

# Pengembangan Modul Dashboarding Pada Aplikasi Mobile Hidup Bebas Tanpa Tuberkulosis (HEBAT)

Arya Tri Putra Majiah<sup>#1</sup>, Tjatur Kandaga Gautama<sup>\*2</sup>

<sup>#</sup>Program Studi SI Teknik Informatika, Universitas Kristen Maranatha  
Jl. Surya Sumantri No. 65 Bandung, 40174

<sup>1</sup>2072023@maranatha.ac.id

<sup>2</sup>tjatur.kandaga@it.maranatha.edu

**Abstract**— Tuberculosis (TB) is one of the infectious diseases caused by the bacterium *Mycobacterium tuberculosis*. Indonesia is the second-highest country in the world with TB cases, and Purwakarta Regency is the city or regency with the third-highest number of cases in West Java. Hebat is an application designed to track the spread of TB in Purwakarta Regency, specifically in the Plered District. This application consists of various modules, one of the modules is Dashboarding module. The purpose of this module is to manage the various data within the application and present it to stakeholders. The application is expected to identify new cases and serve as an initial screening tool for healthcare professionals for further follow-up.

**Keywords**— Dashboarding, Hebat Application, Purwakarta Regency, Tuberculosis

## I. PENDAHULUAN

Menurut World Health Organization (WHO) dalam laporannya pada Global Tuberculosis Report 2022, terdapat estimasi 10,6 juta kasus tuberkulosis di dunia pada tahun 2021 yang di dalamnya terdapat 6,4 juta kasus yang dilaporkan dan menjalani pengobatan. Terdapat 1.6 juta kasus orang yang meninggal karena tuberkulosis di dalam semua kasus tersebut [1]. Indonesia sendiri merupakan negara kedua dengan kasus TB terbanyak, dengan total kasus 969.000 kasus dan diperkirakan terdapat 144.000 kematian. Dibandingkan dengan tahun 2020, yang mana Indonesia berada pada posisi ketiga. Pada tahun 2021, terdapat kenaikan kasus sebesar 17% dibandingkan dengan tahun 2020 [2].

Salah satu program yang diharapkan dapat untuk membantu menekan penyebaran penyakit tuberkulosis adalah program Desa HEBAT (Hidup Bebas Tanpa Tuberkulosis), yang dilaksanakan di Kecamatan Plered, Kabupaten Purwakarta yang merupakan kota atau kabupaten ketiga dengan jumlah pasien tuberkulosis terbanyak di Jawa Barat pada tahun 2021 [3]. Program Desa HEBAT merupakan salah satu kegiatan yang ada di Kampus Merdeka dan masuk dalam skema MBKM Proyek Kemanusiaan. Terdapat tiga fakultas yang tergabung dalam program ini, yaitu Fakultas Kedokteran, Fakultas Seni Rupa dan Desain, dan Fakultas Teknologi Informasi. Program ini merupakan kerja sama antara Universitas Kristen Maranatha dengan Dinas Kesehatan Kabupaten Purwakarta.

Masyarakat untuk saat ini belum dapat mengakses data mengenai penyakit tuberkulosis di daerahnya, dan data mengenai penyakit tuberkulosis ini masih belum terdapat tempat seperti *database* untuk menyimpan data yang dipakai untuk menampilkan data mengenai tuberkulosis di Kecamatan Plered. Dalam pengembangan aplikasi Desa HEBAT ini juga terdapat banyak modul yang perlu diintegrasikan supaya masyarakat maupun pemangku kepentingan dapat dengan mudah untuk memakai aplikasinya.

## II. KAJIAN TEORI

### A. Flutter

Flutter merupakan SDK (*Software Development Kit*) *open-source* yang dikembangkan oleh Google [4]. Dirilis pertama kali tanggal 4 Desember 2018 pada acara Flutter Live. Dengan flutter, memungkinkan pengembang untuk membuat aplikasi dengan antarmuka pengguna yang menarik secara tampilan dan konsisten di berbagai platform, seperti Android, iOS, web, dan desktop. Flutter menggunakan bahasa pemrograman Dart, yang memiliki sintak yang sederhana dan mudah dipahami, sehingga memudahkan pengembang untuk membuat aplikasi. Keunggulan utama Flutter adalah kemampuannya untuk pengembangan lintas platform. Dengan menggunakan satu basis kode, pengembang dapat menciptakan aplikasi yang berjalan di berbagai perangkat dan sistem operasi, ini dapat menghemat waktu dan juga sumber daya.

