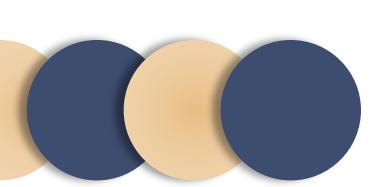




Pemrograman Sisi Cleint Javascript Lanjut

Muhammad Syaifur Rohman, S.Kom, M.CS



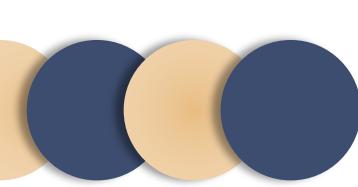


Definisi



FYI

jQuery adalah sebuah perpustakaan JavaScript yang cepat, kecil, dan kaya fitur. Dirilis pada Januari 2006 oleh John Resig, tujuan utama jQuery adalah untuk mempermudah penanganan DOM, event, animasi, dan Ajax dengan sintaks yang mudah dan ringkas. jQuery berhasil mengurangi kompleksitas penulisan kode JavaScript, khususnya untuk browser lintas platform.



Pemilihan id

Jquery

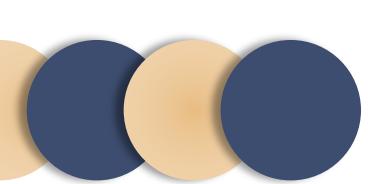


```
$( "#example" );
```

Javascript



```
document.getElementById( 'example' );
```



Manipulasi DOM

Jquery

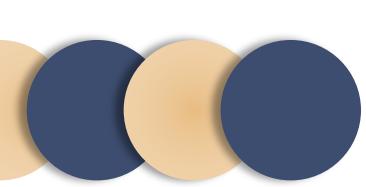


```
$( "#example" ).text( "Tekst baru!" );
```

Javascript



```
document.getElementById( 'example' ).textContent = "Tekst baru!";
```



Mengubah atribut

Jquery

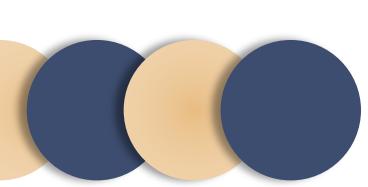


```
$( "#example" ).attr( "href" , "https://www.example.com" );
```

Javascript



```
document.getElementById( 'example' ).setAttribute( "href" , "https://www.example.com" );
```



Event Handling

Jquery



```
$( "#btn" ).click(function() {
    alert( "Tombol diklik!" );
});
```

Javascript



```
document.getElementById( 'btn' ).addEventListener( 'click', function() {
    alert( "Tombol diklik!" );
});
```



Efek dan Animasi

Jquery



```
$( "#example" ).fadeIn( );
```



Javascript

```
let el = document.getElementById('example');
el.style.display = 'block';
el.style.opacity = 0;
let last = +new Date();
let tick = function() {
    el.style.opacity = +el.style.opacity + (new Date() - last) / 400;
    last = +new Date();

    if (+el.style.opacity < 1) {
        (window.requestAnimationFrame && requestAnimationFrame(tick)) || setTimeout(tick, 16);
    }
};
tick();
```



Penggunaan AJAX

Jquery

```
$.ajax({  
  url: "test.txt",  
  success: function(result){  
    $("#example").html(result);  
  }  
});
```

Javascript

```
fetch('test.txt')  
  .then(response => response.text())  
  .then(text => document.getElementById('example').innerHTML = text);
```

Kesimpulan

Sum

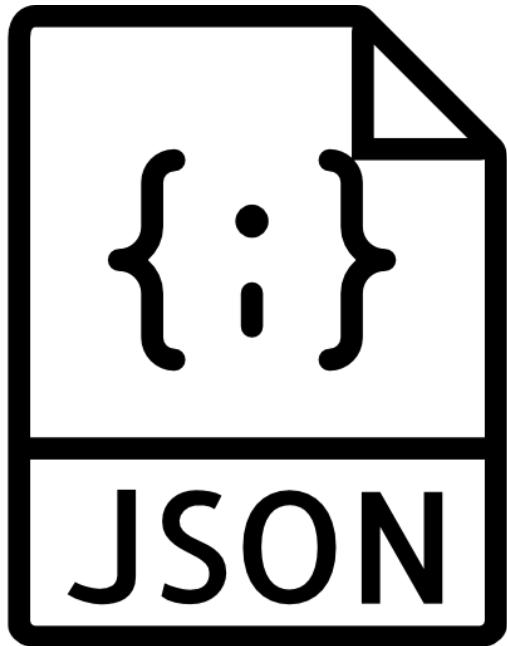
Dalam banyak kasus, jQuery menyederhanakan sintaks dan **mengurangi jumlah kode** yang perlu ditulis untuk mencapai hasil yang sama dibandingkan dengan JavaScript murni. Namun, dengan adanya ES6 dan peningkatan pada API JavaScript murni seperti fetch untuk AJAX dan querySelector untuk seleksi elemen, JavaScript murni menjadi lebih mudah digunakan dan seringkali tidak memerlukan pustaka tambahan seperti jQuery, terutama untuk proyek-proyek baru yang tidak memerlukan dukungan untuk browser lama. Penggunaan JavaScript murni juga mengurangi ketergantungan terhadap pustaka eksternal dan potensi overhead saat memuat halaman.

