

PEMROGRAMAN FRAMEWORK

MODUL 7: HOC

Oleh:

Ashfin Ramadhandy E / 1741720215 TI - 3A / 05



**PROGRAM STUDI D4 TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI POLITEKNIK
NEGERI MALANG**

2020

Praktikum :

1. Buat dan install redux
npx create-react-app contact-redux
install -g create-react-app
npm start
2. Install Form
npm install --save redux react-redux redux-form
3. Ubah file index.html pada folder /public/index.html

```
ReduxDanHoc > contact-redux > public > index.html
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="utf-8" />
5      <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6      <meta name="viewport" content="width=device-width, initial-scale=1" />
7      <meta name="theme-color" content="#000000" />
8
9      <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
10     <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
11     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.6.2/css/bulma.min.css">
12
13
14     <title>React App</title>
15   </head>
16   <body>
17     <noscript>You need to enable JavaScript to run this app.</noscript>
18     <div id="root"></div>
19     <!--
20       This HTML file is a template.
21       If you open it directly in the browser, you will see an empty page.
22
23       You can add webfonts, meta tags, or analytics to this file.
24       The build step will place the bundled scripts into the <body> tag.
25
26       To begin the development, run `npm start` or `yarn start`.
27       To create a production bundle, use `npm run build` or `yarn build`.
28     -->
29   </body>
```

4. Ubah file App.js pada folder /src/App.js

ReduxDanHoc > contact-redux > src > JS App.js > [⌘] default

```
1  import React, { Component } from 'react';
2  import { reduxForm, Field } from 'redux-form';
3  import logo from './logo.svg';
4  import './App.css';
5
6  let SignInForm = props => {
7    const { handleSubmit } = props;
8    return <form onSubmit={handleSubmit} className="form">
9      <div className="field">
10        <div className="control">
11          <Field name="firstName" component={renderField} type="text" label="First Name" />
12        </div>
13      </div>
14
15      <div className="field">
16        <div className="control">
17          <Field name="lastName" component={renderField} type="text" label="Last Name" />
18        </div>
19      </div>
20      <div className="field">
21        <div className="control">
22          <Field name="email" component={renderField} type="email" label="Email Address" />
23        </div>
24      </div>
25
26      <div className="field">
27        <div className="control">
28          <Field name="age" component={renderField} type="number" label="Age" />
29        </div>
30      </div>
```

```

31     <div className="field">
32         <div className="control">
33             <label className="label">Proficiency</label>
34             <div className="select">
35                 <Field className="input" name="proficiency" component="select">
36                     <option />
37                     <option value="beginner">Beginner Dev</option>
38                     <option value="intermediate">Intermediate Dev</option>
39                     <option value="expert">Expert Dev</option>
40                 </Field>
41             </div>
42         </div>
43     </div>
44     <div className="field">
45         <div className="control">
46             <label className="label">Gender</label>
47             <label className="radio">
48                 <Field name="gender" component="input" type="radio" value="male" />
49                 {' '}
50             Male
51         </label>
52         <label className="radio">
53             <Field name="gender" component="input" type="radio" value="female" />
54             {' '}
55             Female
56         </label>

```

```

57 |         </div>
58 |     </div>
59 |     <div className="field">
60 |         <div className="control">
61 |             <label className="checkbox">
62 |                 <Field name="saveDetails" id="saveDetails" component="input" type="checkbox" />
63 |                 Save Details
64 |             </label>
65 |         </div>
66 |     </div>
67 |     <div className="field">
68 |         <div className="control">
69 |             <label className="label">Message</label>
70 |             <Field className="textarea" name="message" component="textarea" />
71 |         </div>
72 |     </div>
73 |     <div className="field">
74 |         <div className="control">
75 |             <button className="button is-link">Submit</button>
76 |         </div>
77 |     </div>
78 | </form>;
79 | };
80 | const validate = val => {
81 |     const errors = {};
82 |     if (!val.firstName) {
83 |         console.log('First Name is required');
84 |         errors.firstName = 'Required';
85 |     }

```

```

86   if (!val.lastName) {
87     console.log('Last Name is required');
88     errors.lastName = 'Required';
89   }
90   if (!val.email) {
91     console.log('email is required');
92     errors.email = 'Required';
93   } else if (!/^.+@.+$/.test(val.email)) {
94     console.log('email is invalid');
95     errors.email = 'Invalid email address';
96   }
97   if (!val.age) {
98     errors.age = 'Required'
99   } else if (isNaN(Number(val.age))) {
100     errors.age = 'Must be a number'
101   } else if (Number(val.age) < 18) {
102     errors.age = 'Sorry, you must be at least 18 years old'
103   }
104   return errors;
105 };
106 const renderField = ({ input, label, type, meta: { touched, error, warning } }) => (
107   <div>
108     <div className="control">
109       <label className="field">{label}</label>
110       <input className="input" {...input} placeholder={label} type={type} />
111       {touched && ((error && <span>{error}</span>) || (warning && <span>{warning}</span>))}
112     </div>
113   </div>
114 )