B. Firebase

Firebase adalah platform pengembangan *mobile* dan web yang dimiliki oleh Google. Firebase dirancang untuk mempermudah pengembangan aplikasi dengan menyediakan berbagai layanan yang mencakup berbagai aspek pengembangan aplikasi, mulai dari *database*, autentikasi pengguna, *hosting*, hingga analisis, kecerdasan buatan (AI), dan perangkat lunak berbasis *cloud*. Firebase membantu pengembang menghemat waktu dan upaya dalam pengembangan aplikasi dengan menyediakan berbagai alat dan layanan yang dapat digunakan secara langsung tanpa harus membangun semuanya dari awal [5].

C. Agile

Agile adalah sebuah pendekatan metodologi pengembangan perangkat lunak yang menekankan fleksibilitas, kolaborasi tim, dan tanggap terhadap perubahan kebutuhan pelanggan [6]. Sejarahnya dimulai pada pertengahan 1990-an ketika sekelompok praktisi perangkat lunak merumuskan Manifesto Agile sebagai panduan untuk menciptakan perangkat lunak yang lebih adaptif dan responsif. Filosofi dasar dari Agile adalah mengutamakan individu dan interaksi, perangkat lunak yang berfungsi, kolaborasi dengan pelanggan, dan respons terhadap perubahan daripada mengikuti proses dan dokumen yang lengkap.

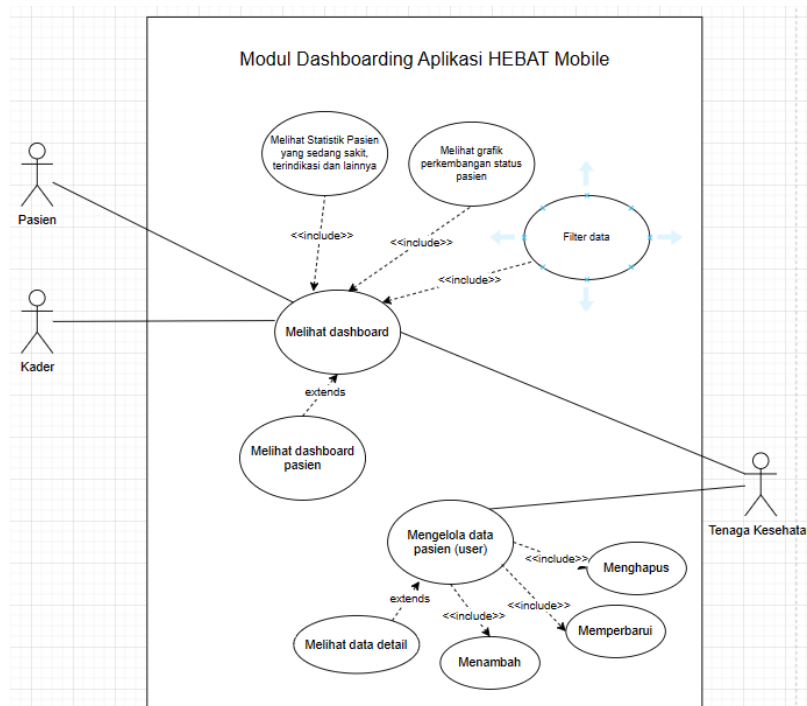
D. Clean Architecture

Dalam proses pengembangan suatu aplikasi, terdapat sebuah konsep arsitektur perangkat lunak yang bertujuan untuk menciptakan sistem yang dapat bertahan lama, mudah dipahami, dan dapat diuji secara efisien dengan melakukan pemisahan yang jelas antara komponen inti aplikasi (*core*) dan lapisan-lapisan lainnya. Arsitektur ini mengusulkan adanya beberapa cincin (ring) yang mengelilingi inti aplikasi, dimulai dari bagian dalam berisi entitas dan *use cases*, diikuti oleh lapisan seperti *Interface Adapters*, *Frameworks* dan *Drivers*, hingga ke lapisan eksternal seperti UI dan *Database*. Dengan menyusun sistem dalam hierarki seperti ini, Clean Architecture tidak hanya memudahkan perubahan dan pemeliharaan, tetapi juga memberikan fleksibilitas dan independen terhadap detail teknologi tertentu [7].

III. ANALISIS DAN RANCANGAN SISTEM

A. Use Case Diagram

Proses pengembangan aplikasi pertama adalah membuat sebuah rancangan. Gambar 1 ini merupakan sebuah use case diagram dari modul *dashboarding*.

