

# Konsep Import-Export

---

ada struktur seperti ini :

- src
  - assets
    - pages
      - SignIn
      - SignUp
      - Home
      - Profile
      - index.js

di dalam folder **SignIn, SignUp, Home, Profile** itu komponen biasa yang export nya *export default SignIn*

di dalam file pages/index.js itu

```
import SignIn from './SignIn'  
import signUp from './SignUp'  
import Home from './Home'  
import Profile from './Profile'  
  
export {SignIn, signUp, Home, Profile}
```

cara kedua export di dalam file pages/index.js

```
export * from './SignIn'
```

# Installasi Project

---

1. `myApps > npx react-native init foodMarketRN`

2. Setting local properties dan emulatornya

untuk installasi project , ini ada di materi react native basic, bisa baca catatan sebelumnya .

# Struktur Project Atomic Design

---

1. buat Folder:

- src/
  - assets/
  - components/

- atoms/
- molecules/
- config/
- pages/
- redux/
- router/
- utils/
- **App.js**

Notes :

folder config : untuk config apps kita, contoh config API  
 redux : seluruh setup untuk statemanagement menggunakan redux  
 router : seluruh configurasi routing apps kita  
 utils : setiap fungsi2 yang kita gunakan untuk mempermudah kerja kita,utils ini bisa digunakan di folder2 lainnya.

2. Pindahkan **App.js** ke dalam **src** folder
3. setting import nya yang error2 ,
4. Setting **App.js** nya , menjadi text biasa (hapus stylennya) dan rapihkan library yang tidak digunakan dengan **option+shift+o** . sama seperti membuat folder kosong.

## Install Library React Native SVG

1. Baca lagi catatan lo di react native basic

2. `myApps > npm install react-native-svg`

3. `myApps > npm install --dev react-native-svg-transformer`

4. ubah file **metro.config.js** dengan codingan berikut ini (amannya sih comment aja sebelumnya)

```
const { getDefaultConfig } = require("metro-config");

module.exports = (async () => {
  const {
    resolver: { sourceExts, assetExts }
  } = await getDefaultConfig();
  return {
    transformer: {
      babelTransformerPath: require.resolve("react-native-svg-transformer")
    },
    resolver: {
      assetExts: assetExts.filter(ext => ext !== "svg"),
      sourceExts: [...sourceExts, "svg"]
    }
  };
})();
```

5. restart dan install ulang servernya , (yarn start dan yarn android)

# Slicing Splash Screen

1. Buat folder Splashescreen dan index.js di halaman pages

- pages
  - **SplashScreen**
    - **index.js**
    - **index.js**

2. index.js di dalam folder pages, import index.js yang dari SplashScreen folder, lalu export .

3. import index.js dari pages dari splashscreen .

contoh :

- **pages/index.js**

```
import SplashScreen from './SplashScreen';  
  
export {SplashScreen};
```

- **App.js**

```
import {SplashScreen} from './pages';  
  
<SplashScreen/>
```

Intinya, import index.js yang ada di folder splashscreen ke dalam App.js

4. Import Logo :

konsep :

"setiap **folder** wajib memiliki **index.js** dan import harus melalui **index.js**"

- Buat folder assets/**Illustrations**
- Masukkan gambarnya di folder tsb
- buat file **index.js** di folder tsb
- di file nomor 3, import gambarnya dan export menggunakan cara nomor 3

```
import NamaSomething from './Logo.svg'  
  
export {NamaSomething}
```

- dalam folder **assets**, buat **index.js** juga

```
export * from './Illustration';
```

- Langsung aja deh panggil **Logo** nya di **App.js**

PAHAMAI KONSEP INI YA , INI PENTING

## 5. App.js

```
return (
  <View>
    <Text></Text>
    <Logo/>
  </View>
);
```

## 6. SplashScreen/index.js

atur background color dan view nya sehingga logo dan textnya berada di tengah tengah dengan menggunakan **flex, justifyContent, alignItems**

Atur textnya sesuai figma, fontsize dan colornya

# INSTALL REACT NAVIGATION

1. <https://reactnavigation.org/docs/getting-started/>
2. Install react navigationnya
3. Install dependencies yang dibutuhkan react navigationnya
4. edit file **App.js** menjadi seperti ini :

```
import 'react-native-gesture-handler';
import * as React from 'react';
import { NavigationContainer } from '@react-navigation/native';

export default function App() {
  return (
    <NavigationContainer>
      <SplashScreen/>
    </NavigationContainer>
  );
}
```

# Install Custom Fonts

1. download di <https://fonts.google.com/specimen/Poppins>
2. copy yang reguler, medium dan light , paste di folder  
MyApps/android/app/src/main/assets/fonts/

# SLICING - SIGNIN PAGE

## Cara membuat Customize button :

```
dibawah ini merupakan default button :  
  
import React from 'react'  
import { StyleSheet, Text, View } from 'react-native'  
  
const Button = () => {  
    return (  
        <View style={styles.container}>  
            <Text style={styles.text}>Sign In</Text>  
        </View>  
    )  
}  
  
export default Button  
  
const styles = StyleSheet.create({  
    text :{ fontSize: 15, fontFamily: 'Poppins-Medium', color : '#020202' },  
    container: {backgroundColor: '#FFC700', borderRadius: 8, padding: 12 }  
})
```

gimana cara membuat customize button ? contoh, background color nya jd abu abu in certain condition.

dibuat menggunakan props ,

1. Ganti text menggunakan props (easy)
2. Ganti color menggunakan props (medium)

caranya , kita buat dulu,

ini kan color mengirimkan prop berupa #1212  
di dalam function button, kita buat props color,  
dan di dalam style container, kita jadikan style container tersebut menjadi function yang  
menerima parameterer color .  
dan di parameter color tsb, kita taro warna color #1212 tsb,  
contoh :

file App.js

```
const App = () => {
  return (
    <View style={{ paddingHorizontal: 15, paddingTop: 10 }}>
      <View style={{ height:10 }}></View>
      <Button text="Sign In" color="#FFC700"></Button>
      <View style={{ height:10 }}></View>
      <Button text="Cancel" color="#8D92A3"></Button>
    </View>
  )
}
```

file Button/index.js

```
const Button = ({text, color}) => {
  return (
    <View style={styles.container(color)}>
      <Text style={styles.text}>{text}</Text>
    </View>
  )
}

export default Button

const styles = StyleSheet.create({
  text :{ fontSize: 15, fontFamily: 'Poppins-Medium', color : '#020202' },
  container: (color) => ({backgroundColor: color, borderRadius: 8,
padding: 12 })
})
```

**INTINYA, KETIKA INGIN PASANG PROPS DI DALAM STYLES, HARUS MENJADIKAN ISI  
STYLESNYA MENJADI FUNCTION , BUKAN OBJECT BIASA LAGI dan PASING  
PARAMETER**

belum selesai catatannya, intinya sih kalau bisa styles dan props , lu bisa buatnya .

## CARA PERPINDAHAN ANTAR HALAMAN

link lebih lanjut : <https://reactnavigation.org/docs/stack-navigator>

CONTOH CASE :

```
return (
  <NavigationContainer>
    <SplashScreen/>
    <Home/>
  </NavigationContainer>
)
```

gimana caranya agar dari splash bisa ke home ? gamungkin kan splashscreennya di comment terus pindah ke home .  
caranya pakai di bawah ini .

Router/StackNavigator.

1. install library nya (stack navigator dari link di atas)

```
myApps > npm install @react-navigation/stack
```

2. Buat component (rnfe) di dalam folder router/**index.js**.

3. import createStackNavigator dari library yang sudah diinstall

```
import {createStackNavigator} from '@react-navigation/native'
```

4. Buat variable bernama stack yang isinya fungsi createStackNavigator();

```
const Stack = createStackNavigator();
```

5. Isi fungsi router sebagai berikut :

```
const Router = () => {
  return (
    <Stack.Navigator>
      <Stack.Screen name="SplashScreen" component={SplashScreen}></Stack.Screen>
    </Stack.Navigator>
  )
}
```

6. Export componentnya dan masukkan ke dalam app.js

```
export default Router;
```

```
//file App.js

export default function App() {
  return (
    <NavigationContainer>
      <Router/> // sudah tidak tulisan <Splashscreen/> <Home/> lg, allin
      router
      </NavigationContainer>
    );
}
```

## 7. LALU GIMANA ?

logic aja sih, di component splashscreen lu buat useeffect , selama 2 detik, berubah halaman ke halaman selanjutnya .

8. Edit SplashScreen/index.js

9. buat function useeffect di dalamnya ,

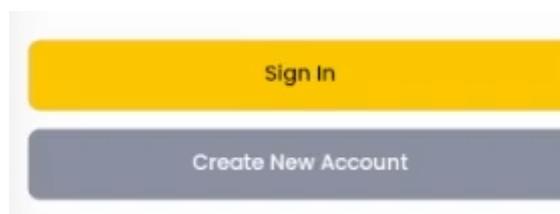
```
useEffect(() => {
  }, [])
```

10. di dalam function useEffect , buat function setTimeout

```
useEffect(() => {
  setTimeout(()=>{
    navigation.replace('signIn');
  }, 2000);
}, [])
```

# PINDAH HALAMAN DENGAN BUTTON

contoh :



button create new Account langsung menuju ke halaman sign up ,

konsep :

**"Caranya gunakan onPress dan isi dari onPressnya itu function  
navigation.navigate("signup")"**

cara :

1. di dalam komponen button, buat touchable oppacity sehingga viewnya dapat diklik

```
<TouchableOpacity>
  <View style={styles.container(color)}>
    <Text style={styles.text}>{text}</Text>
  </View>
</TouchableOpacity>
```

2. masukkan properti onPress dalam tag touchable di atas

```
<TouchableOpacity onPress="onPress">
```

3. pada komponen SignIn , komponen button dibuat properti juga yang isinya function navigate ke signup

```
//file SignIn/index.js
<Button onPress={()=>navigation.navigate("SignUp")}>
```

INTINYA , ketika ingin **pindah halaman**, gunakan **navigation.navigate("SignUp")** pakai action **onPress**

## Button onBack / Back SignUp



gimana caranya sign up punya back button, sedangkan sign in tidak ?

gampang

konsep : "**Jika header memiliki props onback , maka tampilkan icon back"**

cara :

1. buat props onBack pada component kecil (header)
2. buat kondisi pada komponen kecil seperti ini :

```
{
  onBack && (
    <View>
      <Icon/>
    </View>
  )
}
```

notes : jika user memasukan props onback maka tampilkan icon IcBack

3. di komponen besar (komponen signup) masukkan props onBack nya yang isinya fungsi.

intinya dari sign up itu , sama kaya sign in, tapi ada kondisi munculkan icon di atas.

## 11 Slicing Address Page pada Sign Up

intinya sama , membuat form dan button, tapi ini ada tambahan **FROM SELECT**,  
gimana caranya membuat form select ?

1. Install library nya di sini <https://github.com/react-native-picker/picker>

```
yarn add @react-native-picker/picker  
atau  
npm install @react-native-picker/picker --save
```

2. Import di component yang diinginkan

```
import {Picker} from '@react-native-picker/picker';
```

3. masukkan componentnya di dalam return

```
<Picker  
    selectedValue={this.state.language}  
    style={{height: 50, width: 100}}  
    onValueChange={(itemValue, itemIndex) =>  
        this.setState({language: itemValue})  
    }>  
    <Picker.Item label="Java" value="java" />  
    <Picker.Item label="JavaScript" value="js" />  
</Picker>
```

4. Styling sesuai dengan design.

## CARA MENAMBAHKAN BUTTOM NAVIGATOR

Di dalam home itu ada button untuk berpindah antar component/page,  
gimana caranya menambahkan itu semua ?  
konsepnya seperti ini :

"kita kan sudah pernah belajar dan menambahkan stack navigator untuk berpindah halaman, jadi di dalam stack navigator (router) kita buat stackscreen bernama **MainApp** lalu di screen mainapp tersebut, kita buat functional component untuk memanggil component2 navigatoriannya"

contoh :

1. Install Librarynya di <https://reactnavigation.org/docs/tab-based-navigation/>

```
npm install @react-navigation/bottom-tabs  
atau  
yarn add @react-navigation/bottom-tabs
```

2. Daftarkan navigation Component App di dalam stackNavigator

```
const Router = () => {  
  return (  
    <Stack.Navigator>  
      <Stack.Screen  
        name="SplashScreen"  
        component={SplashScreen}  
        options={{headerShown: false}}  
      />  
      <Stack.Screen  
        name="SignIn"  
        component={SignIn}  
        options={{headerShown: false}}  
      />  
      <Stack.Screen  
        name="SignUp"  
        component={SignUp}  
        options={{headerShown: false}}  
      />  
      <Stack.Screen  
        name="SignUpAddress"  
        component={SignUpAddress}  
        options={{headerShown: false}}  
      />  
      <Stack.Screen  
        name="SuccessSignUp"  
        component={SuccessSignUp}  
        options={{headerShown: false}}  
      />  
  
      //kita buat disini  
      <Stack.Screen  
        name="MainApp"  
        component={MainApp}  
        options={{headerShown: false}}  
      />  
      //kita buat disini  
  
    </Stack.Navigator>  
  );
```

```
};
```

Ioh main app ini manggil apa ? manggil function di atasnya (di dalam file yang sama)

