



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Tinjauan Teori

2.1.1 COVID-19

COVID-19 atau *Corona Virus Disease* adalah salah satu pantogen utama (virus atau bakteri) yang menyerang sistem pernafasan manusia. Virus corona memiliki kemiripan dengan virus pada wabah SARS (*Severe Acute Respiratory Syndrome*) dan MERS (*Middle East Respiratory Syndrome*) dimana ketiga virus ini merupakan virus yang sangat mematikan bagi manusia. Virus ini pertama kali menangkap perhatian dunia kesehatan pada awal bulan Desember 2019 dimana terdapat banyak pasien dengan diagnosa awal penyakit pneumonia pada rumah sakit – rumah sakit pada kota Wuhan, China [5]. Walaupun memiliki kemiripan dengan virus SARS dan MERS, virus corona memiliki karakteristik penularan yang lebih mudah jika dibandingkan kedua virus tersebut.

Virus corona akan membuat pasien yang terkena menjadi kesulitan bernafas dengan gejala demam, batuk dan badan lemas. Gejala jika terkena virus ini akan mulai terlihat setelah 5 hari setelah terjadinya kontak dengan virus. Dikarenakan obat untuk virus COVID-19 belum ditemukan maka pasien yang terkena virus hanya dapat dirawat dan diberikan bantuan pernafasan (jika parah). Virus COVID-19 selain menyerang pernafasan juga akan melemahkan imun tubuh sehingga pasien yang terinfeksi virus ini akan sangat rentan terkena penyakit lainnya. Empat belas hari setelah terinfeksi virus merupakan saat saat yang paling kritis dalam menentukan apakah pasien yang terjangkit virus COVID-19 dapat bertahan. Setelah melewati masa tersebut maka sistem imun pada tubuh pasien pada umumnya akan mulai bekerja dan menciptakan *anti-body* untuk melawan virus COVID-19. Virus COVID-19 mulai terdeteksi di negara Indonesia pada bulan Maret 2020. Jumlah pasien yang terjangkit virus COVID-

19 pada negara Indonesia per tanggal 20 Oktober 2020 sejumlah 369.000.000 orang dengan rata-rata penambahan pasien sejumlah 3.000 orang per-hari.

2.1.2 Cross-Platform Application

Cross platform atau dalam Bahasa Indonesia disebut dengan lintas platform merupakan sebuah konsep dimana sebuah aplikasi dapat digunakan pada sistem operasi yang berbeda. Konsep ini dikembangkan agar dapat digunakan pada platform *mobile* karena hadirnya lebih dari satu sistem operasi yang umum digunakan pada Telepon genggam yaitu sistem operasi Android dan iOS. Konsep *cross-platform* pertama kali digunakan pada platform *desktop* dimana terdapat tiga sistem operasi yang umum digunakan yaitu Windows, Linux dan Mac OS. Konsep ini memungkinkan penggunaan aplikasi yang sama pada suatu sistem operasi dengan sistem operasi yang lain.

Konsep *Cross-Platform* terbagi menjadi tiga jenis yaitu [3] :

- *Hybrid Web Application*

Hybrid web application adalah metode pengembangannya aplikasi *cross-platform* yang pengembangannya ditujukan untuk digunakan pada sebuah *web browser*. *Hybrid web application* adalah jenis *cross-platform* yang paling mudah untuk dikembangkan karena aplikasi yang dikembangkan menggunakan metode *cross-platform* jenis ini dapat bekerja di seluruh perangkat yang memiliki sebuah *web browser* sehingga tidak terbatas pada *desktop* dan *mobile*. Tetapi aplikasi yang dikembangkan menggunakan metode ini dinilai tidak memiliki *user experience* yang baik karena aplikasi yang dikembangkan dengan metode ini tidak dapat menggunakan *Native UI component* dari perangkat dan memiliki kecepatan respon yang lambat karena seluruh prosesnya bergantung pada koneksi internet dan kapabilitas *web browser* yang digunakan.

