

TUGAS PERORANGAN/INDIVIDU

LAPORAN MINGGU-8

Disusun sebagai

MATA KULIAH : Pemrograman Berbasis Framework

Oleh :

Mery Kristiani

1741720104

3B/17



PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

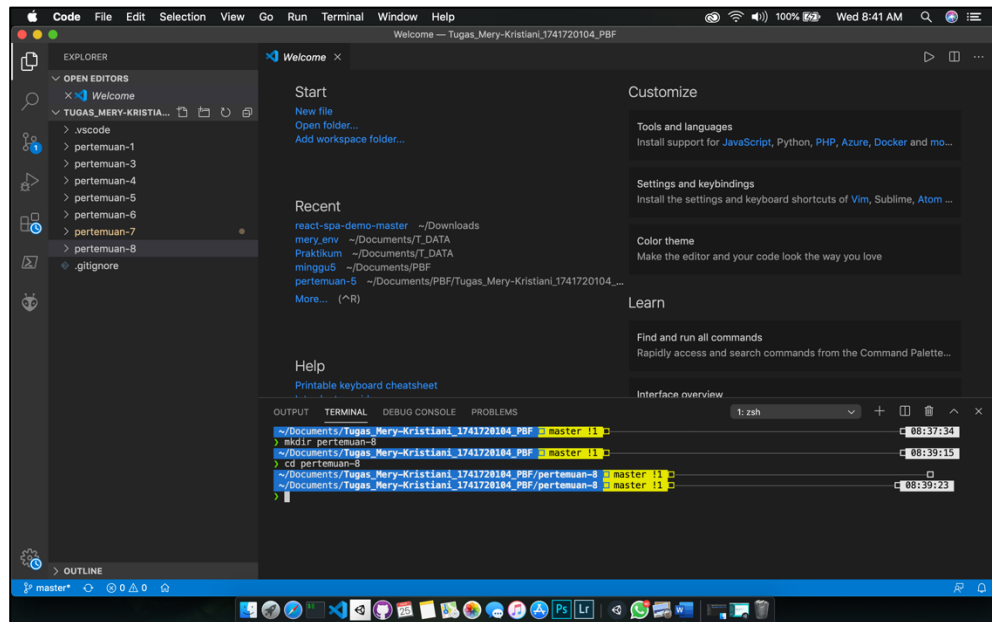
POLITEKNIK NEGERI MALANG

2020

Praktikum

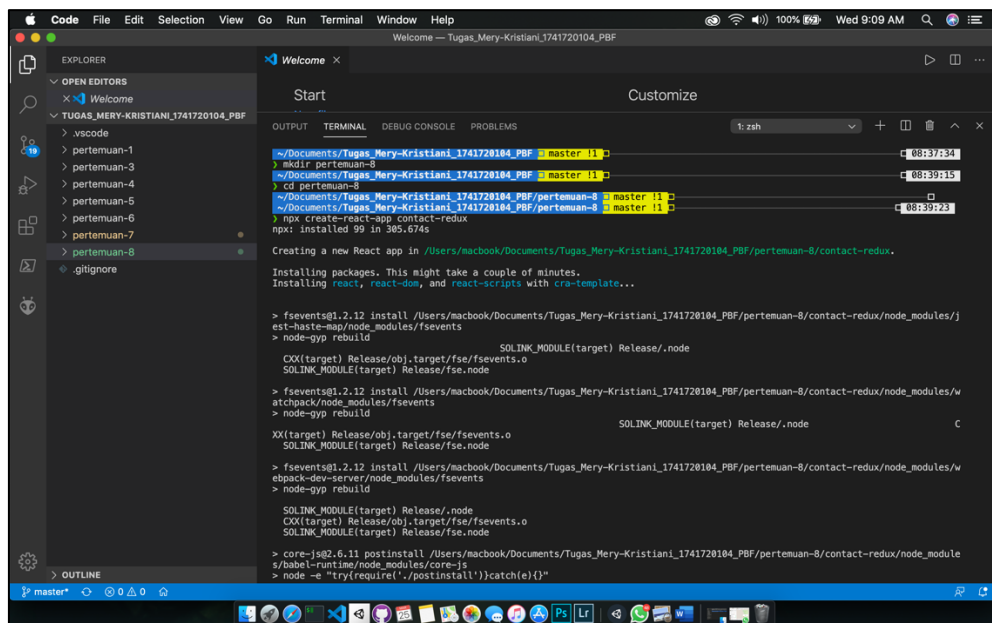
1. Langkah pertama

Bikin folder baru terlebih dahulu dengan perintah **“mkdir pertemuan-8”** dibawah ini. Selanjutnya masuk ke folder yang sudah dibuat dengan perintah **“cd pertemuan-8”**



