

BAB I

PENDAHULUAN

A. Latar Belakang

Permainan olahraga badminton memerlukan sebuah lapangan. Lapangan terbuka memiliki faktor seperti angin yang dapat mempengaruhi permainan olahraga tersebut. Sehingga karena alasan tersebut, pemain yang ingin memainkan olahraga badminton biasanya menyewa sebuah lapangan atau Gedung Olahraga (GOR). Kebanyakan penyewaan lapangan dilakukan secara konvensional dengan cara datang langsung pada lokasi lapangan atau melalui telepon atau sosial media seperti *WhatsApp*. Hal tersebut kerap kali memicu terjadinya *miss communication* karena pencatatan yang tidak jelas. Calon penyewa lapangan kerap kali mengalami beberapa masalah terkait penyewaan lapangan. Dikarenakan sistemnya masih secara manual, tidak jarang terjadinya kesalahan pada sistem penyewaan lapangan. Calon penyewa telah melakukan penyewaan pada tanggal dan jadwal tertentu di awal namun kerap diberikan kepada penyewa lapangan lain sehingga terjadi masalah baru yaitu lapangan yang statusnya sedang disewa oleh 2 pihak yang berbeda.

Untuk mengatasi persoalan tersebut, GOR Depok Sport Center (DSC) ingin membangun sebuah aplikasi berbasis *mobile* untuk melakukan penyewaan lapangan menggunakan *framework flutter*. Aplikasi ini akan dapat mempermudah calon penyewa lapangan dalam melihat status dari lapangan dalam kurung waktu tertentu sehingga dengan mudah melakukan penyewaan lapangan. Selain itu juga dapat mempermudah pengelola GOR dalam mencatat status dari lapangan yang mereka miliki hingga tidak terjadi kesalahan dalam ketersediaan lapangan. Aplikasi ini juga diharapkan dapat mencatat segala jenis transaksi yang berhubungan dengan penyewaan lapangan seperti dapat melakukan pencatatan laporan.

Flutter adalah *framework open-source* dari Google yang memungkinkan para *developer* aplikasi dengan basis *code* tunggal dan mampu melakukan lintas platform dan proses pengembangannya yang terhitung

cepat[1]. *Flutter* memiliki beberapa keunggulan seperti dapat melakukan *hot-reload* dan *hot-restart* yang memungkinkan para penggunanya untuk melihat perubahan hasil *code* secara *realtime* tanpa harus menghentikan atau melakukan *build* aplikasi ulang[2]. Fitur ini sangat memudahkan pada pengembang aplikasi dan membuat waktu pengembangan aplikasi lebih efisien. *Flutter* juga menerapkan prinsip *Flutter Clean Architectur*. Prinsip ini membantu dalam menyusun aplikasi dengan cara yang terstruktur dan dan efisien serta dapat menyelesaikan masalah yang berkaitan dengan *management state* (pengolahan keadaan) dalam aplikasi[3]. Dengan menggunakan *clean arsitektur*, para pengembang aplikasi dapat memisahkan logika dari tampilan, peningkatan kemampuan pengujian serta fleksibilitas dalam pengembangan sehingga membuat aplikasi lebih mudah dalam proses pemeliharaan dan pengembangan[4].

B. Rumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah yang telah diuraikan sebelumnya, maka dapat dirumuskan bahwa permasalahan yang diteliti adalah “Bagaimana cara membangun sistem aplikasi yang akan mempermudah calon penyewa lapangan serta pihak GORD dalam melakukan transaksi sehingga tidak terjadi kesalahan yang menyebabkan satu lapangan dapat disewakan oleh dua pihak atau lebih”.

C. Batasan Penelitian

Dalam melakukan penelitian ini, penulis menerapkan beberapa batasan agar penelitian dapat dilakukan secara terarah, di antaranya sebagai berikut:

1. Bagaimana merancang sebuah aplikasi *mobile* yang dapat memudahkan pihak pengelola dan calon penyewa lapangan badminton.
2. Aplikasi *mobile* yang di bangun akan menggunakan *framework* terbaru yaitu *flutter* sehingga performa aplikasi diharapkan dapat lebih cepat.

D. Tujuan Penelitian

Tujuan penulis pengambil topik “Implementasi Aplikasi Mobile pada Penyewaan Lapangan Badminton Berbasis *Flutter*” adalah menyediakan aplikasi untuk mempermudah proses penyewaan lapangan dan mengelola lapangan yang ada.

E. Metode Penelitian

Pada penelitian ini, peneliti menggunakan metode pengembangan perangkat lunak *waterfall*. *Waterfall* dipilih oleh peneliti karena dianggap pantas dan tepat karena merupakan aplikasi pertama di mana kebanyakan aplikasi dibangun pertama kali menggunakan *waterfall*. Hal ini dilakukan karena beberapa faktor di antaranya waktu, biaya dan keefisienan kerja. *Waterfall* adalah metode pengembangan perangkat lunak dengan menggunakan sistem di mana pengerjaannya dilakukan fase per fase secara berurutan dan tidak dapat kembali ke fase sebelumnya[5]. Peneliti membagi menjadi beberapa bagian di antaranya Analisis kebutuhan, Desain, Implementasi Program, Evaluasi, dan Pemeliharaan.

