



# Pemrograman Perangkat Bergerak

# REACT FUNDAMENTALS

Febri Damatraseta Fairuz S.T, M.Kom

# React Expo CLI

Create project and running the apps



# React Native Expo CLI

## Running your React Native application with EXPO Dev

### Step 1: Install Expo

Buka terminal dan masukan syntax dibawah ini:

```
npm install -g expo-cli
```

Jika sudah berhasil terinstall maka dapat dicek dengan syntax

```
expo -V
```

```
Last login: Fri Mar  3 22:23:53 on ttys000
[febfryfairuz@Febrys-MacBook-Air ~ % expo --version
WARNING: The legacy expo-cli does not support Node
Expo CLI (npx expo).
6.3.2
febfryfairuz@Febrys-MacBook-Air ~ %
```

```
my-mobile-apps — node ↵ node /opt/homebrew/bin/yarn expo start — 103x24
febryfairuz@Febrys-MacBook-Air workspace % yarn create expo-app my-mobile-apps
yarn create v1.22.17
[1/4] ⚡  Resolving packages...
[2/4] 🚛  Fetching packages...
[3/4] 📺  Linking dependencies...
[4/4] 📥  Building fresh packages...
expo-app@1.3.2" with binaries:
project files.

s...
Please upgrade to version 7 or higher. Older versions may use Math.random()
which is known to be problematic. See https://v8.dev/blog/math-random for
ts > uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may
in circumstances, which is known to be problematic. See https://v8.dev/blog/
/math-random for details.
warning expo > expo-file-system > uuid@3.4.0: Please upgrade to version 7 or higher. Older versions m
ay use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/bl
og/math-random for details.
warning expo > @expo/cli > cacache > @npmcli/move-file@1.1.2: This functionality has been moved to @npm
@cli/fs
```

### Step 2: Create Project Expo

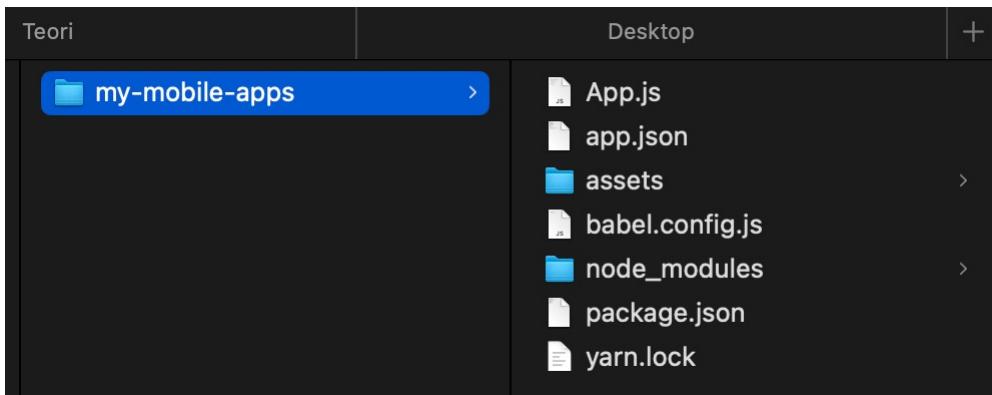
Buka terminal dan masukan syntax dibawah ini:

- npx create-expo-app MyFirstProjectReactExpo
- cd MyFirstProjectReactExpo
- npx expo start



# React Native Expo CLI

## Struktur Project React Native Expo



### Struktur folder

Berbeda dengan struktur sebelumnya, pada struktur Expo tidak dibedakan apakah code native untuk android ataupun ios. File App.js adalah main module yang pertama dieksekusi pada project ini.



**Kampus  
Merdeka**  
INDONESIA JAYA

## Running your React Native application with EXPO Dev

### Step 3: Run Expo Server

Setelah menjalankan npm start maka pada terminal akan menampilkan tampilan seperti gambar dibawah.

Jangan ditutup untuk terminal ini. Pada terminal tertulis address yang dapat kita akses melalui aplikasi browser dengan cara menuliskan

<http://localhost:19000>

# React Native Expo CLI

```
my-mobile-apps — node -v node /opt/homebrew/bin/yarn expo start —1
febryfairuz@Febrys-MacBook-Air my-mobile-apps % yarn expo start
yarn run v1.22.17
$ '/Users/febryfairuz/Documents/IBIK/2022-2023/Genap/Pemrograman Perangkat Bergerak
apps/node_modules/.bin/expo' start
Starting project at /Users/febryfairuz/Documents/IBIK/2022-2023/Genap/Pemrograman P
kspace/my-mobile-apps
Starting Metro Bundler

> Metro waiting on exp://192.168.1.10:19000
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Press a | open Android
> Press i | open iOS simulator
> Press w | open web

> Press j | open debugger
> Press r | reload app
> Press m | toggle menu

> Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
> Opening on iOS...
> Opening exp://192.168.1.10:19000 on iPhone 14 Pro
Downloading the Expo Go app [=====] 100% 0.0
```



**Kampus  
Merdeka**  
INDONESIA JAYA

# React Native Expo CLI

## Running your React Native application with EXPO CLI

### Step 4: Run Expo Application on Smartphone

Pada tampilan expo server terdapat bentuk QR CODE yang dapat kita scan setelah mendaftar sebagai member di [expo.dev](https://expo.dev) dan telah menginstall aplikasi Expo di Google Play Store atau App Store.



# React Fundamentals

React Component, JSX, State

# React Fundamentals



**RCC**

React Class Component

**JSX**

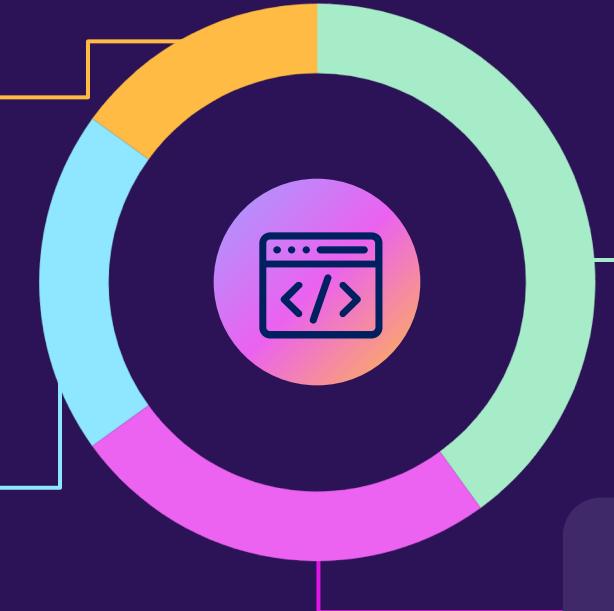
sintaks yang  
memungkinkan Anda  
menulis elemen di  
dalam JavaScript

**RFC**

React Function  
Component

**STATE**

State dan Life cycle





Kampus  
Merdeka  
INDONESIA JAYA

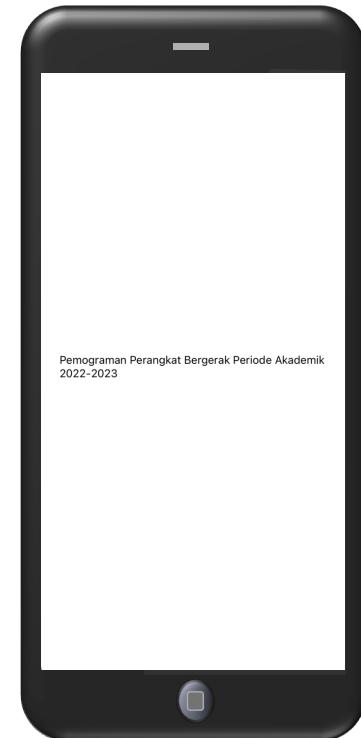
# React Fundamentals

Component

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return [
    <View style={styles.container}>
      <Text>Pemrograman Perangkat Bergerak Periode Akademik 2022-2023</Text>
      <StatusBar style="auto" />
    </View>
  ];
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```





