



MODUL FRAMEWORK

REDUX

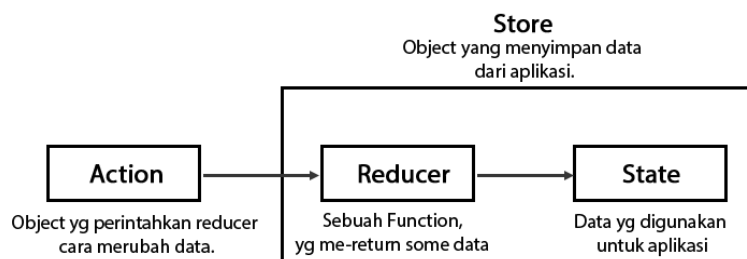
Apa itu Redux?

[Redux](#) adalah predictable state container untuk JavaScript aplikasi.

Redux membantu anda untuk membuat aplikasi dengan efisien dan menjaga konsistensi pada environment yang berbeda, seperti pada client, server, dan native, serta mudah untuk melakukan testing. Selain daripada itu, Redux memberikan pengalaman yang luar biasa kepada para pengembang, seperti live code editing dan time traveling debugger.

Anda dapat menggunakan Redux untuk aplikasi React, atau dengan view library yang lain, dengan ukuran size yang cukup kecil berkisar 2kB, termasuk dependencies.

DIAGRAM KONSEP REDUX



Detail Diagram

Reducer adalah function yang memiliki kembalian data (return some data).

Action adalah plain **JavaScript Object** ("{ }") yang memberitahukan Reducer cara memodifikasi data. Syarat dari Action adalah harus memiliki sebuah **type** property.

State adalah Data dari sebuah aplikasi yang digunakan, misalnya: value dari input field, data list, gambar, seluruh data yang berada di dalam aplikasi, dikumpulkan menjadi **State**.

Store adalah container yang menyimpan Reducer & State pada sebuah aplikasi.

Cara membuat Redux

```
// Step membuat Redux
// 1-Membuat function reducer
const reducer = () => [];
// 2-Membuat store
const store = Redux.createStore(reducer);

// 3-Menampilkan current state
store.getState();
```

Praktikum :

1. Buat Folder dalam src yaitu Action
2. Beri nama actionCreator

```
import { ADD_TODO, REMOVE_TODO, TOGGLE_TODO, SET_VISIBILITY_FILTER } from
'./actionsTypes'

let TodoId = 2

export const addTodo = text => ({
  type: ADD_TODO,
  id: TodoId++,
  text
})

export const deleteTodo = (id) => ({
  type: REMOVE_TODO,
  id: id
})

export const toggleTodo = (id) => ({
  type: TOGGLE_TODO,
  id: id
})

export const setVisibilityFilter = filter => ({
  type: SET_VISIBILITY_FILTER,
  filter
})
```

3. Masih dalam Folder Action buatlah file actionsTypes.js

```
export const ADD_TODO = 'ADD_TODO'
export const TOGGLE_TODO = 'TOGGLE_TODO'
export const REMOVE_TODO = 'REMOVE_TODO'
export const SHOW_ALL = 'SHOW_ALL'
export const SHOW_COMPLETED = 'SHOW_COMPLETED'
export const SHOW_ACTIVE = 'SHOW_ACTIVE'
export const SET_VISIBILITY_FILTER = 'SET_VISIBILITY_FILTER'
```

4. Kemudian dalam folder src buatlah folder dengan nama containers dan buatlah file dengan judul CreateTodo

```
import React, {Component} from 'react'
import { connect } from 'react-redux'
import { addToDo } from '../actions/actionCreator'
import { bindActionCreators } from 'redux'

class CreateTodo extends Component {
  constructor(props){
    super(props)
    this.state = {
      todotext: "",
    }
    this.onChangeTodoText = this.onChangeTodoText.bind(this)
  }

  onChangeTodoText(e){
    this.setState({
      todotext: e.target.value
    })
  }

  render(){
    return (
      <div className="form-group row">
        <div className="col-sm-10">
          <input onChange={this.onChangeTodoText} value={this.state.todotext}
type="text" className="form-control" id="inputEmail3" placeholder="add todo here"/>
          <button type="button" onClick={ () => this.setState({ todotext: "" }) }
style={{marginTop: "25px", marginRight: "15px"}} className="btn btn-
danger">Cancel</button>
          <button type="button" onClick={() => {
this.props.addToDo(this.state.todotext); this.setState({ todotext: "" }) } }
style={{marginTop: "25px"}} className="btn btn-success">Add Todo</button>
        </div>
      </div>
    );
  }
}

const mapDispatchToProps = (dispatch) => {
  return bindActionCreators({
    addToDo
  }, dispatch)
}

export default connect(null, mapDispatchToProps)(CreateTodo)
```

5. Masih dalam folder containers buatlah file dengan judul Table.js

```
import React, { Component } from "react";
import { connect } from "react-redux";
import {
  deleteTodo,
  toggleTodo,
  setVisibilityFilter
} from "../actions/actionCreator";
import { SHOW_ALL, SHOW_COMPLETED, SHOW_ACTIVE } from "../actions/actionsTypes";
import { bindActionCreators } from "redux";

class Table extends Component {
  render() {
    return (
      <div className="col-lg-10 offset-lg-1 col-md-10 col-sm-12 col-xs-12">
        <nav style={{ marginTop: "60px" }}>
          <ol className="breadcrumb">
            <li
              className={"breadcrumb-item " + (this.props.visibilityFilter === SHOW_ALL ?
'active' : '') }
              onClick={() => this.props.setVisibilityFilter(SHOW_ALL)}
            >
              All
            </li>
            <li
              className={"breadcrumb-item " + (this.props.visibilityFilter ===
SHOW_COMPLETED ? 'active' : '') }
              onClick={() => this.props.setVisibilityFilter(SHOW_COMPLETED)}
            >
              Completed
            </li>
            <li
              className={"breadcrumb-item " + (this.props.visibilityFilter === SHOW_ACTIVE ?
'active' : '') }
              onClick={() => this.props.setVisibilityFilter(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}>
```



```

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 ini

```

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';

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

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

```

7. Buat folder dalam src dengan nama reducers dan beri nama file FilterReducer.js dan ketikkan program dibawah ini :

```
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 didalam folder reducers buatlah file dengan nama TodoReducer.js

```
// if you want to show initial data :)
// const INITIAL_DATA = [
//   {
//     id: 0,
//     text: 'Walk the Dog',
//   },
//   {
//     id: 1,
//     text: 'learn Redux',
//   },
// ]

import { ADD_TODO, REMOVE_TODO, TOGGLE_TODO } from '../actions/actionTypes'

const INITIAL_DATA = []

const TodoReducer = (state=INITIAL_DATA, action) => {
  switch (action.type){
    case ADD_TODO:
      return [
        ...state,{
          id: action.id,
          text: action.text,
          completed: false,
        }
      ]
    case TOGGLE_TODO:
      return state.map(todo =>
        (todo.id === action.id)
        ? {...todo, completed: !todo.completed}
        : todo
      )
    case REMOVE_TODO:
```



```
    const numIndex = parseInt(action.id)
    return state.filter(todo => todo.id !== numIndex);
  default:
    return state
  }
}

export default TodoReducer
```

9. Buat file dengan nama MainReducer.js dan simpan

```
import { combineReducers } from 'redux'
import todos from './TodoReducer'
import visibilityFilter from './FilterReducer'

export default combineReducers({
  todos,
  visibilityFilter
})
```

10. Running semua program dengan npm start dan liat hasilnya dan laporkan didalam laporan