

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam proposal ini, penulis mengambil referensi tinjauan pustaka yang ditunjukkan pada table di bawah ini :

Table 2.1 Perbandingan Penelitian

Penulis	Judul	Bahasa Pemrograman	Hasil
Urfan, Aghnia Fi'la (2017)	Kalender Seni Kontemporer di DIY dengan Menggunakan Notifikasi Google Calender dan Pengarahan Lokasi Menggunakan Google Map Berbasis Android	Java	Menghasilkan aplikasi Android untuk memudahkan pengguna dalam mengakses informasi mengenai event di Yogyakarta.
Kosidin, dkk (2016)	Pemodelan Aplikasi Mobile Reminder Berbasis Android	Java	Menghasilkan aplikasi Android untuk membantu seseorang yang memiliki jabatan tinggi agar dapat mengingat kegiatan mereka.

Ramadhan, Taufik dan Victor Gayuh Utomo (2014)	Rancang Bangun Aplikasi Mobile Untuk Notifikasi Jadwal Kuliah	Java	Menghasilkan aplikasi Android untuk menyediakan informasi mengenai jadwal kegiatan mahasiswa pada STMIK Provisi Semarang.
Fransisco S B Thorion (diajukan)	Implementasi Push Notification Menggunakan Flutter Pada Aplikasi Reservasi Dokter Gigi Berbasis Android (Studi Kasus : Klinik Opal Dental).	Dart dan Kotlin	Aplikasi yang bisa digunakan untuk melakukan reservasi dokter gigi secara online.

Penelitian ini menggunakan beberapa pustaka yang berkaitan dengan Penelitian ini. Hal ini berfungsi untuk pedoman dan pembanding Penelitian yang akan dilakukan.

Urfan(2017) melakukan Penelitian berjudul Aplikasi Kalender Event Seni Kontemporer di DIY dengan Menggunakan Notifikasi Google Calendar dan Pengarahan Lokasi Menggunakan Google Map Berbasis Android. Aplikasi yang akan dibuat mengacu pada data dinas Kebudayaan dan dinas Pariwisata DIY mengenai *event* seni kontemporer yang sedang berlangsung di Yogyakarta yang bertujuan untuk memberikan informasi mengenai *event* seni kontemporer yang telah berlangsung, sedang berlangsung maupun yang akan diselenggarakan pertahunnya. Harapan dari aplikasi ini adalah pengguna dari aplikasi ini nantinya dapat mengetahui *event* seni kontemporer yang sedang berlangsung baik dari informasi *event* tersebut, lokasi, susunan acara, *review* mengenai *event* tersebut. Aplikasi yang buat nantinya akan

terhubung dengan google map sehingga pengguna tidak hanya diberikan informasi mengenai *event* nya saja namun juga akan diarahkan menuju lokasi event tersebut dan aplikasi ini terhubung dengan google calendar sehingga pengguna juga bisa menambahkan agenda pada google calendar sebagai pemberitahuan jika event yang akan dikunjungi akan diselenggarakan.

Kodisin dkk. (2016) melakukan Penelitian yang dilakukan oleh Kosidin dan timnya ditujukan untuk seseorang yang memiliki pangkat di suatu organisasi atau perusahaan. Yang menjadi sasaran adalah mereka yang berada di *top management* menjadi sasaran mereka karena orang-orang seperti merekalah yang memiliki kesibukan yang luar biasa. Sehingga mereka sering melupakan jadwal mereka. Harapan dari aplikasi ini adalah aplikasi yang dapat membantu para *top management* dalam mengatur jadwalnya. Pemodelan yang dilakukan oleh Kosidin berupa tahap Requirement Gathering, pemodelan diagrams *Use Case*, Pemodelan diagram *Activity*, Pemodelan diagram *Sequence*, dan perancangan tampilan aplikasi. Selain itu Kebutuhan dari perangkat keras dan perangkat lunak untuk aplikasi ini berupa perangkat yang memiliki ruang penyimpanan 4GB, perangkat bergerak dengan sistem operasi Android 4.2 yang memiliki jaringan 3G/2G/EDGE, alat yang digunakan untuk membangun aplikasi berupa Eclipse IDE Kepler, Android SDK, dan Android Development Tools pada sebuah komputer dengan 11 Hardisk 400GB dan prosesor Core i3.