## Class Component

```
import { Component } from 'react/cjs/react.production.min';
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends Component{
  render(){
    const styles = StyleSheet.create({
      container: {
        flex: 1,
        backgroundColor: '#fff',
        alignItems: 'center',
        justifyContent: 'center',
      },
    });

    return (
      <View style={styles.container}>
        <Text>Pemrograman Perangkat Bergerak Periode Akademik 2022-2023</Text>
        <StatusBar style="auto" />
      </View>
    )
  }
}
```

## Functional Component

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return [
    <View style={styles.container}>
      <Text>Pemrograman Perangkat Bergerak Periode Akademik 2022-2023</Text>
      <StatusBar style="auto" />
    </View>
  ];
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```



## Structure Class

```
import { Component } from 'react/cjs/react.production.min';
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends Component{
  render(){
    const styles = StyleSheet.create({
      container: {
        flex: 1,
        backgroundColor: '#fff',
        alignItems: 'center',
        justifyContent: 'center',
      },
    });

    return (
      <View style={styles.container}>
        <Text>Pemrograman Perangkat Bergerak Periode Akademik 2022-2023</Text>
        <StatusBar style="auto" />
      </View>
    )
  }
}
```

# React Fundamentals

## Class Component

Setiap component pada react selalu memiliki satu buah class

**Import** → library pada javascript yang dapat digunakan pada project react. Library ini tersimpan pada folder node\_module

**Render** → sebuah properties yang digunakan untuk menampilkan komponen yang telah dibentuk. Untuk dapat me-render dibutuhkan sebuah return statement. Biasanya return statement menampilkan expresi JSX

**JSX** → Javascript XML, memungkinkan menuliskan syntax HTML kedalam React

**Export default** → berfungsi untuk menginformasikan bahwa main programnya ada di komponen tersebut.

Data handling pada RCC menggunakan **state** atau **props** sebagai *stored data*



# React Fundamentals

## Function Component

### Structure Class

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return [
    <View style={styles.container}>
      <Text>Pemrograman Perangkat Bergerak Periode Akademik |2022-2023</Text>
      <StatusBar style="auto" />
    </View>
  ];
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

RFC adalah jenis React component yang memiliki syntax yang lebih simple dan memungkinkan kita untuk menggunakan React Hooks.

**JSX** → Javascript XML, memungkinkan menuliskan syntax HTML kedalam React

**Export default** → berfungsi untuk menginformasikan bahwa main programnya ada di komponen tersebut.

Data handling RFC menggunakan **props** untuk *stored data*.



## Functional vs class-based React components

### Functional

1 Functional programming style

2 Minimal boilerplate  
Clean and simple

### Class-based

1 Object-oriented programming style

2 Can have state

3 Can have lifecycle methods

perform actions when the component is mounted, unmounted, about to be updated, etc.

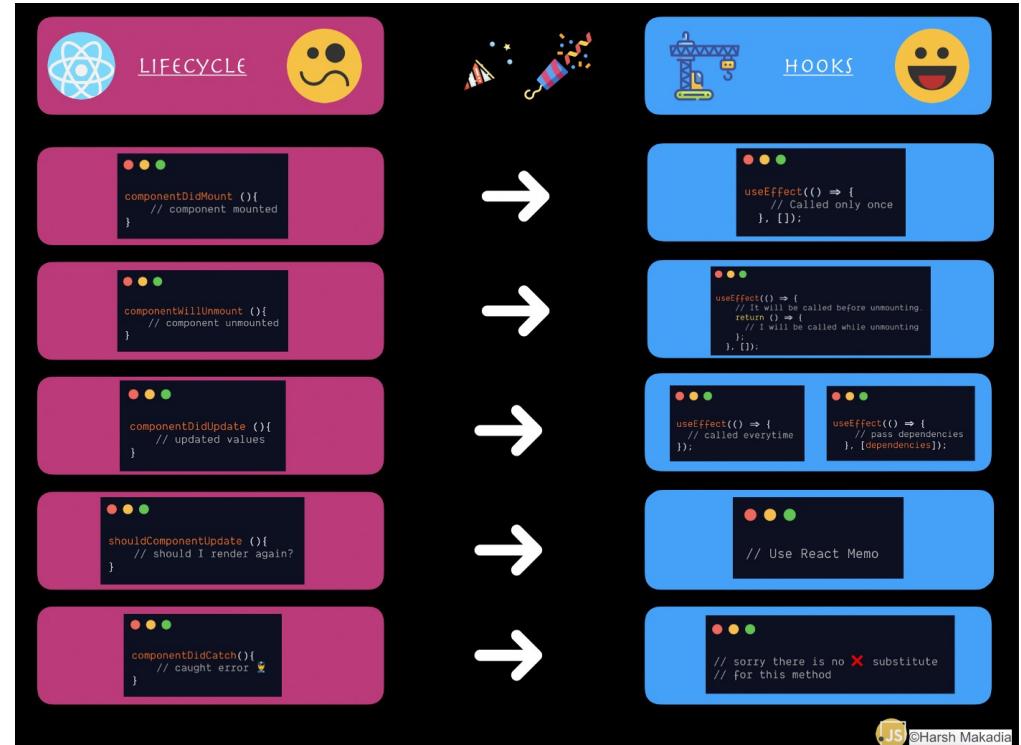
4 Can have refs  
Reference and manipulate underlying DOM elements

5 Performance optimisation

With  
shouldComponentUpdate  
and PureComponent  
Use with caution!

# React Fundamentals

RFC vs RCC



Dalam proses menampilkan component pada browser, React mengeksekusi beberapa method untuk stored data. RCC menggunakan Lifecycle Method, sedangkan RFC menggunakan Hooks



## Class Diagram



## TypeScript

```
component > dashboard > JS Dashboard.js > ...
1  import { Component } from 'react/cjs/react.production.min';
2  import { StatusBar } from 'expo-status-bar';
3  import { StyleSheet, Text, View } from 'react-native';
4  import Header from './Header';
5
6  export default class Dashboard extends Component{
7    render(){
8      const styles = StyleSheet.create({
9        container: {
10          flex: 1,
11          backgroundColor: '#fff',
12          alignItems: 'center',
13          justifyContent: 'center',
14        },
15      });
16
17      return (
18        <View style={styles.container}>
19          <Text>Dashboard</Text>
20          <Header />
21          <StatusBar style="auto" />
22        </View>
23      )
24    }
25 }
```

