

TUGAS PERORANGAN/INDIVIDU

REDUX

Disusun sebagai

MATA KULIAH : Pemrograman Berbasis Framework

Oleh :

Nurcahya Nania Anabela

TI-3A / 24

1741720156



PROGRAM STUDI D-IV T.INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

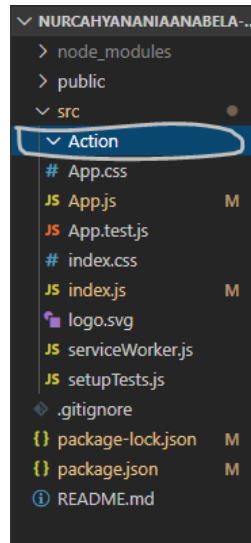
POLITEKNIK NEGERI MALANG

2020

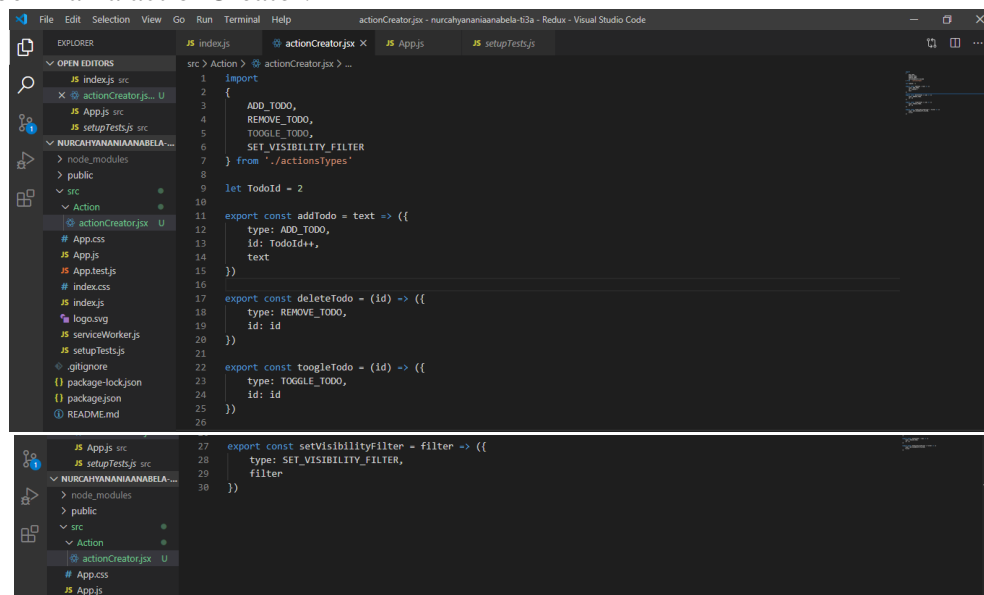
LATIHAN PRAKTIKUM REDUX

Langkah-Langkah :

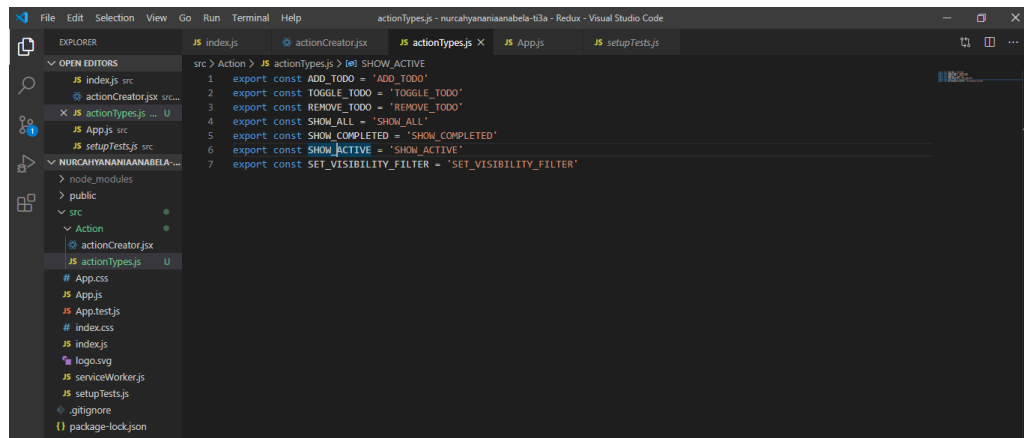
1. Membuat Folder dalam src yaitu **Action**.