3. Buat fungsi MainApp nya

```
//isinya ini

const Tab = createBottomTabNavigator();
const MainApp = () => {
  return (
    <Tab.Navigator tabBar={({props}) => <BottomNavigator {...props} />}>
      <Tab.Screen name="Home" component={Home} />
      <Tab.Screen name="Order" component={Order} />
      <Tab.Screen name="Profile" component={Profile} />
    </Tab.Navigator>
  );
};
```

ful lengkapnya :

```
import React from 'react';
import {createStackNavigator} from '@react-navigation/stack';
import {createBottomTabNavigator} from '@react-navigation/bottom-tabs';
import {
  EditProfile,
  FoodDetail,
  Home,
  Order,
  OrderDetail,
  OrderSummary,
  Profile,
  SignIn,
  SignUp,
  SignUpAddress,
  SplashScreen,
  SuccessOrder,
  SuccessSignUp,
} from '../pages';
import {BottomNavigator} from '../components';

const Stack = createStackNavigator();
const Tab = createBottomTabNavigator();

const MainApp = () => {
  return (
    <Tab.Navigator tabBar={({props}) => <BottomNavigator {...props} />}>
      <Tab.Screen name="Home" component={Home} />
      <Tab.Screen name="Order" component={Order} />
      <Tab.Screen name="Profile" component={Profile} />
    </Tab.Navigator>
  );
};

const Router = () => {
  return (
    <Stack.Navigator>
```

```
<Stack.Screen
  name="SplashScreen"
  component={SplashScreen}
  options={{headerShown: false}}
/>
<Stack.Screen
  name="signIn"
  component={SignIn}
  options={{headerShown: false}}
/>
<Stack.Screen
  name="SignUp"
  component={SignUp}
  options={{headerShown: false}}
/>
<Stack.Screen
  name="SignUpAddress"
  component={SignUpAddress}
  options={{headerShown: false}}
/>
<Stack.Screen
  name="SuccessSignUp"
  component={SuccessSignUp}
  options={{headerShown: false}}
/>
<Stack.Screen
  name="MainApp"
  component={MainApp}
  options={{headerShown: false}}
/>
<Stack.Screen
  name="FoodDetail"
  component={FoodDetail}
  options={{headerShown: false}}
/>
<Stack.Screen
  name="OrderSummary"
  component={OrderSummary}
  options={{headerShown: false}}
/>
<Stack.Screen
  name="SuccessOrder"
  component={SuccessOrder}
  options={{headerShown: false}}
/>
<Stack.Screen
  name="OrderDetail"
  component={OrderDetail}
  options={{headerShown: false}}
/>
<Stack.Screen
  name="EditProfile"
  component={EditProfile}
  options={{headerShown: false}}
/>
</Stack.Navigator>
);
};
```

```
export default Router;
```

# Create Custom Bottom Navigation

cara nya :

1. Buat Component nya dulu lah (rnfes), namanya bebas , tp better **BottomNavigator**, lalu export
2. menurut info di <https://reactnavigation.org/docs/bottom-tab-navigator/> , untuk menambahkan custom tab navigator, di dalam folder / route component kita, kita tambahkan properti dibawah ini pada tab Tab.Navigator di function MainApp

```
tabBar={props => <MyTabBar {...props} />}
```

jadinya

```
<Tab.Navigator tabBar={props => <BottomNavigator {...props} />}>
  {...}
</Tab.Navigator>
```

3. Masukkan code block di bawah ini (dari docs reactnavigation) ke dalam return pada component yang sudah kita buat di nomor 1 dan masukkan propsnya serta librarynya.

```
const focusedOptions = descriptors[state.routes[state.index].key].options;

if (focusedOptions.tabBarVisible === false) {
  return null;
}

return (
  <View style={{ flexDirection: 'row' }}>
    {state.routes.map((route, index) => {
      const { options } = descriptors[route.key];
      const label =
        options.tabBarLabel !== undefined
          ? options.tabBarLabel
          : options.title !== undefined
            ? options.title
            : route.name;

      const isFocused = state.index === index;

      const onPress = () => {
        const event = navigation.emit({
          type: 'tabPress',
          target: route.key,
          canPreventDefault: true,
        });
        if (!isFocused && !event.defaultPrevented) {

```

```

        navigation.navigate(route.name);
    }
};

const onLongPress = () => {
    navigation.emit({
        type: 'tabLongPress',
        target: route.key,
    });
};

return (
    <TouchableOpacity
        accessibilityRole="button"
        accessibilityState={isFocused ? { selected: true } : {}}
        accessibilityLabel={options.tabBarAccessibilityLabel}
        testID={options.tabBarTestID}
        onPress={onPress}
        onLongPress={onLongPress}
        style={{ flex: 1 }}
    >
        <Text style={{ color: isFocused ? '#673ab7' : '#222' }}>
            {label}
        </Text>
    </TouchableOpacity>
);
);
})}
</View>
);

```

jadinya seperti ini :

```

import React from 'react';
import {StyleSheet, TouchableOpacity, View} from 'react-native';

const BottomNavigator = ({state, descriptors, navigation}) => {

    const focusedOptions = descriptors[state.routes[state.index].key].options;

    if (focusedOptions.tabBarVisible === false) {
        return null;
    }

    return (
        <View style={{ flexDirection: 'row' }}>
            {state.routes.map((route, index) => {
                const { options } = descriptors[route.key];
                const label =
                    options.tabBarLabel !== undefined
                        ? options.tabBarLabel
                        : options.title !== undefined
                            ? options.title
                            : route.name;

                const isFocused = state.index === index;

```

```

const onPress = () => {
  const event = navigation.emit({
    type: 'tabPress',
    target: route.key,
    canPreventDefault: true,
  });

  if (!isFocused && !event.defaultPrevented) {
    navigation.navigate(route.name);
  }
};

const onLongPress = () => {
  navigation.emit({
    type: 'tabLongPress',
    target: route.key,
  });
};

return (
  <TouchableOpacity
    accessibilityRole="button"
    accessibilityState={isFocused ? { selected: true } : {}}
    accessibilityLabel={options.tabBarAccessibilityLabel}
    testID={options.tabBarTestID}
    onPress={onPress}
    onLongPress={onLongPress}
    style={{ flex: 1 }}
  >
    <Text style={{ color: isFocused ? '#673ab7' : '#222' }}>
      {label}
    </Text>
  </TouchableOpacity>
);
})}
</View>
);
};

export default BottomNavigator;

```

#### 4. perantik component / file BottomNavigator/index.js

- o Tambahkan Key pada touchable opacity (icon bottomnav) nya .  
key={index}
- o Tambahkan icon di atas tag (sebelumnya wajib import dan export dulu)
- o Buat function icon yang isinya logic switch case return icon2 yang sudah diimport
- o pada function Icon , masukkan parameter **label** nya . dan logic switch nya mengacu pada label,
- o pada function Icon, masukkan parameter **focus** dan masukkan kondisi if else oneline , isinya paramter focus itu , **isFocused**
- o yang perlu diperhatikan yaitu **Fungsi Icon** dan **Return BottomNavigator** nya

```
const Icon = ({label, focus}) => {
  switch (label) {
    case 'Home':
      return focus ? <IconHomeOn /> : <IconHomeOff />;
    case 'Order':
      return focus ? <IconOrderOn /> : <IconOrderOff />;
    case 'Profile':
      return focus ? <IconProfileOn /> : <IconProfileOff />;
    default:
      return <IconOrderOn />;
  }
};
```

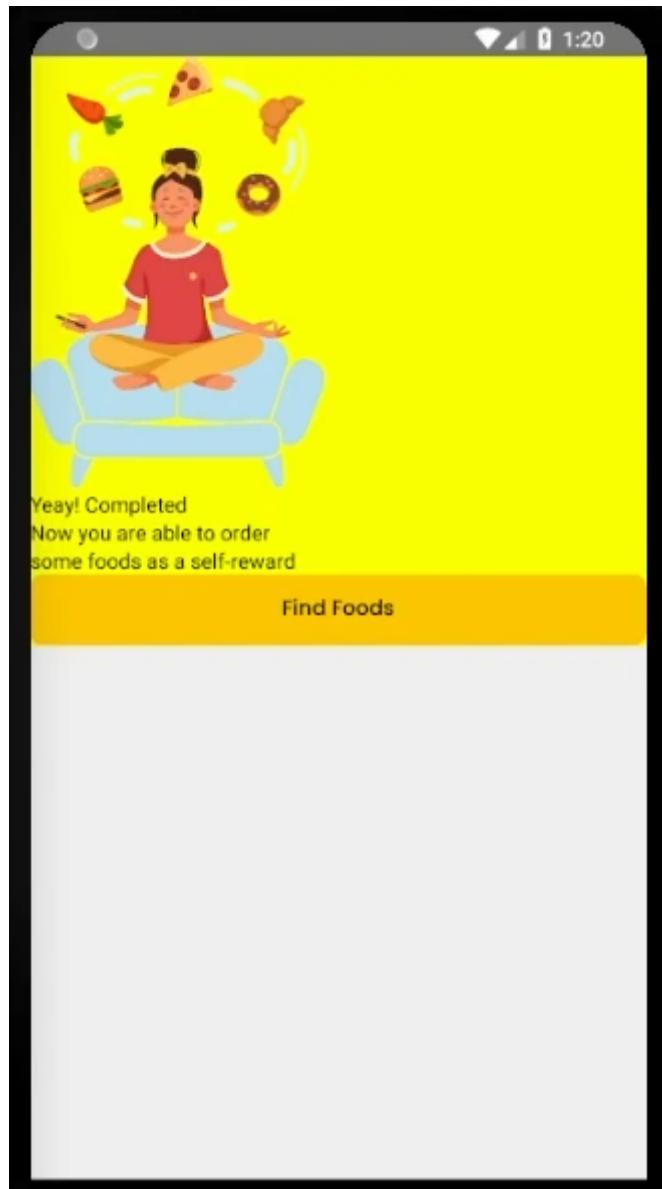
```
return (
  <TouchableOpacity
    key={index}
    accessibilityRole="button"
    accessibilityState={isFocused ? {selected: true} : {}}
    accessibilityLabel={options.tabBarAccessibilityLabel}
    testID={options.tabBarTestID}
    onPress={onPress}
    onLongPress={onLongPress}>
    <Icon label={label} focus={isFocused} /> //ini fungsi yang di atas
  </TouchableOpacity>
);
```

## 5. SEMPLENYA : KITA BUAT FUNGSI ICON UNTUK MENGISI RETURN ICON

# SIGN UP PAGE SUCCESS

---

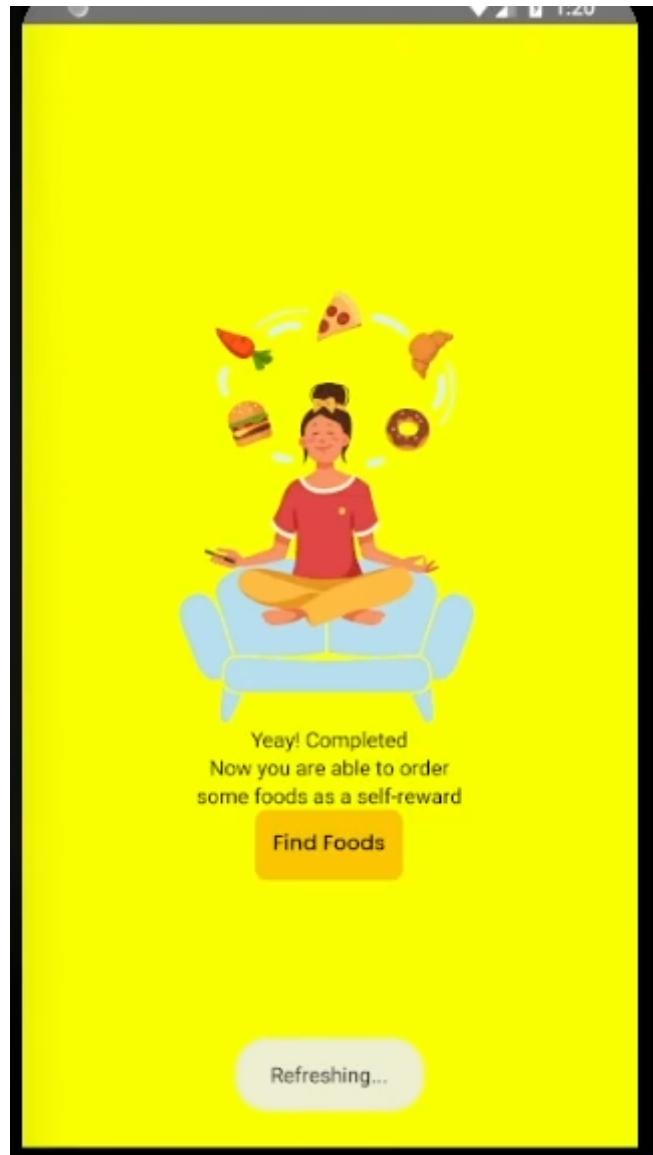
gimana caranya membuat button text yang di bawah ini menjadi di tengah ?



gampang, lu tinggal masukkan :

- Ilustrasi
- Text
- dan Component Button

ke dalam View , dan setting viewnya menjadi full (**flex : 1**) dan **justifyContent: 'center'** , **alignItems : 'center'**



setelah kaya gini ,

gimana cara memperbesar buttonnya ?

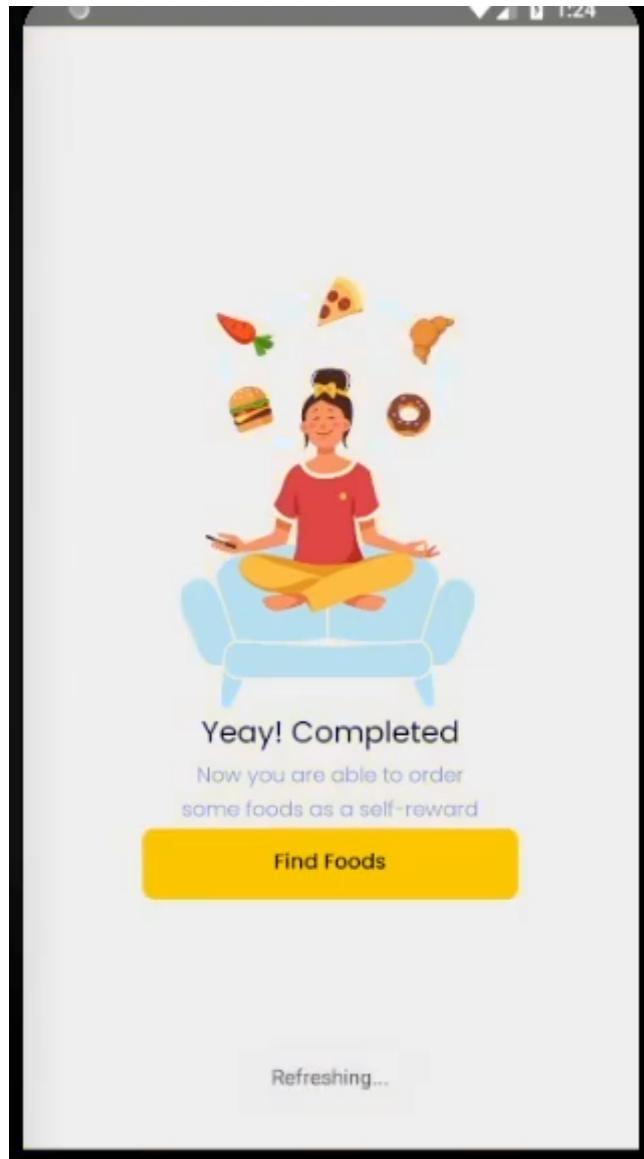
gampang :

**masukkan** komponen buttonnya **ke** dalam , lalu setting view nya menjadi **flex: 1, width: '100%'**.

kan jadi lebar tuh, jadi kaya gini nanti :



kalau udah kaya gini , nanti tinggal atur view nya aja memiliki padding kiri kanan 80 px (**paddingHorizontal: 80**) . jadi deh gini :



## 15 Home Page Welcoming User Section

---

Styling biasa , lu bisa insya Alah .

ntar weekend belajar aja buat , tp inget, jgn nyontek ya !



