

LAPORAN PEMROGRAMAN FRAMEWORK

“SESSION 2 Modern JavaScript”

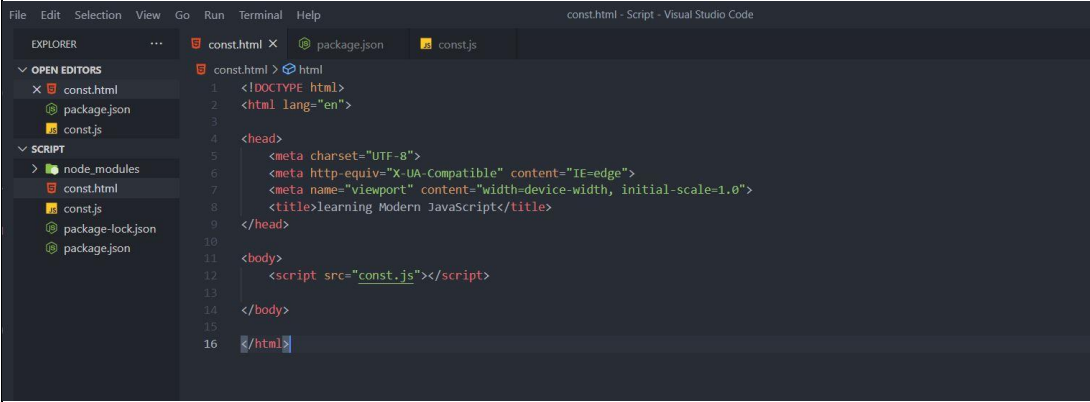

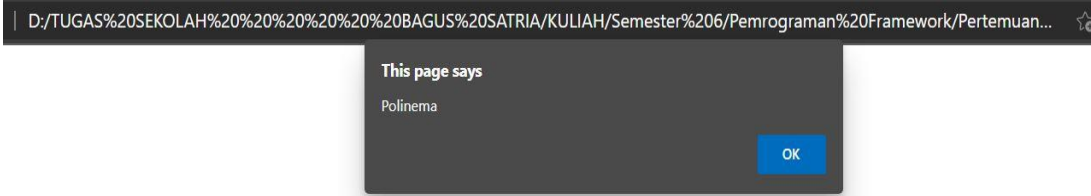


Oleh:

Nama : Bagus Satria Putra
Kelas : 3F
Absen/NIM : 8 / 1841720146

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

16 FEBRUARI 2021

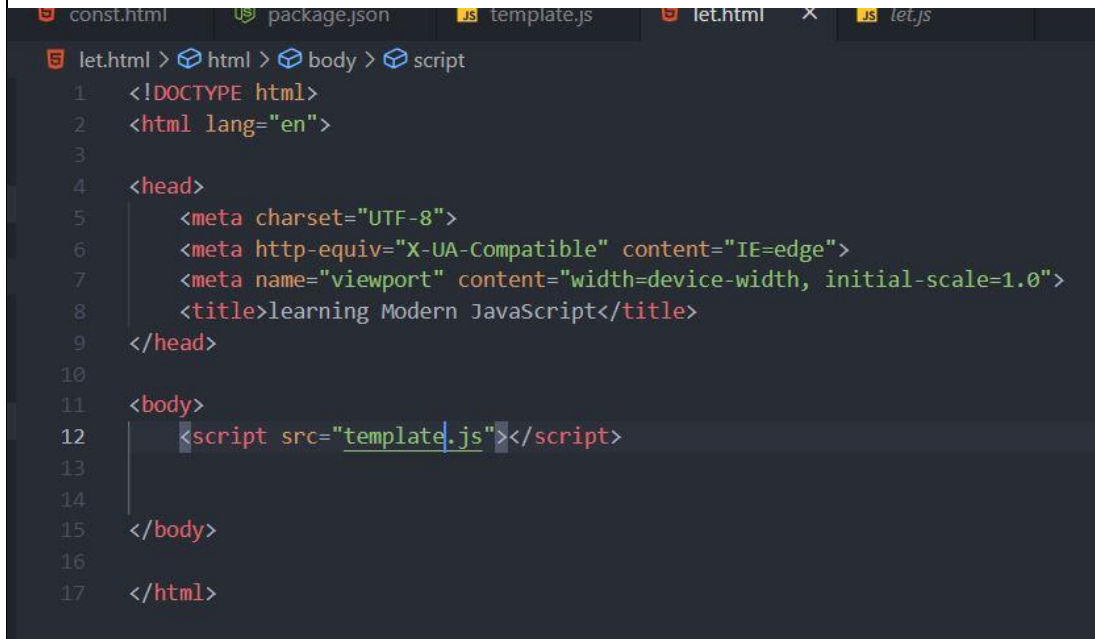
No	Keterangan
a.	<p>Membuat variabel menggunakan const</p> <ol style="list-style-type: none"> 1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya. <p>const.html Jawaban :</p>  <p>const.js Jawaban :</p>  <ol style="list-style-type: none"> 2. Selanjutnya buka file const.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini. <p>Jawaban :</p>  <p>Penjelasan : Variable const akan mengijikan untuk melakukan perubahan data sesuai dengan method yang telah disediakan oleh array maupun object dan ini disebut dengan mutable.</p>

b. Membuat variabel menggunakan let

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

let.html

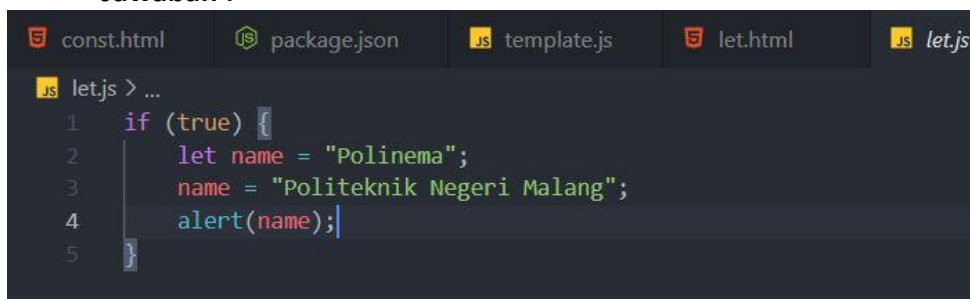
Jawaban :



```
const.html package.json template.js let.html x let.js
let.html > html > body > script
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>learning Modern JavaScript</title>
9 </head>
10
11 <body>
12   <script src="template.js"></script>
13
14
15 </body>
16
17 </html>
```

let.js

Jawaban :



```
const.html package.json template.js let.html let.js
let.js > ...
1 if (true) {
2   let name = "Polinema";
3   name = "Politeknik Negeri Malang";
4   alert(name);
5 }
```

2. Selanjutnya buka file let.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.

Jawaban :

This page says
Politeknik Negeri Malang

OK

Penjelasan :

fitur baru yang diperkenalkan oleh ES6 yang pada dasarnya pengembangan dari var.

