

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metode penelitian yang dilakukan untuk menyusun dan mengerjakan penelitian ini adalah sebagai berikut.

1. Studi Literatur

Studi Literatur yang dilakukan adalah pengumpulan informasi berupa teori tentang ReactJS, metode gamifikasi, *Framework* Gamifikasi Marczewski, *Hedonic-Motivation System Adoption Model* (HMSAM), perhitungan menggunakan *Skala Likert*. Mempelajari ReactJS lebih mendalam untuk mendapatkan pemahaman lebih lanjut untuk memvariasikan soal-soal pada aplikasi seperti *Stateful* dan *Stateless*, *functional* dan *class function*. Adapun pencarian informasi tentang *TicTacToe* yang dapat dijadikan soal dan dapat dimainkan jika soal tersebut selesai sekaligus *reward* untuk hasil akhir dari aplikasi jika semua soal sudah diselesaikan.

Tabel 3.1 merupakan beberapa hal yang membandingkan antara aplikasi pembelajaran dengan gamifikasi dan gim pembelajaran.

Tabel 3.1 Perbandingan aplikasi pembelajaran dengan gamifikasi dan gim pembelajaran (Al-Azawi, 2016)

Aplikasi pembelajaran dengan gamifikasi	Gim pembelajaran
<ol style="list-style-type: none"> 1. Gamifikasi adalah metode menambahkan konsep <i>game elements</i> dan memberikan penghargaan untuk perilaku tertentu. 2. Pengalaman belajar 3. Mendapatkan umpan balik secara langsung 4. Berkompetisi dengan orang lain 	<ol style="list-style-type: none"> 1. Adalah aplikasi permainan untuk membantu pengguna dalam belajar sesuatu 2. Fokus dan daya ingat 3. Kreativitas 4. Interaksi dan kenikmatan dalam bermain

Dalam penelitian sebelumnya yang berjudul Pengembangan Sistem Informasi Pengelolaan Persediaan Barang dengan ReactJS dan React Native Menggunakan *Prototype* oleh Faisal Fajar Nursaid, Adam Hendra Brata, dan Agi Putra Kharisma (2020) menyimpulkan hasil reaksi pengguna terhadap sistem dengan persentase 80% kategori “Setuju”, reaksi terhadap antarmuka memiliki persentase 68% kategori “Cukup”, reaksi terhadap pengoperasian sistem memiliki persentase 96% kategori ”Sangat Mudah”, dan reaksi terhadap fungsionalitas sistem memiliki persentase 76% kategori “Setuju”.

Dalam penelitian gamifikasi sebelumnya yang berjudul Penerapan Gamifikasi pada Sistem Informasi Penilaian Ujian Mahasiswa untuk Meningkatkan Kinerja Dosen oleh Qurotul Aini, Untung Rahardja, Anoesyirwan Moeins, dan Dewi Mariana Apriani (2018) menyimpulkan dengan adanya pengembangan konsep gamifikasi pada sistem PEN+ (Penilaian Plus) dapat berjalan efektif karena dosen memasukkan nilai dengan tepat waktu, dan memotivasi dosen untuk

berlomba-lomba memasukkan nilai secara *ontime* dengan adanya *trophy* kemenangan sehingga mutu dosen dapat terbentuk dengan baik.

2. Perancangan dan Pembuatan Sistem

Perancangan yang dilakukan pada penelitian ini adalah *level design*, pembuatan *mockup* aplikasi, dan perancangan dan pembangunan aplikasi menggunakan *framework* React Native yang didukung dengan *hardware* dan *software* lainnya.

3. Implementasi

Implementasi yang dilakukan adalah dengan menerapkan metode gamifikasi kepada aplikasi dan juga pengimplementasian antarmuka. Aplikasi ini akan dirancang dan dibangun bagian *frontend* saja dengan menggunakan *framework* React Native yang berbasis *Android* juga menggunakan *local storage* sebagai penyimpanan data. Aplikasi tidak memerlukan internet karena aplikasi difokuskan kepada pembelajaran ReactJS agar dapat dimainkan di manapun dan kapanpun.

4. Uji Coba

Uji coba aplikasi akan dilakukan untuk memastikan aplikasi dapat berfungsi dengan sesuai dan benar dengan menguji aplikasi selama dan setelah dibuatnya aplikasi. Lalu aplikasi akan disebarakan untuk diuji bersamaan dengan pemberian kuesioner.

5. Peluncuran

Aplikasi yang sudah selesai dibangun akan langsung dapat digunakan oleh para pengguna di dalam dan di luar Universitas Multimedia Nusantara baik yang memiliki pengetahuan di bidang pemrograman maupun tidak. Setelah aplikasi selesai digunakan, para pengguna diminta untuk mengisi kuesioner untuk diukur nilai persetujuan dari aplikasi yang dibangun menggunakan Skala *Likert*.