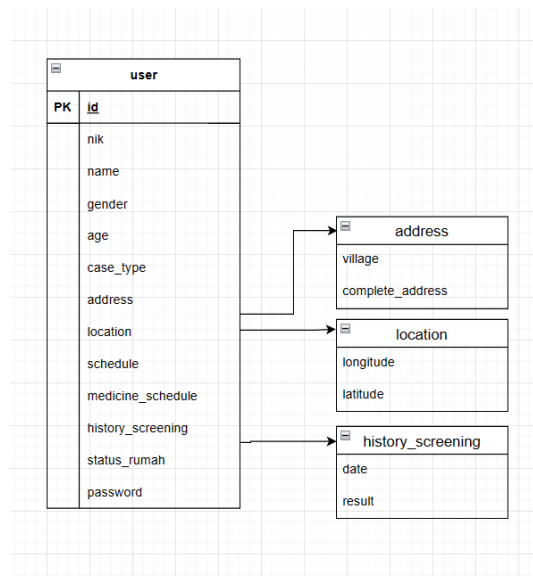


Gambar 1. Use Case Diagram

Diagram ini terdapat beberapa aktor yang terlibat, yang pertama yaitu tenaga kesehatan, yang memiliki akses untuk melihat *dashboard* secara detail dan mengelola data *user* (pasien) seperti menambah, menghapus dan memperbarui data. Kedua yaitu pemerintahan yang memiliki akses terhadap *dashboard* sama seperti tenaga kesehatan. Lalu kedua aktor terakhir yaitu pasien dan kader, yang dimana keduanya hanya memiliki akses terhadap *dashboard* yang datanya disesuaikan. Untuk mengakses *dashboard* diperlukan autentikasi berupa akun. Data yang dimasukkan pertama kali akan dipakai untuk fitur yang lain sehingga tidak perlu untuk memasukkan data kembali ketika mengakses fitur yang lain. Setelah berhasil melakukan autentikasi, akan terbuka sebuah halaman *dashboard*.

### B. Database

Gambar 2 merupakan struktur data dari user yang dipakai dalam proses bisnis ini.

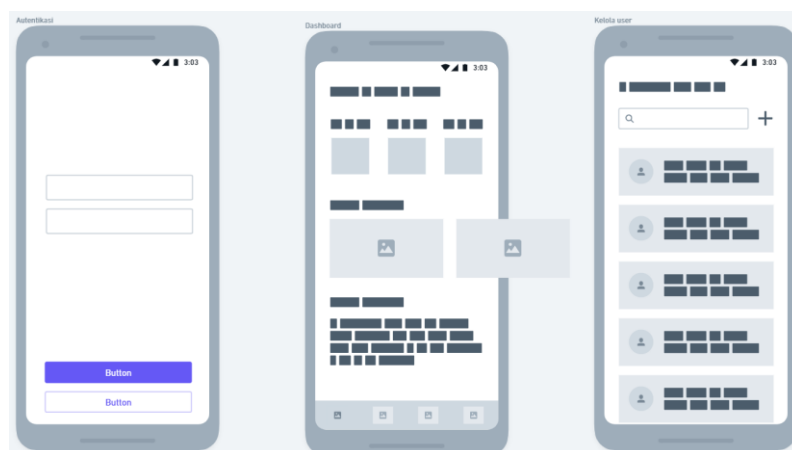


Gambar 2. Struktur Database

Struktur data ini juga yang dipakai dalam keseluruhan fitur didalam aplikasi. Tabel ini disimpan didalam sebuah *database collection* di dalam Firebase Firestore. Dikarenakan penyimpanan yang bersifat *No-SQL* ataupun tanpa skema, maka dalam praktiknya beberapa data mungkin untuk tidak memiliki salah satu kolom seperti dalam gambar.

