



MODUL WEB PROGRAMMER (REACT JS)

Latihan

1. Buat file InputNama.js dengan output

Input Nama

Nama Anda:

Halo, Denny!

Latihan

```
1  import React, { useState } from 'react';
2
3  function InputNama() {
4    const [nama, setNama] = useState('');
5
6    return (
7      <div>
8        <h1>Input Nama</h1>
9        <form>
10         <div>
11           <label>Nama Anda: </label>
12           <input
13             value={nama}
14             type="text"
15             onChange={(e) => setNama(e.target.value)}
16             placeholder="Masukkan nama"
17           />
18         </div>
19       </form>
20       {nama && (
21         <div>
22           <h2>Halo, {nama}!</h2>
23         </div>
24       )}
25     </div>
26   );
27 }
28
29 export default InputNama;
```

Latihan

2. Buat seperti berikut

Calculator

Input 1:

Input 2:

Hasil: 20

React Props (Properties)

Di React, Props (Properties) adalah mekanisme untuk mengirimkan data dari komponen induk (parent) ke komponen anak (child). Props memungkinkan kita untuk membuat komponen yang dapat digunakan kembali (reusable) dengan memberikan data yang berbeda untuk setiap instance-nya.

1. Contoh Dasar Props

```
JS ChildComponent.js U ×
src > JS ChildComponent.js > ...
1  import React from 'react';
2
   Tabnine | Edit | Test | Explain | Document | Ask | Open ChildComponent in Preview.js
3  function ChildComponent(props) {
4    return (
5      <div>
6        <h2>Komponen Anak</h2>
7        <p>Nama: {props.name}</p>
8        <p>Usia: {props.age}</p>
9      </div>
10   );
11 }
12
13 export default ChildComponent;
14
```

React Props (Properties)

JS ParentComponent.js U X

src > JS ParentComponent.js > ...

```

1  import React from 'react';
2  import ChildComponent from './ChildComponent';
3
4  function ParentComponent() {
5      return (
6          <div>
7              <h1>Komponen Induk</h1>
8              <ChildComponent name="John" age={30} />
9          </div>
10     );
11 }
12
13 export default ParentComponent;
14
    
```

Komponen Induk

Komponen Anak

Nama: John

Usia: 30

React Props (Properties)

2. Props dengan Fungsi Callback

JS ChildCallbackComponent.js U X

src > JS ChildCallbackComponent.js > ...

```
1 import React from 'react';
```

```
2
```

Tabnine | Edit | Test | Explain | Document | Ask | Open ChildCallbackComponent in Preview.js

```
3 function ChildCallbackComponent({ onMessageChange }) {
```

```
4   return (
```

```
5     <div>
```

```
6       <h2>Komponen Anak</h2>
```

```
7       <button onClick={() => onMessageChange('Pesan baru dari anak!')}>
```

```
8         Kirim Pesan ke Induk
```

```
9       </button>
```

```
10    </div>
```

```
11  );
```

```
12 }
```

```
13
```

```
14 export default ChildCallbackComponent;
```

```
15
```

React Props (Properties)

JS ParentCallbackComponent.js U X

src > JS ParentCallbackComponent.js > ...

```
1 import React, { useState } from 'react';
2 import ChildCallbackComponent from './ChildCallbackComponent';
3
4 function ParentCallbackComponent() {
5   const [message, setMessage] = useState('');
6
7   const handleMessageChange = (newMessage) => {
8     setMessage(newMessage);
9   };
10
11   return (
12     <div>
13       <h1>Komponen Induk</h1>
14       <ChildCallbackComponent onMessageChange={handleMessageChange} />
15       <p>Pesan dari Anak: {message}</p>
16     </div>
17   );
18 }
19
20 export default ParentCallbackComponent;
```

Komponen Induk

Komponen Anak

Kirim Pesan ke Induk

Pesan dari Anak: Pesan baru dari anak!

React Hooks

- Di React, hooks adalah fungsi-fungsi yang memungkinkan kita untuk menggunakan fitur-fitur React (seperti state, effect, context, dll.) di dalam komponen fungsional.
- Sebelum hooks diperkenalkan pada React versi 16.8, untuk mengelola state dan metode siklus hidup, kita harus menggunakan Class Component. Hooks memungkinkan kita melakukan semua hal yang sebelumnya hanya bisa dilakukan di komponen kelas (seperti state, metode siklus hidup, konteks, dll.) di dalam komponen fungsional yang lebih sederhana dan lebih ringkas.

React Hooks

- **useState** adalah hook yang memungkinkan kita untuk menambahkan state ke dalam komponen fungsional.

