

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka merupakan acuan utama dalam beberapa studi yang pernah dilakukan yang berkaitan dengan penelitian ini. Terdapat beberapa penelitian yang digunakan sebagai acuan dalam penelitian ini.

Penelitian yang dilakukan oleh Sasmita Implementasi Framework Codeigniter pada pemesanan tiket kapal berbasis website dengan memanfaatkan Codeigniter sebagai pembuatan aplikasi yang akan membantu dalam memesan tiket secara *online* (Sasmita, 2021).

Aplikasi pada penelitian yang dilakukan oleh penulis ini menggunakan Framework Codeigniter dalam pembuatannya dan menggunakan database MySQL. Keunggulan Framework ini ringan, cepat minim konfigurasi dan *maintenance* yang mudah.

Penelitian yang dilakukan oleh Pembelian tiket kapal cepat *online* menggunakan teknologi Web *Service* pada aplikasi *android* yang akan membantu orang membeli tiket melalui *smartphone android* tanpa harus mengantri (Abdul Jafar, 2019).

Aplikasi pada penelitian yang dilakukan oleh penulis ini menggunakan aplikasi *mobile* yang mempermudah user dalam pemesanan tiket. Memiliki fitur reservasi mencetak tiket lewat aplikasi dan menggunakan teknologi web *service* dalam pengembangannya.

Penelitian yang dilakukan oleh Himawan Udin Hatari penulis 1, Sakina Sudin penulis 2, dan Gamaria Mandar penulis 3. Membuat Aplikasi pemesanan tiket kapal laut *online* berbasis *android*. Aplikasi ini mempermudah pelayanan masyarakat dalam pembelian tiket, mengetahui informasi jadwal keberangkatan, mengetahui harga tiket,

sehingga dengan adanya aplikasi ini proses pemesanan dapat dilakukan dengan cepat, mudah dan efisien (Hatari. Himawan et al., 2019).

Aplikasi pada penelitian yang dilakukan oleh penulis ini dibuat untuk membantu calon penumpang dalam hal mempermudah pemesanan tiket secara praktis. Pengembangan aplikasi ini menggunakan Android Studio sebagai pengembangan aplikasi dan website sebagai penyedia data untuk mempermudah pengelolaan tiket dan administrasi lainnya.

Penelitian yang dilakukan oleh Emanuel Krowe Goran Aplikasi pemesanan tiket kapal feri berbasis *android*. Aplikasi ini bertujuan untuk mempermudah masyarakat kota Nunukan dalam menggunakan transportasi kapal laut feri (Emanuel Krowe Goran, 2020). Aplikasi pada penelitian yang dilakukan oleh penulis ini menggunakan Android Studio sebagai alat pembuatannya dan firebase sebagai penyimpanannya atau databasenya. Aplikasi ini memiliki keunggulan dalam *real-time* databasenya yang dapat memberikan *user experience* lebih cepat dan lebih baik.

Penelitian yang dilakukan oleh Handayani dan Mochammad Rizki Romdoni Aplikasi *E-Booking* tiket kapal pada wilayah tanjungpinang dan kijang berbasis *android* dan web untuk memberikan kemudahan akses bagi pengguna memesan tiket pada wilayah tanjung pinang (Handayani & Mochammad Rizki Romdoni, 2019).

Aplikasi pada penelitian yang dilakukan oleh penulis ini menggunakan web dan *android* dalam pemesanan tiketnya yang membuat *user* lebih banyak pilihan dalam memesan tiket lewat aplikasi atau web. Keunggulan dari sistem ini *user* lebih nyaman dengan tersedianya semua layanan yang diberikan.

Penelitian yang dilakukan oleh Khairatin Pemilihan kapal laut menggunakan algoritma apriori dalam aplikasi reservasi tiket berbasis web dengan mengimplementasikan algoritma apriori untuk pemilihan kapal laut dan mengetahui

aturan asosiasi yang terbentuk berdasarkan nilai minimum *support* dan nilai minimum *confidence* yang ditentukan (Khairatin, 2019).

Aplikasi pada penelitian yang dilakukan oleh penulis ini menggunakan website yang memiliki keunggulan dengan merekomendasikan pemilihan kapal dengan Algoritma Apriori yang diberikan dalam website.