[Create Form](#) [Delete Form](#)

JSON



Definisi



FYI

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diparsing dan dihasilkan oleh mesin. JSON banyak digunakan dalam pengembangan web untuk pertukaran data antara client dan server.

Struktur

- **Objek:** Kumpulan pasangan nama/nilai. Ditulis dalam kurung kurawal {}. Mirip dengan objek JavaScript.
- **Array:** Kumpulan nilai yang berurutan. Ditulis dalam kurung siku []. **Nilai:** Bisa berupa string, angka, objek, array, true, false, atau null.

- Objek: {"nama": "John Doe", "umur": 30}
- Array: ["Apple", "Banana", "Cherry"]

```
{  
  "employees": [  
    {"nama": "John Doe", "umur": 30},  
    {"nama": "Anna Smith", "umur": 25}  
  ]  
}
```



Penggunaan

- **Pertukaran Data:** JSON sering digunakan untuk mengirim data dari server ke halaman web dan sebaliknya.
- **Konfigurasi:** JSON digunakan dalam berbagai file konfigurasi dan pengaturan karena mudah dibaca.
- **Penyimpanan Data:** Format yang efisien untuk menyimpan data kompleks dalam database NoSQL seperti MongoDB.

AJAX





Definisi



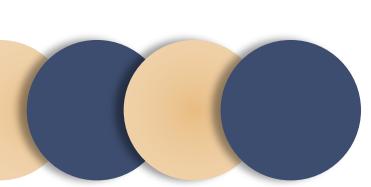
FYI

AJAX (Asynchronous JavaScript And XML) adalah teknik yang digunakan dalam pengembangan web untuk membuat aplikasi web interaktif. Dengan AJAX, web aplikasi dapat mengirim dan menerima data dari server secara asinkron tanpa perlu memuat ulang seluruh halaman. Meskipun nama AJAX mengandung XML, JSON sekarang lebih sering digunakan karena formatnya yang lebih ringkas dan mudah diolah. Ini memungkinkan pembuatan pengalaman pengguna yang lebih cepat dan lebih responsif.



Struktur

- **HTTP Requests:** AJAX mengirim permintaan HTTP ke server dan menerima respons. Metode HTTP yang umum termasuk GET (mengambil data), POST (mengirim data baru), PUT (memperbarui data), dan DELETE (menghapus data).
- **Callback Functions:** Fungsi yang dipanggil ketika respons dari permintaan AJAX diterima, memungkinkan pengembang untuk memanipulasi respons tersebut di sisi klien.
- **JSON:** Format pertukaran data yang ringkas, yang digunakan untuk mengirim dan menerima data melalui AJAX.
- **Asynchronous Processing:** AJAX memungkinkan pengolahan data secara asinkron, artinya halaman web tidak perlu menunggu respons dari server untuk melakukan tindakan lain.



Penggunaan - GET

GET



```
$.ajax({
  url: 'https://jsonplaceholder.typicode.com/posts/1',
  type: 'GET',
  dataType: 'json',
  success: function(data) {
    console.log(data);
  },
  error: function(error) {
    console.error("Terjadi kesalahan: ", error);
  }
});
```

Penggunaan - POST

POST

```
$.ajax({
  url: 'https://jsonplaceholder.typicode.com/posts',
  type: 'POST',
  dataType: 'json',
  data: JSON.stringify({
    title: 'foo',
    body: 'bar',
    userId: 1,
  }),
  contentType: 'application/json; charset=utf-8',
  success: function(data) {
    console.log(data);
  },
  error: function(error) {
    console.error("Terjadi kesalahan: ", error);
  }
});
```

Full of joy

HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>AJAX dengan jQuery</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  </head>
  <body>

