



Pemrograman Perangkat Bergerak Routing & Life Cycle

Febri Damatraseta Fairuz S.T, M.Kom

Component



Component

React Class Component

```
import { Text, View } from 'react-native'  
import React, { Component } from 'react'  
  
export class Hello extends Component {  
  render() {  
    return (  
      <View>  
        <Text>Hello</Text>  
      </View>  
    )  
  }  
}  
  
export default Hello
```

React Function Component

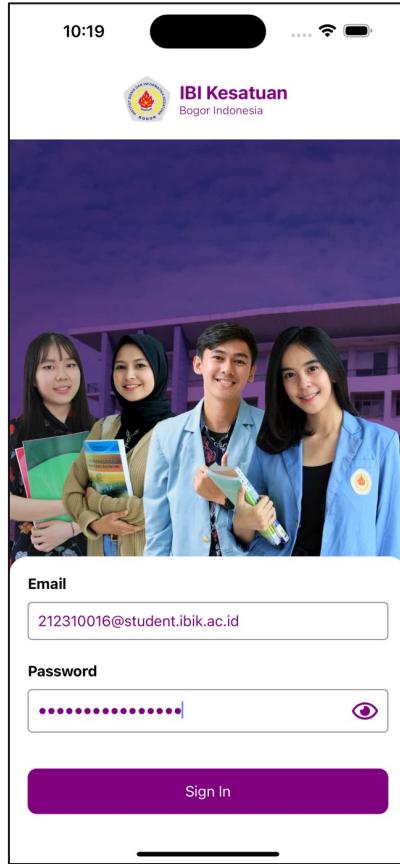
```
import { View, Text } from 'react-native'  
import React from 'react'  
  
const Hellow = () => {  
  return (  
    <View>  
      <Text>ExpClass</Text>  
    </View>  
  )  
}  
  
export default Hellow
```



**Kampus
Merdeka**
INDONESIA JAYA

Tentukan mana **RCC**
dan mana **RFC**
Berdasarkan MOCK UP
disamping?

Component

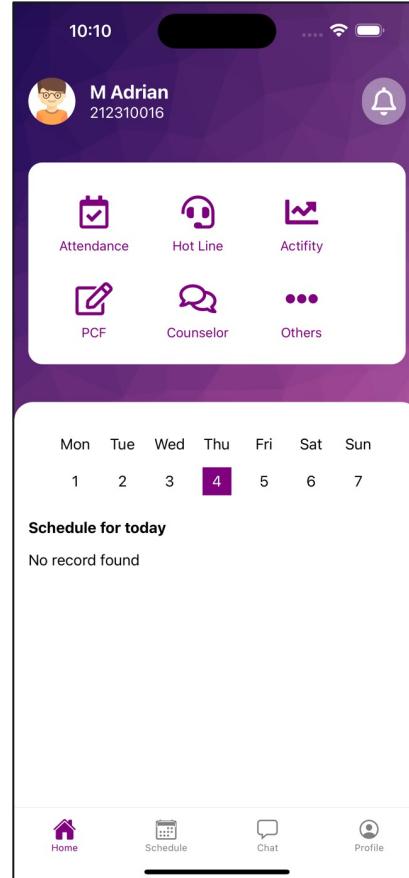




Kampus
Merdeka
INDONESIA JAYA

Tentukan container apa saja yang ada di dalam MOCK UP disamping?

Container



Container

View

```
import { View } from  
  "react-native";  
  <view> ... </view>
```

ScrollView

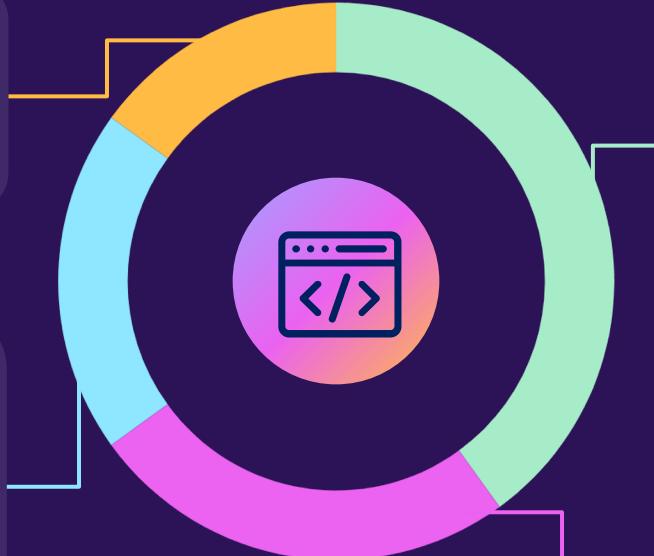
```
import {ScrollView}  
from "react-native";  
<ScrollView>  
  ...  
</ScrollView>
```

