

**TUGAS INDIVIDU**  
**PEMROGRAMAN BERBASIS FRAMEWORK**  
**“Redux”**

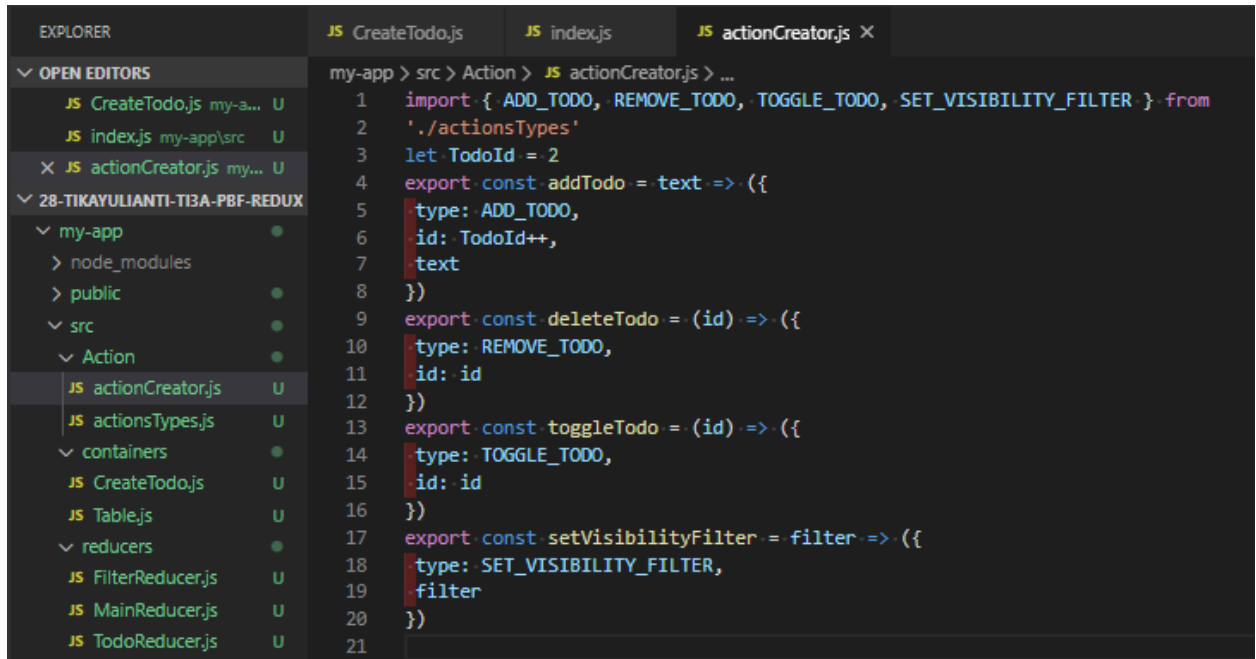


oleh:  
**Tika Yulianti / 1741720153**  
**TI-3A**

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2020**

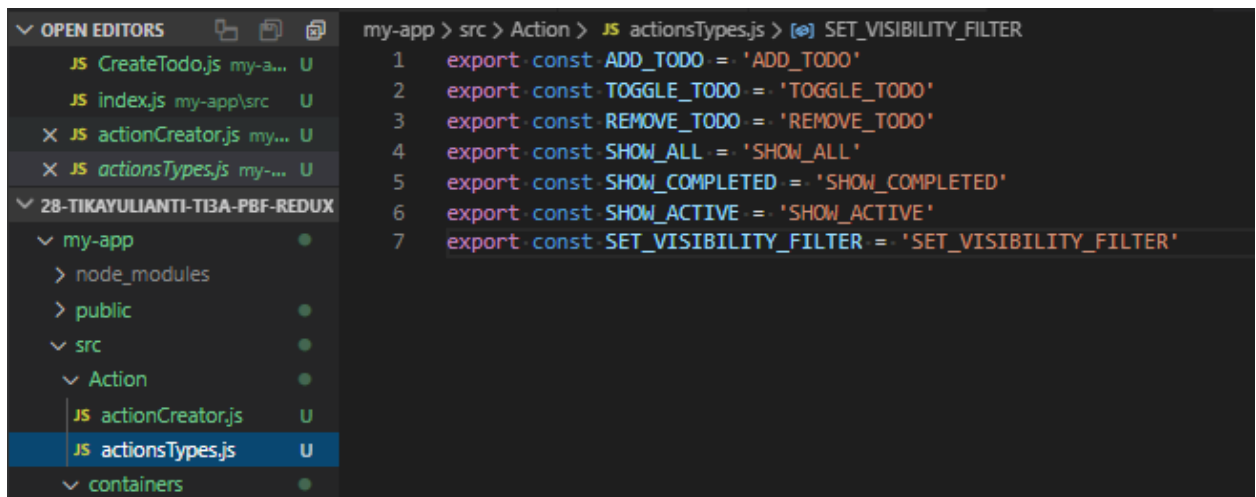
## Praktikum

1. Membuat folder di dalam src yaitu folder **Action**
2. Membuat file di dalam folder **Action** dengan nama **actionCreator.js**



```
EXPLORER
my-app > src > Action > JS actionCreator.js > ...
1 import { ADD_TODO, REMOVE_TODO, TOGGLE_TODO, SET_VISIBILITY_FILTER } from
2 './actionsTypes'
3 let TodoId = 2
4 export const addTodo = text => ({
5   type: ADD_TODO,
6   id: TodoId++,
7   text
8 })
9 export const deleteTodo = (id) => ({
10  type: REMOVE_TODO,
11  id: id
12 })
13 export const toggleTodo = (id) => ({
14  type: TOGGLE_TODO,
15  id: id
16 })
17 export const setVisibilityFilter = filter => ({
18  type: SET_VISIBILITY_FILTER,
19  filter
20 })
21
```

3. Membuat file di dalam folder **Action** dengan nama **actionsTypes.js**



```
OPEN EDITORS
my-app > src > Action > JS actionsTypes.js > [e] SET_VISIBILITY_FILTER
1 export const ADD_TODO = 'ADD_TODO'
2 export const TOGGLE_TODO = 'TOGGLE_TODO'
3 export const REMOVE_TODO = 'REMOVE_TODO'
4 export const SHOW_ALL = 'SHOW_ALL'
5 export const SHOW_COMPLETED = 'SHOW_COMPLETED'
6 export const SHOW_ACTIVE = 'SHOW_ACTIVE'
7 export const SET_VISIBILITY_FILTER = 'SET_VISIBILITY_FILTER'
```

4. Kemudian menambahkan folder dengan nama **containers** didalam src
5. Membuat file di dalam folder **containers** dengan nama **CreateTodo.js**

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'my-app', 'src', 'containers', and 'reducers'. The code editor shows the content of 'CreateTodo.js'.

```

my-app > src > containers > JS CreateTodo.js > CreateTodo > onChangeTodoText
1  import React, {Component} from './node_modules/react'
2  import { connect } from './node_modules/react-redux'
3  import { addToDo } from '../actions/actionCreator'
4  import { bindActionCreators } from './node_modules/redux'
5  class CreateTodo extends Component {
6      constructor(props){
7          super(props)
8          this.state = {
