

MAY-MMXXIII



Data Rendering

INTRODUCTION TO INTERNET TECHNOLOGY

Kelompok-X

KELOMPOK - 5

Membahas Data Rendering



Dhika

77703



M Raehan

78572



Desquita

78745



Khaerul

78279

Bahasan

first

1

Definisi

second

2

JavaScript ES6
Map/Filter/Reduce

third

3

JavaScript ES6
Arrow Function

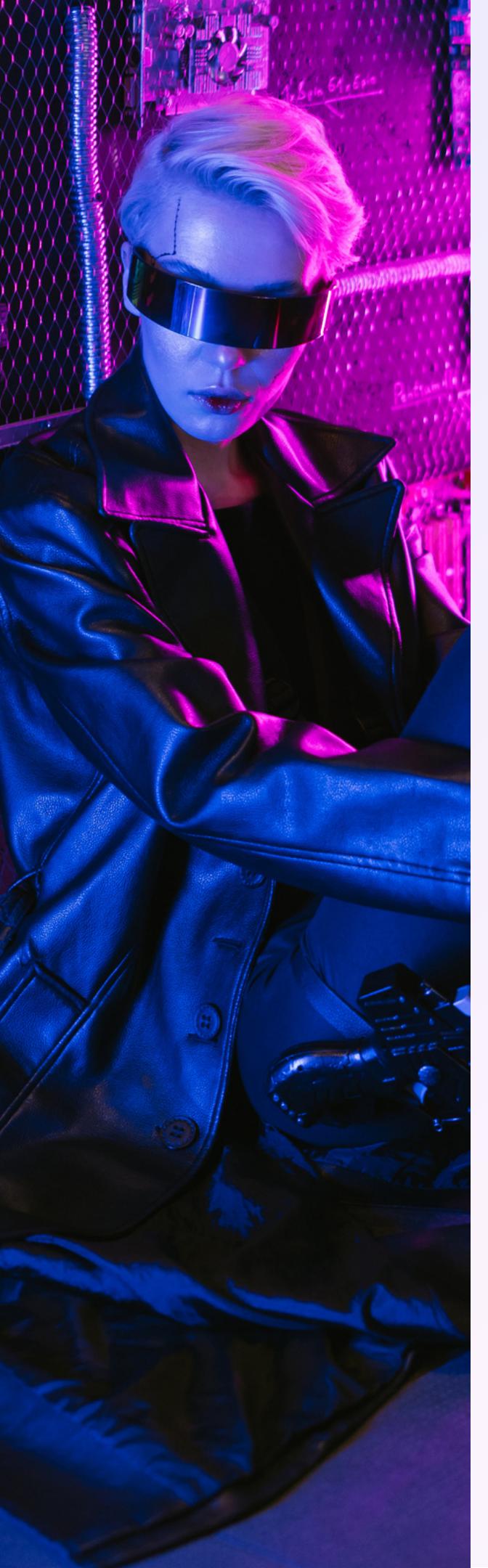
fourth

4

React Conditional
Rendering



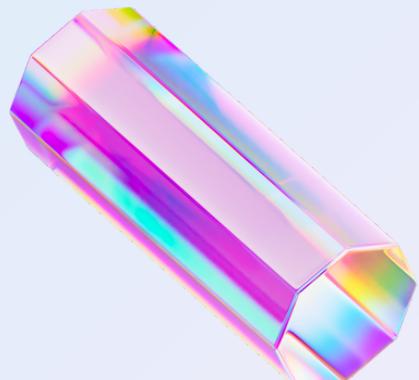
Definisi Data rendering





Definisi Data rendering

Data rendering adalah proses mengubah data mentah menjadi visualisasi yang lebih mudah dipahami. Ini dilakukan dengan menggunakan algoritma, teknik, dan software khusus untuk menghasilkan tampilan visual yang menarik dan akurat dari data mentah. Proses ini membantu mempermudah pengambilan keputusan atau pemahaman terhadap suatu fenomena atau data.



A photograph of a woman with her head down, looking at her arm. She has multiple braids in her hair and a large, colorful floral tattoo on her upper arm. She is wearing a black tank top and dark pants. The background features vertical red and white neon lights.