SafeAreaView

```
import {SafeAreaView}  
from "react-native";  
<SafeAreaView>  
  ...  
</SafeAreaView>
```

ImageBackground

```
import  
{ImageBackground}  
from "react-native";  
<ImageBackground>  
  ...  
</ImageBackground>
```



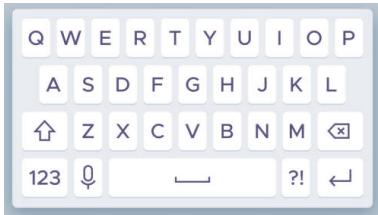


**Kampus
Merdeka**
INDONESIA JAYA

Container

KeyboardAvoidView

```
import { KeyboardAvoidingView } from "react-native";
```



Berpasangan dengan



ScrollView

```
import { ScrollView } from "react-native";
```

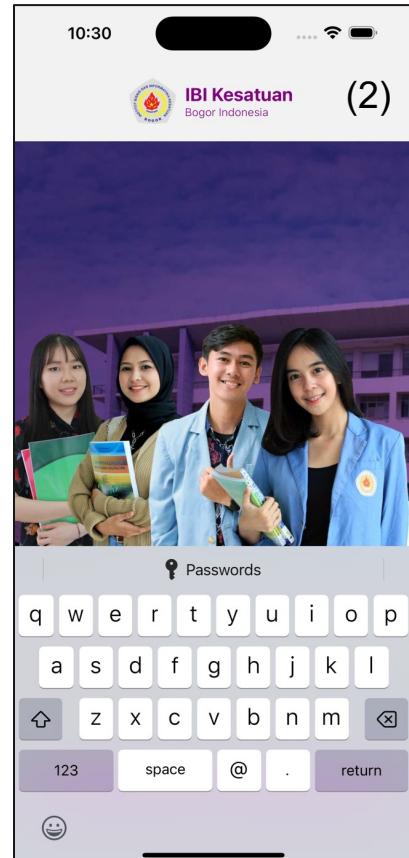


Container

Contoh tanpa menggunakan container
KeyboardAvoidView:

Pada gambar 1 memiliki sebuah form input yg terletak di posisi paling bawah layar. Dimana form tersebut memerlukan inputan data dengan keyboard ponsel.

Pada gambar 2, Ketika memfokuskan pada salah satu form input, maka akan menampilkan keyboard ponsel. Namun form isian tersebut menjadi tidak terlihat, tertutup oleh layer keyboard.



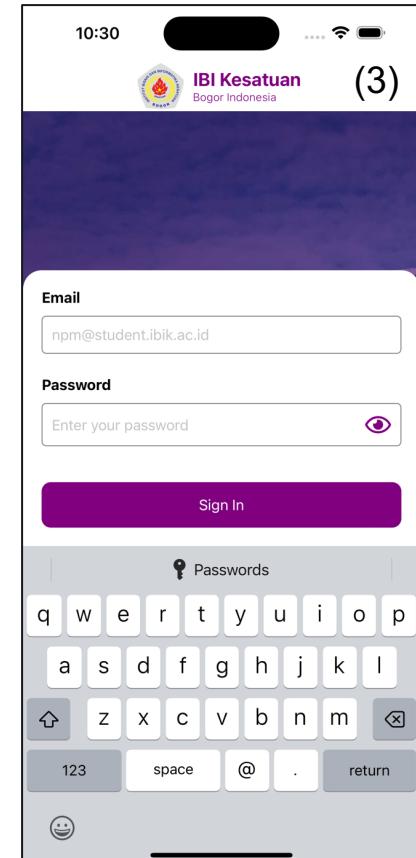


Container

Contoh menggunakan container
KeyboardAvoidView:

Pada gambar 1 memiliki sebuah form input yg terletak di posisi paling bawah layar. Dimana form tersebut memerlukan inputan data dengan keyboard ponsel.