1. Analisis Kebutuhan

Pada tahap ini peneliti melakukan analisa mengenai apa saja yang dibutuhkan terhadap aplikasi ke depannya. Peneliti melakukan beberapa wawancara terhadap pemilik lapangan, pengguna lapangan dan semua *stackholder* yang berperan dalam kegiatan penyewaan lapangan. Hasil analisa akan dirangkum menjadi daftar fungsionalitas yang akan ditambahkan ke dalam aplikasi ke depannya. Hal ini dilakukan demi mewujudkan aplikasi yang dapat memenuhi kebutuhan baik pihak pemilik lapangan maupun calon penyewa lapangan badminton.

2. Desain

Pada tahap desain, peneliti akan membuat desain dari aplikasi yang akan dibuat sesuai dengan kebutuhan yang telah dilakukan pada tahap sebelumnya. Tahap ini akan menghasilkan sebuah *output* berupa sebuah *mockup* atau *prototype* yang dibuat menggunakan *tools* desain seperti *figma*. *Prototype* yang akan dibuat oleh penulis akan dapat

dinavigasikan selayaknya aplikasi sungguhan sehingga *stackholder* dapat melihat gambaran aplikasi secara luas. Selain itu *stackholder* juga dapat memberikan masukan sebelum dilanjutkan pada tahap development aplikasi.

3. Implementasi Program

Pada tahap implementasi, akan dilakukan *pengkodean* aplikasi sesuai dengan mockup yang telah ada sebelumnya. Peneliti menggunakan *framework flutter* sebagai *frontend* dengan bahasa pemrograman *dart*. Peneliti juga akan menggunakan *express.js* sebagai *backend* dari program dengan menggunakan bahasa pemrograman *dart*. Peneliti juga menggunakan *Integrated Development Environment (IDE)* atau *text editor Visual Studio Code*. Pada tahap ini akan menghasilkan *output* berupa sebuah program aplikasi. Peneliti akan membangun aplikasi tersebut di atas *Operating System Android*.

4. Evaluasi

Pada tahap ini, Peneliti akan memberikan output pada program sebelumnya berupa aplikasi dan akan diberikan kepada calon pengguna untuk dilakukan uji coba fungsionalitas yang telah disepakati sebelumnya. Kelayakan aplikasi akan didasarkan atas saran dan masukan yang akan diberikan oleh calon pengguna aplikasi apakah semua kebutuhan fungsionalitas telah berjalan sesuai dengan kebutuhan calon pengguna. Pada tahap ini semua calon pengguna serta penguji akan mencari apakah adanya *bug*. Jika ditemukan *bug* atau kesalahan akan fungsionalitas akan diperbaiki dan akan dilakukan uji coba kembali hingga sesuai dengan kebutuhan calon pengguna.

5. Pemeliharaan

Tahap pemeliharaan akan dilakukan setelah tahap evaluasi selesai. Aplikasi akan dapat digunakan oleh pengguna secara konvensional. Pada tahap ini juga akan melakukan perbaikan jika ditemukan *bug* ketika aplikasi telah digunakan untuk umum. Pemeliharaan biasanya akan dilakukan secara bertahap sesuai dengan

permintaan dari pengguna. Pemeliharaan juga dapat dilakukan ketika adanya urgensi atau *bug* yang sangat fatal sehingga dapat menghambat kinerja dari aplikasi. Namun jarang terjadi karena biasanya *bug* fatal akan semakin minim ditemukan pada tahap pemeliharaan karena sebagian besar sudah di *fix* ditahap sebelumnya.

F. Sistematika Penulisan

Sistematika penulisan tugas akhir ini dibagi menjadi enam bab. Setiap bab akan memaparkan tujuan masing-masing berdasarkan peranannya. Berikut penjelasan dari setiap bab:

a. BAB I PENDAHULUAN

Pada bab ini akan menjelaskan mengenai latar belakang masalah, batasan masalah, tujuan dari penelitian, landasan teori dan sistematika dari penulisan.

b. BAB II TINJAUAN PUSTAKA

Pada bab dua akan menjelaskan perbandingan akan penelitian sebelumnya yang memiliki keterkaitan dengan tugas akhir yang ditulis oleh penulis sekarang.

c. BAB III LANDASAN TEORI

Pada bab tiga menjelaskan mengenai teori-teori berdasarkan sumber, buku ataupun jurnal. Teori yang dijelaskan hanya yang memiliki hubungan dengan penelitian yang sedang dikerjakan oleh penulis sekarang.

d. BAB IV ANALISIS PERANCANGAN SISTEM

Pada bab empat akan menjelaskan hasil analisis dari penelitian yang dilakukan oleh penulis. Analisis-analisis yang dijelaskan adalah analisis sistem, analisis lingkup masalah, perspektif aplikasi, fungsi aplikasi, kebutuhan antarmuka, perancangan data, perancangan arsitektur dan perancangan antarmuka.

e. BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab lima akan menjelaskan hasil dari implementasi *code* terhadap aplikasi, menjelaskan fitur dan menjelaskan hasil pengujian fungsi-fungsi dari aplikasi.

f. BAB VI PENUTUTUP

Pada bab enam akan menjelaskan mengenai kesimpulan dan saran dari tugas akhir penulis.

