

SKRIPSI

APLIKASI PENCARIAN PETUGAS PMI TERDEKAT UNTUK
PENDONOR DARAH DAN INFORMASI STOK DARAH SECARA
REALTIME BERBASIS ANDROID



DISUSUN OLEH

Hafiz Dharmawan

145410112

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

AKAKOM

YOGYAKARTA

2018

HALAMAN JUDUL

APLIKASI PENCARIAN PETUGAS PMI TERDEKAT UNTUK PENDONOR DARAH DAN INFORMASI STOK DARAH SECARA REALTIME BERBASIS ANDROID

Diajukan sebagai salah satu syarat untuk menyelesaikan studi jenjang strata satu
(S1)

Program Studi Teknik Informatika

Sekolah Tinggi Manajemen Informatika dan Komputer AKAKOM

YOGYAKARTA

Disusun Oleh

Hafiz Dharmawan

145410112

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

AKAKOM

YOGYAKARTA

2018

HALAMAN PERSETUJUAN

Judul : Aplikasi Pencarian Petugas PMI Terdekat Untuk Pendoror Darah
Dan Informasi Stok Darah Secara Realtime Berbasis Android

Nama : Hafiz Dharmawan

N I M : 145410112

Jurusan : Teknik Informatika

Jenjang : Strata Satu (S-1)

Semester : Genap/TA.2018/2019

Telah diperiksa dan disetujui

Yogyakarta, 23 - 08 - 2018

Dosen Pembimbing,



(Wagito, S.T, M.T)

HALAMAN PENGESAHAN

SKRIPSI

APLIKASI Pencarian Petugas PMI Terdekat Untuk Pendonor Darah dan Informasi Stok Darah Secara Realtime Berbasis Android

Telah dipertahankan di depan Dewan Penguji Skripsi dan dinyatakan diterima untuk memenuhi sebagai syarat guna memperoleh Gelar Sarjana Komputer Sekolah Tinggi Manajemen Informatika dan Komputer

YOGYAKARTA

Yogyakarta, 23 - 08 - 2018

Mengesahkan

1. Wagito, S.T, M.T.
2. Indra Yatini, S.Kom., M.Kom.
3. LN Harnaningrum S.Si., M.T.

Mengetahui

Ketua Program Studi Teknik Informatika

23 AUG 2018



Dini Pakta Sari, S.T., M.T.

HALAMAN PERSEMBAHAN

Alhamdulillahirobbil'alamin

Puji Syukur kehadiran Allah Subhanahu Wa Ta'ala atas taburan kasih sayang, rizki, nikmat dan karunia-Nya.

Engkau berikan berbagai kemudahan dan kekuatan, sehingga karya ini dapat terselesaikan dengan baik.

Kupersembahkan karyaku ini untuk ...

Ibundaku tersayang Astuti Suheryani dan Ayahanda tercinta Susmantoro, serta adikku Sheryl Evania yang selalu membantuku, mendukungku, memotivasiku dan mendoakanku setiap waktu.

Keluargaku ... semoga kita selalu dilindungi oleh Allah Subhanahu Wa Ta'ala sebagaimana Ia melindungi orang-orang yang Ia lindungi.

Teman main, teman di perum pringgading permai, teman seperjuangan TI-3 Mbak Olen, Heppy, Jati, Shinta, Agata, Anis, Silpi, Junior, Bastian dan teman gabah yang super Iwan, Zaky, Frendi, Riansyah serta semuanya yang tidak dapat ditulis satu persatu, terima kasih atas motivasinya, dukungannya dan kebersamaanya semoga kita dapat bertemu kembali dilain waktu.

Semua pihak yang telah membantu baik materi, tenaga dan doa.

Semoga kita semua selalu dalam lindungan-Nya.

HALAMAN MOTTO

“Bekerja keras dan bersikap baiklah. Hal luar biasa akan terjadi.”

-Conan O’ Brien-

“Ubah pikiranmu dan kau mengubah duniamu.”

-Norman Vincent Peale-

“Hidup dapat dipahami dengan berpikir ke belakang. Tapi ia juga harus dijalani dengan berpikir ke depan.

-Soren Kierkega-

INTISARI

Menurut data yang ada pada tahun 2013, sekitar 17 % dari orang-orang yang tidak mau menderma darahnya beralasan tidak pernah memikirkan donor darah, dan sekitar 15 % masyarakat menyatakan terlalu sibuk sehingga tidak berkesempatan menderma darahnya. Karena terbatasnya stok darah membuat resipien (penerima donor) yang membutuhkan darah menjadi susah mendapatkan transfusi darah. Untuk memenuhi kekurangan stok darah, PMI mengirimkan SMS ke 1.500 nomor telepon genggam pendonor yang terdaftar setiap hari. Fakta tersebut menunjukkan kesulitan petugas PMI untuk mencari pendonor darah agar stok darah di PMI tidak mengalami kekurangan.

Sistem yang dibangun berbasis mobile dan web dengan bahasa pemrograman *Java*, *Javascript* menggunakan basis data real time *Firebase*. Dalam penelitian ini *firebase* digunakan untuk informasi stok darah dan pencarian lokasi petugas PMI secara real time. Untuk melakukan pencarian lokasi petugas, diperlukan data lokasi *latitude longitude* dari pendonor dan petugas. Pencarian tersebut menggunakan *library Firebase* yaitu *GeoFire*. Selain itu sistem ini menggunakan peta *Mapbox* untuk menampilkan peta dan navigasi.

Hasil penelitian ini menunjukkan bahwa aplikasi mampu melakukan pencarian petugas PMI terdekat dalam radius 10km dan menampilkan informasi stok darah secara real *time*. Serta dapat melakukan navigasi peta menuju lokasi petugas PMI.

Kata Kunci : *firebase, donor darah, android, java, javascript, geofire*

KATA PENGANTAR

Alhamdulillah Robbil ‘Alamiin, Puji dan syukur penulis panjatkan kehadiran Allah Ta’ala, berkat kasih dan karunia-Nya sehingga penulis mampu menyelesaikan penyusunan skripsi yang berjudul “Aplikasi Pencarian Petugas PMI Terdekat Untuk Pendorong Darah Dan Informasi Stok Darah Secara Realtime Berbasis Android”. Sholawat dan salam semoga senantiasa tercurahkan kepada Nabi Muhammad Shallallahu Alaihi Wa Sallam, keluarganya, para sahabatnya, dan umatnya yang senantiasa mengikuti petunjuknya hingga akhir zaman.

Penyusunan skripsi ini tidak terlepas dari bantuan berupa bimbingan, saran, dukungan, dan semangat dari berbagai pihak, maka penulis mengucapkan terimakasih kepada:

1. Bapak Ir. Totok Suprawoto, M.M., M.T. selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer AKAKOM Yogyakarta,
2. Ibu Dini Fakta Sari, S.T., M.T.,, selaku Ketua Program Studi Teknik Informatika Jenjang Strata Satu (S1) STMIK AKAKOM Yogyakarta yang telah memberikan izin penelitian.
3. Bapak Wagito, S.T, M.T. selaku dosen pertama yang mengenalkan penulis mengenai android sekaligus pembimbing yang telah memberikan

bimbingan, pengarahan, dan semangat dalam penyusunan tugas akhir ini.

4. Ibu Indra Yatini, S.Kom., M.Kom. dan LN Harnaningrum S.Si., M.T. yang telah banyak memberikan nasehat, arahan, motivasi dan masukan selama penelitian ini.
5. Kedua orang tua dan seluruh keluarga yang telah memberikan doa dan dukungannya selama ini.
6. Semua pihak yang telah membantu dalam penyusunan skripsi ini dan tidak dapat penulis sebutkan satu per satu.

Semoga bantuan yang diberikan selama penelitian hingga terselesaikannya skripsi ini mendapatkan balasan kebaikan dari Allah Subhanahu Wa Ta'ala. Penulis menyadari bahwa skripsi ini belum sempurna, sehingga penulis mengharapkan kritik dan saran yang membangun. Semoga skripsi ini bermanfaat bagi penulis dan pembaca.

Yogyakarta, 9 Juli 2018

Penulis,

DAFTAR ISI

COVER.....	i
HALAMAN JUDUL.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO.....	vi
INTISARI.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Ruang Lingkup.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	6
2.1 Tinjauan Pustaka.....	6

2.2	Dasar Teori.....	92.2.1
	PMI	9
2.2.2	Android	9
2.2.3	Firestore	10
2.2.4	JSON.....	12
2.2.5	Android Studio.....	13
2.2.6	Mapbox	13
2.2.7	Rational Rose	14
BAB III	METODE PENELITIAN.....	15
3.1	Analisis Kebutuhan.....	15
3.1.1	Analisis Kebutuhan Masukan	15
3.1.2	Analisis Kebutuhan Proses	15
3.1.3	Analisis Kebutuhan Keluaran	16
3.1.4	Analisis Perangkat Keras (<i>Hardware</i>)	16
3.1.5	Analisis Perangkat Lunak (<i>Software</i>)	16
3.2	Perancangan Sistem	17
3.2.1	Use Case Diagram.....	17
3.2.2	Activity Diagram	18
3.2.3	Sequence Diagram	21
3.3	Perancangan Antarmuka	26
3.4	Perancangan Tabel.....	28
3.5	Arsitektur Teknologi.....	31
BAB IV	IMPLEMENTASI DAN PEMBAHASAN.....	32

4.1	Implementasi Dan Pengujian.....	32
4.2	Pembahasan.....	44
BAB V	PENUTUP.....	53
5.1	Kesimpulan	53
5.2	Saran	53
	DAFTAR PUSTAKA.....	55

DAFTAR GAMBAR

Gambar 3.1 Use Case Diagram.....	17
Gambar 3.2 Activity Diagram Pendonor	18
Gambar 3.3 Activity Diagram Petugas	19
Gambar 3.4 Activity Diagram Admin.....	20
Gambar 3.5 Sequence Diagram Login Pendonor.....	21
Gambar 3.6 Sequence Diagram Pendonor Mencari Petugas Terdekat	22
Gambar 3.7 Sequence Diagram Pendonor Melihat Informasi Stok Darah ...	23
Gambar 3.8 Sequence Diagram Login Petugas	24
Gambar 3.9 Sequence Diagram Petugas Dicari Pendonor.....	25
Gambar 3.10 Sequence Diagram Admin Melihat Grafik Donor Darah	26
Gambar 3.11 Login Aplikasi.....	27
Gambar 3.12 Halaman Utama Pendonor	27
Gambar 3.14 Profil Petugas	27
Gambar 3.13 Halaman Utama Petugas	27
Gambar 3.15 Arsitektur Teknologi	31
Gambar 4.1 Mendaftarkan Aplikasi <i>Android</i> Ke <i>Firestore Console</i>	34
Gambar 4.2 Script Pada Build Gradle.....	34
Gambar 4.3 Script Pada App/build.gradle	35
Gambar 4.4 Script Menghubungkan Firestore Dan Aplikasi Web.....	35
Gambar 4.5 Fungsi aktif/tidak aktif	36

Gambar 4.6 Fungsi Untuk Mencari Petugas	37
Gambar 4.7 <i>Callback Query</i> Pencarian Petugas	38
Gambar 4.8 Listener Untuk Mendapatkan Notifikasi	39
Gambar 4.9 Fungsi Update Stok Darah Pada Admin	40
Gambar 4.10 Fungsi <i>Push</i> Darah Pada Aplikasi Petugas	41
Gambar 4.11 Fungsi Dan <i>Callback</i> Menampilkan Stok Darah	41
Gambar 4.12 Builder untuk mendapatkan rute	42
Gambar 4.13 Fungsi Untuk Menampilkan Grafik	43
Gambar 4.14 Tampilan <i>Splashscreen</i>	44
Gambar 4.15 Tampilan Aplikasi Pendonor.....	45
Gambar 4.16 Lokasi Petugas PMI	46
Gambar 4.17 Aplikasi Pendonor Setelah Pencarian	47
Gambar 4.18 Tampilan Aplikasi Petugas	48
Gambar 4.19 Tampilan <i>Popup</i> Aplikas Petugas	49
Gambar 4.20 Aplikasi Petugas Setelah Pencarian	50
Gambar 4.21 Input Golongan Darah.....	51
Gambar 4.22 Tampilan Interface Web Admin.....	52

DAFTAR TABEL

Tabel 2.1 Perbedaan dan Tinjauan Pustaka	6
Tabel 2.2 Api Level Versi Code dan Distribusi <i>Android</i>	10
Tabel 2.3 Cara Menulis Data Di Firebase.....	11
Tabel 2.4 <i>Listener</i> Untuk Membaca Data.....	12
Tabel 3.1 Struktur Tabel Pendonor.....	28
Tabel 3.2 Struktur Tabel Petugas.....	28
Tabel 3.3 Struktur Tabel Pencarian	29
Tabel 3.5 Struktur Tabel Geo Pendonor	29
Tabel 3.5 Struktur Tabel Geo Petugas	30
Tabel 3.7 Struktur Tabel Stok Darah	30

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Di era zaman sekarang ini jumlah pemakaian darah lebih besar daripada jumlah pendonor darah. Banyak masyarakat yang tidak tahu informasi dan lokasi pos PMI sehingga jumlah pendonor menjadi sedikit. Belum lagi masyarakat yang siap donor darah namun sibuk akan aktivitasnya sehingga tidak sempat pergi ke pos PMI. Hal tersebut mengakibatkan stok kantong darah di PMI terbatas. Menurut data yang ada pada tahun 2013, sekitar 17 % dari orang-orang yang tidak mau menderma darahnya beralasan tidak pernah memikirkan donor darah, dan sekitar 15 % masyarakat menyatakan terlalu sibuk sehingga tidak berkesempatan menderma darahnya (Komandoko, 2013).