Perbandingan penelitian terdahulu dengan yang dikerjakan pada penelitian ini dapat dilihat pada Tabel 2.1 dibawah ini,

Tabel 2.1 Perbandingan penelitian terdahulu

Nama Penulis	Judul	Teknologi	Perbandingan
Muh Atang Sasmita (2021)	Implementasi Framework Codeigniter pada pemesanan tiket kapal berbasis web	Web	Aplikasi ini menggunakan Framework Codeigniter dalam pembuatannya dan menggunakan database MySQL. Keunggulan Framework ini ringan, cepat minim konfigurasi dan <i>maintenance</i> yang mudah.
Abdul Jafar (2019)	Pembelian tiket kapal cepat online menggunakan teknologi web service	Web Service	Aplikasi <i>mobile</i> yang dapat mempermudah user dalam pemesanan tiket ini memiliki fitur reservasi mencetak tiket lewat aplikasi dan menggunakan teknologi web <i>service</i> dalam pengembangannya

Himawan Udin Hatari penulis 1, Sakina Sudin penulis 2, dan Gamaria Mandar penulis 3 (2019)	Aplikasi pemesanan tiket kapal laut online berbasis android	Android	Aplikasi <i>android</i> ini dibuat untuk membantu calon penumpang dalam hal mempermudah pemesanan tiket secara praktis. Pengembangan aplikasi ini menggunakan Android Studio sebagai pengembangan aplikasi dan website sebagai penyedia data untuk mempermudah pengelolaan tiket dan administrasi lainnya.
Emanuel Krowe Goran (2020)	Aplikasi pemesanan tiket kapal feri berbasis android	Android	Aplikasi pemesanan tiket feri ini menggunakan Android Studio sebagai alat pembuatannya dan firebase sebagai penyimpanannya atau databasenyanya. Aplikasi ini memiliki keunggulan dalam <i>real-time</i> databasenyanya yang dapat memberikan <i>user experience</i> lebih cepat dan lebih baik.
Setya Handayani (2019)	Aplikasi e-booking tiket kapal pada wilayah tanjungpinang dan kijang berbasis android dan web	Android dan Web	Aplikasi ini menggunakan web dan <i>android</i> dalam pemesanan tiketnya yang membuat <i>user</i> lebih banyak pilihan dalam memesan tiket lewat aplikasi atau web. Keunggulan dari sistem ini <i>user</i> lebih nyaman dengan tersedianya semua layanan yang diberikan.
Khairatin (2019)	Pemilihan kapal laut menggunakan algoritma apriori dalam aplikasi reservasi tiket berbasis web	Web	Pemesanan kapal laut dengan website ini memiliki keunggulan dengan merekomendasikan pemilihan kapal dengan Algoritma Apriori yang diberikan dalam website

2.2 Dasar Teori

2.2.1 Framework

Framework adalah sebuah kerangka kerja yang digunakan untuk mempermudah para *developer software* dalam membuat dan mengembangkan aplikasi. Framework

berisikan perintah dan fungsi dasar yang umum digunakan untuk membangun sebuah *software* aplikasi sehingga diharapkan aplikasi dapat dibangun dengan lebih cepat serta tersusun dan terstruktur dengan cukup rapi. Framework juga bisa diartikan sebagai komponen – komponen pemrograman yang sudah jadi dan siap untuk digunakan kapan saja, sehingga pengembang aplikasi tidak perlu lagi membuat *script* yang sama untuk tugas – tugas yang sama. Beberapa alasan dari digunakan *framework* dalam membuat aplikasi adalah sebagai berikut.

1. Aplikasi akan memiliki standar pemrograman yang universal.
2. Menghindari *repetitive work*.
3. Membantu dalam *team work*.
4. Membantu dalam *maintenance* dan pengembangan aplikasi dimasa mendatang.
5. Hemat waktu dan biaya.

(Setiawan, 2021).

2.2.2 Flutter

Flutter adalah salah satu produk ciptaan Google yang dapat digunakan oleh para *developer* dalam membuat aplikasi lintas *platform* hanya dengan satu basis *code* (*codebase*). Teknologi *open source* ini memungkinkan *developer* untuk menghasilkan aplikasi yang dapat dipakai di berbagai *platform*, baik *mobile* Android, iOS, web, maupun desktop (Dharma, 2022).

Seperti diketahui, ketika ingin membangun sebuah aplikasi iOS, maka harus menulis *code* Swift/Objective C dan menggunakan *iOS Development Environment*. Begitu pun ketika membangun sebuah aplikasi Android yang harus menuliskan *code* dengan bahasa pemrograman Java atau Kotlin beserta penggunaan *Android Development Environment* (Dharma, 2022).

Ketika menggunakan Flutter, *developer* hanya memerlukan satu bahasa pemrograman yaitu Dart dan satu basis *code* untuk membuat aplikasi di *platform* berbeda. Dart sendiri merupakan bahasa yang diciptakan oleh Google pada tahun 2011 dan berfokus pada pengembangan *front-end* sehingga bisa dimanfaatkan untuk membuat aplikasi *mobile* dan web (Dharma, 2022).

Flutter memiliki dua komponen penting, yaitu *Software Development Kit* (SDK) dan *Framework User Interface* (UI).