    <h2>Daftar Post</h2>
    <ul id="post-list"></ul>

    <h2>Buat Post Baru</h2>
    <form id="new-post-form">
      <input type="text" id="title" placeholder="Judul" required><br>
      <textarea id="body" placeholder="Isi Post" required></textarea><br>
      <button type="submit">Kirim Post</button>
    </form>

    <h2>Detail Post</h2>
    <div id="post-detail"></div>

    <script src="script.js"></script>

  </body>
</html>
```

```
$(<document>).ready(function() {
  // GET request untuk mengambil daftar post
  function fetchPosts() {
    $.ajax({
      url: 'https://jsonplaceholder.typicode.com/posts',
      type: 'GET',
      success: function(posts) {
        $('#post-list').empty(); // Bersihkan list sebelumnya
        posts.slice(0, 5).forEach(function(post) { // Batasi hanya 5 post
          $('#post-list').append('<li>' + post.title + '<button'
            onclick="fetchPostDetail(" + post.id + ")">Lihat Detail</button></li>');
        });
      }
    });
  }

  // Fungsi untuk melihat detail post
  window.fetchPostDetail = function(postId) {
    $.ajax({
      url: `https://jsonplaceholder.typicode.com/posts/${postId}`,
      type: 'GET',
      success: function(post) {
        $('#post-detail').html(`<h3>${post.title}</h3><p>${post.body}</p>`);
      }
    });
  }

  // POST request untuk membuat post baru
  $('#new-post-form').submit(function(e) {
    e.preventDefault();
    $.ajax({
      url: 'https://jsonplaceholder.typicode.com/posts',
      type: 'POST',
      data: JSON.stringify({
        title: $('#title').val(),
        body: $('#body').val(),
        userId: 1,
      }),
      contentType: "application/json; charset=utf-8",
      success: function(post) {
        alert("Post berhasil dikirim!");
        $('#title').val('');
        $('#body').val('');
        fetchPosts(); // Muat ulang daftar post
        // $('#title').val('');
        // $('#body').val('');
        // $('#post-list').append('<li>' + post.title + '<button'
        //   onclick="fetchPostDetail(" + post.id + ")">Lihat Detail</button></li>');
      }
    });
  });

  // Muat daftar post saat halaman dimuat
  fetchPosts();
});
```

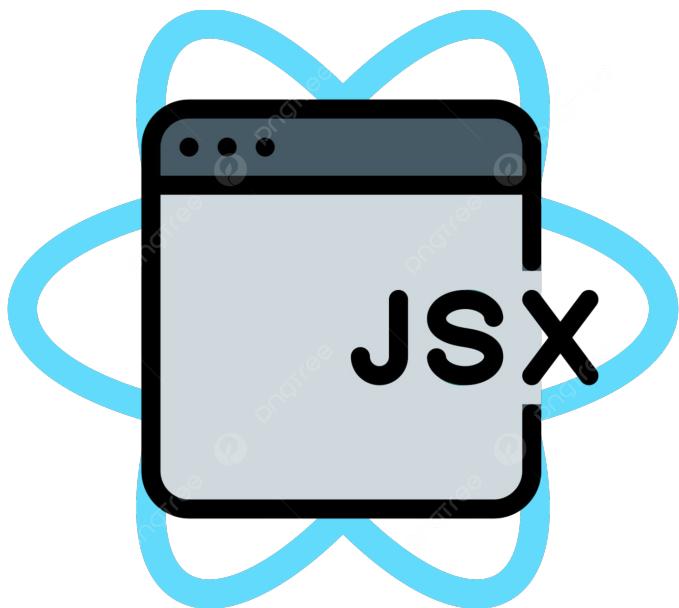
TRY...



JSX



Definisi



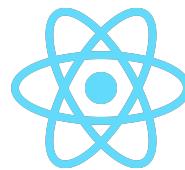
FYI

JSX, singkatan dari JavaScript Syntax eXpression, adalah ekstensi sintaks untuk JavaScript yang direkomendasikan oleh beberapa library dan framework, terutama React. Walaupun sering dikaitkan erat dengan React, JSX sebenarnya independen dan dapat digunakan dengan library lain atau bahkan tanpa library. **JSX memungkinkan pengembang menulis elemen UI dalam kode yang mirip dengan HTML di dalam JavaScript.** Ini membuat kode lebih mudah dibaca dan ditulis, terutama untuk struktur komponen UI yang kompleks.

Definisi

????

JSX memungkinkan Anda menulis elemen HTML dalam JavaScript dan menempatkan HTML tersebut dalam kode JS. Ini mirip dengan template language, tapi dengan kekuatan penuh dari JavaScript. JSX menghasilkan elemen React, yang memungkinkan Anda menggabungkan UI dan logika aplikasi dengan cara yang lebih dekat dan intuitif daripada mengandalkan fungsi-fungsi JavaScript murni.



React menggunakan JSX untuk membuat tampilan komponen yang deklaratif. Ini membantu developer memvisualisasikan bagaimana UI akan terlihat berdasarkan state aplikasi. Dengan JSX, kode yang mendeskripsikan UI juga merupakan ekspresi JavaScript yang valid, membuat proses pengembangan menjadi lebih efisien dan lebih mudah untuk dipahami.



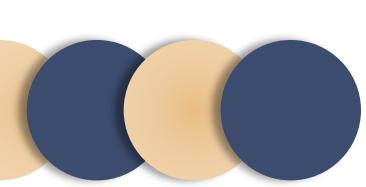
Sintax Dasar



