

BAB III

METODOLOGI DAN PERANCANGAN APLIKASI

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan untuk perancangan dan pembangunan aplikasi adalah sebagai berikut.

a. Observasi

Observasi dilakukan untuk mengetahui masalah yang sedang terjadi dalam dunia pemrograman. Pada tahap ini ditemukan bahwa terdapat kesulitan dalam memahami ekosistem dan konsep dari React dan pemrograman deklaratif. Sehingga pembelajaran React Native menjadi sulit bagi *software developer* yang ingin mempelajari *framework* tersebut (Ogrodowczyk, 2017).

b. Studi Literatur

Studi literatur dilakukan untuk mencari dan mempelajari bahan referensi yang akan digunakan perancangan dan pembuatan aplikasi pembelajaran React Native dengan menggunakan metode gamifikasi Octalysis.

c. Perancangan Aplikasi

Perancangan aplikasi dilakukan untuk merancang aplikasi pembelajaran sehingga menarik untuk digunakan dan dapat berfungsi dengan baik. Perancangan aplikasi ini juga mencakup perancangan API. Aplikasi pembelajaran React Native memiliki beberapa fitur utama yaitu *edit profile*, *journey*, *achievements*, *friends*, *shop*, *leaderboard*, dan beberapa fitur lainnya. Aplikasi pembelajaran React Native yang dirancang berbasis *web* dan hanya berjalan pada layar *desktop*.

d. Pembuatan Aplikasi

Pembuatan aplikasi dilakukan setelah selesai merancang aplikasi. Aplikasi dibuat dengan menggunakan *framework* React Native. API dibuat dengan menggunakan *framework* Node JS dengan Firebase.

e. Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk memastikan aplikasi yang dibuat berjalan dengan baik. Pengujian aplikasi dilakukan dalam dua tahap yaitu pada saat pengerjaan dan setelah aplikasi selesai. Setelah selesai mengerjakan sebuah fitur, maka akan dilakukan pengujian untuk memastikan fitur yang dibuat berjalan dengan baik. Setelah semua fitur pada aplikasi telah selesai dibuat, maka akan dilakukan pengujian secara menyeluruh. Pengujian dilakukan dengan menyebarkan kuesioner dan tingkat kesuksesan aplikasi ini diukur menggunakan *Hedonic-Motivation System Adoption Model* (HMSAM)

f. Pembuatan Laporan

Pembuatan laporan dilakukan setelah aplikasi telah selesai dibuat dan diuji. Pembuatan laporan bertujuan untuk mendokumentasikan aplikasi yang telah dibuat dan dapat menjadi referensi bagi penelitian selanjutnya.

3.2 Perancangan Aplikasi

Aplikasi Reactive yang dirancang berbasis *web* dan hanya berjalan pada layar *desktop*. Tahapan dari perancangan aplikasi terdiri dari perancangan gamifikasi dengan metode Octalysis, model aplikasi, perancangan *flowchart*, perancangan *mockup* aplikasi, dan pemilihan aset.

3.2.1 Perancangan Gamifikasi

Aplikasi pembelajaran React Native yang dibangun didasarkan delapan inti yang terdapat pada metode gamifikasi Octalysis.

1. *Meaning*

Poin pada inti *meaning* yang diambil adalah *beginners' luck* dan *elitism*. *Beginners' luck* diterapkan melalui sebuah sistem peluang hadiah diawal permainan. *Elitism* diterapkan melalui sistem *avatar* pada aplikasi. Apabila pengguna mendapatkan *avatar* yang langka atau dapat membeli *avatar* yang mahal maka akan memberikan rasa elitisme pada pengguna.

2. *Accomplishment*

Poin pada inti *accomplishment* yang diambil adalah *leaderboard* dan *progress bar*. *Leaderboard* diterapkan melalui fitur *leaderboard* yang dimana pengguna dapat melihat skor pengguna lain dan teman dari pengguna. *Progress bar* diterapkan melalui elemen *progress bar level* pengguna.

3. *Empowerment*

Poin pada inti *empowerment* yang diambil adalah *milestone unlocks* dan *blank fills*. *Milestone unlocks* diterapkan melalui sistem level permainan. Setiap *stage* memiliki beberapa *level*. Untuk membuka *level* selanjutnya, pengguna harus menyelesaikan *level* sebelumnya. *Blank fills* diterapkan melalui sistem penulisan jawaban pada aplikasi. Untuk menjawab sebuah pertanyaan pengguna harus menuliskan potongan kode yang benar dari sebuah kode.

4. *Ownership*

Poin pada inti *ownership* yang diambil adalah *exchangable points*, *virtual goods*, dan *avatar*. *Exchangable points* diterapkan melalui sistem *shop*. Pengguna akan mendapatkan *in-game currency* yaitu *coffee beans*. *Coffee beans* yang didapat kemudian dapat ditukarkan menjadi *virtual goods* yaitu *avatars* dan *phone skins*.

5. *Social Influence*

Poin pada inti *social influence* yang diambil adalah *friending*. *Friending* diterapkan melalui sistem pertemanan yang ada pada aplikasi. Pengguna dapat melihat *level* dan *avatar* dari teman pengguna sehingga dapat memberikan rasa kompetitif.

6. *Scarcity*

Poin pada inti *scarcity* yang diambil adalah *prize pacing*. *Prize pacing* diterapkan melalui sistem hadiah. Semakin cepat pengguna menyelesaikan *level*, maka *coffee beans* yang didapat akan semakin banyak.

7. *Unpredictability*

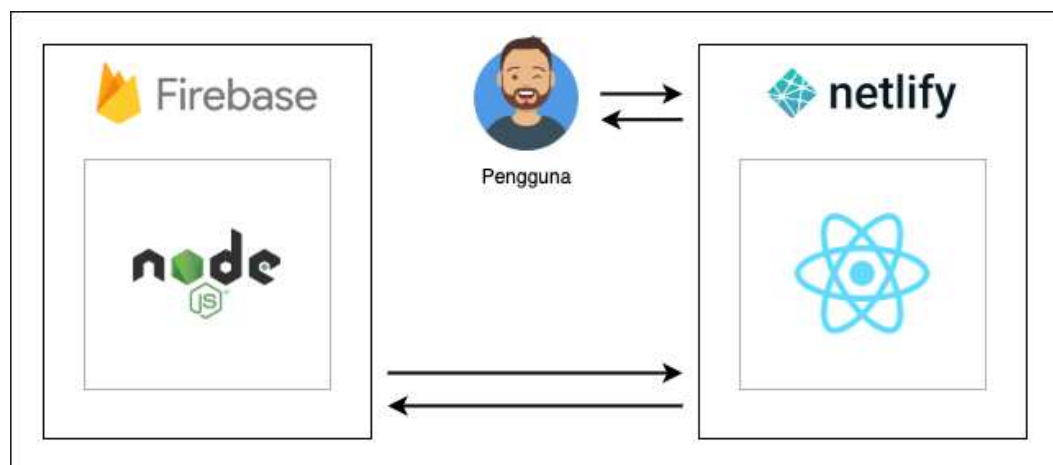
Poin pada inti *unpredictability* yang diambil adalah *easter egg*. *Easter egg* dapat dilihat di berbagai elemen aplikasi. Nama *in-game currency* pada aplikasi ini adalah *coffee beans* karena kebiasaan meminum kopi pada *programmer*. Terdapat sebuah *avatar* yang bernama The Atom yang merupakan sebuah *superhero* dan logo dari React Native adalah atom. *Easter egg* lain dapat dilihat dengan menekan 10 kali kata *master* pada bagian *home*.

8. *Avoidance*

Poin pada inti *avoidance* yang diambil adalah *evanescence opportunity*. *Evanescence opportunity* diterapkan pada sistem hadiah *coffee beans*. Apabila pengguna tidak menyelesaikan level dengan cepat maka pengguna akan melewatkan kesempatan mendapatkan *coffee beans* lebih banyak.

3.2.2 Model Aplikasi

Model dari aplikasi yang akan digunakan dalam pembuatan aplikasi pembelajaran React Native dapat dilihat pada Gambar 3.1.



Gambar 3.1 Model Aplikasi Pembelajaran React Native

Aplikasi pembelajaran React Native bernama Reactive dirancang untuk membantu pengguna dalam membuat komponen React Native serta memahami *properties* dan *style* dari tiap komponen. Komponen yang akan dipelajari adalah Text, View, Image, Activity Indicator, Text Input, dan Touchable Opacity. Reactive dirancang dengan menggunakan metode gamifikasi Octalysis sehingga dapat mempermudah proses pembelajaran.

Setiap komponen yang akan diajarkan memiliki beberapa *level*. Semua komponen terbuka sehingga pengguna bebas mencoba semua komponen yang tersedia. Namun, untuk membuka *level* selanjutnya, pengguna harus

menyelesaikan *level* sebelumnya. Tingkat kompleksitas meningkat seiring bertambahnya *level*. Setelah menyelesaikan *level*, maka pengguna akan mendapatkan hadiah berupa *coffee beans* yang dapat ditukarkan menjadi *avatars* dan *phone skins*. *Coffee beans* yang diperoleh beragam tergantung dengan tingkat kesulitan dari *level* tersebut dan waktu penyelesaian. Selain *coffee beans*, pengguna juga akan mendapatkan *experience* yang dapat membuat *level* pengguna bertambah.