Perbedaannya ada pada scope, apabila var cangkupannya adalah **function scope**, bila berada diluar dari fungsi akan dinyatakan sebagai global objek. Berbeda halnya dengan let yang cangkupannya adalah **block scope**.

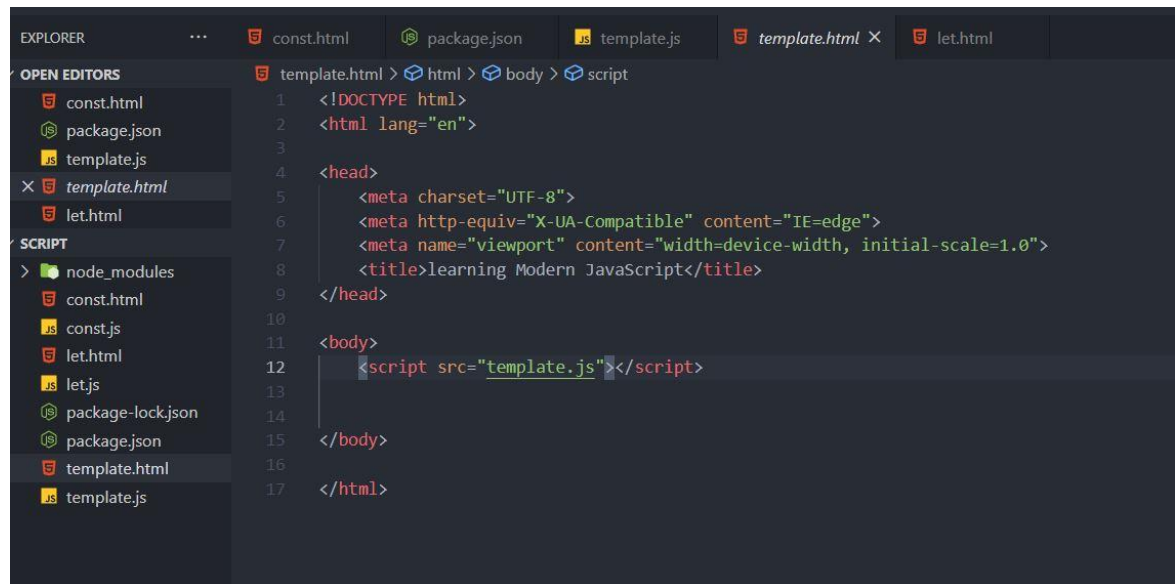
Block scope dengan menggunakan let yaitu cangkupannya terbatas pada block tempat variable let didefenisikan. Sehingga dengan konsep block scope yang dianut oleh let dapat menjadi solusi atas permasalahan yang dialami oleh var.

C. Membuat Template Strings

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

template.html

Jawaban :



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>learning Modern JavaScript</title>
9 </head>
10
11 <body>
12   <script src="template.js"></script>
13
14
15 </body>
16
17 </html>
```

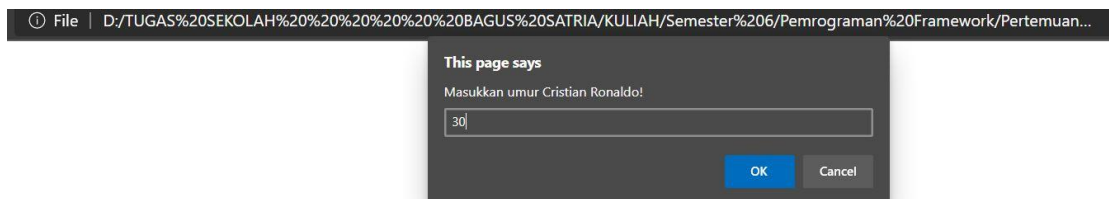
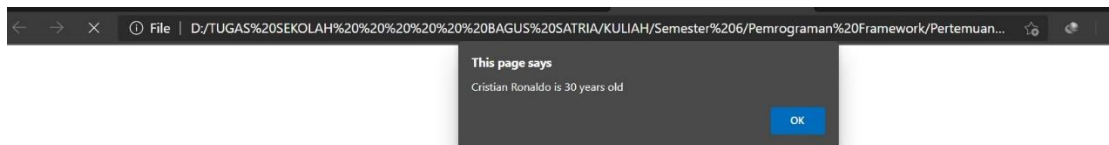
template.js

Jawaban :

