "name": "string", "phoneNumber": "string", "score": "string" }] }]</pre>

Gambar 11. Struktur basis data

C. Implementasi

Tahap ini mencakup implementasi kode program, basis data, serta implementasi antarmuka pengguna. Penulisan kode aplikasi dilakukan dengan menggunakan IDE Android Studio, sedangkan kode *backend* penghubung antara aplikasi dan layanan *Cloud Messaging* ditulis dengan menggunakan IDE Visual Studio Code.

1) Implementasi Firebase

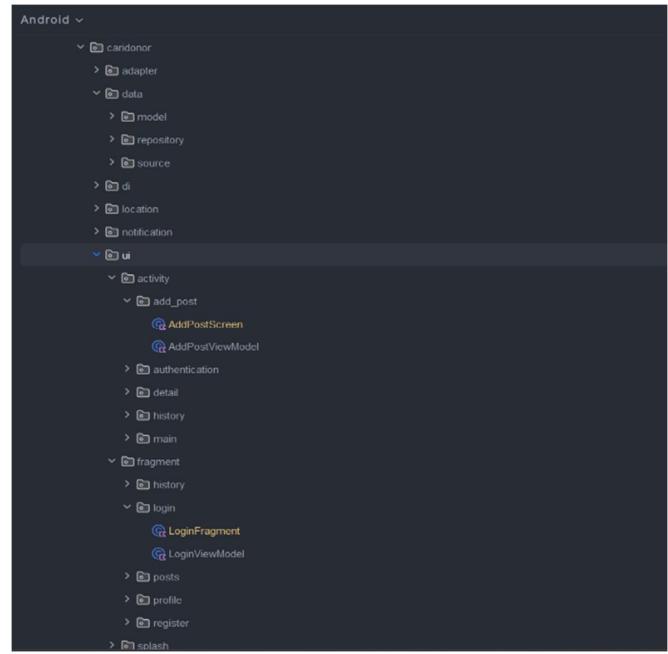
Penelitian ini memanfaatkan tiga layanan Firebase, yaitu *Realtime database*, *Authentication*, dan *Cloud Messaging*. Layanan *Realtime database* digunakan sebagai basis data. Layanan *Authentication* berfungsi untuk mengelola autentifikasi pengguna. *Cloud Messaging* digunakan untuk mengirimkan *push notification* kepada pengguna.

2) Implementasi Model-View-ViewModel

Pada penelitian ini, MVVM diterapkan dengan memisahkan tanggung jawab dari setiap komponen ke dalam folder serta *file-file* yang berbeda. Struktur folder dari aplikasi ini dapat dilihat pada gambar 12.

Folder *Data* menampung *file-file* yang merepresentasikan lapisan *Model* dalam arsitektur MVVM. *Model* di sini akan berinteraksi dengan basis data dan API untuk mengirim, menyimpan, dan memproses data terkait permintaan kebutuhan darah. Folder *Activity* dan *Fragment* menampung tiga jenis *file*, yaitu *file* yang berisikan kelas *Activity*, *file* yang berisikan kelas *Fragment*, dan *file* yang berisikan kelas *ViewModel*. Lapisan *View* direpresentasikan oleh kelas *Activity* dan *Fragment*. Sebagai contoh, di dalam folder ‘add_post’ terdapat *file* ‘AddPostScreen.kt’. *File* ini berisikan kelas *Activity* yang bertugas untuk menampilkan data kepada pengguna dan

menangani interaksi pengguna, begitu juga *file* ‘LoginFragment.kt’. Lapisan *ViewModel* direpresentasikan oleh *file-file* yang berisikan kelas yang mewarisi kelas *ViewModel*. Dalam struktur folder yang ada pada gambar 12, kelas-kelas *ViewModel* ditulis dalam *file* yang berakhiran ‘*ViewModel*’, seperti ‘LoginViewModel.kt’.