Karena terbatasnya stok darah membuat resipien (penerima donor) yang membutuhkan darah menjadi susah mendapatkan transfusi darah. Selain itu untuk mengecek informasi stok darah, keluarga resipien harus datang ke pos PMI untuk menanyakan stok darah yang ada. Sedangkan resipien sangat membutuhkan darah tersebut dan harus ditangani dengan cepat.

Untuk memenuhi kekurangan stok darah, PMI mengirimkan SMS ke 1.500

nomor telepon genggam pendonor yang terdaftar setiap hari. Fakta tersebut menunjukkan kesulitan petugas PMI untuk mencari pendonor darah agar stok darah di PMI tidak mengalami kekurangan. Hal yang biasa dilakukan oleh petugas PMI untuk meningkatkan stok darah yaitu menunggu diadakannya *event* donor darah.

Pengguna *smartphone* di Indonesia sangat tinggi . Menurut Guntur Sanjoyo bahwa Indonesia memiliki populasi *online* sangat aktif dimana rata-rata 5,5 jam sehari dihabiskan untuk mengakses sekitar 46 aplikasi dan domain web melalui *smartphone* mereka setiap hari (Sukma , 2016).

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas maka dibuatlah rumusan masalah yaitu membuat aplikasi pencarian petugas PMI terdekat untuk pendonor darah dan informasi stok darah secara *realtime* berbasis *Android*.

1.3 Ruang Lingkup

Adapun ruang lingkup yang digunakan sebagai berikut :

- a) Aplikasi dapat digunakan oleh pendonor darah, admin PMI dan petugas PMI.
- b) Aplikasi menggunakan layanan peta dari *Mapbox*.

- c) Batas maksimal radius pencarian adalah 10 km.
- d) Penanda GPS petugas PMI dapat dilihat oleh pendonor.
- e) Penanda GPS pendonor darah dapat dilihat oleh petugas PMI secara *real-time* apabila pendonor sedang dalam perjalanan menuju lokasi pendonor darah.
- f) Admin PMI dapat melihat grafik informasi donor darah.
- g) Admin dan petugas PMI dapat melakukan *input* stok darah.
- h) Pendonor dapat melihat informasi stok darah.
- i) Aplikasi dibangun pada sistem operasi *Android* menggunakan layanan *firebase*.
- j) Aplikasi dijalankan pada *Android* dengan minimal versi 6.0 *Marshmallow*.

1.4 Tujuan Penelitian

Tujuan dari pembuatan sistem ini adalah untuk menghasilkan sebuah perangkat lunak aplikasi pencarian petugas PMI untuk pendonor darah yang beroperasi pada sistem *Android* menggunakan *Firebase*.

1.5 Manfaat Penelitian

Manfaat dari dibuatnya sistem ini yaitu membantu pengguna yang ingin mencari informasi stok darah dan membantu petugas PMI untuk mencari

pendonor darah.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi dengan judul “APLIKASI PENCARIAN PETUGAS PMI TERDEKAT UNTUK PENDONOR DARAH DAN INFORMASI STOK DARAH SECARA REALTIME BERBASIS ANDROID” disusun guna memberi gambaran umum tentang penelitian yang dilakukan. Sistematika penulisan skripsi ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Menguraikan tentang pokok yang menjadi latar belakang permasalahan yang diambil, menguraikan rumusan masalah yang dihadapi, menentukan ruang lingkup masalah untuk membatasi aplikasi, menentukan tujuan, manfaat dan kegunaan sistem, serta sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA DAN DASAR TEORI

Membahas mengenai tinjauan pustaka yaitu mengacu penelitian-penelitian yang ada sebelumnya dengan meninjau kelebihan dan kekurangan dari penelitian tersebut sehingga dapat digunakan sebagai referensi. Sedangkan dasar teori berisi konsep dasar serta teori-teori yang berkaitan dengan topik penelitian yang digunakan untuk mendukung proses analisis permasalahan. Selain itu memberikan

gambaran teknologi-teknologi yang digunakan sebagai pendukung penelitian yang akan dilakukan.

BAB 3 METODE PENELITIAN

Menganalisis sistem dalam aplikasi yang dibangun dimulai dari aspek-aspek yang berkaitan serta merancang sistem dimulai dari segi analisis kebutuhan, terdiri dari kebutuhan masukan, proses, keluaran, perangkat keras, dan perangkat lunak, selanjutnya berisi pemodelan-pemodelan dengan diagram-diagram, sampai berisi desain tampilan yang dapat mempermudah pengguna layanan dalam penggunaan sistem.

BAB 4 IMPLEMENTASI DAN PEMBAHASAN

Berisi kesimpulan serta menjawab permasalahan yang dihadapi sampai mengetahui keunggulan dan kekurangan dari sistem yang dirancangan dengan yang sudah ada, serta hasil implementasi diikuti pengujian telah mencapai tujuan yang diinginkan dalam pembuatan aplikasi ini.

BAB 5 PENUTUP

Berisi kesimpulan serta menjawab permasalahan yang dihadapi sampai mengetahui keunggulan dan kekurangan dari sistem yang dirancangan dengan yang sudah ada, serta hasil implementasi diikuti pengujian telah mencapai tujuan yang diinginkan dalam pembuatan aplikasi ini.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Beberapa acuan yang digunakan dalam pengembangan sistem dan aplikasi ini ditampilkan pada tabel 2.1 di bawah ini :

Tabel 2.1 Perbedaan dan Tinjauan Pustaka

No	Penulis	Objek	Teknologi	Pelacakan <i>Real-time</i>	Bahasa Pemrograman
1	Moh., Mustofa Ali (2015)	Pencarian Lokasi Hotel	<i>Webservice</i>	Tidak	Java, PHP
2	Nova Agustina dan Slamet Risnanto (2016)	Pencarian Lokasi Wisata Kota Cimahi	<i>Webservice</i>	Tidak	Jquery, Javascript, Node.JS
2	Buru, Dadang Juwoto (2017)	Perangkat Bergerak	<i>Firebase</i>	Ya	Java, JSON
3	Fadillah (2017)	Pencarian Lokasi Kendaraan Antar Jemput	<i>Webservice</i>	Ya	Java, PHP
4	Wicaksono (2017)	Pencarian Lokasi Apotek	<i>Webservice</i>	Tidak	Java, PHP
5	Hafiz Dharmawan (2018)	Pencarian Petugas PMI & Pendonor Darah	<i>Firebase</i>	Ya	Java, JSON

Pada tahun 2015 pernah dilakukan penelitian dengan judul “Pencarian Lokasi Hotel Berbasis *Android* Menggunakan Metode *Location Based Service* (Studi Kasus Kota Semarang)” yang dibuat oleh Mustofa Ali Moh dari Universitas Dian Nuswantoro Semarang. Aplikasi tersebut dapat melakukan pencarian lokasi hotel terdekat di kota Semarang. Teknologi yang digunakan yaitu web servis untuk pengambilan dan pengiriman data.

Pada tahun 2016 pernah dilakukan penelitian dengan judul “Aplikasi *Location Based Service* Untuk Informasi Dan Pencarian Lokasi Pariwisata Di Kota Cimahi Berbasis *Android*” yang dibuat oleh Nova Agustina dan Slamet Risnanto dari Teknik Informatika Sekolah Tinggi Teknologi Bandung. Aplikasi tersebut dapat melakukan pencarian lokasi wisata yang berada di kota Cimahi dan menentukan rute jalan terdekat. Teknologi yang digunakan yaitu *Node.js* sebagai *backend* dan menggunakan *framework phonegap* untuk membangun *frontend* berbasis *Android*.

Pada tahun 2017 dilakukan penelitian dengan judul “Aplikasi Pendeteksi Lokasi Perangkat Bergerak Menggunakan Teknologi *Cloud Computing* Dengan *Firebase Relatime Database* Berbasis *Android*” sebuah aplikasi real-time berbasis android untuk pelacakan perangkat bergerak” oleh Dadang Juwoto dari STMIK Akakom Yogyakarta. Aplikasi ini melakukan pelacakan lokasi pada perangkat bergerak secara real-time untuk dibagikan kepada pengguna yang dapat menolong

pengguna. Aplikasi ini menggunakan layanan *Firebase* untuk kebutuhan berbagi lokasi secara *realtime*.

Pada tahun 2017 dilakukan penelitian dengan judul “Aplikasi Sistem Pelacakan Lokasi Kendaraan Antar Jemput Anak Sekolah Berbasis *Android*” oleh Fadillah dari Universitas Bandar Lampung. Aplikasi ini dapat melacak kendaraan pada saat menjemput anak sekolah secara *realtime*. Teknologi yang digunakan adalah web servis REST API dan menggunakan *looping* pada *Android* untuk mengirimkan lokasi *latitude* dan *longitude* secara berkala.

Pada tahun 2017 dilakukan penelitian dengan judul “Aplikasi Pencarian Lokasi Apotek Di Kabupaten Sleman” oleh Cahyo Wicaksono dari Universitas Sanata Darma Yogyakarta. Aplikasi ini dapat melakukan pencarian apotek terdekat di wilayah Sleman. Teknologi yang digunakan adalah web servis REST API. Aplikasi ini juga dapat membantu pengguna untuk mencari obat lebih efisien.

Sedangkan pada penelitian ini, aplikasi akan dikembangkan untuk melakukan pencarian petugas PMI dan pendonor darah untuk kebutuhan permintaan donor darah. Aplikasi ini menggunakan *Firebase* yang digunakan untuk melakukan pelacakan dan memberikan informasi stok darah secara *real-time*. Aplikasi ini diberikan fitur pencarian lokasi petugas PMI terdekat dengan radius 10 km.

2.2 Dasar Teori

2.2.1 PMI

PMI atau Palang Merah Indonesia adalah suatu organisasi netral dan independen yang ada di Indonesia dan bergerak di bidang kemanusiaan. PMI terbentuk atas dasar bangsa Indonesia sendiri meski dipengaruhi oleh palang merah yang sikapnya universal. Terbentuknya PMI bermula dari dorongan jiwa manusia dan atas ke dasaran nasional. Ketika melaksanakan kegiatan PMI selalu menjaga prinsip yang dimilikinya yaitu kemanusiaan, kenetralan, sukarela, kemandirian, kesemestaan, dan kesatuan (Bobsusanto , 2016).

2.2.2 Android

Android adalah nama resmi dari sistem operasi atau OS (Operating System) yang berbasis dari *kernel Linux*. Sistem operasi ini banyak digunakan pada perangkat bergerak seperti ponsel cerdas atau *smartphone* yang didukung dan dikembangkan oleh Google (Gunawan , 2017).

Android memiliki *code* versi dan API (*Application Programming Interface*) level yang berbeda-beda. API Level adalah nilai integer yang secara unik mengidentifikasi revisi API kerangka kerja yang ditawarkan oleh versi platform *Android*. Penulis akan menggunakan *Android* versi 6.0 *Marshmallow* karena merupakan versi *Android* yang terbanyak digunakan pada saat ini. Berikut

distribusi versi *android* menurut tabel 2.2 berikut.

Tabel 2.2 Api Level Versi Code dan Distribusi *Android*

No	Version Platform	Api Level	Version Code	Distribusi
1	Android 2.3.7	10	GINGERBREAD	0.3%
2	Android 4.0	15	ICE CREAM SANDWICH	0.4%
3	Android 4.1	16	JELLY BEAN	1.5%
4	Android 4.2	17		2.2%
5	Android 4.3	18		0.6%
6	Android 4.4	19	KIT KAT	10.3%
7	Android 5.0	21	LOLLIPOP	22.4%
8	Android 6.0	23	MARSHMALLOW	25.5%
9	Android 7.0	24	NOUGAT	22.9%
10	Android 7.1.1	25		8.2%
11	Android 8.0	26	OREO	4.9%

(Android , 2018)

2.2.3 Firebase

Firebase adalah *realtime database* yang di-host di *cloud*. Data disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika Anda membuat aplikasi lintas-platform dengan SDK *Android*, *iOS*, dan *JavaScript*, semua klien akan berbagi sebuah *instance realtime database* dan

menerima update data terbaru secara otomatis (Firebase , 2018).

