

MODUL 1
MODERN JAVASCRIPT

PEMROGRAMAN BERBASIS FRAMEWORK



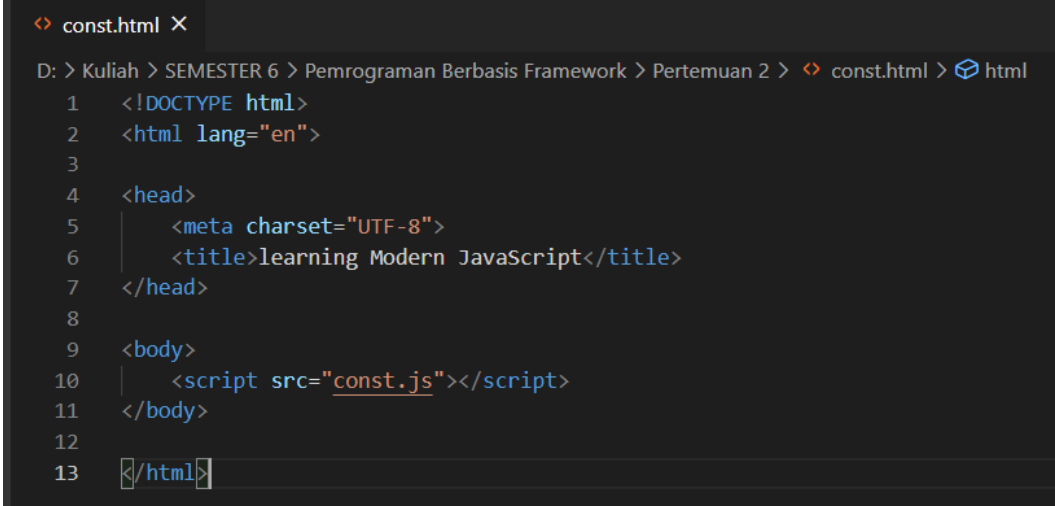
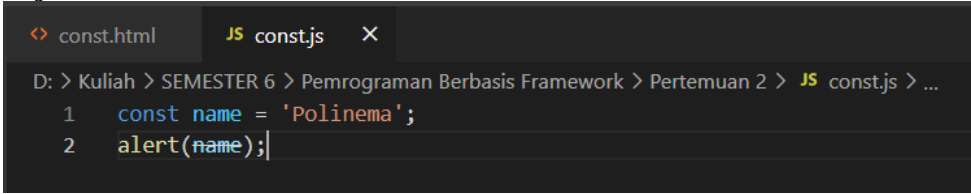
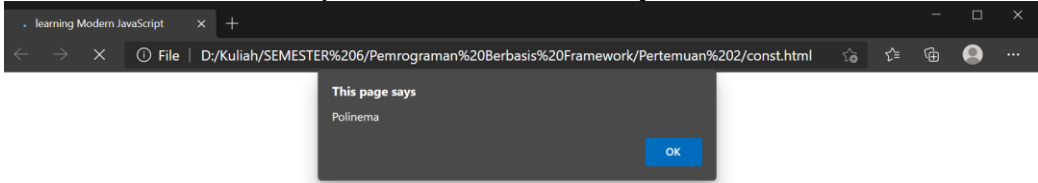
Disusun oleh:
Livia Yurike Khuril Maula
D4 TI – 3F / 15
NIM : 1841720025

POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI D4 TEKNIK INFORMATIKA

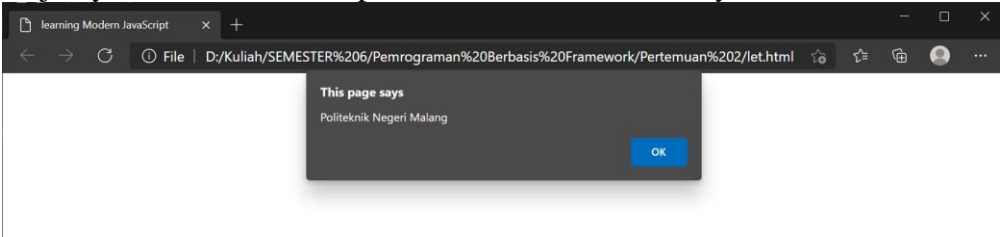
FEBRUARI 2021

PRAKTIKUM

a. Membuat variabel menggunakan const

1	<p>Membuat halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>const.html</p>  <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>learning Modern JavaScript</title> </head> <body> <script src="const.js"></script> </body> </html></pre> <p>const.js</p>  <pre>const name = 'Polinema'; alert(name);</pre> <ul style="list-style-type: none">• Keyword const digunakan untuk variabel yang tidak bisa di-reassign. Variabel const bersifat konstan, artinya tidak mungkin berubah. Lebih tepatnya variabel const tidak bisa di-reassign atau diberikan nilai yang berbeda setelah dideklarasikan.
2	<p>Lalu buka file const.html pada browser, dan hasilnya :</p> 

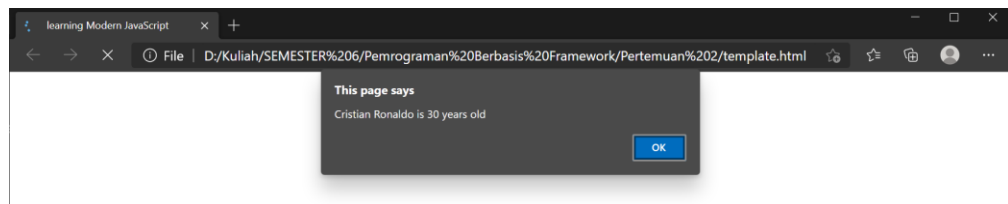
b. Membuat variabel menggunakan let

1	<p>Membuat halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>let.html</p>  <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>learning Modern JavaScript</title> </head> <body> <script src="let.js"></script> </body> </html></pre> <p>let.js</p>  <pre>if (true) { let name = 'Polinema'; name = 'Politeknik Negeri Malang'; alert(name); }</pre> <ul style="list-style-type: none">• Keyword let digunakan untuk variabel yang bisa di-reassign setelah dideklarasi.
2	<p>Selanjutnya buka file let.html pada browser, , maka hasilnya :</p> 

c. Membuat template Strings

1	<p>Membuat halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>template.html</p> <pre>template.html X D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > template.html > html > body > script 1 <!DOCTYPE html> 2 <html lang="en"> 3 4 <head> 5 <meta charset="UTF-8"> 6 <title>learning Modern JavaScript</title> 7 </head> 8 9 <body> 10 <script src="template.js"></script> 11 </body> 12 13 </html></pre> <p>template.js</p> <pre>template.html X JS template.js X D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > JS template.js > ... 1 let fname = 'Cristian'; 2 let lname = 'Ronaldo'; 3 let age = prompt("Masukkan umur Cristian Ronaldo!"); 4 5 // Cara lama 6 // let result = fname + ' ' + lname + 'is' + age + 'years old'; 7 // alert(result); 8 9 //Memakai template strings 10 let result = `\${fname} \${lname} is \${age} years old`; 11 alert(result);</pre> <ul style="list-style-type: none">• Penulisan template literals menggunakan karakter petik ``. Memasukkan nilai ke dalam string menjadi lebih mudah menggunakan sintaks \${}. Tidak perlu menggunakan operator• Untuk menggabungkan string Tidak perlu menulis baris baru dengan karakter newline (\n) atau carriage return (r). Bisa menggunakan tanda kutip bersarang (dengan kutip satu atau kutip dua) di dalam template literals
2	<p>Selanjutnya buka file template.html pada browser, maka hasilnya :</p> 

Klik Ok, lalu hasilnya seperti dibawah ini



d. Membuat default parameters

- 1 Membuat halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

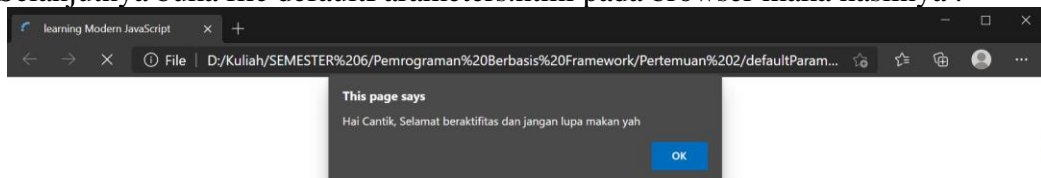
defaultParameters.html

```
defaultParameters.html X
D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > defaultParameters.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <title>learning Modern JavaScript</title>
7  </head>
8
9  <body>
10   <script src="defaultParameters.js"></script>
11 </body>
12
13 </html>
```

