BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kedudukan penulis dalam pelaksanaan kerja magang ini berada di bawah pengawasan dari KodeFox. Penulis kemudian ditetapkan sebagai *Front-End Software Engineer* pada Divisi Software Engineer.

Koordinasi pelaksanaan kerja magang dibawah pengawasan Daniel Sukmana selaku Software Engineer dari KodeFox. Selain itu, koordinasi kerja magang juga dibimbing oleh Ibu Jovita Zhang selaku HR & PR Manager di KodeFox.

3.2 Tugas Yang Dilakukan

Tugas yang dilakukan adalah membuat sebuah aplikasi berbasis *mobile* dengan konsep *social media* yang bernama Snapin dengan menggunakan bahasa pemrograman TypeScript dan menggunakan *framework* React-Native. Pembuatan *front-end* ditujukan untuk menghasilkan *user interface* yang memenuhi ketentuan *user experience* yang baik dan benar.

Aplikasi Snapin dikerjakan oleh sebuah tim yang terdiri dari Daniel Sukmana, Daniel Indra Cahyadi, Jesslyn Tanmas, Kevin Lie, Malvin Hariyanto, dan Vincent Wendy. Pembagian tugas dalam pembuatan aplikasi dapat dilihat pada tabel 3.1.

Tabel 3.1 Pembagian Tugas Pembuatan Aplikasi Snapin

Nama	Tugas yang dilakukan
Daniel Sukmana	Pembimbing Lapangan
Daniel Indra Cahyadi	*Back-End Engineer*

Tabel 3.1 Pembagian Tugas Pembuatan Aplikasi Snapin (lanjutan)

Nama	Tugas yang dilakukan
Jesslyn Tanmas	*Project Manager*
Kevin Lie	*Front-End Engineer*
Malvin Hariyanto	*Front-End Engineer*
Vincent Wendy	*Designer*

3.3 Uraian Pelaksanaan Kerja Magang

Pelaksanaan kerja magang yang terdiri dari proses pelaksanaan, kendala yang ditemukan, dan solusi akan kendala yang ditemukan akan dijelaskan pada bagian ini.

3.3.1 Proses Pelaksanaan

Sebelum melakukan proses pelaksanaan proyek, pembimbing lapangan memberikan pengarahan terhadap *tools* yang akan digunakan dalam pembuatan aplikasi. *Tools* yang digunakan adalah Git & Github dan Expo untuk membantu dalam pengerjaan aplikasi Snapin. Dibuatlah *timeline* untuk pembuatan aplikasi Snapin yang diatur oleh *Project Manager*. Tabel realisasi kerja magang ditunjukkan pada Tabel 3.1 di bawah ini.

Tabel 3.2 Realisasi Kerja Magang

Minggu	Kegiatan
1	- Melakukan pembelajaran terhadap React-DOM atau React-JS - Melakukan pembelajaran React-Native dan React-Navigation
2	- Melakukan pembelajaran terhadap Redux dan Saga - Melakukan pembelajaran terhadap TypeScript - Melakukan pembelajaran terhadap Git & Github
3	- Pembuatan *repository* GitHub - Konfigurasi *environment* yang dibutuhkan dalam proyek - Membuat *core-ui Icon dan core-ui Button* - Melakukan *stand up* dengan *Project Manager* (PM) - Menunggu dan melakukan *review* di GitHub
4	- Melakukan revisi terhadap *core-ui Button* - Membuat *core-ui TextInput* dan *core-ui Floating Action Button* - Melakukan *stand up* dengan PM - Menunggu dan melakukan *review* di GitHub

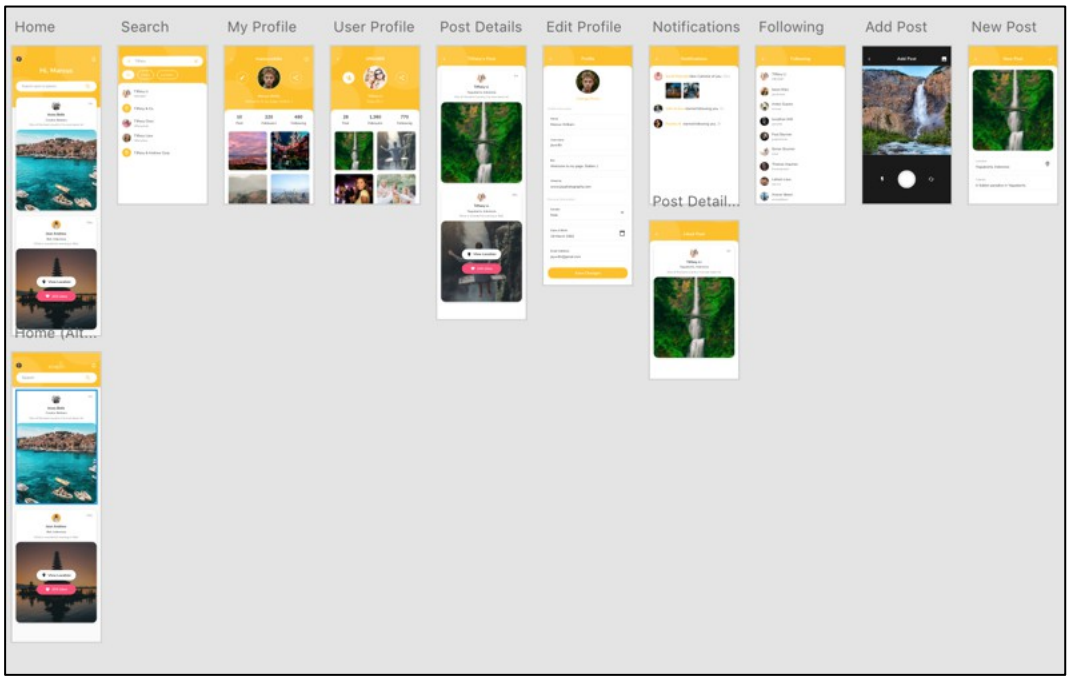
Tabel 3.2 Realisasi Kerja Magang (lanjutan)

