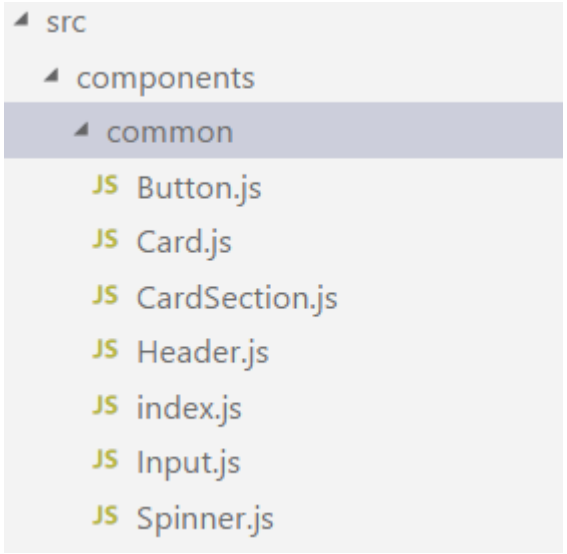
React Native Development

# Tech_Stack Apps Walkthrough

Purwadhika
Startup and Coding School

# Step 1

- Copy semua common component yang kita punya diproject lalu ke project yang sekarang

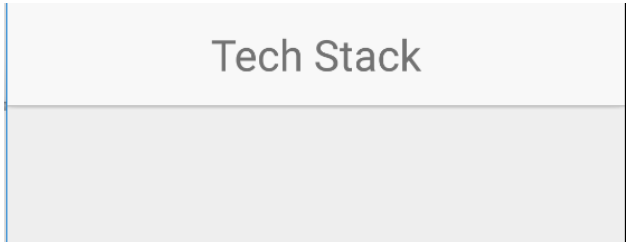



Purwadhika
Startup and Coding School

# Step 2

- Hapus semua code yang ada di App.js, dan diganti dengan ini :

```
import React from 'react';
import { View } from 'react-native';
import { Header } from './src/components/common';

const App = () => {
    return (
        <View>
          <Header headerText="Tech Stack" />
        </View>
    );
};

export default App;
```




Purwadhika
Startup and Coding School

# Step 3

- Install Redux dan React Redux dengan mengetik ini di Integrated Terminal
- npm install --save redux react-redux

# Step 4

- Import createStore dari Redux dan Provider dari React-Redux di App.js

```
import React from 'react';
import { View } from 'react-native';
import { Provider } from 'react-redux';
import { createStore } from 'redux';
```

- Kemudian pasang

```
return (
    <Provider store={createStore}>
      <View>
        <Header headerText="Tech Stack" />
      </View>
    </Provider>
);
```

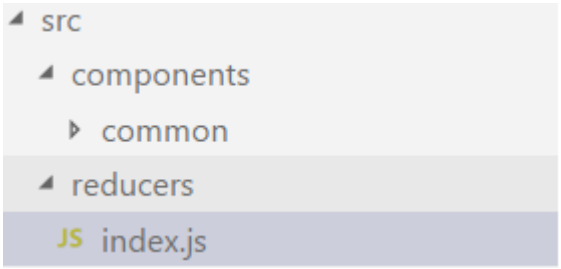
Warning: Failed prop type: Invalid prop `store` of type `function` supplied to `Provider`, expected `object`....

Warning: Failed child context type: Invalid child context `store` of type `function` supplied to...