JavaScriptES6

Map/Filter/Reduce



Map



Metode map digunakan untuk mengambil setiap elemen dalam array, memproses nya, dan menghasilkan array baru dengan hasilnya. Metode ini menerima satu argumen, yaitu sebuah fungsi yang diterapkan pada setiap elemen array. Fungsi ini mengembalikan nilai baru, yang kemudian disimpan dalam array baru.

1

Map



```
1 const numbers = [1, 2, 3, 4];
2 const doubled = numbers.map(num => num * 2);
3 console.log(doubled); // [2, 4, 6, 8]
```

Filter



Metode filter digunakan untuk memfilter elemen dalam array berdasarkan kondisi tertentu dan menghasilkan array baru yang hanya berisi elemen yang lolos filter. Metode ini menerima satu argumen, yaitu sebuah fungsi yang mengembalikan nilai boolean untuk setiap elemen array.

2 Filter



```
1 const numbers = [1, 2, 3, 4];
2 const evenNumbers = numbers.filter(num => num % 2 === 0);
3 console.log(evenNumbers); // [2, 4]
```

3

Reduce



Metode Reduce digunakan untuk menggabungkan semua elemen dalam Array menjadi satu nilai tunggal, berdasarkan sebuah fungsi yang ditentukan. Secara sederhana, metode Reduce dapat digunakan untuk menjumlahkan semua elemen dalam Array atau melakukan operasi matematika lainnya.

3 Reduce



```
1 const numbers = [1, 2, 3, 4];
2 const sum = numbers.reduce((acc, num) => acc + num, 0);
3 console.log(sum); // 10
```



JavaScript ES6 Arrow Function





JavaScript ES6 Arrow Function

`() => {}`

Arrow Function in JavaScript

Merupakan fitur baru dalam ES6 yang memungkinkan penulisan fungsi dengan sintaks yang lebih ringkas dan mudah dibaca.

Secara garis besar arrow function adalah fungsi yang dideklarasikan dengan **tanda arrow(panah) "=>"**

Dalam arrow function, kita tidak perlu menuliskan kata kunci "function" dan tanda kurung kurawal pada body function jika hanya terdiri dari satu statement.



"Contoh Penggunaan"

JavaScript ES6 Arrow Function



```
1 var hello;  
2  
3 hello = function() {  
4     return "Hello World!";  
5 }  
6  
7 document.getElementById("demo").innerHTML = hello();
```