```
const name = 'Dunia';
const element = <h1>Halo, {name}</h1>;
```

Sintax

- JSX menggabungkan markup HTML dengan logika JavaScript. Dengan JSX, Anda dapat menyisipkan ekspresi JavaScript di dalam kurung kurawal {}.



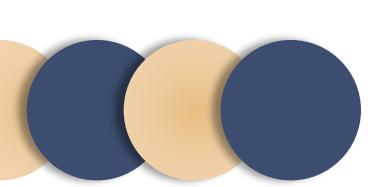
Sintax Dasar



```
const myElement = <div className="myClass">Hello World</div>;
```

Elemen HTML

- Elemen JSX menggambarkan apa yang ingin Anda tampilkan di layar. Elemen JSX bisa berupa tag HTML biasa atau komponen yang didefinisikan pengguna.



Sintax Dasar

Function



```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

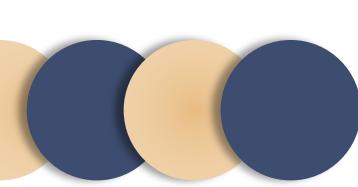
Event Handling



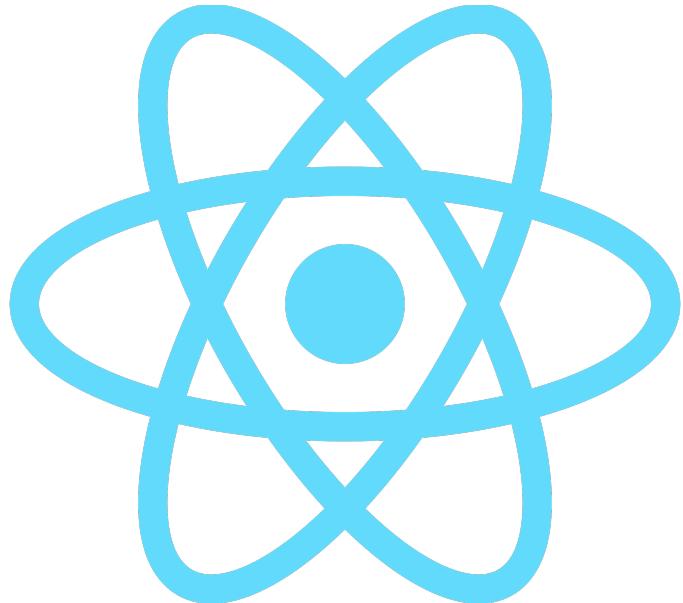
```
<button onClick={this.handleClick}>  
  Click me  
</button>
```

Reactjs





Definisi



FYI

React (juga dikenal sebagai React.js atau ReactJS) adalah pustaka JavaScript yang populer untuk membangun antarmuka pengguna, atau UI. Dikembangkan oleh Facebook dan dirilis pada tahun 2013, React telah menjadi salah satu pilihan utama bagi pengembang web di seluruh dunia. Materi ini akan memperkenalkan Anda kepada konsep dasar React dan bagaimana menggunakan untuk membangun aplikasi web yang responsif dan dinamis.



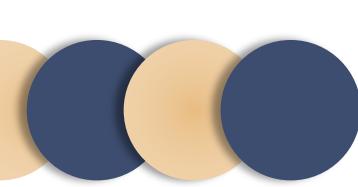
Definisi

????

React adalah pustaka JavaScript untuk membangun antarmuka pengguna yang efisien dan interaktif. Dengan React, Anda dapat membuat komponen UI yang besar terdiri dari komponen-komponen kecil, terisolasi, dan dapat digunakan kembali. React berfokus pada 'view layer' saja, membuatnya dapat digabungkan dengan pustaka lain seperti Redux untuk manajemen state atau React Router untuk navigasi.