Purwadhika
Startup and Coding School

# Step 5

- Kemudian buat folder reducers didalam folder src dan buat file index.js didalam folder reducers




- Isi index.js nya

```
import { combineReducers } from 'redux';

export default combineReducers({
    libraries: () => []
});
```

Purwadhika
Startup and Coding School

# Step 5 (continue)

- Import reducers ke App.js

```
import reducers from './src/reducers';
```

- Kemudian pasang

```
return (
    <Provider store={createStore(reducers)}>
      <View>
        <Header headerText="Tech Stack" />
      </View>
    </Provider>
);
```

- Sekarang warning diemulatornya akan hilang karena createStore sudah kita kasih reducer

Purwadhika
Startup and Coding School

# Step 6

- Kemudian buat file LibraryReducer.js didalam folder reducers isinya cuma

```
export default () => [];
```

- Kemudian ubah isi index.js di folder reducers jadi

```
import { combineReducers } from 'redux';
import LibraryReducer from './LibraryReducer';

export default combineReducers({
    libraries: LibraryReducer
});
```

# Step 7

- Kemudian buat file LibraryList.json di folder reducers juga, isinya array of objects

```
[
    {
        "id": 0,
        "title": "Webpack",
        "description": "Webpack is a module bundler. It packs CommonJ
    },
    {
        "id": 1,
        "title": "React",
        "description": "React makes it painless to create interactive
    },
    {
        "id": 2,
        "title": "Redux",
        "description": "Redux is a predictable state container for Ja
    },
    {
        "id": 3,
        "title": "React-Redux",
        "description": "React-Redux is the official set of bindings b
    },
```

Purwadhika
Startup and Coding School

# Step 7 (continue)

- Import dan pakai LibraryList.json tersebut di LibraryReducer

```
import data from './LibraryList.json';

export default () => data;
```

# Step 8

- Kemudian buat file component baru LibraryList.js difolder components

```
import React, { Component } from 'react';
import { View, Text } from 'react-native';
import { connect } from 'react-redux';

class LibraryList extends Component {
    render() {
        console.log(this.props);
        return (
            <View>
                <Text>Library List</Text>
            </View>
        );
    }
}

const mapStateToProps = state => {
    return { libraries: state.libraries };
};

export default connect(mapStateToProps)(LibraryList);
```

Purwadhika
Startup and Coding School

# Step 9

- Kemudian import dan pasang LibraryList di App.js

```
import LibraryList from './src/components/LibraryList';

const App = () => {
    return (
        <Provider store={createStore(reducers)}>
          <View>
            <Header headerText="Tech Stack" />
            <LibraryList />
          </View>
```

- Muncul di console browser seperti itu =>


Purwadhika
Startup and Coding School

# Step 10

- Ubah import di LibraryList.js yang import dari react-native jadi ini :

```
import { ListView } from 'react-native';
```

- Kemudian ubah isi function render() jadi ini

```
return (
    <ListView
        dataSource={this.dataSource}
    />
);
```

- Kemudian tambahkan lifeCycleMethod componentWillMount untuk mengisi this.dataSource (dataSource dari class LibraryList, sudah tersedia karena turunan dari class Component di React)

```
componentWillMount() {
    const ds = new ListView.DataSource({
        rowHasChanged: (r1, r2) => r1 !== r2
    });

    this.dataSource = ds.cloneWithRows(this.props.libraries);
}
```