```
const.html package.json template.js X template.html let.html
template.js > result
1 let fname = "Cristian";
2 let lname = "Ronaldo";
3 let age = prompt("Masukkan umur Cristian Ronaldo!");
4
5 let result = `${fname} ${lname} is ${age} years old`;
6 alert(result);
```

2.Selanjutnya buka file template.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.

Jawaban :



Penjelasan :

menggabungkan teks dan variable dengan tampilan yang lebih rapi.
Prinsip penggunaan template literal adalah

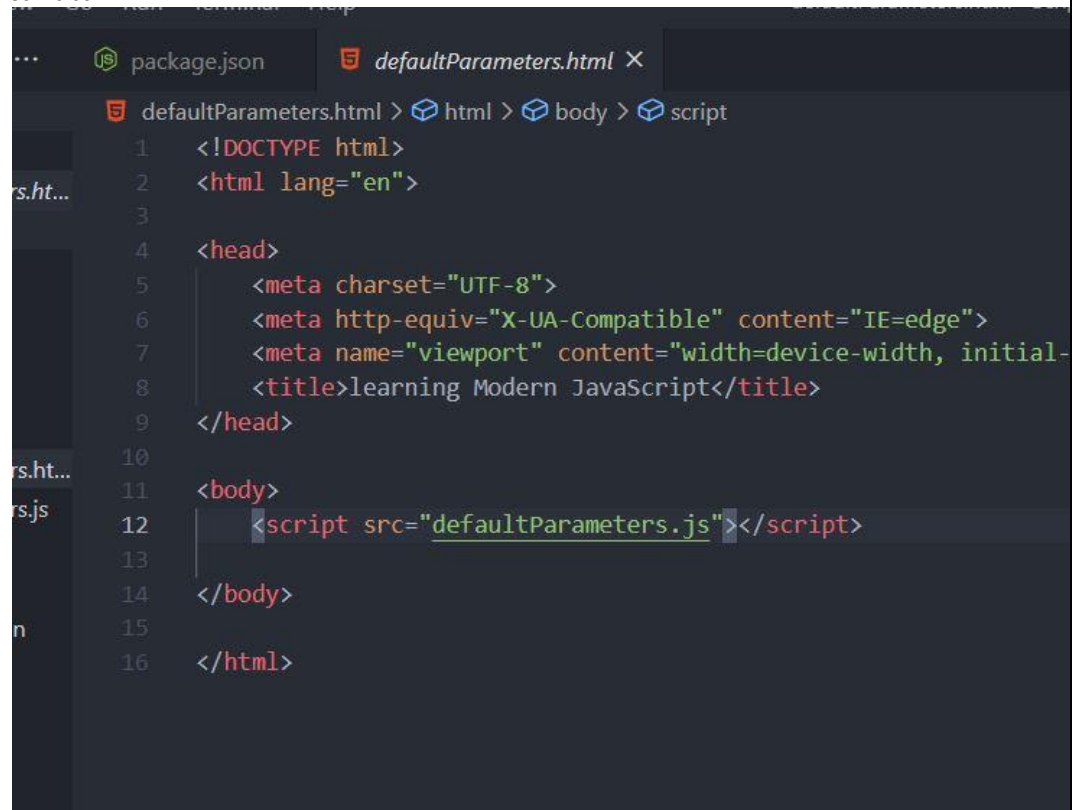
1. Dimulai dan diakhir dengan `
2. Penggunaan `\${}` untuk menuliskan suatu variable atau javascript statement lainnya

d. Membuat default parameters

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

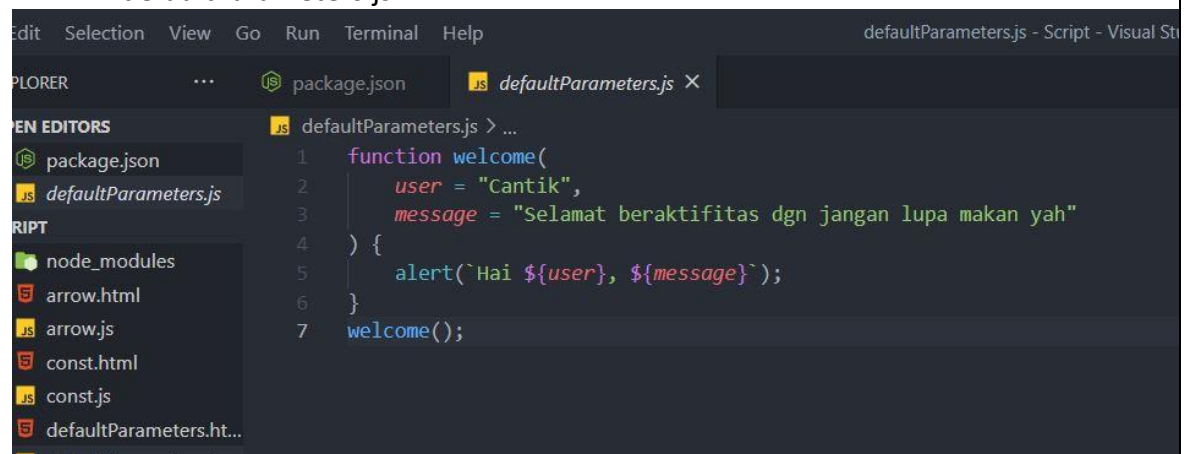
defaultParameters.html

Jawaban :



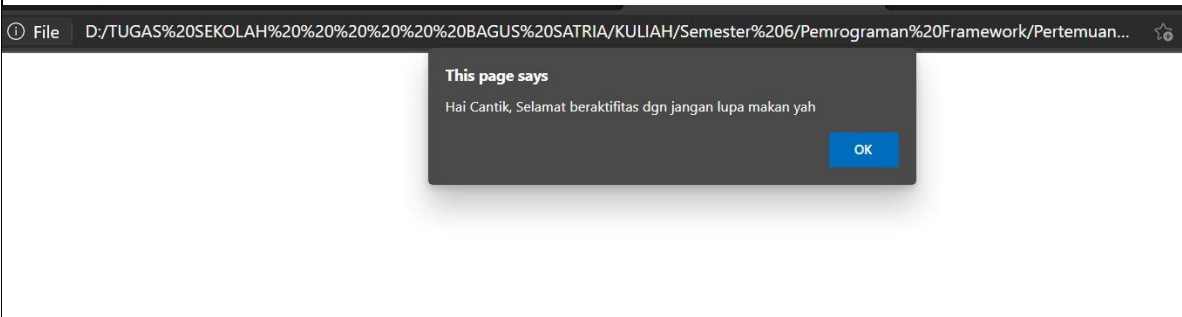
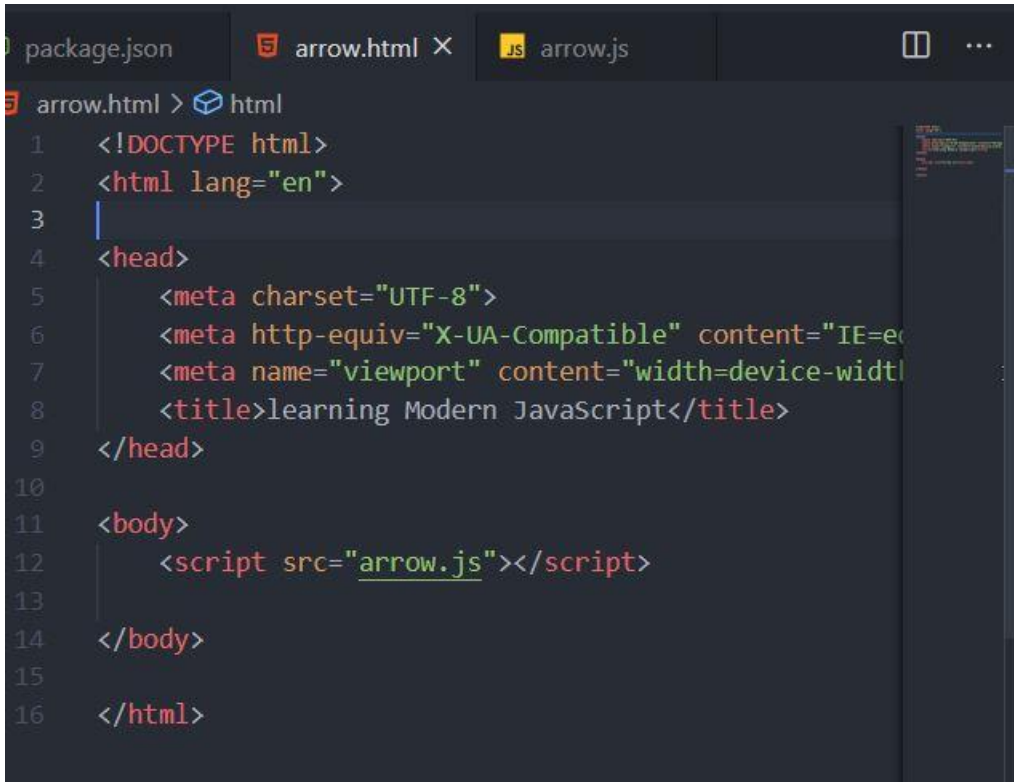
```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-
8   <title>learning Modern JavaScript</title>
9 </head>
10
11 <body>
12   <script src="defaultParameters.js"></script>
13
14 </body>
15
16 </html>
```

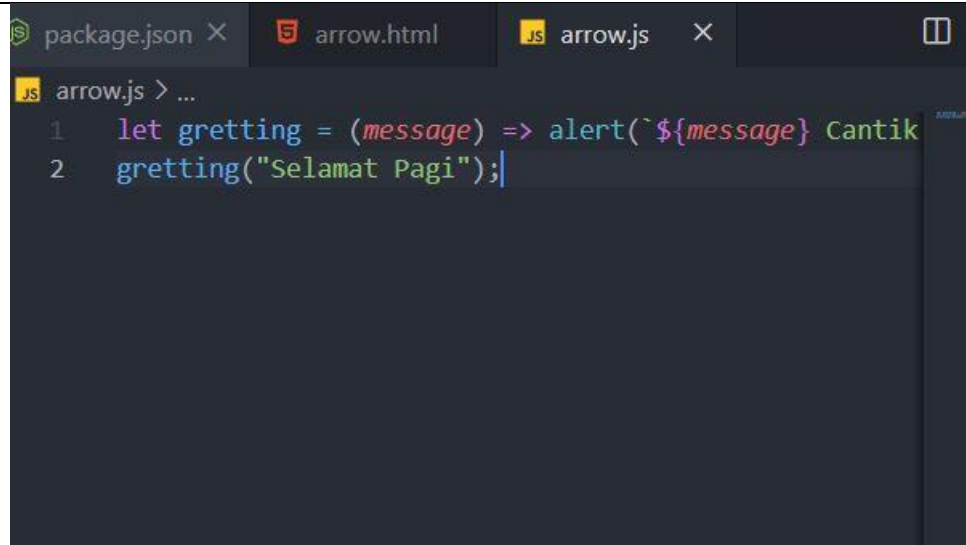
defaultParameters.js