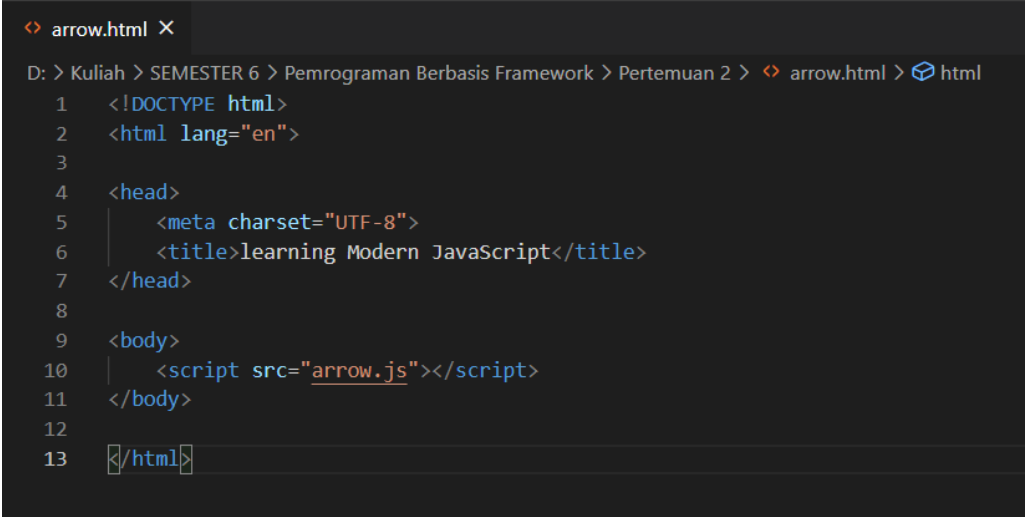
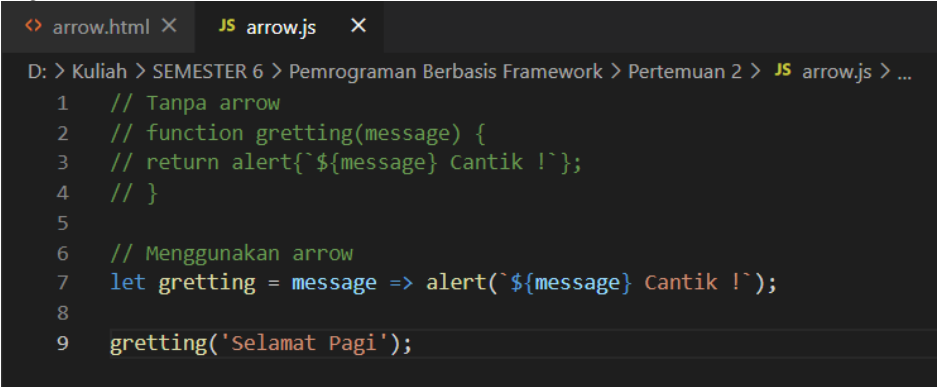
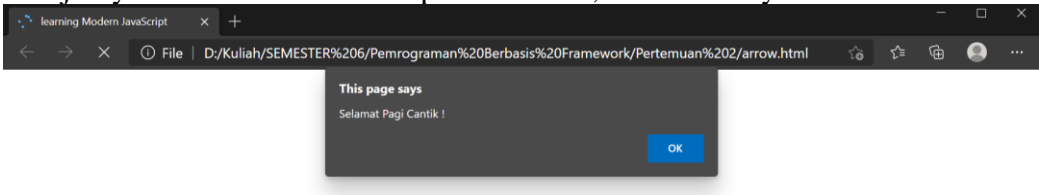
defaultParameters.js

```
defaultParameters.html JS defaultParameters.js X
D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > JS defaultParameters.js > ...
1  function welcome(user = 'Cantik', message = 'Selamat beraktifitas dan jangan lupa makan yah') {
2    alert(`Hai ${user}, ${message}`);
3  }
4  welcome();
```