```

```

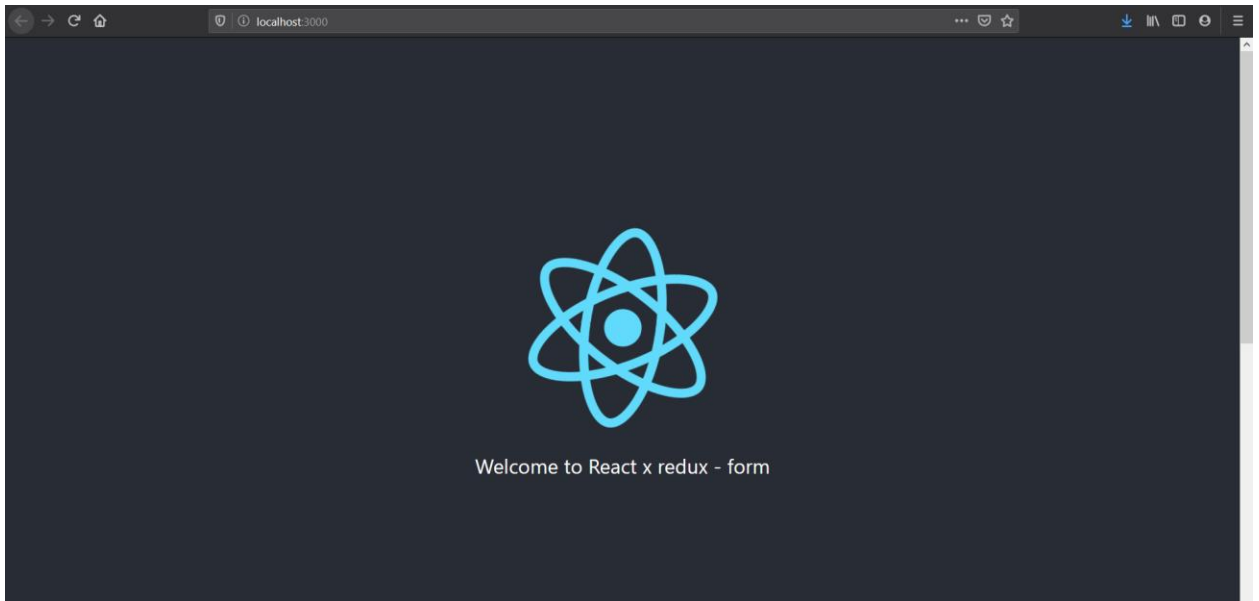
115   SignInForm = reduxForm({
116     form: 'signIn',
117     validate,
118   })(SignInForm);
119   class App extends Component {
120     handleSignIn = values => {
121       console.log(values);
122     };
123     render() {
124       return (
125         <div className="App">
126           <header className="App-header">
127             <img src={logo} className="App-logo" alt="logo" />
128             <h1 className="App-title">Welcome to React x redux-form</h1>
129           </header>
130           <div className="container">
131             <p className="App-intro">
132               Contact Form
133             </p>
134             <SignInForm onSubmit={this.handleSignIn} />
135           </div>
136         </div>
137       );
138     }
139   }
140   export default App;

```

5. Ubah file /src/index.js

```
ReduxDanHoc > contact-redux > src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import { createStore, combineReducers } from 'redux';
4  import { Provider } from 'react-redux';
5  import { reducer as formReducer } from 'redux-form';
6  import './index.css';
7  import App from './App';
8  import * as serviceWorker from './serviceWorker';
9
10 const rootReducer = combineReducers({
11   form: formReducer,
12 });
13
14 const store = createStore(rootReducer);
15
16 ReactDOM.render(
17   <Provider store={store}>
18     <App />
19   </Provider>,
20   document.getElementById('root')
21 );
22
23 // If you want your app to work offline and load faster, you can change
24 // unregister() to register() below. Note this comes with some pitfalls.
25 // Learn more about service workers: https://bit.ly/CRA-PWA
26 serviceWorker.register();
```

6. Run



First Name

Last Name

Email Address

Age

Proficiency

Gender

☐ Male ☐ Female

☐ Save Details

Message

Submit

```
{...}
  age: "21"
  email: "ashfinramadhandy10@gmail.com"
  firstName: "Ashfin"
  gender: "male"
  lastName: "Ramadhandy"
  proficiency: "beginner"
  saveDetails: true
  > <prototype>: Object { ... }
```

>>