### C. Wireframe

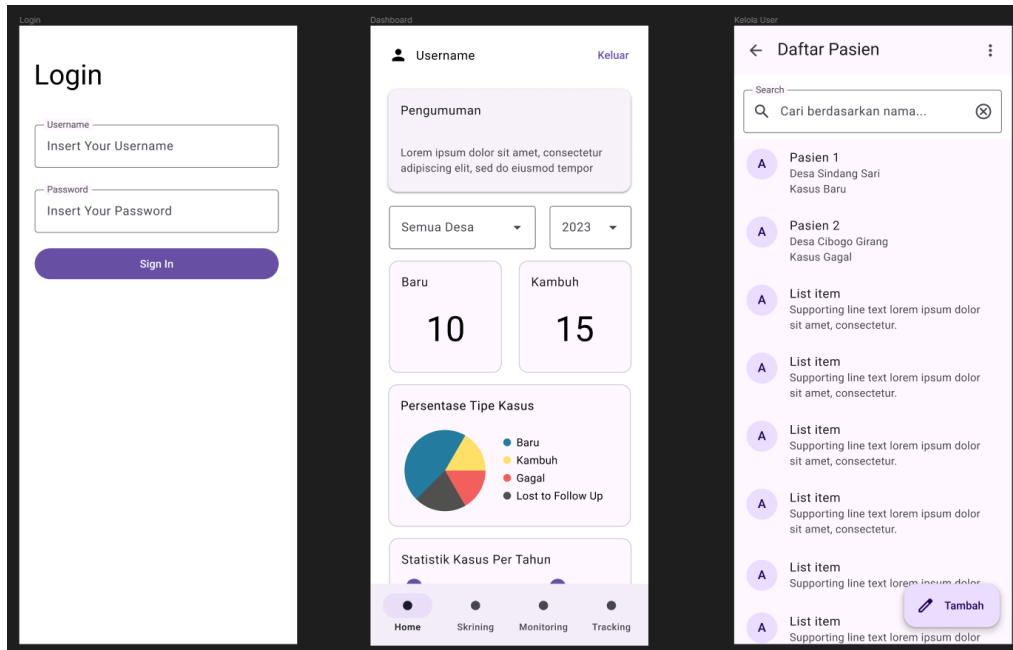
Setelah semua alur mengenai aplikasi telah dirancang, dilanjutkan dengan membuat sebuah *wireframe* dan juga *prototype*. Pada Gambar 3 ini merupakan *wireframe* yang dibuat untuk aplikasi Hebat, yang hanya fitur *dashboard* saja. Didalam *wireframe* ini tercantum tampilan *login* (autentikasi), *dashboard*, dan juga halaman daftar pasien.



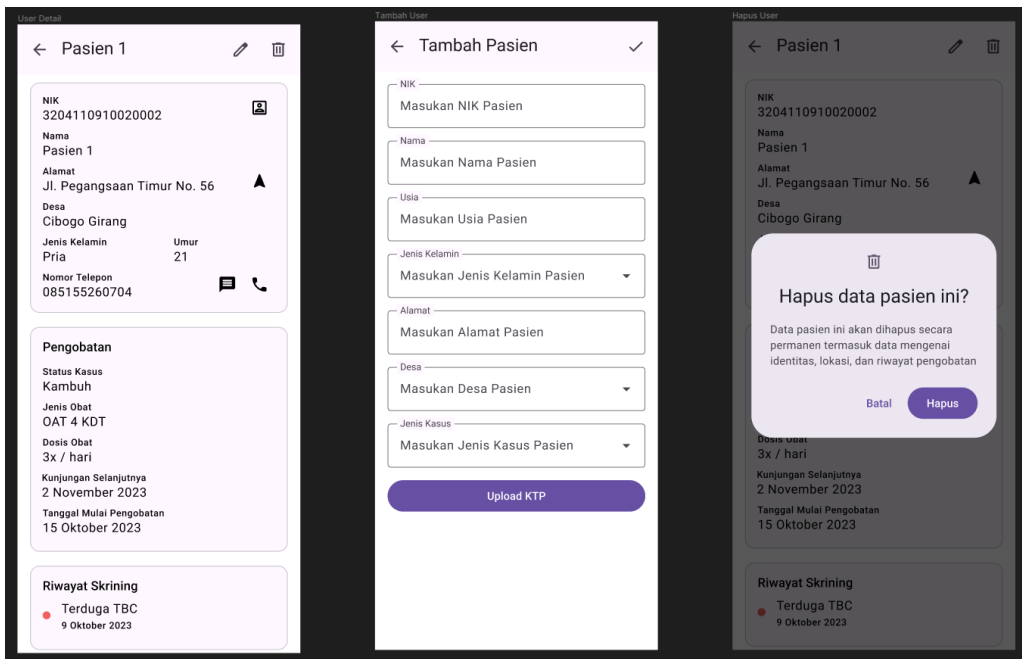
Gambar 3. Wireframe Aplikasi.

D. Prototype

Pada Gambar 4 merupakan prototype yang merupakan rancangan tampilan lanjutan setelah wireframe. Dalam prototype ini, sudah mencakup keseluruhan tampilan yang akan dibuat berdasarkan rancangan use case sebelumnya. Pada gambar ini, terdapat tampilan untuk login, dashboard dan daftar pasien.



Gambar 4. Prototype 1.



Gambar 5. Prototype 2.