- 2 Selanjutnya buka file defaultParameters.html pada browser maka hasilnya :



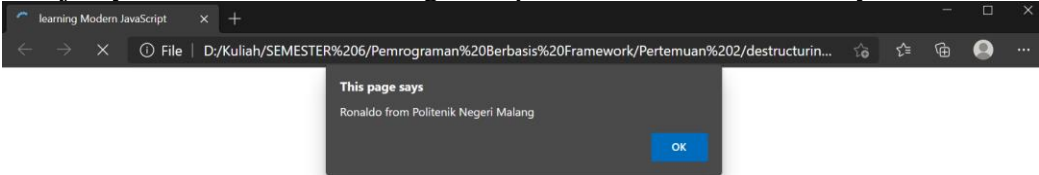
e. Membuat Arrow Function 1

1	<p>Membuat halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>arrow.html</p>  <pre><?xml version="1.0" encoding="UTF-8" ?> <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>learning Modern JavaScript</title> </head> <body> <script src="arrow.js"></script> </body> </html></pre> <p>arrow.js</p>  <pre>// Tanpa arrow function greeting(message) { return alert(`\${message} Cantik !`); } // Menggunakan arrow let greeting = message => alert(`\${message} Cantik !`); greeting('Selamat Pagi');</pre>
2	<p>Selanjutnya buka file arrow.html pada browser, maka hasilnya akan :</p> 

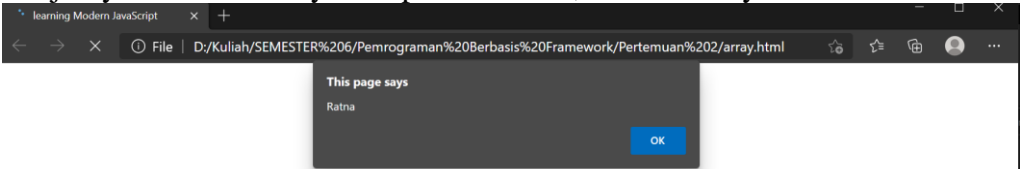
f. Membuat Arrow Function 2

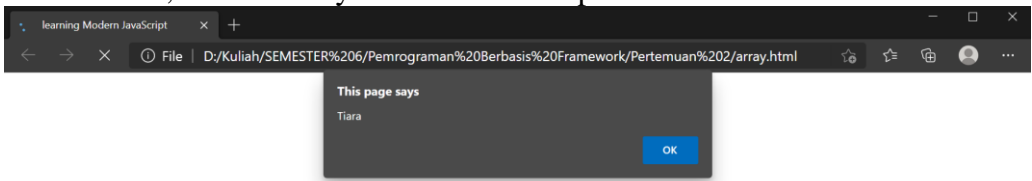
1	<p>Membuat halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>arrowf.html</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="no" ?> <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>learning Modern JavaScript</title> </head> <body> <script src="arrowf.js"></script> </body> </html></pre> <p>arrowf.js</p> <pre>// Tanpa arrow // const func = function(a, b) { // return a + b; // }; // alert(func(3, 3)); //Menggunakan arrow const func = (a, b) => { return a + b; }; alert(func(5, 4));</pre> <ul style="list-style-type: none">• Arrow function memungkinkan kita untuk menulis function dengan sintaks yang lebih singkat. Menggantikan keyword return dan body function dengan tanda panah: =>.• Tanda kurung pada parameter bisa dihapus jika hanya ada 1 parameter.• Kurung kurawal (curly braces) pada body function bisa dihapus.
2	<p>Selanjutnya buka file arrowf.html pada browser, maka hasilnya :</p> 