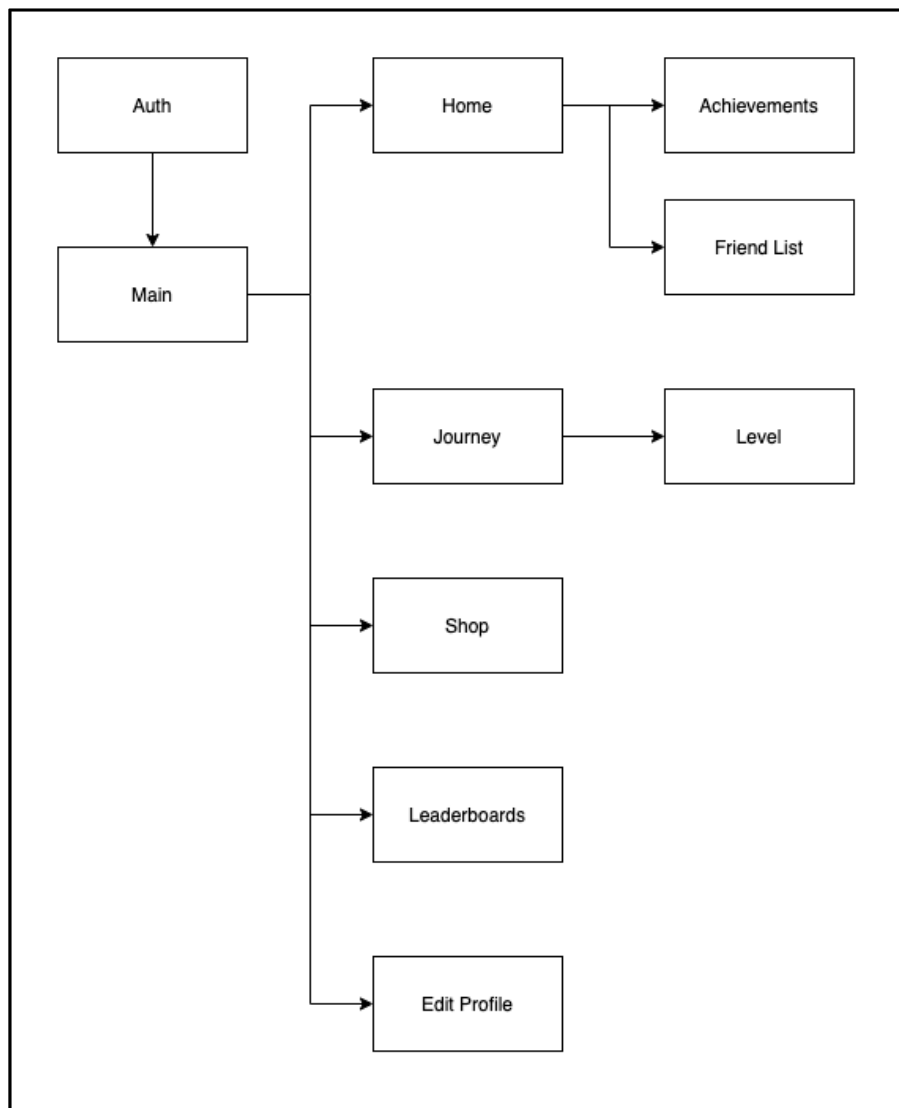
Pengguna dapat melihat posisi dari pengguna yang lain dengan fitur *leaderboard*. Fitur *leaderboard* terbagi menjadi dua yaitu *global* dan *friend*. *Global leaderboard* berisi urutan 10 pengguna teratas yang memiliki *score experience* tertinggi. *Friend leaderboard* sama seperti *global leaderboard* namun pengguna yang akan dimunculkan adalah teman dari pengguna.

Pengguna dapat melihat *achievement* atau pencapaian yang telah dicapai pada menu *achievement* pada menu *home*. Selain melihat *achievement*, pengguna dapat menambahkan pengguna lain sebagai teman. Pengguna dapat melihat *avatar*, nama, dan *level* dari teman pengguna sehingga dapat memberikan rasa kompetitif untuk bermain.

Halaman *edit profile* dapat diakses dengan menekan *avatar* pengguna yang terletak pada bagian atas dari *side bar menu*. Halaman *edit profile* berfungsi untuk mengubah *avatars* dan *phone skins* pengguna dengan yang telah diperoleh.

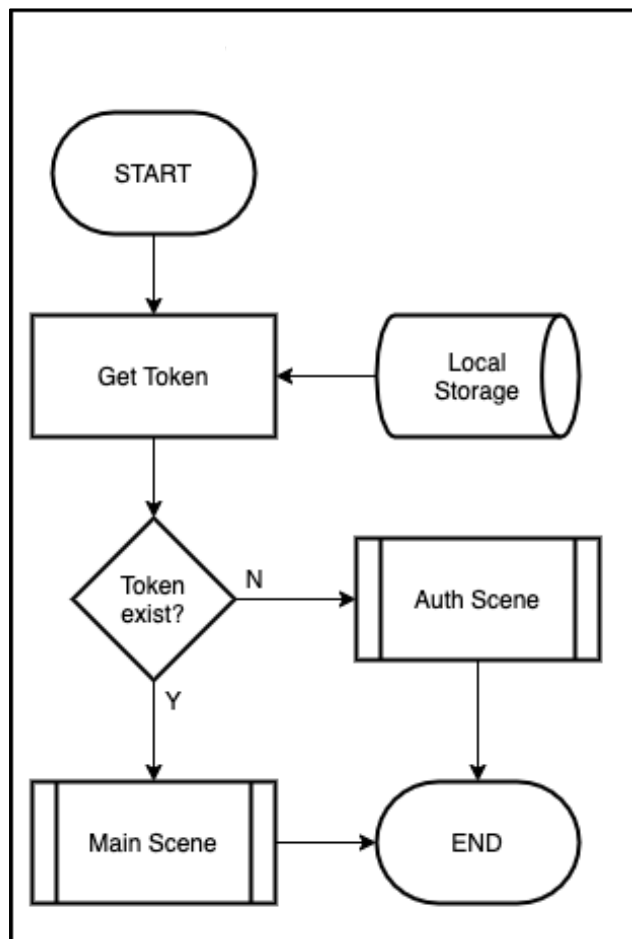
3.2.3 Flowchart

Perancangan *flowchart* bertujuan untuk merancang alur dari aplikasi pembelajaran React Native. Gambar 3.2 merupakan *sitemap* dari aplikasi Reactive.



Gambar 3.2 *Sitemap* dari Aplikasi Reactive

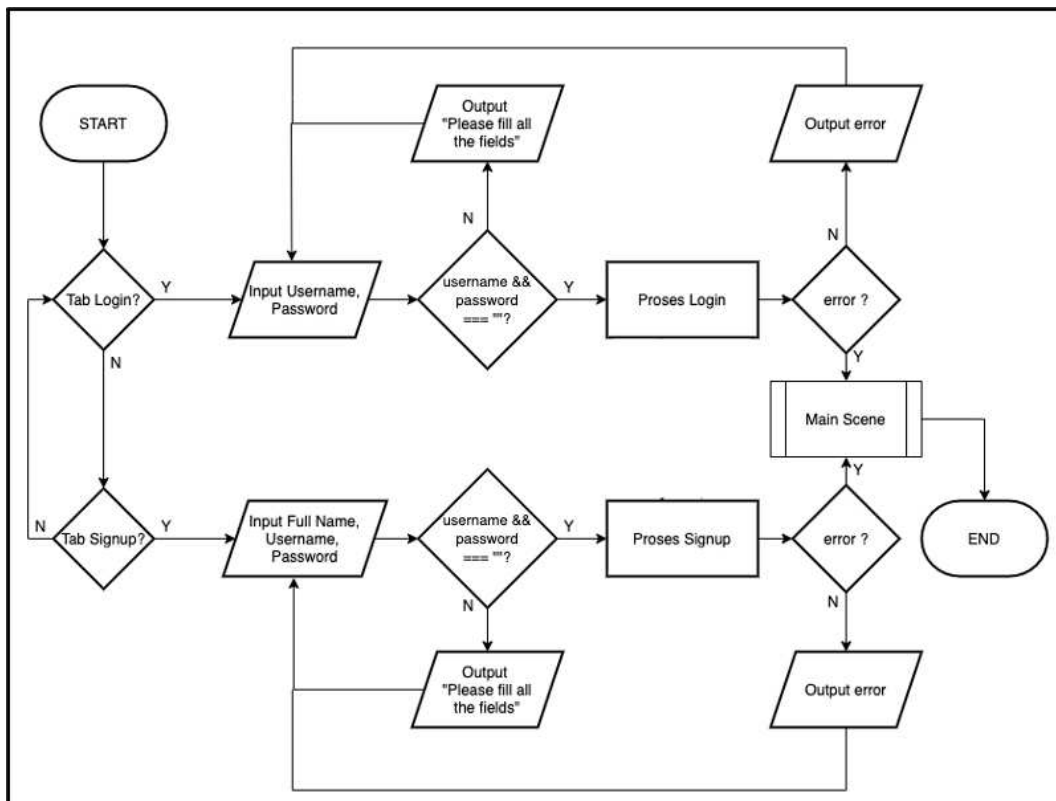
Pada saat pengguna pertama kali menggunakan aplikasi Reactive, pengguna akan dibawa pada halaman Auth yang dimana pengguna dapat melakukan *login* atau *signup*. Setelah pengguna berhasil *login* atau *signup*, maka pengguna akan dibawa ke halaman Main. Pada halaman Main, pengguna dapat melakukan navigasi ke modul Home, Journey, Shop, Leaderboard, dan Edit Profile melalui *menu drawer* disebelah kiri.