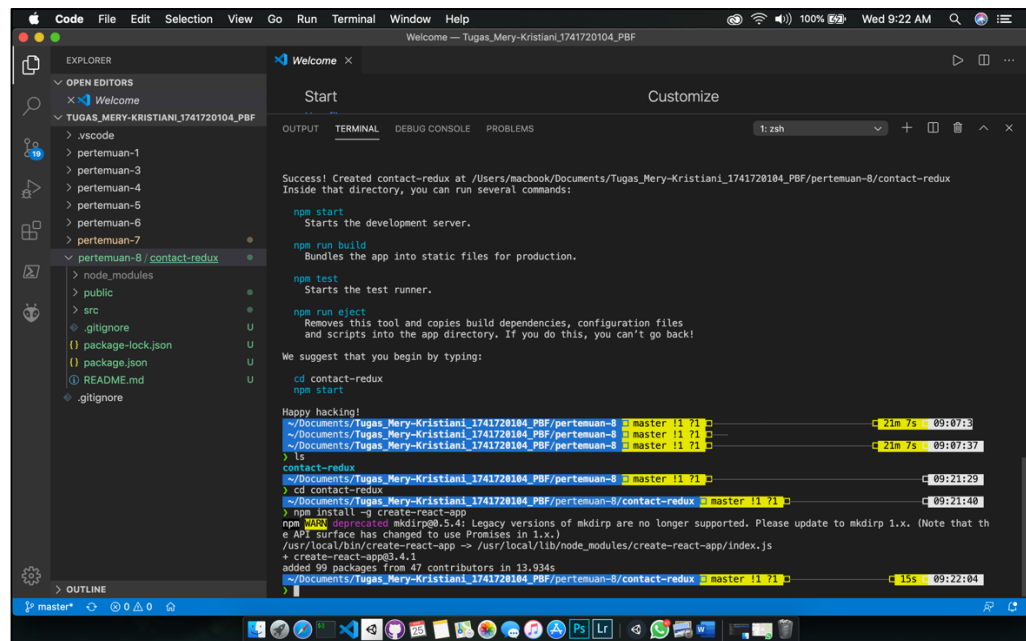
2. Langkah kedua

Masukkan perintah **“npx create-react-app contact-redux”**



3. Langkah ketiga

Masukkan perintah **"npm install -g create-react-app"**



The screenshot shows the VS Code interface with a terminal window open. The terminal displays the output of the command `npm install -g create-react-app`. The output indicates that the installation was successful and provides instructions on how to start the development server. The Explorer panel on the left shows the project structure, including the `contact-redux` directory.

```
npm install -g create-react-app

