

Pengembangan Aplikasi Web

TI2109

By:
Participant Handbook

Course Version: 2023

Universitas Mikroskil, Copyright ©2023

COURSE OVERVIEW

COURSE OVERVIEW

Mata kuliah ini, mahasiswa akan mempelajari tentang bagaimana membangun dan mengembangkan aplikasi web mulai dari front end, back-end server, aplikasi dan basis data sampai pembuatan RestAPI dengan memanfaatkan React Js, Node Js dan Express Js.

COURSE GOALS

Capaian Pembelajaran Lulusan yang dibebankan:

- ☐ Mahasiswa mampu memahami dan menerapkan konsep-konsep dasar dalam pengembangan web menggunakan React Js.
- ☐ Mahasiswa Mampu mengimplementasikan teknik routing dan styling dalam pengembangan web menggunakan React Js.
- ☐ Mahasiswa mampu membangun dan mengelola server menggunakan Express Js.
- ☐ Mahasiswa mampu mengintegrasikan API dalam pengembangan web menggunakan React Js.

COURSE OBJECTIVES

Capaian Pembelajaran Mata Kuliah:

- ☐ Mahasiswa mampu memahami dan menerapkan konsep-konsep ke dalam HTML.
- ☐ Mahasiswa mampu memilih element dan semantik yang sesuai pada HTML.
- ☐ Mahasiswa mampu menerapkan konsep tipografi dan multimedia pada halaman HTML.
- ☐ Mahasiswa mampu memilih CSS format yang sesuai untuk diterapkan ke dalam HTML.
- ☐ Mahasiswa mampu merencanakan penggunaan GIT untuk kolaborasi dalam proses pengembangan aplikasi.
- ☐ Mahasiswa mampu menerapkan konsep dari layout dan grid pada CSS.
- ☐ Mahasiswa mampu membangun sebuah halaman web yang responsive secara design dengan menggunakan media queries.
- ☐ Mahasiswa mampu menerapkan penggunaan dari list untuk pembuatan daftar pada halaman web.
- ☐ Mahasiswa mampu membangun tabel dan form untuk proses pendataan pada halaman web.
- ☐ Mahasiswa mampu menyusun web dengan menerapkan konsep javascript.



UNIT 3

STATE DAN PROPS

UNIT OVERVIEW

Pada minggu ini, kita akan mendalami dua konsep kunci dalam pengembangan aplikasi React, yaitu "State dan Props". Dalam React, setiap komponen memiliki 'state' dan 'props' yang memungkinkan kita untuk mengelola data dan interaksi antar komponen dengan lebih efisien. Melalui pemahaman mendalam tentang state dan props, kita akan dapat membangun aplikasi React yang responsif dan dinamis, serta memahami alur data dan komunikasi antar komponen..

UNIT OBJECTIVES

Pada minggu ini, mahasiswa akan fokus pada praktik langsung dalam menerapkan konsep "State dan Props" pada aplikasi React. Berikut adalah capaian yang diharapkan setelah menyelesaikan topik ini:

1. Memahami Konsep State: Mahasiswa akan mampu menjelaskan apa itu state, bagaimana cara kerjanya, dan mengapa state penting dalam pengembangan aplikasi React.
2. Memahami Konsep Props: Mahasiswa akan mampu mendefinisikan props, memahami perbedaan antara state dan props, serta bagaimana props memungkinkan komunikasi antar komponen.
3. Menerapkan State dan Props dalam Aplikasi: Mahasiswa akan mampu mengimplementasikan state dan props dalam aplikasi React, memanfaatkan kedua konsep tersebut untuk membangun antarmuka pengguna yang dinamis dan responsif.
4. Mengelola Alur Data dengan State dan Props: Mahasiswa akan memahami bagaimana data mengalir dalam aplikasi React, bagaimana state dan props mempengaruhi alur data tersebut, dan bagaimana mengelola perubahan data dengan benar.
5. Menerapkan Event Handling dengan State: Mahasiswa akan mampu mengintegrasikan event handling dengan state untuk menciptakan interaksi pengguna yang dinamis dan responsif.

