

LAPORAN PENGEMBANGAN APLIKASI MOBILE

REACT-REDUX APLIKASI SEDERHANA

“Kalkulator sederhana”



Disusun oleh :

Putri Ayuni S NIM. 118140118

Kelas PAM – RB

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK ELEKTRO, INFORMATIKA DAN SISTEM FISIS

INSTITUT TEKNOLOGI SUMATERA

2021

PEMBAHASAN

React Native merupakan *framework mobile app development* yang memungkinkan pengembangan aplikasi secara *multi-platform* yaitu Android dan iOS. *framework* ini memungkinkan *developer* membuat aplikasi untuk berbagai *platform* dengan menggunakan basis kode yang sama.

Redux merupakan *back-end* yang berkolaborasi dengan **react native**. Adapun peran Redux dalam sebuah aplikasi adalah melakukan perubahan *state* yang dibutuhkan oleh setiap fungsional pada aplikasi. Untuk melakukan perubahan itu terdapat tiga komponen utama dari aplikasi Redux, yaitu **action, reducer, store**.

Berikut ini aplikasi sederhana yang menggunakan React Native yaitu kalkulator menggunakan redux dijabarkan :

1. Pertama buka Command Prompt dan Create Project : `npx create-react-app mycalcuRedux`
2. Kemudian tambahkan library Redux dan react-redux : `npm install - -save react-redux`
3. Selanjutnya lihat pada directory file dan buka menggunakan visualStudio
4. Buat folder Public dan create file index.html

```
index.html X
public > index.html > html
1 <html lang="en">
2 <head>
3   <meta charset="utf-8">
4   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
5   <meta name="theme-color" content="#000000">
6
7 <!-- Bootstrap CSS -->
8 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
9   integrity="sha384-MCw98/SF76GwEOngsV7Zt27NXFoaoApmYm81iuXoPkPfJFWJ8ERdnLPMO"
10  crossorigin="anonymous">
11 <!-- font awesome -->
12 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css"
13   integrity="sha384-mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU" crossorigin="anonymous">
14 <style>
15   .screen {
16     font-size: 40px;
17     text-align: right;
18     padding: 10px;
19     margin: 10px 0;
20     border: 1px solid #gainsboro;
21     border-radius: 3px;
22     width: 100%;
23     display: block;
24   }
25 </style>
26 <title>React Redux Calculator Apps</title>
```

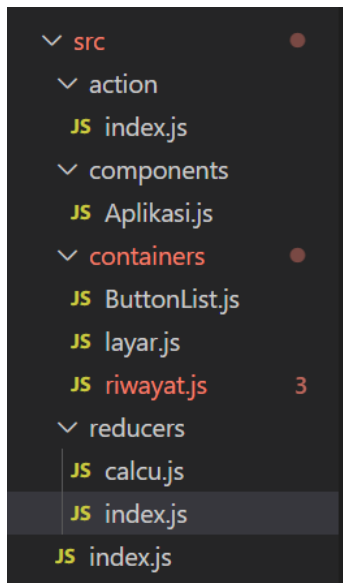
```

27 </head>
28 <body>
29 <noscript>
30 You need to enable JavaScript to run this app.
31 </noscript>
32 <div class="container" id="root"></div>
33 </body>
34 </html>

```

Index.html merupakan file berfungsi memperindah tampilan dari aplikasi sederhana. Dengan menggunakan css dan bootstrap.

5. Buat folder src.



Folder ini bertujuan untuk menampung components yang akan digunakan dalam aplikasi.

6. Selanjutnya buat folder action didalamnya dan tambahkan file index.js

```

JS index.js X
src > action > JS index.js > selectHistory
1  export const PRESS_BUTTON = 'PRESS_BUTTON'
2  export const PRESS_EQUALS = 'PRESS_EQUALS'
3  export const PRESS_RESET = 'PRESS_RESET'
4  export const SELECT_HISTORY = 'SELECT_HISTORY'
5  export const CLEAR_HISTORY = 'CLEAR_HISTORY'
6
7  export function pressButton (value){
8      console.log('PRESSED BUTTON : ', value)
9      return {
10         type : PRESS_BUTTON,
11         value
12     }
13 }
14
15 export function pressEqual () {
16     console.log('PRESSED BUTTON : equals')
17     return {
18         type : PRESS_EQUALS
19     }
20 }
21
22 export function pressReset () {
23     console.log('PRESSED RESET')
24     return {
25         type : PRESS_RESET
26     }
27 }
28
29 export function selectHistory (id) {
30     console.log('SELECT HISTORY', id)
31     return {
32         type : SELECT_HISTORY,
33         id
34     }
35 }
36
37 export function clearHistory() {
38     console.log('CLEAR_HISTORY')
39     return {
40         type : CLEAR_HISTROY
41     }
42 }