Gambar 3.3 *Flowchart* Main Navigator

Pada saat aplikasi dibuka akan dilakukan pengecekan melalui modul Main Navigator. Main Navigator akan mengambil *token* melalui *local storage*. Apabila

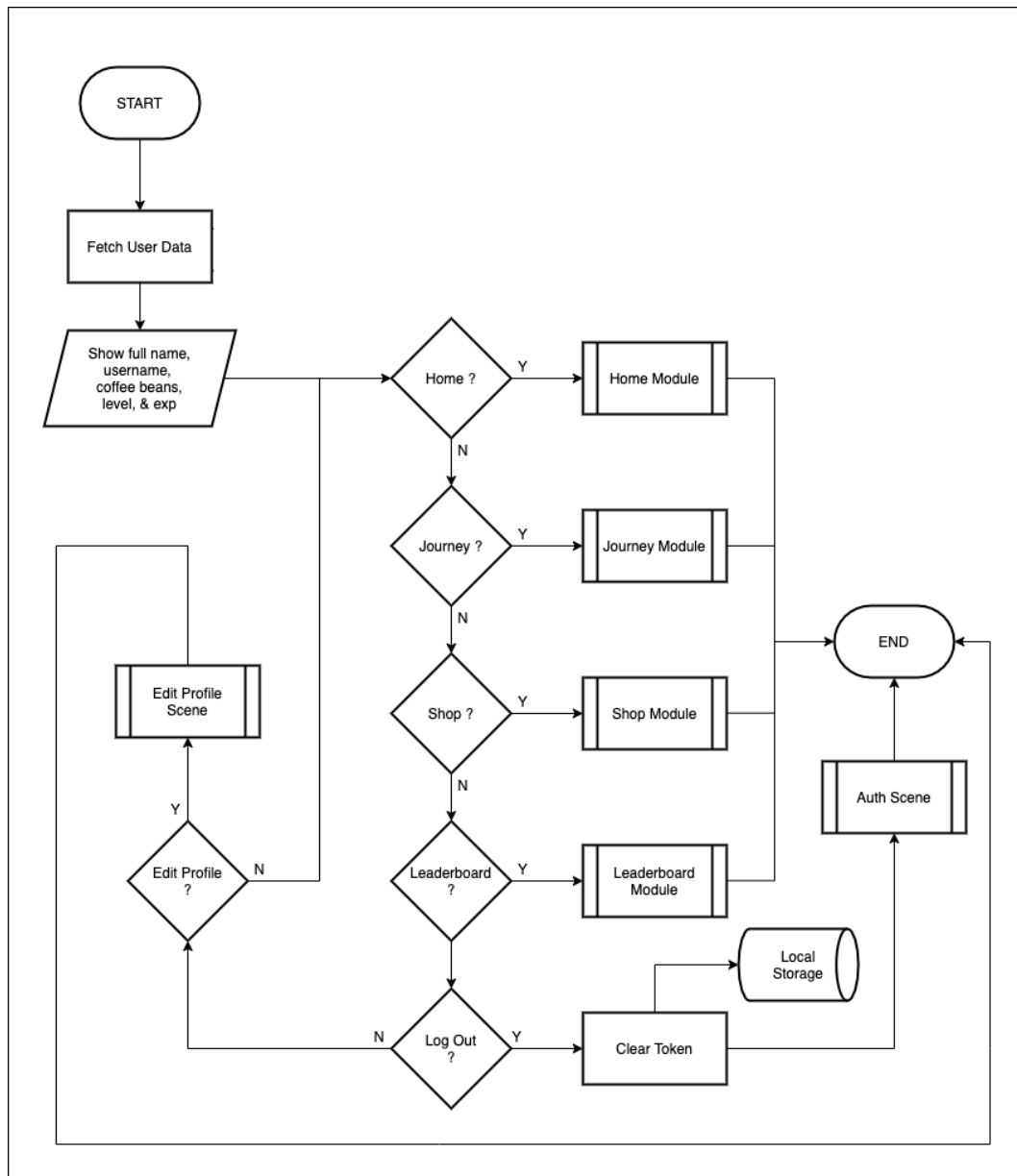
tidak terdapat *token*, maka aplikasi akan menampilkan halaman Auth Scene untuk melakukan proses *login* atau *signup*. Apabila terdapat *token*, maka aplikasi akan menampilkan halaman Main Scene. Proses yang terdapat pada halaman Auth Scene dapat dilihat pada Gambar 3.4 dan proses yang terdapat pada halaman Main Scene dapat dilihat pada Gambar 3.5.



Gambar 3.4 *Flowchart* Halaman Auth Scene

Pada Gambar 3.4, dilakukan pengecekan apakah pengguna sedang berada pada *tab login* atau *signup*. Secara *default*, *tab* yang aktif adalah Login. Pengguna kemudian memasukkan *username* dan *password* yang telah dibuat sebelumnya.

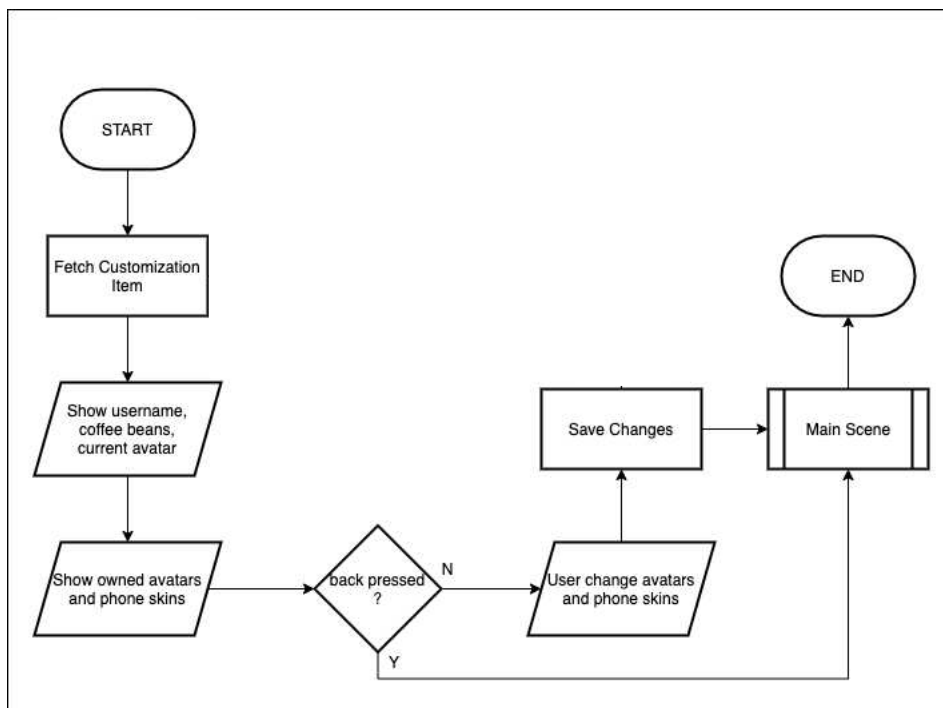
Setelah menekan tombol *login*, maka akan dilakukan pengecekan apakah *username* dan *password* yang dimasukkan kosong. Apabila terdapat *field* kosong maka akan ditampilkan sebuah teks merah dengan tulisan "*Please fill all the fields*". Apabila pengguna mengisi *username* dan *password* maka akan dilakukan proses *login*. Proses *login* akan memanggil API Login dan akan dilakukan pengecekan terhadap *username* dan *password* yang telah dikirim oleh pengguna. Apabila *username* atau *password* tidak sesuai atau tidak terdaftar, maka *server* akan mengirim status *error* dan aplikasi akan menampilkan *error message* dari *response*. Apabila *server* tidak mengirimkan status *error* maka aplikasi akan menyimpan *token* yang dikirim oleh *server* di *local storage*. Proses mendaftar hampir sama dengan proses *login*. Setelah memilih *tab signup*, maka pengguna memasukkan nama lengkap, *username*, dan *password*. Kemudian akan dilakukan pengecekan apakah terdapat *field* yang kosong. Apabila terdapat *field* yang kosong, maka akan ditampilkan teks berwarna merah dengan tulisan "*Please fill all the fields*". Apabila tidak ada *field* yang kosong maka akan dilakukan proses *signup* dengan mengirim nama lengkap, *username*, dan *password* ke API Signup. Apabila sudah terdapat *username* yang sama maka *server* akan mengirimkan status *error* sehingga aplikasi akan menampilkan *error message* dari *response*. Apabila *server* tidak mengirimkan status *error* maka aplikasi akan menyimpan *token* yang dikirim oleh *server* di *local storage*. Setelah proses *login* dan *signup* telah selesai maka akan dilakukan navigasi ke halaman Main Scene. Proses dari Main Scene dapat dilihat pada Gambar 3.5.



