

A dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date '1/1/2023'. In the bottom-left corner, there are several thin, curved, light blue lines that sweep upwards and to the right.

1/1/2023

# React Native

**Belajar Mengenai React Hooks dan  
Latihan Aritmatika**

Novinaldi .

## Daftar Isi

<b>Mengenal Hooks .....</b>	<b>2</b>
<b>Latihan Membuat Perhitungan Aritmatika .....</b>	<b>7</b>
<b>Tugas Yang Wajib di-Kerjakan .....</b>	<b>10</b>

## Mengenal Hooks

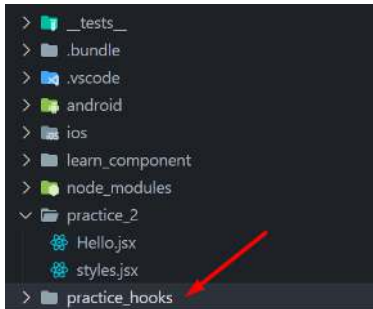
**Hooks** adalah fitur yang diperkenalkan di **React 16.8** yang memungkinkan Anda menggunakan state dan fitur-fitur React lainnya tanpa menulis sebuah class. Berikut adalah beberapa **hooks** dasar dan contoh penggunaannya dalam konteks React Native:

1. **useState()**: Digunakan untuk menambahkan state lokal ke function components.

Contoh perintahnya :

```
const [count, setCount] = useState(0);
```

Kita akan coba buat contoh latihannya. Silahkan kalian terlebih dahulu buat **Folder Baru** dengan nama folder **practive\_hooks**



Selanjutnya buat file baru simpan dalam folder tersebut dengan nama file **LatUseState.jsx**



Kemudian kalian ketikkan perintah berikut :

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

export default function LatUseState() {
  const [count, setCount] = useState(0);

  return (
    <View style={{flex: 1, justifyContent: 'center', alignItems: 'center'}}>
      <Text
        style={{
          fontSize: 40,
          fontWeight: 'bold',
          color: 'red',
          marginVertical: 10,
        }}>
        {count}
      </Text>
      <TouchableOpacity
        style={{
          backgroundColor: 'blue',

```

```

        padding: 10,
        borderRadius: 10,
    }}
    onPress={() => setCount(count + 1)}>
    <Text
        style={{
            color: '#FFF',
        }}>
        Click Me
    </Text>
</TouchableOpacity>
<TouchableOpacity
    style={{
        backgroundColor: '#ff8188',
        padding: 10,
        borderRadius: 10,
        marginTop: 10,
    }}
    onPress={() => setCount(0)}>
    <Text
        style={{
            color: '#FFF',
        }}>
        Reset
    </Text>
</TouchableOpacity>
</View>
);
}

const styles = StyleSheet.create({});

```

Jika sudah silahkan kalian coba jalankan, terlebih dahulu edit pada bagian file **index.js** menjadi seperti berikut

```

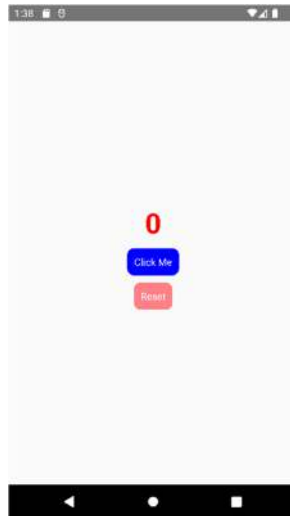
/**
 * @format
 */

import {AppRegistry} from 'react-native';
import App from './practice_hooks/LatUseState';
import {name as appName} from './app.json';

