

BAB I PENDAHULUAN

1.1 Latar Belakang

Teknologi Informasi telah menjadi bagian dari kehidupan masyarakat dunia dengan perkembangannya yang sangat cepat dan terbukti berperan dalam berbagai kegiatan. Pengimplementasinya mendukung kinerja dan juga peningkatan efisiensi, efektivitas dan produktivitas bagi berbagai instansi, baik itu instansi pemerintah negeri, swasta atau bahkan perorangan, serta mendorong masyarakat untuk beradaptasi dengan kemajuan teknologi sehingga membentuk perwujudan masyarakat yang maju dan sejahtera.

Android menjadi sistem operasi yang paling banyak digunakan diantara sekian banyak pilihan sistem operasi untuk perangkat seluler. sifatnya yang terbuka untuk siapa saja, baik menggunakan atau berkontribusi dalam pengembangan aplikasi Android atau disebut dengan *open source* memudahkan pengembang untuk membuat aplikasi Android. Menurut (Statcounter, 2022), Pada bulan Oktober 2022 sebanyak 89,82% konsumen indonesia memilih *platform* Android untuk sistem operasi peranti *mobile* mereka, ini menandakan bahwa kebutuhan akan aplikasi berbasis android sangat tinggi.

Sistem Kewaspadaan Dini dan Respon adalah sebuah sistem yang memiliki fungsi penting dalam mendeteksi adanya ancaman indikasi (Kejadian Luar Biasa) KLB penyakit menular. Data yang digunakan untuk

sistem tersebut harus akurat, karena hasil data yang akurat sangat penting untuk dapat membuat keputusan yang sesuai.

Pola arsitektur (*Architectural Pattern*) dalam pengembangan perangkat lunak seperti *Clean Architecture* (Robert C. Martin, 2017), *Onion Architecture* (Jeffrey Palermo, 2008). *Onion Architecture* diperuntukan untuk pengembangan *website* berskala besar dan memiliki keberlanjutan proses bisnis dan juga memiliki perilaku yang kompleks. *Clean Architecture* mempunyai kelebihan ketika dihadapkan dengan proyek aplikasi berbasis Android dikarenakan kemudahan dalam penerapan aturan ketergantungan (*dependency rule*).

Pola desain (*Design Pattern*) seperti *Creational Pattern*, *Structural Pattern*, *Behavioral* adalah solusi umum yang telah teruji dan bisa digunakan kembali untuk menyelesaikan suatu masalah yang sering terjadi pada perancangan perangkat lunak. *Dependency Injection* merupakan bagian dari *Creational Pattern*, digunakan untuk mengurangi ketergantungan antar kelas-kelas ketika terjadi proses instansiasi atau penciptaan objek yang dibutuhkan untuk melakukan suatu proses.

Pada penelitian ini akan dilakukan perancangan dan pengembangan aplikasi berbasis Android yang dapat mengelola data untuk kebutuhan SKDR (Sistem Kewaspadaan Dini dan Respon) agar terhindar dari kesalahan perhitungan data. *Clean Architecture* dan *Dependency Injection* akan digunakan, sehingga akan menghasilkan aplikasi dengan kode yang bersih dan kokoh.

1.2 Rumusan Masalah

Dari uraian latar belakang penulis dapat merumuskan beberapa masalah sebagai berikut:

1. Bagaimana cara merancang dan membangun aplikasi yang dapat mengelola data untuk keperluan Sistem Kewaspadaan Dini dan Respon berbasis Android?
2. Bagaimana proses pengolahan data-data penyakit tersebut sampai menjadi informasi yang dapat digunakan untuk Sistem Kewaspadaan Dini dan Respon?
3. Bagaimana pengimplementasian *clean architecture* dan *dependency injection* pada pengkodean aplikasi?

1.3 Batasan Masalah

Untuk mempermudah pelaksanaan perancangan dan pengembangan sistem aplikasi pada penelitian ini agar tidak terlalu luas, maka penulis membatasi masalah yang akan dibahas pada penelitian ini sebagai berikut:

1. Penulis membahas mengenai pengembangan aplikasi yang dapat melakukan proses pengolahan data untuk keperluan Sistem Kewaspadaan Dini dan Respon yang dilakukan oleh survailans di UPTD Puskesmas Natar.
2. *Create, Read, Update, Delete* (CRUD) adalah proses pengolahan data yang akan digunakan dalam pengembangan aplikasi ini.
3. Penerapan *Clean Architecture* yaitu membagi aplikasi menjadi 3 (tiga) bagian atau lapisan (*layering*) yaitu, *Presentation, Domain dan Data*

juga memisahkan model untuk *domain* dan model untuk *data* (*separation model*).

4. Penerapan *Dependency Injection* yaitu dengan menerapkan *module* yang dapat melakukan proses *inject* kepada kelas-kelas yang membutuhkan seperti *ViewModel*, *Repository* dan *Database*.
5. Penggunaan 2 (dua) Prinsip dari *SOLID* yaitu *single responsibillity* dan *dependency inversion* sebagai panduan dari kode yang bersih (*clean code*) dan aturan ketergantungan untuk meminimalisir adanya ketergantungan yang sangat erat antar kelas (*tight coupling*).

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah penelitian, maka tujuan dari penelitian ini adalah:

1. Melakukan perancangan dan pengembangan aplikasi yang dapat mengelola (memasukan, menampilkan, memperbaharui, menghapus) data jenis-jenis penyakit mingguan yang dibutuhkan oleh survailans di UPTD Puskesmas Natar.
2. Untuk membantu survailans dalam melakukan proses perhitungan dan pengiriman laporan mingguan.
3. Membangun aplikasi yang bersih dari segi pengkodean dengan menggunakan *clean architecture* dan *dependecy injection*