Gambar 3.5 Flowchart Halaman Main Scene

Pada Gambar 3.5, aplikasi akan mengirimkan *token* untuk mengambil data pengguna dari API. Data pengguna seperti nama lengkap, username, *coffee beans*, *level*, dan *experience* kemudian akan ditampilkan. Pengguna dapat melakukan navigasi ke halaman Edit Profile dengan menekan tombol Edit Profile yang prosesnya dapat dilihat pada Gambar 3.6. Apabila pengguna memilih Home, maka

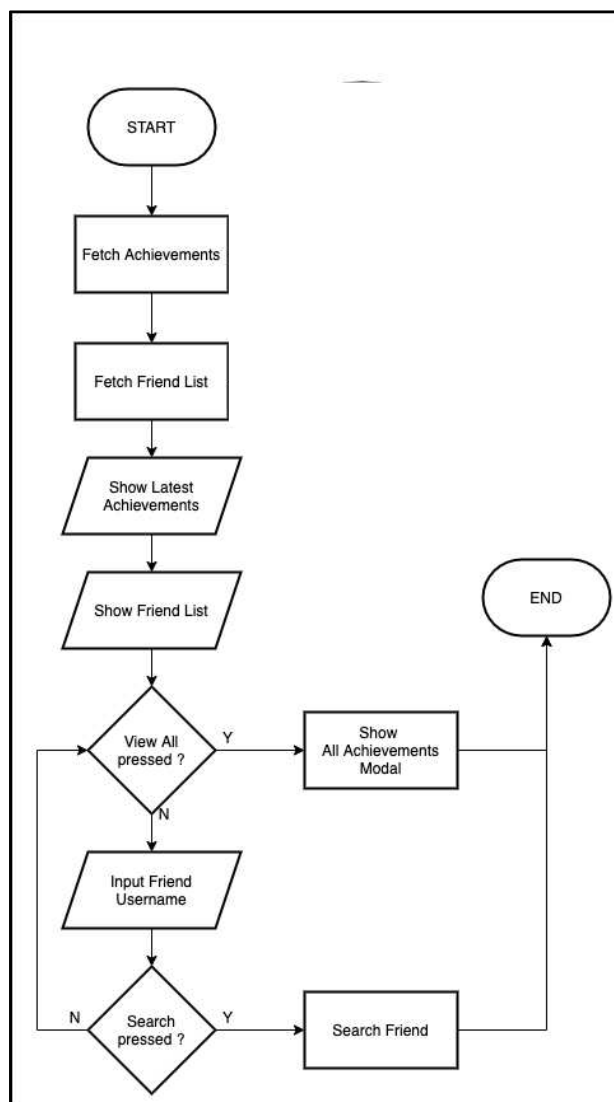
modul Home akan ditampilkan yang prosesnya dapat dilihat pada Gambar 3.7. Apabila pengguna memilih Journey, maka modul Journey akan ditampilkan yang prosesnya dapat dilihat pada Gambar 3.8. Apabila pengguna memilih halaman Shop, maka modul Shop akan ditampilkan yang prosesnya dapat dilihat pada Gambar 3.10. Apabila pengguna memilih Leaderboard, maka modul Leaderboard akan ditampilkan yang prosesnya dapat dilihat pada Gambar 3.11. Apabila pengguna ingin *log out* dari aplikasi dapat menekan tombol Log Out. Ketika tombol Log Out ditekan, maka aplikasi akan menghapus *token* dari *local storage* kemudian melakukan navigasi ke halaman Auth Scene.



Gambar 3.6 *Flowchart* Edit Profile Scene

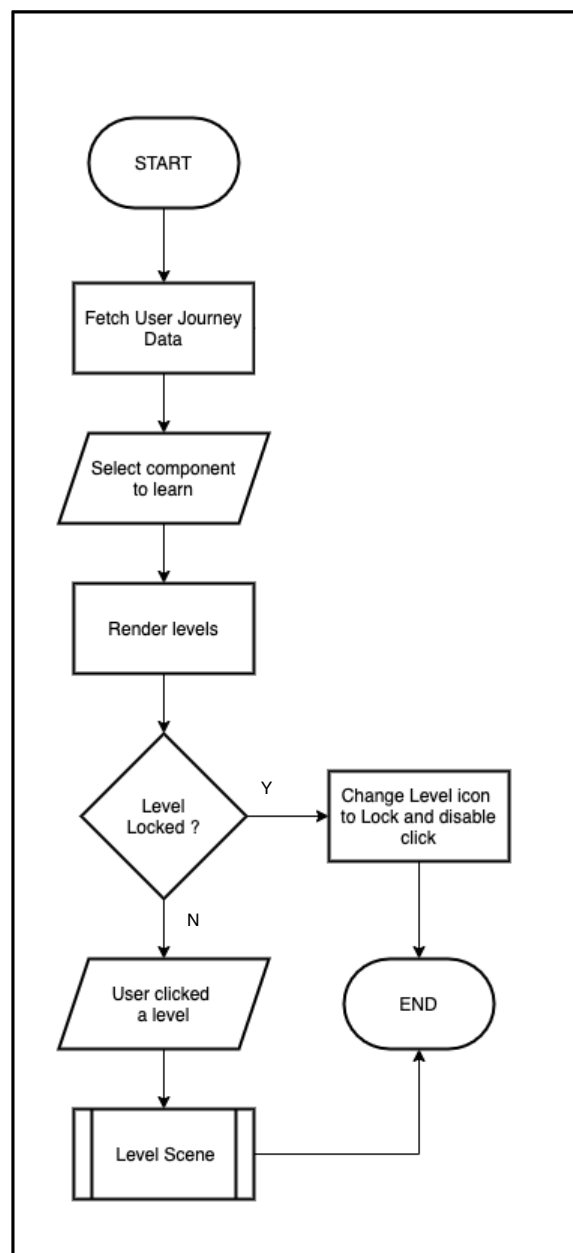
Pada Gambar 3.6, aplikasi akan mengambil *customization item* seperti *avatars* dan *phone skins* dari API. Kemudian aplikasi akan menampilkan *username*, jumlah *coffee beans*, *avatar* sekarang, beserta *avatars* dan *phone skins*

yang telah dimiliki. Pengguna dapat kembali ke halaman Main Scene dengan menekan tombol *back*. Pengguna kemudian dapat memilih *avatars* dan *phone skins* yang diinginkan kemudian apabila pengguna telah selesai memilih maka pengguna dapat menekan tombol Save Changes untuk menyimpan perubahan yang dibuat. Ketika tombol Save Changes ditekan maka aplikasi akan memanggil API untuk menyimpan perubahan yang telah dibuat. Setelah selesai, maka aplikasi akan navigasi kembali ke halaman Main Scene.



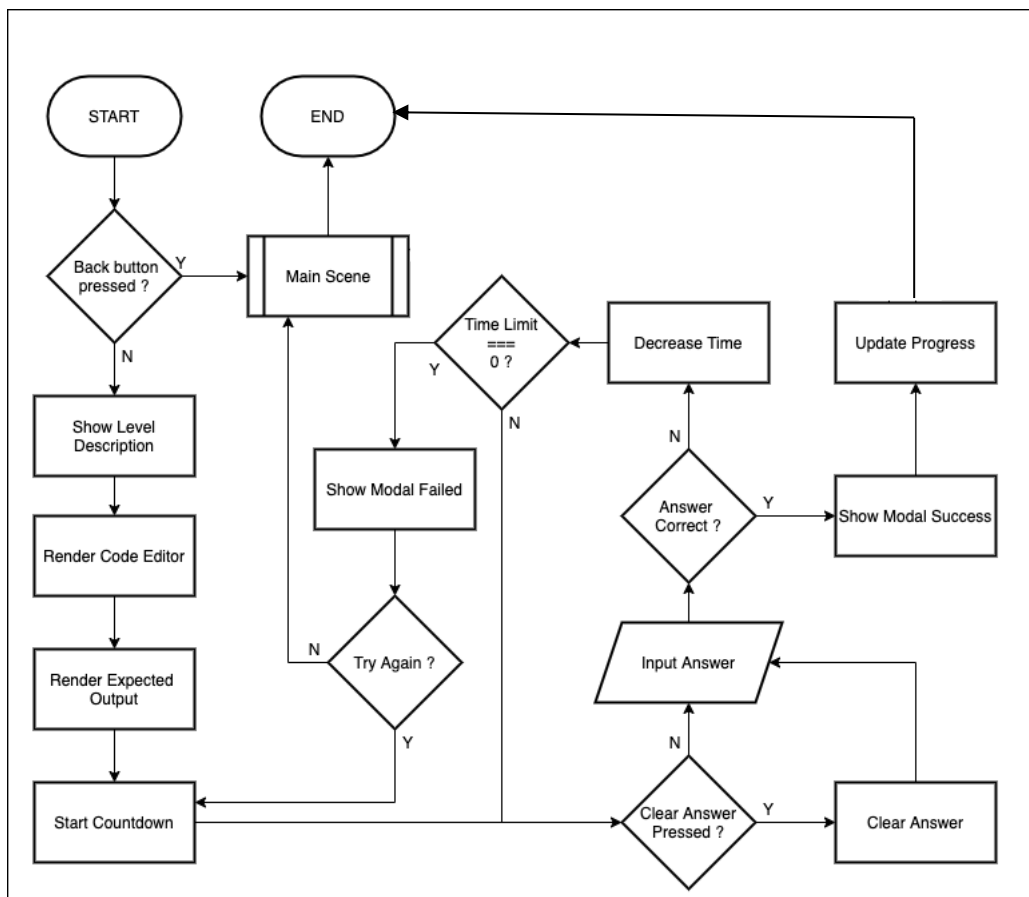
Gambar 3.7 *Flowchart* Modul Home

Pada Gambar 3.7, disaat modul Home dimunculkan, aplikasi akan mengambil data *achievements* dan *friend list* dari API. Apabila tombol *view all* ditekan maka akan keluar *modal* yang berisi daftar *achievements*. Pengguna juga dapat menambah teman dengan mengetik *username* yang ingin dicari dan kemudian menekan tombol *search*.