Gambar 12. Struktur folder

3) Implementasi Push Notification

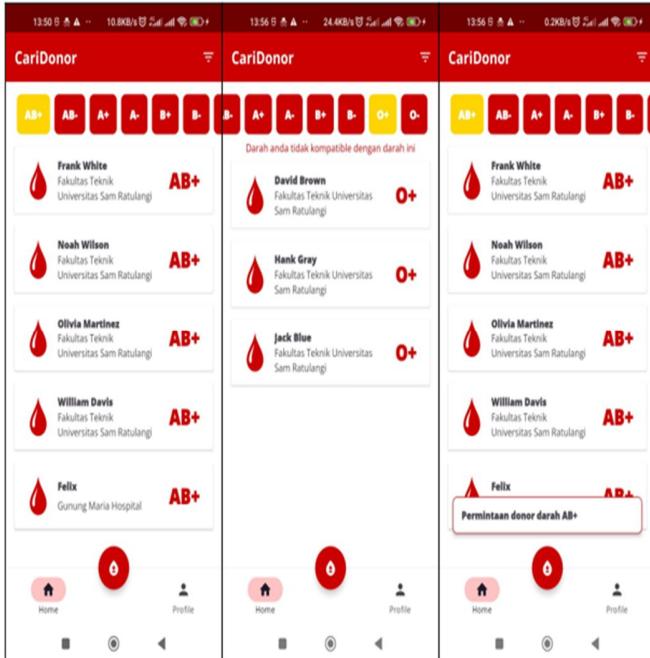
Implementasi push *notification* terbagi menjadi dua, yaitu dari sisi *backend* dan dari sisi *client*. Implementasi pada sisi *backend* bertujuan sebagai perantara yang menghubungkan aplikasi *client* dengan server *cloud messaging* dari Firebase, sekaligus melakukan filter perangkat tujuan notifikasi berdasarkan kompatibilitas golongan darah, sedangkan implementasi pada sisi *client* melibatkan implementasi *service* yang akan menangani pesan yang masuk.

4) Implementasi Formula Haversine

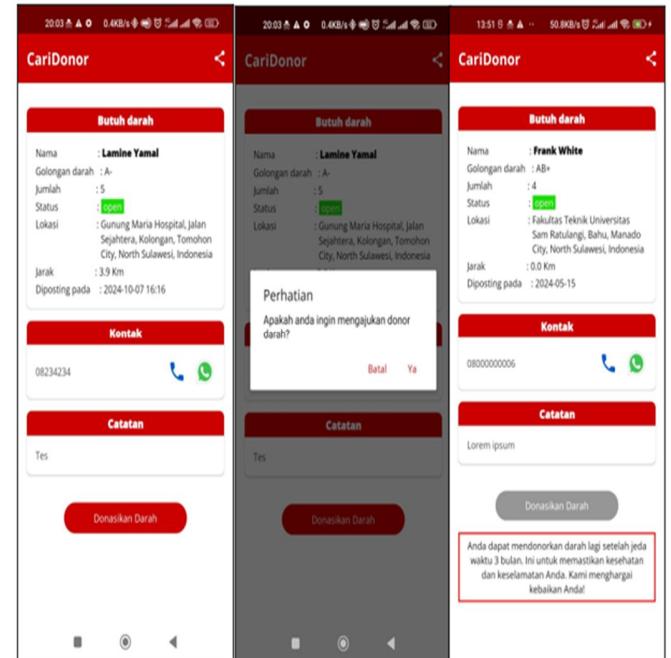
Implementasi formula *haversine* pertama-tama dilakukan dengan mendapatkan data koordinat (*latitude* dan *longitude*) dari pengguna dan alamat pondonoran yang tertera di permintaan kebutuhan darah. Titik koordinat pengguna didapatkan dengan mengakses lokasi GPS dari perangkat, sedangkan titik koordinat pasien didapatkan dengan mengonversi alamat pondonoran ke koordinat *latitude* dan *longitude*. Setelah melakukan konversi, titik koordinat dari pengguna dan pasien kemudian dapat dilakukan perhitungan jaraknya mengacu pada (1).