Minggu	Kegiatan
5	- Melakukan revisi terhadap *core-ui TextInput* - Membuat *component Item Box TextInput, Item Box Gender,* dan *Item Box Calendar* - Melakukan *stand up* dengan PM - Menunggu dan melakukan *review* di GitHub
6	- Melakukan revisi terhadap semua component Item Box - Membuat *component Post* dan *SearchBar* - Melakukan *stand up* dengan PM - Menunggu dan melakukan *review* di GitHub
7	- Membuat *Home Scene, Forgot Password Scene & Notification Scene (coming soon), Post Detail Scene* - Melakukan *stand up* dengan PM - Menunggu dan melakukan *review* di GitHub
8	- Membuat *Search Scene, My Profile Scene, User Profile Scene,* dan *Edit Profile Scene* - Melakukan *stand up* dengan PM - Menunggu dan melakukan *review* di GitHub
9	Melakukan testing, memperbaiki *bug*, menyempurnakan, dan melakukan presentasi aplikasi ke seluruh karyawan kantor

Aplikasi yang dibuat adalah aplikasi dengan konsep *social media* yang dapat memungkinkan *sharing* foto antar pengguna (apabila sudah melakukan "*follow"*), melakukan *follow* atau *unfollow* pengguna lain, melakukan *like* atau *unlike post* pengguna lain, menulis *caption* yang menarik, dan mencantumkan lokasi foto tersebut diambil (*tag location*). Aplikasi ini bernama Snapin yang merupakan gabungan kata dari "Snap" dan "Pin", yang berarti foto dan melakukan pin ke suatu lokasi atau tag suatu lokasi dan membagikannya di aplikasi Snapin.

Proses perancangan desain aplikasi ini dilakukan oleh seorang dari tim desain dari KodeFox yang ditugaskan untuk masuk kedalam *project* Snapin ini. Perancangan dan permintaan pembuatan aplikasi diminta langsung oleh KodeFox dan desain yang dibuat menggunakan aplikasi Adobe XD.

Pada gambar 3.1 merupakan desain keseluruhan untuk aplikasi Snapin yang sudah dibuatkan oleh Vincent Wendy sebagai tim desain di proyek ini.



Gambar 3.1 Desain Aplikasi Snapin

Dibawah ini adalah hasil *screenshot* aplikasi Snapin dengan menggunakan simulator *mobile iOS iPhone XS*.

a. Core-UI Icon

Pada gambar 3.2 merupakan *core-ui icon* dalam aplikasi Snapin.

src > core-ui > Icon.tsx > ...

```tsx
import React from 'react';
import {View} from 'react-native';
import {Icon as ReactIcon} from 'react-native-elements';

import {WHITE} from '../constants/color';

type Props = {
  nameIcon?: string;
  typeIcon?:
    | 'material'
    | 'material-community'
    | 'font-awesome'
    | 'octicon'
    | 'ionicon'
    | 'foundation'
    | 'evilicon'
    | 'simple-line-icon'
    | 'zocial'
    | 'entypo'
    | 'feather'
    | 'antdesign';
  sizeIcon?: number;
  colorIcon?: string;
};

export default function Icon(props: Props) {
  let {nameIcon, typeIcon, sizeIcon, colorIcon, ...otherProps} = props;

  return nameIcon ? (
    <ReactIcon
      name={nameIcon}
      type={typeIcon || 'material'}
      size={sizeIcon || 30}
      color={colorIcon || WHITE}
      {...otherProps}
    />
  ) : (
    <View />
  );
}

```

Gambar 3.2 *Core-UI Icon*

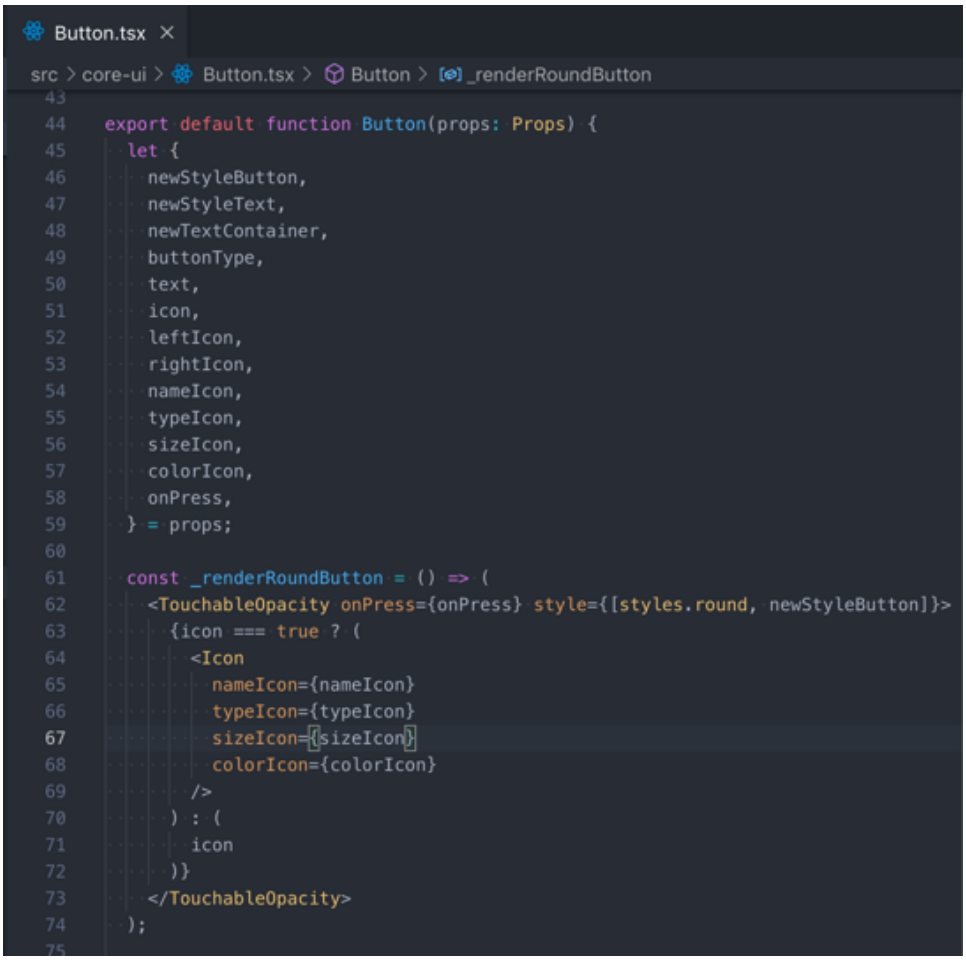
Pada gambar 3.2 menunjukkan *core-ui icon* berfungsi untuk menampilkan dan memfasilitasi segala kebutuhan aplikasi Snapin dalam menampilkan *icon-icon* yang beragam. *Default value* juga sudah diberikan didalam *core-ui icon*, agar ketika *error* tidak muncul *icon* tanda tanya melainkan icon default tersebut.

b. Core-UI Button

Berikut adalah *core-ui button* dalam aplikasi Snapin. *Core-ui button* berfungsi untuk membantu penggunaan tombol-tombol yang dibutuhkan dalam aplikasi Snapin ini. *Core-ui button* berfungsi untuk meng-*handle* segala kebutuhan yang diperlukan dalam aplikasi Snapin ini. Mulai dari menyiapkan berbagai bentuk yang tersedia, *styling*, dan fungsi-fungsi yang dapat diterima dan memproses fungsi tersebut. Fungsi yang diterima dalam *props* hanya berupa fungsi dengan *return void* karena fungsinya ada di *scene* ketika dibutuhkan disuatu *scene* tersebut *logic*-nya seperti apa.