Pada gambar 3, Ketika memfokuskan pada salah satu form input, maka akan menampilkan layer keyboard ponsel, namun posisi form isian akan avoid keatas.





Kampus
Merdeka
INDONESIA JAYA



Container

Implementasi KeyboardAvoidView

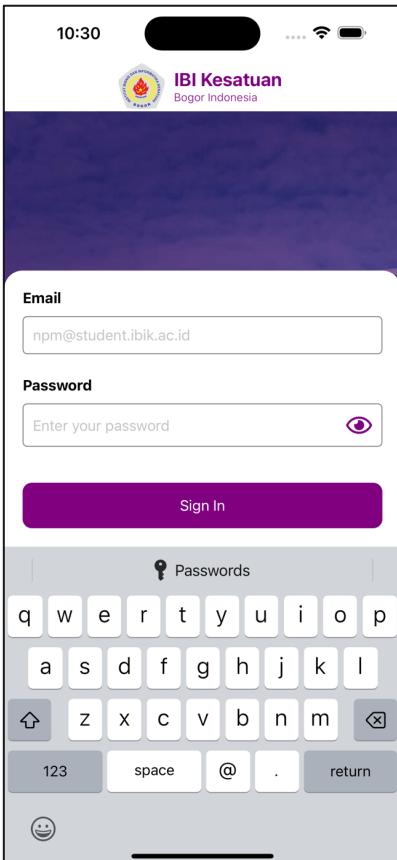
Pada code sebelumnya container awal menggunakan SafeAreaView.

```
export class Auth extends Component {  
  render() {  
    return (  
      <SafeAreaView>  
        .....  
      </SafeAreaView>  
    );  
  }  
}
```

Memasang KeyboardAvoidView didalam component dengan menambahkan code sebagai berikut:



Kampus
Merdeka
INDONESIA JAYA



Container

Implementasi KeyboardAvoidView

```
export class Auth extends Component {
  render() {
    return (
      <KeyboardAvoidingView
        style={{ flex: 1 }}
        behavior={Platform.OS === "ios" ? "padding" : "height"}>
        <ScrollView contentContainerStyle={{ flex: 1 }}>
          <bounces={false}>
            <SafeAreaView>
              .....
            </SafeAreaView>
          </ScrollView>
        </KeyboardAvoidingView>
      );
    }
}
```

Routing

Navigator

Routing

Stack

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

Drawer

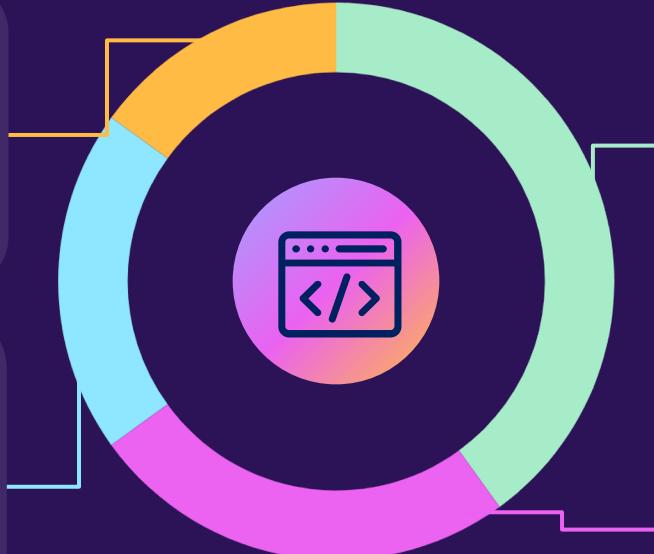
```
import {  
  createDrawerNavigator }  
from '@react-  
navigation/drawer';
```

Bottom Tab

```
import {  
  createBottomTabNavigator }  
from '@react-  
navigation/bottom-tabs';
```

Tab

```
import {  
  createBottomTabNavigator }  
from '@react-  
navigation/bottom-tabs';
```