## 16 . Cara membuat foodcard scrollview ke samping

Konsep :

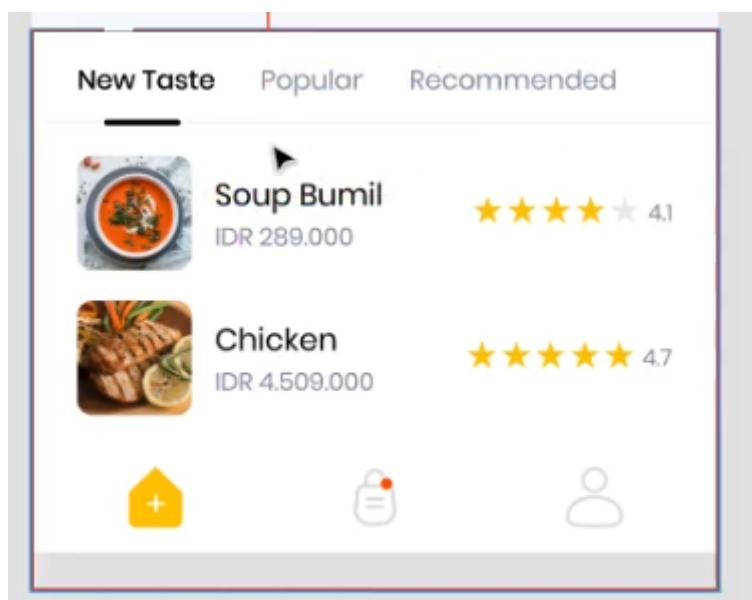
1. Buat comonent biasa isinya :
  - gambar
  - text
  - iconstaron, iconstarof dan text rating
2. Buat gambarnya ke bawah dulu gapapa
3. abis itu kalau udah semua sesuai (style tulisan dan gambar icon star nya)
4. styling containernya (permasing masing item) memiliki **shadowOffSet** dan **shadowOpacity** dan shadow radius
5. kalau udah semua ke bawah, tinggal di styling2 aja sesuai dengan arahan UI/UX.
6. buat container nya menjadi rata ke samping.
7. setelah itu masukkan ke dalam dan horizontal serta showhorizontalscrollindocatornya menjadi false .

stylenya :

```
container: {  
    width: 200,  
    backgroundColor: 'white',  
    borderRadius: 8,  
    shadowColor: 'black',  
    shadowOffset: {width: 0, height: 7},  
    shadowOpacity: 0.5,  
    shadowRadius: 10,  
    elevation: 14,  
    overflow: 'hidden',  
    marginRight: 24,  
},  
image: {width: 200, height: 140, resizeMode: 'cover'},  
content: {padding: 12},  
text: {fontSize: 16, fontFamily: 'Poppins-Regular', color: '#020202'},
```

## Swipe able Tabs and Click able Tabs

contoh case :



ini bisa di swipe dan ganti ke komponent populer dan bisa juga ke recommended,

gimana caranya ? **Pakai Library : [react-native-tab-view \(clickme\)](#)**.

langsung install aja

abis itu ikutin petunjuknya :

1. `MyApps>$ yarn add react-native-tab-view`

### 2. Quick Start from docs

```
import * as React from 'react';  
import { View, StyleSheet, Dimensions } from 'react-native';  
import { TabView, SceneMap } from 'react-native-tab-view';
```

```

const FirstRoute = () => (
  <view style={[styles.scene, { backgroundColor: '#ff4081' }]} />
);

const SecondRoute = () => (
  <view style={[styles.scene, { backgroundColor: '#673ab7' }]} />
);

const initialLayout = { width: Dimensions.get('window').width };

export default function TabviewExample() {
  const [index, setIndex] = React.useState(0);
  const [routes] = React.useState([
    { key: 'first', title: 'First' },
    { key: 'second', title: 'Second' },
  ]);

  const renderScene = SceneMap({
    first: FirstRoute,
    second: SecondRoute,
  });

  return (
    <TabView
      navigationState={{ index, routes }}
      renderScene={renderScene}
      onIndexChange={setIndex}
      initialLayout={initialLayout}
    />
  );
}

const styles = StyleSheet.create({
  scene: {
    flex: 1,
  },
});

```

Cara membuat :

1. Install done
  2. buat component view dan isikan tag <TabView....> di atas
- (notes : FirstRoute /SecondRoute itu maksudnya tab1 tab2 dst..)

```

<view>
  <TabView
    navigationState={{ index, routes }}
    renderScene={renderScene}
    onIndexChange={setIndex}
    initialLayout={initialLayout}
  />
</view>

```

3. Copy Use State s.d. render scenenya , jadinya :

```

const [index, setIndex] = React.useState(0);
const [routes] = React.useState([
  { key: 'first', title: 'First' },
  { key: 'second', title: 'Second' },
]);

const renderScene = SceneMap({
  first: FirstRoute,
  second: SecondRoute,
});

<view>
  <TabView
    navigationState={{ index, routes }}
    renderScene={renderScene}
    onIndexChange={setIndex}
    initialLayout={initialLayout}
  />
</view>

```

4. Buat fungsi yang isinya component view dan berikan stylenya (BG DAN FLEX)

```

const FirstRoute = () => (
  <view style={[styles.scene, { backgroundColor: '#ff4081', flex: 1}]} />
);

const SecondRoute = () => (
  <view style={[styles.scene, { backgroundColor: '#673ab7', flex: 1 }]} />
);

```

kurang lebih jd seperti ini :

```

const FirstRoute = () => (
  <view style={{ backgroundColor: '#ff4081', flex: 1 }} />
);

const SecondRoute = () => (
  <view style={{ backgroundColor: '#673ab7', flex: 1 }} />
);

const Home = () => {
  const [index, setIndex] = React.useState(0);
  const [routes] = React.useState([
    { key: 'first', title: 'First' },
    { key: 'second', title: 'Second' },
  ]);

  const renderScene = SceneMap({
    first: FirstRoute,
    second: SecondRoute,
  });
  return (
    <view style={{flex:1}}> //notes : INI ISI FLEX 1
      <TabView
        navigationstate={{ index, routes }}
        renderScene={renderScene}
  
```

```

        onIndexChange={setIndex}
        initialLayout={initialLayout}
      />
    </View>
  )
}

export default Home

```

## Custom Swipe able Tabs from *react-native-tab-view*



di Bab sebelumnya kan kita udah buat tuh yah <Tabview... /> nya, sekarang kita tambahin property namanya **renderTabBar={renderTabBar}** dan kita buat component baru di (di file yang sama aja, di atasnya komponent utama Home) dengan nama **renderTabBar**

contoh :

```

<TabView
  navigationState={{ index, routes }}
  renderScene={renderScene}
  onIndexChange={setIndex}
  initialLayout={initialLayout}
  renderTabBar={renderTabBar}
/>

```

```

const renderTabBar = (props) => (
  <TabBar
    {...props}
    indicatorStyle={{ backgroundColor: 'white' }}
    style={{ backgroundColor: 'pink' }}
  />
  //NOTES : KETIKA INGIN MENAMBAHKAN TAG TABBARINI HARUS IMPORT TABBAR FROM
  LIBRARY
);

```

ketika kita ingin memodifikasi tabBar nya , kita masukkan property **tabStyle** ke dalam property di tag <TabBar .../>

```
<TabBar tabStyle={{width: 'auto'}}/>
```

**gimana kalau mau styling pada tulisan *text, focus, dan color* nya ?**

caranya masukkan aja property namanya **renderLabel** isinya function memiliki parameter, (**route, focused, color**) dan mereturn text .

contoh :

```
<TabBar

  renderLabel={({route, focused, color}) =>

    <Text style={{fontFamily: 'Poppins-Medium', color: focused ? '#item' : '#abuabu'}}>{route.title}</Text>

  }
/>
```

intinya : **Cara di atas digunakan untuk MEMBUAT TEXTNYA FOCUS DAN STYLE FONTNYA SESUAI KEINGINAN**

ALURNYA GINI :

BUAT COMPONENT UTAMA ISINYA USE EFFECT DAN TABVIEW->

TAMBAHKAN PROPERTY **RENDERTABBAR** PADA TABVIEW->

BUAT FUNGSI RENDERTABBAR -> MASUKKAN TAG TABBAR -> STYLE TAG TABBAR TSB DAN RETURN TEXT

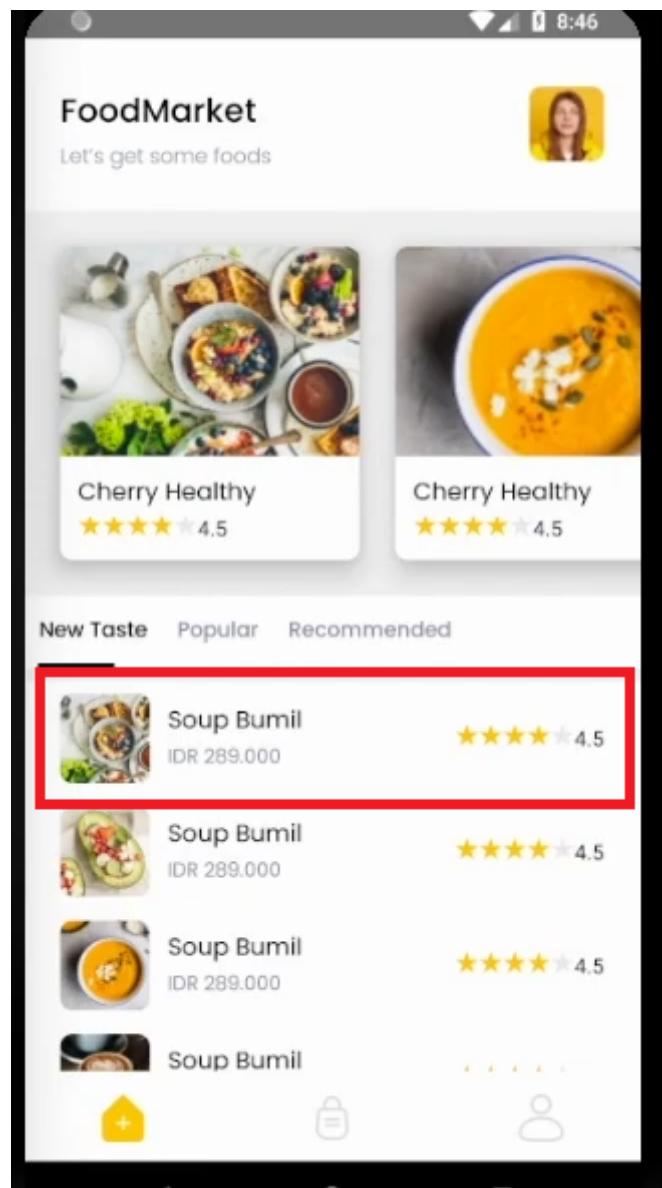
## 19 Home Page Food List Item Section

---

ini intinya sama kaya slicing biasa , tapi di video ini di **restruktur** sehingga **Home Component** nya menjadi **Simple** , kaya tag or video sebelumnya aja , kaya slicing2 biasa, bisa kok lo ,

## 20 Food Detail

---



kita ingin membuat yang page untuk yang dimearhin itu, (Food Detail).



### Cherry Healthy

★★★★★ 4.5

- 14 +

Makanan khas Bandung yang cukup sering dipesan oleh anak muda dengan pola makan yang cukup tinggi dengan mengutamakan diet yang sehat dan teratur.

#### Ingredients:

Seledri, telur, blueberry, madu.

Total Price:

IDR 12.289.000

Order Now

menjadi seperti ini kira kira tampilan akhirnya, gimana caranya ?

Logic .

1. Buat Komponen RNFES biasa
2. Daftarkan di dalam stack navigator, bukan page screen . karena itu tidak ke dalam bottom navigator.  
(navigation.navigate('something')) hanya bisa digunakan pada yang levelnya page (home, food, profile) food detail tidak termasuk ke dalam page.

contoh,

biasanya kan kita buat fungsi agar bisa navigation kan gini ya :

```

const Something={({navigation}) => {
    return (
        <View>
            <Button onPress={() => navigation.navigate('halaman_home')}>
        </View>
    );
}

```

**INI HANYA BISA JIKA HALAMAN HOME ITU BERUPA PAGE! BUKAN KOMPONEN (BUKAN FOOD DETAIL)**

CARANYA UNTUK FOOD DETAIL KAYA GINI :

```

import {useNavigation} from '@react-navigation/native';
const Something=() => {                                         //LINE INI ILANGIN
NAV.. nya
    const navigation = useNavigation();
    return (
        <View>
            <Button onPress={() => navigation.navigate('halaman_home')}>
        </View>
    );
}

```

3. Setelah buat RNFES, Masukkan Image nya , **Di sini tidak memakai komponen <Image> biasa, karena di dalam image ada komponen BUTTON** jadi harus pakai **<ImageBackground />**

```

<View>
    <ImageBackground source = {FoodDummy6} style={styles.something}>
</View>

```

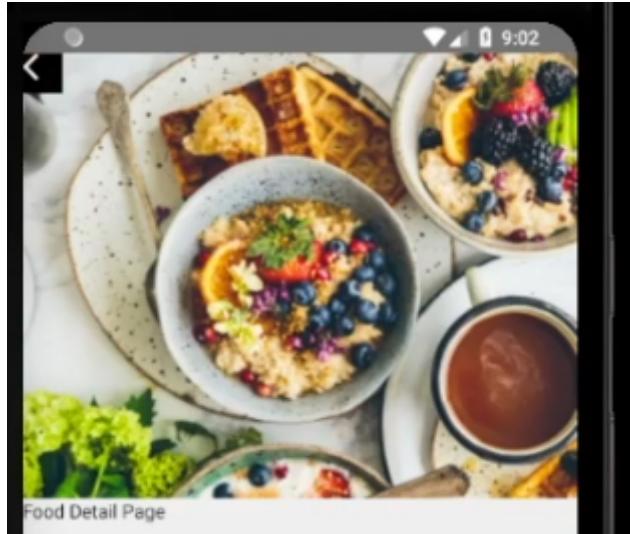
4. masukkan icon back icon ke dalam touchable opacity , dan masukkan touchable opacity tersebut ke dalam view utama .

```

<TouchableOpacity>
    <IconBackwhite/>
<TouchableOpacity>