2. Memberi nama actionCreator.

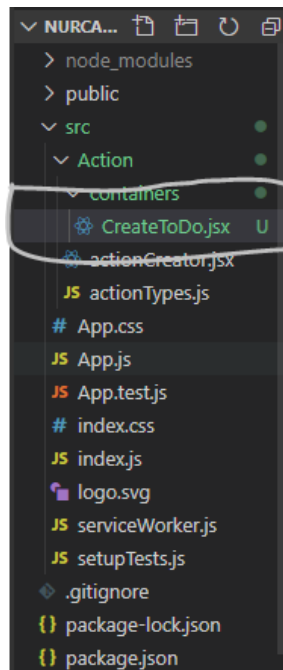


3. Masih dalam Folder Action buat file **actionsTypes.js**

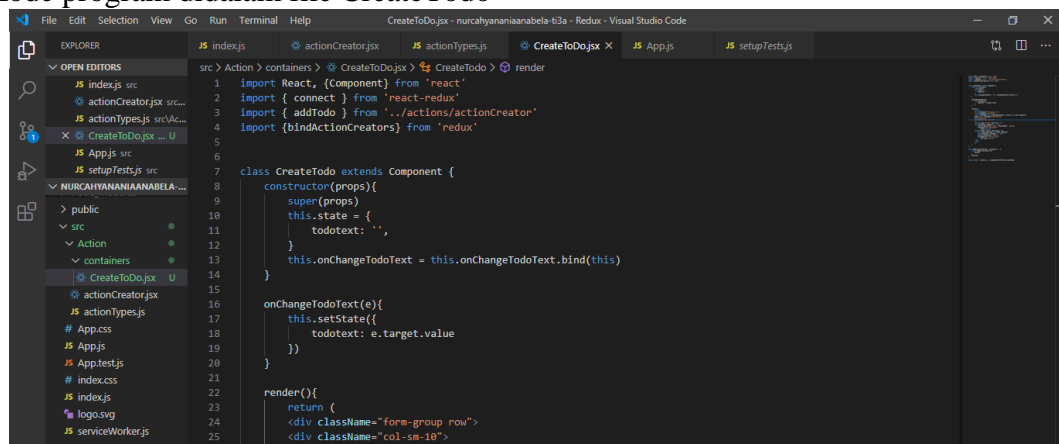


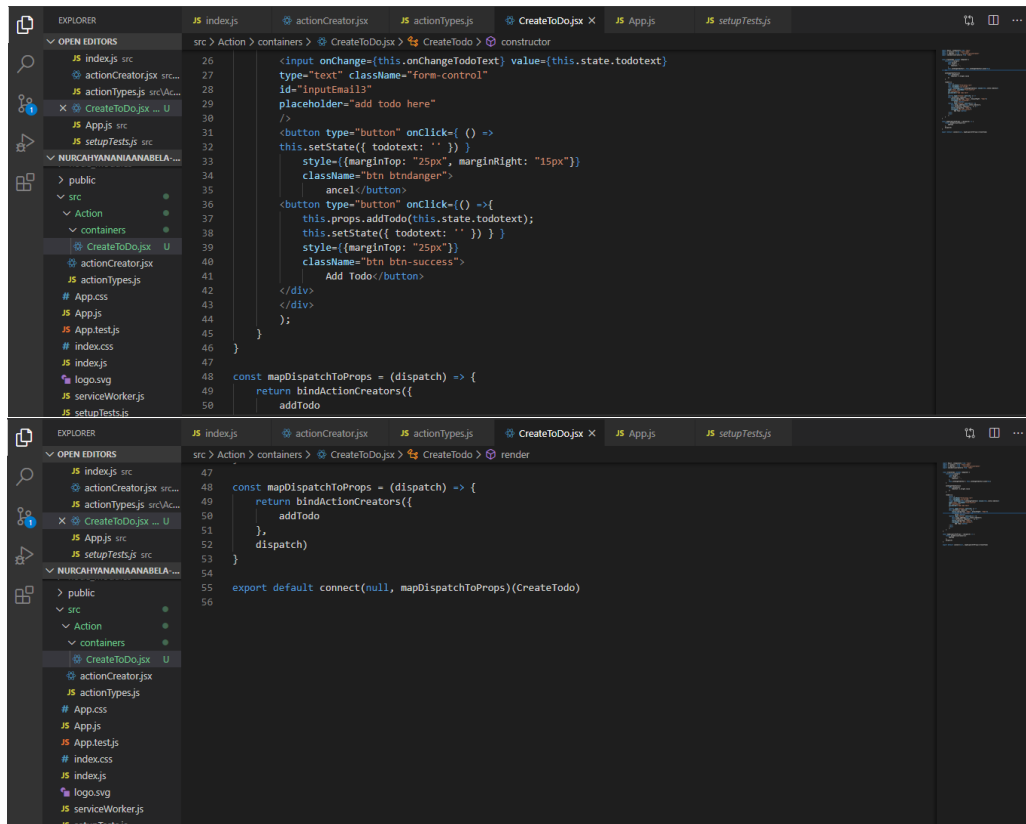
4. Kemudian dalam folder `src` buat folder dengan nama `containers` dan buat file dengan judul **CreateTodo**.