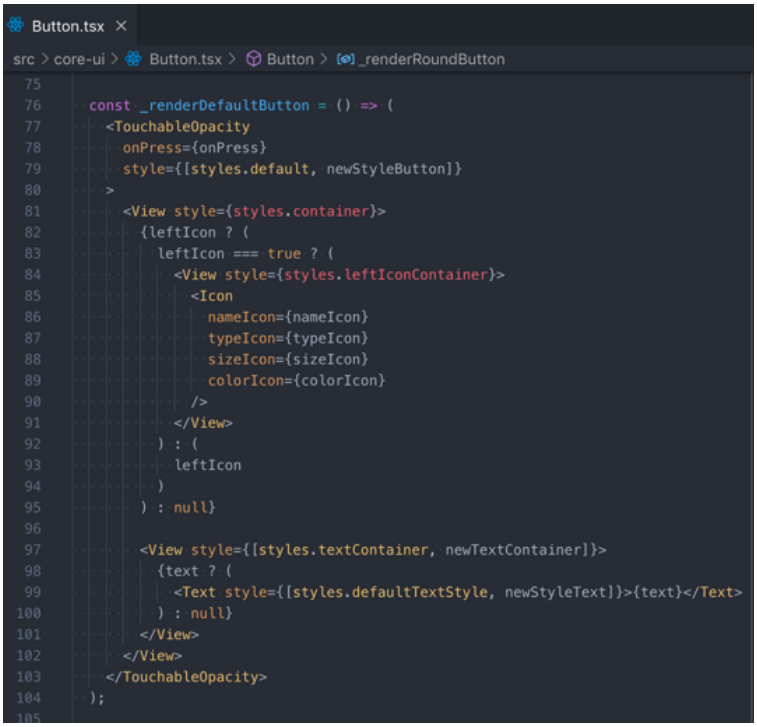
Gambar 3.3 *Core-UI Button Props*

Gambar 3.3 menunjukkan berbagai *props* yang dapat diterima di dalam *core-ui button* ini. Beserta segala import yang dibutuhkan baik dalam *react* maupun konfigurasi atau *core-ui* lain dalam *core-ui button* ini.



Gambar 3.4 *Core-UI Button Round*

Gambar 3.4 menunjukkan *core-ui button* untuk me-*render* jenis *button* yang berbentuk bulat. Karena dalam aplikasi Snapin ini ada *button* dengan bentuk bulat dan memiliki *icon* di dalam *button* tersebut. Sehingga dibutuhkan *core-ui icon* yang juga sudah dibuat sendiri.



Gambar 3.5 *Core-UI Button Default*

Pada gambar 3.5 menunjukkan *core-ui button* yang me-*render button* dengan bentuk *default* yaitu kotak dengan warna *primary color* yang sudah ditentukan dari awal.

Gambar 3.6 *Core-UI Button Search*

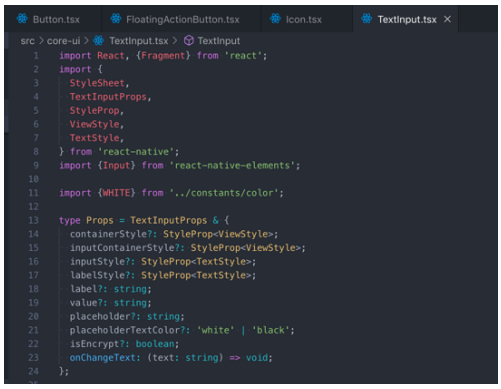
Pada gambar 3.6 menunjukkan *core-ui button* me-*render button* khusus untuk *search button*. Karena dalam aplikasi ini yang digunakan pada halaman *Home* hanyalah berbentuk *search button*, ketika ditekan baru akan dialihkan ke halaman *Search*. Lalu yang di *return* adalah tergantung dari *buttonType props* yang diberikan kedalam *core-ui button*, *core-ui button* ini sudah meng-*cover* segala kebutuhkan *button* dalam aplikasi Snapin.

Gambar 3.7 *Core-UI Button Styling*

Pada gambar 3.7 menunjukkan segala *styling* yang dipakai dalam *core-ui button* ini.

c. Core-UI TextInput

Berikut adalah *core-ui textinput* yang digunakan untuk memfasilitasi segala kebutuhan textinput dalam aplikasi Snapin.



Gambar 3.8 *Core-UI TextInput Props*

Pada gambar 3.8 menunjukkan segala kebutuhan *import* dan *props* yang dapat diterima oleh *core-ui textinput*. *Props* tersebut dibutuhkan untuk me-render sebuah *textinput*.

Gambar 3.9 *Core-UI TextInput*

Pada gambar 3.9 menunjukkan *textinput* yang akan di-*render* oleh *core-ui textinput*. Dimulai dari *value-value* yang dibuat dan kondisi-kondisi yang ditentukan untuk me-*render* segala kebutuhan dan kondisi untuk sebuah *textinput* yang sesuai dengan keperluan aplikasi Snapin.