```

```
<view>
  <ImageBackground source = {FoodDummy6} style={styles.something}/>
  <TouchableOpacity>
    <IconBackwhite/>
  </TouchableOpacity>
</view>
```



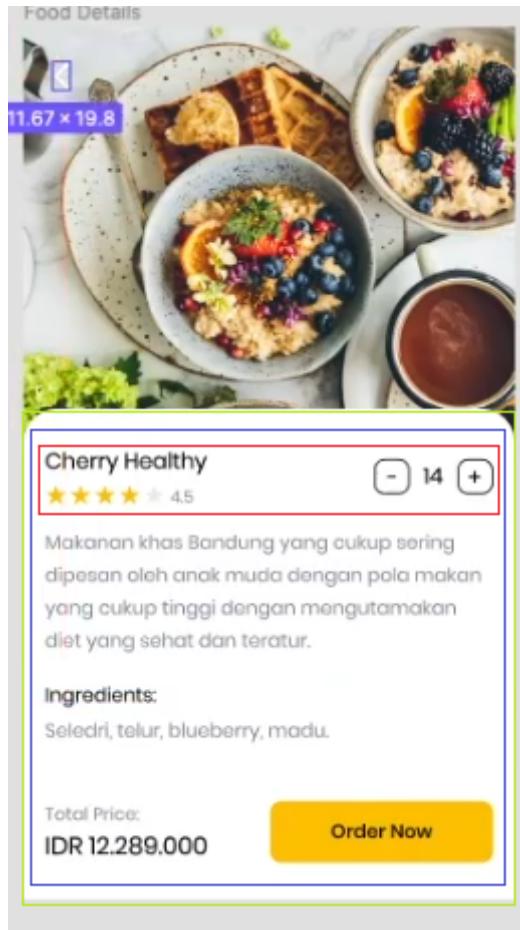
icon backnya kan mojok ke atas kanan gitu ya, supaya rapih, kita buat styling , cara membuat bisa berubah geser sedikit, kita kasih padding di dalam styles imagebackground nya ,

contoh :

```
const styles = StyleSheet.create({
  cover: {height: 330, paddingTop: 26, paddingLeft: 22},
})
```

5. Style icon back nya , panjang 30 tinggi 30 dan di tengah2 item2 itu.

```
back: {
  width: 30,
  height: 30,
  justifyContent: 'center',
  alignItems: 'center',
},
```

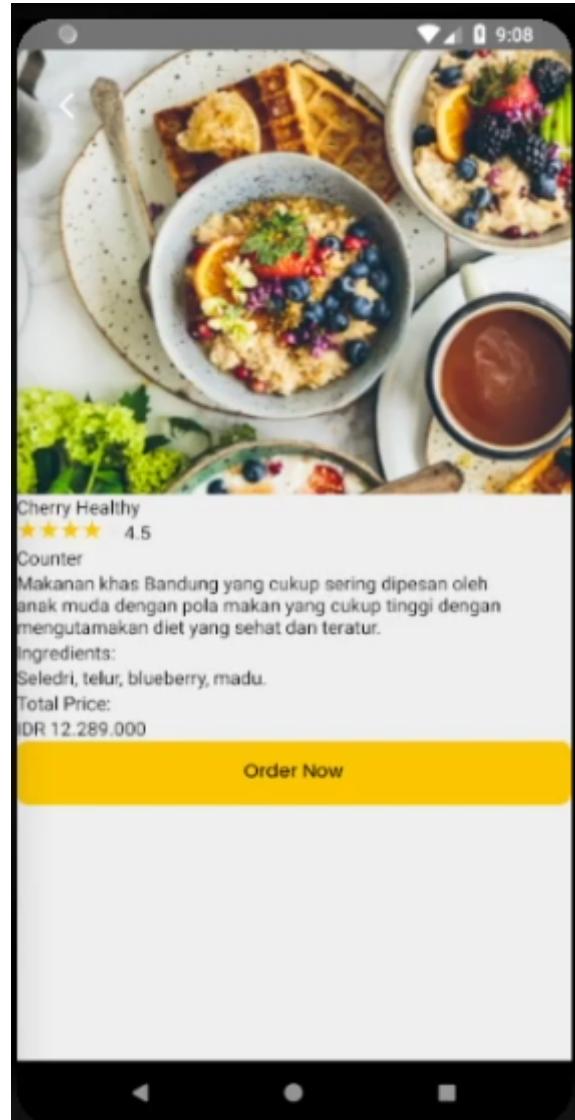


kita ingin buat yang merah2 itu , caranya , kita buat view (utama, semua putih2 - line ijo) lalu di dalam view itu , ada view yang bungkusan biru untuk semua componennya itu , lalu di dalam view biru biru itu ada view utnuk ngebungkus yang merah2 itu,

jadi jatuhnya kaya gini :

```
<view>
    <view>
        <view>
            <Text>Cherry Healthy</Text>
            <Rating/>
            <Counter/>
        <view>
            <Text>Makanan Khas bandung ....Makanan Khas bandung ...Makanan Khas
            bandung             <Text/>
            <Text>Ingredients : blablabla</Text>
            <view>
                <view>
                    <Text>Total Price: </Text>
                    <Text>Rp12.000.000</Text>
                </view>
                <view>
                    <Button>
                    </View>
                </View>
            <View>
        <View>
```

jika sudah semua, tinggal styling , nanti jadi kaya gini :



## Membuat style pada page di atas :

1. Konten keseluruhan (yang putih2 itu).
  - a. Naikkan putih2nya ke atas, berarti **pakai margin top -40 jadi naik ke atas** dan kiri kanan kita pakaikan lengkung (Border Radius)

```
const styles = StyleSheet.create({
  content: {
    marginTop : -40
    borderTopLeftRadius : 40,
    borderTopRightRadius : 40,
  }
})
```

- b. Kita kasih jarak antara tulisan ke samping dan ke atas, (padding top dan padding horizontal)

```

const styles = StyleSheet.create({
  content: {
    marginTop : -40
    borderTopLeftRadius : 40,
    borderTopRightRadius : 40,
    paddingTop : 26,
    paddingHorizontal : 16,
  }
})

```

c. Gambar di atas kan ga full ke bawah ya, nah kita kasih **flex 1** sehingga jadi full ke bawah , tetapi , ga bisa gitu , karena ptuih2 itu ada di dalam view Page, sehingga view page nya juga harus kita beri flex 1.

```

const styles = StyleSheet.create({
  page : {flex: 1},
  content: {
    marginTop : -40
    borderTopLeftRadius : 40,
    borderTopRightRadius : 40,
    paddingTop : 26,
    paddingHorizontal : 16,
    flex: 1
  }
})

```

d. Setting title (nama makanan sesuai design )

```

const styles = StyleSheet.create({
  page : {flex: 1},
  content: {
    marginTop : -40
    borderTopLeftRadius : 40,
    borderTopRightRadius : 40,
    paddingTop : 26,
    paddingHorizontal : 16,
    flex: 1
  }
  title: {
    fontSize: 16, fontFamily: 'Poppins-Regular', color: '#020202'
  }
})

```

e. Setting product counter , gimana caranya ?

logic :

1. Buat ke samping pakai apa ? **flexDirection Row!**
2. buat jadiin ke paling kanan pakai apa ? **justifyContent space-between!**
3. buat supaya rata gimana ? **alignItems center!**

**INGAT! LOGIC IS IMPORTANT DARIPADA NIRU!**

```

const styles = StyleSheet.create({
  page : {flex: 1},
  content: {

```

```
        marginTop : -40
        borderTopLeftRadius : 40,
        borderTopRightRadius : 40,
        paddingTop : 26,
        paddingHorizontal : 16,
        flex: 1
    }
    title: {
        fontSize: 16, fontFamily: 'Poppins-Regular', color: '#020202'
    },
    productContainer: {
        flexDirection: 'row',
        justifyContent: 'space-between',
        alignItems: 'center',
        marginBottom: 14,
    },
)
}
```

F. Setting text untuk description dan ingredients, (IKUTIN DESIGN AJA):

### 1. Berapa Fontnya

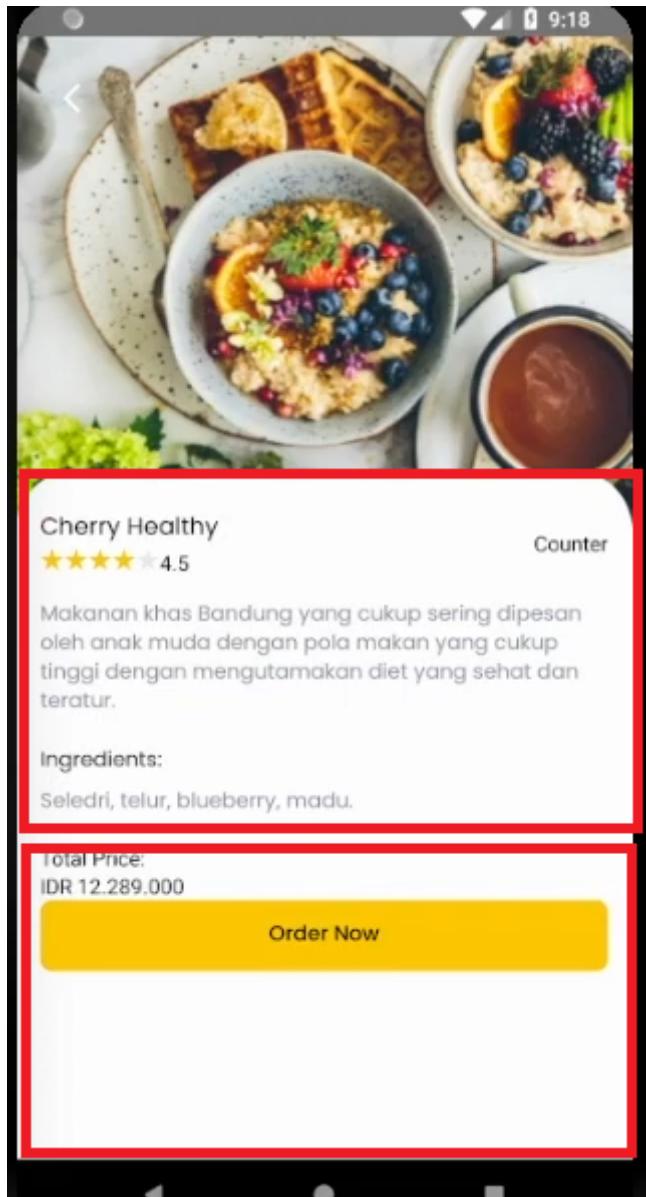
2. \*\*Apa Fontnya\*\*
3. \*\*Apa Colornya\*\*
4. \*\*Berapa margin nya\*\*

```
label: {
    fontSize: 14,
    fontFamily: 'Poppins-Regular',
    color: '#020202',
    marginBottom: 4,
},
desc: {
    fontSize: 14,
    fontFamily: 'Poppins-Regular',
    color: '#8D92A3',
    marginBottom: 16,
},
}s
```



jadi kan nih udah kaya gini, lalu gimana caranya Totalpricedan button bisa kebawah ?

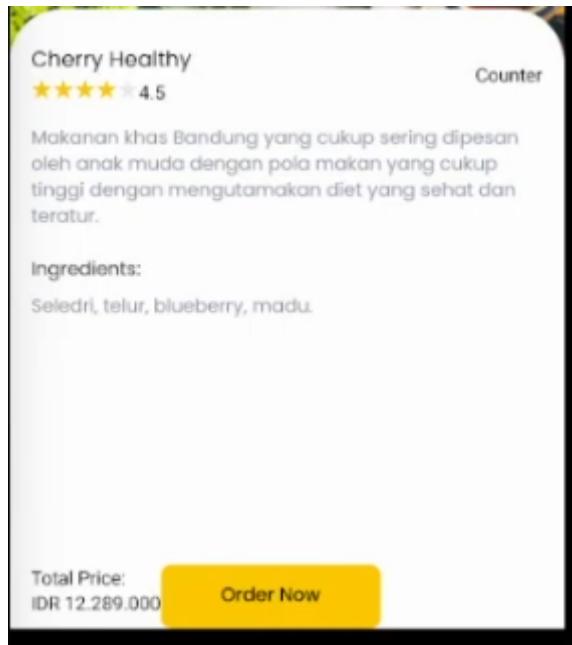
pikir coba ,



1. kita buat 2 view, lalu kita masukkin yang atas ke dalam view stylenya **mainContent**
2. bawahnya kita masukin ke dalam view kita namain **footer**
3. udah gini kan, lalu kita buat yang atas **flexnya menjadi 1!**

KALAU UDAH KITA BUAT VIEW FOOTERNYA MENJADI ROW SEHINGGA BUTTONNYA KE SAMPING

*footer: {flexDirection: 'row'}*



Kalau udah jadi kaya gini, kita buat view yang di dalam Total price menjadi **Flex 1** supaya **LEBAR TOTAL PRICE MEMENUHI KE SAMPING KANAN** (KONSEP FLEX 1)



kan udah jadi kaya gini nih , kan jelek ya , lalu kita kasih padding vertikal supaya bagus dan align item center,

```
footer: {flexDirection: 'row', paddingVertical: 16, alignItems: 'center'},  
priceContainer: {flex: 1},  
  
//ini keseluruhannya kaya gini
```

TERAKHIR, KITA SESUAIN FONT TOTAL PRICE NYA DAN ANGKANYA ITU SESUAI DESIGN SESUAI 4 KONSEP .

