Nama : Satrio Kunto Birowo

NIM : 118140001

Kelas : Pemrograman Aplikasi Mobile – RB

Pada tugas ini saya melakukan percobaan pembuatan kalkulator sederhana menggunakan react-native redux dimana saya menggunakan contoh yang ada pada link medium, berikut saya sertakan link yang ada pada website tersebut : https://medium.com/skyshidigital/memulai-react-redux-dengan-aplikasi-calculator-sederhana-86afc742f3a1

Melakukan inisisiasi project

```
C:\Windows\System32\PAM>npx create-react-app kalkulator-redux

Creating a new React app in C:\Windows\System32\PAM\kalkulator-redux.
```

Menambahkan library redux kedalam project kalkulator-redux yang berguna untuk melakukan koneksi antara react dan redux

```
C:\WINDOWS\system32>npm install --save redux react-redux
```

Melakukan modifikasi pada file struktur agar dapat membaca file yang ada pada file lain

```
public
  -index.html
src
  -actions
    -index.js
  -components
    -App.js
  -containers
    -ButtonList.js
    -History.js
    -Screen.js
  -reducers
    -calculator.js
    -index.js
  -index.js
```

Menambahkan file bootsrap pada index.html

```
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta name="theme-color" content="#000000">
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/
SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">
<!-- font awesome -->
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css" integrity="sha384-mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU"
crossorigin="anonymous">
<style>
.screen {
font-size: 40px;
text-align: right;
padding: 10px;
margin: 10px 0;
border: 1px gainsboro solid;
border-radius: 3px;
width: 100%;
display: block;
}
</style>
<title>React Redux Calculator Apps</title>
</head>
<body>
<noscript>
You need to enable JavaScript to run this app.
</noscript>
<div class="container" id="root"></div>
</body>
</html>
```

Menambahkan file calculator.js pada Reducers yang berguna untuk menyimpan state aplikasi

```
import { PRESS_BUTTON, PRESS_EQUALS, PRESS_RESET, SELECT_HISTORY, CLEAR_HISTORY } from '../actions/index'
const INITIAL_STATE = { screen: '', history: [] }
const calculator = (state = INITIAL_STATE, action) => {
    switch (action.type) {
        case PRESS_BUTTON:
        return {
            ...state,
            screen: (state.screen + '' + action.value)
        }
        case PRESS_EQUALS:
            const newState = { ...state }
            newState.screen = (eval(state.screen))
            newState.history = state.history.concat(`${state.screen}`)
        return newState
        case PRESS_RESET:
            return {
            ...state,
            screen: ''
        }
        case SELECT_HISTORY:
            return {
            ...state,
            screen: state.history[action.id]
            }
        case CLEAR_HISTORY:
            return {
            ...state,
            history: []
            }
            default:
                return state
        }
}
export default calculator
```

Kemudian menggabungkan reducer yang ada pada calculator dengan rootReducer yang ada di index.js

```
import { combineReducers } from 'redux'
import calculator from './calculator'
export default combineReducers({
  calculator
})
```

Memasukan action yang akan digunakan untuk menyimpan action creator untuk melakukan perubahan state

```
export const PRESS_BUTTON = 'PRESS_BUTTON'
export const PRESS_EQUALS = 'PRESS_EQUALS'
export const PRESS_RESET = 'PRESS_RESET'
export const SELECT_HISTORY = 'SELECT_HISTORY'
export const CLEAR_HISTORY = 'CLEAR_HISTORY'

export function pressButton(value){
    console.log('PRESS_BUTTON : ', value)
    return{
        type: PRESS_BUTTON,
        value
    }
}
export function pressEqual(){
    console.log('PRESSED BUTTON : equals')
    return{
        type: PRESS_EQUALS
    }
}
export function pressReset(){
    console.log('PRESSED RESET')
    return{
        type: PRESS_RESET
    }
}
export function selectHistory(id){
    console.log('SELECT HISTORY', id)
    return{
        type: SELECT_HISTORY,
        id
    }
}
export function clearHistory(){
    console.log('CLEAR_HISTORY')
    return{
        type: CLEAR_HISTORY
    }
}
```

Menambahkan container screen js yaitu component yang melakukan komunikasi ke app state sehingga bisa akses data dari reducers dan melakukan action dispatch dari actions, dimaa container screendigunaka untuk menampilkan layar perhitungan setelah user memasukan inputan angka dan operator dari button yang ada di reducers

```
import React, { Component } from 'react'
import { connect } from 'react-redux'

class Screen extends Component {
    render(){
        return (
            <div className='row'>
                <div className='screen'>
                    {(!this.props.screen)?'0':this.props.screen}
                </div>
            </div>
        )
    }
}
function mapStateToProps (state) {
    return{
        screen: state.calculator.screen
    }
}
export default connect(mapStateToProps)(Screen)
```