AppRegistry.registerComponent(appName, () => App);

```

Hasilnya :



2. **useEffect** merupakan Mirip dengan `'componentDidMount'`, `'componentDidUpdate'`, dan `'componentWillUnmount'` yang dikombinasikan. Digunakan untuk **side effects** dalam komponen.

Seperti apa contohnya, silahkan kalian buat file baru lagi misalkan nama file nya **LatUseEffect.jsx** (*ingat didalam folder **practice\_hooks** yang telah kalian buat tadi*), ketikkan perintah berikut :

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

export default function LatUseEffect() {
  const [message, setMessage] = useState('');

  useEffect(() => {
    setMessage('Selamat datang di aplikasi kami!');
    return () => {
      // Bersihkan resource jika perlu, seperti event listeners atau timers.
    };
  }, []); // Array kosong berarti useEffect hanya dijalankan sekali setelah render pertama.

  return (
    <View style={{flex: 1, justifyContent: 'center', alignItems: 'center'}}>
      <Text>{message}</Text>
    </View>
  );
}

const styles = StyleSheet.create({});
```

Jika sudah, silahkan kalian jalankan file di android kalian. Lihat lah hasilnya.

### 3. useContext

Memudahkan penggunaan React Context untuk berbagi nilai antar komponen tanpa harus melewatinya melalui props.

kita akan coba contohkan, silahkan kalian buat file baru dengan nama file **LatUseContext.jsx**, kemudian ketikkan perintah berikut :

```
import {View, Text, Button, StyleSheet} from 'react-native';
import React, {useContext, useState} from 'react';

const ThemeContext = React.createContext();

function ThemeButton() {
  const theme = useContext(ThemeContext);
  return <Button title="Ubah Tema" onPress={theme.toggleTheme} />;
}

export default function LatUseContext() {
  const [darkTheme, setDarkTheme] = useState(false);
  const toggleTheme = () => {
    setDarkTheme(prevTheme => !prevTheme);
  };

  return (
    <ThemeContext.Provider value={{isDark: darkTheme, toggleTheme}}>
      <View
        style={{
          flex: 1,
          justifyContent: 'center',
          alignItems: 'center',
          backgroundColor: darkTheme ? 'black' : 'white',
        }}>
        <Text style={{color: darkTheme ? 'white' : 'black'}}>
          Tema saat ini: {darkTheme ? 'Gelap' : 'Terang'}
        </Text>
        <ThemeButton />
      </View>
    </ThemeContext.Provider>
  );
}

const styles = StyleSheet.create({});
```

Silahkan jalankan dan lihat hasilnya.

#### 4. **useReducer**

Alternatif untuk **useState**, biasanya digunakan untuk **state logic** yang lebih kompleks atau ketika **state** berikutnya bergantung pada **state** sebelumnya.

Kita coba Latihan, silahkan kalian buat file baru dengan nama file **LatUseReducer.jsx** ketikkan perintah berikut :

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

const initialState = {count: 0};

function reducer(state, action) {
  switch (action.type) {
    case 'increment':
      return {count: state.count + 1};
    case 'decrement':
      return {count: state.count - 1};
    default:
      throw new Error();
  }
}

export default function LatUseReducer() {
  const [state, dispatch] = useReducer(reducer, initialState);

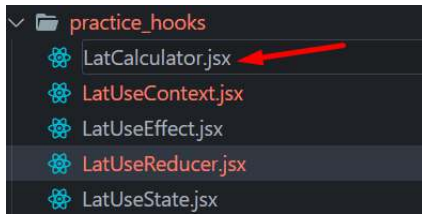
  return (
    <View style={{flex: 1, justifyContent: 'center', alignItems: 'center'}}>
      <Text>Count: {state.count}</Text>
      <Button title="Tambah" onPress={() => dispatch({type: 'increment'})} />
      <Button title="Kurangi" onPress={() => dispatch({type: 'decrement'})} />
    </View>
  );
}

const styles = StyleSheet.create({});
```


Simpan filenya, dan lihat hasilnya...

## Latihan Membuat Perhitungan Aritmatika

1. Kita akan coba membuat Latihan perhitungan sederhana, silahkan kalian buat file baru dengan nama file **LatCalculator.jsx**,



Oiya sebelumnya kita membutuhkan sebuah **Package** yang namanya **Picker** untuk membuat **Dropdown**.

Silahkan kalian tutup **node.js** nya yang gambar ini . Tekan aja **CTRL+C** pada keyboard, kemudian tutup. Kemudian pada bagian **git bash** kalian ketik perintah berikut :

**yarn add @react-native-picker/picker**