3.2 Perancangan Aplikasi

Sistem yang dibuat adalah aplikasi berbasis *Android*. Perancangan sistem dilakukan dengan menerapkan *framework* metode gamifikasi Marczewski, perancangan *flowchart*, perancangan *mockup*, dan menyusun aset yang diperlukan.

3.2.1 Perancangan Gamifikasi

Perancangan gamifikasi yang dibangun berdasarkan 8 *questions to ask yourself* yang diambil dari *framework* metode gamifikasi Marczewski, sebagai berikut:

1. *What is being gamified.*

Aplikasi yang akan digamifikasikan adalah aplikasi pembelajaran ReactJS dengan menambahkan konsep gim.

2. *Why is it being gamified.*

Dengan menambahkan unsur gamifikasi, aplikasi pembelajaran ReactJS ini dapat membantu pengguna dalam belajar ReactJS dan meningkatkan motivasi pengguna secara intrinsik (Malone, 1987).

3. *Who are the users.*

Pengguna aplikasi ini adalah pengguna yang memiliki keinginan untuk belajar tentang ReactJS, baik bagi pemula, yang sudah berpengalaman dalam pemrograman, maupun yang tidak berpengalaman dalam pemrograman.

4. *How is it being gamified.*

Pengguna akan mendapatkan sebuah *Achivement* berupa sebuah permainan yang sudah dibuat oleh pengguna itu sendiri dengan menyelesaikan soal-soal yang sudah diberikan. Permainan ini adalah *TicTacToe* yang dapat dimainkan jika semua soal sudah terjawab dengan benar. Pengguna juga akan mendapatkan sebuah *Reward* berupa sertifikat sebagai tanda bahwa pengguna telah menyelesaikan semua soal dengan benar.

5. *Analytics are set up.*

Menganalisis tingkatan seberapa mudah atau sulit setiap soal yang diberikan kepada pengguna untuk dijawab.

6. *Tested with user.*

Aplikasi akan diuji dan dicoba oleh pengguna Universitas Multimedia Nusantara program studi Informatika angkatan 2016 yang belum memiliki pengalaman belajar ReactJS. Aplikasi yang akan diuji dan dicoba mengenai tampilan yang *user friendly* atau tidak, apakah terdapat masalah saat penggunaan aplikasi, dan apakah soal yang diberikan terlalu sulit.

7. *Acted/Iterated on feedback.*

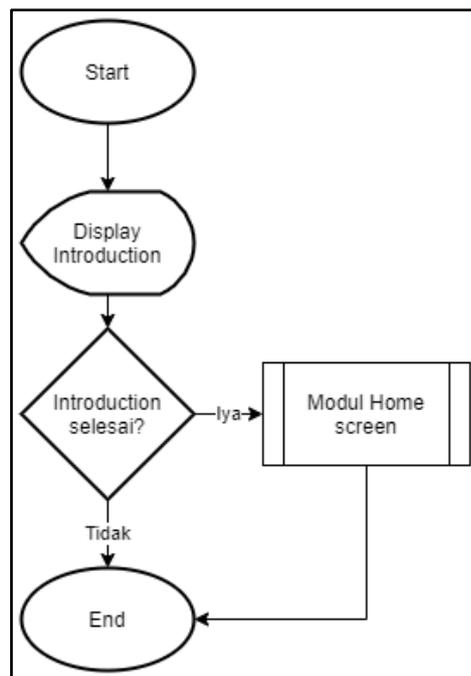
Feedback yang didapatkan dari pengguna akan diimplementasikan jika masih terdapat waktu pengembangan untuk melakukan perbaruan pada aplikasi.

8. *Released the solution*

Setelah aplikasi sudah selesai dikembangkan, maka aplikasi siap diluncurkan untuk digunakan oleh seluruh pengguna.