Salah satu kemampuan utama *Firestore* yaitu bersifat *real-time*. Sebagai ganti permintaan HTTP biasa, *Firestore Realtime Database* menggunakan sinkronisasi data—setiap kali data berubah, semua perangkat yang terhubung akan menerima update dalam waktu milidetik. Memberikan pengalaman yang kolaboratif dan *immersif* tanpa perlu memikirkan kode jaringan (Firebase , 2018).

Firestore merupakan database *NoSQL*, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan database terkait. API *real-time Database* dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan anda untuk membangun pengalaman *real-time* yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan tersebut. Berikut cara menulis data di *Firestore* :

Tabel 2.3 Cara Menulis Data Di Firestore

No	Method	Penjelasan
1	set	Menulis atau mengganti data ke lokasi yang ditetapkan, seperti messages/users/<username>
2	update	Mengupdate beberapa kunci untuk lokasi yang ditetapkan tanpa mengganti semua data.
3	push	Menambahkan ke daftar data dalam database. Setiap kali mendorong node baru ke dalam daftar, database menghasilkan kunci unik,

		seperti messages/users/<unique-user-id>/<username>
4	transaction	Gunakan fitur transaction ketika bekerja dengan data kompleks yang bisa rusak karena update bersamaan

(Firebase , 2018).

Adapun cara untuk membaca data atau *listener* peristiwa untuk ketika terdapat perubahan data:

Tabel 2.4 *Listener* Untuk Membaca Data

No	Method Callback	Penjelasan
1	ValueChanged	Membaca dan mendeteksi perubahan di seluruh konten lokasi.
2	ChildAdded	Mengambil daftar item atau mendeteksi penambahan daftar item. Disarankan untuk digunakan dengan ChildChanged dan ChildRemoved untuk memantau perubahan daftar.
3	ChildChanged	Mendeteksi perubahan pada item dalam daftar. Gunakan dengan ChildAdded dan ChildRemoved untuk memantau perubahan daftar.
4	ChildRemoved	Mendeteksi item yang sedang dihapus dari daftar. Gunakan dengan ChildAdded dan ChildChanged untuk memantau perubahan daftar.
5	ChildMoved	Mendeteksi perubahan pada urutan item dalam urutan daftar. Peristiwa ChildMoved selalu mengikuti peristiwa ChildChanged, yang menyebabkan urutan item berubah (berdasarkan metode 'urutkan berdasarkan' saat ini).

(Firebase , 2018).

2.2.4 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan

dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, *JavaScript*, *Perl*, *Python* (Json , 2018).

2.2.5 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi *Android*, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, *Android studio* juga dibuat untuk mempercepat pengembangan dan membantu Anda membuat aplikasi berkualitas tinggi untuk setiap perangkat *Android* (Android , 2018).

2.2.6 Mapbox

Mapbox adalah platform data lokasi untuk aplikasi seluler dan web. *Mapbox* menyediakan blok bangunan untuk menambahkan fitur lokasi seperti peta, penelusuran, dan navigasi ke setiap pengalaman yang anda buat. *Mapbox* mengubah cara orang bergerak di sekitar kota dan menjelajahi dunia kita. *Mapbox* menjangkau lebih dari 300 juta orang setiap bulan. (Mapbox, 2018)

2.2.7 Rational Rose

Rational Rose adalah *tools* permodelan visual untuk pengembangan system berbasis objek yang handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan system. *Rational Rose* mendukung permodelan bisnis yang membantu para pengembang memahami system secara komprehensif. Aplikasi ini juga membantu analisis system dengan cara pengembang membuat diagram *use case* untuk melihat fungsionalitas system secara keseluruhan sesuai dengan harapan dan keinginan pengguna (Mulyadi , 2013).

BAB III

METODE PENELITIAN

3.1 Analisis Kebutuhan

3.1.1 Analisis Kebutuhan Masukan

Kebutuhan masukan merupakan sekumpulan data yang akan diproses oleh sistem. Adapun kebutuhan masukan yang dibutuhkan oleh sistem ini yaitu :

- a. Data nama, nomor telepon, dan email dari petugas PMI.
- b. Data nama, nomor telepon, dan email dari pengguna aplikasi pendonor.
- c. Foto petugas PMI.
- d. Koordinat Lokasi.
- e. Data darah yang diambil.

3.1.2 Analisis Kebutuhan Proses

Kebutuhan proses merupakan data yang akan diproses oleh sistem. Adapun kebutuhan proses dalam aplikasi ini yaitu :

- a. Proses Login.
- b. Proses Pelacakan lokasi petugas PMI / pendonor.
- c. Kirim data ke cloud firebase.

3.1.3 Analisis Kebutuhan Keluaran

Kebutuhan keluaran merupakan hasil dari data masukan yang telah diproses oleh sistem. Adapun keluaran yang dihasilkan oleh aplikasi ini yaitu :

- a. Grafik informasi donor darah.
- b. Informasi stok darah.
- c. Pelacakan lokasi terdekat petugas PMI dengan radius 10 km.

3.1.4 Analisis Perangkat Keras (*Hardware*)

Perangkat keras merupakan perangkat yang dibutuhkan untuk membuat aplikasi ini. Perangkat keras dengan spesifikasi sebagai berikut :

- a. Laptop dengan prosesor 2.50GHz.
- b. RAM 4,00GB.
- c. Android version 6.0.

3.1.5 Analisis Perangkat Lunak (*Software*)

Perangkat lunak merupakan perangkat yang digunakan untuk membuat dan menjalankan aplikasi ini. Adapun perangkat lunak yang digunakan adalah :

- a. Sistem Operasi *MacOS Sierra*.
- b. *Android Studio*.

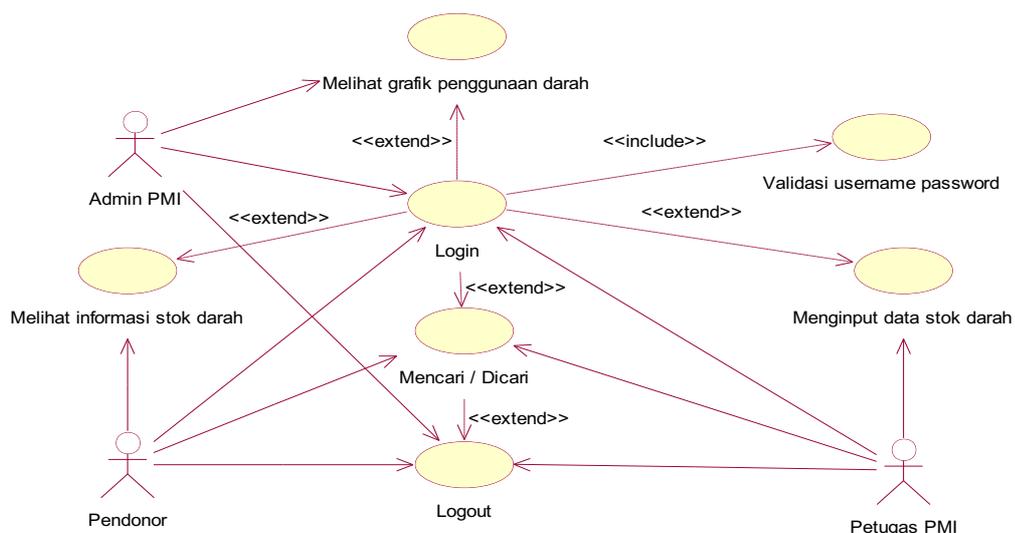
- c. Xml.(*Extensible Markup Language*)
- d. SDK (*Standard Development Kit*) Android.
- e. JDK 8 (*Java Developmen Kit*).

3.2 Perancangan Sistem

Perancangan sistem merupakan garis besar dari seluruh proses komputerisasi pengolahan data yang akan dilakukan. Perancangan ini mencakup bagaimana sistem tersebut dapat dijalankan, masalah yang dihadapi, langkah pemrosesan datanya, serta hasil keluarannya.

3.2.1 Use Case Diagram

Use Case Diagram merupakan diagram yang menggambarkan perilaku sistem yang akan dibuat.

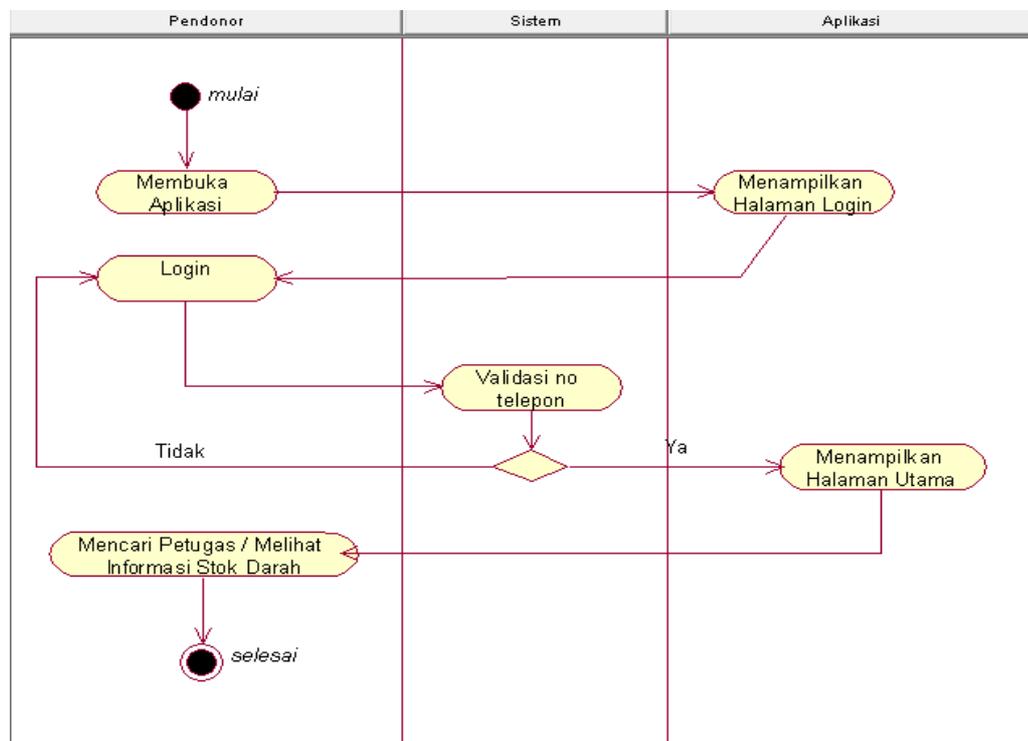


Gambar 3.1 Use Case Diagram

Deskripsi Gambar 3.1 : Terdapat 3 aktor yaitu Admin PMI, Pendonor dan Petugas PMI. Untuk Admin PMI dapat melakukan *login*, melihat grafik donor darah dan *logout*. Petugas PMI dapat melakukan *login*, dicari oleh pendonor, menginput data darah pendonor dan *logout*. Pendonor dapat melakukan *login*, melihat informasi stok darah, mencari petugas PMI, dan *logout*.

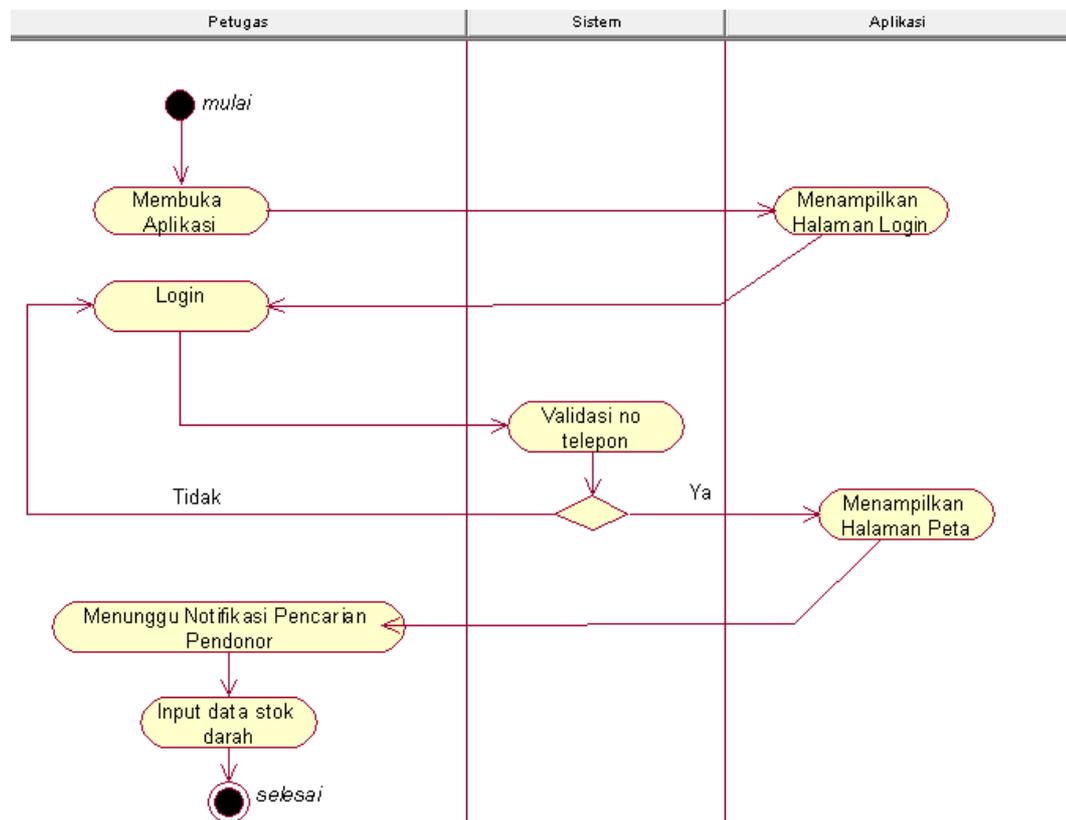
3.2.2 Activity Diagram

Activity Diagram merupakan diagram yang menggambarkan tentang alir kerja/aktivitas sistem dari aktivitas awal hingga aktivitas akhir.