Bentuk :

const [state, setState] = useState(initialState);

state = nilai saat ini

setState = Fungsi yang digunakan untuk memperbarui state. Fungsi ini dapat menerima nilai state baru secara langsung, atau sebuah fungsi yang mengembalikan state baru berdasarkan state sebelumnya.

initialState = Nilai awal state (dapat berupa angka, string, objek, array, atau nilai lainnya)

Contoh:

const [count, setCount] = useState(0); // menginisialisasi state dengan angka

React Hooks

- useEffect adalah hook yang memungkinkan kita untuk melakukan efek samping di dalam komponen fungsional, seperti pengambilan data, subscribe, atau manipulasi DOM secara manual.

Bentuk:

```
useEffect(() => {  
    // Kode yang akan dijalankan saat komponen di-render  
}, [dependencies]);
```

- Argumen pertama adalah fungsi callback yang akan dijalankan saat komponen pertama kali dirender atau ketika ada perubahan pada dependensi.
- Argumen kedua adalah array dependensi. Jika ada nilai dalam array ini yang berubah, efek tersebut akan dijalankan ulang. Jika kita memberikan array kosong, efek hanya akan dijalankan sekali setelah render pertama.

React Hooks

```
1 import React, { useState, useEffect } from 'react';
2
3 Tabnine | Edit | Test | Explain | Document | Ask | Open ExampleComponent in Preview.js
4 function ExampleComponent() {
5   const [message, setMessage] = useState("Hello, world!");
6
7   // Efek yang hanya dijalankan saat pertama kali komponen dimuat
8   useEffect(() => {
9     console.log("Komponen ExampleComponent sudah dirender!");
10    // Cleanup function yang dijalankan saat komponen di-unmount
11    return () => {
12      console.log("Komponen ExampleComponent akan di-unmount!");
13    };
14  }, []); // Menggunakan array kosong agar hanya dijalankan sekali pada mount dan unmount
15
16  // Efek yang dijalankan ketika nilai message berubah
17  useEffect(() => {
18    console.log("State message berubah:", message);
19  }, [message]);
20
21  return (
22    <div>
23      <p>{message}</p>
24      <button onClick={() => setMessage("Hello, React!")}>Ubah Pesan</button>
25    </div>
26  );
27 export default ExampleComponent;
```

React Hooks

Selamat datang di Aplikasi React!

Hello, React!

Ubah Pesan

not available

✕ Expression

not available

Komponen ExampleComponent sudah dirender!
State message berubah: Hello, world!
Komponen ExampleComponent akan di-unmount!
Komponen ExampleComponent sudah dirender!
State message berubah: Hello, world!
State message berubah: Hello, React!
Komponen ExampleComponent akan di-unmount!
Komponen ExampleComponent sudah dirender!
State message berubah: Hello, React!

Single Page Application

- Single Page Application (SPA) adalah jenis aplikasi web di mana seluruh aplikasi dimuat dalam satu halaman HTML, dan interaksi dengan aplikasi tidak memerlukan pemuatan ulang halaman dari server. Sebagai gantinya, konten dan data diperbarui dinamis di dalam halaman tersebut, memberikan pengalaman pengguna yang lebih cepat dan lancar.
- React, sebagai pustaka JavaScript untuk membangun antarmuka pengguna (UI), dirancang untuk mengembangkan SPA secara efisien. Di React, aplikasi umumnya dimulai dengan memuat satu file HTML, dan kemudian hanya data atau komponen UI yang berubah ketika pengguna berinteraksi dengan aplikasi, tanpa melakukan refresh halaman secara keseluruhan.

Konsep SPA di React

1. Routing dan Navigasi tanpa Refresh Halaman:

- Di aplikasi SPA, ketika Anda mengklik tautan atau berinteraksi dengan elemen UI yang seharusnya mengarah ke halaman lain, React tidak akan memuat halaman baru dari server. Sebagai gantinya, React akan mengganti bagian UI yang relevan dengan komponen yang baru tanpa me-refresh seluruh halaman.
- Untuk mengimplementasikan navigasi dalam SPA, kita biasanya menggunakan pustaka seperti React Router. React Router memungkinkan kita untuk menentukan rute yang berbeda (URL) dalam aplikasi, dan akan mengganti komponen yang sesuai dengan rute tersebut tanpa memuat ulang halaman.

Konsep SPA di React

2. Komponen Dinamis dan Pembaruan UI

React menggunakan konsep komponen untuk membagi UI aplikasi menjadi bagian-bagian yang lebih kecil dan dapat digunakan kembali. Komponen React bertanggung jawab untuk menampilkan dan mengelola bagian UI tertentu.

1. Saat state atau props dalam komponen berubah, React akan secara otomatis merender ulang hanya bagian yang terpengaruh, bukan seluruh halaman.
2. Ini berarti bahwa aplikasi tetap responsif dan cepat karena hanya bagian UI yang relevan yang diperbarui, bukan seluruh halaman.

