Untuk menggantikan URL API dengan sumber data berita nyata, kita bisa menggunakan **NewsAPI** (<https://newsapi.org/>), yang memungkinkan kita untuk mendapatkan berita terbaru dari berbagai sumber berita global.

Langkah-langkah berikut menjelaskan bagaimana kita bisa mengintegrasikan **NewsAPI** ke dalam aplikasi React yang kamu buat.

**1. Mendaftar untuk API Key dari NewsAPI**

* Kunjungi [NewsAPI](https://newsapi.org/), dan daftar untuk mendapatkan **API Key**.
* API key ini digunakan untuk otentikasi setiap permintaan yang kamu buat ke NewsAPI.

**2. Update Kode untuk Menggunakan NewsAPI**

Setelah kamu mendapatkan **API Key** dari NewsAPI, ganti URL API di kode dengan URL NewsAPI dan sertakan API Key di header permintaan.

Berikut adalah modifikasi kode yang sudah terintegrasi dengan NewsAPI:

**Kode Setelah Mengintegrasikan NewsAPI:**

import React, { Component } from "react";

import axios from "axios"; // Import axios

import 'bootstrap/dist/css/bootstrap.min.css';

import { Button, Card, Col, Container, Form, InputGroup, Row } from "react-bootstrap";

export default class Bootstrap extends Component {

constructor(props) {

super(props);

this.state = {

searchQuery: "", // Query pencarian

newsData: [], // Data berita yang akan ditampilkan

loading: false, // Status loading

error: null, // Pesan error jika ada

};

}

// Method untuk mengambil data berita dari NewsAPI

fetchData = async () => {

this.setState({ loading: true });

const { searchQuery } = this.state;

const apiKey = "YOUR\_NEWSAPI\_KEY"; // Ganti dengan API Key NewsAPI-mu

const url = `https://newsapi.org/v2/everything?q=${searchQuery}&apiKey=${apiKey}`;

try {

const response = await axios.get(url);

this.setState({

newsData: response.data.articles, // Ambil artikel dari response

loading: false,

});

} catch (error) {

this.setState({ error: error.message, loading: false });

}

};

// Method untuk menangani perubahan input pencarian

handleSearchChange = (e) => {

this.setState({ searchQuery: e.target.value });

};

// Method untuk menangani pengiriman pencarian

handleSearchSubmit = (e) => {

e.preventDefault();

this.fetchData(); // Panggil fetchData saat form disubmit

};

// Memanggil fetchData saat komponen pertama kali dimuat

componentDidMount() {

this.fetchData(); // Memuat berita secara otomatis pada saat pertama kali

}

render() {

const { newsData, loading, error, searchQuery } = this.state;

return (

<Container fluid>

<Row className="bg-dark text-white">

<Col>

<h2>Radar Hot News</h2>

</Col>

</Row>

{/\* Form Pencarian \*/}

<Form onSubmit={this.handleSearchSubmit}>

<InputGroup className="mt-4">

<InputGroup.Text id="basic-addon1">SRC</InputGroup.Text>

<Form.Control

placeholder="Search"

aria-label="Search"

aria-describedby="basic-addon1"

value={searchQuery}

onChange={this.handleSearchChange}

/>

<Button type="submit" variant="primary">Search</Button>

</InputGroup>

</Form>

{/\* Menampilkan loading spinner jika data sedang diambil \*/}

{loading && <div className="mt-4">Loading...</div>}

{/\* Menampilkan error jika terjadi masalah \*/}

{error && <div className="mt-4 text-danger">Error: {error}</div>}

{/\* Menampilkan data berita jika sudah ada \*/}

<Row xs={1} md={4} className="g-4 mt-5">

{newsData.length === 0 && !loading && !error && (

<div>No news found.</div>

)}

{newsData.map((news, idx) => (

<Col key={idx}>

<Card>

<Card.Img variant="top" src={news.urlToImage || "https://via.placeholder.com/100"} />

<Card.Body>

<Card.Title>{news.title}</Card.Title>

<Card.Text>

{news.description || "This is a longer card with supporting text below as a natural lead-in to additional content."}

</Card.Text>

<Button variant="primary" href={news.url} target="\_blank">Read More</Button>

</Card.Body>

</Card>

</Col>

))}

</Row>

</Container>

);

}

}

**Penjelasan Perubahan:**

1. **API Key dan URL**:
   * Ganti YOUR\_NEWSAPI\_KEY dengan API Key yang kamu dapatkan setelah mendaftar di [NewsAPI](https://newsapi.org/).
   * URL API yang digunakan adalah https://newsapi.org/v2/everything?q=${searchQuery}&apiKey=${apiKey}. Di sini, q adalah parameter pencarian yang bisa diubah berdasarkan query pencarian pengguna.
2. **Menyusun Data**:
   * **response.data.articles** adalah tempat dimana artikel-artikel berita disimpan. Kemudian, kita simpan di newsData untuk ditampilkan di UI.
3. **Card dan Gambar**:
   * Setiap artikel berita menampilkan judul (news.title), deskripsi singkat (news.description), dan gambar (news.urlToImage).
   * Jika tidak ada gambar yang tersedia, maka menggunakan gambar placeholder: https://via.placeholder.com/100.
4. **Tombol "Read More"**:
   * Setiap card juga menyediakan tombol "Read More" yang mengarah ke URL lengkap artikel menggunakan news.url.
5. **Pencarian**:
   * Fitur pencarian bekerja dengan mengubah query di state searchQuery, dan saat form disubmit, fetchData() akan dipanggil untuk mengambil berita sesuai dengan kata kunci pencarian.

**Catatan:**

* **Rate Limits**: Perhatikan bahwa NewsAPI memiliki batasan penggunaan gratis. Kamu hanya dapat melakukan sejumlah permintaan dalam satu hari dengan API gratis. Untuk lebih banyak permintaan, kamu perlu berlangganan paket berbayar.
* **Query Pencarian**: Kamu bisa mengubah query pencarian untuk mencari berita berdasarkan topik tertentu, seperti "technology", "sports", atau apa pun yang relevan dengan data yang ingin kamu ambil.

Dengan langkah-langkah di atas, kamu sekarang bisa menggunakan NewsAPI untuk menampilkan berita langsung di aplikasi React kamu.