- Membuat folder `containers` dan membuat file `CreateTodo`

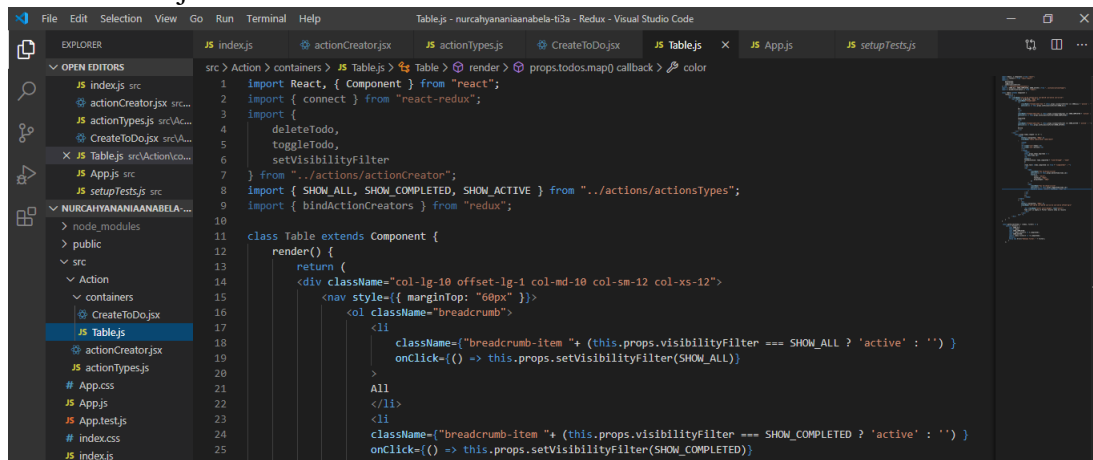


- Kode program didalam file `CreateTodo`





5. Buat file Table.js dalam folder containers



```
File Edit Selection View Go Run Terminal Help
Tablejs - nurcahyananianabela-t3a - Redux - Visual Studio Code

EXPLORER
src > Action > containers > JS Tablejs > Table > render > props.todos.map() callback > color

JS indexjs src 26
actionCreator.jsx src... 27
actionTypes.js src/Ac... 28
CreateToDo.jsx src/Ac... 29
X JS Tablejs src(ActionCo... 30
JS Appjs src 31
JS setupTests.js src 32
NURCAHYANANIANABELA... 33
node_modules 34
public 35
src 36
Action 37
containers 38
CreateToDo.jsx 39
JS Tablejs 40
actionCreator.jsx 41
actionTypes.js 42
App.css 43
App.js 44
App.test.js 45
index.css 46
index.js 47

Completed
</li>
</li>
className={"breadcrumb-item "+ (this.props.visibilityFilter === SHOW_ACTIVE ? 'active' : '') }
onClick={() => this.props.setVisibillityFilter(SHOW_ACTIVE)}
>
Active
</li>
</ol>
</nav>
(this.props.todos.length !== 0 ? (
<table
style={{ marginTop: "60px" }}
className="table table-hover table-dark"
>
<thead>
<tr>
<th scope="col">Todos</th>
<th scope="col">Actions</th>
</tr>
</thead>
<tbody>
(this.props.todos.map(todo => (
<tr key={todo.id}>
```

```
File Edit Selection View Go Run Terminal Help
Tablejs - nurcahyananianabela-t3a - Redux - Visual Studio Code

EXPLORER
src > Action > containers > JS Tablejs > Table > render > props.todos.map() callback > color

JS indexjs src 51
actionCreator.jsx src... 52
actionTypes.js src/Ac... 53
CreateToDo.jsx src/Ac... 54
X JS Tablejs src(ActionCo... 55
JS Appjs src 56
JS setupTests.js src 57
NURCAHYANANIANABELA... 58
node_modules 59
public 60
src 61
Action 62