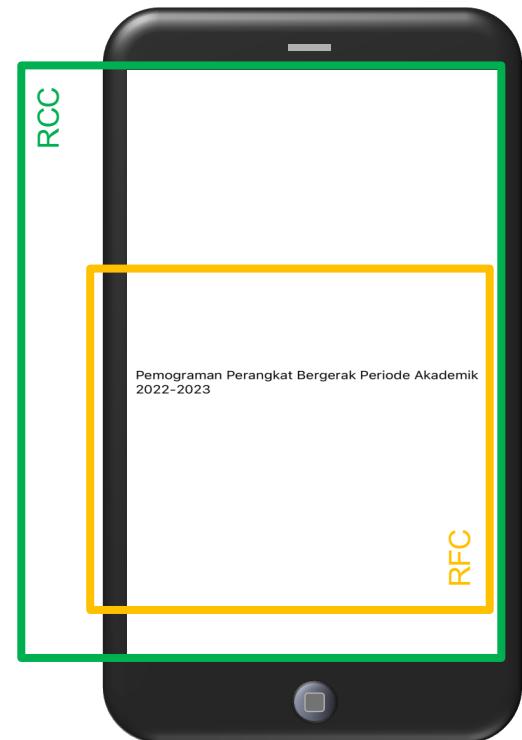
RCC

```
component > dashboard > JS Header.js > ...
1  import { Text } from 'react-native';
2
3  export default function Header(){
4    return (
5      <Text>Pemograman Perangkat Bergerak Periode Akademik 2022-2023</Text>
6    )
7 }
```

RFC

# React Component

Contoh halaman depan bernama komponen *Dashboard*





# React Fundamentals

Roles in naming

## Penamaan dalam React Native

1. Huruf besar pada awal frasa
2. Menggunakan frasa dalam bentuk *nouns*
3. Tidak boleh menggunakan space, angka atau symbol dalam membuat penamaan pada RCC atau RFC
4. Penamaan bertingkat dapat menggunakan huruf besar disetiap frasa

## Contoh

Project		RCC & RFC		Variable	
ProjectReact1	Project React 1	Dashboard	dashboard	satu	1
Project_React-1	Project,React.1	DetailItem	Detail Item	positionX	Position X
Project1_react_1	1project-react	Check_out	Check-out	Position_X	Position-X



# React Fundamentals

Tipe data pada variable

Name	Scope	Desc
const	Block scope	Berisi nilai tetap dan tidak bisa diubah-ubah
let	Block scope	Nilai dapat diubah
var	Functional scope	Nilai dapat diubah dan diakses diluar block kecuali diluar function

Contoh:

```
const radian = 1
console.log(radian)
```

```
const bulan = 'mei'
bulan = 'juni'
console.log(bulan)
```

```
const arrayObj = [{title:'Pem Perangkat Bergerak', id:1}, {title:'Pem Web', id:2}]
console.log(arrayObj)
```

```
let isActive = true
console.log(isActive)
```

```
let name ='Febry'
console.log(name)
```

```
var total = 10.3
console.log(total)
```

```
var obj ={title:'Pem Perangkat Bergerak', id:1}
console.log(obj)
```



Kampus  
Merdeka  
INDONESIA JAYA

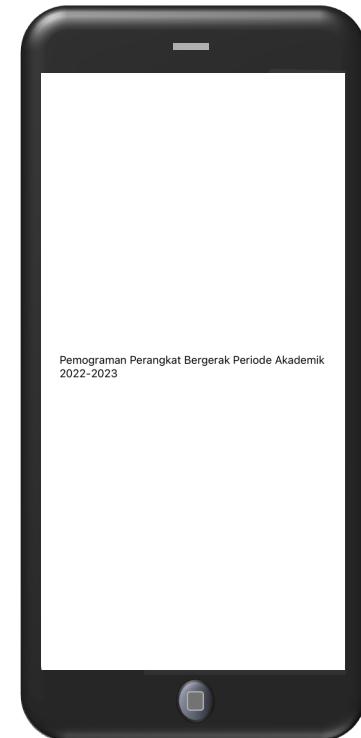
# React Fundamentals

Component

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return [
    <View style={styles.container}>
      <Text>Pemrograman Perangkat Bergerak Periode Akademik 2022-2023</Text>
      <StatusBar style="auto" />
    </View>
  ];
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```





## Class Component

```
import { Component } from 'react/cjs/react.production.min';
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends Component{
  render(){
    const styles = StyleSheet.create({
      container: {
        flex: 1,
        backgroundColor: '#fff',
        alignItems: 'center',
        justifyContent: 'center',
      },
    });

    return (
      <View style={styles.container}>
        <Text>Pemrograman Perangkat Bergerak Periode Akademik 2022-2023</Text>
        <StatusBar style="auto" />
      </View>
    )
  }
}
```

## Functional Component

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return [
    <View style={styles.container}>
      <Text>Pemrograman Perangkat Bergerak Periode Akademik 2022-2023</Text>
      <StatusBar style="auto" />
    </View>
  ];
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```



## Structure Class

```
import { Component } from 'react/cjs/react.production.min';
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends Component{
  render(){
    const styles = StyleSheet.create({
      container: {
        flex: 1,
        backgroundColor: '#fff',
        alignItems: 'center',
        justifyContent: 'center',
      },
    });

    return (
      <View style={styles.container}>
        <Text>Pemrograman Perangkat Bergerak Periode Akademik 2022-2023</Text>
        <StatusBar style="auto" />
      </View>
    )
  }
}
```

# React Fundamentals

## Class Component

Setiap component pada react selalu memiliki satu buah class

**Import** → library pada javascript yang dapat digunakan pada project react. Library ini tersimpan pada folder node\_module

**Render** → sebuah properties yang digunakan untuk menampilkan komponen yang telah dibentuk. Untuk dapat me-render dibutuhkan sebuah return statement. Biasanya return statement menampilkan expresi JSX

**JSX** → Javascript XML, memungkinkan menuliskan syntax HTML kedalam React

**Export default** → berfungsi untuk menginformasikan bahwa main programnya ada di komponen tersebut.

Data handling pada RCC menggunakan **state** atau **props** sebagai *stored data*



# React Fundamentals

## Function Component

### Structure Class

```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return [
    <View style={styles.container}>
      <Text>Pemrograman Perangkat Bergerak Periode Akademik |2022-2023</Text>
      <StatusBar style="auto" />
    </View>
  ];
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

RFC adalah jenis React component yang memiliki syntax yang lebih simple dan memungkinkan kita untuk menggunakan React Hooks.

**JSX** → Javascript XML, memungkinkan menuliskan syntax HTML kedalam React

**Export default** → berfungsi untuk menginformasikan bahwa main programnya ada di komponen tersebut.

Data handling RFC menggunakan **props** untuk *stored data*.



# React Fundamentals

## Function Components

Sample of rfc:

```
import { Text, View } from 'react-native'  
import React from 'react'  
  
export function Latihan() {  
  return (  
    <View>  
      <Text>Latihan</Text>  
    </View>  
  )  
}  
  
export default Latihan;
```

```
import { Text, View } from "react-native";  
import React from "react";  
  
const Latihan = () => {  
  return (  
    <View>  
      <Text>Latihan</Text>  
    </View>  
  );  
};  
  
export default Latihan;
```



# React Fundamentals

## Function Components

Sample of rfc:

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

function textTitle2() {
  return (
    <View>
      <Text>Pemograman Perangkat Bergerak</Text>
    </View>
  )
}

const TextTitle = () =>{
  return <Text>Pemograman Perangkat Bergerak</Text>
}

export default function App() {
  return (
    <SafeAreaView>
      <TextTitle />
      {textTitle2()}
      {TextTitle()}
    </SafeAreaView>
  );
}
```



## Functional vs class-based React components

### Functional

1 Functional programming style

2 Minimal boilerplate  
Clean and simple

### Class-based

1 Object-oriented programming style

2 Can have state

3 Can have lifecycle methods

perform actions when the component is mounted, unmounted, about to be updated, etc.