**FONT , SIZE, COLOR, MARGIN PADA TIAP TIAP TULISAN**

## 20.1 Membuat component Counter



1. RNFES
2. Import dan export icon - dan + tsb ke dalam SVG .
3. masukkan ke dalam return

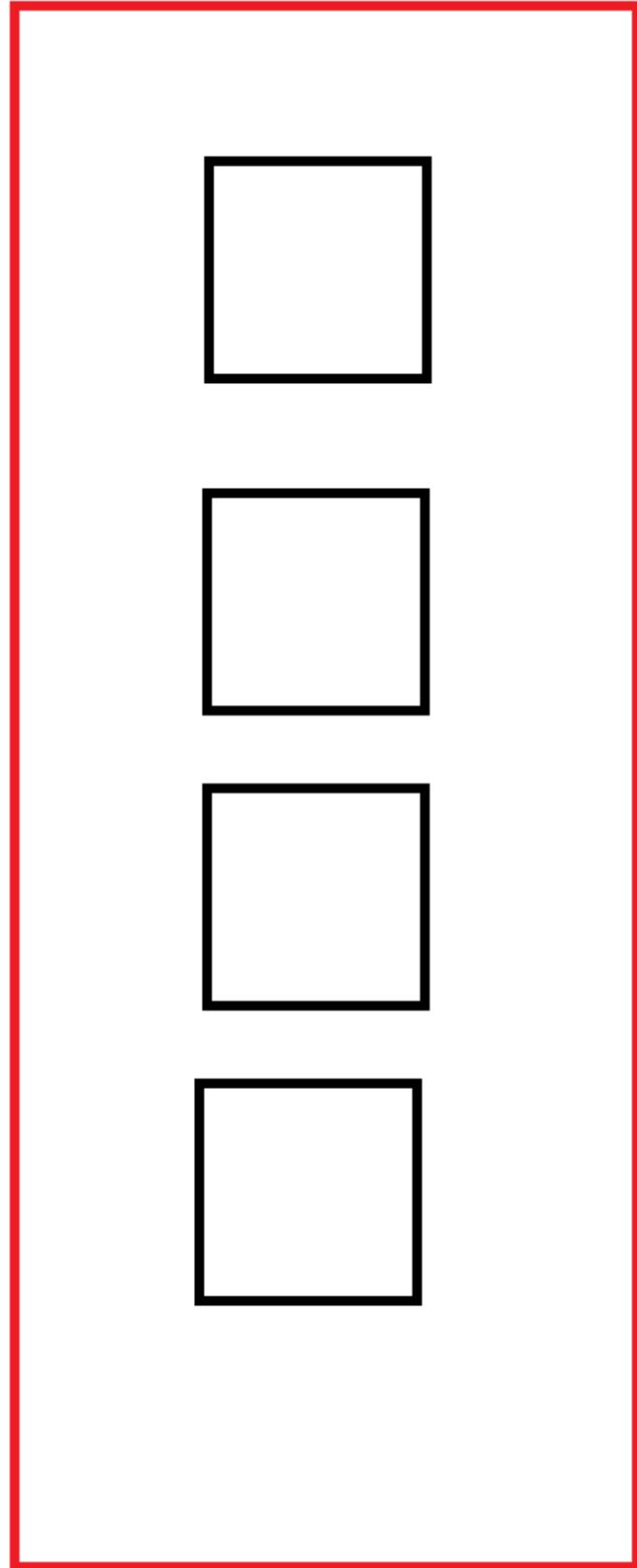
```
<view>
  <IconMin/>
  <Text><Text/>
  <IconPlus/>
</view>
```

ini kan nanti jadinya kaya gini ya,



gimana caranya supaya bisa kiri kanan ? **PAKAI FLEXDIRECTION ROW**

CONTOH KONSEP :



LU PUNYA KOMPONEN ITEM ITEMINI KE BAWAH, SUPAYA KE SAMPING, KOMPONEN MERAH2  
(VIEW) LU BUAT FLEXDIRECTIONNYA MENJADI ROW ! **AGAR KE SAMPING**

Setelah udah jadi kaya gini, kita styling text nya sesuai desgin , menggunakan 4 KONSEP FONT,

#### **FONT, SIZE, COLOR, MARGIN**

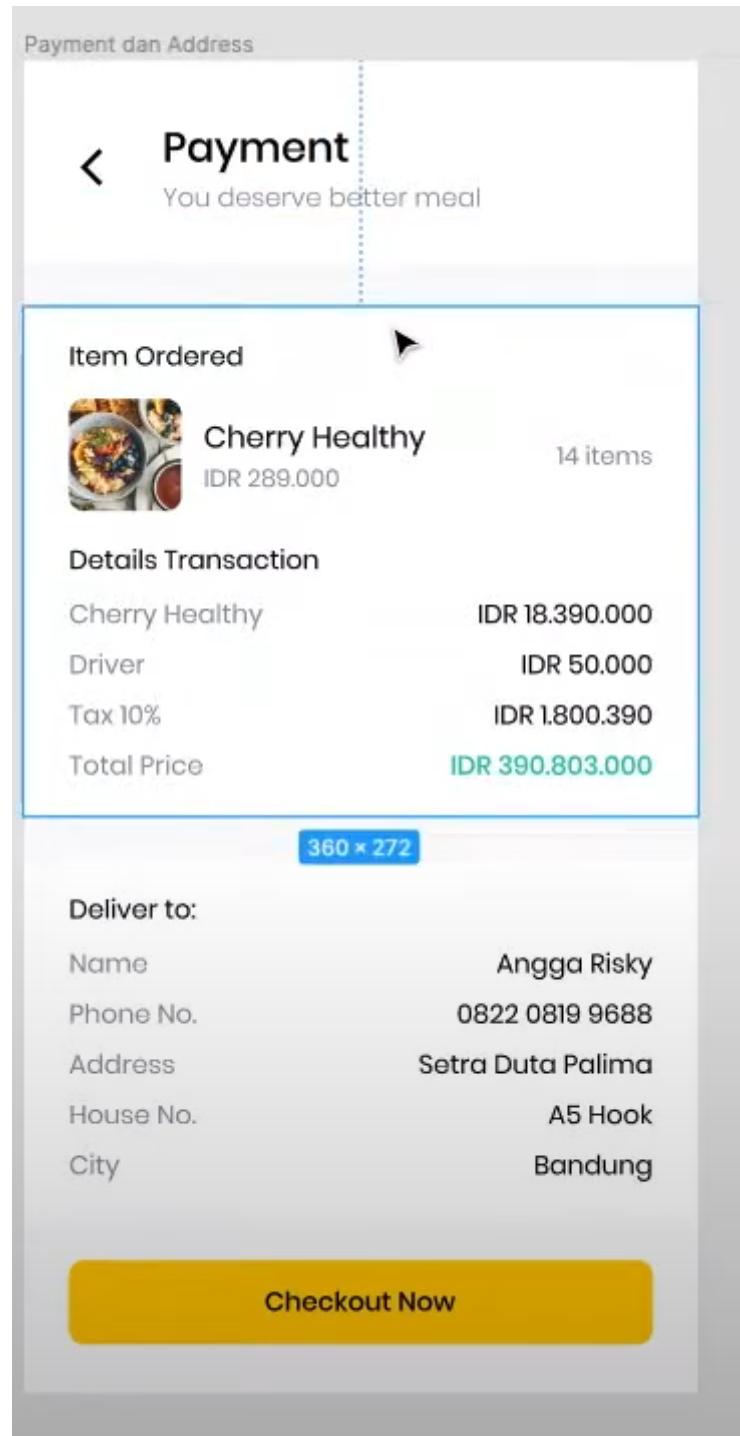
```
<Text style={styles.value}>{value}</Text>

const styles = StyleSheet.create({
  container: {flexDirection: 'row', alignItems: 'center'},
  value: {
    fontSize: 16,
    fontFamily: 'Poppins-Regular',
    color: '#020202',
    marginHorizontal: 10,
  },
});
```

5. Buat icon - dan + nya menjadi clickable dengan dibungkus < **TouchableOpacity** />

## **VIDEO 21 SLICING ORDER SUMMARY PAGE**

---



langkah langkah :

1. Buat Header / Import Headernya

```
<view>
  <Header
    title="Payment"
    subTitle="You deserve better meal"
    onBack={() => setIsPaymentOpen(false)}
  />
</view>
```

2. buat yang dibiru2in itu (order detail)

kalau kita lihat di bagian home, itu kan gambarnya sama tuh kaya gambar di home ,

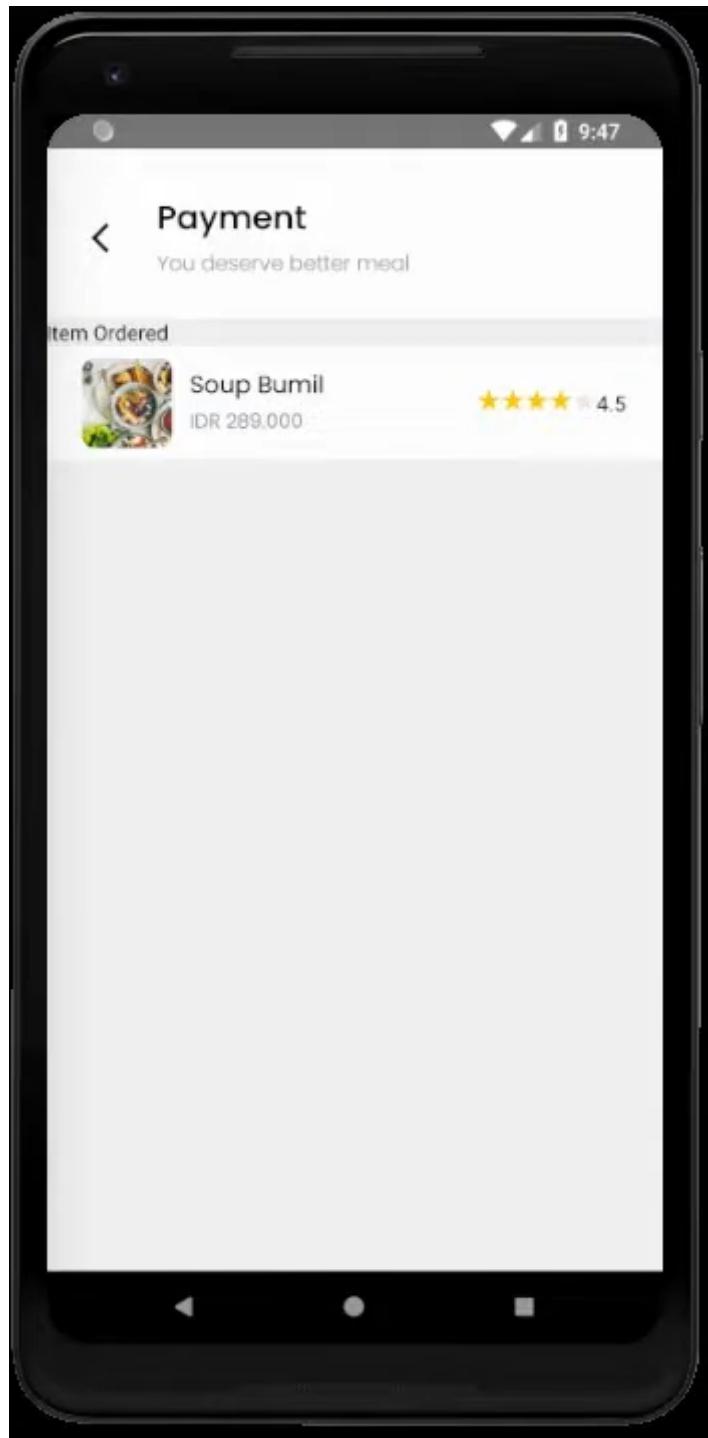
The screenshot shows a mobile application interface for a food delivery service. On the left, there's a list of food items under the heading 'New Taste'. The first item, 'Soup Bumil', is highlighted with a red box. It includes a small thumbnail image, the name, a star rating of 4.1, and the price IDR 289.000. Below it are 'Chicken' (IDR 4.509.000, 4.7 rating) and 'Shrimp' (IDR 999.000, 3.2 rating). At the bottom of this list are three icons: a house with a plus sign, a delivery bag, and a user profile. On the right, a red box highlights the 'Item Ordered' section. It shows an order for 'Cherry Healthy' from 'Cherry Healthy' at IDR 289.000. A notification bubble indicates '14 items'. Below this is a 'Details Transaction' table with the following data:

Cherry Healthy	IDR 18.390.000
Driver	IDR 50.000
Tax 10%	IDR 1.800.390
Total Price	<b>IDR 390.803.000</b>

berarti kita tinggal pakai komponen tersebut ke component ini, TAPI KITA KASIH LOGIC,

```
< View >  
  
    < Text > Item Ordered< /Text >  
  
    < ItemListFood  image={foodDummy}>  
  
</ View >
```

kalau kita lihat, nanti muncul nya itu seperti ini ,



tapi bagaimana kita ingin munculnya itu tidak seperti ini ? (kalau di page ini tidak ada rating) caranya kita tambahkan props di component < ItemListFood image={foodDummy}> nya , coba kita masuk ke component item List Foodnya , kodingannya kan kaya gini ya ,

```

import React from 'react';
import {Image, StyleSheet, Text, TouchableOpacity, View} from 'react-native'
import Rating from '../Rating';

const ItemListFood = ({image, onPress}) => {
  return (
    <TouchableOpacity activeOpacity={0.7} onPress={onPress}>
      <View style={styles.container}>
        <Image source={image} style={styles.image} />
        <View style={styles.content}>
          <Text style={styles.title}>Soup Bumil</Text>
          <Text style={styles.price}>IDR 289.000</Text>
        </View>
        <Rating />
      </View>
    </TouchableOpacity>
  );
}

```

kita ubah aja jadi ratingnya dengan text . EITSSS TIDAK BISA! ntar di halaman home, yang ada ratingnya hilang dong ? berarti kita buat logic nih di sini ,

kita buat jadi kaya gini ,

```

const ItemListFood = ({image, onPress, items, rating}) => {
  return (
    <TouchableOpacity activeOpacity={0.7} onPress={onPress}>
      <View style={styles.container}>
        <Image source={image} style={styles.image} />
        <View style={styles.content}>
          <Text style={styles.title}>Soup Bumil</Text>
          <Text style={styles.price}>IDR 289.000</Text>
        </View>
        {items && !rating && <Text>{items} items</Text>}
        {[rating && !items && <Rating />]}
      </View>
    </TouchableOpacity>
  );
}

```

naah, maksudnya gini,

di kodingan kita di awal kan gini nih <ItemListFood image={foodDummy}> , misalnya kita kirim props jadi kaya gini <ItemListFood image={foodDummy} items={14}> yaaa keluarnya items = 14, tapi kalau kita kirim <ItemListFood image={foodDummy} rating={4}> yaaa, yang muncul rating.

Payment dan Address

## Payment

You deserve better meal

Item Ordered

 **Cherry Healthy** IDR 289.000 14 items

Details Transaction	
Cherry Healthy	IDR 18.390.000
Driver	IDR 50.000
Tax 10%	IDR 1.800.390
Total Price	<b>IDR 390.803.000</b>

360 x 272

Deliver to:

Name	Angga Risky
Phone No.	0822 0819 9688
Address	Setra Duta Palima
House No.	A5 Hook
City	Bandung

Checkout Now

**kita lihat ini  
strukturnya sama**

ini kan sama nih ya, jadi kita buat komponen tersendiri, sehingga kita tidak perlu buat dua kali dalam 1 page,

kira kira nanti jadinya kaya gini :

```

<View>
    <Text>Item Ordered</Text>
    <ItemListFood image={FoodDummy1} items={14} />
    <Text>Details Transaction</Text>
    <ItemValue />
    <ItemValue />
    <ItemValue />
    <ItemValue />
</View>

<View>
    <Text>Deliver to:</Text>
    <ItemValue />
    <ItemValue />
    <ItemValue />
    <ItemValue />
    <ItemValue />
</View>
<View>
    <Button text="Checkout Now" />
</View>

```

mari kita edit Komonent < ItemValue > tersebut.