5) Implementasi Antarmuka Pengguna

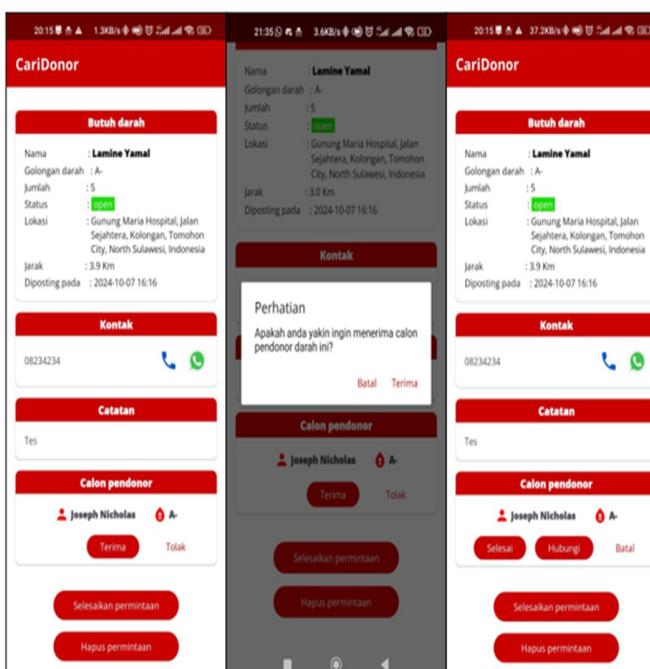
Tahap ini merupakan tahap di mana hasil desain sebelumnya yang berbentuk *wireframe* diimplementasikan menjadi bentuk yang lebih interaktif dan fungsional. Implementasi antarmuka pengguna ditulis dengan menggunakan bahasa XML melalui IDE Android Studio. Berikut adalah implementasi antarmuka pengguna dari aplikasi pencarian donor darah.



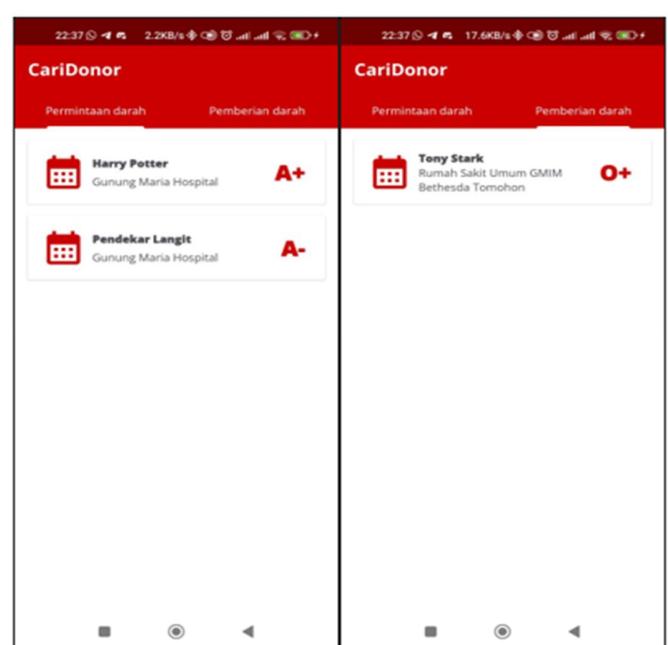
Halaman 13. Halaman daftar permintaan kebutuhan darah



Gambar 15. Halaman detail unggahan permintaan kebutuhan darah dari sisi pendoron



Gambar 14. Halaman detail permintaan butuh darah dari sisi pencari donor



Gambar 16. Halaman riwayat

D.Pengujian

Pengujian pada aplikasi pencarian pendonor darah dilakukan dengan menggunakan metode *Black box* dengan fokus utama yaitu pada sisi fungsional aplikasi. Terdapat tujuh fitur utama yang diuji pada tahap ini, yaitu fitur *login*, registrasi, daftar permintaan kebutuhan darah, membuat unggahan permintaan kebutuhan darah, mengajukan donasi darah, pemrosesan permintaan oleh pencari donor dan fitur riwayat.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil pembahasan dari pengembangan aplikasi mobile pencarian donor darah, maka dapat ambil kesimpulan bahwa aplikasi pencarian pendonor darah telah berhasil dikembangkan untuk perangkat Android sesuai dengan metode dan arsitektur yang telah ditetapkan, memudahkan pengguna untuk mencari donor serta menawarkan donasi darah dengan lebih cepat dan efisien.

Adapun saran yang dapat dipertimbangkan untuk pengembangan sistem yang serupa untuk penelitian selanjutnya adalah mengembangkan sistem pada sistem operasi lain selain Android, serta menerapkan konsep gamifikasi untuk meningkatkan keterlibatan dan motivasi pengguna untuk melakukan donor darah.