Ramadhan dan Utomo (2014) melakukan Penelitian yang berjudul Rancang Bangun Aplikasi Mobile untuk Notifikasi Jadwal Kuliah Berbasis Android ini

merupakan fasilitas yang akan disediakan oleh STMIK Provisi Semarang. Adanya aplikasi ini merupakan wujud dari pelayanan bidang akademik yang ini menyediakan informasi jadwal kegiatan mahasiswa. Aplikasi yang dibangun akan berjalan pada system operasi Android. Metode perancangan yang digunakan adalah metode SDLC dengan 12 pendekatan waterfall. Mereka menggunakan metode ini dikarenakan tahapan dari metode ini membagi perancangan ke dalam fase-fase yang berurutan. Hasilnya adalah terciptanya aplikasi mobile pengingat berbasis mobile. Aplikasi ini memiliki 3 bagian. Yaitu halaman login. Halaman ini adalah yang pertama kali muncul pada aplikasi ini. Setelah pengguna masuk ke system maka akan muncul halaman menu. Pada halaman menu ini pengguna bisa memilih fungsi apa saja yang akan digunakan. Mahasiswa dapat melihat jadwal kuliah, kalender kegiatan, dan halaman atur jadwal. Selain itu aplikasi ini terdapat notifikasi yang digunakan sebagai pengingat. Adapun perbandingan tinjauan pustaka tersebut dapat dilihat pada Tabel Perbandingan Penelitian Aplikasi Reservasi Dokter Gigi Menggunakan Dengan Implementasi *Push Notification* Menggunakan Flutter Berbasis Mobile.

2.2 Dasar Teori

2.2.1 Flutter

Flutter merupakan platform pengembangan aplikasi *mobile* yang dikembangkan oleh Google untuk membuat aplikasi di *ios*, *Android*, *web app* dan juga *desktop app*. (Kurosaki, 2020) Beberapa kemampuan flutter yaitu memiliki *hot reload*, widget banyak dan memiliki tampilan yang menarik. (Giordano, 2019) Flutter

memiliki tujuan untuk menjadi pengubah dalam pengembangan aplikasi seluler. Flutter menyediakan semua alat yang dibutuhkan oleh pengembang untuk membuat aplikasi yang mengagumkan tanpa kekurangan dalam kinerja dan skalabilitas. Flutter dalam struktur intinya memiliki beberapa konsep yang fokus pada kinerja aplikasi dan antarmuka pengguna. Flutter menggunakan dart untuk menyediakan alat yang berkontribusi pada produktivitas pengembang dalam fase pengembangan dan untuk membangun aplikasi yang dioptimalkan untuk publikasi. (Biessek, 2019)

Flutter merupakan proyek *open-source* yang dihosting di github dengan kontribusi dari google dan komunitas. Flutter menggunakan dart, bahasa berorientasi objek modern yang mengompilasi ke kode *native ARM* dan kode javascript. Flutter menggunakan mesin *rendering Skia 2D* yang bekerja dengan berbagai jenis platform perangkat keras dan perangkat lunak dan juga digunakan oleh google chrome, chrome os, android, mozilla firefox, firefox os dan lain-lain.

Flutter menggunakan dart untuk membuat UI (*user interface*) pengguna. Flutter bersifat deklaratif, dengan kata lain flutter membangun UI untuk mencerminkan *state* aplikasi. Ketika *state* (data) berubah, UI akan berubah dan flutter membuat instance baru dengan widget. (Napoli, 2019).

2.2.2 Notifikasi

Menurut KBBI (Kamus Besar Bahasa Indonesia), notifikasi adalah pemberitahuan atau kabar tentang penawaran barang dan sebagainya. Notifikasi yang berkaitan dengan sebuah *system* dapat diartikan sebuah pemberitahuan yang dapat

diberikan suatu *system* kepada pengguna baik melalui *email*, ponsel, maupun internet. Notifikasi dapat berupa pemberitahuan yang berisi teks kata, gambar, video, maupun suara. (KBBI, 2019).

2.2.3 Smartphone

Smartphone merupakan *platform* yang menjalankan aplikasi-aplikasi *mobile*. (Prasetya, 2013) Sedangkan menurut Ali Zaki, *smartphone* yang berarti telepon pintar, merupakan telepon seluler yang memiliki kemampuan seperti PC walaupun terbatas. Selain itu, *smartphone* juga mendukung email dan organizer. Fitur lainnya adalah kemampuannya untuk ditambah aplikasi-aplikasi baru. (Zaki, 1999)

2.2.5 Kotlin

Kotlin adalah Bahasa pemrograman modern namun sudah matang yang bertujuan untuk membuat developer lebih bahagia. Ringkas, aman, dapat dioperasikan dengan Java dan Bahasa lain, dan menyediakan banyak cara untuk menggunakan kembali kode di antara berbagai platform untuk pemrograman yang produktif (kotlinlang.org, 2021)