```
1 function welcome(
2   user = "Cantik",
3   message = "Selamat beraktifitas dgn jangan lupa makan yah"
4 ) {
5   alert(`Hai ${user}, ${message}`);
6 }
7 welcome();
```

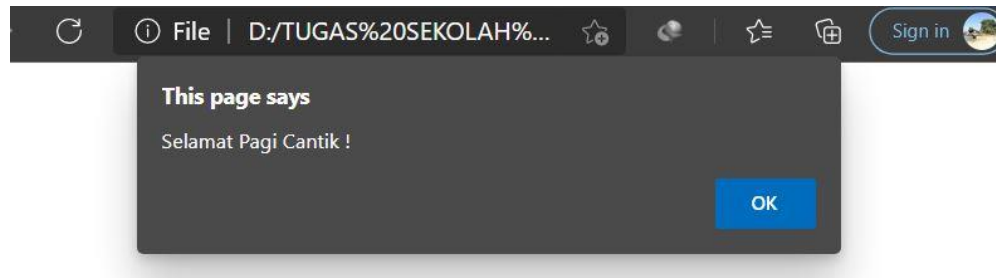
2. Selanjutnya buka file defaultParameters.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.

	 <p>Penjelasan :</p> <p>Default parameter merupakan fitur yang bisa dimanfaatkan untuk membuat fungsi kita lebih fleksibel, karena cara pemanggilan fungsi dapat dirancang dengan lebih sederhana. Untuk merancang fungsi yang lebih rumit, kita bisa membuat jumlah argumen yang tidak terbatas.</p>
e.	<p>Membuat Arrow Function 1</p> <ol style="list-style-type: none"> 1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya. <p>arrow.html</p>  <pre> 1 <!DOCTYPE html> 2 <html lang="en"> 3 4 <head> 5 <meta charset="UTF-8"> 6 <meta http-equiv="X-UA-Compatible" content="IE=edge"> 7 <meta name="viewport" content="width=device-width, initial-scale=1"> 8 <title>learning Modern JavaScript</title> 9 </head> 10 11 <body> 12 <script src="arrow.js"></script> 13 14 </body> 15 16 </html> </pre> <p>arrow.js</p>