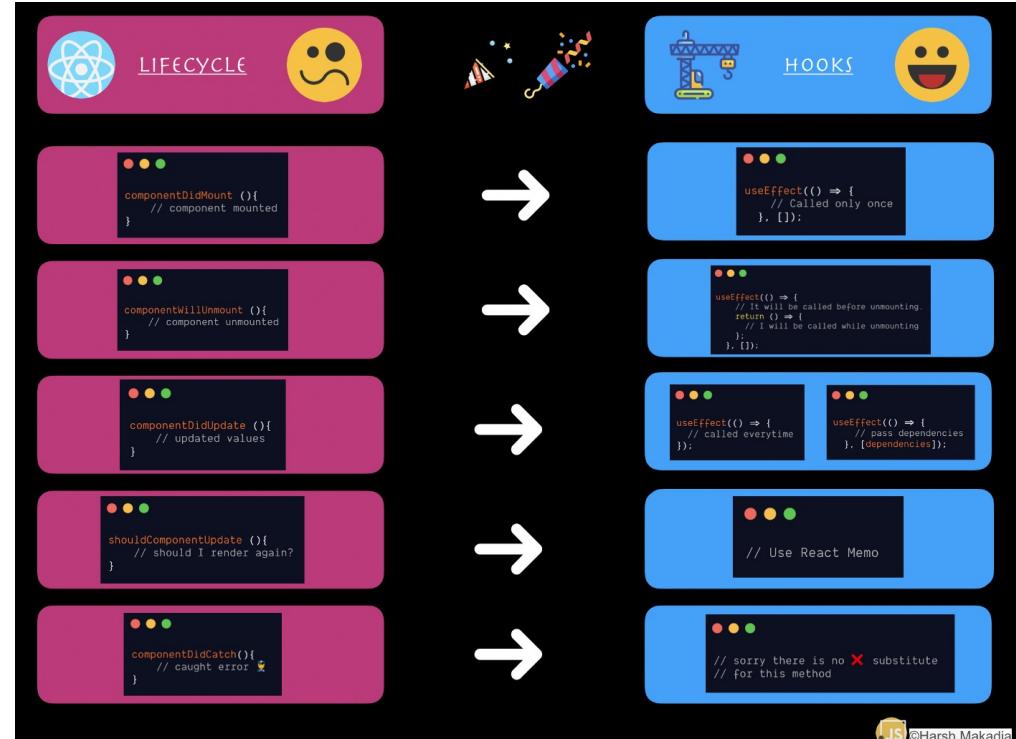
4 Can have refs

Reference and manipulate underlying DOM elements

5 Performance optimisation

With shouldComponentUpdate and PureComponent

Use with caution!



Dalam proses menampilkan component pada browser, React mengeksekusi beberapa method untuk stored data. RCC menggunakan Lifecycle Method, sedangkan RFC menggunakan Hooks



# Question 1

Tentukan component react dibawah ini:

```
import { Text, View } from 'react-native'
import React, { Component } from 'react'

export class Latihan extends Component {
  render() {
    return (
      <View>
        <Text>Latihan</Text>
      </View>
    )
  }
}

export default Latihan;
```

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

export function Latihan() {
  return (
    <View>
      <Text>Latihan</Text>
    </View>
  )
}

export default Latihan;
```

```
import { Text, View } from "react-native";
import React from "react";

const Latihan = () => {
  return (
    <View>
      <Text>Latihan</Text>
    </View>
  );
};

export default Latihan;
```

(A) RCC / Ferdy

(B) RFC / Firdaus

(C) RFC / Askah

Where is the JSX ? ZakyA



# Question 2

Manakah yang benar ?

```
import { Text, View } from 'react-native'  
import React from 'react'  
  
const TextTitle = () => {  
  return(  
    <Text>Pemograman Perangkat Bergerak</Text>  
  )  
}  
  
export function Latihan() {  
  return (  
    <View>  
      <Text>Latihan</Text>  
      <TextTitle />  
    </View>  
  )  
}  
  
export default Latihan;
```

(?)

```
import { SafeAreaView, Text, View } from 'react-native';  
  
function TextTitle() {  
  return (  
    <View>  
      <Text>Pemograman Perangkat Bergerak</Text>  
    </View>  
  )  
}  
  
export default function App() {  
  return (  
    <SafeAreaView>  
      {TextTitle()}  
    </SafeAreaView>  
  );  
}
```



# Question 3

Manakah yang benar ?

```
import { SafeAreaView, Text, View } from 'react-native';

function textTitle() {
    return (
        <View>
            <Text>Pemograman Perangkat Bergerak</Text>
        </View>
    )
}

const TextTitle = () =>{
    return <Text>Pemograman Perangkat Bergerak</Text>
}

export default function App() {
    return (
        <SafeAreaView>
            <TextTitle />
            {textTitle()}
        </SafeAreaView>
    );
}
```



# React Fundamentals

Tipe data pada variable

Name	Scope	Desc
const	Block scope	Berisi nilai tetap dan tidak bisa diubah-ubah
let	Block scope	Nilai dapat diubah
var	Functional scope	Nilai dapat diubah dan diakses diluar block kecuali diluar function

Contoh:

```
const radian = 1
console.log(radian)
```

```
const bulan = 'mei'
bulan = 'juni'
console.log(bulan)
```

```
const arrayObj = [{title:'Pem Perangkat Bergerak', id:1}, {title:'Pem Web', id:2}]
console.log(arrayObj)
```

```
let isActive = true
console.log(isActive)
```

```
let name ='Febry'
console.log(name)
```

```
var total = 10.3
console.log(total)
```

```
var obj ={title:'Pem Perangkat Bergerak', id:1}
console.log(obj)
```



# Expression & State

React Function Component

```
import { SafeAreaView, Text } from 'react-native';

export default function App() {
  const fname = "MUHAMAD";
  let mname = "AGUS";
  var lname = "SETIAWAN";

  return (
    <SafeAreaView>
      <MyNPM />
      <Text>My name is {fname} {mname} {lname}</Text>
    </SafeAreaView>
  );
}

const MyNPM = () =>{
  const npm = 212310004;
  return <Text>NPM {npm}</Text>
}
```

Untuk menampilkan nilai variable / function menggunakan symbol kurung bracket {...}



# Expression & State

React Class Component

```
export class HelloWorld extends Component {  
  constructor(props) {  
    super(props); //menandakan Parent class  
    this.MyNPM = this.MyNPM.bind(this); //init function  
    this.state = { //stored data on state  
      npm: 212310005,  
      fname: "ADJIE",  
      mname: "SYERAFFI",  
      lname: "RAHMAT",  
    };  
  }  
}
```

constructor() – Mengatur nilai state awal dan properti komponen. Inisialisasi varibel dan function berada dalam constructor teknik ini disebut dengan *Lifecycle Method*

```
MyNPM() {  
  return <Text>My NPM</Text>  
}
```

```
render() {  
  return (  
    <View>  
      {this.MyNPM()}  
      <Text>  
        | My name is {this.state.fname} {this.state.mname} {this.state.lname}" "  
      </Text>  
    </View>  
  );  
}  
}  
export default HelloWorld;
```

Untuk menampilkan nilai variable / function menggunakan symbol kurung bracket {...}

# Expression & State

Passing parameters

Passing parameter  
via JSX function

Passing parameter  
via regular function

Receive parameter  
as object