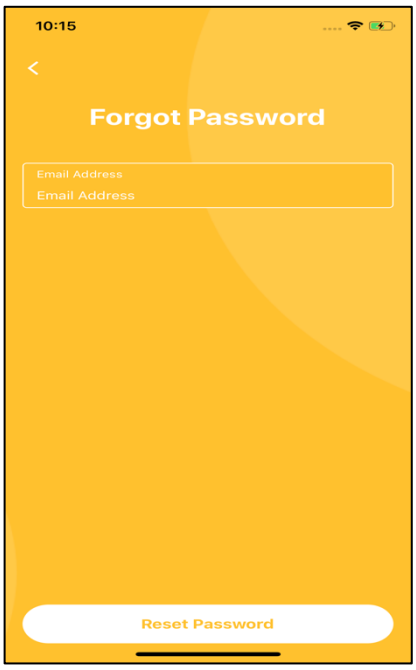
```
73
74  const styles = StyleSheet.create({
75    defaultInputContainerStyle: {
76      alignSelf: 'stretch',
77      borderBottomWidth: 0,
78    },
79    defaultLabelStyle: {
80      color: WHITE,
81      fontSize: 12,
82    },
83    defaultTextStyle: {
84      color: WHITE,
85      fontSize: 14,
86    },
87  });
```

Gambar 3.10 *Core-UI TextInput Styling*

Pada gambar 3.10 menunjukkan *styling* yang digunakan dalam *core-ui textinput.*

d. Halaman *Forgot Password*

Berikut adalah halaman *Forgot Password* dalam aplikasi Snapin.

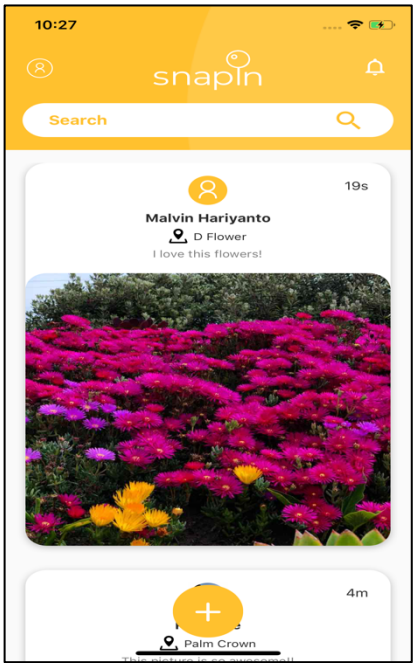


Gambar 3.11 Halaman *Forgot Password*

Pada gambar 3.11 pengguna dapat memasukkan alamat *e-mail* dan melakukan *reset password*. Akan tetapi, fitur ini belum dapat diimplementasikan, hanya dalam bentuk tampilannya saja, sehingga ketika ditekan tombol "Reset Password" maka akan menampilkan halaman *Notification*. Fitur ini tidak sempat terselesaikan dikarenakan keterbatasan waktu.

e. Halaman *Home*

Berikut adalah halaman *Home* dalam aplikasi Snapin.



Gambar 3.12 Halaman *Home*

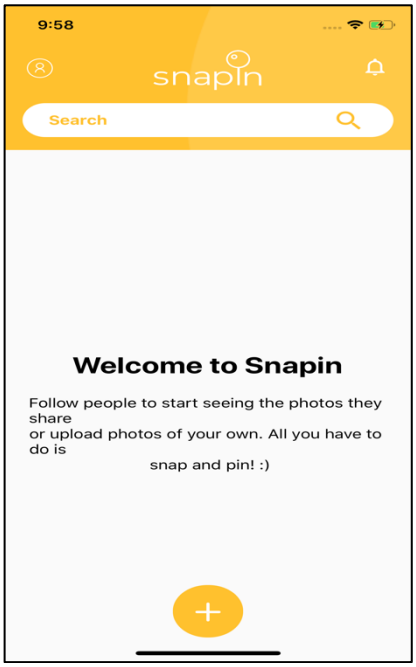
Pada gambar 3.12, tampilan *home* terdiri dari banyak *card* yang merupakan *list* dari *post* yang dimiliki oleh pengguna lain yang diikuti oleh pengguna maupun *post* yang dimiliki oleh diri sendiri. Jika terdapat lebih dari satu *post* maka pengguna dapat melakukan *scrolling* untuk melihat *post* yang lain.

Floating action button akan mengarahkan pengguna ke halaman *upload* jika ditekan. Kemudian, *floating action button* ini pun dapat menghilang ketika pengguna sedang melakukan *scrolling* dan muncul kembali apabila sudah berhenti melakukan *scrolling*.

Di bagian kiri atas terdapat sebuah *icon* atau gambar *avatar* dari pengguna yang dapat ditekan supaya diarahkan ke halaman *My Profile*.

Bagian kanan atas terdapat *icon* yang mengarahkan pengguna ke halaman *Notification*.

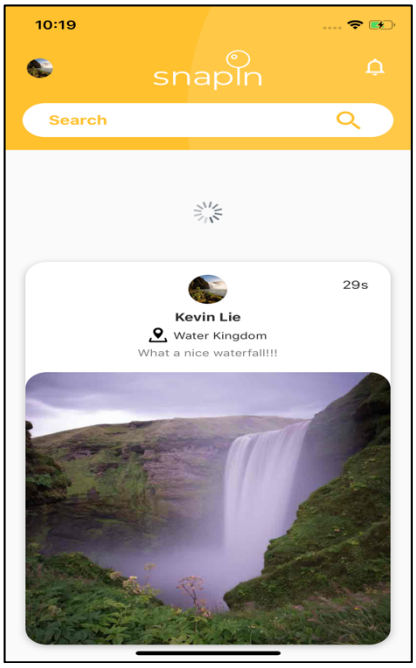
Lalu terdapat *button* berbentuk *searchbar* yang terletak di bawah logo Snapin berguna untuk mencari pengguna atau lokasi yang terdapat pada *post,* ketika button tersebut ditekan akan diarahkan ke halaman *Search*.



Gambar 3.13 Halaman *Home Default*

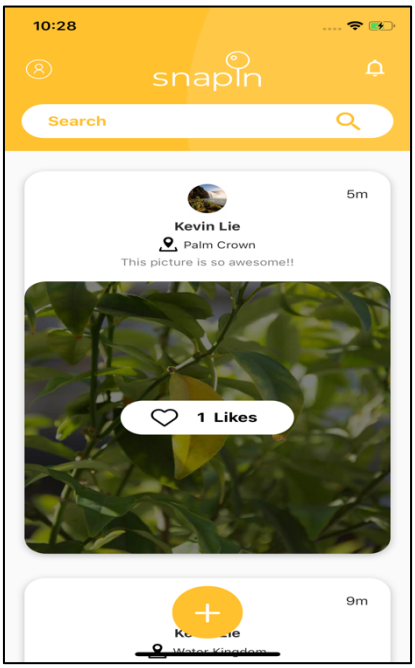
Namun jika baru pertama kali *login* dan belum "*follow*" siapapun maka akan muncul *UI* seperti pada gambar 3.13.