3.2.2 Flowchart

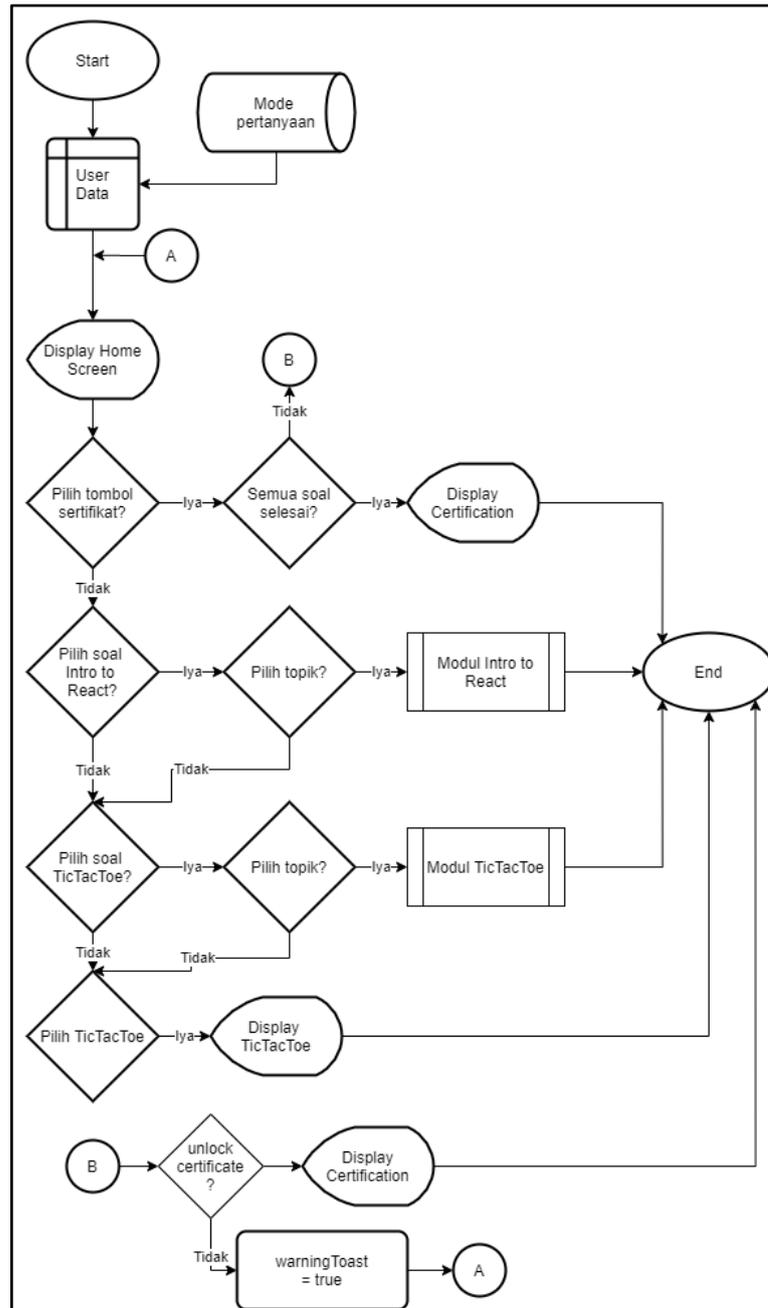
Perancangan *flowchart* dilakukan untuk menjelaskan alur kerja dari aplikasi pembelajaran ReactJS.



Gambar 3.1 *Flowchart* keseluruhan

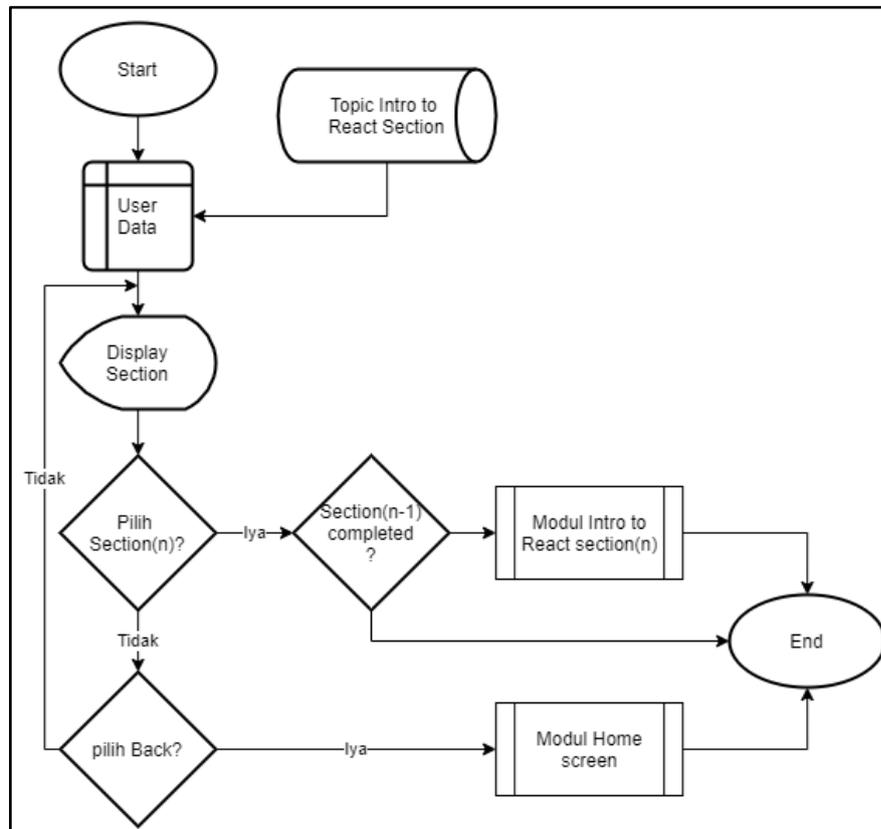
Gambar 3.1 merupakan *Flowchart* aplikasi pembelajaran ReactJS secara keseluruhan. Saat aplikasi digunakan pertama kali, pengguna akan mendapatkan sebuah *introduction* mengenai aplikasi secara singkat. *Introduction* akan

dimunculkan hanya satu kali dan tidak akan ditampilkan kembali jika *introduction* sudah selesai. Setelah itu pengguna akan diteruskan ke layar utama yaitu *Home Screen* yang dapat dilihat *flowchart*-nya seperti pada gambar 3.2.