9              todotext: '',
10             }
11             this.onChangeTodoText = this.onChangeTodoText.bind(this)
12         }
13         onChangeTodoText(e){
14             this.setState({
15                 todotext: e.target.value
16             })
17         }
18         render(){
19             return (
20                 <div className="form-group row">
21                     <div className="col-sm-10">
22                         <input onChange={this.onChangeTodoText} value={this.state.todotext}
23                             type="text" className="form-control" id="inputEmail3" placeholder="add todo here"/>
24                         <button type="button" onClick={() => this.setState({ todotext: '' })}
25                             style={{marginTop: "25px", marginRight: "15px"}} className="btn btn-danger">Cancel</button>
26                         <button type="button" onClick={() => {
27                             this.props.addToDo(this.state.todotext); this.setState({ todotext: '' })
28                             style={{marginTop: "25px"}} className="btn btn-success">Add Todo</button>
29                     }
30                 </div>
31             );
32         }
33     }
34     const mapDispatchToProps = (dispatch) => {
35         return bindActionCreators({
36             addToDo
37         }, dispatch)
38     }
39     export default connect(null, mapDispatchToProps)(CreateTodo)

```

6. Membuat file di dalam folder **containers** dengan nama **Table.js**

OPEN EDITORS

JS CreateTodo.js my-app U

X JS Table.js my-app/src U

JS index.js my-app/src U

X JS actionCreator.js my-app U

28-TIKAYULIANTI-TIBA-PBF-REDUX

my-app

node\_modules

public

src

Action

JS actionCreator.js U

JS actionTypes.js U

containers

JS CreateTodo.js U

JS Table.js U

reducers

JS FilterReducer.js U

JS MainReducer.js U

JS TodoReducer.js U

# App.css U

JS App.js U

JS App.test.js U

# index.css U

JS index.js U

logo.svg U

JS serviceWorker.js U

JS setupTests.js U

.gitignore U

package-lock.json U

package.json U

README.md U

.gitignore U

package-lock.json U

OUTLINE

my-app > src > containers > JS Table.js > ...