Gambar 3.2 Activity Diagram Pendonor

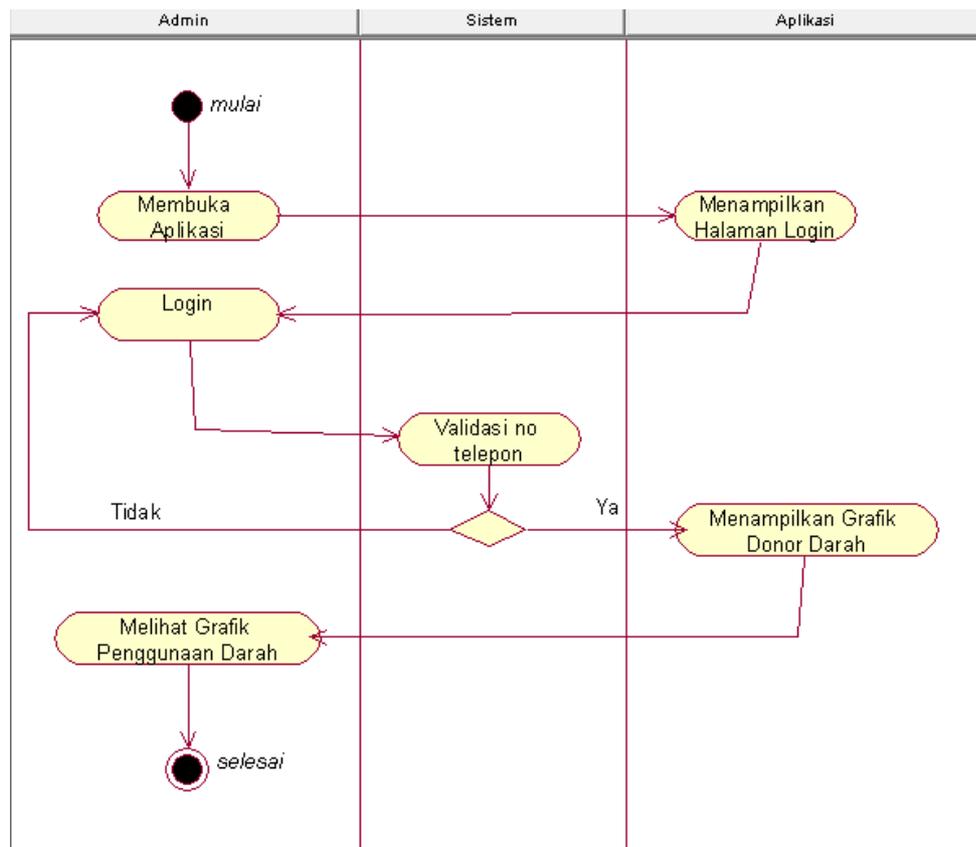
Deskripsi Gambar 3.2 : Activity diagram tersebut memiliki 3 swimlane yaitu Pendoron , Sistem dan Aplikasi. Pertama pendonor membuka aplikasi, lalu aplikasi menampilkan halaman *login*. Pendoron melakukan *login* dan divalidasi apabila valid maka aplikasi menampilkan halaman utama jika tidak, maka kembali *login*. Setelah *login* pendonor dapat mencari petugas atau melihat informasi stok darah.



Gambar 3.3 Activity Diagram Petugas

Deskripsi gambar 3.3 : Activity diagram tersebut memiliki 3 *swimlane* yaitu Petugas , Sistem dan Aplikasi. Pertama petugas membuka aplikasi, lalu aplikasi

menampilkan halaman *login*. Petugas melakukan *login* dan di validasi apabila valid maka aplikasi menampilkan halaman peta jika tidak maka kembali *login*. Setelah *login* petugas menunggu notifikasi pencarian dan *input* data stok darah.

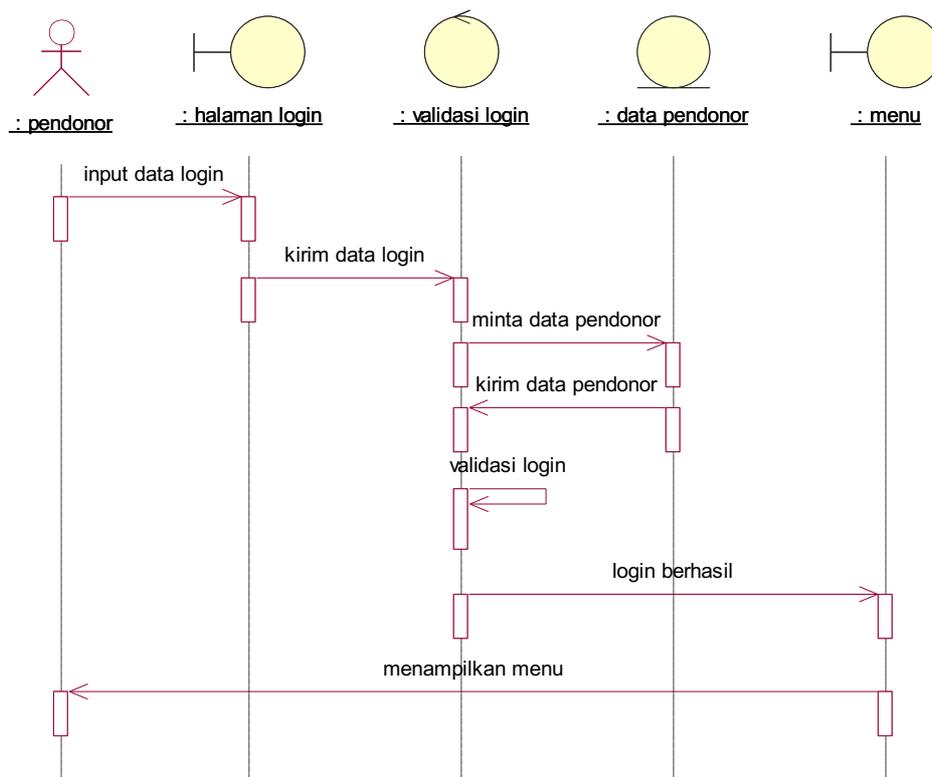


Gambar 3.4 Activity Diagram Admin

Deskripsi Gambar 3.4 : *Activity* diagram tersebut memiliki 3 *swimlane* yaitu Admin , Sistem dan Aplikasi. Pertama admin membuka aplikasi, lalu aplikasi menampilkan halaman login. Admin melakukan login dan di validasi apabila valid maka aplikasi menampilkan grafik informasi donor darah jika tidak maka kembali login. Setelah login admin dapat melihat grafik tersebut.

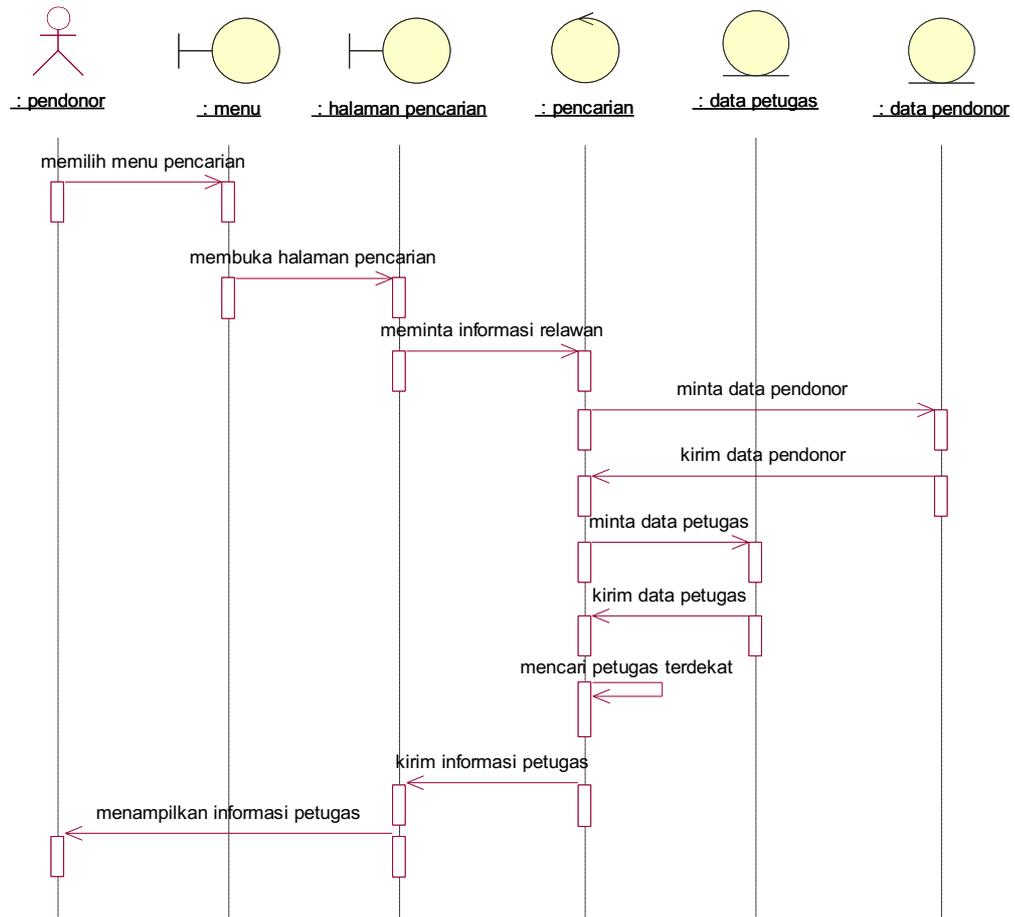
3.2.3 Sequence Diagram

Sequence Diagram merupakan diagram yang menggambarkan interaksi antar objek di dalam sebuah sistem yang akan dibuat



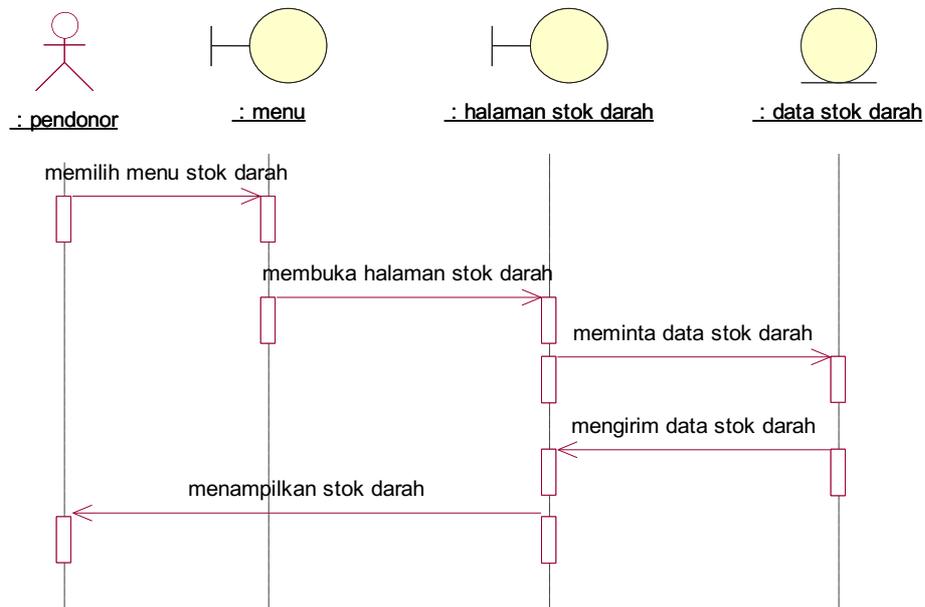
Gambar 3.5 Sequence Diagram Login Pendonor

Deskripsi Gambar 3.5 : Pendonor melakukan login dengan menginputkan data login yang kemudian divalidasi dengan data pendonor dari entitas database. Jika berhasil maka tampilkan menu.