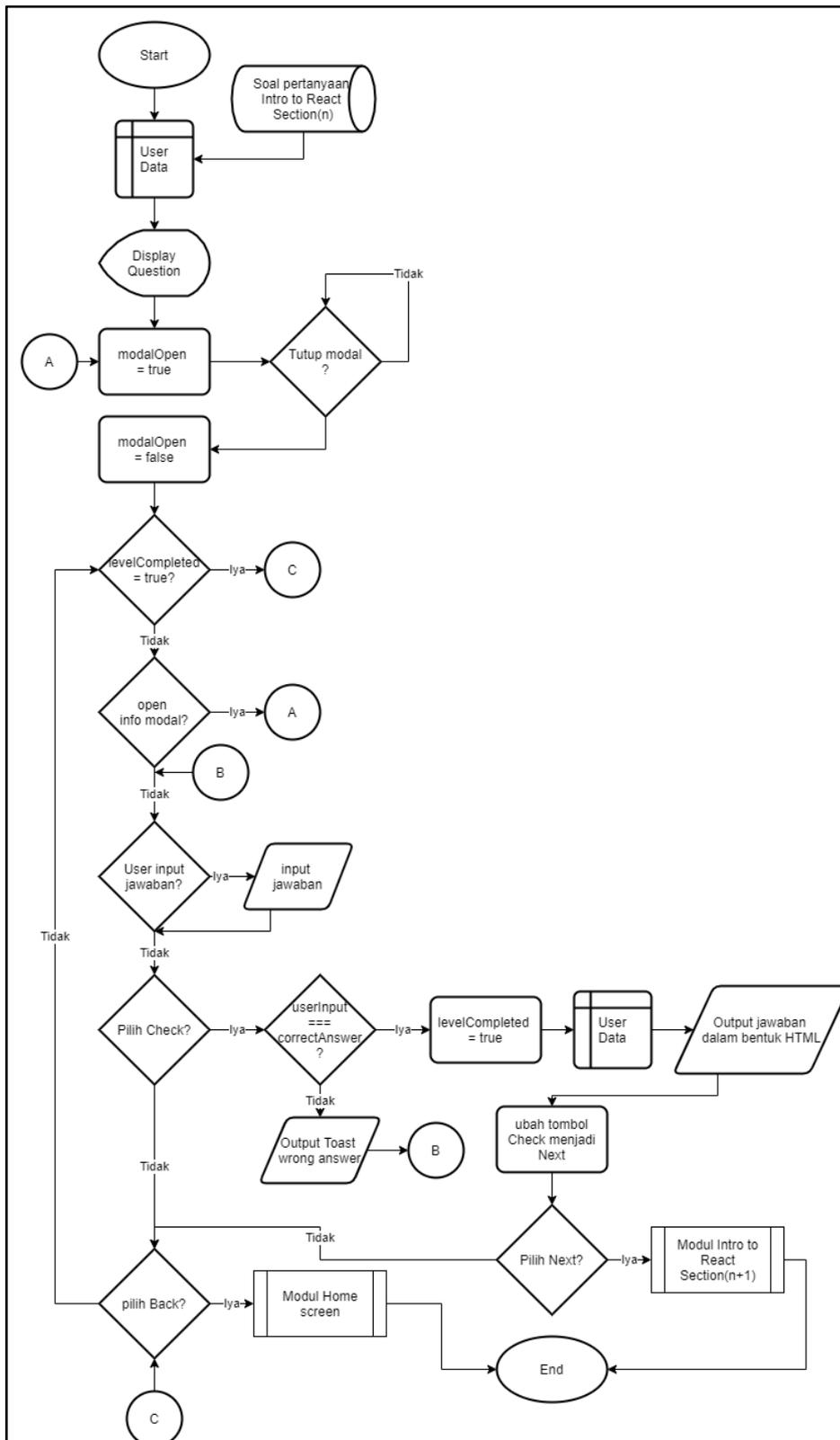
Gambar 3.2 Home Screen Flowchart

Gambar 3.2 merupakan *Flowchart* layar utama. Sebelum layar muncul, aplikasi akan menerima data-data berupa mode soal dan *section* setiap mode. *Home screen* ini memiliki sebuah tombol untuk melihat sertifikat sebagai *rewards* dan tombol untuk melihat progres dari permainan *TicTacToe*. Jika semua soal sudah berhasil dijawab dengan benar, maka sertifikat dapat dilihat hasilnya dengan memunculkan sebuah modal sertifikat. Tetapi untuk tombol progres permainan *TicTacToe* dapat dilihat kapan saja walaupun semua soal belum dijawab. Pada *home screen*, pengguna dapat memilih 2 macam mode, yang pertama adalah mode *Intro to React* (contoh *flowchart* pada gambar 3.3) yang memiliki beberapa topik untuk dibahas di dalamnya. Jenis mode yang kedua adalah