Setiap *card* berisi gambar *avatar*, nama pengguna, lokasi, *caption*, gambar yang di *post*, dan *timestamp* pada kanan atas *card*. Apabila gambar *avatar* dari post tersebut ditekan, akan mengarahkan *user* ke halaman *My Profile* atau *User Profile* tergantung *post* siapa yang ditekan. Lalu apabila gambar dari suatu *card* dalam suatu *post* ditekan, akan muncul *button likes* yang menunjukkan informasi berapa jumlah *like* dalam foto tersebut dan apakah *user* yang *login* sekarang sudah *like* atau belum foto tersebut serta *user* dapat melakukan *like* atau *unlike post* tersebut.



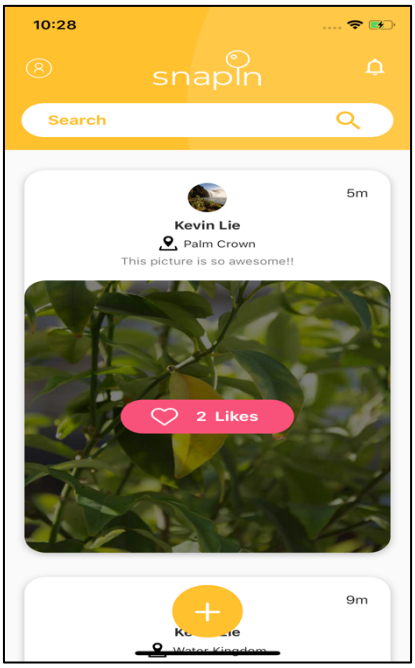
Gambar 3.14 Halaman *Home Refresh*

Pada gambar 3.14 terlihat fitur untuk *refresh* halaman. Seperti pada umumnya untuk melakukan *refresh* halaman cukup *scroll* ke bawah hingga muncul *icon refresh* dan akan *refresh* halaman *Home.*



Gambar 3.15 *Unliked Post* Pada Halaman *Home*

Pada gambar 3.15 berarti pengguna yang *login* sekarang belum melakukan *like* ke foto tersebut sehingga tampilan yang muncul adalah *button* berwarna putih.

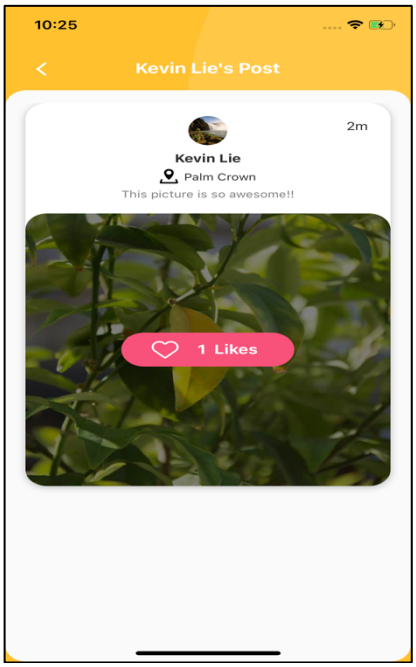


Gambar 3.16 *Liked Post* Pada Halaman *Home*

Terlihat bahwa *post* yang sudah di *like* berubah warna menjadi pink. Pada gambar 3.16 menandakan *post* tersebut sudah di *like* oleh pengguna.

f. Halaman *Post Detail*

Berikut adalah halaman *Post Detail* dalam aplikasi Snapin. Halaman ini berfungsi untuk menampilkan suatu *post* yang spesifik yang dipilih pengguna melalui halaman *My Profile* ataupun *User Profile*, tergantung pengguna ingin melihat lebih *detail post* siapa dan yang mana.

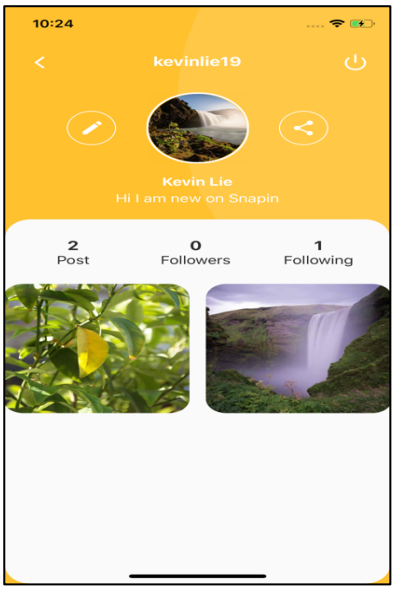


Gambar 3.17 Halaman *Post Detail*

Halaman *Post Detail* pada gambar 3.17 berisi rincian dari *post* pengguna, halaman ini dapat diakses ketika kita sudah berada di halaman *My Profile* atau *User Profile* dan memilih suatu *post* dari *user* tersebut. Pengguna juga dapat melihat berapa jumlah *likes post* tersebut dan melakukan *like* atau *unlike post* tersebut. Halaman ini mirip seperti halaman *Home*, hanya saja ini khusus menampilkan suatu *post* yang dipilih saja.

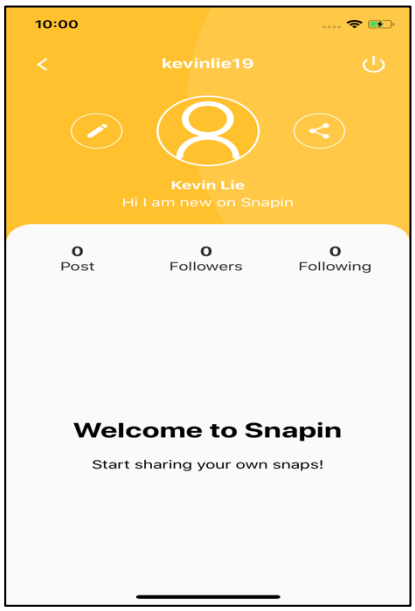
g. Halaman *My Profile*

Berikut adalah halaman *My Profile* dalam aplikasi Snapin. Halaman *My Profile* berisi *detail* informasi tentang *user* yang sedang *login*. Pengguna dapat melihat *detail* informasinya dan mengganti informasi tentang dirinya ataupun melakukan *logout*. Ada fitur *share user* namun fitur ini juga belum dapat direalisasikan karena keterbatasan waktu.