g. Membuat Destructuring Object

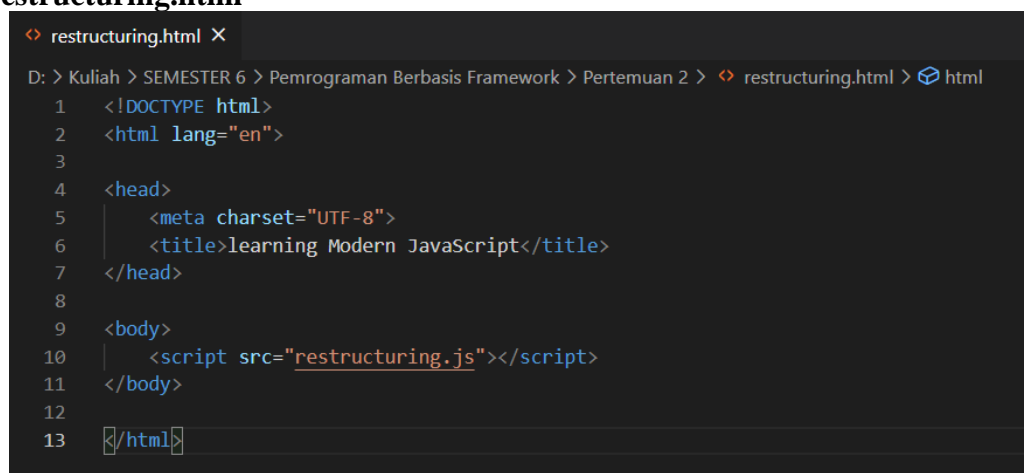
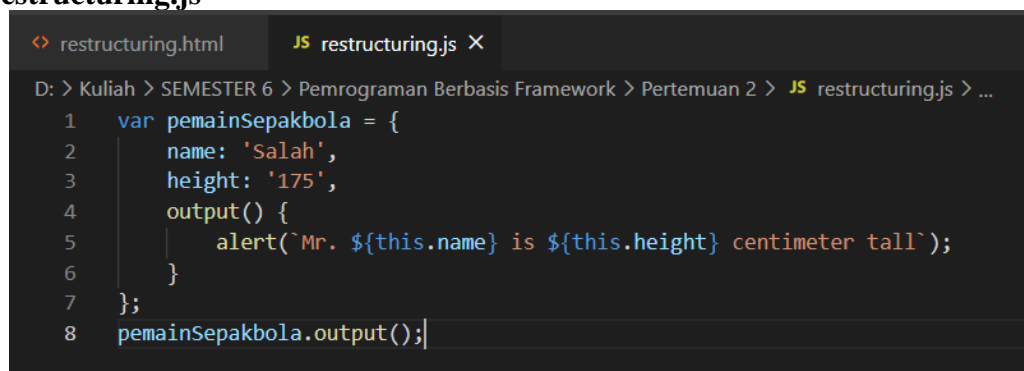
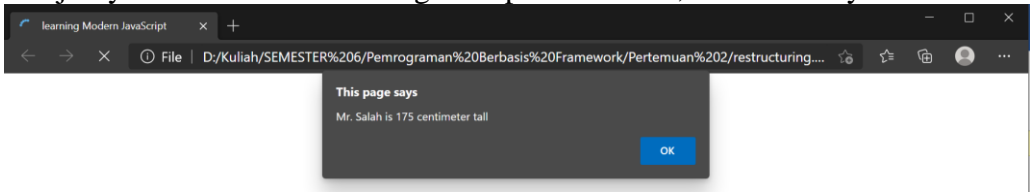
1	<p>Membuat halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>destructuring.html</p> <pre><? destructuring.html X D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > <? destructuring.html > html 1 <!DOCTYPE html> 2 <html lang="en"> 3 4 <head> 5 <meta charset="UTF-8"> 6 <title>learning Modern JavaScript</title> 7 </head> 8 9 <body> 10 <script src="destructuring.js"></script> 11 </body> 12 13 </html></pre> <p>destructuring.js</p> <pre><? destructuring.html JS destructuring.js X D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > JS destructuring.js > ... 1 let polStudent = ({ name, polytechnic }) => { 2 alert(`\${name} from \${polytechnic}`); 3 }; 4 5 polStudent({ 6 name: 'Ronaldo', 7 polytechnic: 'Politenik Negeri Malang' 8 }) 9</pre>
2	<p>Selanjutnya buka file destructuring.html pada browser, maka hasilnya :</p> 