```

Untuk action/index.js ini merupakan penyimpanan creator action apa saja yang dilakukan untuk meminimalkan duplikasi kode. Caranya dengan mengirimkan action type dan payload ke reducers lalu diolah sesuai dengan state nya.

7. Lalu src/components dan didalamnya buat file aplikasi.js

```

15 index.js      15 Aplikasi.js X
src > components > 15 Aplikasi.js > Aplikasi > render
1  import React, { Component } from 'react'
2  import Screen from '../containers/layar'
3  import ButtonList from '../containers/ButtonList'
4  import History from '../containers/riwayat'
5
6  class Aplikasi extends Component {
7    render(){
8      return (
9        <div className = 'row'>
10         <div className = 'col-sm-12'>
11           <h3 className = 'mt-4'><i className = 'fa fa-calculator' /> Kalkulator React-redux</h3>
12         </div>
13         <div className = 'col-sm-8'>
14           <Screen />
15         </div>
16         <div className = 'col-sm-8'>
17           <h5>Tombol</h5>
18           <ButtonList />
19         </div>
20         <div className = 'col-sm-4'>
21           <History />
22         </div>
23       </div>
24     )
25   }
26 }
27
28 export default Aplikasi

```

Aplikasi.js merupakan komponen utama aplikasi (main program) yaitu file yang menyusun tampilan aplikasi dalam redux.

8. Lalu src/containers dan didalamnya buat file ButtonList.js

```

15 ButtonList.js •
src > containers > 15 ButtonList.js > ButtonList > render
1  import React, { Component } from 'react'
2  import { connect } from 'react-redux'
3  import { bindActionCreators } from 'redux'
4  import { pressButton, pressEqual, pressReset } from '../action/index'
5
6  class ButtonList extends Component{
7    render (){
8      return (
9        <div className = 'row'>
10         <div className = 'col-sm-9'>
11           <div className = 'row mb-2'>
12             <div className = 'col-sm-4'>
13               <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButto
14             </div>
15             <div className = 'col-sm-4'>
16               <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButto
17             </div>
18             <div className = 'col-sm-4'>
19               <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButton(9)
20             </div>
21           </div>
22         </div>
23         <div className = 'col-sm-4'>
24           <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButton(4)
25         </div>
26         <div className = 'col-sm-4'>
27           <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButton(5)
28         </div>

```

```

JS ButtonList.js
src > containers > JS ButtonList.js > ButtonList > render
29
30 <div className = 'col-sm-4'>
31 <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButton(6)
32 </div>
33 <div className = 'row mb-2'>
34 <div className = 'col-sm-4'>
35 <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButton(1)
36 </div>
37 <div className = 'col-sm-4'>
38 <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButton(2)
39 </div>
40 <div className = 'col-sm-4'>
41 <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButton(3)
42 </div>
43 </div>
44 <div className = 'row mb-2'>
45 <div className = 'col-sm-4'>
46 <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButton('
47 </div>
48 <div className = 'col-sm-4'>
49 <button className = 'btn-block btn-lg btn-default' onClick={() => { this.props.pressButton(0)
50 </div>
51 <div className = 'col-sm-4'>
52 <button className = 'btn-block btn-lg btn-danger' onClick={() => { this.props.pressButton()}}
53 </div>
54 </div>
55 </div>
56 <div className = 'col-sm-3'>
57 <div className = 'row mb-2'>
58 <div className = 'col-sm-6'>
59 <button className = 'btn-block btn-lg btn-primary' onClick={() => { this.props.pressButton('+
60 </div>
61 <div className = 'col-sm-6'>
62 <button className = 'btn-block btn-lg btn-primary' onClick={() => { this.props.pressButton('-
63 </div>
64 </div>
65 <div className = 'row mb-2'>
66 <div className = 'col-sm-6'>
67 <button className = 'btn-block btn-lg btn-primary' onClick={() => { this.props.pressButton('*
68 </div>
69 <div className = 'col-sm-6'>
70 <button className = 'btn-block btn-lg btn-primary' onClick={() => { this.props.pressButton('/
71 </div>
72 </div>
73 <button className = 'btn btn-block btn-success' onClick={() => { this.props.pressEqual() }}> = [//butt
74 </div>
75 </div>
76
77 )
78 }
79 }
80
81 function mapDispatchToProps (dispatch){
82   return bindActionCreators({ pressButton, pressEqual, pressReset }, dispatch)
83 }
84
85 export default connect(null, mapDispatchToProps)(ButtonList)