1. Kita Style dulu di halaman page nya padding ,

```

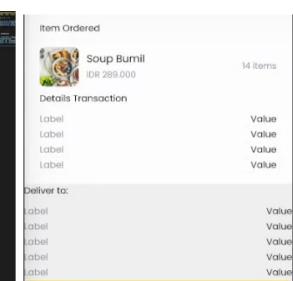
import React from 'react';
import {StyleSheet, Text, View} from 'react-native';

const ItemValue = () => {
  return (
    <View style={styles.container}>
      <Text style={styles.label}>Label</Text>
      <Text style={styles.value}>Value</Text>
    </View>
  );
};

export default ItemValue;

const styles = StyleSheet.create({
  container: {flexDirection: 'row', justifyContent: 'space-between'},
  label: {fontSize: 14, fontFamily: 'Poppins-Regular', color: '#8D92A3'},
  value: {fontSize: 14, fontFamily: 'Poppins-Regular', color: '#020202'},
});

```



kira kira nanti item dan valuenya kaya gini , kita perlu masukkan props di dalam component itemView ini, sehingga tulisan label dan valuenya bisa lebih dinamis,

contoh, jadinya kaya gni :

```
const ItemValue = ({label, value}) => {
  return [
    <View style={styles.container}>
      <Text style={styles.label}>{label}</Text>
      <Text style={styles.value}>{value}</Text>
    </View>
  ];
};
```

sehingga nanti di Component induk OrderSummary, kita bisa masukkan :

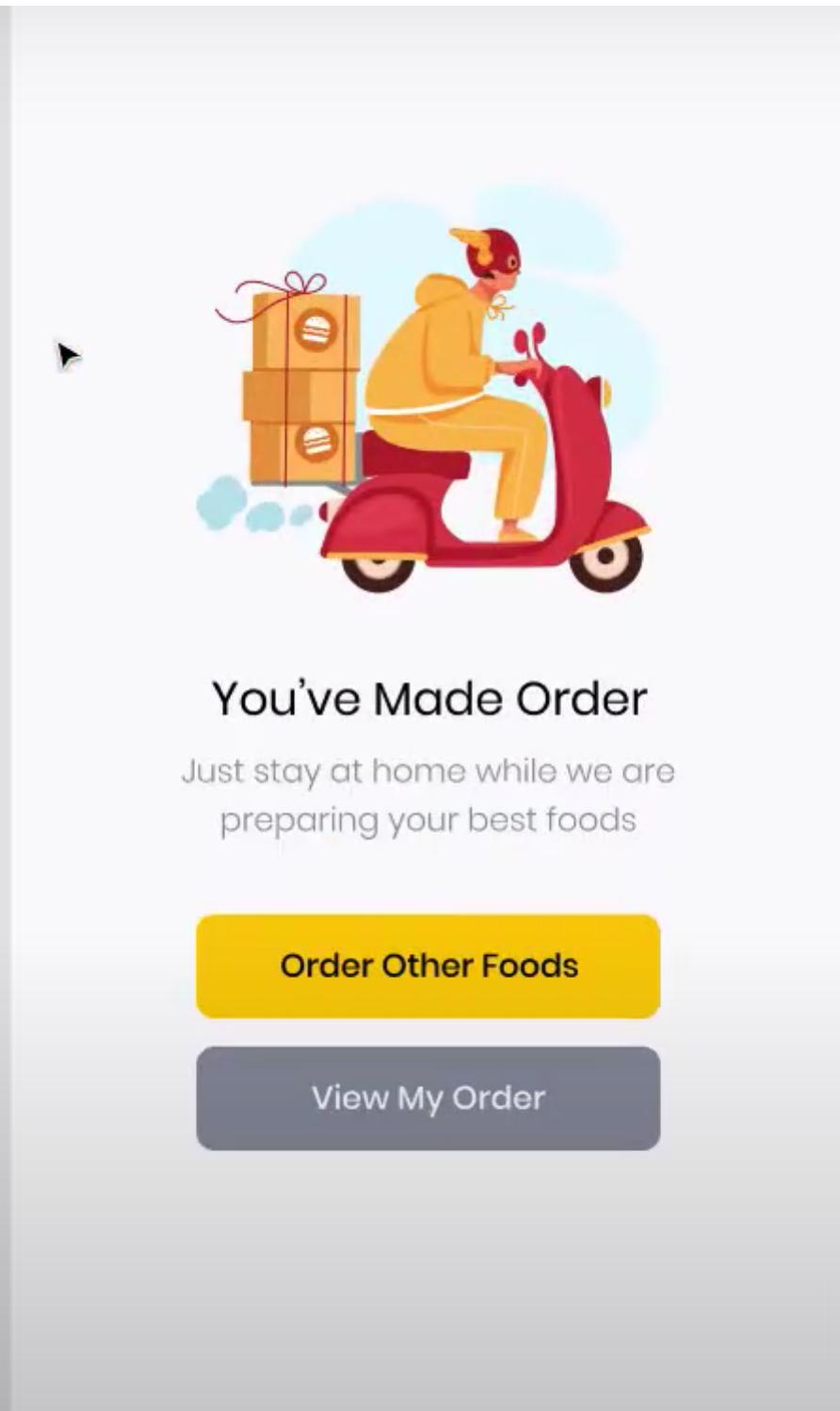
```
< ItemValue label ="Ayam Bakar" value="IDR 40000" >
```

gitu . kalau udah semua, tinggal styling sesuai yang diinginkan oleh tim UI/UX

## Video 22 : SLICING SUCCESS ORDER

---

kita ingin membuat seperti ini :



## You've Made Order

Just stay at home while we are  
preparing your best foods

**Order Other Foods**

**View My Order**

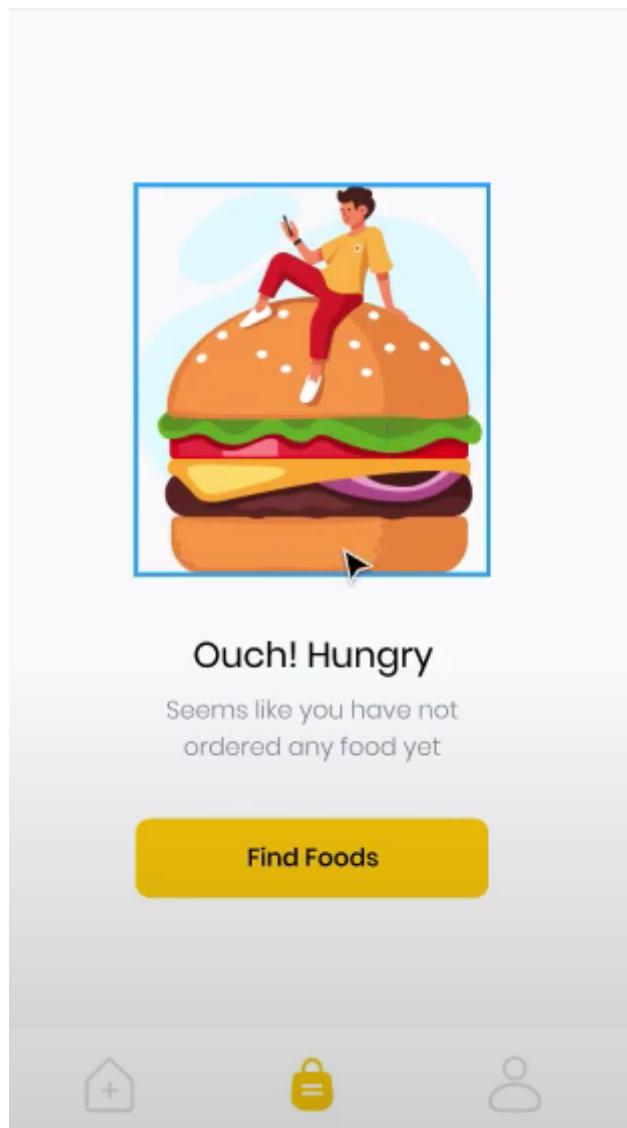
1. Buat folder baru namanya SuccessOrder, buat index.js nya, export, import route Stack.Screennya,
2. RNFES
3. Import Ilustrasinya ke dalam komponen, dna ini sebenarnya caranya sampa persis kaya SuccessSignUp, tinggal kita kopas aja kodingannya dan stylingnya, lalu kita ubah import dan imagennya.

di atas gambar tersebut kan ada dua button tuh ya, satunya gampang , tinggal  
navigation.replace('MainApp'), TAPI YANG SATUNYA BEDA, ga bisa ke navigation.replace('Order'),  
karena order page itu ada di main app, tetapi di beda screen,

jadi kalau mau ke navigation.replace('Order') harus seperti ini ===>

```
navigation.replace('MainApp', {screen: 'order'})
```

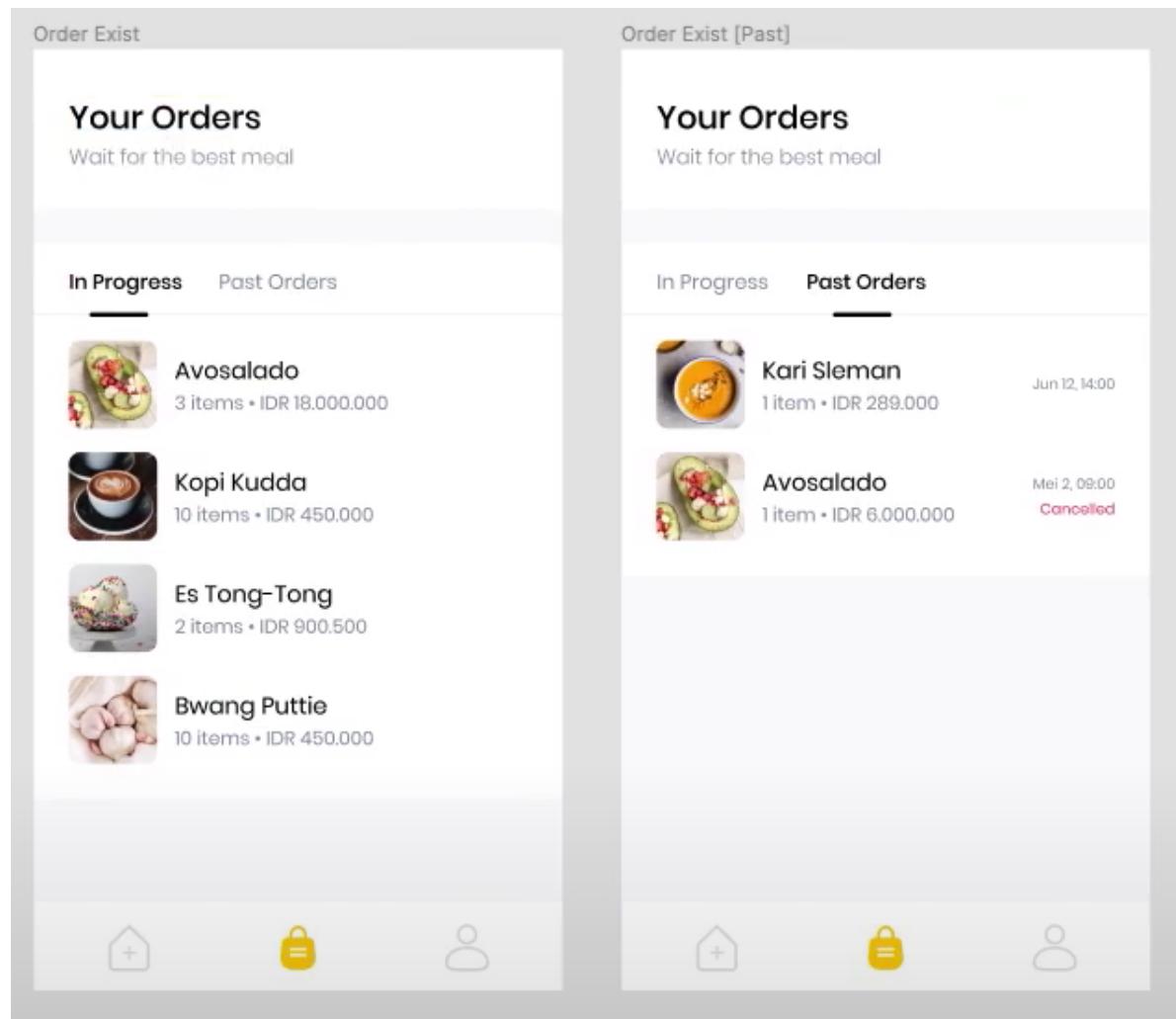
## VIDEO 23 : ORDER HISTORY (EMPTY)



Kita buat kira kira kaya gini ,

kalau empty , caranya sami mawon kaya di video 22, cuma gambar dan tulisannya aja yang berbeda.

## 24. History in Progress & Past Order



kita ingin buat kaya gini ,

1. kita buat folder dan index.js serta rnfs,
2. kan tadi di atas kita udah buat tuh ya yang empry order kaya gimana,  
sekarang kita buat, kalau misalnya ga empty kaya gini tulisannya, nah kita buat logic pagi  
**isEmpty** contoh :

```
const [isEmpty] = useState(false);

return (
  <View>
  {
    isEmpty?
      <EmptyOrder> :

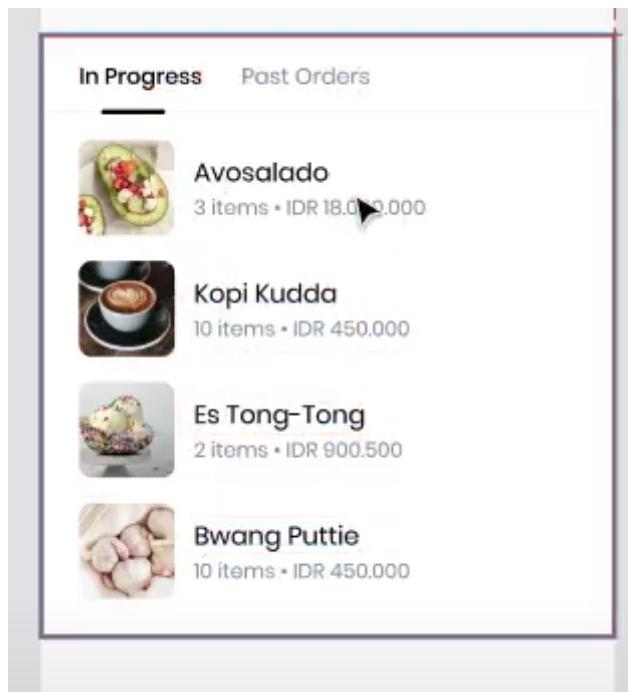
      <View>
        <View>
      </View>
    }
  </View>
);
```

kita buat default false, berarti kodingan kita di bawah empty order itu , (else nya).