Kampus
Merdeka
INDONESIA JAYA

Navigator

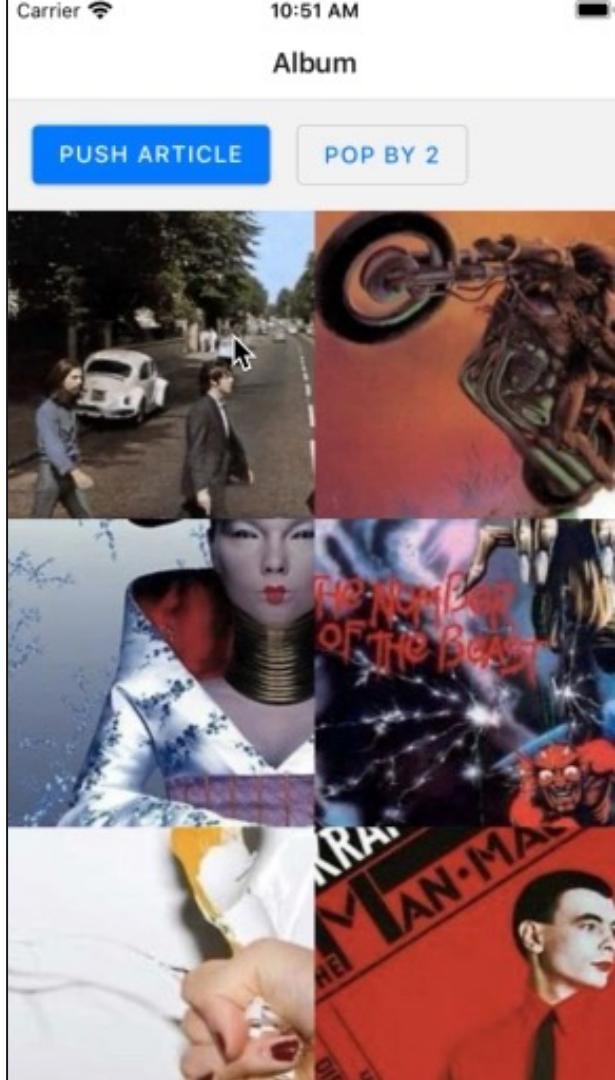
STACK

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

const Stack = createStackNavigator();

function MyStack() {
  return (
    <Stack.Navigator>
      <Stack.Screen name="Home" component={Home} />
      <Stack.Screen name="Profile" component={Profile} />
      <Stack.Screen name="Settings" component={Settings} />
    </Stack.Navigator>
  );
}
```

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





Navigator

Drawer

```
import { createDrawerNavigator } from '@react-navigation/drawer';

const Drawer = createDrawerNavigator();

function MyDrawer() {
  return (
    <Drawer.Navigator>
      <Drawer.Screen name="Feed" component={Feed} />
      <Drawer.Screen name="Article" component={Article} />
    </Drawer.Navigator>
  );
}
```

Home screen!



Kampus
Merdeka
INDONESIA JAYA

Navigator

Bottom Tabs

```
import { createBottomTabNavigator }  
from '@react-navigation/bottom-tabs';  
  
const Tab = createBottomTabNavigator();  
  
function MyTabs() {  
  return (  
    <Tab.Navigator>  
      <Tab.Screen name="Home" component={HomeScreen} />  
      <Tab.Screen name="Settings" component={SettingsScreen} />  
    </Tab.Navigator>  
  );  
}
```

Carrier 2:08 PM

Examples Bottom Tabs

[REPLACE WITH FEED](#) [POP SCREEN](#)



Gandalf
1st Jan 2025

Lorem Ipsum

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old.

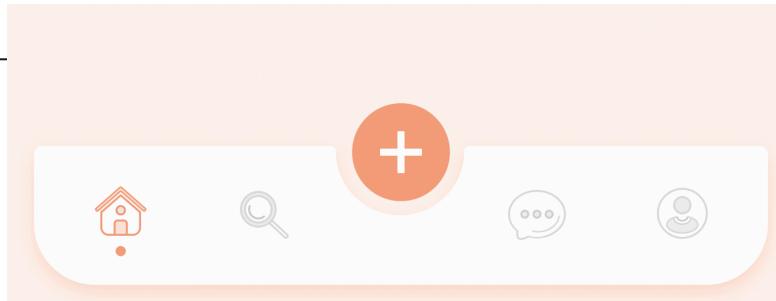




Navigator

Tabs