Gambar 3.18 Halaman *My Profile*

Pada gambar 3.18 terdapat *icon shutdown* di kanan atas yang berarti ketika ditekan akan melakukan *logout user* dan kembali ke halaman *Welcome.* Di kiri atas terdapat *icon back* yang berarti akan me-*navigate user* kembali ke halaman sebelumnya. Lalu ada *icon button pencil* untuk menuju ke halaman *Edit Profile.* Pada halaman ini terdapat juga jumlah *post*, jumlah *followers* , dan juga jumlah *following* serta semua *post* yang sudah di *post* oleh pengguna. Serta *post-post* pengguna dapat ditekan dan akan menuju ke halaman *Post Detail*.

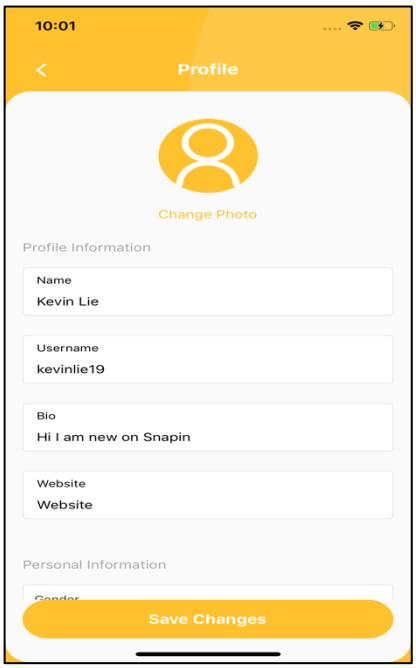


Gambar 3.19 Halaman *My Profile* Default

Gambar 3.19 merupakan tampilan *default* dari halaman *My Profile*. Ada beberapa konfigurasi yang diambil dari data pengguna ketika melakukan *register* dan ada beberapa konfigurasi dengan data *default* yang sudah ditentukan dari bagian Back-End aplikasi Snapin agar ada data yang ditampilkan dan dapat digunakan untuk menampilkannya di aplikasi Snapin.

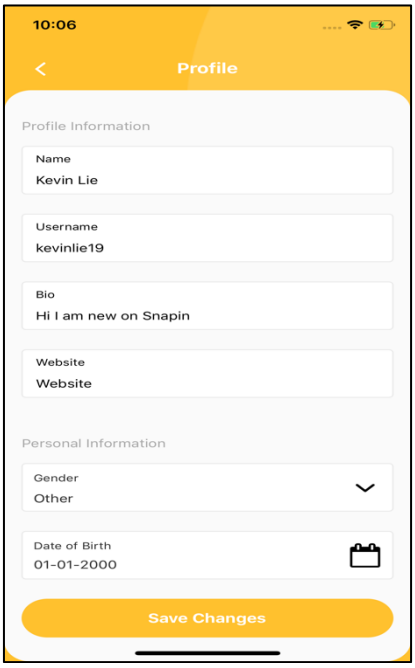
h. Halaman *Edit Profile*

Berikut adalah halaman *Edit Profile* dalam aplikasi Snapin. Pada halaman ini dibagi menjadi 2 jenis informasi, yaitu bagian *Profile Information* dan *Personal Information*. Bagian *profile information* merupakan bagian yang ditampilkan di halaman *My Profile* dan dapat dilihat oleh pengguna lain bila memiliki *value*, kecuali website karena website tidak ditampilkan. Bagian *personal information* merupakan informasi yang hanya disimpan oleh diri sendiri dan tidak dapat dilihat oleh pengguna lain dan tidak ditampilkan dimanapun, hanya sebagai preferensial diri sendiri dan disimpan dalam *database* aplikasi Snapin.



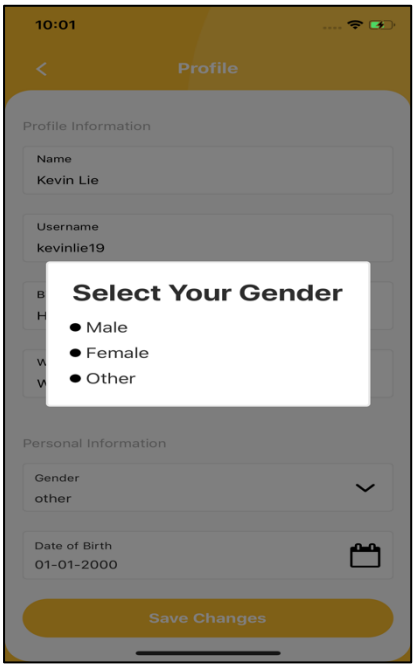
Gambar 3.20 Halaman *Edit Profile* Bagian *Profile Information*

Pada gambar 3.20 data diri pengguna dapat di-*edit* sesuai dengan kemauan pengguna, namun bagian *username* tidak dapat di-*edit*. *Profile Information* berisi *detail* data pengguna dengan *name* dan *username* dari *user* ketika mereka *register* dan *default value untuk bio* dan *website*.



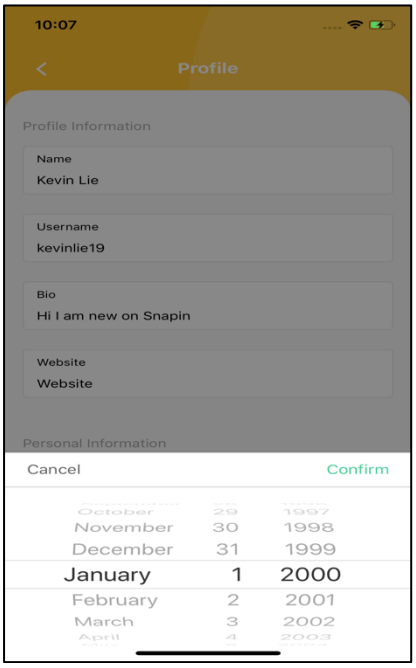
Gambar 3.21 Halaman *Edit Profile* Bagian *Personal Information*

Pada gambar 3.21, bagian *personal information* berisi data pribadi pengguna yang memiliki *default value* untuk *gender* dan *Date of Birth*.



Gambar 3.22 Halaman *Edit Profile* Saat *Select Gender*

Untuk pemilihan *gender* maka akan muncul sebuah *modal* yang berada di tengah layar aplikasi seperti pada gambar 3.22.