1. *Software Development Kit* (SDK) merupakan sekumpulan *tools* yang berfungsi untuk membuat aplikasi agar bisa dijalankan di berbagai *platform*.
2. *Framework UI* merupakan komponen UI yang menyediakan fungsi, seperti teks, tombol, navigasi, dan lain-lainnya, yang dapat dikustomisasi sesuai kebutuhan (Dharma, 2022).

2.2.3 Dart

Dart merupakan bahasa modern dan berfitur lengkap. Dart juga memiliki banyak kemiripan dengan bahasa lain yang sudah banyak dikenal seperti Java, C#, Javascript, Swift, dan Kotlin. Salah satu rancangan utama dari Dart adalah supaya bahasa ini familiar bagi *developer* Javascript dan Java/C#. Artinya, yang telah familiar dengan kedua bahasa tersebut dapat memulai belajar bahasa Dart dengan lebih mudah. Namun, jika Anda adalah calon *developer* yang baru memulai belajar pemrograman dan memutuskan Dart sebagai *first language*, tenang saja. Dart adalah bahasa yang nyaman dan mudah dipelajari untuk memulai pemrograman.

Dart sebagai bahasa memiliki beberapa karakteristik berikut:

1. *Statically typed*,
2. *Type inference*,

3. *String Interpolation*,
4. *Multi-paradigm: OOP & Functional*.

(Dicoding, 2021).

Dart adalah bahasa yang *statically typed*, artinya kita perlu mendefinisikan variabel sebelum bisa menggunakannya. Potongan kode berikut adalah contoh deklarasi variabel pada Dart. Dart digunakan untuk menulis aplikasi *multi-platform*. Teknologi *compiler* yang fleksibel memungkinkan *code* Dart dapat dijalankan dengan cara yang berbeda, tergantung target *platform* yang dituju.

- **Dart Native** : Ditujukan untuk program yang menargetkan perangkat seluler, desktop, server, dan lainnya. Dart Native mencakup **Dart VM** dengan *compilation JIT (just-in-time)* dan *compiler AOT (ahead-of-time)* untuk menghasilkan kode mesin.
- **Dart Web** : Ditujukan untuk program yang menargetkan web. Dart Web menyertakan *compiler development (dartdevc)* dan *compiler production (dart2js)*.

(Dicoding, 2021).

2.2.4 State Management

State Management adalah sebuah cara untuk mengatur data / *state* kita bekerja, bisa juga untuk memisahkan antara *logic* dan *view*, dimana *logic* tersebut juga bisa *reusable*. Cara kerja *state management* seperti *provide* dan *listen*, artinya adalah kamu bisa memasukan *state* yang kemungkinan bisa berubah sewaktu waktu, lalu widget yang *subscribe (listen)* dengan *provider* yang kita buat akan berubah sesuai dengan *state* yang berubah. Setiap bagian dalam Flutter terbuat dalam bentuk widget. Terdapat dua widget

yang memiliki perlakuan *state* yang berbeda, yaitu *stateful* dan *stateless* widget (Alwi, 2022).

Stateless merupakan widget yang dimuat secara statis dimana seluruh konfigurasi yang dimuat di dalamnya telah diinisialisasikan sejak awal widget itu dimuat. Artinya, *state* jenis ini tidak dapat diubah dan tidak akan pernah berubah. *Stateless* biasa digunakan hanya untuk tampilan saja seperti *button*, *item*, *box container*, dan lain-lain (Alwi, 2022).

Stateful merupakan suatu widget yang sifatnya dinamis atau dapat berubah-ubah, kebalikan dari *stateless* widget. Pada *stateful*, kamu dapat menggunakan *property init State* yang berfungsi untuk menginisialisasi *state* yang akan pertama kali dijalankan. *Stateful* widget dapat mengubah atau mengupdate tampilan, menambah widget lainnya, mengubah nilai variabel, *icon*, warna, dan masih banyak lagi (Alwi, 2022).

2.2.4.1 Provider

Provider merupakan *state management* yang paling sederhana dan mudah digunakan. *Provider* menyediakan sebuah teknik mengolah *state* yang dapat digunakan untuk memmanage data didalam aplikasi. Manfaat menggunakan *provider* antara lain :

- a. Mengalokasikan resource menjadi lebih sederhana.
- b. *Lazy-loading*.
- c. Mengurangi *boilerplate* saat membuat kelas baru setiap saat.
- d. *Support* dengan *devtool*, karena status aplikasi kamu akan terlihat di Flutter *devtool*.
- e. Peningkatan skalabilitas untuk *class* yang memanfaatkan mekanisme *listen* yang dibangun secara kompleks.

(Alwi, 2022).