1 import React, { Component } from "react";  
2 import { connect } from "react-redux";  
3 import {  
4 deleteTodo,  
5 toggleTodo,  
6 setVisibilityFilter  
7 } from "../actions/actionCreator";  
8 import { SHOW\_ALL, SHOW\_COMPLETED, SHOW\_ACTIVE } from "../actions/actionTypes";  
9 import { bindActionCreators } from "redux";  
10 class Table extends Component {  
11 render() {  
12 return (  
13 <div className="col-lg-10 offset-lg-1 col-md-10 col-sm-12 col-xs-12">  
14 <nav style={{ marginTop: "60px" }}>  
15 <ol className="breadcrumb">  
16 <li  
17 className="breadcrumb-item "+ (this.props.visibilityFilter === SHOW\_ALL ?  
18 'active' : '') )>  
19 onClick={() => this.props.setVisibilityFilter(SHOW\_ALL)}  
20 >  
21 All  
22 </li>  
23 <li  
24 className="breadcrumb-item "+ (this.props.visibilityFilter ===  
25 SHOW\_COMPLETED ? 'active' : '') )>  
26 onClick={() => this.props.setVisibilityFilter(SHOW\_COMPLETED)}  
27 >  
28 Completed  
29 </li>  
30 <li  
31 className="breadcrumb-item "+ (this.props.visibilityFilter === SHOW\_ACTIVE ?  
32 'active' : '') )>  
33 onClick={() => this.props.setVisibilityFilter(SHOW\_ACTIVE)}  
34 >  
35 Active  
36 </li>  
37 </ol>  
38 </nav>  
39 {this.props.todos.length !== 0 ? (  
40 <table

OPEN EDITORS	my-app > src > containers > JS Table.js > ...
JS CreateTodo.js my-a... U	40 <table
X JS Table.js my-app/src/... U	41 style={{ marginTop: "60px" }}
JS index.js my-app/src U	42 className="table table-hover table-dark"
JS actionCreator.js my... U	43 >
28-TIKAYULIANTI-T3A-P8F-REDUX	44 <thead>
my-app	45 <tr>
node_modules	46 <th scope="col">Todos</th>
public	47 <th scope="col">Actions</th>
src	48 </tr>
Action	49 </thead>
JS actionCreator.js U	50 <tbody>
JS actionTypes.js U	51 {this.props.todos.map(todo => {
containers	52 <tr key={todo.id}>
JS CreateTodo.js U	53 <td
JS Table.js U	54 style={{
reducers	55 textDecoration: todo.completed ? "line-through" : "none"
JS FilterReducer.js U	56 >
JS MainReducer.js U	57 >
JS TodoReducer.js U	58 {todo.text} {todo.completed === true ? "(completed)" : ""}
# App.css U	59 </td>
JS App.js U	60 <td>
JS App.test.js U	61 <span
# index.css U	62 className="fas fa-minus-circle"
JS index.js U	63 onClick={() => this.props.deleteTodo(todo.id)}
logo.svg U	64 style={{
JS serviceWorker.js U	65 color: "white",
JS setupTests.js U	66 fontSize: "20pt",
.gitignore U	67 marginRight: "20px"
package-lock.json U	68 >
package.json U	69 </span>
README.md U	70 <span
.gitignore U	71 className="fas fa-check-circle"
package-lock.json U	72 onClick={() => this.props.toggleTodo(todo.id)}
	73 style={{ color: "white", fontSize: "20pt" }}
	74 >
	75 </td>
	76 </tr>
	77 >}}
	78 </tbody>
	79 </table>
OUTLINE	

```

my-app > src > containers > JS Table.js > ...
82 <div style={style} className={className} >
83 <div className="alert alert-danger" role="alert">
84 <div className="alert alert-danger" role="alert">
85 <div className="alert alert-danger" role="alert">
86 <div className="alert alert-danger" role="alert">
87 <div className="alert alert-danger" role="alert">
88 <div className="alert alert-danger" role="alert">
89 <div className="alert alert-danger" role="alert">
90 <div className="alert alert-danger" role="alert">
91 <div className="alert alert-danger" role="alert">
92 <div className="alert alert-danger" role="alert">
93 <div className="alert alert-danger" role="alert">
94 const getVisibleTodos = (todos, filter) => {
95   switch (filter) {
96     case SHOW_ALL:
97       return todos;
98     case SHOW_COMPLETED:
99       return todos.filter(t => t.completed);
100    case SHOW_ACTIVE:
101      return todos.filter(t => !t.completed);
102    default:
103      throw new Error("Unknown filter: " + filter);
104    }
105  };
106  const mapStateToProps = state => {
107    return { todos: getVisibleTodos(state.todos, state.visibilityFilter),
108            visibilityFilter: state.visibilityFilter
109          };
110  };
111  const mapDispatchToProps = dispatch => {
112    return {
113      deleteTodo,
114      toggleTodo,
115      setVisibilityFilter
116    },
117    dispatch
118  };
119  };
120  };
121  export default connect(mapStateToProps, mapDispatchToProps)(Table);