```
package.json X arrow.html JS arrow.js X  
JS arrow.js > ...  
1 let gretting = (message) => alert(`${message} Cantik  
2 gretting("Selamat Pagi");
```

2. Selanjutnya buka file arrow.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



Penjelasan :

sintak penulisan fungsi yang bisa dibilang lebih singkat, menggunakan token baru yaitu "=>", fungsi yang dideklarasikan menggunakan panah ini bersifat anonim. Mirip lambda arrow operator di bahasa lain. Dengan arrow function kita bisa menulis fungsi tanpa menggunakan kata kunci "function", "return", dan kurung kurawal. Oke mari kita langsung saja membahas sedikit.

F. Membuat Arrow Function 2

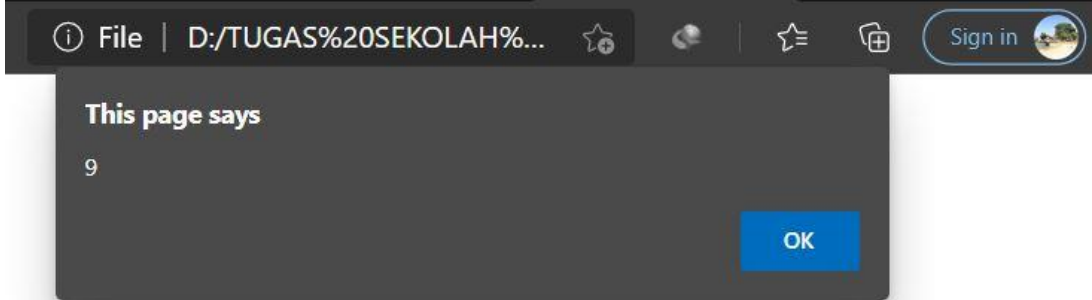
3. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.
arrowf.html

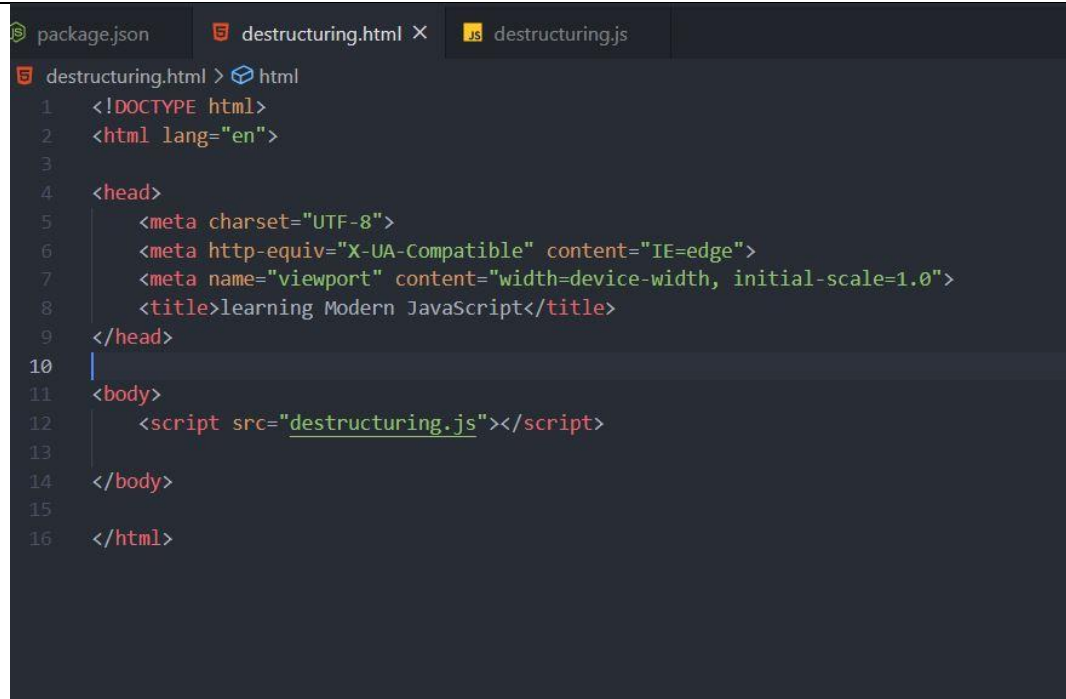

```
package.json x arrowf.html x JS arrowf.js
arrowf.html > html > body > script
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>learning Modern JavaScript</title>
9 </head>
10
11 <body>
12   <script src="arrowf.js"></script>
13
14 </body>
15
16 </html>
```

arrowf.js