3. Masukkan komponen header pada else yang view di atas itu.

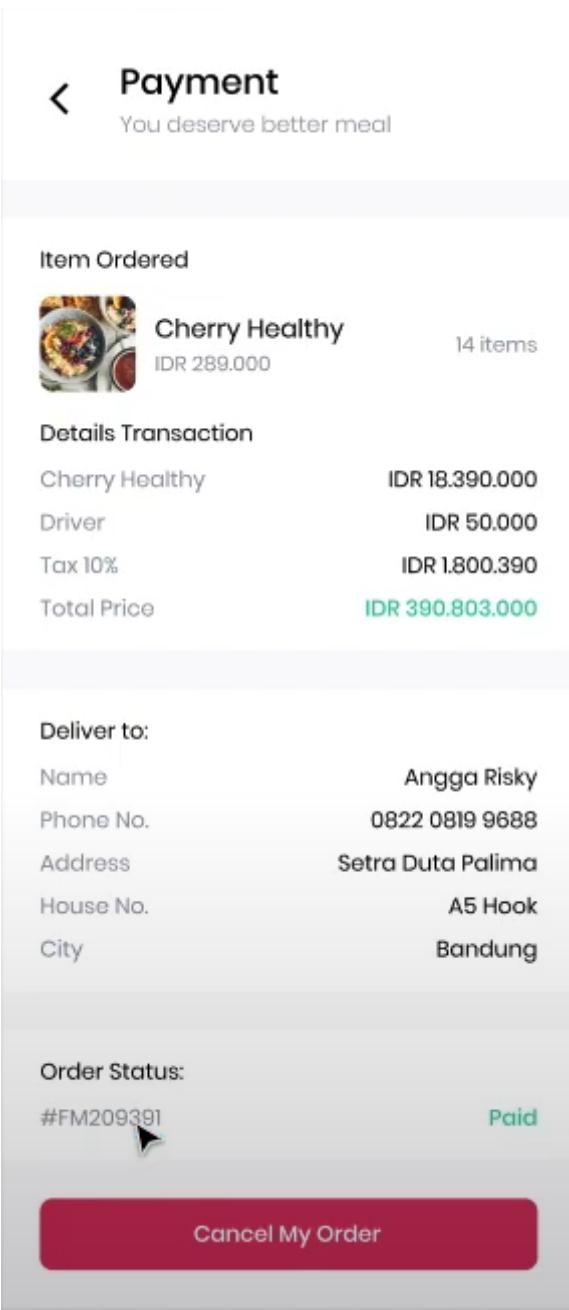
```
<View>
  <Header title="Your Orders" subTitle="Wait for the best meal" />
</View>
```

4. Kita lihat gambar di bawah ini :



ini itu struktur nya sama kaya di hometabsection, jadi kita ikutin aja kodingan dari hometabsection.

## VIDEO 25 : ORDER DETAIL CANCEL ORDER



Coba deh perhatiin baik baik, kita lihat kan di sini, kalau order detail itu sama kaya order summary, jadi kita copy paste aja kodingan ordersummary ke sini dan kita edit edit ,

1. Copy paste dan edit dari ordersummary ke order detail,
2. styling copy, import copy,
3. kita copy kodingan berikut, sehingga ada order statusnya dan sesuai dengan yang diinginkan orderdetail.

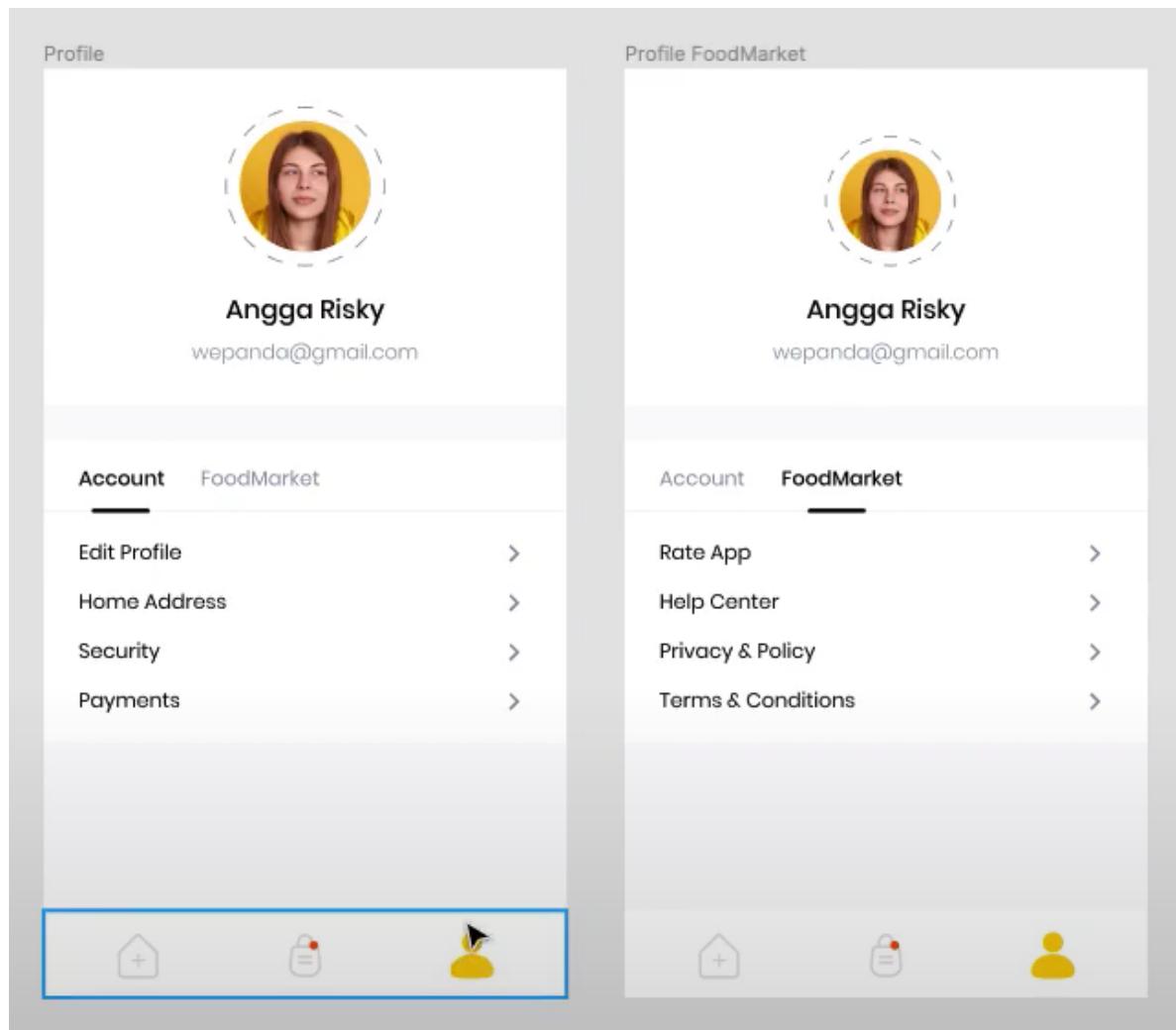
```
<View style={styles.content}>
  <Text style={styles.label}>Order Status:</Text>
  <ItemValue label="#FM209391" value="Paid" />
</View>
```

4. ganti buttonnya jadi kaya gini :

```
<View style={styles.button}>
  <Button
    text="Cancel My Order"
    onPress={() => navigation.replace('SuccessOrder')}
    color="#D9435E"
    textColor="white"
  />
</View>
```

sehingga buttonnya jadi merah.

## VIDEO 26 : SLICING PROFILE PAGE

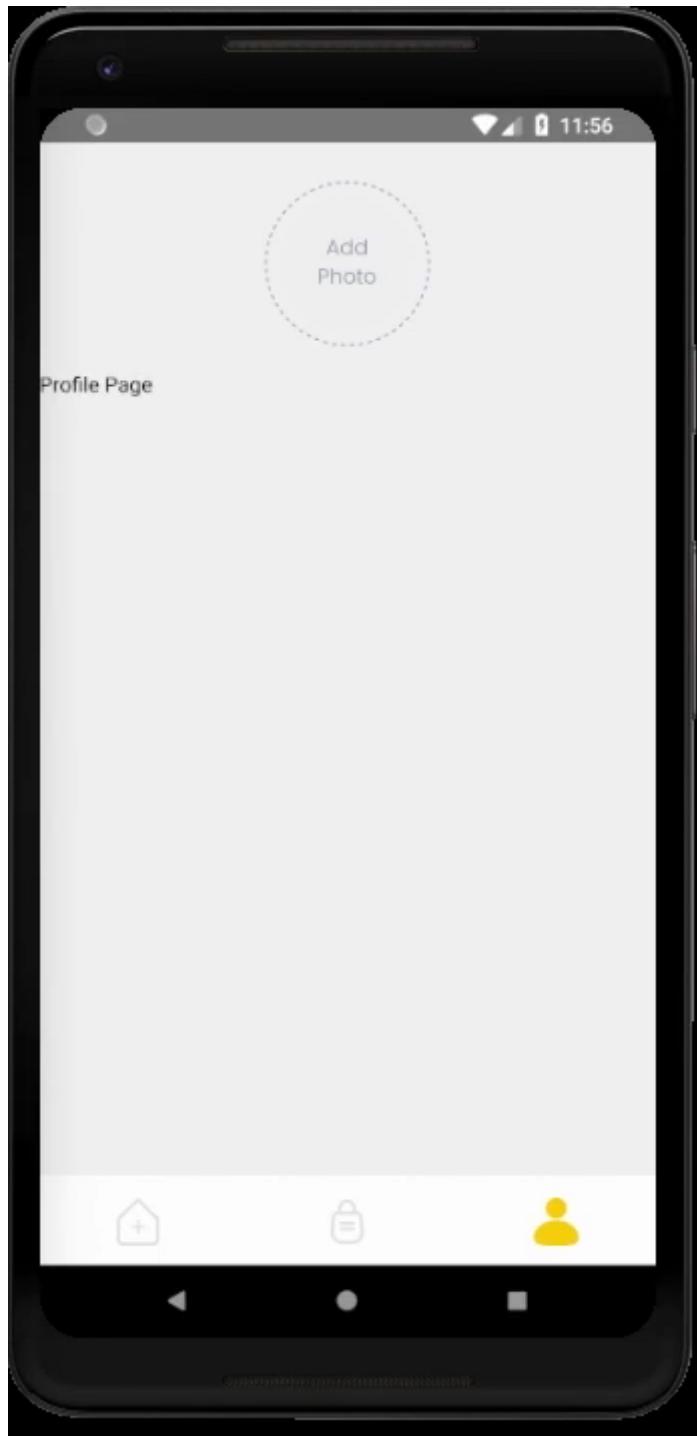


1. Buat komponennya (folder dan index.js nya)
2. ini kan sama yah kaya di proses sign up yang gambar2 fotonya itu, jadi kita bisa copy component dan stylingnya aja, jadi kira kira kodingannya jadinya kaya gini :

```
const Profile = () => {
  return (
    <View>
      <View style={styles.photo}>
        <View style={styles.borderPhoto}>
          <View style={styles.photoContainer}>
            <Text style={styles.addPhoto}>Add Photo</Text>
          </View>
        </View>
      <Text>Profile Page</Text>
    </View>
  );
};
```

```
const styles = StyleSheet.create([
  photo: {alignItems: 'center', marginTop: 26, marginBottom: 16},
  borderPhoto: {
    borderWidth: 1,
    borderColor: '#8D92A3',
    width: 110,
    height: 110,
    borderRadius: 110,
    borderStyle: 'dashed',
    justifyContent: 'center',
    alignItems: 'center',
  },
  photoContainer: {
    width: 90,
    height: 90,
    borderRadius: 90,
    backgroundColor: '#F0F0F0',
    padding: 24,
  },
  addPhoto: {
    fontSize: 14,
    fontFamily: 'Poppins-Light',
    color: '#8D92A3',
    textAlign: 'center',
  },
]).
```

yang atas komponentnya yang bawah stylingnya. tapi gambarnya jadinya kaya gini :