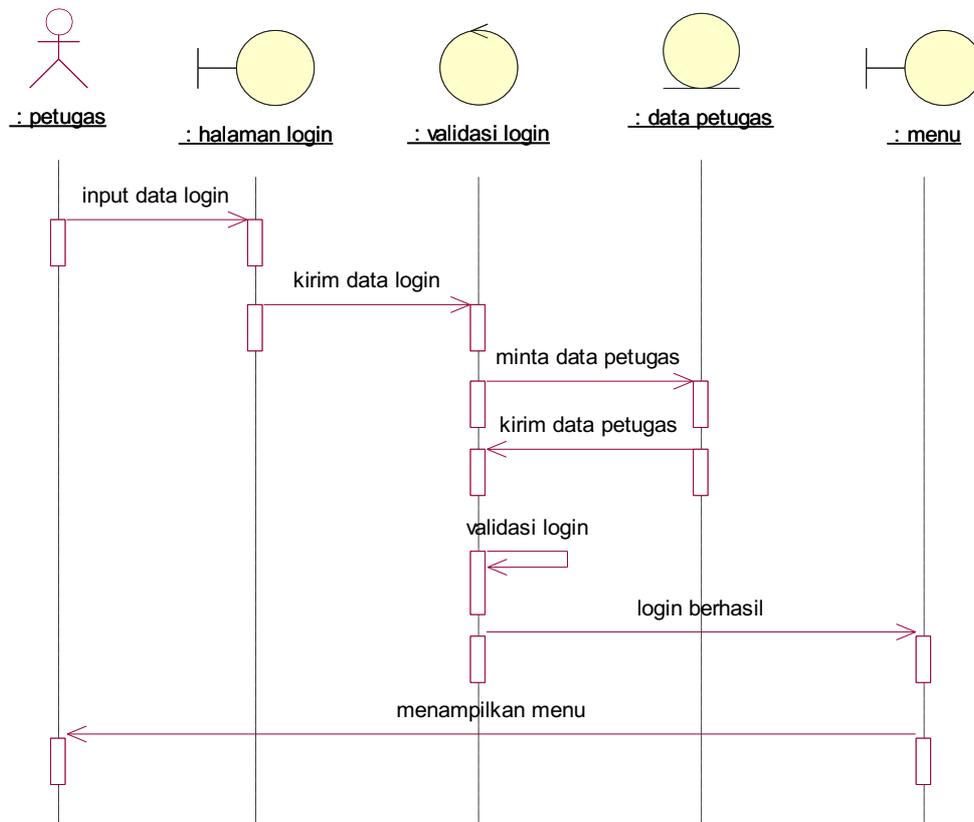
Gambar 3.6 Sequence Diagram Pendonor Mencari Petugas Terdekat

Deskripsi Gambar 3.6 : Pendonor melakukan pencarian petugas PMI terdekat dengan membuka halaman pencarian. Proses pencarian petugas PMI terdekat membutuhkan data petugas dan data pendonor. Setelah di dapatkan, tampilkan informasi petugas.



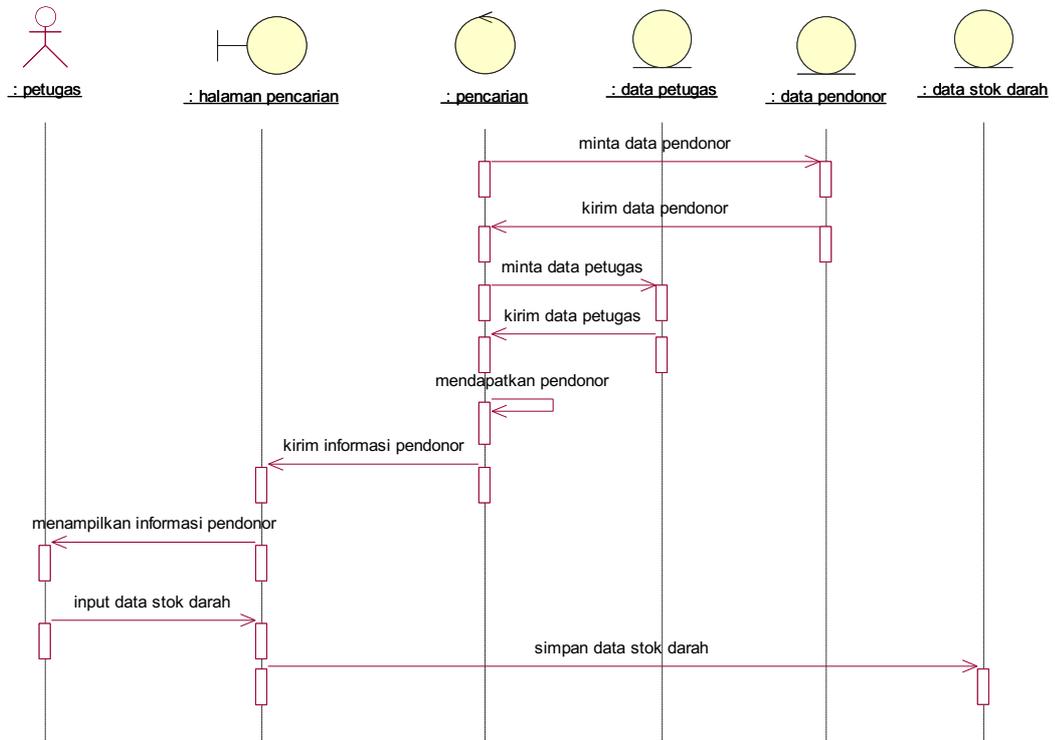
Gambar 3.7 Sequence Diagram Pendonor Melihat Informasi Stok Darah

Deskripsi Gambar 3.7 : Pendonor membuka halaman stok darah, halaman stok darah meminta data dari database. Database stok darah mengirimkan data stok darah untuk ditampilkan kepada pendonor.



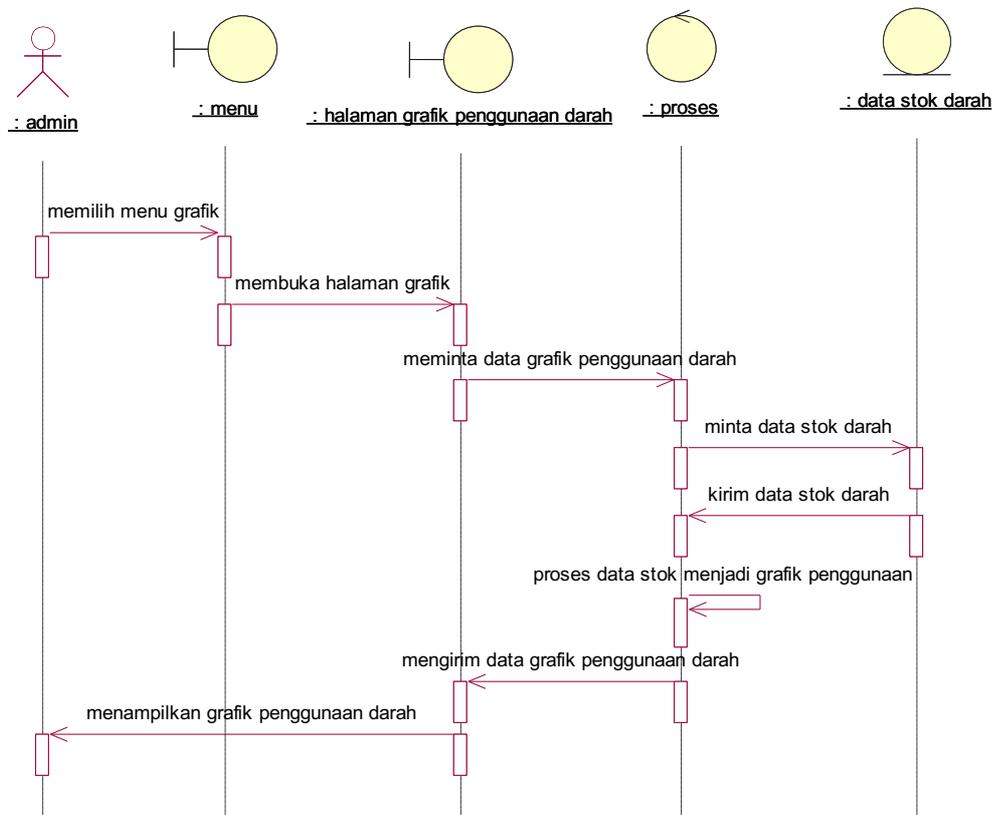
Gambar 3.8 Sequence Diagram Login Petugas

Deskripsi Gambar 3.8 : Petugas melakukan login dengan menginputkan data login yang kemudian divalidasi dengan data petugas dari entitas database. Jika berhasil maka tampilkan menu.



Gambar 3.9 Sequence Diagram Petugas Dicari Pendoror

Deskripsi Gambar 3.9 : Petugas menunggu pencarian dari pendonor pada halaman pencarian. Proses pencarian membutuhkan data petugas dan data pendonor. Setelah mendapatkan pendonor yang cocok, tampilkan informasi pendonor. Kemudian petugas melakukan input data stok darah untuk disimpan di data stok darah.



Gambar 3.10 Sequence Diagram Admin Melihat Grafik Donor Darah

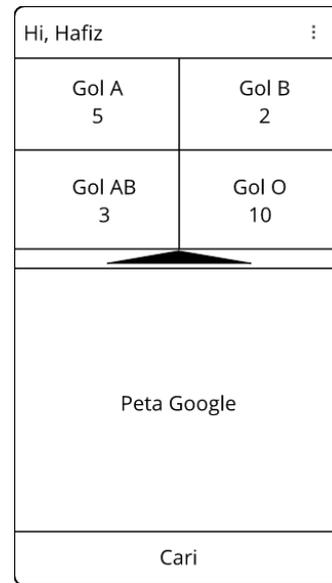
Deskripsi Gambar 3.10 : Admin membuka halaman grafik informasi donor darah, halaman grafik tersebut meminta data dari hasil proses stok darah. Database stok darah mengirim data untuk diproses menjadi grafik lalu ditampilkan ke admin.

3.3 Perancangan Antarmuka

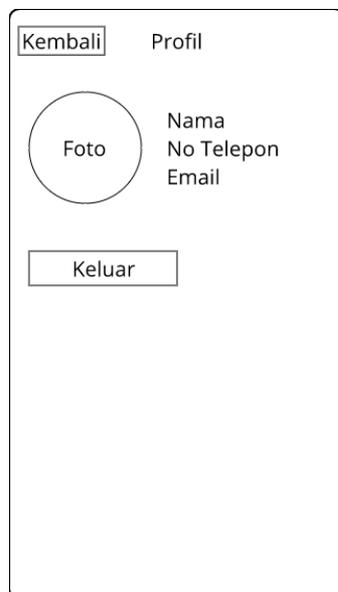
Perancangan antarmuka (interface) aplikasi merupakan tahap awal sebelum implementasi aplikasi sebagai berikut :



Gambar 3.11 Login Aplikasi



Gambar 3.12 Halaman Utama Pendoror



Gambar 3.14 Profil Petugas



Gambar 3.13 Halaman Utama Petugas

3.4 Perancangan Tabel

Untuk mempermudah pengolahan data, beberapa data yang ada dikumpulkan ke dalam *Firestore* dibentuk dalam tabel-tabel, dengan tabel sebagai berikut :

1. Tabel Pendonor

Tabel Pendonor berisi data informasi untuk kebutuhan mengidentifikasi pengguna yang akan masuk ke dalam aplikasi pendonor.

Tabel 3.1 Struktur Tabel Pendonor

Nama Field	Tipe	Lebar	Keterangan
No Telepon	Varchar	20	Nomor telepon pengguna sebagai kunci primer
Name	Varchar	40	Nama Pengguna
Email	Varchar	30	Email Pengguna

2. Tabel Petugas

Tabel petugas berisi data informasi untuk kebutuhan mengidentifikasi pengguna yang akan masuk ke dalam aplikasi petugas.

Tabel 3.2 Struktur Tabel Petugas

Nama Field	Tipe	Lebar	Keterangan
No Telepon	Varchar	20	Nomor telepon pengguna sebagai kunci primer
Name Depan	Varchar	20	Nama Depan

Nama Belakang	Varchar	20	Nama Belakang
Email	Varchar	30	Email Pengguna
url	Text	-	Url foto dari petugas untuk ditampilkan

3. Tabel Donor

Tabel Donor berisi relasi data antara tabel pendonor dan tabel petugas yang menandakan bahwa pendonor telah mendonorkan darahnya ke petugas PMI.

Tabel 3.3 Struktur Tabel Pencarian

Nama Field	Tipe	Lebar	Keterangan
Timestamp	varchar	30	Kunci primer diambil dari tanggal yang di generate dari <i>firebase</i>
phonePendonor	varchar	20	Nomor telepon dari pendonor
PhonePetugas	varchar	20	Nomor telepon dari petugas
tgl	Date	10	Tanggal donor darah
gol	Enum	-	Golongan darah yang telah diambil petugas

4. Tabel Geo Pendonor

Tabel geo pendonor berisi data lokasi *latitude* dan *longitude* dari pendonor.

Data tersebut digunakan untuk pelacakan lokasi pendonor oleh petugas PMI.