soal *TicTacToe* (contoh *flowchart* pada gambar 3.4) yang juga memiliki beberapa topik, di mana pengguna akan menjawab soal (*fill-in-the-blanks*) untuk membuat sebuah permainan *TicTacToe*.



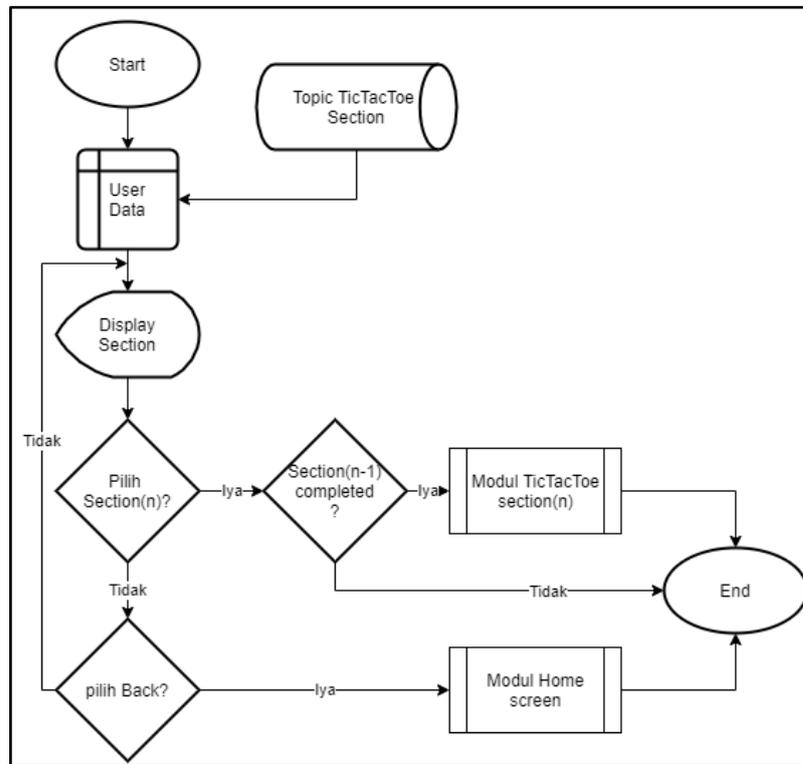
Gambar 3.3 *Intro to React Flowchart*

Gambar 3.3 adalah *flowchart* ketika pengguna memilih mode soal *Intro to React* dan juga topik tertentu. Pada *flowchart* ini, aplikasi akan menampilkan sebuah *list* yang berisi *Section*, di mana setiap *section* ini memiliki pembelajaran soal yang berbeda. Ketika pengguna memilih salah satu *section*, maka pengguna akan diarahkan ke layar berikutnya untuk menyelesaikan beberapa soal seperti pada contoh *flowchart* gambar 3.4. Jika pengguna menekan tombol *Back*, maka pengguna akan diarahkan kembali ke *home screen* seperti pada gambar 3.2.



