PEMROGRAMAN FRAMEWORK

MODUL 7: HOC

Oleh:

IRHAM RIZALDY PRATAMA / 1741720103
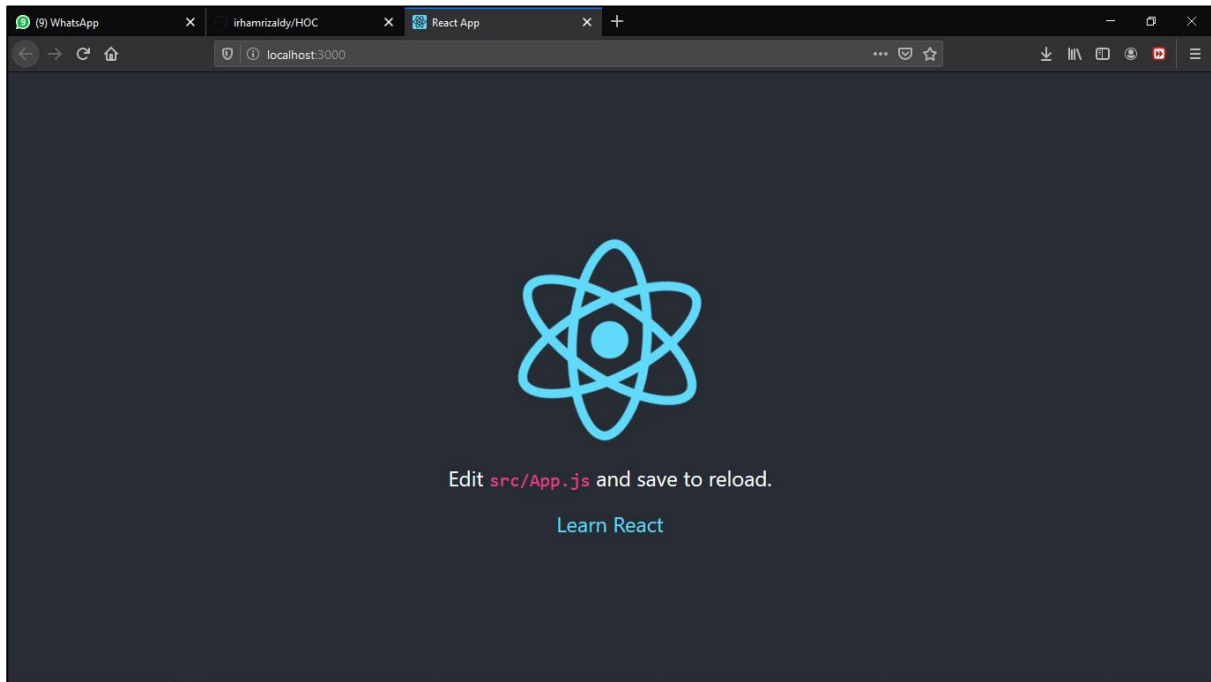
TI - 3A / 14

PROGRAM STUDI D4 TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

1. Buat dan install redux
   npx create-react-app contact-redux
   npm start

2. Install Form
   npm install –save redux react-redux redux-form

3. Ubah file index.html pada folder /public/index.html

```html
<!DOCTYPE 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">
  <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
  <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.6.2/css/bulma.min.css">
  <title>React App</title>
</head>

<body>
  <noscript>
    You need to enable JavaScript to run this app.
  </noscript>
  <div id="root"></div>
</body>

</html>
```

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

```javascript
import React, { Component } from 'react';
import { reduxForm, Field } from 'redux-form';
import logo from './logo.svg';
import './App.css';
let SignInForm = props => {
  const { handleSubmit } = props;
  return <form onSubmit={handleSubmit} className="form">
    <div className="field">
      <div className="control">
        <Field name="firstName" component={renderField} type="text" label="First Name" />
      </div>
    </div>
    <div className="field">
      <div className="control">
        <Field name="lastName" component={renderField} type="text" label="Last Name" />
      </div>
    </div>
    <div className="field">
      <div className="control">
        <Field name="email" component={renderField} type="email" label="Email Address" />
      </div>
    </div>
    <div className="field">
      <div className="control">
        <Field name="age" component={renderField} type="number" label="Age" />
      </div>
    </div>
    <div className="field">
      <div className="control">
        <label className="label">Proficiency</label>
        <div className="select">
          <Field className="input" name="proficiency" component="select">
            <option />
            <option value="beginner">Beginner Dev</option>
```

```jsx
          <option value="intermediate">Intermediate Dev</option>
          <option value="expert">Expert Dev</option>
        </Field>
      </div>
    </div>
    <div className="field">
      <div className="control">
        <label className="label">Gender</label>
        <label className="radio">
          <Field name="gender" component="input" type="radio" value="male" />
          {' '}
Male
</label>
        <label className="radio">
          <Field name="gender" component="input" type="radio" value="female" />
          {' '}
Female
</label>
      </div>
    </div>
    <div className="field">
      <div className="control">
        <label className="checkbox">
          <Field name="saveDetails" id="saveDetails" component="input" type="checkbox" />
Save Details
</label>
      </div>
    </div>
    <div className="field">
      <div className="control">
        <label className="label">Message</label>
        <Field className="textarea" name="message" component="textarea" />
      </div>
```

```jsx
    <div className="field">
      <div className="control">
        <label className="label">Message</label>
        <Field className="textarea" name="message" component="textarea" />
      </div>
    </div>
    <div className="field">
      <div className="control">
        <button className="button is-link">Submit</button>
      </div>
    </div>
  </form>;
};
const validate = val => {
  const errors = {};
  if (!val.firstName) {
    console.log('First Name is required');
    errors.firstName = 'Required';
  }
  if (!val.lastName) {
    console.log('Last Name is required');
    errors.lastName = 'Required';
  }
  if (!val.email) {
    console.log('email is required');
    errors.email = 'Required';
  } else if (!/^.+@.+$/i.test(val.email)) {
    console.log('email is invalid');
    errors.email = 'Invalid email address';
  }
  if (!val.age) {
    errors.age = 'Required'
  } else if (isNaN(Number(val.age))) {
```

```
  <div>
    <div className="control">
      <label className="field">{label}</label>
      <input className="input" {...input} placeholder={label} type={type} />
      {touched && ((error && <span>{error}</span>) || (warning && <span>{warning}</span>))}
    </div>
  </div>
)
SignInForm = reduxForm({
  form: 'signIn',
  validate,
})(SignInForm);
class App extends Component {
  handleSignIn = values => {
    console.log(values);
  };
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h1 className="App-title">Welcome to React x redux-form</h1>
        </header>
        <div className="container">
          <p className="App-intro">
            Contact Form
          </p>
          <SignInForm onSubmit={this.handleSignIn} />
        </div>
      </div>
    );
  }
}
export default App;
```

5. Ubah file /src/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();
27
```

6. Run hasil

Object { firstName: "Irham", lastName: "Rizaldy", email: App.js:118
"irhamrizaldy@email.com", age: "20", proficiency: "beginner",
gender: "male", saveDetails: true, message: "Hello HOC!" }