Tabel 3.5 Struktur Tabel Geo Pendonor

Nama Field	Tipe	Lebar	Keterangan
No telepon	varchar	20	Nomor telepon dari pendonor sebagai key
latitude	double	30	Letak latitude dari pendonor
longitude	double	30	Letak longitude dari pendonor

5. Tabel Geo Petugas

Tabel geo petugas berisi data lokasi latitude dan longitude dari petugas. Data tersebut digunakan untuk pencarian lokasi donor oleh pendonor yang akan mendonorkan darahnya.

Tabel 3.5 Struktur Tabel Geo Petugas

Nama Field	Tipe	Lebar	Keterangan
No telepon	varchar	20	Nomor telepon dari petugas sebagai key
latitude	double	30	Letak latitude dari petugas
longitude	double	30	Letak longitude dari petugas

6. Tabel Stok Darah

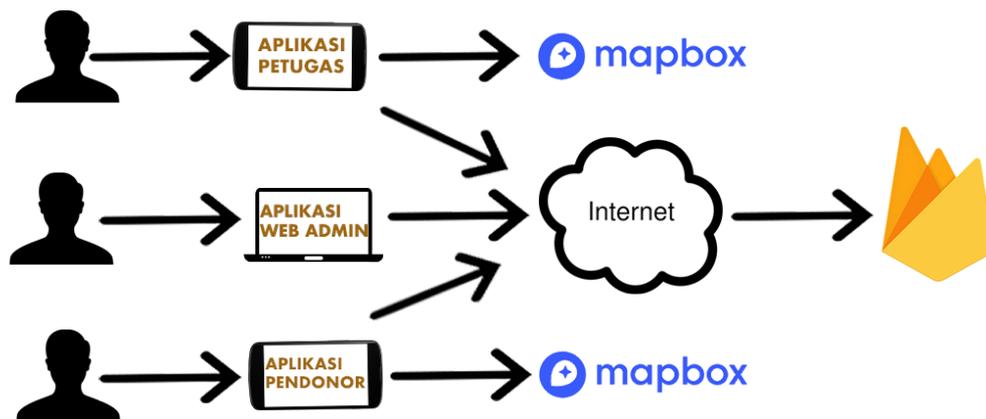
Tabel stok darah berisi data stok darah yang diinputkan oleh petugas atau admin PMI.

Tabel 3.7 Struktur Tabel Stok Darah

Nama Field	Tipe	Lebar	Keterangan
A	Integer	5	Jumlah stok golongan darah A
B	Integer	5	Jumlah stok golongan darah B
AB	Integer	5	Jumlah stok golongan darah AB
O	Integer	5	Jumlah stok golongan darah O

3.5 Arsitektur Teknologi

Arsitektur teknologi dalam pembuatan aplikasi ini adalah sebagai berikut :



Gambar 3.15 Arsitektur Teknologi

Gambar 3.15 merupakan gambar arsitektur teknologi, terlihat terdapat 3 aplikasi yaitu aplikasi petugas, web admin, dan pendonor. Untuk aplikasi petugas dan aplikasi pendonor dijalankan menggunakan android dan membutuhkan peta dari mapbox. Sedangkan untuk aplikasi web admin dijalankan menggunakan personal komputer atau laptop. Keseluruhan aplikasi terhubung dengan internet untuk dapat mengirimkan data ke firebase.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi Dan Pengujian

Tahap implementasi dan pengujian merupakan tahapan penerapan perancangan perangkat lunak sekaligus uji coba aplikasi. Tahapan implementasi *Firestore* pada aplikasi pencarian petugas *pmi* terdekat untuk pendonor berbasis *android* disesuaikan dengan perancangan perangkat lunak yang telah dijabarkan, sehingga mendapatkan keluaran sesuai yang telah dirancang pada bab sebelumnya, antara lain:

Tahapan pembuatan *project* pada *Firestore*

- Masuk dan *login* ke halaman : <https://console.firebase.google.com/u/0/>.
- Tambahkan *project* baru.
- Menghubungkan *Firestore* ke *project client*.

Alur pencarian petugas PMI terdekat

- Inialisasi lokasi petugas PMI dan Pendonor.
- Implementasi pencarian petugas dengan radius 10km.
- Dialog saat petugas ditemukan.

Alur informasi stok darah

- Input data stok darah
- Insialisasi data stok darah dari *Firebase*.
- Menampilkan informasi stok darah.

Alur mendapatkan jarak tempuh

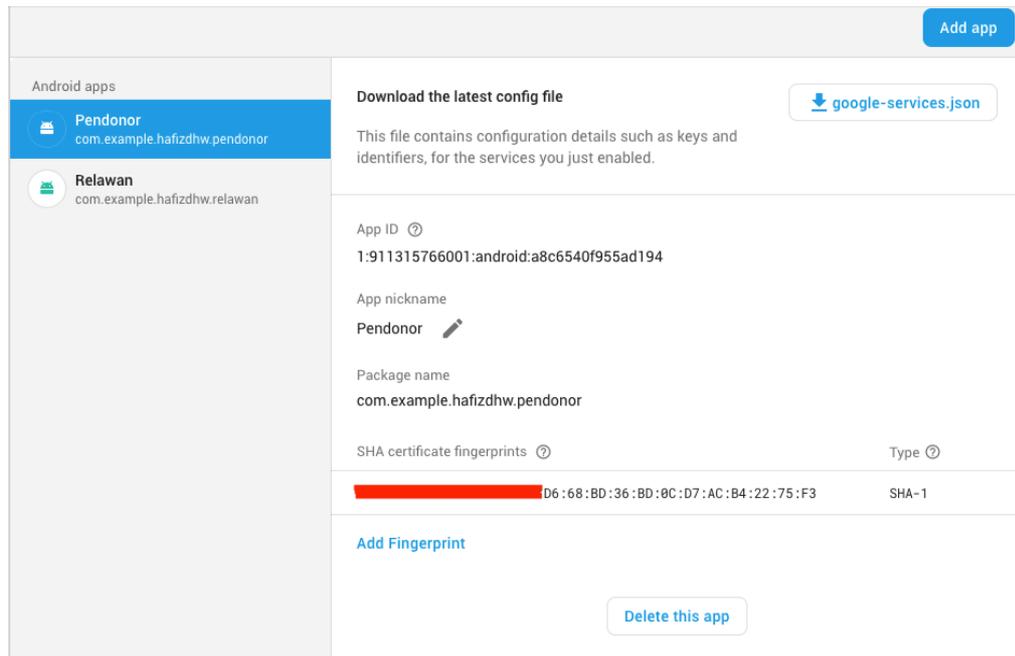
- Insialisasi rute mapbox
- Mendapatkan jarak tempuh dari rute mapbox

Alur menampilkan grafik donor darah untuk Admin PMI

- Insialisasi data donor dari *Firebase*.
- Menampilkan grafik informasi donor darah.

A. Tahap Pembuatan Project Pada *Firebase*

Pada tahapan ini menguraikan tentang pembuatan *project* pada *Firebase*. Untuk membuat sebuah *project* pada *Firebase*, hal pertama dilakukan adalah masuk ke dalam situs *Firebase Console* (<https://console.firebase.google.com/>) selanjutya tambahkan *project* baru. Kemudian setelah *project* dibuat, daftarkan aplikasi android pada *firebase* menggunakan SHA *certifacte fingerprints* console seperti yang terlihat pada Gambar 4.1 berikut.



Gambar 4.1 Mendaftarkan Aplikasi *Android* Ke *Firebase Console*

Selanjutnya setelah aplikasi didaftarkan pada *Firebase console*, hubungkan *Firebase* ke dalam *project* aplikasi. Untuk aplikasi *Android* dapat dilihat pada gambar 4.2 dan 4.3 sedangkan untuk aplikasi web dapat dilihat pada gambar 4.4.

```

buildscript {
    // ...
    dependencies {
        // ...
        classpath 'com.google.gms:google-services:4.0.1' // google-services plugin
    }
}

allprojects {
    // ...
    repositories {
        // ...
        google() // Google's Maven repository
    }
}

```

Gambar 4.2 Script Pada Build Gradle

Untuk dapat menambahkan *library Firebase* pada aplikasi Android Studio dibutuhkan *google services plugin* dan *repository* dari Google Maven. Script pada gambar 4.2 di tambahkan pada *build.gradle* di Android Studio kemudian di sinkronisasi.

```

apply plugin: 'com.android.application'

android {
    // ...
}

dependencies {
    // ...
    implementation 'com.google.firebase:firebase-core:16.0.1'

    // Getting a "Could not find" error? Make sure you have
    // added the Google maven repository to your root build.gradle
}

// ADD THIS AT THE BOTTOM
apply plugin: 'com.google.gms.google-services'

```

Gambar 4.3 Script Pada App/build.gradle

Setelah *repository* berhasil didaftarkan, tambahkan potongan kode seperti gambar 4.3 pada *app/build.gradle* untuk menginstall firebase pada aplikasi. Setelah menambahkan script, sinkronisasi Android Studio.

```

<script src="https://www.gstatic.com/firebasejs/5.2.0/firebase.js"></script>
<script>
    // Initialize Firebase
    // TODO: Replace with your project's customized code snippet
    var config = {
        apiKey: "<API_KEY>",
        authDomain: "<PROJECT_ID>.firebaseapp.com",
        databaseURL: "https://<DATABASE_NAME>.firebaseio.com",
        projectId: "<PROJECT_ID>",
        storageBucket: "<BUCKET>.appspot.com",
        messagingSenderId: "<SENDER_ID>",
    };
    firebase.initializeApp(config);
</script>

```

Gambar 4.4 Script Menghubungkan Firebase Dan Aplikasi Web

Script pada gambar 4.4 digunakan untuk implementasi *Firebase* pada aplikasi web menggunakan bahasa javascript. Setelah *Firebase* terhubung, aplikasi dapat mengirimkan data ke *Firebase* secara *realtime*.

B. Alur Pencarian Petugas PMI terdekat

Tahap pertama dalam melakukan pencarian petugas PMI terdekat yaitu menginisialisasi lokasi petugas agar dapat dicari oleh pendonor. Pada interface aplikasi petugas terdapat tombol aktif/tidak aktif, apabila di aktifkan maka akan menginisialisasi lokasi petugas PMI, lalu disimpan ke *Firebase*.

```
@Override
public void onSwitch(boolean status) {
    if (status) { // jika aktif
        startService();
        // insialisasi lokasi
        mGeo.setLocation(phoneId,
            new GeoLocation(myLocation.getLatitude(),
                myLocation.getLongitude()));
    } else { // jika tidak
        stopService();
        // hapus lokasi
        mGeo.removeLocation(phoneId);
    }
}
```

Gambar 4.5 Fungsi aktif/tidak aktif

Potongan kode pada Gambar 4.5 adalah fungsi yang digunakan ketika petugas mengaktifkan *switch* atau menonaktifkannya. Apabila status aktif maka set lokasi ke *firebase* pada baris ke-6 apabila status tidak aktif maka hapus lokasi petugas pada baris ke-11 di *firebase*. Lokasi dihapus agar tidak dapat dicari oleh pendonor

apabila petugas tidak aktif. Setelah lokasi petugas di insialisasi, selanjutnya yaitu melakukan pencarian petugas dengan radius 10km.

```
private void findWorker(double latitude, double longitude) {
    geoQuery = petugasGeo.queryAtLocation(new GeoLocation(latitude,
        longitude), RADIUS);
    geoQuery.addGeoQueryEventListener(geoQueryCallback);
}
```

Gambar 4.6 Fungsi Untuk Mencari Petugas

Potongan kode pada Gambar 4.6 merupakan fungsi untuk mencari petugas dengan radius tertentu. Terlihat pada baris ke-2 *queryAtLocation* adalah fungsi dari library GeoFire yang digunakan untuk melakukan pencarian query pada firebase dengan paramater latitude dan longitude dari pendonor. Setelah dilakukan pencarian maka diberikan callback pada baris ke-4 untuk mengetahui apakah terdapat petugas di radius 10km.

```
@Override
public void onGeoQueryReady() { // LISTENER INI DIPANGGIL APABILA SUDAH
    SELESAI PENCARIAN SEMUA DATA PETUGAS
    if (!isPetugasAda) {
        // MENAMPILKAN ALERT DIALOG
        final AlertDialog.Builder builder;
        builder = new AlertDialog.Builder(MainActivity.this,
            android.R.style.Theme_Material_Dialog_Alert);
        builder.setTitle("Perhatian")
            .setCancelable(false)
            .setMessage("Maaf, kami tidak menemukan petugas PMI di
                sekitar anda. Ingin menuju ke kantor PMI Bantul?")
            .setPositiveButton("Ya", (dialogInterface, i) -> {
                goToOffice();
            })
            .setNeutralButton("Tidak", (dialog, which) ->
```

```

        dialog.dismiss()
        .setIcon(android.R.drawable.ic_dialog_alert)
        .show();
    } else {
        String keyP;
        PetugasLocation petugasLocation;
        if (listPetugas.size() > 1) { // JIKA ADA LEBIH DARI 1 PETUGAS
            DI RADIUS
            petugasLocation =
                findNearestWorker(myLocation.getLatitude(),
                    myLocation.getLongitude(), listPetugas);
            keyP = petugasLocation.getKey();
        } else { // JIKA HANAYA ADA 1 PETUGAS
            petugasLocation = listPetugas.get(0);
            keyP = petugasLocation.getKey();
        }
        sendNotifToWorker(keyP, petugasLocation);
    }
}

