BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengguna *smartphone* saat ini didominasi oleh dua platform yaitu *Android* dan iOS yang mengharuskan beberapa perusahaan dan *developer* memiliki aplikasi yang harus berjalan pada kedua platform tersebut. Setiap platform memiliki bahasa dan cara pengembangan yang berbeda tentunya membutuhkan *programmer* yang khusus menangani *Android* dan iOS atau keduanya, seperti yang terlihat dalam statistik dari *Developer Economics*, 80% perusahaan masih memiliki tim pengembang yang semata-mata didedikasikan hanya untuk satu platform [1].

Seiring dengan itu *Facebook* mengembangkan sebuah *framework* React Native untuk membuat sebuah aplikasi native menggunakan *React.* Dengan React Native *developer* dapat membuat aplikasi sekaligus untuk *Android* dan iOS dalam satu *project* React Native. Pengguna React Native pun semakin banyak dilihat dari halaman *Github*-nya terdapat sekitar 90.800 lebih orang mengikuti perkembangan React Native. Dilihat dari halaman *website* React Native beberapa aplikasi besar seperti *Facebook, Instagram, Shopify, Skype, Uber, Pinterest, Airbnb* menggunakan React Native dalam pembuatan aplikasi tersebut.

User interface React Native sendiri dibangun berdasarkan component ReactJs. *User interface* menjadi bagian yang sangat penting dalam suatu aplikasi karna dengan *user interface* pengguna dapat berkomunikasi dengan sistem melalui *Command Line Interface* (CLI) ataupun *Graphical User Interface* (GUI). Penggunaan GUI yang mengadopsi grapik dan ikon sebagai *user interface* jauh lebih *user-friendly* dan efektif dibanding CLI [2

Pembuatan *user interface* menggunakan React Native tidaklah mudah karna harus mengetahui banyak *component, props, attribute* dan *style* dari *resource* tersebut. Selain itu metode pembuatan *user* interface manual dengan pengkodean tangan, metode seperti itu memiliki resiko *human error* penulisan *code*.

Pendekatan pada *Model Driven Architecture* dapat memberikan solusi dengan menawarkan cara automatis dalam menghasilkan beberapa bagian pada pembuatan aplikasi. Dengan *Model Driven Architecture* setiap model GUI untuk *mobile* aplikasi ditransformasikan menjadi *Platform Independent Model* (PIM), kemudian diubah menjadi *Platform Specific Model* (PSM) dengan target React Native. Setiap model PSM kemudian ditransformasikan menjadi *source code* untuk React Native yang di improvisasikan melalui sebuah aplikasi yang dapat men-*design user interface* serta secara automatis dapat men-*generate* code sesuai *component, props, attribute* dan *style* dari model PSM React Native.

Dibandingkan dengan pengembangan tradisional seperti model *waterfall* dalam pengembangan perangkat lunak *Model Driven Architecture* dapat melewatkan *coding* setelah *design* dalam pembuatan *user interface* dari tahapan proses model *waterfall*, sehingga dapat memperpendek siklus pengembangan perangkat lunak. Berdasarkan penjelasan tersebut diharapkan *Model Driven Architecture* dapat menjadi solusi dalam pembuatan *user interface* dengan React Native sehingga dapat mempermudah *developer* dalam pengembangan aplikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan maka dapat dirumuskan beberapa permasalahannya yaitu:

1. Bagaimana membuat *user interface* untuk React Native tanpa penulisan *code* secara manual?
2. Bagaimana mengimplementasikan *Model Driven Architecture* untuk React Native?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah diatas dapat ditarik suatu tujuan penelitian sebagai berikut:

1. Mempermudah *developer* dalam pembuatan *user interface* untuk React Native tanpa penulisan *code* secara manual.
2. Mengimplementasikan *Model Driven Architecture* untuk React Native.