Gambar 3.8 *Flowchart* Modul Journey

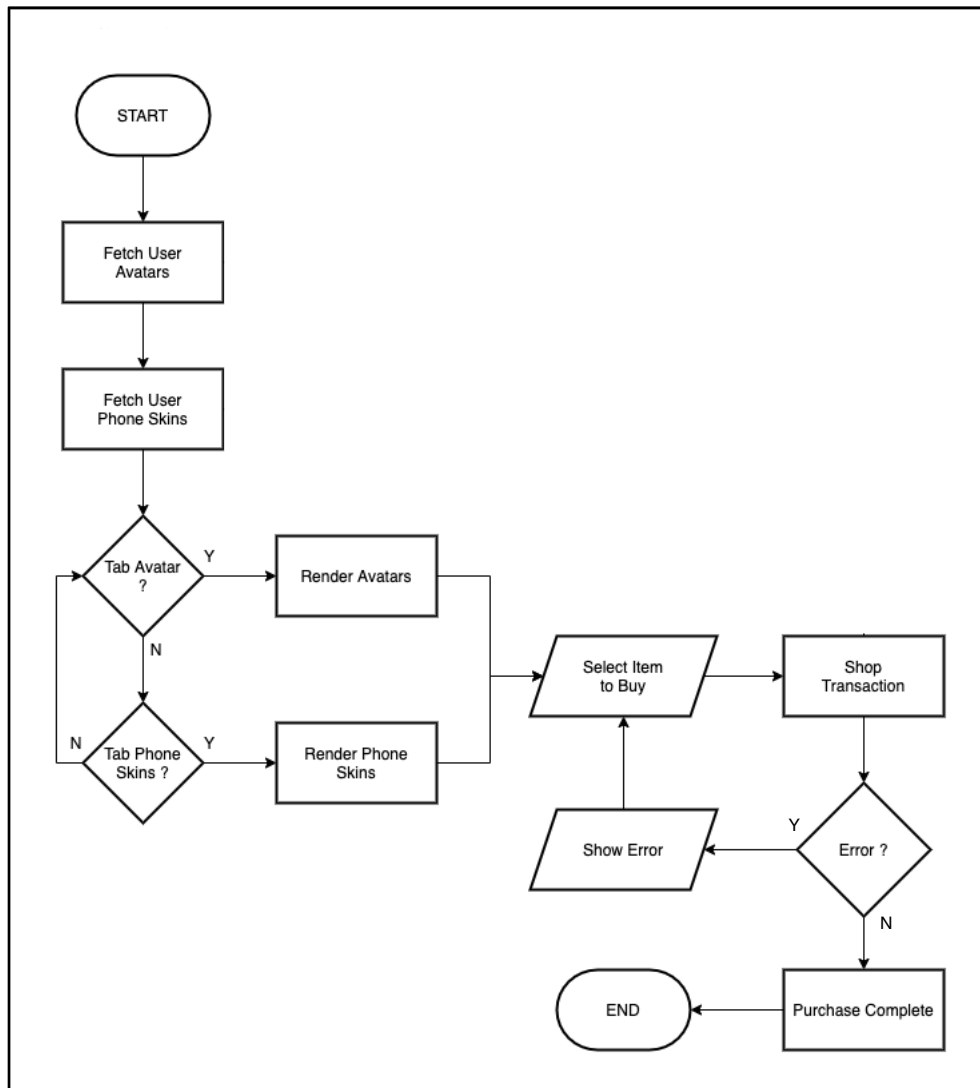
Pada Gambar 3.8, saat pengguna melakukan navigasi ke modul Journey, aplikasi akan mengambil data Journey dari pengguna. Setelah berhasil mengambil data Journey dari pengguna, maka aplikasi akan menampilkan daftar komponen yang dapat dipelajari. Apabila pengguna telah memilih komponen yang ingin dipelajari, maka akan ditampilkan *list level* dari komponen yang dipilih. Apabila *level* belum bisa dimainkan maka *icon* dari *level* tersebut akan berubah menjadi *icon* gembok atau *lock* dan tidak bisa diklik. Apabila pengguna memilih *level* yang dapat dimainkan, maka aplikasi akan melakukan navigasi ke halaman Level Scene.



Gambar 3.9 Flowchart Halaman Level Scene

Pada Gambar 3.9, pada halaman Level Scene, pengguna dapat kembali ke halaman Main Scene dengan menekan tombol *back*. Aplikasi kemudian akan menampilkan deskripsi dari *level*, *code editor*, dan *expected output*. Deskripsi dari level berisi arahan dan soal dari *level* yang dipilih. *Code editor* berisi potongan kode dan *field* kosong untuk jawaban. *Expected output* akan menampilkan hasil yang diharapkan dari potongan kode pada *level* tersebut. Aplikasi kemudian memulai hitung mundur waktu untuk menjawab. Apabila pengguna ingin menghapus *field* jawaban tanpa menggunakan tombol *backspace*, pengguna dapat menekan *icon clear* untuk mengosongkan *field*. Apabila pengguna belum berhasil memasukkan jawaban yang benar pada *field* jawaban, maka waktu akan terus berkurang. Apabila waktu untuk menjawab sudah habis, maka akan muncul *modal failed* dengan pilihan untuk mencoba lagi atau kembali ke halaman Home Scene. Apabila pengguna memilih untuk mencoba lagi, maka aplikasi akan mengembalikan waktu pengerjaan seperti diawal dan pengguna dapat mengerjakan lagi. Apabila pengguna berhasil menjawab dengan benar, maka akan muncul *modal success* yang berisi daftar hadiah yang diterima oleh pengguna. Hadiah yang diterima adalah *coffee beans* dan *experience*. Jumlah *coffee beans* dan *experience* yang diterima relatif dengan lamanya waktu pengerjaan. Beberapa *level* memiliki peluang untuk menghasilkan hadiah tambahan berupa *avatar*. Hadiah *coffee beans* dan *experience* hanya didapat ketika pengguna menyelesaikan *level* tersebut pertama kali. Setelah menjawab dengan benar, maka aplikasi akan memperbarui *progress* dari pengguna dengan memanggil API.

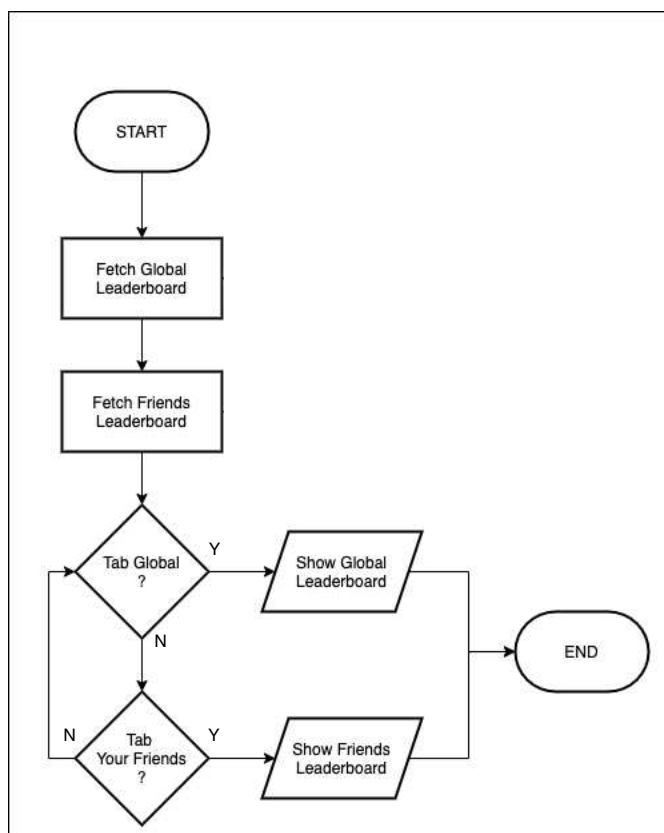
Pengguna dapat melakukan navigasi kembali ke halaman Main Scene setelah menekan tombol *back* pada *modal success* untuk memilih *level* selanjutnya.



Gambar 3.10 *Flowchart* Modul Shop

Gambar 3.10 merupakan proses dari modul Shop. Pada saat pengguna melakukan navigasi ke modul Shop, aplikasi akan mengambil data *avatars* dan *phone skins* pengguna melalui API. Secara *default*, *tab* yang aktif adalah *tab avatar*. Aplikasi akan memunculkan daftar barang atau *item* sesuai dengan *tab* yang aktif. Apabila pengguna sudah memiliki barang tersebut, maka tulisan

"Purchase" akan berubah menjadi "Owned" dan tidak dapat dibeli lagi. Pengguna kemudian akan memilih barang yang diinginkan kemudian aplikasi akan melakukan transaksi dengan API *shop transaction*. Server akan melakukan pengecekan apakah saldo yang dimiliki cukup. Apabila saldo tidak cukup, maka server akan mengirimkan status *error* dan aplikasi akan menampilkan *error message*. Apabila saldo cukup, maka transaksi berhasil.