2.2.4.2 Riverpod

Riverpod mirip dengan *provider* yang *compile-safe* untuk digunakan dan teruji (*testable*). Kamu tidak akan menemukan *Provider Not Found Exception* atau *error* dalam menangani proses dalam mengelola *state*. Karena, selagi kode kamu berhasil di kompilasi, maka aplikasi akan tetap dijalankan. Riverpod mendukung *multiple providers* dengan tipe yang sama, proses *asynchronous*, dan mampu menambahkan *provider* dari file mana saja (Alwi, 2022).

2.2.4.3 Bloc

BLoC atau *Business Logic Component* adalah *design pattern* yang membantu kamu untuk memisahkan *presentation* dengan *business logic*. Sehingga komponen pada *project* terbagi menjadi *presentational component*, BLoC, dan *backend*. *Pattern* ini memperbolehkan *developer* untuk fokus dalam mengkonversikan *event* menjadi *state*. BLoC mengelola *state* dengan menggunakan pendekatan *stream* atau *reactive*. Secara umum, data akan bergerak dari BLoC ke *User Interface*, atau sebaliknya dalam bentuk *streams* (Alwi, 2022).

2.2.4.4 Getx

GetX merupakan salah satu pilihan terbaik untuk *state management* yang memiliki performa tinggi, memiliki *injection dependency* yang cerdas, serta memiliki manajemen *route* yang cepat dan praktis. GetX tidak akan memberatkan aplikasi kamu, meskipun memiliki fitur yang banyak, namun masing-masing fitur berada dalam *container* terpisah dan akan mulai dijalankan setelah dipakai. Misal, jika kamu hanya menggunakan *state management*, maka hanya *state management* lah yang akan di *compile*, tidak termasuk *route* dan lainnya. *State management* ini memiliki komunitas

yang besar, banyak kolaborator aktif. GetX mampu berjalan di OS Android, iOS, Web Linux, Mac, Windows, dan diserver (Alwi, 2022).

2.2.5 Firebase

Firebase memiliki produk utama, yaitu menyediakan *database real-time* dan *backend* sebagai layanan (*Backend as a Service*). Layanan ini menyediakan pengembang aplikasi API (*Application Programming Interface*) yang memungkinkan aplikasi data yang akan di sinkronisasi di klien dan disimpan di *cloud* Firebase ini. Firebase menyediakan *library* untuk berbagai *client platform* yang memungkinkan integrasi dengan Android, iOS, JavaScript, Java, Objective-C dan Node aplikasi Js dan dapat juga disebut sebagai layanan DbaaS (*Database as a Service*) dengan konsep *real-time* (Google, 2015).

Meskipun basis data menggunakan JSON *tree*, data yang tersimpan dalam basis data bisa diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe JSON (*JavaScript Object Notation*) yang tersedia untuk membantu dalam menulis lebih banyak kode yang bisa dipertahankan, struktur *tree* sendiri menggunakan metode *parent and child* (Google, 2015).

2.2.5.1 Firebase Database Firestore

Firebase Database Firestore adalah layanan *database cloud* yang disediakan oleh Firebase, platform pengembangan aplikasi mobile dan web yang dimiliki oleh Google. Firestore dirancang untuk memudahkan pengembangan aplikasi berbasis *cloud* dengan menyediakan *database* NoSQL yang mudah digunakan dan dapat disesuaikan dengan kebutuhan pengguna. Firestore dirancang untuk digunakan dalam pengembangan aplikasi *cloud* dengan kebutuhan data yang kompleks dan besar. Firestore menyediakan berbagai

fitur dan konsep yang memudahkan pengembang untuk membuat aplikasi yang dapat menangani data dalam skala besar, sambil tetap memastikan keamanan dan konsistensi data (Google, 2015).

2.2.5.2 Firebase Authentication

Sebagian besar aplikasi tentu membutuhkan autentikasi untuk mengetahui identitas pengguna. Mengetahui identitas pengguna memungkinkan aplikasi untuk secara aman menyimpan data pengguna di *cloud* dan memberikan pengalaman pribadi yang sama di semua perangkat pengguna. Firebase menyediakan layanan *backend*, SDK yang mudah digunakan dan pustaka siap pakai untuk mengautentikasi pengguna ke aplikasi. Saat ini Firebase mendukung autentikasi menggunakan sandi, sosial media populer seperti Google, Facebook, Twitter dan lainnya (Ilhami, 2018).

2.2.5.3 Firebase Cloud Messaging

Firebase Cloud Messaging merupakan solusi perpesanan lintas *platform* yang memungkinkan aplikasi mengirimkan pesan dan pemberitahuan ke aplikasi pengguna dengan tanpa biaya. Ada Firebase *Cloud Messaging*, aplikasi bisa memberitahu aplikasi pengguna ketika ada email baru atau pemberitahuan dan notifikasi (Ilhami, 2018).