```
import { SafeAreaView, Text } from 'react-native';

export default function App() {
  return (
    <SafeAreaView>
      <MyNPM />
      <Text>My name is {biodata.fullname}</Text>
      <MyHeight value={biodata.height}></MyHeight>
      <MyWeight(biodata.weight)></MyWeight>
    </SafeAreaView>
  );
}

const biodata = {
  npm: 212310012,
  fullname: "FERDY APRILIYANTO",
  height: 170,
  weight: 70.5
}

const MyNPM = () =>{
  return <Text>NPM {biodata.npm}</Text>
}

const MyHeight = ({value})=>{
  return <Text>Height {value} cm</Text>
}

function MyWeight(value) {
  return <Text>Weight {value} kg</Text>
}
```



# Question 4

Bagaimana cara mengisi nilai data object jika datanya ialah sebagai berikut:

NPM	Fullscreen	Address	Age
212310036	MUHAMMAD FIRDAUS	Jakarta	20
212310045	ERDIANA RAGIL SYAWALA	Bogor	21

Rizky Ramdani

# Container

## View

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

## ScrollView

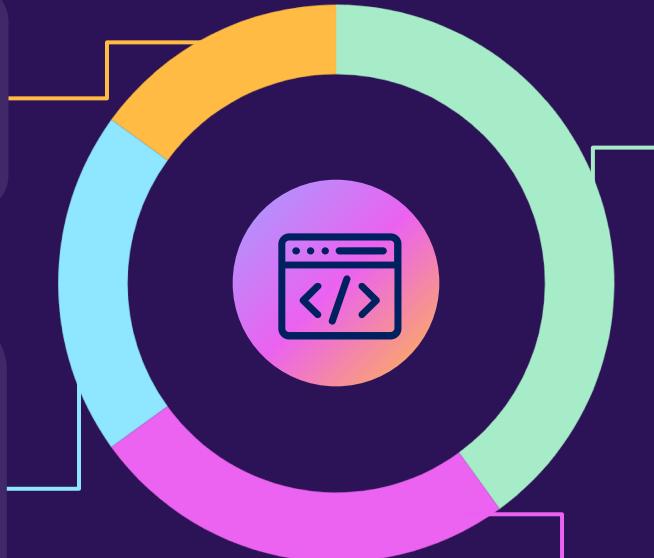
```
import {ScrollView}
from "react-native";
<ScrollView>
  ...
</ScrollView>
```

## SafeAreaView

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

## ImageBackground

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



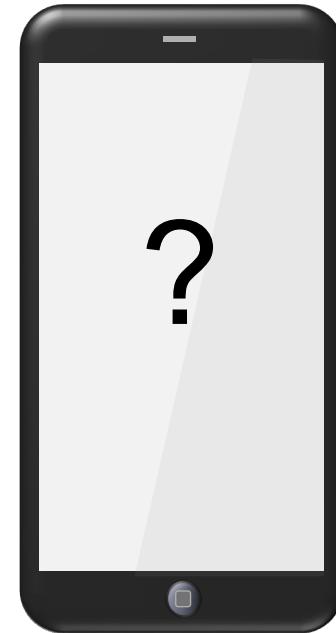


Kampus  
Merdeka  
INDONESIA JAYA

# Question 5

Jika terdapat script seperti ini apa yang akan terjadi ?

```
import {Text} from 'react-native'  
import React from 'react'  
  
export default function Latihan() {  
  return (  
    <Text>Latihan</Text>  
  )  
}
```



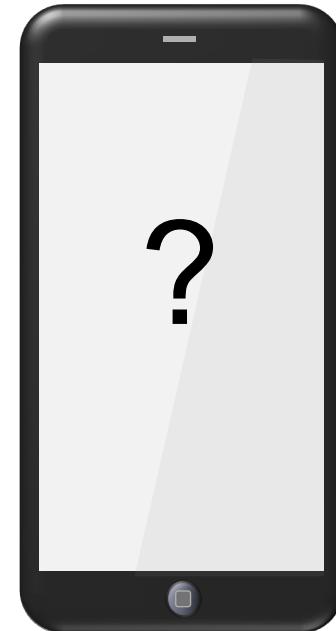


# Question 5

Jika terdapat script seperti ini apa yang akan terjadi ?

```
import {Text} from 'react-native'  
import React from 'react'  
  
export default function Latihan() {  
  return (  
    <Text>Latihan</Text>  
  )  
}
```

Buat multiple element



# Basic Component UI

## Images

```
<Image  
source={  
  require('filename')  
}  
/>
```

## StatusBar

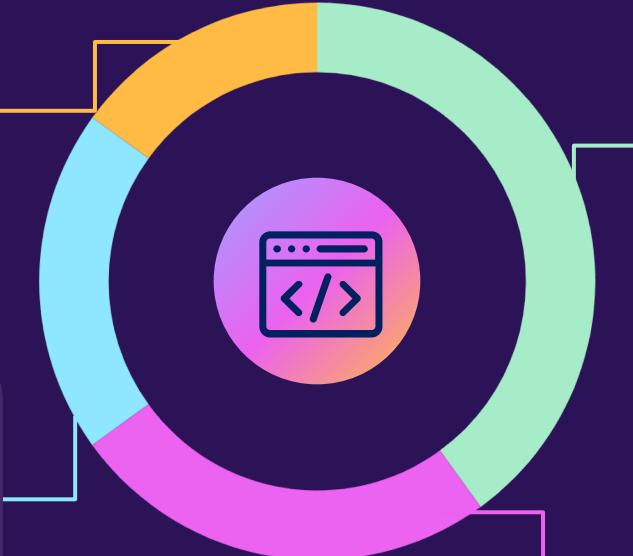
```
<StatusBar  
backgroundColor="#61d  
afb"  
/>
```

## Text

```
<Text>...</Text>
```

## TextInput

```
<TextInput  
value="Hello"  
placeholder="Typing here"  
/>
```





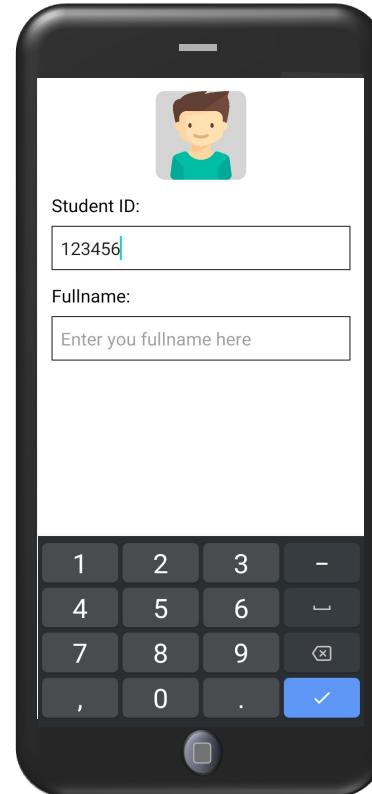
Kampus  
Merdeka  
INDONESIA JAYA

```
export default function LatihanUI() {
  return (
    <SafeAreaView>
      <StatusBar hidden={true} />
      <View>
        <View>
          <Image
            source={require("../assets/ava-boy.png")}
          />
        </View>

        <View>
          <Text>Student ID:</Text>
          <TextInput
            placeholder="Enter your NPM"
            keyboardType="numeric"
          />
        </View>

        <View>
          <Text>Fullname:</Text>
          <TextInput
            placeholder="Enter your fullname here"
          />
        </View>
      </View>
    </SafeAreaView>
  );
}
```

# Sample



# User Interface

Kampus  
Merdeka  
INDONESIA JAYA



## Button