```
import { createBottomTabNavigator }  
from '@react-navigation/bottom-tabs';  
  
const Tab = createBottomTabNavigator();  
  
function MyTabs() {  
  return (  
    <NavigationContainer>  
      <Tab.Navigator>  
        <Tab.Screen name="Home" component={HomeScreen} />  
        <Tab.Screen name="Settings" component={SettingsScreen} />  
      </Tab.Navigator>  
    </NavigationContainer>  
  );  
}
```





Stack

- Stack Navigator menyediakan cara bagi aplikasi untuk bertransisi antar layer ke layar baru.
- Library react yang digunakan berasal dari *@react-navigation/stack*

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

const Stack = createStackNavigator();
function App() {
  return (
    <Stack.Navigator>
      <Stack.Screen name="Home" component={Home} />
      <Stack.Screen name="Profile" component={Profile}>/>
    </Stack.Navigator>
  );
}

Penamaan dari screen name
  ↗
  ↓
Nama screen component RCC/RFC yang akan ditambahkan
```



**Kampus
Merdeka**
INDONESIA JAYA

Screen Options

Each screen can configure various aspects about how it gets presented in the navigator that renders it by specifying certain options, for example, the header title in stack navigator, tab bar icon in bottom tab navigator etc. Different navigators support different set of options.

<Stack.Screen

```
name="OnBoardScreen"  
component={OnBoardScreen}  
options={{ headerShown:false, gestureEnabled:true }} />
```



Screen Options

Title and Header

title: Sets the title for the screen's header.

headerShown: A boolean that determines if the header is shown or hidden.

headerStyle: An object that customizes the style of the header, such as `backgroundColor`.

headerTintColor: The color of the header title text and back button icon.

headerTitleStyle: An object that customizes the style of the header title text.

headerBackTitle: The text that appears next to the back button on iOS. Can be set to null to hide it.



Screen Options

Animation

animation: The type of animation for screen transitions. Examples include 'fade', 'slide_from_right', 'slide_from_left', 'slide_from_bottom', and 'none'.

Gestures

gestureEnabled: Determines whether you can use gestures to go back on iOS.

Appearance

contentStyle: Customizes the style of the screen's content.

Behavior

orientation: Specifies the allowed orientations for the screen.



Stack

Memasang stack pada route Navigation Controller, pada contoh kasus ini berada pada file App.js

```
import OnBoardScreen from "./src/views/screens/OnBoardScreen";
import MainComponent from "./src/views/navigation/MainComponent";
import DetailAttendance from "./src/views/screens/Attendance/DetailAttendance";

export default class App extends Component {
  render() {
    const Stack = createStackNavigator();
    return (
      <NavigationContainer>
        <StatusBar backgroundColor={colors.white} barStyle="dark-content" />
        <Stack.Navigator screenOptions={{ headerShown: false }}>
          <Stack.Screen name="BoardScreen" component={OnBoardScreen} />
          <Stack.Screen name="Main" component={MainComponent} />
          <Stack.Screen
            name="DetailAttendance"
            component={DetailAttendance}
            options={{
              headerTitle: "Detail Attendance",
              headerShown: true,
              gestureEnabled: true,
              headerBackTitle: "Back",
            }}
          />
        </Stack.Navigator>
      </NavigationContainer>
    );
  }
}
```

Memasukan file yang akan ditambahkan kedalam stack

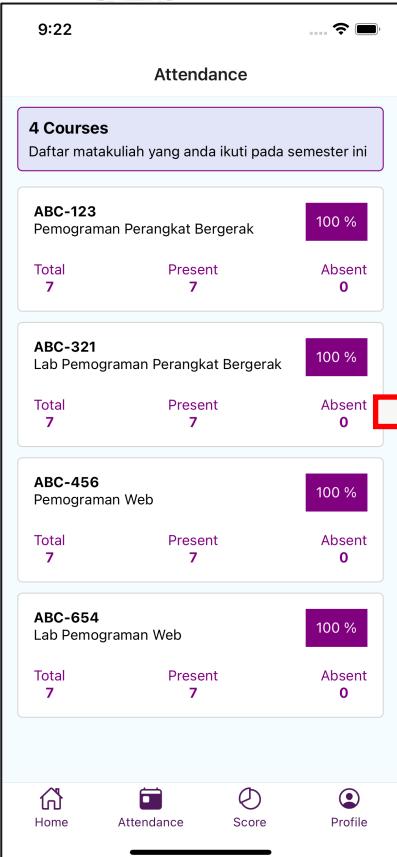
Atribute options ditambahkan pada tag Stack.Screen untuk file DetailAttendance.
Key yang digunakan ialah:

- headerTitle** => Untuk memberikan title pada header screen
- headerShown** => nilai true, untuk menampilkan header screen
- gestureEnabled** => nilai true, untuk menonaktifkan gesture back screen melalui Gerakan sentuh layer
- headerBackTitle** => untuk memberikan nama title pada icon back screen



Stack

9:22



Memasang action click untuk pindah ke screen baru, disini contoh kasus ialah jika mengklik salah satu item dari listing pada halaman Attendance akan mengarah ke halaman DetailAttendance (yg sebelumnya telah ditambahkan).

```
const CourseItem = ({ item, navigation }) => {
  return (
    <TouchableOpacity
      activeOpacity={0.3}
      onPress={() => navigation.navigate("DetailAttendance")}
    >
      <View style={styles.itemContainer}>
        <View style={{ flexDirection: "row" }}>...
        </View>
        <View style={styles.itemFooter}>...
        </View>
      </View>
    </TouchableOpacity>
  );
};
```

Menambahkan parameter navigation kedalam function loop item

Menambahkan element TouchableOpacity untuk membungkus element item agar bisa melakukan event click.

Atribute onPress, berisi value yang mengarahkan kemana screen akan berpindah.

`navigation.navigate ("DetailAttendance")`



Stack

```
export function AttendanceScreen({ navigation })
```

Jika menggunakan RFC cara mengirimkan Parameter navigation cukup menuliskan langsung nama parameteranya, **navigation**

Jika menggunakan RCC parameter navigasi berada dalam props.
Maka untuk menggunakannya tuliskan **this.props.navigation**

Isian attribute navigation diambil dari:

```
<SectionList  
sections={DATA}  
renderItem={({ item }) => (  
  <CourseItem item={item} navigation={navigation} />  
)}
```

```
export class AttendanceScreen extends Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      navigation : this.props.navigation  
    };  
  }
```

```
const CourseItem = ({ item, navigation }) => {  
  return (  
    <TouchableOpacity  
      activeOpacity={0.3}  
      onPress={() => navigation.navigate("DetailAttendance")}  
    >  
      <View style={styles.itemContainer}>  
        <View style={{ flexDirection: "row" }}>...  
        </View>  
        <View style={styles.itemFooter}>...  
        </View>  
      </View>  
    </TouchableOpacity>  
  );  
};
```