UNIT CONTENTS

Lesson 1: Komponen dan Jsx	5 - 6
Lesson 2: Styling dasar Komponen React	6 - 7
Lesson 3: Menyusun Komponen React	7 - 9

PRE LAB

Pre lab, berisi pertanyaan mendasar terkait teori materi yang sedang diajarkan.

QUESTION

- ☐ Apa yang dimaksud dengan komponen dalam React, dan mengapa konsep ini penting dalam pengembangan aplikasi React?
- ☐ Apa itu JSX (JavaScript XML), dan bagaimana JSX digunakan untuk mendefinisikan elemen-elemen dalam tampilan antarmuka pengguna React?
- ☐ Apa perbedaan antara komponen fungsional dan komponen kelas dalam React? Bagaimana cara Anda memilih antara keduanya dalam pengembangan aplikasi?
- ☐ Apa itu props (properti) dalam React, dan bagaimana Anda mengirim data antara komponen menggunakan props?
- ☐ Bagaimana menggabungkan CSS atau Style dalam komponen React?

CONTENT LESSON

CASE STUDY / PROJECT

Perusahaan ABC adalah sebuah organisasi yang bergerak di bidang manajemen proyek. Dengan bertambahnya jumlah proyek dan tugas yang harus dikelola, tim manajemen merasa kesulitan untuk melacak semua tugas yang harus diselesaikan. Mereka membutuhkan sebuah sistem sederhana yang memungkinkan anggota tim untuk mencatat, melacak, dan mengelola tugas-tugas yang harus dikerjakan.

Sebagai solusi, organisasi ABC memutuskan untuk membangun aplikasi to-do list. Aplikasi ini akan memungkinkan setiap anggota tim untuk menambahkan tugas, menandai tugas yang telah selesai, dan menghapus tugas yang tidak lagi relevan. Dengan demikian, seluruh tim dapat bekerja dengan lebih efisien dan terorganisir.

Bantulah organisasi ABC untuk membangun aplikasi to-do list ini agar mereka dapat meningkatkan produktivitas dan efisiensi dalam pengelolaan tugas-tugas mereka.

To-Do List

Tambahkan

Belajar React

Hapus

Beli perlengkapan toko

Hapus

Tambahkan Produk baru

Hapus

To-Do List

Tambahkan

☒ ~~Tambah karyawan~~

Hapus

☒ ~~Gek stok barang~~

Hapus

☐ Hitung pendapatan

Hapus

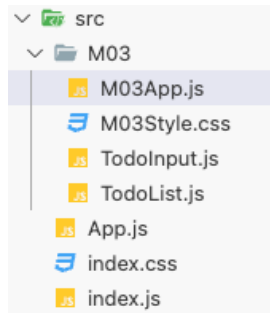
SOLUTION

Berikut kita akan menerapkan state dan props dalam react secara bertahap.

INSTRUCTION

Silahkan ikuti langkah-langkah berikut untuk menyelesaikan permasalahan yang kalian buat:

1. Gunakan folder project react sebelumnya.
2. Buka project react menggunakan vscode dan jalankan Aplikasi React dengan **npm start**.
3. Perhatikan penjelasan dosen untuk struktur web app react.
4. Setelah semua siap, implementasikan struktur folder, dan file berikut :
Buat struktur folder dan file berikut pada project anda :