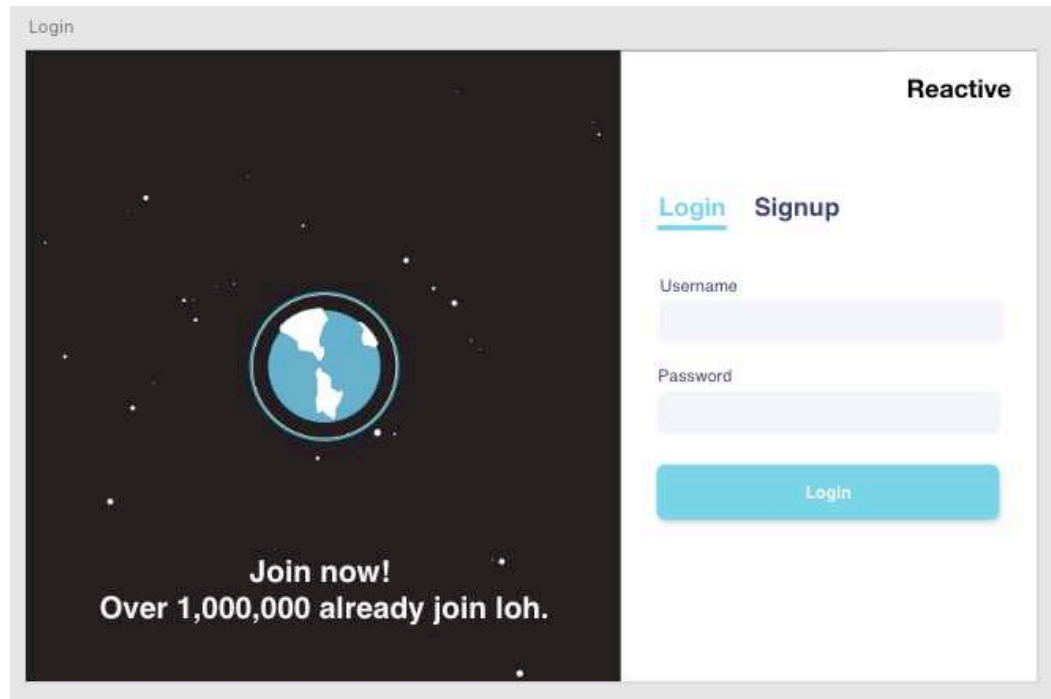
Gambar 3.11 *Flowchart* Modul Leaderboard

Pada Gambar 3.11, saat pertama kali pengguna melakukan navigasi ke modul Leaderboard, aplikasi akan mengambil data *global leaderboard* dan *friends leaderboard* dari API. Apabila *tab* yang dipilih *global* maka aplikasi akan menampilkan 10 pengguna teratas. Apabila *tab* yang dipilih *friends* maka aplikasi akan menampilkan 10 teman pengguna teratas.

3.2.4 Mockup Aplikasi

Pembuatan *mockup* dilakukan untuk merancang tampilan antarmuka aplikasi Reactive yang menarik.

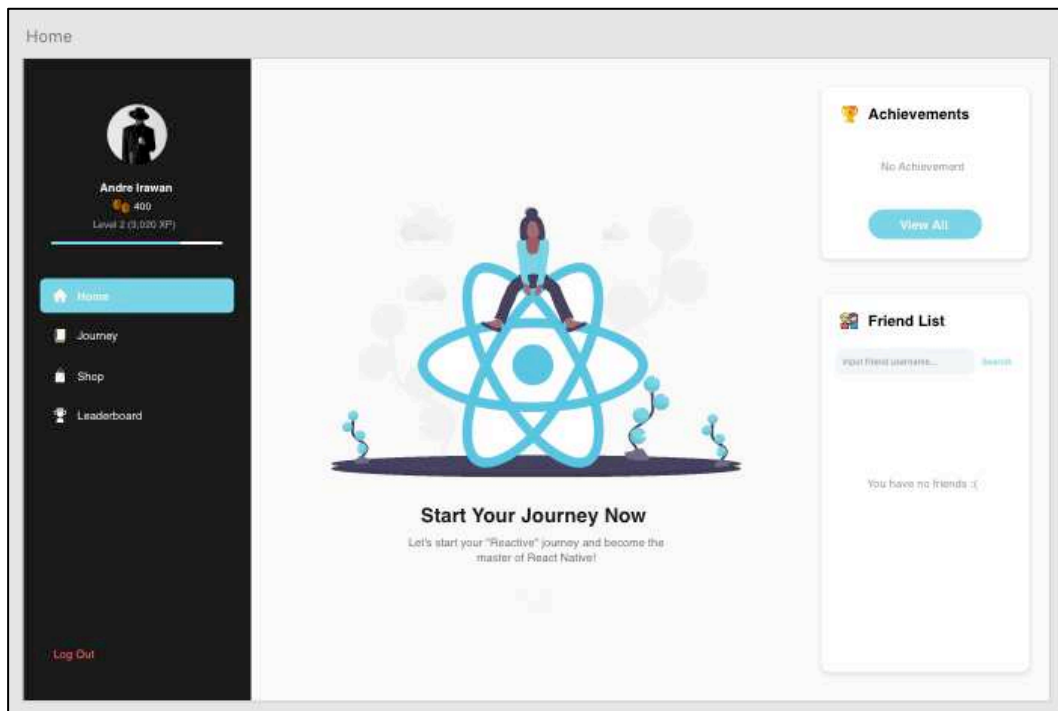
A. Auth Scene



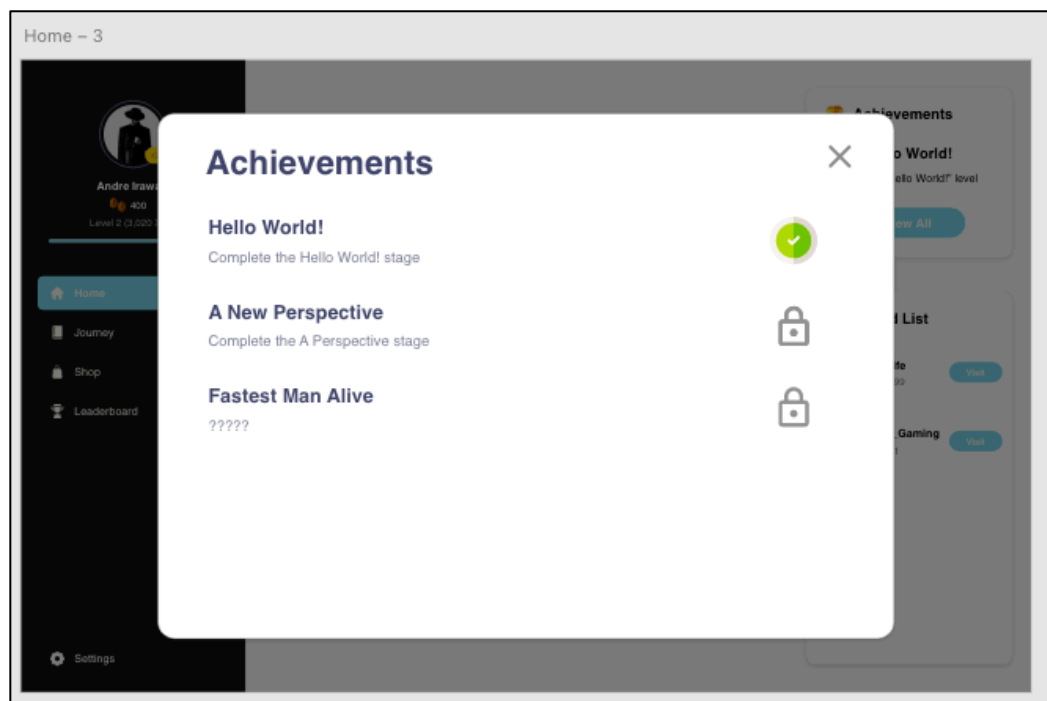
Gambar 3.12 Mockup Halaman Auth Scene

Gambar 3.12 merupakan rancangan dari halaman Auth Scene. Pada halaman Auth Scene terdapat sebuah *tab* untuk *login* dan *signup*. Pada bagian *login*, terdapat kolom teks untuk pengguna memasukkan *username* dan *password*. Setelah memasukkan *username* dan *password*, pengguna kemudian menekan tombol *login*. Tampilan pada bagian *signup* sama seperti *login*, dengan tambahan satu kolom teks untuk nama lengkap dan tulisan *login* menjadi *signup*.