```
<Button  
    title="Click me"  
    onPress={...}  
>
```

## Touchable Opacity

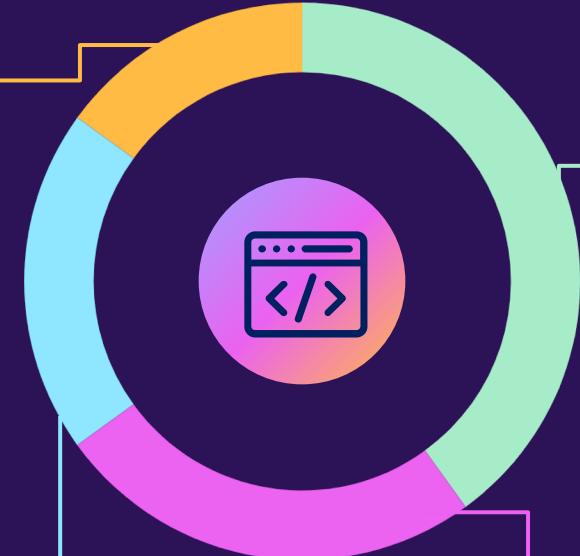
```
<TouchableOpacity  
    onPress={...}  
>  
...  
</TouchableOpacity>
```

## Touchable Highlight

```
<TouchableHighlight  
    onPress={...}>  
...  
</TouchableHighlight>
```

## Touchable Without Feedback

```
<TouchableWithoutFeedback  
    onPress={ ... } >  
...  
</TouchableWithoutFeedback>
```





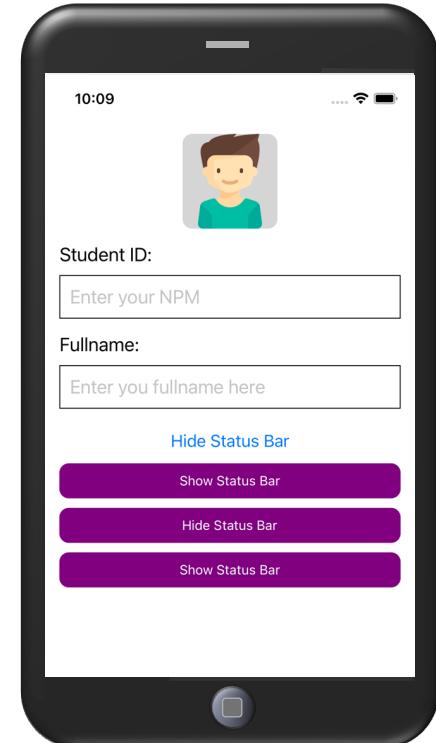
```
const buttonAct = (title) => {
  return (
    <View
      style={{
        backgroundColor: "purple",
        borderRadius: 10,
        padding: 10,
        alignItems: "center",
        marginVertical: 5,
      }}
    >
      <Text style={{ color: "white" }}>{title}</Text>
    </View>
  );
};
```

# Sample

```
<View>
  <Button
    title="Hide Status Bar"
    onPress={() => setToggleStatusBar(true)}
  />
  <TouchableOpacity
    activeOpacity={0.6}
    onPress={() => setToggleStatusBar(false)}
  >
    {buttonAct("Show Status Bar")}
  </TouchableOpacity>

  <TouchableHighlight
    activeOpacity={0.6}
    onPress={() => setToggleStatusBar(true)}
  >
    {buttonAct("Hide Status Bar")}
  </TouchableHighlight>

  <TouchableWithoutFeedback
    activeOpacity={0.6}
    onPress={() => setToggleStatusBar(false)}
  >
    {buttonAct("Show Status Bar")}
  </TouchableWithoutFeedback>
</View>
```





# StyleSheet

## Style

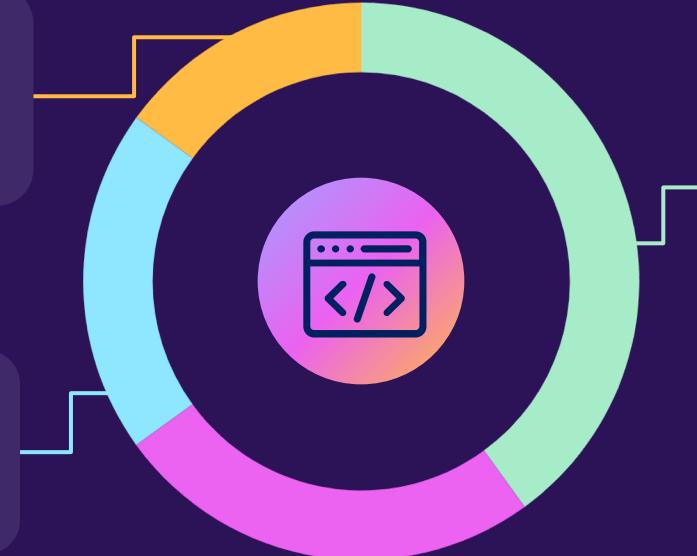
Inline style, internal dan external style

## Layout

Flex  
Flex Direction  
Justify Content  
Align Items  
Align Content  
Flex Wrap

## Dimension

Width and Height



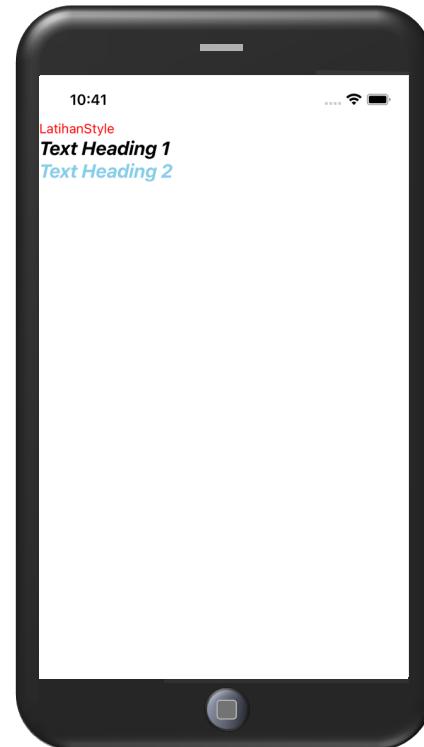


# Style

```
import { SafeAreaView, StyleSheet, Text } from "react-native";
import React from "react";

export default function LatihanStyle() {
  return (
    <SafeAreaView>
      <Text style={{ color: "red" }}>LatihanStyle</Text>
      <Text style={styles.headingText}>Text Heading 1</Text>
      <Text style={{ ...styles.headingText, color: "skyblue" }}>
        Text Heading 2
      </Text>
    </SafeAreaView>
  );
}

const styles = StyleSheet.create({
  headingText: {
    fontSize: 20,
    fontWeight: "bold",
    fontStyle: "italic",
  },
})
```



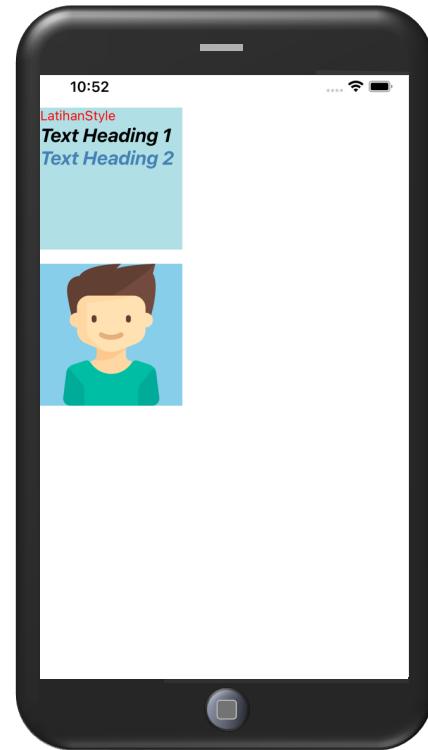


# Dimension