2.2.6 Dart

Dart adalah bahasa pemrograman utama yang dipakai untuk mengembangkan *framework* flutter dan dikembangkan langsung oleh google. (Kurosaki, 2020) Flutter membutuhkan bahasa modern tingkat tinggi agar mampu memberikan pengalaman terbaik kepada pengembang dan memungkinkan pembuatan aplikasi seluler yang mengagumkan. Dart bisa digunakan untuk membuat aplikasi web, android, ios dan juga

menjalankan server. (Biessek, 2019) Dart dapat digunakan untuk membuat satu aplikasi yang kodingannya dapat digunakan di berbagai platform. Aplikasi yang dibuat juga bisa digunakan pada android maupun ios. Dart juga bisa digunakan untuk *web development*. (Wiraganda, 2019)

Dart diluncurkan pada tahun 2011 dan telah berkembang sejak saat itu. Dart merilis versi stabilnya pada tahun 2013 dengan perubahan besar termasuk rilis Dart 2.0 menjelang akhir 2018 :

- Dart fokus pada konsep pengembangan web dengan tujuan utama mengganti JavaScript: namun sekarang dart fokus kepada pengembangan *mobile* pada flutter.
- Dart mencoba memecahkan masalah JavaScript : JavaScript tidak menyediakan ketahanan dari gabungan beberapa bahasa. Jadi, Dart diluncurkan sebagai penerus JavaScript.
- Dart menawarkan kinerja terbaik dengan alat yang lebih baik untuk proyek skala besar: Dart memiliki *tools* yang modern dan stabil yang disediakan oleh plugin IDE, dart telah dirancang untuk mendapatkan kinerja terbaik.
- Dart menambahkan fitur OOP dan menyeimbangkan antara fleksibilitas dan ketahanan.

Dart adalah bahasa lintas platform modern yang terus meningkatkan fitur-fiturnya, membuatnya lebih baik dan fleksibel. Itulah mengapa flutter memilih dart sebagai bahasa yang digunakan. (Biessek, 2019).

2.2.7 Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna. Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas para pengembang untuk membuat aplikasi Android. (Suryana, 2018).

2.2.7 Firebase

Firebase adalah sebuah platform yang menyediakan beberapa layanan untuk memudahkan *developer* membangun aplikasi web, android dan ios maupun unity. (Akbar, 2018).

a. Cloud Firestore

Cloud firestore adalah database yang fleksibel dan skalabel untuk pengembangan seluler, web dan server dari firebase dan *google cloud platform*. *Cloud firestore* membuat data tetap terhubung di aplikasi klien melalui pemroses *realtime* dan menawarkan dukungan secara *offline* untuk seluler dan web. Dengan begitu, *developer* dapat mem-*build* aplikasi yang responsif dan mampu bekerja tanpa harus bergantung pada koneksi jaringan.

b. Firebase Authentication

Sebagian besar aplikasi perlu mengetahui identitas pengguna. Dengan mengetahui identitas pengguna, aplikasi dapat menyimpan data pengguna secara aman di *cloud*. *Firebase authentication* menyediakan layanan *backend*, SDK yang mudah

digunakan dan *library* UI siap pakai untuk mengautentikasi pengguna ke aplikasi yang dibuat. *Firebase authentication* mendukung autentikasi menggunakan sandi, email, nomor telepon, penyedia identitas gabungan yang populer seperti google, facebook, twitter dan lain-lain. *Firebase authentication* terintegrasi erat dengan layanan firebase lainnya dan sistem ini memanfaatkan berbagai standar industri seperti Oauth 2.0 dan OpenID Connect sehingga dapat dengan mudah diintegrasikan dengan *backend* kustom yang diinginkan.

c. **Firebase Cloud Messaging**

Firebase cloud messaging adalah solusi pertukaran pesan lintas platform yang dapat digunakan untuk mengirim pesan secara terpercaya tanpa biaya. Pengguna dapat mengirim pesan notifikasi untuk mendorong interaksi kembali dan retensi pengguna.

2.2.8 Push Notification

Push Notification merupakan salah satu fitur yang ditawarkan oleh Firebase Cloud Messaging yang digunakan untuk interaksi ke *user*. Implementasi dari salah satu fitur yang dimiliki oleh Firebase Cloud Messaging ini mencakup 2 komponen utama untuk mengirim dan menerima pesan, komponen yang pertama adalah lingkungan terpercaya seperti Cloud Functions for Firebase atau server aplikasi yang akan digunakan untuk membuat, menargetkan, dan mengirim pesan. Komponen yang kedua adalah aplikasi klien iOS, Android, atau Web (JavaScript) yang menerima pesan. (Google, 2019).