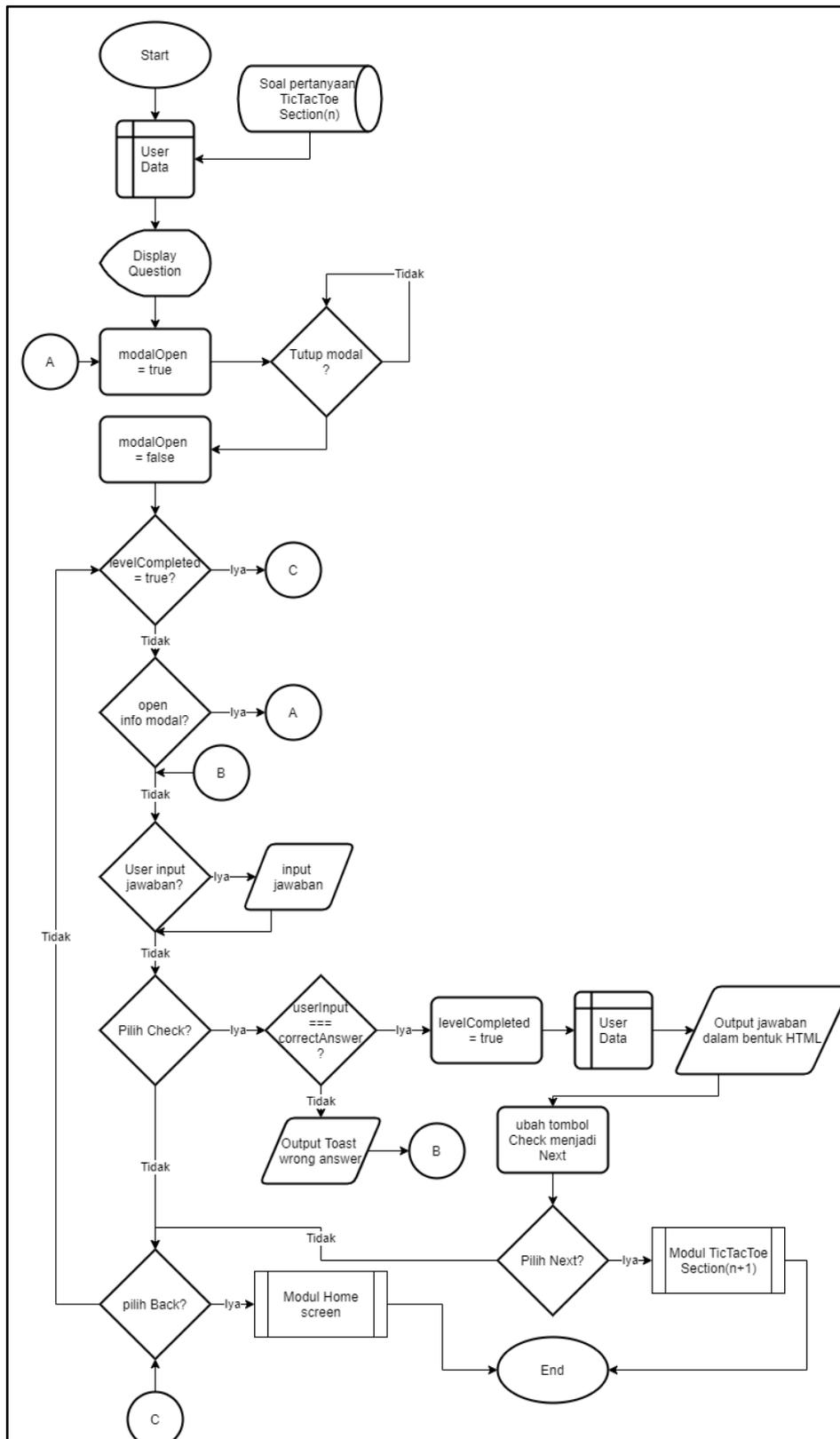
Gambar 3.4 Intro to React Section Flowchart

Gambar 3.4 merupakan *flowchart* ketika pengguna telah memilih salah satu *section*. Halaman ini memiliki beberapa komponen yang pengguna dapat lakukan, yaitu menjawab soal, memeriksa jawaban, dan hasil dari jawaban yang sudah dijawab. Sebelum layar ditampilkan, aplikasi akan mengambil data soal untuk *section* dari topik yang sudah dipilih tadi. Lalu setelah layar ditampilkan, akan muncul sebuah modal sebagai informasi tentang soal yang pengguna sedang pelajari. Ketika pengguna sudah selesai mengisi *fill-in-the-blanks*, pengguna dapat menekan tombol *Check*. Saat tombol ditekan, aplikasi akan memeriksa jawaban dari pengguna, jika jawaban sudah benar maka *variable* `levelCompleted` akan berubah menjadi *true* lalu disimpan ke dalam *local storage*. Selain itu, akan ditampilkan juga sebuah hasil dari soal tersebut di sebuah *HTML renderer* yang terdapat di atas soal, dan tombol *Check* akan berubah menjadi tombol *Next* untuk menuju ke *section* selanjutnya. Sebaliknya, jika soal tidak dijawab dengan benar, maka akan muncul sebuah *toast* untuk memberitahu pengguna bahwa jawaban belum benar. Bila pengguna menekan tombol *Back*, maka pengguna akan diarahkan ke layar *Intro to React* seperti pada *flowchart* gambar 3.3.



Gambar 3.5 *TicTacToe Flowchart*

Gambar 3.5 adalah *flowchart* ketika pengguna memilih mode soal *TicTacToe*. Pada *flowchart* ini, aplikasi akan menampilkan sebuah *list* yang berisi *Section*, di mana setiap *section* ini memiliki soal untuk menyelesaikan pembuatan gim *TicTacToe*. Ketika pengguna memilih salah satu *section*, maka pengguna akan diarahkan ke layar berikutnya untuk menyelesaikan beberapa soal seperti pada contoh *flowchart* gambar 3.6. Jika pengguna menekan tombol *Back*, maka pengguna akan diarahkan kembali ke *home screen* seperti pada gambar 3.2.