```
ASUS@DESKTOP-HKLH0MB MINGW64 /e/Novinaldi/Mengajar/2023-2024/ganjil/Aplikasi Mobile 2/AppNovinaldi
$ yarn add @react-native-picker/picker
yarn add v1.22.19
warning package-lock.json found. Your project contains lock files generated by tools other than Yarn. It is advised not to mix package managers in order to avoid resolution inconsistencies caused by unsynchronized lock files. To clear this warning, remove package-lock.json.
```

Silahkan ditunggu sampai selesai. Jika sudah silahkan jalankan kembali project android nya dengan mengetikkan perintah : **npm run android**

Kemudian kembali ke file **LatCalculator.jsx** kalian ketikkan perintah berikut :

```
import {StyleSheet, View, TextInput, Button, Text} from 'react-native';
import React, {useState} from 'react';
import {Picker} from '@react-native-picker/picker';

export default function LatCalculator() {
  const [number1, setNumber1] = useState('');
  const [number2, setNumber2] = useState('');
  const [operation, setOperation] = useState('+');
  const [result, setResult] = useState('');

  const calculate = () => {
    let n1 = parseFloat(number1);
    let n2 = parseFloat(number2);
    let res = 0;

    switch (operation) {
      case '+':
        res = n1 + n2;
        break;
    }
  }
}
```



```

        case '-':
            res = n1 - n2;
            break;
        case '*':
            res = n1 * n2;
            break;
        case '/':
            res = n1 / n2;
            break;
    }

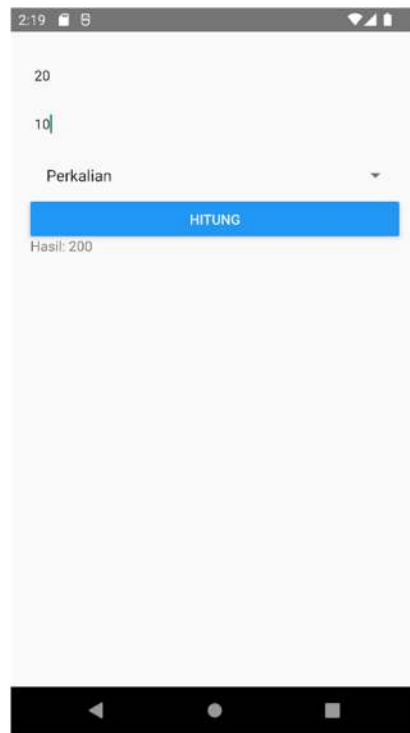
    setResult(res);
};

return (
    <View style={{padding: 20}}>
        <TextInput
            placeholder="Angka Pertama"
            keyboardType="numeric"
            value={number1}
            onChangeText={text => setNumber1(text)}
        />
        <TextInput
            placeholder="Angka Kedua"
            keyboardType="numeric"
            value={number2}
            onChangeText={text => setNumber2(text)}
        />
        <Picker
            selectedValue={operation}
            onValueChange={itemValue => setOperation(itemValue)}>
            <Picker.Item label="Penambahan" value="+" />
            <Picker.Item label="Pengurangan" value="-" />
            <Picker.Item label="Perkalian" value="*" />
            <Picker.Item label="Pembagian" value="/" />
        </Picker>
        <Button title="Hitung" onPress={calculate} />
        <Text>Hasil: {result}</Text>
    </View>
);
}

const styles = StyleSheet.create({});

```

jika sudah silahkan kalian lihat hasilnya :



## Tugas Yang Wajib di-Kerjakan

1. Buatlah **Form Menghitung Persegi Panjang dan Segitiga**.
2. Yang mana Inputannya untuk **Persegi Panjang** : Ada Input Angka Panjang dan Lebar Beserta Pilihan (Picker Dropdown) **Luas dan Keliling**  
Untuk **Segitiga** : ada inputan **Alas dan Tinggi** (*tidak menggunakan **Picker Dropdown***)