Konsep SPA di React

3. Pemrograman Deklaratif dengan JSX

- React memungkinkan pengembangan aplikasi dengan pendekatan deklaratif, di mana Anda menggambarkan bagaimana UI harus terlihat berdasarkan state aplikasi, dan React akan memastikan UI diperbarui ketika state berubah.
- Anda menulis UI menggunakan JSX (JavaScript XML), yang merupakan sintaks untuk mendeskripsikan elemen HTML di dalam JavaScript. JSX memungkinkan Anda menulis kode deklaratif untuk elemen dan komponen UI dengan cara yang lebih mudah dibaca dan dipahami.

Konsep SPA di React

4. Manajemen State Global dan Lokal

- Dalam SPA, status (state) aplikasi sangat penting karena status menentukan bagaimana UI akan tampil. React memungkinkan Anda untuk mengelola status menggunakan `useState`, `useReducer`, dan solusi manajemen status global seperti Redux atau Context API.
- State Lokal: Mengelola data atau informasi yang relevan untuk komponen tertentu.
- State Global: Mengelola data atau informasi yang perlu diakses oleh banyak komponen dalam aplikasi, misalnya menggunakan Context API atau Redux.

Konsep SPA di React

5. Loading Dinamis (Lazy Loading)

- Aplikasi SPA cenderung lebih ringan dan lebih cepat karena hanya bagian UI yang dimuat sesuai kebutuhan.
- Lazy loading adalah teknik di mana komponen atau sumber daya tertentu hanya dimuat ketika diperlukan. Di React, Anda dapat menggunakan `React.lazy()` dan `Suspense` untuk memuat komponen secara dinamis hanya saat dibutuhkan, mengurangi waktu pemuatan aplikasi awal.

Keuntungan SPA di React

1. Pengalaman Pengguna yang Cepat:

- SPA memberikan pengalaman pengguna yang lebih mulus dan interaktif karena aplikasi hanya memuat konten yang dibutuhkan tanpa me-refresh halaman secara keseluruhan.
- Hal ini mengurangi latensi dan membuat aplikasi lebih responsif.

2. Pengurangan Pemanggilan HTTP:

- Setelah aplikasi dimuat pertama kali, hampir semua interaksi selanjutnya hanya melibatkan pemanggilan API untuk memperbarui data, bukan memuat ulang halaman lengkap dari server.

Keuntungan SPA di React

3. Penggunaan Cache dan State:

- Karena aplikasi tidak memuat ulang halaman sepenuhnya, data dan state yang telah dimuat sebelumnya tetap tersedia selama sesi pengguna, yang dapat mengurangi waktu loading dan meningkatkan efisiensi.

4. Navigasi Cepat:

- Karena halaman tidak dimuat ulang secara keseluruhan, navigasi antar halaman atau tampilan dalam aplikasi bisa sangat cepat.

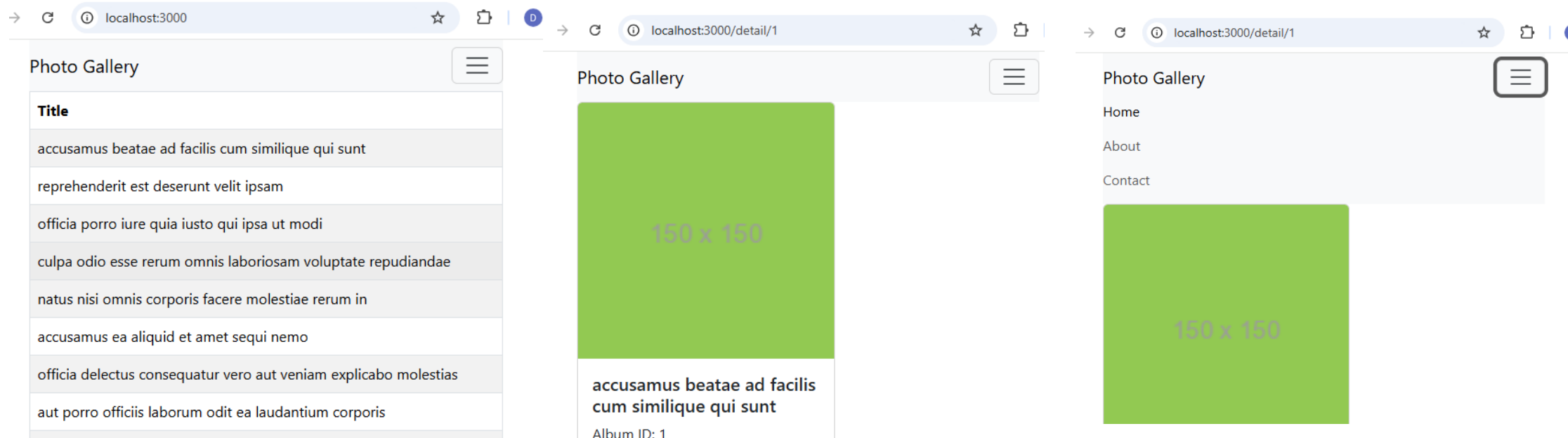
5. Pengelolaan UI dengan Komponen:

- Penggunaan komponen yang dapat digunakan kembali di React membuat pengelolaan dan pembaruan UI menjadi lebih terstruktur dan mudah dipelihara.