```
package.json arrowf.html JS arrowf.js x
JS arrowf.js > ...
1 const func = (a, b) => {
2   return a + b;
3 };
4 alert(func(5, 4));
```

4. Selanjutnya buka file arrowf.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.

	 <p>Penjelasan :</p> <p>jika parameter yang dibutuhkan hanya 1 maka kita bisa meringkasnya kembali dengan menghilangkan tanda kurung pada baris kode tersebut.</p> <p>namun jika kita tidak memiliki parameter yang dibutuhkan maka kita teap harus menggunakan tanda kurung agar block kodenya dapat berjalan dengan benar.</p>
G.	<p>Membuat Destructuring Object</p> <ol style="list-style-type: none"> 1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya. destructuring.html



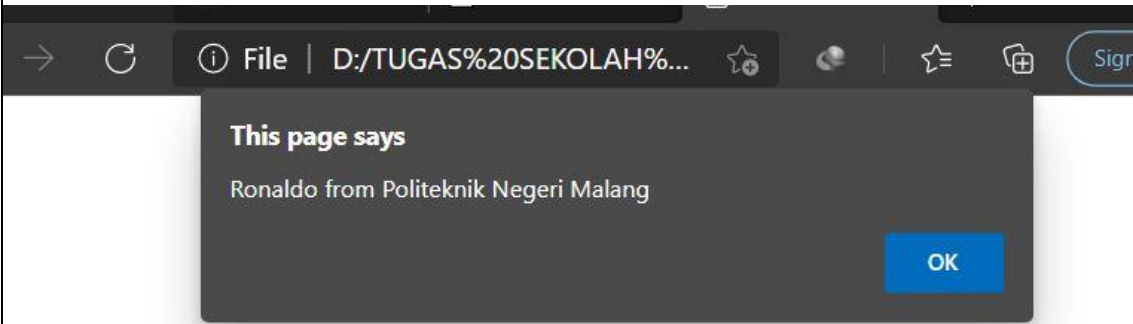
```
package.json  destructuring.html X  destructuring.js
destructuring.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta http-equiv="X-UA-Compatible" content="IE=edge">
7    <meta name="viewport" content="width=device-width, initial-scale=1.0">
8    <title>learning Modern JavaScript</title>
9  </head>
10
11 <body>
12   <script src="destructuring.js"></script>
13
14 </body>
15
16 </html>
```

destructuring.js



```
package.json  destructuring.html  destructuring.js X
destructuring.js > ...
1  let polStudent = ({ name, polytechnic }) => {
2    alert(`${name} from ${polytechnic}`);
3  };
4
5  polStudent({
6    name: "Ronaldo",
7    polytechnic: "Politeknik Negeri Malang",
8  });
```

2. Selanjutnya buka file destructuring.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.

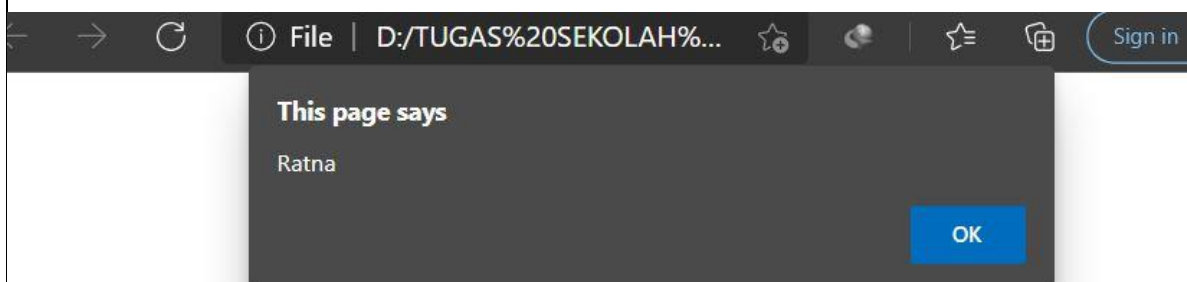
	 <p>Penjelasan :</p> <p>'membongkar' isi array dan objek untuk disimpan pada beberapa variabel. Dengan menggunakan fitur ini, kita dapat menghemat kode secara signifikan. Dalam <i>object destructuring</i>, nama key pada sisi kiri pernyataan harus <i>match</i> dengan nama <i>key</i> pada objek sumber</p>
H.	<p>Membuat Destructuring an Array</p> <ol style="list-style-type: none"> 1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya. array.html

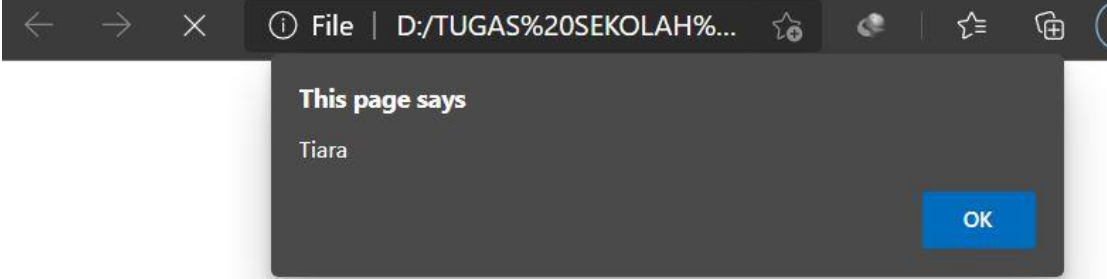
```
package.json array.html X array.js
array.html > html > head
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>learning Modern JavaScript</title>
9 </head>
10
11 <body>
12   <script src="array.js"></script>
13
14 </body>
15
16 </html>
```

array.js