Pada container button terdapat dispatch action creator yang dihubungkan pada mapDispatchToProps menggunakan parameter yang ada sehingga bisa akses action dengan props contohnya pada tag button

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { pressButton, pressEqual, pressReset } from '../actions/index'

class ButtonList extends Component{
    render(){
        return(
           <div className='row'>
               <div className='com-sm-9'>
                   <div className='row mb-2'>
                       <div className='col=sm-4 '>
                            <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton(7)}}>7</button>
                       </div>
                       <div className='col=sm-4 '>
                            <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton(8)}}>8</button>
```

```
                    </div>
                    <div className='col=sm-4 '>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton(9)}}>9</button>
                    </div>
                </div>
                <div className='row mb-2'>
                    <div className='col-sm-4 '>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton(4)}}>4</button>
                    </div>
                    <div className='col-sm-4 '>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton(5)}}>5</button>
                    </div>
                    <div className='col-sm-4 '>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton(6)}}>6</button>
                    </div>
                </div>
                <div className='row mb-2'>
                    <div className='col-sm-4'>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton(1)}}>1</button>
                    </div>
                    <div className='col-sm-4'>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton(2)}}>2</button>
                    </div>
                    <div className='col-sm-4'>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton(3)}}>3</button>
                    </div>
                </div>
                <div className='row mb-2'>
                    <div className='col-sm-4 '>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton('.')}}>.</button>
                    </div>
                    <div className='col-sm-4 '>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton(0)}}>0</button>
                    </div>
                    <div className='col-sm-4 '>
```

```
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressReset()}}>C</button>
                    </div>
                </div>
            </div>
            <div className='col-sm-3'>
                <div className='row mb-2'>
                    <div className='col-sm-6'>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton('+')}}>+</button>
                    </div>
                    <div className='col-sm-6'>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton('-')}}>-</button>
                    </div>
                </div>
                <div className='row mb-2'>
                    <div className='col-sm-6'>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton('')}}></button>
                    </div>
                    <div className='col-sm-6'>
                        <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressButton('/')}}>/</button>
                    </div>
                </div>
                <button className='btn-block btn-lg btn-
default' onClick={() => { this.props.pressEqual()}}>=</button>
            </div>
        </div>
      )
    }
}

function mapDispatchToProps (dispatch){
    return bindActionCreators({pressButton, pressEqual, pressReset}, dispatch)
}

export default connect(null, mapDispatchToProps)(ButtonList)
```

menambahkan container history.js untuk menampilkan history dimana perlu diakses data state dan action dalam satu container agar bisa dikoneksi dengan mapStateToProps dan mapDispatchToProps secara bersamaan

```
import React, { Component } from 'react'
import { connect } from 'react-redux'
import { bindActionCreators } from 'redux'
import { selectHistory, clearHistory } from '../Actions/index'

class History extends Component{
    render(){
        return(
            <div>
                <h4><i className='fa fa-clock' /> History Operator </h4>
                <ul className='list-group'>
                    {this.props.history.map((history, index) =>(
                        <li key ={index} className = 'list-group-item list-group-item-action' onClick={() => {this.props.selectHistory(index)}}>{`${history} = ${eval(history)}`}</li>
                        ))}
                </ul>
                <div className='mt-3 text-right'>
                    <button className = 'btn btn-danger' onClick ={ () => {this.props.clearHistory() }}><i className='fa fa-trash' /> Clear </button>
                </div>
            </div>
            )
        }
}

function mapStateToProps (state){
    return{
        history: state.calculator.history
    }
}

function mapDispatchToProps (dispatch){
    return bindActionCreators({ selectHistory, clearHistory }, dispatch)
}
export default connect(mapStateToProps, mapDispatchToProps)(History)
```

Menambahkan components app.js dimana components tidak membutuhkan akses ke reducers, dimana app js berfungsi memanggil countainer lain untuk disusun dalam tampilan aplikasi

```
import React, { Component } from 'react'
import Screen from '../containers/Screen'
import ButtonList from '../containers/ButtonList'
import History from '../containers/History'

class App extends Component{
  render(){
    return (
      <div className = 'row'>
        <div className = 'col-sm-12'>
          <h3 className = 'mt-4'> <i className = 'fa fa-calculator' /> Calculator React Redux
          </h3>
        </div>
        <div className = 'col-sm-8'>
          <Screen/>
        </div>
        <div className = 'col-sm-8'>
          <h5>Buttons</h5>
          <ButtonList />
        </div>
        <div className = 'col-sm-4'>
          <History />
        </div>
      </div>
    )
  }
}
export default App
```

Pada percobaanya banyak ditemukan kesalan atau error yang terjadi, walaupun saya sudah melakukan debugging sebisa saya namun tetap ditemukan kesalahan