containers 63
CreateToDo.jsx 64
JS Tablejs 65
actionCreator.jsx 66
actionTypes.js 67
App.css 68
App.js 69
App.test.js 70
index.css 71
index.js 72

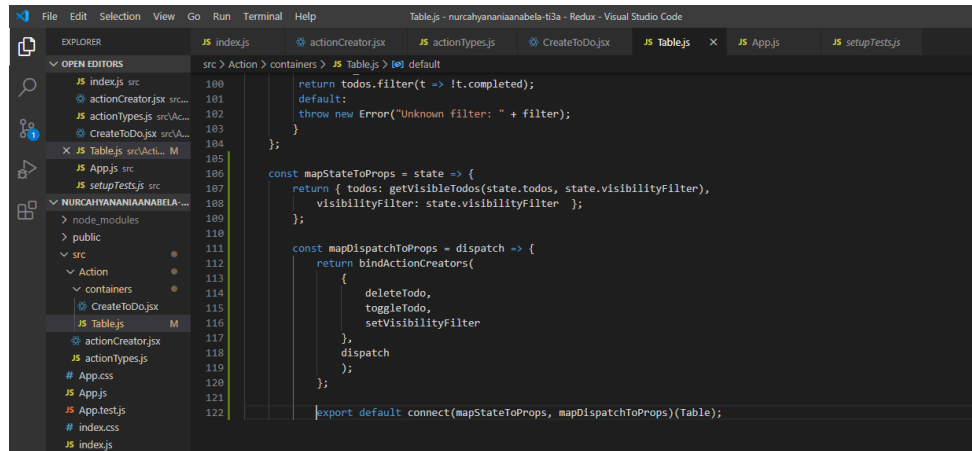
<td
style={{
textDecoration: todo.completed ? "line-through" : "none"
}}
>
{todo.text} {todo.completed === true ? "(completed)" : ""}
</td>
<td>
<span
className="fas fa-minus-circle"
onClick={() => this.props.deleteTodo(todo.id)}
style={{
color: "white",
fontSize: "20pt",
marginRight: "20px"
}}
/>
<span
className="fas fa-check-circle"
onClick={() => this.props.toggleTodo(todo.id)}
style={{ color: "white", fontSize: "20pt" }}
/>
</td>
</tr>
)))
```

```
File Edit Selection View Go Run Terminal Help
Tablejs - nurcahyananianabela-t3a - Redux - Visual Studio Code

EXPLORER
src > Action > containers > JS Tablejs > Table > render > props.todos.map() callback > color

JS indexjs src 76
actionCreator.jsx src... 77
actionTypes.js src/Ac... 78
CreateToDo.jsx src/Ac... 79
X JS Tablejs src(ActionCo... 80
JS Appjs src 81
JS setupTests.js src 82
NURCAHYANANIANABELA... 83
node_modules 84
public 85
src 86
Action 87
containers 88
CreateToDo.jsx 89
JS Tablejs 90
actionCreator.jsx 91
actionTypes.js 92
App.css 93
App.js 94
App.test.js 95
index.css 96
index.js 97

</tbody>
</table>
): (
<div
style={{ marginTop: "50px" }}
className="col-lg-10 col-md-10 col-xs-12 col-sm-12 offset-lg-1"
>
<div className="alert alert-danger" role="alert">
Todo list is empty or Filter results show no results
</div>
</div>
)))( " "
</div>
);
}
}
const getVisibleTodos = (todos, filter) => {
switch (filter) {
case SHOW_ALL:
return todos;
case SHOW_COMPLETED:
return todos.filter(t => t.completed);
case SHOW_ACTIVE:
return todos.filter(t => !t.completed);
```