Praktik Single Page Application dan akses REST API

- Akan dibuat sebuah aplikasi menampilkan data photos dari API <https://jsonplaceholder.typicode.com/>
- Library HTTP client menggunakan Axios : <https://axios-http.com/docs/intro>
- Menggunakan react-bootstrap : <https://react-bootstrap.netlify.app/>
- Router menggunakan react-router-dom : <https://reactrouter.com/en/main>



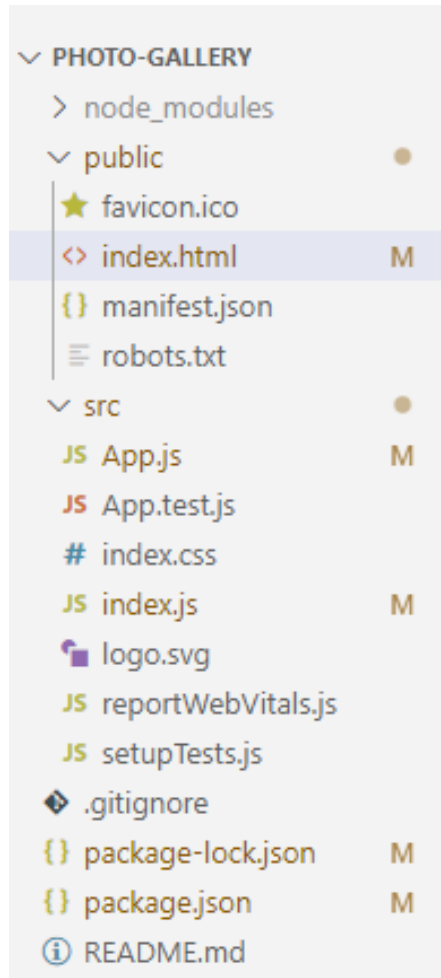
Praktik Single Page Application dan akses REST API

1. Buat project baru (di folder belajar_react) melalui cmd : `npx create-react-app photo-gallery`
2. Buka VSCode, arahkan ke folder photo-gallery
3. Buka terminal , install dependencies:
`npm install axios react-bootstrap bootstrap react-router-dom react-router-bootstrap`
4. Import css di index.js : `import 'bootstrap/dist/css/bootstrap.min.css';`

```
JS index.js M X
src > JS index.js > ...
  You, 10 minutes ago | 1 author (You)
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import 'bootstrap/dist/css/bootstrap.min.css';
4  import App from './App';
5  import reportWebVitals from './reportWebVitals';
6
7  const root = ReactDOM.createRoot(document.getElementById('root'));
8  root.render(
9    // React.StrictMode diperlukan ketika development mode, ketika production mode dihilangkan
10    <React.StrictMode>
11      { /* App is the root component of your React app */ }
12    <App />
13  </React.StrictMode>
14  );
15  // Runs performance checks and sends results, can be used to log or report performance metrics
16  reportWebVitals(console.log);
17
```

Praktik Single Page Application dan akses REST API

5. Delete file yang tidak perlu



Praktik Single Page Application dan akses REST API

6. index.html, sesuaikan

```
<> index.html M X
public > <> index.html > html
You, 1 second ago | 1 author (You)
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="utf-8" />
5      <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6      <meta name="viewport" content="width=device-width, initial-scale=1" />
7      <meta
8        name="description"
9        content="Photo Gallery App"
10     />
11     <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
12     <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
13     <title>Photo Gallery App</title>
14   </head>
15   <body>
16     <noscript>You need to enable JavaScript to run this app.</noscript>
17     <div id="root"></div>
18   </body>
19 </html> You, 1 second ago • Uncommitted changes
```

Praktik Single Page Application dan akses REST API

7. App.js

JS App.js M X

src > JS App.js > ...

You, 4 minutes ago | 1 author (You)

```
1 import React from 'react';
2 import 'bootstrap/dist/css/bootstrap.min.css';
3 import { Container, Button } from 'react-bootstrap';
4
5 function App() {
6   return (
7     <Container className="p-3">
8       <div className="p-5 mb-4 bg-light rounded-3">
9         <h1>Welcome to the Photo Gallery!</h1>
10        <p>You can search photos by title here</p>
11        <p>
12          <Button variant="primary">Learn more</Button>
13        </p>
14      </div>
15    </Container>
16  );
17 }
18
19 export default App;
```

You, 18 minutes ago • Initialize project using Create React

Praktik Single Page Application dan akses REST API

8. Jalankan, npm start



localhost:3000

Welcome to the Photo Gallery!