Why ?

- **Komponen yang Dapat Digunakan Kembali:** Dengan React, UI dibagi menjadi komponen terisolasi yang dapat digunakan kembali, membuat kode Anda lebih terorganisir dan mudah untuk di-maintain.
- **Deklaratif:** React membuat pembuatan antarmuka pengguna yang interaktif menjadi mudah. Deklarasikan apa yang Anda inginkan tampil, dan React akan secara efisien memperbarui dan merender komponen yang tepat saat data Anda berubah.
- **Performa Tinggi:** React menggunakan virtual DOM untuk meminimalkan jumlah manipulasi DOM, yang merupakan operasi mahal dalam hal performa. Ini meningkatkan kecepatan dan efisiensi aplikasi.



Requirement

Must Install !

- Git bash (optional) or terminal
- Node js
- Npm
- Npx

Command

- Install react app
 - npx create-react-app nama-aplikasi
- Buka projek react
 - cd nama-aplikasi
- Jalankan
 - npm build
 - npm run



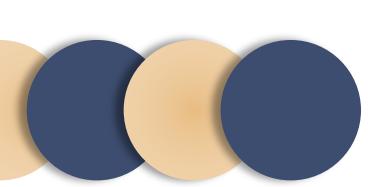
Konsep Dasar

Komponen

Komponen adalah blok bangunan dasar aplikasi React. Setiap komponen adalah sebuah potongan kode JavaScript yang mengembalikan elemen React, yang menggambarkan bagian dari antarmuka pengguna. Komponen bisa berupa fungsi atau kelas.

JSX

JSX adalah sintaks ekstensi untuk JavaScript yang digunakan dalam React untuk menggambarkan UI. Meskipun terlihat mirip dengan HTML, JSX adalah ekspresi JavaScript yang memungkinkan Anda menulis struktur komponen dengan cara yang lebih mudah dibaca dan dipahami.



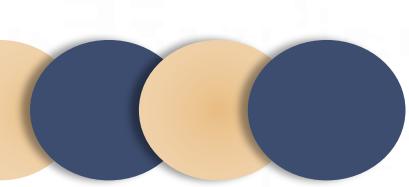
Sintax Dasar



```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

Class

- **Props** adalah singkatan dari Properties. Mereka adalah argumen yang Anda pasok ke komponen React dan merupakan cara untuk memberikan data ke komponen.
- **State** adalah data pribadi komponen yang dapat berubah seiring waktu. Komponen kelas menggunakan **this.state** untuk menyimpan state dan **this.setState()** untuk memperbarui state.



Exercise

```
import React from 'react';

function SimplePage() {
  const title = "Halaman Web Sederhana";
  const tasks = [ 'Belajar JavaScript', 'Belajar React', 'Membuat Aplikasi Web dengan React'];

  return (
    <div>
      <h1>{title}</h1>
      <h2>Daftar Tugas:</h2>
      <ul>
        {tasks.map((task, index) =>
          <li key={index}>{task}</li>
        )}
      </ul>
    </div>
  );
}

export default SimplePage;
```

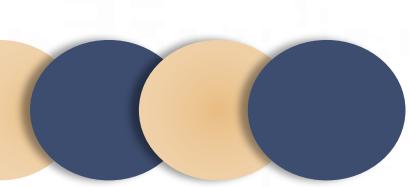
Exercise

```
import React from 'react';
import './App.css';
import SimplePage from './SimplePage';

function App() {
  return (
    <div className="App">
      <SimplePage />
    </div>
  );
}

export default App;
```

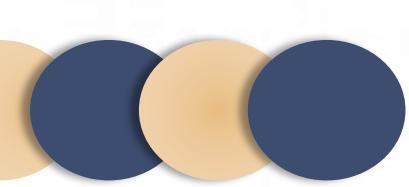
Gunakan komponen SimplePage di dalam file App.js



Exercise



Lets go deep on next class !



Referensi

1. Eran Krinsbruner, **A Frontend Web Developer's Guide to Testing**, 2022
2. Adam Boduch , Roy Derks , Mikhail Sakhniuk, **React and React Native - Fourth Edition**, 2022
3. Maya Shavin , Raymond Camden, **Frontend Development Projects with Vue.js 3 - Second Edition**, 2023
4. Waweru Mwaura, **End-to-End Web Testing with Cypress**, 2021
5. <https://roadmap.sh/frontend>
6. <https://jsonplaceholder.typicode.com/posts>



Thanks

Muhammad Syaifur Rohman, S.Kom, M.CS