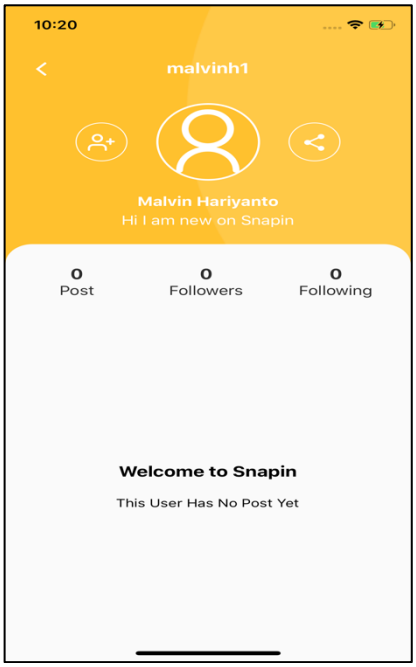


Gambar 3.23 Halaman *Edit Profile* Saat *Select* Date of Birth

Untuk pemilihan *Date of Birth* maka akan muncul modal berupa *datepicker* yang berada di bagian bawah layar aplikasi. Tampilan bervariasi tergantung dari *platform* mana aplikasi Snapin dijalankan (*datepicker* Android berbentuk seperti *calendar*).

i. Halaman *User Profile*

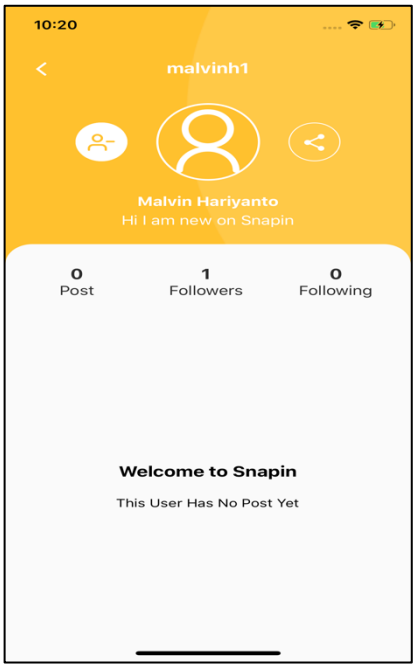
Berikut adalah halaman *User Profile* dalam aplikasi Snapin. Pada halaman ini hampir mirip seperti halaman *My Profile* namun pada halaman ini kita hanya bisa melihat informasi pengguna lain dan dapat *follow* atau *unfollow* pengguna lain. Tidak bisa *edit profile* dan *logout* pada halaman ini, namun sama-sama bisa melihat *post detail* apabila pengguna tersebut pernah melakukan *post* setidaknya minimal satu kali.



Gambar 3.24 Halaman *User Profile Default*

Gambar 3.24 merupakan tampilan *default* dari halaman *User Profile*. Data dari pengguna tersebut adalah data *default* ketika pengguna tersebut baru saja

register dan belum melakukan apapun di aplikasi Snapin, sehingga ketika diakses oleh orang lain maka data yang tampil adalah data yang masih *default*.

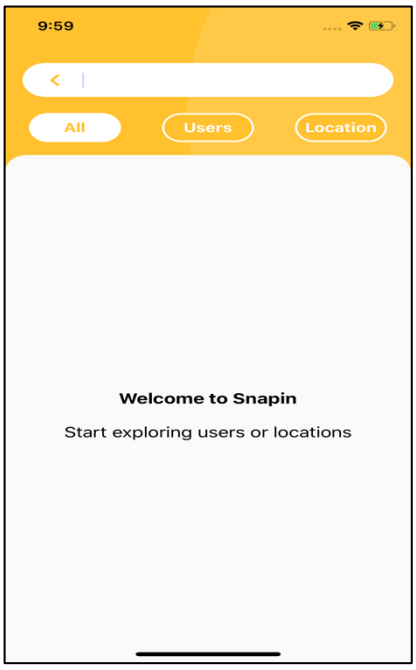


Gambar 3.25 Halaman *User Profile Followed*

Pada Halaman ini, pengguna dapat *follow* (gambar 3.24 belum *follow* pengguna tersebut) atau *unfollow* (gambar 3.25 sudah *follow* pengguna tersebut) pengguna lain dan melihat *post* dari pengguna lainnya. Tampilan *post-post* pengguna tersebut tersusun seperti layaknya halaman *My Profile*. Jika tidak ada *post* sama sekali, maka *default*-nya adalah tulisan "*Welcome to Snapin. This User Has No Post Yet*".

j. Halaman *Search*

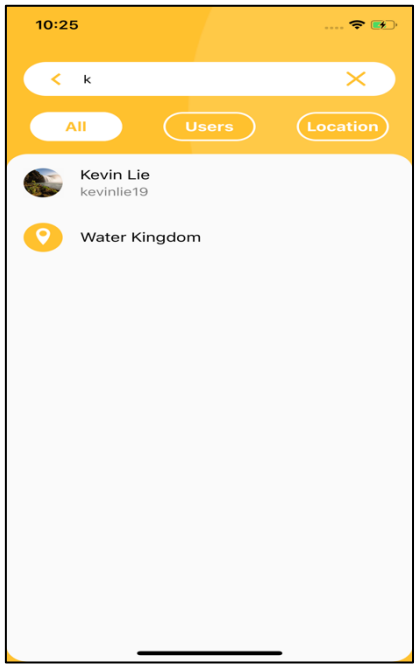
Berikut adalah halaman *Search* dalam aplikasi Snapin. Halaman ini digunakan untuk mencari pengguna ataupun lokasi yang ada dalam *database* Snapin.