```

Pada file ButtonList.js ini merupakan file untuk memproses button/tombol yang digunakan dalam aplikasi kalkulator ini. Terdapat 'dispatch' yang berguna menggabungkan dengan mapDispatchToProps().

9. Lalu src/containers dan didalamnya buat file layar.js

```

1 import React, { Component } from 'react'
2 import { connect } from 'react-redux'
3
4 class Layer extends Component {
5   render(){
6     return (
7       <div className = 'row'>
8         <div className = 'Layar'>
9           {{
10             !this.props.Layar
11             ?'0':this.props.Layar}}
12         </div>
13       </div>
14     )
15   }
16 }
17
18 function mapStateToProps(state) {
19   return {
20     Layar: state.calcu.Layar
21   }
22 }
23
24 export default connect(mapStateToProps)(Layar)

```

File ini merupakan smart component yang melakukan koneksi (komunikasi) langsung kepada application state. Fungsi dari layar.js ini untuk menampilkan layer perhitungan setelah pengguna menginput nilai dan operator terhadap perhitungan pada kalkulator ini. Dengan penyimpanan data yang membutuhkan mapStateToProps.

10. Lalu src/containers dan didalamnya buat file riwayat.js

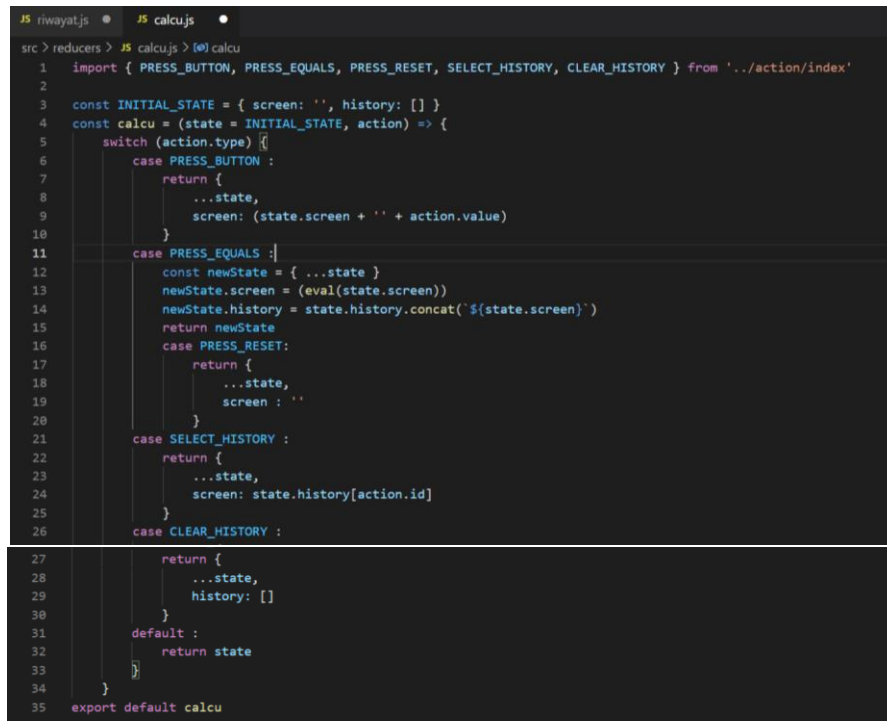
```

JS riwayat.js
src > containers > JS riwayat.js > Riwayat > render
1 import React, { Component } from 'react'
2 import { connect } from 'react-redux'
3 import { bindActionCreators } from 'redux'
4 import { selectHistory, clearHistory } from '../action/index'
5
6 class Riwayat extends Component {
7   render () {
8     return (
9       <div>
10         <h4 <i className = 'fa fa-clock' /> Riwayat Operator </h4>
11         <ul className = 'list-group'>
12           {{ this.props.Riwayat.map(Riwayat, index) => (
13             <li key={index} className = 'list-group-item list-group-item-action' onClick = {{() => {
14               this.props.selectHistory(index) }} > {{`${riwayat} = ${eval(riwayat)}`}} </li>
15             )}}
16         </ul>
17         <div className = 'mt-3 text-right'>
18           <button className = 'btn btn-danger' onClick={{() => {
19             this.props.clearHistory()
20           }}} <i className = 'fa fa-trash' /> Hapus </button>
21         </div>
22       </div>
23     )
24   }
25 }
26
27 function mapStateToProps (state){
28   return {
29     Riwayat: state.calcu.Riwayat
30   }
31 }
32
33 function mapDispatchToProps (dispatch){
34   return bindActionCreators({selectHistory , clearHistory}, dispatch)
35 }
36
37 export default connect(mapStateToProps, mapDispatchToProps)(History)