```
package.json array.html array.js X
array.js > ...
1 let [wife] = ["Ratna", "Bunga", "Tiara"];
2 alert(wife);
3
4 // let [, , wife] = ["Ratna", "Bunga", "Tiara"];
5 // alert(wife);
```

2. Selanjutnya buka file array.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



	<p>3. Kemudian ganti sedikit code pada array.js seperti dibawah ini</p> <pre>1 let [, , wife] = ['Ratna', 'Bunga', 'Tiara']; 2 alert(wife);</pre> <p>4. Bukalah kembali, maka hasilnya akan berbeda seperti gambar dibawah ini</p>  <p>Penjelasan : dalam Javascript akan mengembalikan <code>undefined</code> ketika kita mencoba mengakses nilai di luar batasnya (<i>index out of bounds</i>). Konsekuensinya, jika panjang array yang kita deklarasikan melebihi panjang array sumbernya, kita dapat temukan variabel yang kita <i>assign</i> memiliki nilai <code>undefined</code>. Untuk menanggulangnya, kita dapat memberikan nilai <i>default</i> bagi variabel yang akan di-<i>assign</i> melalui <i>destructuring</i>. Seperti <i>array destructuring</i>, kita juga dapat melakukan <i>destructuring</i> terhadap objek tanpa melakukan deklarasi pada baris yang sama. Perbedaannya adalah kita harus membungkusnya dengan sepasang tanda kurung agar pernyataan pada sisi kiri (<code>{ ..., ...}</code>) tidak dianggap sebagai blok;</p>
i.	<p>Membuat Restructuring</p> <ol style="list-style-type: none"> 1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.

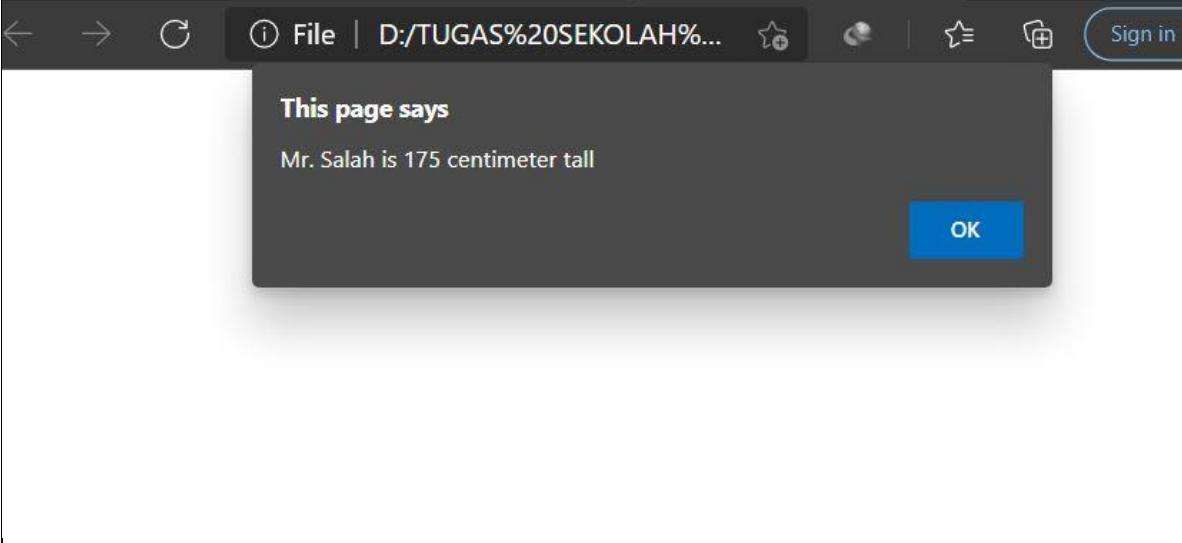
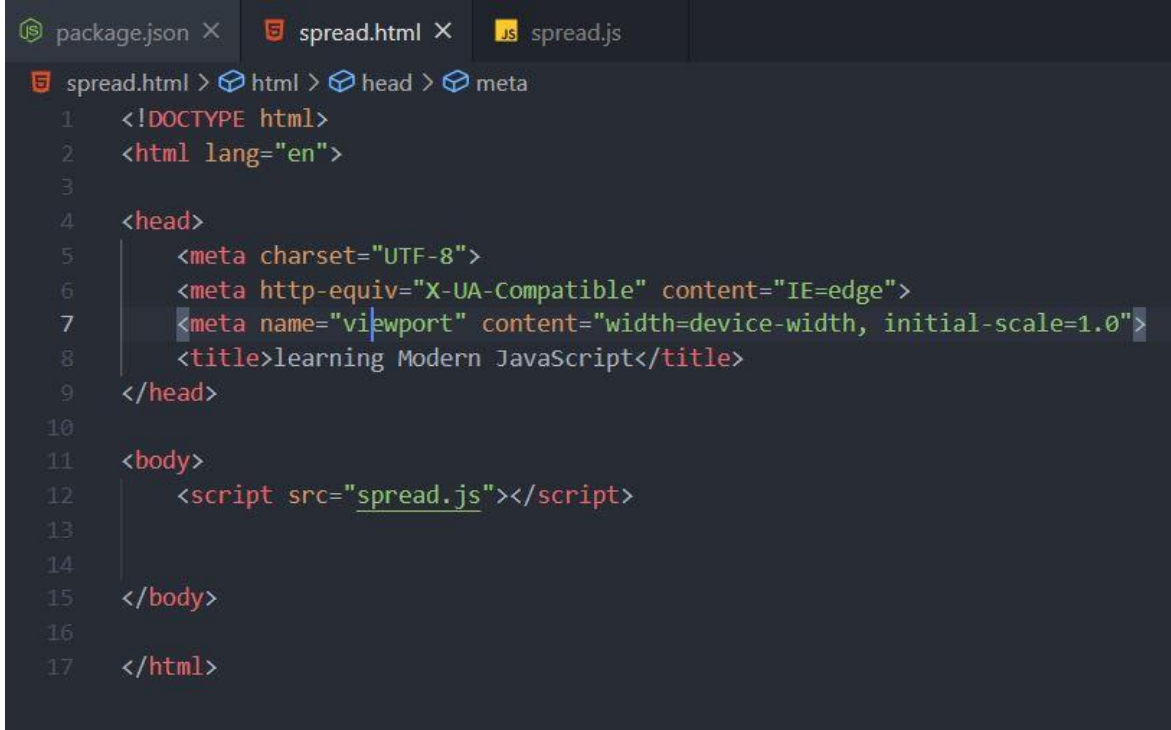
restructuring.html