B. Main Scene



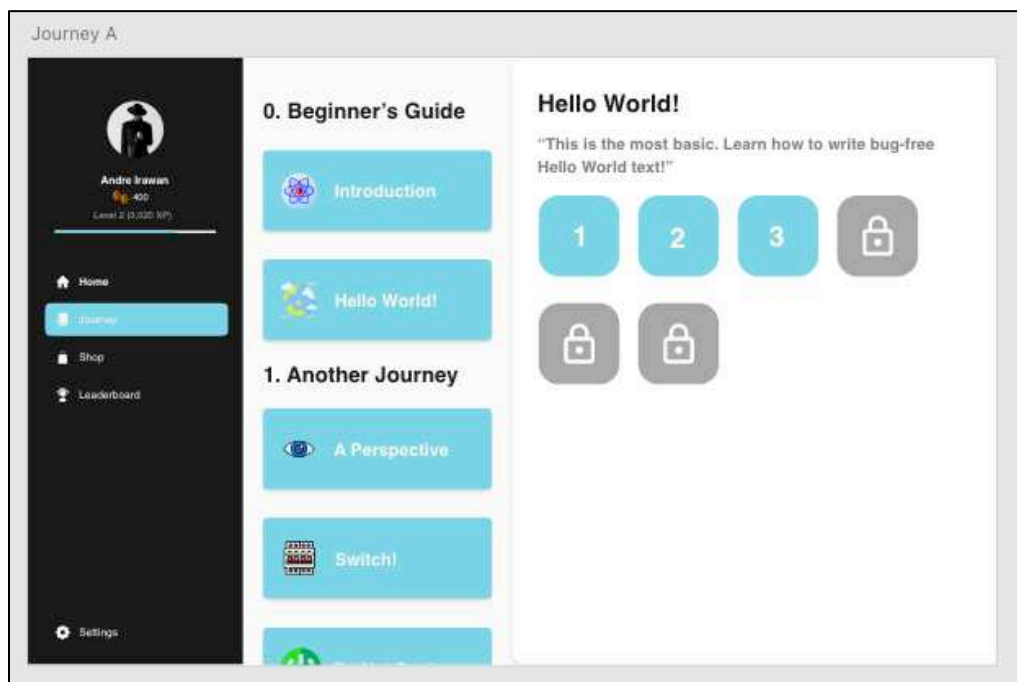
Gambar 3.13 *Mockup* Modul Home



Gambar 3.14 *Mockup* Tampilan *Achievements*

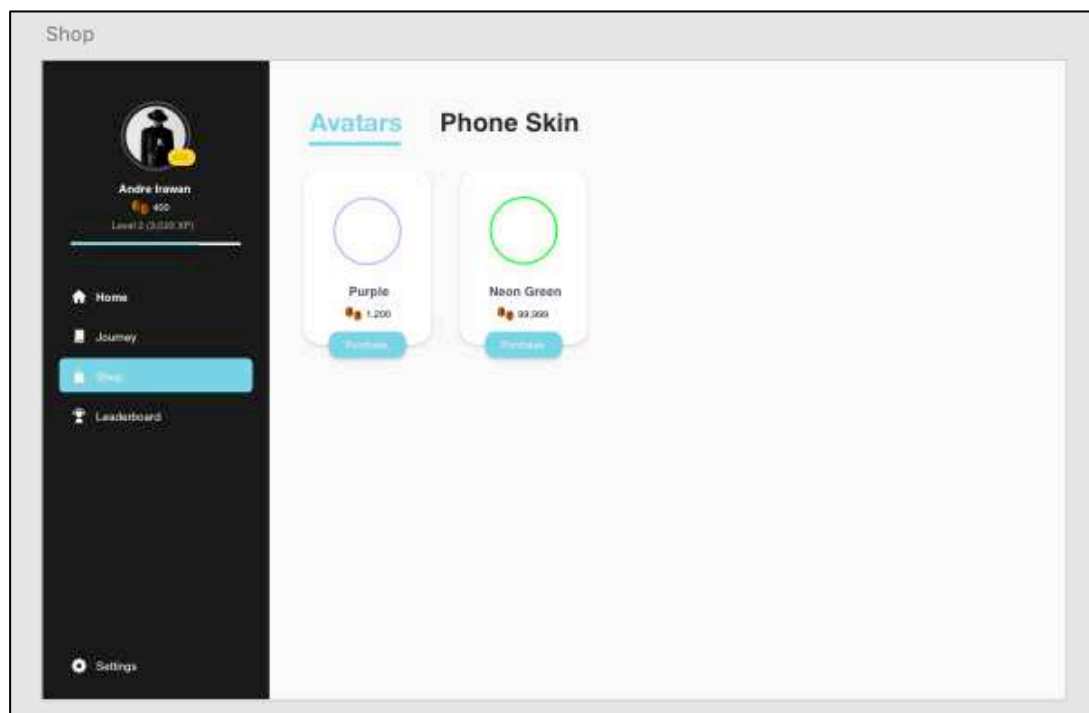
Main Scene terdiri dari sebuah komponen *drawer* yang dapat menampilkan empat modul yaitu Home, Journey, Shop, dan Leaderboard. Secara *default*, modul yang ditampilkan pertama kali adalah modul Home. Selain menampilkan modul, terdapat informasi pengguna yang ditampilkan seperti *avatar*, *coffee beans*, *experience*, dan *level* pengguna. Apabila pengguna ingin mengganti akun, pengguna dapat menekan tombol *log out*.

Seperti Gambar 3.13, pada modul Home, pengguna dapat melihat *achievements* dengan menekan tombol "*View All*". Apabila tombol "*View All*" ditekan maka *modal achievements* akan terlihat seperti pada Gambar 3.14. Apabila pengguna telah mencapai sebuah *achievement*, maka tulisan "*No Achievement*" akan berubah sesuai dengan *achievement* terakhir dari pengguna. Pengguna dapat mencari pengguna lain dengan mengetikkan *username* pengguna lain dan menekan tombol *search* pada bagian *friend list*.



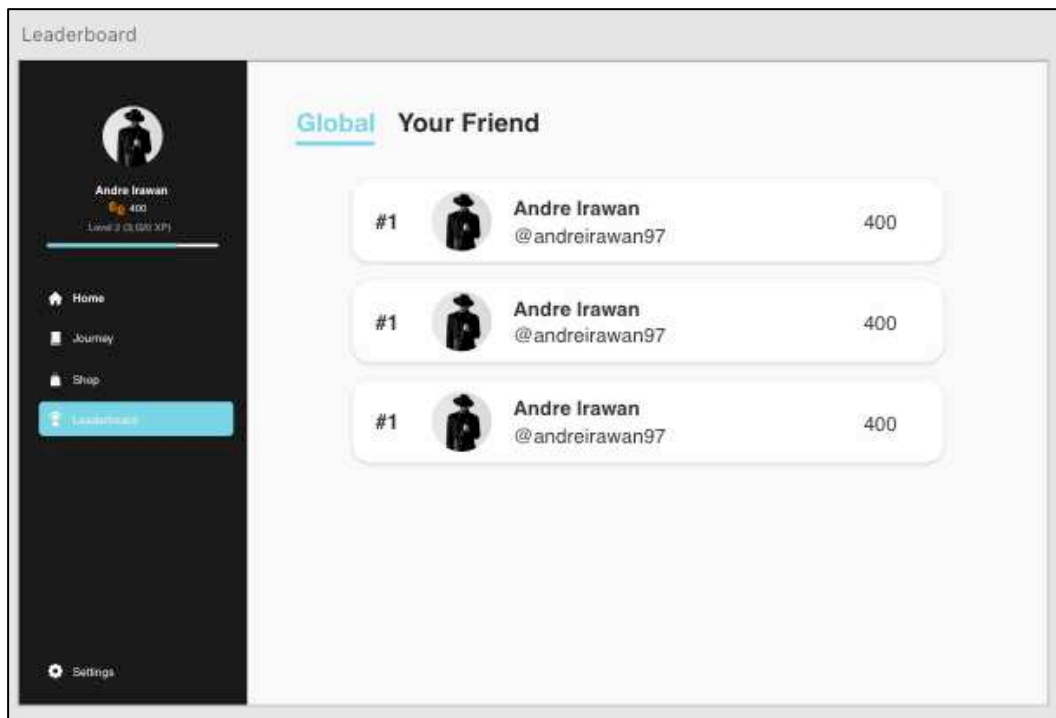
Gambar 3.15 *Mockup* Modul Journey

Gambar 3.15 merupakan *mockup* dari modul Journey yang akan memunculkan *level* dari komponen yang dapat dipelajari. Penamaan dari komponen dibuat menarik seperti komponen *text* dibuat menjadi *hello world* karena *teks* merupakan komponen terdasar pada React Native. Tampilan *level* yang tidak dapat diakses oleh pengguna akan berubah menjadi *icon* gembok dan tidak dapat ditekan. Apabila pengguna menekan *level* yang terbuka, maka pengguna akan dibawa ke halaman Level Scene.



Gambar 3.16 *Mockup* Modul Shop

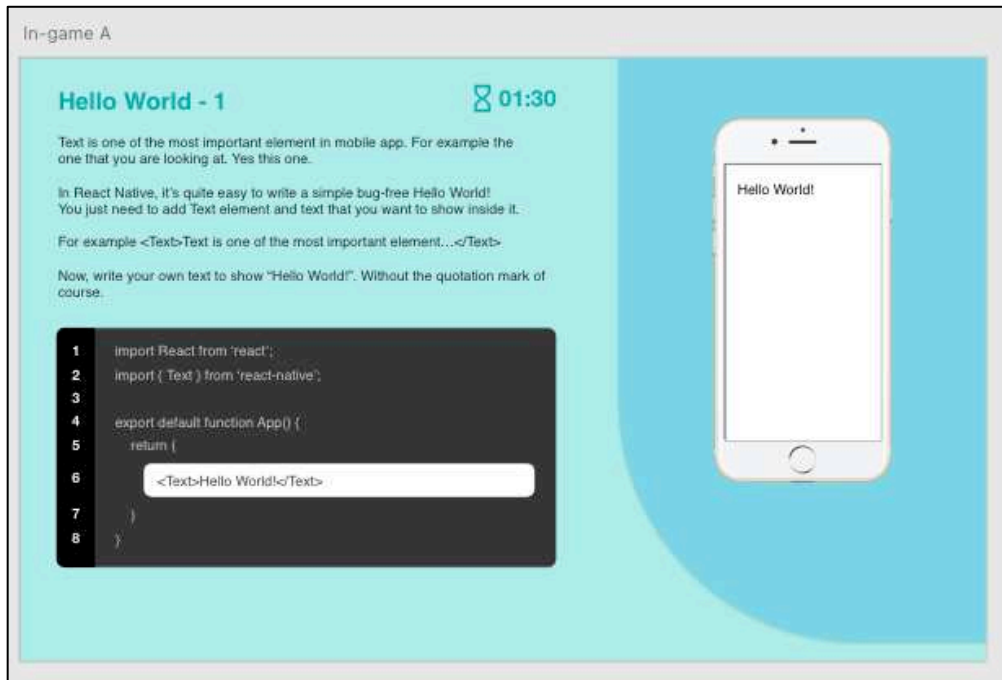
Pada Gambar 3.16, tampilan modul Shop memiliki dua *tab* yaitu *avatars* dan *phone skins*. Pengguna dapat membeli *avatars* pada bagian *avatars* dan *phone skins* pada bagian *phone skins*.