Purwadhika
Startup and Coding School

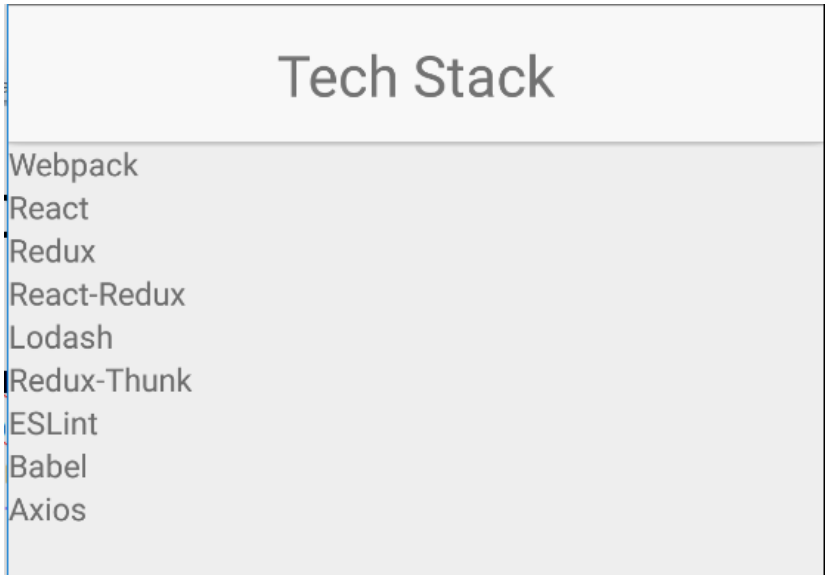
# Step 11

- Kemudian tambahkan function renderRow() diantara render() dan componentWillMount()

```
renderRow(library) {
    return <Text>{library.title}</Text>;
}
```

- Lalu pasang ke ListViewnya

```
<ListView
    dataSource={this.dataSource}
    renderRow={this.renderRow}
/>
```




Purwadhika
Startup and Coding School

# Step 12

- Kemudian buat file ListItem.js difolder yang sama dengan LibraryList.js

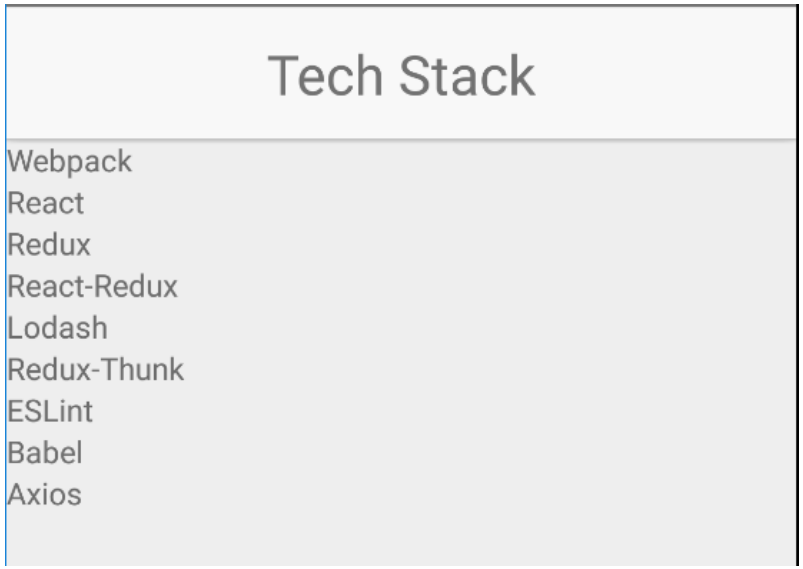
```
import React, { Component } from 'react';
import { Text, View } from 'react-native';

class ListItem extends Component {
    render() {
        const { title } = this.props.library;

        return (
            <View>
                <Text>{title}</Text>
            </View>
        );
    }
}

export default ListItem;
```

Purwadhika
Startup and Coding School

# Step 12 (continue)

- Kemudian import dan pasang ListItem di LibraryList

```
import ListItem from './ListItem';

renderRow(library) {
    return <ListItem library={library} />;
}
```