Pada Gambar 5 merupakan *prototype* untuk fitur selanjutnya. Pada tampilan paling kiri merupakan tampilan untuk detail data dari pasien. Pada tampilan yang berada di tengah merupakan tampilan untuk pengguna dapat menambah pasien baru. Terakhir merupakan tampilan dialog ketika data akan dihapus.

#### IV. IMPLEMENTASI

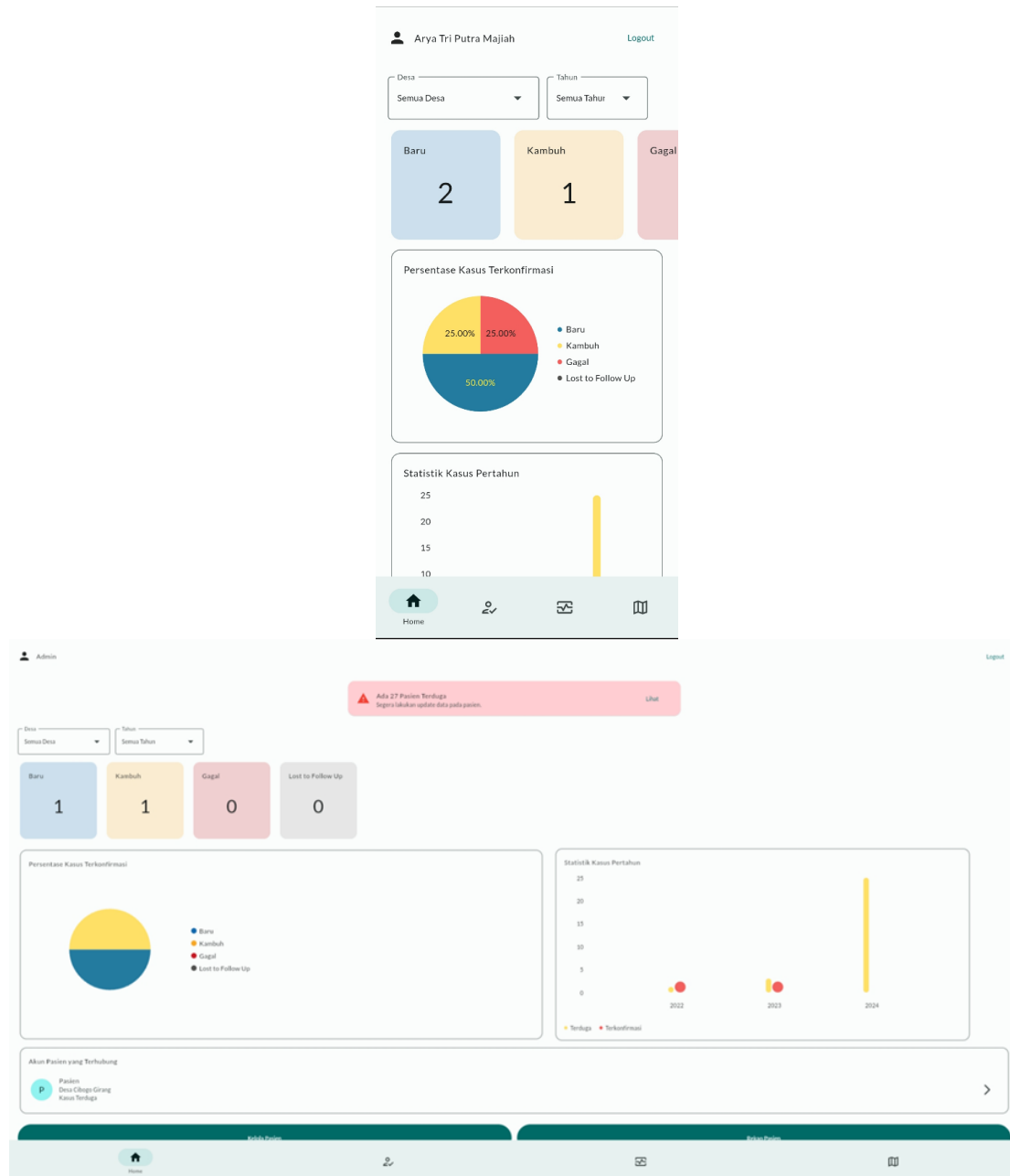
Tahap pengembangan aplikasi dibuat dengan bahasa pemrograman Dart dan menggunakan framework Flutter. Dikarenakan aplikasi ini memiliki beberapa modul, yang di mana di setiap modulnya dikerjakan oleh orang yang berbeda, maka dalam tahap pengembangan penulis memakai desain aplikasi Clean Architecture.