Kampus
Merdeka
INDONESIA JAYA

10:10

M Adrian
212310016

Attendance Hot Line Actifity

PCF Counselor Others

Mon Tue Wed Thu Fri Sat Sun

1 2 3 4 5 6 7

Schedule for today

No record found

Home Schedule Chat Profile

Navigator

Back Attendance

4 Courses

Daftar matakuliah yang anda ikuti pada semester ini

ABC-123 Pemograman Perangkat Bergerak 100 %

Total	Present	Absent
8	8	0

ABC-321 Lab Pemograman Perangkat Bergerak 100 %

Total	Present	Absent
8	8	0

ABC-456 Pemograman Web 100 %

Total	Present	Absent
8	8	0

ABC-654 Lab Pemograman Web 100 %

Total	Present	Absent
8	8	0

Attendance Detail Attendance

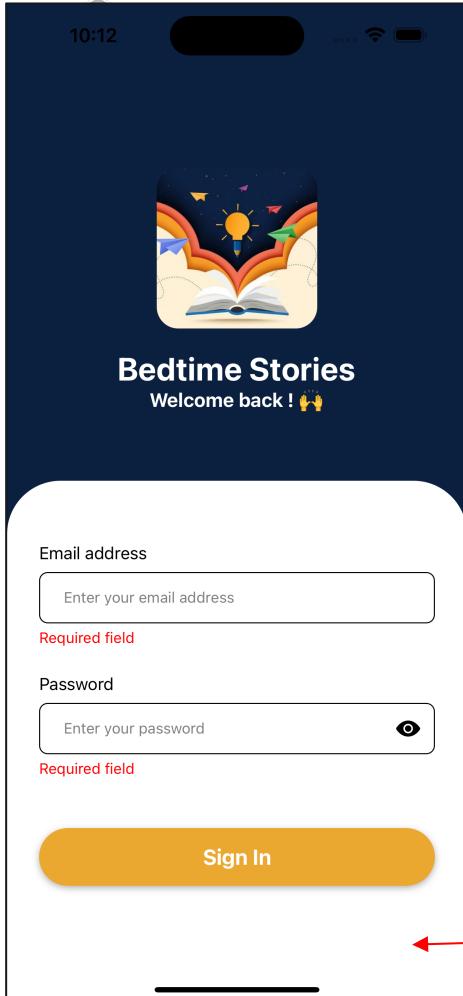
Pemograman Perangkat Bergerak

Lecturer 3320190306 - Febri Damatraseta F.S.T, M.Kom

Code	Credit	Total
ABC-123	3	8

Session Teaching Topic Absent

1	React Native Intro	P
2	React Fundamental	P
3	React Container	P
4	React Typography & Image	P
5	React Navigation Bar	A
6	UTS	P
7	TA Section 01	P
8	React Listing	P



10:13

Create Account

Connect with your friend today!

Email address

Required field

Password

Required field

Mobile Number

Required field

I agree to the terms and conditions

Sign Up

Cancel

A red arrow points from the 'Sign In' link on the join now screen to the 'Sign In' button on the left screen.

Life Cycle



Life Cycle

React Class Component

```
import { Dimensions, Image, SafeAreaView, StyleSheet, Text, TextInput, View, } from "react-native";
import React, { Component } from "react";

export class FormRCC extends Component {
  constructor(props){
    super(props);
    this.state={
      title:"IBI Kesatuan",
      subTitle:"Bogor Indonesia"
    }
  }
  render() {
    return (
      <SafeAreaView>
        <View>
          <Image source={require("../../../../../assets/icons/icon-ibik.png")} />
          <View>
            <Text> {this.state.title} </Text>
            <Text> {this.state.subTitle} </Text>
          </View>
        </View>
      </SafeAreaView>
    )
  }
}
```

Inisialisasi key pada state lifecycle
bernama title, dan subTitle

{this.state.title}
{this.state.subTitle}

Cara menampilkan state
lifecycle pada JSX



Life Cycle

React Class Component

```
<View>
  <View>
    <Text>Change Logo</Text>
  </View>
  <View>
    <Text>Title</Text>
    <TextInput placeholder="Enter title here"
      defaultValue={this.state.title}
      onChangeText={({text})=>this.setState({title:text})} />
  </View>
  <View>
    <Text>Sub Title</Text>
    <TextInput placeholder="Enter sub title here"
      defaultValue={this.state.subTitle}
      onChangeText={({text})=>this.setState({subTitle:text})} />
  </View>
</View>
</SafeAreaView>

};

export default FormRCC;
```

Mengambil nilai value pada TextInput dengan menggunakan properties `onChangeText`, dan untuk mengupdate value pada state gunakan `this.setState({namakey:value})`



Life Cycle

React Function Component

```
import { View, Text, StyleSheet, SafeAreaView, Image, TextInput } from "react-native";
import React, { useState } from "react";

const FormRFC = () => {
  const [title, setTitle] = useState("IBI Kesatuan");
  const [subTitle, setSubTitle] = useState("Bogor Indonesia");

  return (
    <SafeAreaView>
      <View>
        <Image source={require("../../../../../assets/icons/icon-ibik.png")} />
        <View>
          <Text>{title}</Text>
          <Text>{subTitle}</Text>
        </View>
      </View>
      <View>
        <Text>Change Logo</Text>
      </View>
    </SafeAreaView>
  );
}

export default FormRFC;
```

Format penulisan `useState()`:
`[getter, setter] = useState(value)`

Inisialisasi title dan subtitle dengan menggunakan HOOK `useState`

Menampilkan nilai getter dari `useState`

```
<View>
  <Text>Title</Text>
  <TextInput
    placeholder="Enter title here"
    defaultValue={title}
    onChangeText={(text) => setTitle(text)}</TextInput>
  />
</View>

<View>
  <Text>Sub Title</Text>
  <TextInput
    placeholder="Enter sub title here"
    defaultValue={subTitle}
    onChangeText={(text) => setSubTitle(text)}</TextInput>
  />
</View>
</SafeAreaView>
);

export default FormRFC;
```

Mengisi nilai baru pada setter title atau subtitle dengan properties `onChangeText(...)`



Thanks!

Does anyone have any questions?

febrid@ibik.ac.id
+62 81398894710