Gambar 3.6 TicTacToe Section Flowchart

Gambar 3.6 merupakan *flowchart* ketika pengguna telah memilih salah satu *section*. Halaman ini memiliki beberapa komponen yang pengguna dapat lakukan, yaitu menjawab soal, memeriksa jawaban, dan hasil dari jawaban yang sudah dijawab. Soal (*fill-in-the-blanks*) pada mode *TicTacToe* ini akan lebih banyak dari pada mode *Intro to React*. Sebelum layar ditampilkan, aplikasi akan mengambil data soal. Lalu setelah layar ditampilkan, akan muncul sebuah modal sebagai informasi tentang soal yang pengguna sedang pelajari. Ketika pengguna sudah selesai mengisi *fill-in-the-blanks*, pengguna dapat menekan tombol *Check*. Saat tombol ditekan, aplikasi akan memeriksa jawaban dari pengguna, jika jawaban sudah benar maka *variable levelCompleted* akan berubah menjadi *true* lalu disimpan ke dalam *local storage*. Selain itu, akan ditampilkan juga sebuah hasil dari soal tersebut di sebuah *HTML renderer* yang terdapat di atas soal, dan tombol *Check* akan berubah menjadi tombol *Next* untuk menuju ke *section* selanjutnya. Sebaliknya, jika soal tidak dijawab dengan benar, maka akan muncul sebuah *toast* untuk memberitahu pengguna bahwa jawaban belum benar. Bila pengguna menekan tombol *Back*, maka pengguna akan diarahkan ke layar *Intro to React* seperti pada *flowchart* gambar 3.5.

3.2.3 Perancangan Mockup

Mockup dirancang untuk menggambarkan *User Interface* (UI) aplikasi pembelajaran ReactJS.

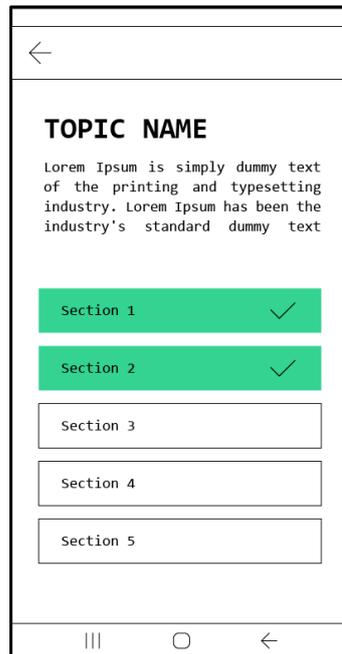
A. Home Screen



Gambar 3.7 *Home Screen Mockup*

Gambar 3.7 adalah tampilan awal di mana terdapat beberapa komponen, salah satunya adalah komponen sertifikat. Sertifikat ini tidak akan memiliki warna jika topik dari setiap mode (*Intro to React & TicTacToe*) sudah selesai. Komponen *accordion* yang ada pada mode *Intro to React* dan *TicTacToe* dapat dipilih langsung dan akan memunculkan topik-topiknya tanpa harus pindah ke layar lain. Lalu terdapat sebuah komponen tombol yaitu *TicTacToe Progress*, tombol ini berfungsi sebagai pelacak sudah berapa persen pengguna berhasil menyelesaikan soal *TicTacToe* dan juga sebagai tombol untuk bermain *TicTacToe*.