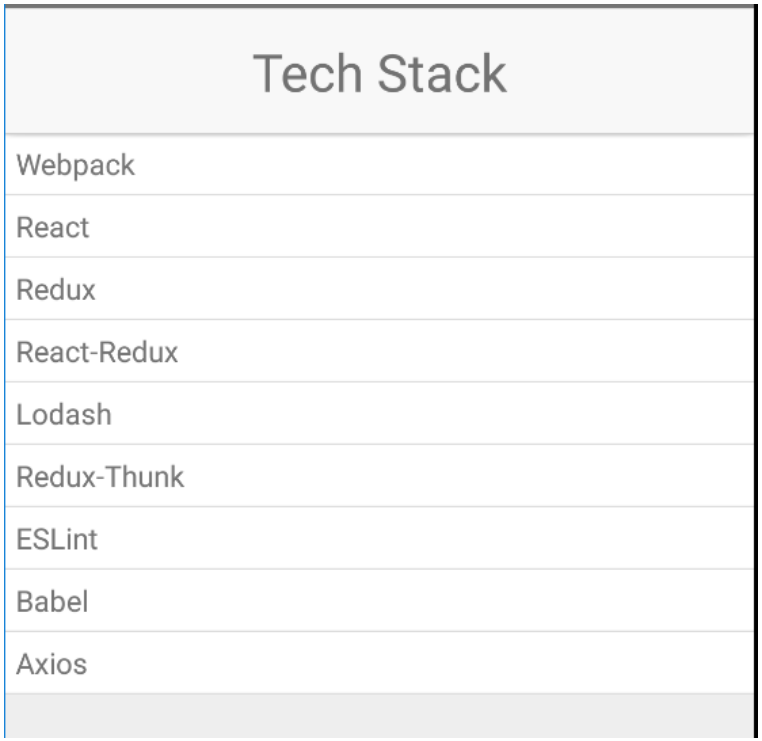



Purwadhika
Startup and Coding School

# Step 13

- Kemudian import CardSection di ListItem.js kemudian pasang

```
import { CardSection } from './common';

return (
    <View>
        <CardSection>
            <Text>{title}</Text>
        </CardSection>
    </View>
);
```




Purwadhika
Startup and Coding School

# Step 14

- Kemudian tambahkan style ke title Textnya

```
const styles = {
    titleStyle: {
        fontSize: 18,
        paddingLeft: 15
    }
};
 return (
     <View>
         <CardSection>
             <Text style={styles.titleStyle}>{title}</Text>
         </CardSection>
     </View>
 );
```

| Tech Stack |
| --- |
| Webpack |
| React |
| Redux |
| React-Redux |
| Lodash |
| Redux-Thunk |
| ESLint |
| Babel |
| Axios |

Purwadhika
Startup and Coding School

# Step 15

- Tambahkan style flex: 1 di tag View di App.js, biar ga kepotong isi dari ListView kita

```
const App = () => {
    return (
        <Provider store={createStore(reducers)}>
          <View style={{ flex: 1 }}>
            <Header headerText="Tech Stack" />
            <LibraryList />
          </View>
        </Provider>
    );
};
```

Purwadhika
Startup and Coding School

# Step 16

- Buat file Reducer baru namanya SelectionReducer.js di folder reducers

```
export default (state = null, action) => {
    switch (action.type) {
        case 'select_library':
            return action.payload;
        default:
            return state;
    }
};
```

- Kemudian import dan pasang SelectionReducer tersebut di combineReducers di file index.js di folder reducers

```
import SelectionReducer from './SelectionReducer';

export default combineReducers({
    libraries: LibraryReducer,
    selectedLibraryId: SelectionReducer
});
```

Purwadhika
Startup and Coding School

# Step 17

- Buat folder baru actions didalam folder src, dan buat file index.js di dalam folder actions tersebut, dan isinya ini:

```
export const selectLibrary = (libraryId) => {
    return {
        type: 'select_library',
        payload: libraryId
    };
};
```

- Kemudian import action creators tersebut ke ListItem.js

```
import * as actions from '../actions';
```

Purwadhika
Startup and Coding School

# Step 18

- Kemudian import connect dari react-redux di ListItem.js

```
import { Text, View } from 'react-native';
import { connect } from 'react-redux';
import { CardSection } from './common';
import * as actions from '../actions';
```

- Kemudian ubah exportnya jadi seperti ini untuk menghubungkan action creator

```
export default connect(null, actions)(ListItem);
```

Purwadhika
Startup and Coding School

# Step 19

- Kemudian import TouchableWithoutFeedback dari react-native di ListItem.js dan pasang

```
import {
    Text,
    View,
    TouchableWithoutFeedback
} from 'react-native';
```

```
<TouchableWithoutFeedback>
    <View>
        <CardSection>
            <Text style={styles.titleStyle}>{title}</Text>
        </CardSection>
    </View>
</TouchableWithoutFeedback>
```

Purwadhika
Startup and Coding School

# Step 20

- Tambahkan destructure id saat destructure title dari this.props.library

```
const { title, id } = this.props.library;
```

