

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 TINJAUAN PUSTAKA

Dari tahun ke tahun, perkembangan dan penggunaan internet secara global di dunia saat ini semakin meningkat. Hal tersebut tidak terlepas dari akses internet yang semakin mudah terjangkau dan tersebar sampai ke pelosok serta biaya akses yang semakin murah. Salah satu implikasi dari hal tersebut adalah terjadinya peningkatan penggunaan media sosial yang kian hari semakin banyak dan umum digunakan oleh manusia (Taufik, M. 2020).

Media sosial adalah sebuah media online, dengan para penggunanya bisa dengan mudah berpartisipasi, berbagi, dan menciptakan isi meliputi blog, jejaring sosial, wiki, forum dan dunia virtual. Blog, jejaring sosial yang merupakan situs dimana jejaring sosial merupakan situs dimana setiap orang bisa membuat *web page* pribadi, kemudian terhubung dengan teman-teman untuk berbagi informasi dan berkomunikasi. Jejaring sosial terbesar antara lain *Facebook*, *Myspace*, dan *Twitter*. Jika media tradisional menggunakan media cetak dan *media broadcast*, maka media sosial menggunakan internet (Fatmawati, Nurul. 2021).

Pesatnya perkembangan media sosial kini dikarenakan semua orang seperti bisa memiliki media sendiri. Jika untuk memiliki media tradisional seperti televisi, radio, atau koran dibutuhkan modal yang besar dan tenaga kerja yang banyak, maka lain halnya dengan media. Seorang pengguna media sosial bisa mengakses menggunakan jaringan internet bahkan yang aksesnya lambat sekalipun tanpa biaya besar, tanpa alat mahal dan dilakukan sendiri tanpa karyawan. Kita sebagai pengguna media sosial dengan bebas bisa mengedit, menambahkan, memodifikasi baik tulisan, gambar, video, grafis, dan berbagai model konten lainnya (Fatmawati, Nurul. 2021).

Berdasarkan laporan *We Are Sosial*, rata-rata pengguna internet yang mengakses media sosial menghabiskan waktu antara 60 menit hingga 180 menit

lebih dalam sehari untuk menggunakan media sosial. Laporan itu menunjukkan, semakin muda usia pengguna media sosial maka semakin lama durasi menggunakan media sosial. Dari semua kelompok usia, pengguna media sosial dengan durasi terlama yakni dari kategori perempuan (Annur Mutia, Cindy. 2022).

Tabel 2.1 Tabel Perbandingan Pustaka

No.	Penulis	Teknologi	Target Pengguna	Hasil
1.	Putra, R. (2023)	Kotlin	Masyarakat umum	Aplikasi Ngampu
2.	Kuncahya, Haris Dwi. (2020)	Kotlin	Masyarakat umum	Aplikasi Pengenalan Pahlawan Nasional dan Revolusi Indonesia
3.	Agustyani, Monika Dwi. (2020)	Kotlin	Masyarakat umum	Aplikasi Chatting
4.	Andika, Dika Sakti. (2022)	Kotlin, Firebase	Masyarakat umum	Aplikasi Pemesanan Jasa Antar-Jemput Laundry
5.	Kuswanto, Teguh Junian. (2021)	Kotlin	Masyarakat umum	Aplikasi Pemesanan Gilingan Padi
6	Edo, Anggoro. (2023)	Kotlin	Masyarakat umum	Aplikasi Samakaki

2.2 Dasar Teori

2.2.1. Kotlin

Kotlin merupakan sebuah bahasa pemrograman modern yang diterapkan dengan pendekatan statis, serta berjalan dibawah *Java Virtual Machine (JVM)*. Pengembangan Kotlin dilakukan oleh *JetBrains*, dan *Google* mengambil ini untuk mengembangkan aplikasi Android melalui *Android Studio*. Bahasa pemrograman Kotlin kini telah menjadi salah satu yang paling prestisius, digemari oleh para

pengembang berkat berbagai fitur bermanfaat dan bagus yang dimilikinya. Sehingga, bahasa ini mampu digunakan lintas berbagai platform dan sistem operasi. Adapun manfaat yang diperoleh dari pemanfaatan Kotlin dalam pengembangan perangkat lunak antara lain:

1. Open Source. Siapapun dapat menggunakan kotlin secara gratis, bahkan dapat berkontribusi untuk pengembangan kotlin.
2. Mudah Dipelajari. Bahasa kotlin sederhana dan ringkas sehingga mudah dipelajari.
3. Lebih ringan dari Java. *Running* dan *debugging* aplikasi pada emulator dan *real device* lebih cepat dari Java.
4. Bebas dan aman dari *NullPointerException* (NPE).
5. Terintegrasi dengan Android Studio.
6. *Interoperable*. Kotlin dapat dikolaborasi dengan Java. Tingkat *interoperability* dengan Java mencapai 100%. Artinya kotlin dapat berjalan pada java dan sebaliknya.
7. Dapat mengembangkan aplikasi server untuk iOS dan web.

Aplikasi Samakaki mengimplementasikan menggunakan Bahasa pemrograman Kotlin.