Gambar 3.26 Halaman *Search*

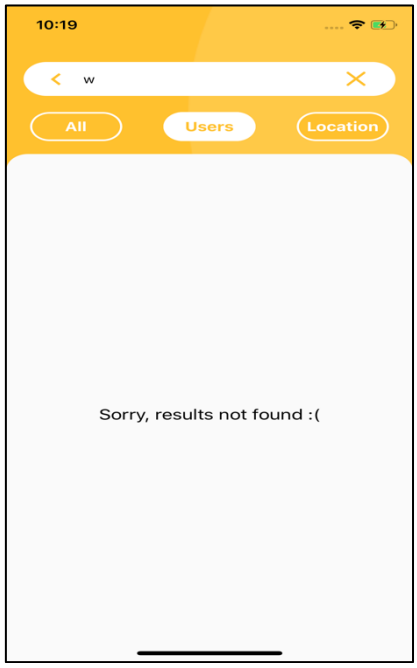
Pada awalnya, muncul tulisan *default* seperti pada gambar 3.26 yaitu "*Welcome to Snapin. Start exploring users or locations*". Pencarian berdasarkan *username* bagi pengguna dan nama lokasi bagi lokasi. Pada halaman *Search*, pengguna dapat mencari pengguna lain yang menggunakan aplikasi Snapin dan lokasi-lokasi yang ada dalam semua *post* pengguna di aplikasi Snapin ketika mereka *post* dan *tag* sebuah tempat serta dibagi menjadi 3 kategori, yaitu "*All"* untuk mencari semua pengguna dan lokasi, "*Users"* untuk mencari pengguna lain saja, dan "Location" untuk mencari suatu lokasi saja. *TextInput SearchBar* memiliki fitur *autofocus* sehingga ketika pengguna masuk ke halaman ini, akan otomatis muncul keyboard untuk dapat mengaktifkan fitur *search*. Pada awalnya diinginkan ketika mencari suatu lokasi dan memilih lokasi tersebut, akan diarahkan ke suatu halaman yang memunculkan semua post yang tag lokasi tersebut, namun fitur tersebut juga

sudah disepakati untuk tidak diimplementasikan terlebih dahulu karena keterbatasan waktu. Lalu ada *icon back* untuk kembali ke halaman Home.



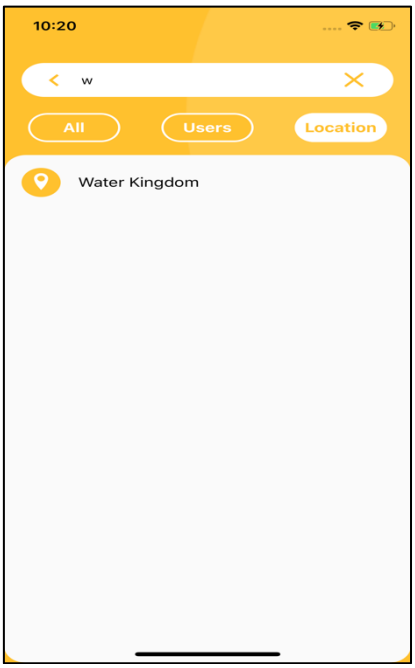
Gambar 3.27 Halaman *Search By All*

Pada gambar 3.27, pengguna dapat mencari semua *users* dan *locations* yang sesuai dengan menekan tombol "*All*".



Gambar 3.28 Halaman *Search By User*

Pada gambar 3.28 pengguna dapat mencari berdasarkan hanya terhadap pengguna lain saja dengan memilih "*Users"*. Lalu apabila tidak ditemukan maka akan muncul tampilan "*Sorry, results not found :(*", tampilan ini berlaku untuk semua halaman *Search By All*, *Search By Users*, dan *Search by Location*

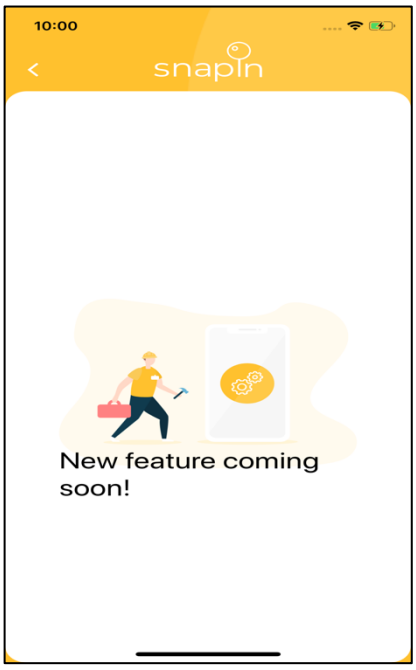


Gambar 3.29 Halaman *Search By Location*

Pada gambar 3.29 pengguna juga dapat mencari berdasarkan lokasi dengan memilih *button* "*Location*".

k. Halaman *Notification*

Berikut adalah halaman *Notification* dalam aplikasi Snapin. Pada halaman ini seharusnya segala bentuk notifikasi akan masuk ke halaman ini dan ditampilkan, baik di-*follow* pengguna lain maupun di-*like* post pengguna oleh pengguna lain.



Gambar 3.30 Halaman *Notification*

Halaman serta fitur pada *notification* belum terselesaikan dikarenakan waktu yang kurang, sehingga dimunculkan tampilan "*coming soon*" seperti pada gambar 3.30.

3.3.2 Kendala Yang Ditemukan

Berikut ini adalah kendala yang ditemukan dalam pembuatan rancang bangun aplikasi Snapin yang dibuat dalam kurang lebih dua bulan.

a. Kurangnya pengalaman serta pembelajaran pada teknologi yang digunakan (*React-Native*, *Typescript, Redux, dan Saga*) dan Git & GitHub sehingga berulang kali melakukan revisi pada saat proses pembuatan aplikasi Snapin dan revisi *setup project* berkali-kali ketika awal proses pembuatan. Hal ini menyebabkan terbuangnya waktu secara sia-sia yang seharusnya bisa digunakan untuk hal lain dalam penyelesaian aplikasi Snapin.
b. Kurangnya sumber daya manusia dalam pembuatan aplikasi sehingga dibutuhkan usaha yang lebih banyak dalam penyelesaiannya. Karena hanya

tiga orang yang melakukan *coding* dengan skala aplikasi yang lumayan besar dan rumit.

3.3.3 Solusi Atas Kendala Yang Ditemukan

Berdasarkan kendala-kendala yang ditemukan tersebut, berikut ini adalah solusi yang dapat digunakan untuk mengatasinya.

a. Mengumpulkan informasi dari internet dan bertanya pada karyawan kantor lainnya untuk mencari pembelajaran. Selain itu, dapat juga mencari contoh *coding* pada *project* lainnya di GitHub KodeFox yang dapat dijadikan sebagai acuan dalam pembuatan aplikasi.
b. Mengurangi jumlah fitur yang ada di dalam aplikasi, sehingga jumlah waktu yang dibutuhkan dalam membuat aplikasi berkurang. Prioritaskan fitur utama dalam aplikasi Snapin agar dapat berjalan dengan baik, sehingga dapat menyelesaikan aplikasi Snapin dengan tenggat waktu yang sudah ditentukan.