You can search photos by title here

[Learn more](#)

Praktik Single Page Application dan akses REST API

9. Buat Home.js

JS Home.js U X

src > JS Home.js > [default]

```
1 import { Button, Container } from "react-bootstrap";
2 import 'bootstrap/dist/css/bootstrap.min.css';
3
4 function Home(){
5   return (
6     <Container className="p-3">
7       <div className="p-5 mb-4 bg-light rounded-3">
8         <h1>Welcome to the Photo Gallery!</h1>
9         <p>You can search photos by title here</p>
10        <p>
11          <Button variant="primary">Learn more</Button>
12        </p>
13      </div>
14    </Container>
15  );
16 }
17 export default Home;
```

Praktik Single Page Application dan akses REST API

10. App.js, update

JS App.js M X

src > JS App.js > ...

You, 44 seconds ago | 1 author (You)

```
1 import React from 'react';
2 import 'bootstrap/dist/css/bootstrap.min.css';
3 import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
4 import Home from './Home';
5
```

Tabnine | Edit | Test | Explain | Document | Ask | Open App in Preview.js

```
6 function App() {
7   return (
8     <Router>
9       <Routes>
10        <Route path="/" element={<Home />} />
11      </Routes>
12    </Router>
13  );
14 }
```

You, 24 minutes ago • Initialize project using Create React App

```
16 export default App;
```

Praktik Single Page Application dan akses REST API

11. Home.js, update

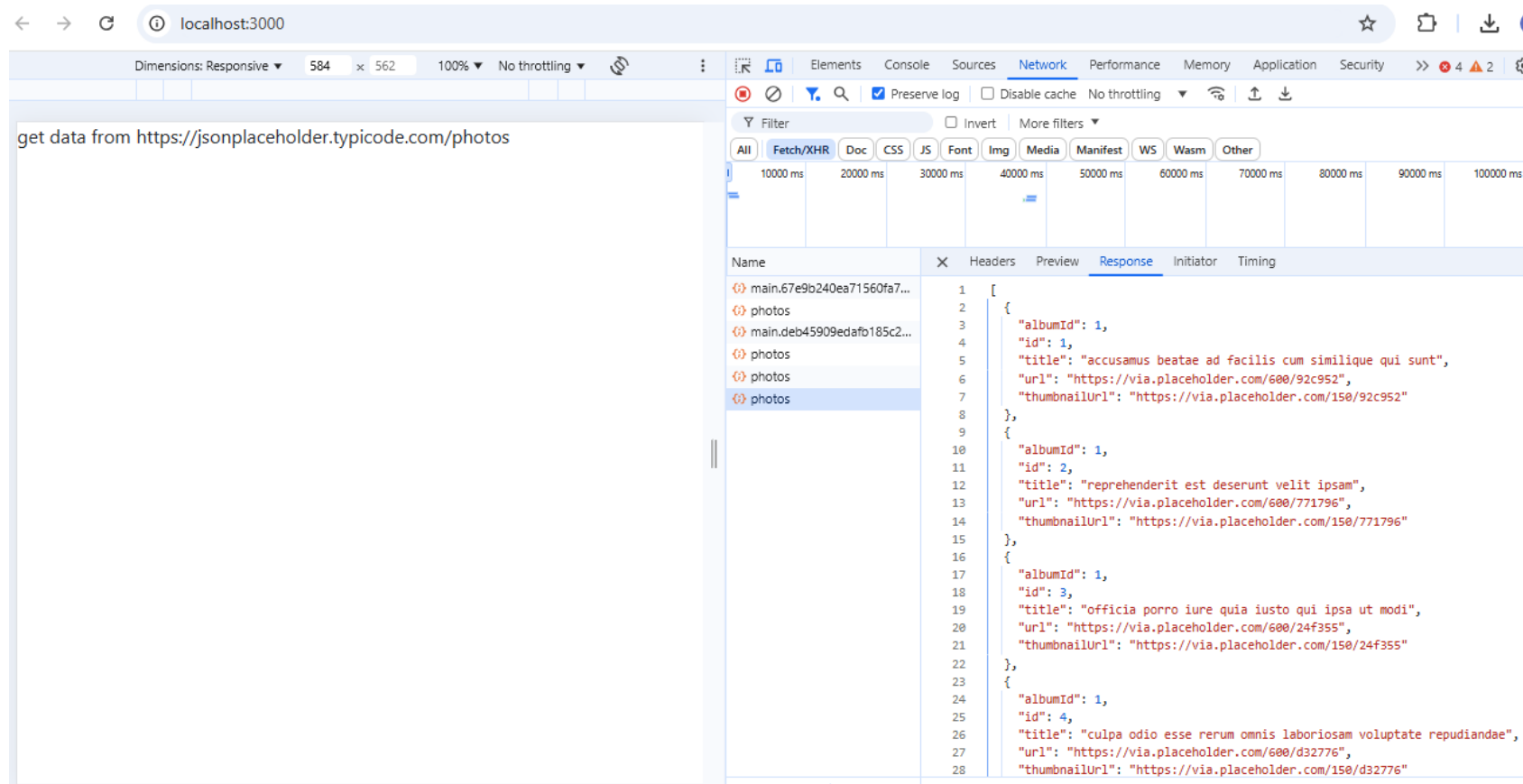
```

1  import 'bootstrap/dist/css/bootstrap.min.css';
2  import { useEffect, useState } from "react";
3  import axios from "axios";
4  |
  Tabnine | Edit | Test | Explain | Document | Ask
5  function Home(){
6    | Open Home in Preview.js
7    const [photos, setPhotos] = useState([]);
8
9    useEffect(() => {
10     | axios.get('https://jsonplaceholder.typicode.com/photos')
11     |   .then(response => setPhotos(response.data));
12   }, []);
13
14   useEffect(() => {
15     | console.log(photos);
16   }, [photos]);
17
18   return (
19     | <div>
20     |   | get data from https://jsonplaceholder.typicode.com/photos
21     |   | </div>
22   );
23 }
24 export default Home;