Gambar 3.17 *Mockup* Modul Leaderboard

Sesuai dengan Gambar 3.17, pada tampilan modul Leaderboard terdapat dua *tab* yaitu *global* dan *your friend*. Untuk melihat urutan dari seluruh pengguna dapat memilih *tab global*. Untuk melihat urutan dari teman pengguna dapat memilih *tab your friend*. Informasi dari pengguna lain yang dapat dilihat pada tampilan modul Leaderboard adalah *avatar*, nama lengkap, *username*, dan *experience*.

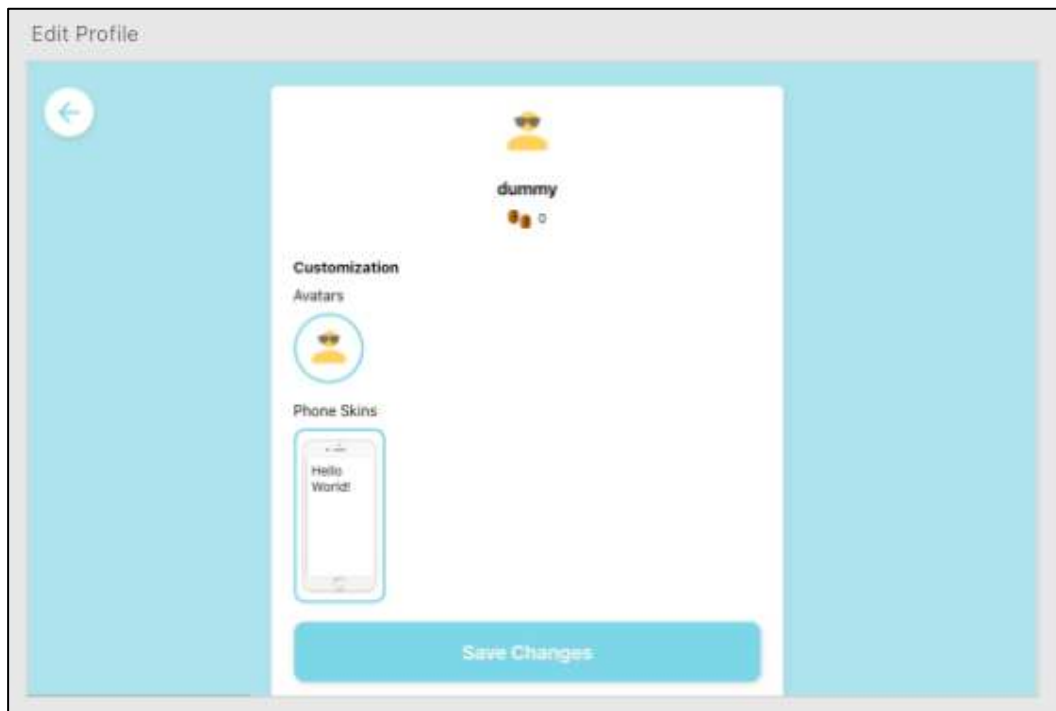
C. Level Scene



Gambar 3.18 *Mockup* Halaman Level Scene

Pada Gambar 3.18, pada Halaman Level Scene terdapat nama dari *level* yang sedang dikerjakan dan terdapat juga sebuah penanda waktu dari *level* yang sedang dikerjakan. Selain itu, terdapat deskripsi dari *level* yang dikerjakan dan sebuah potongan kode yang dimana pengguna dapat menuliskan jawaban. Pada bagian kanan terdapat hasil tampilan yang diharapkan dari potongan kode apabila telah terjawab dengan benar.

D. Edit Profile Scene



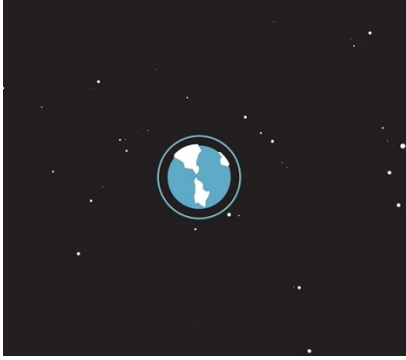
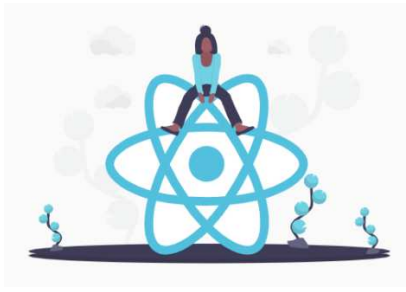
Gambar 3.19 *Mockup* Halaman Edit Profile Scene

Pada Gambar 3.19, halaman Edit Profile Scene terdapat tombol back yang akan membawa pengguna kembali ke halaman Main Scene dan elemen-elemen lain seperti *avatar*, *username*, dan *coffee beans*. Pengguna kemudian dapat mengganti *avatar* dan *phone skin* melalui halaman ini. Apabila pengguna telah mengganti *avatar* dan *phone skin* dengan yang diinginkan, maka pengguna dapat menekan tombol *save changes* untuk menyimpan perubahan.






3.2.5 Pemilihan Aset

Aplikasi Reactive memakai beberapa aset. Daftar gambar yang digunakan pada aplikasi dapat dilihat pada Tabel 3.1. Daftar ikon yang digunakan pada aplikasi dapat dilihat pada Tabel 3.2. Daftar aset *avatar* yang digunakan dalam aplikasi ini dapat dilihat pada Tabel 3.3. Daftar aset *phone skin* dapat dilihat pada Tabel 3.4.

Tabel 3.1 Daftar Gambar

No.	Gambar	Penjelasan	Sumber Gambar
1.		Gambar Auth Scene	www.wallpapercave.com
2.		Gambar pada Home Scene	www.undraw.co

Tabel 3.2 Daftar Ikon

No.	Gambar	Penjelasan	Sumber Gambar
1.		Ikon <i>home</i>	Material Community Icons
2.		Ikon <i>journey</i>	Material Community Icons
3.		Ikon <i>shop</i>	Material Community Icons
4.		Ikon <i>leaderboard</i>	Material Community Icons
5.		Ikon <i>coffee beans</i>	www.flaticon.com






Tabel 3.2 Daftar Ikon (lanjutan)

No.	Gambar	Penjelasan	Sumber Gambar
6.		Ikon <i>experience</i>	www.flaticon.com
7.		Ikon <i>achievements</i>	www.flaticon.com
8.		Ikon <i>friend list</i>	www.flaticon.com
9.		Ikon gembok ketika <i>level</i> atau <i>achievement</i> terkunci	www.flaticon.com
10.		Ikon ketika <i>achievement</i> terbuka dan <i>modal</i> berhasil	www.flaticon.com



Tabel 3.2 Daftar Ikon (lanjutan)

No.	Gambar	Penjelasan	Sumber Gambar
11.		Ikon <i>log out</i>	www.flaticon.com
12.		Ikon <i>back</i> yang dipakai pada halaman Level Scene dan Edit Profile Scene	www.flaticon.com
13.		Ikon <i>timer</i> pada halaman Level Scene	www.flaticon.com
14.		Ikon <i>clear text</i> pada halaman Level Scene	www.flaticon.com
15.		Ikon ketika pengguna berhasil menyelesaikan <i>level</i>	www.flaticon.com


Tabel 3.2 Daftar Ikon (lanjutan)

No.	Gambar	Penjelasan	Sumber Gambar
16.		Ikon ketika pengguna gagal menyelesaikan <i>level</i>	www.flaticon.com
17.		Ikon pada <i>level Hello World</i>	www.flaticon.com
18.		Ikon pada <i>level Perspective</i>	www.flaticon.com
19.		Ikon pada <i>level Portrait</i>	www.flaticon.com
20.		Ikon pada <i>level Loading</i>	www.flaticon.com


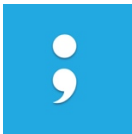


Tabel 3.2 Daftar Ikon (lanjutan)

No.	Gambar	Penjelasan	Sumber Gambar
21.		Ikon pada <i>level Type In</i>	www.flaticon.com
22.		Ikon pada <i>level Touchable</i>	www.flaticon.com


Tabel 3.3 Daftar Aset *Avatar*

No.	Gambar	Penjelasan	Sumber Gambar
1.		<i>Default avatar</i> saat pengguna pertama kali mendaftar	Google Image

Tabel 3.3 Daftar Aset *Avatar* (lanjutan)

No.	Gambar	Penjelasan	Sumber Gambar
2.		<i>Avatar stickman</i>	Google Image
3.		<i>Avatar semicolon</i>	Google Image
4.		<i>Avatar the atom</i>	Google Image
5.		<i>Avatar the fallen coder</i>	Google Image

Tabel 3.3 Daftar Aset *Avatar* (lanjutan)

No.	Gambar	Penjelasan	Sumber Gambar
6.		<i>Avatar millionaire</i>	Google Image

Tabel 3.4 Daftar Aset *Phone Skin*

No.	Gambar	Penjelasan	Sumber Gambar
1.		<i>Phone skin iPhone</i> 6	Google Image
2.		<i>Phone skin Nexus</i> 5	Google Image