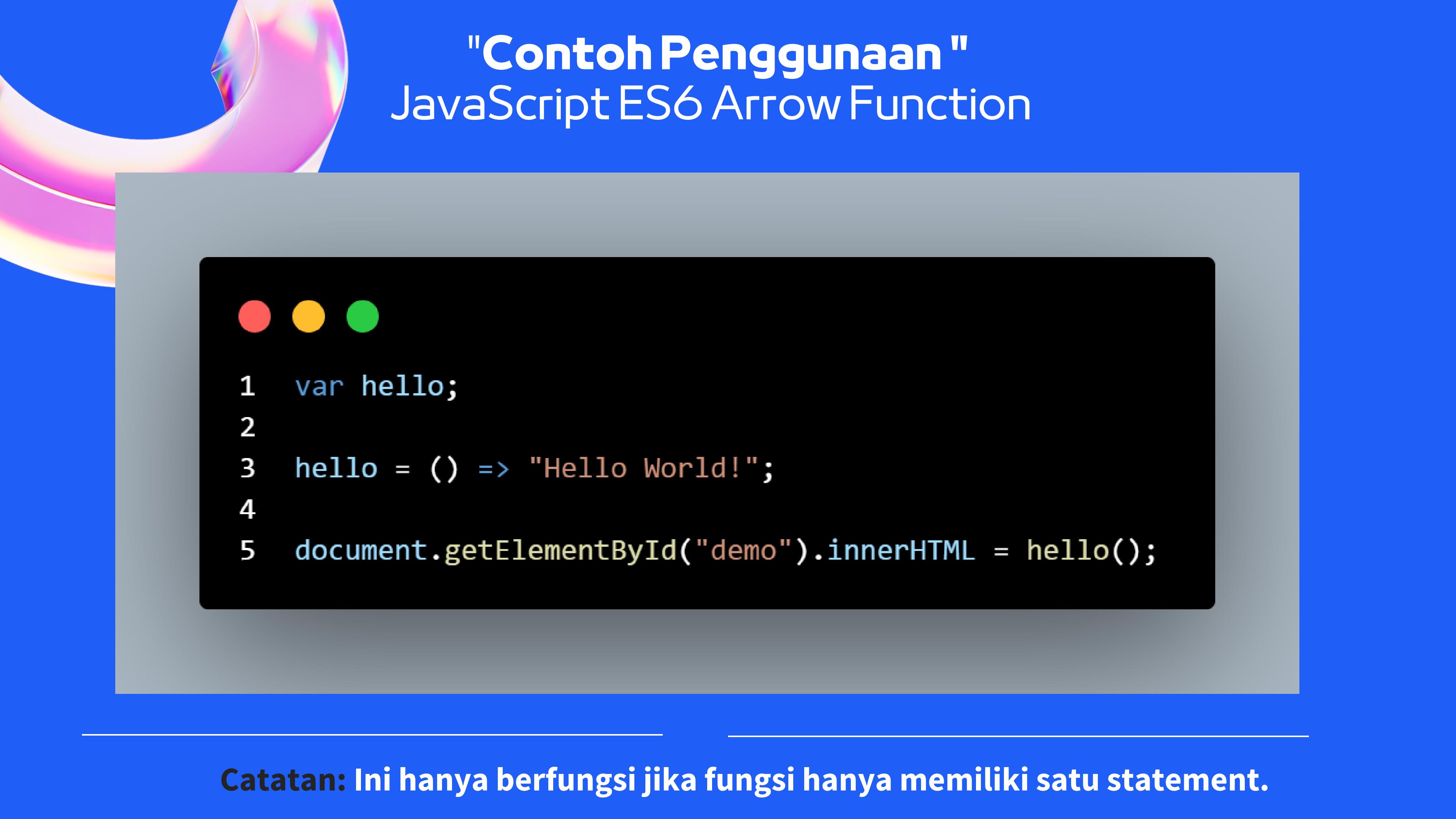
Tanpa Arrow Function

Perbedaan dari 2 program diatas adalah perubahan pada syntax yang awalnya menggunakan source code “function” diubah dengan menggunakan syntax Arrow Function menjadi **tanda panah “=>”**.



```
1 var hello;  
2  
3 hello = () => {  
4     return "Hello World!";  
5 }  
6  
7 document.getElementById("demo").innerHTML = hello();
```

Dengan Arrow Function



"Contoh Penggunaan" JavaScript ES6 Arrow Function



```
1 var hello;  
2  
3 hello = () => "Hello World!";  
4  
5 document.getElementById("demo").innerHTML = hello();
```

Catatan: Ini hanya berfungsi jika fungsi hanya memiliki satu statement.

"Contoh Penggunaan" JavaScript ES6 Arrow Function



```
1 var hello;  
2  
3 hello = (val) => "Hello " + val;  
4  
5 document.getElementById("demo").innerHTML = hello("Universe!");
```

Jika memiliki parameter, maka menaruh atau meneruskannya di dalam tanda kurung,
`hello("Universe!");`

React Conditional Rendering



React Conditional Rendering

Merupakan cara di mana kita dapat menampilkan komponen atau elemen tertentu tergantung pada kondisi tertentu di dalam aplikasi React. Ini memungkinkan kita untuk membuat aplikasi yang lebih interaktif dan responsif terhadap input pengguna.



Contoh penggunaan

penggunaan conditional rendering seperti ini, dapat membuat aplikasi yang lebih dinamis dan memberikan pengguna lebih banyak kontrol atas data yang ditampilkan.

```
1 import React, { useState } from "react";
2
3 function ItemList(props) {
4   const items = props.items;
5   const [showAll, setShowAll] = useState(false);
6
7   const filteredItems = showAll ? items : items.filter(item => item.category === "Food");
8
9   return (
10     <div>
11       <h1>Items</h1>
12       <ul>
13         {filteredItems.map(item => (
14           <li key={item.name}>
15             {item.name} - {item.category}
16           </li>
17         )))
18       </ul>
19       <button onClick={() => setShowAll(!showAll)}>
20         {showAll ? "Show only Food items" : "Show All"}
21       </button>
22     </div>
23   );
24 }
25
26 export default ItemList;
```

selesai.