```

Gambar 4.7 *Callback Query* Pencarian Petugas

Gambar 4.7 merupakan potongan kode pada *callback* pencarian petugas yang menjelaskan bahwa apabila tidak ditemukan petugas maka tampilkan dialog dan menawarkan pendonor untuk menuju ke lokasi kantor PMI. Potongan kode tersebut terletak pada aplikasi pendonor. Pada baris ke-8 sampai ke-21 digunakan untuk menampilkan dialog apabila tidak ada petugas yang ditemukan. Sedangkan apabila petugas ditemukan dan lebih dari 1 maka dicari yang terdekat dengan fungsi *findNearestWorker* seperti pada baris ke-28, lalu kirim notifikasi kepada petugas dengan cara menuliskan data pendonor ke *firebase* petugas menggunakan fungsi *sendNotiftoWorker* seperti yang terdapat pada potongan kode pada Gambar 4.7 baris ke-35. Agar petugas dapat mendapatkan notifikasi secara *realtime*

diperlukan *listener child added* seperti pada Gambar 4.8 berikut.

```

@Override
public void onChildAdded(DataSnapshot dataSnapshot, String s) {
    // MENAMPILKAN ALERT DIALOG
    AlertDialog.Builder builder1 = new
    AlertDialog.Builder(MainActivity.this,
        android.R.style.Theme_Material_Dialog_Alert);
    alertDialog = builder1.create();
    alertDialog.setTitle("Notifikasi");
    alertDialog.setCancelable(false);
    alertDialog.setMessage("Ada pendonor " + pendonor.getName() + "
    dengan nomor " + keyPendonor);
    alertDialog.setButton(Dialog.BUTTON_POSITIVE, "Terima", (dialog,
    which) -> {
        dialog.dismiss();
        statusRef.setValue(1);
        statusPendonor.setValue(1);
    });
    alertDialog.setButton(Dialog.BUTTON_NEGATIVE, "Tolak", (dialog,
    which) -> {
        dialog.dismiss();
        cancelDonorByPetugas();
    });
    alertDialog.setIcon(android.R.drawable.ic_dialog_alert);
    alertDialog.show();
}

```

Gambar 4.8 Listener Untuk Mendapatkan Notifikasi

Potongan kode pada Gambar 4.8 adalah sebuah *listener* untuk mendapatkan notifikasi dengan trigger menggunakan *child added*. Apabila terdapat perubahan pada *child firebase* maka *listener* tersebut akan terpanggil. Pada potongan kode baris ke-12 sampai dengan baris ke-31, terlihat apabila terdapat pendonor yang menambahkan *child* maka akan menampilkan dialog, apakah petugas akan

menerima donor atau tidak. Jika terima maka pendonor akan ditujukan ke lokasi petugas dan petugas menunggu pendonor menuju lokasinya. Kode tersebut terletak pada aplikasi untuk petugas.

C. Alur Informasi Stok Darah

Tahap pertama dalam menampilkan informasi stok darah kepada pendonor yaitu insialisasi data stok darah terlebih dahulu. Data stok darah didapatkan dari *input* admin pada web dan aplikasi petugas. Perbedaanya untuk web admin input darah lebih kompleks, admin dapat merubah stok darah semua golongan darah. Sedangkan untuk aplikasi petugas hanya dapat menambahkan 1 kantong dan memilih salah satu golongan darah tersebut.

```
function simpanDarah() {  
  var a = $("#gol-a").val();  
  var b = $("#gol-b").val();  
  var ab = $("#gol-ab").val();  
  var o = $("#gol-o").val();  
  darahRef.update({A : a});  
  darahRef.update({B : b});  
  darahRef.update({AB : ab});  
  darahRef.update({O : o});  
}
```

Gambar 4.9 Fungsi Update Stok Darah Pada Admin

Potongan kode pada Gambar 4.9 merupakan potongan dari aplikasi web admin yang fungsinya digunakan untuk update data stok darah pada *firebase*. Fungsi simpan darah ini dipanggil ketika admin web menekan tombol simpan. Selanjutnya untuk aplikasi petugas juga memiliki hak untuk menambahkan stok

darah seperti Gambar 4.10 berikut.

```
String gol = spinner.getSelectedItem().toString();
pushDarah(gol);
```

Gambar 4.10 Fungsi *Push* Darah Pada Aplikasi Petugas

Potongan kode pada Gambar 4.10 digunakan untuk inialisasi tipe golongan darah yang akan di inputkan dengan cara *getSelecteditem* dari *spinner*. Setelah didapatkan *string* golongan darahnya, *string* darah tersebut di *push* menuju ke firebase melalui fungsi *pushDarah*.

```
mDatabase.getReference("darah").addValueEventListener(dataDarahCallback);

ValueEventListener dataDarahCallback = new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        mStokA.setText(dataSnapshot.child("A").getValue().toString());
        mStokB.setText(dataSnapshot.child("B").getValue().toString());
        mStokAB.setText(dataSnapshot.child("AB").getValue().toString());
        mStokO.setText(dataSnapshot.child("O").getValue().toString());
    }
    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
};
```

Gambar 4.11 Fungsi Dan *Callback* Menampilkan Stok Darah

Untuk dapat menampilkan stok darah dibutuhkan inialisasi *addValueEventListener* yang artinya menambahkan listener untuk referensi firebase “darah”. Setelah diberikan callback, apabila terdapat perubahan pada data stok darah maka text view akan update value. Jadi tampilan untuk stok darah ini

akan update secara real time apabila terdapat perubahan pada data stok darah.

D. Alur Mendapatkan Jarak Tempuh

Langkah awal untuk mendapatkan jarak tempuh yaitu dengan menginisialisasi rute dari *mapbox*. Berikut ini adalah *builder* untuk mendapatkan rute dari *mapbox* yang dapat dilihat pada gambar 4.12

```

NavigationRoute.builder(this)
    .accessToken(Mapbox.getAccessToken())
    .origin(origin)
    .destination(destination)
    .build()
    .getRoute(new Callback<DirectionsResponse>() {
        @Override
        public void onResponse(@NonNull Call<DirectionsResponse> call,
            @NonNull Response<DirectionsResponse> response) {
            DirectionsRoute currentRoute = response.body().routes().get(0);
            double jarak = currentRoute.distance();
        }
    });

```

Gambar 4.12 Builder untuk mendapatkan rute

Terlihat pada gambar 4.12 terdapat builder untuk mendapatkan rute. Builder tersebut memiliki beberapa fungsi diantaranya, pada baris ke-2 merupakan fungsi untuk meminta access token dari mapbox. Baris ke-3 dan ke-4 merupakan fungsi untuk meminta lokasi asal dan lokasi tujuan. Baris ke 6 digunakan untuk mendapatkan rute, lalu pada baris ke-10 digunakan untuk menyimpan rute yang telah didapatkan ke dalam variabel *currentRoute* dari kelas *DirectionsRoute*. Setelah rute didapatkan gunakan fungsi *distance()* untuk mendapatkan jarak,

seperti yang terlihat pada baris ke-11.

E. Alur Menampilkan Grafik Donor Darah Untuk Admin PMI

Langkah awal untuk menampilkan grafik donor darah yaitu menginisialisasi data donor dari firebase. Berikut beberapa fungsi yang digunakan untuk menampilkan grafik donor darah. Untuk menginisialisasi data donor diperlukan fungsi *donorRef.on("child_added", function(snapshot) {})*;

```
var data = [
    { y: date1, item1: a1, item2: b1, item3: ab1, item4: o1 },
    { y: date2, item1: a2, item2: b2, item3: ab2, item4: o2 },
    { y: date3, item1: a3, item2: b3, item3: ab3, item4: o3 },
    { y: date4, item1: a4, item2: b4, item3: ab4, item4: o4 },
    { y: date5, item1: a5, item2: b5, item3: ab5, item4: o5 },
    { y: date6, item1: a6, item2: b6, item3: ab6, item4: o6 },
    { y: date7, item1: a7, item2: b7, item3: ab7, item4: o7 }
];
area.setData(data);
```

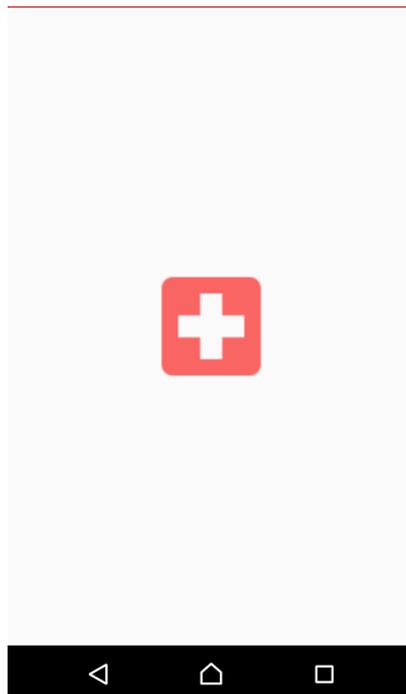
Gambar 4.13 Fungsi Untuk Menampilkan Grafik

Potongan kode pada Gambar 4.13 merupakan proses pembentukan data donor menjadi array agar dapat diproses menjadi grafik informasi donor darah. Untuk y merupakan tanggal, item1 merupakan golongan darah A, item2 golongan darah B, item3 golongan darah AB, dan untuk item4 adalah golongan darah O. Setelah data donor diproses menjadi *array*, lalu *array* tersebut di set ke morris js untuk diproses menjadi grafik.

4.2 Pembahasan

1) Tampilan *Splashscreen*

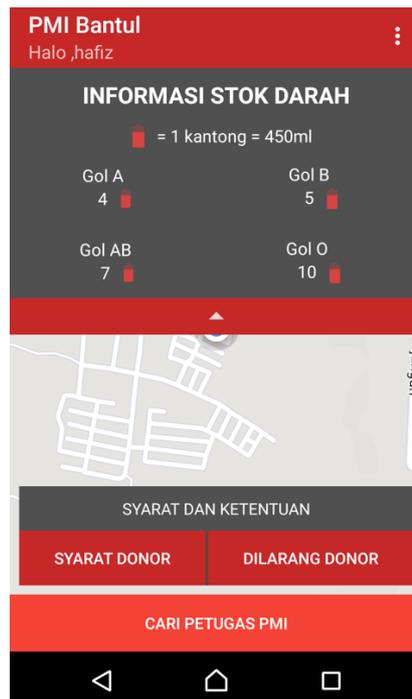
Tampilan pertama yang dijalankan adalah *splashscreen* yang berupa logo berlambang PMI. *Splashscreen* ini digunakan untuk meminta izin *permission* penggunaan lokasi pada *Android*. Selain itu *inteface* ini digunakan untuk mengecek apakah GPS dan internet sudah aktif atau belum. Gambar 4.5 merupakan tampilan *splashscreen* pada aplikasi ini.



Gambar 4.14 Tampilan *Splashscreen*

2) Tampilan Aplikasi Pendoror

Tampilan *inteface* aplikasi pendonor adalah halaman yang muncul setelah *splashscreen* dan apabila penggunanya sudah *login*.



Gambar 4.15 Tampilan Aplikasi Pendoror

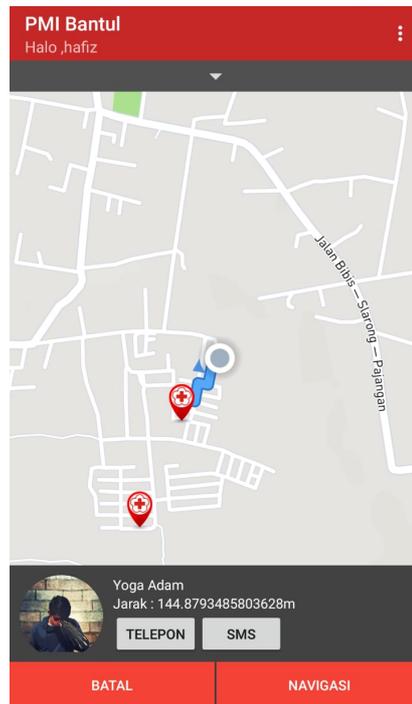
Pada Gambar 4.15 adalah tampilan dari awal dari aplikasi pendonor. Halaman tersebut menampilkan *toolbar* yang berjudul nama kantor PMI dan ber *subtitle* nama pengguna yang *login*. Selain itu *interface* ini menampilkan informasi stok darah dan menampilkan peta yang di dalamnya terdapat *marker* petugas PMI yang sedang aktif mencari pendonor. Halaman ini juga memiliki fitur untuk mencari petugas PMI terdekat untuk pengguna yang ingin mendonorkan darahnya. Kemudian halaman ini juga memiliki fitur untuk bantuan syarat ketentuan yang

terletak di bagian bawah berisi syarat donor dan dilarang donor.