```

Praktik Single Page Application dan akses REST API

12. Cek di browser



Praktik Single Page Application dan akses REST API

13. Home.js, update

```
useEffect(() => {  
  axios.get('https://jsonplaceholder.typicode.com/photos')  
    .then(response => {  
      setPhotos(response.data.slice(0, 10)); // Only take the first 100 items  
    })  
    .catch(error => console.error(error));  
}, []);
```


Praktik Single Page Application dan akses REST API

14. Home.js, update

```
14   return (  
15     <Table striped bordered hover>  
16       <thead>  
17         <tr>  
18           <th>Title</th>  
19         </tr>  
20       </thead>  
21       <tbody>  
22         {photos.map(photo => (  
23           <tr>  
24             <td>{photo.title}</td>  
25           </tr>  
26         ))}  
27       </tbody>  
28     </Table>  
29   );  
30 }  
31 export default Home;
```

Praktik Single Page Application dan akses REST API

15. Hasil

Title
accusamus beatae ad facilis cum similique qui sunt
reprehenderit est deserunt velit ipsam
officia porro iure quia iusto qui ipsa ut modi
culpa odio esse rerum omnis laboriosam voluptate repudiandae
natus nisi omnis corporis facere molestiae rerum in
accusamus ea aliquid et amet sequi nemo
officia delectus consequatur vero aut veniam explicabo molestias
aut porro officiis laborum odit ea laudantium corporis
qui eius qui autem sed
beatae et provident et ut vel

Praktik Single Page Application dan akses REST API

16. Buat Detail.js

JS Detail.js U X

src > JS Detail.js > Detail

```
1 import React, { useEffect, useState } from 'react';
2 import axios from 'axios';
3 import { Card } from 'react-bootstrap';
4 import { useParams } from 'react-router-dom';
5 function Detail() {
6   const { id } = useParams();
7   const [photo, setPhoto] = useState(null);
8
9   useEffect(() => {
10     axios.get(`https://jsonplaceholder.typicode.com/photos/${id}`)
11       .then(response => setPhoto(response.data));
12   }, [id]);
13
14   return photo ? (
15     <Card style={{ width: '18rem' }}>
16       <Card.Img variant="top" src={photo.thumbnailUrl} />
17       <Card.Body>
18         <Card.Title>{photo.title}</Card.Title>
19         <Card.Text>Album ID: {photo.albumId}</Card.Text>
20       </Card.Body>
21     </Card>
22   ) : (
23     <p>Loading...</p>
24   );
25 };
26 export default Detail;
```

Praktik Single Page Application dan akses REST API

17. App.js, update

Tabnine | Edit | Test | Explain | Document | Ask | Open App in Preview.js

```

7  function App() {
8    return (
9      <Router>
10       <Routes>
11         <Route path="/" element={<Home />} />
12         <Route path="/detail/:id" element={<Detail />} />
13       </Routes>
14     </Router>
15   );
16 }
17
18 export default App;