```
export default function LatihanStyle() {
  return (
    <SafeAreaView>
      <View style={{ ...styles.dimension, backgroundColor: "powderblue" }}>
        <Text style={{ color: "red" }}>LatihanStyle</Text>
        <Text style={styles.headingText}>Text Heading 1</Text>
        <Text style={{ ...styles.headingText, color: "steelblue" }}>
          | Text Heading 2
        </Text>
      </View>

      <Image
        source={require("../assets/ava-boy.png")}
        style={{
          ...styles.dimension,
          marginVertical: 15,
          backgroundColor: "skyblue",
        }}
      />
    </SafeAreaView>
  );
}
```

```
const styles = StyleSheet.create({
  headingText: { ... },
  dimension: {
    width: 150,
    height: 150,
  },
});
```





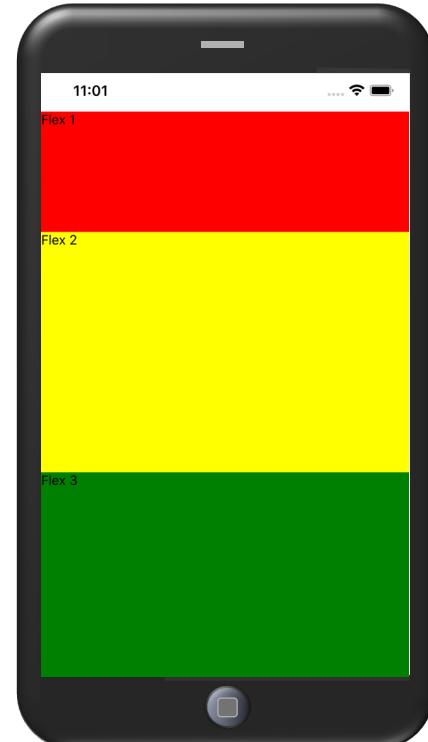
# Layout - Flex

```
export default function LatihanStyle() {
  return (
    <SafeAreaView style={{flex: 1 }}>
      <View style={{ flex: 1, backgroundColor: "red" }}>
        <Text>Flex 1</Text>
      </View>

      <View style={{ flex: 2, backgroundColor: "yellow" }}>
        <Text>Flex 2</Text>
      </View>

      <View style={{ flex: 3, backgroundColor: "green" }}>
        <Text>Flex 3</Text>
      </View>
    </SafeAreaView>
  );
}
```

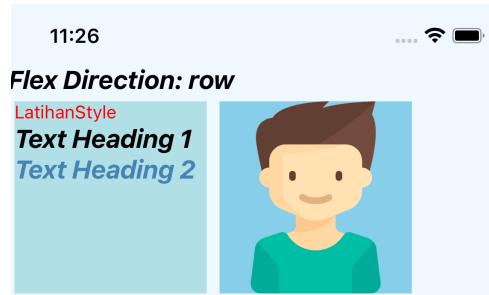
flex akan menentukan bagaimana item akan *"mengisi"* ruang yang tersedia di sepanjang sumbu utama.



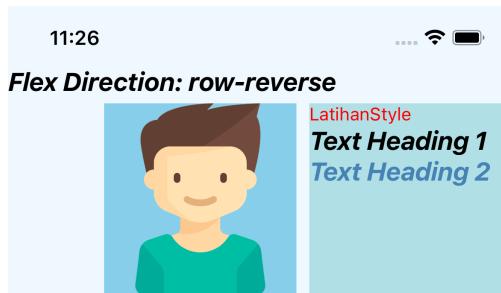


# Layout – Flex Direction

```
return (
  <SafeAreaView style={{ flex: 1, backgroundColor: "aliceblue" }}>
    <View style={{ flexDirection: "row", flex: 1 }}>
      <ItemText />
      <ItemImage />
    </View>
  </SafeAreaView>
);
```



Position  
Left to Right



Position  
Right to Left

```
return (
  <SafeAreaView style={{ flex: 1, backgroundColor: "aliceblue" }}>
    <View style={{ flexDirection: "row-reverse", flex: 1 }}>
      <ItemText />
      <ItemImage />
    </View>
  </SafeAreaView>
);
```



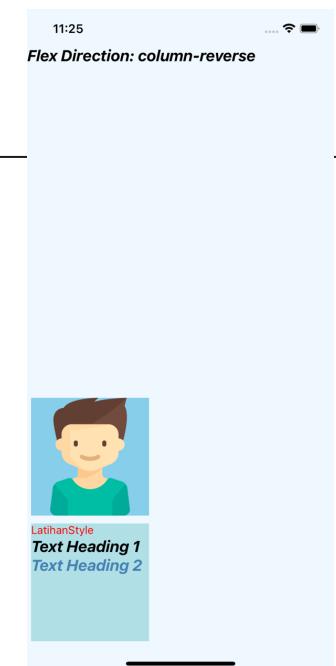
# Layout – Flex Direction

```
return (
  <SafeAreaView style={{ flex: 1, backgroundColor: "aliceblue" }}>
    <View style={{ flexDirection: "column", flex: 1 }}>
      <ItemText />
      <ItemImage />
    </View>
  </SafeAreaView>
);
```