Success! Created contact-redux at /Users/macbook/Documents/Tugas_Mery-Kristiani_1741720104_PBF/pertemuan-8/contact-redux
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

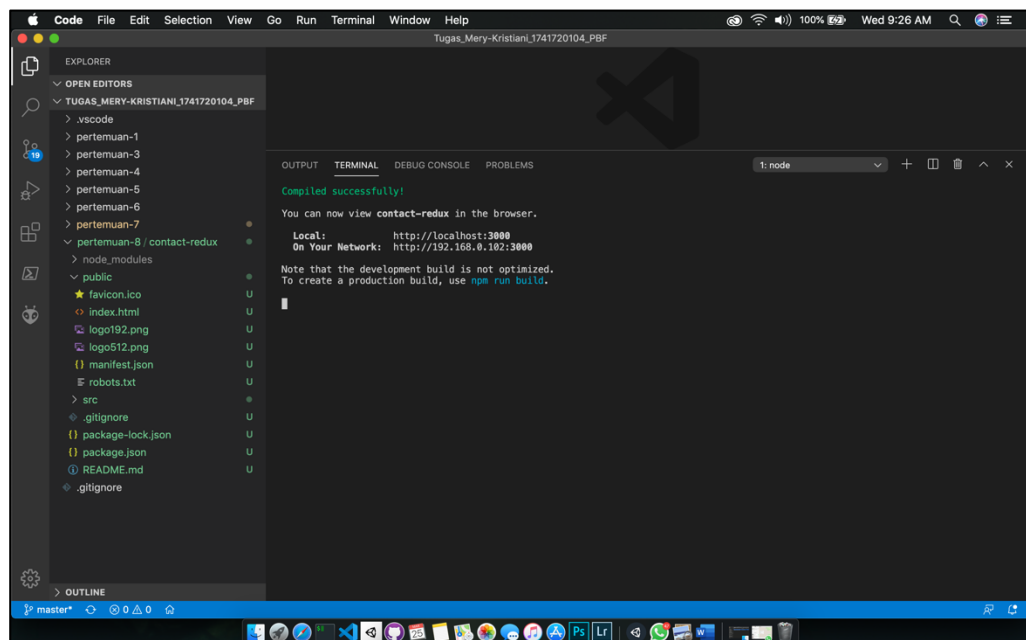
  cd contact-redux
  npm start

Happy hacking!

~/Documents/Tugas_Mery-Kristiani_1741720104_PBF/pertemuan-8 master !1 71 | 21m 7s | 09:07:25
~/Documents/Tugas_Mery-Kristiani_1741720104_PBF/pertemuan-8 master !1 71 | 21m 7s | 09:07:37
~/Documents/Tugas_Mery-Kristiani_1741720104_PBF/pertemuan-8 master !1 71 | 09:21:29
~/Documents/Tugas_Mery-Kristiani_1741720104_PBF/pertemuan-8 master !1 71 | 09:21:40
~/Documents/Tugas_Mery-Kristiani_1741720104_PBF/pertemuan-8 master !1 71 | 15s | 09:22:04
```