```

7. Mengedit **index.js** dengan source code sebagai berikut:

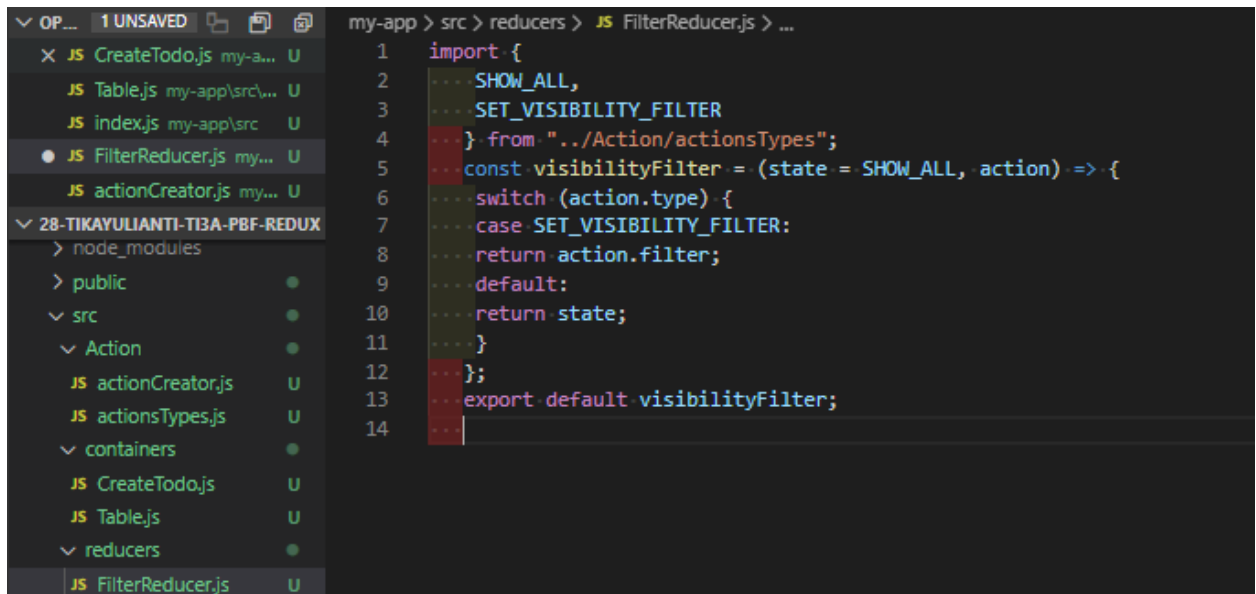
```

my-app > src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import { Provider } from 'react-redux';
4  import { createStore, applyMiddleware, compose } from 'redux';
5  import MainReducer from './reducers/MainReducer';
6  import App from './App';
7  import * as serviceWorker from './serviceWorker';
8  const store = compose(window.devToolsExtension ? window.devToolsExtension() : f =>
9  f)(createStore)(MainReducer);
10 ReactDOM.render(<Provider store={store}>
11 <App />
12 </Provider>
13 , document.getElementById('root'));
14 serviceWorker.unregister();