```
return (
  <SafeAreaView style={{ flex: 1, backgroundColor: "aliceblue" }}>
    <View style={{ flexDirection: "column-reverse", flex: 1 }}>
      <ItemText />
      <ItemImage />
    </View>
  </SafeAreaView>
);
```



Position  
Top to Bottom



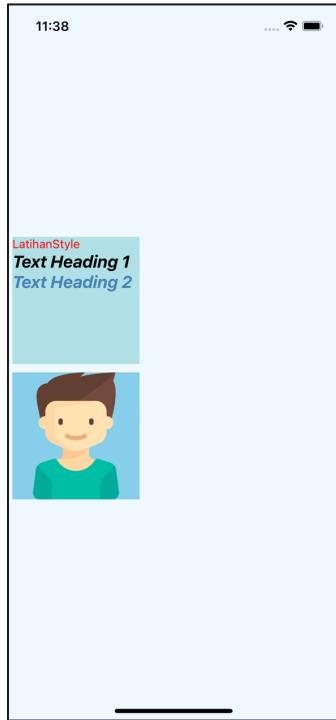
Position  
Bottom to Top



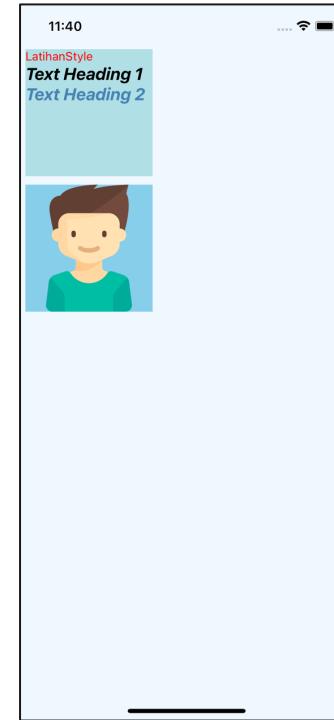
Kampus  
Merdeka  
INDONESIA JAYA

# Layout – Justify Content

style={{ justifyContent:'center' }}



style={{ justifyContent:'flex-start' }}

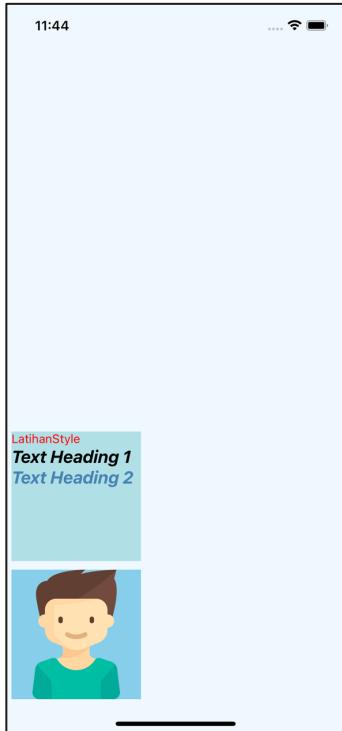




Kampus  
Merdeka  
INDONESIA JAYA

# Layout – Justify Content

style={{ justifyContent:'flex-end' }}



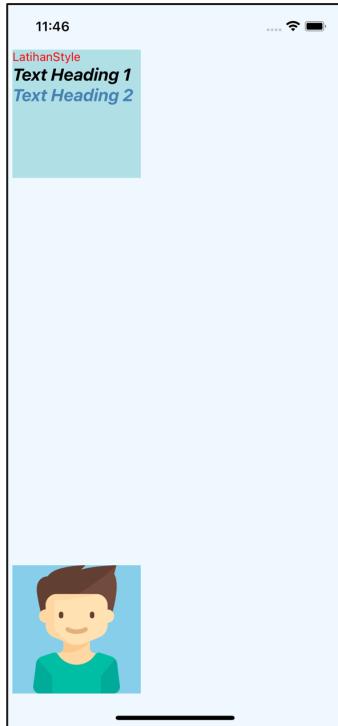
style={{ justifyContent:' space-around' }}



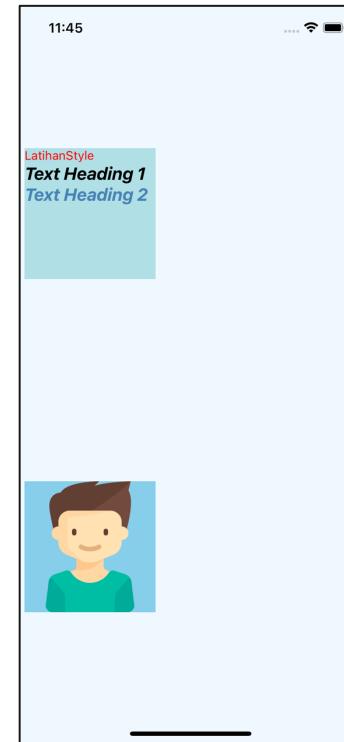


# Layout – Justify Content

style={{ justifyContent:'space-between' }}



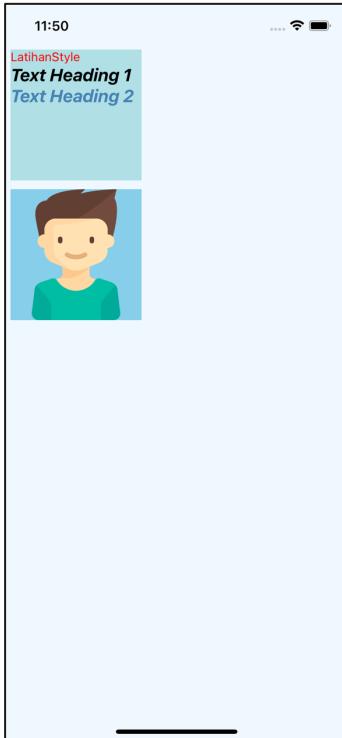
style={{ justifyContent:'space-evenly' }}





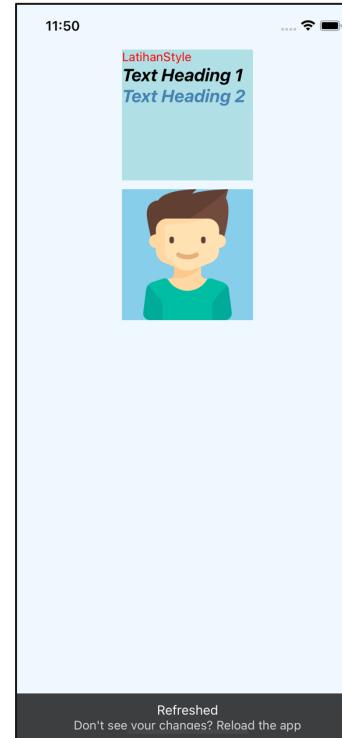
**Kampus  
Merdeka**  
INDONESIA JAYA

style={{alignItems:'baseline' }}



# Layout – Align Items

style={{alignItems:'center' }}

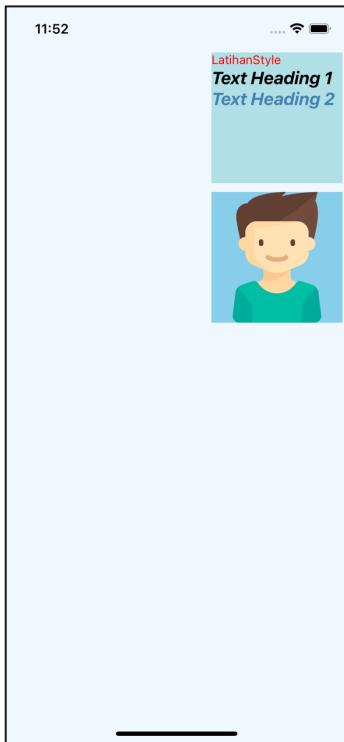


Refreshed  
Don't see your changes? Reload the app

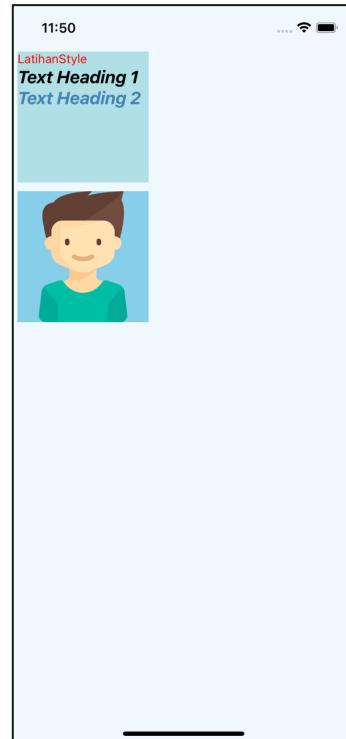


# Layout – Align Items

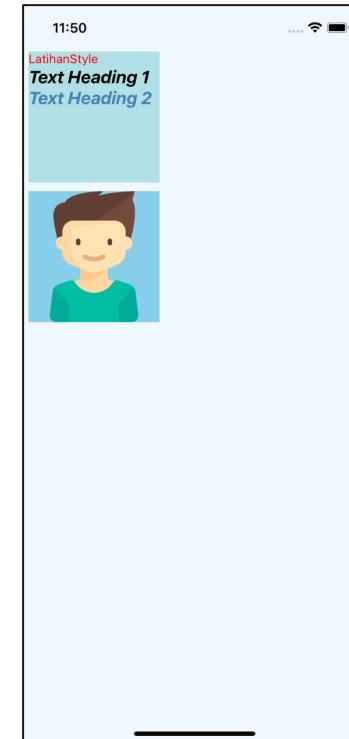
style={{alignItems:'flex-end' }}



style={{alignItems:'flex-start' }}



style={{alignItems:'stretch' }}





**Kampus  
Merdeka**  
INDONESIA JAYA

# Question 6

Buatlah structure untuk layout seperti berikut:





# Thanks!

Does anyone have any questions?

[febrid@ibik.ac.id](mailto:febrid@ibik.ac.id)  
+62 81398894710