```
package.json  restructuring.html ×  JS restructuring.js
restructuring.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>learning Modern JavaScript</title>
9  </head>
10
11 <body>
12     <script src="restructuring.js"></script>
13
14 </body>
15
16 </html>
```

restructuring.js

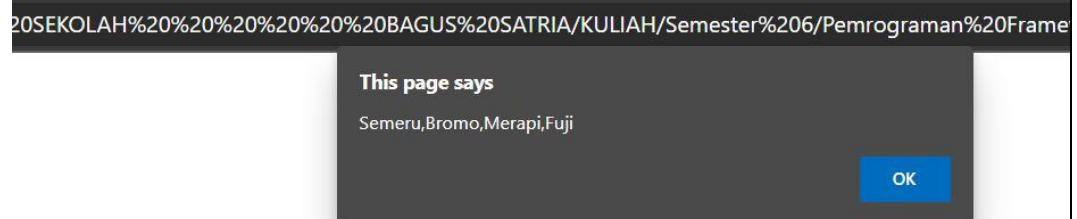
```
package.json  restructuring.html  JS restructuring.js ×
JS restructuring.js > pemainSepakbola
1  var pemainSepakbola = {
2      name: "Salah",
3      height: "175",
4      output() {
5          alert(`Mr. ${this.name} is ${this.height} centimeter tall`);
6      },
7  };
8  pemainSepakbola.output();
```

2.Selanjutnya buka file restructuring.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.

	 <p>Penjelasan :</p> <p>fungsi yang memanggil dirinya sendiri. Fungsi restructuring sering saya bayangkan seperti perulangan. Karena tingkah lakunya yang mengulang-ulang setiap pemanggilan dirinya</p>
J.	<p>Membuat Spread and Rest operator</p> <p>1. Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>spread.html</p>  <p>spread.js</p>


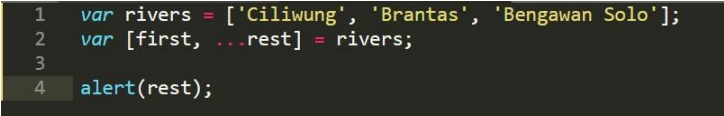
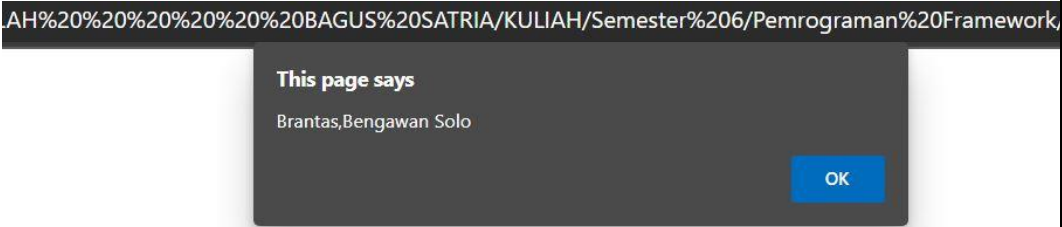

```
package.json  spread.html  JS spread.js X
JS spread.js > ...
1  var mountains = ["Semeru", "Bromo", "Merapi"];
2  var mountainsFromJapan = ["Fuji"];
3
4  var allMountains = [...mountains, ...mountainsFromJapan];
5  alert(allMountains);
```

2.Selanjutnya buka file spread.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.



3.Untuk melakukan praktek rest operation, buatlah dua file .html dan .js seperti dibawah ini
restO.html

```
package.json X  restO.html X  JS restO.js  JS class.js
restO.html > html > head > meta
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5  |   <meta charset="UTF-8">
6  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8  |   <title>learning Modern JavaScript</title>
9  | </head>
10
11 <body>
12 |   <script src="restO.js"></script>
13 |
14 </body>
15
16 </html>
```

	<p>restO.js</p>  <pre> 1 var rivers = ["Ciliwung", "Brantas", "Bengawan Solo"]; 2 var [first, ...rest] = rivers; 3 alert(rest); </pre> <p>4.Selanjutnya buka file restO.html pada browser anda masing-masing, maka</p>  <pre> 1 var rivers = ['Ciliwung', 'Brantas', 'Bengawan Solo']; 2 var [first, ...rest] = rivers; 3 4 alert(rest); </pre> <p>hasilnya akan seperti dibawah ini.</p>  <p>Penjelasan :</p> <p><i>Rest Parameter</i> ini berguna untuk menggabungkan semua paramater pada <i>function</i> ke dalam array. Dengan menggunakan <i>Rest Parameter</i> ini dapat membantu kita mendefinisikan <i>function</i> dengan rapi serta memberikan parameter yang tidak terbatas pada sebuah <i>function</i>. <i>Spread Operator</i> digunakan untuk membagi elemen array atau properti pada objek, sehingga elemen array dapat ditambahkan/dimasukan ke dalam array baru.</p>
K.	<p>Membuat Classes Constructor and Super</p> <p>1.Buatlah halaman file .html untuk menampilkan hasilnya, dan file .js untuk menuliskan code JavaScript-nya.</p> <p>class.html</p>

package.json × class.js class.html ×

class.html > html > head > meta

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>learning Modern JavaScript</title>
9 </head>
10
11 <body>
12   <script src="class.js"></script>
13
14 </body>
15
16 </html>
```

class.js

```

package.json × JS class.js × class.html
JS class.js > Expedition > info
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 }
10            will take $ { this.days }
11            days.`);
12     }
13 }
14
15 //sub class
16 class Expedition extends Holiday {
17     constructor(destination, days, gear) {
18         super(destination, days);
19         this.gear = gear;
20     }
21
22     info() {
23         super.info();
24         alert(`Bring your $ { this.gear.join(' and your ') }`);
25     }
26 }
27
28 const tripWithGear = new Expedition("Semeru", 10, [
29     "Sunglasses",
30     "Flags",
31     "Camera",
32 ]);
33 tripWithGear.info();

```

2.Selanjutnya buka file class.html pada browser anda masing-masing, maka hasilnya akan seperti dibawah ini.

File | D:/TUGAS%20SEKOLAH%20%20%20%20%20%20BAGUS%20SATRIA/KULIAH/Semester%206/Pemrograman%20Framev

This page says

Semeru will take 10 days.

OK

3. Klik OK, maka akan muncul page selanjutnya dengan pesan sebagai berikut.

D:/TUGAS%20SEKOLAH%20%20%20%20%20%20BAGUS%20SATRIA/KULIAH/Semester%206/Pemrograman%20Framework/Pertemuan..

This page says

Bring your Sunglassesand yourFlagsand yourCamera

OK

Penjelasan :

Hanya boleh ada satu metode spesial dengan nama "constructor" dalam sebuah kelas. `SyntaxError` akan dilemparkan jika suatu kelas memiliki lebih dari satu metode konstruktor.

Konstruktor dapat menggunakan kata kunci `super` untuk memanggil konstruktor orang-tua.

Jika metode konstruktor tidak didefinisikan dalam sebuah kelas, maka konstruktor asali yang akan digunakan.

--Alhamdulillah--