5. Buat komponen **TodoInput.js** berikut (abaikan komentar, jangan diketik):

```
1 // Mengimpor modul yang diperlukan dari React
2 import React, { useState } from "react";
3
4 // Mendefinisikan komponen TodoInput sebagai komponen berbasis fungsi
5 const TodoInput = (props) => {
6   // Menggunakan hook useState untuk mendefinisikan state `inputValue` dan fungsi `setInputValue` untuk mengubah state tersebut
7   const [inputValue, setInputValue] = useState("");
8
9   // Fungsi yang akan dipanggil ketika form disubmit
10  const handleSubmit = (e) => {
11    // Mencegah perilaku default dari form (misalnya refresh halaman)
12    e.preventDefault();
13
14    // Memastikan bahwa inputValue tidak kosong atau hanya berisi spasi
15    if (inputValue.trim()) {
16      // Memanggil fungsi addTodo dari props dengan nilai inputValue sebagai argumen
17      props.addTodo(inputValue);
18
19      // Mengatur ulang inputValue menjadi string kosong setelah todo ditambahkan
20      setInputValue("");
21    }
22  };
23
24  // Render komponen
25  return (
26    <form onSubmit={handleSubmit}>
27      { /* Input field untuk menambahkan todo baru.
28       * Nilai dari input diatur oleh state `inputValue` dan setiap perubahan pada input akan mengubah state tersebut */ }
29      <input
30        type="text"
31        value={inputValue}
32        onChange={(e) => setInputValue(e.target.value)}
33        placeholder="Tambahkan Todo..."
34      />
35
36      { /* Tombol untuk menambahkan todo */ }
37      <button type="submit">Tambahkan</button>
38    </form>
39  );
40 };
41
42 // Mengekspor komponen TodoInput agar dapat digunakan di tempat lain
43 export default TodoInput;
```

6. Buat komponent **TodoList.js** (abaikan komentar):



```
1  // Mengimpor modul yang diperlukan dari React
2  import React from "react";
3
4  // Mendefinisikan komponen TodoList sebagai komponen berbasis fungsi
5  const TodoList = (props) => {
6    // Render komponen
7    return (
8      <ul>
9        {/* Menggunakan metode map untuk mengiterasi melalui array todos dari props
10         dan mengembalikan elemen li untuk setiap todo */}
11        {props.todos.map((todo, index) => (
12          <li key={index}>
13            {/* Menampilkan teks todo */}
14            {todo}
15
16            {/* Tombol untuk menghapus todo. Ketika diklik,
17             akan memanggil fungsi removeTodo dari props dengan indeks todo sebagai argumen */}
18            <button onClick={() => props.removeTodo(index)}> Hapus </button>
19          </li>
20        ))}
21      </ul>
22    );
23  };
24
25  // Mengekspor komponen TodoList agar dapat digunakan di tempat lain
26  export default TodoList;
27
```

7. Buat komponen M03App.js



```
1 // Mengimpor modul yang diperlukan dari React dan komponen lainnya
2 import React, { useState } from "react";
3 import TodoList from "../TodoList";
4 import TodoInput from "../TodoInput";
5 import "../M03Style.css"
6
7 // Mendefinisikan komponen M03App sebagai komponen berbasis fungsi
8 const M03App = () => {
9   // Menggunakan hook useState untuk mendefinisikan state `todos` dan fungsi `setTodos` untuk mengubah state tersebut
10   const [todos, setTodos] = useState(["Belajar React"]);
11
12   // Fungsi untuk menambahkan todo baru ke dalam daftar
13   const addTodo = (todo) => {
14     setTodos([...todos, todo]);
15   };
16
17   // Fungsi untuk menghapus todo berdasarkan indeksinya
18   const removeTodo = (index) => {
19     const newTodos = [...todos];
20     newTodos.splice(index, 1);
21     setTodos(newTodos);
22   };
23
24   // Render komponen
25   return (
26     <div className="M03App">
27       <h1>To-Do List</h1>
28
29       {/* Komponen input untuk menambahkan todo baru.
30        Fungsi `addTodo` diteruskan sebagai props agar dapat dipanggil dari dalam komponen TodoInput */}
31       <TodoInput addTodo={addTodo} />
32
33       {/* Komponen untuk menampilkan daftar todo.
34        Daftar `todos` dan fungsi `removeTodo` diteruskan sebagai props agar dapat digunakan dalam komponen TodoList */}
35       <TodoList todos={todos} removeTodo={removeTodo} />
36     </div>
37   );
38 };
39
40 // Mengekspor komponen M03App agar dapat digunakan di tempat lain
41 export default M03App;
42
```