1.4 Batasan Masalah

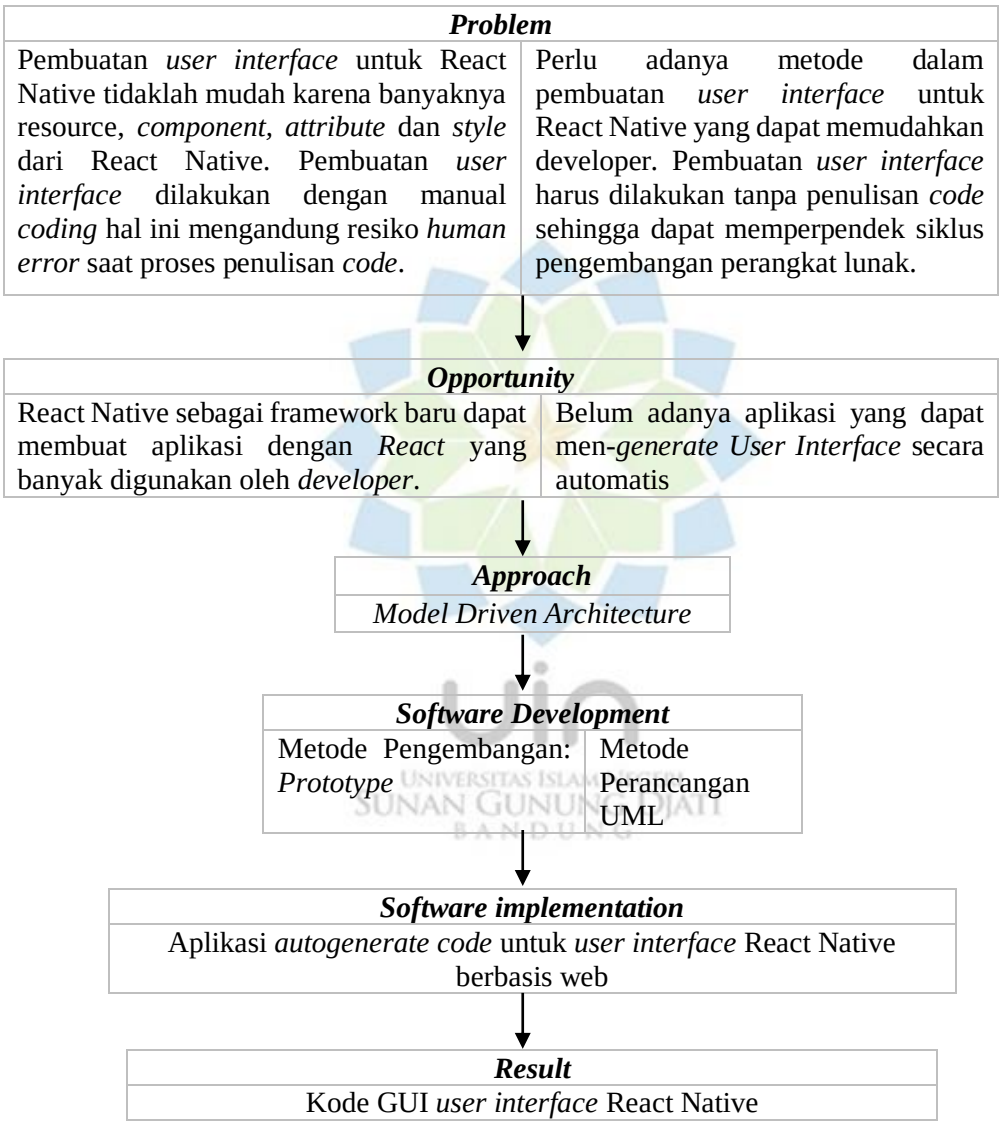
Berdasarkan permasalahan yang ada, maka perlu adanya batasan agar permasalahan tidak melebar, antara lain :

1. Aplikasi berbasis web yang dibangun menggunakan *framework ReactJs*
2. Output hanya menghasilkan *assets* dan *source code* React Native.
3. Metode yang digunakan dalam pengembangan *user interface* ini menggunakan *Model Driven Architecture.*
4. Transformasi otomatis hanya dari level PSM React Native menjadi *cource code.*
5. Aplikasi hanya dapat digunakan untuk tampilan desktop.
6. Aplikasi hanya menyediakan satu ukuran layout untuk *device* iOS-X.

7. Model GUI yang dipakai pada penelitian ini hanya untuk GUI sebagai berikut: View. ScrollView, Text, Text Input, Button, Image, Icon

1.5 Kerangka Pemikiran

Adapun kerangka pemikiran yang digambarkan seperti dibawah ini:



1.6 Metodologi Penelitian

1.6.1 Teknik Pengumpulan Data

Metode yang akan digunakan dalam penelitian ini melalui beberapa teknik pengumpulan data sebagai berikut:

1. Observasi

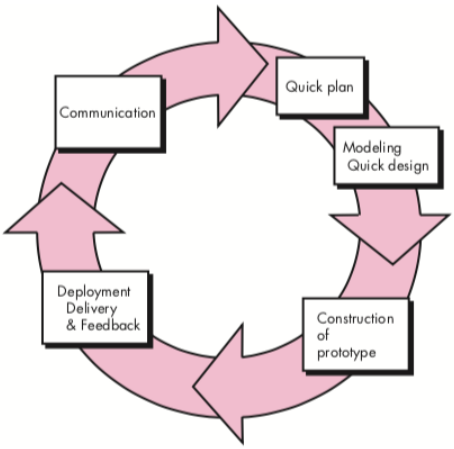
   Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil.