```

File ini berfungsi menampilkan riwayat dari penghitungan. Untuk mengakses `mapStateToProps` dilakukan secara bersamaan.

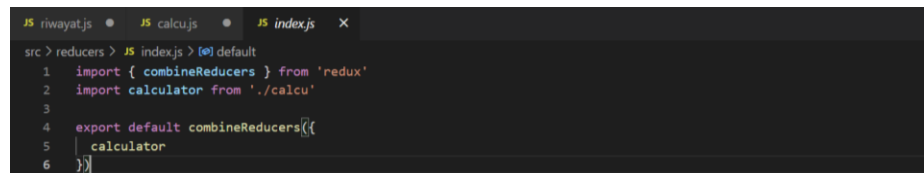
11. Lalu `src/reducers` dan didalamnya buat file `calcu.js`



```
1  import { PRESS_BUTTON, PRESS_EQUALS, PRESS_RESET, SELECT_HISTORY, CLEAR_HISTORY } from '../action/index'
2
3  const INITIAL_STATE = { screen: '', history: [] }
4  const calcu = (state = INITIAL_STATE, action) => {
5    switch (action.type) {
6      case PRESS_BUTTON :
7        return {
8          ...state,
9          screen: (state.screen + '' + action.value)
10        }
11      case PRESS_EQUALS :
12        const newState = { ...state }
13        newState.screen = (eval(state.screen))
14        newState.history = state.history.concat(`${state.screen}`)
15        return newState
16      case PRESS_RESET:
17        return {
18          ...state,
19          screen : ''
20        }
21      case SELECT_HISTORY :
22        return {
23          ...state,
24          screen: state.history[action.id]
25        }
26      case CLEAR_HISTORY :
27        return {
28          ...state,
29          history: []
30        }
31      default :
32        return state
33    }
34  }
35  export default calcu
```

`Calcu.js` merupakan sebuah reducers pada aplikasi sederhana ini. Dengan inisiasi state dan `buttonList` terlebih dahulu.

12. Lalu `src/reducers` dan didalamnya buat file `index.js`



```
1  import { combineReducers } from 'redux'
2  import calculator from './calcu'
3
4  export default combineReducers({
5    calculator
6  })
```

Setelah melakukan penggabungan reducer `calcu.js` ke dalam `rootReducer`.

13. Selanjutnya didalam folder `src` buat file `index.js`


```
JS riwayatjs ● JS calcujs ● JS indexjs X
src > JS indexjs > ...
1 import React from 'react'
2 import { render } from 'react-dom'
3 import { createStore } from 'redux'
4 import { Provider } from 'react-redux'
5 import App from '../components/Aplikasi'
6 import rootReducer from './reducers/index'
7
8 const store = createStore(rootReducer)
9
10 render(() => {
11   <Provider store={store}>
12     <App />
13   </Provider>,
14   document.getElementById('root')
15 })
```

File ini berfungsi mendeklarasikan Aplikasi.js untuk dalam di running dan berjalan.