```

8. Membuat folder baru dengan nama **reducers** di dalam src

9. Membuat file baru di dalam folder **reducers** dengan nama **FilterReducer.js**



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure. The file `FilterReducer.js` is highlighted under the `reducers` folder. The main editor area shows the content of `FilterReducer.js` with the following code:

```
1 import {
2   ...SHOW_ALL,
3   ...SET_VISIBILITY_FILTER
4 } from "../Action/actionTypes";
5 const visibilityFilter = (state = SHOW_ALL, action) => {
6   switch (action.type) {
7     case SET_VISIBILITY_FILTER:
8       return action.filter;
9     default:
10      return state;
11   }
12 };
13 export default visibilityFilter;
```

10. Membuat file baru di dalam folder **reducers** dengan nama **TodoReducer.js**

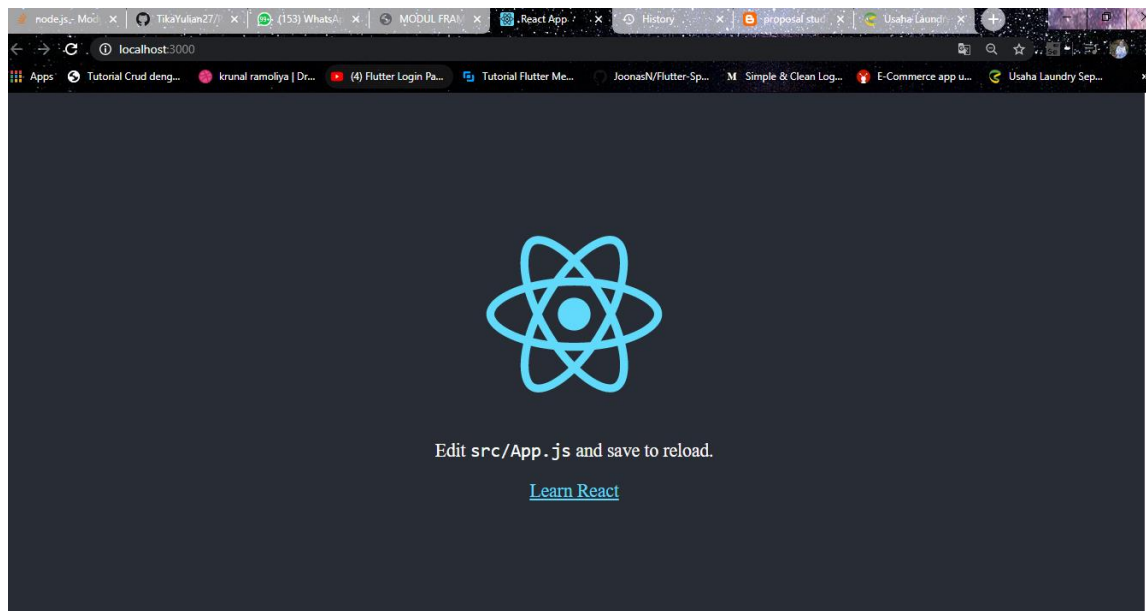
```
my-app > src > reducers > JS TodoReducer.js > [?] TodoReducer > [?] state.map() callback
1  //if you want to show initial data :)
2  //const INITIAL_DATA = [
3  //{
4  //id: 0,
5  //text: 'Walk the Dog',
6  //},
7  //{
8  //id: 1,
9  //text: 'learn Redux',
10 //},
11 //]
12 import { ADD_TODO, REMOVE_TODO, TOGGLE_TODO } from '../Action/actionsTypes'
13 const INITIAL_DATA = []
14 const TodoReducer = (state=INITIAL_DATA, action) => {
15   switch (action.type){
16     case ADD_TODO:
17       return [
18         ...state,{
19           id: action.id,
20           text: action.text,
21           completed: false,
22         }
23       ]
24     case TOGGLE_TODO:
25       return state.map(todo =>
26         [todo.id === action.id]
27         ? {...todo, completed: !todo.completed}
28         : todo
29       )
30     case REMOVE_TODO:
31       const numIndex = parseInt(action.id)
32       return state.filter(todo => todo.id !== numIndex);
33     default:
34       return state
35   }
36 }
37 export default TodoReducer
```

11. Membuat file baru di dalam folder **reducers** dengan nama **MainReducer.js**

```
my-app > src > reducers > JS MainReducer.js > ...
1  import { combineReducers } from 'redux'
2  import todos from './TodoReducer'
3  import visibilityFilter from './FilterReducer'
4  export default combineReducers({
5    todos,
6    visibilityFilter
7  })
8
```

12. Menjalankan program dengan cara **npm start**





13. Selesai