Pada sisi server *backend*, aplikasi ini menggunakan layanan dari Google Cloud Platform (GCP), tepatnya pada produk Firebase. Dikarenakan Firebase sendiri sudah dapat diintegrasikan dengan mudah dengan aplikasi Flutter, maka untuk menghubungkan aplikasi ini dengan server *backend* cukup simpel. Semua data yang ada disimpan dalam Firestore, yang terhubung dengan Firebase Authentication untuk keperluan autentikasi.

Proses pengembangan aplikasi ini dibantu dengan aplikasi manajemen proyek. Produk yang dipakai adalah Zube sebagai *agile management* yang membantu dalam pemantauan proses pengembangan. Produk kedua yang dipakai adalah Github, yang dipakai untuk menyimpan kode sumber dari aplikasi agar bisa terintegrasi dengan tim.

##### A. Halaman Dashboard

Terdapat beberapa halaman dalam modul *dashboarding*, salah satu halaman utama yang dipakai untuk menampilkan halaman pertama yang dilihat pengguna dan juga yang menampilkan *bottom navigation bar*, berada pada *file main screen*. Didalam *file* ini nantinya akan memuat 4 halaman didalamnya yaitu sesuai dengan fitur aplikasi ini yaitu *dashboard*, *screening*, *monitoring*, dan *tracking*.

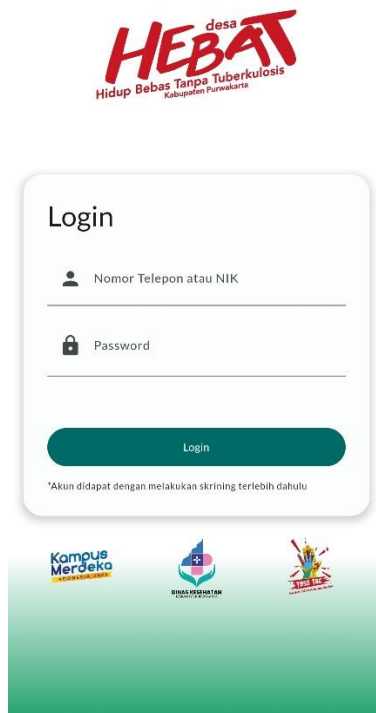


Gambar 6. Halaman Dashboard.

Halaman *dashboard* yang berada pada *file dashboard screen* merupakan halaman pertama ketika pengguna membuka aplikasi. Halaman ini dipanggil didalam *main screen*. Di dalam halaman ini, terdapat beberapa informasi terkait data pasien TBC yang dibutuhkan oleh pemangku kepentingan. Data-data dalam halaman ini juga disesuaikan dengan *role* pengguna yang sedang *login*, seperti misalnya notifikasi terkait ada pasien terduga yang perlu untuk di *follow up* ataupun tombol untuk mengarahkan pengguna ke halaman detail mengenai pasien. Gambar 6 merupakan tampilan dari halaman *dashboarding* ketika dibuka di dalam aplikasi dengan *smartphone*.

### B. Halaman Autentikasi

Pada modul *dashboard* juga termasuk halaman autentikasi ataupun *login*, halaman ini bisa ditampilkan jika pengguna belum melakukan *login* sebelumnya dan menekan tombol “*login*” yang ada pada sisi kanan atas halaman *dashboard*.



Gambar 7. Halaman Login.

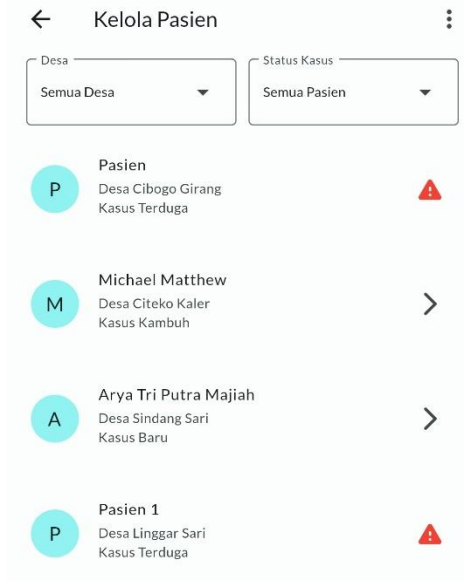
Pada halaman *login* ini memuat *form* dan juga tombol *login* yang dapat diisi oleh pengguna sendiri. Untuk membuat akun atau registrasi akun, pengguna harus melakukan skrining terlebih dahulu yang terdapat dalam modul skrining. Pada halaman *login*, data yang dimasukkan oleh pengguna akan langsung dikirimkan ke server melalui *bloc* yang berhubungan dengan *usecase* yang berkaitan. Pada Gambar 7 merupakan tampilan dari halaman *login* aplikasi ini.