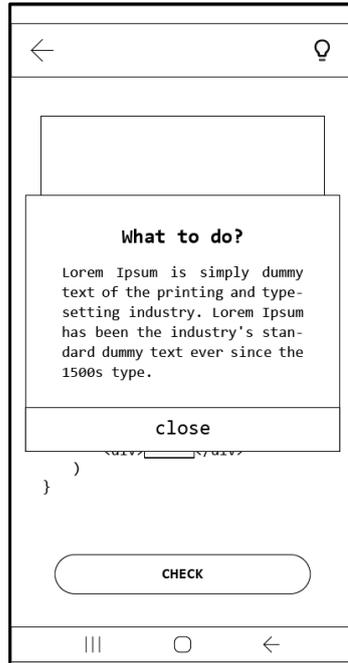
B. Section Selection



Gambar 3.8 *Mockup Section Selection*

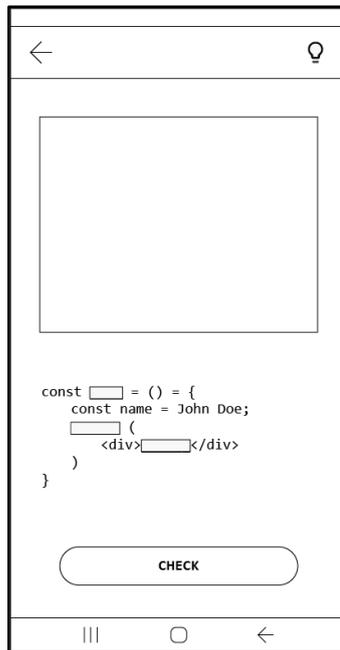
Gambar 3.8 adalah layar *section selection* jika salah satu mode (*Intro to React* dan *TicTacToe*) dipilih seperti pada gambar 3.7. Komponen yang terdapat pada layar ini adalah komponen *title*, *description*, dan *section*. Komponen *title* akan berubah sesuai dengan nama topik apa yang dipilih oleh pengguna. Komponen *description* merupakan sebuah kalimat singkat tentang apa yang akan dibahas pada topik yang pengguna pilih. Lalu komponen *section* merupakan *list* untuk menampilkan ada berapa soal yang dapat dipelajari dan dikerjakan oleh pengguna. Komponen *section* ini tidak akan berwarna hijau jika *section* tersebut belum selesai dikerjakan.

C. Information Modal



Gambar 3.9 *Information Modal Mockup*

Gambar 3.9 merupakan tampilan awal pada saat salah satu *section* (gambar 3.8) dipilih. Komponen modal informasi akan muncul pertama kali untuk memberitahu pengguna apa yang harus dilakukan di *section* tersebut. Komponen modal dapat dibuka berulang-ulang jika pengguna lupa dengan apa yang harus dilakukan.



Gambar 3.10 *Answering Section Mockup*

Gambar 3.10 merupakan layar di mana pengguna dapat memulai proses pembelajarannya dan proses pengisian soal (*fill-in-the-blanks*). Layar ini memiliki beberapa komponen, yaitu komponen informasi, HTML *renderer*, potongan *code*, dan tombol *check*. Komponen informasi (berlogo lampu di atas kanan) akan memunculkan sebuah modal informasi tentang apa yang harus dilakukan pada *section* yang telah dipilih pengguna seperti pada gambar 3.9. Komponen informasi dapat dibuka berulang-ulang sesuai kebutuhan pengguna. Komponen HTML *renderer* adalah komponen untuk menampilkan hasil dari soal ReactJS yang sudah dikerjakan oleh pengguna jika pengguna menjawab dengan sesuai. Komponen HTML *renderer* ini tidak akan menampilkan apapun bila jawaban yang diisi oleh pengguna belum sesuai. Komponen potongan *code* merupakan soal di mana pengguna harus mengisi jawabannya pada *fill-in* yang sudah diberikan oleh

aplikasi. Ketika pengguna sudah selesai mengisi komponen tersebut, maka pengguna dapat menekan komponen tombol *Check* untuk melihat apakah jawaban pengguna sudah sesuai dengan yang diminta oleh aplikasi. Jika jawaban pengguna sudah benar, maka hasil akan langsung ditampilkan di komponen *HTML renderer*, lalu aplikasi akan memunculkan sebuah *Toast* (“*correct*” bila benar, “*wrong*” bila salah), dan tombol *Check* akan berubah menjadi tombol *Next* agar pengguna dapat langsung berpindah ke *section* berikutnya.

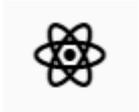
3.2.4 Perancangan Aset

Aplikasi pembelajaran ReactJS menggunakan sejumlah aset yang diambil dari *internet* dan beberapa aset berasal dari pribadi. Aset dari *internet* yang digunakan merupakan gratis dan sudah menjadi pendukung *frontend* untuk pengembangan aplikasi *mobile*.

Tabel 3.2 Daftar aset aplikasi

Gambar	Keterangan	Sumber
	<i>Certificate icon</i> menunjukkan progres belajar belum selesai	Aset pribadi
	<i>Certificate icon</i> menunjukkan progres belajar telah selesai	Aset pribadi

Tabel 3.2 Daftar aset aplikasi (lanjutan)

Gambar	Keterangan	Sumber
	<p><i>Light Bulb icon</i> untuk simbol modal informasi</p>	<p>Ionicons/md-bulb</p>
	<p>X dan O <i>icon</i> untuk simbol pada mode dan permainan <i>TicTacToe</i></p>	<p>FontAwesome/circle-o Feather/x</p>
	<p><i>Lock</i> untuk informasi bahwa section tidak dapat dipilih</p>	<p>Feather/lock</p>
	<p><i>Check</i> untuk <i>section</i> yang sudah selesai dilakukan</p>	<p>Feather/check</p>
	<p><i>React</i> merupakan simbol poin yang didapat oleh pengguna</p>	<p>MaterialCommunityIcons/react</p>
	<p><i>Check Square</i> untuk mode yang sudah selesai dilakukan</p>	<p>Feather/check-square</p>