- *Interpreted Application*

Interpreted Application adalah metode pengembangannya aplikasi *cross-platform* dimana dalam pengembangannya dikembangkan menggunakan bahasa pemrograman umum (dapat dijalankan oleh seluruh sistem operasi yang dituju) tetapi pada saat aplikasi akan dijalankan oleh suatu sistem operasi maka bahasa pemrograman umum tersebut akan dikonversikan oleh *compiler* menjadi bahasa *native* yang digunakan oleh sistem operasi agar dapat mengakses API. Kekurangan dari metode ini adalah dikarenakan bahasa pemrograman yang digunakan oleh pihak pengembang aplikasi berbeda dengan bahasa pemrograman yang dijalankan secara *native* oleh sistem operasi maka proses perubahan bahasa tersebut akan menyebabkan terjadi *performance issue* atau perangkat yang menjalankan aplikasi tersebut akan terasa menjadi lebih lambat dan tidak secepat aplikasi yang dikembangkan secara *native*.

- *Cross-compiled Application*

Cross-compiled application adalah metode pengembangannya aplikasi *cross-platform* dimana aplikasi yang dikembangkan oleh bahasa ini disusun menggunakan bahasa yang sama untuk sistem operasi yang berbeda, tetapi pada saat aplikasi di *compile*, maka bahasa tersebut akan langsung diproses oleh *cross-compiler* menjadi bahasa yang *native* pada sistem operasi yang dituju. Perbedaannya dengan metode *interpreted application* adalah *cross-compiled application* akan mengubah kode dengan bahasa umum tersebut menjadi kode dengan bahasa *native* dari sistem operasi pada saat aplikasi sedang dikompilasi atau diselesaikan. Sedangkan *Interpreted application* mengubah bahasa umum tersebut menjadi bahasa *native* pada saat aplikasi dijalankan. Oleh karena itu aplikasi yang dikembangkan menggunakan metode *cross-compiled application* tidak akan terasa lambat dan kurang *responsive*. Tetapi metode ini memungkinkan perlunya penyesuaian kode pada beberapa sistem operasi khusus. Contohnya adalah akses untuk menggunakan sensor NFC dapat diakses dengan mudah pada sistem operasi Android namun pada sistem

operasi iOS akses untuk menggunakan sensor NFC tidak dapat digunakan dengan bebas karena alasan keamanan.

2.1.3 Flutter

Flutter adalah sebuah *developer SDK (software development kit)* yang dapat ditulis menggunakan bahasa pemrograman C, C++ dan Dart [3]. Flutter dikembangkan oleh pihak google dan memiliki tujuan untuk melakukan pengembangan aplikasi mobile yang bersifat *cross-platform* atau dapat dijalankan pada lebih dari satu jenis sistem operasi. Selain untuk melakukan pengembangan aplikasi mobile, Flutter juga dapat digunakan untuk mengembangkan sebuah aplikasi yang ditujukan untuk *desktop* atau aplikasi yang bersifat *web based*. Flutter menyediakan banyak API (*Application Programming Interface*) dan *widget* UI sehingga pengembangan aplikasi menggunakan Flutter akan memberikan sebuah aplikasi dengan tampilan yang sederhana dan modern. Performa aplikasi merupakan salah satu keunggulan Flutter, aplikasi yang dikembangkan menggunakan Flutter memiliki performa yang hampir sama dengan aplikasi yang dikembangkan secara *Native* (Java dan Kotlin untuk android dan Swift untuk iOS) walaupun Flutter memiliki banyak *widget* UI yang menggunakan animasi-animasi yang cukup berat [4].