Package *bloc* merupakan state management yang dipakai dalam pengembangan aplikasi ini, untuk Halaman login state ketika halaman belum login, proses login, dan ketika sudah login dikendalikan oleh package *bloc* ini.

### C. Halaman Kelola User

Terakhir pada modul *dashboard* juga terdapat beberapa halaman yang dibuat untuk mengelola data *user*. Halaman pertama adalah halaman yang menampilkan *list* semua data *user* yang ada.

Halaman ini dapat dibuka dengan menekan tombol Kelola Pasien yang ada pada halaman *dashboard*. Pada halaman *list* ini, terdapat beberapa fitur diantaranya yaitu filter menurut status kasus dan asal desa, dan juga fitur untuk mengekspor data ke dalam bentuk excel. Fitur untuk mengekspor data ini hanya bisa dilakukan melalui *website*, dikarenakan pada aplikasi dibutuhkan permintaan untuk mengakses data pada *handphone* pengguna. Pada Gambar 8 ini merupakan tampilan dari daftar pasien. Terdapat juga tombol untuk menambahkan data baru pada sisi kanan bawah halaman *list* pasien ini.



Gambar 8. Halaman Kelola User.

Halaman untuk menambahkan data pasien baru ini berisi *form* berupa data yang perlu diisikan oleh pengguna. Halaman ini hanya bertujuan untuk menambahkan data pasien yang sudah terkonfirmasi positif, untuk menambahkan data pasien yang belum terkonfirmasi maka dibutuhkan skrining mandiri terlebih dahulu. Pada halaman ini, penulis tidak menggunakan bloc, melainkan langsung menggunakan *usecase* yang diperlukan untuk menambah data, karena bloc tidak terlalu dibutuhkan dan bisa di-*handle* tanpa bloc. Pada Gambar 9 merupakan tampilan dari halaman untuk menambah pasien baru.

← Tambah Pasien ✓

NIK 0/16

Nama

Usia

Jenis Kelamin

Alamat

Desa

Nomor Telepon

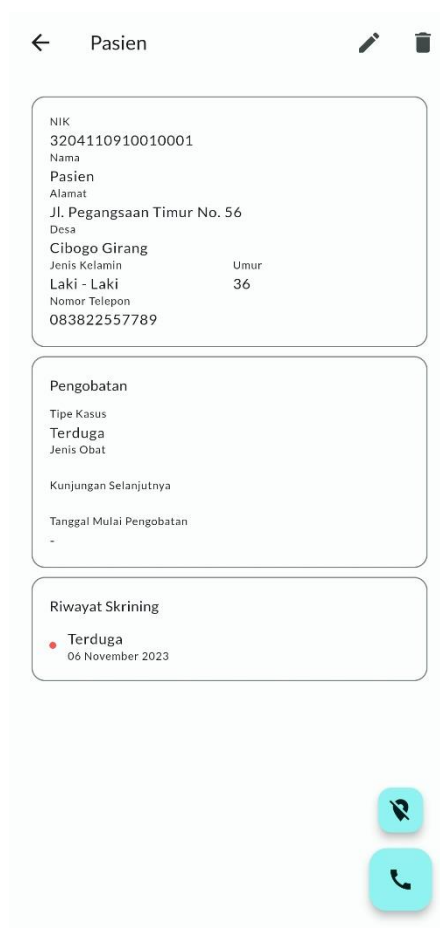
Jenis Kasus

Gambar 9. Halaman Tambah Pasien.



Halaman untuk memperbarui data dari pasien memiliki tampilan yang hampir serupa dengan tampilan untuk menambah data pasien, hanya saja pada halaman untuk memperbarui data, terdapat data *default* yang sudah terisi sebelumnya. Proses dalam pengiriman data juga memiliki cara yang hampir sama seperti menambah data baru. Data *default* yang ditampilkan pada halaman ini didapat dari halaman detail *user* dan dengan cara mengirimkan *id* dari *user*, kemudian memanggil data detail lainnya dari *database* menggunakan bloc.