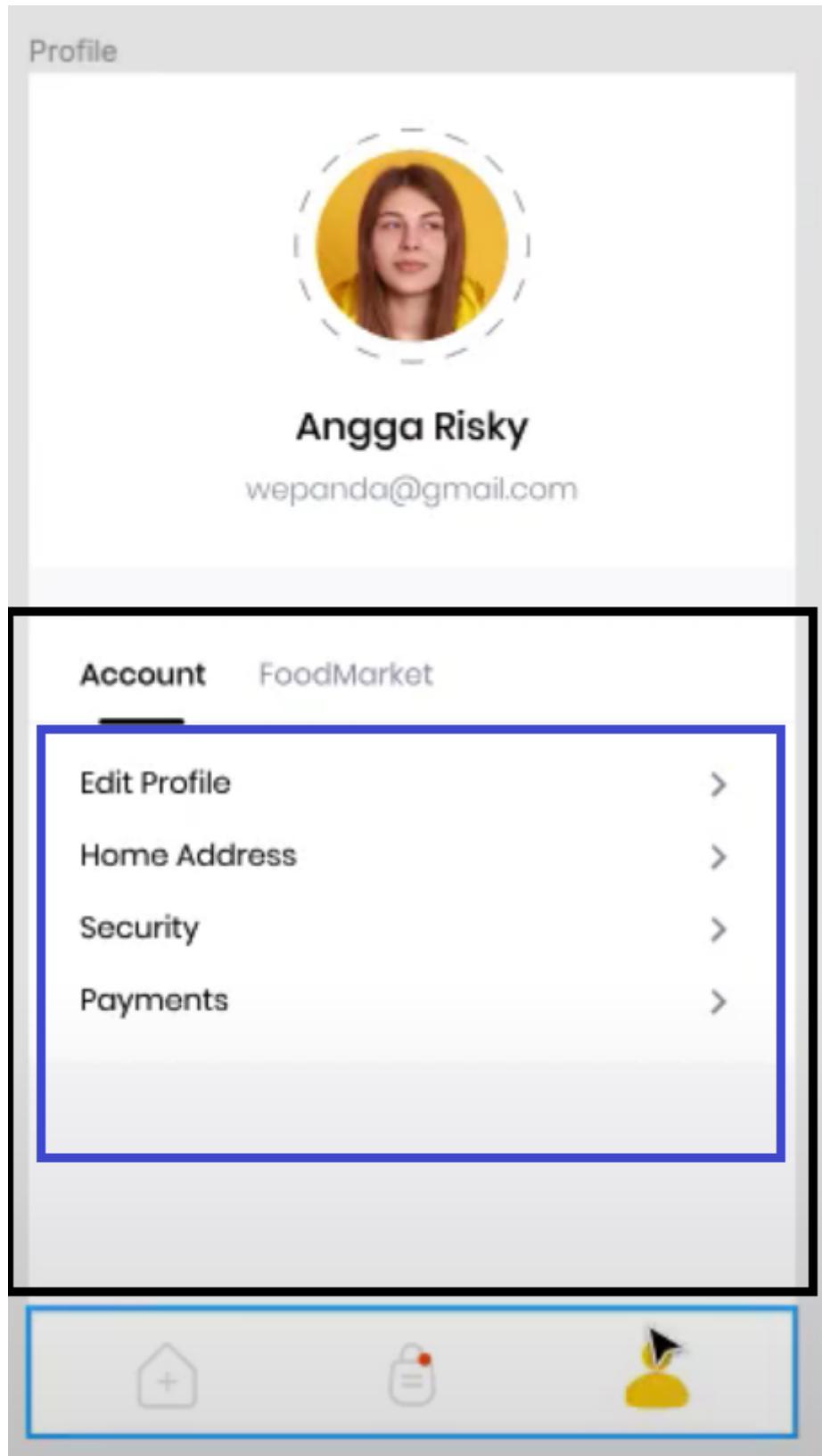


naah kalau kaya gitu berarti kita edit aja ,

jadi di dalam view style border photo , bawahnya kita ganti text AddPhotonya menjadi image ,

```
<View style={styles.borderPhoto}>
| <Image source={ProfileDummy} style={styles.photoContainer} />
|</View>
```

3. tambahkan text component isinya nama dan email dan styling text nama dan email tersebut.
4. kalau kita perhatikan lagi gambar di atas, itu kan ada tab section ya di profile, berarti kita perlu buat component baru namanya ProfileTabSection dan kita masukkan ke dalam komponen profile page ini,
5. isi nya apa ? ya kira kira tab section mirip2 lah, jd kita copy paste dari hometabsection aja, terus nanti kita edit.
- 6.

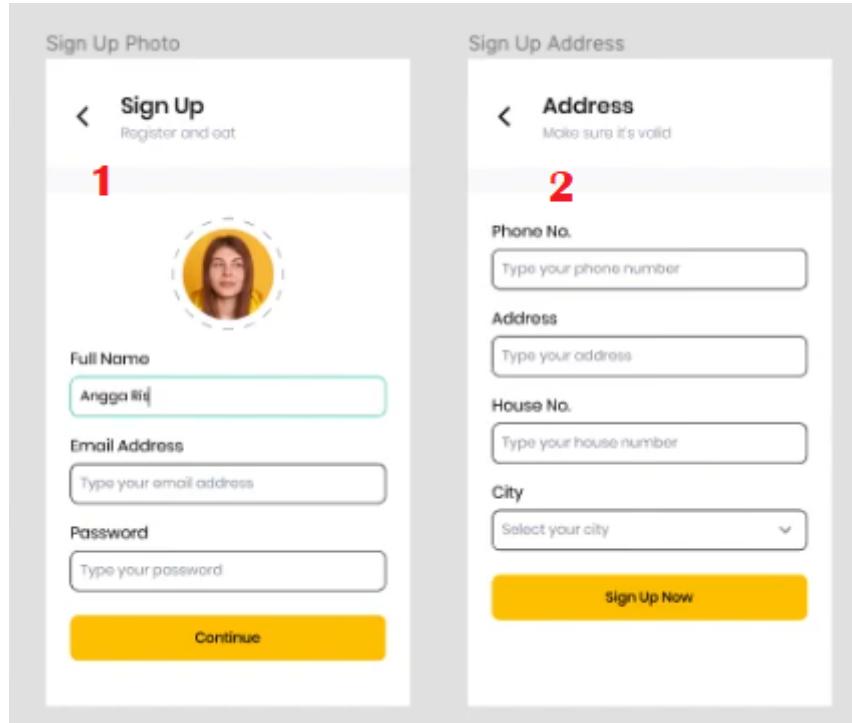


kan tadi kita udah buat tuh ya profile tab sectionnya, nah , di dalam profile tab section itu, kita buat lagi component , isinya kira2 seperti gambar yang di kotak biru di atas tua di atas, isinya cuma text dan icon > , jadi gampang lah, lu bisa buat kaya gitu,

## VIDEO 31 : INSTALLING AND SETUP REDUX

sebelum memasuki bagian ini, disarankan mampu memahami materi react-redux basic , link di bawah ini :

<https://www.youtube.com/watch?v=6x65uPMdV5A>



Kata kunci konsep pada redux :

1. **Store**
2. **Reducer**
3. **Dispatch/Action**

di gambar tersebut kan kita tau, kalau ketika kita ketik 'Continue' itu kita belum mengirimkan http post ke server, nah, di sini kita **MENGGABUNG DATA GAMBAR 1 DAN 2 KE DALAM GLOBAL STATE,**

Caranya menggunakan **REDUX REACT-REDUX**

1. `$> yarn add redux redux-thunk`

### **-MEMBUAT STORE-**

2. buat **folder redux** di dalam folder **src**, dan buat **store.js** nya

store.js :

```
import {createStore} from 'redux';

const store = createStore(reducer);
```

ini kodingan store sangat sangat basic, kalau di native pakai thunk, dan reducer nya yang dikurungin itu dapet dari mana ? dari file reducer.js (folder terpisah) intinya reducer itu sih cuma kaya fungsi biasa aja .

untuk project ini kaya gini :

```

import {applyMiddleware, createStore} from 'redux';
import thunk from 'redux-thunk';

const store = createStore(reducer, applyMiddleware(thunk));

export default store;

```

## -MEMBUAT REDUCER-

3. buat folder **reducer** dan di dalam nya ada **index.js** (untuk mengumpulkan semua reducernya , TAPI INI MENGGUNAKAN COMBINEREDUCER DARI REDUX YAA!!) dan buat reducer nya, contoh **auth.js**

cara menggabungkan semua reducer di index.js:

```

import {combineReducers} from 'redux';
import authReducer from './auth';

const reducer = combineReducers({registerReducer, globalReducer, dst..});

export default reducer;

```

auth.js:

```

const initRegister = {
    name:'',
    email:'',
    password:'',
    password_confirmation:'',
    address:'',
    city:'',
    houseNumber:'',
    phoneNumber: ''
}

export const registerReducer = (state=initRegister, action) => {
    if(action.type === 'SET_REGISTER'){
        return {
            ...state,
            name: action.value.name,
            email: action.value.email,
            password : action.value.password,
            password_confirmation : action.value.password
        }
    }
    if(action.type === 'SET_ADDRESS'){
        ...state,
        address : action.value.address,
        city: action.value.city,
        houseNumber: action.value.houseNumber,
        phoneNumber: action.value.phoneNumber
    }
    return state;
}

```

contoh lain global.js

```

const initGlobalState = {
  isError : false,
  message : 'Error'
}

export const GlobalReducer =(state=initGlobalState, action) => {
  if(action.type ==='SET_ERROR'){
    return {
      ...state,
      isError : action.value.isError,
      message : action.value.message
    }
  }
  return state;
}

```

### **-SETUP PEMANGGILAN REDUX (PROVIDER)-**

4. Pada file App.js , kurungkan < Route /> ke dalam tag < Provider > yang di import dari library 'react-redux'

```

import {Provider} from 'react-redux'
import store from './redux/store'

return (
  <NavigationContainer>
    <Provider store={store}>
      <Router>
        <Provider>
          <NavigationContainer>
    )
)

```

expl : store={store} dapet dr mana ? dari store yang telah kita buat,

*import store from './redux/store'*

## **32. IMPLEMENT REDUX , SETUP FROM ON REGISTER PAGE**

---

Register page itu kan ada 2 tahapan ya, yaitu :

1. Signup (username email pass)
2. Address

nah , pada masing masing ini , kita simpan data nya ke dalam global , yaitu menggunakan redux.

## -cara implementasinya-

1. Pada signup/index.js (signup page), masukkan const useState (setForm) yang sudah kita buat sendiri:

```
const [form, setForm] = useForm({  
    name: '',  
    email: '',  
    password: '',  
    password_confirmation: ''  
});
```

2. Masukkan useForm di atas pada masing masing Text input ,

```
<TextInput  
    label="Full Name"  
    placeholder="Type your full name"  
    value={form.name}  
    onChangeText={(value) => setForm('name', value)}  
/>
```

maksudnya di atas ini yaitu, text input awalnya memiliki value {form.name} , form.name itu apa ? ya kosong dulu , , sesuai point nomor 1, name: ",

lalu **onChangeText** itu maksudnya apa ? **Ketika diketik, (textnya berubah, maka terjadi function**, apa functionnya ? yaitu setForm , dengan mengirimkan parameter value (yang kita ketik) ke dalam variable name.

contoh : kita ketik "rifan", nah onchangetext memiliki fungsi yang mana fungsi tersebut mengirimkan parameter value untuk mengubah name, yang tadinya kosong menjadi 'rifan'.

begitu juga yang kita lakukan pada email dan password

```
<TextInput  
    label="Email Address"  
    placeholder="Type your email address"  
    value={form.email}  
    onChangeText={(value) => setForm('email', value)}  
/>  
<TextInput  
    label="Password"  
    placeholder="Type your password"  
    value={form.password}  
    onChangeText={(value) => setForm('password', value)}  
    secureTextEntry  
/>
```

3. Pada button **Continue** kita kirimkan fungsi onPress untuk menuju ke fungsi baru yang akan kita buat , yaitu onSubmit,

4. <Button text="Continue" onPress={onSubmit} />

buat fungsi onSubmit nya, lalu kita console log,

```
5. const onSubmit = () => {
    console.log('form : ', form);
    dispatch({type: 'SET_REGISTER', value: form});
};
```

```
form: index.bundle?platform=ios&shallow=true:40
  ▼ name: "sule", email: "sule@prawito.com", password: "12345678", pas
    sword_confirmation: "" } ⓘ
      email: "sule@prawito.com"
      name: "sule"
      password: "12345678"
      password_confirmation: ""
    ► __pr password ect
```

naaaahhh, berhasil kita punya form yang akan dikirimkannya.

6. pada fungsi onSubmit, kita panggil deh dispatch, untuk penyimpanan datta form di atas ke dalam global .

caranya :

```
import {useDispatch} from 'react-redux';

const dispatch = useDispatch();

const onSubmit = () => {
    dispatch({type: 'SET_REGISTER', value: form});
};
```

pada useDispatch ini, kita selal mengirimkan 2 parameter, yaitu

- TIPENYA APA
- VALUENYA APA

kita masukkan ke dalam onSubmit (ketika dipencet)

7. Untuk form Addressnya, sama seperti di atas, tp yang sedikit berbeda yaitu, ketika kita memilih OptionSelect menggunakan picker,

- Kalau input text kita menggunakan onChangeText
- Kalau OptionSelect kita menggunakan onSelectChange

```
const [form, setForm] = useForm({
    phoneNumber: '',
    address: '',
    houseNumber: '',
    city: 'Bandung',
});

<TextInput
    label="Phone No."
    placeholder="Type your phone number"
    value={form.phoneNumber}
    onChangeText={(value) => setForm('phoneNumber', value)}
/>

<TextInput
```

```

        label="Address"
        placeholder="Type your address"
        value={form.address}
        onChangeText={(value) => setForm('address', value)}
    />

    <TextInput
        label="House No."
        placeholder="Type your house number"
        value={form.houseNumber}
        onChangeText={(value) => setForm('houseNumber', value)}
    />

    <Select
        label="City"
        value={form.city}
        onSelectChange={(value) => setForm('city', value)}
    />

```

```
<Button text="Sign Up Now" onPress={onSubmit} />
```

```
const onSubmit = () => {
    console.log('form: ', form)
};
```

```

form: index.bundle?platform=web&w=true:sourcemap:39
  ▼ {phoneNumber: "123123", address: "Cipari", houseNumber: "12", city: "Jogja"} ⓘ
    address: "Cipari"
    city: "Jogja"
    houseNumber: "12"
    phoneNumber: "123123"
  ► __proto__: Object

```

di sini kita liat, sudah berhasil, lalu tinggal kita dispatch saja data tersebut , kita kirimkan ke dalam global menggunakan dispatch, **Tentukan Type nya** dan **Tentukan (kirimkan valuenya)**

## 8. DI SINI AGAK BERBEDA, KENAPA ?

di gambar di atas ini an kita tau , kalau form kita itu address, city, house and phone number. **kita gabungkan data global dengan datas di atas**, caranya menggunakan useSelector. caranya :

1. Kita panggil data global di atas (username, email ,pass)

```

import {useSelector} from 'react-redux';

const registerReducer = useSelector((state) => state.registerReducer);

```

di sini kita udah dapet semua data registerReducer (tinggal kita gabungkan aja menggunakan data form yang di gambar di atas):

caranya :

```
2. const onsubmit = () => {
  const data ={
    ... form,
    ...registerReeducer
  }
  console.log("data : " , data);
};
```

nanti hasilnya kaya gini :

```
data Register: index.bundle?platform=web:sourceMap:56
  {phoneNumber: "123123", address: "Ciparigi", houseNumber: "12", city: "Surabaya", name: "sule", ...} ⓘ
    address: "Ciparigi"
    city: "Surabaya"
    email: "sule@prawito.com"
    houseNumber: "12"
    name: "sule"
    password: "12345678"
    password_confirmation: "12345678"
    phoneNumber: "123123"
  ► __proto__: Object
>
```

## 33. Sign Up Integration to API (AXIOS)

Kan kita udah dapet tuh semua datanya yang di gambar di atas, , nah , pada fungsi onSubmit tersebut, kita kirimkan data tersebut ke endpoint post menggunakan axios,

jadinya seperti ini :

```
Axios.post('http://linknyaapa.com/api/register', data).
  .then((res)=>{
    console.log(res.data)
  })
  .catch((err) => {
    console.log(err)
  })
```