```

You, 6 min

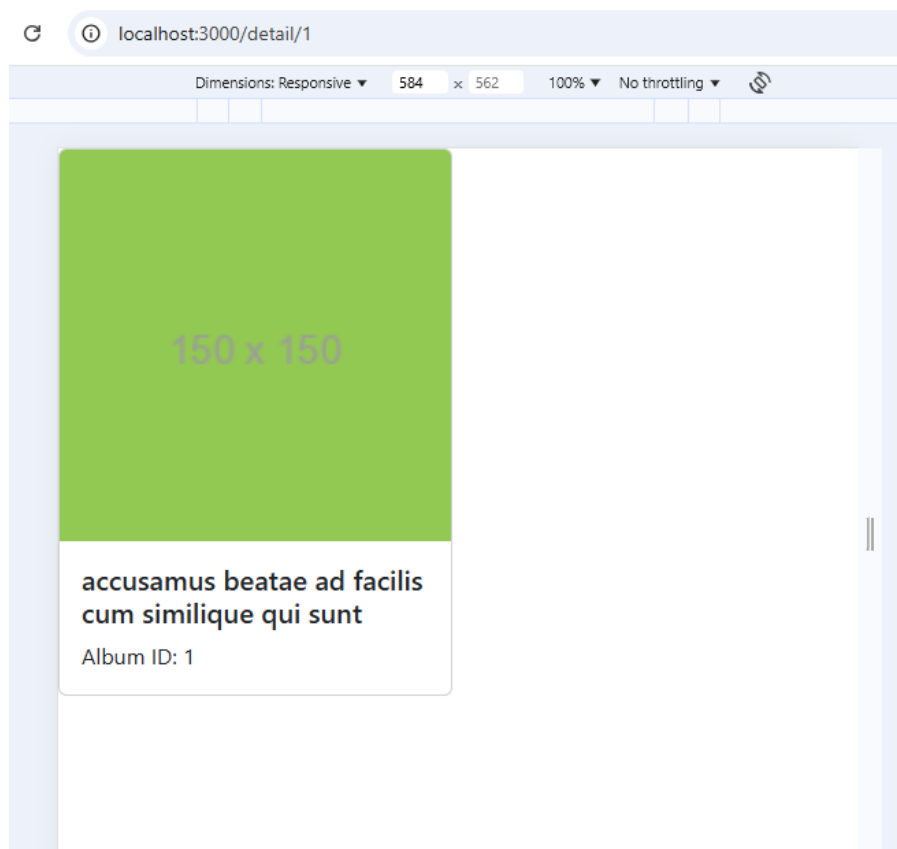
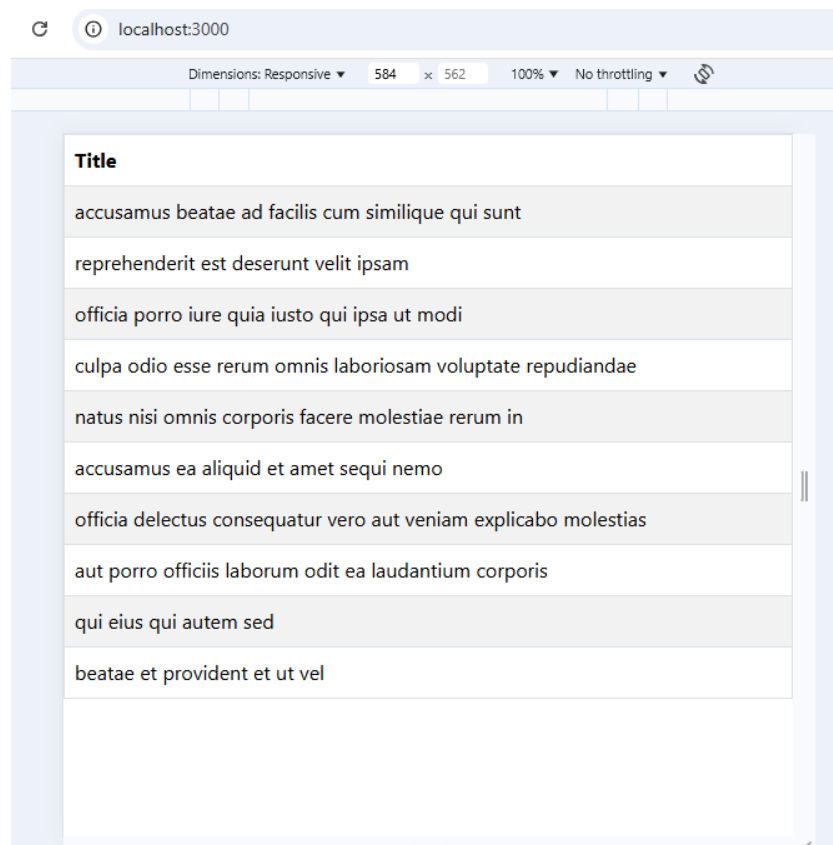
Praktik Single Page Application dan akses REST API