2.2.2. Dagger-Hilt

Dagger-hilt adalah sebuah *library dependency injection* yang dikembangkan oleh *Google* untuk bahasa pemrograman Kotlin. *Library* ini memungkinkan pengembang untuk memanfaatkan fitur *dependency injection* pada aplikasi Android dengan mudah dan efisien. *Dagger-hilt* memiliki beberapa kelebihan antara lain:

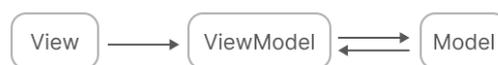
1. Mempermudah pengembangan aplikasi: *Dagger-hilt* memungkinkan kita untuk memisahkan antara kode fungsional dan kode *dependency injection*. Hal ini membuat pengembangan aplikasi menjadi lebih mudah dan terstruktur.

2. Meningkatkan performa aplikasi: *Dagger-hilt* menggunakan teknik *compile-time code generation*, sehingga tidak ada *overhead runtime* yang terjadi pada saat aplikasi dijalankan. Hal ini dapat meningkatkan performa aplikasi secara signifikan.
3. Memudahkan pengujian: *Dagger-hilt* memungkinkan pengembang untuk mengganti *dependency* pada saat pengujian dilakukan. Hal ini memudahkan pengujian dan mempercepat proses *debugging*.
4. Kompatibilitas dengan *Android Studio*: *Dagger-hilt* dapat diintegrasikan dengan *Android Studio*, sehingga pengembang dapat dengan mudah mengelola *dependency* pada aplikasi Android.
5. Dukungan dari komunitas: *Dagger-hilt* merupakan *library dependency injection* yang dikembangkan oleh *Google*, sehingga mendapatkan dukungan dari komunitas pengembang *Android* yang besar.

Dengan kelebihan diatas kita menggunakan *dagger-hilt* sebagai *dependency injection* pada aplikasi Samakaki.

2.2.3. Model-View-ViewModel (MVVM)

Model-View-ViewModel (MVVM) adalah salah satu arsitektur pembuatan aplikasi berbasis *GUI (Graphical User Interface)* yang berfokus pada pemisahan antara kode untuk komponen logika dan tampilan aplikasi. Dalam penerapannya, *MVVM* terbagi atas beberapa layer, yaitu *Model*, *View*, dan *ViewModel*.



Gambar 2.1 Design Pattern MVVM Layer

1. Model. Layer ini adalah model atau entitas yang merepresentasikan data yang akan digunakan pada logika bisnis. Biasanya kelas didalamnya berisi *POJO (Plain Old Java Object)* dan *Data Classes*.

2. **View.** Layer ini berisi UI dari aplikasi untuk mengatur bagaimana informasi akan ditampilkan. Layer ini akan berisi kelas-kelas, seperti *Activity* dan *Fragment*.
3. **ViewModel.** Layer ini akan bekerja untuk berinteraksi dengan layer model dimana data yang ada akan diteruskan ke layer view.

2.2.4. Coroutine

Coroutine adalah cara untuk mengelola eksekusi tugas secara konkuren dalam bahasa pemrograman Kotlin, dan cocok digunakan dalam pengembangan aplikasi Android untuk membuat kode yang berjalan secara asinkron menjadi lebih mudah diimplementasikan dan dipahami.

Di Android, *coroutines* membantu mengatur tugas-tugas yang berjalan dalam waktu lama yang mungkin akan menghambat *thread* utama dan membuat aplikasi menjadi tidak responsif. Berikut adalah beberapa fitur yang berguna untuk pengembangan aplikasi:

1. Ringan. Kita bisa menjalankan banyak *coroutines* pada *thread* tunggal karena *coroutines* mendukung suspensi, yang tidak menghentikan utas di mana *coroutine* berjalan. Suspensi ini menghemat memori dibandingkan dengan penghentian (*blocking*), sambil tetap mendukung banyak operasi konkuren.
2. Lebih sedikit penggunaan memory. Karena menggunakan konkurensi terstruktur untuk menjalankan operasi di dalam cakupan(*scope*).
3. Support cancellation: Pembatalan secara otomatis diinformasikan melalui hierarki *coroutine* yang sedang berjalan.
4. Terintegrasi dengan *Jetpack*. Dalam *library Jetpack* banyak *extensions* yang sudah mendukung penuh *coroutines*.

2.2.5. Extensible Markup Language (XML)

Extensible Markup Language (XML) merupakan markup language yang diciptakan oleh *World Wide Web Consortium*. Bahasa ini digunakan untuk menyederhanakan pertukaran dan penyimpanan data. XML sendiri biasanya

mengandung simbol-simbol khusus yang digunakan untuk mendeskripsikan isi dari sebuah halaman web. Biasanya XML dan HTML digunakan dalam pembuatan aplikasi berbasis website. XML ini memiliki tiga struktur pembentuk, berikut ini adalah ketiga struktur dari XML.

1. Deklarasi. Bagian deklarasi ini menunjukkan versi XML yang sedang kamu gunakan.
2. Atribut. Atribut merupakan bagian yang berisikan keterangan dari objek. Seperti judul atau sejenisnya.
3. Elemen. Struktur terakhir adalah elemen, Elemen ini berguna untuk mendeskripsikan objek dan terdiri dari tiga bagian yaitu tag pembuka, isi, dan tag penutup.