V. KUTIPAN

- [1] R. A. Abbasi *et al.*, “Saving lives using social media: Analysis of the role of twitter for personal blood donation requests and dissemination,” *Telematics and Informatics*, vol. 35, no. 4, pp. 892–912, Jul. 2018, doi: 10.1016/j.tele.2017.01.010.
- [2] M. Xin Ying and H. Hafit, “Emergency Notification and Healthcare Mobile Application for Elderly,” *Applied Information Technology And Computer Science*, vol. 3, no. 1, pp. 264–278, 2022, doi: 10.30880/aites.2022.03.01.017.
- [3] N. J. Stroud, C. Peacock, and A. L. Curry, “The Effects of Mobile Push Notifications on News Consumption and Learning,” *Digital Journalism*, vol. 8, no. 1, pp. 32–48, Jan. 2020, doi: 10.1080/21670811.2019.1655462.
- [4] M. Iqbal, P. Lestari, and N. Kurniati, “Buletin Sistem Informasi dan Teknologi Islam Penerapan Metode Haversine Pada Sistem Informasi Geografis Pencarian Lokasi Laundry Terdekat di Kota Makassar INFORMASI ARTIKEL ABSTRAK,” vol. 2, no. 1, pp. 12–16, 2021.
- [5] W. Wiharti, I. L. Rimra, and R. Putra, “Rancangan Dan Implementasi Aplikasi Mobile Tanggap Darurat Covid19 Berbasis Android Dan Gps,” *Seminar Nasional Informatika ...*, vol. 2020, no. Semnasif, 2020.
- [6] T. Ferga Prasetyo, D. Zalilluddin dan Rohmat, J. K. Abdul Halim, and J. Barat, *Implementasi Sistem Tanggap Darurat Bencana dengan... (Prasetyo dkk.) Implementasi Sistem Tanggap Darurat Bencana Dengan Menggunakan Firebase Cloud Messaging Berbasis Android Dan Sms Gateway*, vol. 1. 2019.
- [7] R. D. P. Simamora, M. S. Sudarma, and I. M. A. Suyadnya, “RANCANG BANGUN APLIKASI AMBULANCE ONLINE BERBASIS ANDROID,” *SINTECH (Science and Information Technology) Journal*, vol. 3, no. 2, 2020, doi: 10.31598/sintechjournal.v3i2.633.
- [8] M. E. Saifudin and H. Zakaria, “Rancang Bangun Aplikasi Antrian Secara Realtime Dengan Fitur Push Notification Menggunakan Smartphone Berbasis Android: Studi Kasus: Klinik Sandiana,” *LOGIC: Jurnal Ilmu Komputer ...*, vol. 1, no. 4, 2023.
- [9] M. Imron *et al.*, “Implementasi Push Notification Pada Sistem Peminjaman Sarana dan Prasarana Berbasis Website,” *JURNAL INFORMATIKA*, vol. 7, no. 2, pp. 174–182, 2020, [Online]. Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>
- [10] Ririn Feriana Basri and R. Rahmita, “PENYULUHAN PROSES DONOR DARAH DAN Pentingnya Donor Darah Sebagai Edukasi Pra-Donasi Pada Masyarakat Pattitanggang, Kecamatan Mappakasunggu, Kabupaten Takalar,” *Jurnal Abdimas Indonesia*, vol. 3, no. 3, 2023, doi: 10.53769/jai.v3i3.481.
- [11] L. Li *et al.*, “Static analysis of android apps: A systematic literature review,” 2017. doi: 10.1016/j.infsof.2017.04.001.
- [12] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global positioning system : theory and practice / B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins*. 2001.
- [13] M. Palash Uddin, M. Zahidul Islam, M. Nadim, and M. Ibn Afjal, “GPS-based Location Tracking System via Android Device,” 2013.
- [14] M. Pielot, A. Vradi, and S. Park, “Dismissed! A detailed exploration of how mobile phone users handle push notifications,” in *MobileHCI*



Joseph Julio Nicholas, lahir di Tomohon, pada tanggal 29 Juli 2002, sebagai anak tertua dari dua bersaudara. Penulis memulai perjalanan pendidikannya di TK Katolik St. Fransiskus Xaverius Kakaskasen pada 2007-2008. Penulis kemudian melanjutkan pendidikan ke jenjang sekolah dasar di SD Katolik St. Fransiskus Xaverius Kakaskasen pada 2008-2014, kemudian melanjutkan pendidikan ke jenjang sekolah menengah di SMP Katolik Gonzaga Tomohon pada 2014-2017, dan menempuh pendidikan menengah atas di SMA Katolik Karitas Tomohon pada 2017-2020. Penulis kemudian melanjutkan pendidikan tinggi di Universitas Sam Ratulangi Manado, mengambil program studi Informatika di bawah Jurusan teknik Elektro, Fakultas Teknik.