h. Membuat Destructuring an Array

1	<p>Membuat halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>array.html</p> <pre><? array.html X D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > <? array.html > html 1 <!DOCTYPE html> 2 <html lang="en"> 3 4 <head> 5 <meta charset="UTF-8"> 6 <title>learning Modern JavaScript</title> 7 </head> 8 9 <body> 10 <script src="array.js"></script> 11 </body> 12 13 </html></pre> <p>array.js</p> <pre><? array.html JS array.js X D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > JS array.js > ... 1 let [wife] = ['Ratna', 'Bunga', 'Tiara']; 2 alert(wife);</pre> <ul style="list-style-type: none">• Destructuring membuat kita tidak perlu mengakses keseluruhan object ketika ingin digunakan.• Dengan Destructuring, kita bisa mengambil apapun nilai yang dibutuhkan dan disimpan ke variabel.• Destructuring juga bisa dilakukan pada array sama seperti object.
2	<p>Selanjutnya buka file array.html pada browser, maka hasilnya :</p> 
3	<p>Kemudian ganti sedikit code pada array.js seperti dibawah ini</p> <pre><? array.html JS array.js X D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > JS array.js > ... 1 let [, , wife] = ['Ratna', 'Bunga', 'Tiara']; 2 alert(wife);</pre>

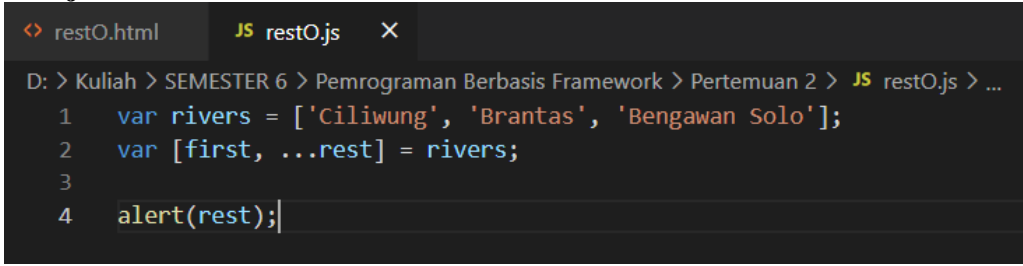
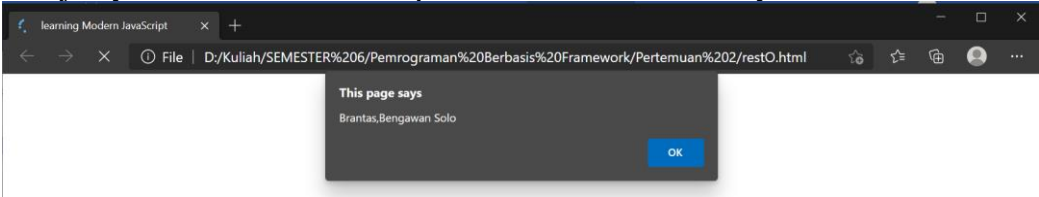
4	<p>Buka kembali, maka hasilnya akan berbeda seperti :</p> 
---	--

i. Membuat Restructuring

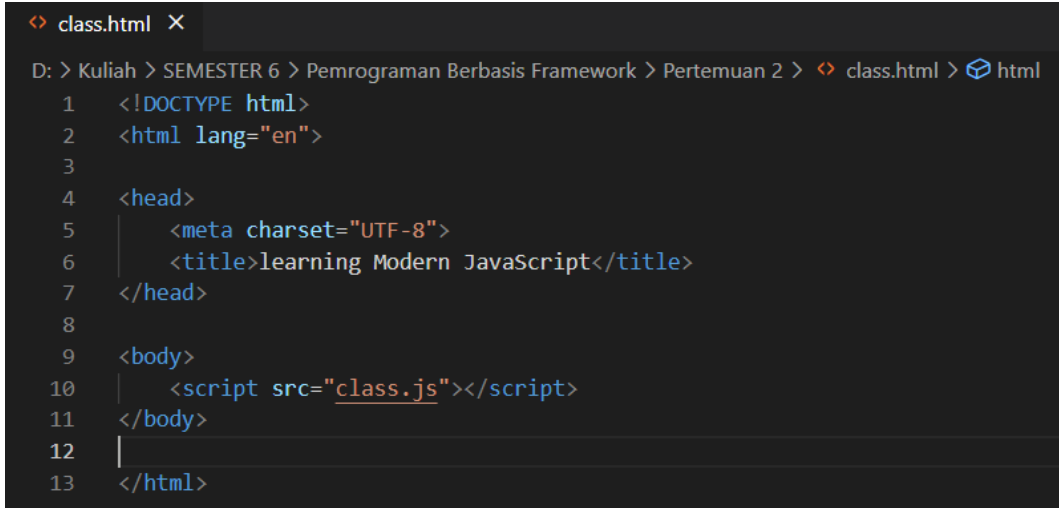
1	<p>Membuat halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>restructuring.html</p>  <pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 4 <head> 5 <meta charset="UTF-8"> 6 <title>learning Modern JavaScript</title> 7 </head> 8 9 <body> 10 <script src="restructuring.js"></script> 11 </body> 12 13 </html> </pre> <p>restructuring.js</p>  <pre> 1 var pemainSepakbola = { 2 name: 'Salah', 3 height: '175', 4 output() { 5 alert(`Mr. \${this.name} is \${this.height} centimeter tall`); 6 } 7 }; 8 pemainSepakbola.output(); </pre>
2	<p>Selanjutnya buka file restructuring.html pada browser, maka hasilnya :</p> 