8. Buat style css pada file M03Style.css



```
1  .M03App {
2    font-family: Arial, sans-serif;
3    max-width: 400px;
4    margin: 50px auto;
5    padding: 20px;
6    border: 1px solid #ddd;
7    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
8  }
9
10 h1 {
11   text-align: center;
12 }
13
14 ul {
15   list-style-type: none;
16   padding: 0;
17 }
18
19 li {
20   display: flex;
21   justify-content: space-between;
22   align-items: center;
23   padding: 10px 0;
24 }
25
26 button {
27   margin-left: 10px;
28   padding: 5px 10px;
29   background-color: #ff6347;
30   color: #fff;
31   border: none;
32   cursor: pointer;
33 }
34
35 button:hover {
36   background-color: #ff4500;
37 }
38
```


9. Ganti isi untuk memanggil M03App pada App.js



```
1  import M03App from "./M03/M03App";
2
3  function App() {
4    return (
5      <M03App />
6    );
7  }
8
9  export default App;
10
```

10. Lihat hasil source code di atas dengan memastikan koding di atas telah **disimpan** dan aplikasi react telah dijalankan dengan **npm start**

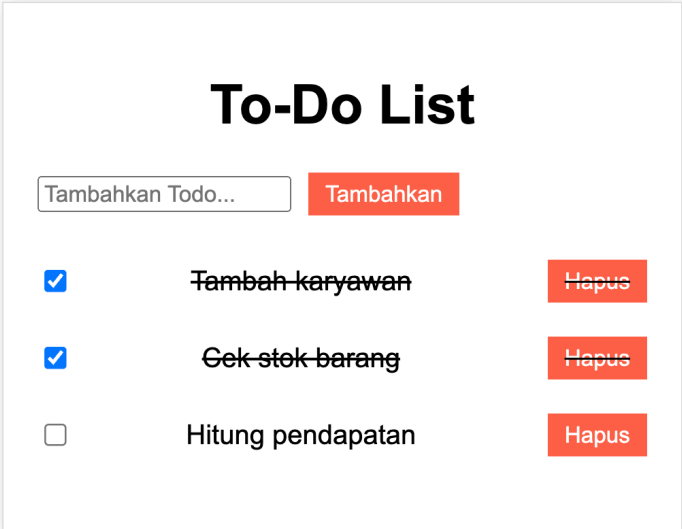
EXERCISE

EXERCISE OBJECTIVES

Pada latihan ini, mahasiswa diharapkan mampu untuk:

- Membuat website sederhana dengan memanfaatkan state, props pada React.

TASK 1: TAMBAHKAN FITUR YANG MEMUNGKINKAN PENGGUNA UNTUK MENANDAI TODO SEBAGAI "SELESAI". SETIAP TODO HARUS MEMILIKI CHECKBOX ATAU TOMBOL YANG, KETIKA DIKLIK, AKAN MENGGANTI TAMPILAN TODO (MISALNYA, DENGAN GARIS TENGAH) UNTUK MENUNJUKKAN BAHWA TELAH SELESAI, SEPERTI GAMBAR BERIKUT:



The screenshot shows a web application titled "To-Do List". At the top, there is a text input field with the placeholder "Tambahkan Todo..." and a red button labeled "Tambahkan". Below this, there is a list of three items. Each item consists of a checkbox, a text label, and a red button labeled "Hapus". The first two items have their checkboxes checked and their text labels are crossed out with a diagonal line. The third item has an unchecked checkbox and its text label is not crossed out.

Checkbox	Item	Action
<input checked="" type="checkbox"/>	Tambah karyawan	Hapus
<input checked="" type="checkbox"/>	Gek stok barang	Hapus
<input type="checkbox"/>	Hitung pendapatan	Hapus

Silahkan lengkapi koding di atas dan buat perubahannya.



UNIVERSITAS
MIKROSKIL