Gambar 4.16 Lokasi Petugas PMI

Gambar 4.16 merupakan tampilan yang terdapat pada aplikasi pendonor apabila di sekitar pendonor terdapat petugas PMI. Terlihat terdapat 2 marker petugas PMI dan 1 marker adalah lokasi si pendonor. Ketika pendonor menekan tombol cari maka lokasi petugas PMI yang terdekat akan menampilkan notifikasi atau dialog pada aplikasi petugas.

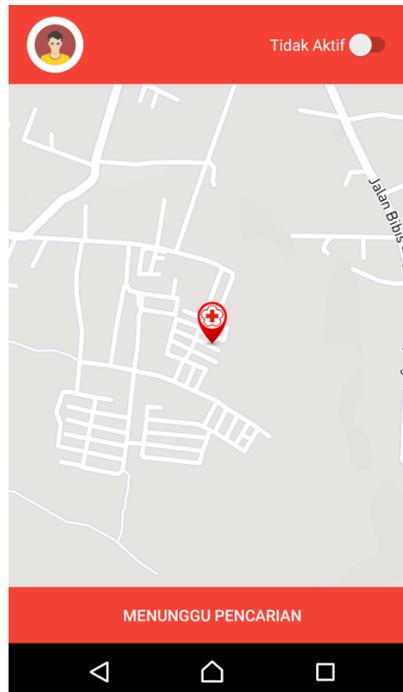


Gambar 4.17 Aplikasi Pendoror Setelah Pencarian

Gambar 4.17 merupakan hasil dari pencarian pada aplikasi pendonor. Setelah pencarian pendonor akan ditampilkan menu informasi nama petugas, jarak dan diberikan tombol untuk telepon atau *sms*. Di bawah menu tersebut terdapat tombol batal yang digunakan untuk membatalkan donor darah. Selain tombol batal terdapat tombol navigasi yang digunakan untuk navigasi peta menuju lokasi petugas PMI yang sedang aktif. Pada peta terdapat rute menuju lokasi petugas PMI yang telah menyetujui untuk donor darah.

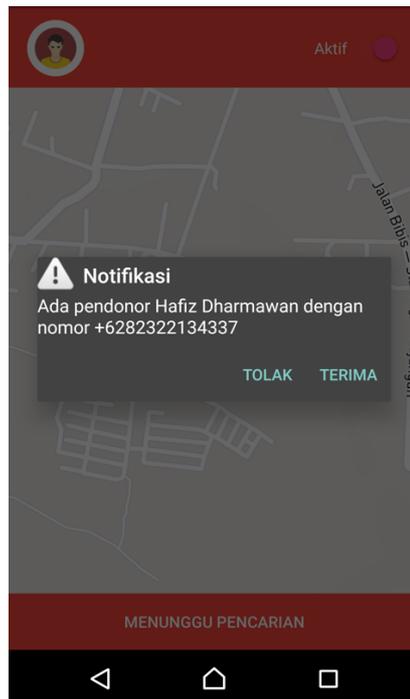
3) Tampilan Aplikasi Petugas

Tampilan *inteface* aplikasi petugas adalah halaman yang muncul setelah *splashscreen* dan apabila penggunaanya sudah *login*.



Gambar 4.18 Tampilan Aplikasi Petugas

Pada Gambar 4.18 merupakan halaman yang terdapat pada aplikasi petugas. Halaman tersebut menampilkan foto profil dari petugas pada toolbar. Pada bagian sudut kanan atas terdapat switch yang digunakan untuk menandakan petugas PMI sedang aktif atau tidak aktif mencari pendonor. Selain itu terdapat peta yang digunakan untuk melacak apabila terdapat pendonor yang ingin mendonorkan darahnya. Fitur yang dimiliki halaman ini yaitu akan memunculkan pop up apabila terdapat pendonor.



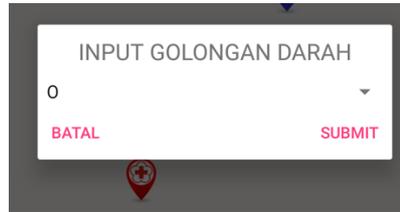
Gambar 4.19 Tampilan *Popup* Aplikasi Petugas

Terlihat pada Gambar 4.19 merupakan tampilan dialog pada aplikasi petugas yang muncul apabila terdapat pendonor yang ini donor. Petugas dapat menolak atau menerima, jika petugas menerima donor tersebut maka akan menampilkan *marker* lokasi dari pendonor seperti pada Gambar 4.25 berikut.



Gambar 4.20 Aplikasi Petugas Setelah Pencarian

Gambar 4.20 merupakan hasil tampilan pada aplikasi petugas ketika pencarian petugas PMI telah ditemukan. Terlihat, terdapat menu tampilan dengan keterangan, nama pendonor yaitu “hafiz” dan diberikan tombol telepon atau *sms*. Pada peta terdapat 2 *marker*, untuk *marker* warna biru merupakan lokasi petugas PMI dan *marker* warna merah merupakan lokasi pendonor. *Marker* tersebut dapat dilacak oleh petugas atau bersifat *realtime*. Pada bagian paling bawah terdapat tombol batal yang gunanya untuk membatalkan donor darah.

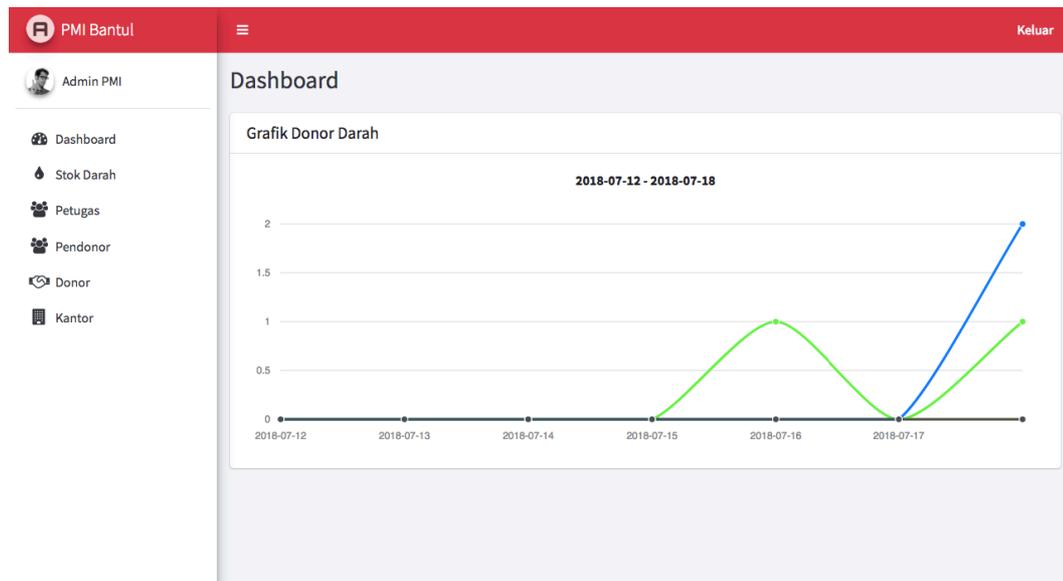


Gambar 4.21 Input Golongan Darah

Gambar 4.21 merupakan input golongan darah pada aplikasi petugas PMI yang muncul apabila petugas PMI telah menyelesaikan proses donor. Terlihat input yang dilakukan untuk pengujian adalah golongan darah O. Setelah di inputkan maka akan mengubah tampilan informasi stok darah yang terdapat pada aplikasi pendonor.

4) Tampilan Web Admin

Tampilan interface web admin digunakan untuk admin dari PMI. Di halaman ini, admin dapat melihat grafik informasi donor darah yang menggambarkan kegiatan donor setiap 1 minggu. Selain itu halaman admin ini disediakan beberapa menu untuk melihat list petugas, pendonor, dan kegiatan donor. Halaman web admin ini juga dapat digunakan untuk menginput stok darah secara manual. Gambar 4.9 adalah gambar tampilan web admin.



Gambar 4.22 Tampilan Interface Web Admin

Dari hasil pencarian petugas PMI pada Gambar 4.12 menunjukkan bahwa aplikasi petugas PMI menerima notifikasi atau dialog ada pendonor yang ingin mendonorkan darahnya. Berikut adalah gambar tampilan hasil pencarian setelah petugas PMI di temukan.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil penelitian, berikut kesimpulan dari Aplikasi ini :

1. Aplikasi telah berhasil dibuat dan dapat melakukan pencarian petugas PMI terdekat dengan radius 10 km untuk pendonor pada sistem *Android* dengan teknologi *Firebase*.
2. Teknologi *Firebase* terbukti dapat membuat aplikasi berjalan secara real time untuk pelacakan lokasi dan informasi stok darah.
3. Peta *MapBox* dapat membantu pendonor darah untuk melakukan navigasi menuju lokasi petugas PMI.

5.2 Saran

Berdasarkan implementasi dan pengujian masih banyak yang dapat dilakukan untuk pengembangan aplikasi ini. Pengembangan dapat dilakukan pada beberapa bagian sebagai berikut :

1. Agar pendonor dan petugas lebih muda dalam berkomunikasi, sebaiknya aplikasi ini diberikan fitur *chat* menggunakan *web socket* atau *firebase cloud message*.

2. Aplikasi sebaiknya ditambahkan fitur rating untuk membantu pendonor memilih petugas PMI yang melayani donor berkualitas.
3. Aplikasi sebaiknya ditambahkan fitur riwayat donor untuk petugas PMI maupun pendonor sehingga petugas atau pendonor dapat mengetahui riwayat donor darahnya.

DAFTAR PUSTAKA

- Android.2018.*Android Studio*. <https://developer.android.com>. Diakses 17/01/2018 17:16:00
- Bobsusanto.2017. *Pengertian PMI Serta Sejarah PMI Lengkap*. <http://www.spengetahuan.com/2016/09/pengertian-pmi-serta-sejarah-pmi-lengkap.html> . diakses 17/01/2018 00:13:00
- Buru, Dadang Juwoto.2017. *Aplikasi Pendeteksi Lokasi Perangkat Bergerak Menggunakan Teknologi Cloud Computing Dengan Firebase Relatime Database Berbasis Android*. STMIK Akakom Yogyakarta
- Fadillah.2017. *Aplikasi Sistem Pelacakan Lokasi Kendaraan Antar Jemput Anak Sekolah Berbasis Android*. Universitas Bandar Lampung
- Firestore.2018.*Firestore*. <https://firebase.google.com>. Diakses 17/01/2018 22:53:00
- Gunawan.2017. *Urutan Tingkatan Versi Sistem Operasi Android*. <https://haiwiki.info/teknologi/urutan-tingkatan-versi-sistem-operasi-android/>. Diakses 17/01/2018 16:39:00
- Json.2018.*Pengenalan JSON*. <https://www.json.org/json-id.html>. Diakses 17/01/2018 23:09:00
- Mapbox.2018.*Tentang Mapbox*. <https://www.mapbox.com/about/>. Diakses 21/01/2018 22:09:00
- Moh., Mustofa Ali .2015. *Pencarian Lokasi Hotel Berbasis Android Menggunakan Metode Location Based Service (Studi Kasus Kota Semarang)*. Universitas Dian Nuswantoro Semarang
- Mulyadi, Tedi.2013. *Pengertian Rational Rose*. <http://tedimulyadi12.blogspot.co.id/2013/06/pengertian-rational-rose.html>. Diakses 21/01/2018 22:35:00
- Muzayyin, Alif. 2017. *Perbedaan Kadar Kolesterol Sebelum Dan Sesudah Donor Darah* . Universitas Muhammadiyah Semarang.
- Setiawan, Hendrawan. 2017 . *Stok Darah Menipis PMI Yogyakarta Perpanjang Jam Layanan* . <https://www.cnnindonesia.com/tv/20170610141109-405-220794/stok-darah-menipis-pmi-yogyakarta-perpanjang-jam-layanan> 16/01/2018 05:29:00
- Sukma, D. 2016. *“Data Gfk: 9 dari 10 orang Indonesia Internetan Lewat*

Smartphone". <http://arenalte.com/berita/industri/data-gfk-terbaru-2016-pengguna-smartphone-indonesia/>. Diakses 16/01/2018 23:03:00

Sukma, Deni. Data Gfk: 9 dari 10 orang Indonesia Internetan Lewat Smartphone. <https://arenalte.com/berita/industri/data-gfk-terbaru-2016-pengguna-smartphone-indonesia/> . Diakses 23/05/2018 5:42:00

Wicaksono, Cahyo.2017. *Aplikasi Pencarian Lokasi Apotek Di Kabupaten Sleman*. Universitas Sanata Darma Yogyakarta