2. Studi Literatur

   Pengumpulan data dengan cara mengumpulkan literatur, jurnal, *paper* dan bacaan-bacaan yang ada kaitannya dengan judul skripsi.

1.6.2 Metodologi Pengembangan

Metode pengembangan perangkat lunak yang digunakan untuk penelitian ini, diantaranya adalah model *prototype*. *Prototype* adalah sebuah metode perancangan software yang banyak digunakan pengembang agar dapat saling berinteraksi dengan pelangan selama proses pembuatan sistem dan terdiri dari 5 tahap yang saling terkait atau mempengaruhi yaitu sebagai berikut[7]:



Gambar 1. 1 Model Prototype

Berdasarkan model *prototype* yang telah digambarkan diatas, maka dapat diuraikan pembahasan masing-masing tahap dalam model tersebut adalah sebagai berikut:

1. *Communication* / Komunikasi

   Tim perancang perangkat lunak melakukan pertemuan dengan para stakeholder untuk menganalisis dan menentukan kebutuhan perangkat lunak yang saat itu diketahui dan untuk menggambarkan area- area dimana definisi lebih jauh untuk iterasi selanjutnya.

2. *Quick Plan* / Perencanaan Secara Cepat

   Dalam perencanaan ini iterasi pembuatan prototipe dilakukan secara cepat. Setelah itu dilakukan pemodelan dalam bentuk "rancangan cepat".

3. *Modeling Quick Design* / Model Rancangan Cepat

   Pada tahap ini dilakukan pemodelan perencanaan ditahap sebelumnya dengan menggunakan pemodelan terstrukturdalam bentuk DFD (Data Flow Diagram), ERD (Entity Relationship Diagram) dan Flowchart untuk menggambarkan analisis dan desain sistem

4. *Construction of Prototype* / Pembuatan *Prototype*

   Dalam pembuatan rancangan cepat berdasarkan pada representasi aspek-aspek perangkat lunak yang akan terlihat oleh para end *user* (misalnya rancangan antarmuka pengguna atau format tampilan). Rancangan cepat merupakan dasar untuk memulai konstruksi pembuatan prototipe.

5. *Deployment Deliery & Feedback*/Penyerahan Dan Memberikan Umpan Balik Terhadap Pengembangan

   Prototype kemudian diserahkan kepada para *stakeholder* untuk mengevaluasi *prototype* yang telah dibuat sebelumnya dan memberikan umpan-balik yang akan digunakan untuk memperbaiki spesifikasi kebutuhan. Iterasi terjadi saat pengembang melakukan perbaikan terhadap prototipe tersebut.

1.7 Sistematika Penulisan

Sistematika penulisan penelitian ini terdiri dari beberapa yang bertujuan untuk mendapatkan arahan dan sistemasi dalam penulisan sehingga mudah dipahami, adapun sistematika secara umum dari penulisan laporan ini adalah:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metode penelitian, serta sistematika penulisan

BAB II STUDI PUSTAKA

Pada bab II akan dijelaskan tentang teori-teori yang digunakan dalam analisa permasalahan yang ada, dan juga teori-teori yang digunakan dalam perancangan dan implementasi.

BAB III ANALISIS DAN PERANCANGAN

Bab III membahas tentang analisis dan perancangan aplikasi yang dibentuk, yaitu berisi cara kerja aplikasi, identifikasi masalah dan evaluasi aplikasi, serta perancangan pembangunan aplikasi.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bab IV dijelaskan tentang spesifikasi aplikasi, kebutuhan aplikasi, implementasi aplikasi dan pengujian yang dilakukan terhadap aplikasi yang dibangun.

BAB V PENUTUP

Bab V berisi kesimpulan dan saran untuk pengembangan aplikasi lebih lanjut dalam upaya memperbaiki kelemahan pada aplikasi guna untuk mendapatkan hasil kinerja aplikasi yang lebih baik dan pengembangan program selanjutnya.