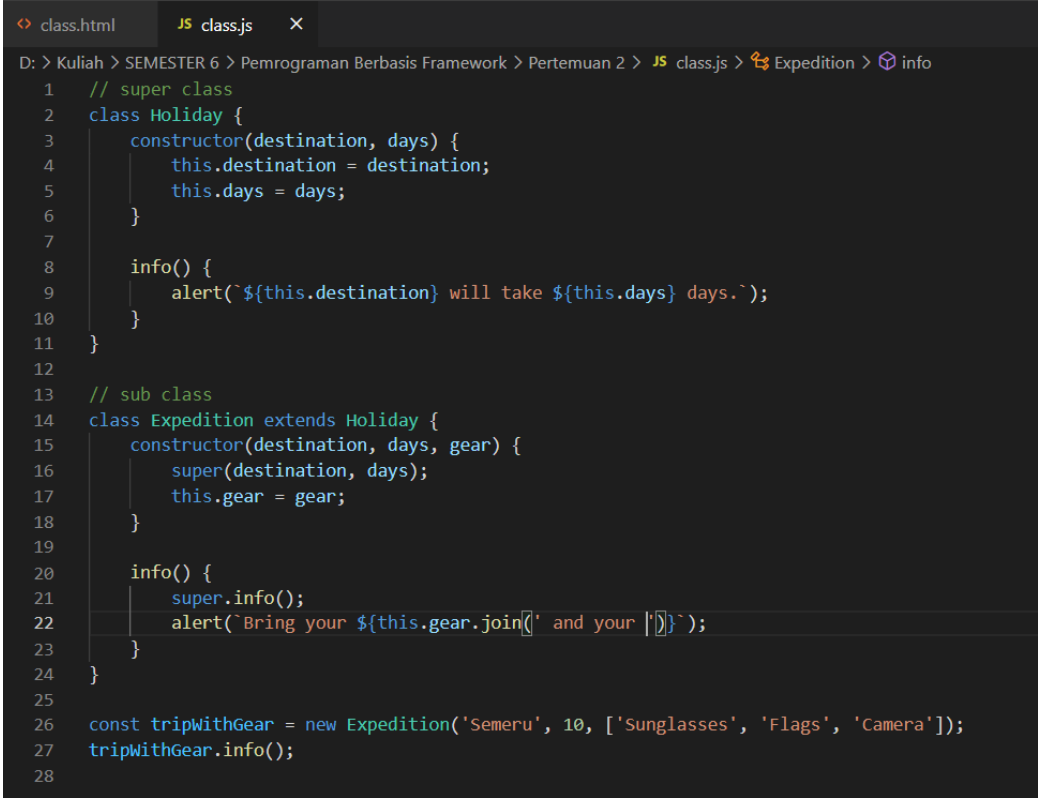
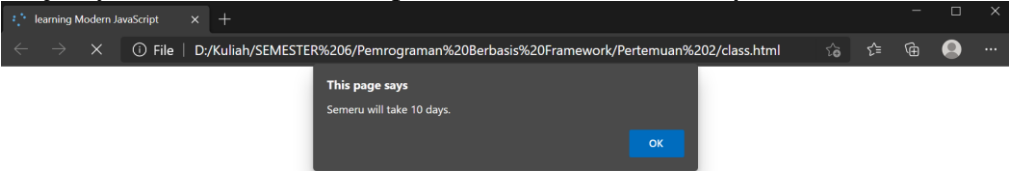
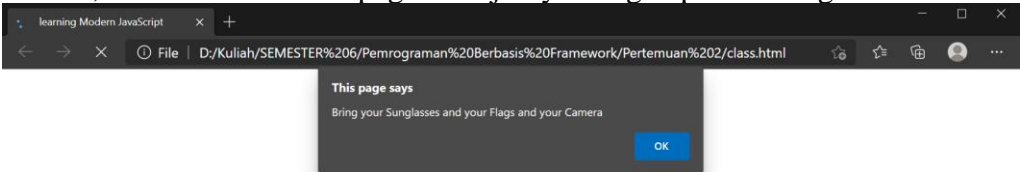
j. Membuat Spread and Rest operator