18. Home.js, update

```
JS Home.js U x
src > JS Home.js > Home
7  function Home(){
9    const navigate = useNavigate();
10
11    useEffect(() => {
12      axios.get('https://jsonplaceholder.typicode.com/photos')
13        .then(response => {
14          setPhotos(response.data.slice(0, 10)); // Only take the first 100 items
15        })
16        .catch(error => console.error(error));
17    }, []);
18
19    return (
20      <Table striped bordered hover>
21        <thead>
22          <tr>
23            <th>Title</th>
24          </tr>
25        </thead>
26        <tbody>
27          {photos.map(photo => (
28            <tr key={photo.id} onClick={() => navigate(`/detail/${photo.id}`)}>
29              <td>{photo.title}</td>
30            </tr>
31          ))}
32        </tbody>
33      </Table>
34    );
35  }
```

Praktik Single Page Application dan akses REST API

19. Hasil



Praktik Single Page Application dan akses REST API

20. Buat Header.js

```
JS Header.js U x
src > JS Header.js > Header
1 import React from "react";
2 import { Navbar, Nav } from "react-bootstrap";
3 import { LinkContainer } from "react-router-bootstrap";
  Tabnine | Edit | Test | Explain | Document | Ask | Open Header in Preview.js
4 function Header() {
5   return (
6     <Navbar bg="light" expand="lg">
7       <LinkContainer to="/">
8         <Navbar.Brand>Photo Gallery</Navbar.Brand>
9       </LinkContainer>
10      <Navbar.Toggle aria-controls="basic-navbar-nav" />
11      <Navbar.Collapse id="basic-navbar-nav">
12        <Nav className="ml-auto">
13          <LinkContainer to="/">
14            <Nav.Link>Home</Nav.Link>
15          </LinkContainer>
16          <LinkContainer to="/about">
17            <Nav.Link>About</Nav.Link>
18          </LinkContainer>
19          <LinkContainer to="/contact">
20            <Nav.Link>Contact</Nav.Link>
21          </LinkContainer>
22        </Nav>
23      </Navbar.Collapse>
24    </Navbar>
25  );
26
27 export default Header;
```

Praktik Single Page Application dan akses REST API

21. App.js, update

```
JS App.js M ●
src > JS App.js > ...
...
1 import React from 'react';
2 import 'bootstrap/dist/css/bootstrap.min.css';
3 import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
4 import Home from './Home';
5 import Detail from './Detail';
6 import Header from './Header';
7
Tabnine | Edit | Test | Explain | Document | Ask
8 function App() {
9   return (
10     <Router>
11       <div className="container">
12         <Header />
13         <Routes>
14           <Route path="/" element={<Home />} />
15           <Route path="/detail/:id" element={<Detail />} />
16         </Routes>
17       </div>
18     </Router>
19   );
20 }
21
22 export default App;
```


Praktik Single Page Application dan akses REST API

22. Tambahkan About.js, Contact.js

23. Tambahkan di App.js route nya

```
lighthouse | Edit | Test | Explain | Document | Ask | Open App in Preview.js
9  function App() {
10    return (
11      <Router>
12        <div className="container">
13          <Header />
14          <Routes>
15            <Route path="/" element={<Home />} />
16            <Route path="/about" element={<About />} />
17            <Route path="/detail/:id" element={<Detail />} />
18          </Routes>
19        </div>
20      </Router>
21    );
22  }
```

Tugas I

- Buat project baru my-profile-app
- Isinya adalah profile pribadi (website pribadi)
- Minimal terdiri dari 4 Page, misal : Home, About Me, Contact, Projects
- Gunakan layout yang menarik, foto diri, bisa melihat ide di internet
- Bisa menggunakan library seperti:
 - Bootstrap : <https://react-bootstrap.netlify.app/>
 - MUI : <https://mui.com/>
 - Ant Design : <https://ant.design/>
 - Prime React: <https://primereact.org/>
 - Atau CSS sendiri

terima kasih njengkepi parikan
 tarima kase terime kaseh
 teghemo kasih nerima nihan
 kurre sumanga' saohagölö
 rutam nuwus kasuwun lias ate
 tarimo kasih teghima kasih
 ngatur nuhun sukur dofu dangke
 ta' kabessa nyo'on surak sabeu
 tarima kasi teurimong geunaseh makase suwun
 epang gawang keso'on tiba teing sura'
 makaseh makapulu sama' mokaseh
 tumpu lalo matur nuwun e saparauni
 tampi asih bujur
 matur suksma sukur moanto
 obrigado diate tupa berijin brejen
 odu'olo terima kaseh kurru sumange
 mauliate waniyam walidi'a
 tumpuno laloku sinmung
 hatur nuhun tampeasu
 minta rela kasumasa
 tinatauan mator sekelangkong
 masurak bagat



BUMN UNTUK
 INDONESIA

Telkom
 Indonesia
the world in your hand