4. Langkah keempat

Jalankan program tersebut dengan perintah **"npm start"**



The screenshot shows the VS Code interface with a terminal window open. The terminal displays the output of the command `npm start`. The output indicates that the application has been compiled successfully and is now running on a local server. The Explorer panel on the left shows the project structure, including the `contact-redux` directory.

```
npm start

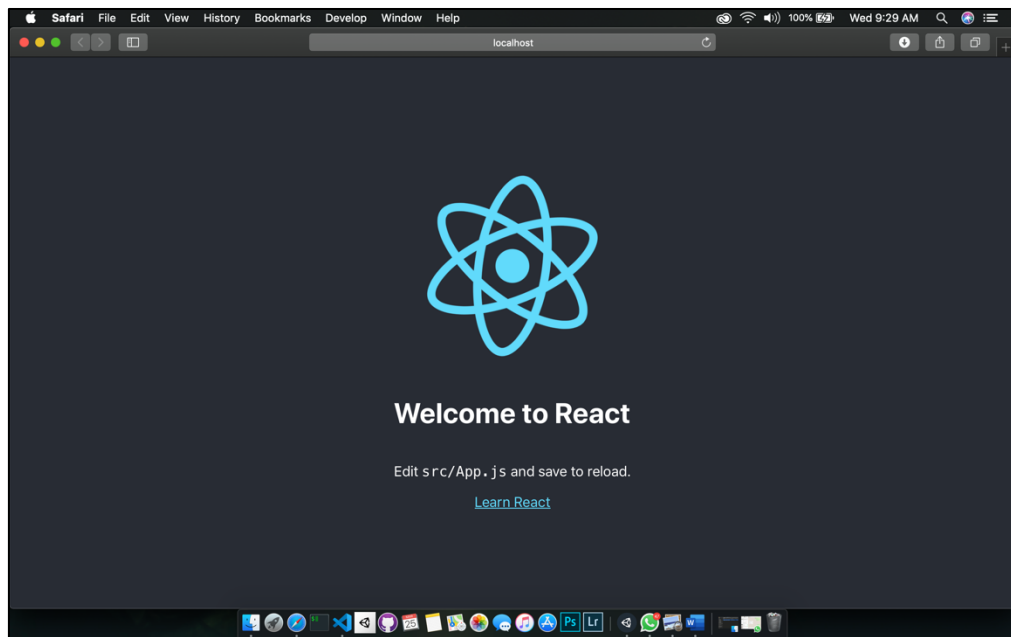
Compiled successfully!

You can now view contact-redux in the browser.

  Local:    http://localhost:3000
  On Your Network: http://192.168.0.102:3000

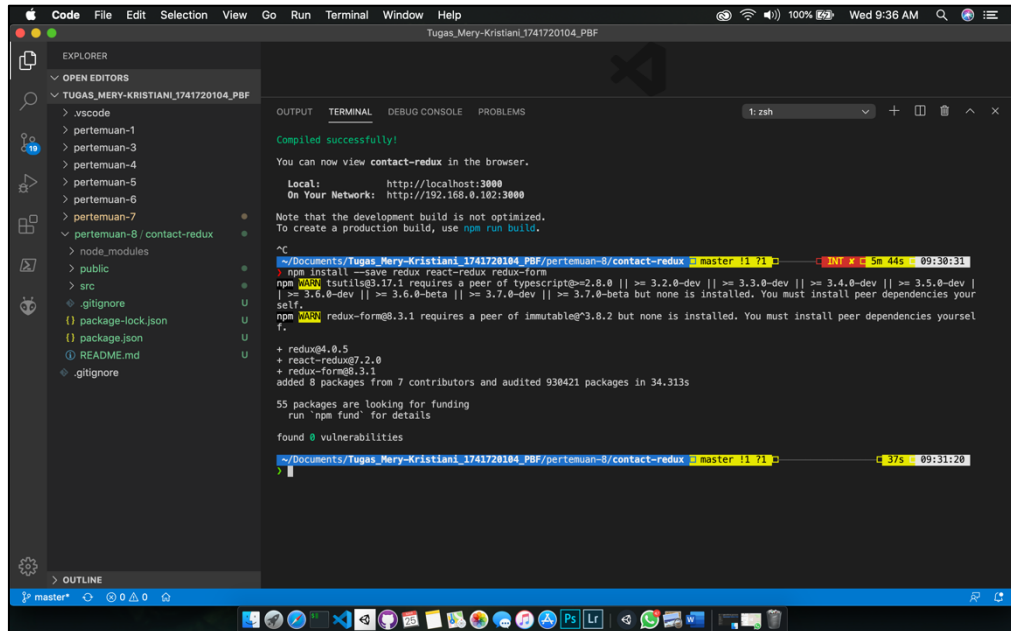
Note that the development build is not optimized.
To create a production build, use npm run build.
```

Hasil



5. Langkah kelima

Masukkan perintah **"npm install --save redux react-redux redux-form"**



6. Langkah keenam

Ubah source code di file **index.html** yang berada di folder **/public/index.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>
```

7. Langkah ketujuh

Ubah source code di file **App.js** yang berada di folder **/src/App.js**