- Kemudian tambahkan event onPress pada TouchableWithoutFeedback

```
<TouchableWithoutFeedback
    onPress={() => this.props.selectLibrary(id)}
>
    <View>
        <CardSection>
            <Text style={styles.titleStyle}>{title}</Text>
        </CardSection>
    </View>
```

Purwadhika
Startup and Coding School

# Step 21

- Kemudian tambahkan mapStateToProps di ListItem, dan hubungkan dengan connect()

```
const mapStateToProps = (state) => {
    return { selectedLibraryId: state.selectedLibraryId };
};

export default connect(mapStateToProps, actions)(ListItem);
```

- Kemudian tambahkan function renderDescription diatas render

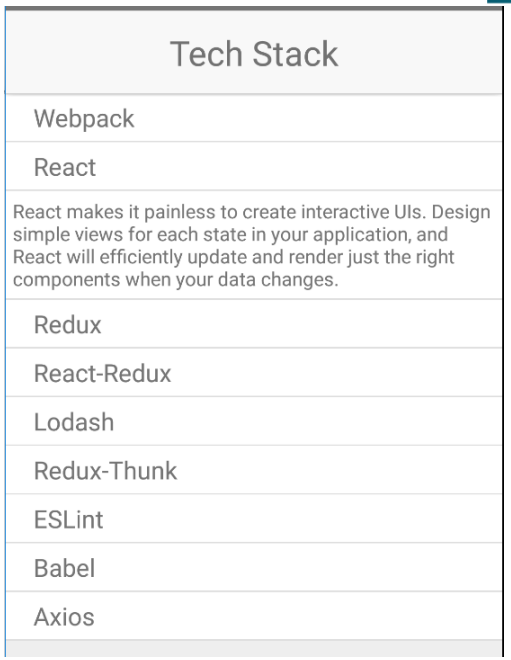
```
renderDescription() {
    const { selectedLibraryId, library } = this.props;

    if (library.id === selectedLibraryId) {
        return (
            <CardSection>
                <Text>
                    {library.description}
                </Text>
            </CardSection>
        );
    }
}
```

Purwadhika
Startup and Coding School

# Step 21 (continue)

- Kemudian pasang renderDescription tersebut didalam function render

```
>
    <View>
        <CardSection>
            <Text style={styles.titleStyle}>{title}</Text>
        </CardSection>
        {this.renderDescription()}
    </View>
</TouchableWithoutFeedback>
```




Purwadhika
Startup and Coding School

# Step 22

- Ubah function mapStateToProps menjadi seperti ini

```
const mapStateToProps = (state, ownProps) => {
    const expanded = state.selectedLibraryId === ownProps.library.id;

    return { expanded };
};
```

- Kemudian ubah isi function renderDescription juga

```
renderDescription() {
    const { expanded, library } = this.props;

    if (expanded) {
        return (
            <CardSection>
                <Text>
                    {library.description}
                </Text>
            </CardSection>
        );
    }
}
```

Purwadhika
Startup and Coding School

# Step 23

- Import LayoutAnimation, UIManager, dan Platform dari react-native di ListItem.js, dan kemudian tambahkan lifeCycleMethod componentWillUpdate()

```
import {
    Text,
    View,
    TouchableWithoutFeedback,
    LayoutAnimation,
    UIManager,
    Platform
} from 'react-native';
```

```
componentWillUpdate() {
    if (Platform.OS === 'android') {
        UIManager.setLayoutAnimationEnabledExperimental(true);
      }

    LayoutAnimation.spring();
}
```

Purwadhika
Startup and Coding School

# Finish!!!

- Apps ketiga kita akhirnya selesai hore!!

| Tech Stack |
| --- |
| Webpack |
| React |
| Redux |
| Redux is a predictable state container for JavaScript apps. It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. |
| React-Redux |
| Lodash |
| Redux-Thunk |
| ESLint |
| Babel |
| Axios |

Purwadhika
Startup and Coding School