The screenshot shows the Visual Studio Code editor with the 'Table.js' file open. The file is located in the 'src/reducers' directory. The code defines a 'default' reducer that filters todos based on a 'filter' prop. It also defines two helper functions: 'mapStateToProps' and 'mapDispatchToProps', and exports a 'connect' function that connects the reducer to the 'Table' component.

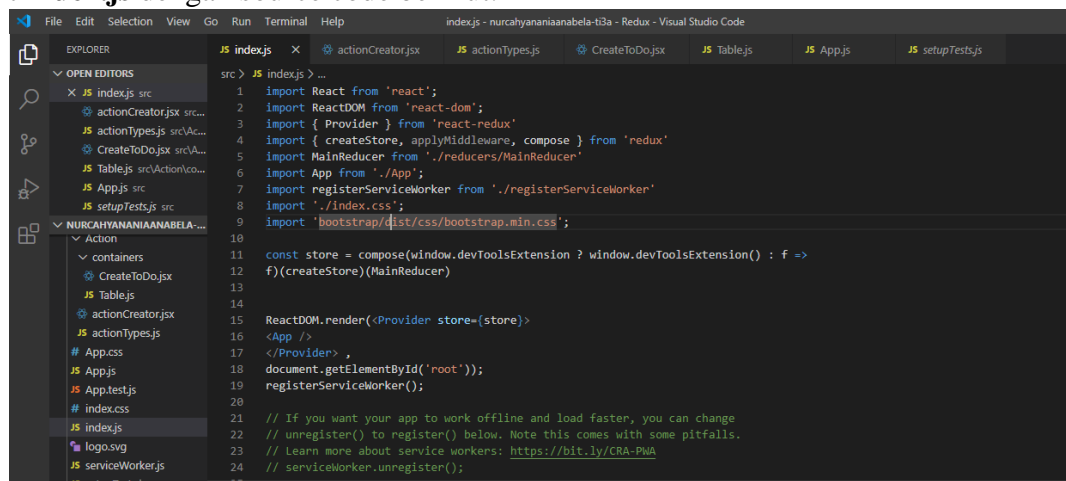
```
return todos.filter(t => !t.completed);
default:
  throw new Error("Unknown filter: " + filter);
};

const mapStateToProps = state => {
  return { todos: getVisibleTodos(state.todos, state.visibilityFilter),
    visibilityFilter: state.visibilityFilter };
};

const mapDispatchToProps = dispatch => {
  return bindActionCreators(
    {
      deleteTodo,
      toggleTodo,
      setVisibilityFilter
    },
    dispatch
  );
};

export default connect(mapStateToProps, mapDispatchToProps)(Table);
```

6. Edit **index.js** dengan source code berikut.



The screenshot shows the Visual Studio Code editor with the 'index.js' file open. The file is located in the 'src' directory. The code imports necessary libraries and components, sets up the Redux store, and renders the application using ReactDOM. It also includes comments about service workers and offline functionality.