```
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>
  </form>
```

```

<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>
        <option value="intermediate">Intermediate Dev</option>
        <option value="expert">Expert Dev</option>
      </Field>
    </div>
  </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>
  </div>

  <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 (!/^.+@.+$/.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))) {
    errors.age = 'Must be a number'
  } else if (Number(val.age) < 18) {
    errors.age = 'Sorry, you must be at least 18 years old'
  }
  return errors;
};

const renderField = ({ input, label, type, meta: { touched, error, warning } }) => (
  <div>
    <div className="control">
      <label className="field">{label}</label>
      <input className="input" {...input} placeholder={label} type={type}/>
    </div>
  </div>
);

```

```

        {touched && ((error && <span>{error}</span>) || (warning &&
<span>{warning}</span>))}
      </div>
    </div>
  )

  SignInForm = reduxForm({
    form: 'signIn',
    validate,
  })(SignInForm);

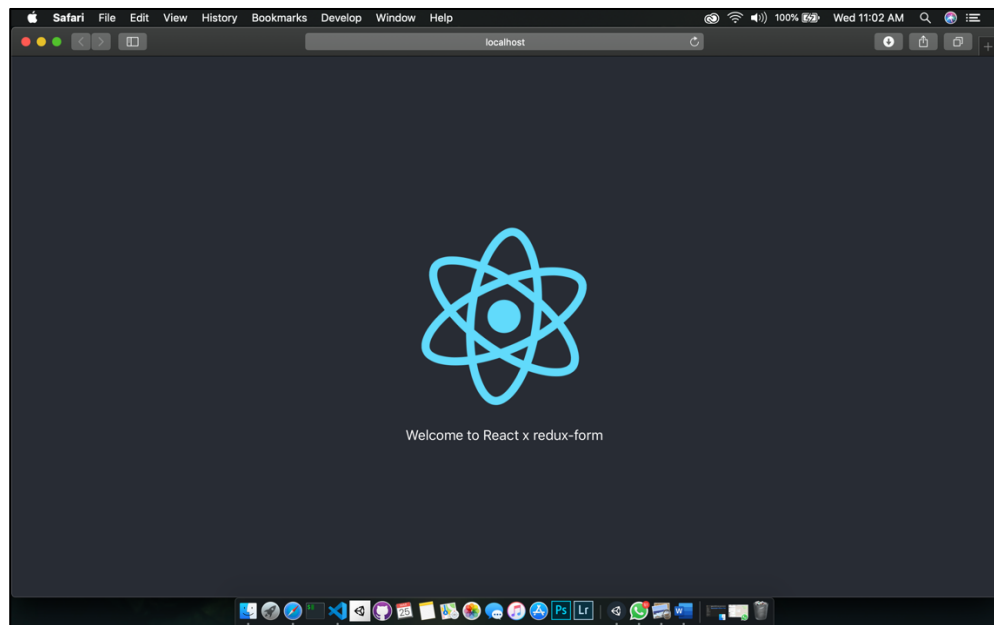
  class App extends Component {
    handleSignIn = values => {
      console.log(values);
    };

    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;

```


8. Hasil Praktikum



A screenshot of a Safari web browser window showing a contact form. The address bar shows 'localhost'. The form is titled 'Contact Form' and is set against a dark blue background. The form fields are as follows:

- First Name**: A text input field with the placeholder text 'First Name'.
- Last Name**: A text input field with the placeholder text 'Last Name'.
- Email Address**: A text input field with the placeholder text 'Email Address'.
- Age**: A text input field with the placeholder text 'Age'.
- Proficiency**: A dropdown menu.
- Gender**: Radio buttons for 'Male' and 'Female'.
- Save Details**: A checkbox.
- Message**: A large text area for a message.
- Submit**: A blue button at the bottom left of the form.

The browser's menu bar and status bar are also visible, showing the time 'Wed 11:02 AM'.