Halaman detail *user* sendiri menggunakan bloc yang sama seperti pada halaman untuk memperbarui data *user*. Halaman ini bisa pengguna buka dengan menekan salah satu data pada halaman *list user*. Pada halaman ini terdapat data yang lebih detail dari pasien, dan pengguna juga dapat menggunakan datanya seperti nomor telepon dan lokasi yang memiliki tombol pada sisi kanan bawah untuk langsung menuju ke aplikasi yang berkaitan. Pada Gambar 10 merupakan tampilan dari halaman detail *user* ini.



Gambar 10. Halaman Detail Pasien.

Pada halaman ini juga terdapat beberapa aksi yang dapat dilakukan oleh pengguna seperti untuk menghapus dan juga memperbarui data. Untuk menghapus data pasien, pengguna dapat menekan tombol hapus berbentuk “tong sampah” yang terletak di sisi kanan atas pada halaman detail *user*. Tombol ini akan memunculkan modal konfirmasi apakah pengguna ingin benar-benar menghapus datanya. Untuk proses menghapus data ini menggunakan bloc yang sama dengan yang dipakai pada halaman detail *user*, hanya saja memanggil *event* bloc yang berbeda.

## V. KESIMPULAN

### A. Simpulan

Beberapa kesimpulan yang dapat penulis sampaikan yaitu:

1) Pembuatan fitur dashboarding pada aplikasi Hebat Mobile, dapat menampilkan data penting yang sesuai dengan pemangku kepentingan. Fitur juga sudah melewati proses pengujian dengan hasil keseluruhan baik, ditambah beberapa

improvisasi yang dapat dilakukan seperti fitur ekspor file pada platform mobile dan juga performa aplikasi yang dapat ditingkatkan kembali.

2) Pada sisi backend aplikasi, implementasi server dan database dapat dilakukan dengan menyimpan data pada Firestore, dan menggunakan Firebase Authentication yang dikombinasikan dengan Cloud Function untuk mengelola fitur autentikasi.

3) Proses Integrasi dalam penggabungan fitur-fitur menggunakan metode Agile dengan bantuan management tools Zube. Pada fitur yang memiliki tingkat kompleksitas yang tinggi dipakai metode branching pada Git.

4) Proses build aplikasi dapat dilakukan tanpa adanya error. Pengguna atau masyarakat bisa memakai aplikasi dengan memasang melalui apk atau membuka halaman website pada browser.

#### B. Saran

Program Desa HEBAT ini dapat membantu Dinas Kesehatan Kabupaten Purwakarta dalam menemukan kasus-kasus tuberkulosis baru. Salah satu saran yang dapat penulis sampaikan adalah untuk memperbesar cakupan dari aplikasi ini, yang sebelumnya hanya diterapkan di satu kecamatan saja.

Setelah mengikuti program Desa HEBAT, penulis mendapatkan pengetahuan baru mengenai penyakit tuberkulosis dan pengalaman untuk mengembangkan aplikasi kerja sama dengan pemerintahan. Selanjutnya, penulis perlu untuk meningkatkan komunikasi khususnya dalam bekerja sama dengan pihak lain.

#### DAFTAR PUSTAKA

- [1] WHO. (2022) Global Tuberculosis Report 2022. [Online]. Tersedia: <https://www.who.int/publications/i/item/9789240061729/>
- [2] Kementerian Kesehatan Republik Indonesia. (2023) Laporan Program Penanggulangan Tuberkulosis Tahun 2022. [Online]. Tersedia: [https://tbindonesia.or.id/pustaka\\_tbc/laporan-program-penanggulangan-tuberkulosis-tahun-2022/](https://tbindonesia.or.id/pustaka_tbc/laporan-program-penanggulangan-tuberkulosis-tahun-2022/)
- [3] Dinas Kesehatan Jawa Barat. Jumlah Kasus Penyakit Tuberkulosis Berdasarkan Kabupaten/Kota di Jawa Barat. [Online]. Tersedia: <https://opendata.jabarprov.go.id/id/dataset/jumlah-kasus-penyakit-tuberkulosis-berdasarkan-kabupatenkota-di-jawa-barat/>
- [4] Flutter. FAQ, Introduction, What is Flutter?. [Online]. Tersedia: <https://docs.flutter.dev/resources/faq/>
- [5] Firebase. Developer documentation for Firebase. [Online]. Tersedia: <https://firebase.google.com/docs/>
- [6] Cohn, M. Agile Estimating and Planning. [Online]. Tersedia: <https://www.oreilly.com/library/view/agile-estimating-and/9780137126347/>
- [7] Martin, R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. [Online]. Tersedia: <https://www.oreilly.com/library/view/clean-architecture-a/9780134494272/>