Flutter dapat mendukung pengembangan aplikasi yang cepat dengan fitur *Hot Reload*. Fitur ini memungkinkan pihak pengembang aplikasi untuk dapat melihat perubahan secara langsung pada saat kode dari aplikasi tersebut diubah. Sehingga perubahan pada aplikasi yang sedang dikembangkan dapat langsung dilihat tanpa harus melewati proses *compiling* aplikasi. Penggunaan flutter juga dinilai sangat ringan jika dibandingkan dengan pengembangan aplikasi yang bersifat *native*. Pengembangan aplikasi flutter hanya memerlukan Flutter SDK dan sebuah *text editor*. Aplikasi yang dikembangkan menggunakan flutter bersifat *cross-platform* yang berarti aplikasi tersebut dapat dijalankan pada sistem operasi Android dan iOS, faktor tersebut memperkecil biaya yang diperlukan oleh pihak pengembang aplikasi dan membantu pihak pengembang aplikasi yang kekurangan tenaga karena dengan

menggunakan satu kode pada Flutter, program dapat berjalan di dua sistem operasi yang berbeda dan dengan konsistensi aplikasi yang sama.

2.1.4 Prototyping

Prototyping atau metode *prototype* adalah sebuah model pengembangan aplikasi yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuah hasil akhir berupa sebuah *prototype* dari ide tersebut [6].

Dalam metode *prototyping* terdapat 7 fase utama yaitu:

1. Fase analisa kebutuhan

Fase analisa kebutuhan adalah fase dimana peneliti selaku pengembang aplikasi atau sistem melakukan identifikasi terhadap kebutuhan-kebutuhan yang diperlukan oleh aplikasi atau sistem yang akan dikembangkan.

2. Fase membangun *prototyping*

Fase membangun *prototyping* adalah fase dimana peneliti sebagai pengembang aplikasi atau sistem melakukan perancangan sementara yang memiliki fokus terhadap proses *input* (masukan) dan *output* (keluaran) yang dibutuhkan dan dihasilkan oleh aplikasi yang akan disusun.

3. Fase evaluasi *prototyping*

Fase evaluasi *prototyping* adalah fase yang bertujuan untuk memastikan apakah seluruh proses *input* dan *output* yang diperlukan dan dihasilkan oleh aplikasi telah sesuai dengan tujuan utama penelitian.

4. Fase mengkodekan sistem

Fase mengkodekan sistem adalah fase dimana peneliti sebagai pengembang aplikasi atau sistem menerjemahkan rancangan aplikasi dan sistem yang telah disusun menjadi sebuah aplikasi yang dapat berjalan menggunakan

kerangka kerja pemograman, bahasa pemograman, dan bahasa struktur data pemograman.

5. Fase pengujian aplikasi atau sistem

Fase pengujian aplikasi atau sistem adalah fase yang berfungsi untuk memastikan bahwa setiap *input* dan *output* yang dibutuhkan dan dihasilkan oleh aplikasi atau sistem tersebut telah berjalan dengan benar.

6. Fase evaluasi sistem

Fase evaluasi sistem adalah fase yang bertujuan untuk memastikan bahwa aplikasi atau sistem yang dikembangkan telah dapat memenuhi tujuan utama dari penelitian yang dilakukan.

7. Fase menggunakan sistem

Fase menggunakan sistem adalah fase yang dilakukan dengan cara melakukan peragaan aplikasi dengan contoh kasus nyata agar hasil dari penelitian dapat diamati.



2.2 Penelitian Terdahulu

Tabel 2.1 Daftar penelitian terdahulu

Pengarang	Nama Artikel	Nama Jurnal	Tahun Jurnal
Kuan-Yu Lin Chih-Hun Chen Zhe-Ming Zhang Sheng-Chuan Ou	<i>NFC-based mobile application design restaurant ordering system APP</i>	<i>Proceedings of 4th IEEE International Conference on Applied System Innovation 2018, ICASI 2018</i>	2018
I Kadek Kris Sanjaya Putu Wira Buana I Made Sukarsa	<i>Designing Mobile Transactional Based Restaurant Management</i>	<i>International Journal of Computer Engineering and Information Technology Vol. 11 No.6, June 2019</i>	2019
Jos Forman T. Steven R. Sentinuwo Alicia A. E. Sinsuw	Rancang Bangun Aplikasi Pemesanan Menu Makanan Restoran Berbasis Android	E-journal Teknik Informatika, Vol 9, No.1 October 2016	2016

Perbedaan penelitian ini dengan penelitian-penelitian yang tertera pada tabel perbandingan di atas adalah penelitian ini akan mengembangkan sebuah aplikasi dengan konsep *cross-platform* dan menerapkan konsep *Flutter* dalam pengembangannya.