1	<p>Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>spread.html</p> <pre>spread.html X D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > spread.html > html 1 <!DOCTYPE html> 2 <html lang="en"> 3 4 <head> 5 <meta charset="UTF-8"> 6 <title>learning Modern JavaScript</title> 7 </head> 8 9 <body> 10 <script src="spread.js"></script> 11 </body> 12 13 </html></pre> <p>spread.js</p> <pre>spread.html JS spread.js X D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > JS spread.js > ... 1 var mountains = ['Semeru', 'Bromo', 'Merapi']; 2 var mountainsFromJapan = ['Fuji']; 3 4 var allMountains = [...mountains, ...mountainsFromJapan]; 5 alert(allMountains);</pre> <ul style="list-style-type: none">• Spread operator dapat digunakan untuk menggabungkan object atau array
2	<p>Selanjutnya buka file spread.html pada browser, maka hasilnya :</p> 
3	<p>Untuk melakukan praktek rest operation, buatlah dua file .html dan .js seperti dibawah ini</p> <p>restO.html</p> <pre>restO.html X D: > Kuliah > SEMESTER 6 > Pemrograman Berbasis Framework > Pertemuan 2 > restO.html > html 1 <!DOCTYPE html> 2 <html lang="en"> 3 4 <head> 5 <meta charset="UTF-8"> 6 <title>learning Modern JavaScript</title> 7 </head> 8 9 <body> 10 <script src="rest0.js"></script> 11 </body> 12 13 </html></pre>

	<p>restO.js</p>  <pre> 1 var rivers = ['Ciliwung', 'Brantas', 'Bengawan Solo']; 2 var [first, ...rest] = rivers; 3 4 alert(rest); </pre>
4	<p>Selanjutnya buka file restO.html pada browser, maka hasilnya :</p> 

k. Membuat Classes Constructor and Super

1	<p>Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>class.html</p>  <pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 4 <head> 5 <meta charset="UTF-8"> 6 <title>learning Modern JavaScript</title> 7 </head> 8 9 <body> 10 <script src="class.js"></script> 11 </body> 12 13 </html> </pre>
---	---

	<p>class.js</p>  <pre> 1 // super class 2 class Holiday { 3 constructor(destination, days) { 4 this.destination = destination; 5 this.days = days; 6 } 7 8 info() { 9 alert(`\${this.destination} will take \${this.days} days.`); 10 } 11 } 12 13 // sub class 14 class Expedition extends Holiday { 15 constructor(destination, days, gear) { 16 super(destination, days); 17 this.gear = gear; 18 } 19 20 info() { 21 super.info(); 22 alert(`Bring your \${this.gear.join(' and your ')} `); 23 } 24 } 25 26 const tripWithGear = new Expedition('Semeru', 10, ['Sunglasses', 'Flags', 'Camera']); 27 tripWithGear.info(); 28 </pre>
2	<p>Selanjutnya buka file class.html pada browser, maka hasilnya :</p> 
3	<p>Klik OK, maka akan muncul page selanjutnya dengan pesan sebagai berikut.</p> 

Link Github :
<https://github.com/liviayurike/Pemrograman-Berbasis-Framework/tree/master/Pertemuan%202>

Link Youtube :
<https://www.youtube.com/watch?v=FsplGis9QFk>