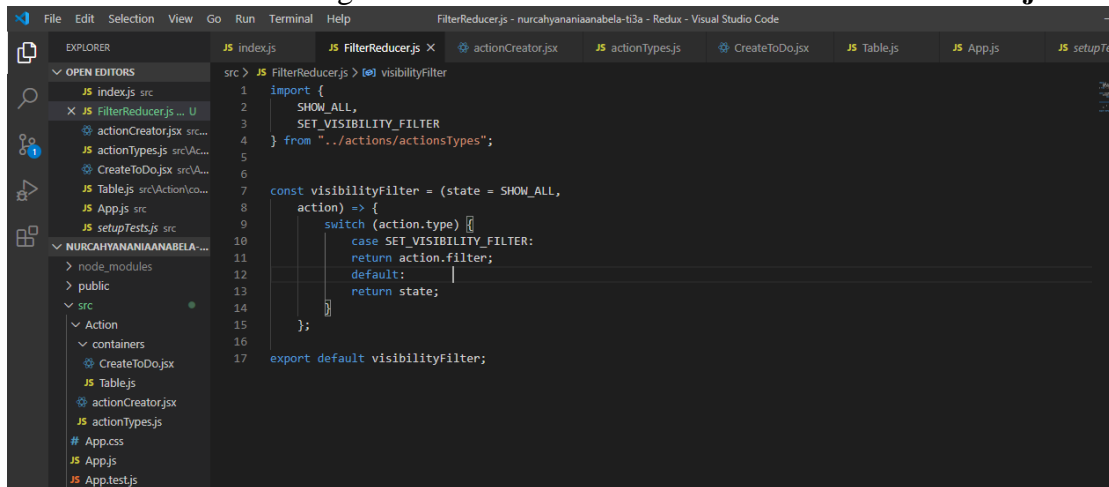
```
import React from 'react';
import ReactDOM from 'react-dom';
import { Provider } from 'react-redux';
import { createStore, applyMiddleware, compose } from 'redux';
import MainReducer from './reducers/MainReducer';
import App from './App';
import registerServiceWorker from './registerServiceWorker';
import './index.css';
import 'bootstrap/dist/css/bootstrap.min.css';

const store = compose(window.devToolsExtension ? window.devToolsExtension() : f =>
f)(createStore)(MainReducer);

ReactDOM.render(<Provider store={store}>
  <App />
</Provider>,
document.getElementById("root"));
registerServiceWorker();

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
// serviceWorker.unregister();
```