Penelitian rancang bangun aplikasi yang dilakukan oleh Kuan-Yu Lin, Chih-Hun Chen, Zhe-Ming Zhang, dan Sheng-Chuan Ou pada tahun 2018 dengan judul *NFC-based mobile application design restaurant ordering system APP*. Pada penelitian ini Kuan-Yu Lin dan rekan-rekannya melakukan rancang bangun aplikasi pemesanan makanan dengan memanfaatkan teknologi RFID dan NFC yang terdapat pada Telepon genggam. Penelitian ini juga membahas tentang teknologi RFID dan metode pengembangan aplikasi Telepon genggam yang dilakukan. Pada penelitian ini proses pengembangan aplikasi masih menggunakan konsep *native* yang pengembangannya hanya ditujukan untuk sistem operasi Android namun arsitektur sistem yang dijabarkan dapat dijadikan referensi pada saat melakukan pembuatan aplikasi Telepon genggam untuk pengembangan dengan konsep *cross-platform* [7].

Penelitian rancang bangun aplikasi yang dilakukan oleh I Kadek Kris Sanjaya, Putu Wira Buana dan I Made Sukarsa pada tahun 2019 dengan judul

Designing Mobile Transactional Based Restaurant Management. Pada penelitian ini I Kadek Kris Sanjaya dan rekan-rekannya melakukan penelitian untuk rancang bangun aplikasi manajemen *restaurant*. Penelitian ini lebih tertuju untuk melakukan analisa *flow* dan arsitektur sistem yang telah ada dan apa yang perlu diperbaiki arsitektur tersebut. Penjelasan *flow* dan arsitektur sistem yang dibahas pada penelitian ini dijadikan referensi pada saat melakukan perancangan alur UI dari aplikasi yang akan dibuat [8].

Penelitian rancang bangun aplikasi yang dilakukan oleh Jos Forman Tompoh, Steven R. Sentinuwo, dan Alicia A. E. Sinsuw pada tahun 2016 dengan judul Rancang Bangun Aplikasi Pemesanan Menu Makanan Restoran Berbasis Android. Pada penelitian ini Jos Forman Tompoh dan rekan rekannya melakukan rancang bangun aplikasi pemesanan menu makanan dengan metode pengembangan *Rapid Application Development* dan menggunakan konsep *cross platform* dengan teknologi Ionic. Penelitian ini menjelaskan pengembangan sistem dari fase analisa kebutuhan pengguna, pembuatan diagram *use case*, perancangan *database*, proses penulisan program dan pembuatan UI serta proses pengujian aplikasi. Seluruh proses tersebut adalah alur dasar rancang bangun sebuah sistem aplikasi Telepon genggam yang dijadikan referensi dalam penelitian ini [9].

Berikut adalah penelitian terdahulu yang mendefinisikan perbedaan konsep *cross-platform* dan *framework* yang digunakan:

Penelitian perbandingan metode rancang bangun aplikasi cross-platform yang dilakukan oleh Awel Eshetu Fentaw pada tahun 2020 dengan judul Cross platform mobile application development: a comparison study of React Native Vs Flutter [3].

Penelitian perbandingan metode rancang bangun aplikasi cross-platform yang dilakukan oleh Wenhao Wu pada tahun 2018 dengan judul React Native vs Flutter, cross-platform mobile application frameworks [4].