7. Buat folder dalam src dengan nama reducers dan beri nama file **FilterReducer.js**.



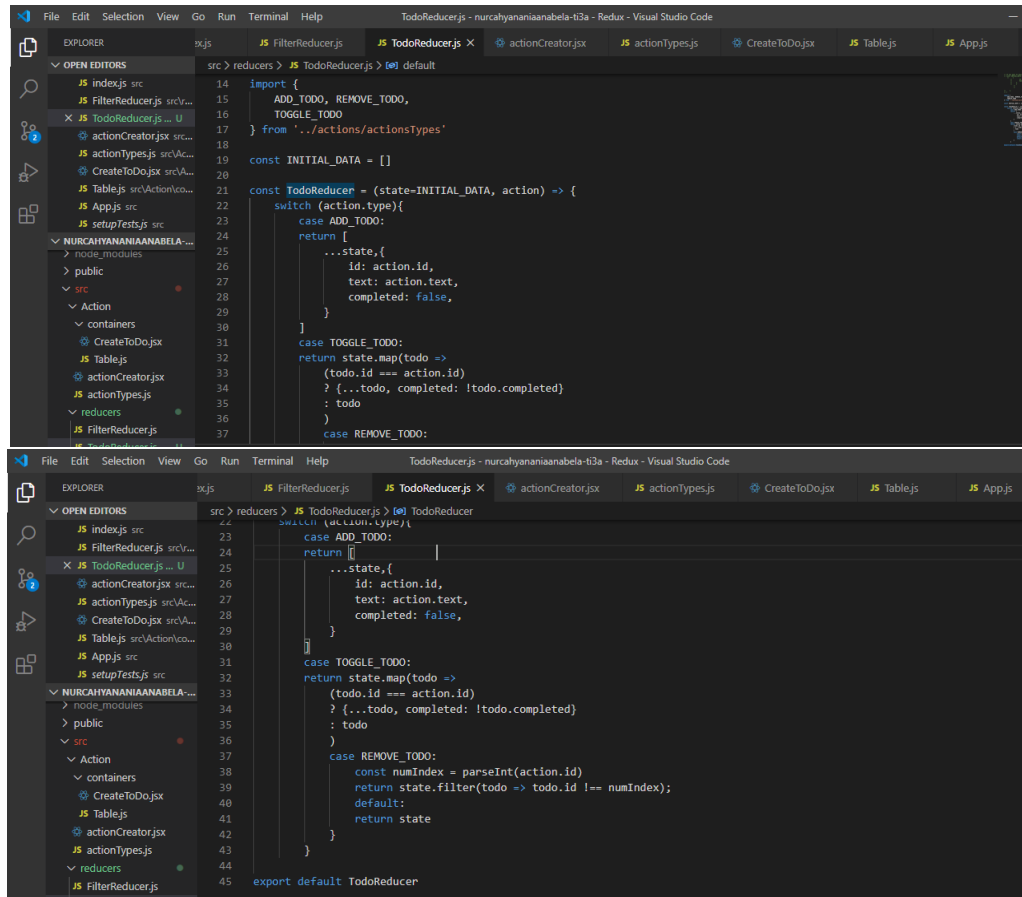
The screenshot shows the Visual Studio Code editor with the 'FilterReducer.js' file open. The file is located in the 'src/reducers' directory. The code imports 'SHOW_ALL' and 'SET_VISIBILITY_FILTER' from 'actions/actionTypes'. It defines a 'visibilityFilter' reducer that updates the filter based on the 'action.type' and exports it as 'default'.

```
import {
  SHOW_ALL,
  SET_VISIBILITY_FILTER
} from "../actions/actionTypes";

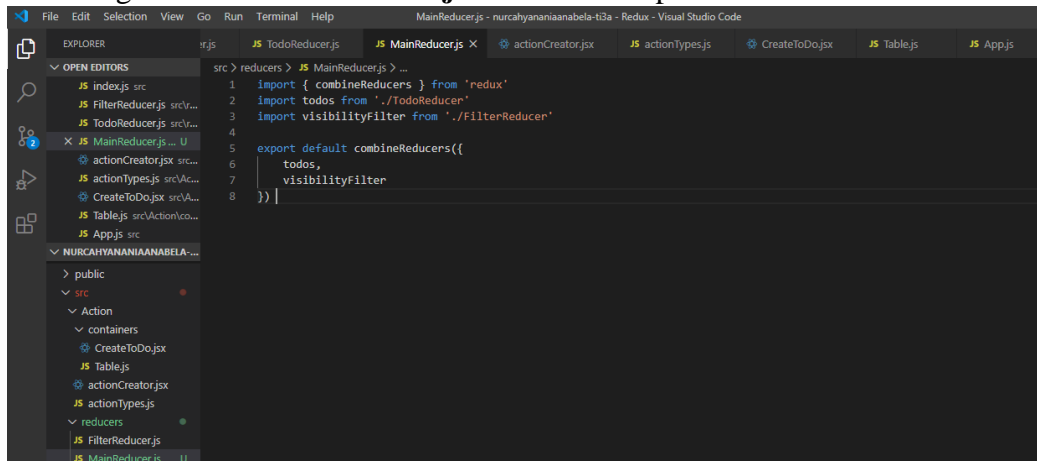
const visibilityFilter = (state = SHOW_ALL,
  action) => {
  switch (action.type) {
    case SET_VISIBILITY_FILTER:
      return action.filter;
    default:
      return state;
  }
};

export default visibilityFilter;
```

8. Masih dalam folder reducers buatlah file dengan nama **TodoReducer.js**



9. Buat file dengan nama **MainReducer.js** kemudian simpan.



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer shows a project structure with a 'src' folder containing 'reducers' and 'containers'. The 'reducers' folder is expanded, showing 'FilterReducer.js' and 'MainReducer.js'. The 'MainReducer.js' file is selected and its content is visible in the editor. The code in 'MainReducer.js' is as follows:

```
1 import { combineReducers } from 'redux'
2 import todos from './TodoReducer'
3 import visibilityFilter from './FilterReducer'
4
5 export default combineReducers({
6   todos,
7   visibilityFilter
8 })
```

10. Running semua program dengan *npm start* dan lihat hasilnya.

11. Hasil Output Running program.

